TOSHIBA

32 Bit RISC Microcontroller
TX00 Series

TMPM061FWFG

TOSHIBA ELECTRONIC DEVICES & STORAGE CORPORATION

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

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

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

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

a. SFR table of each IP as an example

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

Base Address = 0x0000_0000

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

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

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

b. SFR(register)

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

1.2.2 SAMCR(Control register)

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

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

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

R / W	READ WRITE
R	READ
W	WRITE

c. Data descriptopn

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

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

d. Register descriptoption

Registers are described as shown below.

- Register name <Bit Symbol>

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

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

- Register name [Bit]

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

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

General precautions on the use of Toshiba MCUs

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

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

1. The MCUs' operation at power-on

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

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

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

2. Unused pins

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

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

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

3. Clock oscillation stability

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

Revision History

Date	Revision	Comment
2012/2/4	Tentative 1	First Release of Tentative
2012/2/8	Tentative 2	Contents Revised
2012/6/26	Tentative 3	Contents Reviced
2013/8/2	1	First Release
2014/3/10	2	Contents Revised
2015/10/2	3	Contents Revised
2018/10/26	4	Contents Revised

Table of Contents

5. Reset Operation

6. Clock/Mode control

7. Exceptions

8. Input / Output port

9. 16-bit Timer / Event Counters (TMRB)

12. Serial Bus Interface (I2C/SIO)

13. 10-bit Analog/Digital Converter (ADC)

14. 24-bit ΔΣ Analog/Digital Converter (DSADC)

15. Temperature Sensor (TEMP)

16. Real Time Clock (RTC)

18. Low Voltage Detection Circuit (LVD)

19. Watchdog Timer (WDT)

20. Flash Memory Operation

21. Debug Interface

22. Electrical Characteristics

23. Port Section Equivalent Circuit Schematic

24. Package Dimensions

CMOS 32-Bit Microcontroller

TMPM061FWFG

The TMPM061FWFG is a 32-bit RISC microprocessor series with an Arm®Cortex®-M0 microprocessor core.

Features of the TMPM061FWFG are as follows:

1.1 Features

1. Arm Cortex-M0 microprocessor core
   a. Improved code efficiency has been realized through the use of Thumb® −2 instruction.
   b. 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.

      [Low power consumption]

      ・ Optimized design using a low power consumption library

      ・ Standby function that stops the operation of the micro controller core
   c. High-speed interrupt response suitable for real-time control

      ・ An interruptible long instruction.

      ・ Stack push automatically handled by hardware.

2. Endian: Little endian

3. On Chip program memory and data memory
   - On chip Flash ROM: 128 Kbyte
   - On chip RAM: 8 Kbyte

4. Power calculation engine (PCE) : DSP function for power calculation

5. 16-bit timer (TMRB): 2 channels
   ・ 16-bit interval timer mode
   ・ 16-bit event counter mode
   ・ 16-bit PPG output
   ・ Input capture function
   ・ Synchronous mode

6. 16-bit timer (TMR16A): 7 channels

7. General-purpose serial interface (SIO/UART): 4 channels

   Either UART mode or synchronous mode can be selected

   * Port selection is possible (1 channel)

   * IR carrier pulse output is possible (3 channels)

8. Serial bus interface (I2C/SIO): 1channels

   Either I2C bus mode or synchronous mode can be selected.

9. 10-bit AD converter (ADC): 1 unit
   - Fixed channel/scan mode
   - Single/repeat mode
   - AD monitoring 2ch
   - Conversion speed

     16.2μsec (AVDD = 2.7 to 3.6V)

     32.4μsec (AVDD = 1.8 to 3.6V)

10. 24-bit ΔΣAD converter (ADC): 3 units
    - Sampling frequency: 3 KHz, 6 KHz
    - Input voltage range: -0.375 to +0.375 V
    - Programmable gain amp: ×1, ×2, ×4, ×8 or ×16 can be selected.
    - Conversion mode: Single or repeat
    - Adjustment of conversion start
    - Synchronous start of multiple units

11. Temperature sensor (TEMP)

12. Real time clock (RTC): 1 channel
    - Clock (hour, minute and second)
    - Calendar (month, week, date and leap year)
    - Clock adjustment (by software)

13. LCD driver/controller (LCDD)
    - LCD direct drive is possible (40 seg × 4 com)
    - 1/4, 1/3, 1/2 duties or static drive are selectable
    - Bleeder resistance incorporated (external bleeder resistance is also usable)

14. Voltage detection circuit (LVD)

15. Watchdog timer (WDT): 1 channel

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

16. Interrupt source: The order of priority can be set to 4 levels.
    - Internal: 28 factors
    - External: 4 factors

17. Input/output ports (PORT): 64 pins

    Three 5V-Tolerant inputs are prepared

18. Standby mode

    Standby modes: IDLE, SLOW, SLEEP, STOP

19. Clock generator (CG)

    Clock gear function: The high-speed clock can be divided into 1/1, 1/2, 1/4, 1/8 or 1/16.

20. Maximum operating frequency: 16 MHz

21. Operating voltage range: 1.8 to 3.6 V

    2.9 to 3.6 V: Use of 24-bit ΔΣAD converter is possible

2.2 to 3.6 V: Use of LCD driver/controller is possible

2.7 to 3.6 V: Write/erase of Flash ROM is possible

22. Temperature range

- -40 to 85 degrees (except during Flash writing/erasing)
- 0 to 70 degrees (during Flash writing/erasing)

23. Package

LQFP100 (14mm × 14mm, 0.5mm pitch)

1.2 Block Diagram

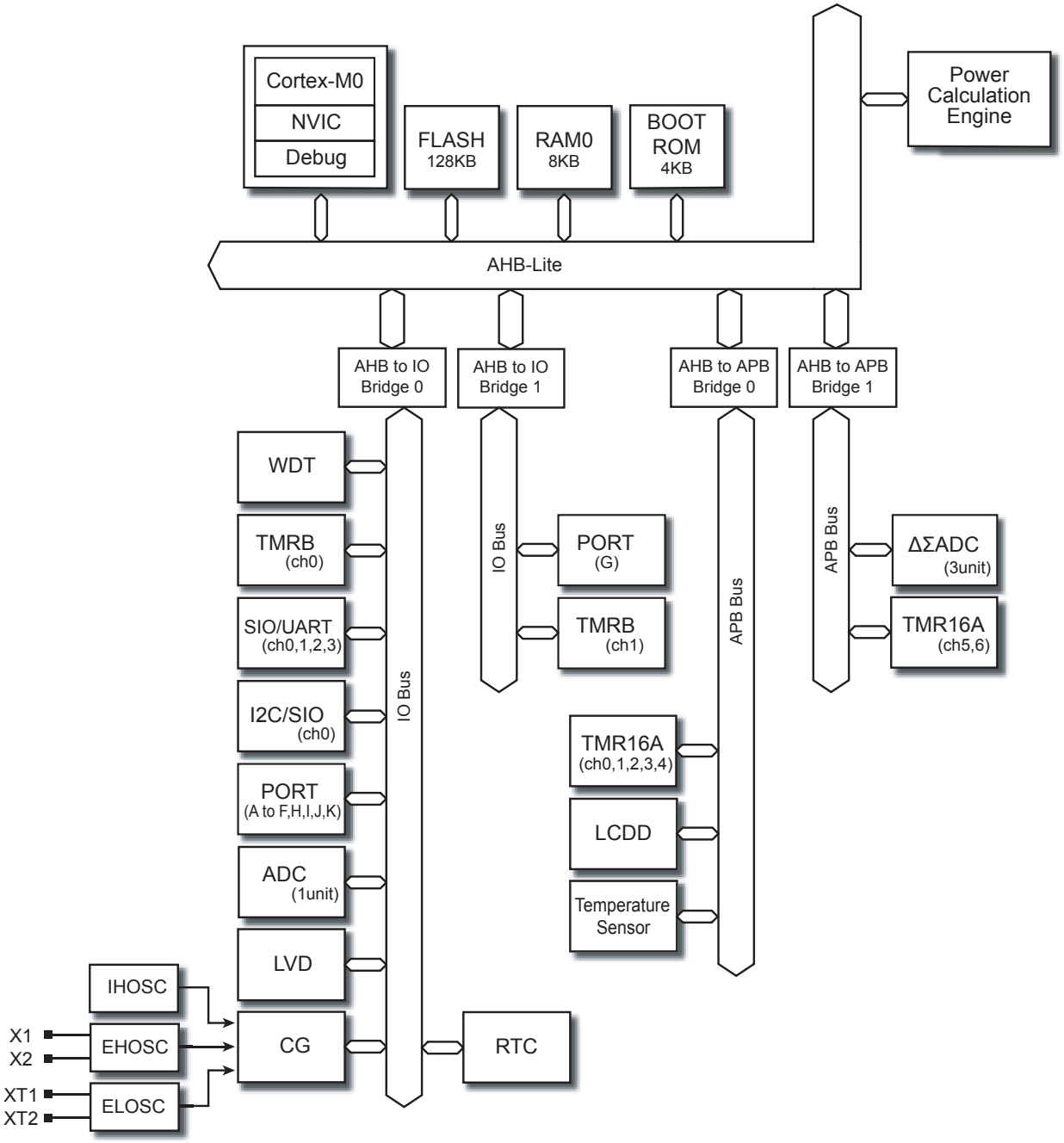


Figure 1-1 Block Diagram

1.3 Pin Layout (Top view)

Figure 1-2 shows the pin layout of TMPM061FWFG.

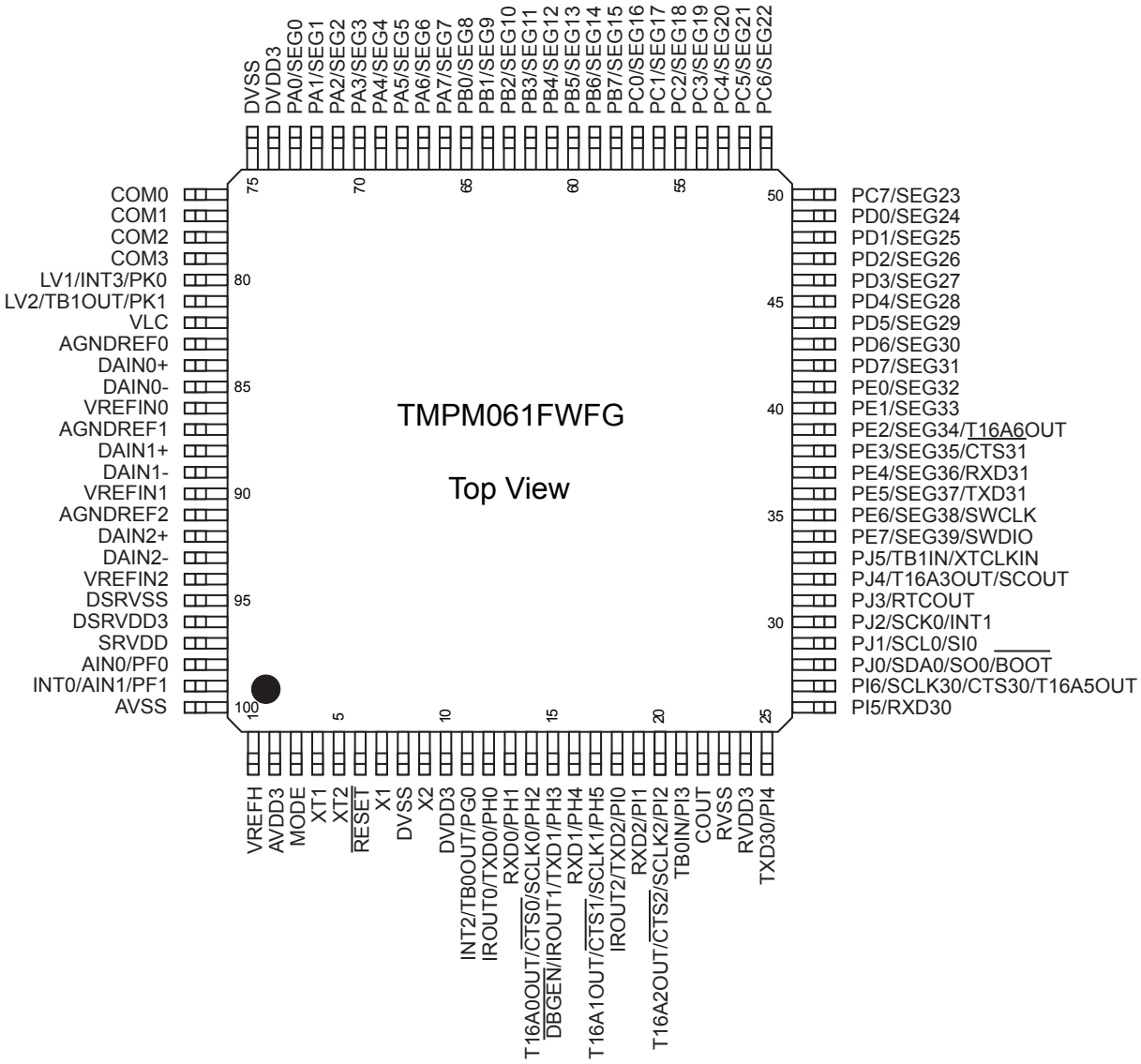


Figure 1-2 Pin Layout

1.4 Pin names and Functions

The input/output pin names and functions of the TMPM061FWFG are as follows:

Table 1-1 Pin Names and Functions Sorted by Pin (1/6)

Type	Pin No.	Pin Name	Input/ Out-put	Function
PS	1	VREFH	−	Supplying the 10bit AD converter with a reference power supply. (note 1)
PS	2	AVDD3	−	Supplying the 10bit AD converter with a power supply. (note 1) (note 3)
Control	3	MODE	I	MODE pin (note) MODE pin must be connected to GND.
Clock	4	XT1	I	Connected to a low-speed oscillator.
Clock	5	XT2	O	Connected to a low-speed oscillator.
Control	6	$\overline{\text{RESET}}$	I	Reset input pin (note) With a pull-up and a noise filter
Clock	7	X1	I	Connected to a high-speed oscillator. High-speed clock input pin.
PS	8	DVSS	−	GND pin
Clock	9	X2	O	Connected to a high-speed oscillator.
PS	10	DVDD3	−	Power supply pin (note 3)
Function	11	PG0 TB0OUT INT2	I/O O I	I/O port Timer B output External interrupt pin
Function	12	PH0 TXD0 IROUT0	I/O O O	I/O port Sending serial data Sending serial data with carrier pulse
Function	13	PH1 RXD0	I/O I	I/O port Receiving serial data
Function	14	PH2 SCLK0 $\overline{\text{CTS0}}$ T16A0OUT	I/O I/O I O	I/O port Serial clock input/ output Handshake input pin Timer 16A output
Function	15	PH3 TXD1 IROUT1 $\overline{\text{DBGEN}}$	I/O O O I	I/O port Sending serial data Sending serial data with carrier pulse Debug enable (note) This pin is fixed to debug function by sampling "Low" at the rise of $\overline{\text{RESET}}$ signal.
Function	16	PH4 RXD1	I/O I/O	I/O port Receiving serial data
Function	17	PH5 SCLK1 $\overline{\text{CTS1}}$ T16A1OUT	I/O I/O I O	I/O port Serial clock input/ output Handshake input pin Timer 16A output

Table 1-1 Pin Names and Functions Sorted by Pin (2/6)

Type	Pin No.	Pin Name	Input/ Out-put	Function
Function	18	PI0 TXD2 IROUT2	I/O O O	I/O port Sending serial data Sending serial data with carrier pulse (note) While $\overline{\text{RESET}}$ pin is "Low", keep PI0 pin from being set to "Low".
Function	19	PI1 RXD2	I/O I	I/O port Receiving serial data
Function	20	PI2 SCLK2 $\overline{\text{CTS2}}$ T16A2OUT	I/O I/O I O	I/O port Serial clock input/ output Handshake input pin Timer 16A output
Function	21	PI3 TB0IN	I/O I	I/O port Inputting the timer B capture trigger
PS	22	COUT	−	Regulator output pin Connect a 1μF capacitor to between COUT and RVSS.
PS	23	RVSS	−	GND pin
PS	24	RVDD3	−	Power supply pin (note 3)
Function	25	PI4 TXD30	I/O O	I/O port Sending serial data
Function	26	PI5 RXD30	I/O I	I/O port (5V tolerant input) (note 7) Receiving serial data
Function	27	PI6 SCLK30 $\overline{\text{CTS30}}$ T16A5OUT	I/O I/O I O	I/O port (5V tolerant input) (note 7) Serial clock input/ output Handshake input pin Timer 16A output
Function	28	PJ0 SDA0 SO0 $\overline{\text{BOOT}}$	I/O I/O I/O I	I/O port I2C mode: data pin SIO mode: data pin Setting a single boot mode: (note) TMPM061FWFG goes into single boot mode by sampling "Low" at the rise of a $\overline{\text{RESET}}$ signal.
Function	29	PJ1 SCL0 SI0	I/O I/O I/O	I/O port I2C mode: clock pin SIO mode: data pin
Function	30	PJ2 SCK0 INT1	I/O I/O I	I/O port (5V tolerant input) (note 7) Inputting and outputting a clock if the serial bus interface operates in the SIO mode. External interrupt pin
Function	31	PJ3 RTCOUT	I/O O	I/O port RTC output
Function	32	PJ4 T16A3OUT SCOUT	I/O O O	I/O port Timer 16A output System clock output
Function	33	PJ5 TB1IN XTCLKIN	I I I	Input port Inputting the timer B capture trigger Low-speed clock input pin

Table 1-1 Pin Names and Functions Sorted by Pin (3/6)

Type	Pin No.	Pin Name	Input/ Out-put	Function
Function / Debug	34	PE7 SEG39 SWDIO	I/O O I/O	I/O port LCD segment output pin Debug pin
Function / Debug	35	PE6 SEG38 SWCLK	I/O O I	I/O port LCD segment output pin Debug pin
Function	36	PE5 SEG37 TXD31	I/O O O	I/O port LCD segment output pin Sending serial data
Function	37	PE4 SEG36 RXD31	I/O O I	I/O port LCD segment output pin Receiving serial data
Function	38	PE3 SEG35 SCLK31 $\overline{\text{CTS31}}$	I/O O I/O I	I/O port LCD segment output pin Handshake input pin Serial clock input/ output
Function	39	PE2 SEG34 T16A6OUT	I/O O O	I/O port LCD segment output pin Timer 16A output
Function	40	PE1 SEG33	I/O O	I/O port LCD segment output pin
Function	41	PE0 SEG32	I/O O	I/O port LCD segment output pin
Function	42	PD7 SEG31	I/O O	I/O port LCD segment output pin
Function	43	PD6 SEG30	I/O O	I/O port LCD segment output pin
Function	44	PD5 SEG29	I/O O	I/O port LCD segment output pin
Function	45	PD4 SEG28	I/O O	I/O port LCD segment output pin
Function	46	PD3 SEG27	I/O O	I/O port LCD segment output pin
Function	47	PD2 SEG26	I/O O	I/O port LCD segment output pin
Function	48	PD1 SEG25	I/O O	I/O port LCD segment output pin
Function	49	PD0 SEG24	I/O O	I/O port LCD segment output pin
Function	50	PC7 SEG23	I/O O	I/O port LCD segment output pin
Function	51	PC6 SEG22	I/O O	I/O port LCD segment output pin

Table 1-1 Pin Names and Functions Sorted by Pin (4/6)

Type	Pin No.	Pin Name	Input/ Out-put	Function
Function	52	PC5 SEG21	I/O O	I/O port LCD segment output pin
Function	53	PC4 SEG20	I/O O	I/O port LCD segment output pin
Function	54	PC3 SEG19	I/O O	I/O port LCD segment output pin
Function	55	PC2 SEG18	I/O O	I/O port LCD segment output pin
Function	56	PC1 SEG17	I/O O	I/O port LCD segment output pin
Function	57	PC0 SEG16	I/O O	I/O port LCD segment output pin
Function	58	PB7 SEG15	I/O O	I/O port LCD segment output pin
Function	59	PB6 SEG14	I/O O	I/O port LCD segment output pin
Function	60	PB5 SEG13	I/O O	I/O port LCD segment output pin
Function	61	PB4 SEG12	I/O O	I/O port LCD segment output pin
Function	62	PB3 SEG11	I/O O	I/O port LCD segment output pin
Function	63	PB2 SEG10	I/O O	I/O port LCD segment output pin
Function	64	PB1 SEG9	I/O O	I/O port LCD segment output pin
Function	65	PB0 SEG8	I/O O	I/O port LCD segment output pin
Function	66	PA7 SEG7	I/O O	I/O port LCD segment output pin
Function	67	PA6 SEG6	I/O O	I/O port LCD segment output pin
Function	68	PA5 SEG5	I/O O	I/O port LCD segment output pin
Function	69	PA4 SEG4	I/O O	I/O port LCD segment output pin
Function	70	PA3 SEG3	I/O O	I/O port LCD segment output pin
Function	71	PA2 SEG2	I/O O	I/O port LCD segment output pin
Function	72	PA1 SEG1	I/O O	I/O port LCD segment output pin

Table 1-1 Pin Names and Functions Sorted by Pin (5/6)

Type	Pin No.	Pin Name	Input/ Out- put	Function
Function	73	PA0 SEG0	I/O O	I/O port LCD segment output pin
PS	74	DVDD3	−	Power supply pin (note 3)
PS	75	DVSS	−	GND pin
Function	76	COM0	O	LCD common output pin
Function	77	COM1	O	LCD common output pin
Function	78	COM2	O	LCD common output pin
Function	79	COM3	O	LCD common output pin
Function	80	PK0 INT3 LV1	I/O I −	I/O port External interrupt pin Connected to a bleeder resistance.
Function	81	PK1 TB1OUT LV2	I O −	I/O port Timer B output Connected to a bleeder resistance.
PS	82	VLC	−	LCD power supply pin (note 3)
PS	83	AGNDREF0	−	24bit ΔΣAD converter: GND pin (note 4) (note 5)
Function	84	DAIN0+	I	24bit ΔΣAD converter analog input
Function	85	DAIN0-	I	24bit ΔΣAD converter analog input
PS	86	VREFIN0	−	Supplying the 24bit ΔΣAD converter with a power supply. (note 4) (note 5)
PS	87	AGNDREF1	−	24bit ΔΣAD converter: GND pin (note 4) (note 5)
Function	88	DAIN1+	I	24bit ΔΣAD converter analog input
Function	89	DAIN1-	I	24bit ΔΣAD converter analog input
PS	90	VREFIN1	−	Supplying the 24bit ΔΣAD converter with a power supply. (note 4) (note 5)
PS	91	AGNDREF2	−	24bit ΔΣAD converter: GND pin (note 4) (note 5)
Function	92	DAIN2+	I	24bit ΔΣAD converter analog input
Function	93	DAIN2-	I	24bit ΔΣAD converter analog input
PS	94	VREFIN2	−	Supplying the 24bit ΔΣAD converter with a power supply. (note 4) (note 5)
PS	95	DSRVSS	−	Voltage reference circuit: GND pin (note 6)
PS	96	DSRVDD3	−	Supplying the amplifier circuit for ΔΣAD converter with a power supply. (note 6)

Table 1-1 Pin Names and Functions Sorted by Pin (6/6)

Type	Pin No.	Pin Name	Input/ Out- put	Function
PS	97	SRVDD	−	Supplying the voltage reference circuit with a power supply. (note 6)
Function	98	PF0 AIN0	I/O I	I/O port 10bit AD converter analog input
Function	99	PF1 AIN1 INT0	I/O I I	I/O port 10bit AD converter analog input External interrupt pin
PS	100	AVSS	−	10bit AD converter: GND pin (note 2)

Note 1: AVDD3 and VREH must be connected to power supply even if 10bit AD converter is not used.
Note 2: AVSS must be connected to GND even if the 10bit AD converter is not used.
Note 3: The same voltage must be supplied to DVDD3, AVDD3, RVDD3, DSRVDD3, SRVDD, VLC.
Note 4: When 24bit ΔΣAD converter is used, provide pin treatments as follows:
- Do not connect VREFINx to a reference voltage.
- Connect AGNDREFx to DVSS level.
- Connect a 1μF capacitor to between VREFINx and AGNDREFx.

Note 5: When 24bit ΔΣAD converter is not used, below settings are required.
- Connect AGNDREFx to DVSS level.

Note 6: When a temperature sensor is also not used, a reference voltage circuit requires below settings.
- Connect DSRVDD3 and SRVDD to DVDD3.
- Connect DSRVSS to DVSS.

Note 7: Only when input is enabled, these pins tolerate 5V inputs.
Note that these pins cannot be pulled up over the power supply voltage when using as open-drain output.

1.5 Pin Numbers and Power Supply Pins

Table 1-2 Pin Numbers and Power Supplies

Power supply	Voltage range	Pin No.	Pin name
AVDD3	1.8 to 3.6V	2	1 to 2 pin, 98 to 100 pin
DVDD3		8, 74	3 to 82 pin
SRVDD		97	95 to 97 pin
VREFIN0	-	86	83 to 86 pin
VREFIN1		90	87 to 90 pin
VREFIN2		94	91 to 94 pin

2. Product Information

This chapter describes the product-specific information about peripherals. Use this chapter in conjunction with Chapter of Peripherals.

2.1 16-bit Timer/Event Counter (TMRB)

TMPM061FWFG contains 2 channels of TMRB. The following are the product-specific functions:

1. Timer flip-flop output used as a capture trigger

   An output of 16-bit timer A(TMR16A) can be used as a capture trigger for TMRB.
   - T16A3OUT → TMRB0
   - T16A6OUT → TMRB1

2. Start trigger in the timer synchronous mode

   A synchronous start of multiple channels is possible.
   - TMRB0 → Start TMRB0 and TMRB1 simultaneously.

Table 2-1 describes the difference of TMRB.

Table 2-1 Difference of TMRB according to channels (n: channel number)

Channel	Pin (Port/pin number)		Trigger function between timers	
	External clock/ capture trigger input pin TBnIN	Timer flip-flop output pin TBnOUT	Capture trigger	Synchronous start trigger channel
TMRB0	PI3 (21)	PG0 (11)	T16A3OUT	-
TMRB1	PJ5 (33)	PK1 (81)	T16A6OUT	TMRB0

2.2 16-bit Timer A (TMR16A)

TMPM061FWFG contains 7 channels of TMR16A. Channel 4 does not provide a match interrupt signal and output signal to the rectangular wave pin.

Table 2-2 shows a list of rectangular wave output pins of TMR16A.

Table 2-2 Used pins of TMR16A (n: channel number)

Channel	T16AnOUT pin
TMR16A0	PH2 (14)
TMR16A1	PH5 (17)
TMR16A2	PI2 (20)
TMR16A3	PJ4 (32)
TMR16A4	-
TMR16A5	PI6 (27)
TMR16A6	PE2 (39)

2.3 Serial Channel (SIO/UART)

TMPM061FWFG contains 4 channels of SIO. The following are the product-specific functions:

1. Timer output used as a transfer clock

   In the UART mode, a timer output can be used as a transfer clock.
   - T16A0OUT → SIO0, SIO1
   - T16A1OUT → SIO2, SIO3

2. Timer output used as carrier pulses

   A timer flip-flop output can be used for transmission using carrier pulses. However, SIO3 does not have this function.
   - T16A4OUT→ SIO0, SIO1, SIO2

Table 2-3 describes the difference of SIO.

SIO3 can choose a port to use. Either port must be valid. SIO3 does not have a function of IROUTn (carrier pulse transmission with data).

Table 2-3 Difference of SIO according to channels (n: channel number)

Channel	Pin (Port/pin number)			Timer output	
	TXDn IROUTn	RXDn	SCLKn $\overline{\text{CTSn}}$	Used as clock	Used as carrier pulse
SIO0	PH0 (12)	PH1 (13)	PH2 (14)	T16A0OUT	T16A4OUT
SIO1	PH3 (15)	PH4 (16)	PH5 (17)		
SIO2	PI0 (18)	PI1 (19)	PI2 (20)	T16A1OUT	
SIO3	PI4 (25)	PI5 (26)	PI6 (27)		
	PE5 (36)	PE4 (37)	PE3 (38)		-

2.4 Analog/Digital Converter (ADC)

2.4.1 Non-Usable Functions

In the TMPM061FWFG, the following ADC functions cannot be used. Do not set the related registers.

Function	Register
Highest priority conversion	ADMOD2, ADREGSP
AD monitoring function	ADMOD3, ADMOD5, ADCMP0, ADCMP1
AD start-up by hardware	ADMOD4 <ADHTG> <ADHS> <HADHTG> <HADHS>

2.4.2 Conversion Channel

In the TMPM061FWFG, 4 channels from 0 to 3 are used as input channels of the AD converter. Analog signals input to each channel are as follows.

Table 2-4 ADC input

Channel	Input
Channel 0	AIN0 pin (PF0/98pin)
Channel 1	AIN1 pin (PF1/99pin)
Channel 2	Temperature sensor output
Channel 3	ΔΣADC reference voltage circuit(BGR) output

Conversion channels are specified with ADMOD0<SCAN>, ADMOD1<ADSCN> and <ADCH>. Table 2-5 describes available settings.

Table 2-5 Channel selection on normal conversion

		ADMOD1<ADCH[3:0]>				
		0000	0001	0010	0011	0100 to 1111
ADMOD0 <SCAN>=0	Fixed channel	AIN0	AIN1	AIN2	AIN3	Not available
ADMOD0 <SCAN>=1	ADMOD1<ADSCN>=00 4-channel scan	AIN0	AIN0~AIN1	AIN0~AIN2	AIN0~AIN3	
	ADMOD1<ADSCN>=01 8-channel scan	AIN0	AIN0~AIN1	AIN0~AIN2	AIN0~AIN3	
	ADMOD1<ADSCN>=10 12-channel scan	AIN0	AIN0~AIN1	AIN0~AIN2	AIN0~AIN3	

2.5 ΔΣ Analog/Digital Converter (DSADC)

TMPM061FWFG contains 3 units of DSADC.

In the synchronous start function of DSADC, the following table is an assignment of a master unit and slave unit.

Table 2-6 Master/slave assignment

Master	Slave
Unit 0	Unit 1 Unit 2

3. Processor Core

The TMPM061FWFG series has a high-performance 32-bit processor core (the Arm Cortex-M0 processor core). For information on the operations of this processor core, please refer to the "Cortex-M0 Technical Reference Manual" issued by Arm Limited. This chapter describes the functions unique to the TMPM061FWFG series that are not explained in that document.

3.1 Information on the processor core

The following table shows the revision of the processor core in the TMPM061FWFG.

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

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

Product Name	Core Revision
TMPM061FWFG	r0p0-03

3.2 Configurable Options

The Cortex-M0 core has optional blocks. The following tables shows the configurable options in the TMPM061FWFG.

Configurable Options	Implementation
Interrpts	32
Data endiannes	Little-endian
SysTick timer	Present
Number of watchpoint comparators	2
Number of breakpoint comparators	4
Halting debug support	Present
Multiplier	Fast

Note: The fast multiplier provides a 32-bit × 32-bit multiply that yields the least-significant 32-bits.

3.3 Exceptions/ Interruptions

Exceptions and interruptions are described in the following section.

3.3.1 Number of Interrupt Inputs

The number of interrupt inputs can optionally be defined in the Cortex-M0 core.

TMPM061FWFG has 32 interrupt inputs.

3.3.2 SysTick

TMPM061FWFG has a SysTick timer which can generate SysTick exception.

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

3.3.3 SYSRESETREQ

The Cortex-M0 core outputs SYSRESETREQ signal when <SYSRESETREQ> bit of Application Interrupt and Reset Control Register are set.

TMPM061FWFG provides the same operation when SYSRESETREQ signal are output.

Note: The reset operation by <SYSRESETREQ> can not used while in SLOW mode.

3.3.4 LOCKUP

When irreparable exception generates, the Cortex-M0 core outputs LOCKUP signal to show a serious error included in software.

TMPM061FWFG does not use this signal. To return from LOCKUP status, it is necessary to use non-maskable interruput (NMI) or reset.

3.4 Events

The Cortex-M0 core has event output signals and event input signals. An event output signal is output by SEV instruction execution. If an event is input, the core returns from low-power consumption mode caused by WFE instruction.

TMPM061FWFG does not use event output signals and event input signals. Please do not use SEV instruction and WFE instruction.

3.5 Power Management

The Cortex-M0 core provides power management system which uses SLEEPING signal and SLEEPDEEP signal. SLEEPDEEP signals are output when <SLEEPDEEP> bit of System Control Register is set.

These signals are output in the following circumstances:

-Wait-For-Interrupt (WFI) instruction execution

-Wait-For-Event (WFE) instruction execution

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

TMPM061FWFG does not use SLEEPDEEP signal so that <SLEEPDEEP> bit must not be set. And also event signal is not used so that please do not use WFE instruction.

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

4. Memory Map

4.1 Memory map

The memory maps for theTMPM061FWFG are based on the Arm Cortex-M0 processor core memory map.

The internal ROM is mapped to the code of the Cortex-M0 core memory, the internal RAM is mapped to the SRAM region and the special function register (SFR) is mapped to the peripheral region respectively.

The special function register (SFR) indicates I/O ports and control registers for the peripheral function. TMPM061FWFG has bit-band feature equivalent to Cortex-M3 and the SRAM and SFR regions of TMPM061FWFG 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-M0 Technical Reference Manual".

Note that access to regions indicated as "Fault" causes a hard fault. Do not access the vendor-specific region and the reserved region.

Figure 4-1 shows the memory map of the TMPM061FWFG.

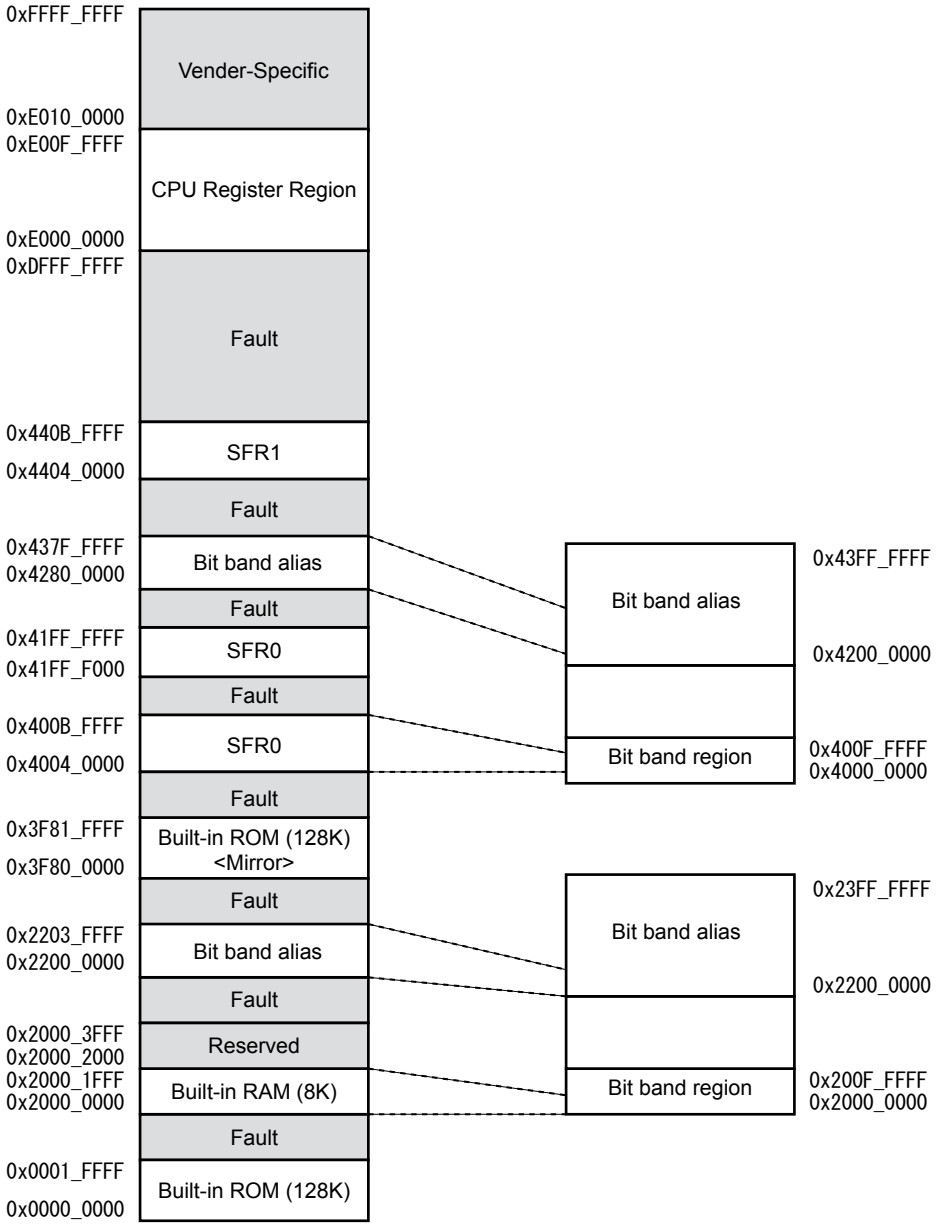


Figure 4-1 Memory Map

4.2 Bus Structure

TMPM061FWFG contains the Cortex-M0 core and PCE acting as a bus master.

Slaves are the built-in ROM, built-in RAM0/1, built-in BOOT ROM, bridge0/1 from AHB-Lite bus to APB bus and bridge 0/1 from AHB-Lite bus to IO bus. Bridges numbered 0 are allocated to SFR0 area and bridges numbered 1 are allocated to SFR1 area. Peripheral functions are connected either the APB bus or IO bus via bridges.

An access to slaves from the core or PCE is executed simultaneously as long as the access is not to the same slave.

Note: PCE cannot access to the built-in BOOT ROM.

Figure 4-2 shows a bus structure.

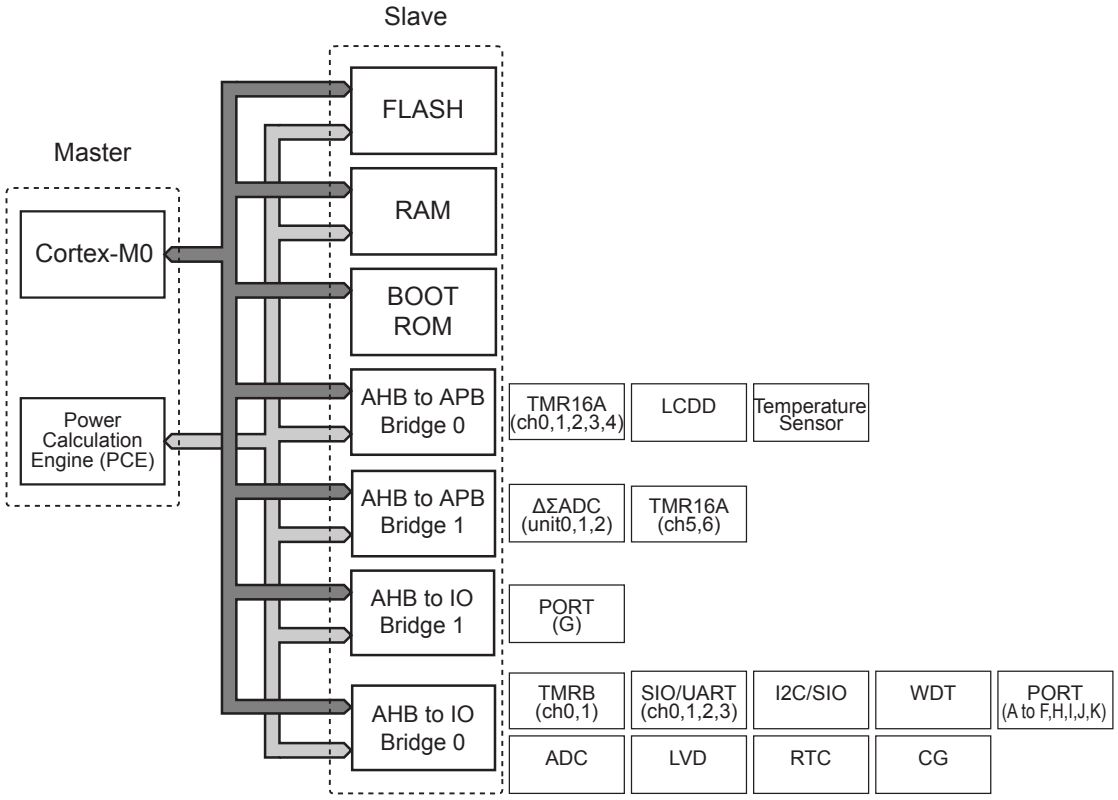


Figure 4-2 Bus structure

4.3 Address lists of peripheral functions

Base addresses of the peripheral functions are shown below. Be careful of which area of SFR0 or SFR1 the peripheral function is allocated.

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

Peripheral Function		Base Address	Area
Clock/Mode control (CG)		0x400F_3000	SFR0
Input/Output Ports	PORTA	0x400C_0000	SFR0
	PORTB	0x400C_0100	SFR0
	PORTC	0x400C_0200	SFR0
	PORTD	0x400C_0300	SFR0
	PORTE	0x400C_0400	SFR0
	PORTF	0x400C_0500	SFR0
	PORTG	0x440C_0600	SFR1
	PORTH	0x400C_0700	SFR0
	PORTI	0x400C_0800	SFR0
	PORTJ	0x400C_0900	SFR0
	PORTK	0x400C_0A00	SFR0
16-bit Timer/Event Counters (TMRB)	ch0	0x400C_4000	SFR0
	ch1	0x440C_4100	SFR1
16-bit Timer A (TMR16A)	ch0	0x4008_D000	SFR0
	ch1	0x4008_E000	SFR0
	ch2	0x4008_F000	SFR0
	ch3	0x4009_0000	SFR0
	ch4	0x4009_1000	SFR0
	ch5	0x4409_2000	SFR1
	ch6	0x4409_3000	SFR1
Serial Channel (SIO/SIO)	ch0	0x400E_1000	SFR0
	ch1	0x400E_1100	SFR0
	ch2	0x400E_1200	SFR0
	ch3	0x400E_1300	SFR0
Serial Bus Interface (I2C/SIO)		0x400E_0000	SFR0
Analog/Digital Converter (ADC)		0x400F_C000	SFR0
ΔΣAnalg/Digital Converter (DSADC)	unit0	0x4406_7000	SFR1
	unit1	0x4406_8000	SFR1
	unit2	0x4406_9000	SFR1
Temperature sensor (TEMP)		0x4005_D000	SFR0
Real Time Clock (RTC)		0x400C_C000	SFR0
LCD Driver (LCD)		0x4006_E000	SFR0
Voltage Detector (LVD)		0x400F_4000	SFR0
Watch Dog Timer (WDT)		0x400F_2000	SFR0
Flash / Debug (FC)		0x41FF_F000	SFR0

5. Reset Operation

The following are sources of reset operation.

- RESET pin ($\overline{\text{RESET}}$)
- Watch-dog timer (WDT)
- Application interrupt by CPU and a signal from the reset register bit <SYSRESETREQ>

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

A reset by WDT is refer to the chapter on the "Watch-dog timer".

A reset by <SYSRESETREQ> is referred to "Cortex-M0 Technical Reference Manual".

Note 1: Once reset operation is done, internal RAM data is not assured.
Note 2: In the SLOW mode, do not use a reset by <SYSRESETREQ>.

5.1 Cold Reset

When turning-on power, $\overline{\text{RESET}}$ pin must be kept "Low".

When turning-on power, it is necessary to take a stable time of built-in regulator into consideration. In the TMPM061FWFG, the internal regulator requires at least approximately 1ms to be stable. At cold reset, $\overline{\text{RESET}}$ pin must be kept "Low" for a duration of time sufficiently long enough for the internal regulator to be stable. Approximately 1.6ms after $\overline{\text{RESET}}$ pin becomes "High", internal reset will be released.

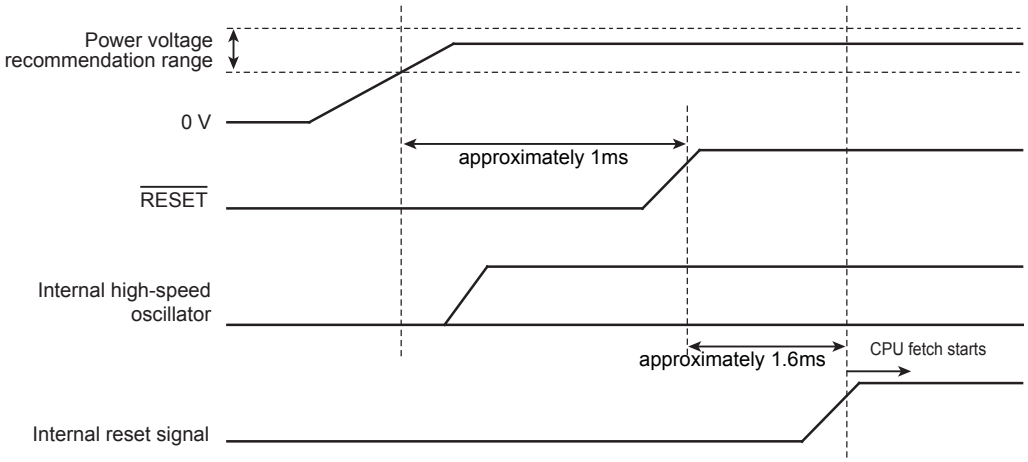


Figure 5-1 Cold Reset Operation Sequence

Note: The above sequence is applied as well when restoring power.

5.2 Warm Reset

To do reset TMPM061FWFG, the following conditions are required; power supply voltage is in the operational range ; $\overline{\text{RESET}}$ pin is kept "Low" at least for 12 internal high-speed clocks. Approximately 1.6ms after $\overline{\text{RESET}}$ pin becomes "High", internal reset will be released.

In case of WDT reset or <SYSRESETREQ> reset, internal reset will be released approximately 30 internal high-speed clocks after reset.

5.3 After reset

All of the control register of the internal core and the peripheral function control register (SFR) are initialized by reset.

When reset is released, TMPM061FWFG starts operation by a clock of internal high-speed oscillator. External clock should be set if necessary.

6. Clock/Mode control

6.1 Features

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

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

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

The clock/mode control block has the following functions:

- Controls the system clock
- Controls the prescaler clock
- Controls the warm-up timer

In addition to NORMAL mode, the TMPM061FWFG can operate in variety of low power modes to reduce power consumption according to its usage conditions.

6.2 Registers

6.2.1 Register List

The table below shows control registers and their addresses.

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

Register name		Address(Base+)
System control register	CGSYSCR	0x0000
Oscillation control register	CGOSCCR	0x0004
Standby control register	CGSTBYCR	0x0008
External high-speed clock select register	CGEHCLKSEL	0x000C
System clock selection register	CGCKSEL	0x0010

6.2.2 CGSYSCR (System control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	FCSTOP	-	-	SCOSEL	
After reset	0	0	0	0	0	0	0	1
	15	14	13	12	11	10	9	8
bit symbol	-	-	FPSEL1	FPSEL0	-	PRCK		
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	GEAR		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-21	−	R	Read as 0.
23	−	R/W	Write "0".
22-21	−	R	Read as 0.
20	FCSTOP	R/W	ADC clock 0: Active 1: Stop This bit enables to stop providing clock to the AD converter and ΣΔAD converter. ADC clock is provided after reset. Confirming that AD conversion has stopped or has been completed before setting "1" (stop) to the register.
19-18	−	R	Read as 0.
17-16	SCOSEL[1:0]	R/W	SCOUT out 00: fs 01: fsys/2 10: fsys 11: φT0 Specifies the clock to output from SCOUT pin.
15-14	−	R	Read as 0.
13	FPSEL1	R/W	fperiph 1 0: φT0 1: fsys

Bit	Bit Symbol	Type	Function
12	FPSEL0	R/W	fperiph 0 0: fgear 1: fc Selecting fc fixes fperiph regardless of the clock gear mode.
11	−	R	Read as 0.
10-8	PRCK[2:0]	R/W	Prescaler clock 000: fperiph 100: fperiph/16 001: fperiph/2 101: fperiph/32 010: fperiph/4 110: Reserved 011: fperiph/8 111: Reserved Specifies the prescaler clock to peripheral I/O.
7-3	−	R	Read as 0.
2-0	GEAR[2:0]	R/W	High-speed clock gear (fc) gear (Note) 000: fc 100: fc/2 001: Reserved 101: fc/4 010: Reserved 110: fc/8 011: Reserved 111: fc/16

Note: You cannot select fc/16 when the SysTick timer is being used.

6.2.3 CGOSCCR (Oscillation control register)

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

Bit	Bit Symbol	Type	Function
31-20	WUPT[11:0]	R/W	Warm-up counter setup value. (Note 1) Setup the 16-bit timer for warm-up timer of upper 12-bits counter value.
19	HWUPSEL2	R/W	Warm-up clock. (Note 2) 0: internal (f_{IHOSC}) 1: external (f_{EHOSC}) Selects warm-up counter by high-speed oscillator. When using STOP/SLEEP mode, please select clock-source that is same as <OSCSEL> to <WUPSEL2> before entering to STOP/SLLEP mode.
18	HOSCEN	R/W	External high-speed oscillator 0: Not using X1/X2(f_{EHOSC}) 1: Using X1/X2(f_{EHOSC}) Set "1" to use the external high-speed oscillator (X1/X2).

Bit	Bit Symbol	Type	Function
17	OSCSEL	R/W	High-speed oscillator 0: internal (f_{IHOSC}) 1: external (f_{EHOSC}) The high-speed oscillator is changed between internal and external. Stopping the internal oscillator after switching to the external oscillator can reduce the power consumption.
16	XEN2	R/W	Internal high-speed oscillator operation (Note 3) 0: Stop 1: Oscillation
15-14	WUPTL[1:0]	R/W	Warm-up counter setup value (Note 1) If high-speed oscillator is selected, <WUODR[1:0]> is set "00".
13-12	−	R/W	Write "0".
11	EHCLKEN	R/W	External high-speed clock input enable 0: Disable 1: Enable
10	LOSCSEL	R/W	Low-speed clock 0: Low-speed oscillator (f_{ELOSC}) 1: Low-speed clock ($f_{ELCLKIN}$)
9	XTEN	R/W	External low-speed oscillator mode 0: Stop 1: Oscillation
8	XEN1	R/W	External high-speed oscillator mode 0: Stop 1: Oscillation
7-4	−	R/W	Write "0011"
3	WUPSEL1	R/W	Warm-up clock 0: High-speed clock 1: Low-speed clock Select source clock for warm-up timer. High-speed clock is followed by <WUPSEL2>. Low-speed clock is selected by CGOSCCR<LOSCSEL>.
2	−	R/W	Write "0".
1	WUEF	R	Operation of warm-up timer (WUP) for oscillator 0: WUP finish 1: WUP active Enables to monitor the status of the warm-up timer.
0	WUEON	W	Operation of warm-up timer (WUP) for oscillator 0: don't care 1: WUP start Enables to start the warm-up timer. Read as 0.

Note 1: Refer to Section "6.3.4 Warm-up function" about the Warm-up setup.

Note 2: the external high-speed clock input ($f_{EHCLKIN}$) cannot be used as warm-up clock.

Note 3: When using internal high-speed oscillator (IHOSC), do not use it as system clock which high accuracy assurance is required.

6.2.4 CGSTBYCR (Standby control register)

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

Bit	Bit Symbol	Type	Function
31-20	−	R	Read as 0.
19-17	−	R/W	Write "0" after reset.
16	DRVE	R/W	Pin status in STOP mode. 0: Inactive in STOP mode 1: Active in STOP mode
15-10	−	R	Read as 0.
9	RXTEN	R/W	Low-speed oscillator after releasing STOP mode. 0: Stop 1: Oscillation This bit specifies the low-speed oscillator operation after releasing the STOP mode when the low-speed oscillator is not used as system clock.
8	RXEN	R/W	High-speed oscillator after releasing STOP mode. 0: Stop 1: Oscillation This bit specifies the operation of the high-speed oscillator at the release of the STOP in case the high-speed oscillator is not used as the system clock. Setting OSCCR<OSCSEL> enables internal oscillation or external oscillation.
7-3	−	R	Read as 0.
2-0	STBY[2:0]	R/W	Low power consumption mode 000: Reserved 001: STOP 010: SLEEP 011: IDLE 100: Reserved 101: Reserved 110: Reserved 111: Reserved

6.2.5 CGEHCLKSEL (External high-speed clock select register)

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

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as 0.
15-12	−	R/W	Write as "0111".
11	−	R	Read as 0.
10-1	−	R/W	Write as "0100001111".
0	EHCLKSEL	R/W	External high-speed clock 0: Use the clock selected by CGOSCCR<OSCSEL> 1: Use the external high-speed clock input

6.2.6 CGCKSEL (System clock selection register)

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

Bit	Bit Symbol	Type	Function
31-2	−	R	Read as 0.
1	SYSCK	R/W	System clock 0: High-speed 1: Low-speed
0	SYSCKFLG	R/W	System clock status 0: High-speed 1: Low-speed

6.3 Clock control

6.3.1 Clock Type

Each clock is defined as follows:

f_{EHOSC}	: External high-speed oscillator clock input from oscillator connected to X1, X2
$f_{EHCLKIN}$	: External high-speed clock input from X1
f_{IHOSC}	: Internal high-speed oscillator clock input from internal oscillator
fs	: External low-speed oscillator clock input from oscillator connected to XT1, XT2
$f_{ELCLKIN}$	: External low-speed clock input from PJ5 (33pin)
fosc	: High-speed clock specified by CGOSCCR<OSCSEL>
fc	: Clock specified by CGEHCLKSEL<EHCLKSEL> (high-speed clock)
fgear	: Clock specified by CGSYSCR<GEAR[2:0]>
fsys	: Clock specified by CGCKSEL<SYSCK> (system clock)
fperiph	: Clock specified by CGSYSCR<FPSEL0>
φT0	: Clock specified by CGSYSCR<PRCK[2:0]> (prescaler clock)

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

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

6.3.2 Initial Values after Reset

Reset operation initializes the clock configuration as follows.

internal high-speed oscillator	: oscillating
external high-speed oscillator	: stop
PLL (phase locked loop circuit)	: stop
High-speed clock gear	: fc (no frequency dividing)

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

$fc = f_{IHOSC}$

$fsys = fc (=f_{IHOSC})$

$fperiph = fc (=f_{IHOSC})$

$φT0 = fperiph (=f_{IHOSC})$

6.3.3 Clock system Diagram

Figure 6-1 shows the clock system diagram.

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

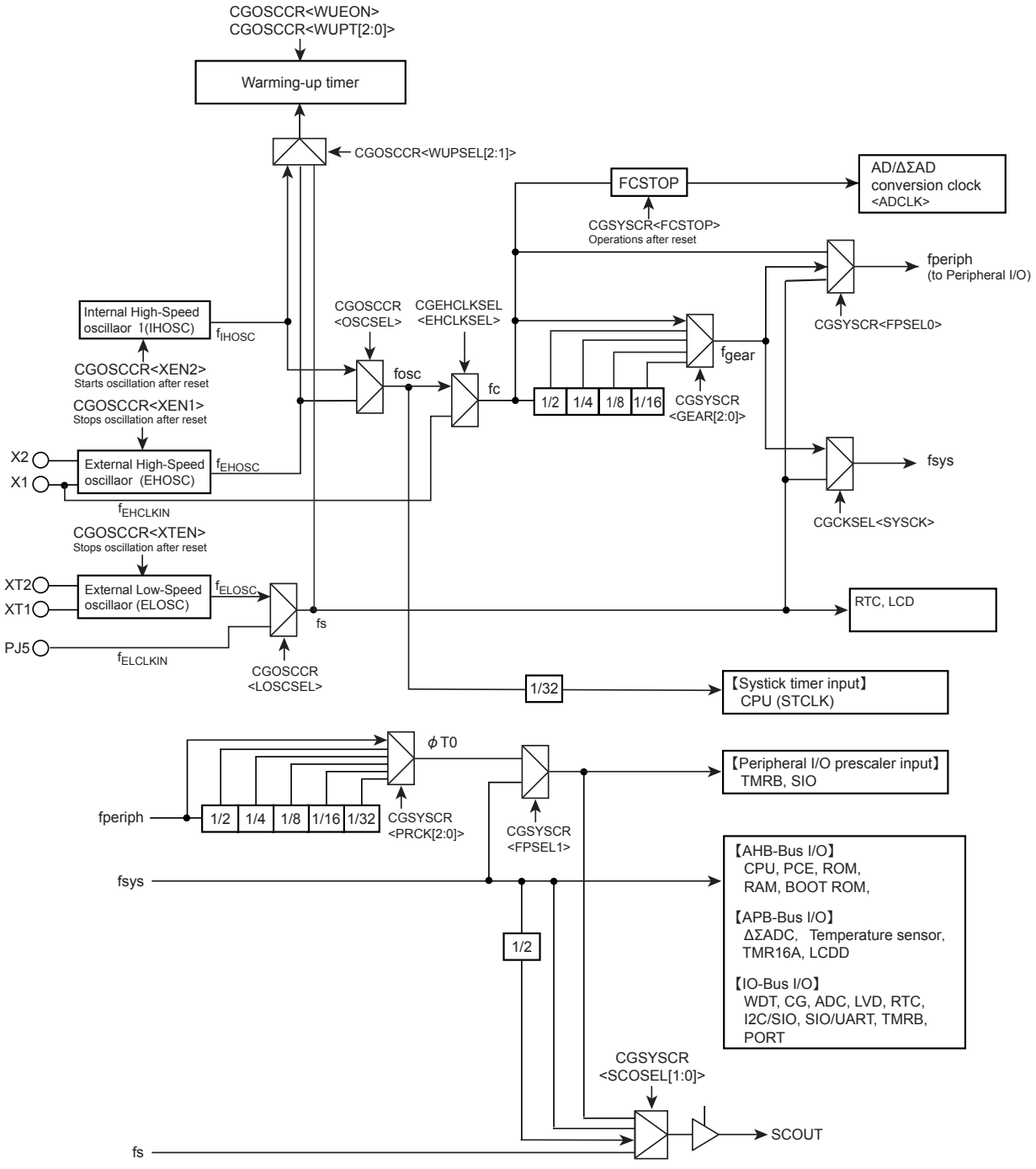


Figure 6-1 Clock Block Diagram

6.3.4 Warm-up function

The warm-up function secures the stability time for the oscillator with the warm-up timer. When using stable external clock, warm-up function is not necessary.

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.

Note: Transition to the low power consumption mode while the warm-up timer is operating is prohibited.

How to configure the warm-up function.

1. Specify the count up clock

   Specify the count up clock for the warm-up counter in the CGOSCCR <WUPSEL2> <WUPSEL1> bit.

   Note: the external high-speed clock input ($f_{EHCLKIN}$) cannot be used as a warm-up clock. To use the $f_{EHCLKIN}$ as a system clock, specify the f_{IOSC}(internal high-speed oscillation) as a warm-up clock. In this case, you cannot stop the f_{IOSC}.

2. Specify the warm-up counter value

   CGOSCCR<WUPT[11:0]><WUPTL[1:0]> uses <WUPT> and <WUPTL> for counting with a low-speed clock and <WUPT> for a high-speed clock. Set "00" to <WUPTL> for using a high-speed clock.

   Setting values can be calculated using the formula shown below and round off the lower four bits.

   Note: Setting warm-up count value to CGOSCCR<WUPT><WUPTL>, wait until this value is reflected, then transit to stand-by mode by executing a command "WFI".

$$\text{number of warm-up cycle} = \frac{\text{warm-up time to set}}{\text{input frequency cycle (s)}}$$

   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.

   <example 1> When using high-speed oscillator 8MHz, and set warm-up time 5ms.

$$\frac{\text{warm-up time to set}}{\text{input frequency cycle (s)}} = \frac{5\text{ms}}{1/8\text{MHz}} = 40{,}000 \text{ cycle} = 0\text{x}9\text{C}40$$

   Round lower 4 bit off, set 0x9C4 to CGOSCCR<WUPT[11:0]>.

3. confirm the start and completion of warm-up

The CGOSCCR<WUEON><WUEF> is used to confirm the start and completion of warm-up through software (instruction). When CGOSCCR<WUEON> is set to "1", the warm-up start a count up. The completion of warm-up can be confirmed with CGOSCCR<WUEF>.

The example of warm-up function setup.

Table 6-1 <example> from STOP mode to NORMAL mode transition (internal high-speed oscillator is selected)

CGOSCCR<WUPT> = "0x9C4"	: Specify the warm-up time
CGOSCCR<WUPT> read	: Confirm warm-up time reflecting Repeat until the read data is "0x9C4".
CGOSCCR<XEN2> = "1"	: high-speed oscillator (fosc) enable
CGOSCCR<WUEON> = "1"	: Start the warm-up timer
CGOSCCR<WUEF> read	: Wait until the state becomes "0" (warm-up is finished)

6.3.5 System clock

One of the following clocks can be used as a source clock of the system clock: internal high-speed oscillation clock, external high-speed oscillation clock (connected to an oscillator or a clock input) and external low-speed clock (connected to an oscillator or a clock input).

Internal high-speed oscillation should not be used if high accuracy assurance is required.

Source clock		Frequency
Internal high-speed oscillation (f_{IHOSC})		10MHz
External high-speed oscillation	Oscillator (f_{EHOSC})	8 to 16MHz
	Input clock ($f_{EHCLKIN}$)	
External low-speed oscillation	Oscillator (f_{ELOSC})	30 to 34kHz
	Input clock ($f_{ELCLKIN}$)	

The system clock can be divided by CGSYSCR<GEAR> when using a high-speed oscillator. Although the settings can be changed while operating, the actual switching takes place after a slight delay.

Operating frequency examples configured by clock gear settings are shown in Table 6-2.

Table 6-2 System clock frequency

fosc frequency (MHz)	Clock gear (CG)				
	1/1	1/2	1/4	1/8	1/16
8	8	4	2	1	-
10	10	5	2.5	1.25	-
16	16	8	4	2	1

↑ Initial value after reset

Note: Do not use 1/16 when using SysTick.

6.3.5.1 System clock switching

The Figure 6-2 shows how to switch the system clock.

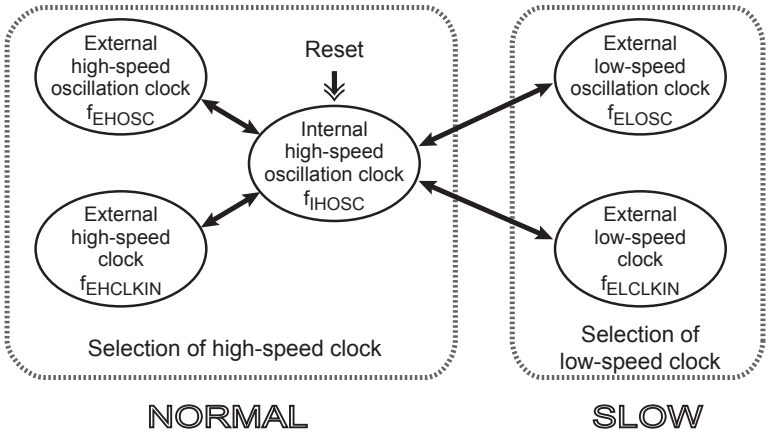


Figure 6-2 System clock switching

The internal high-speed oscillation clock is set as the system clock after releasing the reset signal. A high-speed clock can be selected from the internal high-speed oscillation clock, the external high-speed oscillation clock and external high-speed clock. A low-speed clock can be selected from the external low-speed oscillation clock and the external low-speed clock. A clock cannot be switched between the external high-speed oscillation clock and external high-speed clock, and also between the external low-speed oscillation clock and external low-speed clock.

6.3.5.2 Precautions for transiting to SLOW, STOP and SLEEP mode

To transit to SLOW mode, STOP mode or SLEEP mode, switch the system clock to the internal high-speed oscillation clock by setting CGOSCCR<OSCSEL> first. While using the external high-speed oscillation clock or the external high-speed clock, mode cannot be switched to STOP or SLEEP.

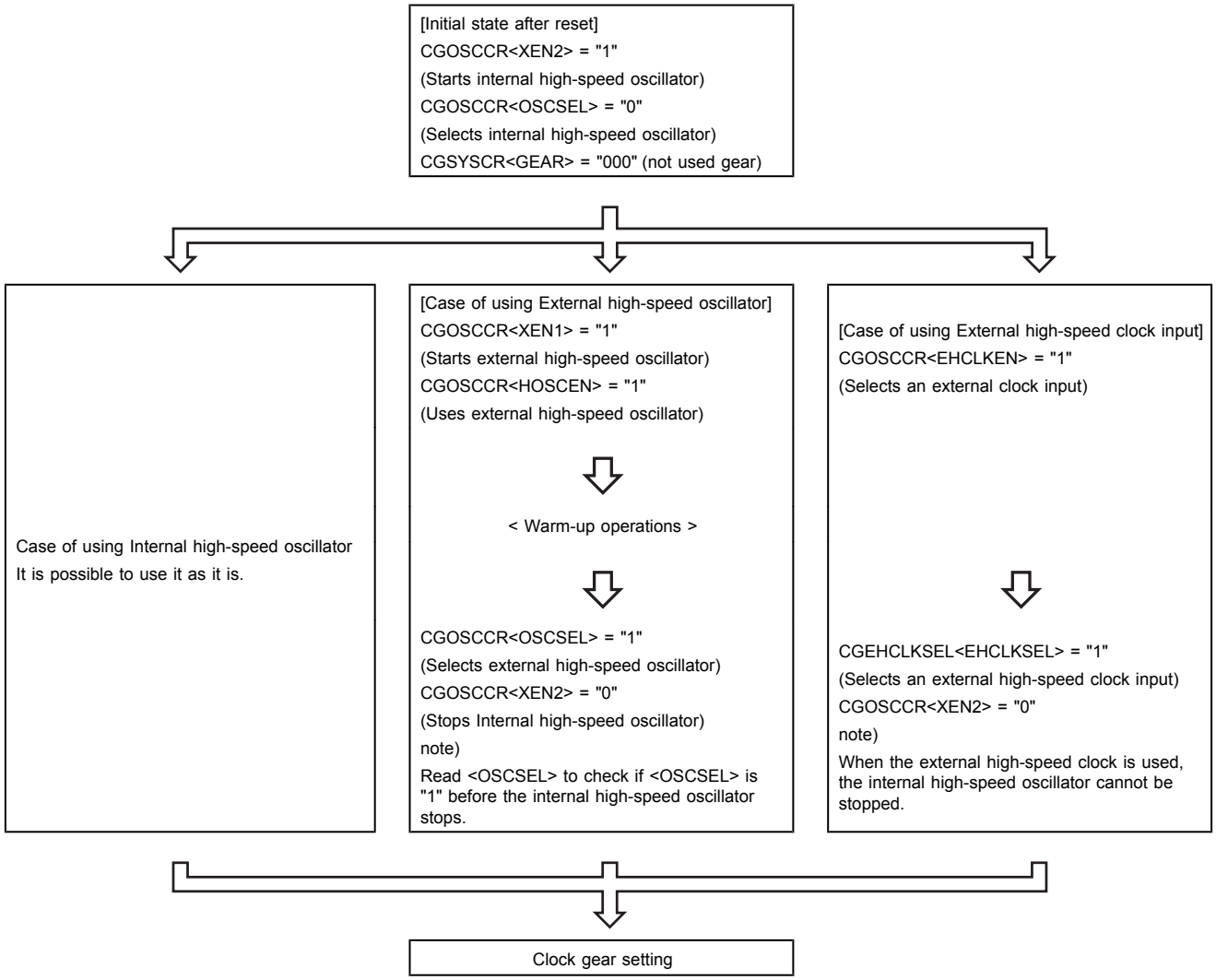
6.3.5.3 Clock setting

The system clock can be selected by setting the CGOSCCR and CGEHCLKSEL. After selecting the clock, set the clock gear by setting the CGSYSCR if required.

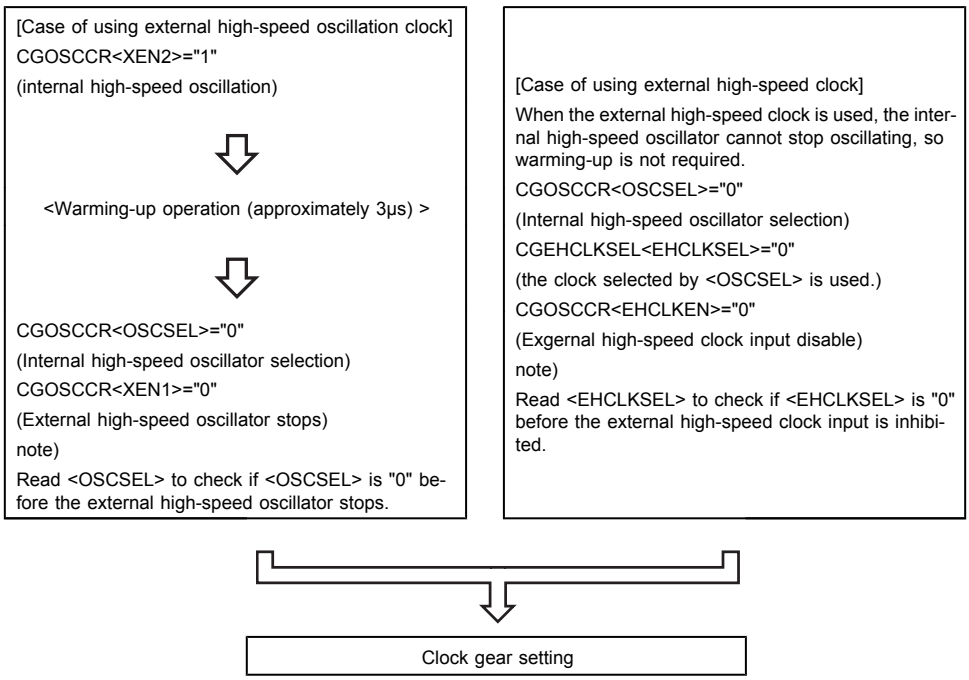
The clock setup sequence is shown as below.

If a low-speed clock is set as the system clock, set the low-speed clock in NORMAL mode first and then shift to the SLOW mode.

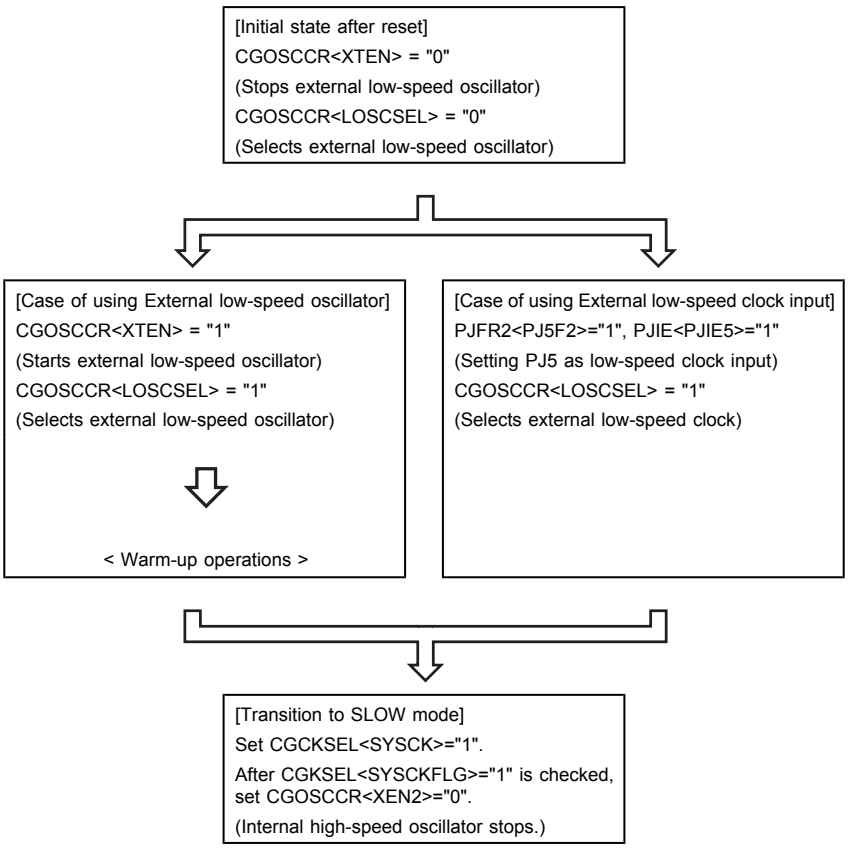
1. High-speed clock setup sequence (Initial setting)



2.High-speed clock setting sequence (Switch from external to internal high-speed clock)



3. Low-speed clock setup sequence



6.3.6 Prescaler Clock Control

Peripheral IO (TMRB,SIO) has a prescaler for dividing a clock. As the clock φT0 to be input to each prescaler, the "fperiph" clock specified in the CGSYSCR<FPSEL0> can be divided according to the setting in the CGSYSCR<PRCK[2:0]>. After the controller is reset, fperiph/1 is selected as φT0.

When the clock fs is being used for the system clock fsys, fs can be used for the prescaler clock by setting "1" to CGSYSCR<FPSEL1>.

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 TMPM061FWFG enables to output the system clock from a pin. The SCOUT pin can output the system clock fsys and fsys/2, and the prescaler input clock for peripheral I/O φT0.

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

By setting the port J registers, the PJCR<PD4C> and PJFR2<PJ4F2> to "1", the PJ4 pin becomes the SCOUT output pin. The output clock is selected by setting the CGSYSCR<SCOSEL[1:0]>.

Table 6-3 shows the pin status in each mode when the SCOUT pin is set to the SCOUT output.

Table 6-3 SCOUT Output Status in Each Mode

Mode / SCOUT selection CGSYSCR	NORMAL	Low power consumption mode		
		IDLE	SLEEP	STOP
<SCOSEL[1:0]> = "00"	Output the fs clock		Fixed to "0" or "1".	
<SCOSEL[1:0]> = "01"	Output the fsys/2 clock			
<SCOSEL[1:0]> = "10"	Output the fsys clock			
<SCOSEL[1:0]> = "11"	Output the φT0 clock			

Note 1: The phase difference (AC timing) between the system clock output by the SCOUT and the internal clock is not guaranteed.

Note 2: If fsys is selected for SCOUT, output waveforms may undergo distortion when the clock gear is switched.

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 the 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.

Figure 6-3 shows a mode transition diagram.

For a detail of sleep-on-exit, refer to "Cortex-M0 Technical Reference Manual."

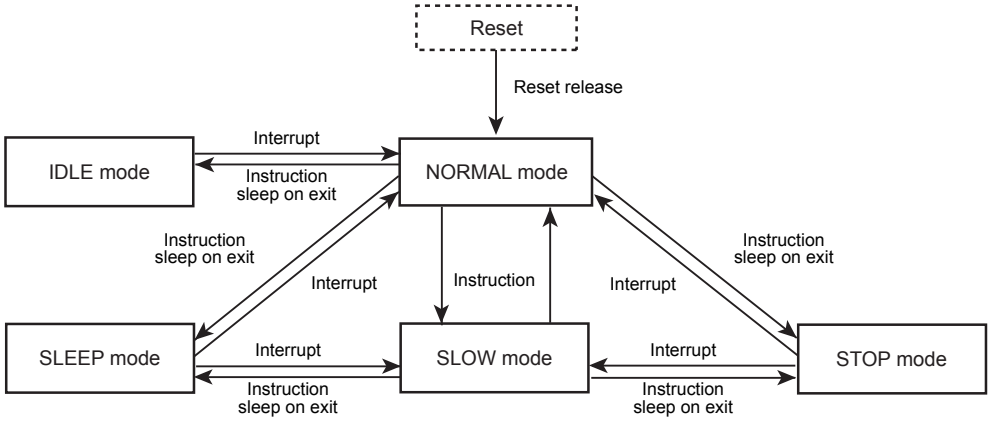


Figure 6-3 Mode Transition Diagram

Note: Operation mode cannot be shifted to STOP or SLEEP mode when using the external high-speed oscillation clock (f_{EHOSC}) and the external high-speed clock ($f_{EHCLKIN}$). Switch the clock to the internal high-speed oscillation clock (f_{IHOSC}) first.

6.5 Operation mode

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.

It is shifted to the NORMAL mode after reset. The low-speed clock can also be used.

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.

Operable peripheral functions are limited in SLOW mode. The following peripheral functions are operable: I/O port (PORT), Power Calculation Engine (PCE), Timer (TMRB and TMR16A), watchdog timer (WDT), real-time clock (RTC), LCD driver, Low Voltage Detect (LVD) and temperature sensor (TEMP).

Note 1: In SLOW mode, stop all functions other than operable peripheral functions before shifting to SLOW mode.

Note 2: In the slow mode, be sure not to perform reset using the Application Interrupt and Reset Control Register<SYSRESETREG> of the Cortex-M0 NVIC register.

6.6 Low Power Consumption Modes

The TMPM061FWFG has three low power consumption modes: IDLE, SLEEP and STOP. To shift to the low power consumption mode, specify the mode in the system control register CGSTBYCR<STBY[2:0]> and execute the WFI (Wait For Interrupt) instruction. In this case, execute reset or generate the interrupt to release the mode. Releasing by the interrupt requires settings in advance. See the chapter "Exceptions" for details.

Note 1: The TMPM061FWFG does not offer any event for releasing the low power consumption mode. Transition to the low power consumption mode by executing the WFE (Wait For Event) instruction is prohibited.

Note 2: The TMPM061FWFG does not support the low power consumption mode configured with the SLEEPDEEP bit in the Cortex-M0 core. Setting the <SLEEPDEEP> bit of the system control register is prohibited.

Note 3: Transition to the low power consumption mode while the warm-up timer is operating is prohibited.

Note 4: The operation mode cannot be shifted to STOP mode or SLEEP mode when using the external high-speed oscillation clock (f_{EHOSC}) and the external high-speed clock ($f_{EHCLKIN}$). Switch the clock to the internal high-speed oscillation clock (f_{IHOSC}) first.

Note 5: To use the LCD driver in SLEEP mode, operation mode needs to be shifted from SLOW mode to SLEEP mode.

The features of IDLE, SLEEP, STOP mode are described as follows.

6.6.1 IDLE mode

This is a mode that the clock of the processor core stops. 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 timer A (TMR16A)
- Serial channel (SIO/UART)
- Serial bus interface (I2C/SIO)
- Analog Digital converter (ADC)
- Watch dog timer (WDT)

6.6.2 SLEEP mode

This is a mode that a high-speed clock stops. A real time clock (RTC) counter that operates with low-speed clock signals keep operation.

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

6.6.3 STOP mode

All the clocks including an internal oscillator stop in STOP mode.

By releasing the STOP mode, the device returns to the preceding mode of the STOP mode and starts operation. Set 200μs or more as the warming up time after releasing the STOP mode.

The STOP mode enables to select the pin status by setting the CGSTBYCR<DRVE>. Table 6-4 shows the pin status in the STOP mode.

Table 6-4 Pin States in the STOP mode

Usage	Function	Port	Configuration	I/O	<DRVE> = 0	<DRVE> = 1
Port	SWCLK	PE6	PEFR2<PE6F2>=1 , PEIE<PE6IE>=1 or $\overline{\text{DBGEN}}$="0" at the rising edge of $\overline{\text{RESET}}$	Input	o	
	SWDIO	PE7	PEF21<PE7F2>=1, PECR<PE7C>=1, PEIE<PE7IE>=1 or $\overline{\text{DBGEN}}$="0" at the rising edge of $\overline{\text{RESET}}$	Input	o	
				Output	Enable when data is valid Disable when data is invalid	
	INT0 INT1 INT2 INT3	PF1 PJ2 PG0 PK0	PxFRn<PxmFn>=1 PxIE<PxmIE>=1	Input	o	
	Function except upper description or general purpose input/output port		PxIE<PxmIE>=1	Input	×	o
			PxCR<PxmC>=1	Output	×	o
Except port	$\overline{\text{RESET}}$, MODE			Input only	o	
	COM0,COM1,COM2,COM3			Output only	o	
	X1,XT1			Input only	×	
	X2,XT2			Output only	"High" level output	

o : Valid input or output.
× : Invalid input or output.

Note: x: port number / m: corresponding bit

6.6.4 Low power Consumption Mode Setting

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

Table 6-5 shows the mode setting in the <STBY[2:0]>.

Table 6-5 Low power consumption mode setting

Mode	CGSTBYCR <STBY[2:0]>
STOP	001
SLEEP	010
IDLE	011

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

6.6.5 Operational Status in Each Mode

Table 6-6 show the operational status in each mode.

Table 6-6 Operational Status in Each Mode

Block	NORMAL			IDLE			SLOW		SLEEP		STOP
	Internal high-speed oscillator use (f_{IHOSC})	External high-speed oscillator use (f_{EHOSC})	External high-speed clock use ($f_{EHCLKIN}$)	Internal high-speed oscillator use (f_{IHOSC})	External high-speed oscillator use (f_{EHOSC})	External high-speed clock use ($f_{EHCLKIN}$)	External low-speed oscillator use (f_{ELOSC})	External low-speed clock use ($f_{ELCLKIN}$)	External low-speed oscillator use (f_{ELOSC})	External low-speed clock use ($f_{ELCLKIN}$)	
Internal high-speed oscillator	o	Δ	o (Note 1)	o	Δ	o (Note 1)	Δ (Note 2)		−		−
External high-speed oscillator	Δ	o	Δ	Δ	o	Δ	Δ (Note 2)		−		−
External low-speed oscillator	Δ	Δ	Δ	Δ	Δ	Δ	o	Δ	o	Δ	−
CG	o			o			o		o		−
Processor core	o			−			o (Note 6)		−		−
Power Calculation Engine	o			o			o		−		−
IO port	o			o			o		o		o(Note 3)
SIO/UART	o			Δ			#(Note 5)		−		−
I2C/SIO	o			Δ			#(Note 5)		−		−
TMRB	o			Δ			o		−		−
TMR16A	o			Δ			o		−		−
WDT	o			Δ(Note 7)			o		−		−
ADC	o			Δ			#(Note 5)		−		−
ΔΣADC	o			o			#(Note 5)		−		−
RTC	o			o			o		o		−
LCD driver	o			o			o		o(Note 4)		−
Temperature sensor	o			o			o		o		−
LVD	o			o			o		o		o

o : Operation is available when in the target mode.

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

: To Stop the function is required when transiting to the target mode.

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

Note 1: When an external high-speed clock is used, an internal high-speed clock cannot be stopped.

Note 2: The high-speed oscillator (EHOSC and IHOSC) does not automatically stop when the operation mode shifted from NORMAL mode to SLOW mode. Set the CGOSCCR<XEN1> or <XEN2> to stop the oscillation. Before the transition from SLOW mode to NORMAL mode is made, set CGOSCCR<XEN1> or <XEN2> to start oscillation, since the high-speed oscillators (EHOSC, IHOSC) do not automatically start oscillation.

Note 3: This setting is depending on the CGSTBYCR<DRVE> setting.

Note 4: To use the LCD driver in SLEEP mode, the operation mode needs to be shifted from SLOW mode to SLEEP mode.

Note 5: Stop ADC, ΔΣADC, SIO/UART and I2C/SIO in SLOW mode.

Note 6: SysTick timer cannot be used in the SLOW mode.

Note 7: In IDLE mode, the watchdog timer cannot be cleared by the processor core.

6.6.6 Releasing the Low Power Consumption Mode

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

Details are shown in Table 6-7.

Table 6-7 Release Source in Each Mode

Releasesource		Low power consumption mode		
		IDLE	SLEEP	STOP
Interrupt	INT0 to 3 (Note1), INTLVD	o	o	o
	INTRTC	o	o	×
	Other interrupts	o	×	×
SysTick interrupt		o	×	×
Non-Maskable Interrupt (INTWDT)		o	×	×
RESET ($\overline{\text{RESET}}$ pin)		o	o	o

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

Note 1: When releasing from low power consumption mode by interrupting level mode, hold the level until the interrupt handling starts. If the level is changed before that, the correct interrupt handling cannnot be started.

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

- Release by interrupt request

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

- Release by SysTick interrupt

  SysTick interrupt can only be used in the IDLE mode.

- Release by Non-Maskable Interrupt (NMI)

  INTWDT can only be used in the IDLE mode.

- Release by reset

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

Refer to "Interrupts" for details.

6.6.7 Warm-up

Warm-up may be required for stable oscillation of internal oscillator in the mode transition.

In the mode transition from STOP to the 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 warm-up time. It is necessary to set a warm-up time in the CGOSCCR<WUPSEL2><WUPSEL1> and to set the warm-up time in the CGOSCCR<WUPT><WUPTL> before executing the instruction to enter the STOP/SLEEP mode.

Note: The external high-speed clock input ($f_{EHCLKIN}$) cannot be used as a warm-up clock.

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.

Table 6-8 shows whether the warm-up setting of each mode transition is required or not.

Table 6-8 Warm-up setting in mode transition

Mode transition	Warm-up setting
NORMAL → IDLE	Not required
NORMAL → SLEEP	(Note1)
NORMAL → SLOW	(Note1)
NORMAL → STOP	Not required
IDLE → NORMAL	Not required
SLEEP → NORMAL	Automatic warm-up with a clock selected by CGOSCCR<WUPSEL2>
SLEEP → SLOW	Not required
SLOW → NORMAL	(Note2)
SLOW → SLEEP	Not required
SLOW → STOP	Not required
STOP → NORMAL (Note 3)	Automatic warm-up with a clock selected by CGOSCCR<WUPSEL2>.
STOP → SLOW (Note 3)	Automatic warm-up with a clock selected by CGOSCCR<LOSCSEL>.

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.
Note 3: Set 200μs or more as the warming up time after releasing the STOP mode.

6.6.8 Clock Operations in Mode Transition

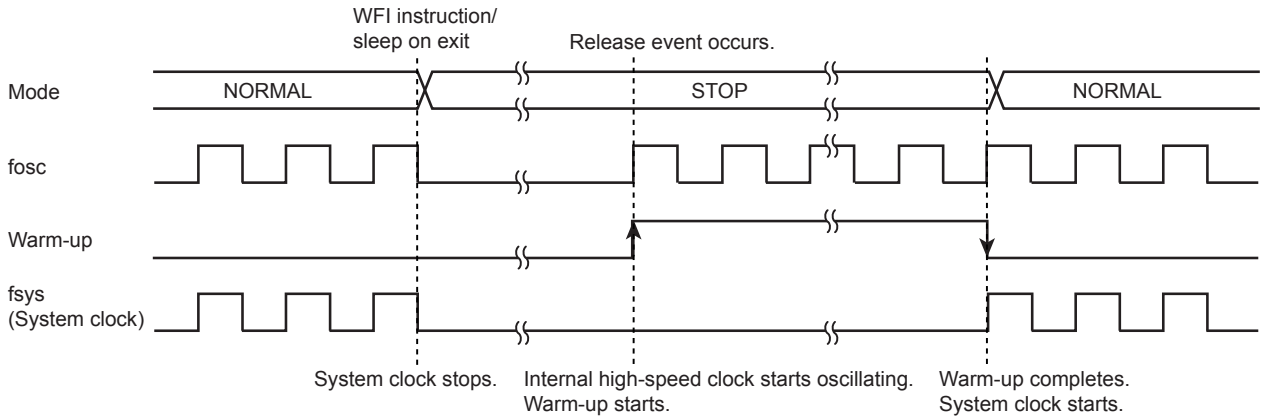
The clock operations in mode transition are described as follows.

6.6.8.1 Transition of operation modes: NORMAL → STOP → NORMAL

When returning to the NORMAL mode from the STOP mode, the warm-up is activated automatically. It is necessary to set the warm-up time to CGOSCCR<WUPT> and select clock-source that is same as CGOSCCR<OSCSEL> to <WUPSEL2><WUPSEL1> before entering the STOP mode.

Note: The operation mode cannot be shifted to STOP mode or SLEEP mode when using the external high-speed oscillation clock (f_{EHOSC}) and the external high-speed clock ($f_{EHCLKIN}$). Switch the clock to the internal high-speed oscillation clock (f_{IHOSC}) first.

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

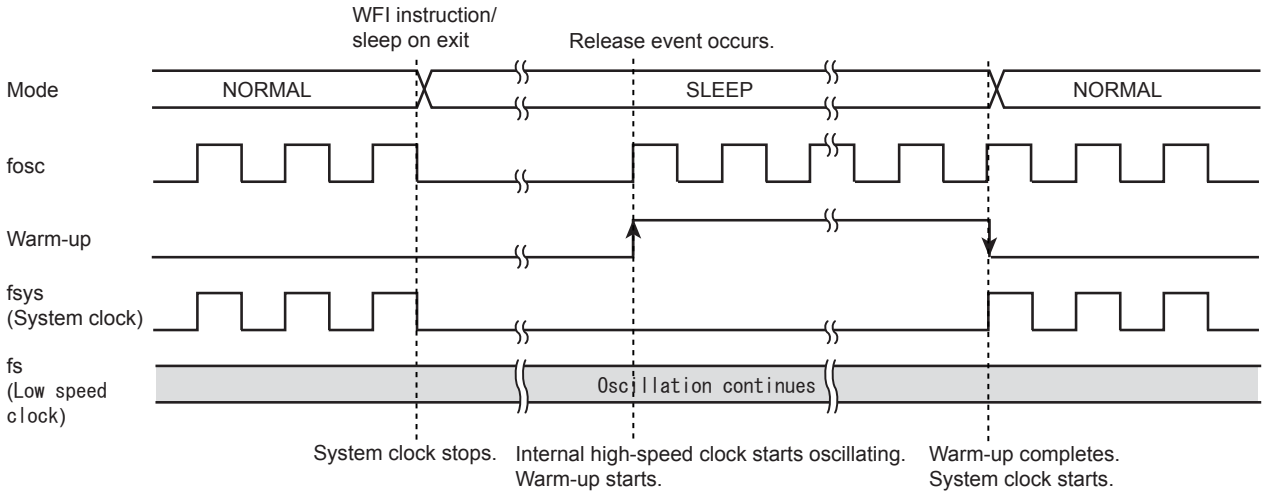


6.6.8.2 Transition of operation modes: NORMAL → SLEEP → NORMAL

When returning to the NORMAL mode from the SLEEP mode, the warm-up is activated automatically. It is necessary to set the warm-up time to CGOSCCR<WUPT> and select clock-source that is same as CGOSCCR<OSCSEL> to <WUPSEL2><WUPSEL1> before entering the SLEEP mode.

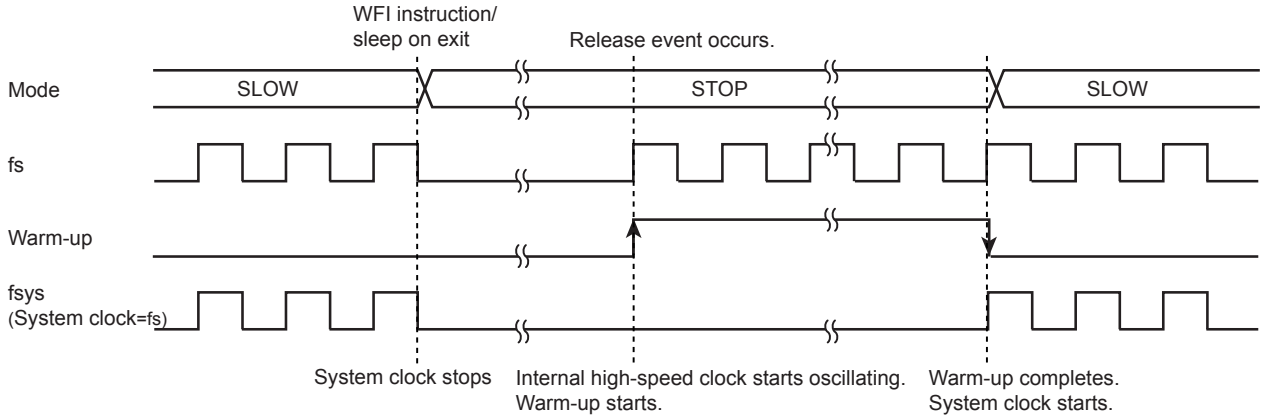
Note: The operation mode cannot be shifted to STOP mode or SLEEP mode when using the external high-speed oscillation clock (f_{EHOSC}) and the external high-speed clock ($f_{EHCLKIN}$). Switch the clock to the internal high-speed oscillation clock (f_{IHOSC}) first.

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



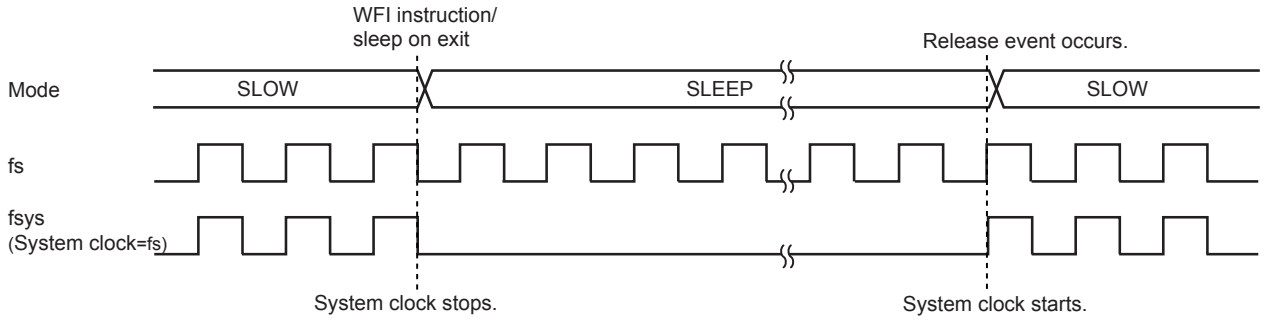
6.6.8.3 Transition of operation modes: SLOW → STOP → SLOW

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



6.6.8.4 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-M0 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-M0.

For detailed descriptions on each exception, refer to "Cortex-M0 Technical Reference Manual".

- Reset
- Non-Maskable Interrupt (NMI)
- Hard Fault
- SVCall (Supervisor Call)
- PendSV
- SysTick
- External Interrupt

7.1.2 Handling Flowchart

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

Each step is described later in this chapter.

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

7.1.2.1 Exception Request and Detection

(1) Exception occurrence

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

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

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

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

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	$\overline{\text{NMI}}$ pin or WDT
3	Hard Fault	−1	Fault that cannot activate because a higher-priority fault is being handled or it is disabled
4~10	Reserved	−	
11	SVCall	Configurable	System service call with SVC instruction
12~13	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: **External interrupts have different sources and numbers in each product. For details, see "7.5.1.5 List of Interrupt Sources".**

(3) Priority setting

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

The configuration of <PRI_n> is two bit, so the priority can be configured in the range from 0 to 3. The highest priority is "0". If multiple elements with the same priority exist, the smaller the number, the higher the priority becomes.

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

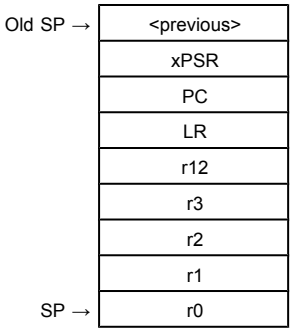
When an exception occurs, the CPU suspends the currently executing process and branches to the interrupt service routine. This is called "pre-emption".

(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

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

Prepare a vector table containing the top addresses of ISRs for each exception.

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

(3) Late-arriving

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

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

(4) Vector table

The vector table is configured as shown below.

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

Offset	Exception	Contents	Setting
0x00	Reset	Initial value of the main stack	Required
0x04	Reset	ISR address	Required
0x08	Non-Maskable Interrupt	ISR address	Required
0x0C	Hard Fault	ISR address	Required
0x10 to 0x28	Reserved		
0x2C	SVCall	ISR address	Optional
0x30 to 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 Interrupts".

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

7.1.2.4 Exception exit

(1) Execution after returning from an ISR

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

- Tail-chaining

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

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

- Returning to the last stacked ISR

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

- Returning to the previous program

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

(2) Exception exit sequence

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

- Pop eight registers

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

- Load current active interrupt number

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

- Select SP

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

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 "Low" to "High".

- Reset exception by WDT

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

- Reset exception by SYSRESETREQ

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

7.3 Non-Maskable Interrupts (NMI)

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

7.4 SysTick

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

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

Note: In this product, the systick timer counts based on fosc which is selected by the bits <OSCSEL> of the register CGOSCCR.

7.5 Interrupts

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

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

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

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

7.5.1 Interrupt Sources

7.5.1.1 Interrupt Route

Figure 7-1 shows an interrupt request route.

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

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

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

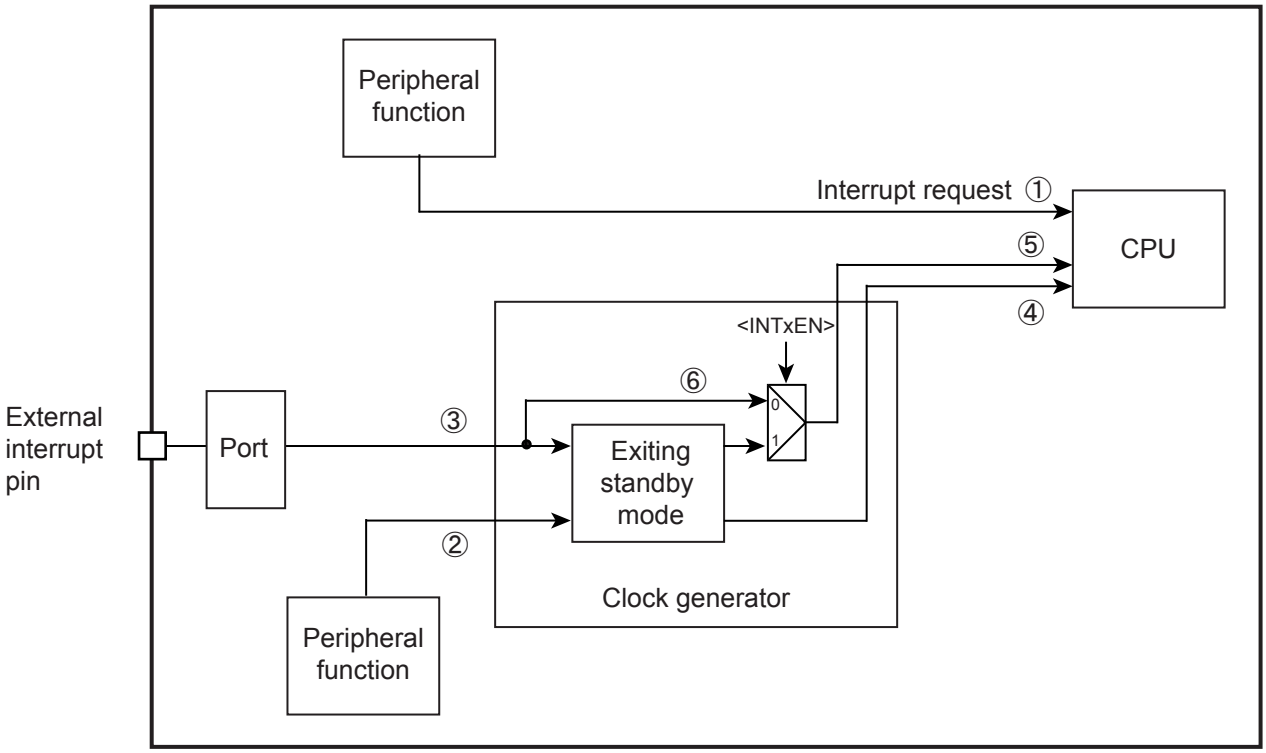


Figure 7-1 Interrupt Route

7.5.1.2 Generation

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

- From external pin

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

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

  See the chapter of each peripheral function for details.
- By setting Interrupt Set-Pending Register (forced pending)

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

7.5.1.3 Transmission

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

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

7.5.1.4 Precautions when using external interrupt pins

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

If input disabled (PxIE<PxmIE>="0"), inputs from external interrupt pins are "High". Also, if external interrupts are not used as a trigger to release standby (route 6 of "Figure 7-1 Interrupt Route"), input signals from the external interrupt pins are directly sent to the CPU. Since the CPU recognizes "High" input as an interrupt, interrupts occur if corresponding interrupts are enabled by the CPU as inputs are being disabled.

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

7.5.1.5 List of Interrupt Sources

Table 7-2 shows the list of interrupt sources.

Table 7-2 List of Interrupt Sources

No.	Interrupt Source		active level (Clearing standby)	CG interrupt mode control register
0	INTPCE	PCE		
1	INTDSAD0	ΔΣA/D conversion complete interrupt (unit0)		
2	INTDSAD1	ΔΣA/D conversion complete interrupt (unit1)		
3	INTDSAD2	ΔΣA/D conversion complete interrupt (unit2)		
4	INTLVD	LVD Interrupt	High edge	CGIMCGA
5	INT0	Interrupt pin 0	High/Low Edge/Level Selectable	
6	INT1	Interrupt pin 1		
7	INT2	Interrupt pin 2		
8	INT3	Interrupt pin 3		CGIMCGB
9	INTRX0	Serial reception (channel.0)		
10	INTTX0	Serial transmission (channel.0)		
11	INTRX1	Serial reception (channel.1)		
12	INTTX1	Serial transmission (channel.1)		
13	INTRX2	Serial reception (channel.2)		
14	INTTX2	Serial transmission (channel.2)		
15	INTRX3	Serial reception (channel.3)		
16	INTTX3	Serial transmission (channel.3)		
17	INTSBI	Serial bus interface		
18	INTTB0	16-bit TMRB match detection 0		
19	INTTB1	16-bit TMRB match detection 1		
20	INTCAP00	16-bit TMRB input capture 0 (channel.0)		
21	INTCAP01	16-bit TMRB input capture 1 (channel.0)		
22	INTCAP10	16-bit TMRB input capture 0 (channel.1)		
23	INTCAP11	16-bit TMRB input capture 1 (channel.1)		
24	INTT16A0	16-bit TMR16A match detection (channel.0)		
25	INTT16A1	16-bit TMR16A match detection (channel.1)		
26	INTT16A2	16-bit TMR16A match detection (channel.2)		
27	INTT16A3	16-bit TMR16A match detection (channel.3)		
28	INTT16A5	16-bit TMR16A match detection (channel.5)		
29	INTT16A6	16-bit TMR16A match detection (channel.6)		
30	INTRTC	Real time clock	Low edge	CGIMCGB
31	INTAD	A/D conversion completion interrupt		

7.5.1.6 Active level

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

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

If an interrupt source is used for clearing a standby mode, setting the relevant clock generator register is also required. Enable the CGIMCGx<INTxEN> bit and specify the active level in the CGIMCGx<EMCGx> bits. You must set the active level for interrupt requests from each peripheral function as shown in Table 7-2.

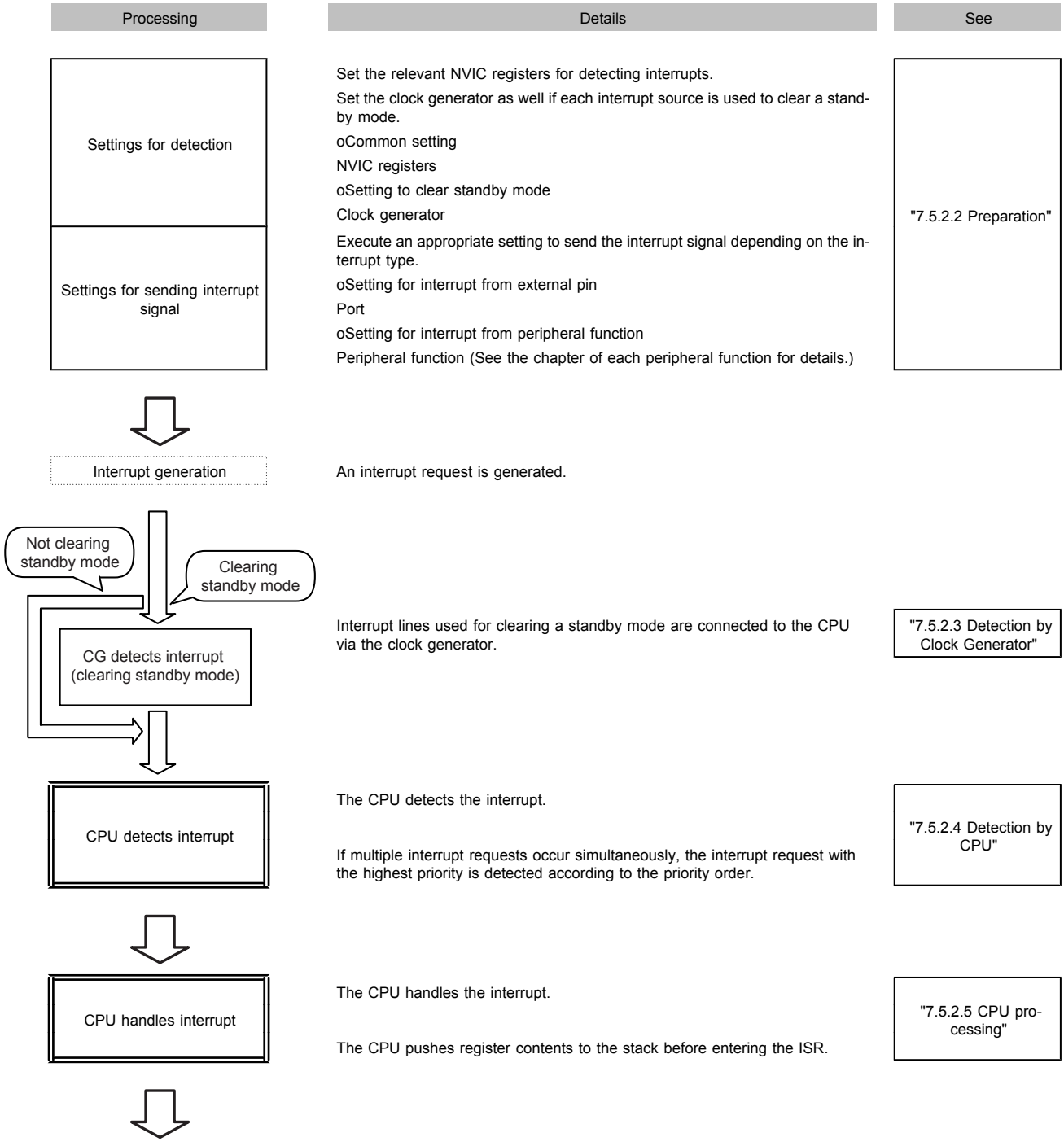
An interrupt request detected by the clock generator is notified to the CPU with a signal in "High" level.

7.5.2 Interrupt Handling

7.5.2.1 Flowchart

The following shows how an interrupt is handled.

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



Processing	Details	See
ISR execution	Program for the ISR. Clear the interrupt source if needed.	"7.5.2.6 Interrupt Service Routine (ISR)"
⇩		
Return to preceding program	Configure to return to the preceding program of the ISR.	

7.5.2.2 Preparation

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

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

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

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

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

(1) Disabling interrupt by CPU

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

Use "MSR" instruction to set this register.

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

(2) CPU registers setting

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

Each interrupt source is provided with two bits for assigning a priority level from 0 to 3. Priority level 0 is the highest priority level. If multiple sources have the same priority, the smallest-numbered interrupt source has the highest priority.

NVIC register		
<PRI_n>	←	"priority"

Note: "n" indicates the corresponding exceptions/interrupts.

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

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

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

Note: x: port number / m: corresponding bit / n: function register number

In modes other than STOP mode, setting PxIE to enable input enables the corresponding interrupt input regardless of the PxFR setting. Be careful not to enable interrupts that are not used. Also, be aware of the description of "7.5.1.4 Precautions when using external interrupt pins".

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

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

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

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

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

Note: **m: corresponding bit**

(6) Configuring the clock generator

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

Before enabling an interrupt, clear the corresponding interrupt request already held. This can avoid unexpected interrupt.To clear corresponding interrupt request, write a value corresponding to the interrupt to be used to the CGICRCG register.See "7.6.3.3 CGICRCG(CG Interrupt Request Clear Register)" for each value.

Interrupt requests from external pins can be used without setting the clock generator if they are not used for exiting a standby mode. However, an "High" pulse or "High"-level signal must be input so that the CPU can detect it as an interrupt request. Also, be aware of the description of"7.5.1.4 Precautions when using external interrupt pins".

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

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

(7) Enabling interrupt by CPU

Enable the interrupt by the CPU as shown below.

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

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

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

At the end, PRIMASK register is zero cleared.

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

Note: m : corresponding bit

7.5.2.3 Detection by Clock Generator

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

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

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

7.5.2.4 Detection by CPU

The CPU detects an interrupt request with the highest priority.

7.5.2.5 CPU processing

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

7.5.2.6 Interrupt Service Routine (ISR)

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

(1) Pushing during ISR

An ISR normally pushes register contents to the stack and handles an interrupt as required. The Cor-tex-M0 core automatically pushes the contents of PC, PSR, r0-r3, r12 and LR to the stack. No extra programming is required for them. Push the contents of other registers if needed.

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

(2) Clearing an interrupt source

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

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

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

7.6 Exception/Interrupt-Related Registers

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

7.6.1 Register List

NVIC registers — Base Address = 0xE000_E000

Register name	Address
SysTick Control and Status Register	0x0010
SysTick Reload Value Register	0x0014
SysTick Current Value Register	0x0018
SysTick Calibration Value Register	0x001C
Interrupt Set-Enable Register	0x0100
Interrupt Clear-Enable Register	0x0180
Interrupt Set-Pending Register	0x0200
Interrupt Clear-Pending Register	0x0280
Interrupt Priority Register	0x0400 ~ 0x0430
Application Interrupt and Reset Control Register	0x0D0C
System Handler Priority Register	0x0D1C, 0x0D20
System Handler Control and State Register	0x0D24

Clock generator registers — Base Address = 0x400F_3000

Register name		Address
CG Interrupt Mode Control Register A	CGIMCGA	0x0020
CG Interrupt Mode Control Register B	CGIMCGB	0x0024
CG Interrupt Request Clear Register	CGICRCG	0x0014
Reset Flag Register	CGRSTFLG	0x001C

7.6.2 NVIC Registers

7.6.2.1 SysTick Control and Status Register

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

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

Note: In this product, the fosc (which is selected by the bits <OSCSEL> of the register CGOSCCR) is used as the external reference clock.

7.6.2.2 SysTick Reload Value Register

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

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

7.6.2.3 SysTick Current Value Register

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

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

7.6.2.4 SysTick Calibration Value Register

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

Bit	Bit Symbol	Type	Function
31	NOREF	R	0: Reference clock provided 1: No reference clock
30	SKEW	R	0: Calibration value is 10 ms. 1: Calibration value is not 10 ms.
29-24	−	R	Read as 0.
23-0	TENMS	R	Calibration value (Note)

Note: TMPM061FWFG does not prepare the calibration value.

7.6.2.5 Interrupt Set-Enable Register

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

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

Note: For descriptions of interrupts and interrupt numbers, see Section "7.5.1.5 List of Interrupt Sources".

7.6.2.6 Interrupt Clear-Enable Register

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

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

Note: For descriptions of interrupts and interrupt numbers, see Section "7.5.1.5 List of Interrupt Sources".

7.6.2.7 Interrupt Set-Pending Register

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

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

Note: For descriptions of interrupts and interrupt numbers, see Section "7.5.1.5 List of Interrupt Sources".

7.6.2.8 Interrupt Clear-Pending Register

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

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

Note: For descriptions of interrupts and interrupt numbers, see Section "7.5.1.5 List of Interrupt Sources".

7.6.2.9 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

Cortex-M0 core uses two bits for assigning a priority.

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

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

Bit	Bit Symbol	Type	Function
31-30	PRI_3	R/W	Priority of interrupt number 3
29-24	−	R	Read as 0.
23-22	PRI_2	R/W	Priority of interrupt number 2
21-16	−	R	Read as 0.
15-14	PRI_1	R/W	Priority of interrupt number 1
13-8	−	R	Read as 0.
7-6	PRI_0	R/W	Priority of interrupt number 0
5-0	−	R	Read as 0.

7.6.2.10 Application Interrupt and Reset Control Register

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

Bit	Bit Symbol	Type	Function
31-16	VECTKEY (Write)/ VECTKEY-STAT(Read)	R/W	Register key [Write] Writing to this register requires 0x5FA in the <VECTKEY> field. [Read] Read as 0xFA05.
15	ENDIANESS	R/W	Endianness bit:(Note1) 1: big endian 0: little endian
14-3	−	R	Read as 0.
2	SYSRESET REQ	R/W	System Reset Request. 1=CPU outputs a SYSRESETREQ signal. (note2)
1	VECTCLR ACTIVE	R/W	Clear active vector bit 1: clear all state information for active NMI, fault, and interrupts 0: do not clear. This bit self-clears. It is the responsibility of the application to reinitialize the stack.
0	−	R	Read as 0.

Note 1: This product can be used as the little-endian memory format only.

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

7.6.2.11 System Handler Priority Register

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

	31 24	23 16	15 8	7 0
0xE000_ED1C	PRI_11 (SVCall)	PRI_10	PRI_9	PRI_8
0xE000_ED20	PRI_15 (SysTick)	PRI_14 (PendSV)	PRI_13	PRI_12

Cortex-M0 core uses two bits for assigning a priority.

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

	31	30	29	28	27	26	25	24
bit symbol	PRI_15		-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	PRI_14		-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	PRI_13		-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PRI_12		-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-30	PRI_15	R/W	Priority of SysTick
29-24	-	R	Read as 0.
23-22	PRI_14	R/W	Priority of PendSV
21-16	-	R	Read as 0.
15-14	PRI_13	R/W	Reserved
13-8	-	R	Read as 0.
7-6	PRI_12	R/W	Reserved
5-0	-	R	Read as 0.

7.6.2.12 System Handler Control and State Register

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

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as 0.
15	SVCALL PENDED	R/W	SVCall 0: Not pended 1: Pended
14-0	-	R	Read as 0.

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

7.6.3 Clock generator registers

7.6.3.1 CGIMCGA(CG Interrupt Mode Control Register A)

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

Bit	Bit Symbol	Type	Function
31	−	R	Read as 0.
30-28	EMCG3[2:0]	R/W	active level setting of INT2 standby clear request. (101~111: setting prohibited) 000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edges
27-26	EMST3[1:0]	R	active level of INT2 standby clear request 00: − 01: Rising edge 10: Falling edge 11: Both edges
25	−	R	Reads as undefined.
24	INT3EN	R/W	INT2 clear input 0:Disable 1: Enable
23	−	R	Read as 0.
22-20	EMCG2[2:0]	R/W	active level setting of INT1 standby clear request. (101~111: setting prohibited) 000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edges
19-18	EMST2[1:0]	R	active level of INT1 standby clear request 00: − 01: Rising edge 10: Falling edge 11: Both edges
17	−	R	Reads as undefined.
16	INT2EN	R/W	INT1 clear input 0:Disable 1: Enable
15	−	R	Read as 0.

Bit	Bit Symbol	Type	Function
14-12	EMCG1[2:0]	R/W	active level setting of INT0 standby clear request. (101~111: setting prohibited) 000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edges
11-10	EMST1[1:0]	R	active level of INT0 standby clear request 00: − 01: Rising edge 10: Falling edge 11: Both edges
9	−	R	Reads as undefined.
8	INT1EN	R/W	INT0 clear input 0:Disable 1: Enable
7	−	R	Read as 0.
6-4	EMCG0[2:0]	R/W	active level setting of INTLVD standby clear request. (Note) Set it as shown below. 011: Rising edge
3-2	EMST0[1:0]	R	active level of INTLVD standby clear request 00: − 01: Rising edge 10: Falling edge 11: Both edges
1	−	R	Reads as undefined.
0	INT0EN	R/W	INTLVD clear input 0:Disable 1: Enable

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

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

7.6.3.2 CGIMCGB(CG Interrupt Mode Control Register B)

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

Bit	Bit Symbol	Type	Function
31	−	R	Read as 0.

Bit	Bit Symbol	Type	Function
30-28	−	R/W	Write any value.
27-25	−	R	Read as 0.
24	−	R/W	Write as 0.
23	−	R	Read as 0.
22-20	−	R/W	Write any value.
19-17	−	R	Read as 0.
16	−	R/W	Write as 0.
15	−	R	Read as 0.
14-12	EMCG5[2:0]	R/W	active level setting of INTRTC standby clear request. (Note) Set it as shown below. 010: Falling edge
11-10	EMST5[1:0]	R	active level of INTRTC standby clear request 00: − 01: Rising edge 10: Falling edge 11: Both edges
9	−	R	Reads as undefined.
8	INT5EN	R/W	INTRTC clear input 0:Disable 1: Enable
7	−	R	Read as 0.
6-4	EMCG4[2:0]	R/W	active level setting of INT3 standby clear request. (101~111: setting prohibited) 000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edges
3-2	EMST4[1:0]	R	active level of INT3 standby clear request 00: − 01: Rising edge 10: Falling edge 11: Both edges
1	−	R	Reads as undefined.
0	INT4EN	R/W	INT3 clear input 0:Disable 1: Enable

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

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

7.6.3.3 CGICRCG(CG Interrupt Request Clear Register)

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

Bit	Bit Symbol	Type	Function
31-5	−	R	Read as 0.
4-0	ICRCG[4:0]	W	Clear interrupt requests. 0_0000: INTLVD 0_0001: INT0 0_0010: INT1 0_0011: INT2 0_0100: INT3 0_0101: INTRTC 0_0110 to 1_1111: setting prohibited. Read as 0.

7.6.3.4 CGRSTFLG (Reset Flag Register)

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

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

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

Note 2: This register is not cleared automatically. Write "0" to clear the register.

8. Input / Output port

8.1 Port Function

8.1.1 Function List

The ports are also used as input / output pins for built-in peripheral functions.

Table 8-1 show list of port function.

Table 8-1 List of Port Function (Port A to Port K)

Port	Pin name	Input / Out-put	Schmitt Input	Noise filter	Function pin
Port A					
	PA0	I/O	o	-	SEG0
	PA1	I/O	o	-	SEG1
	PA2	I/O	o	-	SEG2
	PA3	I/O	o	-	SEG3
	PA4	I/O	o	-	SEG4
	PA5	I/O	o	-	SEG5
	PA6	I/O	o	-	SEG6
	PA7	I/O	o	-	SEG7
Port B					
	PB0	I/O	o	-	SEG8
	PB1	I/O	o	-	SEG9
	PB2	I/O	o	-	SEG10
	PB3	I/O	o	-	SEG11
	PB4	I/O	o	-	SEG12
	PB5	I/O	o	-	SEG13
	PB6	I/O	o	-	SEG14
	PB7	I/O	o	-	SEG15
Port C					
	PC0	I/O	o	-	SEG16
	PC1	I/O	o	-	SEG17
	PC2	I/O	o	-	SEG18
	PC3	I/O	o	-	SEG19
	PC4	I/O	o	-	SEG20
	PC5	I/O	o	-	SEG21
	PC6	I/O	o	-	SEG22
	PC7	I/O	o	-	SEG23
Port D					

Table 8-1 List of Port Function (Port A to Port K)

Port	Pin name	Input / Output	Schmitt Input	Noise filter	Function pin
	PD0	I/O	o	−	SEG24
	PD1	I/O	o	−	SEG25
	PD2	I/O	o	−	SEG26
	PD3	I/O	o	−	SEG27
	PD4	I/O	o	−	SEG28
	PD5	I/O	o	−	SEG29
	PD6	I/O	o	−	SEG30
	PD7	I/O	o	−	SEG31
Port E					
	PE0	I/O	o	−	SEG32
	PE1	I/O	o	−	SEG33
	PE2	I/O	o	−	SEG34/T16A6OUT
	PE3	I/O	o	−	SEG35/SCLK31/$\overline{\text{CTS31}}$
	PE4	I/O	o	−	SEG36/RXD31
	PE5	I/O	o	−	SEG37/TXD31
	PE6	I/O	o	−	SEG38/SWCLK
	PE7	I/O	o	−	SEG39/SWDIO
Port F					
	PF0	I/O	o	−	AIN0
	PF1	I/O	o	o	AIN1/INT0
Port G					
	PG0	I/O	o	o	TB0OUT/INT2
Port H					
	PH0	I/O	o	−	TXD0/IROUT0
	PH1	I/O	o	−	RXD0
	PH2	I/O	o	−	SCLK0/$\overline{\text{CTS0}}$/T16A0OUT
	PH3	I/O	o	−	TXD1/IROUT1
	PH4	I/O	o	−	RXD1
	PH5	I/O	o	−	SCLK1/$\overline{\text{CTS1}}$/T16A1OUT
Port I					
	PI0	I/O	o	−	TXD2/IROUT2
	PI1	I/O	o	−	RXD2
	PI2	I/O	o	−	SCLK2/$\overline{\text{CTS2}}$/T16A2OUT
	PI3	I/O	o	−	TB0IN
	PI4	I/O	o	−	TXD30
	PI5	I/O	o	−	RXD30
	PI6	I/O	o	−	SCLK30/$\overline{\text{CTS30}}$/T16A5OUT
Port J					
	PJ0	I/O	o	−	SDA0/SO0
	PJ1	I/O	o	−	SCL0/SI0
	PJ2	I/O	o	o	SCK0/INT1
	PJ3	I/O	o	−	RTCOUT
	PJ4	I/O	o	−	T16A3OUT/SCOUT
	PJ5	I	o	−	TB1IN/XTCLKIN
Port K					

Table 8-1 List of Port Function (Port A to Port K)

Port	Pin name	Input / Output	Schmitt Input	Noise filter	Function pin
	PK0	I/O	o	o	INT3
	PK1	I/O	o	-	TB1OUT

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

8.2 Port Register General Description

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

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

In the following examination, the 8 bits port are described. About the configuration of the port and the initial value in the each port, please refer to each port's section.

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

8.2.1 PxDATA : Port x data register

This register reads / writes port data.

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	Px7	Px6	Px5	Px4	Px3	Px2	Px1	Px0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0"
7-0	Px7-Px0	R/W	Port x data register

8.2.2 PxCR : Port x output control register

This register controls output.

To enable / disable input with PxIE register.

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	Px7C	Px6C	Px5C	Px4C	Px3C	Pxx2C	Px1C	Px0C
After reset	0	0	0	0	0	0	1	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7-0	Px7C-Px0C	R/W	Output 0: Disable 1: Enable

8.2.3 PxFRn : Port x function register n

This register sets the function.

The assigned function can be enabled by setting "1". This register exists for the each function assigned to the port. In case of having some function, only one function can be enabled.

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	Px7Fn	Px6Fn	Px5Fn	Px4Fn	Px3Fn	Px2Fn	Px1Fn	Px0Fn
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7-0	Px7Fn-Px0Fn	R/W	0: PORT 1: Function

8.2.4 PxOD : Port x open-drain control register

This register controls programmable open-drain outputs.

Programmable open-drain outputs are set with PxOD. When output data is "1", output buffer is disabled and becomes a pseudo-open-drain output.

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	Px7OD	Px6OD	Px5OD	Px4OD	Px3OD	Px2OD	Px1OD	Px0OD
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7-0	Px7OD-Px0OD	R/W	0: Push-pull output 1: Open-drain output

8.2.5 PxPUP : Port x pull-up control register

This register controls programmable pull-ups.

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	Px7UP	Px6UP	Px5UP	Px4UP	Px3UP	Px2UP	Px1UP	Px0UP
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7-0	Px7UP-Px0UP	R/W	Pull-up 0: Disable 1: Enable

8.2.6 PxPDN : Port x pull-down control register

This register controls programmable pull-downs.

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	Px7DN	Px6DN	Px5DN	Px4DN	Px3DN	Px2DN	Px1DN	Px0DN
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7-0	Px7DN-Px0DN	R/W	Pull-down 0: Disable 1: Enable

8.2.7 PxIE : Port x input control register

This register controls inputs.

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	Px7IE	Px6IE	Px5IE	Px4IE	Px3IE	Px2IE	Px1IE	Px0IE
After reset	0	0	0	0	0	0	1	1

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7-0	Px7IE-Px0IE	R/W	Input 0: Disable 1: Enable

8.3 Register List

Port	Base Address
Port A	0x400C_0000
Port B	0x400C_0100
Port C	0x400C_0200
Port D	0x400C_0300
Port E	0x400C_0400
Port F	0x400C_0500
Port G	0x440C_0600
Port H	0x400C_0700
Port I	0x400C_0800
Port J	0x400C_0900
Port K	0x400C_0A00

Register Name	Address (Base+)	Port A	Port B	Port C	Port D	Port E	Port F
Data register	0x0000	PADATA	PBDATA	PCDATA	PDDATA	PEDATA	PFDATA
Output control register	0x0004	PACR	PBCR	PCCR	PDCR	PECR	PFCR
Function register 1	0x0008	PAFR1	PBFR1	PCFR1	PDFR1	PEFR1	PFFR1
Function register 2	0x000C	-	-	-	-	PEFR2	-
Function register 3	0x0010	-	-	-	-	PEFR3	-
Open-drain control register	0x0028	PAOD	PBOD	PCOD	PDOD	PEOD	PFOD
Pull-up control register	0x002C	PAPUP	PBPUP	PCPUP	PDPUP	PEPUP	PFPUP
Pull-down control register	0x0030	PAPDN	PBPDN	PCPDN	PDPDN	PEPDN	PFPDN
Input control register	0x0038	PAIE	PBIE	PCIE	PDIE	PEIE	PFIE

Register Name	Address (Base+)	Port G	Port H	Port I	Port J	Port K
Data register	0x0000	PGDATA	PHDATA	PIDATA	PJDATA	PKDATA
Output control register	0x0004	PGCR	PHCR	PICR	PJCR	PKCR
Function register 1	0x0008	PGFR1	PHFR1	PIFR1	PJFR1	PKFR1
Function register 2	0x000C	PGFR2	PHFR2	PIFR2	PJFR2	-
Function register 3	0x0010	-	PHFR3	PIFR3	-	-
Open-drain control register	0x0028	PGOD	PHOD	PIOD	PJOD	PKOD
Pull-up control register	0x002C	PGPUP	PHPUP	PIPUP	PJPUP	PKPUP
Pull-down control register	0x0030	PGPDN	PHPDN	PIPDN	PJPDN	PKPDN
Input control register	0x0038	PGIE	PHIE	PIIE	PJIE	PKIE

8.4 Function details

This chapter describes the configuration of register, initial value and the function assigned by a function register.

The bit 31 to 8 of the register and the bit hatched is read as "0". Writing to them does not influence.

8.4.1 Port A

	7	6	5	4	3	2	1	0
PADATA	0	0	0	0	0	0	0	0
PAIE	0	0	0	0	0	0	0	0
PACR	0	0	0	0	0	0	0	0
PAPUP	0	0	0	0	0	0	0	0
PAPDN	0	0	0	0	0	0	0	0
PAOD	0	0	0	0	0	0	0	0
PAFR1	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0
	0	0	0	0	0	0	0	0

Note: When port A is used as LCD segment outputs, set "1" to PAFR1 and clear other registers to "0".

8.4.2 Port B

	7	6	5	4	3	2	1	0
PBDATA	0	0	0	0	0	0	0	0
PBIE	0	0	0	0	0	0	0	0
PBCR	0	0	0	0	0	0	0	0
PBPUP	0	0	0	0	0	0	0	0
PBPDN	0	0	0	0	0	0	0	0
PBOD	0	0	0	0	0	0	0	0
PBFR1	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8
	0	0	0	0	0	0	0	0

Note: When port B is used as LCD segment outputs, set "1" to PBFR1 and clear other registers to "0".

8.4.3 Port C

	7	6	5	4	3	2	1	0
PCDATA	0	0	0	0	0	0	0	0
PCIE	0	0	0	0	0	0	0	0
PCCR	0	0	0	0	0	0	0	0
PCPUP	0	0	0	0	0	0	0	0
PCPDN	0	0	0	0	0	0	0	0
PCOD	0	0	0	0	0	0	0	0
PCFR1	SEG23	SEG22	SEG21	SEG20	SEG19	SEG18	SEG17	SEG16
	0	0	0	0	0	0	0	0

Note: When port C is used as LCD segment outputs, set "1" to PCFR1 and clear other registers to "0".

8.4.4 Port D

	7	6	5	4	3	2	1	0
PDDATA	0	0	0	0	0	0	0	0
PDIE	0	0	0	0	0	0	0	0
PDCR	0	0	0	0	0	0	0	0
PDPUP	0	0	0	0	0	0	0	0
PDPDN	0	0	0	0	0	0	0	0
PDOD	0	0	0	0	0	0	0	0
PDFR1	SEG31	SEG30	SEG29	SEG28	SEG27	SEG26	SEG25	SEG24
	0	0	0	0	0	0	0	0

Note: When port D is used as LCD segment outputs, set "1" to PDFR1 and clear other registers to "0".

8.4.5 Port E

	7	6	5	4	3	2	1	0
PEDATA	0	0	0	0	0	0	0	0
PEIE	1	1	0	0	0	0	1 (Note3)	1 (Note3)
PECR	1	0	0	0	0	0	1 (Note3)	0
PEPUP	1	0	0	0	0	0	1 (Note3)	0
PEPDN	0	1	0	0	0	0	0	1 (Note3)
PEOD	0	0	0	0	0	0	0	0
PEFR1	SEG39	SEG38	SEG37	SEG36	SEG35	SEG34	SEG33	SEG32
	0	0	0	0	0	0	0	0
PEFR2	SWDIO	SWCLK	TXD31	RXD31	SCLK31	T16A6OUT	-	-
	1	1	0	0	0	0	1 (Note2)	1 (Note2)
PEFR3					$\overline{\text{CTS31}}$			
					0			

Note 1: When port E is used as LCD segment outputs, set "1" to PEFR1 and clear other registers to "0".
Note 2: Write as "0".
Note 3: Change setting if necessary.

8.4.6 Port F

When this port is used as analog input, the value of all register set to the initial value.

	7	6	5	4	3	2	1	0
PFDATA							0	0
PFIE							0	0
PFCR							0	0
PFPUP							0	0
PFPDN							0	0
PFOD							0	0
PFFR1							INT0	
							0	

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

8.4.7 Port G

	7	6	5	4	3	2	1	0
PGDATA								0
PGIE								0
PGCR								0
PGPUP								0
PGPDN								0
PGOD								0
PGFR1								TB0OUT
								0
PGFR2								INT2
								0

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

8.4.8 Port H

	7	6	5	4	3	2	1	0
PHDATA			0	0	0	0	0	0
PHIE			0	0	0	0	0	0
PHCR			0	0	0	0	0	0
PHPUP			0	0	0	0	0	0
PHPDN			0	0	0	0	0	0
PHOD			0	0	0	0	0	0
PHFR1			SCLK1	RXD1	TXD1	SCLK0	RXD0	TXD0
			0	0	0	0	0	0
PHFR2			$\overline{\text{CTS1}}$		IROUT1	$\overline{\text{CTS0}}$		IROUT0
			0		0	0		0
PHFR3			T16A1OUT			T16A0OUT		
			0			0		

Note: PH3 is shared with $\overline{\text{DBGEN}}$ function. It is enabled to be input and pulled-up while $\overline{\text{RESET}}$ pin is low.

8.4.9 Port I

	7	6	5	4	3	2	1	0
PIDATA		0	0	0	0	0	0	0
PIIE		0	0	0	0	0	0	0
PICR		0	0	0	0	0	0	0
PIPUP		0	0	0	0	0	0	0
PIPDN		0	0	0	0	0	0	0
PIOD		0	0	0	0	0	0	0
PIFR1		SCLK30	RXD30	TXD30	TB0IN	SCLK2	RXD2	TXD2
		0	0	0	0	0	0	0
PIFR2		$\overline{\text{CTS30}}$				$\overline{\text{CTS2}}$		IROUT2
		0				0		0
PIFR3		T16A5OUT				T16A2OUT		
		0				0		

Note 1: PI0 is enabled to be input and pulled-up while $\overline{\text{RESET}}$ pin is low.

Note 2: Only when input is enabled, PI5 and PI6 tolerate 5V inputs.
Note that these pins cannot be pulled up over the power supply voltage when using as open-drain output.

8.4.10 Port J

	7	6	5	4	3	2	1	0
PJDATA			0	0	0	0	0	0
PJIE			0	0	0	0	0	0
PJCR				0	0	0	0	0
PJPUP				0	0	0	0	0
PJPDN				0	0	0	0	0
PJOD				0	0	0	0	0
PJFR1			TB1IN	T16A3OUT	RTCOUT	SCK0	SCL0/SI0	SDA0/SO0
			0	0	0	0	0	0
PJFR2			XTCLKIN	SCOUT		INT1		
			0	0		0		

Note 1: PJ0 is shared with $\overline{\text{BOOT}}$ function. It is enabled to be input and pulled-up while $\overline{\text{RESET}}$ pin is low.

Note 2: PJ5 is used as only input port.

Note 3: In modes other than STOP mode, interrupt input is enabled regardless of the PxFRn register setting if input is enabled in PxIE. Make sure to disable unused interrupts when programming the device.

Note 4: Only when input is enabled, PJ2 tolerates 5V input.
Note that these pins cannot be pulled up over the power supply voltage when using as open-drain output.

8.4.11 Port K

When bleeder resistor is connected with this port, the value of all register set to the initial value.

	7	6	5	4	3	2	1	0
PKDATA							0	0
PKIE							0	0
PKCR							0	0
PKPUP							0	0
PKPDN							0	0
PKOD							0	0
PKFR1							TB1OUT	INT3
							0	0

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

8.5 Block Diagrams of Ports

8.5.1 Port Type

The ports are classified as shown below. Please refer to the following pages for the block diagrams of each port type. A dotted box in the figure indicates the part of the equivalent circuit described in the "Block diagrams of ports".

Port types used in each pin are described in Chapter 8.6.

Table 8-2 Function List

Type	Function					Pull-up	Pull-down	Analog
	I/O	Enable signals exist/not exist		Operation in the STOP mode				
		Input	Output	Input	Output			
FT1	I/O	−	−	−	−	R	R	−
FT2	I/O	−	o	o	o	EnR	EnR	−
FT3	I/O	−	o	−	−	R	R	−
FT4	Input (int)	−	−	o	−	R	R	−
FT5	Input	−	−	−	−	R	R	o
FT6	I/O	o	o	−	o	R	R	−
FT7	Input	−	−	o	−	R	R	−
FT8	I/O	−	−	−	o	R	R	−
FT9	I/O	−	o	−	o	R	R	−
FT10	Output(LCD)	−	−	−	o	R	R	−

int: Interrupt input
−: No exist/Disabled
o: Exist/Enabled
R: Forced disable during reset
EnR: Forced enable during reset

8.5.2 Type FT1

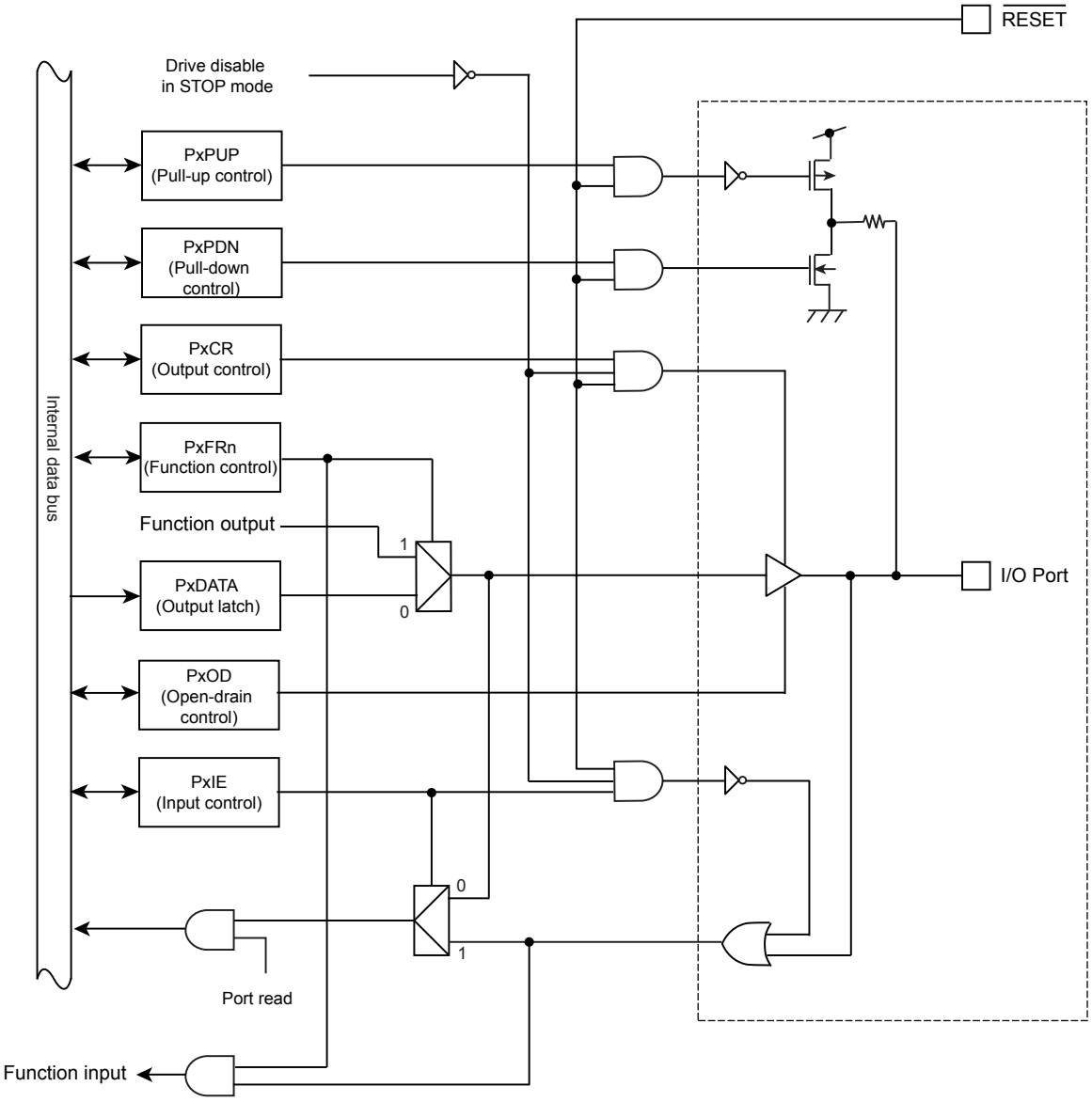


Figure 8-1 Port Type FT1

8.5.3 Type FT2

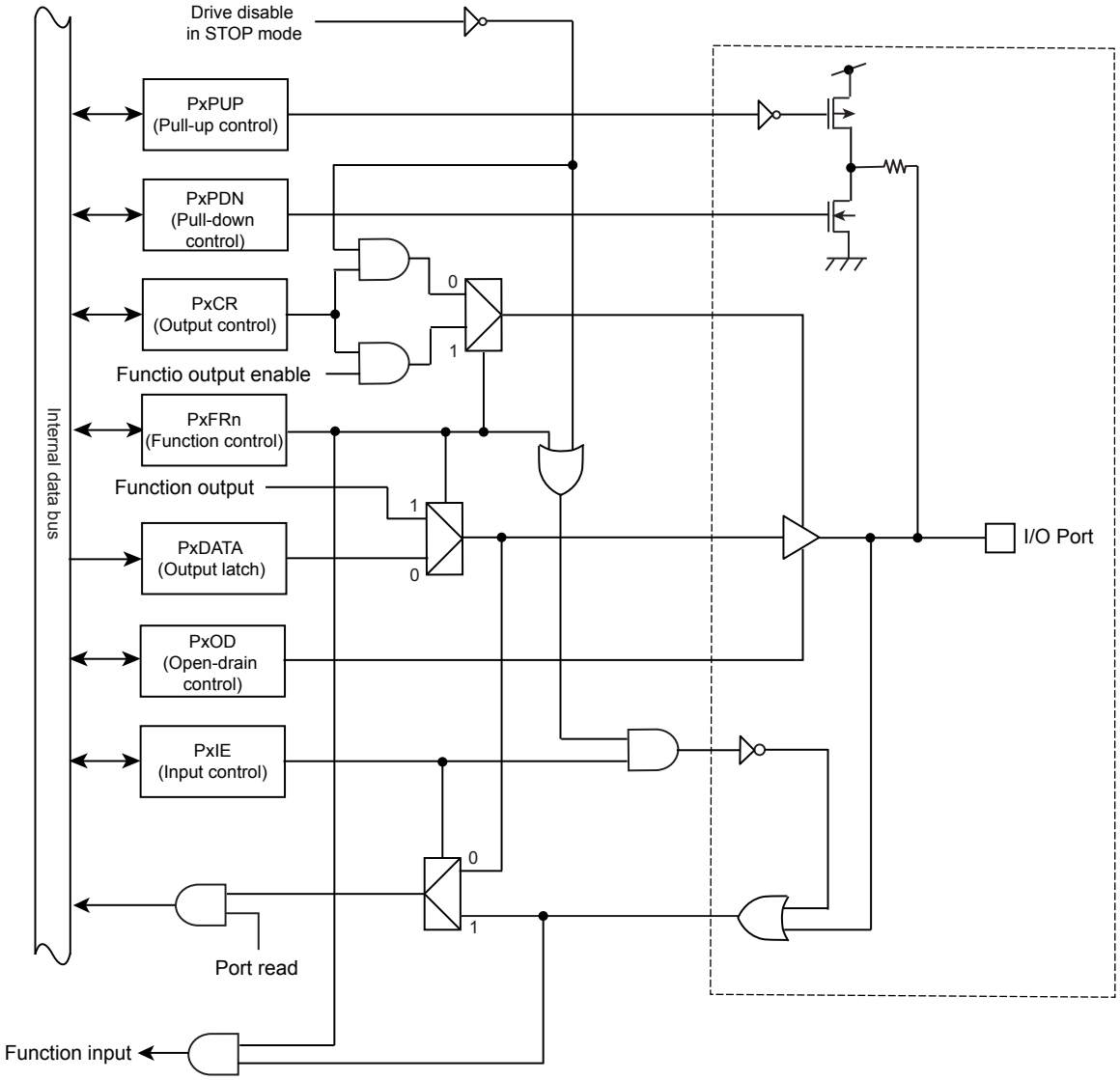


Figure 8-2 Port Type FT2

8.5.4 Type FT3

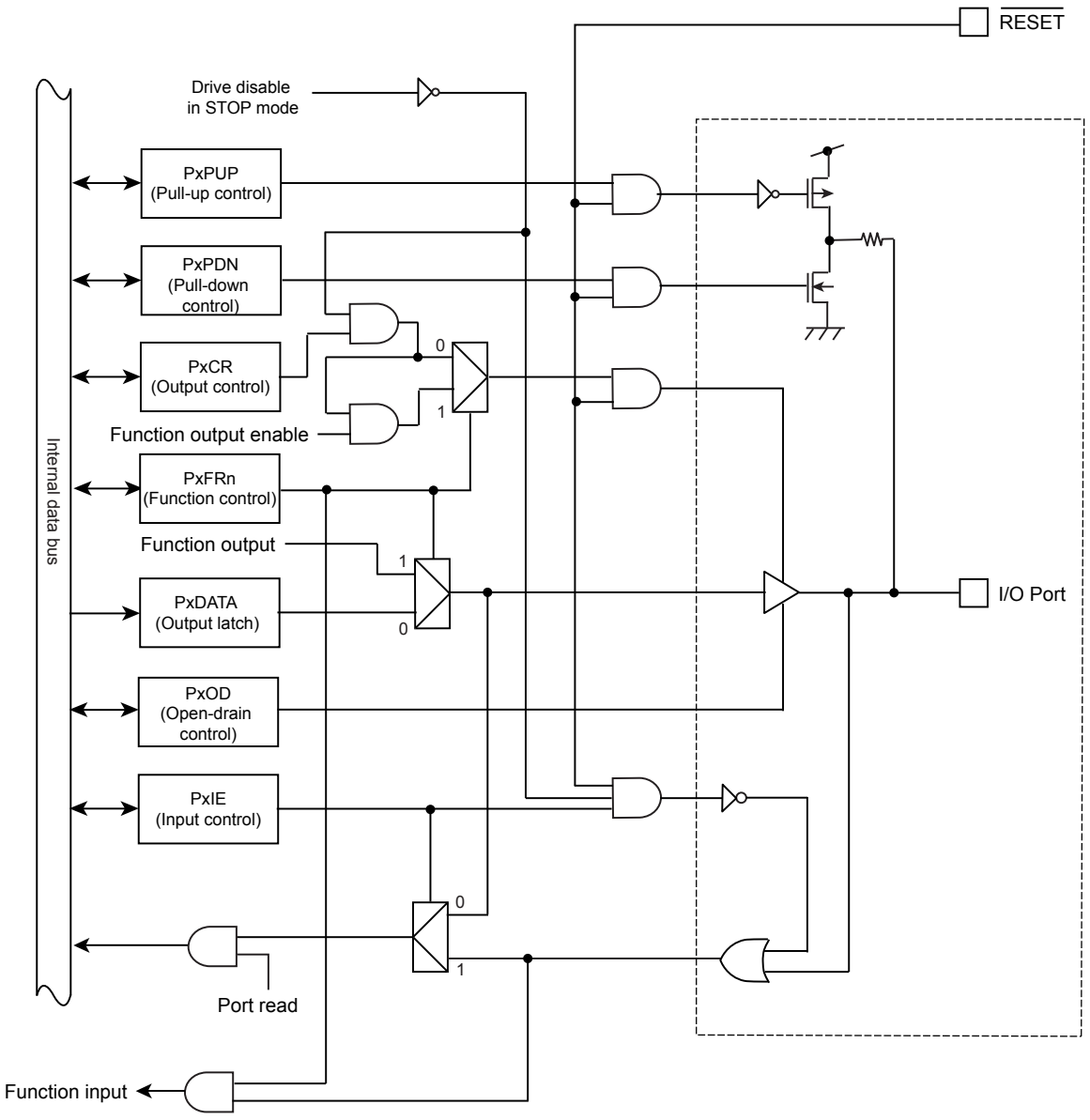


Figure 8-3 Port Type FT3

8.5.5 Type FT4

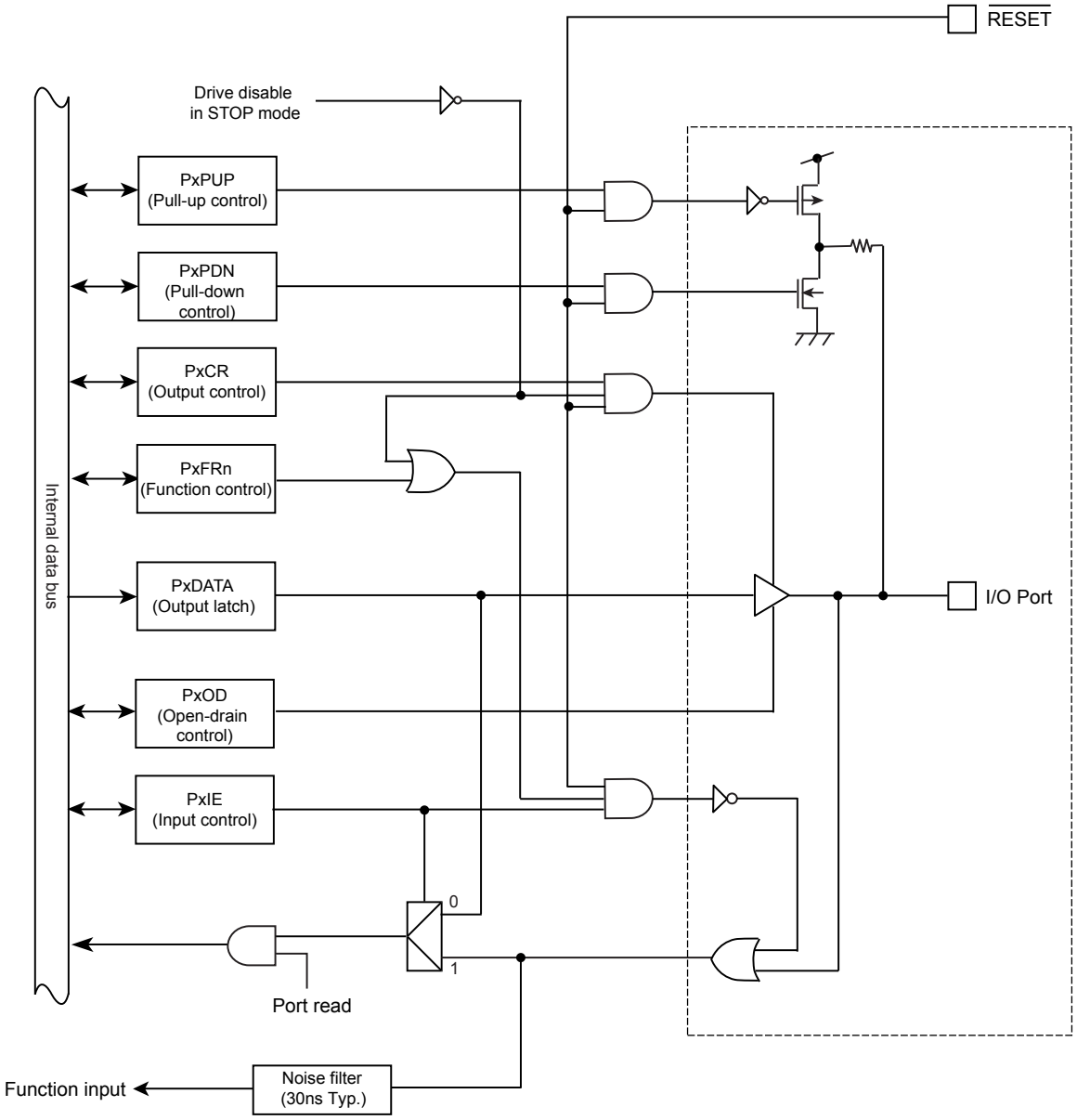


Figure 8-4 Port Type FT4

8.5.6 Type FT5

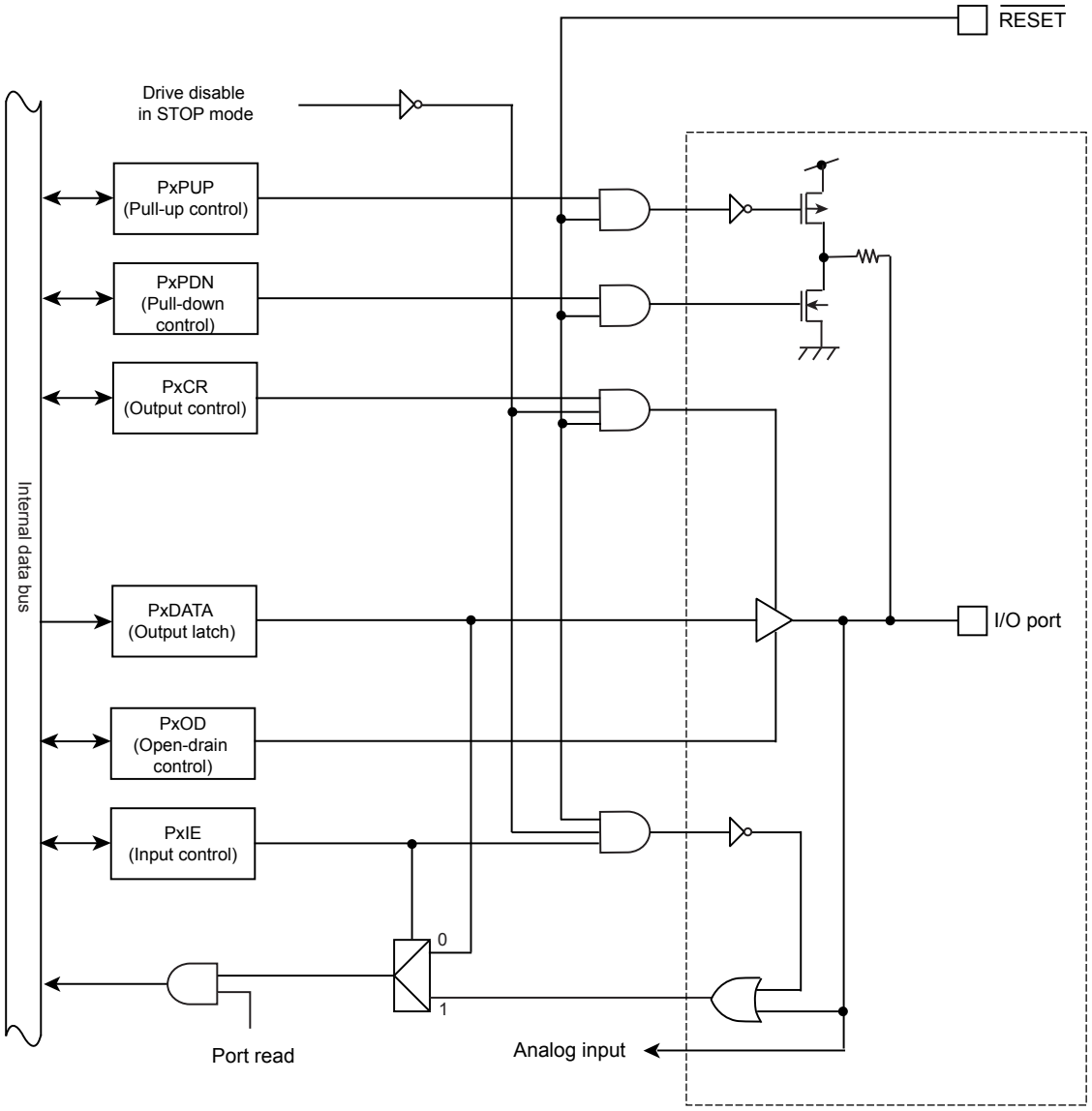


Figure 8-5 Port Type FT5

8.5.7 Type FT6

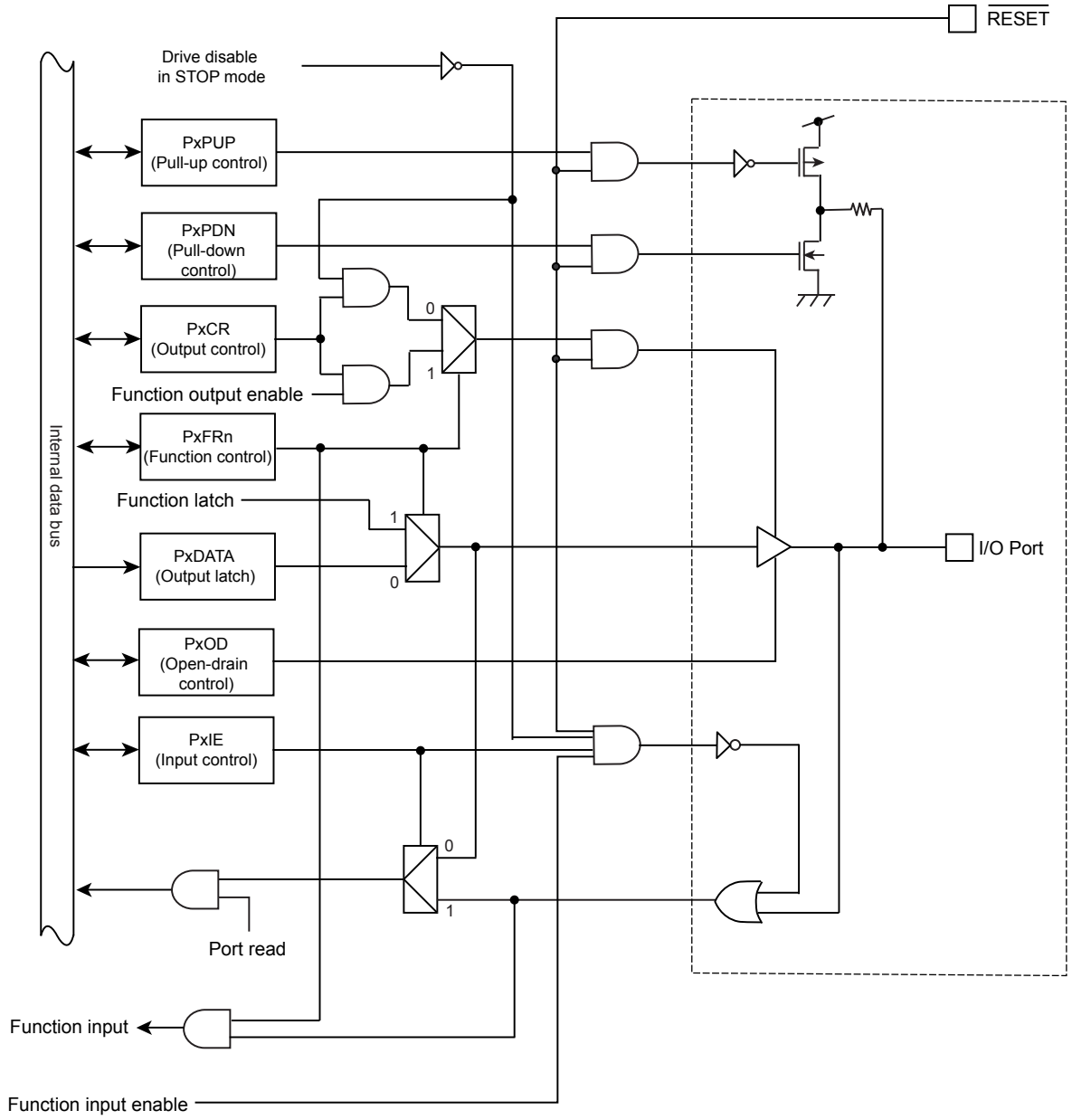


Figure 8-6 Port Type FT6

8.5.8 Type FT7

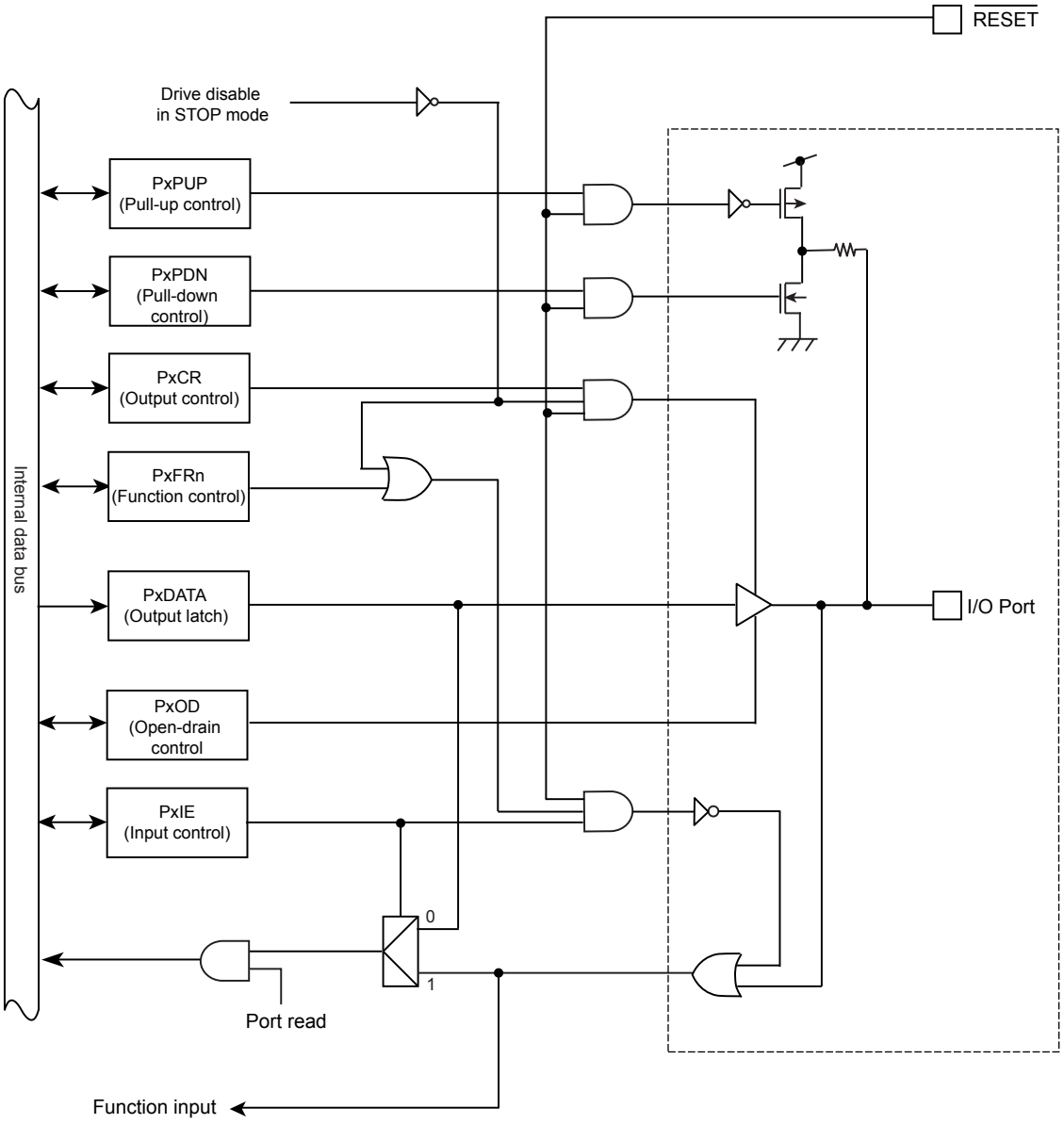


Figure 8-7 Port Type FT7

8.5.9 Type FT8

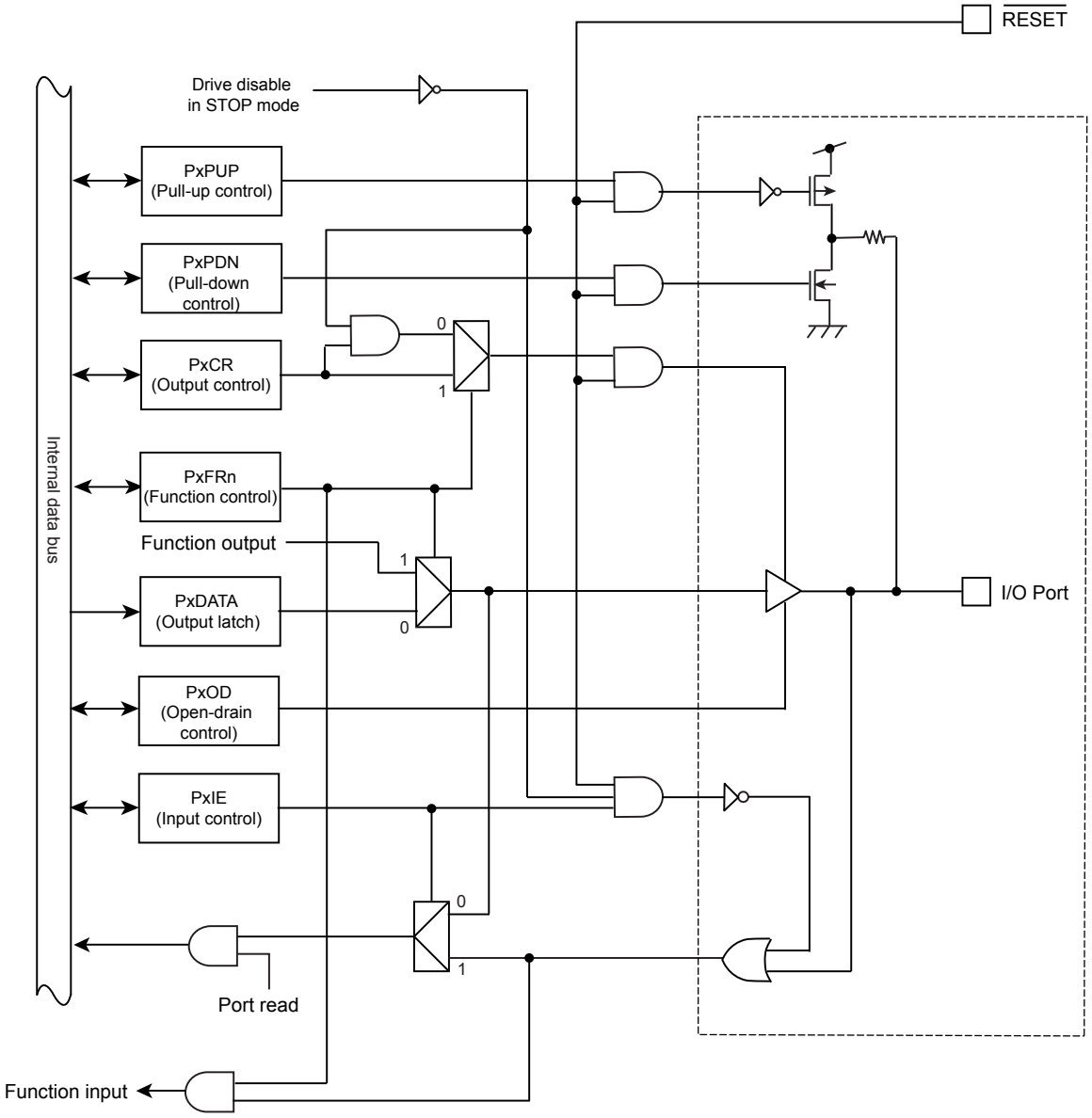


Figure 8-8 Port Type FT8

8.5.10 Type FT9

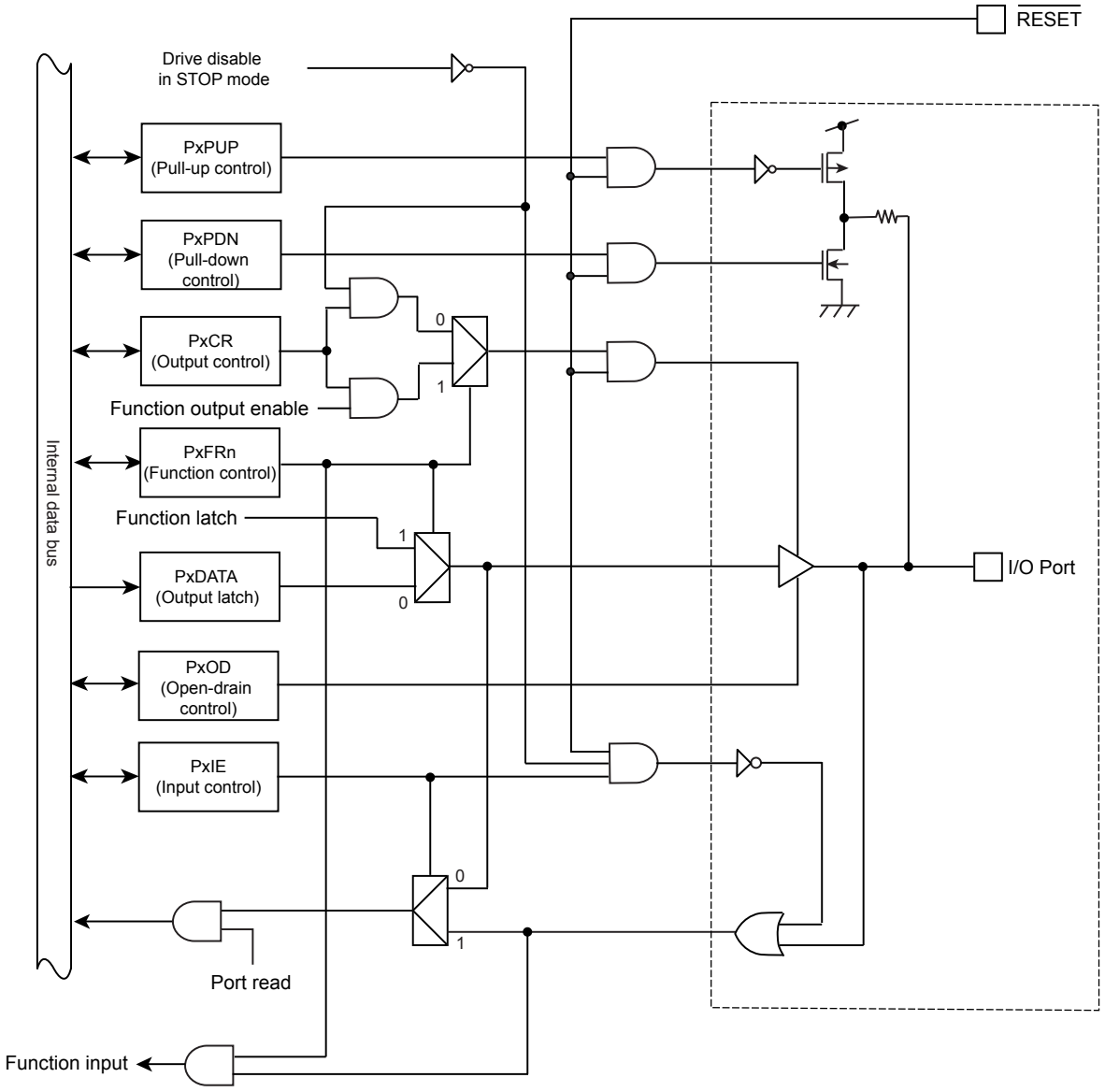


Figure 8-9 Port Type FT9

8.5.11 Type FT10

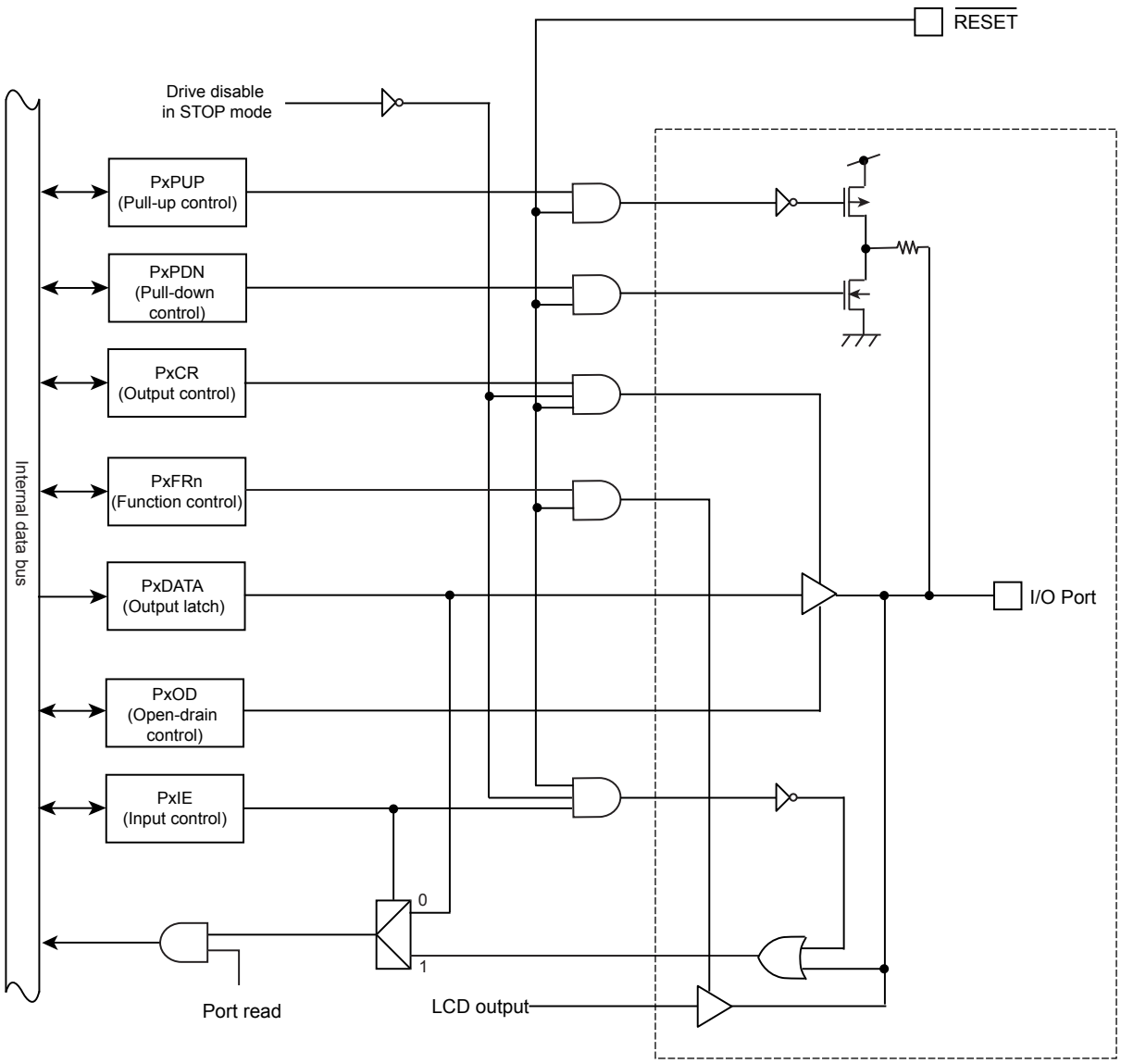


Figure 8-10 Port Type FT10

8.6 Appendix (List of Port Setting)

The following tables show port register settings in each pin.

The numbers "0" and "1" described below register names indicate a setting value and "x" means optional.

8.6.1 The Setting of I/O Port

When I/O port is used as for input port or output port, set its registers as follows:

Pin name	Port type	Function	After reset	Px CR	Px FRn	Px OD	Px PUP	Px PDN	Px IE
Pxn	-	Input port		0	0	x	x	x	1
		Output port		1	0	x	x	x	0

8.6.2 The Setting of Input Dedicated Port

When an input dedicated port is used, set its registers as follows:

Pin name	Port type	Function	After reset	Px CR	Px FRn	Px OD	Px PUP	Px PDN
Pxn	-	Input port		0	x	x	x	1

8.6.3 The Setting of Output Dedicated Port

When an output dedicated port is used, set its registers as follows:

Pin name	Port type	Function	After reset	Px CR	Px FRn	Px OD	Px PUP	Px PDN
Pxn	-	Output port		1	0	x	x	x
		Output port (Hi-Z output)		0	0	x	x	x

8.6.4 The setting of peripheral's I/O port

This section describes the settings in case that ports are used for peripheral functions.

In almost all pins, the initial port state of port registers is all "0" and is inhibited to input/output after reset. Some pins are specified as a certain function after reset. In this case, the symbol "o" is described in the "After reset" Column of the following tables.

The column of PxFRn indicates the function register that is required to set.

In the shaded areas of bits of the following tables indicate that "0" is read and write has no meaning.

8.6.4.1 Port A Setting

Pin name	Port Type	Function	After reset	PA CR	PA FRn	PA OD	PA PUP	PA PDN	PA IE
PA0	FT10	SEG0 (Output)		0	PA0 FR1	0	0	0	0
PA1	FT10	SEG1 (Output)		0	PA1 FR1	0	0	0	0
PA2	FT10	SEG2 (Output)		0	PA2 FR1	0	0	0	0
PA3	FT10	SEG3 (Output)		0	PA3 FR1	0	0	0	0
PA4	FT10	SEG4 (Output)		0	PA4 FR1	0	0	0	0
PA5	FT10	SEG5 (Output)		0	PA5 FR1	0	0	0	0
PA6	FT10	SEG6 (Output)		0	PA6 FR1	0	0	0	0
PA7	FT10	SEG7 (Output)		0	PA7 FR1	0	0	0	0

8.6.4.2 Port B Setting

Pin name	Port Type	Function	After reset	PB CR	PB FRn	PB OD	PB PUP	PB PDN	PB IE
PB0	FT10	SEG8 (Output)		0	PB0 FR1	0	0	0	0
PB1	FT10	SEG9 (Output)		0	PB1 FR1	0	0	0	0
PB2	FT10	SEG10 (Output)		0	PB2 FR1	0	0	0	0
PB3	FT10	SEG11 (Output)		0	PB3 FR1	0	0	0	0
PB4	FT10	SEG12 (Output)		0	PB4 FR1	0	0	0	0
PB5	FT10	SEG13 (Output)		0	PB5 FR1	0	0	0	0
PB6	FT10	SEG14 (Output)		0	PB6 FR1	0	0	0	0
PB7	FT10	SEG15 (Output)		0	PB7 FR1	0	0	0	0

8.6.4.3 Port C Setting

Pin name	Port Type	Function	Pin name	PC CR	PC FRn	PC OD	PC PUP	PC PDN	PC IE
PC0	FT10	SEG16 (Output)		0	PC0 FR1	0	0	0	0
PC1	FT10	SEG17 (Output)		0	PC1 FR1	0	0	0	0
PC2	FT10	SEG18 (Output)		0	PC2 FR1	0	0	0	0
PC3	FT10	SEG19 (Output)		0	PC3 FR1	0	0	0	0
PC4	FT10	SEG20 (Output)		0	PC4 FR1	0	0	0	0
PC5	FT10	SEG21 (Output)		0	PC5 FR1	0	0	0	0
PC6	FT10	SEG22 (Output)		0	PC6 FR1	0	0	0	0
PC7	FT10	SEG23 (Output)		0	PC7 FR1	0	0	0	0

8.6.4.4 Port D Setting

Pin name	Port Type	Function	Pin name	PD DR	PD FRn	PD OD	PD PUP	PD PDN	PD IE
PD0	FT10	SEG24 (Output)		0	PD0 FR1	0	0	0	0
PD1	FT10	SEG25 (Output)		0	PD1 FR1	0	0	0	0
PD2	FT10	SEG26 (Output)		0	PD2 FR1	0	0	0	0
PD3	FT10	SEG27 (Output)		0	PD3 FR1	0	0	0	0
PD4	FT10	SEG28 (Output)		0	PD4 FR1	0	0	0	0
PD5	FT10	SEG29 (Output)		0	PD5 FR1	0	0	0	0
PD6	FT10	SEG30 (Output)		0	PD6 FR1	0	0	0	0
PD7	FT10	SEG31 (Output)		0	PD7 FR1	0	0	0	0

8.6.4.5 Port E Setting

Pin name	Port Type	Function	Pin name	PE CR	PE FRn	PE OD	PE PUP	PE PDN	PE IE
PE0	FT10	SEG32 (Output)		0	PE0 FR1	0	0	0	0
PE1	FT10	SEG33 (Output)		0	PE1 FR1	0	0	0	0
PE2	FT10	SEG34 (Output)		0	PE2 FR1	0	0	0	0
	FT1	T16A6OUT (Output)		1	PE2 FR2	x	x	x	0
PE3	FT10	SEG35 (Output)		0	PE3 FR1	0	0	0	0
	FT1	SCLK31 (Input)		0	PE3 FR2	x	x	x	1
		SCLK31 (Output)		1	PE3 FR2	x	x	x	0
	FT1	$\overline{\text{CTS31}}$ (Output)		1	PE3 FR3	x	x	x	0
PE4	FT10	SEG36 (Output)		0	PE4 FR1	0	0	0	0
	FT1	RXD31 (Input)		0	PE4 FR2	x	x	x	1
PE5	FT10	SEG37 (Output)		0	PE5 FR1	0	0	0	0
	FT1	TXD31 (Input)		1	PE5 FR2	x	x	x	0
PE6	FT10	SEG38 (Output)		0	PE6 FR1	0	0	0	0
	FT2	SWCLK (Input)	o	0	PE6 FR2	0	0	1	1
PE7	FT10	SEG39 (Output)		0	PE7 FR1	0	0	0	0
	FT2	SWDIO (I/O)	o	1	PE7 FR2	0	1	0	1

8.6.4.6 Port F Setting

Pin name	Port Type	Function	Pin name	PF CR	PF FRn	PF OD	PF PUP	PF PDN	PF IE
PF0	FT5	AIN0	o	0	0	0	0	0	0
PF1	FT5	AIN1	o	0	0	0	0	0	0
	FT4	INT0 (Input)		0	PF1 FR1	x	x	x	1

8.6.4.7 Port G Setting

Pin name	Port Type	Function	Pin name	PG CR	PG GRn	PG OD	PG PUP	PG PDN	PG IE
PG0	FT1	TB0OUT(Output)		1	PG0 GR0	x	x	x	0
	FT4	INT2 (Input)		0	PG0 GR1	x	x	x	1

8.6.4.8 Port H Setting

Pin name	Port Type	Function	Pin name	PH CR	PH FRn	PH OD	PH PUP	PH PDN	PH IE
PH0	FT1	TXD0 (Output)		1	PH0 FR1	x	x	x	0
	FT1	IROUT0 (Output)		1	PH0 FR2	x	x	x	0
PH1	FT1	RXD0 (Input)		0	PH1 FR1	x	x	x	1
PH2	FT1	SCLK0 (Input)		0	PH2 FR1	x	x	x	1
	FT1	SCLK0 (Output)		1	PH2 FR1	x	x	x	0
	FT1	$\overline{\text{CTS0}}$ (Output)		1	PH2 FR2	x	x	x	0
	FT1	T16A0OUT (Output)		1	PH2 FR3	x	x	x	0
PH3	FT1	TXD1 (Output)		1	PH3 FR1	x	x	x	0
	FT1	IROUT1 (Output)		1	PH3 FR2	x	x	x	0
PH4	FT1	RXD1 (Input)		0	PH4 FR1	x	x	x	1
PH5	FT1	SCLK1 (Input)		0	PH5 FR1	x	x	x	1
	FT1	SCLK1 (Output)		1	PH5 FR1	x	x	x	0
	FT1	$\overline{\text{CTS1}}$ (Output)		1	PH5 FR2	x	x	x	0
	FT1	T16A1OUT (Output)		1	PH5 FR3	x	x	x	0

8.6.4.9 Port I Setting

Pin name	Port Type	Function	Pin name	PI CR	PI FRn	PI OD	PI PUP	PI PDN	PI IE
PI0	FT1	TXD2 (Output)		1	PI0 FR1	x	x	x	0
	FT1	IROUT2 (Output)		1	PI0 FR2	x	x	x	0
PI1	FT1	RXD2 (Input)		0	PI1 FR1	x	x	x	1
PI2	FT1	SCLK2 (Input)		0	PI2 FR1	x	x	x	1
	FT1	SCLK2 (Output)		1	PI2 FR1	x	x	x	0
	FT1	$\overline{\text{CTS2}}$ (Output)		1	PI2 FR2	x	x	x	0
	FT1	T16A2OUT (Output)		1	PI2 FR3	x	x	x	0
PI3	FT1	TB0IN (Input)		0	PI3 FR1	x	x	x	1
PI4	FT1	TXD30 (Output)		1	PI4 FR1	x	x	x	0
PI5	FT1	RXD30 (Input)		0	PI5 FR1	x	x	x	1
PI6	FT1	SCLK30 (Input)		0	PI6 FR1	x	x	x	1
	FT1	SCLK30 (Output)		1	PI6 FR1	x	x	x	0
	FT1	$\overline{\text{CTS30}}$ (Output)		1	PI6 FR2	x	x	x	0
	FT1	T16A5OUT (Output)		1	PI6 FR3	x	x	x	0

8.6.4.10 Port J Setting

Pin name	Port Type	Function	Pin name	PJ CR	PJ FRn	PJ OD	PJ PUP	PJ PDN	PJ IE
PJ0	FT1	SDA0 (I/O)		1	PJ0 FR1	1	x	x	1
		SO0 (Output)		1	PJ0 FR1	x	x	x	0
PJ1	FT1	SCL0 (I/O)		1	PJ0 FR1	1	x	x	1
		SI0 (Input)		0	PJ0 FR1	x	x	x	1
PJ2	FT1	SCK0 (Input)		0	PJ2 FR1	x	x	x	1
		SCK0 (Output)		1	PJ2 FR1	x	x	x	0
	FT4	INT1 (Input)		0	PJ2 FR2	x	x	x	1
PJ3	FT1	RTCOUT (Output)		1	PJ3 FR1	x	x	x	0
PJ4	FT1	T16A3OUT (Output)		1	PJ4 FR1	x	x	x	0
	FT1	SCOUT (Output)		1	PJ4 FR2	x	x	x	0
PJ5	FT1	TB1IN (Input)			PJ5 FR1				1
	FT1	XTCLKIN (Input)			PJ5 FR2				1

8.6.4.11 Port K Setting

Pin name	Port Type	Function	Pin name	PK CR	PK FRn	PK OD	PK PUP	PK PDN	PK IE
PK0	FT5	LV1 (Input)	o	0	0	0	0	0	0
	FT4	INT3 (Input)		0	PK0 FR1	x	x	x	1
PK1	FT5	LV2 (Input)	o	0	0	0	0	0	0
	FT1	TB1OUT(Output)		1	PK1 FR1	x	x	x	0

9. 16-bit Timer / Event Counters (TMRB)

9.1 Outline

TMRB has the operation modes shown as below.

- Interval timer mode
- Event counter mode
- Programmable pulse generation (PPG) mode
- Programmable pulse generation (PPG) external trigger mode

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

- Frequency measurement
- Pulse width measurement

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

9.2 Block Diagaram

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

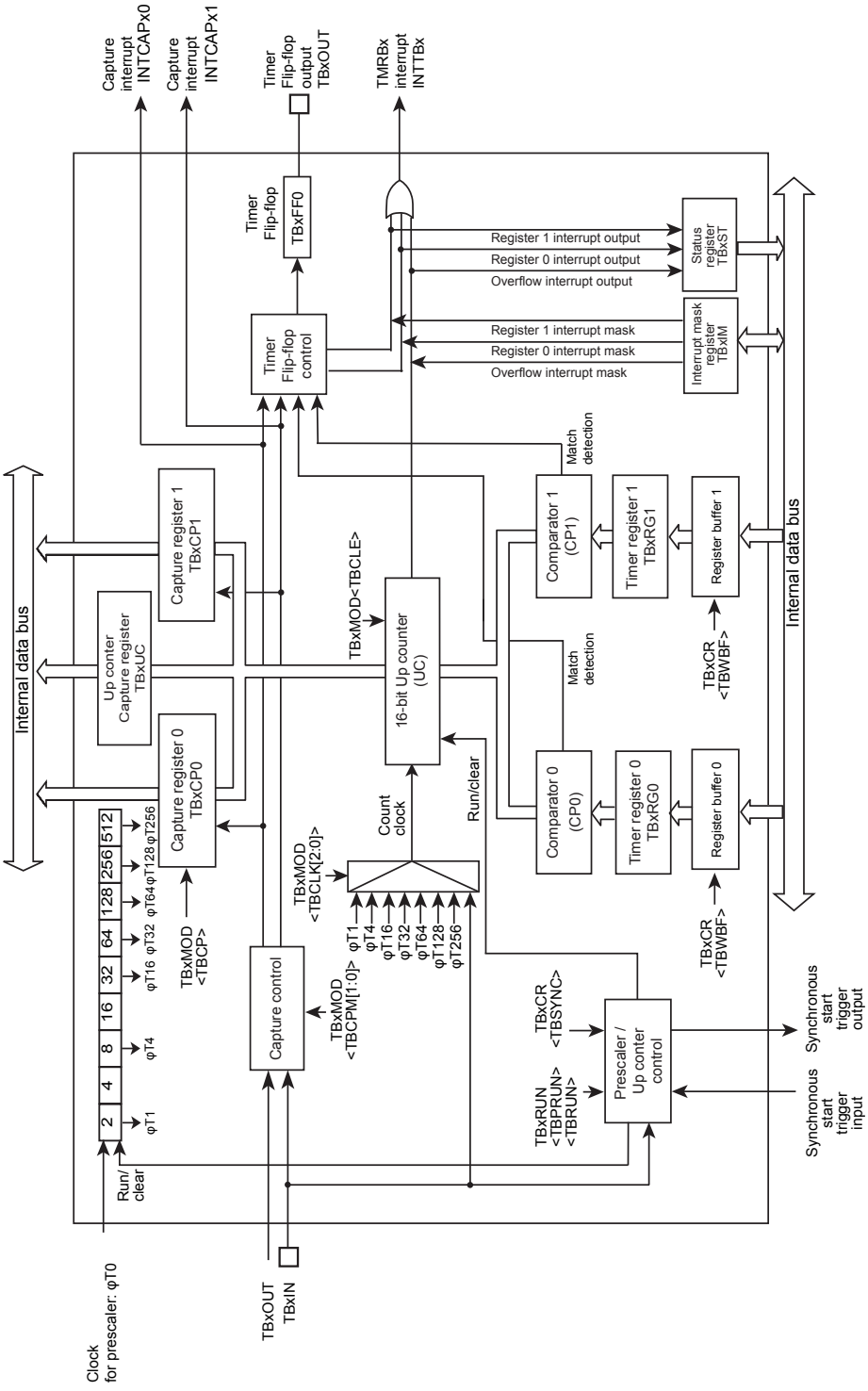


Figure 9-1 TMRBx Block Diagram

9.3 Registers

9.3.1 Register list

The table below shows control registers and their addresses.

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

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

Note: During timer operation, timer control register, timer mode register and timer flip-flop control register should not be modified. After stopping timer operation, they should be modified.

9.3.2 TBxEN (Enable register)

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

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

9.3.3 TBxRUN (RUN register)

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

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

9.3.4 TBxCR (Control register)

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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	TBWBF	R/W	Double Buffer 0: Disabled 1: Enabled
6	−	R/W	Write "0".
5	TBSYNC	R/W	Synchronous mode switching 0: individual (Each channel) 1: synchronous
4	−	R	Read as "0".
3	I2TB	R/W	Operation at IDLE mode 0: Stop 1:Operation
2	−	R/W	Write "0".
1	TRGSEL	R/W	Selects the external triggers. 0: rising 1: falling Controls the edge selection (of signal to TBxIN pin) when the external triggers is selected.
0	CSSEL	R/W	Selects the count start 0: starts by software 1: starts by external trigger

9.3.5 TBxMOD (Mode register)

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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	−	R/W	Write "0".
6	TBCP	W	Capture control by software 0: Capture by software 1: Don't care When "0" is written, the capture register 0 (TBxCP0) takes count value. Read as "1".
5-4	TBCPM[1:0]	R/W	Capture timing 00: Disable Capture 01: Reserved 10: TBxIN↑, TBxIN↓ Takes count values into capture register 0 (TBxCP0) upon rising of TBxIN pin input. Takes count values into capture register 1 (TBxCP1) upon falling of TBxIN pin input. 11: TBxFF0↑, TBxFF0↓ Takes count values into capture register 0 (TBxCP0) upon rising of TBxFF0 and into capture register 1 (TBxCP1) upon falling of TBxFF0.
3	TBCLE	R/W	Up-counter control 0: Disables clearing of the up-counter. 1: Enables clearing of the up-counter. Clears and controls the up-counter. When "0" is written, it disables clearing of the up-counter. When "1" is written, it clears up counter when up-counter matches with timer regsiter1 (TBxRG1).
2-0	TBCLK[2:0]	R/W	Selects the TMRBx source clock. 000: TBxIN pin input 001: φT1 010: φT4 011: φT16 100: φT32 101: φT64 110: φT128 111: φT256

Note: Do not make any changes of TBxMOD register while the TMRBx is running.

9.3.6 TBxFFCR (Flip-flop control register)

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

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

9.3.7 TBxST (Status register)

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

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

Note 1: Even if mask configuration by TBxIM register is valid, the status is set to TBxST register.

Note 2: When the interrupt mask configuration is disabled by the corresponding bit of TBxIM register, the interrupt is issued to the CPU.

Note 3: To clear the flag, TBxST register should be read.

9.3.8 TBxIM (Interrupt mask register)

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

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

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

9.3.9 TBxUC (Up counter capture register)

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

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

9.3.10 TBxRG0 (Timer register 0)

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

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

9.3.11 TBxRG1 (Timer register 1)

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

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

9.3.12 TBxCP0 (Capture register 0)

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

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

9.3.13 TBxCP1 (Capture register 1)

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

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

9.4 Description of Operation

9.4.1 Prescaler

There is prescaler to generate the source clock for up-counter.

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

The operation or the stoppage of a prescaler is set with TBxRUN<TBPRUN> where writing "1" status counting and writing "0" clears and stops counting.

9.4.2 Up-counter (UC)

UC is a 16-bit binary counter.

9.4.2.1 Source clock

UC's source clock is specified by TBxMOD<TBCLK[2:0]>.

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

9.4.2.2 Counter start / stop

To start the counter, there are a software start, external trigger start and synchronous start.

1. Software start

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

2. External trigger start

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

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

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

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

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

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

3. Synchronous start

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

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

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

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

9.4.2.3 Timing to clear UC

1. When a match with TBxRG1 is detected

   By setting TBxMOD<TBCLE> = "1", UC is cleared if when the comparator detects a match between UC and TBxRG1.

2. When UC stops

   UC stops and is cleared if TBxRUN<TBRUN> = "0".

9.4.2.4 UC overflow

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

9.4.3 Timer registers (TBxRG0, TBxRG1)

TBxRG0 and TBxRG1 are registers for setting value 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 timer register and in an up-counter, comparator outputs the match detection signal.

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

Controlling double buffering disable or enable is specified by TBxCR<TBWBF>. if <TBWBF> = 0, the double buffering becomes disable, If <TBWBF> ="1", it becomes enable.

When the double buffering is enabled, data transfers from the register buffer to the timer register (TBxRG0/1) in the case that UC is matched with TBxRG1.

When UC is stopped even if double buffering is enabled, the double buffering operates as a single buffer, and data can be written to the TBxRG0 and TBxRG1 directly.

9.4.4 Capture control

This is a circuit that controls the timing of latching values from UC into the TBxCP0 and TBxCP1. The capture timing of UC is specified by TBxMOD<TBCPM[1:0]>.

Software can also capture the value of UC to capture registers. The value of UC are taken into the TBxCP0 each time "0" is written to TBxMOD<TBCP>.

9.4.5 Capture registers (TBxCP0, TBxCP1)

This register captures the value of UC.

9.4.6 Up-counter capture register (TBxUC)

If TBxUC register is read during the counter operation, the current value of up-counter will be captured and the value will be read. The value captured at the end is held while the counter is stopping.

9.4.7 Comparators (CP0, CP1)

This circuit compares with UC and the value set to TBxRG0/1 and detects match. If a match is detected, INTTBx is occurred.

9.4.8 Timer flip-flop (TBxFF0)

The timer flip-flop (TBxFF0) is reversed by a match signal from the comparator and a latch signal to the capture registers. It can be enabled or disabled to reverse by setting the TBxFFCR<TBC1T1, TBC0T1, TBC1T1, TBC1T0>.

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

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

9.4.9 Capture interrupt (INTCAPx0, INTCAPx1)

INTCAPx0 and INTCAPx1 can be generated at the timing of latching value from UC into the TBxCP0 and TBxCP1.

9.5 Description of Operation for each mode

9.5.1 Interval timer mode

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

		7	6	5	4	3	2	1	0	
TBxEN	←	1	X	X	X	X	X	X	X	Enable TMRBx operation.
TBxRUN	←	X	X	X	X	X	0	X	0	Stops prescaler and counter.
Interrupt set-enable register	←	*	*	*	*	*	*	*	*	Permits INTTBx interrupt by setting corresponding bit to "1".
TBxFFCR	←	X	X	0	0	0	0	1	1	disable to TBÇòFF0 reverse trigger
TBxMOD	←	X	1	0	0	0	*	*	*	Changes to prescaler output clock as input clock. Specifies capture function to disable.
						(*** = 001 to 111)				
TBxRG1	←	*	*	*	*	*	*	*	*	Specifies a time interval. (16 bits)
	←	*	*	*	*	*	*	*	*	
TBxRUN	←	X	X	X	X	X	1	X	1	Starts prescaler and counter.

Note: X; Don't care, *; optional value, -; Don't change

9.5.2 Event counter mode

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

The UC counts up on the rising edge of TBxIN pin input. The value of UC can be captured by soft capture. It is possible to read the count value by reading it.

		7	6	5	4	3	2	1	0	
TBxEN	←	1	X	X	X	X	X	X	X	Enable TMRBx operation.
TBxRUN	←	X	X	X	X	X	0	X	0	Stops prescaler and counter.
Assigns a corresponding port to TBxIN.										
TBxFFCR	←	X	X	0	0	0	0	1	1	Disable to TBxFF0 reverse trigger.
TBxMOD	←	X	1	0	0	0	0	0	0	Set to a source clock as TBxIN pin input.
TBxRUN	←	X	X	X	X	X	1	X	1	Starts prescaler and counter.
TBxMOD	←	X	0	-	-	-	-	-	-	Software capture is done.

Note: X; Don't care, *; optional value, -; Don't change

9.5.3 Programmable pulse generation (PPG) output mode

Square wave with any frequency and any duty can be output. The output pulse can be either low-active or high-active.

TBxFF0 is reversed when UC matches the set value of TBxRG0 and TBxRG1. TBxFF0 can be output from TBxOUT pin.

Note that the set value of TBxRG0 and TBxRG1 must satisfy the following requirement.

Set value of TBxRG0 < Set value of TBxRG1

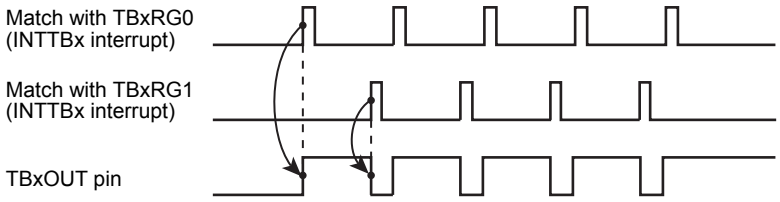


Figure 9-2 Example of Programmable pulse generation output

In this mode, by enabling the double buffering, The value of register buffer 0 and 1 are shifted into TBxRG0 and 1 when UC matches the value of TBxRG1.

It is possible to modify frequency and duty without timing of modifying TBxRG0 and TBxRG1.

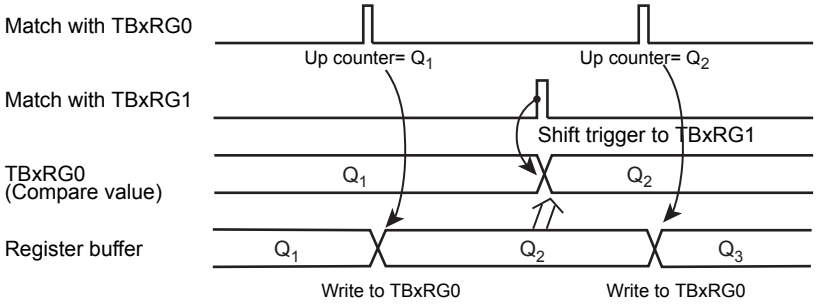


Figure 9-3 Register Buffer Operation

The block diagram of this mode is shown below.

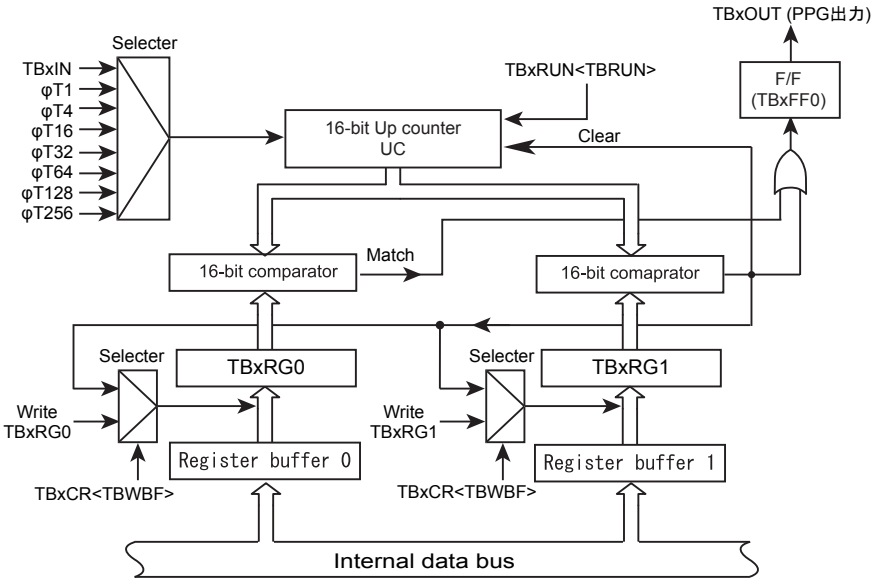


Figure 9-4 Block diagram of 16-bit PPG mode

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

		7	6	5	4	3	2	1	0	
TBxEN	←	1	X	X	X	X	X	X	X	Enables TMRBx operation.
TBxRUN	←	X	X	X	X	X	0	X	0	Stops prescaler and counter.
TBxCR	←	1	0	X	X	X	0	X	X	Enables double-buffering.
TBxRG0	←	*	*	*	*	*	*	*	*	set a duty.
	←	*	*	*	*	*	*	*	*	
TBxRG1	←	*	*	*	*	*	*	*	*	Set a cycle.
	←	*	*	*	*	*	*	*	*	
TBxFFCR	←	X	X	0	0	1	1	1	0	Specifies to trigger TBxFF0 to reverse when a match with TBxRG0 or TBxRG1 is detected. And sets the initial value of TBxFF0 to "0".
TBxMOD	←	X	1	0	0	0	*	*	*	Designates the prescaler output clock as the input clock, and disable the capture function.
						(*** = 001 to 111)				
Assigns a corresponding port to TBxOUT.										
TBxRUN	←	X	X	X	X	X	1	X	1	Starts prescaler and counter.

Note: X; Don't care, *; optional value, -; Don't change

9.5.4 Programmable pulse generation (PPG) external trigger output mode

A PPG wave with a short delay time can be output by using external trigger count start mode.

The example of an one-shot pulse output by external trigger count start mode is shown below.

To start count up by the rising edge of TBxIN, set TBxCR<CSSEL> to "1" and clear

TBxCR<TRGSEL> to "0" in stopping 16-bit up counter.

TBxRG0 is set the delay time (d) from an external trigger signal. TBxRG1 is set the value (d)+(p) which is added the delay time (d) and the width (p) of one-shot pulse.

To reverse TBxFF0 when UC matches TBxRG0 and TBxRG1, TBxFFCR<TBE1T1> and TBxFFCR<TBE1T1> are set to "1".

UC is readied to start UC by setting TBxRUN<TBPRUN> and TBxRUN<TBRUN> to "1".

UC starts by the rising edge of external trigger.

TBxFF0 is reversed when UC counts up to (d) and UC matches TBxRG0. TBxFF0 is "High" level.

TBxFF0 is reversed when UC counts up to (d)+(p) and UC matches TBxRG1. TBxFF0 is "Low" level.

To fix the level of TBxFF0, clear TBxFFCR<TBE1T1> and TBxFFCR<TBE0T1> to "0" or stops UC by TBxRUN<TBPRUN><TBRUN> in INTTBx which is generated when UC matches TBxRG1.

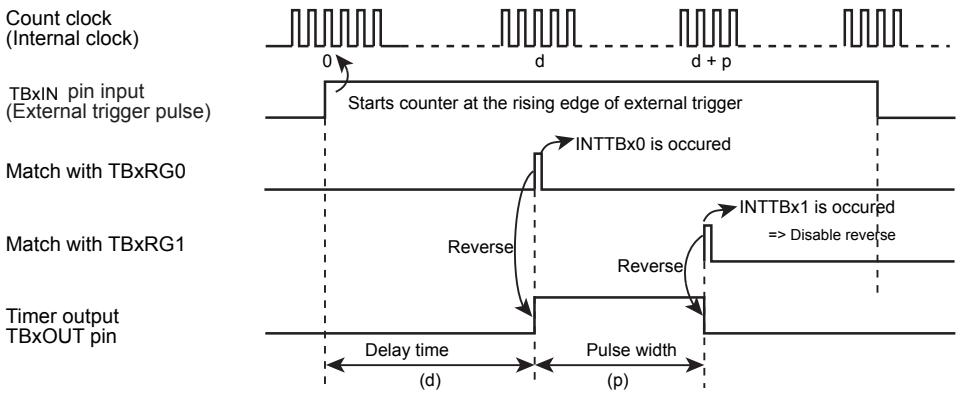


Figure 9-5 One-shot pulse output with delay by external trigger start

The followings shows the setting in the case that 2 ms width one-shot pulse is output after 3 ms by triggering TBxIN input at the rising edge. In this example, the source clock is φT1.

		7	6	5	4	3	2	1	0	
[Main processing]										
Assigns a corresponding port to TBxIN.										
TBxEN	←	1	X	X	X	X	X	X	X	Enables TMRBx operation.
TBxRUN	←	X	X	X	X	X	0	X	0	Stops prescaler and counter.
TBxRG0	←	*	*	*	*	*	*	*	*	Set count value. (3ms/φT1)
TBxRG0	←	*	*	*	*	*	*	*	*	
TBxRG1	←	*	*	*	*	*	*	*	*	Set count value. (3+2)ms/φT1)
TBxRG1	←	*	*	*	*	*	*	*	*	
TBxFFCR	←	X	X	0	0	1	1	1	0	Reverses TBxFF0 if UC matches TBxRG0 and TBxRG1. Clear TBxFF0 to "0".
TBxMOD	←	X	1	0	0	0	0	0	1	Starts UC as free-running. Selects φT1 for the source clock. Disable capture UC.
Assigns a corresponding port to TBxOUT.										
TBxIM	←	X	X	X	X	X	1	0	1	Masks except TBxRG1 interrupt.
Interrupt set-enable register	←	*	*	*	*	*	*	*	*	Permits to generate interrupt specified by INTTBx interrupt corresponding bit setting to "1".
TBxRUN	←	X	X	X	X	X	1	X	1	Starts prescaler and counter.
[Processing of INTTBx interrupt service routine] Output disable										
TBxFFCR	←	X	X	−	−	0	0	−	−	Clears TBxFF0 reverse trigger setting
TBxRUN	←	X	X	X	X	X	0	X	0	Stops prescaler and counter.

Note: X; Don't care, *; optional value, -; Don't change

9.6 Applications using the capture function

The capture function can be used many applications.

The applications are shown below.

1. Frequency measurement
2. Pulse width measurement

9.6.1 Frequency measurement

The frequency of an external clock can be measured.

To measure frequency, TMRBm is used as 16-bit interval timer mode and TMRBn is used as 16-bit event counter mode.

To count UC of TMRBn freely by an external clock, set TMnMOD<TBCLK> to "000" and set TBnRUN<TBE1T1><TBE0T1> to "11".

To reverse TBmFF0 when UC of TMRBm matches TBmRG0 and TBmRG1, set TBmFFCR<TBE1T1><TBE0T1> to "11".

To capture UC to TBnCP0 at rising edge of TBmFF0 and UC to TBmCP1 at falling edge of TBmFF0, set TBxMOD<TBCPM> to "11".

Set TBmRG0 and TBmRG1 to time when UC counts an external clock and start TMRBm.

Rises-up TBmFF0 when UC of TMRBm matches TBmRG0 and captures the value of TMRBn's UC to TBnCP0. Falls-down TBmFF0 when UC of TMRBm matches TBmRG1 and captures the value of TMRBn's UC to TBnCP1.

A frequency is measured from (TBnCP1 - TBnCP0) ÷ (TBmRG1 - TBmRG0) in INTTBm.

For example, the difference between TBmRG1 and TBmRG0 is 0.5 s and the difference between TBnCP1 and TBnCP0 is 100, the frequency is 200 Hz (100 ÷ 0.5 s = 200Hz)

TBnCP1 - TBnCP0 may be less than zero depend on the changing timing of TBmFF0. Please correct the value if TBnCP1 - TBnCP0 is less than zero.

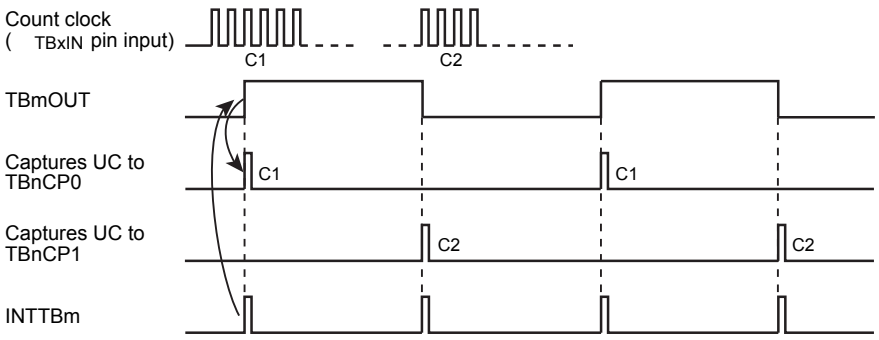


Figure 9-6 Frequency measurement

The following shows in the case that the measured pulse is input to TBxIN. In this example, the source clock is φT1.

		7	6	5	4	3	2	1	0	
[Main processing] Capture setting by TBmFF0										
Assigns a corresponding port to TBxIN.										
TBmEN	←	1	X	X	X	X	X	X	X	Enables TMRBm operation.
TBmRUN	←	X	X	X	X	X	0	X	0	Stops prescaler and counter.
TBnEN	←	1	X	X	X	X	X	X	X	Enables TMRBn operation.
TBnRUN	←	X	X	X	X	X	0	X	0	Stops prescaler and counter.
TBmCR	←	1	0	X	X	X	0	X	X	Enables double-buffering.
TBmRG0	←	*	*	*	*	*	*	*	*	Set the external clock measured time 1.
	←	*	*	*	*	*	*	*	*	
TBmRG1	←	*	*	*	*	*	*	*	*	Set the external clock measured time 2.
	←	*	*	*	*	*	*	*	*	
TBmFFCR	←	X	X	0	0	1	1	1	0	Reverses TBxFF0 if UC matches TBxRG0 and TBxRG1. Clear TBxFF0 to "0".
TBnMOD	←	0	1	1	1	0	0	0	0	Captures at the rising / falling edge. Clears and disables UC. Input clock is TBxIN.
TBmIM	←	X	X	X	X	X	1	0	1	Masks except TBxRG1 interrupt.
Interrupt set-enable register	←	*	*	*	*	*	*	*	*	Permits to generate interrupt specified by INTTBm interrupt corresponding bit setting to "1".
TBnRUN	←	X	X	X	X	X	1	X	1	Starts prescaler and counter.
TBmRUN	←	X	X	X	X	X	1	X	1	Starts prescaler and counter.
[Processing of INTTBm interrupt service routine]										
TBmFFCR	←	X	X	-	-	0	0	-	-	Clears TBxFF0 reverse trigger setting
Interrupt enable clear register	←	*	*	*	*	*	*	*	*	Prohibits interrupt specified by INTTBm corresponding bit by setting to "1".
TBnCP0 and TBnCP1 are read out and the frequency is calculated.										

Note: X; Don't care, *; optional value, -; Don't change

9.6.2 Pulse width measurement

"High" level width of the external pulse can be measured.

To capture UC to TBxCP0 at rising edge of TBxIN and UC to TBxCP1 at falling edge of TBxIN, set TBxMOD<TBCPM> to "10".

Enables INTCAPx1 interrupt.

Enables TMRBx operation.

Captures the vale of UC to TBxCP0 when the rising edge of the external pulse into TBxIN. Captures the value of UC to TBxCP1 when the falling edge of the external pulse into TBxIN and INTCAPx1 interrupt is occurred.

The "High" level width of the external pulse can be calculated by multiplying the difference between TBxCP0 and TBxCP1 by the clock cycle of a prescaler output clock.

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

When the pulse width which is more than maximum count time of UC is measured, please correct the measured value.

The "Low" level width of an external pulse can also be measured.

In this case, enables INTCAPx0 interrupt. In twice process of INTCAPx0 interrupt, the difference between C2 generated the first time and C1 generated the second time in "Figure 9-7 Pulse width measurement" is multiplied by the cycle of the prescaler output clock.

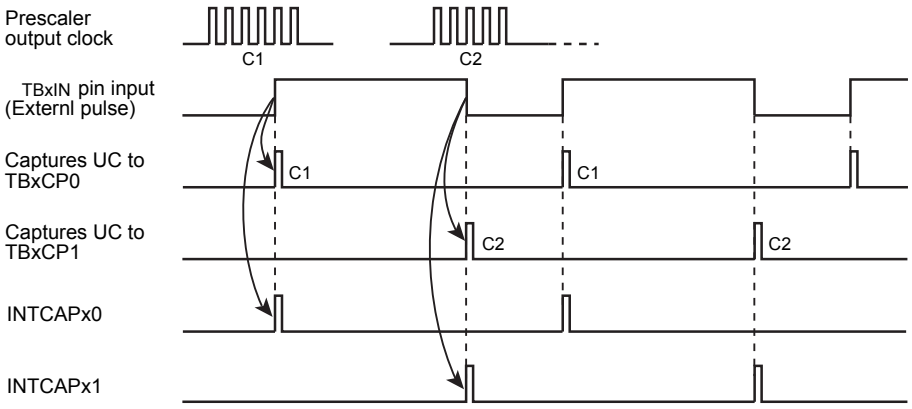


Figure 9-7 Pulse width measurement

The following is shown that the "High" level width of the external pulse into TBxIN is measured. In this example, the source clock is φT1.

		7	6	5	4	3	2	1	0	
[Main processing] Capture setting TBxIN.										
Assigns a corresponding port to TBxIN.										
TBxEN	←	1	X	X	X	X	X	X	X	Enables TMRBx operation
TBxRUN	←	X	X	X	X	X	0	X	0	Stops prescaler and counter.
TBxFFCR	←	X	X	0	0	0	0	1	0	Clears TBxFF0 reverse trigger and TBxFF0.
TBxMOD	←	X	1	1	0	0	0	0	1	Starts UC as free-running. Selects φT1 for the source clock. UC is captured to TBxCP0 at the rising edge of TBxIN. UC is captured to TBxCP1 at the falling edge of TBxIN.
Interrupt set-enable register	←	*	*	*	*	*	*	*	*	Permits to generate interrupt specified by INTCAPx1 interrupt corresponding bit setting to "1".
TBxRUN	←	X	X	X	X	X	1	X	1	Starts prescaler and counter.
[Processing INTCAPx1 interrupt service routine] Calculate the width of "High" level.										
Interrupt enable clear register	←	*	*	*	*	*	*	*	*	Prohibits interrupt specified by INTCAPx1 interrupt corresponding bit setting to "1".
Calculated the width of "High" level by reading TBxRG0 and TBxRG1.										

Note: X; Don't care, *; optional value, -; Don't change

10. 16-Bit Timer A (TMR16A)

10.1 Outline

TMR16A contains the following functions:

- Match interrupt
- Square waveform output
- Read capture

In this chapter, "x" indicates a channel number.

10.2 Block Diagram

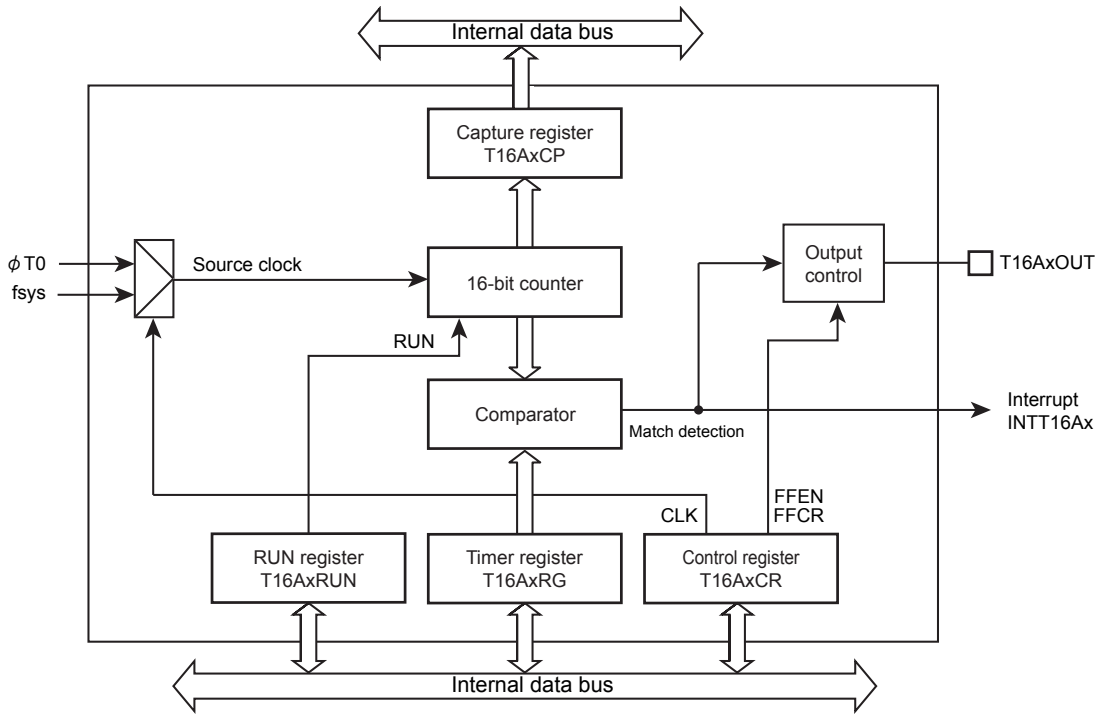


Figure 10-1 Block diagram of TMR16A

10.3 Registers

10.3.1 Register List

The table below shows control registers and their addresses.

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

Register name		Address(Base+)
Enable register	T16AxEN	0x0000
RUN register	T16AxRUN	0x0004
Control register	T16AxCR	0x0008
Timer register	T16AxRG	0x000C
Capture register	T16AxCP	0x0010

Note: When T16ARUN<RUN> is set to "1", do not modify T16AxEN, T16AxCR, T16AxRG and T16AxCP.

10.3.2 Details of Registers

10.3.2.1 T16AxEN (Enable Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	HALT	I2T16A
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	-	R	Read as "0".
1	HALT	R/W	Operation during halt mode debug 0: Operating 1: Stop Specifies the operation during halt mode debug. Write "1" to the bit to stop the operation.
0	I2T16A	R/W	Operation during the IDLE mode 0: Stop 1: Operating Specifies the operation during the IDLE mode. Write "1" to the bit to continue the operation.

10.3.2.2 T16AxRUN (RUN Register)

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

Bit	Bit Symbol	Type	Function
31-1	−	R	Read as "0".
0	RUN	R/W	Counter operation 0: Stop 1: Operating

10.3.2.3 T16AxCR (Control Register)

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

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as "0".
15	UCCR	R/W	Write to "1"
14-8	−	R	Read as "0".
7	FFEN	R/W	Inverse of T16AxOUT 0: Disabled 1: Enabled Write "1" to the bit to invert T16AxOUT when the counter matches with T16ARG.
6	−	R	Read as "0".
5-4	FFCR[1:0]	W	T16AxOUT control 00: Invert 01: Set 10: Clear 11: No operation Write a value to the bit to control T16AxOUT by software. Read as "11".
3-1	−	R	Read as "0".
0	CLK	R/W	Source clock 0: fsys 1: ΦT0 Specifies a source clock.

10.3.2.4 T16AxRG (Timer Register)

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

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as "0".
15-0	RG[15:0]	R/W	Set a value to compare with a counter

Note: Do not set "0x0000".

10.3.2.5 T16AxCP (Capture Register)

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

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as "0".
15-0	CP[15:0]	R/W	Counter value [Read] Reads a current counter value. [Write] Sets a counter value. Since the counter is cleared only when the counter matches with T16AxRG<RG>, write "0x0000" to clear the register before starting the operation.

10.4 Operation Description

10.4.1 Timer Operation

1. Preparation

   Set "1" to T16AxCR<UCCR>.

   Select a source clock with T16AxCR<CLK>. Write "0" to set fsys or write "1" to set ΦT0. Set a counter value to T16AxRG<RG>.

   Modify T16AxCR and T16AxRG while the counter stops (T16AxRUN<RUN> is "0").

2. Counter operation

   Before starting counter operation, set "0x0000" to T16AxCP<CP> to clear the counter.

   To start count-up, set "1" to T16AxRUN<RUN>. If the counter value matches with a value of T16AxRG<RG>, it will be cleared to "0x0000" and continued to count-up.

3. Match detection interrupt generation

   If a counter value matches with a value of T16AxRG, a match detection interrupt INTT16Ax will be output.

4. Stop

   To stop counts, set "0" to T16AxRUN<RUN>. The counter value is held. Then clear the counter before counting is started by setting "1" to <RUN>.

Note: Modification of T16AxCR, T16AxRG and T16AxCP must be performed while the counter is stopping (T16AxRUN<RUN> is set to "0").

10.4.2 T16AxOUT Control

T16AxOUT is modified by register setting or by matching the counter with T16AxRG.

An initial state of T16AxOUT is "0".

1. Control by software

   With T16AxCR<T16AFFC> setting, T16AxOUT can be specified; "1" is to set, "0" is to clear, and also the inverted setting is possible.

   Modify T16AxCR while the counter stops (T16AxRUN<RUN> is "0").

2. Inverse due to matching counter

   Write "1" to T16ACR<FFEN> to invert T16AxOUT. When T16AxRG matches with a counter value, T16AxOUT will invert. When the counter stops, a state of T16AxOUT will remain.

10.4.3 Read Capture

A current value of the counter can be captured by reading T16AxCP.

10.4.4 Automatic Stop

With the setting of T16AxEN, TMR16A automatically stops in the following conditions:

1. Transition to/from IDLE mode

   With T16AEN<I2T16A> setting, TMR16A operation during IDLE mode can be specified. If "1" is set, TMR16A automatically stops count-up when the transition to the IDLE mode occurs. If TMR16A returns from IDLE mode, it will restart counting-up operation.

2. Debug halt

   With T16AEN<HALT> setting, TMR16A operation during debug halt can be specified. If "0" is set, TMR16A automatically stops count-up when the transition to the debug halt mode occurs. If debug halt mode of the core is canceled, count-up will restart.

11. Serial Channel (SIO/UART)

11.1 Overview

Serial channel has the modes shown below.

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

Their features are given in the following.

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

  The usable double buffer function.
- I/O Interface Mode
  - Transfer Mode: the half duplex (transmit/receive), the full duplex
  - Clock: Output (fixed rising edge) /Input (selectable rising/falling edge)
  - Make it possible to specify the interval time of continuous transmission.
- UART Mode
  - Data length: 7 bits, 8bits, 9bits
  - Add parity bit (to be against 9bits data length)
  - Serial links to use wake-up function
  - Handshaking function with $\overline{\text{CTSx}}$ pin
- Output Signal using IR Carrier Pulses

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

11.2 Configuration

Figure 11-1 shows Serial channel block diagram.

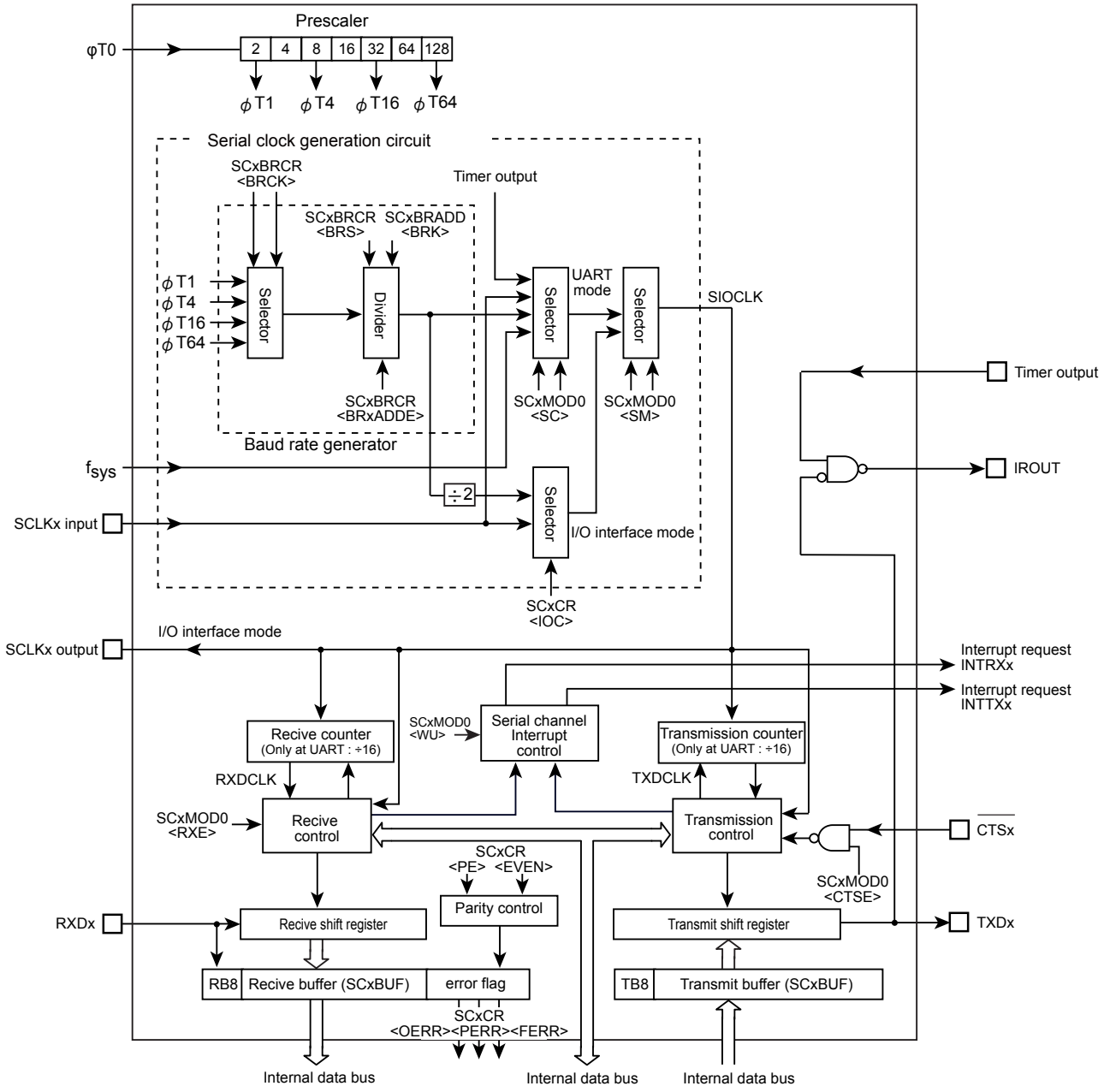


Figure 11-1 Serial Channel Block Diagram

11.3 Registers Description

11.3.1 Registers List

The table below shows control registers and their addresses.

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

Register name		Address (Base+)
Enable register	SCxEN	0x0000
Buffer register	SCxBUF	0x0004
Control register	SCxCR	0x0008
Mode control register 0	SCxMOD0	0x000C
Baud rate generator control register	SCxBRCR	0x0010
Baud rate generator control register 2	SCxBRADD	0x0014
Mode control register 1	SCxMOD1	0x0018
Mode control register 2	SCxMOD2	0x001C

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

11.3.2 SCxEN (Enable Register)

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

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

11.3.3 SCxBUF (Buffer Register)

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

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

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

11.3.4 SCxCR (Control Register)

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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	RB8	R	Receive data bit 8 (For UART) 9th bit of the received data in the 9-bit UART mode.
6	EVEN	R/W	Parity (For UART) 0: Odd 1: Even Selects even or odd parity. "0" : odd parity, "1" : even parity. The parity bit may be used only in the 7- or 8-bit UART mode.
5	PE	R/W	Add parity (For UART) 0: Disabled 1: Enabled Controls enabling/ disabling parity. The parity bit may be used only in the 7- or 8-bit UART mode.
4	OERR	R	Over-run error flag (Note) 0: Normal operation 1: Error
3	PERR	R	Parity / Under-run error flag (Note) 0: Normal operation 1: Error
2	FERR	R	Framing error flag (Note) 0: Normal operation 1: Error
1	SCLKS	R/W	Selecting input clock edge (For I/O Interface) Set to "0" in the clock output mode. 0:Data in the transmit buffer is sent to TXDx pin one bit at a time on the falling edge of SCLKx. Data from RXDx pin is received in the receive buffer one bit at a time on the rising edge of SCLKx. In this case, the SCLKx starts from high level. 1:Data in the transmit buffer is sent to TXDx pin one bit at a time on the rising edge of SCLKx. Data from RXDx pin is received in the receive buffer one bit at a time on the falling edge of SCLKx. In this case, the SCLKx starts from low level.
0	IOC	R/W	Selecting clock (For I/O Interface) 0: Baud rate generator 1: SCLK pin input

Note: Any error flag (OERR, PERR, FERR) is cleared to "0" when read.

11.3.5 SCxMOD0 (Mode Control Register 0)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	TB8	CTSE	RXE	WU	SM		SC	
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	TB8	R/W	Transmit data bit 8 (For UART) Writes the 9th bit of transmit data in the 9-bit UART mode.
6	CTSE	R/W	Handshake function control (For UART) 0: CTS disabled 1: CTS enabled Controls handshake function. Setting "1" enables handshake function using $\overline{CTS}$ pin.
5	RXE	R/W	Receive control (Note1)(Note2) 0: Disabled 1: Enabled
4	WU	R/W	Wake-up function (For UART) 0: Disabled 1: Enabled This function is available only at 9-bit UART mode. In other mode, this function has no meaning. In it is Enabled, Interrupt only when RB9 = "1" at 9-bit UART mode.
3-2	SM[1:0]	R/W	Specifies transfer mode. 00: I/O interface mode 01: 7-bit length UART mode 10: 8-bit length UART mode 11: 9-bit length UART mode
1-0	SC[1:0]	R/W	Serial transfer clock (For UART) 00: Timer output 01: Baud rate generator 10: Internal clock fsys 11: External clock (SCLK input) (As for the I/O interface mode, the serial transfer clock can be set in the control register (SCxCR).

Note 1: Set <RXE> to "1" after setting each mode register (SCxMOD0, SCxMOD1 and SCxMOD2).

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

11.3.6 SCxMOD1 (Mode Control Register 1)

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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	I2SC	R/W	IDLE 0: Stop 1: Operate Specifies the IDLE mode operation.
6-5	FDPX[1:0]	R/W	Transfer mode setting 00: Transfer prohibited 01: Half duplex (Receive) 10: Half duplex (Transmit) 11: Full duplex Configures the transfer mode in the I/O interface mode.
4	TXE	R/W	Transmit control (Note1)(Note2) 0 :Disabled 1: Enabled This bit enables transmission and is valid for all the transfer modes.
3-1	SINT[2:0]	R/W	Interval time of continuous transmission (For I/O interface) 000: None 001: 1 x SCLK 010: 2 x SCLK 011: 4 x SCLK 100: 8 x SCLK 101: 16 x SCLK 110: 32 x SCLK 111: 64 x SCLK This parameter is valid only for the I/O interface mode when SCLK pin output is selected. In other modes, this function has no meaning. Specifies the interval time of continuous transmission when double buffering is enabled in the I/O interface mode.
0	−	R/W	Write "0".

Note 1: Specify the all mode first and then enable the <TXE> bit.

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

11.3.7 SCxMOD2 (Mode Control Register 2)

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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	TBEMP	R	Transmit buffer empty flag. 0: Full 1: Empty If double buffering is disabled, this flag is insignificant. This flag shows that the transmit double buffers are empty. When data in the transmit double buffers is moved to the transmit shift register and the double buffers are empty, this bit is set to "1". Writing data again to the double buffers sets this bit to "0".
6	RBFLL	R	Receive buffer full flag. 0: Empty 1: Full If double buffering is disabled, this flag is insignificant. This is a flag to show that the receive double buffers are full. When a receive operation is completed and received data is moved from the receive shift register to the receive double buffers, this bit changes to "1" while reading this bit changes it to "0".
5	TXRUN	R	In transmission flag 0: Stop 1: Operate This is a status flag to show that data transmission is in progress. <TXRUN> and <TBEMP> bits indicate the following status.
4	SBLEN	R/W	STOP bit (for UART) 0 : 1-bit 1 : 2-bit This specifies the length of transmission stop bit in the UART mode. On the receive side, the decision is made using only a single bit regardless of the <SBLEN> setting.
3	DRCHG	R/W	Setting transfer direction 0: LSB first 1: MSB first Specifies the direction of data transfer in the I/O interface mode. In the UART mode, set this bit to LSB first.
2	WBUF	R/W	Double-buffer 0: Disabled 1 : Enabled This parameter enables or disables the transmit/receive double buffers to transmit (in both SCLK output/input modes) and receive (in SCLK output mode) data in the I/O interface mode and to transmit data in the UART mode. When receiving data in the I/O interface mode (SCLK input) and UART mode, double buffering is enabled in both cases that 0 or 1 is set to <WBUF> bit.

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

Bit	Bit Symbol	Type	Function
1-0	SWRST[1:0]	R/W	Software reset Overwriting "01" in place of "10" generates a software reset. When this software reset is executed, the following bits are initialized and the transmit/receive circuit become initial state (Note1)(Note2).

Register	Bit
SCxMOD0	<RXE>
SCxMOD1	<TXE>
SCxMOD2	<TBEMP>, <RBFLL>, <TXRUN>
SCxCR	<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.

11.3.8 SCxBRCR (Baud Rate Generator Control Register), SCxBRADD (Baud Rate Generator Control Register 2)

The division ratio of the baud rate generator can be specified in the registers shown below.

SCxBRCR

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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	−	R/W	Write "0".
6	BRADDE	R/W	N + (16 − K)/16 divider function (For UART) 0: disabled 1: enabled This division function can only be used in the UART mode.
5-4	BRCK[1:0]	R/W	Select input clock to the baud rate generator. 00: φT1 01: φT4 10: φT16 11: φT64
3-0	BRS[3:0]	R/W	Division ratio "N" 0000: 16 0001: 1 0010: 2 ... 1111: 15

SCxBRADD

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

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

Table 11-1 lists the settings of baud rate generator division ratio.

Table 11-1 Setting division ratio

	<BRADDE> = "0"	<BRADDE> = "1" (Note1) (Only UART mode)
<BRS>	Specify "N" (Note2) (Note3)	
<BRK>	No setting required	Specify "K" (Note4)
Division ratio	Divide by N	$N + \frac{(16 - K)}{16}$ division.

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

11.4 Operation in Each Mode

Table 11-2 shows the modes and data formats.

Table 11-2 Mode and Data format

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

Mode 0 is a synchronous communication and can be used to extend I/O. This mode transmits and receives data in synchronization with SCLK. SCLK can be used for both input and output.

The direction of data transfer can be selected from LSB first and MSB first. This mode is not allowed either to use parity bits or STOP bits.

The mode 1, mode 2 and mode 3 are asynchronous modes and the transfer direction is fixed to the LSB first.

Parity bits can be added in the mode 1 and mode 2. The mode 3 has a wakeup function in which the master controller can start up slave controllers via the serial link (multi-controller system).

STOP bit in transmission can be selected from 1 bit and 2 bits. The STOP bit length in reception is fixed to a one bit.

11.5 Data Format

11.5.1 Data Format List

Figure 11-2 shows data format.

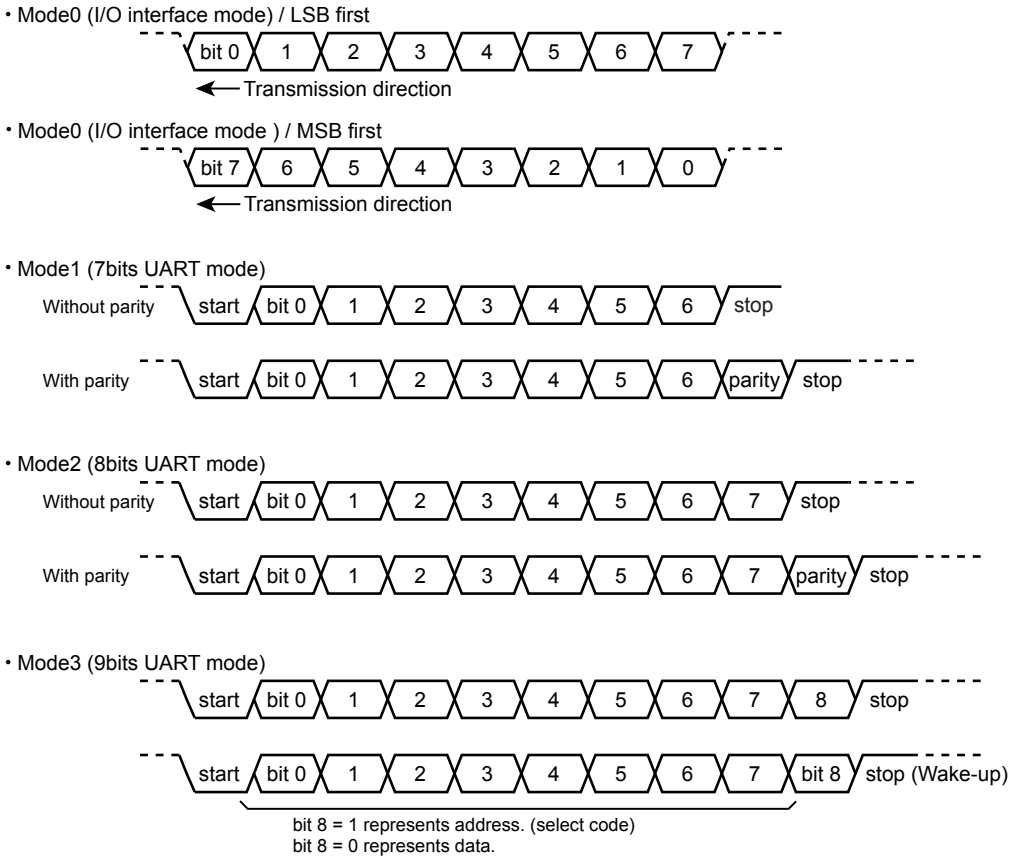


Figure 11-2 Data Format

11.5.2 Parity Control

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

Setting "1" to SCxCR<PE> enables the parity.

The <EVEN> bit of SCxCR selects either even or odd parity.

11.5.2.1 Transmission

Upon data transmission, the parity control circuit automatically generates the parity with the data in the transmit buffer.

The parity bit will be stored in SCxBUF<TB7> in the 7-bit UART mode and SCxMOD<TB8> in the 8-bit UART mode.

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

11.5.2.2 Receiving Data

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

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

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

11.5.3 STOP Bit Length

The length of the STOP bit in the UART transmission mode can be selected from one bit or two bits by setting the SCxMOD2<SBLEN>. The length of the STOP bit data is determined as one-bit when it is received regardless of the setting of this bit.

11.6 Clock Control

The following figure shows the serial clock (SIOCLK) generation circuit. Before changing the serial clock setting, check if the setting satisfies AC electrical characteristics.

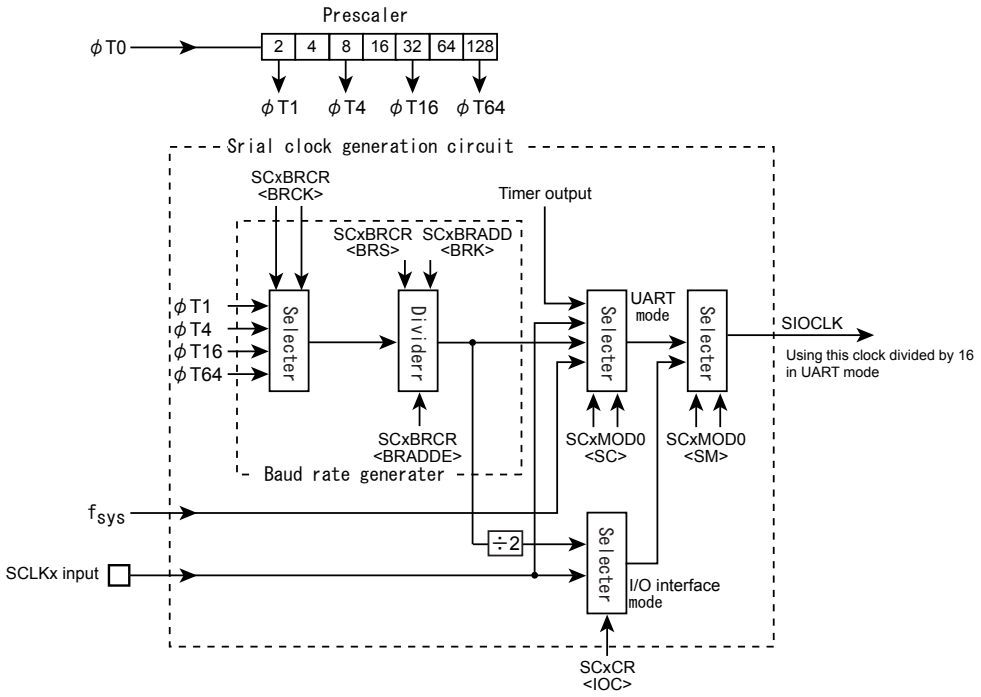


Figure 11-3 Serial clock generation circuit

11.6.1 Prescaler

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

Use the CGSYSCR register in the clock/mode control block to select the input clock φT0 of the prescaler.

The prescaler becomes active only when the baud rate generator is selected as a transfer clock by SCxMOD0<SC[1:0]> = "01".

11.6.2 Serial Clock Generation Circuit

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

11.6.2.1 Baud Rate Generator

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

(1) Baud Rate Generator input clock

The input clock of the baud rate generator is selected from the prescaler outputs divided by 2, 8, 32 and 128.

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

(2) Baud Rate Generator output clock

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

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

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

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

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

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

- Divide by N

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

- N + (16-K)/16 division

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

11.6.2.2 Clock Selection Circuit

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

Modes can be specified by setting the SCxMOD0<SM>.

The input clock in I/O interface mode is selected by setting SCxCR. The clock in UART mode is selected by setting SCxMOD0<SC>.

(1) Transfer Clock in I/O interface mode

Table 11-3 shows clock selection in I/O interface mode.

Table 11-3 Clock Selection in I/O Interface Mode

Mode SCxMOD0<SM>	Input/Output selection SCxCR<IOC>	Clock edge selection SCxCR<SCLKS>	Clock of use
I/O interface mode	SCLK output	Set to "0". (Fixed to the rising edge)	Divided by 2 of the baud rate generator output.
	SCLK input	Rising edge	SCLK input rising edge
		Falling edge	SCLK input falling edge

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

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

(2) Transfer clock in the UART mode

Table 11-4 shows the clock selection in the UART mode. In the UART mode, selected clock is divided by 16 in the receive counter or the transmit counter before use.

Table 11-4 Clock Selection in UART Mode

Mode SCxMOD0<SM>	Clock selection SCxMOD0<SC>
UART Mode	Timer output
	Baud rate generator
	fsys
	SCLKx input

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

- SCLK cycle > 2/fsys

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

Baud rates can be obtained by using the following formula.

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

- 2: In the case the timer prescaler clock ΦT1 (2 divition ratio) is selected.
- (TBxRG1 × 2): One clock cycle is a period that the timer output is inverted twice.

To enable the TMR16A output, a timer output inverts when the value of the counter and that of T16AxRG match. The SIOCLK clock frequency is "Setting value of T16AxRG × 2".

Baud rates can be obtained by using the following formula.

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

One clock cycle is a period that the timer output is inverted twice.

11.6.3 Transmit/Receive Buffer

11.6.3.1 Configuration

Figure 11-4 shows the configuration of transmit buffer, receive buffer.

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

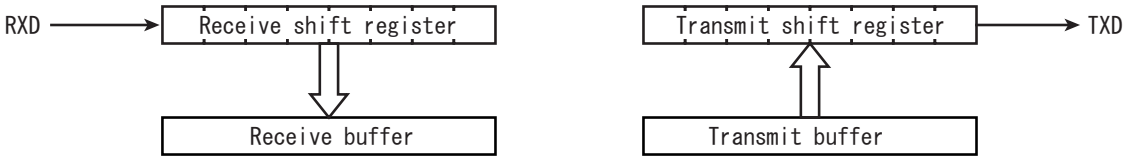


Figure 11-4 The Configuration of Buffer

11.6.3.2 Transmit/Receive Buffer

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

In the case of a receive mode, if SCLK input is set in the I/O interface mode or the UART mode is selected, it's double buffered despite the <WBUF> settings. In other modes, it's according to the <WBUF> settings.

Table 11-5 shows correlation between modes and buffers.

Table 11-5 Mode and buffer Composition

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

11.7 Status Flag

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

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

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

11.8 Error Flag

Three error flags are provided in the SCxCR register. The meaning of the flags is changed depending on the modes. The table below shows the meanings in each mode.

These flags are cleared to "0" after reading the SCxCR register.

Mode	Flag		
	<OERR>	<PERR>	<FERR>
UART	Over-run error	Parity error	Framing error
I/O Interface (SCLK input)	Over-run error	Under-run error (When using double buffer)	Fixed to 0
		Fixed to 0 (When a double buffer unused)	
I/O Interface (SCLK output)	Undefined	Undefined	Fixed to 0

11.8.1 OERR Flag

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

In the I/O interface with SCLK output mode, the SCLK output stops upon setting the flag.

Note: To switch the I/O interface SCLK output mode to other modes, read the SCxCR register and clear the over-run flag.

11.8.2 PERR Flag

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

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

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

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

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

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

11.8.3 FERR Flag

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

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

11.9 Receive

11.9.1 Receive Counter

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

In the UART mode, sixteen SIOCLK clock pulses are used in receiving a single data bit and the data symbol is sampled at the seventh, eighth, and ninth pulses. From these three samples, majority logic is applied to decide the received data.

11.9.2 Receive Control Unit

11.9.2.1 I/O interface mode

In the SCLK output mode with SCxCR <IOC> set to "0", the RXDx pin is sampled on the rising edge of the shift clock outputted to the SCLKx pin.

In the SCLK input mode with SCxCR <IOC> set to "1", the serial receive data RXDx pin is sampled on the rising or falling edge of SCLKx pin input signal depending on the SCxCR <SCLKS> setting.

11.9.2.2 UART Mode

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

11.9.3 Receive Operation

11.9.3.1 Receive Buffer

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

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

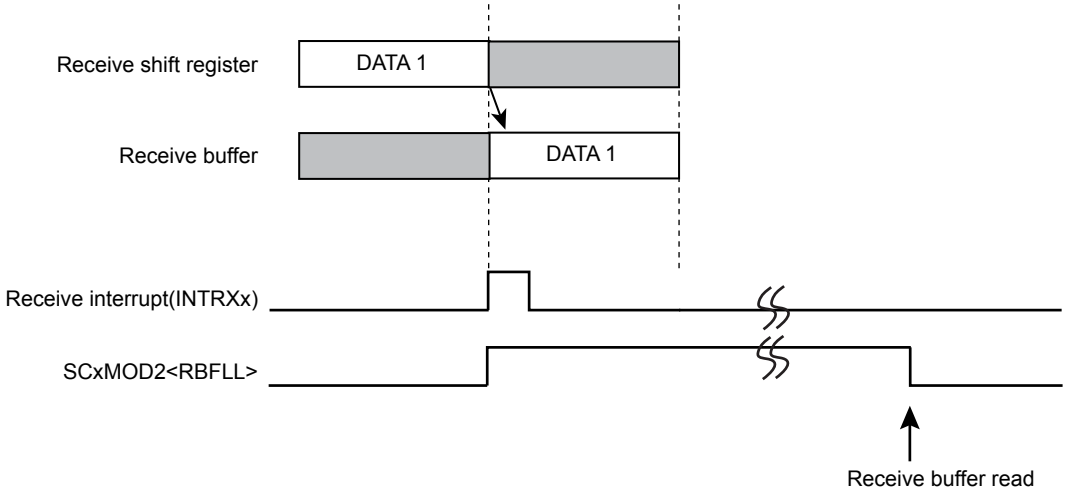


Figure 11-5 Receive Buffer Operation

11.9.3.2 I/O interface mode with SCLK output

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

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

(1) Case of single buffer

Stop SCLK output after receiving a data. In this mode, I/O interface can transfer each data with the transfer device by hand-shake.

When the data in a buffer is read, SCLK output is restarted.

(2) Case of double buffer

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

When the data is read, SCLK output is restarted.

11.9.3.3 Read Received Data

Read the received data from the receive buffer (SCxBUF).

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

11.9.3.4 Wake-up Function

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

11.9.3.5 Over-run Error

The over-run error is occurred and set over-run flag without completing data read before receiving the next data. When over-run error is occurred, a content of receive buffer and SCxCR<RB8> is not lost, but a content of receive shift register is lost.

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

Note: When the mode is changed from I/O interface SCLK output mode to the other mode, read SCxCR and clear over-run flag.

11.10 Transmission

11.10.1 Transmission Counter

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

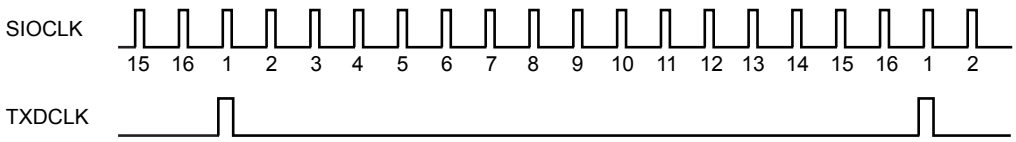


Figure 11-6 Generation of Transmission Clock

11.10.2 Transmission Control

11.10.2.1 I/O Interface Mode

In the SCLK output mode with SCxCR<IOC> set to "0", each bit of data in the transmit buffer is outputted to the TXDx pin on the falling edge of the shift clock outputted from the SCLKx pin.

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

11.10.2.2 UART Mode

When the transmit data is written in the transmit buffer, data transmission is initiated on the rising edge of the next TXDCLK and the transmit shift clock signal is also generated.

11.10.3 Transmit Operation

11.10.3.1 Operation of Transmission Buffer

If double buffering is disabled, the CPU writes data only to transmit shift register and the transmit interrupt INTTXx is generated upon completion of data transmission.

If double buffering is enabled, data written to the transmit buffer is moved to the transmit shift register. The INTTXx interrupt is generated at the same time and the transmit buffer empty flag (SCxMOD2<TBEMP>) is set to "1". This flag indicates that the next transmit data can be written. When the next data is written to the transmit buffer, the <TBEMP> flag is cleared to "0".

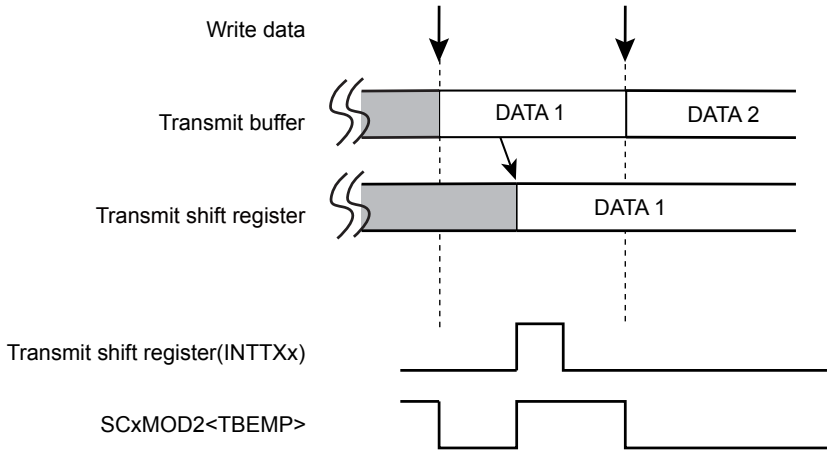


Figure 11-7 Operation of Transmission Buffer (Double-buffer is enabled)

11.10.3.2 I/O interface Mode/Transmission by SCLK Output

If SCLK is set to generate clock the I/O interface mode, the SCLK output automatically stops when all data transmission is completed and underrun error will not occur.

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

(1) Single Buffer

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

(2) Double Buffer

The SCLK output stops upon completion of data transmission of the transmit shift register and the transmit buffer. The SCLK output resumes when the next data is written in the buffer.

11.10.3.3 Under-run error

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

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

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

11.11 Handshake function

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

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

Note 1: If the $\overline{\text{CTS}}$ signal is set to "H" during transmission, the next data transmission is suspended after the current transmission is completed. (Point a in Figure 11-9)

Note 2: Data transmission starts on the first falling edge of the TXDCLK clock after $\overline{\text{CTS}}$ is set to "L". (Point b in Figure 11-9)

Although no $\overline{\text{RTS}}$ pin is provided, a handshake control function can easily implemented by assigning one bit of the port for the $\overline{\text{RTS}}$ function. By setting the port to "High" level upon completion of data reception (in the receive interrupt routine), the transmit side can be requested to suspend data transmission.

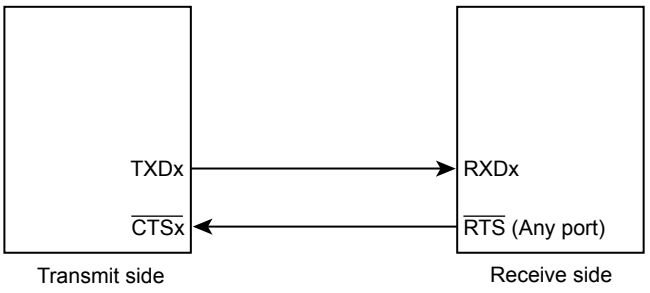


Figure 11-8 Handshake Function

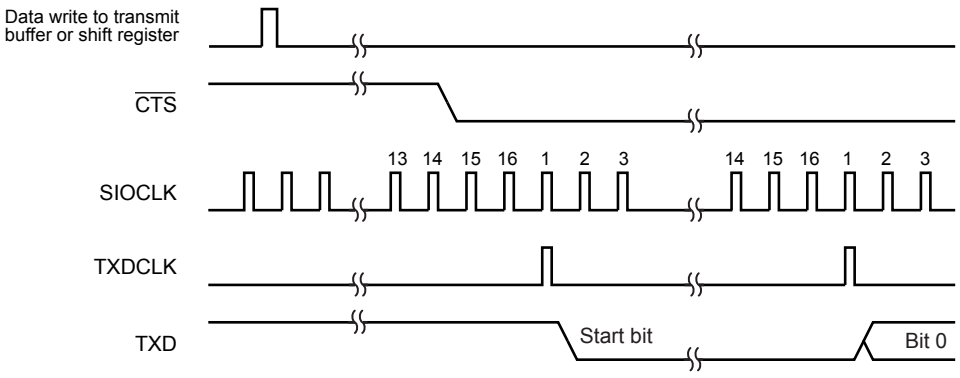


Figure 11-9 $\overline{\text{CTS}}$ Signal timing

11.12 Output Signal using IR Carrier Pulses

IROUT pin outputs a signal using IR carrier pulses.

IROUT signal is generated by the timer output signals and TXD signals. Since the timer used to generate pulses varies depending on products, refer to Chapter Product Information to select timers.

The timer should be set to generate pulse outputs in a desired cycle within a settable range.

To output IROUT signal from a port, in the port where IROUT pin is assigned, specify the function setting of function register (PxFRn) to IROUT, then enable IROUT signal to be output by control register (PxCR).

SIO receives pulses from the timer and combines the pulses with a TXD signal to output a signal to IROUT pin. Timer outputs and TXD signals are asynchronous, so that pulse distortion occurs when TXD signals are changed.

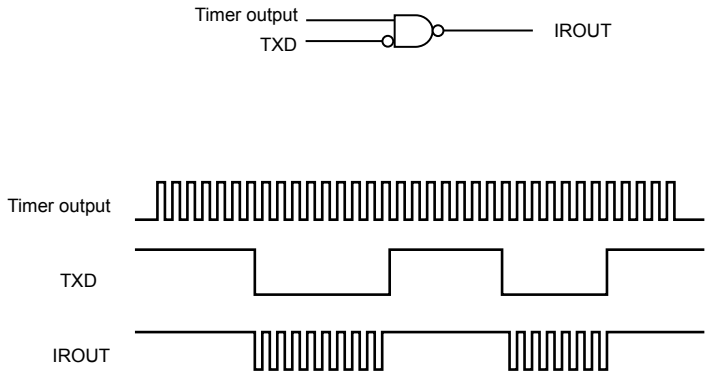


Figure 11-10 Output signal using IR carrier pulse

11.13 Interrupt/Error Generation Timing

11.13.1 RX Interrupts

Figure 11-11 shows the data flow of receive operation and the route of read.

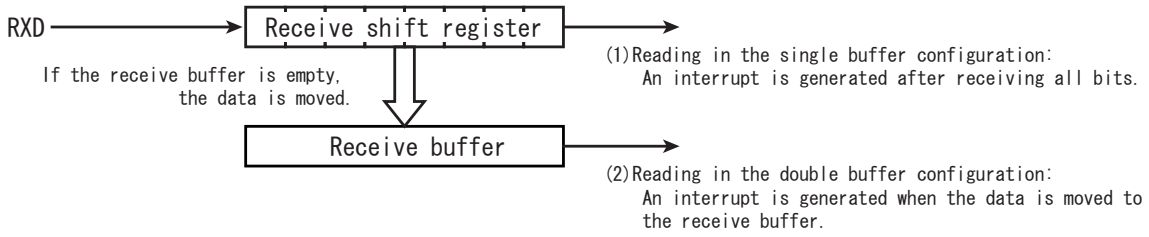


Figure 11-11 Receive Buffer Configuration Diagram

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

Buffer Configurations	UART modes	IO interface modes
Single Buffer	−	・Immediately after the raising / falling edge of the last SCLK (Rising or falling is determined according to SCxCR<SCLKS> setting.)
Double Buffer	・Around the center of the first stop bit	・Immediately after the raising / falling edge of the last SCLK (Rising or falling is determined according to SCxCR<SCLKS> setting.) ・On data transfer from the shift register to the buffer by reading buffer.

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

11.13.2 TX interrupts

Figure 11-12 shows the data flow of transmit operation and the route of read.

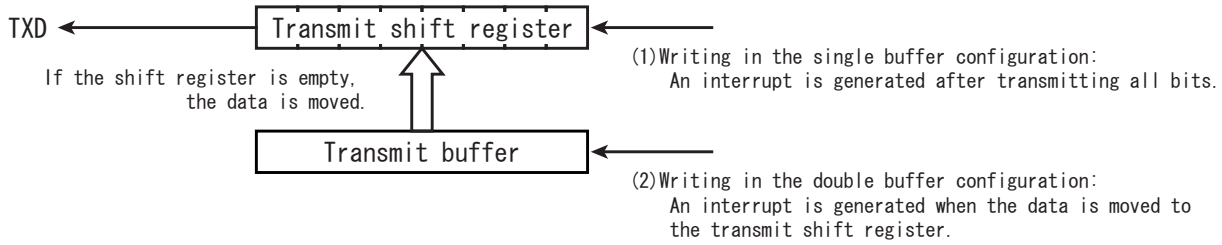


Figure 11-12 Transmit Buffer Configuration Diagram

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

Buffer Configurations	UART modes	IO interface modes
Single Buffer	Just before the stop bit is sent	Immediately after the raising / falling edge of the last SCLK (Rising or falling is determined according to SCxCR<SCLKS> setting.)
Double Buffer	When a data is moved from the transmit buffet to the transmit shift register.	

Note: If double buffer is enabled, a interrupt is also generated when the data is moved from the buffer to the shift register by writing to the buffer.

11.13.3 Error Generation

11.13.3.1 UART Mode

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

11.13.3.2 I/O Interface Mode

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

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

11.14 Software Reset

Software reset is generated by writing SCxMOD2<SWRST[1:0]> as "10" followed by "01".

As a result, SCxMOD0<RXE>, SCxMOD1<TXE>, SCxMOD2<TBEMP><RBFLL><TXRUN>, SCxCR <OERR><PERR><FERR> are initialized. And the receive circuit and the transmit circuit become initial state. Other states are maintained.

11.15 Operation in Each Mode

11.15.1 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.

11.15.1.1 Transmitting Data

(1) SCLK Output Mode

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

  Data is output from the TXDx pin and the clock is output from the SCLKx pin each time the CPU writes data to the transmit buffer. When all data is output, an interrupt (INTTXx) is generated.

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

  Data is moved from the transmit buffer to the transmit shift register when the CPU writes data to the transmit buffer while data transmission is halted or when data transmission from the transmit buffer (shift register) is completed. Simultaneously, the transmit buffer empty flag SCxMOD2<TBEMP> is set to "1", and the INTTXx interrupt is generated.

  When data is moved from the transmit buffer to the transmit shift register, if the transmit buffer has no data to be moved to the transmit shift register, INTTXx interrupt is not generated and the SCLK output stops.

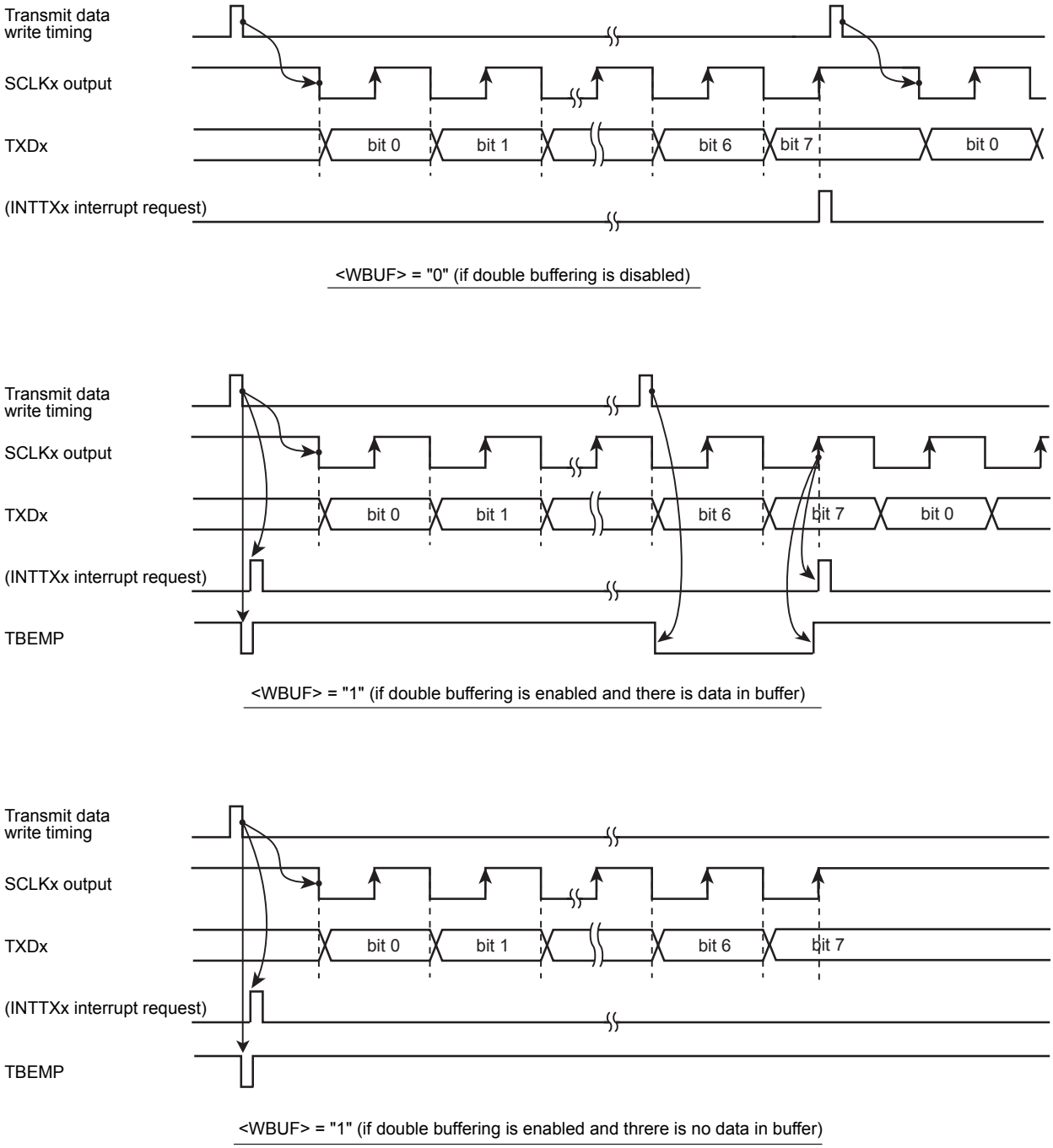


Figure 11-13 Transmit Operation in the I/O Interface Mode (SCLK Output Mode)

(2) SCLK Input Mode

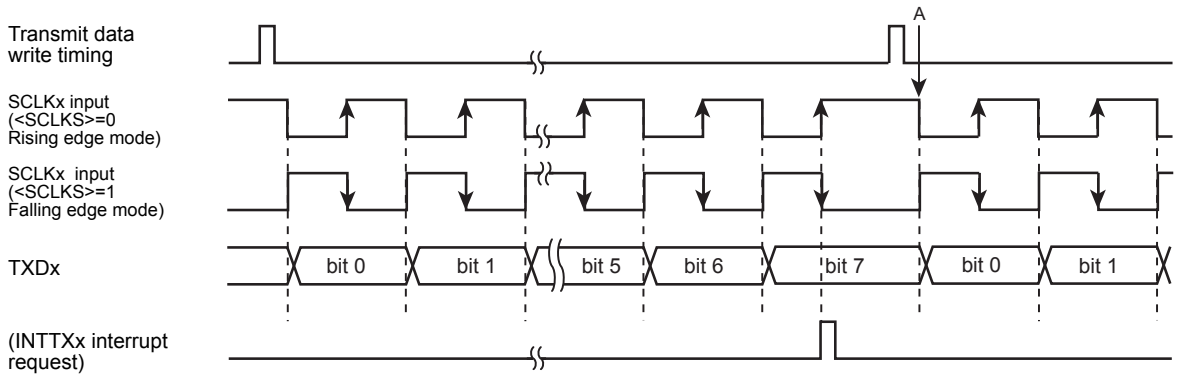
- If double buffering is disabled (SCxMOD2<WBUF> = "0")

  If the SCLK is input in the condition where data is written in the transmit buffer, 8-bit data is outputted from the TXDx pin. When all data is output, an interrupt INTTXx is generated. The next transmit data must be written before the timing point "A" as shown in Figure 11-14.

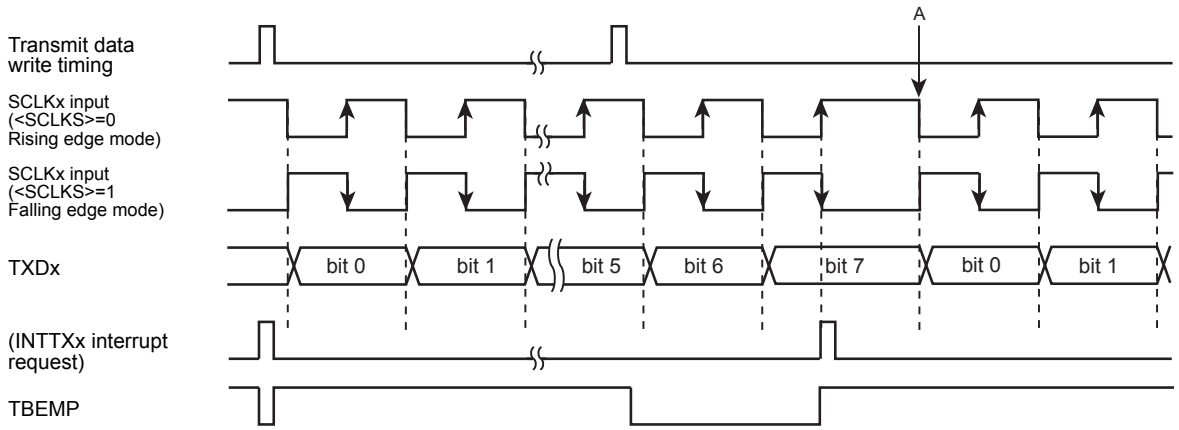
- If double buffer is enabled (SCxMOD2<WBUF> = "1")

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

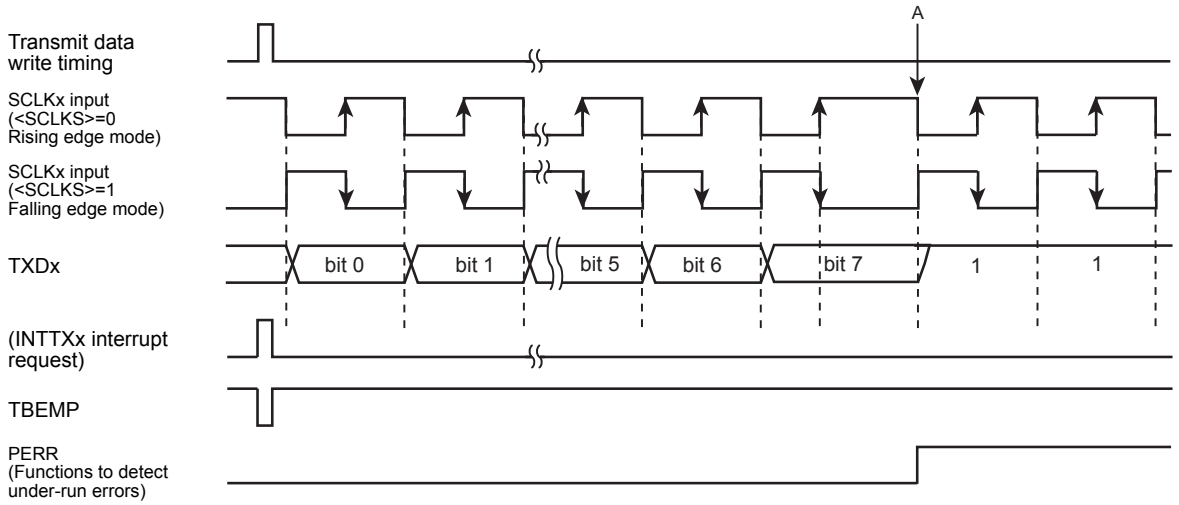
  If the SCLK input becomes active while no data is in the transmit buffer, although the internal bit counter is started, an under-run error occurs and 8-bit dummy data (0xFF) is sent.



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



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



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

Figure 11-14 Transmit Operation in the I/O Interface Mode (SCLK Input Mode)

11.15.1.2 Receive

(1) SCLK Output Mode

The SCLK output can be started by setting the receive enable bit SCxMOD0<RXE> to "1".

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

  A clock pulse is outputted from the SCLKx pin and the next data is stored into the shift register each time the CPU reads received data. When all the 8 bits are received, the INTRXx interrupt is generated.

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

  Data stored in the shift register is moved to the receive buffer and the receive buffer can receive the next frame. A data is moved from the shift register to the receive buffer, the receive buffer full flag SCxMOD2<RBFLL> is set to "1" and the INTRXx is generated.

  While data is in the receive buffer, if the data cannot be read from the receive buffer before completing reception of the next 8 bits, the INTRXx interrupt is not generated and the SCLK output stops. In this state, reading data from the receive buffer allows data in the shift register to move to the receive buffer and thus the INTRXx interrupt is generated and data reception resumes.

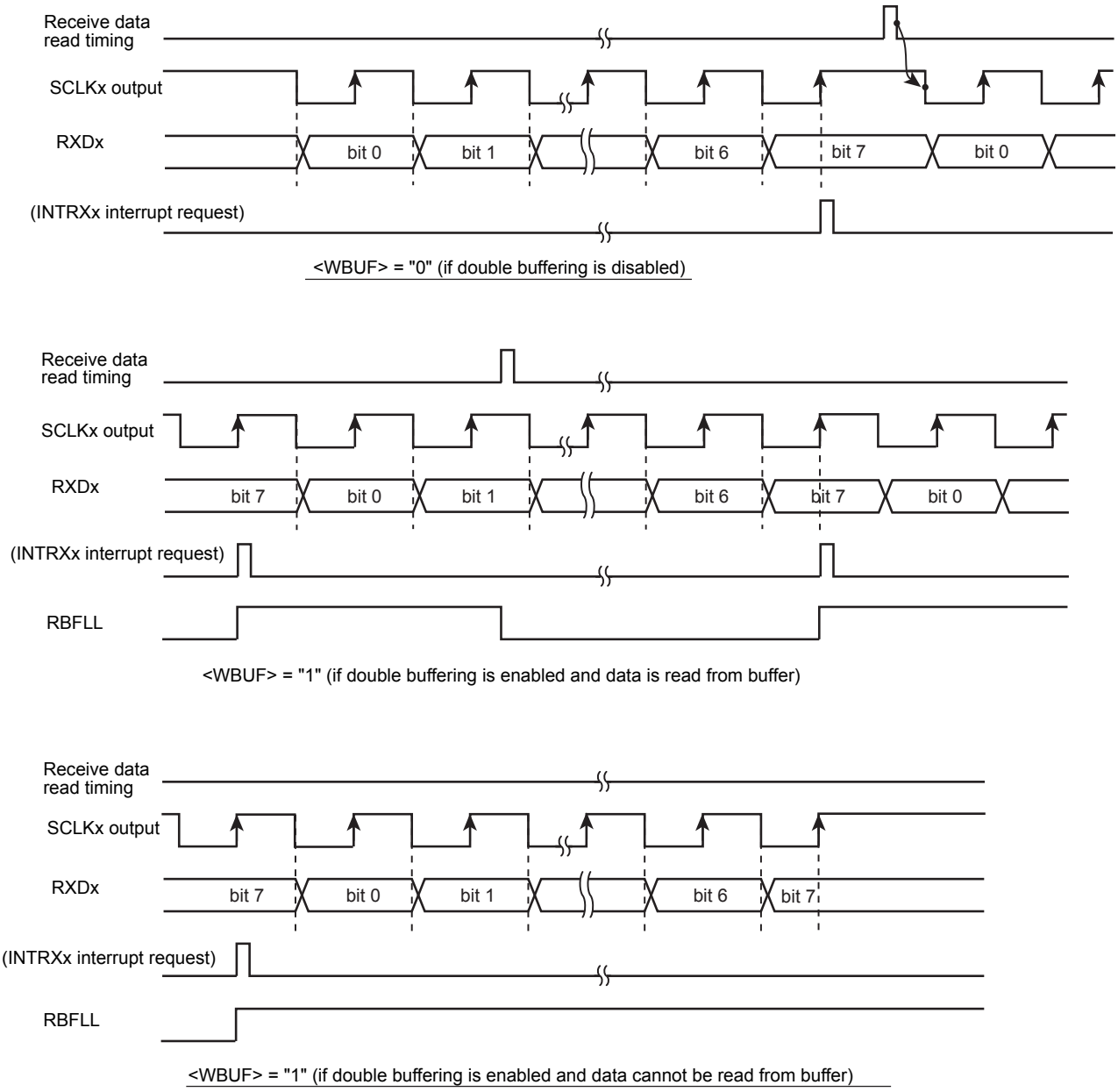


Figure 11-15 Receive Operation in the I/O Interface Mode (SCLK Output Mode)

(2) SCLK Input Mode

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

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

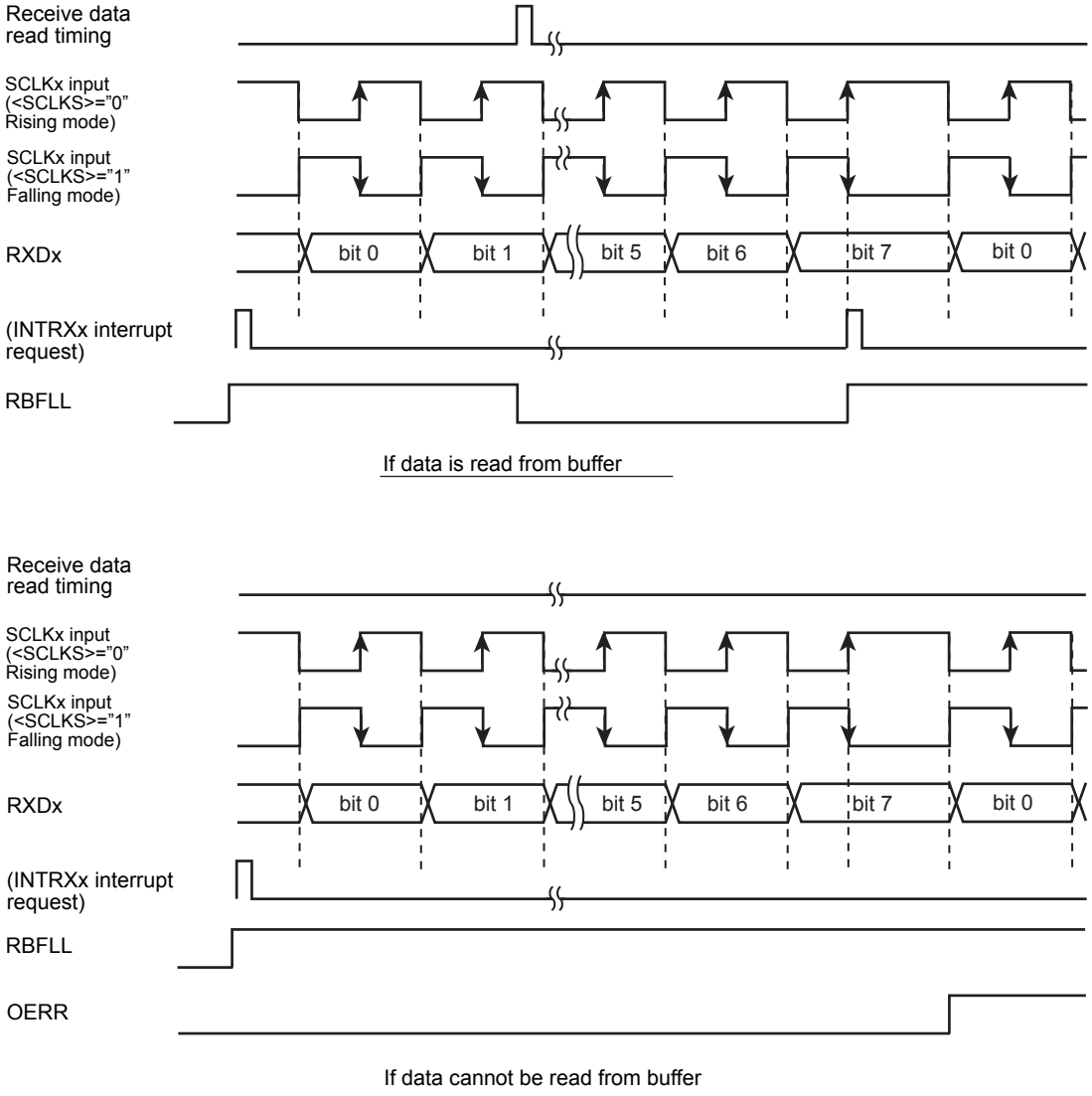


Figure 11-16 Receive Operation in the I/O Interface Mode (SCLK Input Mode)

11.15.1.3 Transmit and Receive (Full-duplex)

(1) SCLK Output Mode

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

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

  Subsequently, 8 bits of data are shifted into receive buffer and the INTRXx receive interrupt is generated. Concurrently, 8 bits of data written to the transmit buffer are outputted from the TXDx pin, the INTTXx transmit interrupt is generated when transmission of all data bits has been completed. Then, the SCLK output stops.

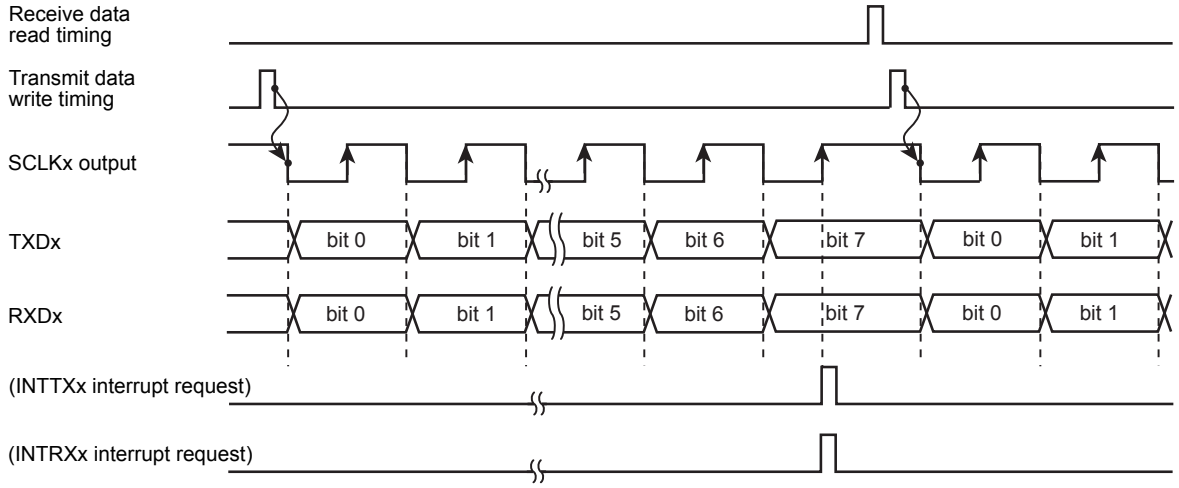
  The next round of data transmission and reception starts when the data is read from the receive buffer and the next transmit data is written to the transmit buffer by the CPU. The order of reading the receive buffer and writing to the transmit buffer can be freely determined. Data transmission is resumed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

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

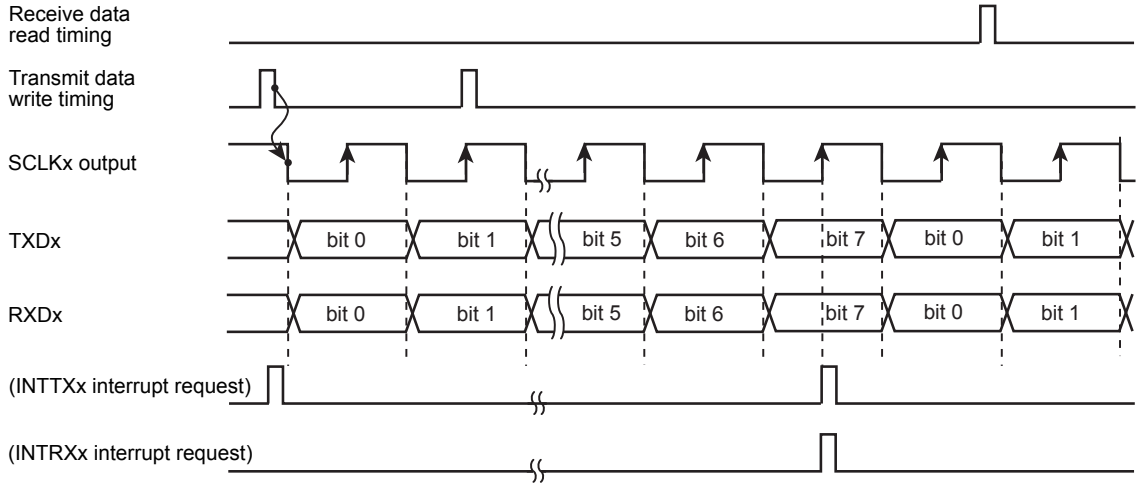
  SCLK is outputted when the CPU writes data to the transmit buffer.

  8 bits of data are shifted into the receive shift register, moved to the receive buffer, and the INTRXx interrupt is generated. While 8 bits of data is received, 8 bits of transmit data is outputted from the TXDx pin. When all data bits are sent out, the INTTXx interrupt is generated and the next data is moved from the transmit buffer to the transmit shift register.

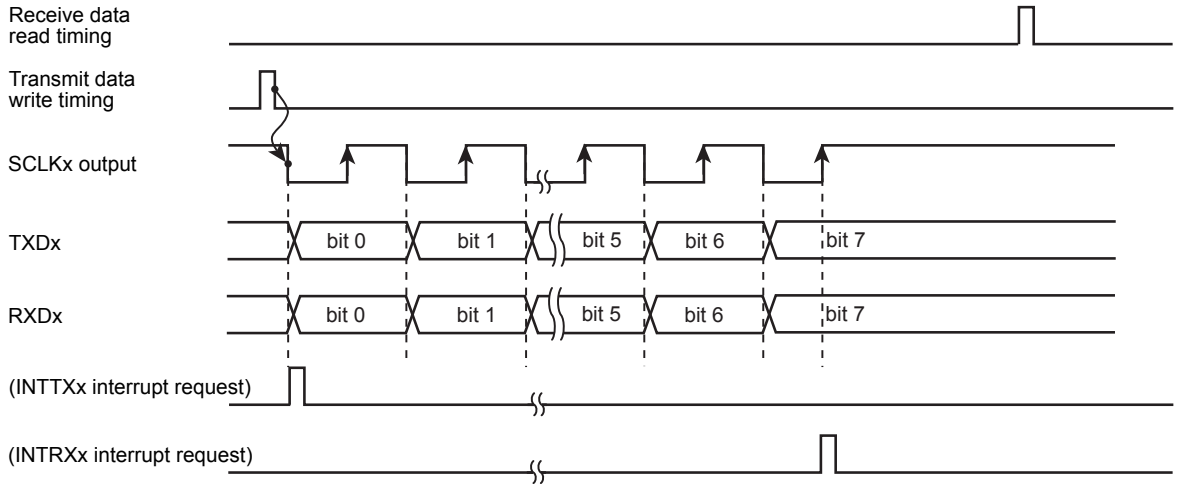
  If the transmit buffer has no data to be moved to the transmit buffer (SCxMOD2<TBEMP> = "1") or when the receive buffer is full (SCxMOD2<RBFULL> = "1"), the SCLK output is stopped. When both conditions, receive data is read and transmit data is written, are satisfied, the SCLK output is resumed and the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



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



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



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

Figure 11-17 Transmit/Receive Operation in the I/O Interface Mode (SCLK Output Mode)

(2) SCLK Input Mode

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

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

  8-bit data written in the transmit buffer is outputted from the TXDx pin and 8 bit of data is shifted into the receive buffer when the SCLK input becomes active. The INTTXx interrupt is generated upon completion of data transmission. The INTRXx interrupt is generated when the data is moved from shift register to receive buffer after completion of data reception.

  Note that transmit data must be written into the transmit buffer before the SCLK input for the next frame (data must be written before the point A in Figure 11-18). Data must be read before completing reception of the next frame data.

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

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

  Note that transmit data must be written into the transmit buffer before the SCLK input for the next frame (data must be written before the point A in Figure 11-18). Data must be read before completing reception of the next frame data.

  Upon the SCLK input for the next frame, transmission from transmit shift register (in which data has been moved from transmit buffer) is started while receive data is shifted into receive shift register simultaneously.

  If data in receive buffer has not been read when the last bit of the frame is received, an over-run 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 and the dummy data (0xff) is output.

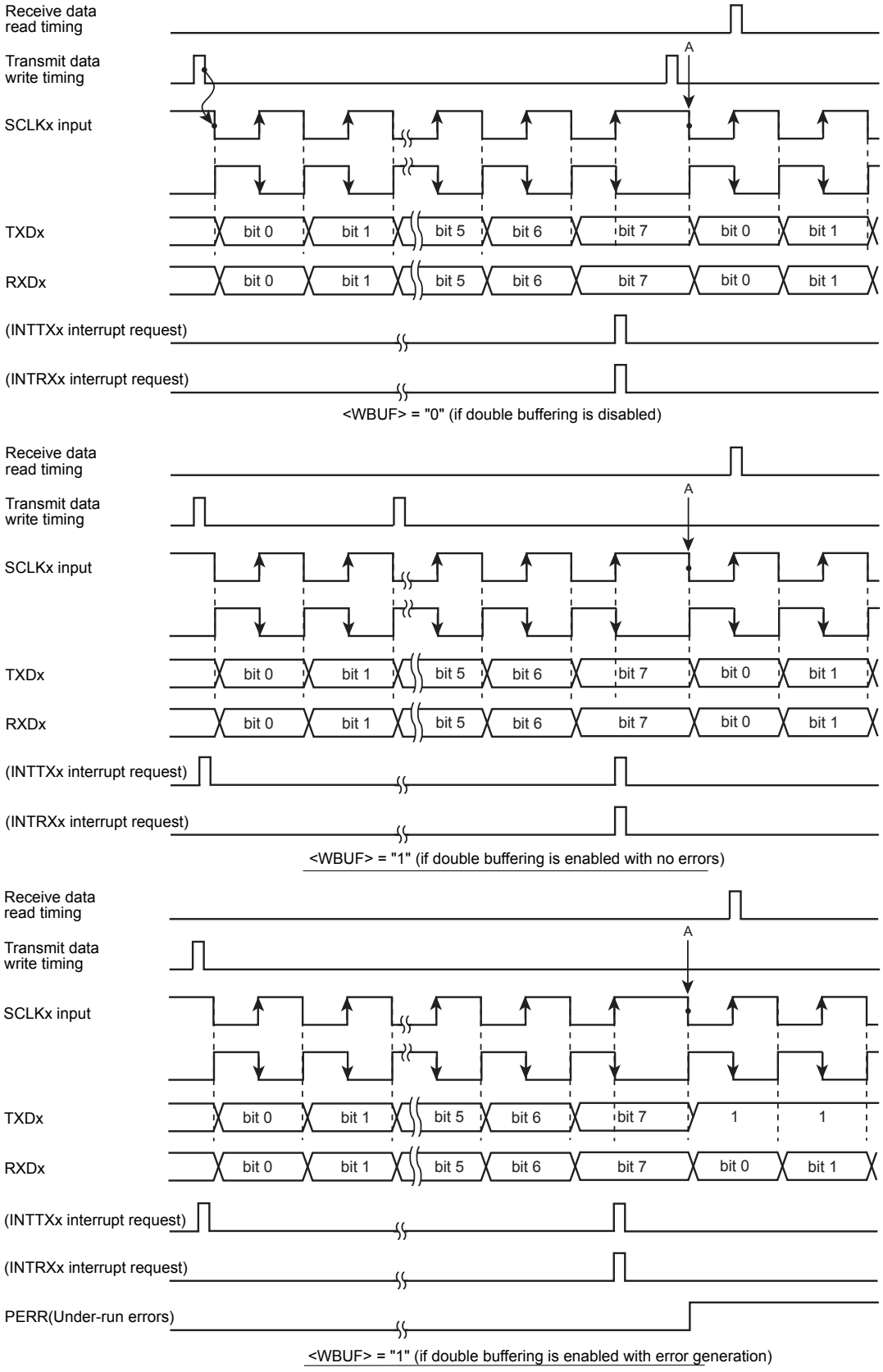


Figure 11-18 Transmit/Receive Operation in the I/O Interface Mode (SCLK Input Mode)

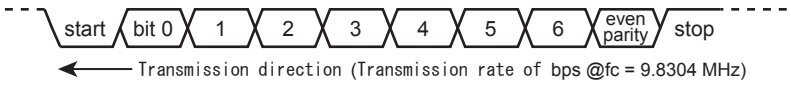
11.15.2 Mode 1 (7-bit UART mode)

The 7-bit UART mode can be selected by setting the serial mode control register (SCxMOD<SM[1:0]>) to "01".

In this mode, parity bits can be added to the transmit data stream; the serial mode control register (SCxCR<PE>) controls the parity enable/disable setting.

When <PE> is set to "1" (enable), either even or odd parity may be selected using the SCxCR<EVEN> bit. The length of the stop bit can be specified using SCxMOD2<SBLEN>.

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



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

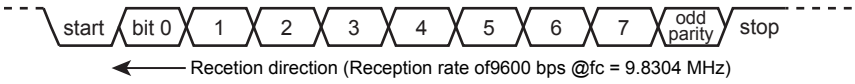
		7	6	5	4	3	2	1	0	
SCxMOD0	←	x	0	-	0	0	1	0	1	Set 7-bit UART mode
SCxCR	←	x	1	1	x	x	x	0	0	Even parity enabled
SCxBRCR	←	0	0	1	0	0	1	0	0	Set 2400bps
SCxBUF	←	*	*	*	*	*	*	*	*	Set transmit data

x: don't care - : no change

11.15.3 Mode 2 (8-bit UART mode)

The 8-bit UART mode can be selected by setting SCxMOD0<SM[1:0]> to "10". In this mode, parity bits can be added and parity enable/disable is controlled using SCxCR<PE>. If <PE> = "1" (enabled), either even or odd parity can be selected using SCxCR<EVEN>.

The control register settings for receiving data in the following format are as follows:



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

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

x: don't care - : no change

11.15.4 Mode 3 (9-bit UART mode)

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

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

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

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

11.15.4.1 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 SCxMOD0<WU> to "1".

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

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

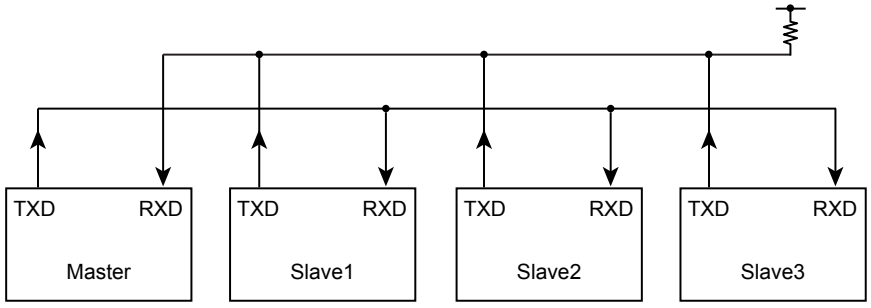
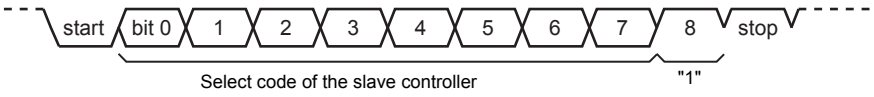


Figure 11-19 Serial Links to Use Wake-up Function

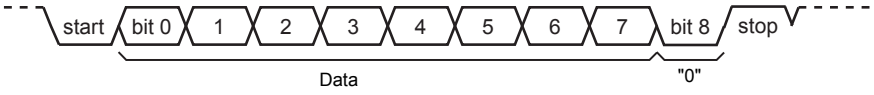
11.15.4.2 Protocol

1. Select the 9-bit UART mode for the master and slave controllers.
2. Set SCxMOD<WU> to "1" for the slave controllers to make them ready to receive data.
3. The master controller is to transmit a single frame of data that includes the slave controller select code (8 bits). In this, the most significant bit (bit 8) <TB8> must be set to "1".



4. Each slave controller receives the above data frame; if the code received matches with the controller's own select code, it clears the WU bit to "0".

5. The master controller transmits data to the designated slave controller (the controller of which SCxMOD<WU> bit is cleared to "0"). In this, the most significant bit (bit 8) <TB8> must be set to "0".



6. The slave controllers with the <WU> bit set to "1" ignore the receive data because the most significant bit (bit 8) <RB8> is set to "0" and thus no interrupt (INTRXx) is generated. Also, the slave controller with the <WU> bit set to "0" can transmit data to the master controller to inform that the data has been successfully received.

12. Serial Bus Interface (I2C/SIO)

12.1 Outline

Serial bus interface has two operation mode shown below.

- I2C bus mode
- Clock-synchronous 8-bit SIO mode

In the following examination, "x" indicates channel number.

12.2 Block Diagram

The block diagram is shown in Figure 12-1.

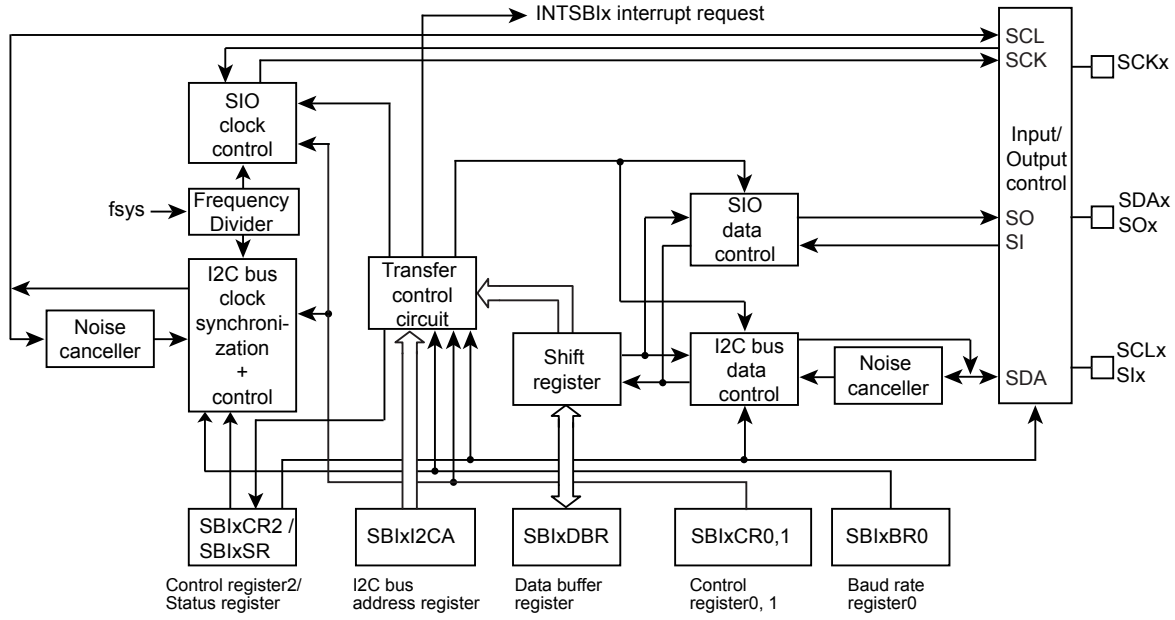
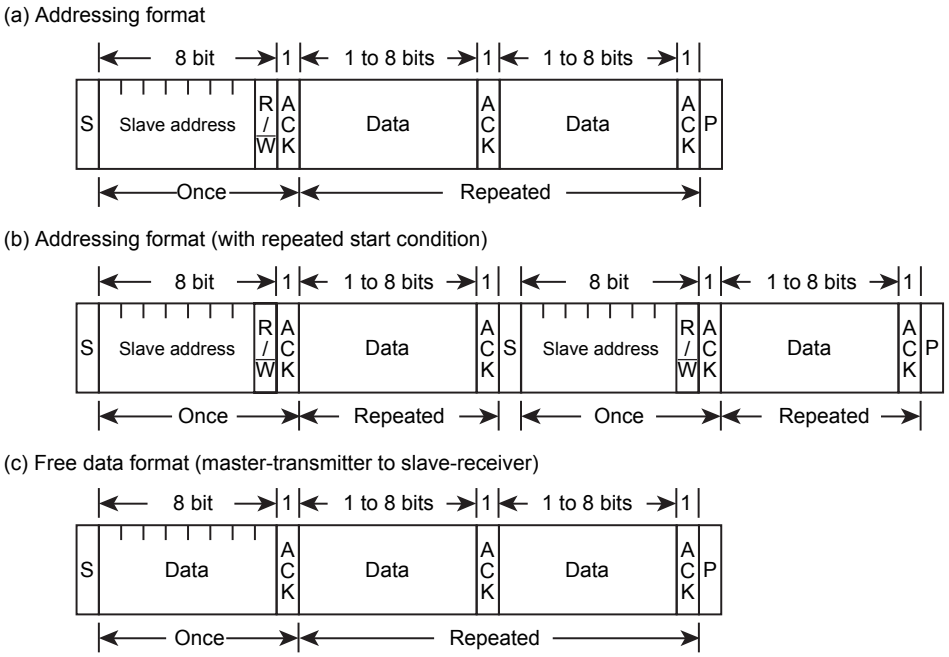


Figure 12-1 Block Diagram of Serial Bus Interface

12.3 I2C Bus Mode Data Format

Figure 12-2 shows the data formats used in the I2C bus mode.



Note) S : Start condition
R/W : Direction bit
ACK : Acknowledge bit
P : Stop condition

Figure 12-2 I2C Bus Mode Data Formats

12.4 Registers

12.4.1 Register List

The table below shows control registers and their addresses.

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

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

12.5 Control Registers in the I2C Bus Mode

The following registers control the serial bus interface in the I2C bus mode and provide its status information for monitoring.

12.5.1 SBIxCR0(Control register 0)

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

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

12.5.2 SBIxCR1(Control register 1)

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.

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

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

Note 2: For details on the SCL line clock frequency, refer to "12.6.2 Serial Clock".

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

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

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

12.5.3 SBIxCR2(Control register 2)

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

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

Note 1: Do not change the serial bus interface operating mode during the transfer.

Note 2: Changes to Port mode after conforming that SDAx/SCLx pin and SOx/SIx/SCKx pin are "High" level.

Note 3: Changes from port mode to I2C bus mode after conforming that SDAx/SCLx pin and SOx/SIx/SCKx pin are "High" level.

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

12.5.4 SBIxSR (Status Register)

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

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

12.5.5 SBIxBR0(Serial bus interface baud rate register 0)

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

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

12.5.6 SBIxDBR (Serial bus interface data buffer register)

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

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

Note 1: The transmission data must be written in to the register from the MSB (bit 7). The received data is stored in the LSB.

Note 2: Since SBIxDBR has independent buffers for writing and reading, a written data cannot be read. Thus, read-modify-write operation cannot be used.

12.5.7 SBIxI2CAR (I2C bus address register)

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7-1	SA[6:0]	R/W	Set the slave address when the SBI acts as a slave device.
0	ALS	R/W	Specify address recognition mode. 0: Recognize its slave address. 1: Do not recognize its slave address (free-data format).

Note 1: Please set theSBIxI2CAR<ALS> to "0", except when you use a free data format. It operates as a free data format when setting it to "1". Selecting the master fixes to transmission. Selecting the slave fixes to reception.

Note 2: Do not set SBIxI2CAR to "0x00" in slave mode. (If SBIxI2CAR is set to "0x00", it's recognized that the slave address matches the START byte ("0x01") of the I2C standard received in slave mode.)

12.6 Control in the I2C bus mode

12.6.1 Setting operation mode

Set operation mode by SBIxCR2<SBIM[1:0]>. When SBI is used as I2C bus mode, set <SBIM[1:0]> to "10".

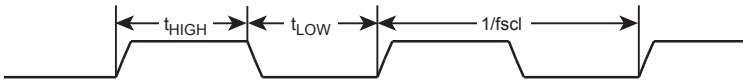
Note 1: Changes to Port mode after conforming that SDAx/SCLx pin and SOx/SIx/SCKx pin are "High" level.

Note 2: Changes from port mode to I2C bus mode after conforming that SDAx/SCLx pin and SOx/SIx/SCKx pin are "High" level.

12.6.2 Serial Clock

12.6.2.1 Clock source

SBIxCR1<SCK[2:0]> specifies the maximum frequency of the serial clock to be output from the SCLx pin in the master mode.



$t_{LOW} = 2^{n-1}/fsys + 58/fsys$
$t_{HIGH} = 2^{n-1}/fsys + 14/fsys$
$fscl = 1/(t_{LOW} + t_{HIGH})$
$= \frac{fsys}{2^n + 72}$

SBIxCR1<SCK[2:0]>	n
000	5
001	6
010	7
011	8
100	9
101	10
110	11

Figure 12-3 Clock source

Note: The maximum speeds in the standard and high-speed modes are specified to 100kHz and 400kHz respectively following the communications standards. Notice that the internal SCL clock frequency is determined by the fsys used and the calculation formula shown above.

12.6.2.2 Clock Synchronization

The I2C bus is driven by using the wired-AND connection due to its pin structure. The first master that pulls its clock line to the "Low" level overrides other masters producing the "High" level on their clock lines. This must be detected and responded by the masters producing the "High" level.

Clock synchronization assures correct data transfer on a bus that has two or more master.

For example, the clock synchronization procedure for a bus with two masters is shown below.

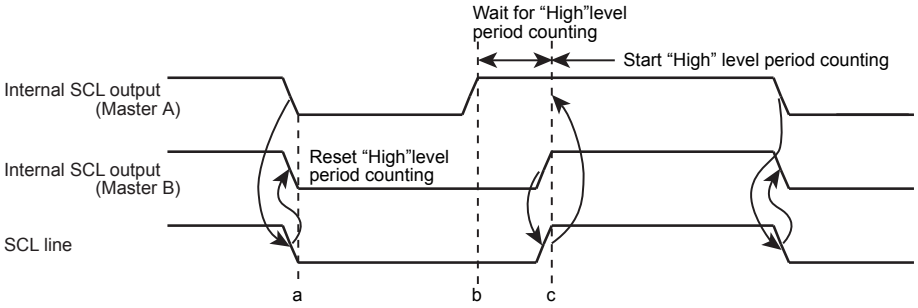


Figure 12-4 Example of Clock Synchronization

At the point a, Master A pulls its internal SCL output to the "Low" level, bringing the SCL bus line to the "Low" level. Master B detects this transition, resets its "High" level period counter, and pulls its internal SCL output level to the "Low" level.

Master A completes counting of its "Low" level period at the point b, and brings its internal SCL output to the "High" level. However, Master B still keeps the SCL bus line at the "Low" level, and Master A stops counting of its "High" level period counting. After Master A detects that Master B brings its internal SCL output to the "High" level and brings the SCL bus line to the "High" level at the point c, it starts counting of its "High" level period.

After that Master finishes counting the "High" level period, the Master pulls the SCLx pin to "Low" and the SCL bus line becomes "Low".

This way, the clock on the bus is determined by the master with the shortest "High" level period and the master with the longest "Low" level period among those connected to the bus.

12.6.3 Setting the Acknowledgement Mode

Setting SBIxCR1<ACK> to "1" selects the acknowledge mode.

When operating as a master, the SBI adds one clock for acknowledgment signal.

In slave mode, the clock for acknowledgement signals is counted.

In transmitter mode, the SBI releases the SDAx pin during clock cycle of acknowledgement to receive acknowledgement signals from the receiver.

In receiver mode, the SBI pulls the SDAx pin to the "Low" level during the clock cycle of acknowledgement and generates acknowledgement signals. Also in slave mode, if a general-call is received, the SBI pulls the SDAx pin to the "Low" level during the clock cycle of acknowledgement and generates acknowledgement signals.

By setting <ACK> to "0", the non-acknowledgment mode is activated. When operating as a master, the SBI does not generate clock for acknowledgement signals. In slave mode, the clock for acknowledgement signals is not counted.

12.6.4 Setting the Number of Bits per Transfer

SBIxCR1<BC[2:0]> specifies the number of bits of the next data to be transmitted or received.

Under the start condition, <BC[2:0]> is set to "000", causing a slave address and the direction bit to be transferred in a packet of eight bits. At other times, <BC[2:0]> keeps a previously programmed value.

12.6.5 Slave Addressing and Address Recognition Mode

Setting "0" to SBIxI2CAR<ALS> and set a slave address in SBIxI2CAR<SA[6:0]> in the addressing format, and then the SBI recognizes a slave address transmitted by the master device and receives data in the addressing format.

If <ALS> is set to "1", the SBI does not recognize a slave address and receives data in the free data format. In the case of free data format, a slave address and a direction bit are not recognized; they are recognized as data immediately after generation of the start condition.

12.6.6 Configuring the SBI as a Master or a Slave

Setting SBIxCR2<MST> to "1" configures the SBI to operate as a master device.

Setting <MST> to "0" configures the SBI as a slave device.

<MST> is cleared to "0" by the hardware when SBI detects the stop condition on the bus or the arbitration lost.

12.6.7 Configuring the SBI as a Transmitter or a Receiver

Setting SBIxCR2<TRX> to "1" configures the SBI as a transmitter.

Setting <TRX> to "0" configures the SBI as a receiver.

<TRX> is cleared to "0" by the hardware when SBI detects the stop condition on the bus or the arbitration lost.

If SBI is used in free data format, <TRX> is not changed by the hardware.

If SBI is used in addressing format, <TRX> is set shown as below.

12.6.7.1 Master mode

As a master mode, if SBI receives acknowledgement from a slave device, <TRX> is set shown as below by a hardware.

If SBI does not acknowledgement, <TRX> retains the previous value.

- When the transmitted direction bit is "1", <TRX> is set to "0".
- When the transmitted direction bit is "0", <TRX> is set to "1".

12.6.7.2 Slave mode

As a slave mode, in case of addressing format, if below condition is satisfied, <TRX> is set depended on the direction bit which is sent by a master device.

- When the received slave address is as same as the value set in SBIxI2CAR.
- When SBI receives general-call

<TRX> is set shown as below.

- When the received direction bit is "1", <TRX> is set to "1".
- When the received direction bit is "0", <TRX> is set to "0".

12.6.8 Bus busy monitor

To conform the state of the bus, read SBIxSR<BB>.

<BB> is set to "1" when SBI detects the start condition on the bus and is cleared to "0" when SBI detects the stop condition on the bus.

When <BB> is "1", it is called as bus busy. When <BB> is "0", it is called as bus free.

The master device can generate the start condition in only bus free. It should be conform that <BB> is "0".

When <BB> is "1", SBI generates the start condition, the start condition is not generated and the arbitration lost is occurred.

12.6.9 Interrupt Service Request and Release

When INTSBIx is generated, SBIxCR2<PIN> is cleared to "0" and SBI is in interrupt service request state.

SBI pulls SCLx pin to "Low" level.

<PIN> is set to "1" when data is written to or read from SBIxDBR. When the program writes "1" to <PIN>, it is set to "1". However, writing "0" does not clear <PIN> to "0".

If <PIN> is set "1", SCLx pin is released. It takes t_{LOW} from setting <PIN> to "1" to releasing SCLx pin.

Note: When arbitration lost is occurred in the master mode, <PIN> is not cleared to "0" if the received slave address does not match. But INTSBIx is occurred.

12.6.10 Arbitration Lost Detection Monitor

The I2C bus has the multi-master capability (there are two or more masters on a bus), and requires the bus arbitration procedure to ensure correct data transfer.

A master that attempts to generate the start condition while the bus is busy loses bus arbitration, with no start condition occurring on the SDA and SCL bus lines.The I2C-bus arbitration takes place on the SDA bus line.

The arbitration procedure for two masters on a bus is shown below.

Up until the point a, Master A and Master B output the same data. At the point a, Master A outputs the "Low" level and Master B outputs the "High" level.

Then Master A pulls the SDA bus line to the "Low" level because the line has the wired-AND connection. When the SCL line goes high at the point b, the slave device reads the SDA line data, i.e., data transmitted by Master A. At this time, data transmitted by Master B becomes invalid.

This condition of Master B is called "Arbitration Lost". Master B releases its SDAx pin, so that it does not affect the data transfer initiated by another master. If two or more masters have transmitted exactly the same first data word, the arbitration procedure continues with the second data word.

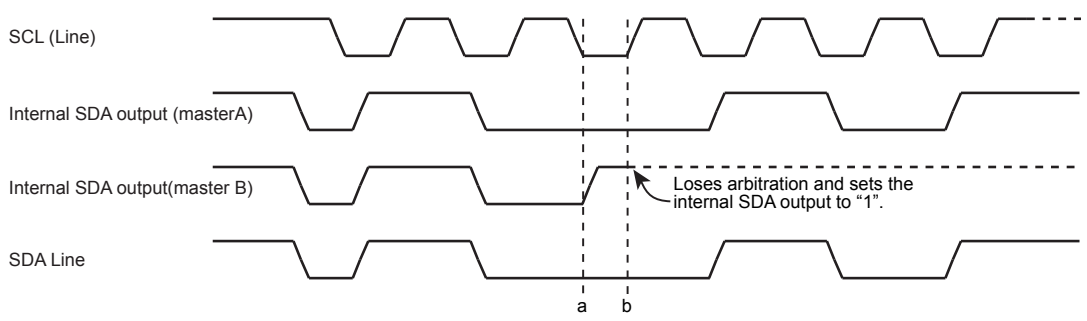


Figure 12-5 Lost Arbitration

A master compares the SDA bus line level and the internal SDA output level at the rising of the SCL line. If there is a difference between these two values, Arbitration Lost occurs and SBIxSR<AL> is set to "1".

When <AL> is set to "1", SBIxSR<MST, TRX> are cleared to "0", causing the SBI to operate as a slave receiver.Therefore, the serial bus interface circuit stops the clock output during data transfer after <AL> is set to "1".

<AL> is cleared to "0" when data is written to or read from SBIxDBR or data is written to SBIxCR2.

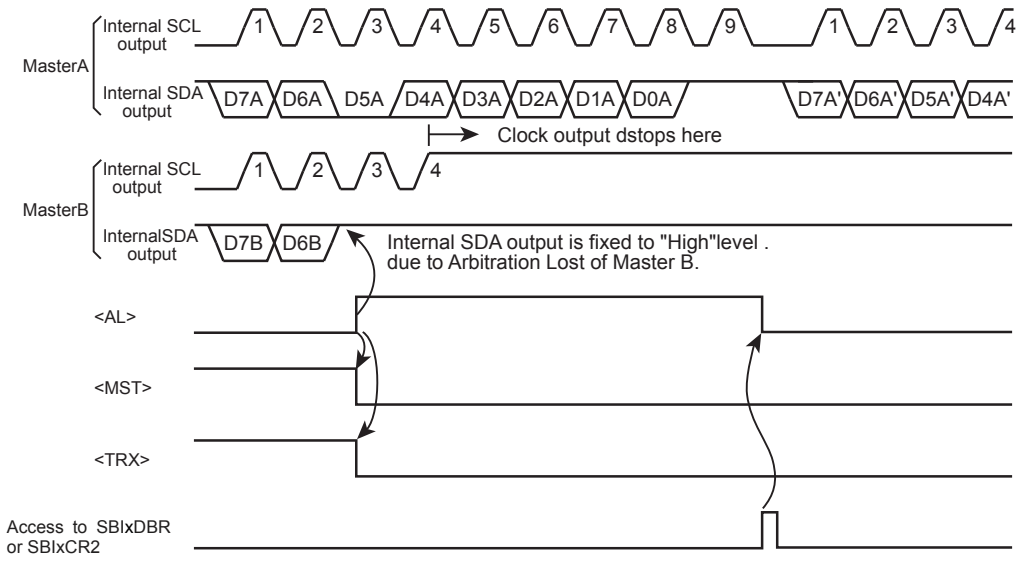


Figure 12-6 Example of Master B Lost Arbitration (D7A = D7B, D6A = D6B)

12.6.11 Slave Address Match Detection Monitor

When the SBI operates as a slave device in the addressing format (SBIxI2CAR<ALS>="0"), SBIxSR<AAS> is set to "1" on receiving the general-call or the slave address that matches the value specified at SBIxI2CAR.

When <ALS> is "1", <AAS> is set to "1" when the first data word has been received.

<AAS> is cleared to "0" when data is written to or read from SBIxDBR.

12.6.12 General-call Detection Monitor

When the SBI operates as a slave device, SBIxSR<ADO> is set to "1" when it receives the general-call address; i.e., the eight bits following the start condition are all zeros.

<ADO> is cleared to "0" when the start or stop condition is detected on the bus.

12.6.13 Last Received Bit Monitor

SBIxSR<LRB> is set to the SDA line value that was read at the rising of the SCL line.

In the acknowledgment mode, reading SBIxSR<LRB> immediately after generation of the INTSBIx interrupt request causes ACK signal to be read.

12.6.14 Data Buffer Register (SBIxDBR)

To read a data or to write a data, SBIxDBR is read or written from or to SBIxDBR.

When the SBI is in the master mode, after writing a slave address and a direction bit to this register in SBIxDBR, the start condition is generated, SBI transmits a slave address and a direction bit to slave device.

12.6.15 Baud Rate Register (SBIxBR0)

The SBIxBR0<I2SBI> register determines if the SBI operates or not when it enters the IDLE mode.

This register must be programmed before executing an instruction to switch to the standby mode.

12.6.16 Software Reset

If SBI locks up due to external noise, it can be initialized by using a software reset.

Writing "10" followed by "01" to SBIxCR2<SWRST[1:0]> generates a reset signal that initializes SBI. When writing SBIxCR2<SWRST[1:0]>, set "10" to SBIxCR2<SBIM[1:0]> for I2C bus mode. After a reset, all control registers and status flags are initialized to their reset values. When SBI is initialized, <SWRST> is automatically cleared to "00".

Note: A software reset causes the SBI operating mode to switch from the I2C mode to the port mode.

12.7 Data transfer Procedure in the I2C Bus mode.

12.7.1 Device Initialization

First, program SBIxCR1<ACK, SCK[2:0]>. Writing "000" to SBIxCR1<BC[2:0]> at the time.

Next, program SBIxI2CAR by specifying a slave address at <SA[6:0]> and an address recognition mode at <ALS>. (<ALS> must be cleared to "0" when using the addressing format).

To configure the Serial Bus Interface as a slave receiver, ensure that the serial bus interface pin is at "High" first. Then write "0" to SBIxCR2<MST, TRX, BB>, "1" to <PIN>, "10" to <SBIM[1:0]> and "0" to the bit 1 and 0.

Note: Initialization of the serial bus interface circuit must be completed within a period that any device does not generate start condition after all devices connected to the bus were initialized. If this rule is not followed, data may not be received correctly because other devices may start transfer before the initialization of the serial bus interface circuit is completed.

		7	6	5	4	3	2	1	0	
SBIxCR1	←	0	0	0	X	0	X	X	X	Specifies ACK and SCL clock.
SBIxI2CAR	←	X	X	X	X	X	X	X	X	Specifies a slave address and an address recognition mode.
SBIxCR2	←	0	0	0	1	1	0	0	0	Configures the SBI as an I2C bus mode and a slave receiver.

Note: X; Don't care

12.7.2 Generating the Start Condition and a Slave Address

The following steps are required to generate the start condition and slave address.

First, ensure that the bus is bus free (SBIxSR<BB> = "0"). then SBIxCR1<ACK> is set to "1" to select the acknowledgment mode. Write to SBIxDBR a slave address and a direction bit to be transmitted.

When <BB> = "0", writing "1111" to SBIxCR2<MST, TRX, BB, PIN> generates the start condition on the bus.

Following the start condition, the SBI generates nine clocks from the SCLx pin.

The SBI outputs the slave address and the direction bit specified at SBIxDBR with the first eight clocks, and releases the SDA line in the ninth clock to receive an acknowledgment signal from the slave device.

The INTSBIx interrupt request is generated on the falling of the ninth clock, and <PIN> is cleared to "0". The SBI holds the SCL line at the "Low" level while <PIN> is "0". <TRX> changes its value according to the transmitted direction bit at generation of the INTSBIx interrupt request, provided that an acknowledgment signal has been returned from the slave device.

Note: To output salve address, check with software that the bus is free before writing to SBIxDBR. If this rule is not followed, data being output on the bus may get ruined.

Settings in main routine

```
                    7  6  5  4  3  2  1  0
Reg.          ←     SBIxSR
Reg.          ←     Reg. e 0x20
if Reg.       ≠     0x00                          Ensures that the bus is free.

Then
SBIxCR1       ←     X  X  X  1  0  X  X  X        Selects the acknowledgement mode.
SBIxDBR       ←     X  X  X  X  X  X  X  X        Specifies the desired slave address and direction.
SBIxCR2       ←     1  1  1  1  1  0  0  0        Generates the start condition.
Example of INTSBI0 interrupt routine
    Clears the interrupt request.
    Processing
    End of interrupt
```

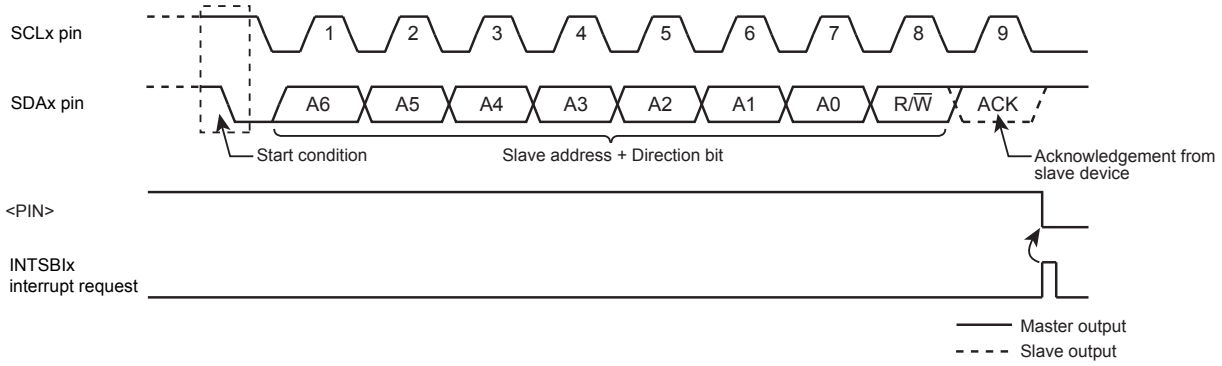


Figure 12-7 Generation of the Start Condition and a Slave Address

12.7.3 Transferring a Data Word

At the end of a data word transfer, the INTSBIx interrupt is generated to test <MST> to determine whether the SBI is in the master or slave mode.

12.7.3.1 Master mode (<MST> = "1")

Test <TRX> to determine whether the SBI is configured as a transmitter or a receiver.

(1) Transmitter mode (<TRX> = "1")

Test <LRB>. If <LRB> is "1", that means the receiver requires no further data.

The master then generates the stop condition as described later to stop transmission.

If <LRB> is "0", that means the receiver requires further data.

If the next data to be transmitted has eight bits, the data is written into SBIxDBR. If the data has different length, <BC[2:0]> are programmed and the transmit data is written into SBIxDBR.

Writing the data makes <PIN> to "1", causing the SCLx pin to generate a serial clock for transferring a next data word, and the SDAx pin to transfer the data word.

After the transfer is completed, the INTSBIx interrupt request is generated, <PIN> is cleared to "0", and the SCLx pin is pulled to the "Low" level.

To transmit more data words, test <LRB> again and repeat the above procedure.

INTSBIx interrupt

if MST = 0

Then go to the slave mode processing.

if TRX = 0

Then go to the receiver mode processing.

if LRB = 0

Then go to processing for generating the stop condition.

SBIxCR1	←	X	X	X	X	0	X	X	X	Specifies the number of bits to be transmitted and specify whether ACK is required.
SBIxDBR	←	X	X	X	X	X	X	X	X	Writes the transmit data.

End of interrupt processing.

Note: X; Don't care

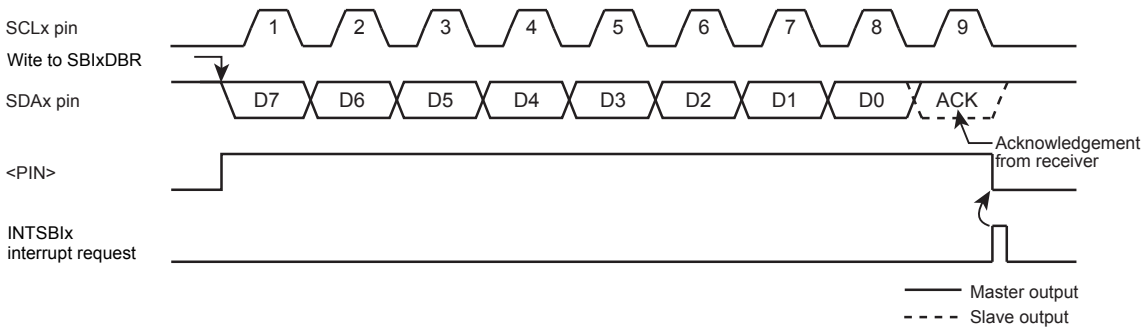


Figure 12-8 <BC[2:0]>= "000",<ACK>= "1" (Transmitter Mode)

(2) Receiver mode (<TRX> = "0")

If the next data to be transmitted has eight bits, the received data is read from into SBIxDBR.

If the data has different length, <BC[2:0]> are programmed and the received data is read from SBIxDBR to release the SCL line. (The data read immediately after transmission of a slave address is undefined.)

On reading the data, <PIN> is set to "1", and the serial clock is output to the SCLx pin to transfer the next data word. In the last bit, when the acknowledgment signal becomes the "Low" level, "0" is output to the SDAx pin.

After that, the INTSBIx interrupt request is generated, and <PIN> is cleared to "0", pulling the SCLx pin to the "Low" level. Each time the received data is read from SBIxDBR, one-word transfer clock and an acknowledgement signal are output.

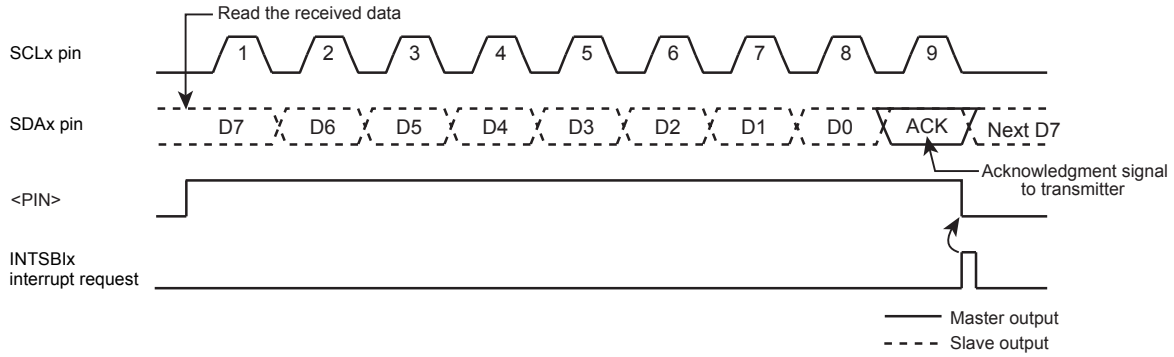


Figure 12-9 <BC[2:0]>= "000",<ACK>= "1" (Receiver Mode)

To terminate the data transmission from the transmitter, <ACK> is cleared to "0" immediately before reading the data word second to last.

This disables generation of an acknowledgment clock for the last data word.

When the transfer is completed, an interrupt request is generated. In tis interrupt processing, <BC [2:0]> must be set to "001" and the data must be read so that a clock is generated for 1-bit transfer.

At this time, so the master is a receiver, the master holds the SDA line at the "High" level. The transmitter receives this "High" level as ACK signal, the master receiver can inform to the transmitter the end of transfer.

In this interrupt processing for terminating the reception of 1-bit data, the stop condition is generated to terminate the data transfer.

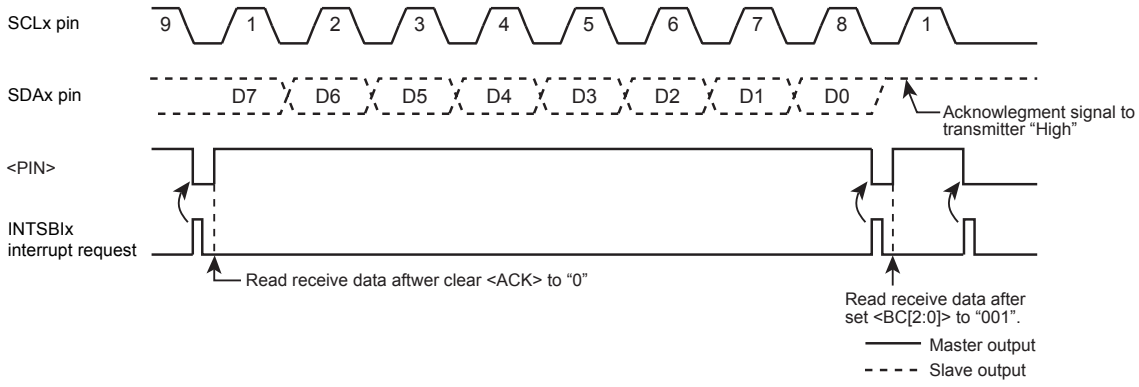


Figure 12-10 Terminating Data Transmission in the Master Receiver Mode

Example: When receiving N data word

INTSBIx interrupt (after data transmission)

		7	6	5	4	3	2	1	0	
SBIxCR1	←	X	X	X	X	0	X	X	X	Sets the number of bits of data to be received and specify whether ACK is required.
Reg.	←	SBIxDBR								Reads dummy data.
End of interrupt										

INTSBIx interrupt (first to (N-2)th data reception)

		7	6	5	4	3	2	1	0	
Reg.	←	SBIxDBR								Reads the first to (N-2)th data words.
End of interrupt										

INTSBIx interrupt ((N-1)th data reception)

		7	6	5	4	3	2	1	0	
SBIxCR1	←	X	X	X	0	0	X	X	X	Disables generation of acknowledgement clock.
Reg.	←	SBIxDBR								Reads the (N-1)th data word.
End of interrupt										

INTSBIx interrupt (Nth data reception)

		7	6	5	4	3	2	1	0	
SBIxCR1	←	0	0	1	0	0	X	X	X	Sets the number of bits to "1".
Reg.	←	SBIxDBR								Reads the Nth data word.
End of interrupt										

INTSBIx interrupt (after completing data reception)

Processing to generate the stop condition.	Terminates the data transmission.
End of interrupt	

Note: X; Don't care

12.7.3.2 Slave mode (<MST> = "0")

In the slave mode, SBI generates the INTSBIx interrupt request when SBI receives any slave address or general-call from master device, when SBI completes to transfers a data after SBI received its slave address or general-call.

Also, if the SBI detects Arbitration Lost in the master mode, it switches to the slave mode. When the completion of data word transfer in which Arbitration Lost is detected, the INTSBIx interrupt request is generated.

When INSBIx interrupt request, <PIN> is cleared to "0", and SCLx pin is pulled to the "Low" level.

When data is written to or read from SBIxDBR or when <PIN> is set to "1", SCLx pin is released after a period of t_{LOW}.

In the slave mode, the normal slave mode processing or the processing as a result of Arbitration Lost is carried out and it changes from master mode to slave mode.

SBIxSR<AL>, <TRX>, <AAS> and <ADO> are tested to determine the processing required.

"Table 12-1 Processing in Slave Mode"shows the slave mode states and required processing.

Example: When the received slave address matches the SBI's slave address and the direction bit is "1" in the slave receiver mode.

INTSBIx interrupt

if TRX = 0

Then go to other processing.

if AL = 0

Then go to other processing.

if AAS = 0

Then go to other processing.

SBIxCR1	←	X	X	X	1	0	X	X	X	Sets the number of bits to be transmitted.
SBIxDBR	←	X	X	X	X	X	X	X	X	Sets the transmit data.

Note: X; Don't care

Table 12-1 Processing in Slave Mode

<TRX>	<AL>	<AAS>	<ADO>	State	Processing
1	1	1	0	Arbitration Lost is detected while the slave address was being transmitted and the SBI received a slave address with the direction bit "1" transmitted by another master.	Set the number of bits in a data word to <BC[2:0]> and write the transmit data into SBIxDBR.
	0	1	0	In the slave receiver mode, the SBI received a slave address with the direction bit "1" transmitted by the master.	
		0	0	In the slave transmitter mode, the SBI has completed a transmission of one data word.	Test LRB. If it has been set to "1", that means the receiver does not require further data. Set <PIN> to 1 and reset <TRX> to 0 to release the bus. If <LRB> has been reset to "0", that means the receiver requires further data. Set the number of bits in the data word to <BC[2:0]> and write the transmit data to the SBIxDBR.
0	1	1	1/0	Arbitration Lost is detected while a slave address is being transmitted, and the SBI receives either a slave address with the direction bit "0" or a general-call address transmitted by another master.	Read the SBIxDBR (a dummy read) to set <PIN> to 1, or write "1" to <PIN>.
		0	0	Arbitration Lost is detected while a slave address or a data word is being transmitted, and the transfer is terminated.	
	0	1	1/0	In the slave receiver mode, the SBI received either a slave address with the direction bit "0" or a general-call address transmitted by the master.	
		0	1/0	In the slave receiver mode, the SBI has completed a reception of a data word.	Set the number of bits in the data word to <BC[2:0].> and read the received data from SBIxDBR.

12.7.4 Generating the Stop Condition

When SBIxSR<BB> is "1", writing "1" to SBIxCR2<MST, TRX, PIN> and "0" to <BB> causes 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, SBI waits until the SCL line is released.

After that, the SDAx pin goes "High", causing the stop condition to be generated.

		7	6	5	4	3	2	1	0	
SBIxCR2	←	1	1	0	1	1	0	0	0	Generates the stop condition.

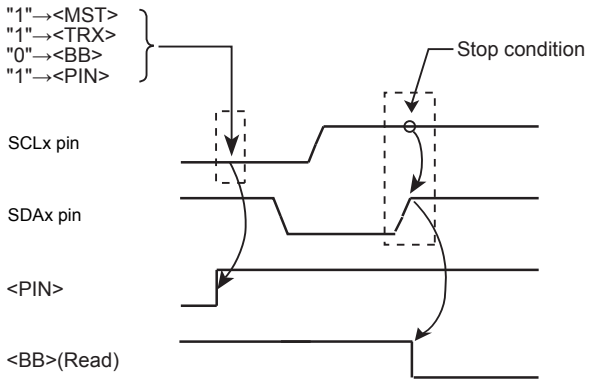


Figure 12-11 Generating the Stop Condition

12.7.5 Restart Procedure

Restart is used when a master device changes the data transfer direction without terminating the transfer to a slave device. The procedure of generating a restart in the master mode is described below.

First, write SBIxCR2<MST, TRX, BB> to "0" and write "1" to <PIN> to release the bus. At this time, the SDAx pin is held at the "High" level and the SCLx pin is released. Because no stop condition is generated on the bus, other devices recognize that the bus is busy.

Then, test SBIxSR<BB> and wait until it becomes "0" to ensure that the SCLx pin is released.

Next, test <LRB> and wait until it becomes "1" to ensure that no other device is pulling the SCL bus line to the "Low" level.

Once the bus is determined to be free by following the above procedures, follow the procedures described in "12.7.2 Generating the Start Condition and a Slave Address" to generate the start condition.

To satisfy the setup time of restart, at least 4.7μs wait period (in the standard mode) must be created by the software after the bus is determined to be bus free.

Note 1: Do not write <MST> to "0" when it is "0". (Restart cannot be initiated.)

Note 2: When the master device is acting as a receiver, data transmission from the slave device which serves as a transmitter must be completed before generating a restart. To complete data transfer, slave device must receive a "High" level acknowledge signal. For this reason, <LBR> before generating a restart becomes "1", the rising edge of the SCL line is not detected even <LBR>= "1" is confirmed by following the restart procedure. To check the status of the SCL line, read the port.

		7	6	5	4	3	2	1	0	
SBIxCR2	←	0	0	0	1	1	0	0	0	Releases the bus.

if SBIxSR<BB> ≠ 0	Checks that the SCLx pin is released.
Then	
if SBIxSR<LRB> ≠ 1	Checks that no other device is pulling the SCLx pin to the "Low".
Then	
4.7 μs Wait	

SBIxCR1	←	X	X	X	1	0	X	X	X	Selects the acknowledgment mode.
SBIxDBR	←	X	X	X	X	X	X	X	X	Sets the desired slave address and direction.
SBIxCR2	←	1	1	1	1	1	0	0	0	Generates the start condition.

Note: X; Don't care

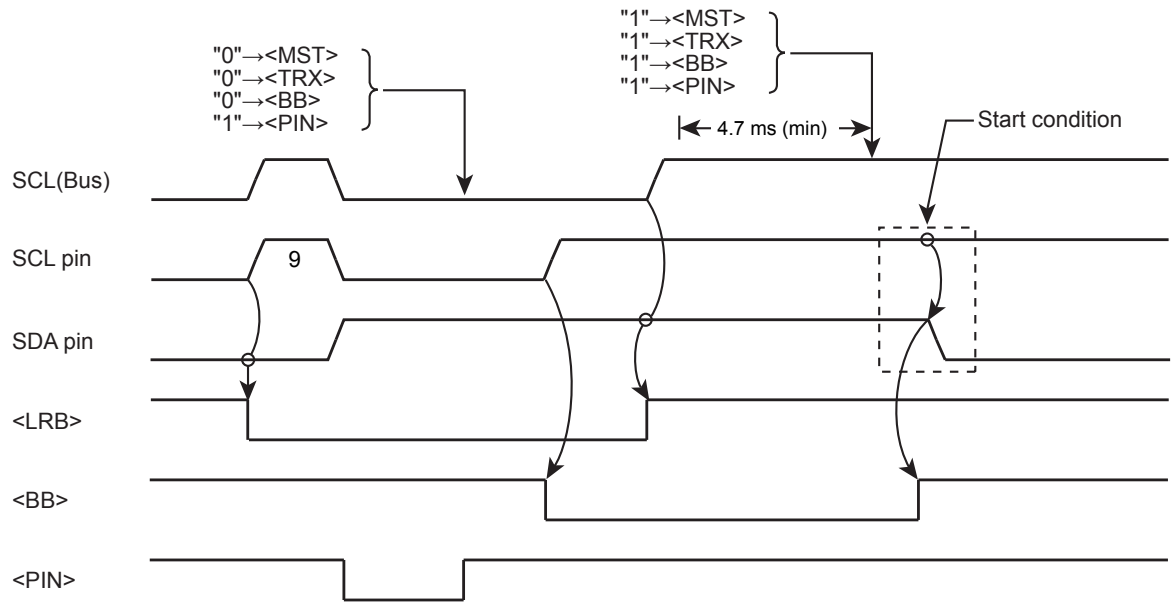


Figure 12-12 Timing Chart of Generating a Restart

Precautions on Use of Multi-master

Prepare recovery process by software in case that communication is in lock state in multi-master mode.

Example of recovery process

1. Start timer for timeout detection synchronizing with starting communication.
2. If a serial interface interrupt (INTSBIx) does not occur within the specified time, a timeout occurs and the MCU determines that communication is locked up.
3. Do software reset on serial bus interface to release the condition that communication is locked up.
4. Adjust transmission timings. (Note)
5. Resend transmission data.

Note: Adjust transmission timing between the MCUs to avoid overlapping the transmission timing.

12.8 Control register of SIO mode

The following registers control the serial bus interface in the clock-synchronous 8-bit SIO mode and provide its status information for monitoring.

12.8.1 SBIxCR0(control register 0)

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7	SBIEN	R/W	Serial bus interface operation. 0:Disable 1: Enable Enable this bit before using the serial bus interface. If this bit is disabled, power consumption can be reduced because all clocks except SBIxCR0 stop. If the serial bus interface operation is enabled and then disabled, the settings will be maintained in each register.
6-0	-	R	Read as "0".

12.8.2 SBIxCR1(Control register 1)

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7	SIOS	R/W	Transfer Start/Stop 0: Stop 1: Start
6	SIOINH	R/W	Transfer 0: Continue 1: Forced termination
5-4	SIOM[1:0]	R/W	Select transfer mode 00: Transmit mode 01: Reserved 10:Transmit/receive mode 11:Receive mode
3	-	R	Read as "1".
2-0	SCK[2:0]	R/W	On writing <SCK[2:0]>: Select serial clock frequency. (Note 1)

000	n = 3	System clock: fsys Clock gear: fc/1 Frequency = $\frac{fsys/2}{2^n}$ [Hz]
001	n = 4	
010	n = 5	
011	n = 6	
100	n = 7	
101	n = 8	
110	n = 9	
111	–	External clock

Note 1: After a reset, the <SCK[0]> bit is read as "1". However, if the SIO mode is selected at the SBIxCR2 register, the initial value is read as "0". In this document, the value written in the column "after reset" is the value after setting the SIO mode in the initial state. The descriptions of the SBIxCR2 register and the SBIxSR register are the same.

Note 2: Set <SIOS> to "0" and <SIOINH> to "1" before programming the transfer mode and the serial clock.

12.8.3 SBIxDBR (Data buffer register)

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

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

Note: Since SBIxDBR has independent buffers for writing and reading, a written data cannot be read. Thus, read-modify-write operation cannot be used.

12.8.4 SBIxCR2(Control register 2)

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7-4	-	R	Read as "1". (Note 1)
3-2	SBIM[1:0]	W	Select serial bus interface operating mode (Note 2) 00: Port mode 01: SIO mode 10: I2Cbus mode 11: Reserved
1-0	-	R	Read as 1. (Note 1)

Note 1: In this document, the value written in the column "after reset" is the value after setting the SIO mode in the initial state.

Note 2: Make sure that modes are not changed during a communication session.

Note 3: Changes from port mode to I2C bus mode after conforming that SDAx/SCLx pin and SOx/SIx/SCKx pin are "High" level.

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

12.8.5 SBIxSR (Status Register)

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7-4	-	R	Read as "1". (Note)
3	SIOF	R	Serial transfer status monitor. 0: Completed 1: In progress
2	SEF	R	Shift operation status monitor 0: Completed. 1: In progress
1-0	-	R	Read as "1". (Note)

Note: In this document, the value written in the column "after reset" is the value after setting the SIO mode in the initial state.

12.8.6 SBIxBR0 (Baud rate register 0)

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7	-	R	Read as "1".
6	I2SBI	R/W	Operation in IDLE mode. 0: Stop 1: Operate
5-1	-	R	Read as "1".
0	-	R/W	Write "0".

12.9 Control in SIO mode

12.9.1 Serial Clock

12.9.1.1 Clock source

Internal or external clocks can be selected by programming SBIxCR1<SCK[2:0]>.

(1) Internal clocks

In the internal clock mode, one of the seven frequencies can be selected as a serial clock, which is output to the outside through the SCKx pin.

At the beginning of a transfer, the SCKx pin output becomes the "High" level.

If the program cannot keep up with this serial clock rate in writing the transmit data or reading the received data, the SBI automatically enters a wait period. During this period, the serial clock is stopped automatically and the next shift operation is suspended until the processing is completed.

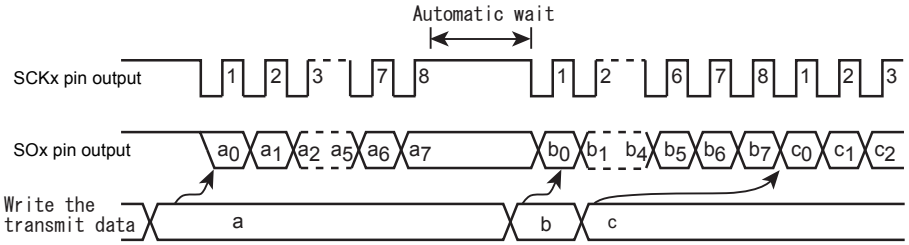


Figure 12-13 Automatic Wait

(2) External clock (<SCK[2:0]> = "111")

The SBI uses an external clock supplied from the outside to the SCKx pin as a serial clock.

For proper shift operations, the serial clock at the "High" and "Low" levels must have the pulse widths as shown below.

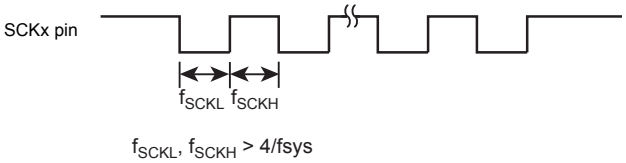


Figure 12-14 Maximum Transfer Frequency of External Clock Input

12.9.1.2 Shift Edge

Leading-edge shift is used in transmission. Trailing-edge shift is used in reception.

- Leading-edge shift

  Data is shifted at the leading edge of the serial clock (or the falling edge of the SCKx pin input/output).

- Trailing-edge shift

  Data is shifted at the trailing edge of the serial clock (or the rising edge of the SCKx pin input/output).

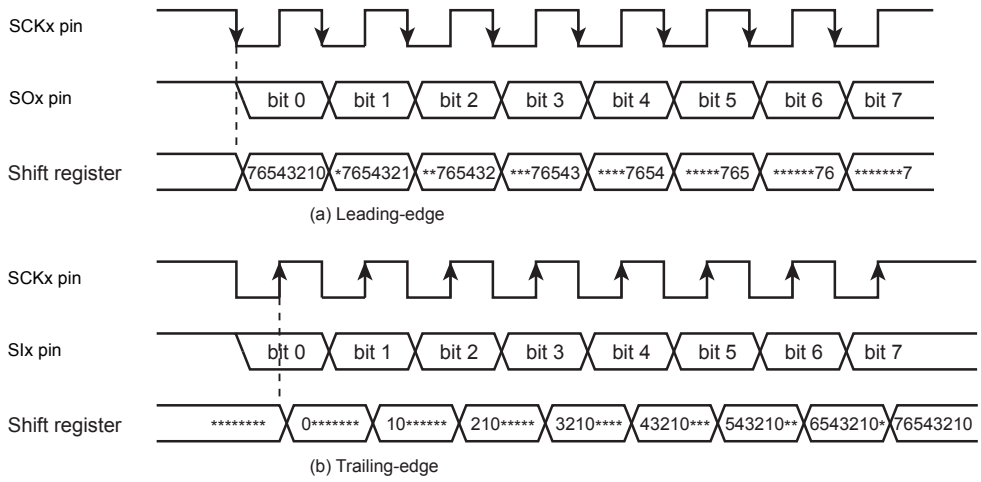


Figure 12-15 Shift Edge

12.9.2 Transfer Modes

The transmit mode, the receive mode or the transmit/receive mode can be selected by programming SBIxCR1<SIOM[1:0]>.

12.9.2.1 8-bit transmit mode

Set the control register to the transmit mode and write the transmit data to SBIxDBR.

After writing the transmit data, writing "1" to SBIxCR1<SIOS> starts the transmission. The transmit data is moved from SBIxDBR to a shift register and output to the SOx pin, with the least-significant bit (LSB) first, in synchronization with the serial clock. Once the transmit data is transferred to the shift register, SBIxDBR becomes empty, and the INTSBIx (buffer-empty) interrupt is generated, requesting the next transmit data.

In the internal clock mode, the serial clock will be stopped and automatically enter the wait state, if next data is not loaded after the 8-bit data has been fully transmitted. The wait state will be cleared when SBIxDBR is loaded with the next transmit data.

In the external clock mode, SBIxDBR must be loaded with data before the next data shift operation is started. Therefore, the data transfer rate varies depending on the maximum latency between when the interrupt request is generated and when SBIxDBR is loaded with data in the interrupt service program.

At the beginning of transmission, the same value as in the last bit of the previously transmitted data is output in a period from setting SBIxSR<SIOF> to "1" to the falling edge of SCK.

Transmission can be terminated by clearing <SIOS> to "0" or setting <SIOINH> to "1" in the INTSBIx interrupt service program. If <SIOS> is cleared, remaining data is output before transmission ends. The program checks SBIxSR<SIOF> to determine whether transmission has come to an end. <SIOF> is cleared to "0" at the end of transmission. If <SIOINH> is set to "1", the transmission is aborted immediately and <SIOF> is cleared to "0".

When in the external clock mode, <SIOS> must be cleared to "0" before next data shifting. If <SIOS> does not be cleared to "0" before next data shifting, SBI output dummy data and stopped.

		7	6	5	4	3	2	1	0	
SBIxCR1	←	0	1	0	0	0	X	X	X	Selects the transmit mode.
SBIxDBR	←	X	X	X	X	X	X	X	X	Writes the transmit data.
SBIxCR1	←	1	0	0	0	0	X	X	X	Starts transmission.

INTSBIx interrupt

SBIxDBR	←	X	X	X	X	X	X	X	X	Writes the transmit data.

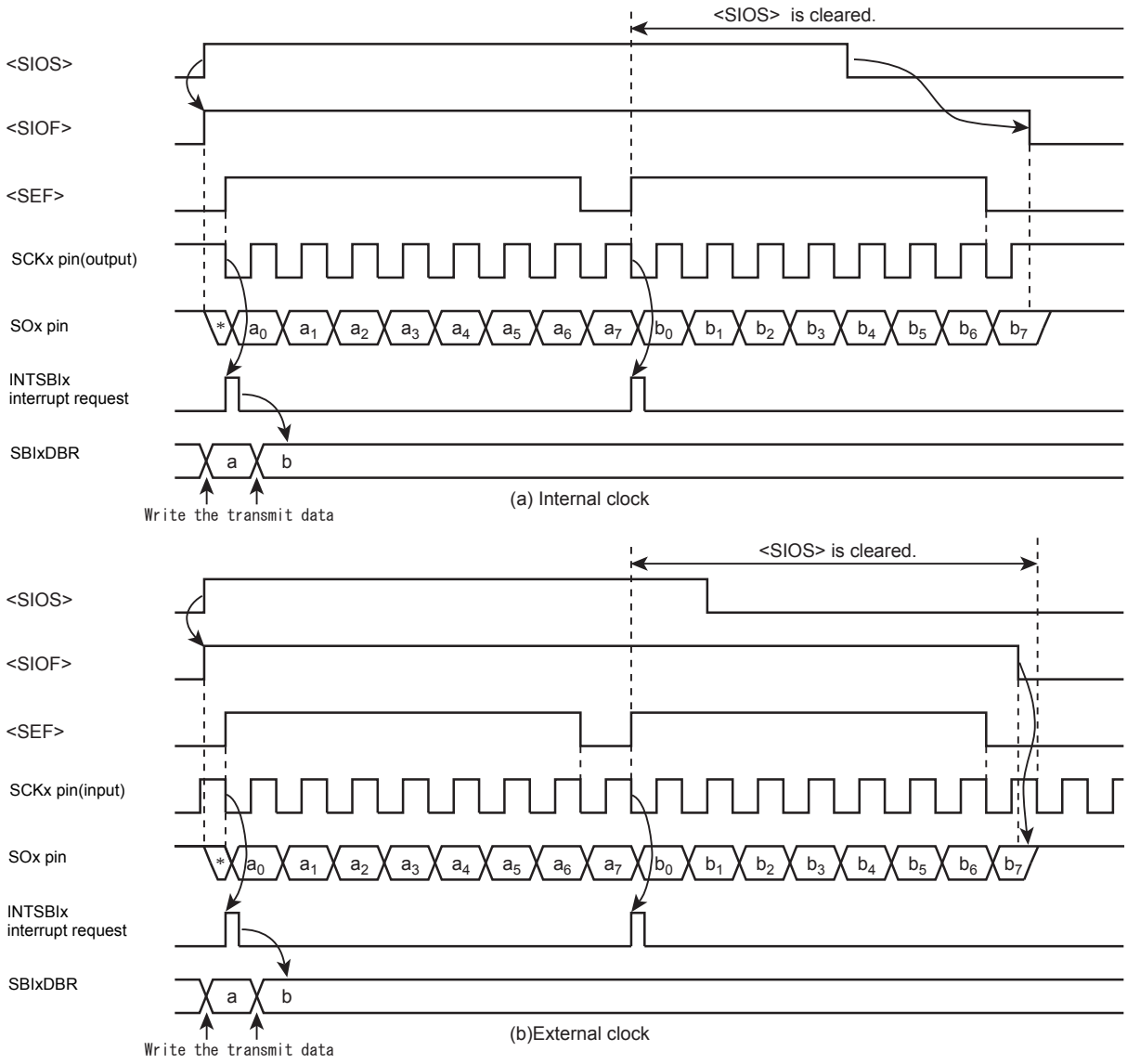


Figure 12-16 Transmit Mode

Example: Example of programming (external clock) to terminate transmission by <SIOS>

	7 6 5 4 3 2 1 0	
if SBIxSR<SIOF> ≠ 0		Recognizes the completion of the transmission.
Then		
if SCK ≠ 1		Recognizes "1" is set to the SCK pin by monitoring the port.
Then		
SBIxCR1	← 0 0 0 0 0 1 1 1	Completes the transmission by setting <SIOS> = 0.

12.9.2.2 8-bit receive mode

Set the control register to the receive mode. Then writing "1" to SBIxCR1<SIOS> enables reception.Data is taken into the shift register from the SIx pin, with the least-significant bit (LSB) first, in synchronization with the serial clock. Once the shift register is loaded with the 8-bit data, it transfers the received data to SBIxDBR and the INTSBIx (buffer-full) interrupt request is generated to request reading the received data. The interrupt service program then reads the received data from SBIxDBR.

In the internal clock mode, the serial clock will be stopped and automatically be in the wait state until the received data is read from SBIxDBR.

In the external clock mode, shift operations are executed in synchronization with the external clock. The maximum data transfer rate varies, depending on the maximum latency between generating the interrupt request and reading the received data

Reception can be terminated by clearing <SIOS> to "0" or setting <SIOINH> to "1" in the INTSBIx interrupt service program. If <SIOS> is cleared, reception continues until all the bits of received data are written to SBIxDBR. The program checks SBIxSR<SIOF> to determine whether reception has come to an end.<SIOF> is cleared to "0" at the end of reception. After confirming the completion of the reception, last received data is read. If <SIOINH> is set to "1", the reception is aborted immediately and <SIOF> is cleared to "0". (The received data becomes invalid, and there is no need to read it out.)

Note: The contents of SBIxDBR will not be retained after the transfer mode is changed. The ongoing reception must be completed by clearing <SIOS> to "0" and the last received data must be read before the transfer mode is changed.

		7	6	5	4	3	2	1	0	
SBIxCR1	←	0	1	1	1	0	X	X	X	Selects the receive mode.
SBIxCR1	←	1	0	1	1	0	X	X	X	Starts reception.

INTSBIx interrupt

Reg.	←	SBIxDBR	Reads the received data.

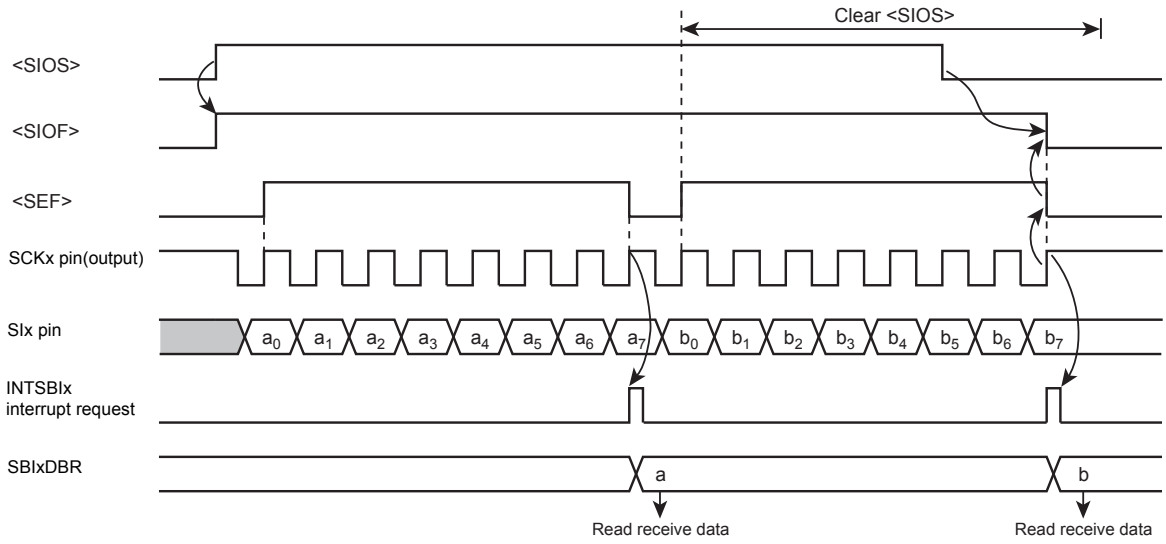


Figure 12-17 Receive Mode (Example: Internal Clock)

12.9.2.3 8-bit transmit/receive mode

Set the control register to the transfer/receive mode. Then writing the transmit data to SBIxDBR and setting SBIxCR1<SIOS> to "1" enables transmission and reception.The transmit data is output through the SOx pin at the falling of the serial clock, and the received data is taken in through the SI pin at the rising of the serial clock, with the least-significant bit (LSB) first. Once the shift register is loaded with the 8-bit data, it transfers the received data to SBIxDBR and the INTSBIx interrupt request is generated.The interrupt service program reads the received data from the data buffer register and writes the next transmit data. Because SBIxDBR is shared between transmit and receive operations, the received data must be read before the next transmit data is written.

In the internal clock operation, the serial clock will be automatically in the wait state until the received data is read and the next transmit data is written.

In the external clock mode, shift operations are executed in synchronization with the external serial clock. Therefore, the received data must be read and the next transmit data must be written before the next shift operation is started.The maximum data transfer rate for the external clock operation varies depending on the maximum latency between when the interrupt request is generated and when the transmit data is written.

At the beginning of transmission, the same value as in the last bit of the previously transmitted data is output in a period from setting <SIOF> to "1" to the falling edge of SCKx.

Transmission and reception can be terminated by clearing <SIOS> to "0" or setting SBIxCR1<SIOINH> to "1" in the INTSBIx interrupt service program. If <SIOS> is cleared, transmission and reception continue until the received data is fully transferred to SBIxDBR. The program checks SBIxSR<SIOF> to determine whether transmission and reception have come to an end. <SIOF> is cleared to "0" at the end of transmission and reception.If <SIOINH> is set to "1", the transmission and reception is aborted immediately and <SIOF> is cleared to "0".

Note: The contents of SBIxDBR will not be retained after the transfer mode is changed. The ongoing transmission and reception must be completed by clearing <SIOS> to "0" and the last received data must be read before the transfer mode is changed.

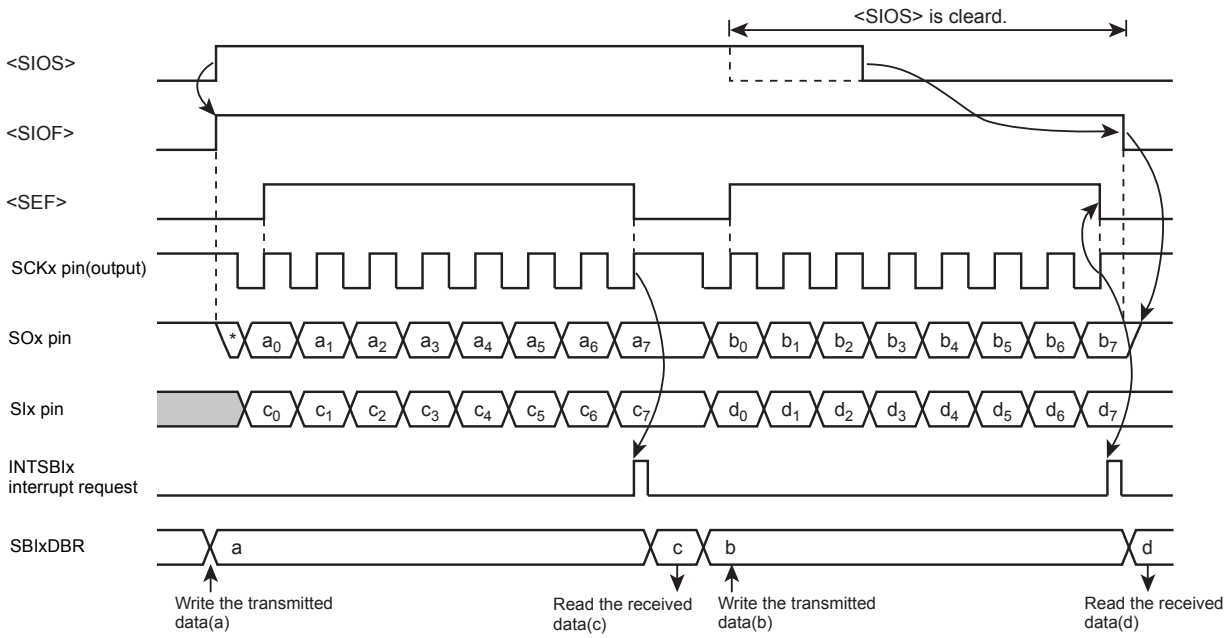


Figure 12-18 Transmit/Receive Mode (Example: Internal Clock)

		7	6	5	4	3	2	1	0	
SBIxCR1	←	0	1	1	0	0	X	X	X	Selects the transmit mode.
SBIxDBR	←	X	X	X	X	X	X	X	X	Writes the transmit data.
SBIxCR1	←	1	0	1	0	0	X	X	X	Starts reception/transmission.

INTSBIx interrupt

Reg.	←	SBIxDBR								Reads the received data.
SBIxDBR	←	X	X	X	X	X	X	X	X	Writes the transmit data.

12.9.2.4 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.

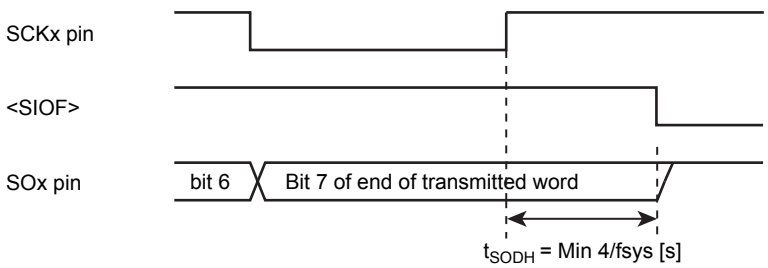


Figure 12-19 Data retention time of the last bit at the end of transmission

13. 10-bit Analog/Digital Converter (ADC)

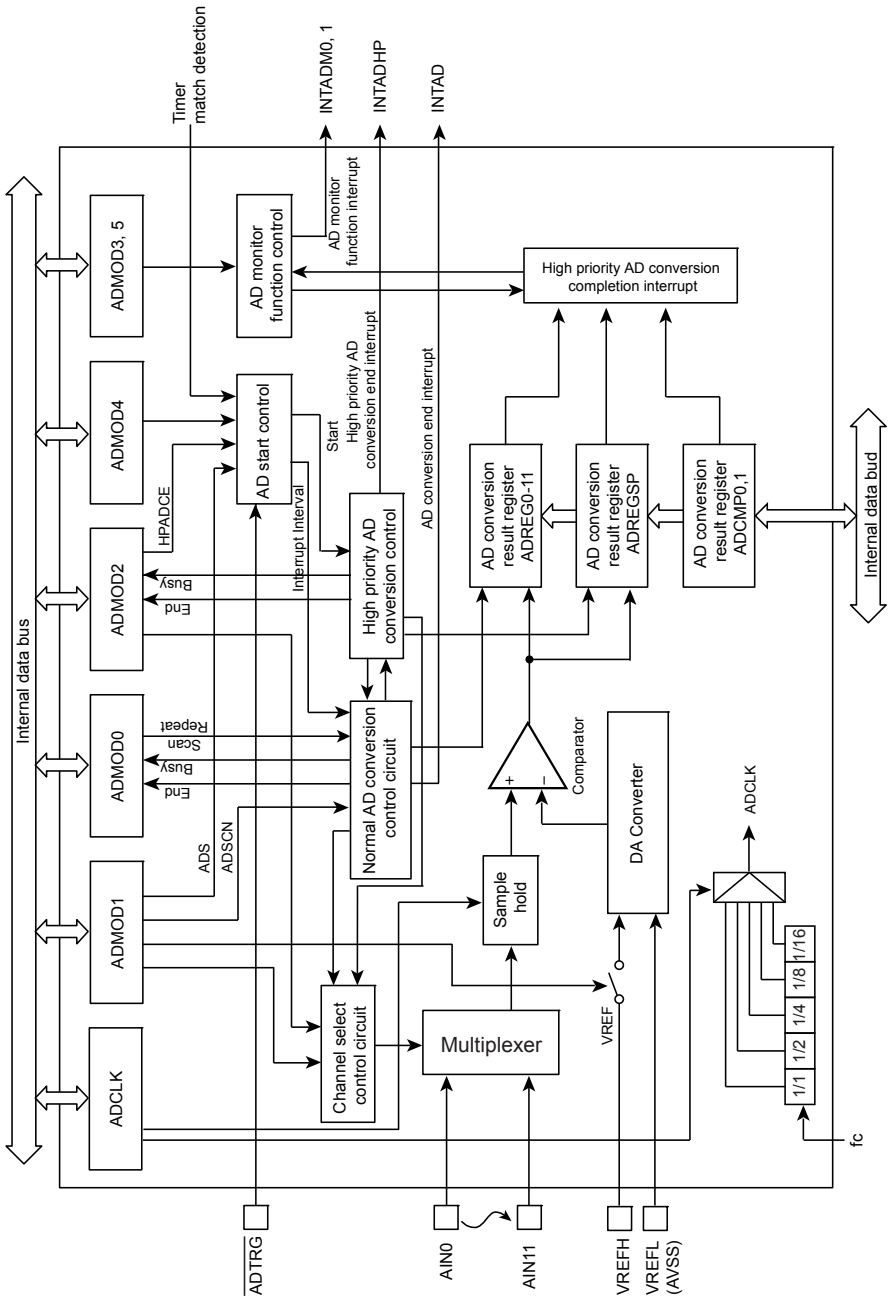
13.1 Outline

A 10-bit, sequential-conversion analog/digital converter (AD converter) is built into the TMPM061FWFG.

For details, refer to "Product information" to confirm usable channels and settings.

13.2 Configuration

Figure 13-1 shows the block diagram of this AD converter.



Note: VREFL and AVSS are shared.

Figure 13-1 10-bit AD Converter Block Diagram

13.3 Registers

13.3.1 Register list

The control registers and addresses of the AD converter are as follows.

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

Register name		Address (Base+)
Conversion Clock Setting Register	ADCLK	0x0000
Mode Control Register 0	ADMOD0	0x0004
Mode Control Register 1	ADMOD1	0x0008
Mode Control Register 2	ADMOD2	0x000C
Mode Control Register 3	ADMOD3	0x0010
Mode Control Register 4	ADMOD4	0x0014
Mode Control Register 5	ADMOD5	0x0018
Conversion Result Register 0	ADREG0	0x0030
Conversion Result Register 1	ADREG1	0x0034
Conversion Result Register 2	ADREG2	0x0038
Conversion Result Register 3	ADREG3	0x003C
Conversion Result Register 4	ADREG4	0x0040
Conversion Result Register 5	ADREG5	0x0044
Conversion Result Register 6	ADREG6	0x0048
Conversion Result Register 7	ADREG7	0x004C
Conversion Result Register 8	ADREG8	0x0050
Conversion Result Register 9	ADREG9	0x0054
Conversion Result Register 10	ADREG10	0x0058
Conversion Result Register 11	ADREG11	0x005C
Conversion Result Register SP	ADREGSP	0x0060
Conversion Result Comparison Register 0	ADCMP0	0x0064
Conversion Result Comparison Register 1	ADCMP1	0x0068

13.3.2 ADCLK (Conversion Clock Setting Register)

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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-4	ADCC[1:0]	R/W	Select the AD conversion clock count 00: 35.5 conversion clock 01: 42 conversion clock 10: 68 conversion clock 11: 81 conversion clock
5-3	−	R	Read as 0.
2-0	ADCLK[2:0]	R/W	Select the AD conversion clock (Note1) (Note2) 000: fc 001: fc/2 010: fc/4 011: fc/8 100: fc/16 101-111: Reserved

Note 1: Do not change the setting of the AD conversion clock during AD conversion.
Note 2: The AD conversion clock ADCLK must be selected so that the relationship "ADCLK ≤ fsys" is satisfied .

A clock count is required to satisfy the condition that described below.

VREFH AVDD	Conversion time
2.7 to 3.6V	16.2 μs or longer
1.8 to 3.6V	32.4μs or longer

13.3.3 ADMOD0 (Mode Control Register 0)

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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7	EOCFN	R	Normal AD conversion completion flag 0: Before or during conversion 1: Completion This bit is "0" cleared when it is read.
6	ADBFN	R	Normal AD conversion BUSY flag 0: Conversion stop 1: During conversion
5	−	R	Read as 0.
4-3	ITM[1:0]	R/W	Interrupt in fixed channel repeat conversion mode 00: Generate in interrupt once every single conversion 01: Generate interrupt once every 4 conversions 10: Generate interrupt once every 8 conversions 11: Setting prohibited It is valid only when it's specified in the fixed channel repeat mode (<REPEAT> ="1", <SCAN> = "0").
2	REPEAT	R/W	Specify repeat mode 0: Single conversion mode 1: Repeat conversion mode
1	SCAN	R/W	Specify scan mode 0: Fixed channel mode 1: Channel scan mode If channel scan mode is selected, set the channel number to ADMOD1<ADSCAN>.
0	ADS	W	Start AD conversion start 0: Don't care 1: Start conversion Conversion must be started after setting the mode. "0" is always read.

13.3.4 ADMOD1 (Mode Control Register 1)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit symbol	VREFON	I2AD	ADSCN		ADCH			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7	VREFON	R/W	VREF application control(Note) 0: OFF 1: ON
6	I2AD	R/W	Specify operation mode in IDLE mode 0: Stop 1: Operation
5-4	ADSCN[1:0]	R/W	Specify operation mode in channel scan mode 00: 4-channel scan 01: 8-channel scan 10: 12-channel scan 11: Reserved Specify operation mode when channel scan mode is selected by ADMOD0<SCAN>. The conversion channel is selected by setting of <ADCH>. Refer to the below table.
3-0	ADCH[3:0]	R/W	Select analog input channel (Refer to the below table.)

Note: Before starting AD conversion, write "1" to the <VREFON> bit, wait for 3μs during which time the internal reference voltage should stabilize, and then write "1" to the ADMOD0<ADS>.

Selection of analog input channel

		<ADCH[3:0]>							
		0000	0001	0010	0011	0100	0101	0110	0111
ADMOD0 <SCAN>=0	Fixed channel	AIN0	AIN1	AIN2	AIN3	AIN4	AIN5	AIN6	AIN7
ADMOD0 <SCAN>=1	<ADSCN>=00 4-channel scan	AIN0	AIN0~ AIN1	AIN0~ AIN2	AIN0~ AIN3	AIN4	AIN4~ AIN5	AIN4~ AIN6	AIN4~ AIN7
	<ADSCN>=01 8-channel scan	AIN0	AIN0~ AIN1	AIN0~ AIN2	AIN0~ AIN3	AIN0~ AIN4	AIN0~ AIN5	AIN0~ AIN6	AIN0~ AIN7
	<ADSCN>=10 12-channel scan	AIN0	AIN0~ AIN1	AIN0~ AIN2	AIN0~ AIN3	AIN0~ AIN4	AIN0~ AIN5	AIN0~ AIN6	AIN0~ AIN7

		<ADCH[3:0]>							
		1000	1001	1010	1011	1100	1101	1110	1111
ADMOD0 <SCAN>=0	Fixed channel	AIN8	AIN9	AIN10	AIN11	AIN12	AIN13	AIN14	AIN15
ADMOD0 <SCAN>=1	<ADSCN>=00 4-channel scan	AIN0	AIN0~ AIN1	AIN0~ AIN2	AIN0~ AIN3	AIN4	AIN4~ AIN5	AIN4~ AIN6	AIN4~ AIN7
	<ADSCN>=01 8-channel scan	AIN0	AIN0~ AIN1	AIN0~ AIN2	AIN0~ AIN3	AIN0~ AIN4	AIN0~ AIN5	AIN0~ AIN6	AIN0~ AIN7
	<ADSCN>=10 12-channel scan	AIN0~ AIN8	AIN0~ AIN9	AIN0~ AIN10	AIN0~ AIN11	-	-	-	-

13.3.5 ADMOD2 (Mode Control Register 2)

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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7	EOCFHP	R	Top-priority AD conversion completion flag (Note) 0: Before or during conversion 1: Completion
6	ADBFHP	R	Top-priority AD conversion BUSY flag 0: During conversion halts 1: During conversion
5	HPADCE	R/W	Activate top-priority conversion 0: Don't care 1: Start conversion "0" is always read.
4	−	R/W	Write "0".
3-0	HPADCH[3:0]	R/W	Select analog input channel when activating top-priority conversion. (See the table below)

Note: This bit is "0" cleared when it is read.

Selection of analog input channel

HPADCH[3:0]	0000	0001	0010	0011	0100	0101	0110	0111
Conversion channel	AIN0	AIN1	AIN2	AIN3	AIN4	AIN5	AIN6	AIN7

HPADCH[3:0]	1000	1001	1010	1011	1100	1101	1110	1111
Conversion channel	AIN8	AIN9	AIN10	AIN11	AIN12	AIN13	AIN14	AIN15

13.3.6 ADMOD3 (Mode Control Register 3)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit symbol	-	-	ADOBIC0	ADREGS0				ADOBSV0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7	−	R/W	Write "0".
6	−	R	Read as 0.
5	ADOBIC0	R/W	Set the AD monitor function interrupt 0 0: If the value of the conversion result is smaller than the comparison register 0, an interrupt is generated. 1: If the value of the conversion result is bigger than the comparison register 0, an interrupt is generated.
4-1	ADREGS0[3:0]	R/W	Select a target conversion result register when using the AD monitor function 0 (See the below table).
0	ADOBSV0	R/W	AD monitor function 0 0: Disable 1: Enable

<ADREGS0[3:0]>	Conversion result register to be compared	<ADREGS0[3:0]>	Conversion result register to be compared
0000	ADREG0	0100	ADREG4
0001	ADREG1	0101	ADREG5
0010	ADREG2	0110	ADREG6
0011	ADREG3	0111	ADREG7
-	-	1xxx	ADREGSP

13.3.7 ADMOD4 (Mode Control Register 4)

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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7	HADHS	R/W	H/W source for activating top-priority AD conversion 0: External trigger 1: Match with timer register (Note 1)
6	HADHTG	R/W	H/W for activating top-priority AD conversion 0: Disable 1: Enable
5	ADHS	R/W	H/W source for activating normal AD conversion (note2) 0: External trigger 1: Match with timer register (Note 1)
4	ADHTG	R/W	HW for activating normal AD conversion 0: Disable 1: Enable
3-2	−	R	Read as 0.
1-0	ADRST[1:0]	W	Overwriting 10 with 01 allows ADC to be software reset.(note 3)

Note 1: For details, refer to "Product information" to confirm H/W source.

Note 2: The external trigger cannot be used for H/W activation of AD conversion when it is used for H/W activation of top priority AD conversion.

Note 3: A software reset initializes all the registers except for ADCLK<ADCLK>.

13.3.8 ADMOD5 (Mode Control Register 5)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit symbol	-	-	ADOBIC1	ADREGS1				ADOBSV1
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-6	−	R	Read as 0.
5	ADOBIC1	R/W	Set the AD monitor function interrupt 1. 0: If the value of the conversion result is smaller than the comparison register 1, an interrupt is generated. 1: If the value of the conversion result is bigger than the comparison register 1, an interrupt is generated.
4-1	ADREGS1[3:0]	R/W	Select a target conversion result register when using the AD monitor function 1 (See the below table).
0	ADOBSV1	R/W	AD monitor function 1 0: Disable 1: Enable

<ADREGS1[3:0]>	Conversion result register to be compared	<ADREGS1[3:0]>	Conversion result register to be compared
0000	ADREG0	0100	ADREG4
0001	ADREG1	0101	ADREG5
0010	ADREG2	0110	ADREG6
0011	ADREG3	0111	ADREG7
−	−	1xxx	ADREGSP

13.3.9 ADREGn (Conversion Result Register n: n = 0 to 11)

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

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as 0.
15-6	ADRn[9:0]	R	AD conversion result Conversion result is stored. For information about the correlation between the conversion channel and the conversion result register, refer to the Table 13-2 and Table 13-3, "13.4.5.7 Interrupt generation timings and AD conversion result storage register".
5-2	−	R	Read as 0.
1	OVRn	R	Overrun flag 0: Not generated 1: Generated If the conversion result is overwritten before reading <ADR0>, "1" is set. This bit is "0" cleared when it is read.
0	ADRnRF	R	AD conversion result storage flag 0: Conversion result is not stored 1: Conversion result is stored. If a conversion result is stored, "1" is set. This bit is "0" cleared when the conversion result is read.

Note: Access to this register must be a half word or a word access.

13.3.10 ADREGSP (AD Conversion Result Register SP)

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

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as 0.
15-6	ADRSP[9:0]	R	AD conversion result. Top-priority AD conversion result is stored
5-2	−	R	Read as 0
1	OVRSP	R	Overrun flag 0: Not generated 1: Generated If a conversion result is overwritten before reading <ADRSP>, "1" is set. This bit is "0" cleared when it is read.
0	ADRSPRF	R	AD conversion result storage flag 0: Conversion result is not stored. 1: Conversion result is stored. If a conversion result is stored, "1" is set. This bit is "0" cleared when the conversion result is read.

Note: Access to this register must be a half word or a word access.

13.3.11 ADCMP0 (AD Conversion Result Comparison Register 0)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit symbol	ADCOM0							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit symbol	ADCOM0		-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as 0.
15-6	ADCOM0[9:0]	R/W	When AD monitor function 0 is enabled, it sets a value to be compared with the value of the conversion result register specified by ADMOD3<ADREGS0>.
5-0	−	R	Read as 0.

Note: To write values into this register, the AD monitor function 0 must be disabled (ADMOD3<ADBSV0>="0").

13.3.12 ADCMP1 (AD Conversion Result Comparison Register 1)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit symbol	ADCOM1							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit symbol	ADCOM1		-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as 0.
15-6	ADCOM1[9:0]	R/W	When AD monitor function 1 is enabled, it sets a value to be compared with the value of the conversion result register specified by ADMODt<ADREGS1>.
5-0	−	R	Read as 0.

Note: **To write values into this register, the AD monitor function 1 must be disabled (ADMOD5<ADBSV1>="0").**

13.4 Description of Operations

13.4.1 Analog Reference Voltage

The "High" level of the analog reference voltage shall be applied to the VRFEH pin, and the "Low" shall be applied to the VREFL pin.

To start AD conversion, make sure that you first write "1" to the <VREFON> bit, wait for 3 μs during which time the internal reference voltage should stabilize, and then write "1" to the ADMOD0<ADS> bit.

If you do not use ADC function, write "0" to the ADMOD1<DACON>. The consumption current of the analog circuit is reduced.

Note: VREFL and AVSS are shared by TMPM061FWFG.

13.4.2 AD Conversion Mode

Two types of AD conversion are supported: normal AD conversion and top-priority AD conversion.

For normal AD conversion, the following four operation modes are supported.

13.4.2.1 Normal AD conversion

For normal AD conversion, the following four operation modes are supported and the operation mode is selected with the ADMOD0 <REPEAT> <SCAN>.

- Fixed channel single conversion mode
- Channel scan single conversion mode
- Fixed channel repeat conversion mode
- Channel scan repeat conversion mode

For channel scan mode, the following three modes are supported and the operation mode is selected with the ADMOD1<ADSCN>.

- 4-channel scan mode
- 8-channel scan mode
- 12-channel scan mode

(1) Fixed channel single conversion mode

If ADMOD0<REPEAT, SCAN> is set to "00", "AD conversion is performed in the fixed channel single conversion mode.

In this mode, AD conversion is performed once for one channel selected. After AD conversion is completed, ADMOD0<EOCFN> is set to "1", ADMOD0<ADBFN> is cleared to "0", and the AD conversion completion interrupt request (INTAD) is generated. <EOCFN> is cleared to "0" upon read.

(2) Channel scan single conversion mode

If ADMOD0 <REPEAT, SCAN> is set to "01," AD conversion is performed in the channel scan single conversion mode.

In this mode, AD conversion is performed once for each scan channel selected. After AD scan conversion is completed, ADMOD0<EOCFN> is set to "1", ADMOD0<ADBFN> is cleared to "0", and the conversion completion interrupt request (INTAD) is generated. <EOCFN> is cleared to "0" upon read.

(3) Fixed channel repeat conversion mode

If ADMOD0<REPEAT, SCAN> is set to "10", AD conversion is performed in fixed channel repeat conversation mode.

In this mode, AD conversion is performed repeatedly for one channel selected. After AD conversion is completed, ADMOD0<EOCFN> is set to "1". ADMOD0<ADBFN> is not cleared to "0". It remains at "1". The timing with which the conversion completion interrupt request (INTAD) is generated can be selected by setting ADMOD0<ITM> to an appropriate setting. <EOCFN> is set with the same timing as this interrupt INTAD is generated.

<EOCFN> is cleared to "0" upon read.

(4) Channel scan repeat conversion mode

If ADMOD0<REPEAT, SCAN> is set to "11", AD conversion is performed in the channel scan repeat conversion mode.

In this mode, AD conversion is performed repeatedly for a scan channel selected. Each time one AD scan conversion is completed, ADMOD0<EOCFN> is set to "1", and the conversion completion interrupt request (INTAD) is generated. ADMOD0<ADBFN> is not cleared to "0". It remains at "1". <EOCFN> is cleared to "0" upon read.

13.4.2.2 Top-priority AD conversion

By interrupting ongoing normal AD conversion, top-priority AD conversion can be performed.

The fixed-channel single conversion is automatically selected, irrespective of the ADMOD0 <REPEAT, SCAN> setting. When conditions to start operation are met, a conversion is performed just once for a channel designated by ADMOD2<HPADCH>. When conversion is completed, the top-priority AD conversion completion interrupt (INTADHP) is generated, and ADMOD2<EOCFHP> showing the completion of AD conversion is set to "1". <ADBFHP> returns to "0". EOCFHP flag is cleared to "0" upon read.

Top-priority AD conversion activated while top-priority AD conversion is under way is ignored.

13.4.3 AD Monitor Function

There are two channels of AD monitor function.

If ADMOD3<ADOBSV0> and ADMOD5<ADOBSV1> are set to "1", the AD monitor function is enabled. If the value of the conversion result register specified by ADMOD3 <ADREGS0> and ADMOD5 <ADREGS1> becomes larger or smaller ("Larger" or "Smaller" to be designated by ADMOD3 <ADOBIC0> and ADMOD5 <ADBIC1>) than the value of a comparison register, the AD monitor function interrupt (INTADM0, INTADM1) is generated. This comparison operation is performed each time a result is stored in a corresponding conversion result register.

If the conversion result register assigned to perform the AD monitor function is continuously used without reading the conversion result, the conversion result is overwritten. The conversion result storage flag <ADRxRF> and the overrun flag <OVRx> remain being set.

13.4.4 Selecting the Input Channel

How the input channel is selected is different depending on AD converter operation mode to be used.

1. Normal AD conversion mode
   - If the analog input channel is used in a fixed state (ADMOD0<SCAN> = "0")

     One channel is selected from analog input pins by setting ADMOD1<ADCH> to an appropriate setting.
   - If the analog input channel is used in a scan state (ADMOD0<SCAN> = "1")

     One scan mode is selected from the scan modes by setting ADMOD1 <ADCH> and <ADSCN> to an appropriate setting.

2. Top-priority AD conversion mode

   One channel is selected from analog input pins by setting ADMOD2<HPADCH> to an appropriate setting.

13.4.5 AD Conversion Details

13.4.5.1 Starting AD Conversion

Normal AD conversion is activated by setting ADMOD0<ADS> to "1". Top-priority AD conversion is activated by setting ADMOD2<HPADCE> to "1".

Four operation modes are made available to normal AD conversion. In performing normal AD conversion, one of these operation modes must be selected by setting ADMOD0<REPEAT,SCAN> to an appropriate setting. For top-priority AD conversion, only one operation mode can be used: fixed channel single conversion mode.

Normal AD conversion can be activated using the H/W activation source selected by ADMOD4 <ADHS>, and top-priority AD conversion can be activated using the HW activation source selected by ADMOD4 <HADHS>. If bits of <ADHS> and <HADHS> are "0", normal and top-priority AD conversions are activated in response to the input of a falling edge through the $\overline{\text{ADTRG}}$ pin. If these bits are "1", conversion is activated in response to match detection of timer.

To permit H/W activation, set ADMOD4 <ADHTG> to "1" for normal AD conversion and set ADMOD4 <HADHTG> to "1" for top-priority AD conversion.

Software activation is still valid even after H/W activation has been permitted.

Note 1: Some products don't provide the $\overline{\text{ADTRG}}$ pin.

Note 2: When an external trigger is used for the HW activation source of a top-priority AD conversion, an external trigger cannot be set for activating normal AD conversion H/W.

Note 3: For details, refer to "Product information" to confirm usable match detection of timer.

13.4.5.2 AD Conversion

When normal AD conversion starts, the AD conversion Busy flag (ADMOD0<ADBFN>) showing that AD conversion is under way is set to "1".

When top-priority AD conversion starts, the top-priority AD conversion Busy flag (ADMOD2 <ADBFHP>) showing that AD conversion is underway is set to "1".

At that time, the value of the Busy flag ADMOD0<ADBFN> for normal AD conversion before the start of top-priority AD conversions are retained. The value of the conversion completion flag ADMOD0<EOCFN> for normal AD conversion before the start of top-priority AD conversion can also be retained.

Note: Normal AD conversion must not be activated when top-priority AD conversion is under way.

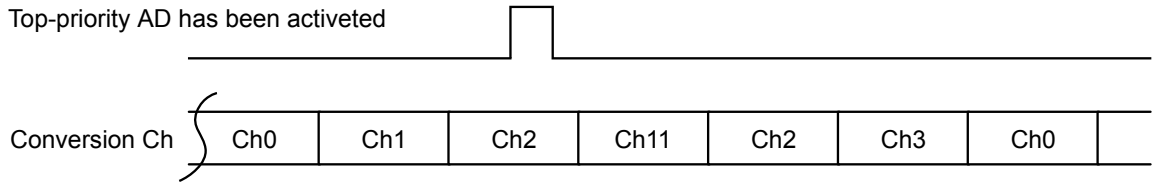
13.4.5.3 Top-priority AD conversion during normal AD conversion

If top-priority AD conversion has been activated during normal AD conversion, ongoing normal AD conversion is suspended, and restarts normal AD conversion after top-priority AD conversion is completed.

If ADMOD2<HPADCE> is set to "1" during normal AD conversion, ongoing normal AD conversion is suspended, and the top-priority AD conversion starts; specifically, AD conversion (fixed-channel single conversion) is executed for a channel designated by ADMOD2<HPADCH>. After the result of this top-priority AD conversion is stored in the storage register ADREGSP, normal AD conversion is resumed.

If H/W activation of top-priority AD conversion is authorized during normal AD conversion, ongoing AD conversion is discontinued when requirements for activation using a H/W activation resource are met, and top-priority AD conversion (fixed-channel single conversion) starts for a channel designated by ADMOD2<HPADCH>. After the result of this top-priority AD conversion is stored in the storage register ADREGSP, normal AD conversion is resumed.

For example, if channel repeat conversion is activated for channels AIN0 through AIN3 and if <HPADCE> is set to "1" during AIN2 conversion, AIN2 conversion is suspended, and conversion is performed for a channel designated by <HPADCH> (AIN11 in the case shown below). After the result of conversion is stored in ADREGSP, channel repeat conversion is resumed, starting from AIN2.



13.4.5.4 Stopping Repeat Conversion Mode

To stop the AD conversion operation in the repeat conversion mode (fixed-channel repeat conversion mode or channel scan conversion mode), write "0" to ADMOD0<REPEAT>. When ongoing AD conversion is completed, the repeat conversion mode terminates, and ADMOD0<ADBFN> is set to "0".

13.4.5.5 Reactivating normal AD conversion

To reactivate normal AD conversion while the conversion is underway, a software reset (ADMOD3 <ADRST>) must be performed before starting AD conversion. The H/W activation method must not be used to reactivate normal AD conversion.

13.4.5.6 Conversion completion

(1) Normal AD conversion completion

When normal AD conversion is completed, the AD conversion completion interrupt (INTAD) is generated. The result of AD conversion is stored in the storage register, and two registers change: the register ADMOD0 <EOCFN> which indicates the completion of AD conversion and the register ADMOD0 <ADBFN>.

For details, refer to Table 13-2 and Table 13-3 to confirm storage register corresponding to the conversion mode.

Interrupt requests, flag changes are as shown below.

- Fixed-channel single conversion mode

  After AD conversion completed, ADMOD0<EOCFN> is set to "1", ADMOD0<ADBFN> is cleared to "0", and the interrupt request is generated.

- Channel scan single conversion mode

  After the channel scan conversion is completed, ADMOD0<EOCFN> is set to "1", ADMOD0<ADBFN> is set to "0", and the interrupt request INTAD is generated.

- Fixed-channel repeat conversion mode

  ADMOD0<ADBFN> is not cleared to "0". It remains at "1". The timing with which the interrupt request INTAD is generated can be selected by setting ADMOD0<ITM> to an appropriate setting. ADMOD0<EOCFN> is set with the same timing as this interrupt INTAD is generated.

- Channel scan repeat conversion mode

  Each time one AD scan conversion is completed, ADMOD0 <EOCF> is set to "1" and interrupt request INTAD is generated. ADMOD0<ADBFN> is not cleared to "0". It remains at "1".

(2) Top-priority AD conversion completion

After the AD conversion is completed, the top-priority AD conversion completion interrupt (INTADHP) is generated, and ADMOD2<EOCFHP> which indicates the completion of top-priority AD conversion is set to "1".

AD conversion results are stored in the AD conversion result register SP.

(3) Data polling

To confirm the completion of AD conversion without using interrupts, data polling can be used. When AD conversion is completed, ADMOD0<EOCFN> is set to "1". To confirm the completion of AD conversion and to obtain the results, poll this bit.

AD conversion result storage register must be read by half word or word access. If <OVRx> = “0” and <ADRxRF> = "1", a correct conversion result has been obtained.

13.4.5.7 Interrupt generation timings and AD conversion result storage register

Table 13-1 shows a relation in the following three items: AD conversion modes, interrupt generation timings and flag operations. Table 13-2 and Table 13-3 shows a relation between analog channel inputs and AD conversion result registers.

Table 13-1 Relations in conversion modes, interrupt generation timings and flag operations

Conversion mode		Scan/repeat mode setting			Interrupt generation timing	ADMOD0<EOCFN>/ ADMOD2<EOCFHP> set timing (Note)	ADMOD0	ADMOD2
		ADMOD0 <REPEAT>	ADMOD0 <SCAIN>	ADMOD0 <ITM[1:0]>			<ADBFN> (After the interrupt is generated)	<ADBFHP>
Normal conversion	Fixed-channel single conversion	0	0	–	After generation is completed.	After conversion is completed.	0	–
	Fixed-channel repeat conversion	1	0	00	Each time one conversion is completed.	After one conversion is completed.	1	–
				01	Each time four conversion is completed.	After four conversions are completed.	1	–
				10	Each time eight conversion is completed.	After eight conversions are completed.	1	–
	Channel scan single conversion	0	1	–	After scan conversion is completed.	After scan conversion is completed.	0	–
	Channel scan repeat conversion	1	1	–	After one scan conversion is completed.	After one scan conversion is completed.	1	–
Top-priority conversion		–	–	–	After completion is completed.	Conversion completion	–	0

Note: ADMOD0<EOCFN> and ADMOD2<EOCFHP> are cleared upon read.

Table 13-2 Result registers (Fixed-channel repet conversion mode)

<ITM[1:0]>	Result register
00 Generate in interrupt once every single conversion	ADREG0
01 Generate interrupt once every 4 conversions	ADREG0 to ADREG3
10 Generate interrupt once every 8 conversions	ADREG0 to ADREG7

Table 13-3 Result registers (Except for fixed-channel repet conversion mode)

ADMOD1 <ADCH[3:0]>	ADMOD0 <SCAN>=0		ADMOD0 <SCAN>=1					
	Fixed channel		<ADSCN>=00 4-channel scan		<ADSCN>=00 8-channel scan		<ADSCN>=00 12-channel scan	
	Conversion channel	Result register	Conversion channel	Result register	Conversion channel	Result register	Conversion channel	Result register
0000	AIN0	ADREG0	AIN0	ADREG0	AIN0	ADREG0	AIN0	ADREG0
0001	AIN1	ADREG1	AIN0 to AIN1	ADREG0 to ADREG1	AIN0 to AIN1	ADREG0 to ADREG1	AIN0 to AIN1	ADREG0 to ADREG1
0010	AIN2	ADREG2	AIN0 to AIN2	ADREG0 to ADREG2	AIN0 to AIN2	ADREG0 to ADREG2	AIN0 to AIN2	ADREG0 to ADREG2
0011	AIN3	ADREG3	AIN0 to AIN3	ADREG0 to ADREG3	AIN0 to AIN3	ADREG0 to ADREG3	AIN0 to AIN3	ADREG0 to ADREG3
0100	AIN4	ADREG4	AIN4	ADREG4	AIN0 to AIN4	ADREG0 to ADREG4	AIN0 to AIN4	ADREG0 to ADREG4
0101	AIN5	ADREG5	AIN4 to AIN5	ADREG4 to ADREG5	AIN0 to AIN5	ADREG0 to ADREG5	AIN0 to AIN5	ADREG0 to ADREG5
0110	AIN6	ADREG6	AIN4 to AIN6	ADREG4 to ADREG6	AIN0 to AIN6	ADREG0 to ADREG6	AIN0 to AIN6	ADREG0 to ADREG6
0111	AIN7	ADREG7	AIN4 to AIN7	ADREG4 to ADREG7	AIN0 to AIN7	ADREG0 to ADREG7	AIN0 to AIN7	ADREG0 to ADREG7
1000	AIN8	ADREG0	AIN8	ADREG0	AIN8	ADREG0	AIN0 to AIN8	ADREG0 to ADREG8
1001	AIN9	ADREG1	AIN8 to AIN9	ADREG0 to ADREG1	AIN8 to AIN9	ADREG0 to ADREG1	AIN0 to AIN9	ADREG0 to ADREG9
1010	AIN10	ADREG2	AIN8 to AIN10	ADREG0 to ADREG2	AIN8 to AIN10	ADREG0 to ADREG2	AIN0 to AIN10	ADREG0 to ADREG10
1011	AIN11	ADREG3	AIN8 to AIN11	ADREG0 to ADREG3	AIN8 to AIN11	ADREG0 to ADREG3	AIN0 to AIN11	ADREG0 to ADREG11
1100	AIN12	ADREG4	AIN12	ADREG4	AIN8 to AIN12	ADREG0 to ADREG4	-	-
1101	AIN13	ADREG5	AIN12 to AIN13	ADREG4 to ADREG5	AIN8 to AIN13	ADREG0 to ADREG5	-	-
1110	AIN14	ADREG6	AIN12 to AIN14	ADREG4 to ADREG6	AIN8 to AIN14	ADREG0 to ADREG6	-	-
1111	AIN15	ADREG7	AIN12 to AIN15	ADREG4 to ADREG7	AIN8 to AIN15	ADREG0 to ADREG7	-	-

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.

14. 24-bit ΔΣ Analog/Digital Converter (DSADC)

A reference voltage circuit (BGR) used in the DSADC is shared with a temperature sensor and needs to set the control register (TEMPEN) of temperature sensor.

14.1 Features

DSADC has the following features:

- Conversion start

  Conversion started by software
- Conversion modes
  - Single conversion
  - Repeat conversion
- Status flags
  - Conversion result store flag
  - Overrun flag
  - Conversion end flag
  - Conversion flag
- Conversion clock can be divided by below ratios.

  fc/1, fc/2, fc/4, fc/8
- Conversion end interrupt output
- Conversion start correct function
- Synchronous start function for multiple units

When DSADC is used, provide pin treatments as follows:

- Do not connect VREFINx to a reference voltage.
- Connect AGNDREFx to DVSS level.
- Connect a 1μF capacitor to between VREFINx and AGNDREFx.

When DSADC is not used, below settings are required.

- Connect AGNDREFx to DVSS level.

When a temperature sensor is also not used, a reference voltage circuit requires below settings.

- Connect DSRVDD3 and SRVDD to DVDD3.
- Connect DSRVSS to DVSS.

14.2 Block Diagram

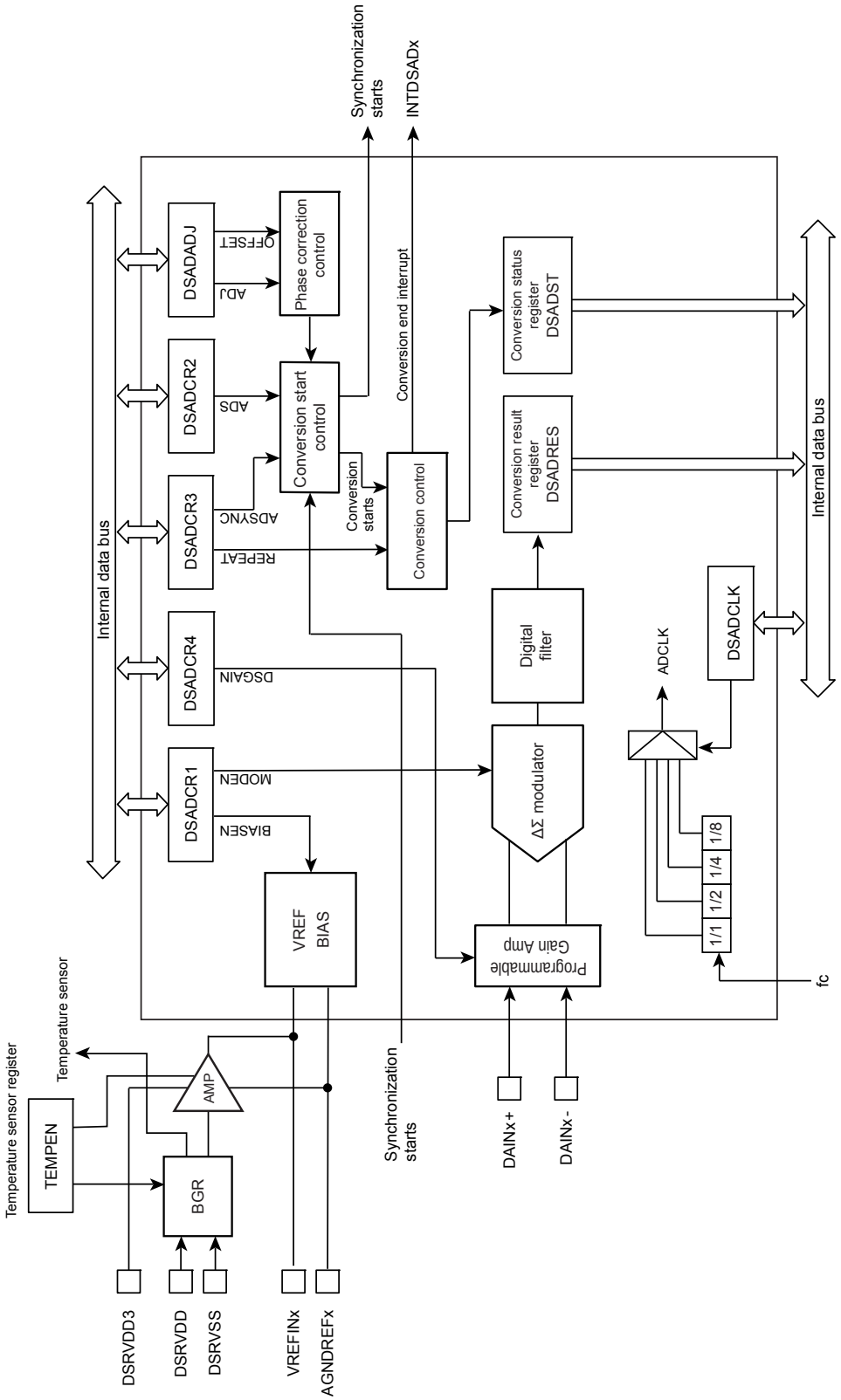


Figure 14-1 Block diagram of 24-bit ΔΣAD converter

14.3 Registers

14.3.1 Register List

The table below shows control registers and their addresses.

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

Register name		Address(Base+)
Clock setting register	DSADCLK	0x0000
Control register 0	DSADCR0	0x0004
Control register 1	DSADCR1	0x0008
Control register 2	DSADCR2	0x000C
Control register 3	DSADCR3	0x0010
Control register 4	DSADCR4	0x0014
Correction register	DSADADJ	0x0030
Conversion status register	DSADST	0x0040
Conversion result stored register	DSADRES	0x0044

14.3.2 Details of Registers

14.3.2.1 DSADCLK (Conversion clock setting register)

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

Bit	Bit Symbol	Type	Function
31-3	-	R	Read as "0".
2-0	ADCLK[2:0]	R/W	AD conversion clock selection 000: fc/1 001: fc/2 010: fc/4 011: fc/8 100-111: Prohibited

Note 1: Change <ADCLK[2:0]> register under the condition where DSADCR1<BIASEN>=<MODEN>=0 and AD conversion stops.

Note 2: When synchronous start function is used, select same conversion clock in all units.

14.3.2.2 DSADCR0 (Control register 0)

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

Bit	Bit Symbol	Type	Function
31-2	−	−	Read as "0".
1-0	ADRST	W	Software reset (Note) Write "10" then "01" to generate a software reset. Internal circuits and registers except DSADCLK are initialized.

Note: Valid only when DSADCR1<BIASEN>="1".

14.3.2.3 DSADCR1 (Control register 1)

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

Bit	Bit Symbol	Type	Function
31-2	−	−	Read as "0".
1	BIASEN	R/W	Bias control 0: Stop 1: Operation
0	MODEN	R/W	Modulator control 0: Stop 1: Control

14.3.2.4 DSADCR2 (Control register 2)

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

Bit	Bit Symbol	Type	Function
31-1	−	−	Read as "0".
0	ADS	W	Conversion starts 1: Start conversion Set "1" to start conversion. Write "0" has no effect. Read as "0".

14.3.2.5 DSADCR3 (Control register 3)

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

Bit	Bit Symbol	Type	Function
31-9	−	−	Read as "0".
8	ADSYNC	R/W	Synchronous mode 0: Asynchronous operation 1: Synchronous operation Set "1" to start conversion simultaneously among multiple units. (note)
7-1	−	R	Read as "0".
0	REPEAT	R/W	Conversion mode 0: Single conversion 1: Repeat conversion Specifies a conversion mode. If single conversion is set while the repeat conversion mode is working, the conversion will automatically stop.

Note: Set "1" to units only used for slaves. A unit used for a master is set to "0". For the combination of master channels and slave channels of this product, refer to Chapter Product Information.

14.3.2.6 DSADCR4 (Control register 4)

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

Bit	Bit Symbol	Type	Function
31-3	−	−	Read as "0".
2-0	DSGAIN[2:0]	R/W	Amplifier setting 000: ×1 001: ×2 010: ×4 011: ×8 100: ×16 101-111: Reserved

14.3.2.7 DSADADJ (Correction register)

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

Bit	Bit Symbol	Type	Function
31-16	OFFSET	R/W	Conversion start correction time Set a time period from setting "1" to DSADCR2<ADS> to starting conversion. A delay of <OFFSET>×1/fsys is added.
15-1	−	R	Read as "0".
0	ADJ	R/W	Conversion start correction 0: No correction 1: Correction Set "1" to start conversion after the time delay set with <OFFSET> has elapsed since conversion started.

14.3.2.8 DSADST (Conversion Status Register)

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

Bit	Bit Symbol	Type	Function
31-2	−	R	Read as "0".
1	EOCF	R	Conversion end flag (note) 0: Conversion on-going 1: Conversion end
0	ADBF	R	Conversion flag 0: No conversion start 1: Conversion starts

Note: This bit is cleared by reading the DSADST register.

14.3.2.9 DSADRES (Conversion result register)

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

Bit	Bit Symbol	Type	Function
31-8	ADR[23:0]	R	Conversion result Conversion result is stored in two's complement format. If the bit is read during conversion, the previous conversion result is read.
7-2	-	R	Read as "0".
1	ADOVR	R	Overrun flag (note) 0: No generation 1: Generation If conversion result is overwritten before reading <ADR>, "1" is set.
0	ADRF	R	Conversion result store flag (note) 0: No result is stored. 1: Result is stored. If a conversion result is stored in <ADR>, "1" is set.

Note: This bit is cleared by reading the DSADRES register.

14.4 Operation Description

14.4.1 Boot-up and Stop Procedures

This section explains the procedure how to start DSADC and how to stop DSADC for transiting to the low power consumption mode. The table below shows registers required to set.

Register	Bit	Description
TEMPEN	EN0, EN1	Reference voltage circuit (note)
CGSYSCR	FCSTOP	Clock feed to DSADC and ADC
DSADCLK	DSADCLK	Divide the conversion clock
DSADCR4	DSGAIN	Gain setting
DSADCR1	BIASEN, MODEN	Bias circuit and modulator circuit operation

Note: A reference voltage circuit is shared with a temperature sensor.

14.4.1.1 Boot-up

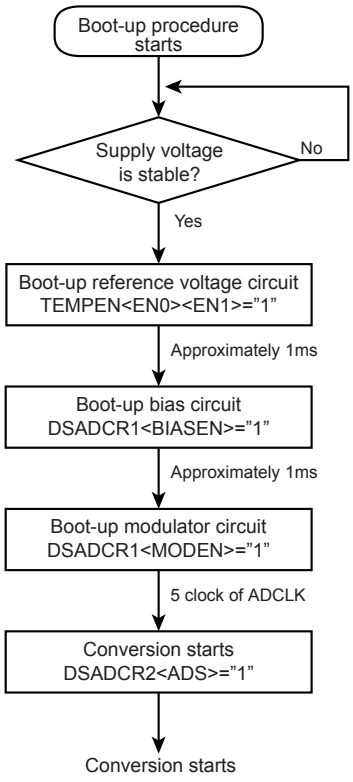


Figure 14-2 Boot-up procedure

Perform following procedure while supply voltage is stable.

1. Boot-up the reference voltage circuit

   Set "1" to TEMPEN<EN0><EN1> and wait 1ms or more to be stable.

   TEMPEN<EN1> must be enabled when TEMPEN<EN0> is enabled. Setting both <EN0> and <EN1> at the same time is capable.

2. Boot-up the bias circuit

   Set "1" to DSADCR1<BIASEN> and wait 1ms or more to be stable.

   Feed a conversion clock before DSADCR1<BIASEN> is set. For details of the conversion clock, refer to "1.4.2 Conversion Clock (ADCLK)".

3. Boot-up the modulator circuit

   Set "1" to DSADCR1<MODEN>.

   After ADCLK will have elapsed for 5 clocks, a conversion will be enabled.

   Set the conversion mode (DSADCR3<REPEAT>) and gain setting (DSADCR4<DSGAIN>) before starting conversion.

14.4.1.2 Stop

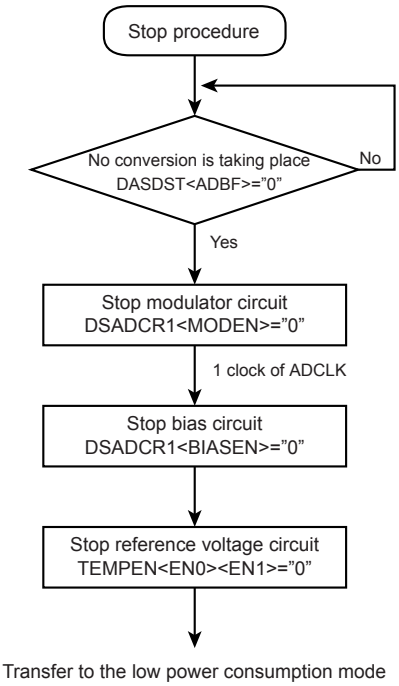


Figure 14-3 Stop procedure

Perform following procedure while the conversion is finished. (DSADST<ADBF>="0")

1. Stopping the modulator circuit

   Set "0" to DSADCR1<MODEN>

2. Stopping the bias circuit

   Set "0" to DSADCR1<BIASEN> after ADCLK will have elapsed one clock or more since the modulator circuit is stopped.

3. Stopping the reference voltage generation circuit

   Set "0" to TEMPEN<EN0><EN1>

Note: The reference voltage circuit is shared with the temperature sensor. While the temperature sensor is operating, do not stop TEMPEN<EN0> ("0" setting means stop).

14.4.2 Conversion Clock (ADCLK)

A conversion clock fed to DSADC is shared with a sequential comparison type AD converter and will operate after reset. The clock is stopped with CGSYSCR<FCSTOP> and also can be divided by DSADCLK.

Change or stop the conversion clock under DSADC stops. (Both DSADCR1<BIASEN> and <MODEN> are set to "0".)

When synchronous start function is used, select same conversion clock in all units.

14.4.2.1 Conversion Time

A conversion time can be calculated by the formula below where the frequency of ADCLK is f_{ADCLK}.

$$\text{Conversion time} = 1 / f_{ADCLK} \times 2640 + \text{Fixed delay time [s]}$$

A fixed delay time is 673 to 675 clocks at the first conversion in the repeat conversion and single conversion. In the repeat conversion, a fixed delay time is 0 clock at the 2nd conversion or later.

For example, where fc/1 is selected at fc=16MHz, a conversion time will be 165μs after the second conversion or later.

14.4.2.2 Transition to Low Power Consumption Mode

When a transition to the SLOW mode takes place, stop DSADC along with the procedure "14.4.1 Boot-up and Stop Procedures" then stop the clock fed to DSADC with CGSYSCR<FCSTOP>.

When a transition to the STOP or SLEEP mode takes place, stop DSADC along with the stopping procedure. The clock fed to DSADC is automatically stopped.

14.4.3 Conversion Mode

The DSADC provides two types of conversion mode such as the single mode and repeat mode. In the single mode, a conversion is performed once and in the repeat mode a conversion is sequentially performed. The mode is set with DSADCR3<REPEAT>.

14.4.4 Conversion Start

Set "1" to DSADCR2<ADS> to start conversion.

14.4.5 Conversion Status

A conversion status can be checked with DSADST.

During the conversion, DSADST<ADBF> becomes "1". After conversion, DSADST<EOCF> becomes "1". To clear <EOCF>, read DSADST.

In the repeat conversion, DSADST<ADBF> holds "1" during operation. When the repeat conversion is complete, DSADST<ADBF> is cleared to "0".

14.4.6 Conversion Stop

In the single mode, when the conversion is complete, DSADC will automatically stop.

In the repeat mode, in order to stop a conversion, set "0" to DSADCR3<REPEAT>. DSADC will suspend the current conversion then stop. At this time a conversion end interrupt will not occur.

Note: If repeat conversion is complete by setting "0" to <REPEAT>, do not modify other bits of DSADCR3.

14.4.7 Conversion End

When the conversion ends, a conversion end interrupt will occur. This conversion result is stored in DSADRES<ADR>, and DSADRES<ADRF> is set to "1".

If the conversion end interrupt is not used, poll DSADST<EOCF>. If <EOCF> is "1", the conversion is complete.

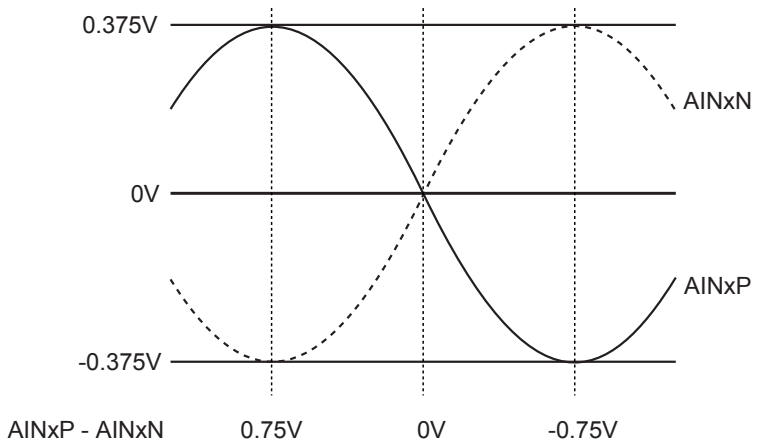
If <ADR> is written to next result before reading the current value, <ADOVR> is set to "1". To clear <ADRF> and <ADOVR>, read DSADRES.

14.4.8 Conversion Result

Because an input range of AINxP to AINxN is -0.375V to 0.375V, the maximum input amplitude is ±0.75V. At this time, conversion results are as follows:

Note: VREFINx = 2.75V

AINxP - AINxN	Conversion result
0.75V	0x4FC2BF
0V	0x000000
-0.75V	0xB03D41



14.5 Synchronous Start Function

A master unit and slave unit can start conversion simultaneously. For details, refer to "Product information" to confirm an assignment of a master and slave in this product.

14.5.1 Boot-up

The following flowchart shows the setting procedure when the synchronous start function is used.

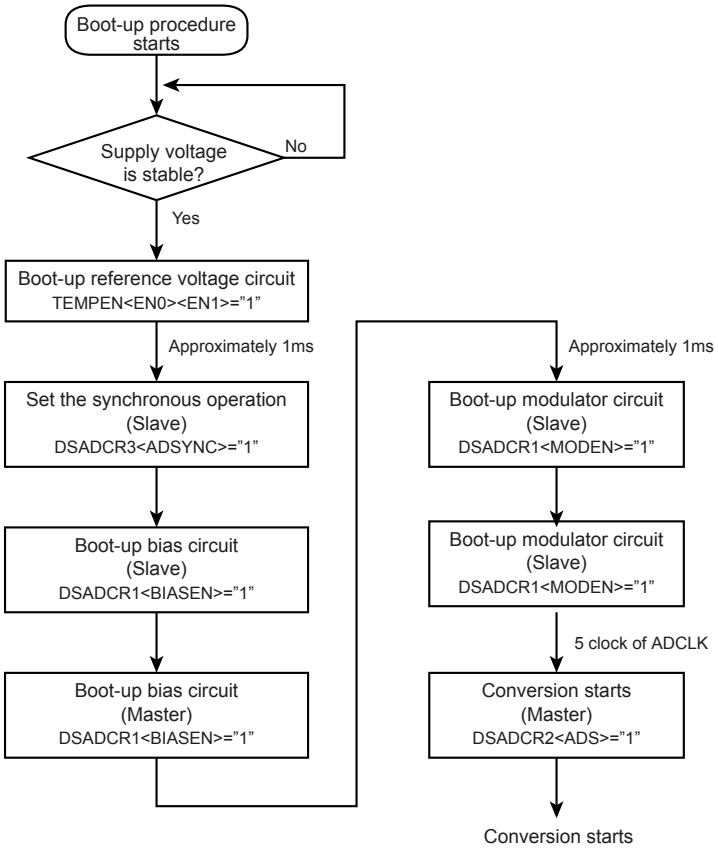


Figure 14-4 Synchronous start function procedure

1. Boot-up the reference voltage circuit

   Set "1" to TEMPEN<EN0><EN1> and wait 1ms or more to be stable.

   TEMPEN<EN1> must be enabled when TEMPEN<EN0> is enabled. Setting both <EN0> and <EN1> at the same time is capable.

2. Set the synchronous operation

   When the synchronous start function is used, set the slave unit DSADCR3<ADSYNC> to "1" and set the master unit <ADSYNC> to "0".

3. Boot-up the bias circuit

   Set "1" to DSADCR1<BIASEN> and wait 1ms or more to be stable. In the case of <BIASEN>, set slave side first, then set the master side.

   Feed a conversion clock before DSADCR<BIASEN> is set. For details of the conversion clock, refer to "1.4.2 Conversion Clock (ADCLK)".

4. Boot-up the modulation circuit

   Set "1" to DSADCR1<MODEN>.

   After ADCLK will have elapsed for 5 clocks, a conversion will be enabled.

Conversion mode (DSADCR3<REPEAT> and gain setting (DSADCR4<DSGAIN>) can be set in each unit. Set them in each unit before conversion starts.

14.5.2 Stop

In the single conversion mode, DSADC stops when conversion is complete in each unit.

In the repeat conversion mode, stop DSADC as follows:

- Stop only slave side

  Change the conversion mode of slave (set DSADCR3<REPEAT> to "0") or perform a software reset.

  At this time, master side continues conversion.
- Stop only master side

  Cancel the synchronous operation by setting DSADCR3<ADSYNC> of the slave to "0", then stop the master side by changing the conversion mode or by performing a software reset.

  At this time, slave side continues conversion.
- Stop both master side and slave side

  Stop slave side then master side by changing the conversion mode or by performing a software reset.

14.6 Conversion Start Correction Function

By using a conversion start correction function, the conversion start time can be delayed from the time when DSADCR2<ADS> is set to "1".

This function is enabled by setting "1" to DSADADJ<ADJ>. The delay time is set with DSADADJ<OFFSET>. After delay time of <OFFSET>×1/fsys has elapsed, conversion will start synchronously with ADCLK.

In the synchronous operation, a desired delay time can be set to <OFFSET> in each unit. Conversion will start after the time defined in <OFFSET> has elapsed from the time when <ADS> of master side is set to "1".

Duration between setting "1" to <ADS> and starting conversion, do not modify a value of <OFFSET>.

15. Temperature Sensor (TEMP)

15.1 Outline

The MCU measures a relative temperature using a temperature sensor.

A temperature sensor outputs a voltage based on the reference voltage circuit (BGR) according to temperatures. The output voltage is input to Channel 2 in the analog/digital converter (ADC)Unit D in the ΔΣ analog/digital converter (DSADC). With AD conversion, a corresponding digital value to temperatures is obtained.

Note: The reference voltage circuit (BGR) is shared with a ΔΣ type analog/digital converter (DSADC).

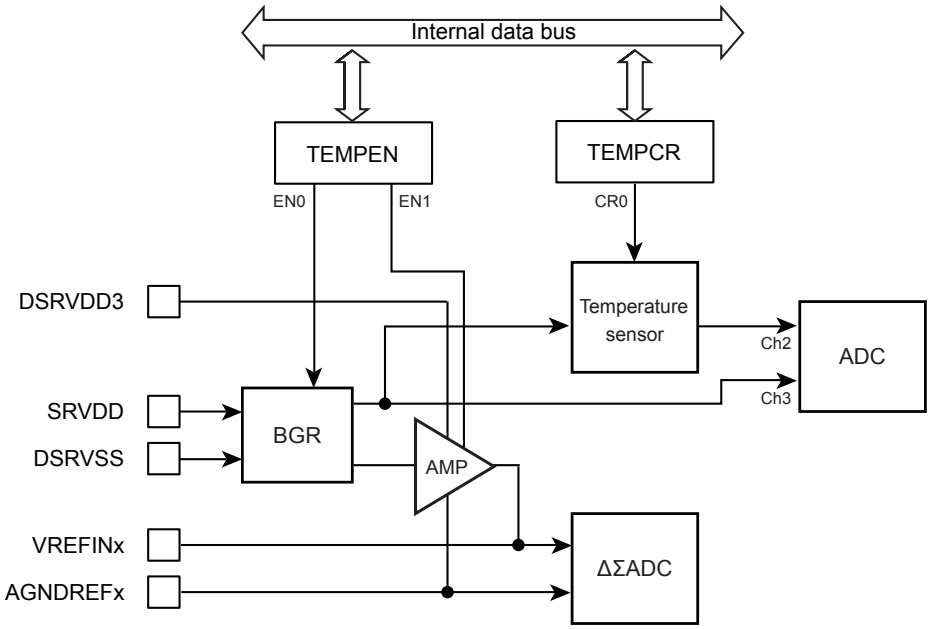
A difference among the temperature sensor output voltages is linearity related to temperature changes. To obtain a relative temperature, collect data under several conditions.

Channel 3 of ADC is input a 1V from BGR. In variable power voltage system, power voltage can be relatively identified by the result where BGR voltage was performed AD conversion.

If the temperature sensor or DSADC is not used, the reference voltage circuit requires below settings.

- Connect DSRVDD3 and SRVDD to DVDD3
- Connect DSRVSS to DGND

15.2 Block diagram



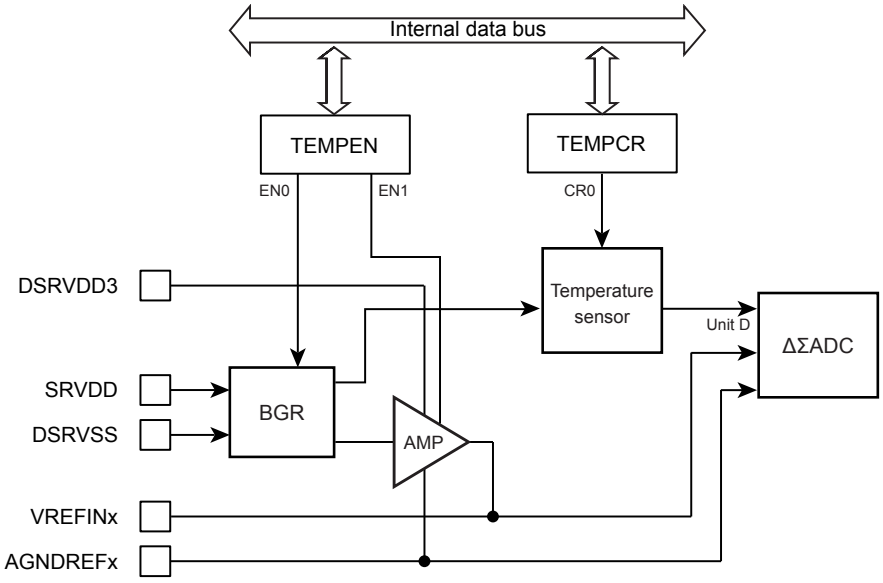


Figure 15-1 Block diagram of Temperature sensor

15.3 Registers

15.3.1 Register List

Then table below shows control registers and their addresses.

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

Register name		Address(Base+)
Enable register	TEMPEN	0x0000
Control register	TEMPCR	0x0004

15.3.2 Details of Register

15.3.2.1 TEMPEN (Enable register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	EN1	EN0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	−	R	Read as "0".
1	EN1	R/W	AMP operation(note1)(note2) 0: Disabled 1: Enabled Sets AMP for ΔΣADC to enable/disable.
0	EN0	R/W	BGR operation 0: Disabled 1: Enabled Sets the reference voltage circuit to enable/disable.

Note: TEMPEN<EN1> must be enabled when TEMPEN<EN0> is enabled. Setting both <EN0> and <EN1> at the same time is possible.

Note 1: TEMPEN<EN1> must be enabled when TEMPEN<EN0> is enabled. Setting both <EN0> and <EN1> at the same time is possible.

Note 2: Do not enable AMP when the reference voltage is applied to VREFINx using the ΔΣADC.

15.3.2.2 TEMPCR (Control Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	-	CR0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-1	−	−	Read as "0".
0	CR0	R/W	Temperature sensor operation 0: Disabled 1: Enabled Sets the temperature sensor to enable/disable.

15.4 Operation Description

1. Boot-up

   Perform the following procedure while power supply voltage is stable.

   1. Boot-up the reference voltage circuit

      Set "1" to TEMPEN<EN0> and wait 1 ms or more to be stable.

   2. Boot-up the temperature sensor

      Set "1" to TEMPCR<CR0>

      Approximately after 10μs, an output voltage is enabled.

2. Stop

   Perform the following procedure.

   1. Stop the temperature sensor

      Set "0" to TEMPCR<CR0>

   2. Stop the reference voltage circuit

      Set "0" to TEMPEN<EN0>

Note: The reference voltage circuit is shared with ΔΣADC. Do not set TEMPEN<EN0> to "0" (stop) during the ΔΣADC operation.

16. Real Time Clock (RTC)

16.1 Function

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 function (available only in the products that have $\overline{\text{ALARM}}$ pin.)
6. Alarm interrupt
7. Clock correction function
8. 1 Hz clock output

16.2 Block Diagram

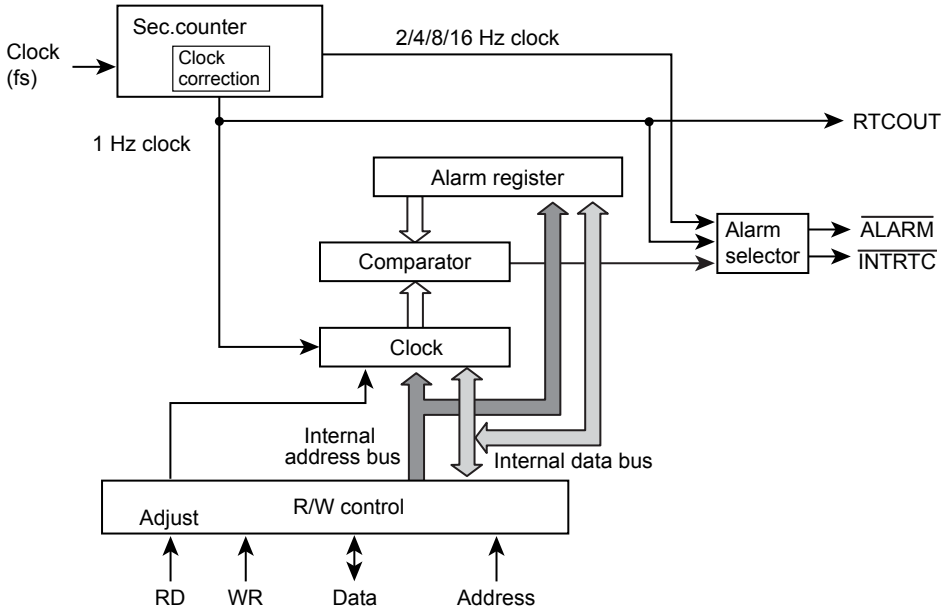


Figure 16-1 Block Diagram

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.

16.3 Detailed Description Register

16.3.1 Register List

The table below shows control registers and their addresses.

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

RTC has two functions, PAGE0 (clock) and PAGE1 (alarm), which share some parts of registers.

The PAGE can be selected by setting RTCPAGER<PAGE >.

Register name		Address(Base+)
Second column register (only PAGE0)	RTCSECR	0x0000
Minute column register	RTCMINR	0x0001
Hour column register	RTCHOURR	0x0002
- (Note)	-	0x0003
Day of the week column register	RTCDAYR	0x0004
Day column register	RTCDATER	0x0005
Month column register (PAGE0) Selection register of 24-hour,12-hour (PAGE1)	RTCMONTHR	0x0006
Year column register (PAGE0) Leap year register (PAGE1)	RTCYEARR	0x0007
PAGE register	RTCPAGER	0x0008
- (Note)	-	0x0009
- (Note)	-	0x000A
- (Note)	-	0x000B
Reset register	RTCRESTR	0x000C
- (Note)	-	0x000D
Protect register	RTCPROTECT	0x000E
Correction Function Control Register	RTCADJCTL	0x000F
Correction Value Register	RTCADJDAT	0x0010, 0x0011

Note:"0" is read by reading the address. Writing is disregarded.

16.3.2 Control Register

Reset operation initializes the following registers.

- RTCPAGER<PAGE>, <ADJUST>, <INTENA>
- RTCRESTR
- RTCPROTECT
- RTCADJCTL
- RTCADJDAT

Other clock-related registers are not initialized by reset operation.

Before using 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 "16.4.3 Entering the Low Power Consumption Mode" for more information.

Table 16-1 PAGE0 (clock function) register

Symbol	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Function
RTCSECR		–	40sec.	20sec.	10sec.	8sec.	4sec.	2sec.	1sec.	Second column
RTCMINR		–	40min.	20min.	10min.	8min.	4min.	2min.	1min.	Minute column
RTCHOURR		–	–	20hours PM/AM	10hour	8hour	4hour	2hour	1hours	Hour column
RTCDAYR		–	–	–	–	–	Day of the week			Day of the week column
RTCDATER		–	–	Day20	Day10	Day8	Day4	Day2	Day1	Day column
RTCMONTHR		–	–	–	Oct.	Aug.	Apr.	Feb.	Jan.	Month column
RTCYEARR		year 80	year 40	year20	year 10	year 8	year 4	year 2	year 1	Year column (lower two columns)
RTCPAGER		Interrupt enable	–	–	Adjust-ment function	Clock en-able	Alarm en-able	–	PAGE setting	PAGE register
RTCRESTR		1 Hz enable	16 Hz enable	Clock reset	Alarm reset	–	2Hz enable	4 Hz enable	8 Hz enable	Reset register
RTCPROTECT		Protect code								Clock correction function register protection
RTCADJCTL		–	–	–	–	Correction reference time			Correction enable	Correction function control
RTCADJDAT		Correction value								Correction value

Note: Reading RTCSECR, RTCMINR, RTCHOURR, RTCDAYR, RTCMONTHR, RTCYEARR of PAGE0 captures the current state.

Table 16-2 PAGE1 (alarm function) registers

Symbol	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Function
RTCSECR		–	–	–	–	–	–	–	–	–
RTCMINR		–	40min.	20min.	10min.	8min.	4min.	2min.	1min.	Minute column
RTCHOURR		–	–	20hours PM/AM	10hour	8hour	4hour	2hour	1hour	Hour column
RTCDAYR		–	–	–	–	–	Day of the week			Day of the week column
RTCDATER		–	–	Day20	Day10	Day8	Day4	Day2	Day1	Day column
RTCMONTHR		–	–	–	–	–	–	–	24/12	24-hour clock mode
RTCYEARR		–	–	–	–	–	–	Leap-year setting		Leap-year mode
RTCPAGER		Interrupt enable	–	–	Adjust-ment function	Clock en-able	Alarm en-able	–	PAGE setting	PAGE register
RTCRESTR		1 Hz Enable	16 Hz Enable	Clock reset	Alarm reset	–	2 Hz enable	4 Hz enable	8 Hz enable	Reset register
RTCPROTECT		Protect code								Clock correction function register protection
RTCADJCTL		–	–	–	–	Correction reference time			Correc-tion ena-ble	Correction function control
RTCADJDAT		Correction value								Correction value

Note 1: Reading RTCMINR, RTCHOURR, RTCDAYR, RTCMONTHR, RTCYEARR of PAGE1 captures the current state.

Note 2: RTCSECR, RTCMINR, RTCHOURR, RTCDAYR, RTCDATER, RTCMONTHR, RTCYEARR of PAGE0 and RTCYEARR of PAGE1 (for leap year) must be read twice and compare the data captured.

16.3.3 Detailed Description of Control Register

16.3.3.1 RTCSECR (Second column register (for PAGE0 only))

	7	6	5	4	3	2	1	0
bit symbol	-	SE						
After reset	0	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit Symbol	Type	Function
7	−	R	Read as 0.
6-0	SE	R/W	Setting digit register of second 000_0000 : 00sec. 001_0000 : 10sec. 010_0000 : 20sec. 000_0001 : 01sec. 001_0001 : 11sec. ・ 000_0010 : 02sec. 001_0010 : 12sec. 011_0000 : 30sec. 000_0011 : 03sec. 001_0011 : 13sec. ・ 000_0100 : 04sec. 001_0100 : 14sec. 100_0000 : 40sec. 000_0101 : 05sec. 001_0101 : 15sec. ・ 000_0110 : 06sec. 001_0110 : 16sec. 101_0000 : 50sec. 000_0111 : 07sec. 001_0111 : 17sec. ・ 000_1000 : 08sec. 001_1000 : 18sec. ・ 000_1001 : 09sec. 001_1001 : 19sec. 101_1001 : 59sec.

Note: The setting other than listed above is prohibited.

16.3.3.2 RTCMINR (Minute column register (PAGE0/1))

	7	6	5	4	3	2	1	0
Bit symbol	-	MI						
After reset	0	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit Symbol	Type	Function
7	−	R	Read as 0.
6-0	MI	R/W	Setting digit register of Minutes. 000_0000 : 00min. 001_0000 : 10min. 010_0000 : 20min. 000_0001 : 01min. 001_0001 : 11min. ・ 000_0010 : 02min. 001_0010 : 12min. 011_0000 : 30min. 000_0011 : 03min. 001_0011 : 13min. ・ 000_0100 : 04min. 001_0100 : 14min. 100_0000 : 40min. 000_0101 : 05min. 001_0101 : 15min. ・ 000_0110 : 06min. 001_0110 : 16min. 101_0000 : 50min. 000_0111 : 07min. 001_0111 : 17min. ・ 000_1000 : 08min. 001_1000 : 18min. ・ 000_1001 : 09min. 001_1001 : 19min. 101_1001 : 59min. 111_1111 : Don't compare Minutes at alarm function.

Note: The setting other than listed above is prohibited.

16.3.3.3 RTCHOURR (Hour column register(PAGE0/1))

(1) 24-hour clock mode (RTCMONTHR<MO0>= "1")

	7	6	5	4	3	2	1	0
Bit symbol	-	-	HO					
After reset	0	0	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit Symbol	Type	Function
7-6	−	R	Read as 0.
5-0	HO	R/W	Setting digit register of Hour. 00_0000 : 0 o'clock 01_0000 : 10 o'clock 10_0000 : 20 o'clock 00_0001 : 1 o'clock 01_0001 : 11 o'clock 10_0001 : 21 o'clock 00_0010 : 2 o'clock 01_0010 : 12 o'clock 10_0010 : 22 o'clock 00_0011 : 3 o'clock 01_0011 : 13 o'clock 10_0011 : 23 o'clock 00_0100 : 4 o'clock 01_0100 : 14 o'clock 00_0101 : 5 o'clock 01_0101 : 15 o'clock 00_0110 : 6 o'clock 01_0110 : 16 o'clock 00_0111 : 7 o'clock 01_0111 : 17 o'clock 00_1000 : 8 o'clock 01_1000 : 18 o'clock 00_1001 : 9 o'clock 01_1001 : 19 o'clock 11_1111 : Don't compare Hour at alarm function.

Note: The setting other than listed above is prohibited.

(2) 12-hour clock mode (RTCMONTHR<MO0> = "0")

	7	6	5	4	3	2	1	0
Bit symbol	-	-	HO					
After reset	0	0	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit Symbol	Type	Function
7-6	−	R	Read as 0.
5-0	HO	R/W	Setting digit register of Hour. (AM) (PM) 00_0000 : 0 o'clock 10_0000 : 0 o'clock 00_0001 : 1 o'clock 10_0001 : 1 o'clock 00_0010 : 2 o'clock 10_0010 : 2 o'clock 00_0011 : 3 o'clock 10_0011 : 3 o'clock 00_0100 : 4 o'clock 10_0100 : 4 o'clock 00_0101 : 5 o'clock 10_0101 : 5 o'clock 00_0110 : 6 o'clock 10_0110 : 6 o'clock 00_0111 : 7 o'clock 10_0111 : 7 o'clock 00_1000 : 8 o'clock 10_1000 : 8 o'clock 00_1001 : 9 o'clock 10_1001 : 9 o'clock 01_0000 : 10 o'clock 11_0000 : 10 o'clock 01_0001 : 11 o'clock 11_0001 : 11 o'clock 11_1111 : Don't compare hour at alarm function.

Note: The setting other than listed above is prohibited.

16.3.3.4 RTCDAYR (Day of the week column register(PAGE0/1))

	7	6	5	4	3	2	1	0
Bit symbol	-	-	-	-	-	WE		
After reset	0	0	0	0	0	Undefined	Undefined	Undefined

Bit	Bit Symbol	Type	Function
7-3	−	R	Read as 0.
2-0	WE	R/W	Setting digit register of day of the week. 000: Sunday 001: Monday 010: Tuesday 011: Wednesday 100: Thursday 101: Friday 110: Saturday 111: Don't' compare day of the week at alarm function.

Note: The setting other than listed above is prohibited.

16.3.3.5 RTCDATER (Day column register (for PAGE0/1 only))

	7	6	5	4	3	2	1	0
Bit symbol	-	-	DA					
After reset	0	0	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit Symbol	Type	Function
7-6	−	R	Read as 0.
5-0	DA	R/W	Setting digit register of day. 00_0001 : 1st day, 00_0010 : 2nd day, 00_0011 : 3rd day, 00_0100 : 4th day, 00_0101 : 5th day, 00_0110 : 6th day, 00_0111 : 7th day, 00_1000 : 8th day, 00_1001 : 9th day 01_0000 : 10th day, 01_0001 : 11th day, 01_0010 : 12th day, 01_0011 : 13th day, 01_0100 : 14th day, 01_0101 : 15th day, 01_0110 : 16th day, 01_0111 : 17th day, 01_1000 : 18th day, 01_1001 : 19th day 10_0000 : 20th day, 10_0001 : 21th day, 10_0010 : 22th day, 10_0011 : 23th day, 10_0100 : 24th day, 10_0101 : 25th day, 10_0110 : 26th day, 10_0111 : 27th day, 10_1000 : 28th day, 10_1001 : 29th day 11_0000 : 30th day, 11_0001 : 31th day 11_1111 : Don't compare day at alarm function.

Note 1: The setting other than listed above is prohibited.
Note 2: Do not set for non-existent days (e.g. 30th Feb.).

16.3.3.6 RTCMONTHR (Month column register (for PAGE0 only))

	7	6	5	4	3	2	1	0
Bit symbol	-	-	-	MO				
After reset	0	0	0	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit Symbol	Type	Function
7-5	−	R	Read as 0.
4-0	MO	R/W	Setting digit register of Month. 0_0001 : January 0_0111 : July 0_0010 : February 0_1000 : August 0_0011 : March 0_1001 : September 0_0100 : April 1_0000 : October 0_0101 : May 1_0001 : November 0_0110 : June 1_0010 : December

Note: The setting other than listed above is prohibited.

16.3.3.7 RTCMONTHR (Selection of 24-hour clock or 12-hour clock (for PAGE1 only))

	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	-	MO0
After reset	0	0	0	0	0	0	0	Undefined

Bit	Bit Symbol	Type	Function
7-1	−	R	Read as 0.
0	MO0	R/W	0: 12-hour 1: 24-hour

Note: Do not change the RTCMONTHR<MO0> while the RTC is in operation.

16.3.3.8 RTCYEARR (Year column register (for PAGE0 only))

	7	6	5	4	3	2	1	0
bit symbol	YE							
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit Symbol	Type	Function
7-0	YE	R/W	Setting digit register of Year. 0000_0000 : 00 year 0000_0001 : 01 years 0000_0010 : 02 years 0000_0011 : 03 years 0000_0100 : 04 years 0000_0101 : 05 years 0000_0110 : 06 years 0000_0111 : 07 years 0000_1000 : 08 years 0000_1001 : 09 years 0001_0000 : 10 years ・ 0010_0000 : 20 years ・ 0011_0000 : 30 years ・ 0100_0000 : 40 years ・ 01001_0000 : 50 years ・ 0110_0000 : 60 years ・ 0111_0000 : 70 years ・ 1000_0000 : 80 years ・ 1001_0000 : 90 years ・ ・ 1001_1001 : 99 years

Note: The setting other than listed above is prohibited.

16.3.3.9 RTCYEARR (Leap year register (for PAGE1 only))

	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	LEAP	
After reset	0	0	0	0	0	0	Undefined	Undefined

Bit	Bit Symbol	Type	Function
7-2	−	R	Read as 0.
1-0	LEAP	R/W	00 : A leap year 01 : one year after a leap year 10 : two years after a leap year 11 : three years after a leap year

16.3.3.10 RTCPAGER(PAGE register(PAGE0/1))

	7	6	5	4	3	2	1	0
Bit symbol	INTENA	-	-	ADJUST	ENATMR	ENAALM	-	PAGE
After reset	0	0	0	0	Undefined	Undefined	0	0

Bit	Bit Symbol	Type	Function
7	INTENA	R/W	INTRTC 0:Disable 1:Enable
6-5	−	R	Read as 0.
4	ADJUST	R/W	[Write] 0: Don't care 1: Sets ADJUST request 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". [Read] 0: ADJUST no request 1: ADJUST requested If "1" is read, it indicates that ADJUST is being executed. If "0" is read, it indicates that the execution is finished.
3	ENATMR	R/W	Clock 0: Disable 1: Enable
2	ENAALM	R/W	ALARM 0: Disable 1: Enable
1	−	R	Read as 0.
0	PAGE	R/W	PAGE selection 0:Selects Page0 1:Selects Page1

Note 1: A read-modify-write operation cannot be performed.

Note 2: To set interrupt enable bits to <ENATMR>, <ENAALM> and <INTENA>, you must follow the order specified here. Make sure not to set them at the same time (make sure that there is time lag between interrupt enable and clock/ alarm enable).To change the setting of <ENATMR> and <ENAALM>, <INTENA> must be disabled first.

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

16.3.3.11 RTCRESTR (Reset register (for PAGE0/1))

	7	6	5	4	3	2	1	0
Bit symbol	DIS1HZ	DIS16HZ	RSTTMR	RSTALM	-	DIS2HZ	DIS4HZ	DIS8HZ
After reset	1	1	0	0	0	1	1	1

Bit	Bit Symbol	Type	Function
7	DIS1HZ	R/W	1 Hz 0:Enable 1: Disable
6	DIS16HZ	R/W	16 Hz 0: Enable 1: Disable
5	RSTTMR	R/W	[Write] 0: Don't care 1: Sec.counter reset Resets the sec counter. The request is sampled using low-speed clock. [Read] 0: No reset request 1: RESET requested If "1" is read, it indicates that RESET is being executed. If "0" is read, it indicates that the execution is finished.
4	RSTALM	R/W	0:Don't care 1: Alarm reset 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
3	−	R	Read as 0.
2	DIS2HZ	R/W	2 Hz 0:Enable 1: Disable
1	DIS4HZ	R/W	4 Hz 0:Enable 1: Disable
0	DIS8HZ	R/W	8 Hz 0:Enable 1: Disable

Note: A read-modify-write operation cannot be performed.

The setting of <DIS1HZ>, <DIW2HZ>, <DIS4HZ> and <DIS16MHZ>, RTCPAGER<ENAALM> used for alarm, 1Hz, 2Hz, 4Hz, 8Hz and 16Hz interrupt is shown as below.

<DIS1HZ>	<DIS2HZ>	<DIS4HZ>	<DIS8HZ>	<DIS16HZ>	RTCPAGER <ENAALM>	Interrupt source signal
1	1	1	1	1	1	Alarm
0	1	1	1	1	0	1 Hz
1	0	1	1	1	0	2 Hz
1	1	0	1	1	0	4Hz
1	1	1	0	1	0	8Hz
1	1	1	1	0	0	16 Hz
Others						Interrupt not generated.

16.3.3.12 RTCPROTECT(Protect register)

	7	6	5	4	3	2	1	0
Bit symbol	RTCPROTECT							
After reset	1	1	0	0	0	0	0	1

Bit	Bit Symbol	Type	Function
7-0	RTCPROTECT	R/W	Clock correction function register protection 0xC1: Write enable. 0xC1: Write disable. In the initial state, RTCPROTECT is "0xC1" and write enable. If RTCPROTECT is set to a value other than "0xC1", RTCADJCTL and RTCADJDAT will be write disable.

16.3.3.13 RTCADJCTL (Correction Function Control Register)

	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	AJSEL			AJEN
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
7-4	−	R	Read as "0".
3-1	AJSEL	R/W	Correction reference time setting 000: 1 second 001: 10 seconds 010: 20 seconds 011: 30 seconds 100: 1 minute 101 - 111: Reserved Set a correction reference time.
0	AJEN	R/W	Correction function control 0: Disabled 1: Enabled

16.3.3.14 RTCADJDAT (Correction Value Register)

	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	ADJDAT
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADJDAT							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
15-9	−	R	Read as "0".
8-0	ADJDAT	R/W	Correction value 0_0000_0000 : No correction 0_0000_0001 : 32768 + 1 0_0000_0010 : 32768 + 2 ・ 0_1111_1110 : 32768 + 254 0_1111_1111 : 32768 + 255 1_0000_0000 : 32768 - 256 1_0000_0001 : 32768 - 255 ・ 1_1111_1110 : 32768 - 2 1_1111_1111 : 32768 - 1 Sets a correction value per second. The 8th is a sign bit. If the 8th bit is "0", a plus correction is applied. If the bit is "1", a minus correction is applied. Specifies a correction value using bit 7 to 0.

16.4 Operational Description

The RTC incorporates a second counter that generates a 1Hz signal from a 32.768 kHz signal.

The second counter operation must be taken into account when using the RTC.

16.4.1 Reading clock data

1. Using 1Hz interrupt

   The 1Hz interrupt is generated being synchronized with counting up of the second 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.

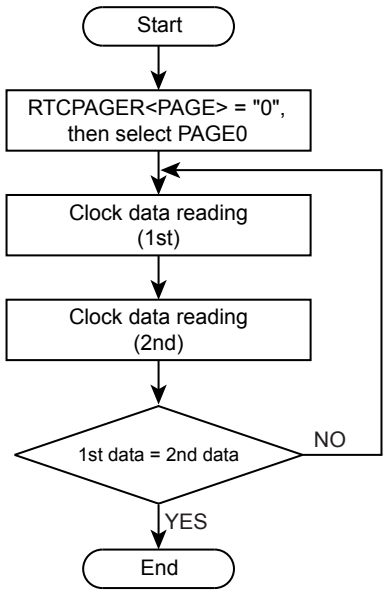


Figure 16-2 Flowchart of the clock data reading

16.4.2 Writing clock data

A carry during writing ruins correct data writing. The following procedure ensures the correct data writing.

1. Using 1 Hz interrupt

   The 1Hz interrupt is generated by being synchronized with counting up of the second 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 second 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.

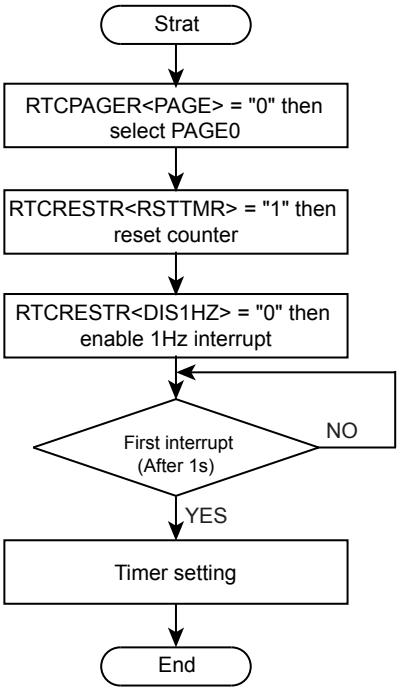


Figure 16-3 Flowchart of the clock data writing

3. Disabling the clock

Writing "0" to RTCPAGER<ENATMR> disables clock operation including a carry.

Stop the clock after the 1Hz-interrupt. The second counter keeps counting.

Set the clock again and enable the clock within one second before next 1Hz-interrupt

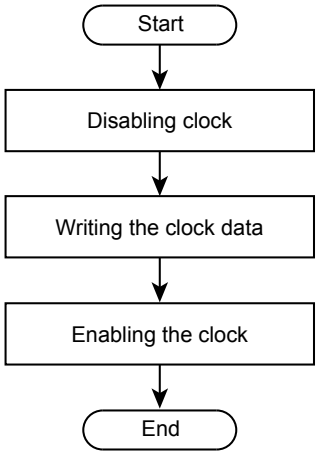


Figure 16-4 Flowchart of the disabling clock

16.4.3 Entering the Low Power Consumption Mode

To enter SLEEP mode, in which the system clock stops, after changing clock data, adjusting seconds or re-setting 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.

16.5 Alarm function

By writing "1" to RTCPAGER<PAGE>, the alarm function of the PAGE1 registers is enabled. One of the following signals is output to the $\overline{\text{ALARM}}$ pin if the product provide the ALARM pin.

1. "Low" pulse (when the alarm register corresponds with the clock)
2. 1, 2, 4, 8 or 16Hz cycle "Low" pulse

In any cases shown above, the RTC 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

16.5.1 Usage of alarm function

"Low" 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 (12:00) 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 interrupts

If some alarm registers are set to "1", RTC doesn't compare the term. For example, if RTCDATER is set to "11_1111" and RTCDAYR is set to "111", the alarm will be output at noon (12:00) every day.

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 fs (about 30μs) may occur for the time register setting to become valid.

16.5.2 1, 2, 4, 8 or 16 Hz cycle "Low" pulse

The RTC outputs a "Low" pulse cycle to the $\overline{\text{ALARM}}$ pin by setting RTCPAGER<INTENA>="1" after setting RTCPAGER<ENAALM>= "0" and RTCRESTR. It is required that one of RTCRESTR<DIS1HZ>, <DIS2HZ>, <DIS4HZ>, <DIS8HZ> or <DIS16HZ> is set to "0".

The RTC outputs one cycle pulse of low-speed clock which correspond to RTCRESTR setting. It generates an INTRTC interrupt simultaneously.

16.6 Clock Correction Function

The clock correction function can precisely adjust the deviation of the clock.

In the Figure 16-5, T1 indicates one second. One second is generated by counting fs (32768Hz) 32768 times. The clock correction function adjusts the number of counts of T2 that is an one second of the correction reference time (Tall). The correction reference time is selected either among 1, 10, 20, 30 seconds or 1 minute with RTCADJCTL<AJSEL>. A count value of T2 can be adjustable from 32768-255 to 32768+256 with RTCADJDAT<ADJDAT>.

Symbol	Item	Description
Tall	Correction reference time	Selects either among 1, 10, 20, 30 seconds or 1 minute with RTCADJCTL<AJSEL>.
T1	1 second	Counts fs 32768 times
T2	Count correction	Adjust a count value with RTCADJDAT<ADJDAT> by plus/minus 32768 counts

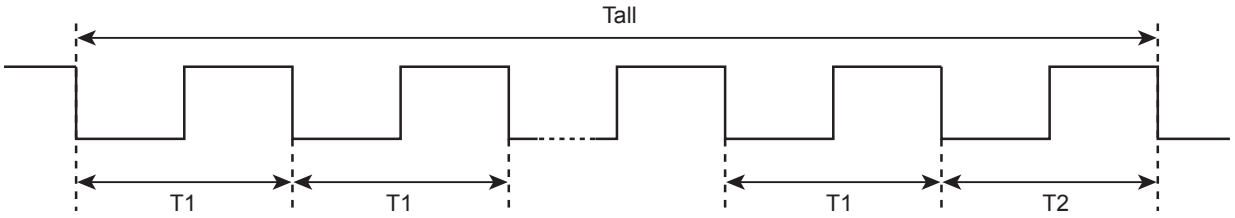


Figure 16-5 Clock Correction

The correction function related register, RTCADJCTL and RTCADJDAT, can be disabled with the RTCPROTECT register. In the initial state, RTCPROTECT is “0xC1” and write enable. If RTCPROTECT is set to a value other than “0xC1”, RTCADJCTL and RTCADJDAT will be write disable.

16.7 1Hz Clock Output Function

RTCOUT pin outputs 1Hz clock. This clock is adjusted to operate on a 50% duty ratio. If the clock correction function is used, a duty ratio may be varied due to the error corrections.

17. LCD Driver

The TMPM061FWFG has a driver and control circuit to directly drive a liquid crystal display (LCD) device. The pins to be connected to the LCD are as follows:

1. Segment output pins : 40 pins (SEG39 to SEG0)
2. Common output pins : 4 pins (COM3 to COM0)

In addition, the VLC pin is provided as a drive power supply pin, and the LV1 and LV2 pins are provided as external bleeder resistance connection pins.

Note: When the static, 1/3 or 1/2 duties are selected, unused common output pins should be opened. (It outputs bias voltage)

The LCD driver can directly drive the following five types of LCD:

1. 1/4 duty (1/3 bias) LCD Max 160 pixels (8 segments × 20 digits)
2. 1/3 duty (1/3 bias) LCD Max 120 pixels (8 segments × 15 digits)
3. 1/3 duty (1/2 bias) LCD Max 120 pixels (8 segments × 15 digits)
4. 1/2 duty (1/2 bias) LCD Max 80 pixels (8 segments × 10 digits)
5. Static LCD Max 40 pixels (8 segments × 5 digits)

17.1 Configuration

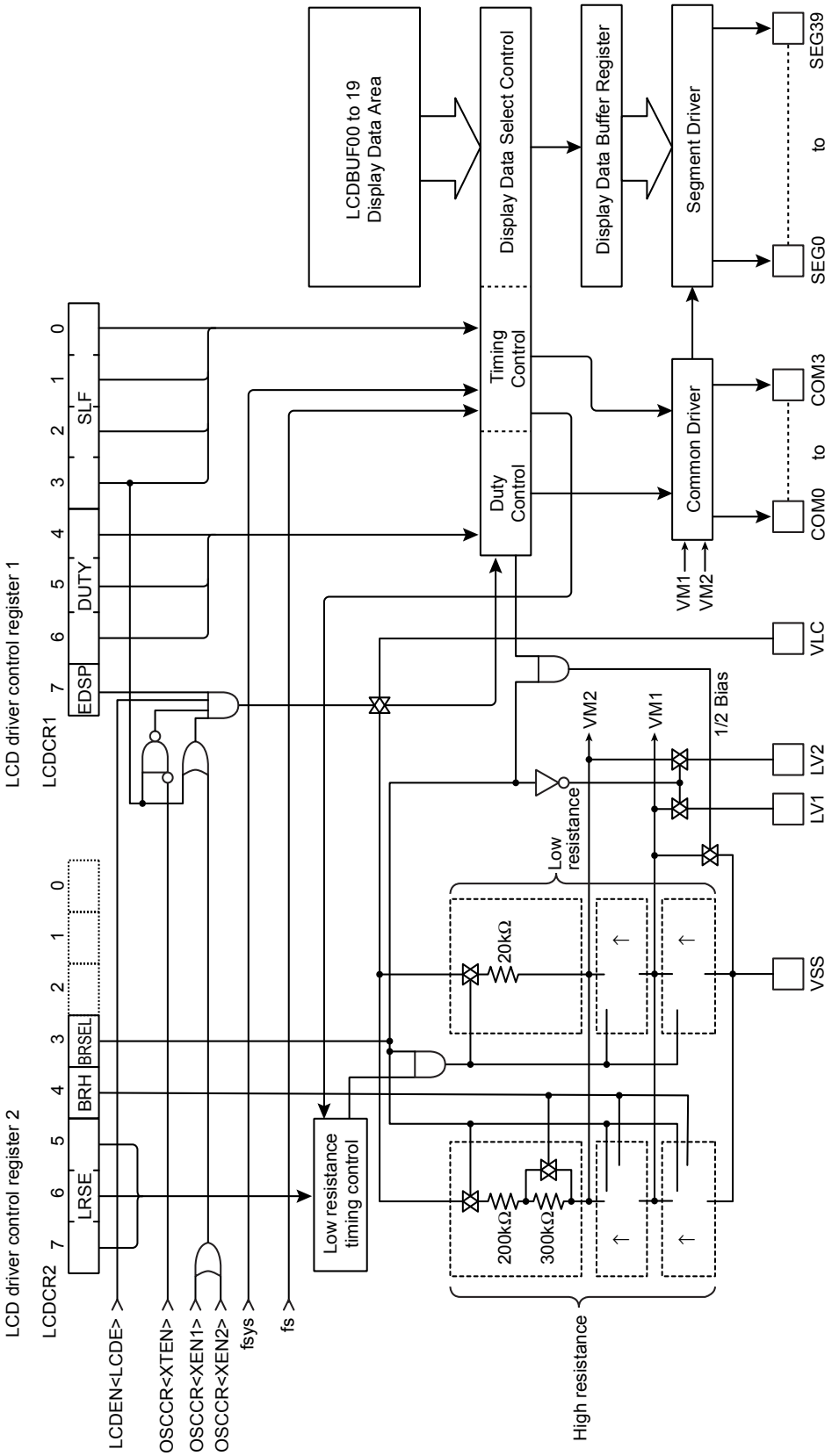


Figure 17-1 LCD Driver

17.2 Registers

17.2.1 Register List

The table below shows control registers and their addresses.

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

Register name		Address(Base+)
Enable register	LCDEN	0x0000
Control register 1	LCDCR1	0x0004
Control register 2	LCDCR2	0x0008
Buffer register 00	LCDBUF00	0x000C
Buffer register 01	LCDBUF01	0x0010
Buffer register 02	LCDBUF02	0x0014
Buffer register 03	LCDBUF03	0x0018
Buffer register 04	LCDBUF04	0x001C
Buffer register 05	LCDBUF05	0x0020
Buffer register 06	LCDBUF06	0x0024
Buffer register 07	LCDBUF07	0x0028
Buffer register 08	LCDBUF08	0x002C
Buffer register 09	LCDBUF09	0x0030
Buffer register 10	LCDBUF10	0x0034
Buffer register 11	LCDBUF11	0x0038
Buffer register 12	LCDBUF12	0x003C
Buffer register 13	LCDBUF13	0x0040
Buffer register 14	LCDBUF14	0x0044
Buffer register 15	LCDBUF15	0x0048
Buffer register 16	LCDBUF16	0x004C
Buffer register 17	LCDBUF17	0x0050
Buffer register 18	LCDBUF18	0x0054
Buffer register 19	LCDBUF19	0x0058

17.2.2 Details of Registers

17.2.2.1 LCDEN (Enable register)

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

Bit	Bit Symbol	Type	Function
31-1	−	R	Read as "0".
0	LCDE	R/W	LCD operation 0: Disable 1: Enable Specified LCD operagion. To use the LCD, set <SIOE> = "1". When the operation is disabled, no clock is supplied to the other registers in the LCD module. This can reduce the power consumption. If the LCD operation is executed and then disabled, the settings will be maintained in each register.

17.2.2.2 LCDCR1 (Control register 1)

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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	EDSP	R/W	LCD display control 0: Blank LCD display 1: Enable LCD display
6-4	DUTY	R/W	LCD drive method selection 000: 1/4 duty (1/3 bias) 001: 1/3 duty (1/3 bias) 010: 1/3 duty (1/2 bias) 011: 1/2 duty (1/2 bias) 100: Static 101: Reserved 110: Reserved 111: Reserved
3-0	SLF	R/W	Base frequency selection (Note) 0000: $fsys/2^{18}$ 0001: $fsys/2^{17}$ 0010: $fsys/2^{16}$ 0011: $fsys/2^{15}$ 0100: $fsys/2^{14}$ 0101: Reserved 0110: Reserved 0111: Reserved 1000: $fs/2^{9}$ 1001: $fs/2^{8}$ 1010 to 1111: Reserved

Note: In SLOW mode, do not set SLF to "0000" to "0100" (i.e. frequencies based on fsys). If SLF is set to one of these frequencies, pulses of an unexpected frame frequency will be output from the common and segment output pins.

17.2.2.3 LCDCR2 (Control register 2)

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7-5	LRSE	R/W	Low internal bleeder resistance connection time selection The time is selected as follows by setting of LCDCR1<SLF>.

	LCDCR1<SLF>						
	0000	0001	0010	0011	0100	1000	1001
000:	Not connected						
001:	2^{11}/fsys	2^{10}/fsys	2^{9}/fsys	2^{8}/fsys	2^{7}/fsys	2^{2}/fs	2/fs
010:	2^{12}/fsys	2^{11}/fsys	2^{10}/fsys	2^{9}/fsys	2^{8}/fsys	2^{3}/fs	2^{2}/fs
011:	2^{13}/fsys	2^{12}/fsys	2^{11}/fsys	2^{10}/fsys	2^{9}/fsys	2^{4}/fs	2^{2}/fs
100:	2^{14}/fsys	2^{13}/fsys	2^{12}/fsys	2^{11}/fsys	2^{10}/fsys	2^{5}/fs	2^{4}/fs
101:	2^{15}/fsys	2^{14}/fsys	2^{13}/fsys	2^{12}/fsys	2^{11}/fsys	2^{6}/fs	2^{5}/fs
110:	Always connected						
111:	Reserved						

4	BRH	R/W	High internal bleeder resistance selection 0: 200kΩ (Typ.) 1: 500kΩ (Typ.)
3	BRSEL	R/W	Internal/external bleeder resistance switching control 0: Use external bleeder resistance 1: Use internal bleeder resistance
2-0	-	R	Read as "0".

Note: The LRSE and BRH settings are effective only when BRSEL is set to select internal bleeder resistance.

17.2.2.4 LCDBUF00 (Buffer register 00)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	SEG1				SEG0			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7-4	SEG1	R/W	Data of SEG1 Specifies the COM3,COM2,COM1 and COM0 data of SEG1.
3-0	SEG0	R/W	Data of SEG0 Specifies the COM3,COM2,COM1 and COM0 data of SEG0.

17.2.2.5 LCDBUF01 (Buffer register 01)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	SEG3				SEG2			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7-4	SEG3	R/W	Data of SEG3 Specifies the COM3,COM2,COM1 and COM0 data of SEG3.
3-0	SEG2	R/W	Data of SEG2 Specifies the COM3,COM2,COM1 and COM0 data of SEG2.

17.2.2.6 LCDBUF02 (Buffer register 02)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	SEG5				SEG4			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7-4	SEG5	R/W	Data of SEG5 Specifies the COM3,COM2,COM1 and COM0 data of SEG5.
3-0	SEG4	R/W	Data of SEG4 Specifies the COM3,COM2,COM1 and COM0 data of SEG4.

17.2.2.7 LCDBUF03 (Buffer register 03)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	SEG7				SEG6			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7-4	SEG7	R/W	Data of SEG7 Specifies the COM3,COM2,COM1 and COM0 data of SEG7.
3-0	SEG6	R/W	Data of SEG6 Specifies the COM3,COM2,COM1 and COM0 data of SEG6.

17.2.2.8 LCDBUF04 (Buffer register 04)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	SEG9				SEG8			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7-4	SEG9	R/W	Data of SEG9 Specifies the COM3,COM2,COM1 and COM0 data of SEG9.
3-0	SEG8	R/W	Data of SEG8 Specifies the COM3,COM2,COM1 and COM0 data of SEG8.

17.2.2.9 LCDBUF05 (Buffer register 05)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	SEG11				SEG10			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7-4	SEG11	R/W	Data of SEG11 Specifies the COM3,COM2,COM1 and COM0 data of SEG11.
3-0	SEG10	R/W	Data of SEG10 Specifies the COM3,COM2,COM1 and COM0 data of SEG10.

17.2.2.10 LCDBUF06 (Buffer register 06)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	SEG13				SEG12			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7-4	SEG13	R/W	Data of SEG13 Specifies the COM3,COM2,COM1 and COM0 data of SEG13.
3-0	SEG12	R/W	Data of SEG12 Specifies the COM3,COM2,COM1 and COM0 data of SEG12.

17.2.2.11 LCDBUF07 (Buffer register 07)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	SEG15				SEG14			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7-4	SEG15	R/W	Data of SEG15 Specifies the COM3,COM2,COM1 and COM0 data of SEG15.
3-0	SEG14	R/W	Data of SEG14 Specifies the COM3,COM2,COM1 and COM0 data of SEG14.

17.2.2.12 LCDBUF08 (Buffer register 08)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	SEG17				SEG16			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7-4	SEG17	R/W	Data of SEG17 Specifies the COM3,COM2,COM1 and COM0 data of SEG17.
3-0	SEG16	R/W	Data of SEG16 Specifies the COM3,COM2,COM1 and COM0 data of SEG16.

17.2.2.13 LCDBUF09 (Buffer register 09)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	SEG19				SEG18			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7-4	SEG19	R/W	Data of SEG19 Specifies the COM3,COM2,COM1 and COM0 data of SEG19.
3-0	SEG18	R/W	Data of SEG18 Specifies the COM3,COM2,COM1 and COM0 data of SEG18.

17.2.2.14 LCDBUF10 (Buffer register 10)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	SEG21				SEG20			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7-4	SEG21	R/W	Data of SEG21 Specifies the COM3,COM2,COM1 and COM0 data of SEG21.
3-0	SEG20	R/W	Data of SEG20 Specifies the COM3,COM2,COM1 and COM0 data of SEG20.

17.2.2.15 LCDBUF11 (Buffer register 11)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	SEG23				SEG22			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7-4	SEG23	R/W	Data of SEG23 Specifies the COM3,COM2,COM1 and COM0 data of SEG23.
3-0	SEG22	R/W	Data of SEG22 Specifies the COM3,COM2,COM1 and COM0 data of SEG22.

17.2.2.16 LCDBUF12 (Buffer register 12)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	SEG25				SEG24			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7-4	SEG25	R/W	Data of SEG25 Specifies the COM3,COM2,COM1 and COM0 data of SEG25.
3-0	SEG24	R/W	Data of SEG24 Specifies the COM3,COM2,COM1 and COM0 data of SEG24.

17.2.2.17 LCDBUF13 (Buffer register 13)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	SEG27				SEG26			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7-4	SEG27	R/W	Data of SEG27 Specifies the COM3,COM2,COM1 and COM0 data of SEG27.
3-0	SEG26	R/W	Data of SEG26 Specifies the COM3,COM2,COM1 and COM0 data of SEG26.

17.2.2.18 LCDBUF14 (Buffer register 14)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	SEG29				SEG28			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7-4	SEG29	R/W	Data of SEG29 Specifies the COM3,COM2,COM1 and COM0 data of SEG29.
3-0	SEG28	R/W	Data of SEG28 Specifies the COM3,COM2,COM1 and COM0 data of SEG28.

17.2.2.19 LCDBUF15 (Buffer register 15)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	SEG31				SEG30			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7-4	SEG31	R/W	Data of SEG31 Specifies the COM3,COM2,COM1 and COM0 data of SEG31.
3-0	SEG30	R/W	Data of SEG30 Specifies the COM3,COM2,COM1 and COM0 data of SEG30.

17.2.2.20 LCDBUF16 (Buffer register 16)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	SEG33				SEG32			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7-4	SEG33	R/W	Data of SEG33 Specifies the COM3,COM2,COM1 and COM0 data of SEG33.
3-0	SEG32	R/W	Data of SEG32 Specifies the COM3,COM2,COM1 and COM0 data of SEG32.

17.2.2.21 LCDBUF17 (Buffer register 17)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	SEG35				SEG34			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7-4	SEG35	R/W	Data of SEG35 Specifies the COM3,COM2,COM1 and COM0 data of SEG35.
3-0	SEG34	R/W	Data of SEG34 Specifies the COM3,COM2,COM1 and COM0 data of SEG34.

17.2.2.22 LCDBUF18 (Buffer register 18)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	SEG37				SEG36			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7-4	SEG37	R/W	Data of SEG37 Specifies the COM3,COM2,COM1 and COM0 data of SEG37.
3-0	SEG36	R/W	Data of SEG36 Specifies the COM3,COM2,COM1 and COM0 data of SEG36.

17.2.2.23 LCDBUF19 (Buffer register 19)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	SEG39				SEG38			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7-4	SEG39	R/W	Data of SEG39 Specifies the COM3,COM2,COM1 and COM0 data of SEG39.
3-0	SEG38	R/W	Data of SEG38 Specifies the COM3,COM2,COM1 and COM0 data of SEG38.

17.3 Functions

17.3.1 LCD Display Control

If LCDCR1<EDSP> is set to "1", the power switch of LCD driver is on and a VLC voltage will be applied to a LCD driver. As a result, the LCD driver display will be enabled. If <EDSP> is set to "0", the power switch of LCD driver is off and a VLC voltage is shutdown. This causes the driver to display blank.

The following tables describes the condition of the LCD connection pins when <EDSP> is "1".

1. Common output pin

   Common output pin is only controlled with LCDCR1<EDSP>.

LCDCR1 <EDSP>	Common output pins COM0,COM1,COM2,COM3
0	"Low" level
1	Common output

2. External bleeder resistor pin

   External bleeder resistor pin is a dual-purpose pin with a general-purpose input/output port. When this pin is used for a external bleeder resistor pin, set "0" to LCDCR2<BRSEL>. A setting of port control register is not required.

LCDCR2 <BRSEL>	External bleeder resistor pins LV1,LV2
1	Input/output port
0	External bleeder resistor

3. Segment output pin

   A segment output pin is a dual-purpose pin with a general-purpose input/output port. To function as a segment output, set "1" to the corresponding function register PxFRn. Settings of other port control registers are not required except PxFRn.

PxFRn	PxCR	External bleeder resistor pins LV1,LV2
0	0	General-purpose port
0	1	
1	0	Segment output
1	1	-

Note) In register names, "x" indicates a port number and "n" indicates a function register number.

17.3.2 Operation at Reset

When a reset occurs, LCDCR1<EDSP> is initialized to "0" and the power switch of the LCD driver is automatically turned off, shutting off VLC voltage. At this time, the common output pins are fixed to the "Low" level. The multiplexed pins (input/output port or segment output) are configured as port input pins (high impedance). Therefore, if external reset operation takes time, the LCD display may become blurred.

17.3.3 Operation in SLEEP/STOP mode

If a transition to SLEEP/STOP mode occurs while LCDCR1<EDSP> is "1", the following conditions will occur.

Note: To use a LCD in the SLEEP mode, perform a transition to SLEEP mode in the SLOW mode. If a transition from the NORMAL to SLEEP mode occurs, a display will be blank.

- SLEEP mode

  A common pin and segment pin remain the same state as before a transition occurs and a display is being used.
- STOP mode

  If a transition to the STOP mode occurs, LCDCR1 <EDSP> will automatically initialized to "0", a display will be blank. To redisplay the LCD after returning from the STOP mode, set "1" to LCDCR1<EDSP>.

17.3.4 Operation in SLOW mode

When the LCD is used in both NORMAL mode and SLOW mode, it is recommended that LCDCR1<SLF> be set to a frequency based on fs ("1000" or "1001"). (This will eliminate the need for changing the LCDCR1<SLF> setting each time the operating mode is switched between NORMAL mode and SLOW mode.)

If a frequency based on fsys is used in NORMAL mode, it is necessary to clear LCDCR1<EDSP> to "0" before switching to SLOW mode. Then, after entering SLOW mode, it is necessary to change LCDCR1<SLF> to a frequency based on fs and to set LCDCR1<EDSP> to "1". Likewise, in switching from SLOW mode to NORMAL mode, it is necessary to clear LCDCR1<EDSP> to "0" before switching to NORMAL mode. Then, after entering NORMAL mode, it is necessary to change LCDCR1<SLF> to a frequency based on fsys and to set LCDCR1<EDSP> to "1".

17.3.5 Fail-safe

Be careful about the following because LCD cannot display.

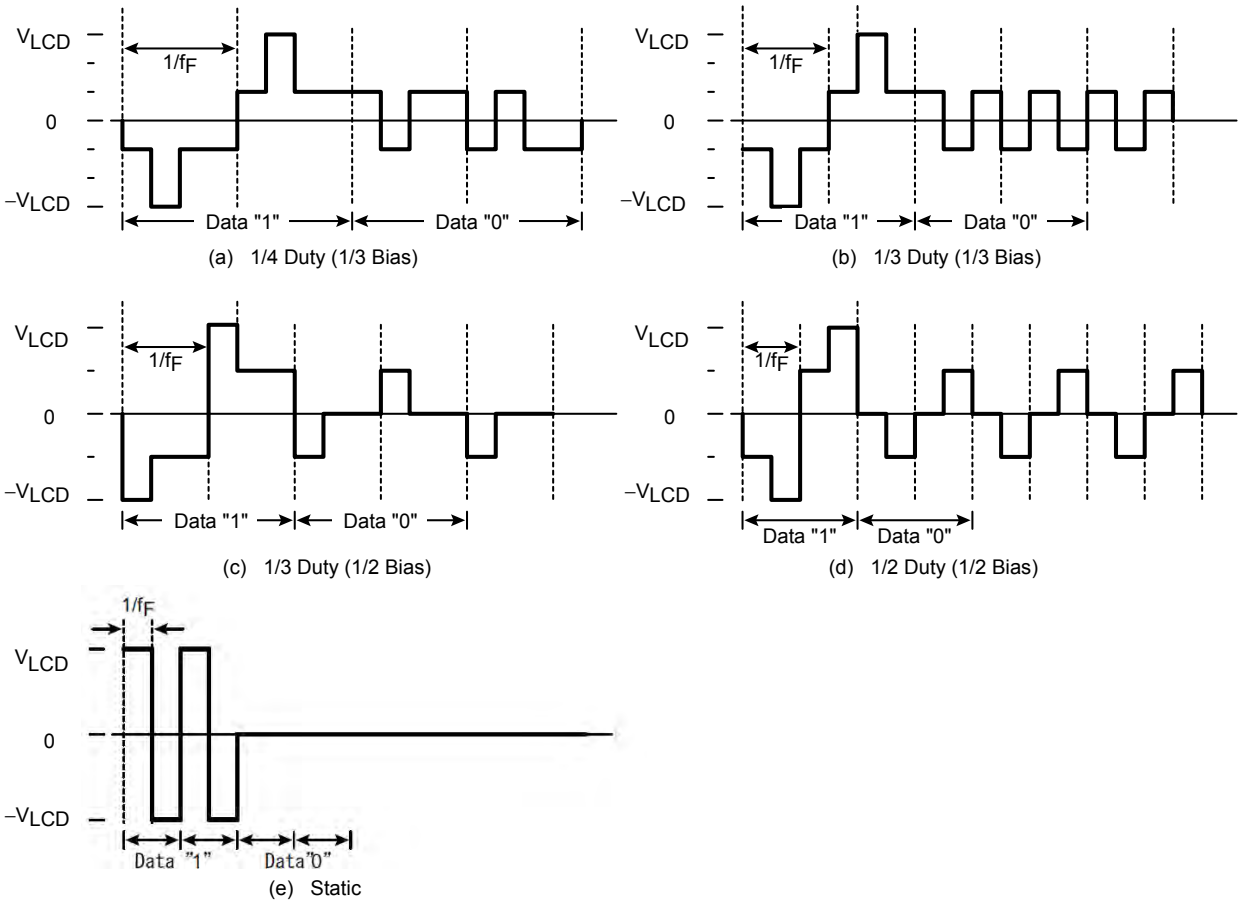
1. When LCDCR1<EDSP>="1", setting LCDEN<LCDE> to "0" blanks the LCD display. The display can be re-enabled by setting LCDEN<LCDE> to "1".

2. When LCDCR1<SLF> is set to "0000" to "0110", the high-speed clock must be activated (OSCCR <XEN1> = "1" or OSCCR <XEN> = "1") and allowed to achieve stable oscillation before LCDCR1 <EDSP> can be set to "1". If LCDCR1<EDSP> is set to "1" while the high-speed clock is stopped, the LCD display cannot be enabled. (Although LCDCR1<EDSP> changes to "1", the LCD display remains blank.)

3. When LCDCR1<SLF> is set to "1000" or "1001", the low-speed clock must be activated (OSCCR <XTEN> = "1") and allowed to achieve stable oscillation before LCDCR1<EDSP> can be set to "1". If LCDCR1<EDSP> is set to "1" while the low-speed clock is stopped, the LCD display cannot be enabled. (Although LCDCR1<EDSP> changes to "1", the LCD display remains blank.)

17.3.6 LCD Drive Methods (LCDCR1<DUTY>)

The LCD drive method can be selected from the following five types by the setting of LCDCR1<DUTY>.



Note 1: f_F = Frame frequency

Note 2: V_{LCD3} = LCD drive voltage (= V_{LC} − V_{SS})

Figure 17-2 LCD Drive Waveforms (Potential Difference between COM and SEG Pins)

17.3.7 Frame Frequency (LCDCR1<SLF>)

The frame frequency (f_F) is determined based on the drive method and base frequency, as shown in Table 17-1. The base frequency is selected by LCDCR1<SLF>.

Table 17-1 Frame Frequency Settings

SLF	Base frequency [Hz]	Frame frequency [Hz]			
		1/4 Duty	1/3 Duty	1/2 Duty	Static
0000	fsys / 2^{18}	fsys / 2^{18}	(4 / 3) x fsys / 2^{18}	(4 / 2) x fsys / 2^{18}	fsys / 2^{18}
	(fsys = 16 MHz)	61	81	122	61
0001	fsys / 2^{17}	fsys / 2^{17}	(4 / 3) x fsys / 2^{17}	(4 / 2) x fsys / 2^{17}	fsys / 2^{17}
	(fsys = 16 MHz)	122	163	244	122
	(fsys= 8 MHz)	61	81	122	61
0010	fsys / 2^{16}	fsys / 2^{16}	(4 / 3) x fsys / 2^{16}	(4 / 2) x fsys / 2^{16}	fsys / 2^{16}
	(fsys = 8 MHz)	122	163	244	122
	(fsys = 4 MHz)	61	81	122	61
0011	fsys / 2^{15}	fsys / 2^{15}	(4 / 3) x fsys / 2^{15}	(4 / 2) x fsys / 2^{15}	fsys / 2^{15}
	(fsys = 4 MHz)	122	163	244	122
	(fsys = 2 MHz)	61	81	122	61
0100	fsys / 2^{14}	fsys / 2^{14}	(4 / 3) x fsys / 2^{14}	(4 / 2) x fsys / 2^{14}	fsys / 2^{14}
	(fsys = 2 MHz)	122	163	244	122
	(fsys = 1 MHz)	61	81	122	61
1000	fs / 2^{9}	fs / 2^{9}	(4 / 3) x fs / 2^{9}	(4 / 2) x fs / 2^{9}	fs / 2^{9}
	(fs = 32.768 kHz)	64	85	128	64
1001	fs / 2^{8}	fs / 2^{8}	(4 / 3) x fs / 2^{8}	(4 / 2) x fs / 2^{8}	fs / 2^{8}
	(fs = 32.768 kHz)	128	171	256	128

Note:fsys = Gear clock frequency [Hz], fs = Low-frequency clock frequency [Hz]

17.3.8 Internal/External Bleeder Resistance Switching Control

The LCD bias voltage is generated by bleeder resistance. Either external or internal bleeder resistance can be used.

To use internal bleeder resistance, set LCDCR2<BRSEL> to "1". In this case, the multiplexed pins (input/output port or external bleeder resistance connection) can be used as input/output ports.

To use external bleeder resistance, set LCDCR2<BRSEL> to "0" and connect external resistance to the external bleeder resistance connection pins (LV1, LV2). In this case, the multiplexed pins (input/output port, external bleeder resistance connection, or segment output) can only be used as external bleeder resistance connection pins.

See Figure 17-4 for how to connect bleeder resistance.

17.3.9 Low Internal Bleeder Resistance Connection Time Selection (LCDCR2<LRSE>)

Internal bleeder resistance is comprised of two parts: high resistance and low resistance. The high and low resistance parts are connected in parallel for each bias voltage. The low bleeder resistance is provided with an analog switch, and the time to turn on the low bleeder resistance can be adjusted by LCDCR2<LRSE>. While the analog switch is turned on, the low resistance is connected in parallel to the high resistance. This reduces the total amount of resistance, allowing the drive capability of the LCD driver to be increased.

Typically, the longer the period of connecting the low resistance, the higher the drive capability of the LCD panel, but the higher the power consumption. Conversely, the shorter the period of connecting the low resistance, the lower the drive capability, but the lower the power consumption. Insufficient drive capability will cause adverse effects on the LCD display, such as blurring. Choose the optimum drive capability for the LCD panel to be used.

Table 17-2 shows the connection time (percentage) of the low bleeder resistance per frame and the estimated amount of current that flows through the entire bleeder resistance in each case.

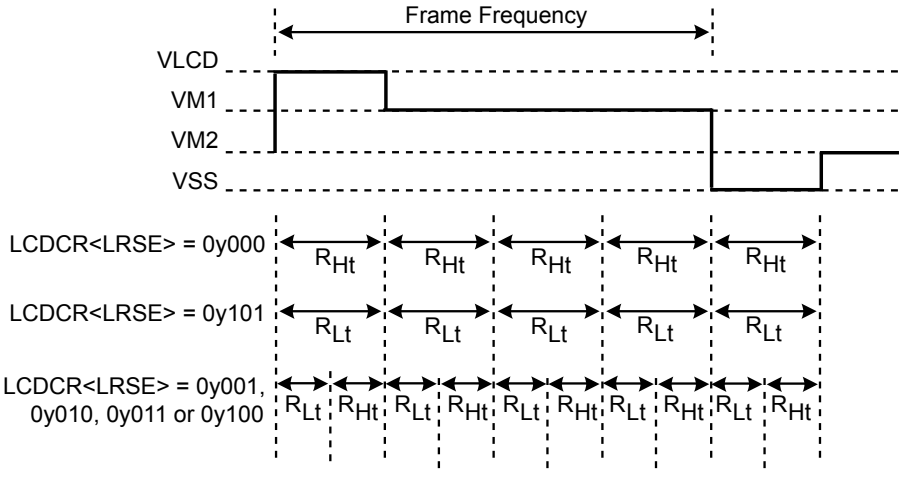
Figure 17-3 shows the bleeder resistance control timing for the 1/4 duty and 1/3 bias LCD.

See "17.3.10 High Internal Bleeder Resistance Selection (LCDCR2<BRH>)"for the setting of LCDCR2<BRH>.

Table 17-2 Low Bleeder Resistance Connection Time (%) and Total Bleeder Current Values (Estimated)

LCDCR2 <LRSE>		1/4 Duty (1/3 Bias)		1/3 Duty (1/3 Bias)		1/3 Duty (1/2 Bias)		1/2 Duty (1/2 Bias)	
		BRH="1"	BRH="0"	BRH="1"	BRH="0"	BRH="1"	BRH="0"	BRH="1"	BRH="0"
000	Low resistance connection time (%)	0% (Always high resistance)							
	Bleeder current	2.00μA	5.00μA	2.00μA	5.00μA	3.00μA	7.50μA	3.00μA	7.50μA
001	Low resistance connection time (%)	3.13%		2.34%		2.34%		1.56%	
	Bleeder current	3.56μA	6.56μA	3.17μA	6.17μA	4.76μA	9.26μA	4.17μA	8.67μA
010	Low resistance connection time (%)	6.25%		4.69%		4.69%		3.13%	
	Bleeder current	5.13μA	8.13μA	4.34μA	7.34μA	6.52μA	11.02μA	5.34μA	9.84μA
011	Low resistance connection time (%)	12.5%		9.38%		9.38%		6.25%	
	Bleeder current	8.25μA	11.25μA	6.69μA	9.69μA	10.03μA	14.53μA	7.69μA	12.19μA
100	Low resistance connection time (%)	25%		18.75%		18.75%		12.5%	
	Bleeder current	14.50μA	17.50μA	11.38μA	14.38μA	17.06μA	21.56μA	12.38μA	16.88μA
101	Low resistance connection time (%)	50%		37.5%		37.5%		25%	
	Bleeder current	27.00μA	30.00μA	20.75μA	23.75μA	31.13μA	35.63μA	21.75μA	26.25μA
110	Low resistance connection time (%)	100% (Always connected)							
	Bleeder current	52.00μA	55.00μA	52.00μA	55.00μA	78.00μA	82.50μA	78.00μA	82.50μA

Note: The bleeder resistance current values shown above are estimated values. The actual current values may vary depending on the amount of LCD load and manufacturing variations in resistance values.



Low Bleeder Resistance Connection Time

R_{Lt} : Period during which low resistance is connected
(high resistance and low resistance are connected in parallel)

R_{Ht} : Period during which low resistance is not connected
(only high resistance is connected)

Figure 17-3 Bleeder Resistance Selection by LCDCR2<LRSE> (1/4 Duty, 1/3 Bias)

17.3.10 High Internal Bleeder Resistance Selection (LCDCR2<BRH>)

The value of high internal bleeder resistance can be selected from two levels (500 kΩ (Typ.) or 200 kΩ (Typ.)) by the setting of LCDCR2<BRH>. Typically, the lower the resistance value, the higher the drive capability of the LCD panel, but the higher the power consumption. Conversely, the higher the resistance value, the lower the drive capability, but the lower the pz ower consumption.

The resistance value of the low resistance is fixed to 20 kΩ (Typ.). Since the low resistance is connected in parallel to the high resistance via an analog switch, the total amount of resistance can be adjusted by the setting of LCDCR2<LRSE> as shown in Table 17-3.

For example, setting LCDCR2<BRH> to "1" selects a synthesized resistance of 19.23 kΩ (Typ.) when the low resistance is connected, and a high resistance of 500 kΩ (Typ.) when the low resistance is not connected.

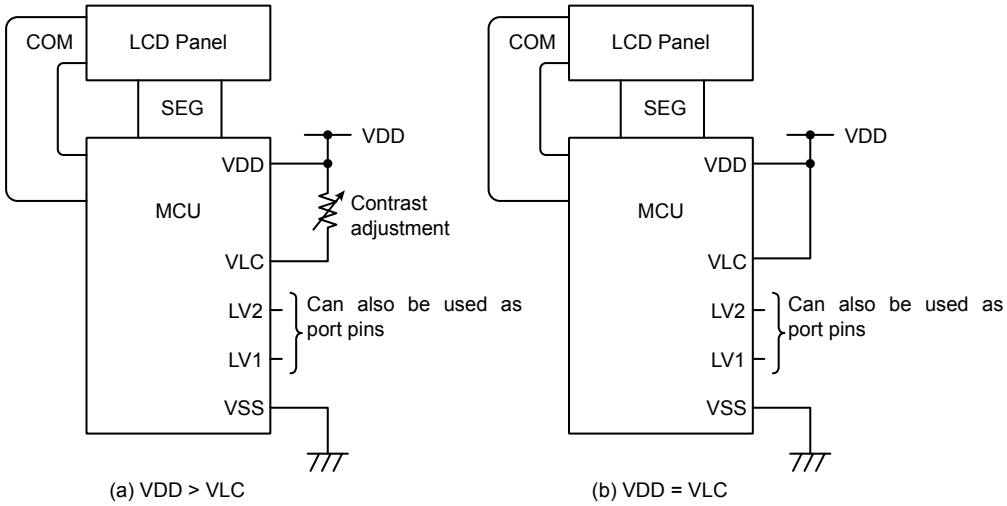
Table 17-3 Bleeder Resistance Values

LCDCR2<BRH>	When low resistance is not connected	When low resistance is connected
1	500 kΩ (Typ.)	19.23 kΩ (Typ.)
0	200 kΩ (Typ.)	18.18 kΩ (Typ.)

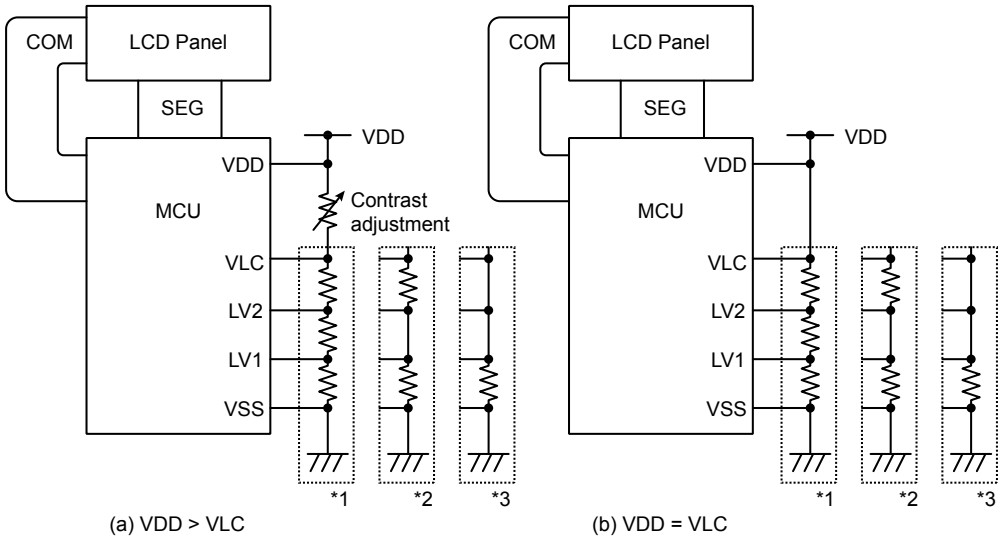
17.3.11 LCD Display Operation

The LCD drive voltage V_{LCD} is given by the potential difference between the VLC and VSS pins ($V_{LC} - V_{SS}$). The LCD lights up when the potential difference between segment and common outputs is $\pm VL_{CD}$. At other times, the LCD is turned off.

Power supply connections should be made to satisfy the condition VLC ≤ VDD. Connection examples are shown in Figure 17-4.



When internal bleeder resistance is used (LCDCR2<BRSEL>="1")



When external bleeder resistance is used (LCDCR2<BRSEL>="0")

*1 : 1/3 bias
*2 : 1/2 bias
*3 : Static

Figure 17-4 Connection Examples

Note 1: When the CPU operating voltage is the same as the LCD drive voltage, the VLC pin should be connected to the VDD pin.

Note 2: At reset, the common output pins become low. However, the multiplexed pins (input/output port or segment output) are configured as port input pins (high impedance). Therefore, if the multiplexed pins (input/output port or segment output) are used as segment output pins and an external reset signal is input for a prolonged period of time, the LCD display may be adversely affected, such as blurring. The multiplexed pins (input/output port or external bleeder resistance connection) are configured as external bleeder resistance connection pins.

17.3.12 Display Data Setting

Display data is stored in the Buffer register 00 to 19.

The display data stored in the display data area is automatically read out and sent to the LCD driver by hardware. The LCD driver generates the segment and common signals according to the display data and drive method. Therefore, display patterns can be changed by simply overwriting the contents of the display data area.Table 17-5 shows the correspondence between the display data area and the SEG and COM pins.

The LCD lights up when display data is "1" and is turned off when display data is "0".

At reset, the data in the Buffer registers are initialized to "0".

Since the number of pixels that can be driven varies with the LCD drive method, the number of bits used for storing display data also varies accordingly. Therefore, the bits not used for storing display data and the data memory locations corresponding to addresses not connected to the LCD can be used for storing general user data (see Table 17-4).

Table 17-4 Bits To Be Used for Storing Display Data

Drive method	Bits 7/3	Bits 6/2	Bits 5/1	Bits 4/0
1/4 duty	COM3	COM2	COM1	COM0
1/3 duty	-	COM2	COM1	COM0
1/2 duty	-	-	COM1	COM0
Static	-	-	-	COM0

Note:" - " denotes bits not used for storing display data.

Table 17-5 LCD Display Data Area

Registar name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Read/ Write	Initial value
LCDBUF00	SEG1				SEG0				R/W	(0000 0000)
LCDBUF01	SEG3				SEG2				R/W	(0000 0000)
LCDBUF02	SEG5				SEG4				R/W	(0000 0000)
LCDBUF03	SEG7				SEG6				R/W	(0000 0000)
LCDBUF04	SEG9				SEG8				R/W	(0000 0000)
LCDBUF05	SEG11				SEG10				R/W	(0000 0000)
LCDBUF06	SEG13				SEG12				R/W	(0000 0000)
LCDBUF07	SEG15				SEG14				R/W	(0000 0000)
LCDBUF08	SEG17				SEG16				R/W	(0000 0000)
LCDBUF09	SEG19				SEG18				R/W	(0000 0000)
LCDBUF10	SEG21				SEG20				R/W	(0000 0000)
LCDBUF11	SEG23				SEG22				R/W	(0000 0000)
LCDBUF12	SEG25				SEG24				R/W	(0000 0000)
LCDBUF13	SEG27				SEG26				R/W	(0000 0000)
LCDBUF14	SEG29				SEG28				R/W	(0000 0000)
LCDBUF15	SEG31				SEG30				R/W	(0000 0000)
LCDBUF16	SEG33				SEG32				R/W	(0000 0000)
LCDBUF17	SEG35				SEG34				R/W	(0000 0000)
LCDBUF18	SEG37				SEG36				R/W	(0000 0000)
LCDBUF19	SEG39				SEG38				R/W	(0000 0000)
	COM3	COM2	COM1	COM0	COM3	COM2	COM1	COM0		

17.4 Examples of How To Control the LCD Driver

17.4.1 Initialization

Figure 17-5 is a flowchart showing the initialization process of the LCD driver.

Example: When the LCD driver is to be operated with the following conditions:
- Drive method: 1/4 duty, 1/3 bias
- LCD frame frequency: fsys/2^{18} [Hz]
- Connection time of low bleeder resistance (internal): 2^{15}/fsys
- High bleeder resistance (internal): 200 kΩ

```
LCDEN              ← 0x01      ; <LCDE> = "1"
LCDCR1             ← 0x00      ; set the LCD drive method and base frequency
LCDCR2             ← 0x28      ; set the connection time of low bleeder resistance and the
                               ; high bleeder resistance value
PxFRn<PxmFn>       ← 0x1       ; set the PxFRn register
                               (x: I/O port number, n; function register number, m; bit number)
:                    :
:                    :         ; set initial display data
LCDCR1             ← 0x80      ; enable the LCD display
```

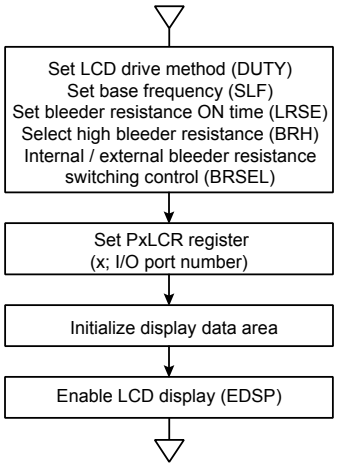


Figure 17-5 Flowchart for the LCD Driver Initialization

17.4.2 Display Data Setting

Display data is normally prepared as fixed data in the program memory (ROM) and transferred to the display data area by instructions.

Example1: The following shows an example of how to set display data for displaying a number corresponding to BCD data stored at address 0x90 in the data memory by using the 1/4 duty and 1/3 bias LCD. Figure 17-6 shows an example of how the COM and SEG pins are connected to the LCD, and Table 17-6 shows how display data is set for this example.

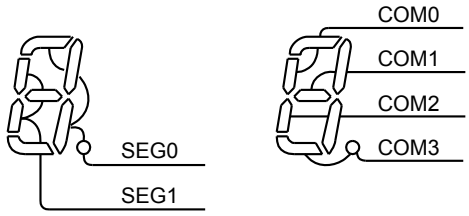


Figure 17-6 Example of COM and SEG Pin Connections (1/4 Duty)

Table 17-6 Example of Display Data (1/4 Duty)

No.	Display	Display data	No.	Display	Display data
0	0.	11011111	5	5	10110101
1	1	00000110	6	6	11110101
2	2	11100011	7	7	00000111
3	3	10100111	8	8	11110111
4	4	00110110	9	9	10110111

Example2: The following shows an example of how to set display data for displaying a number as explained in example 1 by using the 1/2 duty LCD. Figure 17-7 shows an example of how the SEG and COM pins are connected to the LCD, and Table 17-7 shows how display data is set for this example.

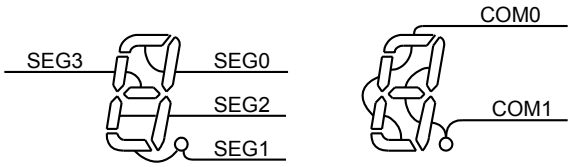


Figure 17-7 Example of COM and SEG Pin Connections

Table 17-7 Example of Display Data (1/2 Duty)

Number	Display data		Number	Display data	
	High-order address	Low-order address		High-order address	Low-order address
0	**01**11	**01**11	5	**11**10	**01**01
1	**00**10	**00**10	6	**11**11	**01**01
2	**10**01	**01**11	7	**01**10	**00**11
3	**10**10	**01**11	8	**11**11	**01**11
4	**11**10	**00**10	9	**11**10	**01**11

Note: An asterisk (*) denotes "don't care".

17.4.3 Drive Output Examples

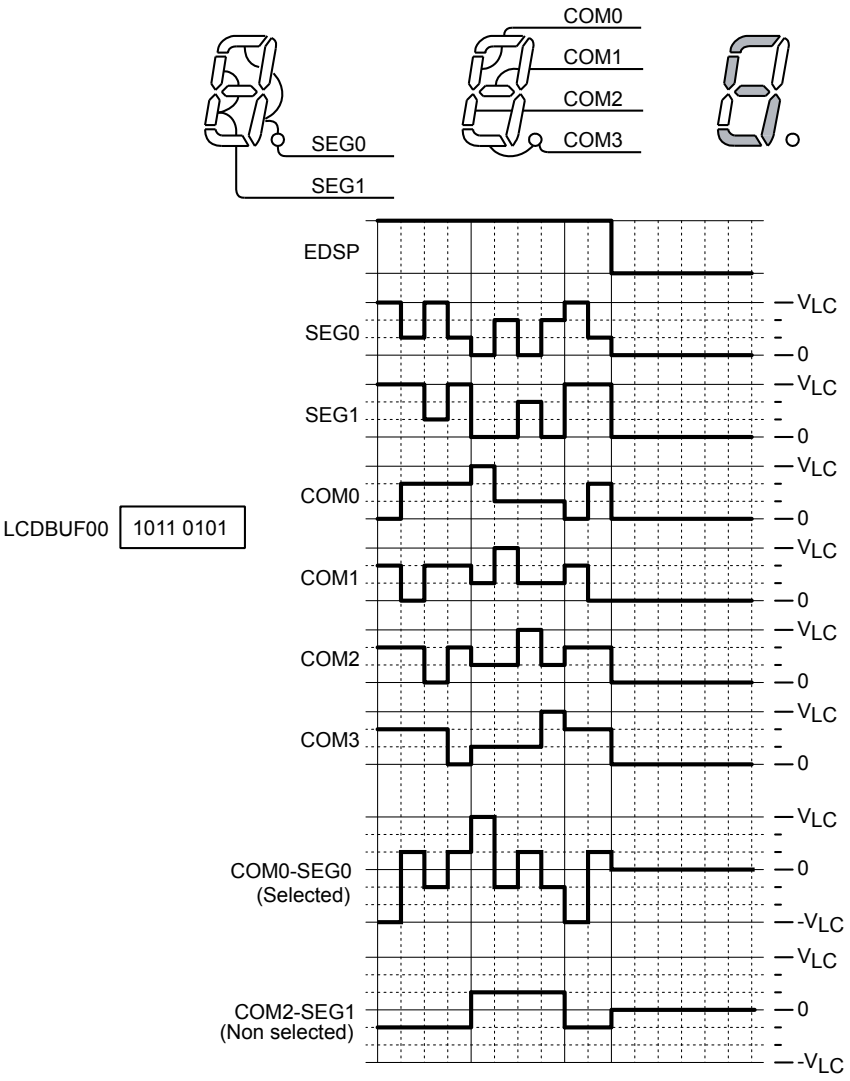


Figure 17-8 1/4 Duty (1/3 Bias) Drive

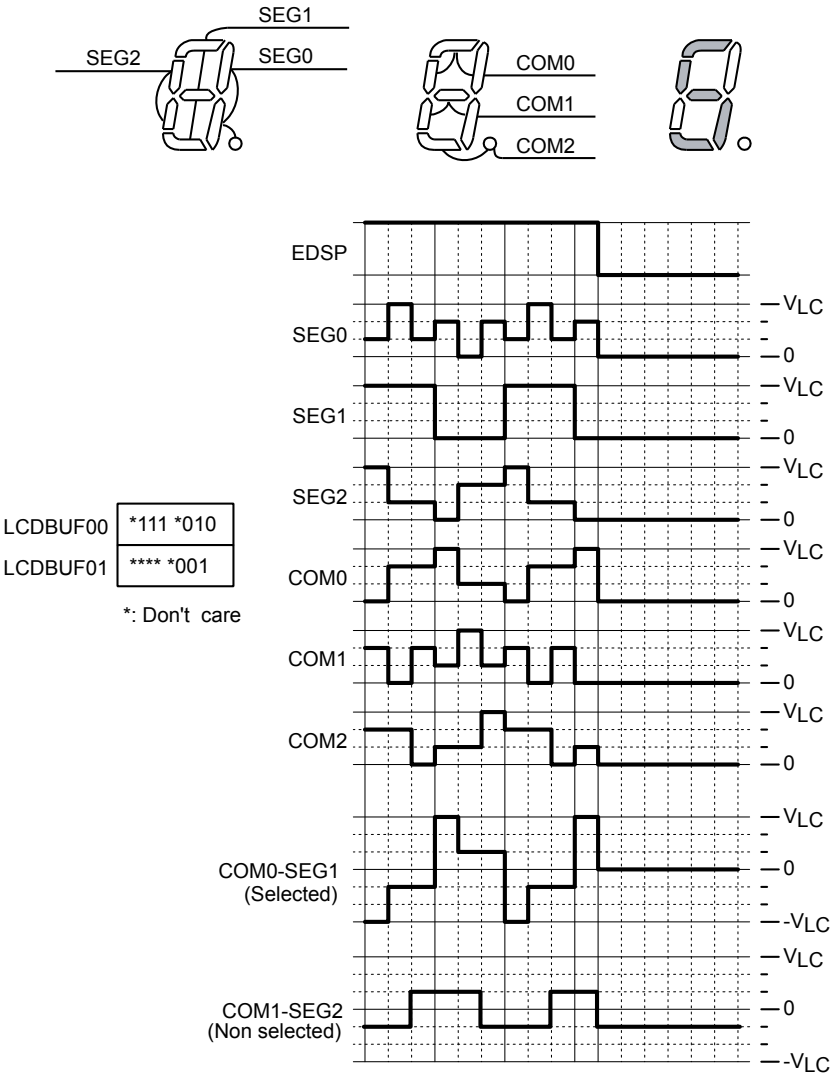


Figure 17-9 1/3 Duty (1/3 Bias) Drive

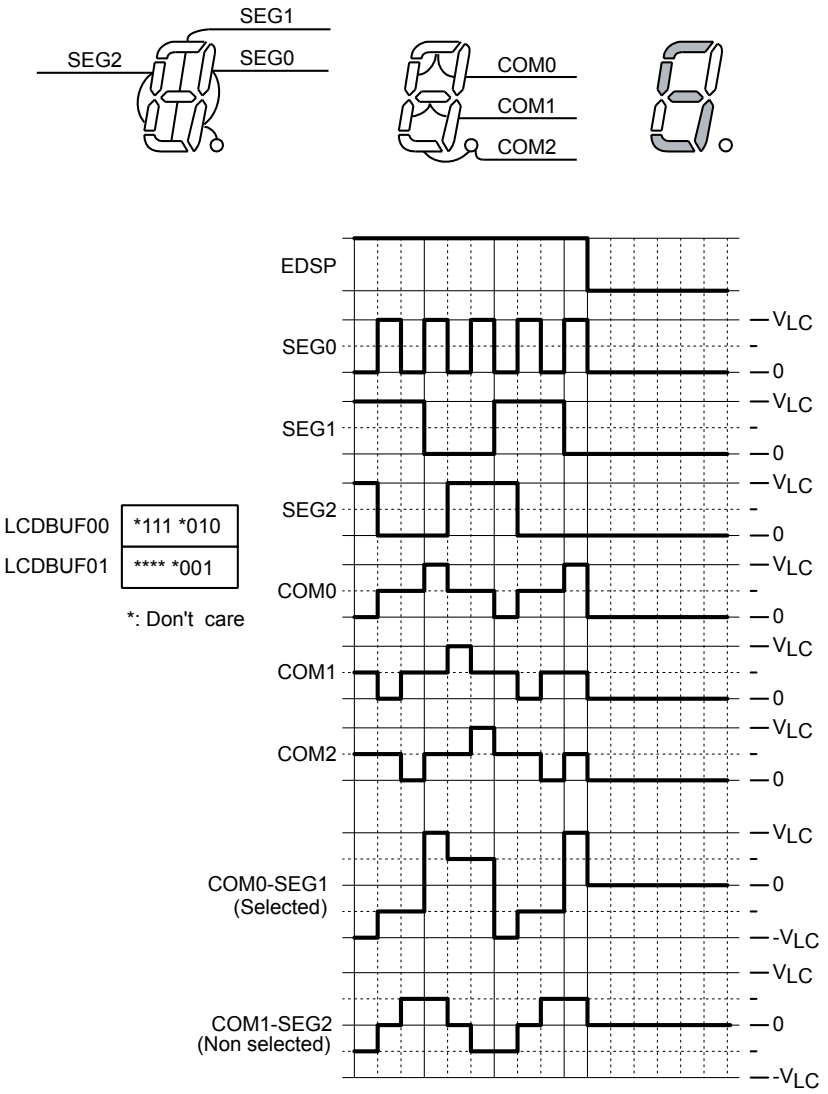


Figure 17-10 1/3 Duty (1/2 Bias) Drive

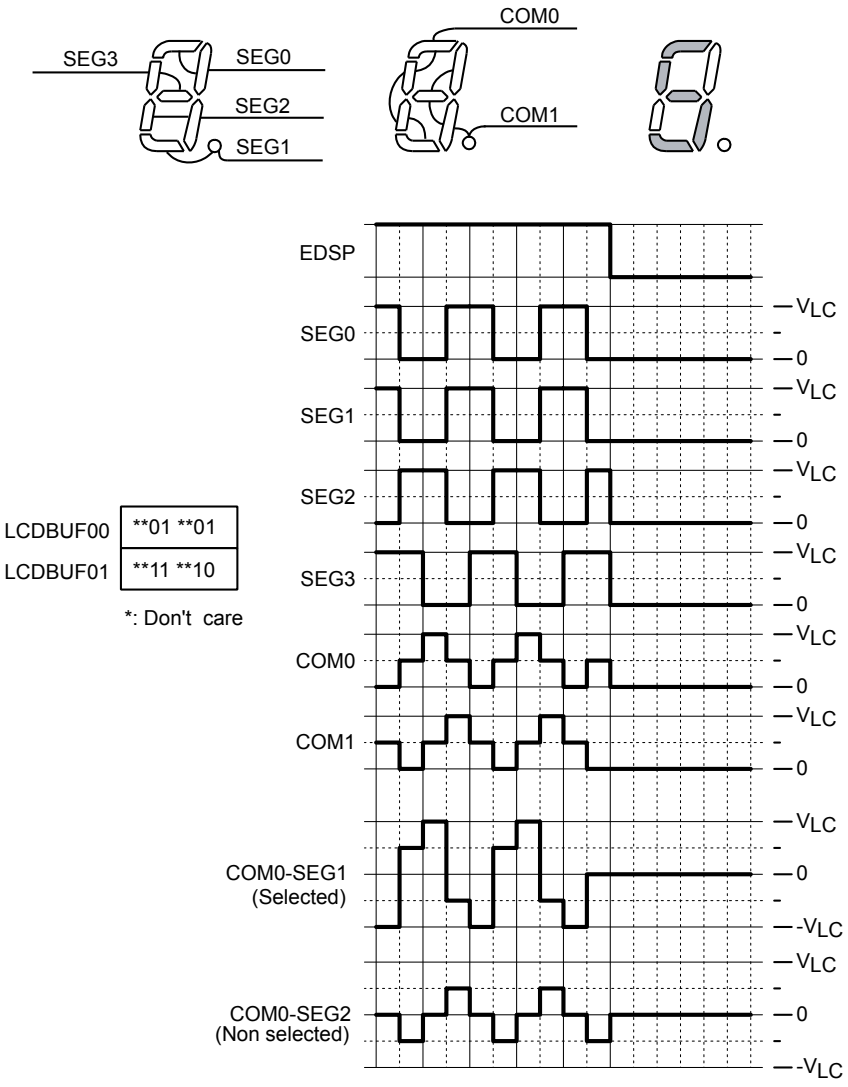


Figure 17-11 1/2 Duty (1/2 Bias) Drive

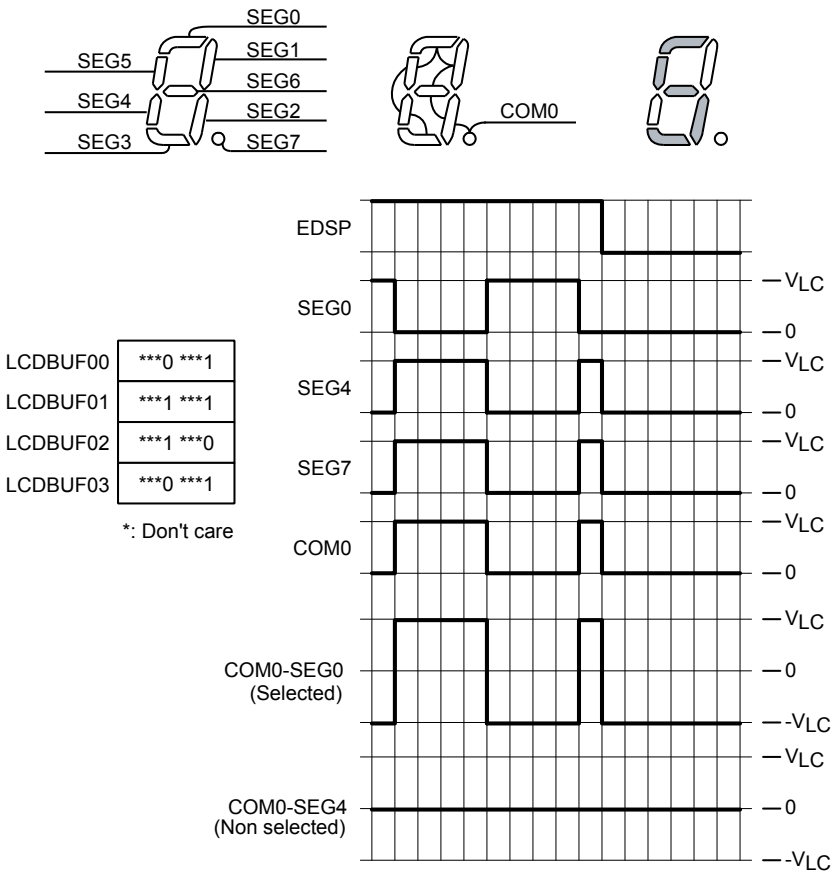


Figure 17-12 Static Drive

18. Low Voltage Detection Circuit (LVD)

Low voltage detection circuit generates an interrupt signal by detecting a decreasing/increasing voltage.

Supply voltage is indicated as RVDD3.

Note: Due to the fluctuation of supply voltage, the power-on reset circuit may not operate properly. Users should give due consideration based on the electrical characteristic in the device designing.

18.1 Structure

The low voltage detection circuit consists of a reference voltage generation circuit, comparators and control registers.

Supply voltage is divided by a ladder resistor and input to the voltage selection circuit. In the voltage selection circuit, a voltage is chosen according to the detected voltage then compared with the reference voltage in the comparator. If the supply voltage is upper/lower than the detected voltage, an interrupt signal occurs.

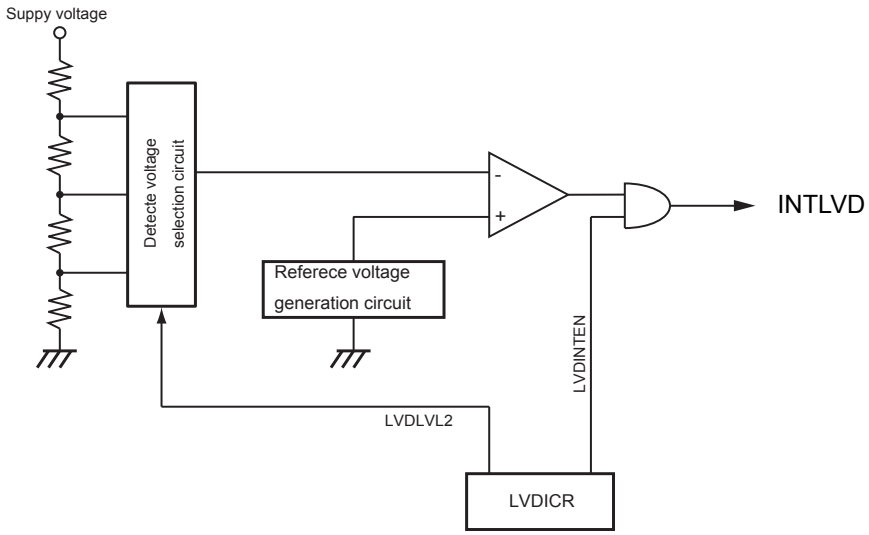


Figure 18-1 Block diagram of LVD (described only LVD interrupt circuit)

18.2 Registers

18.2.1 Register List

The table below shows control registers and their addresses.

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

Register name		Address(Base+)
LVD-INTcontrol register	LVDICR	0x0004
LVD status register	LVDSR	0x0008

18.2.2 LVDICR (LVD-INTcontrol register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	LVDINTEN	INTSEL	LVDLVL2			LVDEN2
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31 - 6	-	R	Read as "0".
5	LVDINTEN	R/W	Controls INTLVD output 0: Disabled 1: Enabled
4	INTSEL	R/W	INT generation condition 0: Only lower than the setting voltage when voltage decreasing. 1: Lower than the setting voltage at decreasing voltage or upper than the setting voltage at increasing voltage.
3 - 1	LVDLVL2[2:0]	R/W	Detected voltage 000: 2.80 ± 0.2V 001: 2.85 ± 0.2V 010: 2.90 ± 0.2V 011: 2.95 ± 0.2V 100: 3.00 ± 0.2V 101: 3.05 ± 0.2V 110: 3.10 ± 0.2V 111: 3.15 ± 0.2V
0	LVDEN2	R/W	Low voltage detection operation 0: Disabled 1: Enabled

Note: LVDICR is initialized by reset with $\overline{\text{RESET}}$ pin.

18.2.3 LVDSR (Status register)

	31	30	29	28	27	26	25	24
bit symbol	–	–	–	–	–	–	–	–
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	–	–	–	–	–	–	–	–
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	–	–	–	–	–	–	–	–
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	–	–	–	–	–	–	LVDST2	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31 - 2	–	R	Read as "0".
1	LVDST2	R	Indicates LVDLVL2 low voltage detection status 0: Power supply voltage is upper than the detection voltage. 1: Power supply voltage is lower than the detection voltage.
0	-	R	Read as undefined.

Note: LVDSR is initialized by reset with $\overline{\text{RESET}}$ pin.

18.3 Operation Description

18.3.1 Detection Voltage Selection and Enabling/Disabling the Operation

The LVDICR register sets the following; choosing the detection voltage, setting the operation to enable/disable, choosing output conditions and setting the output to enable/disable. The LVDICR register is initialized with the reset by $\overline{\text{RESET}}$ pin.

The LVDICR<LVDLVL2[2:0]> bit chooses the detection voltage. If LVDICR<LVDEN2> is set to "1", low voltage detection operation is enabled.

Note: While supply voltage is lower than the detection voltage, if low voltage detection operation is enabled, INTLVD will generate at this timing.

18.3.2 Lower Voltage Detection

If supply voltage is lower than the detection voltage level, INTLVD generates. When the LVDICR<INTSEL> is set to "1", if supply voltage is upper than the detection voltage, INTLVD generates.

After lower voltage detection, to detect INTLVD is required a certain time. If this period is shorter than expected, INTLVD may not generate.

If supply voltage is lower than 1.8V, MCU operation is not guaranteed. In this case, supply voltage must be decreased to 0V and then power-on.

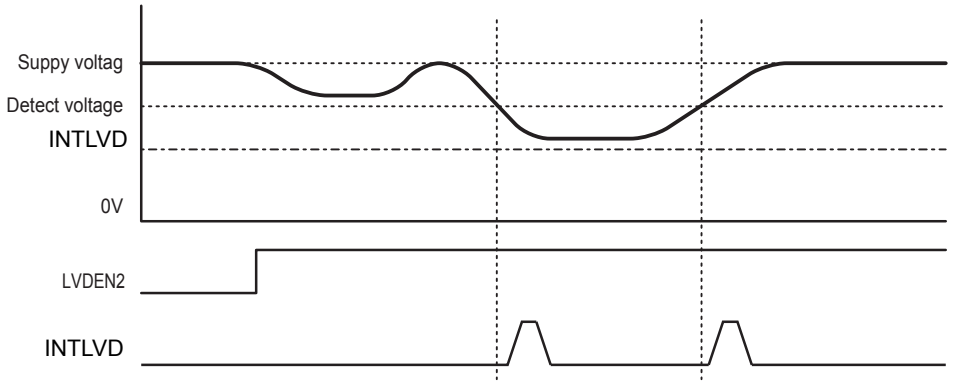


Figure 18-2 low voltage detection Timing

19. Watchdog Timer (WDT)

The watchdog timer (WDT) is for detecting malfunctions (runaway) of the CPU caused by noises or other disturbances and remedying them to return the CPU to normal operation.

If the watchdog timer detects a runaway, it generates a INTWDT interrupt or reset.

Note: INTWDT interrupt is a factor of the non-maskable interrupts (NMI).

Also, the watchdog timer notifies of the detecting malfunction to the external peripheral devices from the watchdog timer pin ($\overline{\text{WDTOUT}}$) by outputting "Low".

Note: TMPM061FWFG does not have the watchdog timer out pin ($\overline{\text{WDTOUT}}$).

19.1 Configuration

Figure 19-1 shows the block diagram of the watchdog timer.

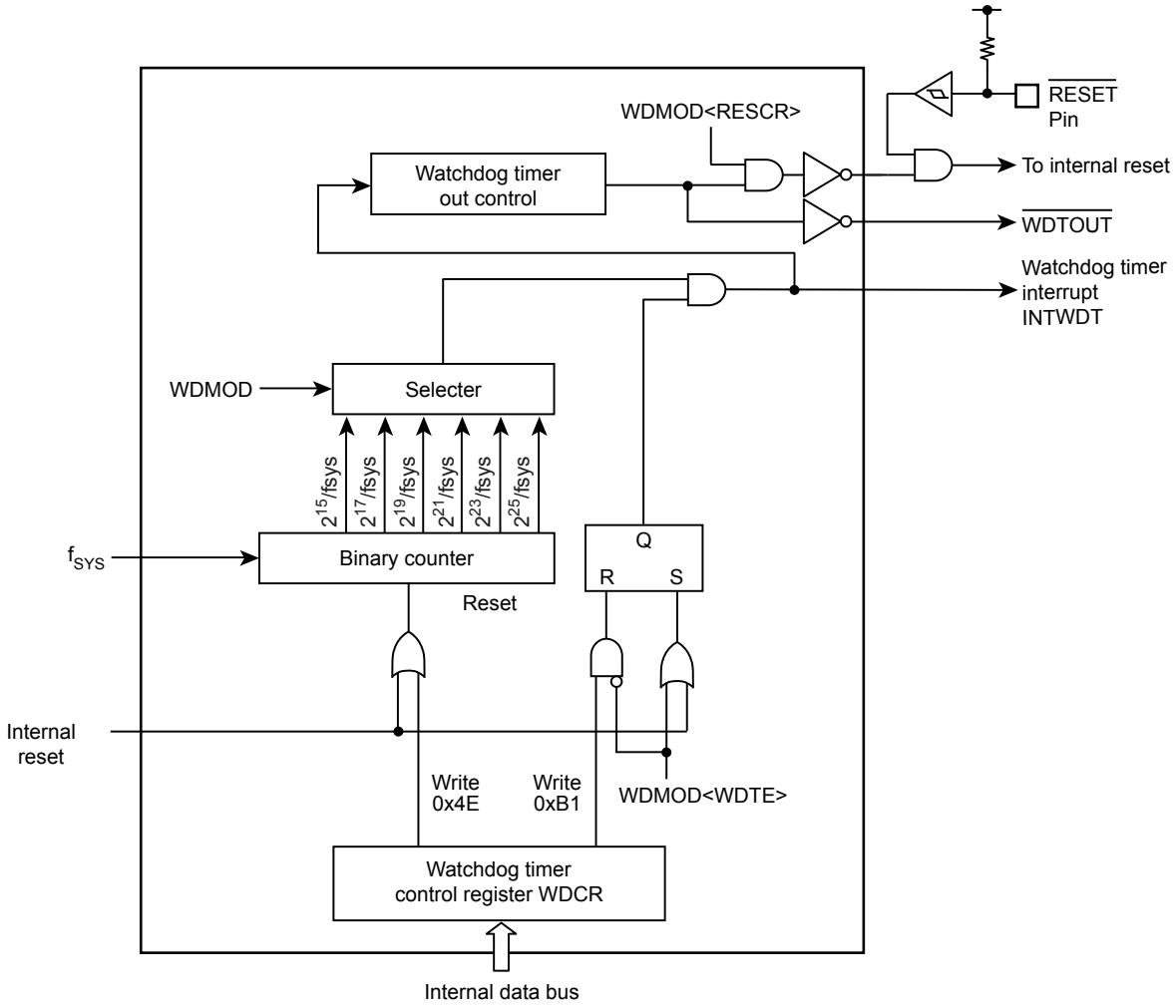


Figure 19-1 Block Diagram of the watchdog Timer

19.2 Register

19.2.1 Register List

The table below shows control registers and their addresses.

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

Register name		Address (Base+)
Watchdog Timer Mode Register	WDMOD	0x0000
Watchdog Timer Control Register	WDCR	0x0004

19.2.2 WDMOD (Watchdog Timer Mode Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	WDTE	WDTP			-	I2WDT	RESCR	-
After reset	1	0	0	0	0	0	1	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	WDTE	R/W	Enable / Disable control 0:Disable 1: Enable To disable the watchdog timer to protect from the error writing by the malfunction, first <WDTE> is set to "0", and then the disable code (0xB1) must be written to WDCR. To change the status of the watchdog timer from "disable" to "enable", set <WDTE> to "1".
6-4	WDTP[2:0]	R/W	Selects WDT detection time 000: $2^{15}/f_{SYS}$ 100: $2^{23}/f_{SYS}$ 001: $2^{17}/f_{SYS}$ 101: $2^{25}/f_{SYS}$ 010: $2^{19}/f_{SYS}$ 110: Reserved 011: $2^{21}/f_{SYS}$ 111: Reserved
3	−	R	Read as "0"
2	I2WDT	R/W	Operation in IDLE mode 0: Stop 1: Operate
1	RESCR	R/W	Operation after detecting malfunction 0: INTWDT interrupt request is generated. Note) 1: Reset
0	−	R/W	Write "0".

Note: INTWDT interrupt is a factor of the non-mask interrupt.

19.2.3 WDCR (Watchdog Timer Control Register)

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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7-0	WDCR	W	Disable / Clear code 0xB1: Disable code 0x4E: Clear code Others : Reserved

19.3 Description of Operation

19.3.1 Basic Operation

The watchdog timer is consists of the binary counter that works using the system clock (fsys) as an input.

Detecting time can be selected between 2^{15}, 2^{17}, 2^{19}, 2^{21}, 2^{23} and 2^{25} by the WDMOD<WDTP[2:0]>.

The detecting time as specified is elapsed, the watchdog timer interrupt (INTWDT) is generated, and the watchdog timer out pin ($\overline{\text{WDTOUT}}$) outputs "Low".

To detect malfunctions (runaways) of the CPU caused by noise or other disturbances, the binary counter of the watchdog timer should be cleared by software instruction before INTWDT interrupt is generated. If the binary counter is not cleared, the non-maskable interrupt is generated by INTWDT. Thus CPU detect malfunction (runaway), malfunction countermeasure program is performed to return to the normal operation.

Additionally, it is possible to resolve the problem of a malfunction (runaway) of the CPU by connecting the watchdog timer out pin to reset pins of peripheral devices.

Note: TMPM061FWFG does not have a watchdog timer out pin ($\overline{\text{WDTOUT}}$).

19.3.2 Operation Mode and Status

The watchdog timer begins operation immediately after a reset is released. If not using the watchdog timer, it should be disabled.

The watchdog timer can not be used at the high-speed frequency clock is stopped. Before transition to below operation modes, the watchdog timer should be disabled.

In IDLE mode, its operation depend on WDMOD<I2WDT> setting.

- STOP mode
- SLEEP mode
- SLOW mode

Also, the binary counter is automatically stopped during debug mode.

19.3.3 Operation when malfunction (runaway) is detected.

19.3.3.1 INTWDT interrupt generation

Figure 19-2 shows the case that INTWDT interrupt is generated (WDMOD<RESCR>="0").

When an overflow of the binary counter occurs, INTWDT interrupt is generated. It is a factor of non-maskable interrupt (NMI). Thus CPU detects non-maskable interrupt and perform the countermeasure program.

When INTWDT interrupt is generated, simultaneously the watchdog timer out ($\overline{\text{WDTOUT}}$) output "Low".

$\overline{\text{WDTOUT}}$ becomes "High" by the watchdog timer clearing that is writing clear code 0x4E to the WDCR.

Note: TMPM061FWFG does not have a watchdog timer out pin ($\overline{\text{WDTOUT}}$).

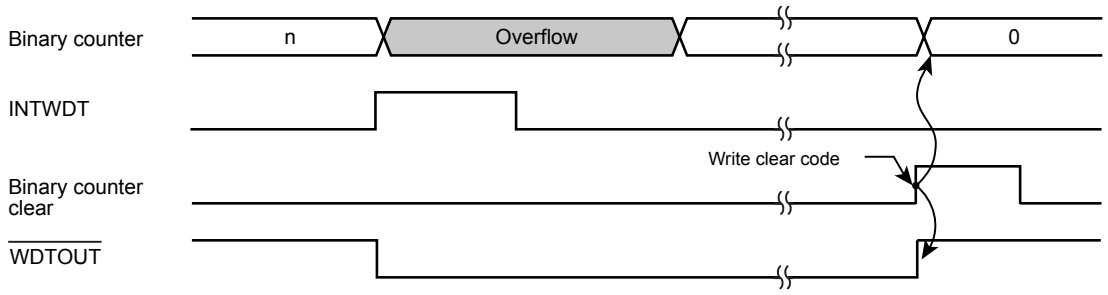


Figure 19-2 INTWDT interrupt generation

19.3.3.2 Internal Reset Generation

Figure 19-3 shows the internal reset generation (WDMOD<RESCR>="1").

MCU is reset by the overflow of the binary counter. In this case, reset status continues for 32 states.

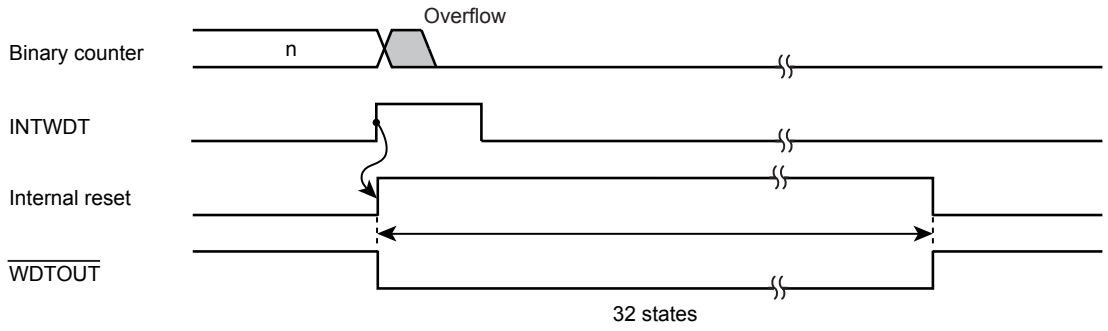


Figure 19-3 Internal reset generation

19.4 Control of the watchdog timer

19.4.1 Disable control

By writing the disable code (0xB1) to WDCR after setting WDMOD<WDTE> to "0", the watchdog timer can be disabled and the binary counter can be cleared.

19.4.2 Enable control

Set WDMOD<WDTE> to "1".

19.4.3 Watchdog timer clearing control

Writing the clear code (0x4E) to WDCR clears the binary counter and it restarts counting.

19.4.4 Detection time of watchdog timer

Set WDMOD<WDTP[2:0]> depend on the detection time.

For example, in the case that $2^{21}/f_{SYS}$ is used, set "011" to WDMOD<WDTP[2:0]>.

20. Flash Memory Operation

This section describes the hardware configuration and operation of Flash memory. In this section, "1-word" means 32 bits.

20.1 Features

20.1.1 Memory Size and Configuration

Table 20-1 and Figure 20-1 show a built-in memory size and configuration of TMPM061FWFG.

Table 20-1 Memory size and configuration

Memory size	Block configuration				# of words per page	# of pages	Write time		Erase time	
	128 KB	64 KB	32 KB	16 KB			1 page	Total area	Block erase	Chip erase
128 KB	-	-	4	-	32	1024	1.25ms	1.28 sec	0.1sec	0.2 sec

Note: The above values are theoretical values not including data transfer time. The write time per chip depends on the write method used by a user.

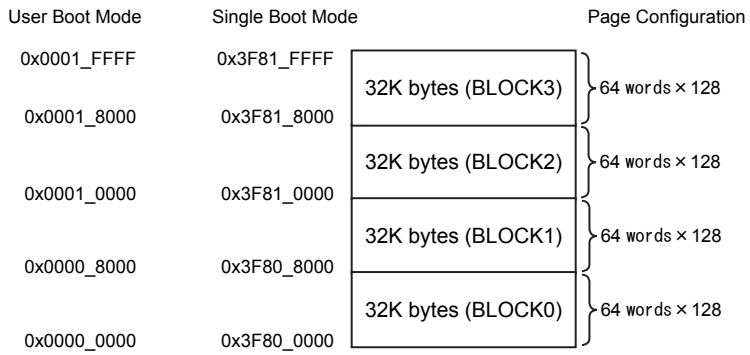


Figure 20-1 Block configuration

Flash memory configuration units ares described as "block" and "page".

- Page

  One page is 32 words. Same address [31:7] is used in a page. First address of the group is [6:0] = 0 and the last address of the group is [6:0] = 0x7F.

- Block

  One block is 32KB and flash memory is consists of four blocks.

Write operation is performed per page. The write time per page is 1.25ms. (Typ.)

Erase is performed per block (auto block erase command use) or performed on entire flash memory (use of auto chip erase command). Erase time varies on commands. If auto block command is used, the erase time will be 0.1 sec per block (Typ.). If the auto chip erase command is used to erase entire area, the time will be 0.2 sec (Typ.).

In addition, the protect function can be used per block. For detail of the protect function, refer to "20.1.5 Protect/Security Function".

20.1.2 Function

Flash memory built-in this device is generally compliant with the JEDEC standards except for some specific functions. Therefore, if a user is currently using a flash memory as an external memory, it is easy to implement the functions into this device. Furthermore, to provide easy write or erase operation, this product contains a dedicated circuit to perform write or chip erase automatically.

JEDEC compliant functions	Modified, added, or deleted functions
・Automatic programming ・Automatic chip erase ・Automatic block erase ・Data polling/toggle bit	<Modified> Block write/erase protect (only software protection is supported) <Deleted> Erase resume - suspend function

20.1.3 Operation Mode

20.1.3.1 Mode Description

This device provides the single chip mode and single boot mode. The single chip mode contains the normal mode and user boot mode. Figure 20-2 shows the mode transition.

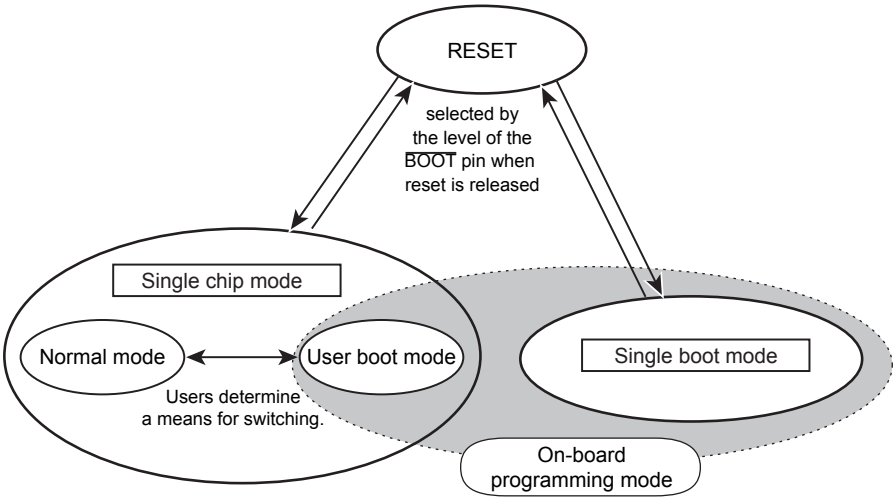


Figure 20-2 Mode transition

(1) Single chip mode

The single chip mode is a mode where the device can boot-up from Flash memory after reset. The mode contains two sub-modes in below.

- Normal mode

  The mode where user application program is executed.
- User boot mode

  The mode where flash memory is re-programmed on the user's set.

Users can switch the normal mode to user boot mode freely. For example, a user can set if PA0 of port A is "1", the mode is the normal mode. If PA0 of port A is "0", the mode is the user boot mode. The user must prepare a routine program in the application program to determine the switching.

(2) Single boot mode

The mode where flash memory can boot-up from the built-in BOOT ROM (Mask ROM) after reset.

The BOOT ROM contains the algorithm that can rewrite Flash memory via serial port of this device on the user's set. With connecting the serial port to external host, data transfer is performed in above-mentioned protocol and re-programed Flash memory.

(3) On-board programming mode

The user boot mode and single boot mode are the modes where flash memory can be re-programmable on the user's set. These two modes are called "on-board programming mode".

20.1.3.2 Mode Determination

Either the single chip or single boot operation mode can be selected by the level of the $\overline{\text{BOOT}}$ pin when reset is released.

Table 20-2 Operation mode setting

Operation mode	Pin	
	$\overline{\text{RESET}}$	$\overline{\text{BOOT}}$
Single chip mode	0 → 1	1
Single boot mode	0 → 1	0

20.1.4 Memory Map

Figure 20-3 shows a comparison of the memory map in the single chip mode and single boot mode. In the single boot mode, built-in Flash memory is mapped to 0x3F80_0000 and subsequent addresses, and the built-in BOOT ROM is mapped to 0x0000_0000 through 0x0000_0FFF.

Flash memory and RAM addresses are shown below.

FLASH size	RAM size	FLASH address	RAM address
128 KB	16 KB	0x0000_0000 to 0x0001_FFFF(single chip mode) 0x3F80_0000 to 0x3F81_FFFF(single boot mode)	0x2000_0000 to 0x2000_3FFF

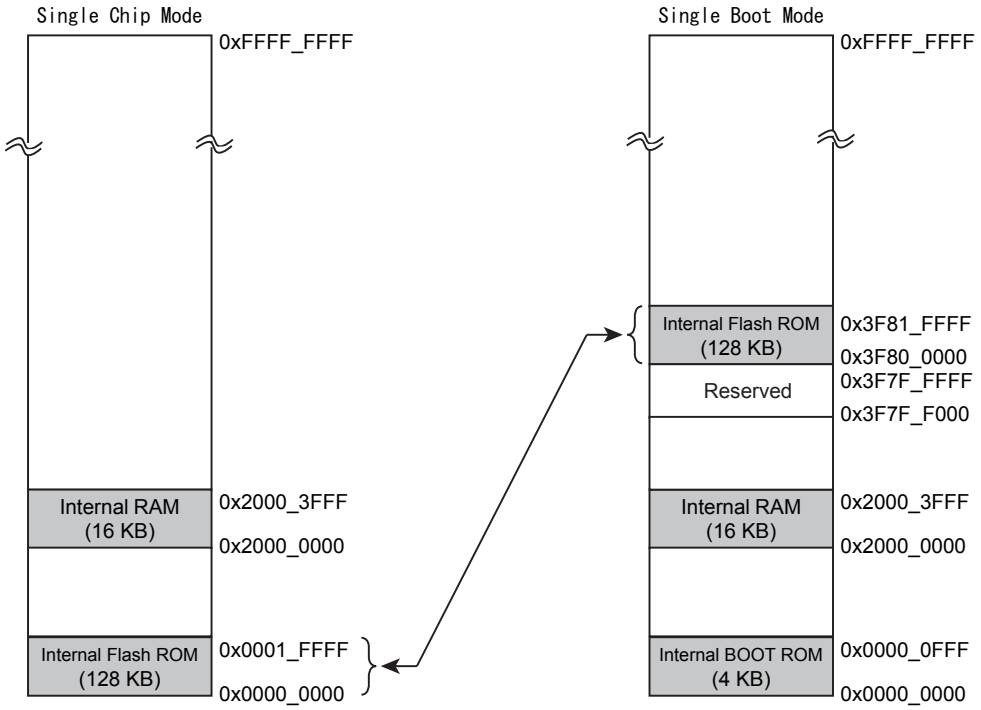


Figure 20-3 Comparison of memory map

20.1.5 Protect/Security Function

This device has the protect and security functions for Flash memory.

1. Protect function

   The write/erase operation can be inhibited per block.

2. Security function

   The read operation from a flash writer can be inhibited.

   Usage restrictions on debug functions

20.1.5.1 Protect Function

This function inhibits the write/erase operation per block.

To enable the protect function, a protect bit corresponding to a block is set to "1" using the protect bit program command. If a protect bit is set to "0" using the protect bit erase command, a block protect can be canceled. The protect bit can be monitored with FCFLCS<BLPRO[3:0]>.

A program of protect bit can be programmed by 1-bit unit and can be erased by 4-bit unit. For detail of programming/erasing of protect bits, refer to "20.2.4 Command Description".

20.1.5.2 Security Function

Table 20-3 shows operations when the security function is enabled.

Table 20-3 Operations when the security function is enabled.

Item	Description
Read flash memory	CPU can read flash memory.
Debug port	JTAG, serial wire or trace communication is disabled.
Command execution to Flash memory	Command write to flash memory is not accepted. If a user tries to erase a protect bit, chip erase is executed and all protect bits are erased.

The security function is enabled under the following conditions;

1. FCSECBIT<SECBIT> is set to "1".
2. All protect bits (FCFLCS<BLPRO>) are set to "1".

FCSECBIT<SECBIT> is set to "1" by the pin reset. Rewriting of FCSECBIT <SECBIT> is described in below.

Note: Use a 32-bit transfer instruction when the following writing operations, item1 and 2.

1. Write the specified code (0xa74a9d23) to FCSECBIT
2. Write data within 16 clocks after the operation of item 1.

20.1.6 Register

20.1.6.1 Register List

Base Address = 0x41FF_F000

Register name		Address(Base+)
Security bit register	FCSECBIT	0x0010
Flash control register	FCFLCS	0x0020

20.1.6.2 FCFLCS (Flash control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	BLPRO3	BLPRO2	BLPRO1	BLPRO0
After reset	0	0	0	0	(Note 2)	(Note 2)	(Note 2)	(Note 2)
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	-	RDY/BSY
After reset	0	0	0	0	0	0	0	1

Bit	Bit Symbol	Type	Function
31-20	−	R	Read as "0".
19-16	BLPRO3- BLPRO0	R	Protection status of Block3 to 0 0: Not protected 1: Protected Protect bit values correspond to protect status of each block. If corresponding bit indicates "1", corresponding block is in the protection status. A block in the protection status cannot be re-programmable.
15-1	−	R	Read as "0".
0	RDY/BSY	R	Ready/Busy (Note 1) (Note3) 0: Busy (during auto operation) 1: Ready (auto operation ends) This bit is a function bit to monitor flash memory from CPU. While flash memory is in auto operation, this bit outputs "0" to indicate that flash memory is busy. Once auto operation is finished, this bit becomes ready state and outputs "1". Then next command is accepted. If a result of auto operation is failed, this bit outputs "0" continuously. The bit returns to "1" by hardware reset.

Note 1: Make sure that flash memory is ready before commands are issued.

Note 2: A value will correspond to the protection status.

Note 3: Access the Flash memory after changing <RDY/BSY> "0" into "1" and passing for 200μs or more.

20.1.6.3 FCSECBIT (Security bit register)

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

Bit	Bit Symbol	Type	Function
31-1	−	R	Read as "0".
0	SECBIT	R/W	Security bit 0: Security function setting is disabled. 1: Security function setting is enabled.

Note: This register is initialized by pin reset.

20.2 Detail of Flash Memory

In on-board programming, the CPU executes commands for reprogramming or erasing Flash memory. This reprogramming/erase control program should be prepared by the user beforehand. Since Flash memory content cannot be read while Flash memory is being written or erased, it is necessary to run the reprogram/erase control program on the built-in RAM. Do not generate interrupt/fault to avoid abnormal program termination.

20.2.1 Function

Flash memory is generally compliant with the JEDEC standards except for some specific functions. However; a method of address designation of operation command is different from standard commands.

If write/erase operation is executed, commands are input to flash memory using 32-bit (1-word) store instruction command. After command input, write or erase operation is automatically executed in inside.

Table 20-4 Flash memory function

Main function	Description
Automatic page program	Writes data automatically.
Automatic chip erase	Erases the entire area of Flash memory automatically.
Automatic block erase	Erases a selected block automatically.
Write/erase protect	The write or erase operation can be individually inhibited for each block.

20.2.2 Operation Mode of Flash Memory

Flash memory provides mainly two types of operation modes;

- The mode to read memory data (Read mode)
- The mode to erase or rewrite memory data automatically (Automatic operation mode)

After power-on, after rest or after automatic operation mode is finished normally, Flash memory becomes read mode. Instruction stored in Flash memory or data read is executed in the read mode.

If commands is input during the read mode, the operation mode becomes the automatic operation. If the command process is normally finished, the operation mode returns to the read mode except the ID-Read command. During the automatic operation, data read and instruction execution stored in Flash memory cannot be performed.

20.2.3 How to Execute Command

The command execution is performed by writing command sequences to Flash memory with a store instruction. Flash memory executes each automatic operation command according to the combination of input addresses and data. For detail of the command execution, refer to "20.2.4 Command Description".

An execution of store instruction to the Flash memory is called "bus write cycle". Each command consists of some bus write cycles. In Flash memory, when address and data of bus write cycle are performed in the specified order, the automatic command operation is performed. When the cycle is performed in non-specified order, Flash memory stops command execution and returns to the read mode.

If you cancel the command during the command sequence or input a different command sequence, execute the read command or read/reset command. Then Flash memory stops command execution and returns to the read mode. The read command and read/reset command are called "software reset".

When write command sequence ends, the automatic operation starts and FCFLCS<RDY/BSY> is set to "0". When the automatic operation normally ends, FCFLCS<RDY/BSY> = "1" is set and Flash memory returns to the read mode. Access the Flash memory after changing <RDY/BSY> "0" into "1" and passing for 200μs or more.

New command sequences are not accepted during the automatic operation.

Notes on the command execution;

1. To recognize command, command sequencer need to be in the read mode before command starting. Confirm FCFLCS<RDY/BSY> = 1 is set prior to the first bus write cycle of each command. Consecutively, it is recommended that the read command is executed.
2. Execute each command sequence from outside of Flash memory.
3. Execute sequentially each bus write cycle by data transfer instruction in one-word (32-bit).
4. Do not access Flash memory during the each command sequence. Do not generate any interrupt or fault.
5. Upon issuing a command, if any address or data is incorrectly written, make sure to return to the read mode by using software reset.

20.2.4 Command Description

This section explains each command content. For detail of specific command sequences, refer to "20.2.5 Command Sequence".

20.2.4.1 Automatic Page Program

(1) Operation Description

The automatic page program writes data per page. When the program writes data to multiple pages, a page command need to be executed in page by page. Writing across pages is not possible.

Writing to Flash memory means that data cell of "1" becomes data of "0". It is not possible to become data cell of "1" from data of "0". To become data cell of "1" from "0", the erase operation is required.

The automatic page program is allowed only once to each page already erased. Either data cell of "1" or "0" cannot be written data twice or more. If rewriting to a page that has already been written once, the automatic page program is needed to be set again after the automatic block erase or automatic chip erase command is executed.

Note 1: Page program execution to the same page twice or more without erasing operation may damage the device.

Note 2: Writing to the protected block is not possible.

(2) How to Set

The 1st to 3rd bus write cycles indicate the automatic page program command.

In the 4th bus write cycle, the first address and data of the page are written. On and after 5th bus cycle, one page data will be written sequentially. Data is written in one-word unit (32-bit).

If a part of the page is written, set "0xFFFFFFFF" as data, which means not required to write, for entire one page.

No automatic verify operation is performed internally in the device. So, be sure to read the data programmed to confirm that it has been correctly written.

If the automatic page program is abnormally terminated, that page has been failed to write. It is recommended not to use the device or not to use the block including the failed address.

20.2.4.2 Automatic Chip Erase

(1) Operation Description

The automatic chip erase is executed to the memory cell of all addresses. If protected blocks are contained, these blocks will not be erased. If all blocks are protected, the automatic chip erase operation will not performed and will return to the read mode after a command sequence is input.

(2) How to Set

The 1st to 6th bus write cycles indicate the automatic chip erase command. After the command sequence is input, the automatic chip erase operation starts.

No automatic verify operation is performed internally in the device. So, be sure to read the data to confirm that it has been correctly erased.

20.2.4.3 Automatic Block Erase

(1) Operation Description

The automatic erase command performs erase operation to the specified block. If the specified block is protected, erase operation is not executed.

(2) How to Set

The 1st to 5th bus write cycles indicate the automatic block erase command. In the 6th bus write cycle, the block to be erased is specified. After the command sequence is input, the automatic block erase operation starts.

No automatic verify operation is performed internally in the device. So, be sure to read the data to confirm that it has been correctly erased.

20.2.4.4 Automatic Protect Bit Program

(1) Operation Description

The automatic protect bit program writes "1" to a protect bit at a time. To set "0" to a protect bit, use the automatic protect bit erase command.

For detail of the protect function, refer to "20.1.5 Protect/Security Function".

(2) How to Set

The 1st to 6th bus write cycles indicate the automatic protect bit program command. In the 7th bus write cycle, the protect bit to be written is specified. After the command sequence is input, the automatic protect bit program starts. Check whether write operation is normally terminated with FCFLCS<BLPRO>.

20.2.4.5 Auto Protect Bit Erase

(1) Operation Description

The automatic protect bit erase command operation depends on the security status. For detail of security status, refer to "20.1.5 Protect/Security Function".

- Non-security status

  Clear the specified protect bit to "0". Protect bit erase is performed in 4-bit unit.
- Security status

  Erase all protect bits after all addresses of Flash memory are erased.

(2) How to Set

The 1st to 6th bus write cycles indicate the automatic protect bit erase command. In the 7th bus write cycle, the protect bit to be erased is specified. After the command sequence is input, the automatic protect bit erase operation starts.

In the non-security status, specified protect bit is erased. Check whether erase operation is normally terminated with FCFLCS<BLPRO>.

In the security status, all addresses and all protect of Flash memory bits are erased. Confirm if data and protect bits are erased normally. If necessary, execute the automatic protect bit erase, automatic chip erase or automatic block erase.

All cases are the same as other commands, FCFLCS<RDY/BSY> becomes "0" during the automatic protect bit erase command operation. After the operation is complete, FCFLCS<RDY/BSY> becomes "1" and Flash memory will return to the read mode.

20.2.4.6 ID-Read

(1) Operation Description

The ID-Read command can read information including Flash memory type and three types of codes such as a maker code, device code and macro code.

(2) How to Set

The 1st to 3rd bus write cycles indicate the ID-Read command. In the 4th bus write cycle, the code to be read is specified. After the 4th bus write cycle, read operation in the arbitrary flash area acquires codes.

The ID-Read can be executed successively. The 4th bus write cycle and reading ID value can be executed repeatedly.

The ID-Read command does not automatically return to the read mode. To return to the read mode, execute the read command or read/reset command.

20.2.4.7 Read Command and Read/reset Command (Software Reset)

(1) Operation Description

A command to return Flash memory to the read mode.

When the ID-Read command is executed, macro stops at the current status without automatically return to the read mode. To return to the read mode from this situation, use the read command or read/reset command. It is also used to cancel the command when commands are input to the middle.

(2) How to Set

The 1st bus cycle indicates the read command. The 1st to 3rd bus write cycles indicate the read/reset command. After either command sequence is executed, Flash memory returns to the read mode.

20.2.5 Command Sequence

20.2.5.1 Command Sequence List

Table 20-5 shows addresses and data of bus write cycle in each command.

All command cycles except the 5th bus cycle of ID-Read command are bus write cycles. A bus write cycle is performed by 32-bit (1-word) data transfer instruction. (Following table shows only lower 8 bits of data.)

For detail of addresses, refer to Table 20-6. Use below values to "command" described in a column of Addr[15:9] in the Table 20-6.

Note 1) Always set to "0" to the address bit [1:0].

Note 2) Set below values to the address bit [19] according to Flash memory size.
Memory size is 1MB or less : Always set to "0"
Memory size is over 1MB : If bus write to 1MB area or less, the bit is set to "0".
If bus write to over 1MB area, the bit is set to "1".

Table 20-5 Command Sequence

Command	1st bus cycle	2nd bus cycle	3rd bus cycle	4th bus cycle	5th bus cycle	6th bus cycle	7th bus cycle
	Addr.	Addr.	Addr.	Addr.	Addr.	Addr.	Addr.
	Data	Data	Data	Data	Data	Data	Data
Read	0xXX	–	–	–	–	–	–
	0xF0	–	–	–	–	–	–
Read/reset	0xX55X	0xXAAX	0xX55X	–	–	–	–
	0xAA	0x55	0xF0	–	–	–	–
ID-Read	0xX55X	0xXAAX	0xX55X	IA	0xXX	–	–
	0xAA	0x55	0x90	0x00	ID	–	–
Automatic page program	0xX55X	0xXAAX	0xX55X	PA	PA	PA	PA
	0xAA	0x55	0xA0	PD0	PD1	PD2	PD3
Automatic chip erase	0xX55X	0xXAAX	0xX55X	0xX55X	0xXAAX	0xX55X	–
	0xAA	0x55	0x80	0xAA	0x55	0x10	–
Automatic block erase	0xX55X	0xXAAX	0xX55X	0xX55X	0xXAAX	BA	–
	0xAA	0x55	0x80	0xAA	0x55	0x30	–
Automatic protect bit program	0xX55X	0xXAAX	0xX55X	0xX55X	0xXAAX	0xX55X	PBA
	0xAA	0x55	0x9A	0xAA	0x55	0x9A	0x9A
Automatic protect bit erase	0xX55X	0xXAAX	0xX55X	0xX55X	0xXAAX	0xX55X	0xXX
	0xAA	0x55	0x6A	0xAA	0x55	0x6A	0x6A

Supplementary explanation

- IA: IDAddress
- ID: ID data
- PA: Program page address
- PD: Program data (32-bit data)

  After the 4th bus cycle, input data in the order of the addresses per page
- BA: Block address (see Table 20-7)

- PBA: Protect bit address (see Table 20-8)

20.2.5.2 Address Bit Configuration in the Bus Cycle

Table 20-6 is used in conjunction with "Table 20-5 Command Sequence".

Set the address setting according to the normal bus write cycle address configuration from the first bus cycle.

Table 20-6 Address bit configuration in the bus write cycle

Address	Addr [31:15]	Addr [14]	Addr [13:12]	Addr [11:9]	Addr [8:7]	Addr [6:4]	Addr [3:0]

Normal Command	Normal bus write cycle address configuration				
	Flash area	"0" is recommended.	Command	Addr[1:0] = "0" (fixed) Other bits = "0" (recommended)	
ID-READ	IA: ID address (Setting of the 4th bus write cycle address for ID-READ)				
	Flash area	"0" is recom-mended.	ID Address	Addr[1:0] = "0" (fixed) Other bits= "0" (recommended)	
Block erase	BA: Block address(Setting of the 6th bus write cycle address for block erase)				
	Block address (Table 20-7)	Addr[1:0] = "0" (fixed) Other bits= "0" (recommended)			
Automatic page pro-gram	PA: Program page address (Setting of the 4th bus write cycle address for page program)				
	Page address			Addr[1:0] = "0" (fixed) Other bits= "0" (recommended)	
Protect bit program	PBA: Protect bit address (Setting of the 7th bus write cycle address for protect bit program)				
	Flash area	Fix to "0"		Protect bit selection (Table 20-8)	Addr[1:0] = "0" (fixed) Other bits= "0" (recommended)

20.2.5.3 Block Address(BA)

Table 20-7 shows block addresses. Specify any address included in the block to be erased in the 6th bus write cycle of the automatic block erase command.

Table 20-7 Block address

Block	Address (User boot mode)	Address (Single boot mode)	Size (Kbyte)
2	0x0001_8000 to 0x0001_FFFF	0x3F81_0000 to 0x3F81_FFFF	32
3	0x0001_0000 to 0x0000_7FFF	0x3F81_0000 to 0x3F80_7FFF	32
1	0x0000_8000 to 0x0000_FFFF	0x3F80_8000 to 0x3F80_FFFF	32
0	0x0000_0000 to 0x0000_7FFF	0x3F80_0000 to 0x3F80_7FFF	32

20.2.5.4 How to Specify Protect Bit (PBA)

The protect bit is specified in 1-bit unit in programming and in 4-bit unit in erasing.

Table 20-8 shows a protect bit selection table of the automatic protect bit program. The column of address example indicates an address described in upper side is used in the use boot mode and the lower side is used in the single boot mode.

Four protect bits are erased by the automatic protect bit erase command in all.

Table 20-8 Protect bit program address

Block	Protect bit	Address of 7th bus write cycle			Address example [31:0]
		Address [14:9]	Address [8]	Address [7]	
Block0	<BLPRO[0]>	Fix to "0"	0	0	0x0000_0000 0x3F80_0000
Block1	<BLPRO[1]>		0	1	0x0000_0080 0x3F80_0080
Block2	<BLPRO[2]>		1	0	0x0000_0100 0x3F80_0100
Block3	<BLPRO[3]>		1	1	0x0000_0180 0x3F80_0180

20.2.5.5 ID-Read Code (IA, ID)

Table 20-9 shows how to specify a code and the content using ID-Read command.

The column of address example indicates an address described in the upper side is used in the use boot mode and the lower side is used in the single boot mode

Table 20-9 ID-Read Command codes and contents

Code	ID[7:0]	IA[13:12]	Address Example [31:0]
Manufacture code	0x98	0y00	0x0000_0000 0x3F80_0000
Device code	0x5A	0y01	0x0000_1000 0x3F80_1000
-	Reserved	0y10	-
Macro code	0x33	0y11	0x0000_3000 0x3F80_3000

20.2.5.6 Example of Command Sequence

(1) use boot mode

Command		Bus cycle						
		1	2	3	4	5	6	7
Read	Address	0x0000_0000	–	–	–	–	–	–
	Data	0x0000_00F0	–	–	–	–	–	–
Read/reset	Address	0x0000_0550	0x0000_0AA0	0x0000_0550	–	–	–	–
	Data	0x0000_00AA	0x0000_0055	0x0000_00F0	–	–	–	–
ID-Read	Address	0x0000_0550	0x0000_0AA0	0x0000_0550	IA	0x0000_0000	–	–
	Data	0x0000_00AA	0x0000_0055	0x0000_0090	0x0000_0000	ID	–	–
Automatic page program	Address	0x0000_0550	0x0000_0AA0	0x0000_0550	PA	In the following cycles, write addresses and data successively per page.		
	Data	0x0000_00AA	0x0000_0055	0x0000_00A0	PD			
Automatic chip erase	Address	0x0000_0550	0x0000_0AA0	0x0000_0550	0x0000_0550	0x0000_0AA0	0x0000_0550	–
	Data	0x0000_00AA	0x0000_0055	0x0000_0080	0x0000_00AA	0x0000_0055	0x0000_0010	–
Automatic block erase	Address	0x0000_0550	0x0000_0AA0	0x0000_0550	0x0000_0550	0x0000_0AA0	BA	–
	Data	0x0000_00AA	0x0000_0055	0x0000_0080	0x0000_00AA	0x0000_0055	0x0000_0030	–
Automatic protect bit program	Address	0x0000_0550	0x0000_0AA0	0x0000_0550	0x0000_0550	0x0000_0AA0	0x0000_0550	PBA
	Data	0x0000_00AA	0x0000_0055	0x0000_009A	0x0000_00AA	0x0000_0055	0x0000_009A	0x0000_009A
Automatic protect bit erase	Address	0x0000_0550	0x0000_0AA0	0x0000_0550	0x0000_0550	0x0000_0AA0	0x0000_0550	0x0000_0550
	Data	0x0000_00AA	0x0000_0055	0x0000_006A	0x0000_00AA	0x0000_0055	0x0000_006A	0x0000_006A

(2) Data single boot mode

Command		Bus cycle						
		1	2	3	4	5	6	7
Read	Address	0x3F80_0000	–	–	–	–	–	–
	Data	0x0000_00F0	–	–	–	–	–	–
Read/reset	Address	0x3F80_0550	0x3F80_0AA0	0x3F80_0550	–	–	–	–
	Data	0x0000_00AA	0x3F80_0055	0x3F80_00F0	–	–	–	–
ID-Read	Address	0x3F80_0550	0x3F80_0AA0	0x3F80_0550	IA	0x0000_0000	–	–
	Data	0x0000_00AA	0x0000_0055	0x0000_0090	0x0000_0000	ID	–	–
Automatic page program	Address	0x3F80_0550	0x3F80_0AA0	0x3F80_0550	PA	In the following cycles, write addresses and data successively per page.		
	Data	0x0000_00AA	0x0000_0055	0x0000_00A0	PD			
Automatic chip erase	Address	0x3F80_0550	0x3F80_0AA0	0x3F80_0550	0x3F80_0550	0x3F80_0AA0	0x3F80_0550	–
	Data	0x0000_00AA	0x0000_0055	0x0000_0080	0x0000_00AA	0x0000_0055	0x0000_0010	–
Automatic block erase	Address	0x3F80_0550	0x3F80_0AA0	0x3F80_0550	0x3F80_0550	0x3F80_0AA0	BA	–
	Data	0x0000_00AA	0x0000_0055	0x0000_0080	0x0000_00AA	0x0000_0055	0x0000_0030	–
Automatic protect bit program	Address	0x3F80_0550	0x3F80_0AA0	0x3F80_0550	0x3F80_0550	0x3F80_0AA0	0x3F80_0550	PBA
	Data	0x0000_00AA	0x0000_0055	0x0000_009A	0x0000_00AA	0x0000_0055	0x0000_009A	0x0000_009A
Automatic protect bit erase	Address	0x3F80_0550	0x3F80_0AA0	0x3F80_0550	0x3F80_0550	0x3F80_0AA0	0x3F80_0550	0x3F80_0550
	Data	0x0000_00AA	0x0000_0055	0x0000_006A	0x0000_00AA	0x0000_0055	0x0000_006A	0x0000_006A

20.2.6 Flowchart

20.2.6.1 Automatic Program

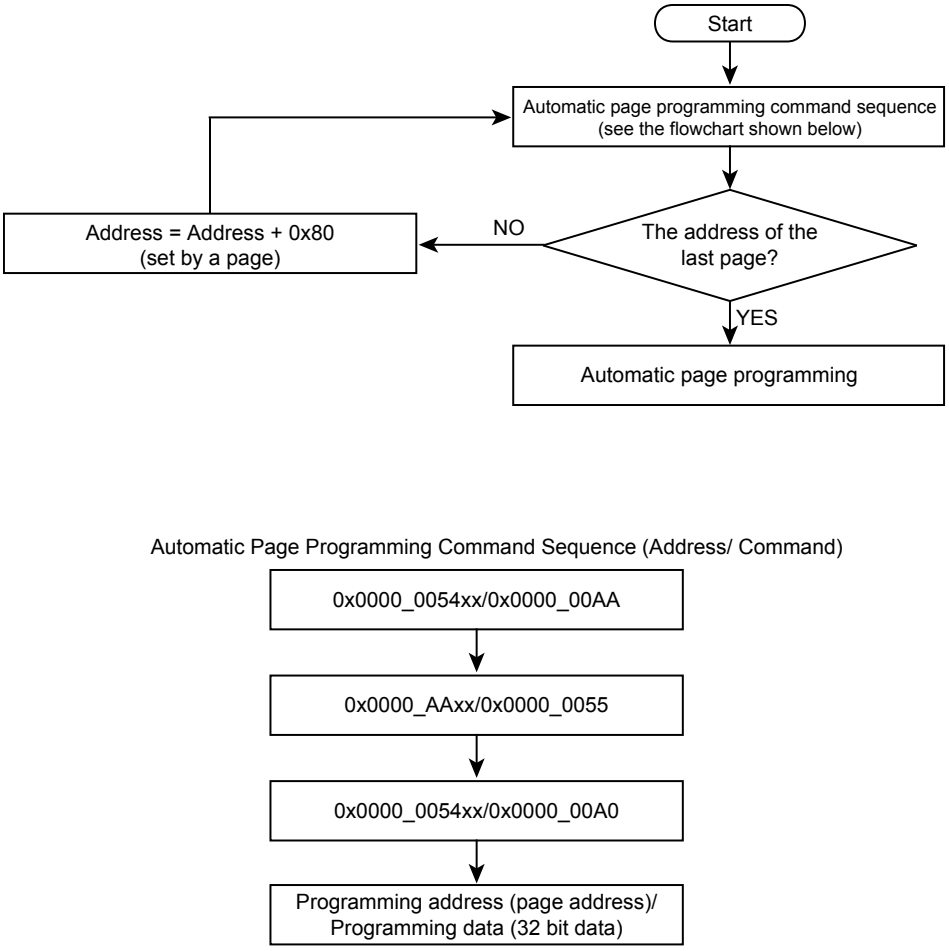


Figure 20-4 Flowchart of automatic program

20.2.6.2 Automatic Erase

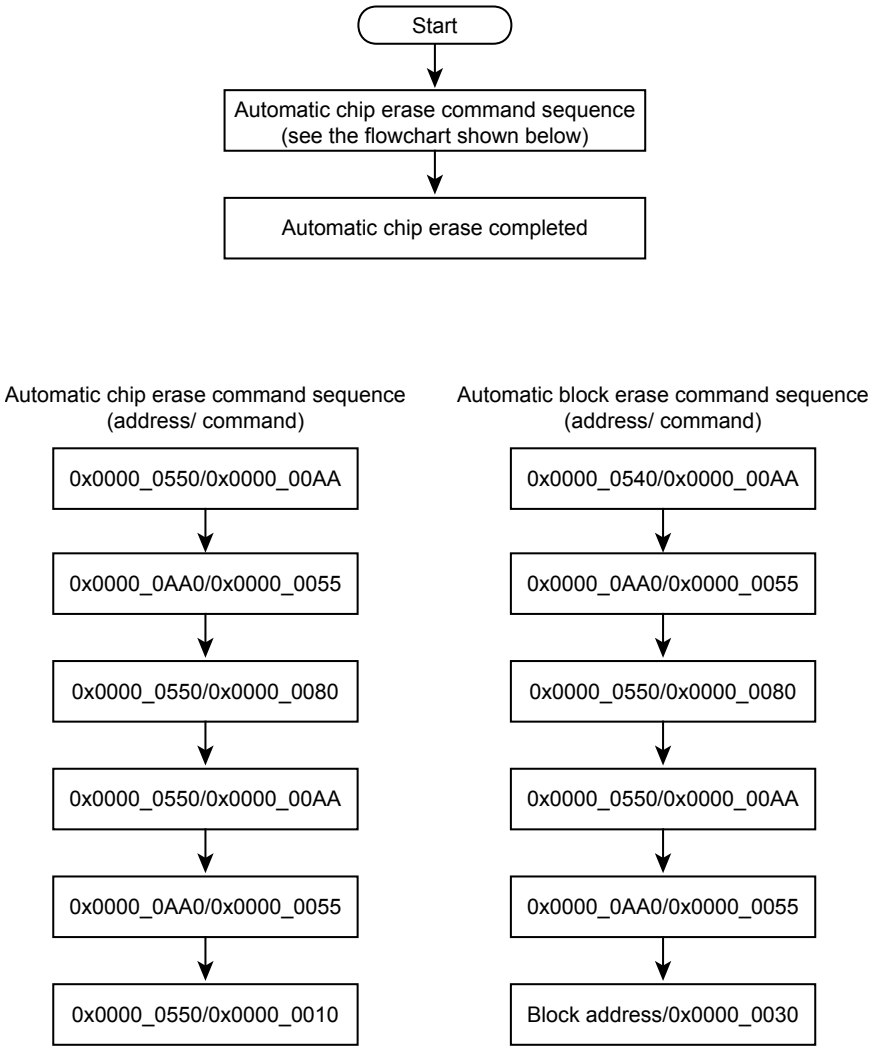


Figure 20-5 Flowchart of automatic erase

20.3 How to Reprogram Flash using Single Boot Mode

The single boot mode utilizes a program contained in built-in BOOT ROM for reprogrammig Flash memory. In this mode, BOOT ROM is mapped to the area containg interrupt vector tables and Flash memory is mapped to another address area other than BOOT ROM area.

In the boot mode, Flash memory is reprogrammed using serial command/data transfer. With connecting serial channel (SIO/UART) of this device to the external host, a reprogramming program is copied from the external host to the built-in RAM. A reprogramming routine in the RAM is executed to reprogram Flash memory. For details of communication with host, follow the protocol described later.

Even in the single boot mode, do not generate interrupt/fault to avoid abnormal program termination.

To secure the contents of Flash memory in the single chip mode (normal operation mode), once re-programming is complete, it is recommended to protect relevant flash blocks against accidental erasure during subsequent single chip operations.

20.3.1 Mode Setting

In order to execute the on-board programming, this device is booted-up in the single boot mode. Below setting is for the single boot mode setting.

$\overline{\text{BOOT}}$ = 0
$\overline{\text{RESET}}$ = 0 → 1

While $\overline{\text{BOOT}}$ pin is set to the above in advance, set $\overline{\text{RESET}}$ pin to "0". Then release $\overline{\text{RESET}}$ pin, the device will boot-up in the single boot mode.

20.3.2 Interface Specification

This section describes SIO/UART communication format in the single boot mode. The serial operation supports both UART (asynchronous communication) and I/O interface modes. In order to execute the on-board programming, set the communication format of the programming controller as well.

- UART communication

  Communication channel: channel 0

  Serial transfer mode: UART (asynchronous), half-duplex, LSB first

  Data length: 8-bit

  Parity bit: None

  STOP bit: 1-bit

  Baud rate: Arbitrary baud rate

- I/O interface mode

  Communication channel: channel 0

  Serial transfer mode: I/O interface, full-duplex, LSB first

  Synchronous signal (SCLK0): Input mode, rising edge setting

  Handshaking signal: PH3 (output mode)

  Baud rate: Arbitrary baud rate

The boot program operates the clock/mode control block setting as an initial condition. For detail of the initial setting of the clock, refer to "Clock/Mode control".

As explained in the "20.3.5.1 Serial Operation Mode Determination", a baud rate is determined by the 16-bit timer (TMRB). When determining the baud rate, communication is executed by 1/16 of a desired baud rate. Therefore, the communication baud rate must be within the measurable range. The timer count clock operates at ΦT1 (fc/2).

A handshaking pin of I/O interface mode outputs "Low" waiting in receive state and outputs "High" in transmission state. Check the handshaking pin before communications and must follow the communication protocol.

Table 20-10 shows the pins used in the boot program. Other than these pins are not used by the boot program.

Table 20-10 Pin connection

Pin		Interface	
		UART	I/O interface mode
Mode setting pin	$\overline{\text{BOOT}}$	o	o
Reset pin	$\overline{\text{RESET}}$	o	o
Communication pin	TXD0 (PH0)	o	o
	RXD0 (PH1)	o	o
	SCLK0 (PH2)	×	o (Input mode)
	PH3	×	o (Output mode)

o:used ×:unused

20.3.3 Restrictions on Internal Memories

Note that the single boot mode places restrictions on the built-in RAM and built-in flash memory as shown in Table 20-11.

Table 20-11 Restrictions on the memories in the single boot mode

Memory	Restrictions
Internal RAM	Boot program uses the memory as a work area through 0x2000_0000 to 0x2000_03FF. Store the program 0x2000_0400 through the end address of RAM. The start address of the program must be even address.
Internal flash memory	The following addresses are assigned for storing software ID information and passowerds. Storing program in below addresses is not recommendable. 0x3F81_FFF0 to 0x3F81_FFFF

Note: If a password is erased data (0xFF), it is difficult to protect data secure due to an easy-to-guess password. Even if the single boot mode is not used, it is recommended to set a unique value as a password.

20.3.4 Operation Command

The boot program provides the following operation commands.

Table 20-12 Operation command data

Operation command data	Operation mode
0x10	RAM transfer
0x40	Flash memory chip erase and protect bit erase

20.3.4.1 RAM Transfer

The RAM transfer is to store data from the controller to the built-in RAM . When the transfer is complete normally, a user program starts. User program can use the memory address of 0x2000_0400 or later except 0x2000_0000 to 0x2000_03FF for the boot program. CPU will start execution from RAM store start address. The start address must be even address.

This RAM transfer function enables user-specific on-board programming control. In order to execute the on-board programming by a user program, use Flash memory command sequence explained in 20.2.5.

20.3.4.2 Flash Memory Chip Erase and Protect Bit Erase

Flash memory chip erase and protect bit erase commands erase the entire blocks of Flash memory and write/erase protects of all blocks regardless of write/erase protect or security status.

Note: Even if a command is completed normally, ACK may be returned as the abnormal termination. In this case, check the erasure again.

20.3.5 Common Operation regardless of Command

This section describes common operation under the boot program execution.

20.3.5.1 Serial Operation Mode Determination

When the controller communicates via UART, set the 1st byte to 0x86 at the desired baud rate. When the controller communicate via I/O interface mode, set the 1st byte to 0x30 at 1/16 of the desired baud rate. Figure 20-6 shows waveforms in each case.

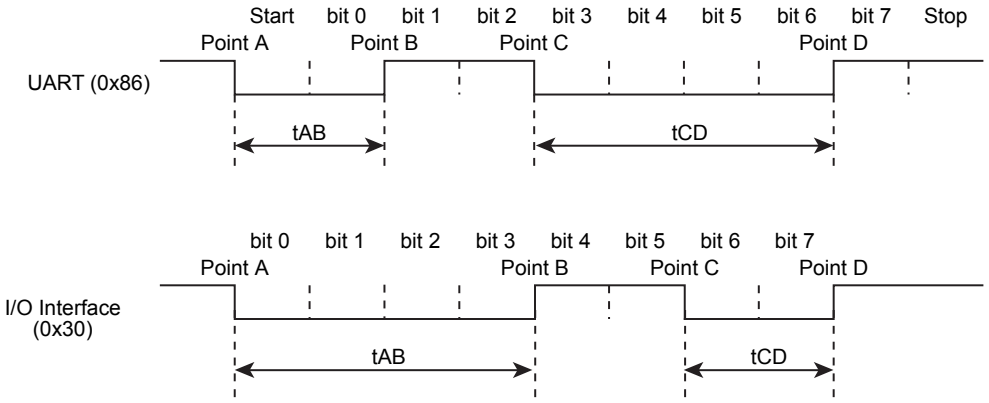


Figure 20-6 Serial operation mode determination data

Figure 20-7 shows a flowchart of boot program. Using 16-bit timer (TMRB) with the time of tAB, tAC and tAD, the 1st byte of serial operation mode determination data (0x86, 0x30) after reset is provided. In Figure 20-7, the CPU monitors level of the receive pin, and obtains a timer value at the moment when the receive pin's level is changed. Consequently, the timer values of tAB, tAC and tAD have a margin of error. In addition, note that if the transfer goes at a high baud rate, the CPU may not be able to determine the level of receive pin. In particular, I/O Interface tends to generate this problem since its baud rate is generally much higher than those of UART. To avoid this, the controller should send data at 1/16 of the desired baud rate in the I/O interface mode.

The flowchart in Figure 20-8 shows the serial operation mode is determined that the time length of the receive pin is long or short. If the length is tAB ≤ tCD, the serial operation mode is determined as UART mode. The time of tAD is used whether the automatic baud rate setting is enable or not. If the length is tAB > tCD, the serial operation mode is determined as I/O Interface mode. Note that timer values of

tAB, tAC and tAD have a margin of error. If the baud rate is high and operation frequency is low, each timer value becomes small. This may generates unexpected determination occurs. (To prevent this problem, re-set UART within the programming routine.)

For example, 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 where the time is expected to receive an echo-back (0x86) from the target board. The controller should give up the communication if it fails to get that echo-back within the allowed time. 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, it is not necessary that the first byte is 0x30 as long as tAB >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>tCD is established and SIO is selected by the resolution of the operation mode determination, the second byte code is 0x30 even though the transmitted code on the first byte is not 0x30 (The first byte code to determine I/O interface mode is described as 0x30).

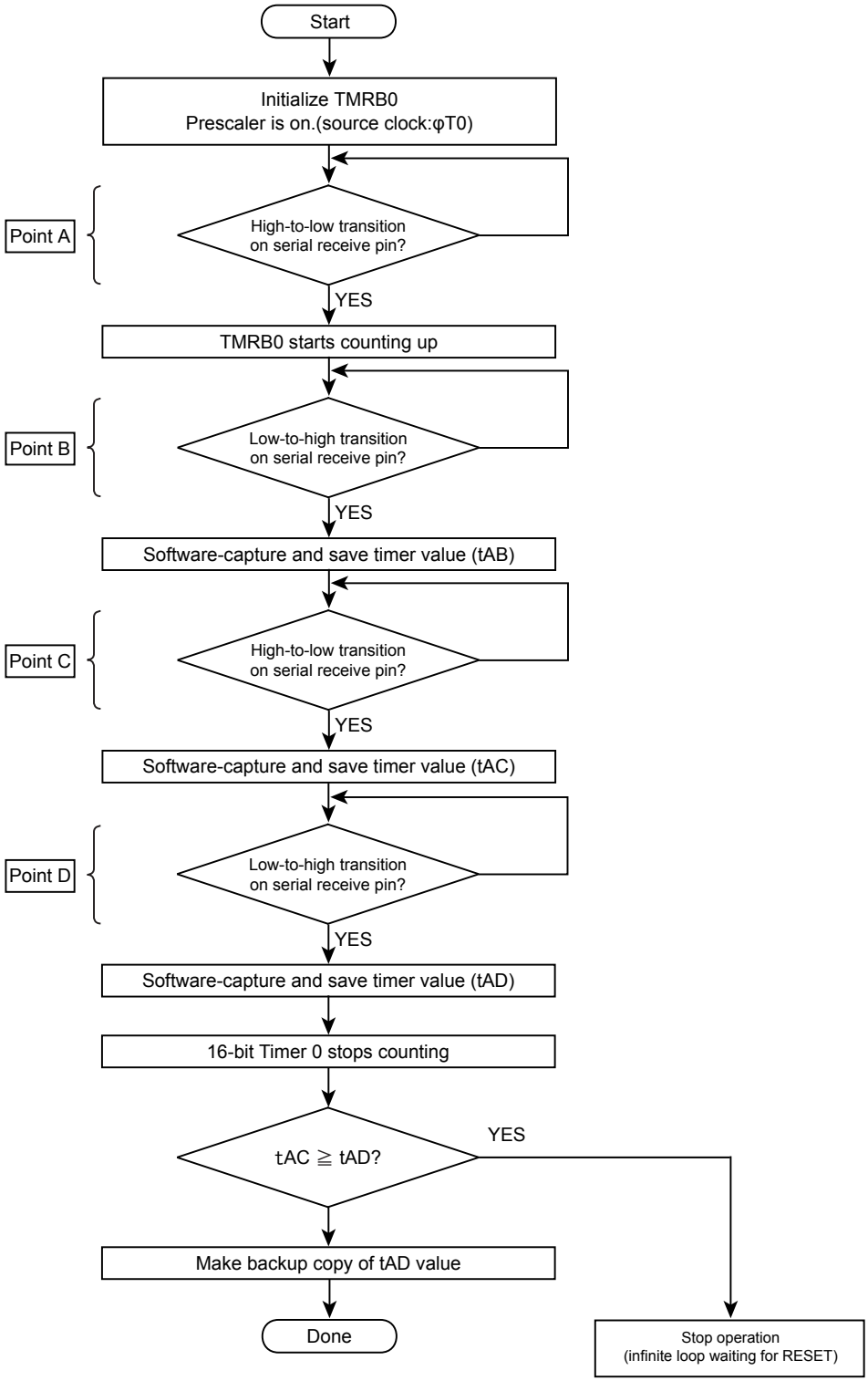


Figure 20-7 Serial operation mode receive flowchart

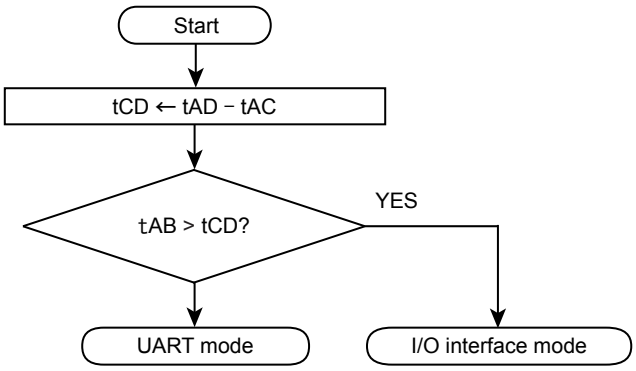


Figure 20-8 Serial operation mode determination flowchart

20.3.5.2 Acknowledge Response Data

The boot program represents processing states in specific codes and sends them to the controller. Table 20-13 to Table 20-16 show the values of acknowledge responses to each receive data.

In Table 20-14 to Table 20-16, the upper four bits of the acknowledge response are equal to those of the operation command data. The 3rd bit indicates a receive error. The 0th bit indicates an invalid operation command error, a checksum error or a password error. The 1st bit and 2nd bit are always "0". Receive error checking is not performed in I/O Interface mode.

Table 20-13 ACK response to the serial operation determination data

Transmit data	Description
0x86	Determined that UART communication is possible. (Note)
0x30	Determined that I/O interface communication is possible.

Note: When the serial operation is determined as UART, if the baud rate setting is determined as unacceptable, the boot program aborts without sending back any response.

Table 20-14 ACK response to the operation command data

Transmit data	Description
0x?8 (Note)	A receive error occurrs in the operation command data
0x?1 (Note)	An undefined operation command data is received normally.
0x10	Determined as a RAM transfer command
0x40	Determined as a flash memory chip erase command

Note: The upper 4 bits of the ACK response data are the same as those of the previous command data.

Table 20-15 ACK response to the CHECK SUM data

Transmit data	Description
0xN8 (Note)	A receive error occurs.
0xN1 (Note)	A CHECK SUM or a password error occurs.
0xN0 (Note)	The CHECK SUM value is correct.

Note: The upper 4 bits of the ACK response data are the same as those of the operation command data.

Table 20-16 ACK response to Flash memory chip erase and protect bit erase operation

Transmit data	Description
0x54	Determined as a erase enable command
0x4F	Erase command is complete.
0x4C	Erase command is abnormally terminated.

20.3.5.3 Password Determination

The boot program use the below area to determine whether a password is required or use as a password.

Area	Address
Password requirement determination	0x3F81_FFF0 (1byte)
Password area	0x3F81_FFF4 to 0x3F81_FFFF (12byte)

The RAM Transfer command performs a password verification regardless of necessity judging data. Flash memory chip erase or protect bit erase command performs a password verification only when necessity judging is determined as "required".

Password requirement setting	Data
Need password	Other than 0xFF
No password	0xFF

If a password is set to 0xFF (erased data), it is difficult to protect data securely due to an easy-to-guess password. Even if Single Boot mode is not used, it is recommended to set a unique value as a password.

(1) Password verification using RAM transfer command

If all these address locations contain the same bytes of data other than 0xFF, this condition is determined as a password area error as shown in Figure 20-9. In this case, the boot program returns an error acknowledge (0x11) in response to the 17th byte of checksum value regardless of the password verification.

The boot program verifies 5th byte to 16th byte of receive data (password data). A password error occurs if all 12 bytes do not match. If the password error is determined, an ACK response data to the 17th of CHECK SUM data is a password error.

The password verification is performed even if the security function is enabled.

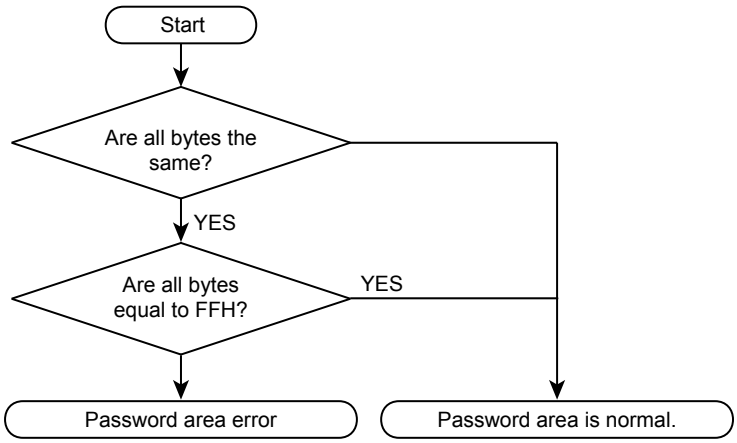


Figure 20-9 Password area check flowchart

(2) Password verification to Flash memory chip erase and protect bit erase command

When a password is enable in the erase password necessity determination area as shown in Figure 20-10 and the passwords are identical data, a password area error occurs. If a password area error is determined, an ACK response to the 17th byte of CHECK SUM sends 0x41 regardless of the password verification.

The boot program verifies 5th byte to 16th byte of receive data (password data). A password error occurs if all 12 bytes do not match. If the password error is determined, an ACK response data to the 17th of CHECK SUM data is a password error.

The password verification is performed even if the security function is enabled.

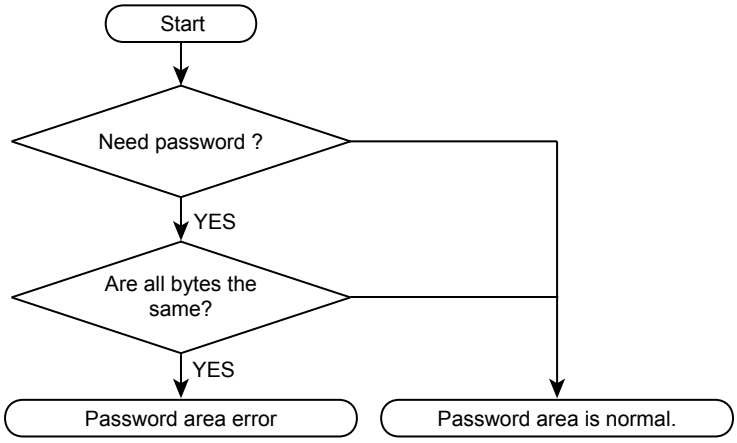


Figure 20-10 Password area check flowchart

20.3.5.4 CHECK SUM Calculation

The checksum is calculated by 8-bit addition to transmit data, dropping the carries, and taking the two's complement of the total sum. The controller must perform the same checksum operation in transmitting checksum bytes.

Example of CHECK SUM

To calculate the checksum for a series of 0xE5 and 0xF6, perform 8-bit addition.

0xE5 + 0xF6 = 0x1DB

Take the two's complement of the sum to the lower 8-bit, and that is a checksum value. So the boot program sends 0x25 to the controller.

0 − 0xDB = 0x25

20.3.6 Transfer Format at RAM Transfer

This section shows a RAM transfer command format. Transfer directions in the table are indicated as follows:

Transfer direction (C→T): Controller to TMPM061FWFG

Transfer direction (C←T): TMPM061FWFG to Controller

Number of transfer bytes	Transfer direction	Transfer data	Description
1	C→T	Serial operation mode and baud rate setting	Sends data to determine the serial operation mode. For detail of mode determination, refer to "20.3.5.1 Serial Operation Mode Determination".
		[UART mode] 0x86	Sends 0x86. If UART mode is determined, the program determines whether a baud setting is possible. If not, the program stops and communication is shutdown.
		[I/O interface mode] 0x30	Sends 0x30 at 1/6 of desired baud rate.The 2nd byte is also sent at 1/6 of desired baud rate. From the 3rd byte or later, you can send the data at a desirable baud rate.
2	C←T	ACK response to serial operation mode	The 2nd byte of transmit data is a ACK response data to the 1st byte that corresponds to the serial operation setting mode data. If the setting is possible, sets SIO/UART. A receive enable timing is set before transmit buffer is written to the data.
		[UART mode] Normal state: 0x86	If the setting is determined to be possible, sends 0x86. If not, the operation aborts without sending back any response. When the controller finished to send the 1st byte of data, requires a time-out time (5 seconds). If data (0x86) is not normally received within a time-out time, communication is not possible.
		[I/O interface mode] Normal state:0x30	Writes data (0x30) to the transmit buffer and waits for SCLK0 clock. When the controller finished to send the 1st byte of data and several ms (idle time) later, outputs SCLK clock. At this time, the baud rate is set to 1/16 of desired baud rate. If receive data is 0x30, communications are possible. After the 3rd byte, sets a desired baud rate to communicate.
3	C→T	Operation command data (0x10)	Sends RAM transfer command data (0x10).
4	C←T	ACK response to operation command Normal state: 0x10 Abnormal state: 0xX1 Communication error: 0xX8	ACK response data to the operation command. First, checks if 3rd byte of receive data has errors. (UART mode only) If receive errors exist, sends a ACK response data 0xX8 that means abnormal communications and waits for a next operation command (3rd byte). Upper 4 bits of transmit data are undefined. (same as upper 4 bits of immediately before operation command.) Note that in the I/O interface, receive error check is not performed. Then, if the 3rd byte of receive data corresponds to either operation command data in Table 20-12, receive data is echoed back. In the case of RAM transfer, 0x10 is echoed back and the transfer data branches to the RAM transfer service routine. If the data does not correspond to the command in Table 20-12, sends a ACK response data 0xX1 that means operation command errors, and waits for next operation command (3rd byte). Upper 4 bits of transmit data are undefined. (same as upper 4 bits of immediately before operation command data.)
5 to 16	C→T	Password data (12-byte) 0x3F81_FF04 to 0x3F81_FF0F	Checks data in the password area. For detail of password area checking, refer to "20.3.5.3 Password Determination". Compares 5th to 16th byte of receive data with 0x3F81_FFF0 to 0x3F81_FFFF of data of Flash memory. If the data does not match the address, a password error flag is set.

Number of transfer bytes	Transfer direction	Transfer data	Description
17	C→T	5th to 16th byte of CHECK SUM values	Send 5th to 16th byte of CHECK SUM values. For detail of CHECK SUM calculation, refer to 20.3.5.4.
18	C←T	ACK response to CHECK SUM value Normal state: 0x10 Abnormal state: 0x11 Communication error: 0x18	First, checks if 5th to 17th byte of receive data have errors.(UART mode only) If receive errors exist, sends a ACK response data 0x18 that means abnormal communications and waits for a next operation command (3rd byte). Then checks 17th byte of CHECK SUM data. If errors exist, sends 0x11 and waits for a next operation command (3rd byte). Finally, checks the result of password verification. If a password error exists, sends a ACK response data 0x11 that means a password error and waits for a next operation command (3rd byte). If all procedure normally ends, sends a normal ACK response data 0x10.
19	C→T	RAM store start Address 31 to 24	Sends a start address of block transfer for RAM store. The 19th byte corresponds to 31st to 24th bit of address.The 22nd byte corresponds to 7th to 0th bit of address. Specify the address to the address 0x2000_0400 through the last address of RAM. The address must be even address.
20	C→T	RAM store start Address 23 to 16	
21	C→T	RAM store start Address 15 to 8	
22	C→T	RAM store start Address 7 to 0	
23	C→T	Number of RAM store bytes 15 to 8	Set the number of bytes to perform block transfer. The 23rd byte corresponds to the15th bit to 8th bit of transfer bytes. The 24th byte corresponds to 7th bit to 0th bit of transfer bytes. Specify the data to be stored in the address from 0x2000_0400 through the last address of RAM.
24	C→T	Number of RAM store bytes 7 to 0	
25	C→T	19th to 24th byte of CHECK SUM value	Send 19th byte to 24th byte of CHECK SUM values
26	C←T	ACK response to CHECK SUM value Normal state: 0x10 Abnormal state: 0x11 Communication error: 0x18	First, checks if 19th byte to 25th byte of receive data have errors.(UART mode only) If receive errors exist, sends a ACK response data 0x18 that means abnormal communications and waits for a next operation command (3rd byte). Then checks 25th byte of CHECK SUM data. If errors exist, sends 0x11 and waits for a next operation command (3rd byte). If all procedure normally ends, sends a normal ACK response data 0x10.
27 to m	C→T	RAM stored data	Sends same bytes of data specified in 23th bytes to 24 byte for RAM stored data.
m+1	C→T	27 to m byte of CHECK SUM value	Sends 27th byte to m byte of CHECK SUM value
m+2	C←T	ACK response to CHECK SUM value Normal state:0x10 Abnormal state: 0x11 Communication error: 0x18	First, checks if 27th byte to m+1 byte of receive data have errors. (UART mode only) If receive errors exist, sends a ACK response data 0xX8 that means abnormal communications and waits for a next operation command (3rd byte). Then checks m+1 byte of CHECK SUM data, if errors exist, sends 0x11 and waits for a next operation command (3rd byte). If all procedure normally ends, sends a normal ACK response data 0x10.
–	–	–	If m + 2nd byte of ACK response data is normal ACK response data, the transfer data branches to the address specified in 19th byte to 22 byte.

20.3.7 Transfer Format of Flash memory Chip Erase and Protect Bit Erase

This section shows a transfer format of Flash memory chip erase and protect bit erase commands. Transfer directions in the table are indicated as follows:

Transfer direction (C→T): Controller to TMPM061FWFG

Transfer direction (C←T): TMPM061FWFG to Controller

Number of transfer bytes	Transfer direction	Transfer data	Description
1	C→T	Serial operation mode and baud rate setting	Sends data to determine the serial operation mode. For detail of mode determination, refer to"20.3.5.1 Serial Operation Mode Determination".
		[UART mode] 0x86	Sends 0x86. If UART mode is determined, checks if baud rate setting can be done. If not, operation stops communications.
		[I/O interface mode] 0x30	Sends 0x30 at 1/16 of desired baud rate. Same as the 1st byte, sends the 2nd byte at 1/16 of desired baud rate. Desired baud rate can be used after 3rd byte.
2	C←T	ACK response to serial operation mode	The 2nd byte of transmit data is a ACK response data to the 1st byte that corresponds to the serial operation setting mode data. If the setting is possible, sets SIO/UART. A receive enable timing is set before transmit buffer is written to the data.
		[UART mode] Normal state: 0x86	If the setting is determined to be possible, sends 0x86. If not, the operation aborts without sending back any response. When the controller finished to send the 1st byte of data, requires a time-out time (5 seconds). If data (0x86) is not normally received within a time-out time, communication is not possible.
		[I/O interface mode] Normal state:0x30	Writes data (0x30) to the transmit buffer and waits for SCLK0 clock. When the controller finished to send the 1st byte of data and several ms (idle time) later, outputs SCLK clock. At this time, the baud rate is set to 1/16 of desired baud rate. If receive data is 0x30, communications are possible. After the 3rd byte, sets a desired baud rate to communicate.
3	C→T	Operation command data (0x40)	Sends Flash memory chip erase and protect bit erase command data (0x40).
4	C←T	ACK response to the operation command Normal state: 0x40 Abnormal state: 0xX1 Communication error: 0xX8	ACK response data to the operation command. First, checks if 3rd byte of receive data has errors. (UART mode only) If receive errors exist, sends a ACK response data 0xX8 that means abnormal communications and waits for a next operation command (3rd byte). Upper 4 bits of transmit data are undefined. (same as upper 4 bits of immediately before operation command data.) Note that in the I/O interface, receive error check is not performed. Then, if the 3rd byte of receive data corresponds to either operation command data in Table 20-12, receive data is echoed back. If the data does not correspond to the command in Table 20-12, sends a ACK response data 0xX1 that means operation command errors, and waits for next operation command. (3rd byte) Upper 4 bits of transmit data are undefined. (Upper 4 bits of immediate before operation command data are used.)
5 to 16	C→T	Password data (12-byte) 0x3F81_FF04 to 0x3F81_FF0F	If password necessity is set to "none", this data is dummy data. If password necessity is set to "necessary", checks data in the password area. For a method of password area data checking, refer to "20.3.5.3 Password Determination". Compares 5th to 16th byte of receive data with 0x3F81_FFF0 to 0x3F81_FFFF of data of Flash memory in order. If the data does not match, a password error flag is set.
17	C→T	5 to 16 th byte CHECK SUMvalue	Sends 5th byte to 16 byte of CHECK SUM value. For a method of CHECK SUM calculation, refer to"20.3.5.4 CHECK SUM Calculation".

Number of transfer bytes	Transfer direction	Transfer data	Description
18	C←T	ACK response to the CHECK SUM value Normal state: 0x40 Abnormal state: 0x41 Communication error: 0x48	If password necessity is set to "none", sends a normal ACK response data 0x40. If password necessity is set to "necessary", first checks if receive errors exist in the 5th byte to 17th byte receive data. (UART mode only) If a receive error exists, sends a ACK response data 0x48 that means abnormal communications and waits for next operation command. (3rd byte) Then checks 17th byte of CHECK SUM data. If error occurs, sends 0x41 and waits for a next operation command (3rd byte) Finally, checks the result of password verification. If a password error exists, sends a ACK response data 0x41that means a password error and waits for a next operation command (3rd byte) If all procedure normally ends, sends a normal ACK response data 0x40.
19	C→T	Erase enable command data (0x54)	Sends an enable command data (0x54).
20	C←T	ACK response to the erase enable command Normal state: 0x54 Abnormal state: 0xX1 Communication error: 0x58	First, checks if 19th byte of receive data has errors. If receive errors exist, sends a ACK response data (bit 3) 0x58 that means abnormal communication and waits for next operation command (3rd byte). Then, if 19th byte of receive data corresponds to the erase enable command, receive data is echoed back (normal ACK response data). In this case, 0x54 is echoed back and the transfer data branches into Flash memory chip erase process routine. If the data does not correspond to the erase enable command, sends a ACK response data (bit 0) 0xX1 and waits for next operation command. Upper 4 bits of transmit data are undefined. (Upper 4 bits of immediate before operation command data are used.)
21	C→T	ACK response to the erase command Normal state: 0x4F Abnormal state: 0x4C	If the operation is normally complete, the end code (0x4F) is returned. If erase error occurs, an error code (0x4C) is returned.
–	–	–	Waits for a next operation command.

20.3.8 Boot Program Whole Flowchart

This section shows a boot program whole flowchart.

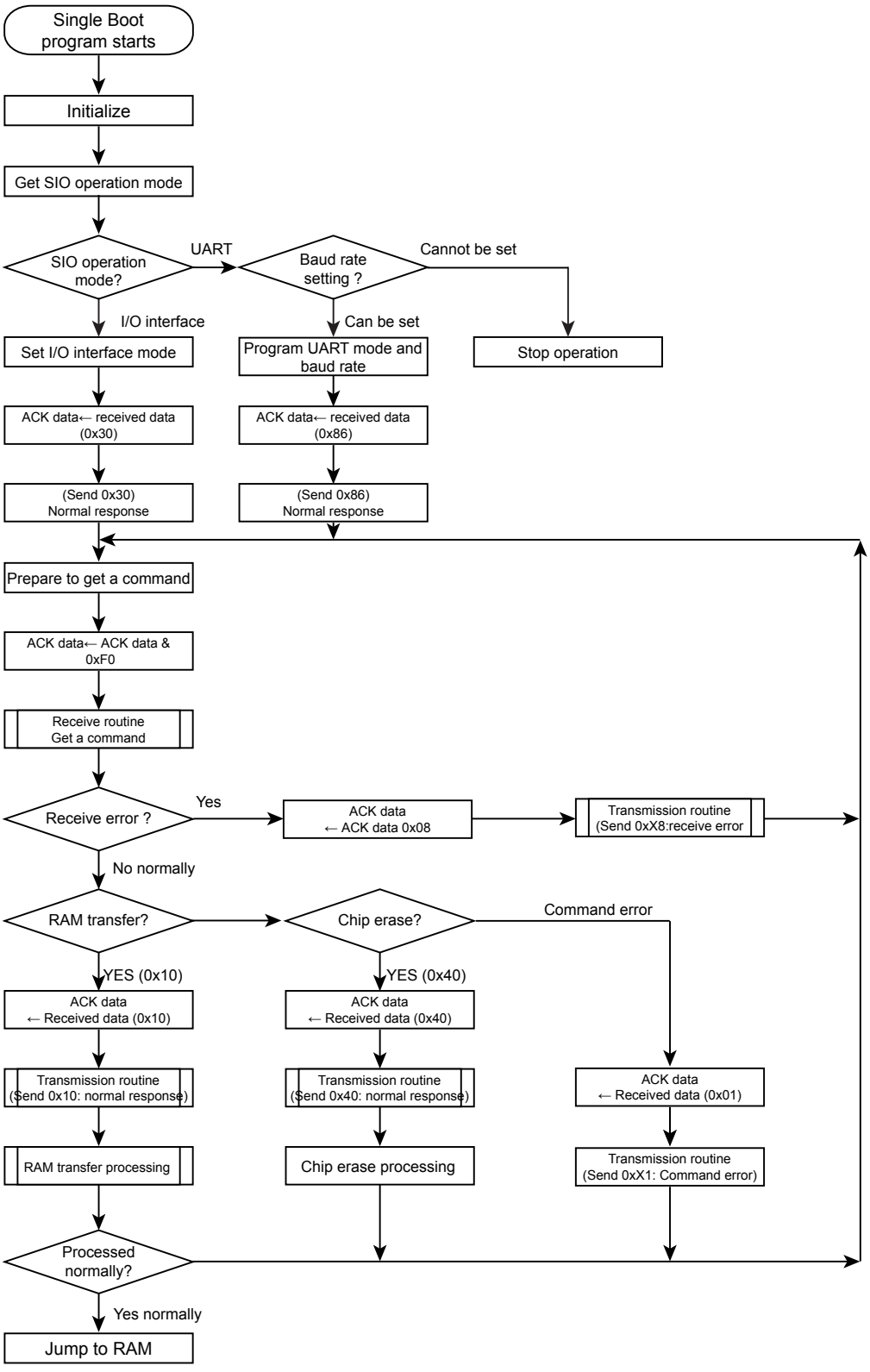


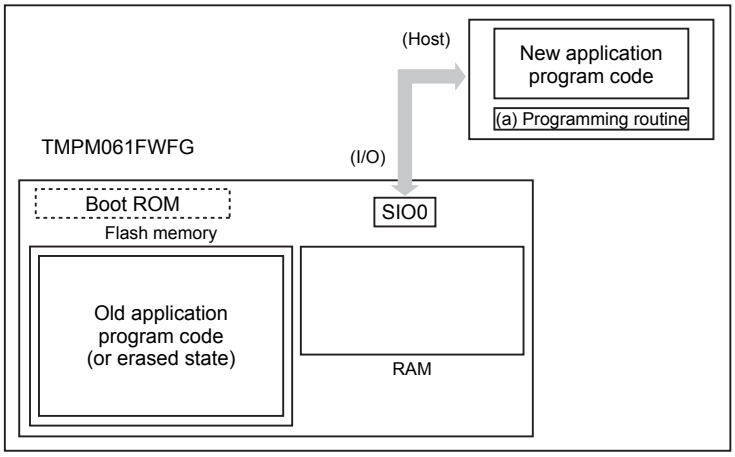
Figure 20-11 Boot program whole flowchart

20.3.9 Reprogramming Procedure of Flash using reprogramming algorithm in the on-chip BOOT ROM

This section describes the reprogramming procedure of the flash using reprogramming algorithm in the on-chip boot ROM.

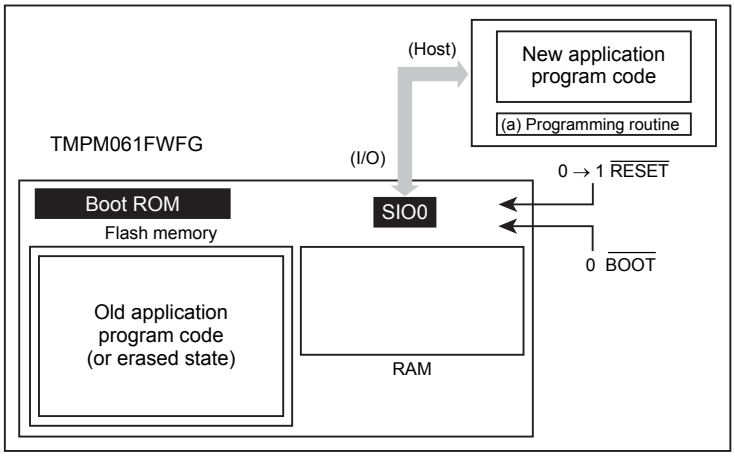
20.3.9.1 Step-1

The condition of Flash memory does not care whether a user program made of former versions has been written or erased. Since a programming routine and programming data are transferred via the SIO (SIO0), the SIO0 must be connected to an external host. A programming routine (a) is prepared on the host.



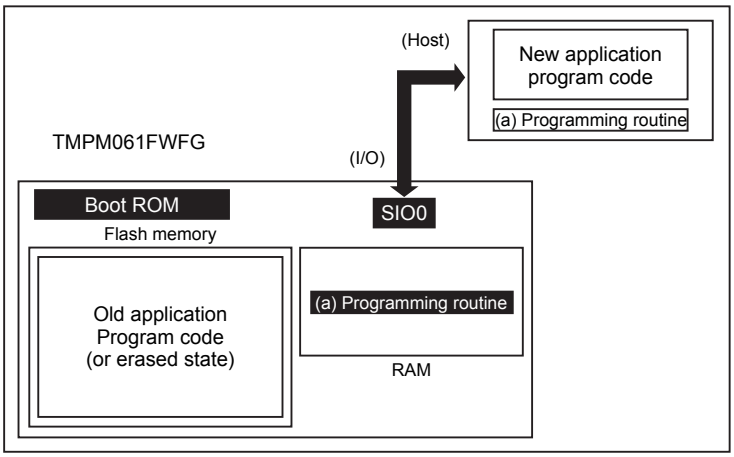
20.3.9.2 Step-2

Release the reset by pin condition setting in the boot mode and boot-up the BOOT ROM. According to the procedure of boot mode, transfer the programming routine (a) via SIO0 from the source (host). A password verification with the the password in the user application program is perfomed. (If Flash memory is erased, an erase data (0xFF) is dealt with a password.)



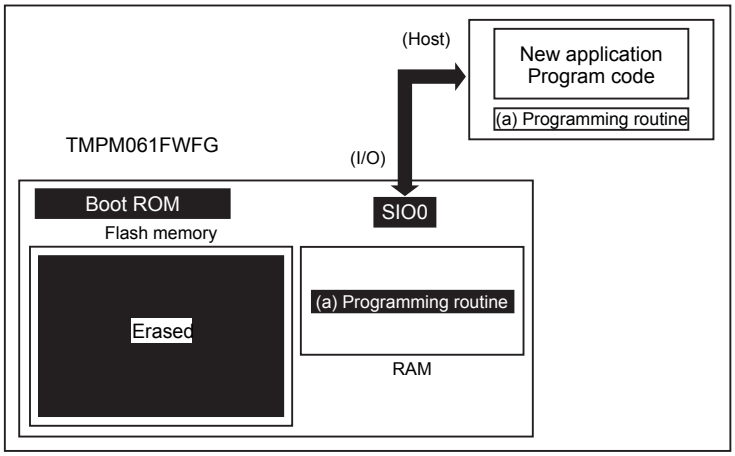
20.3.9.3 Step-3

If the password verification is complete, the boot program transfer a programming routine (a) from the host into the on-chip RAM. The programming routine must be stored in the range from 0x2000_0400 to the end address of RAM.



20.3.9.4 Step-4

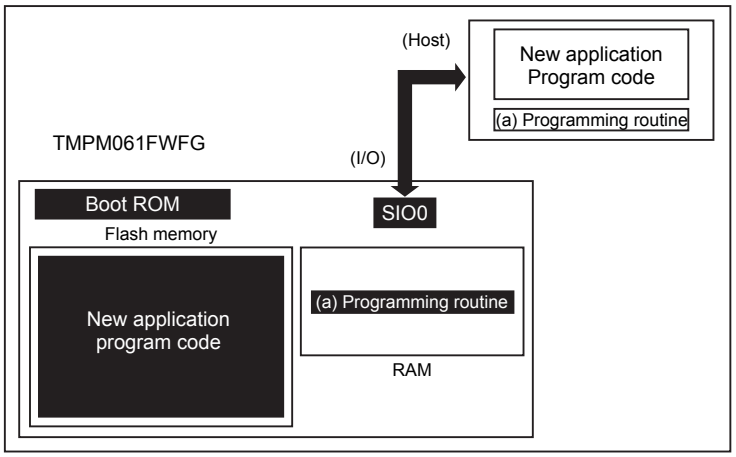
The boot program jumps to the programming routine (a) in the on-chip RAM to erase the flash block containing old application program codes. The Block Erase or Chip Erase command is be used.



20.3.9.5 Step-5

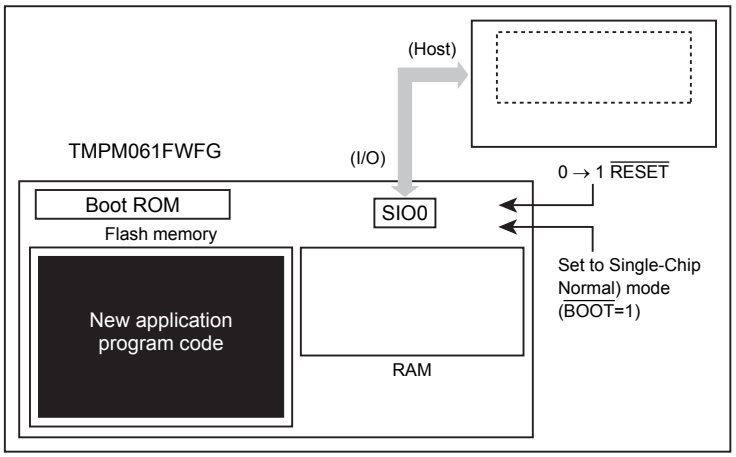
The boot program executes the programming routine (a) to download new application program codes from the host and programs it into the erased flash area. When the programming is complete, the writing or erase protection of that flash area in the user's program must be set.

In the example below, new program code comes from the same host via the same SIO0 channel as for the programming routine. However, once the programming routine has begun to execute, it is free to change the transfer path and the source of the transfer. Create a hardware board and programming routine to suit your particular needs.



20.3.9.6 Step-6

When programming of Flash memory is complete, power off the board and disconnect the cable leading from the host to the target board. Turn on the power again so that the device re-boots in the single-chip (Normal) mode to execute the new program.



20.4 Programming in the User Boot Mode

A user Boot mode is to use flash memory programming routine defined by users. It is used when the data transfer buses for flash memory program code on the user application is different from the serial I/O. It operates in the single chip mode; therefore, a switch from normal mode in which user application is activated in the use boot mode to the user boot mode for programming flash is required. Specifically, add a mode judgment routine to the reset service routine in the user application program.

The condition to switch the modes needs to be set according to the user's system setup condition. Also, a flash memory programming routine that the user uniquely makes up needs to be set in the new application. This routine is used for programming after being switched to the user boot mode. The data in built-in Flash memory cannot be read out during erase/reprogramming mode. Thus, reprogramming routine must be take place while it is stored in the area outside of Flash memory area. Once re-programming is complete, it is recommended to protect relevant flash blocks from accidental reprogramming. Be sure not to generate interrupt/fault to avoid abnormal termination during the user boot mode.

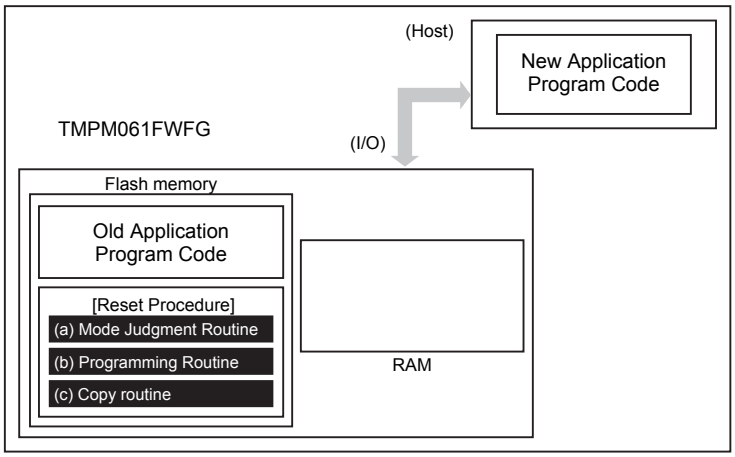
Taking examples from two cases such as the method that reprogramming routine stored in Flash memory (1-A) and transferred from the external device (1-B), the following section explains the procedure. For a detail of the program/erase to Flash memory, refer to "20.2 Detail of Flash Memory".

20.4.1 (1-A) Procedure that a Programming Routine Stored in Flash memory

20.4.1.1 Step-1

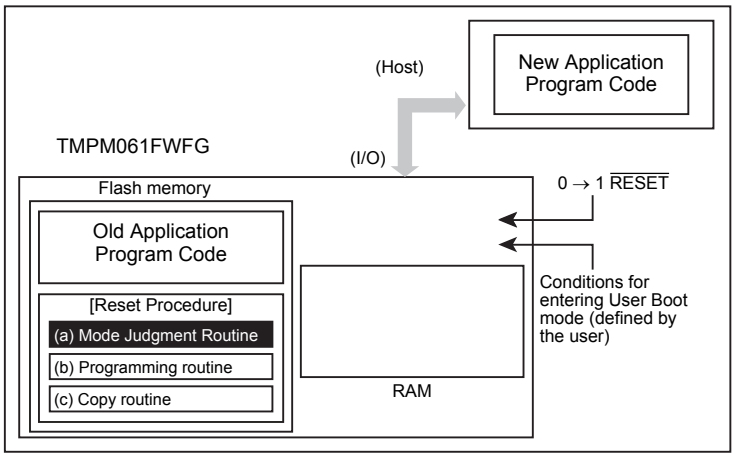
A user determines the conditions (e.g., pin status) to enter the user boot mode and the I/O bus to be used to transfer data. Then suitable circuit design and program are created. Before installing the device on a printed circuit board, write the following three program routines into an arbitrary flash block using programming equipment such as a flash writer.

(a) Mode determination routine:	A program to determine to switch to user boot mode or not
(b) Flash programming routine:	A program to download new program from the host controller and re-program Flash memory
(c) Copy routine:	A program to copy the data described in (a) to the built-in RAM or external memory device



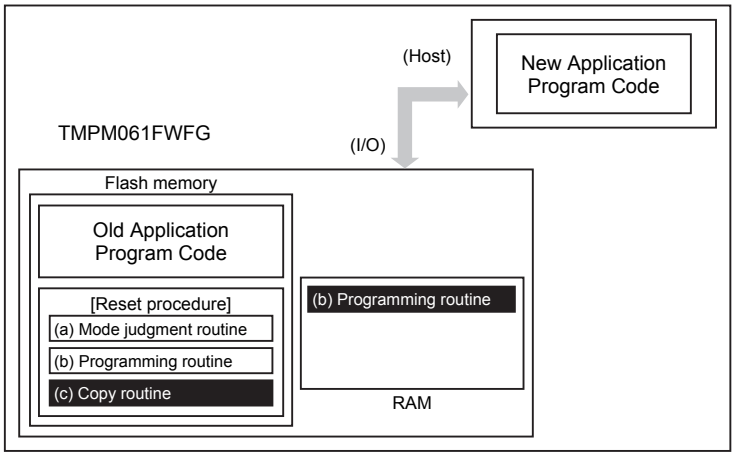
20.4.1.2 Step-2

This section explains the case that a programming routine storied in the reset routine. First, the reset routine determines to enter the user boot mode. If mode switching conditions are met, the device enters the user boot mode to reprogram data.



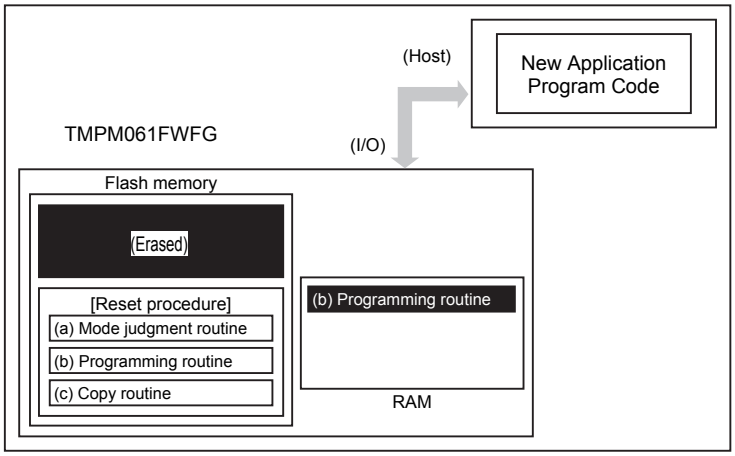
20.4.1.3 Step-3

Once the device enters the user boot mode, execute the copy routine (C) to download the flash programming routine (b) from the host controller to the built-in RAM .



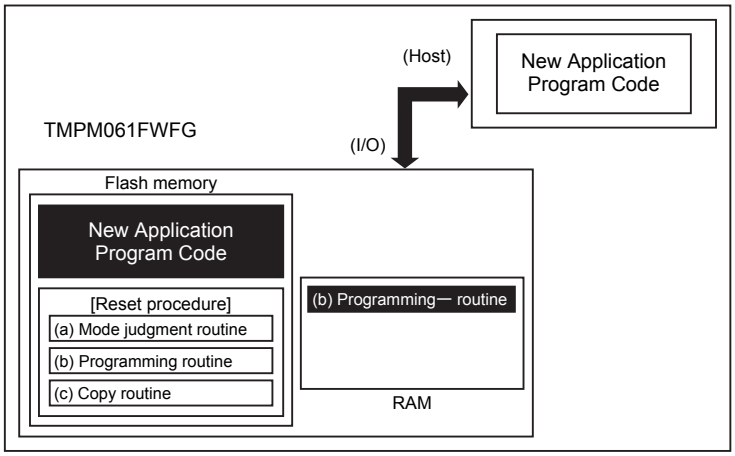
20.4.1.4 Step-4

Jump to the the reprogramming routine in the built-in RAM to release the write/erase protection for the old application program, and to erase a flash in block unit.



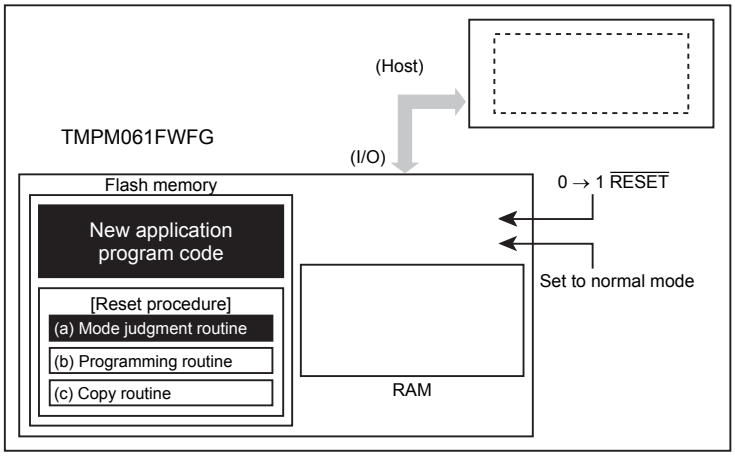
20.4.1.5 Step-5

Continue to execute the flash programming routine to download new program data from the host controller and program it into the erased flash block. When the programming is complete, the write/erase protection of that flash block in the user program area must be set.



20.4.1.6 Step-6

Set $\overline{\text{RESET}}$ to "0". Upon reset, Flash memory is set to the normal mode. After reset, the CPU will start along with the new application program.



20.4.2 (1-B) Procedure that a Programming Routine is transferred from External Host

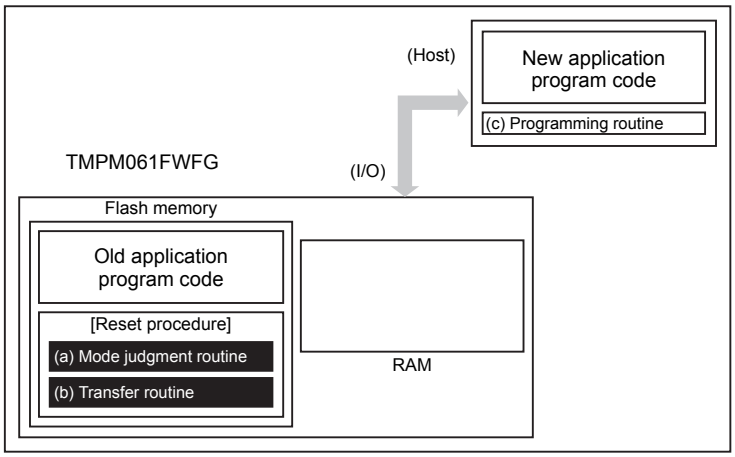
20.4.2.1 Step-1

A user determines the conditions (e.g., pin status) to enter the user boot mode and the I/O bus to be used to transfer data. Then suitable circuit design and program are created. Before installing the device on a printed circuit board, write the following two program routines into an arbitrary flash block using programming equipment such as a flash writer.

(a) Mode determination routine: A program to determine to switch to reprogramming operation
(b) Transfer routine: A program to obtain a reprogramming program from the external device.

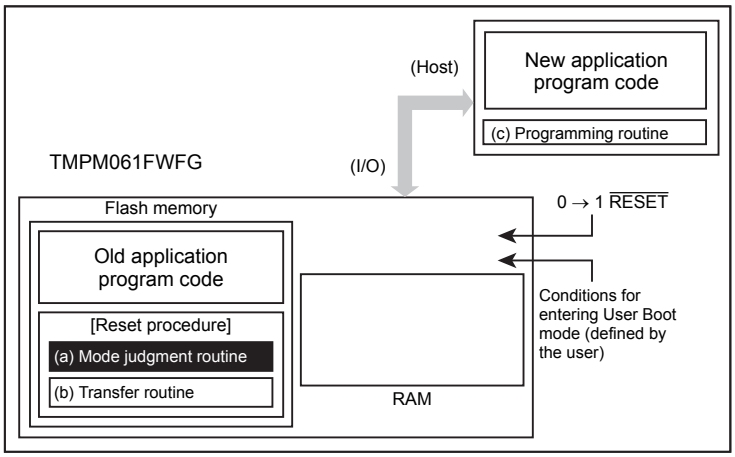
In addition, prepare a reprogramming routine shown below must be stored on the host controller.

(c) Reprogramming routine: A program to reprogram data



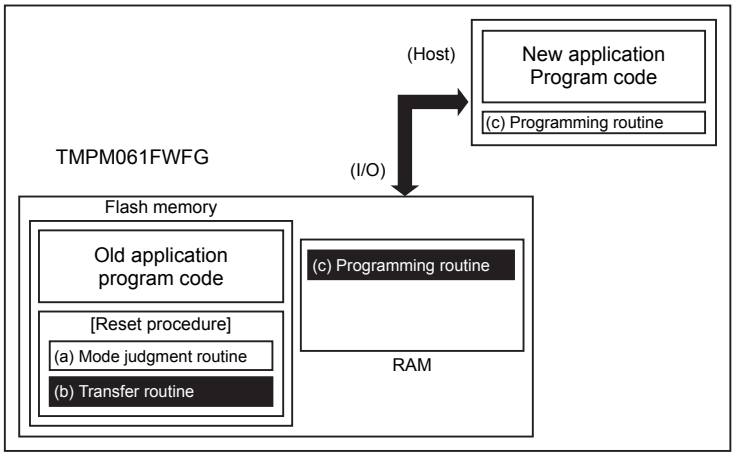
20.4.2.2 Step-2

This section explains the case that a programming routine storied in the reset routine. First, the reset routine determines to enter the user boot mode. If mode switching conditions are met, the device enters the user boot mode to reprogram data.



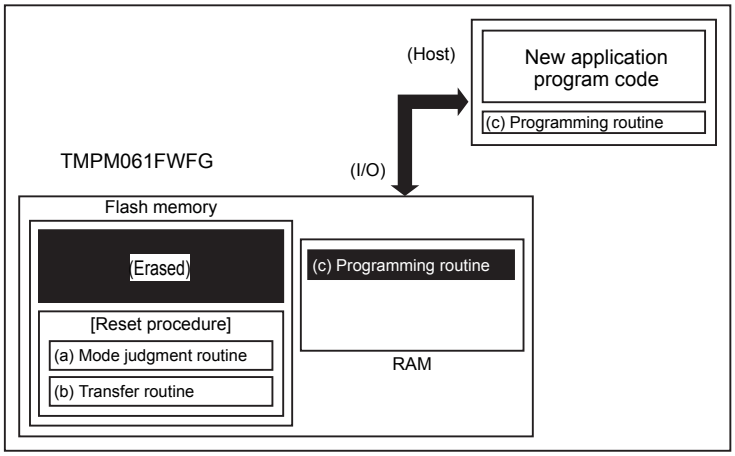
20.4.2.3 Step-3

Once the device enters the user boot mode, execute the transfer routine (b) to download the programming routine (c) from the host controller to the built-in RAM .



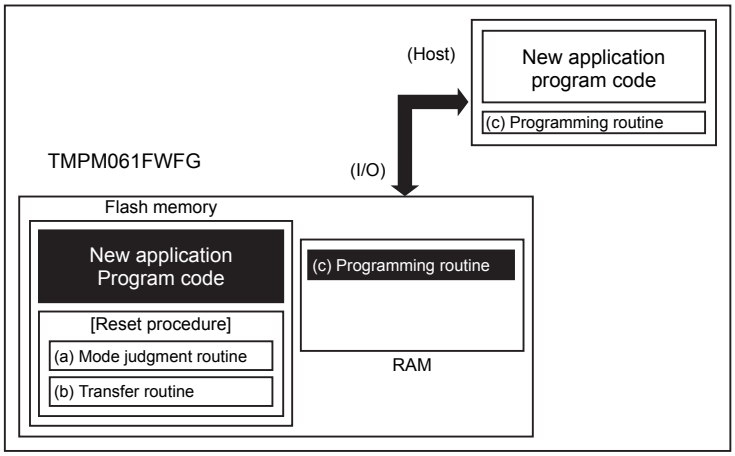
20.4.2.4 Step-4

Jump to the the reprogramming routine in the built-in RAM to release the write/erase protection for the old application program, and to erase a flash in block unit.



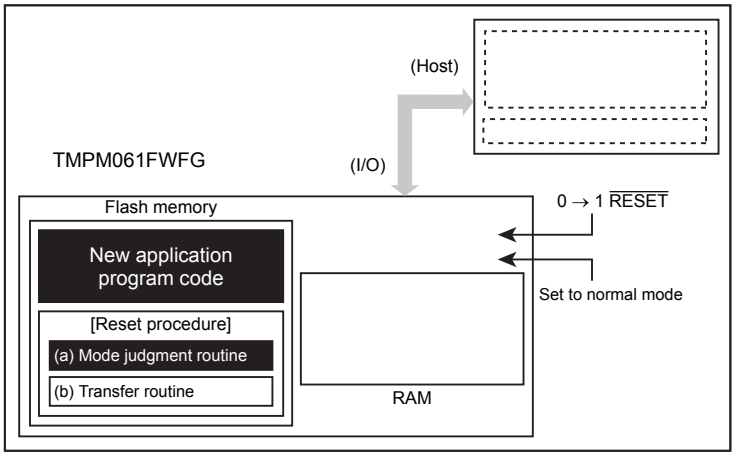
20.4.2.5 Step-5

Continue to execute the flash programming routine (c) to download new program data from the host controller and program it into the erased flash block. When the programming is complete, the write/erase protection of that flash block in the user program area must be set.



20.4.2.6 Step-6

Set $\overline{\text{RESET}}$ to "0". Upon reset, Flash memory is set to the normal mode. After reset, the CPU will start along with the new application program.



21. Debug Interface

21.1 Specification Overview

TMPM061FWFG contains the Serial Wire Debug Port (SW-DP) units as the interface with the debugging tools. SW-DP supports the Serial Wire Debug Port (SWCLK, SWDIO).

For details about SW-DP, refer to the Arm manual "Cortex-M0 Technical Reference Manual".

21.2 Pin Functions

The debug interface pins can also be used as general-purpose ports.

After reset, the debug interface pins are configured as debug port function pins. To use the debug interface pins as other function, change of setting is required.

Table 21-1 shows the settings of the debug interface pins after reset.

Table 21-1 Debug Interface Pins and Related Port Settings after Reset

Debug Function	Value of Related port settings after reset				
	Function (PxFR)	Input (PxIE)	Output (PxCR)	Pull-up (PxPUP)	Pull-down (PxPDN)
SWCLK	1	1	0	0	1
SWDIO	1	1	1	1	0

21.3 Reset Vector Break

TMPM061FWFG is prohibited from transmission with debug tools while reset caused by $\overline{\text{RESET}}$ pin is effective. When setting a break by using reset vector, set the following procedure after reset; set break points from the debug tools, then set the application interrupt and the <SYSRESETREQ> bit of the reset control register to reset again.

21.4 Precautions on Use of Debug Interface Pin Used as General-purpose Port

If a debug interface pin is set to a general-purpose port by a user program after reset, then debug tools cannot control a MCU. In order to connect a debug tool to MCU again, a certain mechanism in which a general-purpose port can be changed to a debugging interface setting must be prepared by a user.

21.5 Debug Enable Pin

TMPM061FWFG prepares debug enable pins ($\overline{\text{DBGEN}}$). These debug enable pins are used for reliable communication with a debug tool.

Debug Enable pins are valid to be input during reset. If "Low" signal is sampled on the rising edge of reset signal, a debug enable condition is set and its function is fixed to the interface. This condition remains until next pin reset.

A debug tool communicates with a processor core after reset. This is effective when a user program changes a port setting of debug interface shortly after reset.

Below register can monitor the debug conditions.

Base Address = 0x41FF_F000

Register name		Address(Base+)
Debug enable monitor register	FCDBGEN	0x005C

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	-	EN0
After reset	1	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-1	−	R	Read as "0".
0	EN0	R	Monitors a debug interface pin 0: Functions of SWCLK0 and SWDIO0 can be changed. 1: Functions of SWCLK0 and SWDIO0 cannot be changed. After reset, if $\overline{\text{DBGEN0}}$ pin is "Low", "1" is set.

21.6 Peripheral Functions in Halt Mode

When Cortex-M0 core enters in the halt mode, the watchdog-timer (WDT) automatically stops. It is selectable that 16-bit Timer (TMRB and TMR16A) continue or stop counting. Other peripherals are continue operating.

21.7 About connection with debug tool

Concerning a connection with debug tools, refer to manufactures recommendations.

Debug interface pins contain a pull-up resistor and a pull-down resistor. When debug interface pins are connected with external pull-up or pull-down, please pay attention to input level.

22. Electrical Characteristics

22.1 Absolute Maximum Ratings

Parameter		Symbol	Rating	Unit
Supply voltage		DVDD3	−0.3 to 3.9	V
		AVDD3	−0.3 to 3.9	
		RVDD3	−0.3 to 3.9	
		DSRVDD3	−0.3 to 3.9	
		SRVDD3	−0.3 to 3.9	
		VLC	-0.3 to 3.9	
Capacitor voltage		COUT	−0.3 to 3.0	V
Input voltage	Other than pins below	V_{IN}	−0.3 to DVDD3 + 0.3	V
	PI5, PI6, PJ2		−0.3 to 5.5	
	DAIN0+, DAIN0-, DAIN1+, DAIN1-, DAIN2+, DAIN3-		−0.375 to DSRVDD3 + 0.3	
Low-level output current	Per pin	I_{OL}	5	mA
	Total	ΣI_{OL}	50	
High-level output current	Per pin	I_{OH}	−5	
	Total	ΣI_{OH}	50	
Power consumption (Ta = 85 °C)		PD	600	mW
Soldering temperature(10 s)		T_{SOLDER}	260	°C
Storage temperature		T_{STG}	−40 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.

22.2 DC Electrical Characteristics (1/3)

Ta = −40 to 85 °C

Parameter		Symbol	Condition		Min	Typ. (Note 1)	Max	Unit
Supply voltage	DVDD3 = AVDD3 = RVDD3 = DSRVDD3 = RVDD3 = VLC (note2) DVSS = AVSS = RVSS = DSRVSS = 0V	DVDD3 AVDD3 RVDD3 DSRVDD3 VLC	f_{OSC} = 8 to 16 MHz fsys = 1 to 16 MHz fs = 30 to 34 kHz		1.8	−	3.6	V
		SRVDD			2.9	−	3.6	V
Low-level input voltage	PF0,PF1	V_{IL1}	1.8 V ≤ AVDD3 ≤ 3.6 V		−0.3	−	0.2 AVDD3	V
	PA0 to 7, PB0 to 7, PC0 to 7, PD0 to 7, PE0 to 7, PF0 to 1, PG0, PH0 to 5, PI0 to 6, PJ0 to 5, PK0 to 1 $\overline{\text{RESET}}$, MODE	V_{IL2}	1.8 V ≤ DVDD3A ≤ 3.6 V				0.2 DVDD3	
High-level input voltage	PF0,PF1	V_{IH1}	1.8 V ≤ DVDD3B ≤ 3.6 V		0.8 AVDD3	−	AVDD3 + 0.3	V
	PA0 to 7, PB0 to 7, PC0 to 7, PD0 to 7, PE0~ to 7, PF0 to 1, PG0, PH0 to 5, PI0 to 4, PJ0 to 1, PJ3 to 5, PK0 to 1 $\overline{\text{RESET}}$, MODE	V_{IH2}	1.8 V ≤ DVDD3A ≤ 3.6 V		0.8 DVDD3		DVDD3 + 0.3	
	PI5, PI6, PJ2	V_{IH3}			0.9 DVDD3		5.5	
Low-level output voltage		V_{OL}	I_{OL} = 2 mA	DVDD3 ≥ 2.7 V AVDD3 ≥ 2.7 V	−	−	0.4	V
			I_{OL} = 0.5 mA	DVDD3 ≥ 1.8 V AVDD3 ≥ 1.8 V			0.2	
High-level output voltage		V_{OH}	I_{OH} = −2 mA	DVDD3 ≥ 2.7 V	2.4	−	DVDD3	V
				AVDD3 ≥ 2.7 V			AVDD3	
			I_{OH} = −0.5 mA	DVDD3 ≥ 1.8 V	1.5		DVDD3	
				AVDD3 ≥ 1.8 V			AVDD3	
Input leakage current		I_{LI1}	0.0 ≤ V_{IN} ≤ DVDD3 0.0 ≤ V_{IN} ≤ AVDD3		−	0.02	±5	μA
Output leakage current		I_{LO}	0.2 ≤ V_{IN} ≤ DVDD3 − 0.2 0.2 ≤ V_{IN} ≤ AVDD3 − 0.2		−	0.05	±10	
Pull-up resister at Reset		RRST	DVDD3 = 1.8 V to 3.6 V		−	50	150	kΩ
Programmable pull-up/pull-down resistor		PKH	DVDD3 = 1.8 V to 3.6 V AVDD3 = 1.8 V to 3.6 V		−	50	150	kΩ
Pin capacitance (Except power supply pins)		C_{IO}	fc = 1 MHz		−	−	10	pF

Note 1: Ta = 25 °C, DVDD3 = RVDD3 = AVDD3 = DSRVDD = SRVDD = VLC = 3.3 V, unless otherwise noted.
Note 2: The same voltage must be supplied to DVDD3, AVDD3, RVDD3, DSRVDD3, SRVDD and VLC.

22.3 DC Electrical Characteristics (2/3)

DVDD3 = AVDD3 = RVDD3 = DSRVDD3 = SRVDD = VLC = 1.8 V to 3.6 V, Ta = −40 to 85 °C

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Low-level output current	I_{OL}	Per pin	−	−	2	mA
	ΣI_{OL}	Total, all Port	−	−	35	mA
High-level output current	I_{OH}	Per pin	−	−	-2	mA
	ΣI_{OH}	Total, all Port	−	−	-35	mA

Note: The same voltage must be supplied to DVDD3, AVDD3, RVDD3, DSRVDD3, SRVDD and VLC.

22.4 DC Electrical Characteristics (3/3)

DVDD3 = AVDD3 = RVDD3 = DSRVDD3 = SRVDD = VLC = 1.8 V ~ 3.6 V, Ta = -40 ~ 85 °C

Parameter	Symbol	Condition	Min	Typ. (Note1)	Max	Unit
NORMAL (Note2)	I_{DD}	fsys = 16 MHz	−	6.5	9.45	mA
		fsys = 10 MHz	−	4	7.45	
IDLE (Note3)		fsys = 16 MHz	−	2.2	3.4	
		fsys = 10 MHz	−	1.5	2.6	
SLOW		fs = 32.768 kHz	−	300	1000	μA
SLEEP			−	15	350	
STOP		−	−	14	300	

Note 1: Ta = 25 °C, DVDD3 = AVDD3 = RVDD3 = DSRVDD3 = SRVDD = VLC = 3.3 V, unless otherwise noted.

Note 2: I_{DD} NORMAL: Measured with the dhrystone ver. 2.1 operated in FLASH.
All functions operates excluding reference voltage of the AD converter, ΔΣ AD converter, temperature sensor, bleeder resistance of LCD and I/O port.

Note 3: I_{DD} IDLE: Measured with all functions stopped. The currents flow through DVDD3, AVDD3, RVDD3, DSRVDD3, SRVDD and VLC are included.

22.5 10-bit ADC Electrical Characteristics

DVDD3 = AVDD3 = RVDD3 = DSRVDD3 = SRVDD = VLC =1.8 V to 3.6 V
AVSS = DVSS = 0V, Ta = -40 to 85 °C

Parameter		Symbol	Condition	Min	Typ.	Max	Unit
Analog reference voltage(+)		AVREFH	−	1.8	2.7	3.6	V
Analog input voltage		VAIN	−	AVSS	−	AVREFH	V
Power supply current of analog reference voltage	AD conversion	IREF	DVSS = AVSS	−	0.4	0.55	mA
	Non-AD conversion			−	0.05	10	μA
INL error		−	AVDD = 1.8V AVSS = 0.0V VREFH = 1.8V	−	±3	±4	LSB
DNL error				−	±3	±4	
Zero-scale error				−	±3	±4	
Full-scale error				−	±3	±4	
Total error				−	±3	±4	
INL error		−	AVDD = 3.0V AVSS = 0.0V VREFH = 3.0V	−	±2	±3	
DNL error				−	±2	±3	
Zero-scale error				−	±2	±3	
Full-scale error				−	±2	±3	
Total error				−	±2	±3	

Note 1: 1LSB = (AVREFH − AVREFL)/1024 [V]
Note 2: This characteristics is shown in operating only ADC.

22.6 24-bit ΔΣADC Electrical Characteristics

DVDD3 = AVDD3 = RVDD3 = DSRVDD3 = SRVDD = VLC = 2.9 V to 3.6 V
DSRVSS = DVSS = 0V, Ta = -40 to 85 °C

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Analog reference voltage(+)	dsVREFIN	−	−	2.75	−	V
Consumption current of AMP (common to 3units)	$dsAMPI_{CC}$	-	-	0.5	1.0	m A
Consumption current of conversion (per 1unit)	$dsADI_{CC}$	-	-	1.4	2.5	mA
SNDR	dsSNDR	GAIN = ×1 Conversion time ≥ 330 μs	-	84	-	dB
Input range	AINP AINN		−0.375	−	0.375	V
Input impedance	dsinp		−	49.5 (note 2)	−	kΩ
Conversion time	Tconvds	−	165	−	−	μs

Note 1: This characteristics is shown in operating only ADC.
Note 2: A value when AINN=0V.

22.7 Temperature Sensor Characteristics

DVDD3 = AVDD3 = RVDD3 = DSRVDD3 = SRVDD = VLC = 1.8 V ~ 3.6 V, Ta = -40 ~ 85 °C

Parameter	Symbol	Condition	Min	Typ.	Max	íPàþ
Consumption current of reference voltage circuit (Note 1)	$BGRI_{CC}$	−	−	0.2	0.5	mA
Consumption current of tepera-ture sensor	$TEMPI_{CC}$	−	−	0.7	1.0	mA
Relative error (Note 2)	−	Ta = -20 ~ 85 °C	−	−	±3	°C
		Ta = -40 ~ 85 °C	−	−	±5	

Note 1: The reference voltage circuit is shared with a ΔΣ analog/digital converter.

Note 2: These are design assurance values of single temperature sensor, which were obtained from a straight-line approxima-tion based on the measured values at 30 °C and 60 °C.

22.8 LCD Characteristics

DVDD3 = AVDD3 = RVDD3 = DSRVDD3 = SRVDD = VLC = 1.8 V to 3.6 V, Ta = −40 to 85 °C

Parameter		Symbol	Condition	Min	Typ.	Max
Power supply for LCD driver	VLC	LCD driver is enable	2.2	−	3.6	V
		LCD driver is not enable	1.8	−	3.6	
Internal bleeder resistance	RH1	−	−	500	−	kΩ
	RH2		−	200	−	
	RL		−	20	−	

22.9 AC Electrical Characteristics

22.9.1 AC measurement condition

The AC characteristics data of this chapter is measured under the following conditions unless otherwise noted

- Output levels: High = 0.8 × DVDD3A, 0.8 × DVDD3, Low = 0.2 × DVDD3A, 0.2 × DVDD3B
- Input levels: Refer to low-level input voltage and high-level input voltage in "DC Electrical Characteristics".
- Load capacity: CL = 30pF

22.9.2 Serial Channel (SIO/UART)

22.9.2.1 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

[Input]

Parameter	Symbol	Equation		16 MHz		Unit
		Min	Max	Min	Max	
SCLK Clock High width (input)	t_{SCH}	4x	−	250	−	ns
SCLK Clock Low width (input)	t_{SCL}	4x	−	250	−	
SCLK cycle	t_{SCY}	$t_{SCH} + t_{SCL}$	−	500	−	
Valid Data input ← SCLK rise or fall (Note)	t_{SRD}	30	−	30	−	
SCLK rise → Input Data hold or fall (Note)	t_{HSR}	x + 30	−	92.5	−	

[Output]

Parameter	Symbol	Equation		16 MHz		Unit
		Min	Max	Min	Max	
SCLK Clock High width (input)	t_{SCH}	4x	−	250	−	ns
SCLK Clock Low width (input)	t_{SCL}	4x	−	250	−	
SCLK cycle	t_{SCY}	$t_{SCH} + t_{SCL}$	−	500	−	
Output Data ← SCLK rise or fall (Note)	t_{OSS}	$t_{SCY}/2 - 3x - 45$	−	17.5	−	
SCLK rise → Output Data hold or fall (Note)	t_{OHS}	$t_{SCY}/2$	−	250	−	

Note: SCLK rise or fall ; Measured relative to the programmed active edge of SCLK.

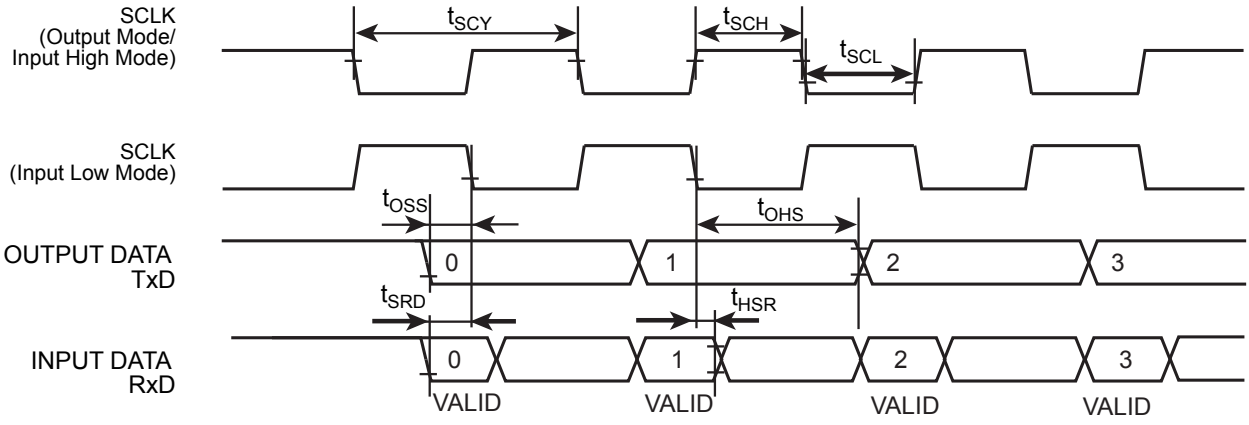
(2) SCLK output mode

DVDD3 = AVDD3 = RVDD3 = DSRVDD3 = SRVDD = VLC = 2.7 to 3.6V

Parameter	Symbol	Equation		16 MHz		Unit
		Min	Max	Min	Max	
SCLK cycle (programmable)	t_{SCY}	4x	−	250	−	ns
Output Data ← SCLK rise	t_{OSS}	$t_{SCY}/2 - 30$	−	95	−	
SCLK rise → Output Data hold	t_{OHS}	$t_{SCY}/2 - 30$	−	95	−	
Valid Data Input ← SCLK rise	t_{SRD}	45	−	45	−	
SCLK rise → Input Data hold	t_{HSR}	0	−	0	−	

DVDD3 = AVDD3 = RVDD3 = DSRVDD3 = SRVDD = VLC = 1.8 V

Parameter	Symbol	Equation		16 MHz		Unit
		Min	Max	Min	Max	
SCLK cycle (programmable)	t_{SCY}	4x	−	250	−	ns
Output Data ← SCLK rise	t_{OSS}	$t_{SCY}/2 - 30$	−	95	−	
SCLK rise → Output Data hold	t_{OHS}	$t_{SCY}/2 - 70$	−	55	−	
Valid Data Input ← SCLK rise	t_{SRD}	90	−	90	−	
SCLK rise → Input Data hold	t_{HSR}	0	−	0	−	



22.9.3 Serial Bus Interface (I2C/SIO)

22.9.3.1 I2C Mode

In the table below, the letter x represents the I2C operation clock cycle time which is identical to the fsys cycle time. It varies depending on the programming of the clock gear function.

n denotes the value of n programmed into the SCK (SCL output frequency select) field in the SBIxCR.

Parameter	Symbol	Equation		Standard Mode		Fast Mode		Unit
		Min	Max	Min	Max	Min	Max	
SCL clock frequency	t_{SCL}	0	−	0	100	0	400	kHz
Hold time for START condition	$t_{HD;\ STA}$	−	−	4.0	−	0.6	−	μs
SCL Low width (Input) (Note 1)	t_{LOW}	−	−	4.7	−	1.3	−	μs
SCL High width (Input) (Note 2)	t_{HIGH}	−	−	4.0	−	0.6	−	μs
Setup time for a repeated START condition	$t_{SU;\ STA}$	(Note5)	−	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}	(Note5)	−	4.7	−	1.3	−	μs

Note 1: SCL clock Low width (output) = $(2^{n-1} + 58)/x$

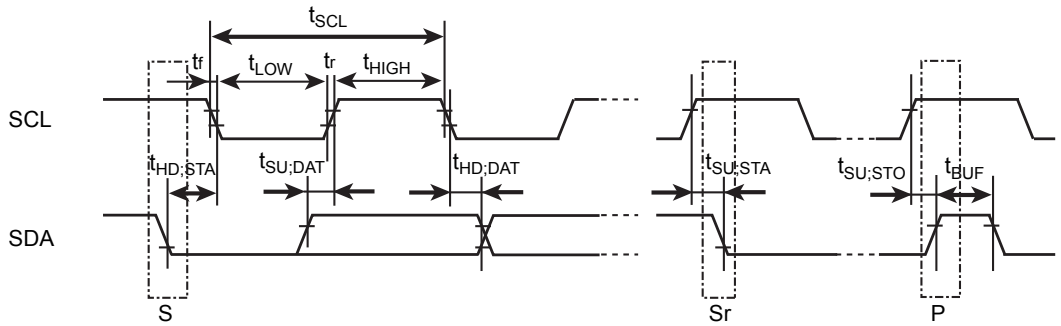
Note 2: SCL clock High width (output) = $(2^{n-1} + 14)/x$ On I2C-bus specification, Maximum Speed of Standard Mode is 100kHz, Fast mode is 400khz. Internal SCL Frequency setting should comply with Note1 & Note2 shown above.

Note 3: The output data hold time is equal to 4x of internal SCL.

Note 4: The Philips I2C-bus specification states that a device must internally provide a hold time of at least 300 ns for the SDA signal to bridge the undefined region of the falling edge of SCL. However, this SBI does not satisfy this requirement. Also, the output buffer for SCL does not incorporate slope control of the falling edges; therefore, the equipment manufacturer should design so that the input data hold time shown in the table is satisfied, including tr/tf of the SCL and SDA lines.

Note 5: Software -dependent

Note 6: The Philips I2C-bus specification instructs that if the power supply to a Fast-mode device is switched off, the SDA and SCL I/O pins must be floating so that they don't obstruct the bus lines. However, this SBI does not satisfy this requirement.



S: Start condition
Sr: Repeated start condition
P: Stop condition

22.9.3.2 Clock-Synchronous 8-Bit SIO mode

In the table below, the letter x represents the I2C operation clock cycle time which is identical to the fsys cycle time. It varies depending on the programming of the clock gear function.

(1) SCK Input Mode (The electrical specifications below are for an SCK signal with a 50% duty cycle.)

[Input]

Parameter	Symbol	Equation		16 MHz		Unit
		Min	Max	Min	Max	
SCK Clock High width (input)	t_{SCH}	4x	−	250	−	ns
SCK Clock Low width (input)	t_{SCL}	4x	−	250	−	
SCK cycle	t_{SCY}	$t_{SCH} + t_{SCL}$	−	500	−	
Valid Data input ← SCK rise	t_{SRD}	30 − x	−	-32.5	−	
SCK rise → Input Data hold	t_{HSR}	2x + 30	−	155	−	

[Output]

Parameter	Symbol	Equation		16 MHz		Unit
		Min	Max	Min	Max	
SCK Clock High width (input)	t_{SCH}	4x	−	250	−	ns
SCK Clock Low width (input)	t_{SCL}	4x	−	250	−	
SCK cycle	t_{SCY}	$t_{SCH} + t_{SCL}$	−	500	−	
Output Data ← SCK rise	t_{OSS}	$t_{SCY}/2 - 3x - 45$	−	17.5	−	
SCK rise → Output Data hold	t_{OHS}	$t_{SCY}/2 + x$	−	312.5	−	

(2) SCK Output Mode (The electrical specifications below are for an SCK signal with a 50% duty cycle.)

DVDD3 = AVDD3 = RVDD3 = DSRVDD3 = SRVDD = VLC = 2.7 to 3.6V

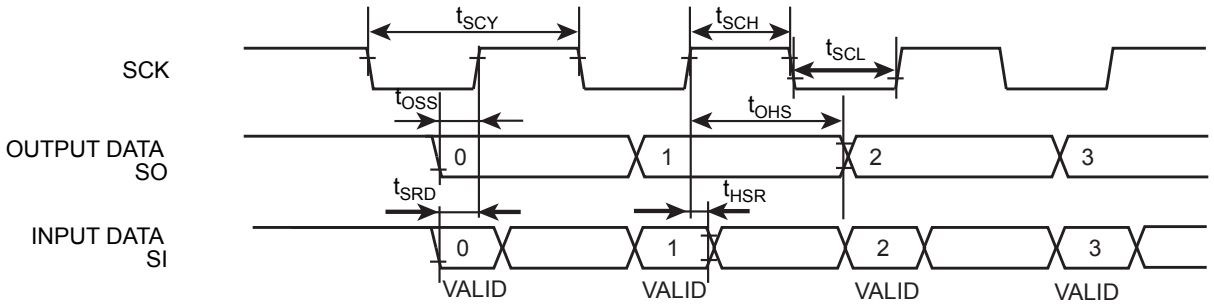
Parameter	Symbol	Equation		16 MHz		Unit
		Min	Max	Min	Max	
SCK cycle (programmable)	t_{SCY}	16x (Note 1)	−	1000	−	ns
Output Data ← SCK rise	t_{OSS}	$t_{SCY}/2 - 30$ (Note2)	−	470	−	
SCK rise → Output Data hold	t_{OHS}	$t_{SCY}/2 - 30$	−	470	−	
Valid Data input ← SCK rise	t_{SRD}	x + 45	−	107.5	−	
SCK rise → Input Data hold	t_{HSR}	0	−	0	−	

DVDD3 = AVDD3 = RVDD3 = DSRVDD3 = = SRVDD = VLC = 1.8 V

Parameter	Symbol	Equation		40 MHz		Unit
		Min	Max	Min	Max	
SCK cycle (programmable)	t_{SCY}	16x (Note 1)	−	1000	−	ns
Output Data ← SCK rise	t_{OSS}	$t_{SCY}/2 - 30$ (Note2)	−	470	−	
SCK rise → Output Data hold	t_{OHS}	$t_{SCY}/2 - 70$	−	430	−	
Valid Data input ← SCK rise	t_{SRD}	x + 90	−	152.5	−	
SCK rise → Input Data hold	t_{HSR}	0	−	0	−	

Note 1: SCK cycle after automatic wait becomes 14x.

Note 2: SO data output after automatic wait may be $t_{SCY}/2$ - x - 20.



22.9.4 Event Counter

In the table below, the letter x represents the TMRB operation clock cycle time which is identical to the fsys cycle time. It varies depending on the programming of the clock gear function.

Parameter	Symbol	Equation		16 MHz		Unit
		Min	Max	Min	Max	
Clock Low pulse width	t_{VCKL}	2x + 100	−	225	−	ns
Clock High pulse width	t_{VCKH}	2x + 100	−	225	−	ns

22.9.5 Capture

In the table below, the letter x represents the TMRB operation clock cycle time which is identical to the fsys cycle time. It varies depending on the programming of the clock gear function.

Parameter	Symbol	Equation		16 MHz		Unit
		Min	Max	Min	Max	
Low pulse width	t_{CPL}	2x + 100	−	225	−	ns
High pulse width	t_{CPH}	2x + 100	−	225	−	ns

22.9.6 External Interrupt

In the table below, the letter x represents the fsys cycle time.

1. Except STOP release interrupts

Parameter	Symbol	Equation		16 MHz		Unit
		Min	Max	Min	Max	
INT0 to 3 low level pulse width	t_{INTAL}	x + 100	−	162.5	−	ns
INT0 to 3 high level pulse width	t_{INTAH}	x + 100	−	162.5	−	ns

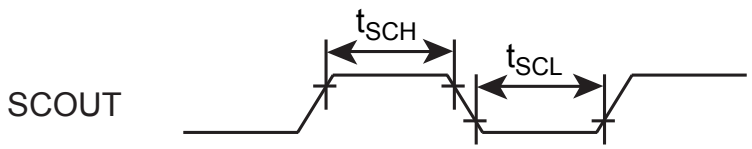
2. STOP release interrupts

Parameter	Symbol	Min	Max	Unit
INT0 to 3 low level pulse width	t_{INTBL}	100	–	ns
INT0 to 3 high level pulse width	t_{INTBH}	100	–	ns

22.9.7 SCOUT Pin AC Characteristic

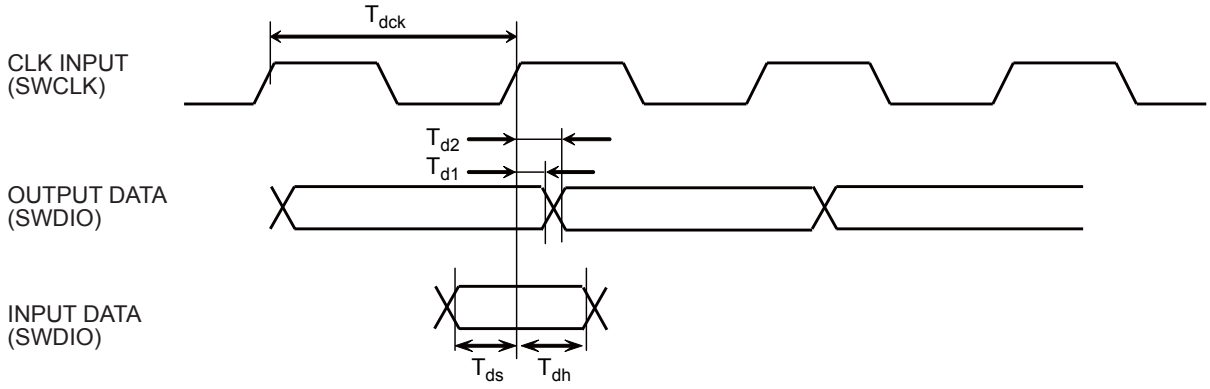
Parameter	Symbol	Equation		16 MHz		Unit
		Min	Max	Min	Max	
High level pulse width	t_{SCH}	0.5T − 5	–	26.25	–	ns
Low level pulse width	t_{SCL}	0.5T − 5	–	26.25	–	ns

Note: In the above table, the letter T represents the cycle time of the SCOUT output clock.



22.9.8 Debug Communication

Parameter	Symbol	Min	Max	Unit
CLK cycle	T_{dck}	100	–	ns
CLK rise → Output data hold	T_{d1}	4	–	
CLK fall → Output data valid	T_{d2}	–	30	
Input data valid → CLK rise	T_{ds}	20	–	
CLK rise → Input data hold	T_{dh}	15	–	



22.9.9 Flash Characteristics

Parameter	Condition	Min	Typ.	Max	Unit
Guarantee on Flash-memory rewriting	DVDD3 = RVDD3 = 2.7 V ~ 3.6 V, Ta = 0 to 70°C	-	-	100	Times

22.9.10 On chip oscillator

Parameter	Symbol	Min	Typ.	Max	Unit
Oscillating frequency	IHOSC	9	10	11	MHz

22.10 Recommended Oscillation Circuit

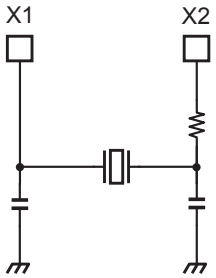


Figure 22-1 High-frequency oscillation connection

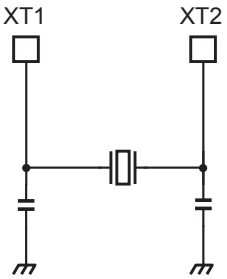


Figure 22-2 Low-frequency oscillation connection

Note: To obtain a stable oscillation, load capacity and the position of the oscillator must be configured properly. Since these factors are strongly affected by substratepatterns, please evaluate oscillation stability using the substrate you use.

22.10.1 Ceramic oscillator

This product has been evaluated by the ceramic oscillator by Murata Manufacturing Co., Ltd.

Please refer to the company's website for details.

22.10.2 Crystal oscillator

This product has been evaluated by the crystal oscillator by Seiko Instruments Inc..

Please refer to the company's website for details.

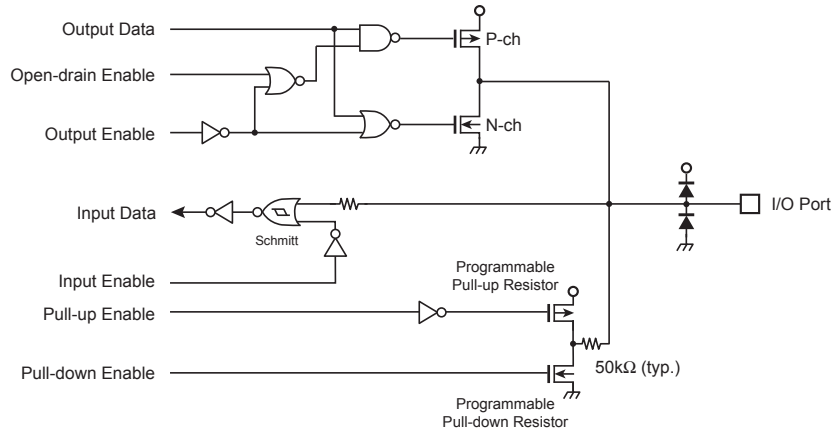
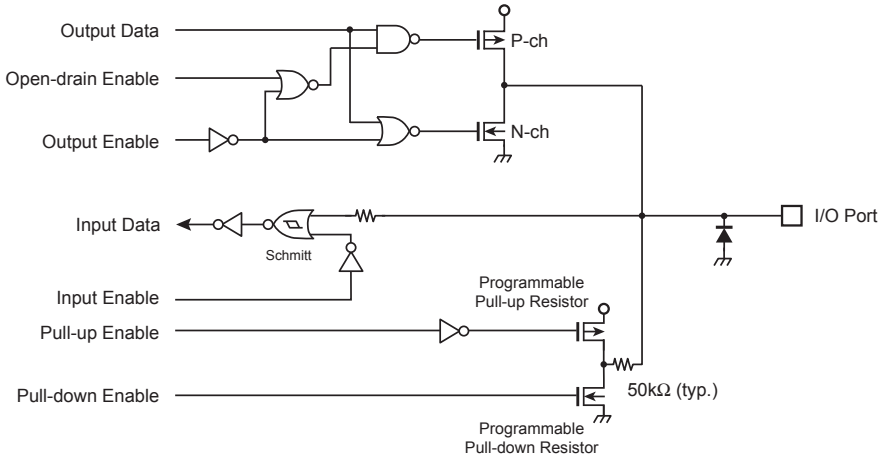
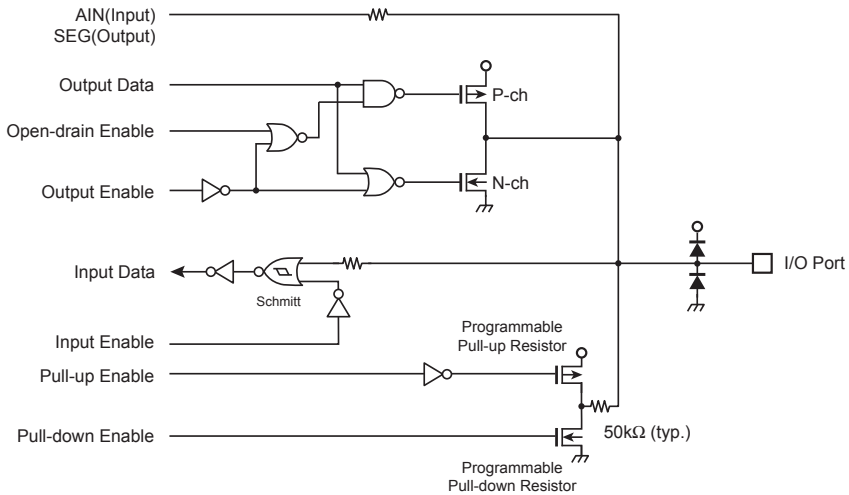
23. Port Section Equivalent Circuit Schematic

Basically, the gate symbols written are the same as those used for the standard CMOS logic IC [74HCXX] series.

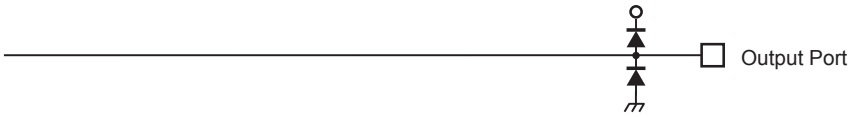
The input protection resistance ranges from several tens of Ω to several hundreds of Ω. Damping resistors X2 and XT2 are shown with a typical value.

Note: Resistors without values in the figure show input protection resistors.

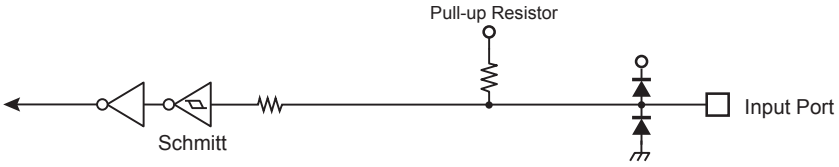
23.1 PORT pin

PE0 to 5 (Port type FT1) PE6, 7 (Port type FT2) PF1 (Port type FT4) PG0 PH0 to 5 PI0~4 PJ0~1, 3~4 PK0 (Port type FT4) PK1 (Port type FT1)	
PI5, PI6, PJ2	 Note : Only when input is enabled, these pins tolerate 5V inputs.
PA0 to 7 PB0 to 7 PC0 to 7 PD0 to 7 PE0 to 7 (Port type FT10) PF0 PF1 (Port type FT5) PK0, 1 (Port type FT5)	
PJ5	

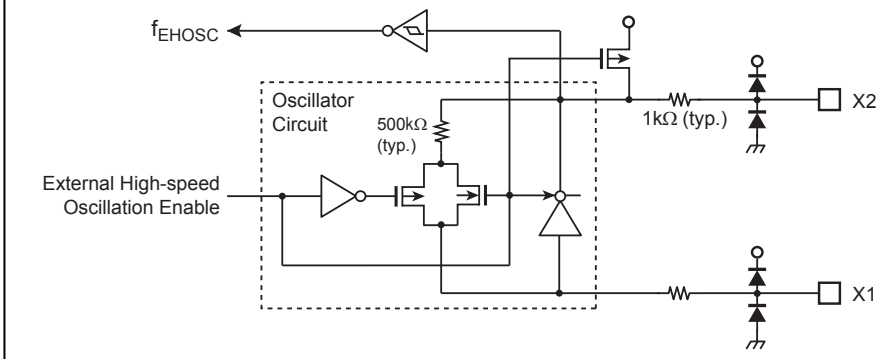
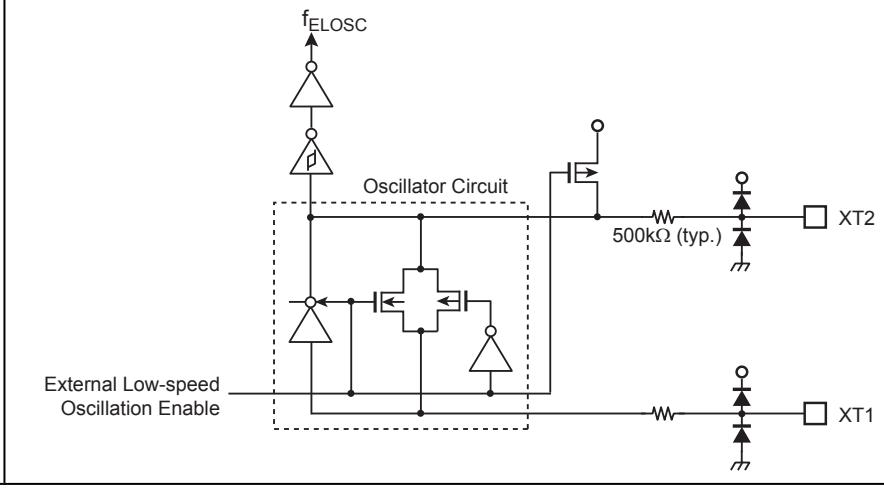
23.2 Analog pin

COM0, COM1, COM2, COM3	
DAIN0+, DAIN0-, DAIN1+, DAIN1-, DAIN2+, DAIN2-,	

23.3 Control pin

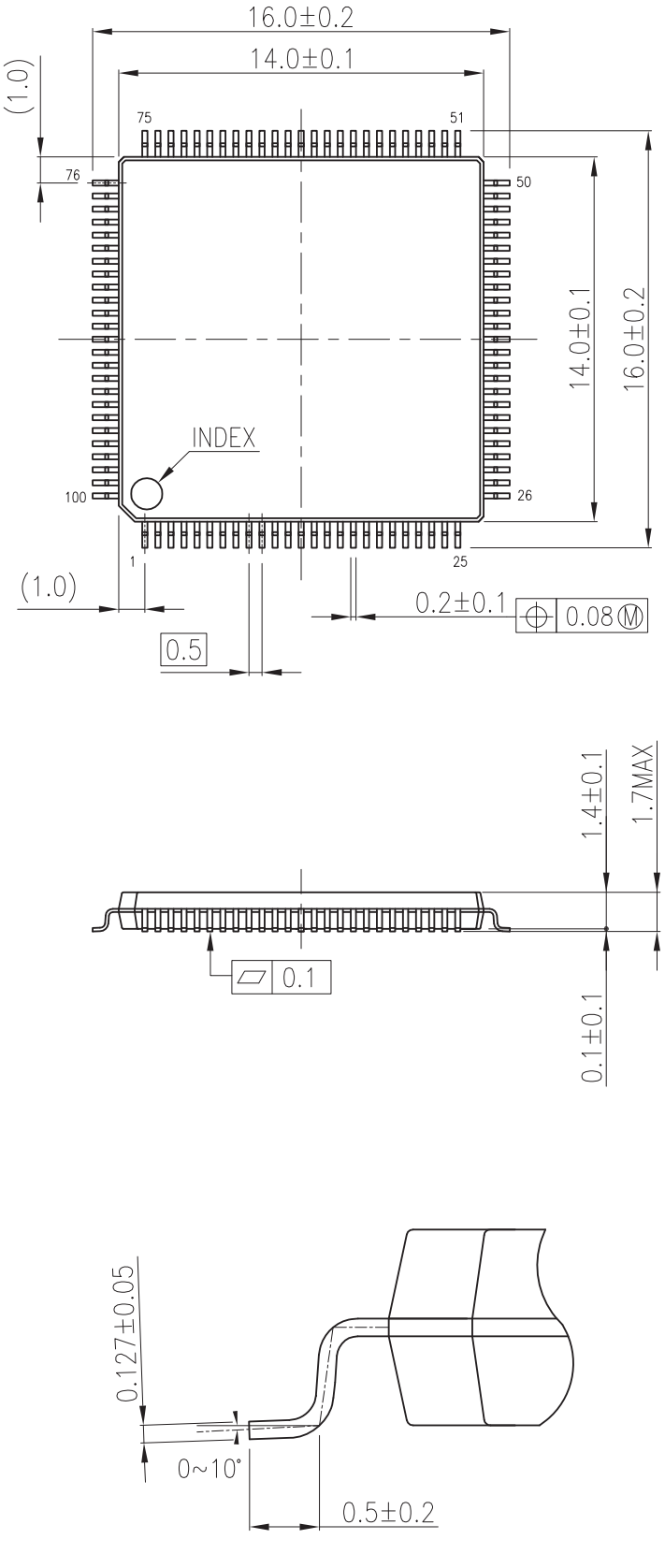
MODE	 Mode pin must be connected to GND.
$\overline{\text{RESET}}$	

23.4 Clock pin

X1、X2	
XT1, XT2	

24. Package Dimensions

Type: LQFP100-P-1414-0.50G



RESTRICTIONS ON PRODUCT USE

Toshiba Corporation and its subsidiaries and affiliates are collectively referred to as “TOSHIBA”.
Hardware, software and systems described in this document are collectively referred to as “Product”.

- TOSHIBA reserves the right to make changes to the information in this document and related Product without notice.
- This document and any information herein may not be reproduced without prior written permission from TOSHIBA. Even with TOSHIBA's written permission, reproduction is permissible only if reproduction is without alteration/omission.
- Though TOSHIBA works continually to improve Product's quality and reliability, Product can malfunction or fail. Customers are responsible for complying with safety standards and for providing adequate designs and safeguards for their hardware, software and systems which minimize risk and avoid situations in which a malfunction or failure of Product could cause loss of human life, bodily injury or damage to property, including data loss or corruption. Before customers use the Product, create designs including the Product, or incorporate the Product into their own applications, customers must also refer to and comply with (a) the latest versions of all relevant TOSHIBA information, including without limitation, this document, the specifications, the data sheets and application notes for Product and the precautions and conditions set forth in the "TOSHIBA Semiconductor Reliability Handbook" and (b) the instructions for the application with which the Product will be used with or for. Customers are solely responsible for all aspects of their own product design or applications, including but not limited to (a) determining the appropriateness of the use of this Product in such design or applications; (b) evaluating and determining the applicability of any information contained in this document, or in charts, diagrams, programs, algorithms, sample application circuits, or any other referenced documents; and (c) validating all operating parameters for such designs and applications. **TOSHIBA ASSUMES NO LIABILITY FOR CUSTOMERS' PRODUCT DESIGN OR APPLICATIONS.**
- **PRODUCT IS NEITHER INTENDED NOR WARRANTED FOR USE IN EQUIPMENTS OR SYSTEMS THAT REQUIRE EXTRAORDINARILY HIGH LEVELS OF QUALITY AND/OR RELIABILITY, AND/OR A MALFUNCTION OR FAILURE OF WHICH MAY CAUSE LOSS OF HUMAN LIFE, BODILY INJURY, SERIOUS PROPERTY DAMAGE AND/OR SERIOUS PUBLIC IMPACT ("UNINTENDED USE").** Except for specific applications as expressly stated in this document, Unintended Use includes, without limitation, equipment used in nuclear facilities, equipment used in the aerospace industry, lifesaving and/or life supporting medical equipment, equipment used for automobiles, trains, ships and other transportation, traffic signaling equipment, equipment used to control combustions or explosions, safety devices, elevators and escalators, and devices related to power plant. IF YOU USE PRODUCT FOR UNINTENDED USE, TOSHIBA ASSUMES NO LIABILITY FOR PRODUCT. For details, please contact your TOSHIBA sales representative or contact us via our website.
- Do not disassemble, analyze, reverse-engineer, alter, modify, translate or copy Product, whether in whole or in part.
- Product shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable laws or regulations.
- The information contained herein is presented only as guidance for Product use. No responsibility is assumed by TOSHIBA for any infringement of patents or any other intellectual property rights of third parties that may result from the use of Product. No license to any intellectual property right is granted by this document, whether express or implied, by estoppel or otherwise.
- **ABSENT A WRITTEN SIGNED AGREEMENT, EXCEPT AS PROVIDED IN THE RELEVANT TERMS AND CONDITIONS OF SALE FOR PRODUCT, AND TO THE MAXIMUM EXTENT ALLOWABLE BY LAW, TOSHIBA (1) ASSUMES NO LIABILITY WHATSOEVER, INCLUDING WITHOUT LIMITATION, INDIRECT, CONSEQUENTIAL, SPECIAL, OR INCIDENTAL DAMAGES OR LOSS, INCLUDING WITHOUT LIMITATION, LOSS OF PROFITS, LOSS OF OPPORTUNITIES, BUSINESS INTERRUPTION AND LOSS OF DATA, AND (2) DISCLAIMS ANY AND ALL EXPRESS OR IMPLIED WARRANTIES AND CONDITIONS RELATED TO SALE, USE OF PRODUCT, OR INFORMATION, INCLUDING WARRANTIES OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, ACCURACY OF INFORMATION, OR NONINFRINGEMENT.**
- Do not use or otherwise make available Product or related software or technology for any military purposes, including without limitation, for the design, development, use, stockpiling or manufacturing of nuclear, chemical, or biological weapons or missile technology products (mass destruction weapons). Product and related software and technology may be controlled under the applicable export laws and regulations including, without limitation, the Japanese Foreign Exchange and Foreign Trade Law and the U.S. Export Administration Regulations. Export and re-export of Product or related software or technology are strictly prohibited except in compliance with all applicable export laws and regulations.
- Please contact your TOSHIBA sales representative for details as to environmental matters such as the RoHS compatibility of Product. Please use Product in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. **TOSHIBA ASSUMES NO LIABILITY FOR DAMAGES OR LOSSES OCCURRING AS A RESULT OF NONCOMPLIANCE WITH APPLICABLE LAWS AND REGULATIONS.**

单击下面可查看定价，库存，交付和生命周期等信息

>>Toshiba(东芝)