8-bit Microcontroller

TLCS-870/C1 Series

TMP89FS60BFG
TMP89FS60BUG
TMP89FS62BUG
TMP89FS63BUG

TOSHIBA ELECTRONIC DEVICES & STORAGE CORPORATION

Start of commercial production
2020-08

Revision History

Date	Revision	Description
2020-08-12	1.0	First Release

Table of Contents

List of Tables

Table of Figures

1. Description

The TMP89FS60BFG, the TMP89FS60BUG, the TMP89FS62BUG and the TMP89FS63BUG are the single-chip 8-bit high-speed and high-functionality microcontrollers incorporating 61440 bytes of Flash Memory

Product No	Flash Memory	RAM	Package
TMP89FS60BFG	61440 bytes	3072 bytes	LQFP64 (14 mm ×14 mm, 0.8 mm pitch)
TMP89FS60BUG			LQFP64 (10 mm ×10 mm, 0.5 mm pitch)
TMP89FS62BUG			LQFP44 (10 mm ×10 mm, 0.8 mm pitch)
TMP89FS63BUG			LQFP52 (10 mm ×10 mm, 0.65 mm pitch)

Hereafter, the TMP89FS60BFG and the TMP89FS60BUG are named as the TMP89FS60B, the TMP89FS62BUG as the TMP89FS62B and the TMP89FS63BUG as the TMP89FS63B. And all of these products are named as the TMP89FS60B/62B/63B.

1.1. Features

- 8-bit single chip microcontroller: TLCS-870/C1 series
  - Minimum instruction execution time:
    - 100 [ns] (when operating at 10 [MHz])
    - 122 [μs] (when operating at 32.768 [kHz])
  - Basic instructions: 133 types and 732 instructions

- Interrupt sources:
  - TMP89FS60B: 27 interrupt sources (6 external interrupt sources and 21 internal interrupt sources except the reset)
  - TMP89FS62B: 24 interrupt sources (4 external interrupt sources and 20 internal interrupt sources except the reset)
  - TMP89FS63B: 25 interrupt sources (4 external interrupt sources and 21 internal interrupt sources except the reset)

- The number of input/output ports
  - TMP89FS60B: 56 ports including 8 large current (20 [mA] (typ.)) ports
  - TMP89FS62B: 37 ports including 6 large current (20 [mA] (typ.)) ports
  - TMP89FS63B: 44 ports including 6 large current (20 [mA] (typ.)) ports

- Watchdog timer (WDT)
  - The operation when a watchdog counter is overflowed: Interrupt or reset can be selected

- Power-on reset circuit (POR)

- Voltage detection circuit (VLTD)
  - Detecting voltage can be set.

- Divider output (DVO)

- Time base timer (TBT)

- 16-bit timer counter (TCA): 2 channels
  - Timer, external trigger, event counter, window, pulse width measurement, PPG output functions

- 8-bit timer counter (TC0): 4 channels
  - Timer, event counter, PWM output, PPG output functions
  - Usable as a 16-bit timer, 12-bit PWM output and 16-bit PPG output by the cascade connection of two channels.

- Real time clock (RTC)

- Serial interface
  - TMP89FS60B
    - UART: 1 channel
    - UART/SIO: 3 channels
    - I^{2}C/SIO: 1 channel

Note: The maximum channel number of UART simultaneously is 3 channels. And One of SIO is 2 channels.

  - TMP89FS62B
    - UART: 2 channels
    - UART/SIO: 2 channels
    - I^{2}C/SIO: Not available

Note: The maximum channel number of UART simultaneously is 3 channels.

  - TMP89FS63B
    - UART: 2 channels
    - UART/SIO: 2 channels
    - I^{2}C/SIO: 1channel

Note: The maximum channel number of UART simultaneously is 3 channels. And One of SIO is 2 channels.

- Key-on wake-up (KWU): 8 channels

- 10-bit AD converter (ADC)
  - successive approximation type
  - Analog input
    - TMP89FS60B: 16 channels
    - TMP89FS62B: 8 channels
    - TMP89FS63B: 13 channels

- On-chip debug function (OCD)
  - Break/Event
  - Trace
  - RAM monitor
  - Flash memory writing

- Clock oscillation circuits: 2 circuits (High-speed and low-speed oscillation circuits)
  - Single clock or dual clock mode can be selected.

- Low power consumption operation (8 modes)
  - STOP mode: The high-frequency and the low-frequency oscillation circuits stop.
    - The CPU stops.
    - The peripheral circuits stop.

  - SLOW1 mode: The high-frequency oscillation circuit stops. The low-frequency oscillation circuit operates.
    - The CPU operates by the low-frequency clock.
    - The peripheral circuits operate by the low-frequency clock.

  - SLOW2 mode: The high-frequency and the low-frequency oscillation circuits operate.
    - The CPU operates by the low-frequency clock.
    - The peripheral circuits operate by the low-frequency clock.

  - IDLE0 mode: The high-frequency oscillation circuit operates. The low-frequency oscillation circuit stops.
    - The CPU stops
    - Only Time base timer of the peripheral circuits operates by the high-frequency clock. The IDLE0 mode is released by an interrupt request of the time base timer and the CPU restarts.

  - IDLE1 mode: The high-frequency oscillation circuit operates. The low-frequency oscillation circuit stops.
    - The CPU stops
    - All peripheral circuits operate by the high-frequency clock. The IDLE1 mode is released by an interrupt request of the peripheral circuits and the CPU restarts.

  - IDLE2 mode: The high-frequency and the low-frequency oscillation circuits operate.
    - The CPU stops
    - All peripheral circuits operate by the high-frequency or low-frequency clock. The IDLE2 mode is released by an interrupt request of the peripheral circuits and the CPU restarts.

  - SLEEP0 mode: The high-frequency oscillation circuit stops. The low-frequency oscillation circuit operates.
    - The CPU stops
    - Only time base timer of the peripheral circuits operates by the low-frequency clock. The SLEEP0 mode is released by an interrupt request of the time base timer and the CPU restarts.

  - SLEEP1 mode: The high-frequency oscillation circuit stops. The low-frequency oscillation circuit operates.
    - The CPU stops
    - All peripheral circuits operate by the low-frequency clock. The SLEEP1 mode is released by an interrupt request of the peripheral circuits and the CPU restarts.

- Operation voltage and frequency:
  - $4.5 \leq V_{DD} \leq 5.5$ [V], 10 [MHz]/32.768 [kHz]
    - The TMP89FS60B/62B/63B satisfy all functions and all electrical characteristics.

  - $4.2 \leq V_{DD} < 4.5$ [V], 10 [MHz]/32.768 [kHz]
    - The TMP89FS60B/62B/63B satisfy all functions except 10-bit AD converter’s AD conversion characteristics, DC/AC characteristics and write characteristics of Flash memory.

1.2. List of difference between the TMP89FS60B, theTMP89FS62B and the TMP89FS63B

The list of the difference between the TMP89FS60B/62B/63B is shown below.

Table 1-1 List of Difference between the TMP89FS60B, the TMP89FS62B and theTMP89FS63B

Products Name		TMP89FS60BFG	TMP89FS60BUG	TMP89FS62BUG	TMP89FS63BUG
Flash memory size		61440 bytes			
RAM size		3072 bytes			
Number of input/output port	Total number of ports	56		37	44
	Number of large current ports	8		6	6
Number of interrupt sources except reset	Total number of interrupt sources	27		24	25
	Number of external interrupt source	6		4	4
	Number of internal interrupt sources	21		20	21
Watchdog timer (WDT)		A			
Power-on reset (POR)		A			
Voltage detection (VLTD)		A			
Divider output (DVO)		A			
Time base timer (TBT)		A			
16-bit timer/counter (TCA)		2 channels			
8-bit timer/counter (TC0)		4 channels			
Real time clock (RTC)		A			
UART (Note 2)		1 channel		2 channels	2 channels
UART/SIO (Note 2)(Note 3)		3 channels		2 channels	2 channels
I^2C/SIO (Note 3)		1 channel		NA	1 channel
Key-on wakeup (KWU)		8 channels			
Number of analog input for 10-bit AD converter (ADC)		16 channels		8 channels	13 channels
Low power consumption operation	STOP mode	A			
	SLOW1 mode	A			
	SLOW2 mode	A			
	IDLE0 mode	A			
	IDLE1 mode	A			
	IDLE2 mode	A			
	SLEEP0 mode	A			
	SLEEP1 mode	A			
Operation voltage and frequency	4.5 ≤ V_{DD} ≤ 5.5 [V] 10 [MHz]/32.768 [kHz]	The TMP89FS60B/62B/63B satisfy all functions and all electrical characteristics			
	4.2 ≤ V_{DD} < 4.5 [V] 10 [MHz]/32.768 [kHz]	The TMP89FS60B/62B/63B satisfy all functions except 10-bit successive approximation type AD converter, DC/AC characteristics and write characteristics of Flash memory			
Number of erasing and writing to Flash memory		1000 [times]			
On-chip debug function		A			
Package		LQFP64 (14 [mm] × 14 [mm]、0.8 [mm] pitch)	LQFP64 (10 [mm] × 10 [mm]、0.5 [mm] pitch)	LQFP44 (10 [mm] × 10 [mm]、0.8 [mm] pitch)	LQFP52 (10 [mm] × 10 [mm]、0.65 [mm] pitch)

Note 1: A: Available, NA: Not available

Note 2: The maximum number of UART channels simultaneously is 3.

Note 3: The maximum number of SIO channels simultaneously is 2.

1.3. Pin Assignment

1.3.1. Pin Assignment of the TMP89FS60BFG/TMP89FS60BUG (Top view)

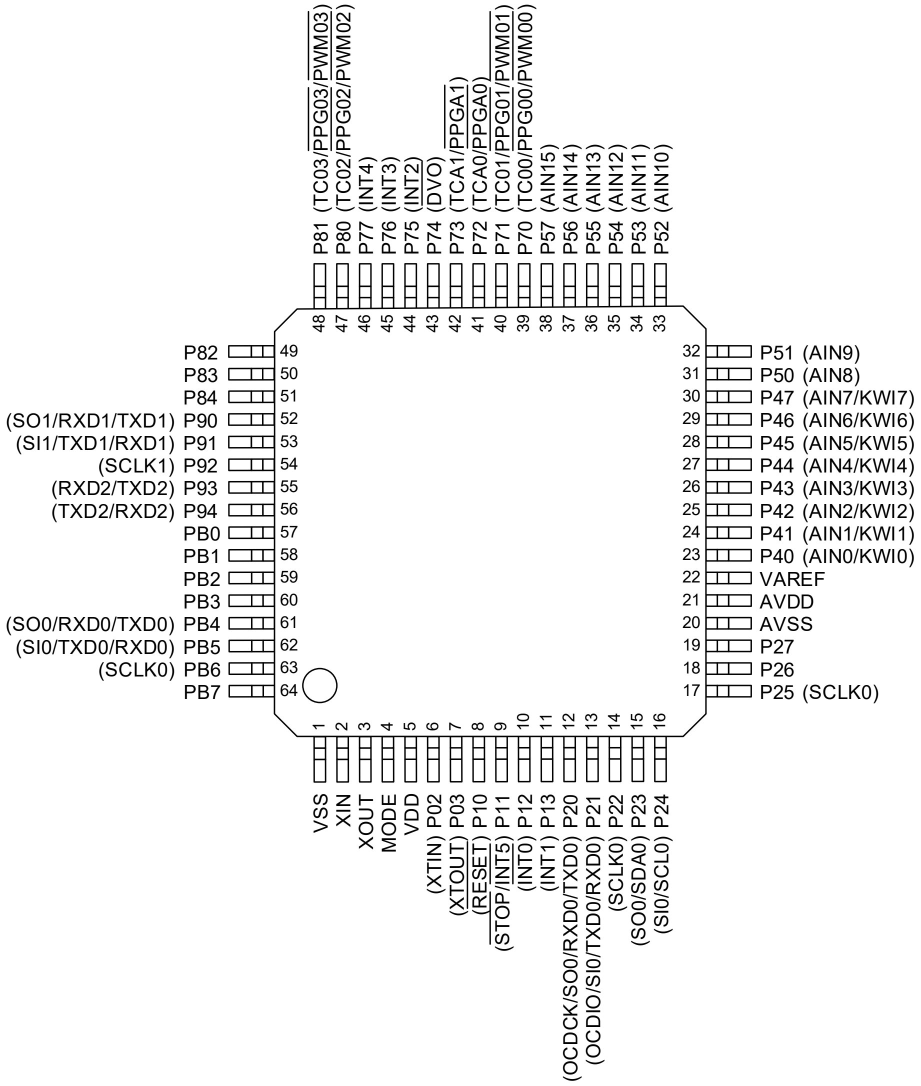


Figure 1-1 Pin Assignment (TMP89FS60BFG/TMP89FS60BUG)

1.3.2. Pin Assignment of the TMP89FS62BUG (Top View)

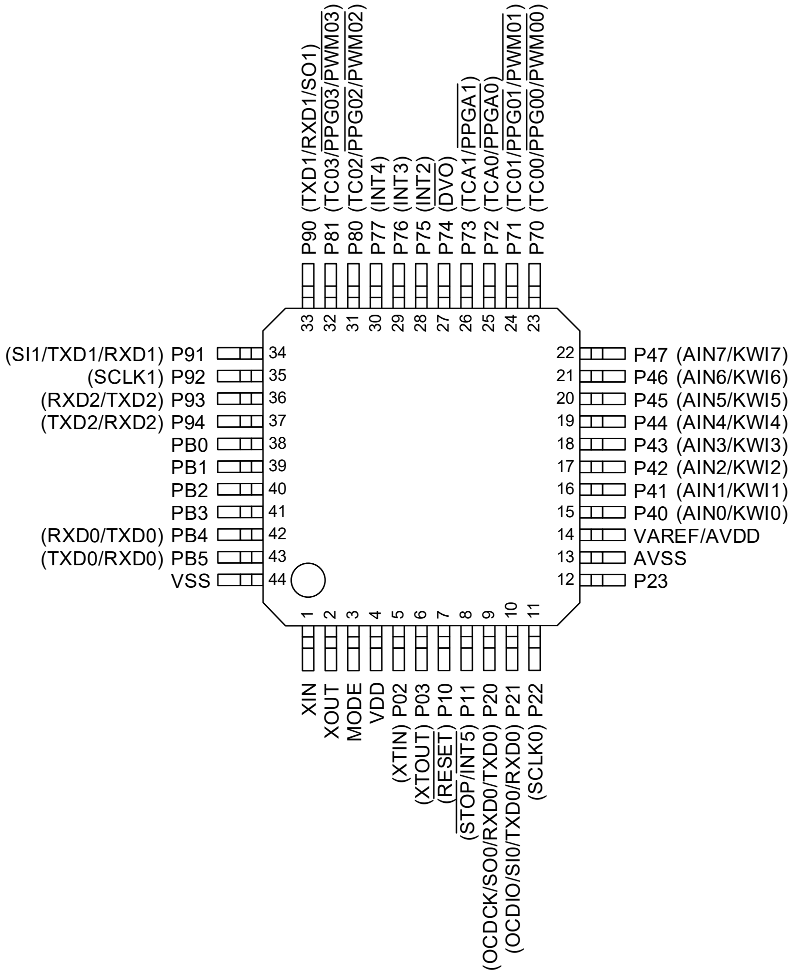


Figure 1-2 Pin Assignment (TMP89FS62BUG)

1.3.3. Pin Assignment of the TMP89FS63BUG (Top view)

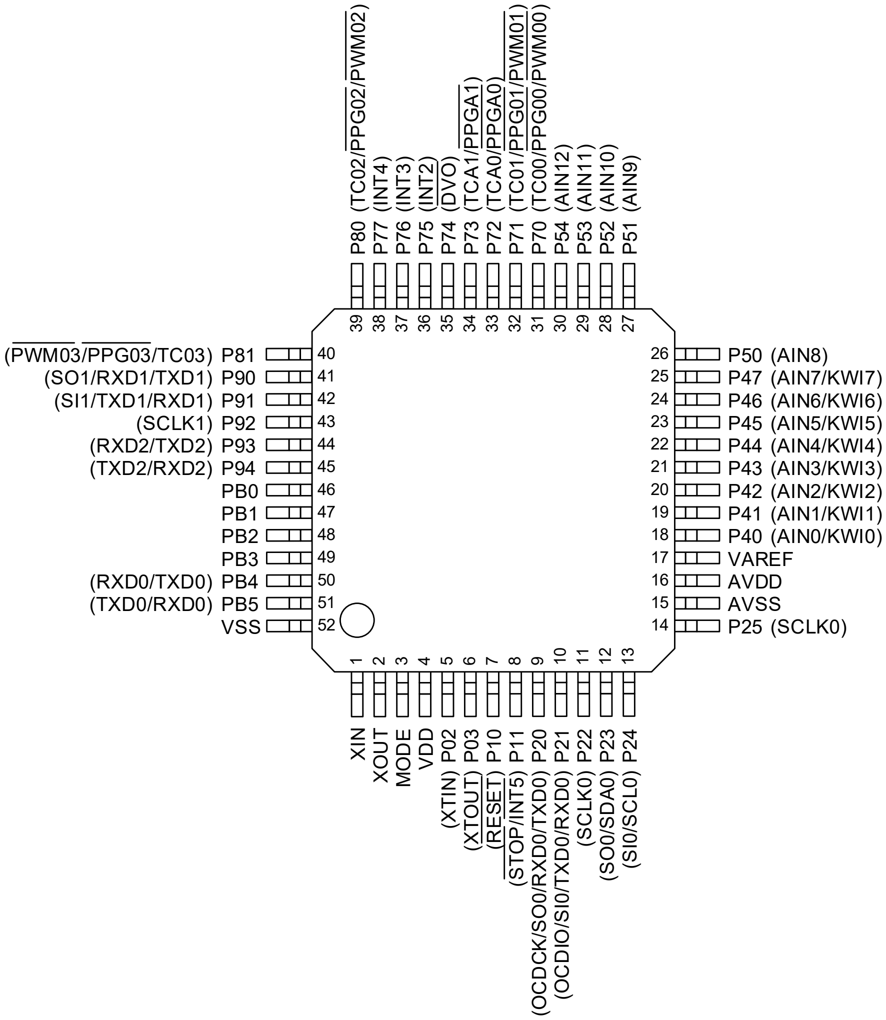


Figure 1-3 Pin Assignment (TMP89FS63BUG)

1.4. Block Diagram

1.4.1. Block Diagram of the TMP89FS60B

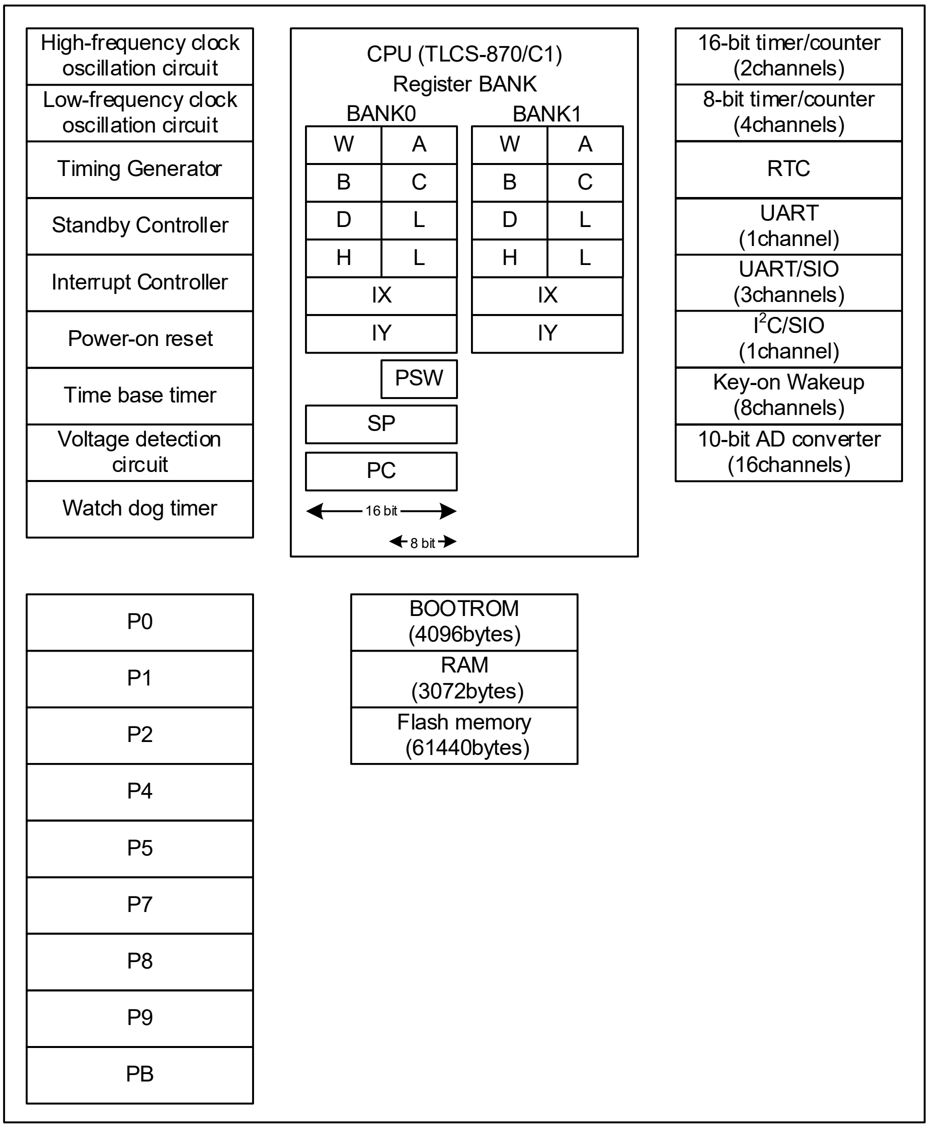


Figure 1-4 Block Diagram

1.4.2. Block Diagram of the TMP89FS62B

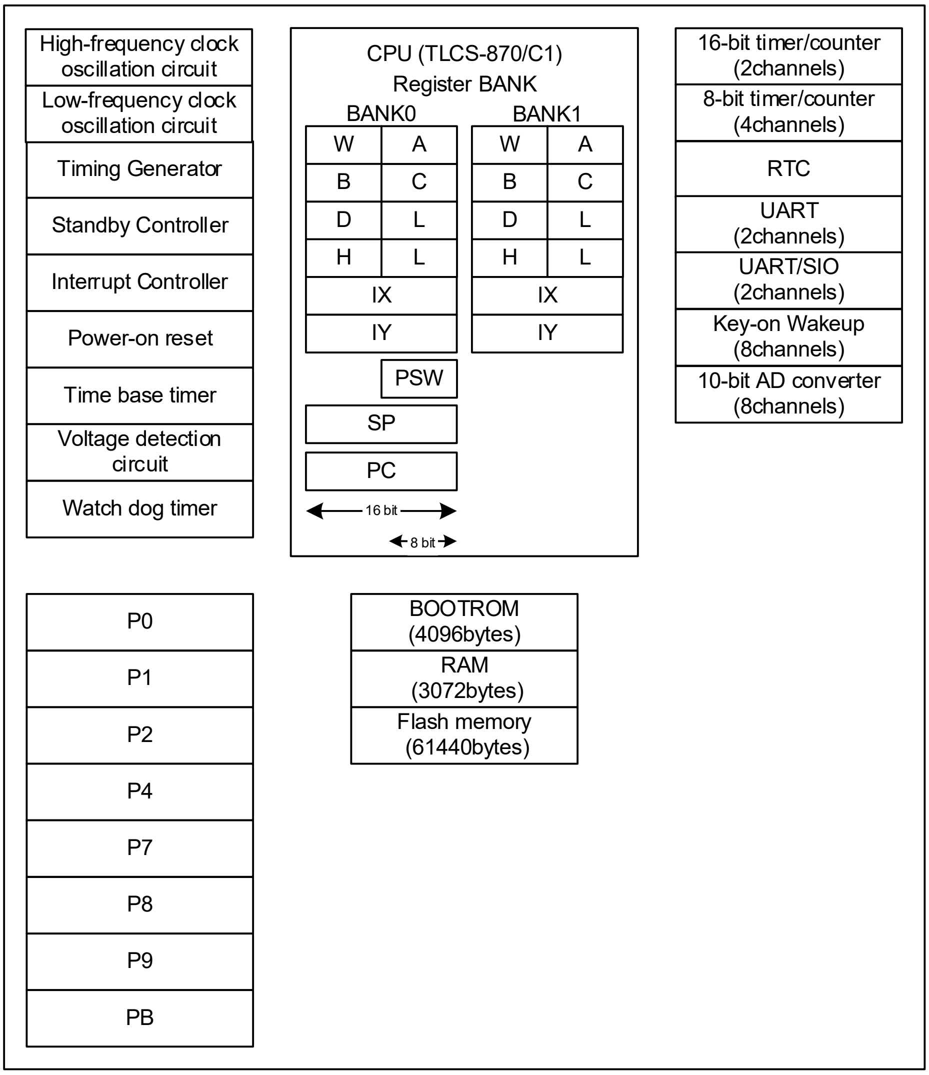


Figure 1-5 Block Diagram

1.4.3. Block Diagram of the TMP89FS63B

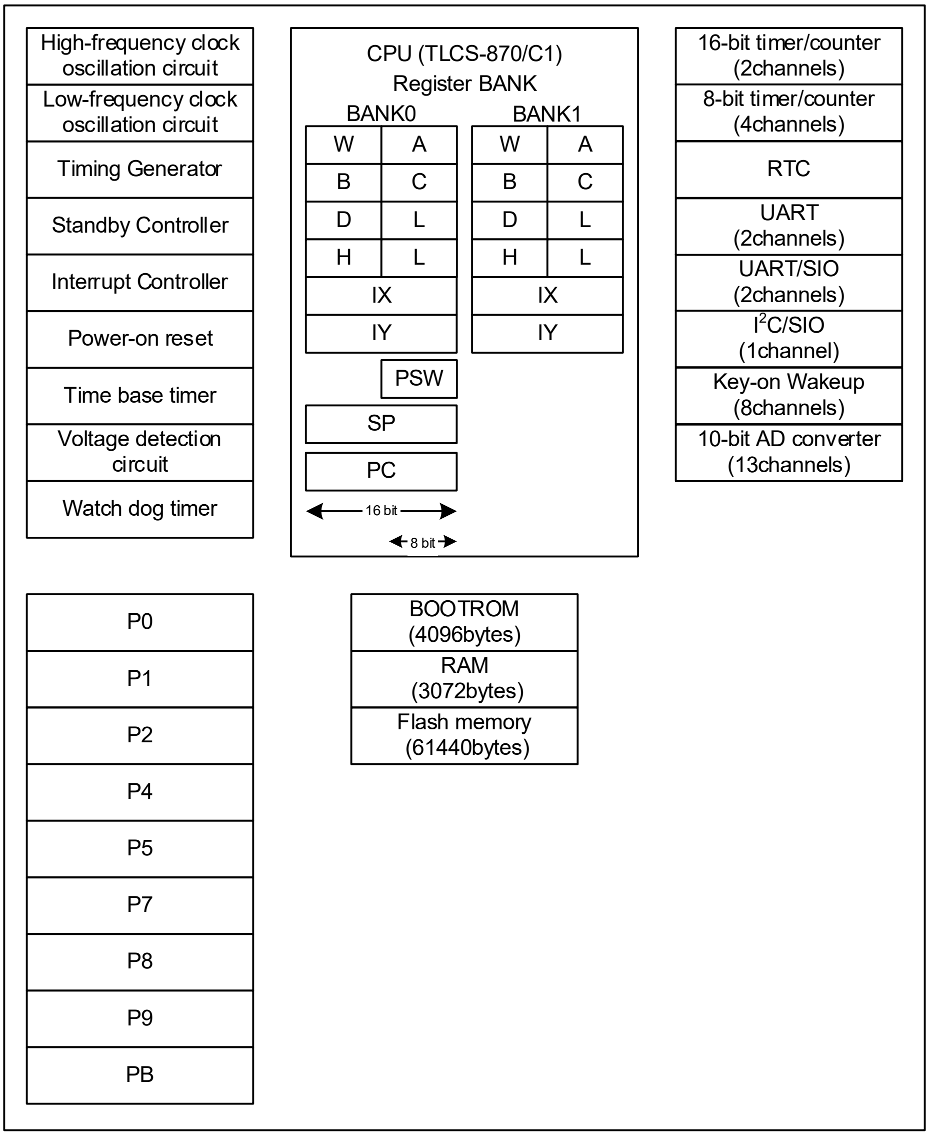


Figure 1-6 Block Diagram

1.5. Pin Names and Functions

The TMP89FS60B/62B/63B have the MCU mode and the Serial PROM mode.

Table 1-2 shows the pin functions in MCU mode.

Regarding to the pin function in the Serial PROM mode, refer to “22. Serial PROM Mode”.

Table 1-2 Pin Function and Number of pin in the MCU mode

NA: Not available
I/O: input and output, I: input, O: output

A number of pin for each product			Pin Name	Input /Output	Pin Functions
TMP89FS60B	TMP89FS62B	TMP89FS63B			
7	6	6	P03 XTOUT	I/O O	PORT03 The pin connected with a low-frequency resonator.
6	5	5	P02 XTIN	I/O I	PORT02 The pin connected with a low-frequency resonator.
3	2	2	XOUT	O	The pin connected with a high-frequency resonator.
2	1	1	XIN	I	The pin connected with a high-frequency resonator.
11	NA	NA	P13 INT1	I/O I	PORT13 External interrupt 1 input
10	NA	NA	P12 $\overline{\text{INT0}}$	I/O I	PORT12 External interrupt 0 input
9	8	8	P11 $\overline{\text{INT5}}$ $\overline{\text{STOP}}$	I/O I I	PORT11 External interrupt 5 input STOP mode release input
8	7	7	P10 $\overline{\text{RESET}}$	I/O I	PORT10 Reset signal input
19	NA	NA	P27	I/O	PORT27
18	NA	NA	P26	I/O	PORT26
17	NA	14	P25 SCLK0	I/O I/O	PORT25 Serial clock input/output 0
16	NA	13	P24 SCL0 SI0	I/O I/O I	PORT24 I^{2}C bus clock input/output 0 (Note 1) Serial data input 0 (Note 1)
15	12	12	P23 SDA0 SO0	I/O I/O O	PORT23 I^{2}C bus data input/output 0 (Note 1) Serial data output 0 (Note 1)
14	11	11	P22 SCLK0	I/O I/O	PORT22 Serial clock input/output 0
13	10	10	P21 RXD0 TXD0 SI0 OCDIO	I/O I O I I/O	PORT21 UART data input 0 UART data output 0 Serial data input 0 OCD data input/output

NA: Not available
I/O: input and output, I: input, O: output

A number of pin for each product			Pin Name	Input /Output	Pin Functions
TMP89FS60B	TMP89FS62B	TMP89FS63B			
12	9	9	P20 TXD0 RXD0 SO0 OCDCK	I/O O I O I	PORT20 UART data output 0 UART data input 0 Serial data output 0 OCD clock input
30	22	25	P47 AIN7 KWI7	I/O I I	PORT47 Analog input 7 Key-on wake-up input 7
29	21	24	P46 AIN6 KWI6	I/O I I	PORT46 Analog input 6 Key-on wake-up input 6
28	20	23	P45 AIN5 KWI5	I/O I I	PORT45 Analog input 5 Key-on wake-up input 5
27	19	22	P44 AIN4 KWI4	I/O I I	PORT44 Analog input 4 Key-on wake-up input 4
26	18	21	P43 AIN3 KWI3	I/O I I	PORT43 Analog input 3 Key-on wake-up input 3
25	17	20	P42 AIN2 KWI2	I/O I I	PORT42 Analog input 2 Key-on wake-up input 2
24	16	19	P41 AIN1 KWI1	I/O I I	PORT41 Analog input 1 Key-on wake-up input 1
23	15	18	P40 AIN0 KWI0	I/O I I	PORT40 Analog input 0 Key-on wake-up input 0
38	NA	NA	P57	I/O	PORT57
			AIN15	I	Analog input 15
37	NA	NA	P56	I/O	PORT56
			AIN14	I	Analog input 14
36	NA	NA	P55 AIN13	I/O I	PORT55 Analog input 13
35	NA	30	P54 AIN12	I/O I	PORT54 Analog input 12
34	NA	29	P53 AIN11	I/O I	PORT53 Analog input 11

NA: Not available
I/O: input and output, I: input, O: output

A number of pin for each product			Pin Name	Input /Output	Pin Functions
TMP89FS60B	TMP89FS62B	TMP89FS63B			
33	NA	28	P52 AIN10	I/O I	PORT52 Analog input 10
32	NA	27	P51 AIN9	I/O I	PORT51 Analog input 9
31	NA	26	P50 AIN8	I/O I	PORT50 Analog input 8
46	30	38	P77 INT4	I/O I	PORT77 External interrupt 4 input
45	29	37	P76 INT3	I/O I	PORT76 External interrupt 3 input
44	28	36	P75 INT2	I/O I	PORT75 External interrupt 2 input
43	27	35	P74 $\overline{DVO}$	I/O O	PORT74 Divider output
42	26	34	P73 TCA1 $\overline{PPGA1}$	I/O I O	PORT73 TCA1 input PPGA1 output
41	25	33	P72 TCA0 $\overline{PPGA0}$	I/O I O	PORT72 TCA0 input PPGA0 output
40	24	32	P71 TC01 $\overline{PPG01}$ $\overline{PWM01}$	I/O I O O	PORT71 TC01 input PPG01 output PWM01 output
39	23	31	P70 TC00 $\overline{PPG00}$ $\overline{PWM00}$	I/O I O O	PORT70 TC00 input PPG00 output PWM00 output
51	NA	NA	P84	I/O	PORT84
50	NA	NA	P83	I/O	PORT83
49	NA	NA	P82	I/O	PORT82
48	32	40	P81 TC03 $\overline{PPG03}$ $\overline{PWM03}$	I/O I O O	PORT81 TC03 input PPG03 output PWM03 output
47	31	39	P80 TC02 $\overline{PPG02}$ $\overline{PWM02}$	I/O I O O	PORT80 TC02 input PPG02 output PWM02 output

NA: Not available
I/O: input and output, I: input, O: output

A number of pin for each product			Pin Name	Input /Output	Pin Functions
TMP89FS60B	TMP89FS62B	TMP89FS63B			
56	37	45	P94	I/O	PORT94
			RXD2	I	UART data input 2
			TXD2	O	UART data output 2
55	36	44	P93	I/O	PORT93
			TXD2	O	UART data output 2
			RXD2	I	UART data input 2
54	35	43	P92	I/O	PORT92
			SCLK1	I/O	Serial clock input/output 1
53	34	42	P91	I/O	PORT91
			RXD1	I	UART data input 1
			TXD1	O	UART data output 1
			SI1	I	Serial data input 1
52	33	41	P90	I/O	PORT90
			TXD1	O	UART data output 1
			RXD1	I	UART data input 1
			SO1	O	Serial data output 1
64	NA	NA	PB7	I/O	PORTB7
63	NA	NA	PB6	I/O	PORTB6
			SCLK0	I/O	Serial clock input/output 0 (Note 2)
62	43	51	PB5	I/O	PORTB5
			RXD0	I	UART data input 0
			TXD0	O	UART data output 0
			SI0	I	Serial data input 0 (Note 2)
61	42	50	PB4	I/O	PORTB4
			TXD0	O	UART data output 0
			RXD0	I	UART data input 0
			SO0	O	Serial data output 0 (Note 2)
60	41	49	PB3	I/O	PORTB3
59	40	48	PB2	I/O	PORTB2
58	39	47	PB1	I/O	PORTB1
57	38	46	PB0	I/O	PORTB0
4	3	3	MODE	I	Test pin for shipping test This pin must keep “Low” level.
22	14	17	VAREF	-	Analog reference voltage input pin for AD conversion
21		16	AVDD	-	Power supply pin for the analog circuits
20	13	15	AVSS	-	GND pin for the analog circuits
5	4	4	VDD	-	Power supply pin
1	44	52	VSS	-	GND pin

Note 1: This function is not available for the TMP89FS62B.

Note 2: This function is not available for the TMP89FS62B and TMP89FS63B.

1.6. Regarding “Reserved” in this data sheet

This data sheet explains three products; TMP89FS60B, TMP89FS62B and TMP89FS63B.

These products have some different peripheral circuits each other. Therefore, the particular addresses, functions and bit symbol of a register cannot be accessed. And it is necessary that they are written a specific value.

These limitations are described as some kinds of “Reserved” shown below;

(1) In case only “Reserved” is written

1. In case “Reserved” is written with the address, do not write to or read from the address.

Example: Regarding the below SRF1 table, “Reserved” is written with address 0x0003. Therefore, do not write to or read from address 0x0003.

Table x-x SFR1

Address	Register Name
0x0000	P0DR
0x0001	P1DR
0x0002	P2DR
0x0003	Reserved
0x0004	P4DR
0x0005	P5DR

Address	Register Name
0x0020	SIO0SR
0x0021	SIO0BUF
0x0022	SBI0CR1
0x0023	SBI0CR2/SBI0SR2
0x0024	I2C0AR
0x0025	SBI0DBR

2. In case “Reserved” is written with the function, this function is not available.

Example: Regarding the below all interrupt sources of the TMP89FS60B/62B/63B table, INTSBI0/INTSIO0 is “Reserved” for TMP89FS62B. Therefore, INTSBI0/INTSIO0 for the TMP89FS62B is not available and cannot be used. And, because INT0 and INT1 are “Reserved” for the TMP89FS62B and TMP89FS63B, INT0 and INT1 for the TMP89FS62B and TMP89FS63B are not available and cannot be used.

Table x-x All interrupt sources of the TMP89FS60B/62B/63B

Basic priority	Interrupt sources		Enable condition	Interrupt latch	Vector Address (MCU mode)	
					SYSSR4 <RVCTR> = 0	SYSSR4 <RVCTR> = 1
16	Internal	INTSBI0/INTSIO0 (Note 4)	<IMF> AND EIRH<EF15> = 1	ILH<IL15>	0xFFE0	0x01E0
17	External	INT0 (Note 5)	<IMF> AND EIRE<EF16> = 1	ILE<IL16>	0xFFDE	0x01DE
18	External	INT1 (Note 5)	<IMF> AND EIRE<EF17> = 1	ILE<IL17>	0xFFDC	0x01DC

Note 4: The INTSBI0/INTSIO0 is “Reserved” for the TMP89FS62B.

Note 5: The INT0 and INT1 is “Reserved” for the TMP89FS62B and TMP89FS63B.

3. In case "Reserved" is written with the value which is set to a bit symbol, do not set the value to the bit symbol.

Example: Regarding the value of <SAIN> in the below table, do not set to <SAIN> from 0x8 to 0xF for the TMP89FS62B and do not set to <SAIN> from 0xD to 0xF for the TMP89FS63B.

AD converter control register 1

ADCCR1		7	6	5	4	3	2	1	0
(0x0034)	Bit Symbol	ADRS	AMD		AINEN	SAIN			
	Read/Write	R/W	R/W		R/W	R/W			
	After reset	0	0	0	0	0	0	0	0

ADRS	AD conversion start	0: 1:	- AD conversion start		
AMD	AD operating mode	00: 01:	AD operation disable, forcibly stop AD operation Single mode		
		10: 11:	Reserved Repeat mode		
AINEN	Analog input control	0: 1:	Analog input disable Analog input enable		
SAIN	Analog input channel select		TMP89FS60B	TMP89FS62B	TMP89FS63B
		0000:	AIN0	AIN0	AIN0
		0001:	AIN1	AIN1	AIN1
		0010:	AIN2	AIN2	AIN2
		0011:	AIN3	AIN3	AIN3
		0100:	AIN4	AIN4	AIN4
		0101:	AIN5	AIN5	AIN5
		0110:	AIN6	AIN6	AIN6
		0111:	AIN7	AIN7	AIN7
		1000:	AIN8	Reserved	AIN8
		1001:	AIN9	Reserved	AIN9
		1010:	AIN10	Reserved	AIN10
		1011:	AIN11	Reserved	AIN11
		1100:	AIN12	Reserved	AIN12
		1101:	AIN13	Reserved	Reserved
		1110:	AIN14	Reserved	Reserved
		1111:	AIN15	Reserved	Reserved

(2) In case “Reserved” with “(this bit symbol must be cleared to “0”.)” or “(this bit symbol must be set to “1”.)” is written

1. In case of “Reserved (this bit symbol must be clear to “0”.)”

Example: Regarding the Interrupt priority change control register 4, ILPRS4<IL17P> and ILPRS4<IL16P> are “Reserved (this bit symbol must be cleared to “0”.)”. Therefore, clear <IL17P> and <IL16P> are cleared to “00” when set an immediate value to ILPRS4 or other bit symbol in the register which includes these bits is modified.

Interrupt priority change control register 4

ILPRS4 (0x0FF3)		7	6	5	4	3	2	1	0
	Bit Symbol	IL19P		IL18P		IL17P		IL16P	
	Read/Write	R/W		R/W		R/W		R/W	
	After reset	0	0	0	0	0	0	0	0

IL19P	Sets the interrupt priority of <IL19>.	00:	Level 0 (lower priority)
IL18P	Sets the interrupt priority of <IL18>.	01:	Level 1
IL17P (Note)	Sets the interrupt priority of <IL17>.	10:	Level 2
IL16P (Note)	Sets the interrupt priority of <IL16>.	11:	Level 3 (higher priority)

Note: <IL17P> and <IL16P> are “Reserved (this bit symbol must be cleared to “0”.)” for the TMP89FS62B and the TMP89FS63B.

2. In case of “Reserved (this bit symbol must be set to “1”.)”

Example: Regarding the port P2 input/output control, P2CR<P2CR7> and P2CR<P2CR6> are “Reserved (this bit symbol must be set to “1”.)”. Therefore, clear <P2CR7> and <P2CR6> are set to “1” when set an immediate value to P2CR or other bit symbol in the register which includes these bits is modified.

Port P2 input/output control

P2CR		7	6	5	4	3	2	1	0
(0x0F1C)	Bit Symbol	P2CR7	P2CR6	P2CR5	P2CR4	P2CR3	P2CR2	P2CR1	P2CR0
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

			TMP89FS60B	TMP89FS62B	TMP89FS63B
P2CR7	Selects input mode or output mode.	0: 1:	Input mode Output mode	Reserved (this symbol must be set to “1”.)	
P2CR6	Selects input mode or output mode.	0: 1:	Input mode Output mode	Reserved (this symbol must be set to “1”.)	
P2CR5	Selects input mode or output mode.	0: 1:	Input mode SCLK0 (I) Output mode SCLK0 (O)	Reserved (this symbol must be set to “1”.)	Input mode SCLK0 (I) Output mode SCLK0 (O)
P2CR4	Selects input mode or output mode.	0: 1:	Input mode SI0 (I) Output mode SCL0 (I/O)	Reserved (this symbol must be set to “1”.)	Input mode SI0 (I) Output mode SCL0 (I/O)

2. CPU Core

2.1. Configuration

The CPU core consists of the CPU, the system clock control circuit and the reset control circuit.

This chapter describes the CPU core address space, the system clock control circuit and the reset control circuit.

2.2. Memory Space

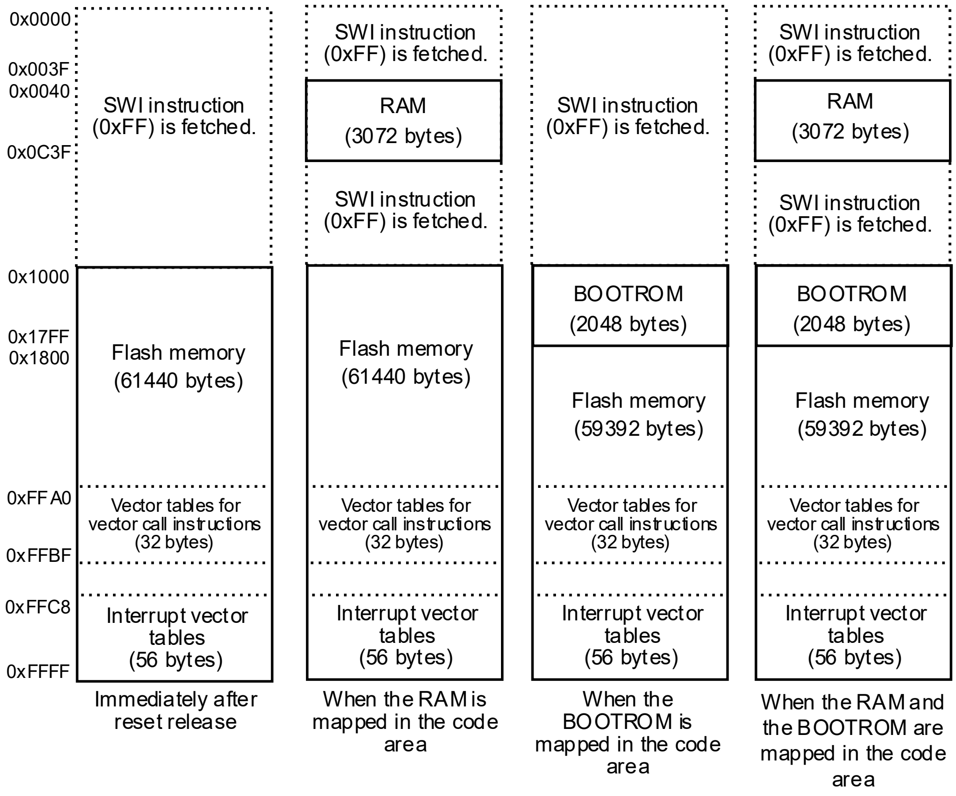
The TLCS-870/C1 CPU memory space consists of the code area to be accessed as instruction operation codes and operands, and the data area to be accessed as sources and destinations of transfer and calculation instructions.

Both the code and data areas have independent 64-Kbytes address spaces.

2.2.1. Code Area

The code area stores operation codes, operands, vector tables for vector call instructions and interrupt vector tables.

The RAM, the BOOTROM and the Flash memory are mapped in the code area.


0x0000
0x003F
0x0040
0x0C3F
0x1000
0x17FF
0x1800
0xFFA0
0xFFBF
0xFFC8
0xFFFF
SWI instruction (0xFF) is fetched.
RAM (3072 bytes)
BOOTROM (2048 bytes)
Flash memory (61440 bytes)
Flash memory (59392 bytes)
Vector tables for vector call instructions (32 bytes)
Interrupt vector tables (56 bytes)
Immediately after reset release
When the RAM is mapped in the code area
When the BOOTROM is mapped in the code area
When the RAM and the BOOTROM are mapped in the code area

Note: Only the first 2 Kbytes of the BOOTROM are mapped in the memory map, except in the Serial PROM mode.

Figure 2-1 Memory Map in the Code Area

2.2.1.1. RAM

The RAM is mapped in the data area immediately after reset release.

By setting SYSCR3<RAREA> to "1" and writing "0xD4" to SYSCR4, RAM can be mapped to "0x0040" to "0x0C3F" in the code area to execute the program.

At this time, by setting SYSCR3<RVCTR> to "1" and writing "0xD4" to SYSCR4, vector table for vector call instructions and interrupt except reset can be mapped to RAM.

For the vector call instruction, refer to “TLCS-870/C1 series CPU”, and for the interrupt vector table, refer to "3. Interrupt Control Circuit".

Note 1: When the RAM is not mapped in the code area, the SWI instruction is fetched from "0x0040" to "0x0C3F".

Note 2: The contents of the RAM become unstable when the power is turned on and immediately after a reset is released. To execute the program by using the RAM, transfer the program to be executed in the initialization routine.

System control register 3

SYSCR3		7	6	5	4	3	2	1	0
(0x0FDE)	Bit Symbol	-	-	-	-	-	RVCTR	RAREA	(RSTDIS)
	Read/Write	R	R	R	R	R	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

RAREA	Specifies mapping of the RAM in the code area	0:	The RAM is not mapped from "0x0040" to "0x0C3F" in the code area.	
		1:	The RAM is mapped from "0x0040" to "0x0C3F" in the code area.	
RVCTR	Specifies mapping of the vector table for vector call instructions and interrupts		Vector table for vector call instructions	Vector table for interrupt
		0:	"0xFFA0" to "0xFFBF" in the code area	"0xFFC8" to "0xFFFF" in the code area
		1:	"0x01A0" to "0x01BF" in the code area	"0x01C8" to "0x01FD" in the code area

Note 1: The value of SYSCR3<RAREA> is invalid until "0xD4" is written into SYSCR4.

Note 2: To assign the vector tables to RAM, write “0xD4” to SYSCR4 after setting SYSCR3<RVCTR> and SYSCR3<RAREA> to "1".

Note 3: When a read instruction is executed on SYSCR3, bits 7 to 3 are read as "0".

System control register 4

SYSCR4		7	6	5	4	3	2	1	0
(0x0FDF)	Bit Symbol	SYSCR4							
	Read/Write	W							
	After reset	0	0	0	0	0	0	0	0

SYSCR4	Writes the control code to make the contents in SYSCR3 and IRSTSR valid	0xB2:	Enables the contents of SYSCR3<RSTDIS>.
		0xD4:	Enables the contents of SYSCR3<RAREA> and SYSCR3 <RVCTR>.
		0x71:	Enables the contents of IRSTSR<FCLR>.
		Others:	Invalid

Note 1: SYSCR4 is a write-only register, and must not be accessed by using a read-modify-write instruction, such as a bit operation.

Note 2: When the SYSCR4 is written to “0xB2” (Enable code for SYSCR3<RSTDIS>) after SYSCR3<RSTDIS> is modified, these operation should be executed continuously in NORMAL mode with fcgck = fc/4 (CGCR<FCGCKSEL> = 00). Otherwise, SYSCR3<RSTDIS> may be enabled at unexpected timing.

Note 3: When the SYSCR4 is written to “0x71” (Enable code for IRSTSR<FCLR>) after IRSTSR<FCLR> is modified, these operation should be executed continuously in NORMAL mode with fcgck = fc/4 (CGCR<FCGCKSEL> = 00). Otherwise, IRSTSR<FCLR> may be enabled at unexpected timing.

System control status register 4

SYSSR4		7	6	5	4	3	2	1	0
(0x0FDF)	Bit Symbol	-	-	-	-	-	RVCTRS	RAREAS	(RSTDIS)
	Read/Write	R	R	R	R	R	R	R	R
	After reset	0	0	0	0	0	0	0	0

RAREAS	Status of mapping of the RAM in the code area	0:	The enabled SYSCR3<RAREA> data is "0".
		1:	The enabled SYSCR3<RAREA> data is "1".
RVCTRS	Status of mapping of the vector table for the vector call instruction and interrupts	0:	The enabled SYSCR3<RVCTR> data is "0".
		1:	The enabled SYSCR3<RVCTR> data is "1".

Note: When a read instruction is executed on SYSSR4, bits 7 to 3 are read as "0".

Example: Transfer the program stored in the data area to the RAM.

```
            LD     HL,TRANSFER_START_ADDRESS   ; Destination RAM address
            LD     DE,PROGRAM_START_ADDRESS    ; Source ROM address
            LD     BC,BYTE_OF_PROGRAM          ; Number of bytes of the transferred program -1
TRANS_RAM:  LD     A,(DE)                      ; Reading the program to be transferred
            LD     (HL),A                      ; Writing the program to be transferred
            INC    HL                          ; Destination address increment
            INC    DE                          ; Source address increment
            DEC    BC                          ; Have all the programs been transferred?
            J      F,TRANS_RAM
```

2.2.1.2. BOOTROM

The BOOTROM is not mapped in the code area or the data area after reset release.

Setting FLSCR1<BAREA> to "1" and writing "0xD5" to FLSCR2 map the BOOTROM to "0x1000" to "0x17FF" in the code area and to "0x1000" to "0x17FF" in the data area. Flash memory can be easily erased and written by using the API (Application Programming Interface) contained in the BOOTROM.

Note 1: When the BOOTROM is not mapped in the code area, an instruction is fetched from the Flash memory or an SWI instruction is fetched, depending on the capacity of the internal Flash memory.

Note 2: Only the first 2 Kbytes of the BOOTROM are mapped in the memory map, except in the Serial PROM mode.

Flash memory control register 1

FLSCR1		7	6	5	4	3	2	1	0
(0x0FD0)	Bit Symbol	(FLSMD)			BAREA	(FAREA)		(ROMSEL)	
	Read/Write	R/W			R/W	R/W		R/W	
	After reset	0	1	0	0	0	0	0	0

BAREA	Specifies mapping of the BOOTROM in the code and data areas	0:	The BOOTROM is not mapped to "0x1000" to "0x17FF" in the code area and to "0x1000" to "0x17FF" in the data area.
		1:	The BOOTROM is mapped to "0x1000" to "0x17FF" in the code area and to "0x1000" to "0x17FF" in the data area.

Note: The contents of FLSCR1 is invalid until "0xD5” is written into FLSCR2.

Flash memory control register 2

FLSCR2		7	6	5	4	3	2	1	0
(0x0FD1)	Bit Symbol	CR1EN							
	Read/Write	W							
	After reset	*	*	*	*	*	*	*	*

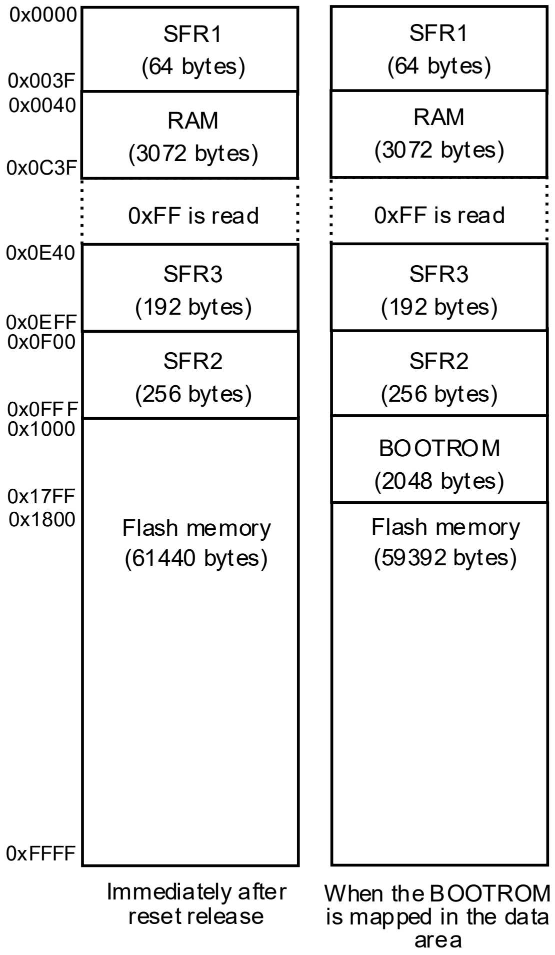
CR1EN	Writes the control code to make the contents in FLSCR1 valid	0xD5:	Enables the contents of FLSCR1.
		Others:	invalid

2.2.1.3. Flash Memory

The Flash memory is mapped to "0x1000" to "0xFFFF" in the code area after reset release.

2.2.2. Data Area

The data area stores the data to be accessed as sources and destinations of transfer and calculation instructions. The SFR, the RAM, the BOOTROM and the Flash memory are mapped in the data area.



Note: Only the first 2 Kbytes of the BOOTROM are mapped in the memory map, except in the Serial PROM mode.

Figure 2-2 Memory Map in the Data Area

2.2.2.1. SFR

The SFR is mapped to "0x0000" to "0x003F" (SFR1), "0x0F00" to "0x0FFF" (SFR2) and "0x0E40" to "0x0EFF" (SFR3) in the data area after reset release.

Note: Don’t read from or write to the address of the SFR with “Reserved”.

2.2.2.2. RAM

The RAM is mapped to "0x0040" to "0x0C3F" in the data area after reset release.

Note: The contents of the RAM become unstable when the power is turned on and immediately after a reset is released. When the program in the RAM is executed, it should be transferred to the RAM before executing.

Example: Initialize contents of the RAM

```
            LD     HL,RAM_TOP_ADDRESS        ; Head of address of the RAM to be initialized
            LD     A, 0x00                   ; Initialization data
            LD     BC,BYTE_OF_CLEAR_BYTES    ; Number of bytes of RAM to be initialized -1
CLR_RAM:    LD     (HL),A                    ; Initialization of the RAM
            INC    HL                        ; Initialization address increment
            DEC    BC                        ; Have all contents of the RAM been initialized?
            J      F,CLR_RAM
```

2.2.2.3. BOOTROM

The BOOTROM is not mapped in the code area and the data area after reset release.

Setting FLSCR1<BAREA> to "1" and writing "0xD5" to FLSCR2 maps the BOOTROM to "0x1000" to "0x17FF" in the code area and to "0x1000" to "0x17FF" in the data area. Flash memory can be easily erased and written by using the API (Application Programming Interface) contained in the BOOTROM.

Note: Only the first 2 Kbytes of the BOOTROM are mapped in the memory map, except in the Serial PROM mode.

Flash memory control register 1

FLSCR1		7	6	5	4	3	2	1	0
(0x0FD0)	Bit Symbol	(FLSMD)			BAREA	(FAREA)		(ROMSEL)	
	Read/Write	R/W			R/W	R/W		R/W	
	After reset	0	1	0	0	0	0	0	0

BAREA	Specifies mapping of the BOOTROM in the code and data areas	0: 1:	The BOOTROM is not mapped to "0x1000" to "0x17FF" in the code area and to "0x1000" to "0x17FF" in the data area. The BOOTROM is mapped to "0x1000" to "0x17FF" in the code area and to "0x1000" to "0x17FF" in the data area.

Note: The Flash memory control register 1 has a double-buffer structure comprised of the register FLSCR1 and a shift register. Writing "0xD5" to the register FLSCR2 allows a register setting to be reflected and take effect in the shift register. This means that a register setting value does not take effect until "0xD5" is written to the register FLSCR2.

Flash memory control register 2

FLSCR2		7	6	5	4	3	2	1	0
(0x0FD1)	Bit Symbol	CR1EN							
	Read/Write	W							
	After reset	*	*	*	*	*	*	*	*

CR1EN	Writes the control code to make the contents in FLSCR1 valid	0xD5: Others:	Enables the contents of FLSCR1. Invalid

2.2.2.4. Flash Memory

The Flash memory is mapped to "0x1000" to "0xFFFF" in the data area after reset release.

2.3. System Clock Controller Circuit

2.3.1. Configuration

The system clock controller circuit consists of a clock generator, a clock gear, a timing generator, a warm-up counter and an operation mode control circuit.

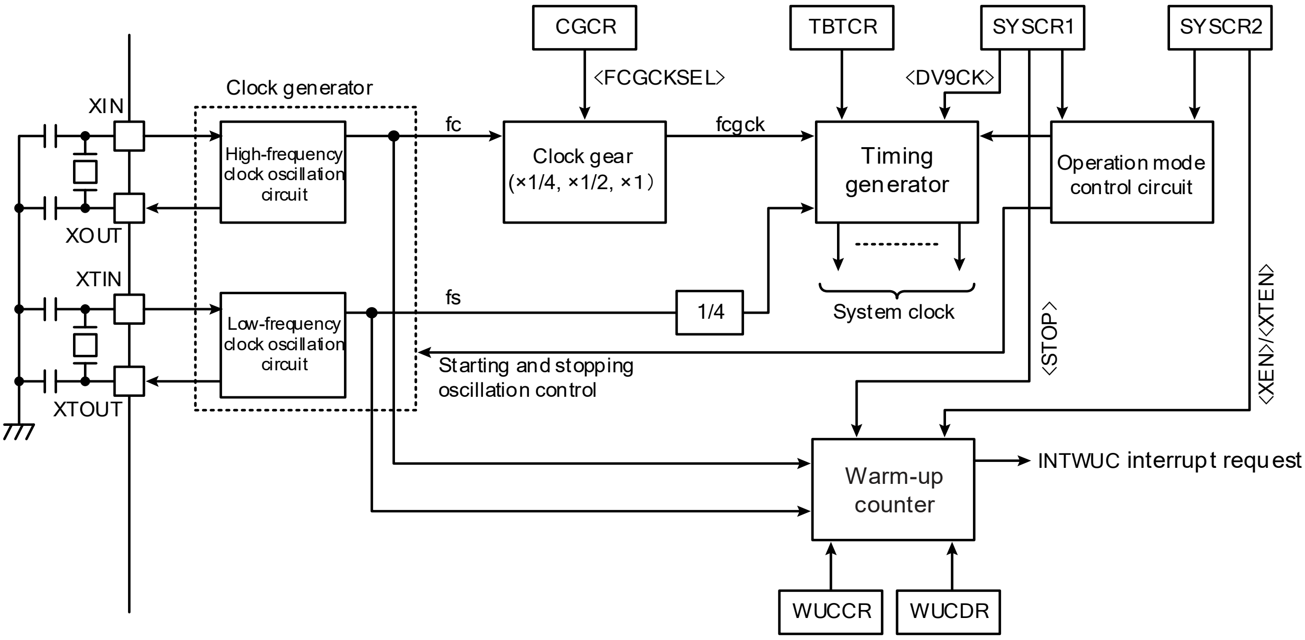

CGCR
TBTCR
SYSCR1
SYSCR2
<FCGCKSEL>
<DV9CK>
Clock generator
XIN
XOUT
XTIN
XTOUT
High-frequency clock oscillation circuit
Low-frequency clock oscillation circuit
fc
fcgck
fs
Clock gear (×1/4, ×1/2, ×1)
Timing generator
Operation mode control circuit
1/4
System clock
Starting and stopping oscillation control
<STOP>
<XEN>/<XTEN>
Warm-up counter
INTWUC interrupt request
WUCCR
WUCDR

Figure 2-3 System Clock Controller circuit

2.3.2. Control

The system clock controller circuit is controlled by system control register 1 (SYSCR1), system control register 2 (SYSCR2), the warm-up counter control register (WUCCR), the warm-up counter data register (WUCDR) and the clock gear control register (CGCR).

System control register 1

SYSCR1		7	6	5	4	3	2	1	0
(0x0FDC)	Bit Symbol	STOP	RELM	OUTEN	DV9CK	-	-	-	-
	Read/Write	R/W	R/W	R/W	R/W	R	R	R	R
	After reset	0	0	0	0	1	0	0	0

STOP	Activates the STOP mode	0: 1:	Operates the CPU and the peripheral circuits. Stops the CPU and the peripheral circuits (activate the STOP mode).
RELM	Selects the STOP mode release method	0: 1:	Edge-sensitive release mode (Release the STOP mode at the rising edge of the STOP mode release signal) Level-sensitive release mode (Release the STOP mode at the “High” level of the STOP mode release signal)
OUTEN	Selects the port output state in the STOP mode	0: 1:	Output state is high impedance. Output state is held.
DV9CK	Selects the input clock to stage 9 of the divider	0: 1:	fcgck/2^9 fs/4

Note 1: fcgck: Gear clock [Hz], fs: Low-frequency clock [Hz]

Note 2: When a read instruction is executed on SYSCR1, bits 2 to 0 are read as "0" and bit 3 is read as “1”.

Note 3: When the STOP mode is activated by setting SYSCR1<OUTEN> to "0", the internal input level of the port is fixed to "Low" level. Therefore, the internal input level of the port may fall down depending on the pin state when the STOP mode is activated. For that reason, the interrupt latch of an external interrupt whose factor is the falling edge may be set

Note 4: The P11 pin is also used as the $\overline{\text{STOP}}$ pin. When the STOP mode is activated, the pin reverts to high impedance and is put in input mode, regardless of the value of SYSCR1<OUTEN>.

Note 5: Writing of the second byte data will be executed improperly when the operation is switched to the STOP state by an instruction, such as LDW, which executes 2-byte data transfer at a time.

Note 6: Don't set SYSCR1<DV9CK> to "1" before the oscillation of the low-frequency clock oscillation circuit becomes stable.

Note 7: In the SLOW1/2 or SLEEP1 mode, fs / 4 is input to stage 9 of the divider, regardless of the value of SYSCR1 <DV9CK>.

System control register 2

SYSCR2		7	6	5	4	3	2	1	0
(0x0FDD)	Bit Symbol	-	XEN	XTEN	SYSCK	IDLE	TGHALT	-	-
	Read/Write	R	R/W	R/W	R/W	R/W	R/W	R	R
	After reset	0	1	0	0	0	0	0	0

XEN	Controls the high-frequency clock oscillation circuit	0: 1:	Stops oscillation Continues or starts oscillation
XTEN	Controls the low-frequency clock oscillation circuit	0: 1:	Stops oscillation Continues or starts oscillation
SYSCK	Selects the source clock of the main system clock	0: 1:	Gear clock (fcgck) (NORMAL1/2 or IDLE1/2 mode) Quarter of Low-frequency clock (fs/4) (SLOW1/2 or SLEEP1 mode)
IDLE	Controls the CPU and the WDT (IDLE1/2 or SLEEP1 mode)	0: 1:	Enables the CPU and the WDT Stops the CPU and the WDT (Activates IDLE1/2 or SLEEP1 mode)
TGHALT	Controls the TG (IDLE0 or SLEEP0 mode)	0: 1:	Enables the clock supply from the TG to all the peripheral circuits Disables the clock supply from the TG to the peripheral circuits except the TBT (Activate IDLE0 or SLEEP0 mode)

Note 1: fcgck: Gear clock [Hz], fs: Low-frequency clock [Hz]

Note 2: WDT: Watchdog timer, TG: Timing generator

Note 3: Don't set both SYSCR2<IDLE> and SYSCR2<TGHALT> to "1" simultaneously.

Note 4: Writing of the second byte data will be executed improperly when the operation is switched to the IDLE state by an instruction, such as LDW, which executes 2-byte data transfer at a time.

Note 5: When the IDLE1/2 or SLEEP1 mode is released, SYSCR2<IDLE> is cleared to "0" automatically.

Note 6: When the IDLE0 or SLEEP0 mode is released, SYSCR2<TGHALT> is cleared to "0" automatically.

Note 7: When a read instruction is executed on SYSCR2, bits 7, 1 and 0 are read as "0".

Warm-up counter control register

WUCCR		7	6	5	4	3	2	1	0
(0x0FCD)	Bit Symbol	WUCRST	-	-	-	WUCDIV		WUCSEL	-
	Read/Write	W	R	R	R	R/W		R/W	R
	After reset	0	0	0	0	1	1	0	1

WUCRST	Resets and stops the warm-up counter	0:	-
		1:	Clears and stops the counter
WUCDIV	Selects the frequency division rate of the warm-up counter source clock	00:	Source clock
		01:	Source clock / 2
		10:	Source clock / 2^2
		11:	Source clock / 2^3
WUCSEL	Selects the warm-up counter source clock	0:	High-frequency clock (fc)
		1:	Low-frequency clock (fs)

Note 1: fc: High-frequency clock [Hz], fs: Low-frequency clock [Hz]

Note 2: WUCCR<WUCRST> is cleared to "0" automatically. There is no need to clear it to "0" after it is set to "1".

Note 3: When a read instruction is executed on WUCCR, bits 6 to 4 are read as "0".

Note 4: Before starting the warm-up counter operation, set the source clock and the frequency division rate at WUCCR and set the warm-up time at WUCDR.

Warm-up counter data register

WUCDR		7	6	5	4	3	2	1	0
(0x0FCE)	Bit Symbol	WUCDR							
	Read/Write	R/W							
	After reset	0	1	1	0	0	1	1	0

WUCDR	Warm-up time setting

Note: Don't start the warm-up counter operation with WUCDR set at "0x00".

Clock gear control register

CGCR		7	6	5	4	3	2	1	0
(0x0FCF)	Bit Symbol	-	-	-	-	-	-	FCGCKSEL	
	Read/Write	R	R	R	R	R	R	R/W	
	After reset	0	0	0	0	0	0	0	0

FCGCKSEL	Clock gear setting	00:	fc / 4
		01:	fc / 2
		10:	fc
		11:	Reserved

Note 1: fcgck: Gear clock [Hz], fc: High-frequency clock [Hz]

Note 2: Don't change CGCR<FCGCKSEL> in the SLOW mode.

Note 3: When a read instruction is executed on CGCR, bits 7 to 2 are read as "0".

2.3.3. Functions

2.3.3.1. Clock generator

The clock generator generates the basic clock for the system clocks to be supplied to the CPU core and peripheral circuits.

It contains two oscillation circuits: one for the high-frequency clock and the other for the low-frequency clock.

The oscillation circuit pins are shared with ports P0. For the setting to use them as ports, refer to "8. I/O Ports".

To use ports P02 and P03 as the low-frequency clock oscillation circuit (the XTIN and XTOUT pins), set <P0FC2> to "1" and then set SYSCR2<XTEN> to "1".

The high-frequency (fc) clock and the low-frequency (fs) clock can easily be obtained by connecting a resonator between the XIN and XOUT pins and between the XTIN and XTOUT pins respectively.

It is possible to apply a clock from an external oscillator . In this case, external clocks are applied to the XIN/XTIN pins and the XOUT/XTOUT pins are kept open.

Enabling/disabling the oscillation of the high-frequency clock oscillation circuit and the low-frequency clock oscillation circuit, and switching the pin function of the low-frequency clock oscillation circuit to ports are controlled by the software and hardware.

The software control is executed by SYSCR2<XEN>, SYSCR2<XTEN> and the P0 port function control register P0FC.

The hardware control is executed by reset release and the operation mode control circuit when the operation mode is switched to the STOP mode as described in "2.3.5. Operation mode control circuit".

Note: No hardware function is available for an external direct monitoring of the basic clock for the system clock. But the pulse signal with a constant frequency is generated by the software in disabling the interrupt and the WDT. It can be monitored. When the oscillation frequency of a system needs to be adjusted, the program for adjusting a frequency should be built in beforehand.

To prevent the dead lock of the CPU core due to the software-controlled enabling/disabling of the oscillation, an internal factor reset is generated depending on the combination of values of SYSCR2<SYSCK> which specifies the source clock of the main system clock, SYSCR2<XEN> and SYSCR2<XTEN>.

Table 2-1 Prohibited Combinations of Selecting Main Source Clock and Oscillation Enable Register

SYSCR2 <XEN>	SYSCR2 <XTEN>	SYSCR2 <SYSCK>	State
0	0	Don't care	All the oscillation circuits are stopped.
Don't care	0	1	The quarter of the low-frequency clock (fs/4) is selected as the source clock of the main system clock, but the low-frequency clock oscillation circuit is stopped.
0	Don't care	0	The gear clock (fcgck) is selected as the source clock of the main system clock, but the high-frequency clock oscillation circuit is stopped.

Note: It takes a certain period of time after SYSCR2<SYSCK> is changed before the main system clock is switched. When the currently operating oscillation circuit is stopped before the main system clock is switched, the internal condition becomes as shown in Table 2-1 and a system clock reset occurs. For details of clock switching, refer to "2.3.6. Operation Mode Control".

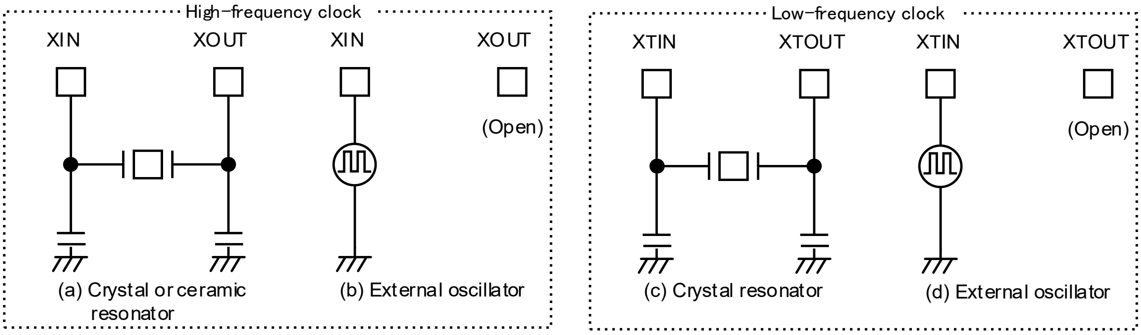


Figure 2-4 Examples of Oscillator Connection

2.3.3.2. Clock gear

The clock gear is a circuit that selects a gear clock (fcgck) obtained by dividing the high-frequency clock (fc) and inputs it to the timing generator.

Selects a divided clock at CGCR<FCGCKSEL>.

Two machine cycles are needed after CGCR<FCGCKSEL> is changed before the gear clock (fcgck) is changed.

The gear clock (fcgck) may be longer than the set clock width, immediately after CGCR<FCGCKSEL> is changed.

Immediately after reset release, the gear clock (fcgck) becomes the clock that is a quarter of the high-frequency clock (fc).

Table 2-2 Gear Clock (fcgck)

CGCR<FCGCKSEL>	fcgck
00	fc / 4
01	fc / 2
10	fc
11	Reserved

Note 1: fc: High-frequency clock [Hz]

Note 2: Don't change CGCR<FCGCKSEL> in the SLOW mode. The fcgck may not be changed as expected.

2.3.3.3. Timing generator

The timing generator is a circuit that generates system clocks to be supplied to the CPU core and the peripheral circuits, from the gear clock (fcgck) or the clock that is a quarter of the low-frequency clock (fs). The timing generator has the following functions:

1. Generation of the main system clock (fm)
2. Generation of clocks for the timer counter, the time base timer and other peripheral circuits

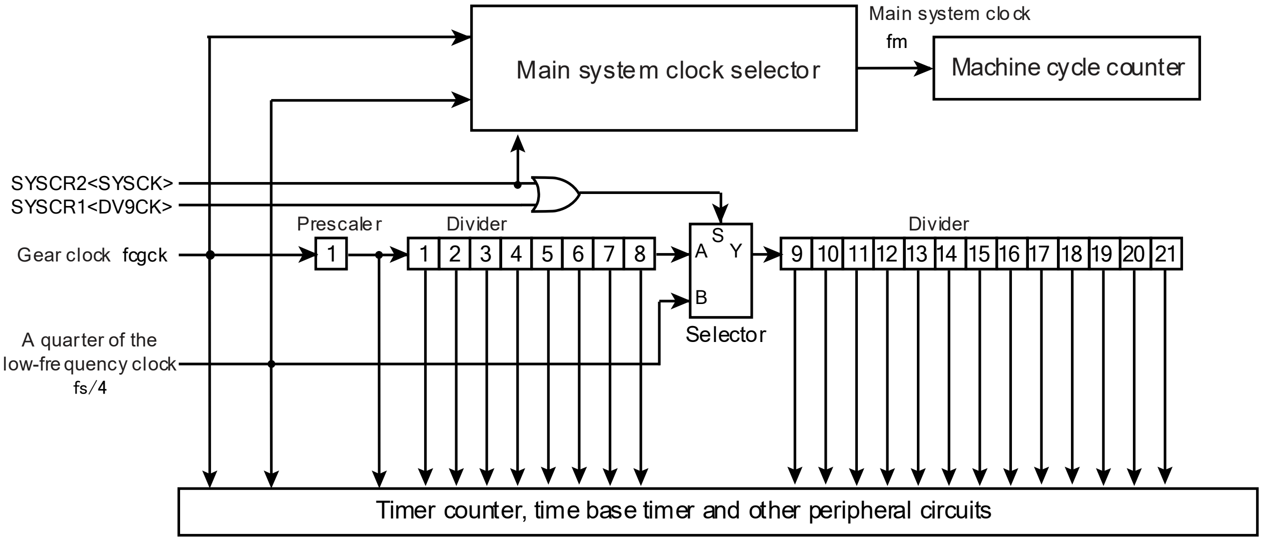


Figure 2-5 Configuration of Timing Generator

(1) Configuration of timing generator

The timing generator consists of a main system clock selector, a prescaler, a 21-stage divider and a machine cycle counter.

1. Main system clock selector

   This circuit selects the gear clock (fcgck) or the clock that is a quarter of the low-frequency clock (fs) for the main system clock (fm) to operate the CPU core.

   Clearing SYSCR2<SYSCK> to "0" selects the gear clock (fcgck). Setting it to "1" selects the clock that is a quarter of the low-frequency clock (fs).

   It takes a certain period of time after SYSCR2<SYSCK> is changed before the main system clock is switched. When the currently operating oscillation circuit is stopped before the main system clock is switched, the internal condition becomes as shown in Table 2-1 and a system clock reset occurs. For details of clock switching, refer to "2.3.6. Operation Mode Control".

2. Prescaler and divider

   These circuits divide fcgck. The divided clocks are supplied to the timer counter, the time base timer and other peripheral circuits.

   When both SYSCR1<DV9CK> and SYSCR2<SYSCK> are "0", the input clock to stage 9 of the divider becomes the output of stage 8 of the divider.

   When SYSCR1<DV9CK> or SYSCR2<SYSCK> is "1", the input clock to stage 9 of the divider becomes the clock that is a quarter of the low-frequency clock (fs). When SYSCR2<SYSCK> is "1", the outputs of stages 1 to 8 of the divider and prescaler are stopped.

   The prescaler and divider are cleared to "0" at a reset and at the end of the warm-up operation that follows the release of STOP mode.

3. Machine cycle

   Instruction execution is synchronized with the main system clock (fm).

   The minimum instruction execution unit is called a "machine cycle". One machine cycle corresponds to one main system clock.

   There are a total of 11 different types of instructions for the TLCS-870/C1 Series: 10 types ranging from 1-cycle instructions, which require one machine cycle for execution, to 10-cycle instructions, which require 10 machine cycles for execution, and 13-cycle instructions, which require 13 machine cycles for execution.

2.3.4. Warm-up counter

The warm-up counter is a circuit that counts the high-frequency clock (fc) and the low-frequency clock (fs), and it consists of a source clock selection circuit, a 3-stage frequency division circuit and a 14-stage counter.

The warm-up counter is used to secure the time after a power-on reset is released before the supply voltage becomes stable and secure the time after the STOP mode is released or the operation mode is changed before the oscillation by the oscillation circuit becomes stable.

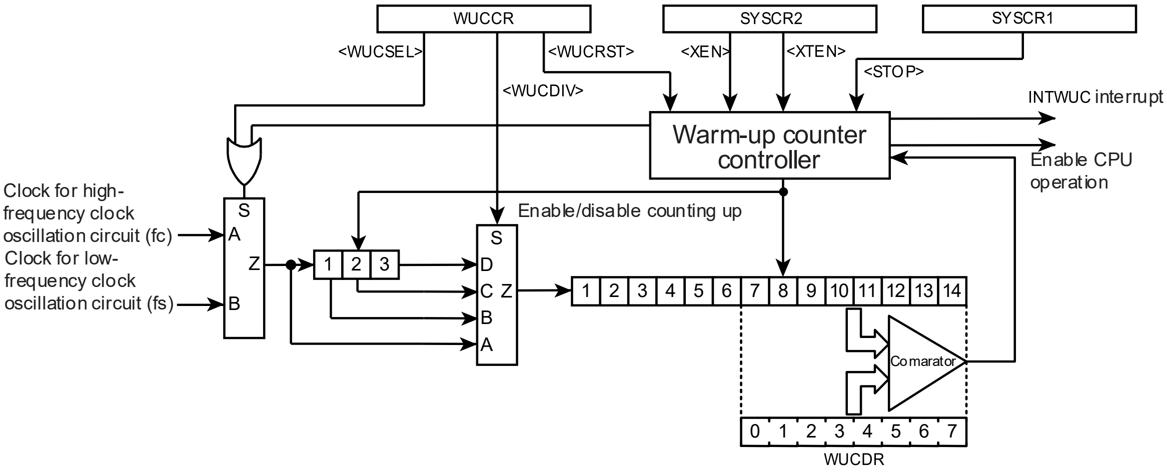


Figure 2-6 Warm-up Counter Circuit

2.3.4.1. Warm-up counter operation when the oscillation is enabled by the hardware

(1) When a power-on reset is released or a reset is released

The warm-up counter serves to secure the time after a power-on reset is released before the supply voltage becomes stable and the time after a reset is released before the oscillation by the high-frequency clock oscillation circuit becomes stable.

When the power is turned on and the supply voltage exceeds the power-on reset release voltage (V_{PROFF}), the warm-up counter reset signal is released. At this time, the CPU and the peripheral circuits are held in the reset state.

A reset signal initializes WUCCR<WUCSEL> to "0" and WUCCR<WUCDIV> to "11", which selects the high-frequency clock (fc) as the input clock to the warm-up counter.

When a reset is released for the warm-up counter, the high-frequency clock (fc) is input to the warm-up counter, and the 14-stage counter starts counting the high-frequency clock (fc).

When the upper 8-bit warm-up counter become equal to WUCDR, counting is stopped and a reset is released for the CPU and the peripheral circuits.

WUCDR is initialized to "0x66" after reset release, which makes the warm-up time "0x66" × 2^9/fc [s].

Note: The clock output from the oscillation circuit is used as the input clock to the warm-up counter. The warm-up time contains errors because the oscillation frequency is unstable until the oscillation circuit becomes stable.

(2) When the STOP mode is released

The warm-up counter serves to secure the time after the oscillation is enabled by the hardware before the oscillation becomes stable at the release of the STOP mode.

The high-frequency clock (fc) or the low-frequency clock (fs), which generates the main system clock when the STOP mode is activated, is selected as the input clock for frequency division circuit, regardless of WUCCR<WUCSEL>.

Before the STOP mode is activated, select the division rate of the input clock to the warm-up counter at WUCCR<WUCDIV> and set the warm-up time at WUCDR.

When the STOP mode is released, the 14-stage counter starts counting the input clock selected in the frequency division circuit.

When the upper 8-bit of warm-up counter become equal to WUCDR, counting is stopped and the operation is restarted by an instruction that follows the STOP mode activation instruction.

Clock that generates the main system clock when the STOP mode is activated	WUCCR <WUCSEL>	WUCCR <WUCDIV>	Counter input clock	Warm-up time
fc	Don’t care	00	fc	2^6 / fc to 255 x 2^6 / fc
		01	fc / 2	2^7 / fc to 255 x 2^7 / fc
		10	fc / 2^2	2^8 / fc to 255 x 2^8 / fc
		11	fc / 2^3	2^9 / fc to 255 x 2^9 / fc
fs	Don't care	00	fs	2^6 / fs to 255 x 2^6 / fs
		01	fs / 2	2^7 / fs to 255 x 2^7 / fs
		10	fs / 2^2	2^8 / fs to 255 x 2^8 / fs
		11	fs / 2^3	2^9 / fs to 255 x 2^9 / fs

Note 1: When the operation is switched to the STOP mode during the warm-up for the oscillation enabled by the software, the warm-up counter holds the value at the time, and restarts counting after the STOP mode is released. In this case, the warm-up time at the release of the STOP mode becomes insufficient. Don't switch the operation to the STOP mode during the warm-up for the oscillation enabled by the software.

Note 2: The clock output from the oscillation circuit is used as the input clock to the warm-up counter. The warm-up time contains errors because the oscillation frequency is unstable until the oscillation circuit becomes stable. Set the sufficient time for the oscillation start property of the oscillator.

2.3.4.2. Warm-up counter operation when the oscillation is enabled by the software

The warm-up counter serves to secure the time after the oscillation is enabled by the software before the oscillation becomes stable, at a mode change from NORMAL1 to NORMAL2 or from SLOW1 to SLOW2.

Select the input clock to the frequency division circuit at WUCCR<WUCSEL>.

Select the input clock to the 14-stage counter at WUCCR<WUCDIV>.

After the warm-up time is set at WUCDR, setting SYSCR2<XEN> or SYSCR2<XTEN> to "1" allows the stopped oscillation circuit to start oscillation and the 14-stage counter to start counting the selected input clock.

When the upper 8 bits of the counter become equal to WUCDR, an INTWUC interrupt occurs, counting is stopped and the counter is cleared.

Set WUCCR<WUCRST> to "1" to discontinue the warm-up operation.

By setting it to "1", the count-up operation is stopped, the warm-up counter is cleared, and WUCCR<WUCRST> is cleared to "0".

SYSCR2<XEN> and SYSCR2<XTEN> hold the values when WUCCR<WUCRST> is set to "1". To restart the warm-up operation, SYSCR2<XEN> or SYSCR2<XTEN> must be cleared to "0".

Note: The warm-up counter starts counting when SYSCR2<XEN> or SYSCR2<XTEN> is changed from "0" to "1". The counter will not start counting by writing "1" to SYSCR2<XEN> or SYSCR2<XTEN> when it is in the state of "1".

WUCCR <WUCSEL>	WUCCR <WUCDIV>	Counter input clock	Warm-up time
0	00	fc	2^6 / fc to 255 x 2^6 / fc
	01	fc / 2	2^7 / fc to 255 x 2^7 / fc
	10	fc / 2^2	2^8 / fc to 255 x 2^8 / fc
	11	fc / 2^3	2^9 / fc to 255 x 2^9 / fc
1	00	fs	2^6 / fs to 255 x 2^6 / fs
	01	fs / 2	2^7 / fs to 255 x 2^7 / fs
	10	fs / 2^2	2^8 / fs to 255 x 2^8 / fs
	11	fs / 2^3	2^9 / fs to 255 x 2^9 / fs

Note: The clock output from the oscillation circuit is used as the input clock to the warm-up counter. The warm-up time contains errors because the oscillation frequency is unstable until the oscillation circuit becomes stable. Set the sufficient time for the oscillation start property of the oscillator.

2.3.5. Operation mode control circuit

The operation mode control circuit starts and stops the oscillation circuits for the high-frequency and low-frequency clocks, and switches the main system clock (fm).

There are three operating modes: the single-clock mode, the dual-clock mode and the STOP mode. These modes are controlled by the system control registers (SYSCR1 and SYSCR2).

Figure 2-7 shows the operating mode transition diagram.

2.3.5.1. Single-clock mode

Only the gear clock (fcgck) is used for the operation in the single-clock mode.

The main system clock (fm) is generated from the gear clock (fcgck). Therefore, one machine cycle time is 1 / fcgck [s].

The gear clock (fcgck) is generated from the high-frequency clock (fc).

In the single-clock mode, the low-frequency clock generation circuit pins P02 (XTIN) and P03 (XTOUT) can be used as the I/O ports.

(1) NORMAL1 mode

In this mode, the CPU core and the peripheral circuits operate using the gear clock (fcgck).

The NORMAL1 mode becomes active after reset release.

(2) IDLE1 mode

In this mode, the CPU and the watchdog timer stop and the peripheral circuits operate using the gear clock (fcgck).

The IDLE1 mode is activated by setting SYSCR2<IDLE> to "1" in the NORMAL1 mode.

When the IDLE1 mode is activated, the CPU and the watchdog timer stop.

When the interrupt latch enabled by the interrupt enable register EIR becomes "1", the IDLE1 mode is released to the NORMAL1 mode.

When the <IMF> (interrupt master enable flag) is "1" (interrupts enabled), the operation returns normal after the interrupt processing is completed.

When the <IMF> is "0" (interrupts disabled), the operation is restarted by the instruction that follows the IDLE1 mode activation instruction.

(3) IDLE0 mode

In this mode, the CPU and the peripheral circuits stop, except the oscillation circuits and the time base timer.

In the IDLE0 mode, the peripheral circuits stop in the states when the IDLE0 mode is activated or become the same as the states when a reset is released. For operations of the peripheral circuits in the IDLE0 mode, refer to the section of each peripheral circuit.

The IDLE0 mode is activated by setting SYSCR2<TGHALT> to "1" in the NORMAL1 mode. When the IDLE0 mode is activated, the CPU stops and the timing generator stops the clock supply to the peripheral circuits except the time base timer.

When the falling edge of the source clock selected at TBTCR<TBTCK> is detected, the IDLE0 mode is released, the timing generator starts the clock supply to all the peripheral circuits and the NORMAL1 mode is restored.

Note that the IDLE0 mode is activated and restarted, regardless of the setting of TBTCR<TBTEN>.

When the IDLE0 mode is activated with TBTCR<TBTEN> set at "1", the INTTBT interrupt latch is set after the NORMAL mode is restored.

When the <IMF> is "1" and the <EF5> (the individual interrupt enable flag for the time base timer) is "1", the operation returns normal after the interrupt processing is completed.

When the <IMF> is "0" or when the <IMF> is "1" and the <EF5> (the individual interrupt enable flag for the time base timer) is "0", the operation is restarted by the instruction that follows the IDLE0 mode activation instruction.

2.3.5.2. Dual-clock mode

The gear clock (fcgck) and the low-frequency clock (fs) are used for the operation in the dual-clock mode.

The main system clock (fm) is generated from the gear clock (fcgck) in the NORMAL2 or IDLE2 mode. And it is generated from the clock that is a quarter of the low-frequency clock (fs) in the SLOW1/2 or SLEEP0/1 mode. Therefore, one machine cycle time is 1/fcgck [s] in the NORMAL2 or IDLE2 mode and is 4/fs [s] in the SLOW1/2 or SLEEP0/1 mode.

In the dual-clock mode, P02 (XTIN) and P03 (XTOUT) are used as the low-frequency clock oscillation circuit pins. Therefore, these pins cannot be used as I/O ports.

The operation of the TLCS-870/C1 Series becomes the single-clock mode after reset release. To operate it in the dual-clock mode, the program starts to oscillate the low-frequency clock oscillation circuit.

(1) NORMAL2 mode

In this mode, the CPU core operates using the gear clock (fcgck), and the peripheral circuits operate using the gear clock (fcgck) or the clock that is a quarter of the low-frequency clock (fs).

(2) SLOW2 mode

In this mode, the CPU core and the peripheral circuits operate using the clock that is a quarter of the low-frequency clock (fs).

In the SLOW mode, some peripheral circuits become the same as the states when a reset is released. For operations of the peripheral circuits in the SLOW mode, refer to the section of each peripheral circuit.

Set SYSCR2<SYSCK> to switch the operation mode from NORMAL2 to SLOW2 or from SLOW2 to NORMAL2.

In the SLOW2 mode, outputs of the prescaler and stages 1 to 8 of the divider stop.

(3) SLOW1 mode

In this mode, the high-frequency clock oscillation circuit stops operation and the CPU core and the peripheral circuits operate using the clock that is a quarter of the low-frequency clock (fs).

This mode requires less power to operate the high-frequency clock oscillation circuit than in the SLOW2 mode.

In the SLOW mode, some peripheral circuits become the same as the states when a reset is released. For operations of the peripheral circuits in the SLOW mode, refer to the section of each peripheral circuit.

Set SYSCR2<XEN> to switch the operation between the SLOW1 and SLOW2 modes.

In the SLOW1 mode, outputs of the prescaler and stages 1 to 8 of the divider stop.

(4) IDLE2 mode

In this mode, the CPU and the watchdog timer stop and the peripheral circuits operate using the gear clock (fcgck) or the clock that is a quarter of the low-frequency clock (fs).

The IDLE2 mode can be activated and released in the same way as for the IDLE1 mode. The operation returns to the NORMAL2 mode after this mode is released.

(5) SLEEP1 mode

In this mode, the high-frequency clock oscillation circuit stops operation, the CPU and the watchdog timer stop, and the peripheral circuits operate using the clock that is a quarter of the low-frequency clock (fs).

In the SLEEP1 mode, some peripheral circuits become the same as the states when a reset is released. For operations of the peripheral circuits in the SLEEP1 mode, refer to the section of each peripheral circuit.

The SLEEP1 mode can be activated and released in the same way as for the IDLE1 mode. The operation returns to the SLOW1 mode after this mode is released.

In the SLEEP1 mode, outputs of the prescaler and stages 1 to 8 of the divider stop.

(6) SLEEP0 mode

In this mode, the high-frequency clock oscillation circuit stops operation, the time base timer operates using the clock that is a quarter of the low-frequency clock (fs), and the core and the peripheral circuits stop.

In the SLEEP0 mode, the peripheral circuits stop in the states when the SLEEP0 mode is activated or become the same as the states when a reset is released. For operations of the peripheral circuits in the SLEEP0 mode, refer to the section of each peripheral circuit.

The SLEEP0 mode can be activated and released in the same way as for the IDLE0 mode. The operation returns to the SLOW1 mode after this mode is released.

In the SLEEP0 mode, the CPU stops and the timing generator stops the clock supply to the peripheral circuits except the time base timer.

2.3.5.3. STOP mode

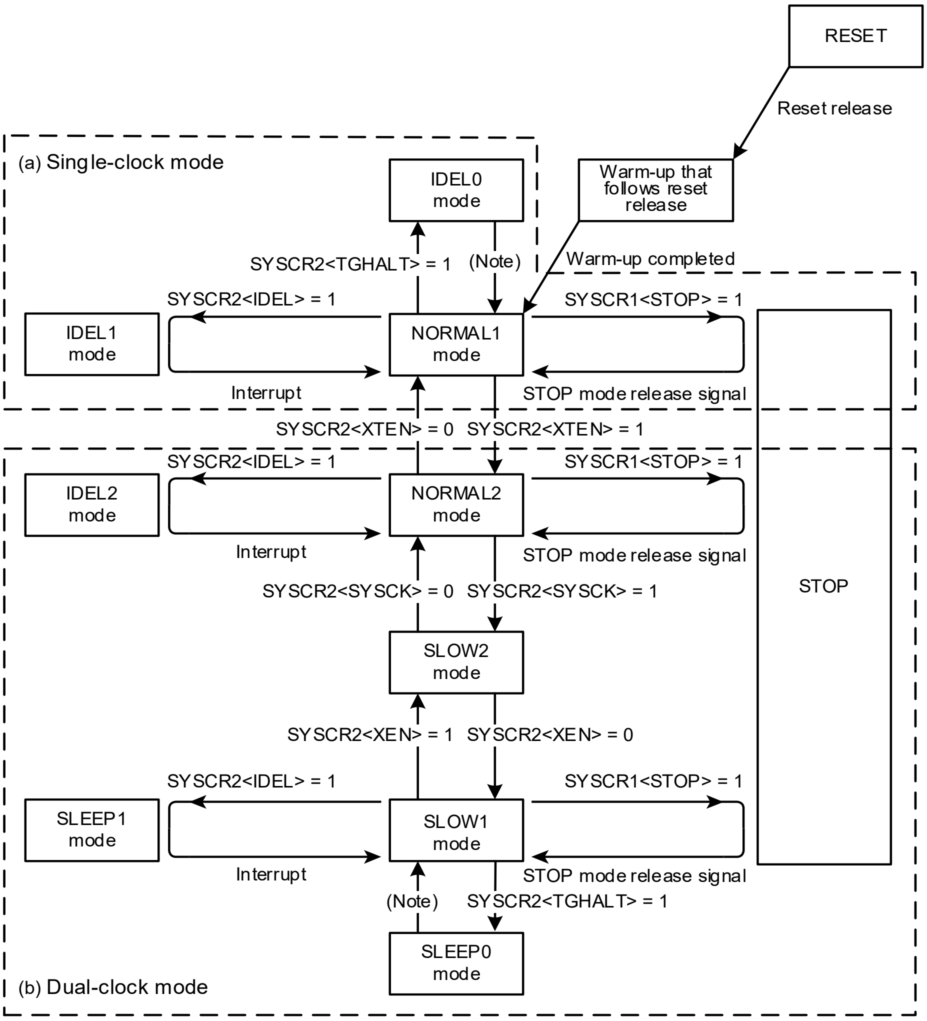
In this mode, all the operations in the TMP89FS60B/62B/63B, including the oscillation circuits, are stopped and the internal states in effect before the TMP89FS60B/62B/63B was stopped are held with low power consumption.

In the STOP mode, the peripheral circuits stop in the states when the STOP mode is activated or become the same as the states when a reset is released. For operations of the peripheral circuits in the STOP mode, refer to the section of each peripheral circuit.

The STOP mode is activated by setting SYSCR1<STOP> to "1".

The STOP mode is released by the STOP mode release signals. After the warm-up time has elapsed, the operation returns to the mode that was active before the STOP mode, and the operation is restarted by the instruction that follows the STOP mode activation instruction.

2.3.5.4. Transition of each operation mode



Note: The mode is released by the falling edge of the source clock selected at TBTCR<TBTCK>.

Figure 2-7 Operation Mode Transition Diagram

Table 2-3 Operation Modes and Conditions

Operation mode		Oscillation circuit		CPU core	Watchdog timer	Time base timer	AD converter	Other peripheral circuits	One Machine cycle time
		For high frequency	For low frequency						
Single clock	RESET	Oscillation	Stop	Reset	Reset	Reset	Reset	Reset	1 / fcgck [s]
	NORMAL1			Operation	Operation	Operation	Operation	Operation	
	IDLE1			Stop	Stop				
	IDLE0						Stop	Stop	
	STOP	Stop				Stop			-
Dual clock	NORMAL2	Oscillation	Oscillation	Operation with the high-frequency clock	Operation with the high- or low-frequency clock	Operation	Operation	Operation	1 / fcgck [s]
	IDLE2			Stop	Stop				
	SLOW2			Operation with the low-frequency clock	Operation with the low-frequency clock		Stop		4 / fs [s]
	SLOW1	Stop		Operation with the low-frequency clock	Operation with the low-frequency clock				
	SLEEP1			Stop	Stop				
	SLEEP0							Stop	
	STOP		Stop			Stop			-

2.3.6. Operation Mode Control

2.3.6.1. STOP mode

The STOP mode is controlled by system control register 1 (SYSCR1) and the STOP mode release signals.

(1) Start the STOP mode

The STOP mode is started by setting SYSCR1<STOP> to "1". In the STOP mode, the following states are maintained:

1. Both the high-frequency and low-frequency clock oscillation circuits stop oscillation and all internal operations are stopped.
2. The RAM, the registers and the program status word are all held in the states in effect before STOP mode was started.
   The port output latch is determined by the value of SYSCR1<OUTEN>.
3. The prescaler and the divider of the timing generator are cleared to "0".
4. The program counter holds the address of the instruction 2 ahead of the instruction (e.g., SET (SYSCR1).7) which started the STOP mode.

(2) Release the STOP mode

The STOP mode is released by the following STOP mode release signals. It is also released by a reset by the $\overline{\text{RESET}}$ pin, a power-on reset and a reset by the voltage detection circuits. When a reset is released, the warm-up starts. After the warm-up is completed, the operation mode becomes the NORMAL1 mode.

1. Release by the $\overline{\text{STOP}}$ pin
2. Release by the key-on wakeup
3. Release by the voltage detection circuit (Release by the voltage detection reset)

Note: During the STOP period (from the start of the STOP mode to the end of the warm-up), due to changes in the level of the external interrupt pin, interrupt latches may be set to "1" and interrupts may be accepted immediately after the STOP mode is released. Before starting the STOP mode, therefore, disable interrupts. Also, before enabling interrupts after STOP mode is released, clear unnecessary interrupt latches.

Regarding to release STOP mode by each STOP mode releasing signal, the details are explained belows.

1. The STOP mode is release by the $\overline{\text{STOP}}$ pin

   The STOP mode is released by using the $\overline{\text{STOP}}$ pin.

   The STOP mode release by the $\overline{\text{STOP}}$ pin includes the level-sensitive release mode and the edge-sensitive release mode, either of which can be selected at SYSCR1<RELM>.

   The $\overline{\text{STOP}}$ pin is also used as the P11 port and the $\overline{\text{INT5}}$ (external interrupt input 5) pin.

   - Level-sensitive release mode

   The STOP mode is released by setting the $\overline{\text{STOP}}$ pin “High” level. Setting SYSCR1<RELM> to "1" selects the level-sensitive release mode. This mode is used for the capacitor backup when the main power supply is cut off and the long term battery backup.

   When an instruction for starting the STOP mode is executed while the $\overline{\text{STOP}}$ pin input is “High” level , the STOP mode does not start. Thus, to start the STOP mode in the level-sensitive release mode, it is necessary for the program to first confirm that the $\overline{\text{STOP}}$ pin input is “Low” level.

   This can be confirmed by testing the port by the software or using interrupts

Note: When the STOP mode is released, the warm-up counter source clock automatically changes to the clock that generated the main system clock when the STOP mode was started, regardless of WUCCR<WUCSEL>.

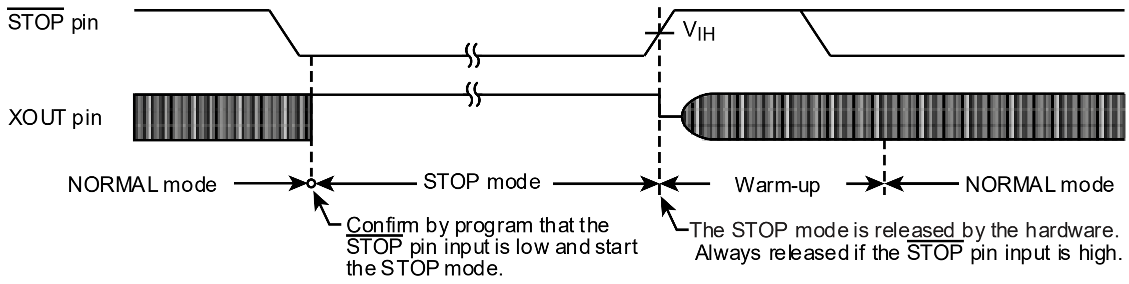
Example: Starting the STOP mode from NORMAL mode after testing P11 ($\overline{\text{STOP}}$) pin.
(Warm-up time at release of the STOP mode is about 300 [μs] at fc = 10 [MHz].)

```
        LD      (SYSCR1), 0x40      ; Set up the level-sensitive release mode
SSTOPH: TEST    (P1PRD).1           ; Wait until STOP pin becomes “Low” level.
        J       F, SSTOPH
        LD      (WUCCR), 0x01       ; WUCCR<WUCDIV> ← "00" (No division) (Note)
        LD      (WUCDR), 0x2F       ; Set the warm-up time
                                    ; 300 [μs] / 6.4 [μs] = 46.9 → round up to "0x2F"
        DI                          ; <IMF> ← "0"
        SET     (SYSCR1).7          ; Start the STOP mode with the level-sensitive release mode selected
```

Example: Starting the STOP mode from the SLOW mode with an INT5 interrupt
(Warm-up time at release of the STOP mode is about 450 [ms] at fs = 32.768 [kHz].)

```
PINT5:  TEST    (P1PRD).1           ; To reject noise, the STOP mode does not start
        J       F, SINT5            ; until the STOP pin input becomes “Low” level
        LD      (SYSCR1), 0x40      ; Set up the level-sensitive release mode
        LD      (WUCCR), 0x03       ; WUCCR<WUCDIV> ← "00" (No division) (Note)
        LD      (WUCDR), 0xE8       ; Set the warm-up time
                                    ; 450 [ms] / 1.953 [ms] = 230.4 → round up to "0xE8"
        DI                          ; <IMF> ← "0"
        SET     (SYSCR1).7          ; Start the STOP mode with the level-sensitive release mode selected
SINT5:  RETI
                :
VINT5:  DW      PINT5               ; INT5 vector table
```

Note: When the STOP mode is released, the warm-up counter source clock automatically changes to the clock that generated the main system clock when the STOP mode was started, regardless of WUCCR<WUCSEL>.


STOP pin
V_IH
XOUT pin
NORMAL mode
STOP mode
Warm-up
NORMAL mode
Confirm by program that the STOP pin input is low and start the STOP mode.
The STOP mode is released by the hardware. Always released if the STOP pin input is high.

Note: When the $\overline{\text{STOP}}$ pin input returns to “Low” level after the warm-up starts, the STOP mode is not restarted.

Figure 2-8 Level-sensitive Release Mode
(Example when the high-frequency clock oscillation circuit is selected)

- Edge-sensitive release mode

  In this mode, the STOP mode is released at the rising edge of the $\overline{\text{STOP}}$ pin input.

  Setting SYSCR1<RELM> to "0" selects the edge-sensitive release mode.

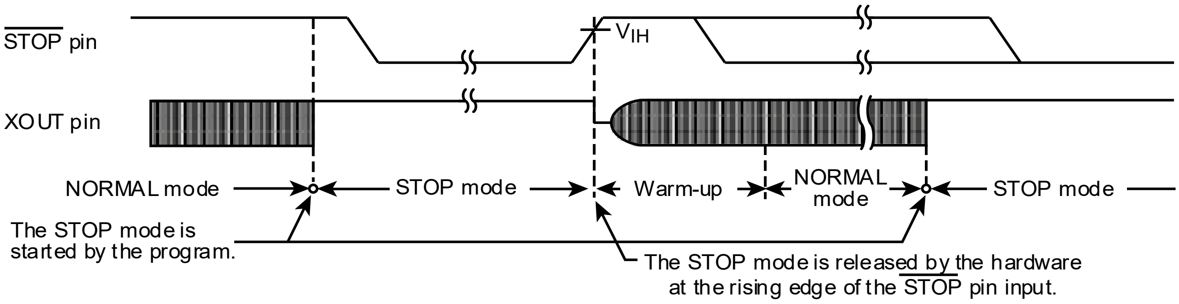
  This mode is used in applications where a relatively short program is executed repeatedly at periodic intervals. This periodic signal (for example, a clock from a low-power consumption oscillator) is applied to the $\overline{\text{STOP}}$ pin.

  In the edge-sensitive release mode, the STOP mode is started even when the $\overline{\text{STOP}}$ pin input is "High" level.

Example: Starting the STOP mode from the NORMAL mode
(The warm-up time at release of the STOP mode is about 200 [μs] at fc = 10 [MHz].)

```
LD      (WUCCR), 0x01       ; WUCCR<WUCDIV> ← "00" (No division) (Note)
LD      (WUCDR), 0x20       ; Set the warm-up time
                            ; 200 [μs] / 6.4 [μs] = 31.25 → round up to "0x20"
DI                          ; <IMF> ← "0"
LD      (SYSCR1), 0x80      ; Start the STOP mode with the edge-sensitive release mode selected
```

Note: When the STOP mode is released, the warm-up counter source clock automatically changes to the clock that generated the main system clock when the STOP mode was started, regardless of WUCCR<WUCSEL>.


STOP pin
VIH
XOUT pin
NORMAL mode
STOP mode
Warm-up
NORMAL mode
STOP mode
The STOP mode is started by the program.
The STOP mode is released by the hardware at the rising edge of the STOP pin input.

Note: When the rising edge is input to the $\overline{\text{STOP}}$ pin within 1 machine cycle after SYSCR1<STOP> is set to "1", the STOP mode will not be released.

Figure 2-9 Edge-sensitive Release Mode
(Example when the high-frequency clock oscillation circuit is selected)

2. Release by the key-on wakeup

   The STOP mode is released by applying the prescribed level to the key-on wakeup pin. The level to release the STOP mode can be selected from “High” level or "Low" level.

   For releasing by the key-on wakeup, refer to section "19 Key-on Wakeup (KWU)".

Note: When the key-on wakeup pin input becomes the opposite level to the release level after the warm-up starts, the STOP mode is not restarted.

3. Release by the voltage detection circuits

   The STOP mode is released by the supply voltage detection by the voltage detection circuits.

   When the voltage detection operation mode of the voltage detection circuits is set to "Generates a voltage detection reset signal", a reset is applied as soon as the supply voltage becomes lower than the detection voltage and the STOP mode is released immediately.

   For details, refer to "7. Voltage Detection Circuit".

Note: When the supply voltage becomes equal to or higher than the detection voltage within 1 machine cycle after SYSCR1<STOP> is set to "1", the STOP mode will not be released.

(3) STOP mode release operation

The STOP mode is released in the following sequence:

1. Oscillation starts. For the oscillation start operation in each mode, refer to "Table 2-4 Oscillation Start Operation at Release of the STOP Mode".
2. Warm-up is executed to secure the time required to stabilize oscillation. The internal operations remain stopped during warm-up. The warm-up time is set by the warm-up counter, depending on the oscillator characteristics.
3. After the warm-up time has elapsed, the normal operation is restarted by the instruction that follows the STOP mode start instruction. At this time, the prescaler and the divider of the timing generator are cleared to "0".

Note 1: After releasing STOP mode, it takes 120 [μs] (maximum) until the voltage in the device is stable and the oscillation is started.

Note 2: When the STOP mode is released with a low hold voltage, the following cautions must be observed. The supply voltage must be at the operating voltage level before releasing the STOP mode. The $\overline{\text{RESET}}$ pin input must also be “High” level, rising together with the supply voltage. In this case, when an external time constant circuit has been connected, the $\overline{\text{RESET}}$ pin input voltage will increase at a slower pace than the power supply voltage. At this time, there is a danger that a reset may occur when the input voltage level of the $\overline{\text{RESET}}$ pin drops below the non-inverting high-level input voltage (Hysteresis input).

Table 2-4 Oscillation Start Operation at Release of the STOP Mode

Operation mode before the STOP mode is started		High-frequency clock oscillation circuit	Low-frequency clock oscillation circuit	Oscillation start operation after release
Single-clock mode	NORMAL1	High-frequency clock oscillation circuit	-	The high-frequency clock oscillation circuit starts oscillation. The low-frequency clock oscillation circuit stops oscillation.
Dual-clock mode	NORMAL2	High-frequency clock oscillation circuit	Low-frequency clock oscillation circuit	The high-frequency clock oscillation circuit starts oscillation. The low-frequency clock oscillation circuit starts oscillation.
	SLOW1	-	Low-frequency clock oscillation circuit	The high-frequency clock oscillation circuit stops oscillation. The low-frequency clock oscillation circuit starts oscillation.

Note: When the operation returns to the NORMAL2 mode, fc is input to the frequency division circuit of the warm-up counter.

2.3.6.2. IDLE1/2 and SLEEP1 modes

The IDLE1/2 and SLEEP1 modes are controlled by the system control register 2 (SYSCR2) and maskable interrupts.

The TMP89FS60B/62B/63B is in the following states during these modes.

(1) The CPU and the watchdog timer stop their operations. The peripheral circuits continue to operate.

(2) The RAM, the registers, the program status word and the port output latches are all held in the status in effect before IDLE1/2 or SLEEP1 mode was started.

(3) The program counter holds the address of the instruction 2 ahead of the instruction which starts the IDLE1/2 or SLEEP1 mode.

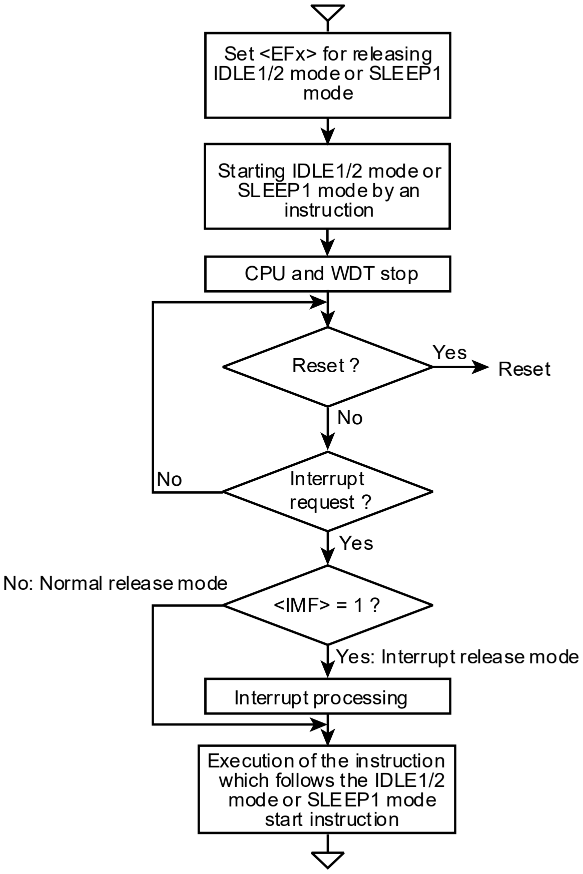

Set <EFx> for releasing IDLE1/2 mode or SLEEP1 mode
Starting IDLE1/2 mode or SLEEP1 mode by an instruction
CPU and WDT stop
Reset ?
Yes
Reset
No
No
Interrupt request ?
Yes
No: Normal release mode
<IMF> = 1 ?
Yes: Interrupt release mode
Interrupt processing
Execution of the instruction which follows the IDLE1/2 mode or SLEEP1 mode start instruction

Figure 2-10 IDLE1/2 and SLEEP1 Modes

1. Start the IDLE1/2 and SLEEP1 modes

   After the interrupt master enable flag (<IMF>) is set to "0", set the individual interrupt enable flag (<EFx>) to "1", which releases IDLE1/2 and SLEEP1 modes.

   To start the IDLE1/2 or SLEEP1 mode, set SYSCR2<IDLE> to "1".

   When the release condition is satisfied when it is attempted to start the IDLE1/2 or SLEEP1 mode, SYSCR2<IDLE> remains cleared and the IDLE1/2 or SLEEP1 mode will not be started.

Note 1: When a watchdog timer interrupt is generated immediately before the IDLE1/2 or SLEEP1 mode is started, the watchdog timer interrupt will be processed but the IDLE1/2 or SLEEP1 mode will not be started.

Note 2: Before starting the IDLE1/2 or SLEEP1 mode, enable the interrupt requests to be generated to release the IDLE1/2 or SLEEP1 mode and set the individual interrupt enable flag.

2. Release the IDLE1/2 and SLEEP1 modes

   The IDLE1/2 and SLEEP1 modes include a normal release mode and an interrupt release mode. These modes are selected at the interrupt master enable flag (<IMF>). After releasing IDLE1/2 or SLEEP1 mode, SYSCR2<IDLE> is automatically cleared to "0" and the operation mode is returned to the mode preceding the IDLE1/2 or SLEEP1 mode.

   The IDLE1/2 and SLEEP1 modes are also released by a reset by the $\overline{\text{RESET}}$ pin, a power-on reset and a reset by the voltage detection circuits. After releasing the reset, the warm-up starts. After the warm-up is completed, the NORMAL1 mode becomes active.

   - Normal release mode (In <IMF> = 0)

   The IDLE1/2 or SLEEP1 mode is released when the interrupt latch enabled by the individual interrupt enable flag (<EFx>) is "1". The operation is restarted by the instruction that follows the IDLE1/2 or SLEEP1 mode start instruction. Normally, the interrupt latch (<ILx>) of the interrupt source used for releasing must be cleared to "0" by load instructions.

   - Interrupt release mode (In <IMF> = 1)

   The IDLE1/2 or SLEEP1 mode is released when the interrupt latch enabled by the individual interrupt enable flag (<EFx>) is "1" and the interrupt procedure is starting. After the interrupt procedure is completed, the operation is restarted by the instruction that follows the IDLE1/2 or SLEEP1 mode start instruction.

2.3.6.3. IDLE0 and SLEEP0 modes

The IDLE0 and SLEEP0 modes are controlled by the system control register 2 (SYSCR2) and the time base timer control register (TBTCR).

The TMP89FS60B/62B/63B is in the following states during these modes.

(1) The timing generator stops the clock supply to the peripheral circuits except the time base timer.

(2) The RAM, the registers, the program status word and the port output latches are all held in the states in effect before the IDLE0 or SLEEP0 mode was started.

(3) The program counter holds the address of the instruction 2 ahead of the instruction which starts the IDLE0 or SLEEP0 mode.

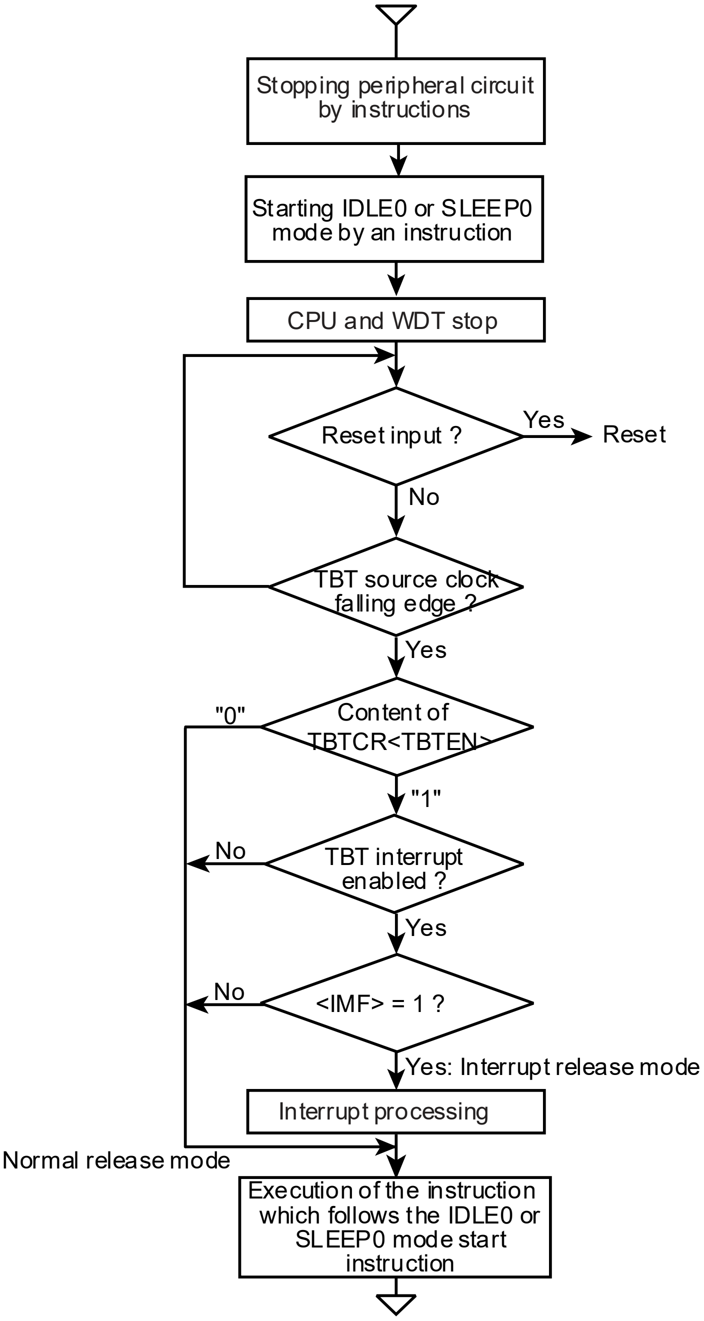

Stopping peripheral circuit by instructions
Starting IDLE0 or SLEEP0 mode by an instruction
CPU and WDT stop
Reset input ?
Yes
Reset
No
TBT source clock falling edge ?
Yes
"0"
Content of TBTCR<TBTEN>
"1"
No
TBT interrupt enabled ?
Yes
No
<IMF> = 1 ?
Yes: Interrupt release mode
Interrupt processing
Normal release mode
Execution of the instruction which follows the IDLE0 or SLEEP0 mode start instruction

Figure 2-11 IDLE0 and SLEEP0 Modes

1. Start the IDLE0 and SLEEP0 modes

   Stops the peripheral circuit such as a timer counter.

   To start the IDLE0 or SLEEP0 mode, set SYSCR2<TGHALT> to "1".

2. Release the IDLE0 and SLEEP0 modes

   The IDLE0 and SLEEP0 modes include a normal release mode and an interrupt release mode. These modes are selected at the interrupt master enable flag (<IMF>), the individual interrupt enable flag (<EF5>) for the time base timer and TBTCR<TBTEN>. After releasing the IDLE0 or SLEEP0 mode, SYSCR2<TGHALT> is automatically cleared to "0" and the operation mode is returned to the mode preceding the IDLE0 or SLEEP0 mode. When TBTCR<TBTEN> has been set at "1", the INTTBT interrupt latch is set.

   The IDLE0 and SLEEP0 modes are also released by a reset by the $\overline{\text{RESET}}$ pin, a power-on reset and a reset by the voltage detection circuits. When a reset is released, the warm-up starts. After the warm-up is completed, the NORMAL1 mode becomes active.

- Normal release mode (In <IMF> AND <EF5> AND TBTCR<TBTEN> = 0)

  The IDLE0 or SLEEP0 mode is released when the falling edge of the source clock selected at TBTCR<TBTCK>is detected. After the IDLE0 or SLEEP0 mode is released, the operation is restarted by the instruction that follows the IDLE0 or SLEEP0 mode start instruction.

  When TBTCR<TBTEN> is "1", the time base timer interrupt latch is set.

- Interrupt release mode (In <IMF> AND <EF5> AND TBTCR<TBTEN> = 1)

  The IDLE0 or SLEEP0 mode is released when the falling edge of the source clock selected at TBTCR<TBTCK> is detected. After the release, the INTTBT interrupt processing is started.

Note 1: The IDLE0 or SLEEP0 mode is released to the NORMAL1 or SLOW1 mode by the asynchronous internal clock selected at TBTCR<TBTCK>. Therefore, the period from the start to the release of the IDLE0 or SLEEP0 mode may be shorter than the time specified at TBTCR<TBTCK>.

Note 2: When a watchdog timer interrupt is generated immediately before the IDLE0 or SLEEP0 mode is started, the watchdog timer interrupt will be processed but the IDLE0 or SLEEP0 mode will not be started.

2.3.6.4. SLOW mode

The SLOW mode is controlled by system control register 2 (SYSCR2).

(1) Switching from the NORMAL2 mode to the SLOW1 mode

Set SYSCR2<SYSCK> to "1".

When a maximum of 2/fcgck + 10/fs [s] has elapsed since SYSCR2<SYSCK> is set to "1", the main system clock (fm) is switched to fs/4.

After switching, wait for 2 machine cycles or longer, and then clear SYSCR2<XEN> to "0" to turn off the high-frequency clock oscillation circuit.

The time which is stable oscillation of the low-frequency clock (fs) is elapsed by the warm-up counter before implementing the procedure described above.

Note 1: Be sure to follow this procedure to switch the operation from the NORMAL2 mode to the SLOW1 mode.

Note 2: It is also possible to allow the basic clock for the high-frequency clock to oscillate continuously to return to NORMAL2 mode. However, be sure to turn off the oscillation of the basic clock for the high-frequency clock when the STOP mode is started from the SLOW mode.

Note 3: After switching SYSCR2<SYSCK>, be sure to wait for 2 machine cycles or longer before clearing SYSCR2<XEN> to "0". Clearing it within 2 machine cycles causes a system clock reset.

Note 4: When the main system clock (fm) is switched, the gear clock (fcgck) is synchronized with the clock that is a quarter of the low-frequency clock (fs). For the synchronization, fm is stopped for a period of 10 / fs or shorter.

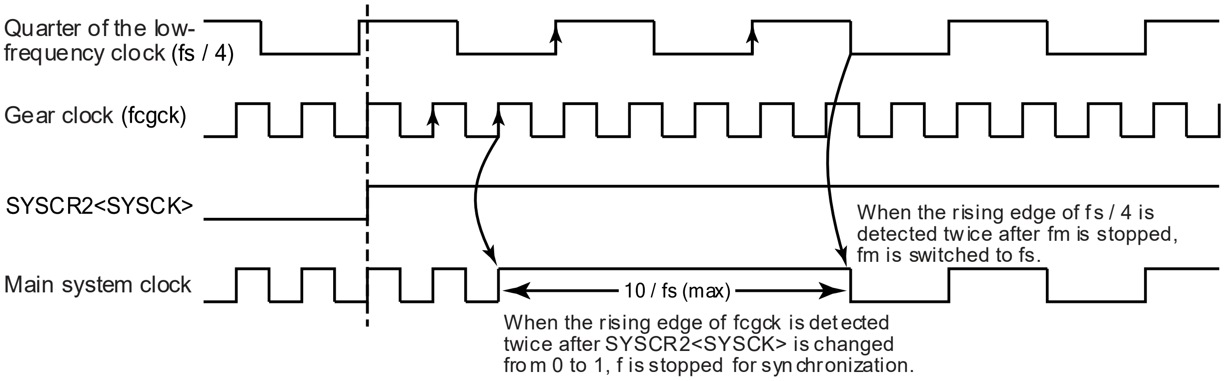

Quarter of the low-frequency clock (fs / 4)
Gear clock (fcgck)
SYSCR2<SYSCK>
Main system clock
When the rising edge of fs / 4 is detected twice after fm is stopped, fm is switched to fs.
10 / fs (max)
When the rising edge of fcgck is detected twice after SYSCR2<SYSCK> is changed from 0 to 1, f is stopped for synchronization.

Figure 2-12 Switching of the Main System Clock (fm) (Switching from fcgck to fs / 4)

Example 1: Switching from the NORMAL2 mode to the SLOW1 mode

```
                SET     (SYSCR2).4      ; SYSCR2<SYSCK> ← "1"
                                        ; (Switches the main system clock to the quarter of the
                                        ; low-frequency clock for the SLOW2 mode)
                NOP                     ; Waits for 2 machine cycles
                NOP
                CLR     (SYSCR2).6      ; SYSCR2<XEN> ← "0"
                                        ; (Turns off the high-frequency clock oscillation circuit for SLOW1 mode)
```

Example 2: When the TMP89FS60B/62B/63B operate by high-frequency clock, switching to the SLOW1 mode after the stable oscillation of the low-frequency clock oscillation circuit is confirmed at the warm-up counter (The warm-up time is about 100 [ms] at fs = 32.768 [kHz].)

```
                ;#### Initialize routine ####
                SET     (P0FC).2          ; <P0FC2> ← "1" (Uses P02/03 as oscillators)
                        :
                LD      (WUCCR), 0x02     ; WUCCR<WUCDIV> ← "00" (No division),
                                          ; WUCCR<WUCSEL> ← "1" (Selects fs as the source clock)
                LD      (WUCDR), 0x33     ; Sets the warm-up time
                                          ; (Determines the time depending on the oscillator characteristics)
                                          ; 100 [ms] / 1.95 [ms] = 51.2 → round up to "0x33"
                SET     (EIRL).4          ; Enables INTWUC interrupts
                SET     (SYSCR2).5        ; SYSCR2<XTEN> ← "1"
                                          ; (Starts the low-frequency clock oscillation circuit and starts the
                                          ; warm-up counter)
                        :
                ;#### Interrupt service routine of warm-up counter interrupts ####
PINTWUC:        SET     (SYSCR2).4        ; SYSCR2<SYSCK> ← "1"
                                          ; (Switches the main system clock to the low-frequency clock)
                NOP                       ; Waits for 2 machine cycles
                NOP
                CLR     (SYSCR2).6        ; SYSCR2<XEN> ← "0"
                                          ; (Turns off the high-frequency clock oscillation circuit)
                RETI
                        :
VINTWUC:        DW      PINTWUC           ; INTWUC vector table
```

(2) Switching from the SLOW1 mode to the NORMAL1 mode

Set SYSCR2<XEN> to "1" to enable the high-frequency clock (fc) to oscillate. After Securing the time to stabilize the oscillation of the high-frequency clock by the warm-up counter, clear SYSCR2<SYSCK> to "0".

When a maximum of 8/fs + 2.5/fcgck [s] has elapsed since SYSCR2<SYSCK> is cleared to "0", the main system clock (fm) is switched to fcgck.

After switching, wait for 2 machine cycles or longer, and then clear SYSCR2<XTEN> to "0" to turn off the low-frequency clock oscillator.

The SLOW mode is also released by a reset by the $\overline{\text{RESET}}$ pin, a power-on reset and a reset by the voltage detection circuits. When a reset is released, the warm-up starts. After the warm-up is completed, the NORMAL1 mode becomes active.

Note 1: Be sure to follow this procedure to switch the operation from the SLOW1 mode to the NORMAL1 mode.

Note 2: After switching SYSCR2<SYSCK>, be sure to wait for 2 machine cycles or longer before clearing SYSCR2<XTEN> to "0". Clearing it within 2 machine cycles causes a system clock reset.

Note 3: When the main system clock (fm) is switched, the gear clock (fcgck) is synchronized with the clock that is a quarter of the low-frequency clock. For the synchronization, fm is stopped for a period of 2.5 / fcgck [s] or shorter.

Note 4: When SYSCR2<XEN> is set at "1", writing "1" to SYSCR2<XEN> does not cause the warm-up counter to start counting the source clock.

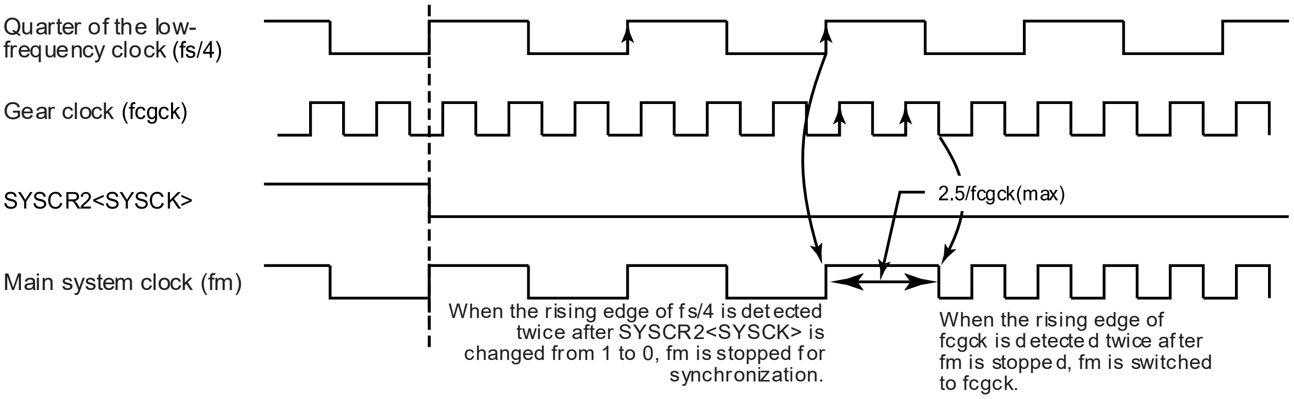

Quarter of the low-frequency clock (fs/4)
Gear clock (fcgck)
SYSCR2<SYSCK>
Main system clock (fm)
2.5/fcgck(max)
When the rising edge of fs/4 is detected twice after SYSCR2<SYSCK> is changed from 1 to 0, fm is stopped for synchronization.
When the rising edge of fcgck is detected twice after fm is stopped, fm is switched to fcgck.

Figure 2-13 Switching the Main System Clock (fm) (Switching from fs / 4 to fcgck)

Example : Switching from the SLOW1 mode to the NORMAL1 mode after the stability of the high-frequency clock oscillation circuit is confirmed at the warm-up counter (The warm-up time is about 4.0 [ms] at fc = 10 [MHz].)

```
                    ;#### Initialize routine ####
                    SET       (P0FC).2             ; P0FC<P0FC2> ← "1" (Uses P02/03 as oscillators)
                              :
                    LD        (WUCCR), 0x09        ; WUCCR<WUCDIV> ← "10" (Divided by 2)
                                                   ; WUCCR<WUCSEL> ← "0" (Selects fc as the source clock)
                    LD        (WUCDR), 0x9D        ; Sets the warm-up time
                                                   ; (Determines the time depending on the frequency and the
                                                   ; oscillator characteristics)
                                                   ; 4 [ms] / 25.6 [μs] = 156.25 → round up to "0x9D"
                    SET       (EIRL). 4            ; Enables INTWUC interrupts
                    SET       (SYSCR2) .6          ; SYSCR2<XEN> ← "1"
                                                   ; (Starts the high-frequency clock oscillation circuit and starts
                              :                    ; the warm-up counter)
                    ;#### Interrupt service routine of warm-up counter interrupts ####
PINTWUC:            CLR       (SYSCR2). 4          ; SYSCR2<SYSCK> ← "0"
                                                   ; (Switches the main system clock to the gear clock)
                    NOP                            ; Waits for 2 machine cycles
                    NOP
                    CLR       (SYSCR2). 5          ; SYSCR2<XTEN> ← "0"
                                                   ; (Turns off the low-frequency clock oscillation circuit)
                    RETI
                              :
VINTWUC:            DW        PINTWUC              ; INTWUC vector table
```

2.4. Reset Control Circuit

The reset circuit controls the external and internal factor resets and initializes the system.

2.4.1. Configuration

The reset control circuit consists of the following reset signal generation circuits:

(1) External reset input (external factor)
(2) Power-on reset (internal factor)
(3) Voltage detection reset 1 (internal factor)
(4) Watchdog timer reset (internal factor)
(5) System clock reset (internal factor)
(6) Trimming data reset (internal factor)
(7) Flash memory standby reset (internal factor)

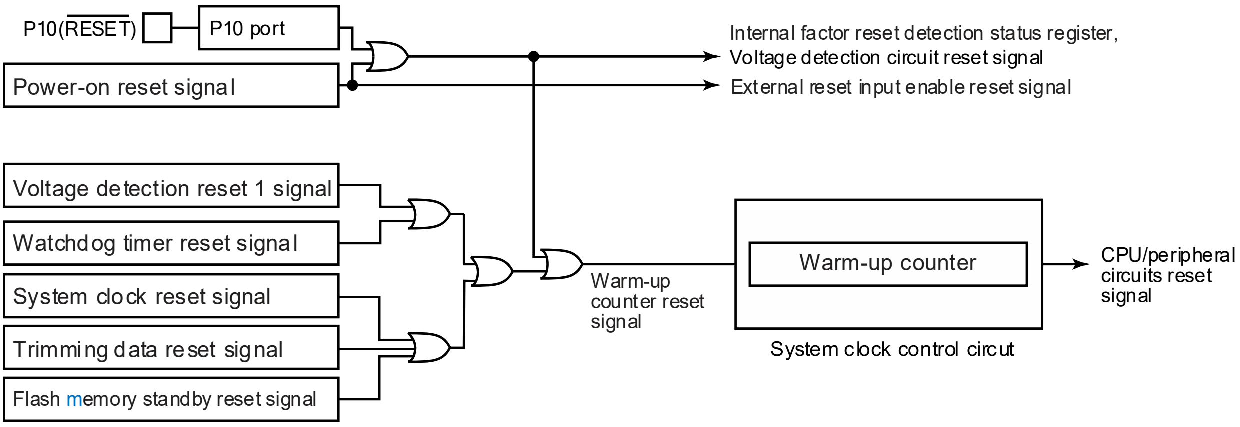

P10(RESET)
P10 port
Power-on reset signal
Internal factor reset detection status register,
Voltage detection circuit reset signal
External reset input enable reset signal
Voltage detection reset 1 signal
Watchdog timer reset signal
System clock reset signal
Trimming data reset signal
Flash memory standby reset signal
Warm-up counter reset signal
Warm-up counter
System clock control circut
CPU/peripheral circuits reset signal

Figure 2-14 Reset Control Circuit

2.4.2. Control

The reset control circuit is controlled by system control register 3 (SYSCR3), system control register 4 (SYSCR4), system control status register (SYSSR4) and the internal factor reset detection status register (IRSTSR).

System control register 3

SYSCR3		7	6	5	4	3	2	1	0
(0x0FDE)	Bit Symbol	-	-	-	-	-	(RVCTR)	(RAREA)	RSTDIS
	Read/Write	R	R	R	R	R	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

RSTDIS	External reset input enable register	0:	Enables the external reset input.
		1:	Disables the external reset input.

Note 1: The enabled SYSCR3<RSTDIS> is initialized by a power-on reset only, and cannot be initialized by an external reset input or internal factor reset. The value written in SYSCR3 is reset by a power-on reset, external reset input or internal factor reset.

Note 2: The value of SYSCR3<RSTDIS> is invalid until "0xB2" is written into SYSCR4.

Note 3: When the SYSCR4 is written to “0xB2” (Enable code for SYSCR3<RSTDIS>) after SYSCR3<RSTDIS> is modified, these operation should be executed continuously in NORMAL mode with fcgck = fc / 4 (CGCR<FCGCKSEL> = 00). Otherwise, SYSCR3<RSTDIS> may be enabled at unexpected timing.

Note 4: When a read instruction is executed on SYSCR3, bits 7 to 3 are read as "0".

System control register 4

SYSCR4		7	6	5	4	3	2	1	0
(0x0FDF)	Bit Symbol	SYSCR4							
	Read/Write	W							
	After reset	0	0	0	0	0	0	0	0

SYSCR4	Writes the SYSCR3 data control code.	0xB2:	Enables the contents of SYSCR3<RSTDIS>
		0xD4:	Enables the contents of SYSCR3<RAREA> and SYSCR3 <RVCTR>
		0x71:	Enables the contents of IRSTSR<FCLR>
		Others:	Invalid

Note 1: SYSCR4 is a write-only register, and must not be accessed by using a read-modify-write instruction, such as a bit operation.

Note 2: After SYSCR3<RSTDIS> is modified, SYSCR4 should be written "0xB2" (Enable code for SYSCR3<RSTDIS>) in NORMAL mode when fcgck is fc / 4 (CGCR<FCGCKSEL> = 00). Otherwise, SYSCR3<RSTDIS> may be enabled at unexpected timing.

Note 3: When the SYSCR4 is written to “0x71” (Enable code for IRSTSR) after IRSTSR is modified, these operation should be executed continuously in NORMAL mode with fcgck = fc / 4 (CGCR<FCGCKSEL> = 00). Otherwise, IRSTSR may be enabled at unexpected timing.

System control status register 4

SYSSR4		7	6	5	4	3	2	1	0
(0x0FDF)	Bit Symbol	-	-	-	-	-	(RVCTRS)	(RAREAS)	RSTDISS
	Read/Write	R	R	R	R	R	R	R	R
	After reset	0	0	0	0	0	0	0	0

RSTDISS	External reset input enable status	0: 1:	The enabled SYSCR3<RSTDIS> data is "0". The enabled SYSCR3<RSTDIS> data is "1".

Note 1: The enabled SYSCR3<RSTDIS> is initialized by a power-on reset only, and cannot be initialized by any other reset signals. The value written in SYSCR3 is reset by a power-on reset and other reset signals.

Note 2: When a read instruction is executed on SYSSR4, bits 7 to 3 are read as "0".

Internal factor reset detection status register

IRSTSR		7	6	5	4	3	2	1	0
(0x0FCC)	Bit Symbol	FCLR	FLSRF	TRMDS	TRMRF	-	LVD1RF	SYSRF	WDTRF
	Read/Write	W	R	R	R	R	R	R	R
	After reset	0	0	0	0	0	0	0	0

FCLR	Flag initialization control	0: 1:	- Clears the internal factor reset flag to "0".
FLSRF	Flash memory standby reset detection flag	0: 1:	- Detects the Flash memory standby reset.
TRMDS	Trimming data status	0: 1:	- The trimming data is incorrect status.
TRMRF	Trimming data reset detection flag	0: 1:	- Detects the trimming data reset.
LVD1RF	Voltage detection reset 1 detection flag	0: 1:	- Detects the voltage detection 1 reset
SYSRF	System clock reset detection flag	0: 1:	- Detects the system clock reset.
WDTRF	Watchdog timer reset detection flag	0: 1:	- Detects the watchdog timer reset

Note 1: The trimming data status (IRSTSR<TRMDS>) is initialized only by a power-on reset and an external reset input. Internal factor reset flag (IRSTSR<FLSRF>, <TRMRF>, <LVD1RF>, <SYSRF> and <WDTRF>) is not initialized by an internal factor reset. It is initialized only by a power-on reset, an external reset input or IRSTSR <FCLR>.

Note 2: Care must be taken in system designing since the IRSTSR may not fulfill its functions due to disturbing noise and other effects.

Note 3: When SYSCR4 is set to "0x71" after IRSTSR<FCLR> is set to "1", internal factor reset flag is cleared to "0" and IRSTSR<FCLR> is automatically cleared to "0".

Note 4: When the SYSCR4 is written to “0x71” (Enable code for IRSTSR<FCLR>) after IRSTSR<FCLR> is modified, these operation should be executed continuously in NORMAL mode with fcgck = fc / 4 (CGCR<FCGCKSEL> = 00). Otherwise, IRSTSR<FCLR> may be enabled at unexpected timing.

Note 5: When a read instruction is executed on IRSTSR, bits 7 and 3 are read as "0".

2.4.3. Functions

The reset signal of the power-on reset, the external reset input and the internal factor reset are input to the warm-up circuit of the clock generator.

During reset, the warm-up counter circuit is reset, and the CPU and the peripheral circuits are reset.

After reset is released, the warm-up counter starts counting the high-frequency clock (fc), and executes the warm-up operation that follows reset release.

During the warm-up operation that follows reset release, the trimming data is loaded from the non-volatile exclusive use memory for the adjustment.

When the warm-up operation that follows reset release is finished, the CPU starts execution of the program from the reset vector address stored in addresses "0xFFFE" to "0xFFFF".

When a reset signal is input during the warm-up operation that follows reset release, the warm-up counter circuit is reset.

The reset operation is common to the power-on reset, the external reset input and the internal factor resets, except for the initialization of some special function registers and the initialization of the voltage detection circuits.

When a reset signal is applied, the peripheral circuits become the states as shown in Table 2-5.

Table 2-5 Initialization of Built-in Hardware by Reset Operation and Its Status after Release

Built-in hardware	During reset	During the warm-up operation that follows reset release	Immediately after the warm-up operation that follows reset release
Program counter (PC)	MCU mode: 0xFFFE Serial PROM mode: 0x1FFE	MCU mode: 0xFFFE Serial PROM mode: 0x1FFE	MCU mode: 0xFFFE Serial PROM mode: 0x1FFE
Stack pointer (SP)	0x00FF	0x00FF	0x00FF
RAM	Undefined	Undefined	Undefined
General-purpose registers (W, A, B, C, D, E, H, L, IX and IY)	Undefined	Undefined	Undefined
Register bank selector (RBS)	0	0	0
Jump status flag (JF)	Undefined	Undefined	Undefined
Zero flag (ZF)	Undefined	Undefined	Undefined
Carry flag (CF)	Undefined	Undefined	Undefined
Half carry flag (HF)	Undefined	Undefined	Undefined
Sign flag (SF)	Undefined	Undefined	Undefined
Overflow flag (VF)	Undefined	Undefined	Undefined
Interrupt master enable flag (<IMF>)	0	0	0
Individual interrupt enable flag (<EFx>)	0	0	0
Interrupt latch (<ILx>)	0	0	0
High-frequency clock oscillation circuit	Oscillation enabled	Oscillation enabled	Oscillation enabled
Low-frequency clock oscillation circuit	Oscillation disabled	Oscillation disabled	Oscillation disabled
Warm-up counter	Reset	Start	Stop
Timing generator prescaler and divider	0	0	0
Watchdog timer	Disabled	Disabled	Enabled
Voltage detection circuit	Disabled or enabled	Disabled or enabled	Disabled or enabled
I/O port pin status	Hi-Z	Hi-Z	Hi-Z
Special function register	Refer to the SFR map.	Refer to the SFR map.	Refer to the SFR map.

Note: The voltage detection circuits are disabled by an external reset input or power-on reset only.

2.4.4. Reset Signal Generating Factors

Reset signals are generated by each factor as follows:

2.4.4.1. Power-on reset

The power-on reset is an internal reset that occurs when power is turned on.

During power-up, a power-on reset signal is generated while the supply voltage is the power-on reset release voltage (V_{PROFF}) or less. When the supply voltage rises above the power-on reset release voltage (V_{PROFF}), the power-on reset signal is released.

During power-down, a power-on reset signal is generated when the supply voltage is the power-on reset detection voltage or less.

For more details, refer to "6. Power-on Reset Circuit".

2.4.4.2. External reset input ($\overline{\text{RESET}}$ pin input)

This is an external reset that is generated by the $\overline{\text{RESET}}$ pin input. Port P10 is also used as the $\overline{\text{RESET}}$ pin, and it is configured as the $\overline{\text{RESET}}$ pin at power-up.

- During power-up
  - When the supply voltage rises rapidly

    When the power supply rise time (t_{VDD}) is shorter than 5 [ms] with enough margin, the reset can be released by a power-on reset or an external reset ($\overline{\text{RESET}}$ pin input).

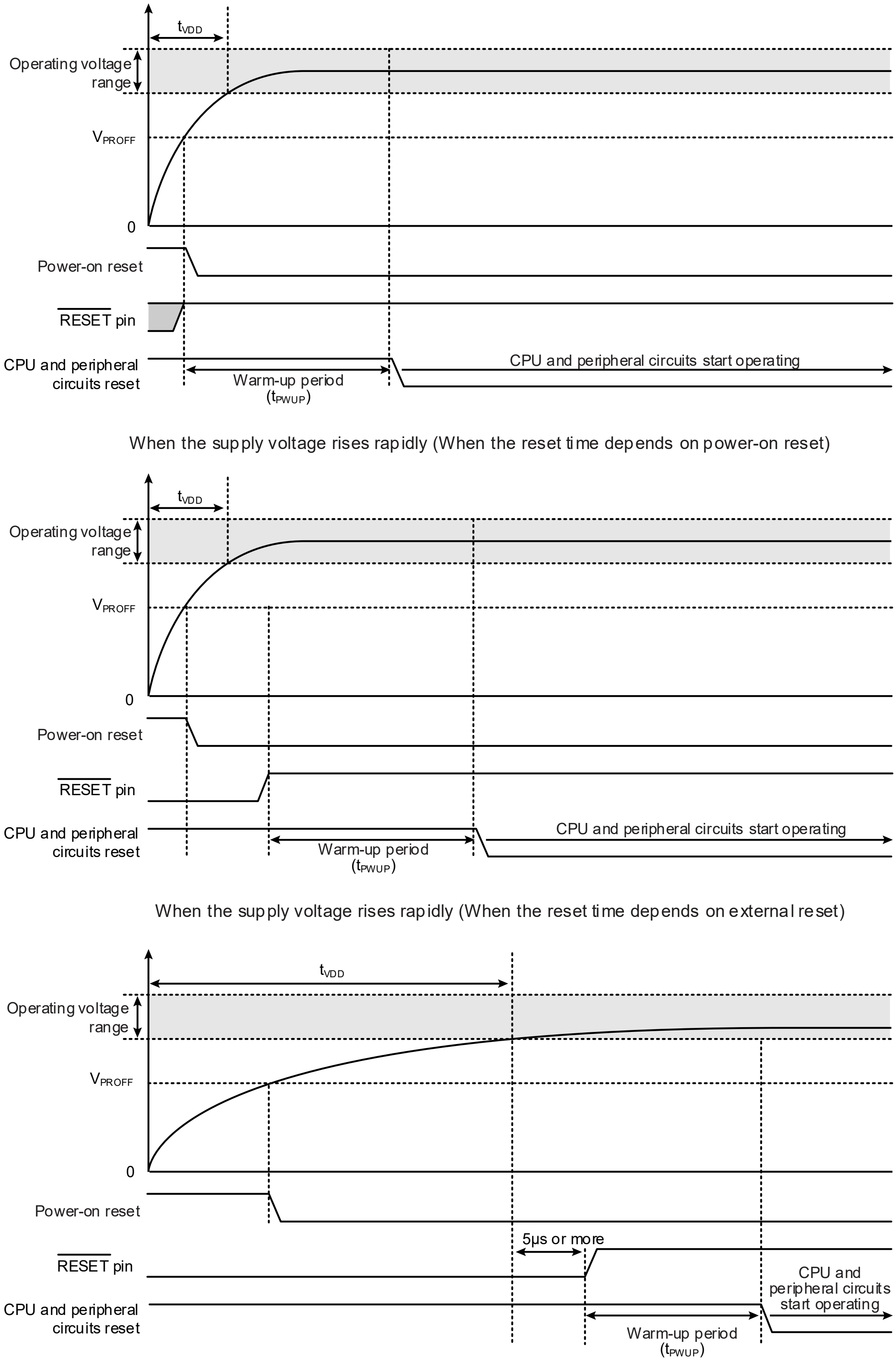
    The power-on reset logic and external reset ($\overline{\text{RESET}}$ pin input) logic are ORed. This means that the TMP89FS60B/62B/63B is reset when either or both of these reset sources are asserted.

    Therefore, the reset time is determined by the reset source with a longer reset period. When the $\overline{\text{RESET}}$ pin level changes from “Low” to “High” level before the supply voltage rises above the power-on-reset release voltage (or when the $\overline{\text{RESET}}$ pin level is “High” level from the beginning), the reset time depends on the power-on reset. When the $\overline{\text{RESET}}$ pin level changes from “Low” level to “High” level after the supply voltage rises above the power-on-reset release voltage, the reset time depends on the external reset.

    In the former case, a warm-up period begins when the power-on reset signal is released. In the latter case, a warm-up period begins when the $\overline{\text{RESET}}$ pin level becomes High. Upon completion of the warm-up period, the CPU and peripheral circuits start operating (Figure 2-15).

  - When the supply voltage rises slowly

    When the power supply rise time (t_{VDD}) is longer than 5 [ms], the reset must be released by using the $\overline{\text{RESET}}$ pin. In this case, hold the $\overline{\text{RESET}}$ pin “Low” level until the supply voltage rises to the operating voltage range and oscillation is stabilized. When this state is achieved, wait at least 5 [μs] and then pull the $\overline{\text{RESET}}$ pin “High” level. Changing the $\overline{\text{RESET}}$ pin level to “High” starts a warm-up period. Upon completion of the warm-up period, the CPU and peripheral circuits start operating (Figure 2-15).



t_{VDD}
Operating voltage range
V_{PROFF}
0
Power-on reset
$\overline{\text{RESET}}$ pin
CPU and peripheral circuits reset
Warm-up period (t_{PWUP})
CPU and peripheral circuits start operating

When the supply voltage rises rapidly (When the reset time depends on power-on reset)

t_{VDD}
Operating voltage range
V_{PROFF}
0
Power-on reset
$\overline{\text{RESET}}$ pin
CPU and peripheral circuits reset
Warm-up period (t_{PWUP})
CPU and peripheral circuits start operating

When the supply voltage rises rapidly (When the reset time depends on external reset)

t_{VDD}
Operating voltage range
V_{PROFF}
0
Power-on reset
5μs or more
$\overline{\text{RESET}}$ pin
CPU and peripheral circuits reset
Warm-up period (t_{PWUP})
CPU and peripheral circuits start operating

When the supply voltage rises slowly

Figure 2-15 External Reset Input (During Power Up)

– When the supply voltage is within the operating voltage range

When the supply voltage is within the operating voltage range and stable oscillation is achieved, holding the $\overline{\text{RESET}}$ pin “Low” level for 5 [µs] or longer generates a reset. Then, changing the $\overline{\text{RESET}}$ pin level to “High” starts a warm-up period. Upon completion of the warm-up period, the CPU and peripheral circuits start operating (Figure 2-16).

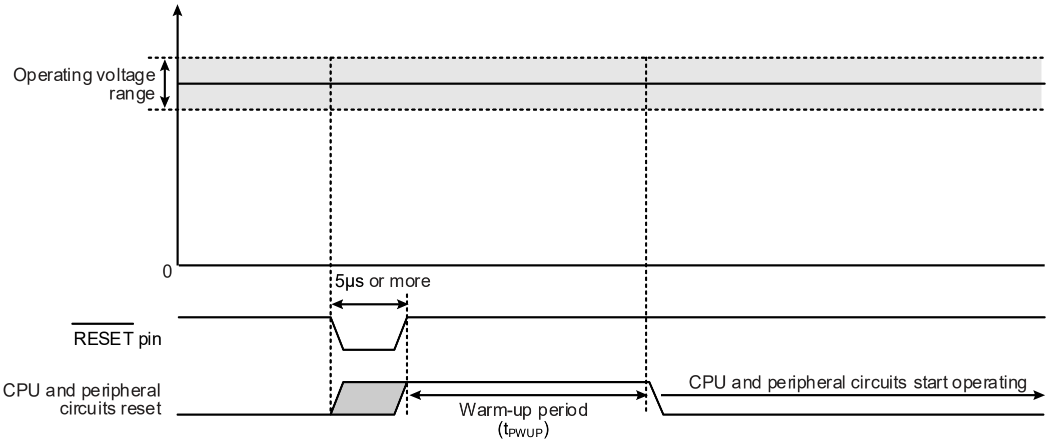

Operating voltage range
0
5µs or more
$\overline{\text{RESET}}$ pin
CPU and peripheral circuits reset
Warm-up period (t_{PWUP})
CPU and peripheral circuits start operating

Figure 2-16 External Reset Input (When the Power Supply is Stable)

2.4.4.3. Voltage detection reset

The voltage detection reset is an internal factor reset that occurs when it is detected that the supply voltage has matched a predetermined detection voltage.

Refer to "7. Voltage Detection Circuit".

2.4.4.4. Watchdog timer reset

The watchdog timer reset is an internal factor reset that occurs when an overflow of the watchdog timer is detected.

Refer to "5. Watchdog Timer (WDT)".

2.4.4.5. System clock reset

The system clock reset is an internal factor reset that occurs when it is detected that the setting of the oscillation circuit and the timing generator register is set to a combination that puts the CPU into deadlock.

Refer to "2.3. System Clock Controller".

2.4.4.6. Trimming data reset

The trimming data reset is an internal factor reset that occurs when the trimming data latched in the internal circuit is broken down during operation due to noise or other factors.

The trimming data is a data bit provided for adjustment of the internal circuits.

This bit is loaded from the non-volatile exclusive use memory during the warm-up time that follows reset release (t_{PWUP}) and latched into the internal circuit.

When the trimming data loaded from the non-volatile exclusive use memory during the warm-up operation that follows reset release is abnormal, IRSTSR<TRMDS> is set to "1".

When IRSTSR<TRMDS> is read as "1" in the initialize routine immediately after reset release, the trimming data need to be reloaded by generating an internal factor reset, such as a system clock reset, and activating the warm-up operation again.

When IRSTSR<TRMDS> is still set to "1" after repeated reading, the internal circuits cannot operation correctly. Design the system so that the system will not be damaged in such a case.

2.4.4.7. Flash memory standby reset

The Flash memory standby reset which is used to keep compatibility when a software for the TMP89FS60 is applied to TMP89FS60B/62B/63B is an internal factor reset.

When a software is programmed anew for the TMP89FS60B/62B/63B, the Flash memory reset doesn’t need to be used. For more details, refer to "21. Flash Memory".

2.4.4.8. Internal factor reset detection status register

By reading the internal factor reset detection status register IRSTSR after the release of an internal factor reset, except the power-on reset, the factor which causes a reset can be detected.

The internal factor reset detection status register is initialized by an external reset input, power-on reset or IRSTSR<FCLR>.

Set IRSTSR<FCLR> to "1" and write "0x71" to SYSCR4. This enables IRSTSR<FCLR> and the bit 6 and bits 4 to 0 of IRSTR are cleared to "0". At the same time, IRSTSR<FCLR> is cleared to "0" automatically

Note 1: Care must be taken in system designing since the IRSTSR may not fulfill its functions due to disturbing noise and other effects.

Note 2: When the SYSCR4 is written to “0x71” (Enable code for IRSTSR<FCLR>) after IRSTSR<FCLR> is modified, these operation should be executed continuously in NORMAL mode with fcgck = fc / 4 (CGCR<FCGCKSEL> = 00). Otherwise, IRSTSR<FCLR> may be enabled at unexpected timing.

2.4.4.9. How to use the external reset input pin as a port

To use the external reset input pin as a port, keep the external reset input pin at the “High” level until the power is turned on and the warm-up operation that follows reset release is finished.

After the warm-up operation that follows reset release is finished, set <P1PU0> to "1" and <P1CR0> to "0", and connect a pull-up resistor for a port. Then set SYSCR3<RSTDIS> to "1" and write "0xB2" to SYSCR4. This disables the external reset function and makes the external reset input pin usable as a normal port.

To use the pin as an external reset pin when it is used as a port, set <P1PU0> to "1" and <P1CR0> to "0" and connect the pull-up resistor to put the pin to the input mode. Then clear SYSCR3<RSTDIS> to "0" and write "0xB2" to SYSCR4. This enables the external reset function and makes the pin usable as the external reset input pin.

Note 1: When switching the external reset input pin to a port or switching the pin used as a port to the external reset input pin, do it when the pin is stabilized at the “High” level. Switching the pin function when the “Low” level is input may cause a reset.

Note 2: When the external reset input is used as a port, the statement which clears SYSCR3<RSTDIS> to "0" should be not written in a program. When it is in a program, the external reset input set as a port may be changed as the external reset input at unexpected timing by the abnormal execution of program,.

Note 3: When the SYSCR4 is written to “0xB2” (Enable code for SYSCR3<RSTDIS>) after SYSCR3<RSTDIS> is modified, these operation should be executed continuously in NORMAL mode with fcgck = fc / 4 (CGCR<FCGCKSEL> = 00). Otherwise, SYSCR3<RSTDIS> may be enabled at unexpected timing.

3. Interrupt Control Circuit

The TMP89FS60B/62B/63B has a total of 27 interrupt sources excluding reset. Interrupts can be nested with priorities. Three of the internal interrupt sources are non-maskable interrupt and others are maskable interrupt.

Table 3-1 All interrupt sources of the TMP89FS60B/62B/63B

Basic priority	Interrupt sources		Enable condition	Interrupt latch	Vector Address (MCU mode)	
					SYSSR4 <RVCTR> = 0	SYSSR4 <RVCTR> = 1
1	Internal/ External	(Reset)	Non-maskable	-	0xFFFE	-
2	Internal	INTSWI	Non-maskable	-	0xFFFC	0x01FC
2	Internal	INTUNDEF	Non-maskable	-	0xFFFC	0x01FC
2	Internal	INTWDT	Non-maskable	ILL<IL3>	0xFFF8	0x01F8
5	Internal	INTWUC	<IMF> AND EIRL<EF4> = 1	ILL<IL4>	0xFFF6	0x01F6
6	Internal	INTTBT	<IMF> AND EIRL<EF5> = 1	ILL<IL5>	0xFFF4	0x01F4
7	Internal	INTRXD0/INTSIO0	<IMF> AND EIRL<EF6> = 1	ILL<IL6>	0xFFF2	0x01F2
8	Internal	INTTXD0	<IMF> AND EIRL<EF7> = 1	ILL<IL7>	0xFFF0	0x01F0
9	External	INT5	<IMF> AND EIRH<EF8> = 1	ILH<IL8>	0xFFEE	0x01EE
10	Internal	INTVLTD	<IMF> AND EIRH<EF9> = 1	ILH<IL9>	0xFFEC	0x01EC
11	Internal	INTADC	<IMF> AND EIRH<EF10> = 1	ILH<IL10>	0xFFEA	0x01EA
12	Internal	INTRTC	<IMF> AND EIRH<EF11> = 1	ILH<IL11>	0xFFE8	0x01E8
13	Internal	INTTC00	<IMF> AND EIRH<EF12> = 1	ILH<IL12>	0xFFE6	0x01E6
14	Internal	INTTC01	<IMF> AND EIRH<EF13> = 1	ILH<IL13>	0xFFE4	0x01E4
15	Internal	INTTCA0	<IMF> AND EIRH<EF14> = 1	ILH<IL14>	0xFFE2	0x01E2
16	Internal	INTSBI0/INTSIO0 (Note 5)	<IMF> AND EIRH<EF15> = 1	ILH<IL15>	0xFFE0	0x01E0
17	External	INT0 (Note 6)	<IMF> AND EIRE<EF16> = 1	ILE<IL16>	0xFFDE	0x01DE
18	External	INT1 (Note 6)	<IMF> AND EIRE<EF17> = 1	ILE<IL17>	0xFFDC	0x01DC
19	External	INT2	<IMF> AND EIRE<EF18> = 1	ILE<IL18>	0xFFDA	0x01DA
20	External	INT3	<IMF> AND EIRE<EF19> = 1	ILE<IL19>	0xFFD8	0x01D8
21	External	INT4	<IMF> AND EIRE<EF20> = 1	ILE<IL20>	0xFFD6	0x01D6
22	Internal	INTTCA1	<IMF> AND EIRE<EF21> = 1	ILE<IL21>	0xFFD4	0x01D4
23	Internal	INTRXD1/INTSIO1	<IMF> AND EIRE<EF22> = 1	ILE<IL22>	0xFFD2	0x01D2
24	Internal	INTTXD1	<IMF> AND EIRE<EF23> = 1	ILE<IL23>	0xFFD0	0x01D0
25	Internal	INTTC02	<IMF> AND EIRD<EF24> = 1	ILD<IL24>	0xFFCE	0x01CE
26	Internal	INTTC03	<IMF> AND EIRD<EF25> = 1	ILD<IL25>	0xFFCC	0x01CC
27	Internal	INTRXD2	<IMF> AND EIRD<EF26> = 1	ILD<IL26>	0xFFCA	0x01CA
28	Internal	INTTXD2	<IMF> AND EIRD<EF27> = 1	ILD<IL27>	0xFFC8	0x01C8

Note 1: Vector address areas can be changed by the SYSCR3<RVCTR> setting. To assign vector address areas to RAM, set SYSCR3<RVCTR> to "1" and SYSCR3<RAREA> to "1" and write “0xD4” to SYSCR4 to be valid them.

Note 2: "0xFFFA" and "0xFFFB" function not as interrupt vectors but as option codes in the Serial PROM mode. For details, refer to "22. Serial PROM Mode".

Note 3: When the watchdog timer interrupt (INTWDT) is used, clear WDTCR1<EDTOUT> to "0". (After releasing a reset, After releasing reset, the WDT outputs "Watchdog timer reset signal".)

Note 4: According to selecting the shared interrupt sources, refer to "8.5. Serial Interface Selecting Function".

Note 5: The INTSBI0/INTSIO0 is “Reserved” for the TMP89FS62B.

Note 6: The INT0 and INT1 is “Reserved” for the TMP89FS62B and TMP89FS63B.

3.1. Configuration

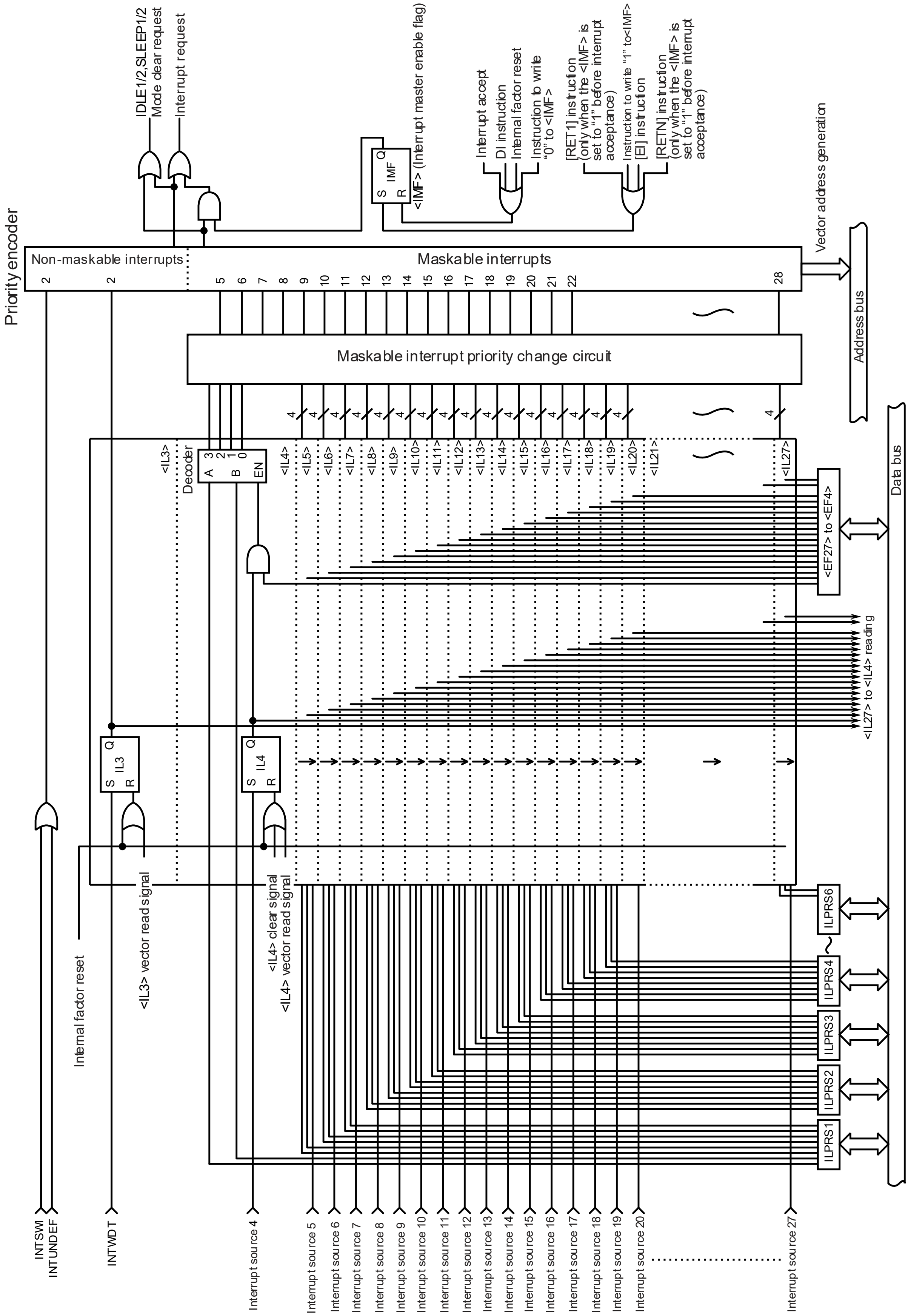


Figure 3-1 Interrupt Control Circuit

3.2. Interrupt Sources

Interrupt sources are provided with interrupt latches (<ILx>), which hold interrupt requests, and have independent vector addresses. When a request for an interrupt is generated, its interrupt latch is set to "1", which requests the CPU to accept the interrupt. Acceptance of interrupts is enabled or disabled by software using the interrupt master enable flag (<IMF>) and individual enable flag (<EFx>) for each interrupt source.

3.3. Interrupt Priority

When multiple maskable interrupts are generated simultaneously, the interrupts are accepted in order of descending priority. The priorities are determined by the interrupt priority change control register (ILPRS1 to ILPRS6) as Levels and determined by the hardware as the basic priorities.

However, there are no prioritized interrupt sources among non-maskable interrupts.

3.4. Interrupt Sources for Each Product

The interrupt sources for each product are shown in Table 3-2.
The individual interrupt enable flag (EFx) whose interrupt source is not available should be cleared to “0”.

Table 3-2 Interrupt sources for each product

Basic priority	Interrupt sources		TMP89FS60B	TMP89FS62B	TMP89FS63B
1	Internal/ External	(Reset)	A	A	A
2	Internal	INTSWI	A	A	A
2	Internal	INTUNDEF	A	A	A
2	Internal	INTWDT	A	A	A
5	Internal	INTWUC	A	A	A
6	Internal	INTTBT	A	A	A
7	Internal	INTRXD0/INTSIO0	A	A	A
8	Internal	INTTXD0	A	A	A
9	External	INT5	A	A	A
10	Internal	INTVLTD	A	A	A
11	Internal	INTADC	A	A	A
12	Internal	INTRTC	A	A	A
13	Internal	INTTC00	A	A	A
14	Internal	INTTC01	A	A	A
15	Internal	INTTCA0	A	A	A
16	Internal	INTSBI0/INTSIO0	A	NA	A
17	External	INT0	A	NA	NA
18	External	INT1	A	NA	NA
19	External	INT2	A	A	A
20	External	INT3	A	A	A
21	External	INT4	A	A	A
22	Internal	INTTCA1	A	A	A
23	Internal	INTRXD1/INTSIO1	A	A	A
24	Internal	INTTXD1	A	A	A
25	Internal	INTTC02	A	A	A
26	Internal	INTTC03	A	A	A
27	Internal	INTRXD2	A	A	A
28	Internal	INTTXD2	A	A	A

Note: A: Available, NA: Not available

3.5. Interrupt Latches (<IL27> to <IL3>)

An interrupt latch is provided for each interrupt source, except for a software interrupt and an undefined instruction execution interrupt. When an interrupt request is generated, the latch is set to "1", and the CPU is requested to accept the interrupt when its acceptance is enabled. The interrupt latch is cleared to "0" immediately after the interrupt is accepted. All interrupt latches are initialized to "0" during reset.

The interrupt latches are located at addresses "0x0FE0", "0x0FE1", "0x0FE2", "0x0FE3" in SFR area. Each latch can be cleared to "0" individually by an instruction. However, <IL3> interrupt latch cannot be cleared by instructions.

Do not use any read-modify-write instruction, such as a bit manipulation or operation instruction, because it may clear interrupt requests generated while the instruction is executed.

Interrupt latches cannot be set to "1" by using an instruction. Writing "1" to an interrupt latch is equivalent to denying clearing of the interrupt latch, and not setting the interrupt latch.

Since interrupt latches can be read by instructions, the status of interrupt requests can be monitored by software.

Note: In the main program, before manipulating the interrupt enable flag (<ILx>), be sure to clear the master enable flag (<IMF>) to "0" (Clearing <IMF> to "0" by [DI] instruction). Then set the <IMF> to "1" as required after operating the <ILx> (Setting <IMF> to "1" by [EI] instruction).
In the interrupt service routine, the <IMF> becomes "0" automatically and needs not be cleared to "0" normally. However, when using multiple interrupt in the interrupt service routine, manipulate the <ILx> before setting the <IMF> to "1".

Example 1: Clears interrupt latches

```
        DI                                  ; <IMF> ← "0"
        LD      (ILL), 0x3F                 ; <IL7> to <IL6> ← "0"
        LD      (ILH), 0xE8                 ; <IL12>, <IL10> to <IL8> ← "0"
        EI                                  ; <IMF> ← "1"
```

Example 2: Reads interrupt latches

```
        LD      WA, (ILL)                   ; W ← ILH, A ← ILL
```

Example 3: Tests interrupt latches

```
        TEST    (ILL).7                     ; When <IL7> = 1 then jump to SSET
        JR      F, SSET
                :
                :
    SSET:
```

3.6. Interrupt Enable Register (EIR)

The interrupt enable register (EIR) enables and disables the acceptance of interrupts, except for the non-maskable interrupts (software interrupt, undefined instruction interrupt and watchdog interrupt). Non-maskable interrupts are accepted regardless of the contents of the EIR.

The EIR consists of the interrupt master enable flag (<IMF>) and the individual interrupt enable flags (<EFx>). These registers are located at addresses "0x003A", "0x003B", "0x003C", "0x003D" in the SFR area, and they can be read and written by instructions (including read-modify-write instructions such as bit manipulation or operation instructions).

3.6.1. Interrupt master enable flag (<IMF>)

The interrupt master enable flag (<IMF>) enables and disables the acceptance of all maskable interrupts. Clearing the IMF to "0" disables the acceptance of all maskable interrupts. Setting the IMF to "1" enables the acceptance of the interrupts that are specified by the individual interrupt enable flags.

When an interrupt is accepted, the <IMF> is stacked and then cleared to "0", which temporarily disables the subsequent maskable interrupts. After the interrupt service routine is executed, the stacked data, which was the status before interrupt acceptance, reloaded on the <IMF> by return interrupt instruction [RETI]/[RETN].

The <IMF> is located on bit 0 in EIRL (Address: "0x003A" in SFR), and can be read and written by instructions. The <IMF> is normally set and cleared by [EI] and [DI] instructions respectively. During reset, the <IMF> is initialized to "0".

3.6.2. Individual interrupt enable flags (<EF27> to <EF4>)

Each of these flags enables and disables the acceptance of its maskable interrupt. Setting the corresponding bit of an individual interrupt enable flag to "1" enables acceptance of its interrupt, and setting the bit to "0" disables acceptance.

During reset, all the individual interrupt enable flags are initialized to "0" and no maskable interrupts are accepted until the flags are set to "1".

Note: In the main program, before manipulating the interrupt enable flag (<EFx>), be sure to clear the master enable flag (<IMF>) to "0" (Clearing <IMF> to “0” by [DI] instruction). Then set the <IMF> to "1" as required after operating the <EFx> (Setting <IMF> to “1” by [EI] instruction).
In the interrupt service routine, the <IMF> becomes "0" automatically and needs not be cleared to "0" normally. However, when using multiple interrupt in the interrupt service routine, manipulate the <EFx> before setting the <IMF> to "1".

Example: Enables interrupts individually and sets <IMF>

```
DI                                          ; <IMF> ← "0"
LDW     (EIRL), 0xE8A0                      ; <EF15> to <EF13>, <EF11>, <EF7>, <EF5> ← "1"
        :                                   ; <IMF> should not be set simultaneously with <EFx>.
        :
EI                                          ; <IMF> ← "1"
```

Interrupt latch (ILL)

ILL		7	6	5	4	3	2	1	0
(0x0FE0)	Bit Symbol	IL7	IL6	IL5	IL4	IL3	-	-	-
	Read/Write	R/W	R/W	R/W	R/W	R	R	R	R
	After reset	0	0	0	0	0	0	0	0
	Function	INTTXD0	INTRXD0/ INTSIO0	INTTBT	INTWUC	INTWDT	-	-	-

Interrupt latch (ILH)

ILH		7	6	5	4	3	2	1	0
(0x0FE1)	Bit Symbol	IL15	IL14	IL13	IL12	IL11	IL10	IL9	IL8
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0
	Function	INTSBI0/ INTSIO0 (Note 6)	INTTCA0	INTTC01	INTTC00	INTRTC	INTADC	INTVLTD	INT5

Interrupt latch (ILE)

ILE		7	6	5	4	3	2	1	0
(0x0FE2)	Bit Symbol	IL23	IL22	IL21	IL20	IL19	IL18	IL17	IL16
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0
	Function	INTTXD1	INTRXD1/ INTSIO1	INTTCA1	INT4	INT3	INT2	INT1 (Note 7)	INT0 (Note 7)

Interrupt latch (ILD)

ILD		7	6	5	4	3	2	1	0
(0x0FE3)	Bit Symbol	-	-	-	-	IL27	IL26	IL25	IL24
	Read/Write	R	R	R	R	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0
	Function	-	-	-	-	INTTXD2	INTRXD2	INTTC03	INTTC02

			When reading	When writing
IL27 to IL4	Interrupt latch	0:	No interrupt request	Clears the interrupt request (Notes 2 and 3)
		1:	Interrupt request	<ILx> is not set to “1” by writing "1".
IL3		0:	No interrupt request	(Note 1)
		1:	Interrupt request	

Note 1: <IL3> is a read-only register. Writing the register does not affect <IL3>.

Note 2: In the main program, before manipulating the interrupt enable flag (<ILx>), be sure to clear the master enable flag (<IMF>) to "0" (Clearing <IMF> to “0” by [DI] instruction). Then set the <IMF> to "1" as required after operating the <ILx> (Setting <IMF> to “1” by [EI] instruction).
In the interrupt service routine, the <IMF> becomes "0" automatically and needs not be cleared to "0" normally. However, when using multiple interrupt in the interrupt service routine, manipulate the <ILx> before setting the <IMF> to "1".

Note 3: Do not clear <ILx> with read-modify-write instructions such as bit operations.

Note 4: When a read instruction is executed on ILL, bits 2 to 0 are read as "0".

Note 5: When a read instruction is executed on ILD, bits 7 to 4 are read as “0”.

Note 6: <IL15> is “Reserved (this bit symbol must be cleared to “0”.)” for the TMP89FS62B.

Note 7: <IL17> and <IL16> are “Reserved (this bit symbol must be cleared to “0”.)” for the TMP89FS62B and TMP89FS63B.

Interrupt enable register (EIRL)

EIRL		7	6	5	4	3	2	1	0
(0x003A)	Bit Symbol	EF7	EF6	EF5	EF4	-	-	-	IMF
	Read/Write	R/W	R/W	R/W	R/W	R	R	R	R/W
	After reset	0	0	0	0	0	0	0	0
	Function	INTTXD0	INTRXD0/ INTSIO0	INTTBT	INTWUC	-	-	-	Interrupt master enable flag

Interrupt enable register (EIRH)

EIRH		7	6	5	4	3	2	1	0
(0x003B)	Bit Symbol	EF15	EF14	EF13	EF12	EF11	EF10	EF9	EF8
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0
	Function	INTSBI0/ INTSIO0 (Note 5)	INTTCA0	INTTC01	INTTC00	INTRTC	INTADC	INTVLTD	INT5

Interrupt enable register (EIRE)

EIRE		7	6	5	4	3	2	1	0
(0x003C)	Bit Symbol	EF23	EF22	EF21	EF20	EF19	EF18	EF17	EF16
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0
	Function	INTTXD1	INTRXD1/ INTSIO1	INTTCA1	INT4	INT3	INT2	INT1 (Note 6)	INT0 (Note 6)

Interrupt enable register (EIRD)

EIRD		7	6	5	4	3	2	1	0
(0x003D)	Bit Symbol	-	-	-	-	EF27	EF26	EF25	EF24
	Read/Write	R	R	R	R	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0
	Function	-	-	-	-	INTTXD2	INTRXD2	INTTC03	INTTC02

EF27 to EF4	Enables the acceptance of all maskable interrupts.	0: 1:	Disables the acceptance of each maskable interrupt. Enables the acceptance of each maskable interrupt.
IMF	Interrupt master enable flag	0: 1:	Disables the acceptance of all maskable interrupts. Enables the acceptance of all maskable interrupts.

Note 1: Do not set the <IMF> and the interrupt enable flag (<EF15> to <EF4>) to "1" at the same time.

Note 2: In the main program, before manipulating the interrupt enable flag (<EFx>), be sure to clear the master enable flag (<IMF>) to "0" (Clearing <IMF> to “0” by [DI] instruction). Then set the <IMF> to "1" as required after operating the <EFx> (Setting <IMF> to “1” by [EI] instruction).
In the interrupt service routine, the <IMF> becomes "0" automatically and needs not be cleared to "0" normally. However, when using multiple interrupt in the interrupt service routine, manipulate the <EFx> before setting the <IMF> to "1".

Note 3: When a read instruction is executed on EIRL, bits 3 to 1 are read as "0".

Note 4: When a read instruction is executed on EIRD, bits 7 to 4 are read as “0”.

Note 5: <EF15> is “Reserved (this bit symbol must be cleared to “0”.)” for the TMP89FS62B.

Note 6: <EF17> and <EF16> are “Reserved (this bit symbol must be cleared to “0”.)” for the TMP89FS62B and the TMP89FS63B.

3.7. Maskable Interrupt Priority Change Function

The priority of maskable interrupts (<IL4> to <IL27>) can be changed to four levels, Levels 0 to 3 by the interrupt priority change control register (ILPRS1 to ILPRS6), regardless of the basic priorities 5 to 28.

To raise the interrupt priority, set the Level to a larger number. To lower the interrupt priority, set the Level to a smaller number. When different maskable interrupts are generated simultaneously at the same level, the interrupt with higher basic priority is processed preferentially.

For example, when the ILPRS1 register is set to "0xC0" and interrupts of <IL4> and <IL7> are generated at the same time, the interrupt of <IL7> is preferentially processed (<EF4> and <EF7> have been enabled.).

After reset is released, all maskable interrupts are set to priority level 0 (the lowest priority).

Note: In the main program, before manipulating the interrupt priority change control register (ILPRS1 to 6), be sure to clear the master enable flag (<IMF>) to "0" (Disable interrupt by [DI] instruction).
Set the <IMF> to "1" as required after operating ILPRS1 to 6 (Enable interrupt by [EI] instruction).
In the interrupt service routine, the <IMF> becomes "0" automatically and needs not be cleared to "0" normally.
However, when using multiple interrupt in the interrupt service routine, manipulate ILPRS1 to 6 before setting the <IMF> to "1".

Interrupt priority change control register 1

ILPRS1		7	6	5	4	3	2	1	0
(0x0FF0)	Bit Symbol	IL07P		IL06P		IL05P		IL04P	
	Read/Write	R/W		R/W		R/W		R/W	
	After reset	0	0	0	0	0	0	0	0

IL07P	Sets the interrupt priority of <IL7>.	00:	Level 0 (lower priority)
IL06P	Sets the interrupt priority of <IL6>.	01:	Level 1
IL05P	Sets the interrupt priority of <IL5>.	10:	Level 2
IL04P	Sets the interrupt priority of <IL4>.	11:	Level 3 (higher priority)

Interrupt priority change control register 2

ILPRS2		7	6	5	4	3	2	1	0
(0x0FF1)	Bit Symbol	IL11P		IL10P		IL09P		IL08P	
	Read/Write	R/W		R/W		R/W		R/W	
	After reset	0	0	0	0	0	0	0	0

IL11P	Sets the interrupt priority of <IL11>.	00:	Level 0 (lower priority)
IL10P	Sets the interrupt priority of <IL10>.	01:	Level 1
IL09P	Sets the interrupt priority of <IL9>.	10:	Level 2
IL08P	Sets the interrupt priority of <IL8>.	11:	Level 3 (higher priority)

Interrupt priority change control register 3

ILPRS3		7	6	5	4	3	2	1	0
(0x0FF2)	Bit Symbol	IL15P		IL14P		IL13P		IL12P	
	Read/Write	R/W		R/W		R/W		R/W	
	After reset	0	0	0	0	0	0	0	0

IL15P (Note)	Sets the interrupt priority of <IL15>.	00:	Level 0 (lower priority)
IL14P	Sets the interrupt priority of <IL14>.	01:	Level 1
IL13P	Sets the interrupt priority of <IL13>.	10:	Level 2
IL12P	Sets the interrupt priority of <IL12>.	11:	Level 3 (higher priority)

Note: <IL15P> is “Reserved (this bit symbol must be cleared to “00”.)” for the TMP89FS62B.

Interrupt priority change control register 4

ILPRS4		7	6	5	4	3	2	1	0
(0x0FF3)	Bit Symbol	IL19P		IL18P		IL17P		IL16P	
	Read/Write	R/W		R/W		R/W		R/W	
	After reset	0	0	0	0	0	0	0	0

IL19P	Sets the interrupt priority of <IL19>.	00:	Level 0 (lower priority)
IL18P	Sets the interrupt priority of <IL18>.	01:	Level 1
IL17P (Note)	Sets the interrupt priority of <IL17>.	10:	Level 2
IL16P (Note)	Sets the interrupt priority of <IL16>.	11:	Level 3 (higher priority)

Note: <IL17P> and <IL16P> are “Reserved (this bit symbol must be cleared to “00”.)” for the TMP89FS62B and the TMP89FS63B.

Interrupt priority change control register 5

ILPRS5		7	6	5	4	3	2	1	0
(0x0FF4)	Bit Symbol	IL23P		IL22P		IL21P		IL20P	
	Read/Write	R/W		R/W		R/W		R/W	
	After reset	0	0	0	0	0	0	0	0

IL23P	Sets the interrupt priority of <IL23>.	00:	Level 0 (lower priority)
IL22P	Sets the interrupt priority of <IL22>.	01:	Level 1
IL21P	Sets the interrupt priority of <IL21>.	10:	Level 2
IL20P	Sets the interrupt priority of <IL20>.	11:	Level 3 (higher priority)

Interrupt priority change control register 6

ILPRS6		7	6	5	4	3	2	1	0
(0x0FF5)	Bit Symbol	IL27P		IL26P		IL25P		IL24P	
	Read/Write	R/W		R/W		R/W		R/W	
	After reset	0	0	0	0	0	0	0	0

IL27P	Sets the interrupt priority of <IL27>.	00:	Level 0 (lower priority)
IL26P	Sets the interrupt priority of <IL26>.	01:	Level 1
IL25P	Sets the interrupt priority of <IL25>.	10:	Level 2
IL24P	Sets the interrupt priority of <IL24>.	11:	Level 3 (higher priority)

3.8. Interrupt Sequence

An interrupt request is held, until interrupt is accepted or interrupt latch is cleared to “0” by a reset or an instruction. Interrupt acceptance sequence requires 8-machine cycles after the completion of the current instruction. The interrupt service task terminates upon execution of an interrupt return instruction [RETI] (for maskable interrupts) or [RETN] (for non-maskable interrupts).

3.8.1. Initial Setting

Using an interrupt requires specifying an SP (stack pointer) for it in advance. The SP is a 16-bit register pointing at the start address of a stack. The SP is post-decremented when a subroutine call or a push instruction is executed or when an interrupt request is accepted. It is pre-incremented when a return or pop instruction is executed. Therefore, the stack becomes deeper toward lower stack location addresses. Be sure to reserve a stack area having an appropriate size based on the SP setting.

The SP is initialized to "0x00FF" after a reset. When the SP must be changed, do so right after a reset or while the interrupt master enable flag (<IMF>) is "0".

Example :The stack pointer (SP) setting

```
LD      SP, 0x023F      ; SP ← "0x023F"
LD      SP, SP+0x04     ; SP ← SP + "0x04"
ADD     SP, 0x0010      ; SP ← SP + "0x0010"
```

3.8.2. Interrupt Acceptance Processing

The below processing are automatically executed in the interrupt acceptance processing.

(1) The interrupt master enable flag (<IMF>) is cleared to "0" in order to disable the acceptance of any following interrupt.

(2) The interrupt latch (<ILx>) for the interrupt source accepted is cleared to "0".

(3) The contents of the program counter (PC) and the program status word (PSW), including the interrupt master enable flag (<IMF>) and the register bank select (RBS), are saved (Pushed) on the stack in sequence of PSW, PCH, PCL. Meanwhile, the stack pointer (SP) is decremented by 3.

(4) The entry address (Interrupt vector) of the corresponding interrupt service program, loaded on the vector table, is transferred to the program counter.

(5) The instruction stored at the entry address of the interrupt service program is executed.

Example: Correspondence between vector table address for INTTBT and the entry address of the interrupt service program

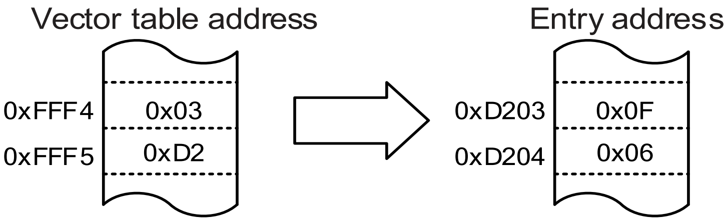


Figure 3-2 Vector table address and Entry address

A maskable interrupt is not accepted until the <IMF> is set to "1" when the maskable interrupt is requested in the interrupt service routine.

In order to utilize nested interrupt service, the <IMF> must be set to "1" in the interrupt service program.

In this case, acceptable interrupt sources are selectively enabled by the individual interrupt enable flags (<EFx>).

To avoid overloaded nesting, clear <EFx> whose interrupt is currently serviced, before setting <IMF> to "1".

As for non-maskable interrupt, keep interrupt service shorter compared with length between interrupt requests.

3.8.3. Saving/restoring General-purpose Registers

During interrupt acceptance processing, the program counter (PC) and the program status word (PSW, includes <IMF> and RBS) are automatically saved to the stack, but the general-purpose registers are not. These registers must be saved by software if necessary. When multiple interrupt services are nested, it is also necessary to avoid using the same RAM area for saving registers.

The following methods are used to save/restore the general-purpose registers.

3.8.3.1. Saving/restoring General-purpose Register by Using PUSH or POP Instructions

To save only a specific register with nested interrupts of the same sources, the general-purpose registers are saved or restored by PUSH or POP instructions.

Example :Saving/restoring general-purpose registers by using PUSH or POP instructions

```
             ; #### Interrupt service routine ####
PINTxx:      PUSH  WA              ; Save WA register to the stack
                   :               ; Interrupt service processing
             POP   WA              ; Restore WA register from the stack
             RETI                  ; Return to the main task
```

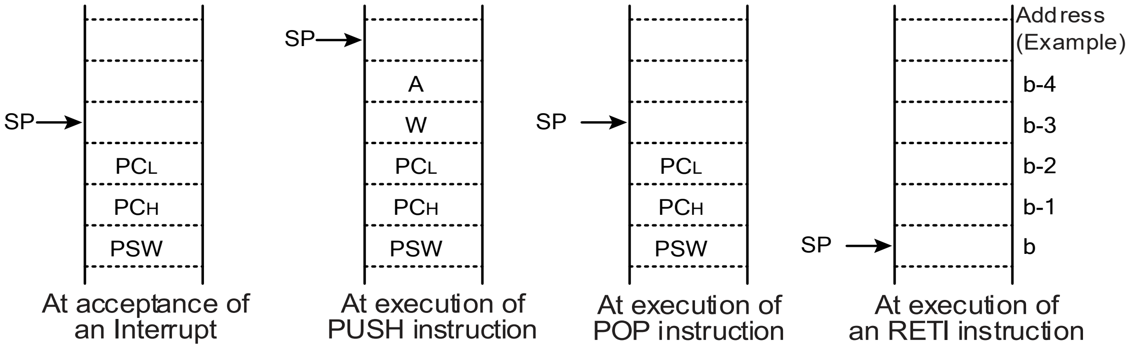

SP
PCL
PCH
PSW
At acceptance of an Interrupt
SP
A
W
PCL
PCH
PSW
At execution of PUSH instruction
SP
PCL
PCH
PSW
At execution of POP instruction
Address (Example)
b-4
b-3
b-2
b-1
b
SP
At execution of an RETI instruction

Figure 3-3 Saving/restoring General-purpose Registers by Using PUSH or POP Instructions

3.8.3.2. Saving/restoring General-purpose Register by Using Data Transfer Instructions

To save only a specific register without nested interrupts, the general-purpose registers are saved and restored by the data transfer instructions to or from the RAM.

Example: Saving/restoring only a specific register by using data transfer instructions to or from the RAM

```
                ; #### Interrupt service routine ####
PINTxx:         LD      (GSAVA), A      ; Save A register
                        :               ; Interrupt service processing
                LD      A, (GSAVA)      ; Restore A register
                RETI                    ; Return to the main task
```

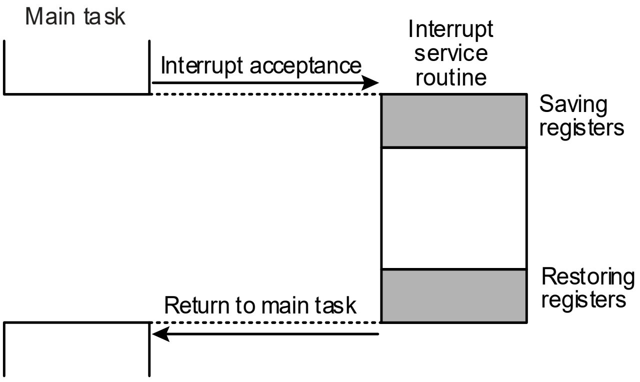


Figure 3-4 Saving/restoring General-purpose Register by Using Data Transfer Instructions

3.8.3.3. Saving/restoring General-purpose Registers by Using a Register Bank

In non-multiple interrupt handling, the register bank function can be used to save/restore the general-purpose registers at a time. The register bank function saves (switches) the general-purpose registers by executing a register bank manipulation instruction (such as LD RBS,1) at the beginning of an interrupt service task. It is unnecessary to re-execute the register bank manipulation instruction at the end of the interrupt service task because executing the RETI instruction makes a return automatically to the register bank that was being used by the main task according to the content of the PSW.

Note: Two register banks (BANK0 and BANK1) are available. Each bank consists of 8-bit general-purpose registers (W, A, B, C, D, E, H, and L) and 16-bit general-purpose registers (IX and IY).

Example: Saving/restoring the general-purpose registers by using a register bank
(with the main task using the register bank BANK0)

```
          ; #### Interrupt service routine
PINTxx:   LD      RBS, 1          ; Switches the register bank to BANK1
                  :               ; Interrupt service processing
          RETI                    ; Return to the main task (Makes a return automatically to BANK0
                                  : when the PSW is restored by RETI instruction)
```

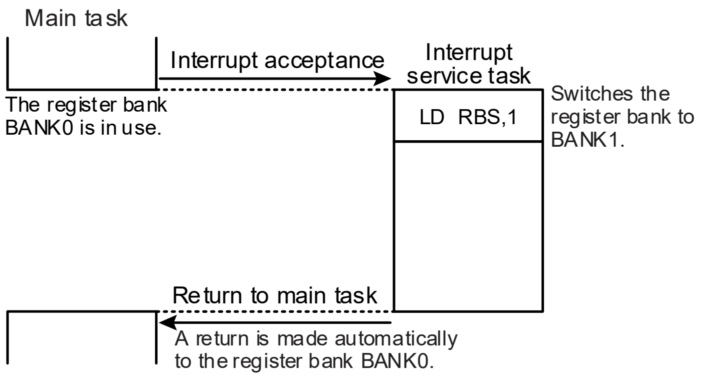

Main task
Interrupt acceptance
Interrupt service task
The register bank BANK0 is in use.
LD RBS,1
Switches the register bank to BANK1.
Return to main task
A return is made automatically to the register bank BANK0.

Figure 3-5 Saving/restoring General-purpose Registers by Using a Register Bank

3.8.4. Interrupt return

Interrupt return instructions [RETI]/[RETN] perform as follows.

[RETI]/[RETN] Interrupt Return
1. Program counter (PC) and program status word (including <IMF> and RBS) are restored from the stack.
2. Stack pointer (SP) is incremented by 3.

3.9. Software Interrupt (INTSWI)

Executing the SWI instruction generates a software interrupt and immediately starts interrupt processing (INTSWI is the top-priority interrupt).

Use the SWI instruction only for address error detection or for debugging described below.

3.9.1. Address error detection

"0xFF" is read when some cause such as noise the CPU attempts to fetch an instruction from a non-existent memory address. Code "0xFF" is an SWI instruction, so a software interrupt is generated and an address error is detected. The address error detection range can be further expanded by writing "0xFF" to unused areas in the program memory.

When a software interrupt is generated, it is recommended to generate system clock reset as the program below.

Example: The TMP89FS60B/62B/63B restart by generating system clock reset after the address error is detected by the SWI instruction.

```
INTSWI:     LD          (SYSCR2), 0x10     ; The TMP89FS60B/62B/63B restart by the system clock reset
            RETN                           ;

vector section romdata abs = 0xFFFC
            DW          INTSWI
```

3.9.2. Debugging

Debugging efficiency can be increased by placing the SWI instruction at the software break point setting address.

3.10. Undefined Instruction Interrupt (INTUNDEF)

When the CPU tries to fetch and execute an instruction that is not defined, INTUNDEF is generated and starts the interrupt processing. INTUNDEF is accepted during not only maskable interrupt but another non-maskable interrupt are in process. The current process is discontinued and the INTUNDEF interrupt process starts soon after it is requested.

Note: When the undefined instruction interrupt (INTUNDEF) occurs, the contents of the interrupt vector as same as the software interrupt's one is set to the PC.

4. External Interrupt Control Circuit

The external interrupt detects the change of the input signal and generates an interrupt request. Noise can be removed by the built-in digital noise canceller.

4.1. External Interrupt Control Circuits for Each Product

The external interrupt control circuits for each product are shown at Table 4-1.

The corresponding bit in the low power consumption register (POFFCR3) for the not available external interrupt control circuit should be cleared to “0”.

Table 4-1 External Interrupt Control Circuits for Each Product

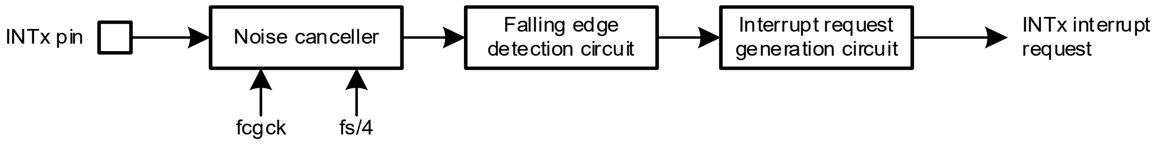
External interrupt control circuit	TMP89FS60B	TMP89FS62B	TMP89FS63B
External interrupt 0	A	NA	NA
External interrupt 1	A	NA	NA
External interrupt 2	A	A	A
External interrupt 3	A	A	A
External interrupt 4	A	A	A
External interrupt 5	A	A	A

Note: A: Available, NA: Not available

4.2. Configuration

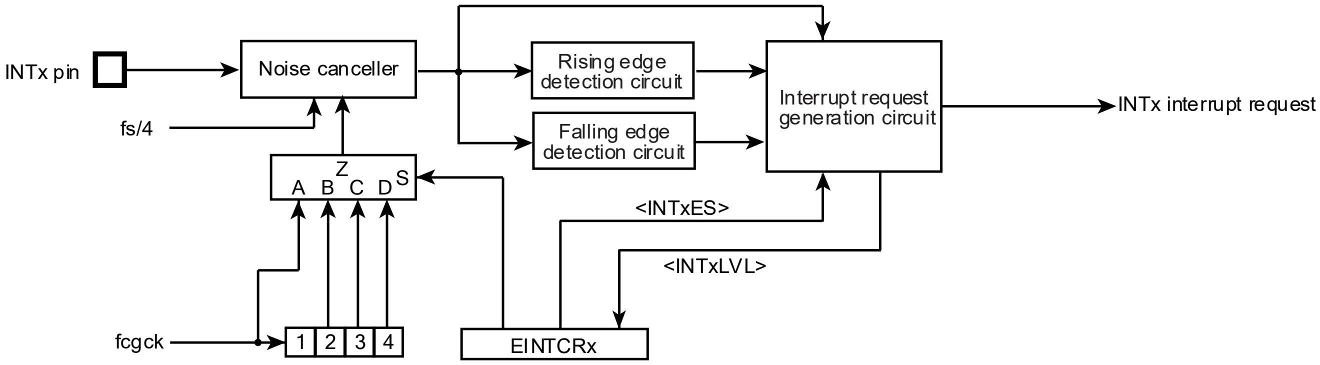
The external interrupt control circuit consists of a noise canceller, an edge detection circuit, a level detection circuit and an interrupt request generation circuit.

The externally input signals are input to the rising edge or falling edge or level detection circuit for each external interrupt, after noise is removed by the noise canceller.


INTx pin
Noise canceller
fcgck
fs/4
Falling edge detection circuit
Interrupt request generation circuit
INTx interrupt request

Note: x = 5 and 0

Figure 4-1 External Interrupts 0/5


INTx pin
Noise canceller
fs/4
Rising edge detection circuit
Falling edge detection circuit
Interrupt request generation circuit
INTx interrupt request
Z
A B C D S
<INTxES>
<INTxLVL>
fcgck
1 2 3 4
EINTCRx

Note: x = 3 to 1

Figure 4-2 External Interrupts 1/2/3

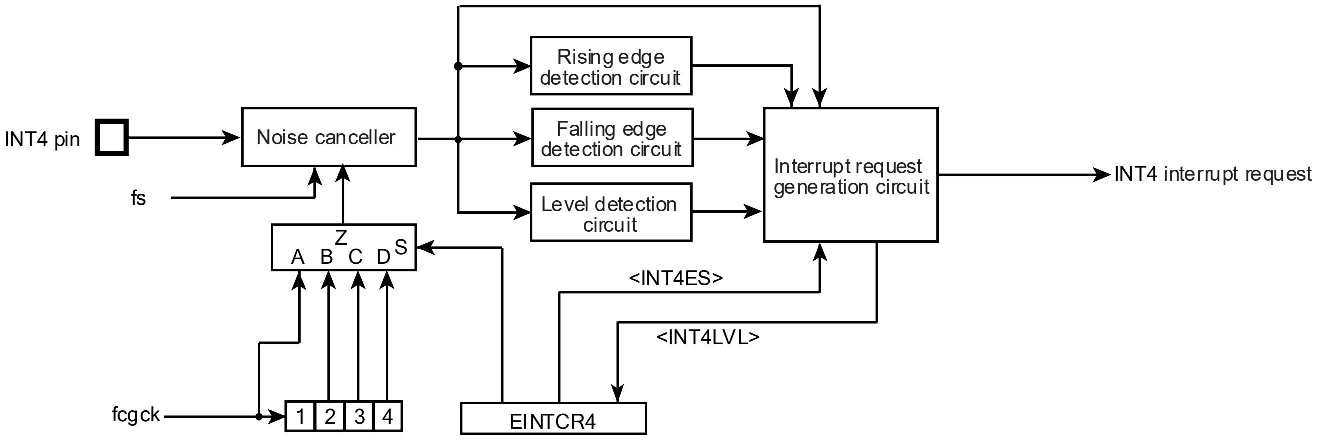


Figure 4-3 External Interrupt 4

4.3. Control

External interrupts are controlled by the following registers:

Low power consumption register 3

POFFCR3		7	6	5	4	3	2	1	0
(0x0F77)	Bit Symbol	-	-	INT5EN	INT4EN	INT3EN	INT2EN	INT1EN	INT0EN
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

INT5EN	INT5 control	0: 1:	Disable Enable
INT4EN	INT4 control	0: 1:	Disable Enable
INT3EN	INT3 control	0: 1:	Disable Enable
INT2EN	INT2 control	0: 1:	Disable Enable
INT1EN (Note 4)	INT1 control	0: 1:	Disable Enable
INT0EN (Note 4)	INT0 control	0: 1:	Disable Enable

Note 1: Clearing <INTxEN> (x = 5 to 0) to "0" stops the clock supply to the external interrupts. This invalidates the data written in the control register for each external interrupt. When using the external interrupts, set <INTxEN> to "1" and then write data into the control register for each external interrupt.

Note 2: Interrupt requests may be generated when <INTxEN> is changed. Before changing <INTxEN>, clear the corresponding interrupt enable flag to "0" to disable the generation of interrupt. Wait for 2 / fcgck + 3 / fspl [s] in the NORMAL1/2 or IDLE1/2 mode and clears the corresponding interrupt latch to “0”. Wait for 12 / fs [s] in the SLOW1/2 or SLEEP1 mode, clears the corresponding one.

Note 3: Bits 7 and 6 of POFFCR3 are read as the value written. They should be clear to “0”.

Note 4: <INT1EN> and <INT0EN> are “Reserved (this bit symbol must be cleared to “0”.)” for the TMP89FS62B and TMP89FS63B.

External interrupt control register 1

EINTCR1		7	6	5	4	3	2	1	0
(0x0FD8)	Bit Symbol	-	-	-	INT1LVL	INT1ES		INT1NC	
	Read/Write	R	R	R	R	R/W		R/W	
	After reset	0	0	0	0	0	0	0	0

INT1LVL	Noise canceller pass signal level when the interrupt request is generated for external interrupt 1	0:	Initial state or signal level “Low”		
		1:	Signal level “High”		
INT1ES	Selects the interrupt request generating condition for external interrupt 1	00:	An interrupt request is generated at the rising edge of the noise canceller pass signal		
		01:	An interrupt request is generated at the falling edge of the noise canceller pass signal		
		10:	An interrupt request is generated at both edges of the noise canceller pass signal		
		11:	Reserved		
INT1NC	Sets the noise canceller sampling interval for external interrupt 1	NORMAL1/2 or IDLE1/2		SLOW1/2 or SLEEP1	
		00:	fcgck [Hz]	00:	fs / 4 [Hz]
		01:	fcgck / 2^2 [Hz]	01:	fs / 4 [Hz]
		10:	fcgck / 2^3 [Hz]	10:	fs / 4 [Hz]
		11:	fcgck / 2^4 [Hz]	11:	fs / 4 [Hz]

Note 1: fcgck: Gear clock [Hz], fs: Low-frequency clock [Hz]

Note 2: Interrupt requests may be generated during transition of the operation mode. Before changing the operation mode, clear the corresponding interrupt enable flag to "0" to disable the generation of interrupt. When the operation mode is changed from NORMAL1/2 or IDLE1/2 to SLOW1/2 or SLEEP1, wait 12 / fs [s] after the operation mode is changed and clear the corresponding interrupt latch.
And when the operation mode is changed from SLOW1/2 or SLEEP1 to NORMAL1/2 or IDLE1/2, wait 2 / fcgck + 3 / fspl [s] after the operation mode is changed and clear the corresponding interrupt latch.

Note 3: Interrupt requests may be generated when EINTCR1 is changed. Before changing EINTCR1 clear the corresponding interrupt enable flag to "0" to disable the generation of interrupt. Wait for 2 / fcgck + 3 / fspl [s] in the NORMAL1/2 or IDLE1/2 mode and clears the corresponding interrupt latch to “0”. Wait for 12 / fs [s] in the SLOW1/2 or SLEEP1 mode, clears the corresponding one.

Note 4: The content of EINTCR1<INT1LVL> is updated each time an interrupt request is generated.

Note 5: When a read instruction is executed on EINTCR1, bits 7 to 5 are read as "0".

External interrupt control register 2

EINTCR2		7	6	5	4	3	2	1	0
(0x0FD9)	Bit Symbol	-	-	-	INT2LVL	INT2ES		INT2NC	
	Read/Write	R	R	R	R	R/W		R/W	
	After reset	0	0	0	0	0	0	0	0

INT2LVL	Noise canceller pass signal level when the interrupt request is generated for external interrupt 2	0: 1:	Initial state or signal level “Low” Signal level “High”		
INT2ES	Selects the interrupt request generating condition for external interrupt 2	00: 01: 10: 11:	An interrupt request is generated at the rising edge of the noise canceller pass signal An interrupt request is generated at the falling edge of the noise canceller pass signal An interrupt request is generated at both edges of the noise canceller pass signal Reserved		
INT2NC	Sets the noise canceller sampling interval for external interrupt 2	NORMAL1/2 or IDLE1/2		SLOW1/2 or SLEEP1	
		00: 01: 10: 11:	fcgck [Hz] fcgck / 2^2 [Hz] fcgck / 2^3 [Hz] fcgck / 2^4 [Hz]	00: 01: 10: 11:	fs / 4 [Hz] fs / 4 [Hz] fs / 4 [Hz] fs / 4 [Hz]

Note 1: fcgck: Gear clock [Hz], fs: Low-frequency clock [Hz]

Note 2: Interrupt requests may be generated during transition of the operation mode. Before changing the operation mode, clear the corresponding interrupt enable flag to "0" to disable the generation of interrupt. When the operation mode is changed from NORMAL1/2 or IDLE1/2 to SLOW1/2 or SLEEP1, wait 12 / fs [s] after the operation mode is changed and clear the corresponding interrupt latch.
And when the operation mode is changed from SLOW1/2 or SLEEP1 to NORMAL1/2 or IDLE1/2, wait 2 / fcgck + 3 / fspl [s] after the operation mode is changed and clear the corresponding interrupt latch.

Note 3: Interrupt requests may be generated when EINTCR2 is changed. Before changing EINTCR2 clear the corresponding interrupt enable flag to "0" to disable the generation of interrupt. Wait for 2 / fcgck + 3 / fspl [s] in the NORMAL1/2 or IDLE1/2 mode and clears the corresponding interrupt latch to “0”. Wait for 12 / fs [s] in the SLOW1/2 or SLEEP1 mode, clears the corresponding one.

Note 4: The content of EINTCR2<INT2LVL> is updated each time an interrupt request is generated.

Note 5: When a read instruction is executed on EINTCR2, bits 7 to 5 are read as "0".

External interrupt control register 3

EINTCR3		7	6	5	4	3	2	1	0
(0x0FDA)	Bit Symbol	-	-	-	INT3LVL	INT3ES		INT3NC	
	Read/Write	R	R	R	R	R/W		R/W	
	After reset	0	0	0	0	0	0	0	0

INT3LVL	Noise canceller pass signal level when the interrupt request is generated for external interrupt 3	0:	Initial state or signal level “Low”		
		1:	Signal level “High”		
INT3ES	Selects the interrupt request generating condition for external interrupt 3	00:	An interrupt request is generated at the rising edge of the noise canceller pass signal		
		01:	An interrupt request is generated at the falling edge of the noise canceller pass signal		
		10:	An interrupt request is generated at both edges of the noise canceller pass signal		
		11:	Reserved		
INT3NC	Sets the noise canceller sampling interval for external interrupt 3	NORMAL1/2 or IDLE1/2		SLOW1/2 or SLEEP1	
		00:	fcgck [Hz]	00:	fs / 4 [Hz]
		01:	fcgck / 2^2 [Hz]	01:	fs / 4 [Hz]
		10:	fcgck / 2^3 [Hz]	10:	fs / 4 [Hz]
		11:	fcgck / 2^4 [Hz]	11:	fs / 4 [Hz]

Note 1: fcgck: Gear clock [Hz], fs: Low-frequency clock [Hz]

Note 2: Interrupt requests may be generated during transition of the operation mode. Before changing the operation mode, clear the corresponding interrupt enable flag to "0" to disable the generation of interrupt. When the operation mode is changed from NORMAL1/2 or IDLE1/2 to SLOW1/2 or SLEEP1, wait 12 / fs [s] after the operation mode is changed and clear the corresponding interrupt latch.
And when the operation mode is changed from SLOW1/2 or SLEEP1 to NORMAL1/2 or IDLE1/2, wait 2 / fcgck + 3 / fspl [s] after the operation mode is changed and clear the corresponding interrupt latch.

Note 3: Interrupt requests may be generated when EINTCR3 is changed. Before changing EINTCR3 clear the corresponding interrupt enable flag to "0" to disable the generation of interrupt. Wait for 2 / fcgck + 3 / fspl [s] in the NORMAL1/2 or IDLE1/2 mode and clears the corresponding interrupt latch to “0”. Wait for 12 / fs [s] in the SLOW1/2 or SLEEP1 mode, clears the corresponding one.

Note 4: The content of EINTCR3<INT3LVL> is updated each time an interrupt request is generated.

Note 5: When a read instruction is executed on EINTCR3, bits 7 to 5 are read as "0".

External interrupt control register 4

EINTCR4		7	6	5	4	3	2	1	0
(0x0FDB)	Bit Symbol	-	-	-	INT4LVL	INT4ES		INT4NC	
	Read/Write	R	R	R	R	R/W		R/W	
	After reset	0	0	0	0	0	0	0	0

INT4LVL	Noise canceller pass signal level when the interrupt request is generated for external interrupt 4	0: 1:	Initial state or signal level “Low” Signal level “High”		
INT4ES	Selects the interrupt request generating condition for external interrupt 4	00: 01: 10: 11:	An interrupt request is generated at the rising edge of the noise canceller pass signal An interrupt request is generated at the falling edge of the noise canceller pass signal An interrupt request is generated at both edges of the noise canceller pass signal An interrupt request is generated at “High” of the noise canceller pass signal		
INT4NC	Sets the noise canceller sampling interval for external interrupt 4	NORMAL1/2 or IDLE1/2		SLOW1/2 or SLEEP1	
		00: 01: 10: 11:	fcgck [Hz] fcgck / 2^2 [Hz] fcgck / 2^3 [Hz] fcgck / 2^4 [Hz]	00: 01: 10: 11:	fs / 4 [Hz] fs / 4 [Hz] fs / 4 [Hz] fs / 4 [Hz]

Note 1: fcgck: Gear clock [Hz], fs: Low-frequency clock [Hz]

Note 2: Interrupt requests may be generated during transition of the operation mode. Before changing the operation mode, clear the corresponding interrupt enable flag to "0" to disable the generation of interrupt. When the operation mode is changed from NORMAL1/2 or IDLE1/2 to SLOW1/2 or SLEEP1, wait 12 / fs [s] after the operation mode is changed and clear the corresponding interrupt latch.
And when the operation mode is changed from SLOW1/2 or SLEEP1 to NORMAL1/2 or IDLE1/2, wait 2 / fcgck + 3 / fspl [s] after the operation mode is changed and clear the corresponding interrupt latch.

Note 3: Interrupt requests may be generated when EINTCR4 is changed. Before changing EINTCR4 clear the corresponding interrupt enable flag to "0" to disable the generation of interrupt. Wait for 2 / fcgck+3 / fspl [s] in the NORMAL1/2 or IDLE1/2 mode and clears the corresponding interrupt latch to “0”. Wait for 12 / fs [s] in the SLOW1/2 or SLEEP1 mode, clears the corresponding one.

Note 4: The content of EINTCR4<INT4LVL> is updated each time an interrupt request is generated.

Note 5: When a read instruction is executed on EINTCR4, bits 7 to 5 are read as "0".

4.4. Low Power Consumption Control

The external interrupts have a function that saves power by using the low power consumption register (POFFCR3) when they are not used.

Setting POFFCR3<INTxEN> to "0" stops (disables) the basic clock for the external interrupts and helps save power. Note that this makes the external interrupts unavailable. Setting POFFCR3<INTxEN> to "1" supplies (enables) the basic clock for the external interrupts and makes the external interrupts available.

After reset, POFFCR3<INTxEN> is initialized to "0" and the external interrupts become unavailable. When using the external interrupt function for the first time, be sure to set POFFCR3<INTxEN> to "1" in the initial setting of software (before operating the external interrupt control registers).

Note: Interrupt requests may be generated when <INTxEN> is changed. Before changing <INTxEN>, clear the corresponding interrupt enable flag to "0" to disable the generation of interrupt. Wait for 2 / fcgck + 3 / fspl [s] in the NORMAL1/2 or IDLE1/2 mode and clears the corresponding interrupt latch to “0”. Wait for 12 / fs [s] in the SLOW1/2 or SLEEP1 mode, clears the corresponding one.

4.5. Function

The condition for generating interrupt requests and the noise cancel time can be set for external interrupts 1 to 4.

The condition for generating interrupt requests and the noise cancel time are fixed for external interrupts 0 and 5.

Table 4-2 Enable Condition and Interrupt Request Generated condition for each external Interrupt source

Source	Pin name	Enable condition	Interrupt request generated condition
INT0	$\overline{\text{INT0}}$	<IMF> AND <EF16> = 1	Falling edge
INT1	INT1	<IMF> AND <EF17> = 1	Falling edge Rising edge Both edges
INT2	INT2	<IMF> AND <EF18> = 1	Falling edge Rising edge Both edges
INT3	INT3	<IMF> AND <EF19> = 1	Falling edge Rising edge Both edges
INT4	INT4	<IMF> AND <EF20> = 1	Falling edge Rising edge Both edges “High” level
INT5	$\overline{\text{INT5}}$	<IMF> AND <EF8> = 1	Falling edge

Table 4-3 External interrupt pin input signal width (w) and noise cancel time

Source	Pin name	External interrupt pin input signal width (w) and noise cancel time	
		NORMAL1/2 or IDLE1/2	SLOW1/2 or SLEEP1
INT0	$\overline{\text{INT0}}$	w < 1 / fcgck : Noise 1 / fcgck ≤ w < 2 / fcgck : Either noise or signal 2 / fcgck ≤ w : Signal	w < 4 / fs: Noise 4 / fs ≤ w < 8 / fs: Either noise or signal 8 / fs ≤ w : Signal
INT1	INT1	w < 2 / fspl : Noise 2 / fspl ≤ w < 3 / fspl + 1 / fcgck: Either noise or signal 3 / fspl + 1 / fcgck ≤ w : Signal	w < 4 / fs: Noise 4 / fs ≤ w < 8 / fs: Either noise or signal 8 / fs ≤ w : Signal
INT2	INT2	w < 2 / fspl : Noise 2 / fspl ≤ w < 3 / fspl + 1 / fcgck: Either noise or signal 3 / fspl + 1 / fcgck ≤ w : Signal	w < 4 / fs: Noise 4 / fs ≤ w < 8 / fs: Either noise or signal 8 / fs ≤ w : Signal
INT3	INT3	w < 2 / fspl : Noise 2 / fspl ≤ w < 3 / fspl + 1 / fcgck: Either noise or signal 3 / fspl + 1 / fcgck ≤ w : Signal	w < 4 / fs: Noise 4 / fs ≤ w < 8 / fs: Either noise or signal 8 / fs ≤ w : Signal
INT4	INT4	w < 2 / fspl : Noise 2 / fspl ≤ w < 3 / fspl + 1 / fcgck: Either noise or signal 3 / fspl + 1 / fcgck ≤ w : Signal	w < 4 / fs: Noise 4 / fs ≤ w < 8 / fs: Either noise or signal 8 / fs ≤ w : Signal
INT5	$\overline{\text{INT5}}$	w < 1 / fcgck : Noise 1 / fcgck ≤ w < 2 / fcgck : Either noise or signal 2 / fcgck ≤ w : Signal	w < 4 / fs: Noise 4 / fs ≤ w < 8 / fs: Either noise or signal 8 / fs ≤ w : Signal

Note: fcgck: gear clock [Hz], fs: low-frequency clock [Hz], fspl: Sampling frequency [Hz]

4.5.1. External Interrupt 0

The external interrupt 0 detects the falling edge of the $\overline{\text{INT0}}$ pin and generates interrupt requests.

In NORMAL1/2 or IDLE1/2 mode, pulses of less than 1 / fcgck are removed as noise and pulses of 2 / fcgck or more are recognized as signals.

In SLOW/SLEEP mode, pulses of less than 4 / fs are removed as noise and pulses of 8 / fs or more are recognized as signals.

4.5.2. External Interrupts 1/2/3

External interrupts 1/2/3 detect the falling edge, the rising edge or both edges of the INT1, INT2 and INT3 pins and generate interrupt requests.

4.5.2.1. Interrupt Request Generating Condition Detection Function

Select interrupt request generating conditions at EINTCRx<INTxES> for external interrupts 1/2/3.

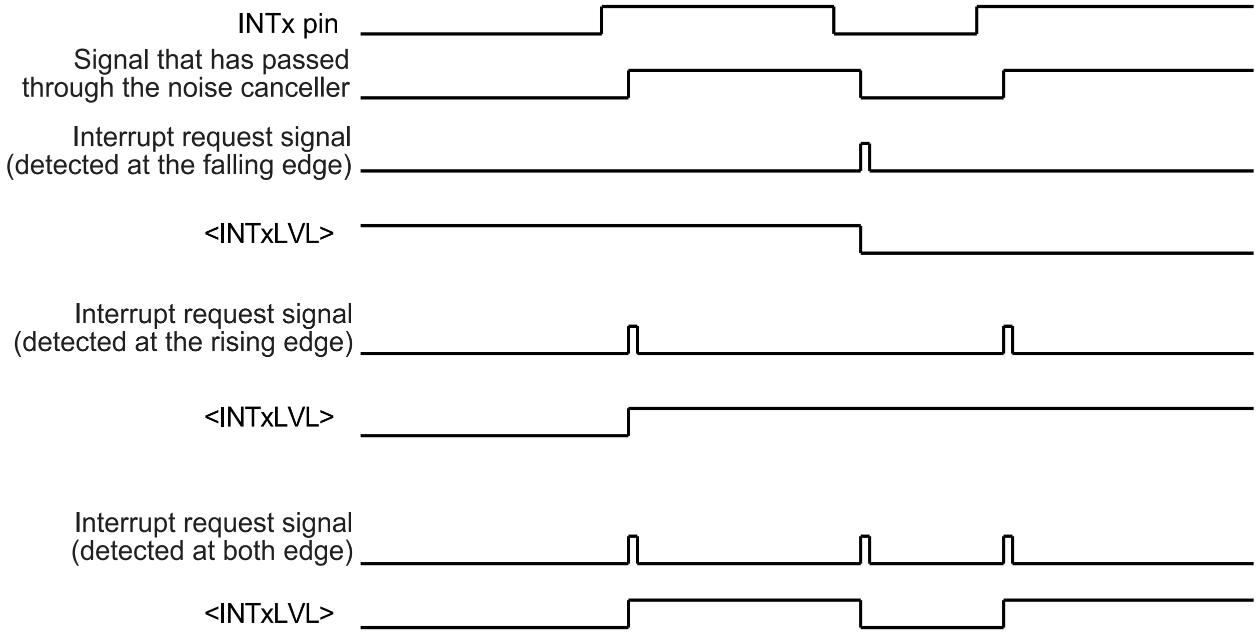
Table 4-4 Selection of Interrupt Request Generation Condition

EINTCRx<INTxES>	Interrupt request generating condition
00	Rising edge
01	Falling edge
10	Both edges
11	Reserved

Note: x = 3 to 1

4.5.2.2. Noise Canceller Passed Signal Monitoring Function when Interrupt Requests Are Generated

The level of a signal that has passed through the noise canceller when an interrupt request is generated can be read by using EINTCRx<INTxLVL>. When both edges are selected as detection edges, the edge where an interrupt is generated can be detected by reading EINTCRx<INTxLVL>.


INTx pin
Signal that has passed through the noise canceller
Interrupt request signal (detected at the falling edge)
<INTxLVL>
Interrupt request signal (detected at the rising edge)
<INTxLVL>
Interrupt request signal (detected at both edge)
<INTxLVL>

Note: x=1 to 3

Figure 4-4 Interrupt Request Generation and EINTCRx<INTxLVL>

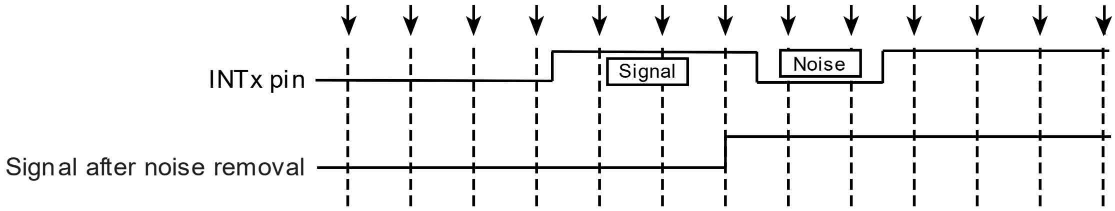
4.5.2.3. Noise Cancel Time Selection Function

In NORMAL1/2 or IDLE1/2 mode, a signal that has been sampled by fcgck is sampled at the sampling interval selected at EINTCRx<INTxNC>. When the same level is detected three consecutive times, the signal is recognized as a signal. When not, the signal is removed as noise.

For details of the signal width passed as a signal, and the signal width removed as a noise, refer to Table 4-3.

Table 4-5 Noise Canceller Sampling Clock

EINTCRx<INTxNC>	Sampling clock
00	fcgck
01	fcgck / 2^2
10	fcgck / 2^3
11	fcgck / 2^4



Note: x = 3 to 1

Figure 4-5 Noise Cancel Operation

In SLOW1/2 or SLEEP1 mode, a signal is sampled by a quarter of the low-frequency clock. When the same level is detected twice consecutively, the signal is recognized as a signal.

For details of the signal width passed as a signal, and the signal width removed as a noise, refer to Table 4-3.

In IDLE0, SLEEP0 or STOP mode, the noise canceller sampling operation is stopped and an external interrupts are unavailable. When the operation mode returns to NORMAL1/2, IDLE1/2, SLOW1/2 or SLEEP1 mode, sampling operation restarts.

Note 1: When noise is input consecutively during sampling of external interrupt pins, the noise cancel function does not work properly. Set EINTCRx<INTxNC> according to the cycle of externally input noise.

Note 2: When an external interrupt pin is used as an output port, the input signal from the port is fixed to “Low” when switching to the output mode, and thus an interrupt request occurs. To use the external interrupt pin as an output port, clear the corresponding interrupt enable flag to "0" to disable the generation of interrupt request.

Note 3: Interrupt requests may be generated during transition of the operation mode. Before changing the operation mode, clear the corresponding interrupt enable register to "0" to disable the generation of interrupt. When the operation mode is changed from NORMAL1/2 or IDLE1/2 to SLOW1/2 or SLEEP1, wait 12 / fs [s] after the operation mode is changed and clear the interrupt latch. And when the operation mode is changed from SLOW1/2 or SLEEP1 to NORMAL1/2 or IDLE1/2, wait 2 / fcgck + 3 / fspl [s] after the operation mode is changed and clear the interrupt latch.

4.5.3. External Interrupt 4

External interrupt 4 detects the falling edge, the rising edge, both edges or “High” level of the INT4 pin and generates interrupt requests.

4.5.3.1. Interrupt request generating condition detection function

Select an interrupt request generating condition at EINTCR4<INT4ES> for external interrupt 4.

Table 4-6 Selection of Interrupt Request Generating Condition

EINTCR4<INT4ES>	Interrupt request generating condition
00	Rising edge
01	Falling edge
10	Both edges
11	“High” level interrupt

4.5.3.2. Noise Canceller Passed Signal Monitoring Function When Interrupt Requests are Generated

The level of a signal that has passed through the noise canceller when an interrupt request is generated can be read by using EINTCR4<INT4LVL>. When both edges are selected as detection edges, the edge where an interrupt is generated can be detected by reading EINTCR4<INT4LVL>.

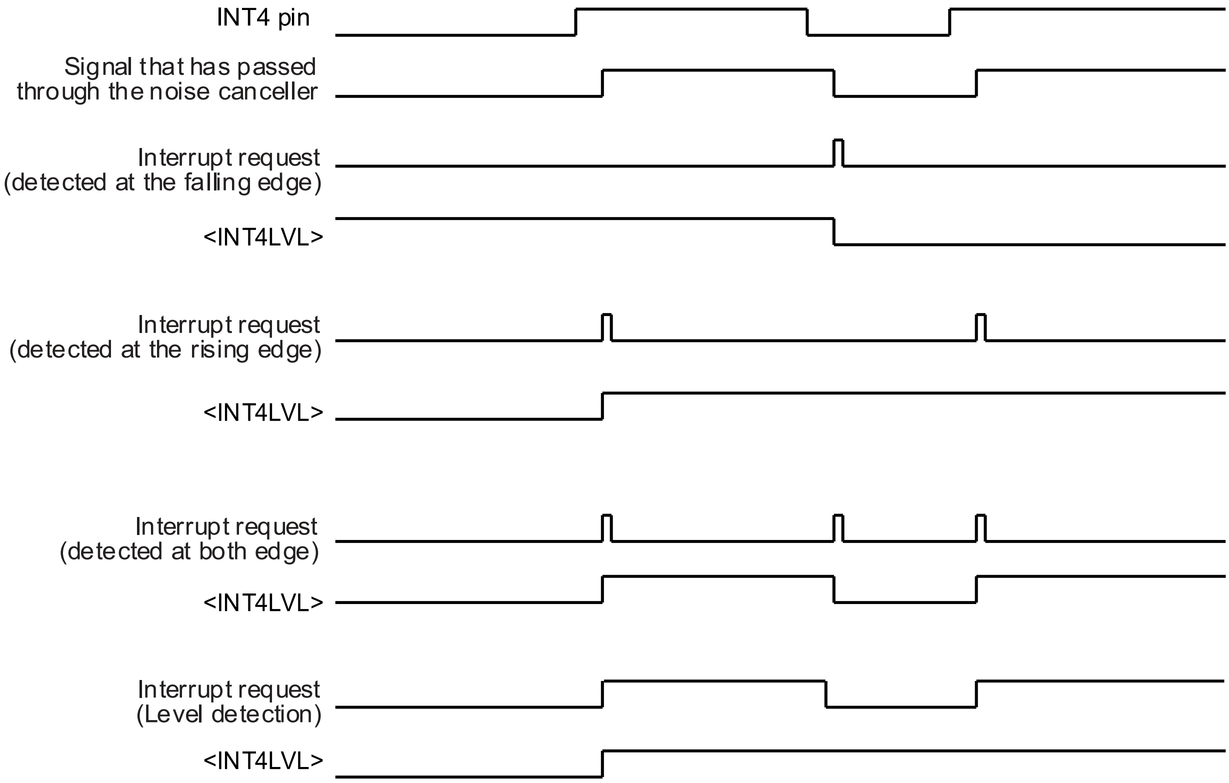

INT4 pin
Signal that has passed through the noise canceller
Interrupt request (detected at the falling edge)
<INT4LVL>
Interrupt request (detected at the rising edge)
<INT4LVL>
Interrupt request (detected at both edge)
<INT4LVL>
Interrupt request (Level detection)
<INT4LVL>

Figure 4-6 Interrupt Request Generation and EINTCR4<INT4LVL>

4.5.3.3. Noise Cancel Time Selection Function

In NORMAL1/2 or IDLE1/2 mode, a signal that has been sampled by fcgck is sampled at the sampling interval selected at EINTCR4<INT4NC>. When the same level is detected three consecutive times, the signal is recognized as a signal. When not, the signal is removed as noise.

For details of the signal width passed as a signal, and the signal width removed as a noise, refer to Table 4-3.

Table 4-7 Noise Canceller Sampling Lock

EINTCR4<INT4NC>	Sampling clock
00	fcgck
01	fcgck / 2^2
10	fcgck / 2^3
11	fcgck / 2^4

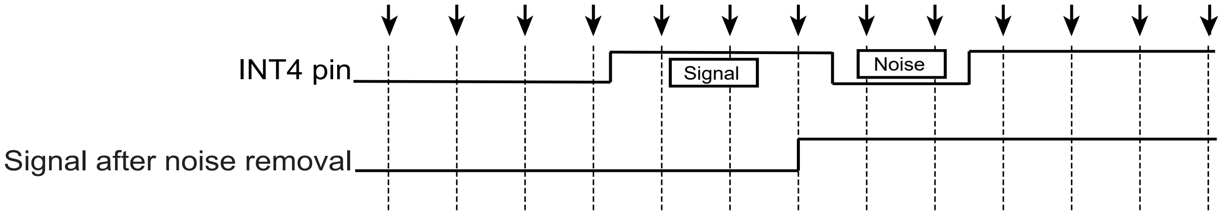


Figure 4-7 Noise Cancel Operation

In SLOW1/2 or SLEEP1 mode, a signal is sampled by the low-frequency clock divided by 4. When the same level is detected twice consecutively, the signal is recognized as a signal.

For details of the signal width passed as a signal, and the signal width removed as a noise, refer to Table 4-3.

In IDLE0, SLEEP0 or STOP mode, the noise canceller sampling operation is stopped and an external interrupts are unavailable. When operation returns to NORMAL1/2, IDLE1/2, SLOW1/2 or SLEEP1 mode, sampling operation restarts.

Note 1: When noise is input consecutively during sampling of external interrupt pins, the noise cancel function does not work properly. Set EINTCR4<INT4NC> according to the cycle of externally input noise.

Note 2: When an external interrupt pin is used as an output port, the input signal to the port is fixed to “Low” when the mode is switched to the output mode, and thus an interrupt request occurs. To use the pin as an output port, clear the corresponding interrupt enable register to "0" to disable the generation of interrupt.

Note 3: Interrupt requests may be generated during transition of the operation mode. Before changing the operation mode, clear the corresponding interrupt enable register to "0" to disable the generation of interrupt. When the operation mode is changed from NORMAL1/2 or IDLE1/2 to SLOW1/2 or SLEEP1, wait 12 / fs [s] after the operation mode is changed and clear the interrupt latch. And when the operation mode is changed from SLOW1/2 or SLEEP1 to NORMAL1/2 or IDLE1/2, wait 2 / fcgck + 3 / fspl [s] after the operation mode is changed and clear the interrupt latch.

4.5.4. External Interrupt 5

The external interrupt 5 detects the falling edge of the $\overline{\text{INT5}}$ pin and generates interrupt requests.

In NORMAL1/2 or IDLE1/2 mode, pulses of less than 1 / fcgck are removed as noise and pulses of 2 / fcgck or more are recognized as signals.

In SLOW/SLEEP mode, pulses of less than 4 / fs are removed as noise and pulses of 8 / fs or more are recognized as signals.

For details of the signal width passed as a signal, and the signal width removed as a noise, refer to Table 4-3.

5. Watchdog Timer (WDT)

The watchdog timer is a fail-safe function to detect rapidly the CPU malfunctions such as endless loops due to spurious noises or the deadlock conditions, and to return a correct operation.

The watchdog timer signals used for detecting malfunctions can be programmed as watchdog interrupt requests or watchdog timer reset signals.

Note: Care must be taken in system designing since the watchdog timer may not fulfill its functions due to disturbing noise and other effects.

5.1. Configuration

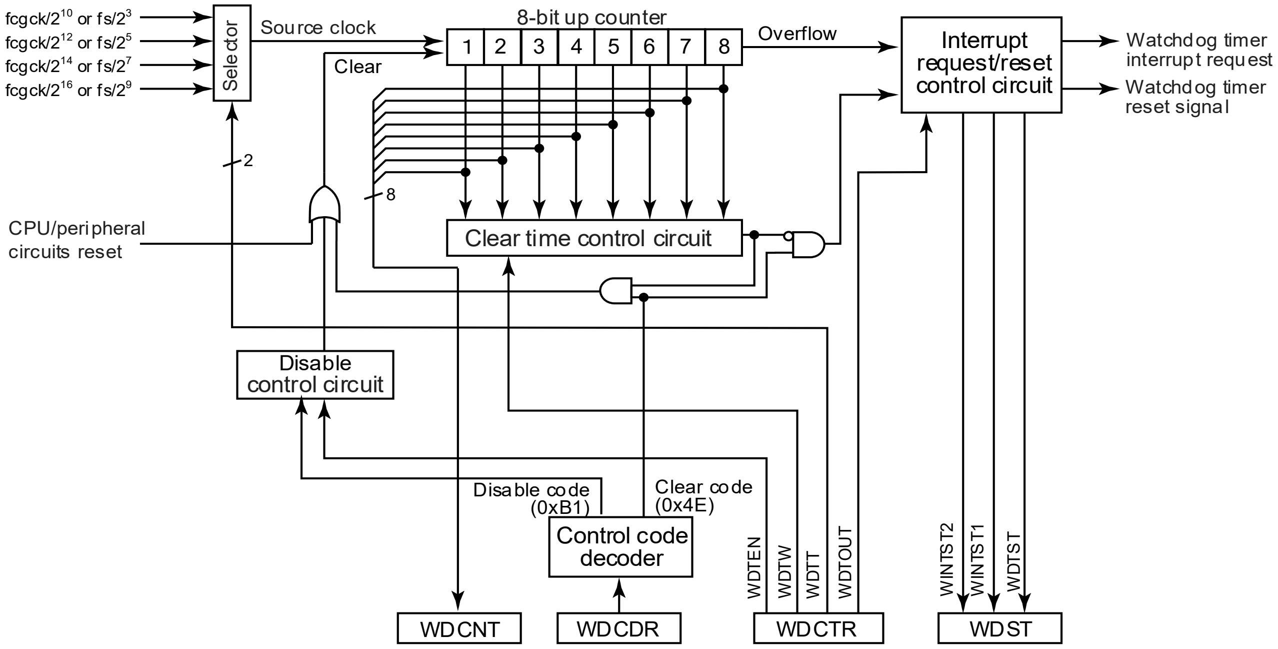


Figure 5-1 Watchdog Timer Configuration

5.2. Control

The watchdog timer is controlled by the watchdog timer control register (WDCTR), the watchdog timer control code register (WDCDR), the watchdog timer counter monitor (WDCNT) and the watchdog timer status (WDST).

The watchdog timer is enabled automatically just after the warm-up operation that follows reset is finished.

Watchdog timer control register

WDCTR		7	6	5	4	3	2	1	0
(0x0FD4)	Bit Symbol	-	-	WDTEN	WDTW		WDTT		WDTOUT
	Read/Write	R	R	R/W	R/W		R/W		R/W
	After reset	1	0	1	0	0	1	1	0

WDTEN	Enables/disables the watchdog timer operation.	0:	Disable		
		1:	Enable		
WDTW	Sets the clear timing of the 8-bit up counter.	00:	The 8-bit up counter is cleared by writing the clear code at any point within the overflow time of the 8-bit up counter.		
		01:	A watchdog timer interrupt request is generated by writing the clear code at a point within the first quarter of the overflow time of the 8-bit up counter. The 8-bit up counter is cleared by writing the clear code after the first quarter of the overflow time has elapsed.		
		10:	A watchdog timer interrupt request is generated by writing the clear code at a point within the first half of the overflow time of the 8-bit up counter. The 8-bit up counter is cleared by writing the clear code after the first half of the overflow time has elapsed.		
		11:	A watchdog timer interrupt request is generated by writing the clear code at a point within the first three quarters of the overflow time of the 8-bit up counter. The 8-bit up counter is cleared by writing the clear code after the first three quarters of the overflow time have elapsed.		
WDTT	Sets the overflow time of the 8-bit up counter.		NORMAL mode		SLOW mode
			<DV9CK> = 0	<DV9CK> = 1	
		00:	2^{18} / fcgck	2^{11} / fs	2^{11} / fs
		01:	2^{20} / fcgck	2^{13} / fs	2^{13} / fs
		10:	2^{22} / fcgck	2^{15} / fs	2^{15} / fs
		11:	2^{24} / fcgck	2^{17} / fs	2^{17} / fs
WDTOUT	Selects an overflow detection signal of the 8-bit up counter.	0:	Watchdog timer interrupt request		
		1:	Watchdog timer reset signal		

Note 1: fcgck: gear clock [Hz], fs: low-frequency clock [Hz]

Note 2: WDCTR<WDTW>, <WDTT> and <WDTOUT> cannot be changed when WDCTR<WDTEN> is "1". When WDCTR<WDTEN> is "1", clear WDCTR<WDTEN> to "0" and write the disable code (0xB1) into WDCDR to disable the watchdog timer operation. Note that WDCTR<WDTW>, <WDTT> and <WDTOUT> can be changed at the same time as setting WDCTR<WDTEN> to "1".

Note 3: Bit 7 and bit 6 of WDCTR are read as "1" and "0" respectively.

Watchdog timer control code register

WDCDR		7	6	5	4	3	2	1	0
(0x0FD5)	Bit Symbol	WDTCR2							
	Read/Write	W							
	After reset	0	0	0	0	0	0	0	0

WDTCR2	Writes watchdog timer control codes.	0x4E:	Clears the watchdog timer. (Clear code)
		0xB1:	Disables the watchdog timer operation and clears the 8-bit up counter when WDCTR<WDTEN> is "0". (Disable code)
		Others:	Invalid

Note: WDCDR is a write-only register and must not be accessed by using a read-modify-write instruction, such as a bit operation.

8-bit up counter monitor

WDCNT		7	6	5	4	3	2	1	0
(0x0FD6)	Bit Symbol	WDCNT							
	Read/Write	R							
	After reset	0	0	0	0	0	0	0	0

WDCNT	Monitors the count value of the 8-bit up counter	The count value of the 8-bit up counter is read.

Watchdog timer status

WDST		7	6	5	4	3	2	1	0
(0x0FD7)	Bit Symbol	-	-	-	-	-	WINTST2	WINTST1	WDTST
	Read/Write	R	R	R	R	R	R	R	R
	After reset	0	1	0	1	1	0	0	1

WINTST2	Watchdog timer interrupt request factor status 2	0:	No watchdog timer interrupt request has occurred.
		1:	A watchdog timer interrupt request has occurred due to the overflow of the 8-bit up counter.
WINTST1	Watchdog timer interrupt request factor status 1	0:	No watchdog timer interrupt request has occurred.
		1:	A watchdog timer interrupt request has occurred due to releasing of the 8-bit up counter outside the clear time.
WDTST	Watchdog timer operating state status	0:	Operation disabled
		1:	Operation enabled

Note 1: WDST<WINTST2> and <WINTST1> are cleared to "0" by reading WDST.

Note 2: Values after reset are read from bits 7 to 3 of WDST.

5.3. Functions

The watchdog timer can detect the CPU malfunctions and deadlock by detecting the overflow of the 8-bit up counter and detecting clearing of the 8-bit up counter outside the clear time.

The watchdog timer stoppage and other abnormalities can be detected by reading the count value of the 8-bit up counter at random times and comparing the value to the last read value.

5.3.1. Setting of Enabling/disabling the Watchdog Timer Operation

Setting WDCTR<WDTEN> to "1" enables the watchdog timer operation, and the 8-bit up counter starts counting the source clock.

WDCTR<WDTEN> is initialized to "1" after the warm-up operation that follows reset is released. This means that the watchdog timer is enabled.

To disable the watchdog timer operation, clear WDCTR<WDTEN> to "0" and write "0xB1" into WDCDR. Disabling the watchdog timer operation clears the 8-bit up counter to "0".

Note: When the overflow of the 8-bit up counter occurs at the same time as "0xB1" (disable code) is written into WDCDR with WDCTR<WDTEN> set at "1", the watchdog timer operation is disabled preferentially and the overflow detection is not executed.

To re-enable the watchdog timer operation, set WDCTR<WDTEN> to "1". There is no need to write a control code into WDCDR.

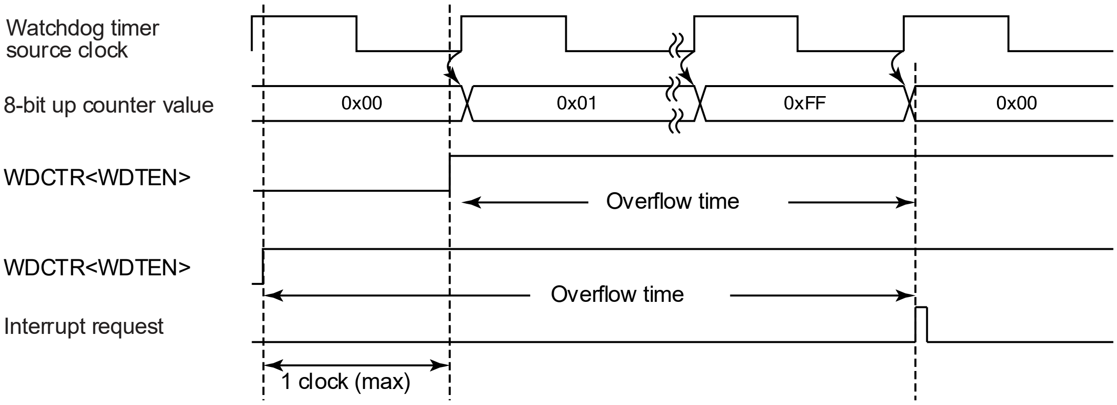


Figure 5-2 WDCTR<WDTEN> Set Timing and Overflow Time

Note: The 8-bit up counter source clock operates out of synchronization with WDCTR<WDTEN>. Therefore, the first overflow time of the 8-bit up counter after WDCTR<WDTEN> is set to "1" may get shorter by a maximum of 1 source clock. The 8-bit up counter must be cleared within the period of the overflow time minus 1 source clock cycle.

5.3.2. Setting the Clear Time of the 8-bit Up Counter

WDCTR<WDTW> sets the clear time of the 8-bit up counter.

When WDCTR<WDTW> is "00", the clear time is equal to the overflow time of the 8-bit up counter, and the 8-bit up counter can be cleared at any time.

When WDCTR<WDTW> is not "00", the clear time is fixed to only a certain period within the overflow time of the 8-bit up counter. When the operation for clearing the 8-bit up counter is attempted outside the clear time, a watchdog timer interrupt request occurs.

At this time, the watchdog timer is not cleared but continues counting. When the 8-bit up counter is not cleared within the clear time, a watchdog timer reset request or a watchdog timer interrupt request occurs due to the overflow, depending on the WDCTR<WDTOUT> setting.

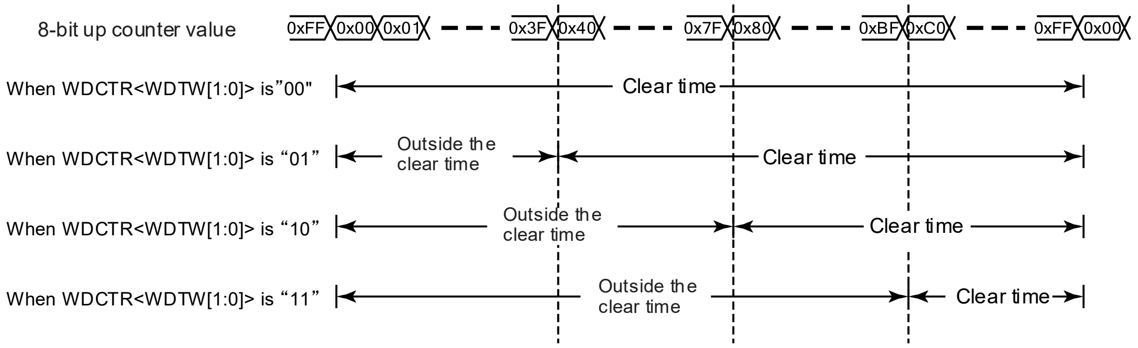


Figure 5-3 WDCTR<WDTW> and the 8-bit up Counter Clear Time

5.3.3. Setting the Overflow Time of the 8-bit Up Counter

WDCTR<WDTT> sets the overflow time of the 8-bit up counter.

When the 8-bit up counter overflows, a watchdog timer reset request signal or a watchdog timer interrupt request occurs, depending on the WDCTR<WDTOUT> setting.

When the watchdog timer interrupt request is selected as the malfunction detection signal, the watchdog counter continues counting, even after the overflow has occurred.

The watchdog timer temporarily stops counting up in the STOP mode (including warm-up) or in the IDLE/SLEEP mode, and restarts counting up after the STOP/IDLE/SLEEP mode is released.

To prevent the 8-bit up counter from overflowing immediately after the STOP/IDLE/SLEEP mode is released, it is recommended to clear the 8-bit up counter before the operation mode is changed.

Table 5-1 Watchdog Timer Overflow Time (fcgck = 10.0 [MHz], fs = 32.768 [kHz])

<WDTT>	Watchdog timer overflow time		
	NORMAL mode		SLOW mode
	SYSCR1<DV9CK> = 0	SYSCR1<DV9CK> = 1	
00	26.21 [ms]	62.50 [ms]	62.50 [ms]
01	104.86 [ms]	250.00 [ms]	250.00 [ms]
10	419.43 [ms]	1.000 [s]	1.000 [s]
11	1.678 [s]	4.000 [s]	4.000 [s]

Note: The 8-bit up counter source clock operates out of synchronization with WDCTR<WDTEN>. Therefore, the first overflow time of the 8-bit up counter after WDCTR<WDTEN> is set to "1" may get shorter by a maximum of 1 source clock. The 8-bit up counter must be cleared within a period of the overflow time minus 1 source clock cycle.

5.3.4. Setting an Overflow Detection Signal of the 8-bit Up Counter

WDCTR<WDTOUT> selects a signal to be generated when the overflow of the 8-bit up counter is detected.

(1) When the watchdog timer interrupt request is selected (when WDCTR<WDTOUT> is "0")

Clearing WDCTR<WDTOUT> to "0" causes a watchdog timer interrupt request to occur when the 8-bit up counter overflows.

A watchdog timer interrupt is a non-maskable interrupt, and its request is always accepted, regardless of the interrupt master enable flag (<IMF>) setting.

Note: When a watchdog timer interrupt occurs while another interrupt, including a watchdog timer interrupt, is already accepted, the new watchdog timer interrupt is processed immediately and the preceding interrupt is put on hold. Therefore, when watchdog timer interrupts occur continuously without execution of the RETN instruction, too many levels of nesting may cause a malfunction of the TMP89FS60B/62B/63B.

(2) When the watchdog timer reset request signal is selected (when WDCTR<WDTOUT> is "1")

Setting WDCTR<WDTOUT> to "1" causes a watchdog timer reset request signal to occur when the 8-bit up counter overflows.

This watchdog timer reset request signal resets the TMP89FS60B/62B/63B.

5.3.5. Writing the Watchdog Timer Control Codes

The watchdog timer control codes are written into WDCDR.

By writing "0x4E" (clear code) into WDCDR, the 8-bit up counter is cleared to "0" and continues counting the source clock.

When WDCTR<WDTEN> is "0", writing "0xB1" (disable code) into WDCDR disables the watchdog timer operation.

To prevent the 8-bit up counter from overflowing, clear the 8-bit up counter in a period shorter than the overflow time of the 8-bit up counter and within the clear time.

By designing the program so that no overflow will occur, the program malfunctions and deadlock can be detected through interrupts generated by watchdog timer interrupt requests.

By applying a reset to the microcontroller using watchdog timer reset request signals, the CPU can be restored from malfunctions and deadlock.

Example: When WDCTR<WDTEN> is "0", set the watchdog timer detection time to 2^{20} / fcgck [s], set the counter clear time to half of the overflow time, and allow a watchdog timer reset request signal to occur when an overflow is detected.

```
                                                LD    (WDCTR), 0x33    ; <WDTW> ← "10"、<WDTT> ← "01"、<WDTOUT> ← "1"
Clear the 8-bit up counter at a point after           :
[a half of the overflow time] and within              :
[a period of the overflow time] minus
[1 source clock cycle].                         LD    (WDCDR), 0x4E    ; Clear the 8-bit up counter

Clear the 8-bit up counter at a point after           :
[a half of the overflow time] and within              :
[a period of the overflow time] minus
[1 source clock cycle].                         LD    (WDCDR), 0x4E    ; Clear the 8-bit up counter
```

Note: When the overflow of the 8-bit up counter and writing of "0x4E" (clear code) into WDCDR occur simultaneously, the 8-bit up counter is cleared preferentially and the overflow detection is not executed.

5.3.6. Reading the 8-bit Up Counter

The counter value of the 8-bit up counter can be read by reading WDCNT.

The stoppage of the 8-bit up counter can be detected by reading WDCNT at random times and comparing the value to the last read value.

5.3.7. Reading the watchdog timer status

The watchdog timer status can be read at WDST.

WDST<WDTST> is set to "1" when the watchdog timer operation is enabled, and it is cleared to "0" when the watchdog timer operation is disabled.

WDST<WINTST2> is set to "1" when a watchdog timer interrupt request occurs due to the overflow of the 8-bit up counter.

WDST<WINTST1> is set to "1" when a watchdog timer interrupt request occurs due to the operation for releasing the 8-bit up counter outside the clear time.

You can know which factor has caused a watchdog timer interrupt request by reading WDST<WINTST2> and WDST<WINTST1> in the watchdog timer interrupt service routine.

WDST<WINTST2> and <WINTST1> are cleared to "0" when WDST is read. When WDST is read at the same time as the condition for turning WDST<WINTST2> or <WINTST1> to "1" is satisfied, WDST<WINTST2> or <WINTST1> is set to "1", rather than being cleared.

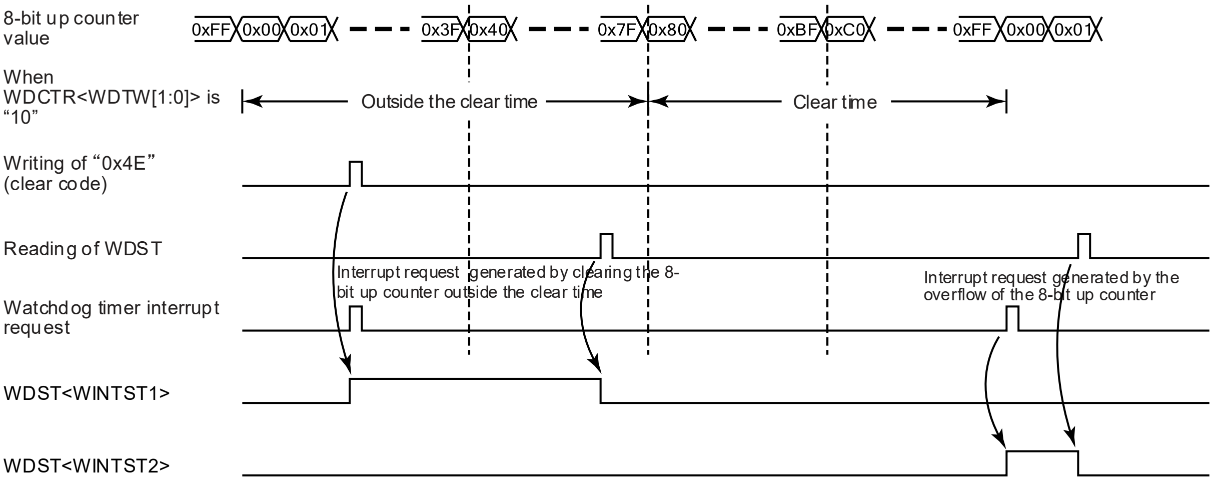

8-bit up counter value
0xFF
0x00
0x01
0x3F
0x40
0x7F
0x80
0xBF
0xC0
0xFF
0x00
0x01
When WDCTR<WDTW[1:0]> is "10"
Outside the clear time
Clear time
Writing of "0x4E" (clear code)
Reading of WDST
Interrupt request generated by clearing the 8-bit up counter outside the clear time
Interrupt request generated by the overflow of the 8-bit up counter
Watchdog timer interrupt request
WDST<WINTST1>
WDST<WINTST2>

Figure 5-4 Changes in the Watchdog Timer Status

6. Power-on Reset Circuit (POR)

The power-on reset circuit generates a reset when the power is turned on. When the supply voltage is lower than the detection voltage of the power-on reset circuit, a power-on reset signal is generated.

6.1. Configuration

The power-on reset circuit consists of a reference voltage generation circuit and a comparator. The supply voltage divided by ladder resistor is compared with the voltage generated by the reference voltage generation circuit by the comparator.

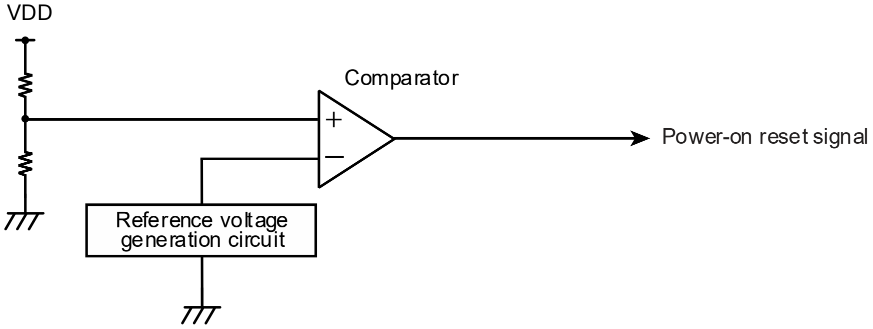


Figure 6-1 Power-on Reset Circuit

6.2. Function

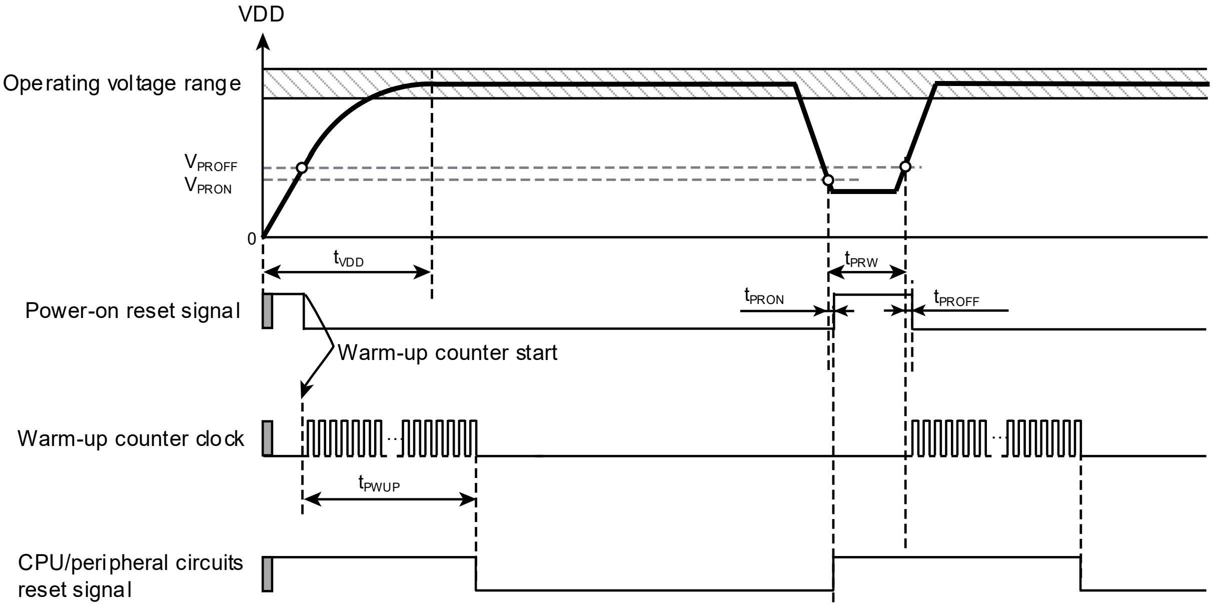
When the power supply voltage goes on, when the supply voltage is equal to or lower than the releasing voltage of the power-on reset circuit (V_{PROFF}), a power-on reset signal is generated and when it is higher than the releasing voltage of the power-on reset circuit, a power-on reset signal is released.

When power supply voltage goes down, when the supply voltage is the detecting voltage of the power-on reset circuit (V_{PRON}) or lower, a power-on reset signal is generated.

While the power-on reset signal is generated, the warm-up circuit, the CPU and the peripheral circuits are reset.

When the power-on reset signal is released, the warm-up circuit is activated. The reset of the CPU and peripheral circuits is released after the warm-up time that follows reset release has elapsed.

Increase the supply voltage into the operating range during the period from detection of the power-on reset release voltage (V_{PROFF}) until the end of the warm-up time that follows reset release. When the supply voltage has not reached the operating voltage range by the end of the warm-up time that follows reset release, the TMP89FS60B/62B/63B cannot operate properly.



Note 1: The power-on reset circuit may operate improperly, depending on fluctuations in the supply voltage (V_{DD}). Refer to the electrical characteristics and take them into consideration when designing equipment.

Note 2: About turning on power supply after turning off, be sure to lower the power supply voltage below the detecting voltage of the power-on reset circuit (V_{PRON}) and hold it for 1 [ms] or more. After that, please follow the same constraints as when turning on the power and turn on the power supply voltage.

Note 3: For the AC timing, refer to "25. Electrical Characteristics".

Figure 6-2 Operation Timing of Power-on Reset

7. Voltage Detection Circuit (VLTD)

The voltage detection circuit detects any decrease in the supply voltage and generates the INTVLTD interrupt request and voltage detection reset signals.

Note: The voltage detection circuit may operate improperly, depending on fluctuations in the supply voltage (V_{DD}). Refer to the electrical characteristics and take them into consideration when designing equipment.

7.1. Configuration

The voltage detection circuit consists of a reference voltage generation circuit, a detection voltage level selection circuit, a comparator and control registers.

The supply voltage (V_{DD}) is divided by the ladder resistor and input to the detection voltage selection circuit. The detection voltage selection circuit selects a voltage according to the specified detection voltage (<VD1LVL>), and the comparator compares it with the reference voltage. When the comparator detects the selected voltage, the voltage detection reset signal or the INTVLTD interrupt request can occur.

Whether to generate the voltage detection reset signal or the INTVLTD interrupt request can be selected by software. In the former case, the voltage detection reset signal is generated when the supply voltage (V_{DD}) becomes lower than the detection voltage (<VD1LVL>). In the latter case, the INTVLTD interrupt request is generated when the supply voltage (V_{DD}) becomes lower than the detection voltage (<VD1LVL>).

Note: Since the comparators used for voltage detection do not have a hysteresis. Therefore, the INTVLTD interrupt requests may be generated frequently when the supply voltage (V_{DD}) is close to the detection voltage (<VD1LVL>). The INTVLTD interrupt requests may be generated not only when the supply voltage falls and is close to the detection voltage but also when it rises and is close to the detection voltage.

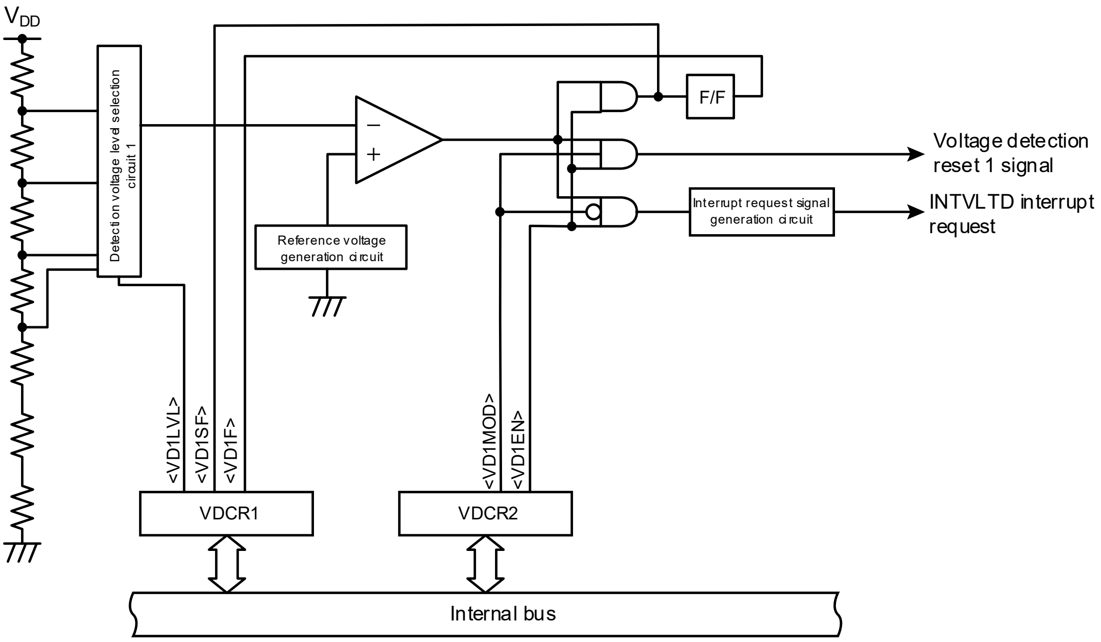

VDD
Detection voltage level selection circuit 1
−
+
Reference voltage generation circuit
F/F
Voltage detection reset 1 signal
Interrupt request signal generation circuit
INTVLTD interrupt request
<VD1LVL>
<VD1SF>
<VD1F>
<VD1MOD>
<VD1EN>
VDCR1
VDCR2
Internal bus

Figure 7-1 Voltage Detection Circuit

7.2. Control

The voltage detection circuit is controlled by voltage detection control registers 1 and 2.

Voltage detection control register 1

VDCR1		7	6	5	4	3	2	1	0
(0x0FC6)	Bit Symbol	"0"	-	"1"	"0"	VD1F	VD1SF	VD1LVL	
	Read/Write	R/W	R	R/W	R/W	R/W	R	R/W	
	After reset	0	0	1	0	0	0	0	0

			Read	Write
VD1F	Voltage detection 1 flag (Retains the magnitude relation of V_{DD} and the detection voltage 1 when V_{DD} < detection voltage 1 is detected.)	0:	V_{DD} ≥ Detection voltage 1	Clears <VD1F> to "0"
		1:	V_{DD} < Detection voltage 1	-
VD1SF	Voltage detection 1 status flag (Shows the magnitude relation of V_{DD} and the detection voltage 1 when VDCR1 is read)	0:	V_{DD} ≥ Detection voltage 1	
		1:	V_{DD} < Detection voltage 1	
VD1LVL	Selection for detection voltage 1	00:	4.5 ± 0.1 [V]	
		01:	4.3 ± 0.1 [V]	
		10:	Reserved	
		11:	Reserved	

Note 1: VDCR1 is initialized by a power-on reset or an external reset input.

Note 2: When <VD1F> is cleared by the software and is set due to voltage detection at the same time, the setting due to voltage detection is given priority.

Note 3: <VD1F> cannot be set to "1" by the software.

Note 4: Bit 6 of VDCR1 is read as "0".

Note 5: Bit 7 and 4 of VDCR1 are set to "0". Bit 5 is set to "1".

Voltage detection control register 2

VDCR2		7	6	5	4	3	2	1	0
(0x0FC7)	Bit Symbol	-	-	"0"	"0"	"0"	"0"	VD1MOD	VD1EN
	Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

VD1MOD	Selects the operation mode of voltage detection 1	0: 1:	Generates an INTVLTD interrupt request Generates a voltage detection reset 1 signal
VD1EN	Enables/disables the operation of voltage detection 1	0: 1:	Disables the operation of voltage detection 1 Enables the operation of voltage detection 1

Note 1: VDCR2 is initialized by a power-on reset or an external reset input.

Note 2: When a read instruction is executed on VDCR2, bits 7 and 6 are read as "0".

Note 3: Bit 5 to 2 of VDCR2 must be cleared to "0".

Note 4: When releasing the STOP mode, the operation mode of the voltage detection 1 must be set "Generates a voltage detection reset 1 signal" (<VD1MOD> = 1).

7.3. Function

The detection voltages (<VD1LVL>) can be set in the voltage detection circuit. Enabling/disabling the voltage detection and the operation can be specified by a software when the supply voltage (V_{DD}) is below the detection voltage (<VD1LVL>).

7.3.1. Enabling/disabling the voltage detection operation

Setting VDCR2<VD1EN> to "1" enables the voltage detection operation. Setting it to "0" disables the operation.

VDCR2<VD1EN> is cleared to "0" immediately after a power-on reset or a reset by an external reset input is released.

Note: When the supply voltage (V_{DD}) is lower than the detection voltage (<VD1LVL>), setting VDCR2<VD1EN> to "1" generates an INTVLTD interrupt request or a voltage detection reset signal at the time.

7.3.2. Selecting the voltage detection operation mode

When VDCR2<VD1MOD> is set to "0", the voltage detection operation mode is set to generate the INTVLTD interrupt request. When VDCR2<VD1MOD> is set to "1", the operation mode is set to generate voltage detection reset signals.

- When the operation mode is set to generate INTVLTD interrupt request (VDCR2<VD1MOD> = 0)

  When VDCR2<VD1EN> = 1, an INTVLTD interrupt request is generated when the supply voltage (V_{DD}) falls and equals to the detection voltage (<VD1LVL>).

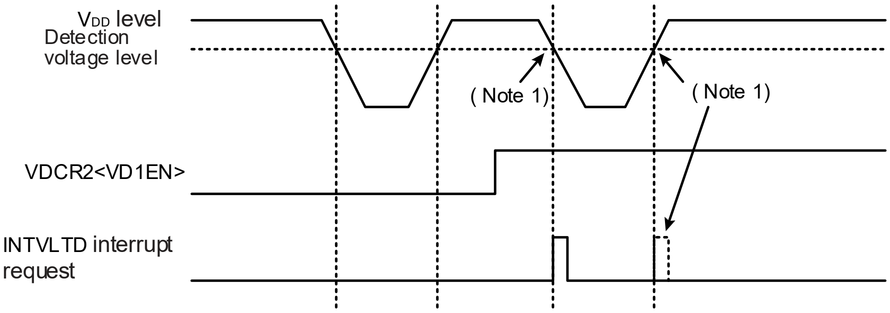


Figure 7-2 Voltage Detection Interrupt Request INTVLTD

Note 1: Since the comparators used for voltage detection do not have a hysteresis. Therefore, the INTVLTD interrupt requests may be generated frequently when the supply voltage (V_{DD}) is close to the detection voltage (<VD1LVL>). The INTVLTD interrupt requests may be generated not only when the supply voltage falls and is close to the detection voltage but also when it rises and is close to the detection voltage.

Note 2: When the supply voltage (V_{DD}) falls and is less or the detection voltage (<VD1LVL>) during IDLE0 or SLEEP0 mode, an INTVLTD interrupt request is generated after the TBT counts the specified period and IDLE0 or SLEEP mode is released. In the case of STOP mode, an INTLVTD interrupt request is generated after STOP mode is released by the $\overline{\text{STOP}}$ pin.

- When the operation mode is set to generate voltage detection reset signals (VDCR2<VD1MOD> = 1)

  When VDCR2<VD1EN> = 1, a voltage detection reset signal is generated when the supply voltage (V_{DD}) becomes lower than the detection voltage (<VD1LVL>).

  VDCR1 and VDCR2 are initialized by a power-on reset or an external reset input only. A voltage detection reset signal is generated continuously as long as the supply voltage (V_{DD}) is lower than the detection voltage (<VD1LVL>).

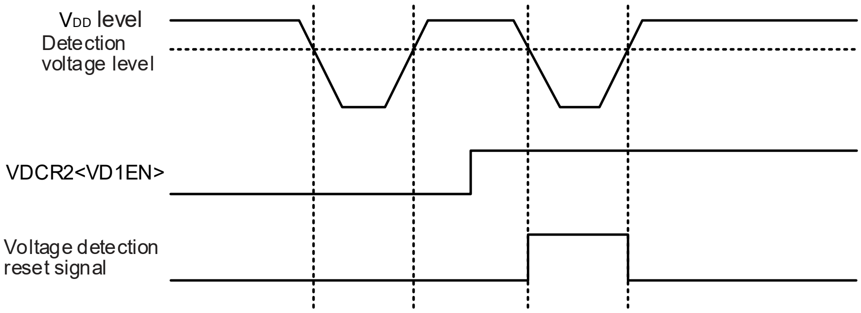


Figure 7-3 Voltage Detection Reset Signal

7.3.3. Selecting the detection voltage level

Select a detection voltage at VDCR1<VD1LVL>.

7.3.4. Voltage detection flag and voltage detection status flag

The magnitude relation between the supply voltage (V_{DD}) and the detection voltage (<VD1LVL>) can be checked by reading VDCR1<VD1F> and <VD1SF>.

When VDCR2<VD1EN> is set at "1", when the supply voltage (V_{DD}) becomes lower than the detection voltage (<VD1LVL>), VDCR1<VD1F> is set to "1" and is held in this state. VDCR1<VD1F> is not cleared to "0" when the supply voltage (V_{DD}) becomes equal to or higher than the detection voltage (<VD1LVL>).

When VDCR2<VD1EN> is cleared to "0" after VDCR1<VD1F> is set to "1", the previous state is still held. To clear VDCR1<VD1F>, "0" must be written to it.

When VDCR2<VD1EN> is set at "1", when the supply voltage (V_{DD}) becomes lower than the detection voltage (<VD1LVL>), VDCR1<VD1SF> is set to "1". When the supply voltage (V_{DD}) becomes equal to or higher than the detection voltage (<VD1LVL>), VDCR1<VD1SF> is cleared to "0".

Unlike VDCR1<VD1F>, VDCR1<VD1SF> does not hold the set state.

Note 1: When the supply voltage (V_{DD}) becomes lower than the detection voltage (<VD1LVL>) in the IDLE0 or SLEEP0 mode, the voltage detection flag and the voltage detection status flag are changed after the operation mode is returned to NORMAL or SLOW mode.

Note 2: Depending on the voltage detection timing, the voltage detection status flag (VD1SF) may be changed earlier than the voltage detection flag (VD1F) by a maximum of 2 / fcgck [s].

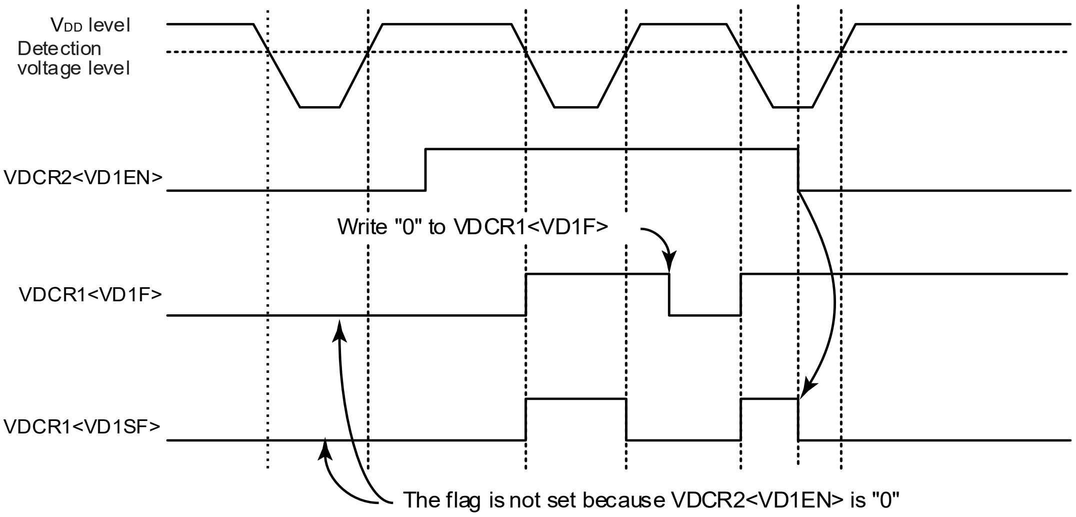


Figure 7-4 Changes in the Voltage Detection Flag and the Voltage Detection Status Flag

7.4. Register Settings

7.4.1. Setting procedure when the operation mode is set to generate INTVLTD interrupt request

When the operation mode is set to generate INTVLTD interrupt request, make the following setting

(1) Clear the voltage detection circuit interrupt enable flag to "0".
(2) Set the detection voltage at VDCR1<VD1LVL>.
(3) Clear VDCR2<VD1MOD> to "0" to set the operation mode to generate INTVLTD interrupt request.
(4) Set VDCR2<VD1EN> to "1" to enable the voltage detection operation.
(5) Wait for 5 [μs] or more until the voltage detection circuit becomes stable.
(6) Make sure that VDCR1<VD1SF> is "0".
(7) Clear VDCR1<VD1F> and the voltage detection circuit interrupt latch to "0" and set the interrupt enable flag to "1" to enable interrupt.

Note: When the supply voltage (V_{DD}) is close to the detection voltage (<VD1LVL>), INTVLTD interrupt request may be generated frequently. When this may pose any problem, execute appropriate wait processing depending on fluctuations in the system power supply and clear the interrupt latch before returning from the INTVLTD interrupt service routine.

To disable the voltage detection circuit while it is enabled with the INTVLTD interrupt request, make the following setting:

(1) Clear the voltage detection circuit interrupt enable flag to "0".
(2) Clear VDCR2<VD1EN> to "0" to disable the voltage detection operation.

Note: When the voltage detection circuit is disabled without clearing interrupt enable flag, unexpected interrupt request may occur.

7.4.2. Setting procedure when the operation mode is set to generate voltage detection reset signals

When the operation mode is set to generate voltage detection reset signals, make the following setting:

(1) Clear the voltage detection circuit interrupt enable flag to "0".
(2) Set the detection voltage at VDCR1<VD1LVL>.
(3) Clear VDCR2<VD1MOD> to "0" to set the operation mode to generate INTVLTD interrupt request.
(4) Set VDCR2<VD1EN> to "1" to enable the voltage detection operation.
(5) Wait for 5 [μs] or more until the voltage detection circuit becomes stable.
(6) Make sure that VDCR1<VD1SF> is "0".
(7) Clear VDCR1<VD1F> to "0".
(8) Set VDCR2<VD1MOD> to "1" to set the operation mode to generate voltage detection reset signals.

VDCR1 and VDCR2 are initialized by a power-on reset or an external reset input only. Because the voltage detection operation is not disabled by the voltage detection reset, the voltage detection reset signals are generated continuously while the supply voltage (V_{DD}) is lower than the detection voltage. Also, when the supply voltage (V_{DD}) once becomes the detection voltage or higher, the supply voltage (V_{DD}) falls lower than the detection voltage again before VDCR2<VD1EN> is cleared to “0”, the voltage detection reset signals are generated immediately.

To disable the voltage detection circuit while it is enabled with the voltage detection reset, make the following setting:

(1) Clear the voltage detection circuit interrupt enable flag to "0".
(2) Clear VDCR2<VD1MOD> to "0" to set the operation mode to generate INTVLTD interrupt requests.
(3) Clear VDCR2<VD1EN> to "0" to disable the voltage detection operation.

Note: When the voltage detection circuit is disabled without clearing interrupt enable flag, unexpected interrupt request may occur.

8. I/O Ports

The TMP89FS60B/62B/63B has input/output ports as follows.

The available ports are depend on a product. Refer to “8.2. Input/output ports for Each Product”.

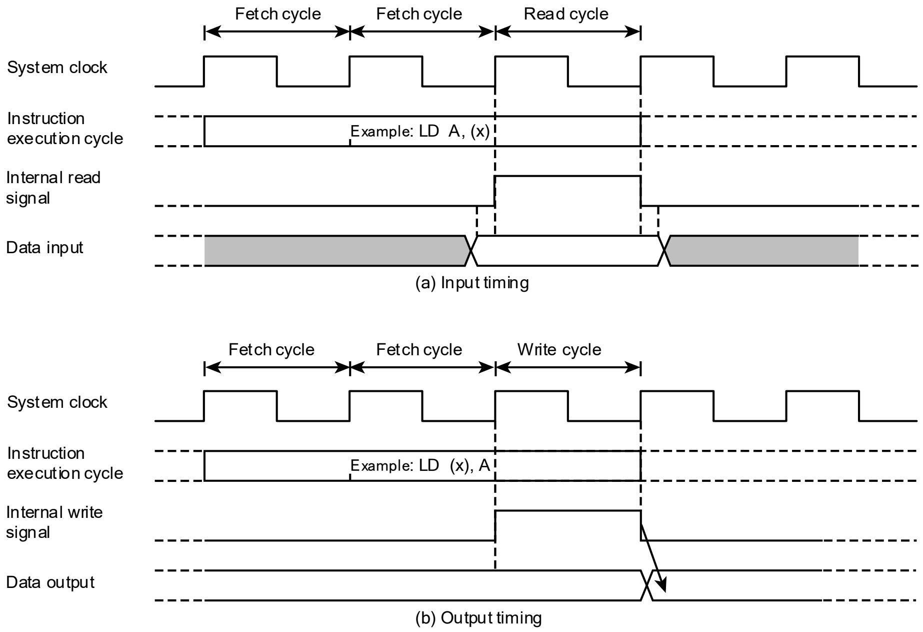
Table 8-1 List of all I/O Ports

Port name	Pin name	Number of pins	Input/output	Secondary functions
Port P0	P03 to P02	2	Input/output	Also used as the low-frequency oscillator connection pin
Port P1	P13 to P10	4	Input/output	Also used as the external reset input, the external interrupt input and the STOP mode release signal input
Port P2	P27 to P20	8	Input/output	Also used as the UART input/output, the serial interface input/output and the serial bus interface input/output
Port P4	P47 to P40	8	Input/output	Also used as the analog input and the key-on wakeup input
Port P5	P57 to P50	8	Input/output	Also used as the analog input
Port P7	P77 to P70	8	Input/output	Also used as the timer counter input/output, the divider output and the external interrupt input
Port P8	P84 to P80	5	Input/output	Also used as the timer counter input/output
Port P9	P94 to P90	5	Input/output	Also used as the UART input/output and the serial interface input/output
Port PB	PB7 to PB0	8	Input/output	Also used as the UART input/output and the serial interface input/output

All output ports contain a latch, which holds the output data. No input port has a latch, so the external input data should be externally held until the input data is read or reading should be performed several times before processing.

Figure 8-1 shows input/output timing examples.

External data is read from an I/O port in the read cycle during execution of the read instruction. This timing cannot be recognized from outside, so that transient input such as chattering must be processed by the program. Data is output to an I/O port in the next cycle of the write cycle during execution of the write instruction.



Note: The positions of the read and write cycles may vary, depending on the instruction.

Figure 8-1 Input/output Timing Example

8.1. Input/output Port Control Registers

The following control registers are used for I/O ports. (The port number is indicated in place of x.) Registers that can be set vary depending on the port. For details, refer to the description of each port.

- PxDR register

  This is the register for setting output data. When a port is set to the output mode, the value specified at PxDR is output from the port.

- PxPRD register

  This is the register for reading input data. When a port is set to the input mode, the current port input status can be read by reading PxPRD.

- PxCR register

  This register switches a port between input and output. A port can be switched between the input mode and the output mode.

- PxFC register

  This register enables the secondary function output of each port. The secondary function output of each port can be enabled or disabled.

- PxOUTCR register

  This register switches the port output between the CMOS output and the open drain output.

- PxPU register

  This register determines whether or not the built-in pull-up resistor is connected when a port is used in the input mode or as the open drain output.

8.2. Input/output ports for Each Product

The input/output ports for each product are shown in Table 8-2.
About setting for unavailable port, refer to each port section.

Table 8-2 Input/output Ports for Each Product

Name of port	Pin name	TMP89FS60B	TMP89FS62B	TMP89FS63B
P0 Port	P03	A	A	A
	P02	A	A	A
P1 Port	P13	A	NA	NA
	P12	A	NA	NA
	P11	A	A	A
	P10	A	A	A
P2 Port	P27	A	NA	NA
	P26	A	NA	NA
	P25	A	NA	A
	P24	A	NA	A
	P23	A	A	A
	P22	A	A	A
	P21	A	A	A
	P20	A	A	A
P4 Port	P47	A	A	A
	P46	A	A	A
	P45	A	A	A
	P44	A	A	A
	P43	A	A	A
	P42	A	A	A
	P41	A	A	A
	P40	A	A	A
P5 Port	P57	A	NA	NA
	P56	A	NA	NA
	P55	A	NA	NA
	P54	A	NA	A
	P53	A	NA	A
	P52	A	NA	A
	P51	A	NA	A
	P50	A	NA	A

Name of port	Pin name	TMP89FS60B	TMP89FS62B	TMP89FS63B
P7 Port	P77	A	A	A
	P76	A	A	A
	P75	A	A	A
	P74	A	A	A
	P73	A	A	A
	P72	A	A	A
	P71	A	A	A
	P70	A	A	A
P8 Port	P84	A	NA	NA
	P83	A	NA	NA
	P82	A	NA	NA
	P81	A	A	A
	P80	A	A	A
P9 Port	P94	A	A	A
	P93	A	A	A
	P92	A	A	A
	P91	A	A	A
	P90	A	A	A
PB Port	PB7	A	NA	NA
	PB6	A	NA	NA
	PB5	A	A	A
	PB4	A	A	A
	PB3	A	A	A
	PB2	A	A	A
	PB1	A	A	A
	PB0	A	A	A

Note: A: Available, NA: not available

8.3. List of I/O Port Settings

For the settings for the I/O ports, refer to Table 8-3.

Table 8-3 List of I/O Ports Settings

Port name	Pin name	Function	Register set value			
			PxCR	PxOUTCR	PxFC	Other required settings
Port P0	P03 to P02	Port input	0	Without register	0	-
		Port output	1		0	-
	P03	XTOUT	*		Without register	-
	P02	XTIN	*		1	-
Port P1	P13 to P11	Port input	0	Without register	Without register	-
		Port output	1			-
	P10	Port input	0			(Note 1)
		Port output	1			(Note 1)
	P13	INT1 input	0			-
	P12	$\overline{\text{INT0}}$ input	0			-
	P11	$\overline{\text{INT5}}$ input	0			-
		$\overline{\text{STOP}}$ input	0			-
	P10	$\overline{\text{RESET}}$ input	*			(Note 1)
Port P2	P27 to P20	Port input	0	*	*	-
		Port output	1	**	0	-
	P25	SCLK0 input (Note 4)	0	*	*	SERSEL<SRSEL0> = 01
		SCLK0 output (Note 4)	1	**	1	SERSEL<SRSEL0> = 01
	P24	SCL0 input/output (Note 4)	1	Without register	1	SERSEL<SRSEL0> = *0
		SI0 input (Note 4)	0		*	SERSEL<SRSEL0> = 01
	P23	SDA0 input/output (Note 4)	1	Without register	1	SERSEL<SRSEL0> = *0
		SO0 output (Note 4)	1		1	SERSEL<SRSEL0> = 01
	P22	SCLK0 input	0	*	*	SERSEL<SRSEL0> = 10 SERSEL<SRSEL2> = 0
		SCLK0 output	1	**	1	SERSEL<SRSEL0> = 10 SERSEL<SRSEL2> = 0
	P21	RXD0 input	0	*	*	SERSEL<SRSEL0> = 0* SERSEL<SRSEL2> = 0 UATCNG<UAT0IO> = 0
		TXD0 output	1	**	1	SERSEL<SRSEL0> = 0* SERSEL<SRSEL2> = 0 UATCNG<UAT0IO> = 1
		SI0 input	0	*	*	SERSEL<SRSEL0> = 10 SERSEL<SRSEL2> = 0

Port name	Pin name	Function	Register set value			
			PxCR	PxOUTCR	PxFC	Other required settings
Port P2	P20	TXD0 output	1	**	1	SERSEL<SRSEL0> = 0* SERSEL<SRSEL2> = 0 UATCNG<UAT0IO> = 0
		RXD0 input	0	*	*	SERSEL<SRSEL0> = 0* SERSEL<SRSEL2> = 0 UATCNG<UAT0IO> = 1
		SO0 output	1	**	1	SERSEL<SRSEL0> = 10 SERSEL<SRSEL2> = 0
Port P4	P47 to P40	Port input	0	Without register	*	-
		Port output	1		0	-
		AIN7 to AIN0	0		1	-
		KWI7 to KWI4	*		*	KWUCR1
		KWI3 to KWI0	*		*	KWUCR0
Port P5	P57 to P50	Port input	0	Without register	0	-
		Port output	1		0	-
		AIN15 to AIN8 (Note 6)	0		1	-
Port P7	P77 to P70	Port input	0	Without register	*	-
		Port output	1		0	-
	P77	INT4 input	0		Without register	-
	P76	INT3 input	0		Without register	-
	P75	INT2 input	0		Without register	-
	P74	$\overline{\text{DVO}}$ output	1		1	-
	P73	TCA1 input	0		*	-
		$\overline{\text{PPGA1}}$ output	1		1	-
	P72	TCA0 input	0		*	SERSEL<TCA0SEL> = 00
		$\overline{\text{PPGA0}}$ output	1		1	-
	P71	TC01 input	0		*	-
		$\overline{\text{PPG01}}$/$\overline{\text{PWM01}}$ output	1		1	-
	P70	TC00 input	0		*	-
		$\overline{\text{PPG00}}$/$\overline{\text{PWM00}}$ output	1		1	-
Port P8	P84 to P80	Port input	0	Without register	*	-
		Port output	1		0	-
	P81	TC03 input	0		*	-
		$\overline{\text{PPG03}}$/$\overline{\text{PWM03}}$ output	1		1	-
	P80	TC02 input	0		*	-
		$\overline{\text{PPG02}}$/$\overline{\text{PWM02}}$ output	1		1	-

Port name	Pin name	Function	Register set value			
			PxCR	PxOUTCR	PxFC	Other required settings
Port P9	P94 to P90	Port input	0	*	*	-
		Port output	1	**	0	-
	P94	RXD2 input	0	*	*	UATCNG<UAT2IO> = 0
		TXD2 output	1	**	1	UATCNG<UAT2IO> = 1
	P93	TXD2 output	1	**	1	UATCNG<UAT2IO> = 0
		RXD2 input	0	*	*	UATCNG<UAT2IO> = 1
	P92	SCLK1 input	0	*	*	SERSEL<SRSEL1> = 10
		SCLK1 output	1	**	1	SERSEL<SRSEL1> = 10
	P91	RXD1 input	0	*	*	SERSEL<SRSEL1> = 0* UATCNG<UAT1IO> = 0
		TXD1 output	1	**	1	SERSEL<SRSEL1> = 0* UATCNG<UAT1IO> = 1
		SI1 input	0	*	*	SERSEL<SRSEL1> = 10
	P90	TXD1 output	1	**	1	SERSEL<SRSEL1> = 0* UATCNG<UAT1IO> = 0
		RXD1 input	0	*	*	SERSEL<SRSEL1> = 0* UATCNG<UAT1IO> = 1
		SO1 output	1	**	1	SERSEL<SRSEL1> = 10
Port PB	PB7 to PB0	Port input	0	*	*	-
		Port output	1	**	0	-
	PB6	SCLK0 input (Note 5)	0	*	*	SERSEL<SRSEL0> = 10 SERSEL<SRSEL2> = 1
		SCLK0 output (Note 5)	1	**	1	SERSEL<SRSEL0> = 10 SERSEL<SRSEL2> = 1
	PB5	RXD0 input	0	*	*	SERSEL<SRSEL0> = 0* SERSEL<SRSEL2> = 1 UATCNG<UAT0IO> = 0
		TXD0 output	1	**	1	SERSEL<SRSEL0> = 0* SERSEL<SRSEL2> = 1 UATCNG<UAT0IO> = 1
		SI0 input (Note 5)	0	*	*	SERSEL<SRSEL0> = 10 SERSEL<SRSEL2> = 1
	PB4	TXD0 output	1	**	1	SERSEL<SRSEL0> = 0* SERSEL<SRSEL2> = 1 UATCNG<UAT0IO> = 0
		RXD0 input	0	*	*	SERSEL<SRSEL0> = 0* SERSEL<SRSEL2> = 1 UATCNG<UAT0IO> = 1
		SO0 output (Note 5)	1	**	1	SERSEL<SRSEL0> = 10 SERSEL<SRSEL2> = 1

Note 1: After the power is turned on, pin P10 operates as an external reset input. To use pin P10 as a port, refer to "2.4.4.9. How to use the external reset input pin as a port".

Note 2: About SERSEL and UATCNG, please refer to “8.5. Serial Interface Selecting Function”.

Note 3: About KWUCR0 and KWUCR1, please refer to “19. Key-on Wakeup (KWU)”.

Note 4: This function is not available for the TMP89FS62B.

Note 5: This function is not available for the TMP89FS62B and TMP89FS63B.

Note 6: The TMP89FS62B does not have AIN15 to AIN8. And, the TMP89FS63B does not have AIN15 to AIN13.

Note 7: The symbol and numeric characters in the table have the following meanings:

Symbol and numeric characters	Meaning
0	Set to "0".
1	Set to "1".
*	Don’t care. (Operation is the same whether "1" or "0" is selected.)
**	The open-drain output or the CMOS output can be selected.
Without register	There is no register that corresponds to the bit.

8.4. I/O Port Registers

8.4.1. Port P0

Port P0 is a 2-bit input/output port that can be set to input or output for each bit individually, and it is also used as the low-frequency resonator connection pin.

Port P0 contains a programmable pull-up resistor on the VDD pin side. This pull-up resistor can be used when the port is used in the input mode.

Table 8-4 Port P0

	-	-	-	-	P03	P02	-	-
Secondary function	-	-	-	-	XTOUT	XTIN	-	-

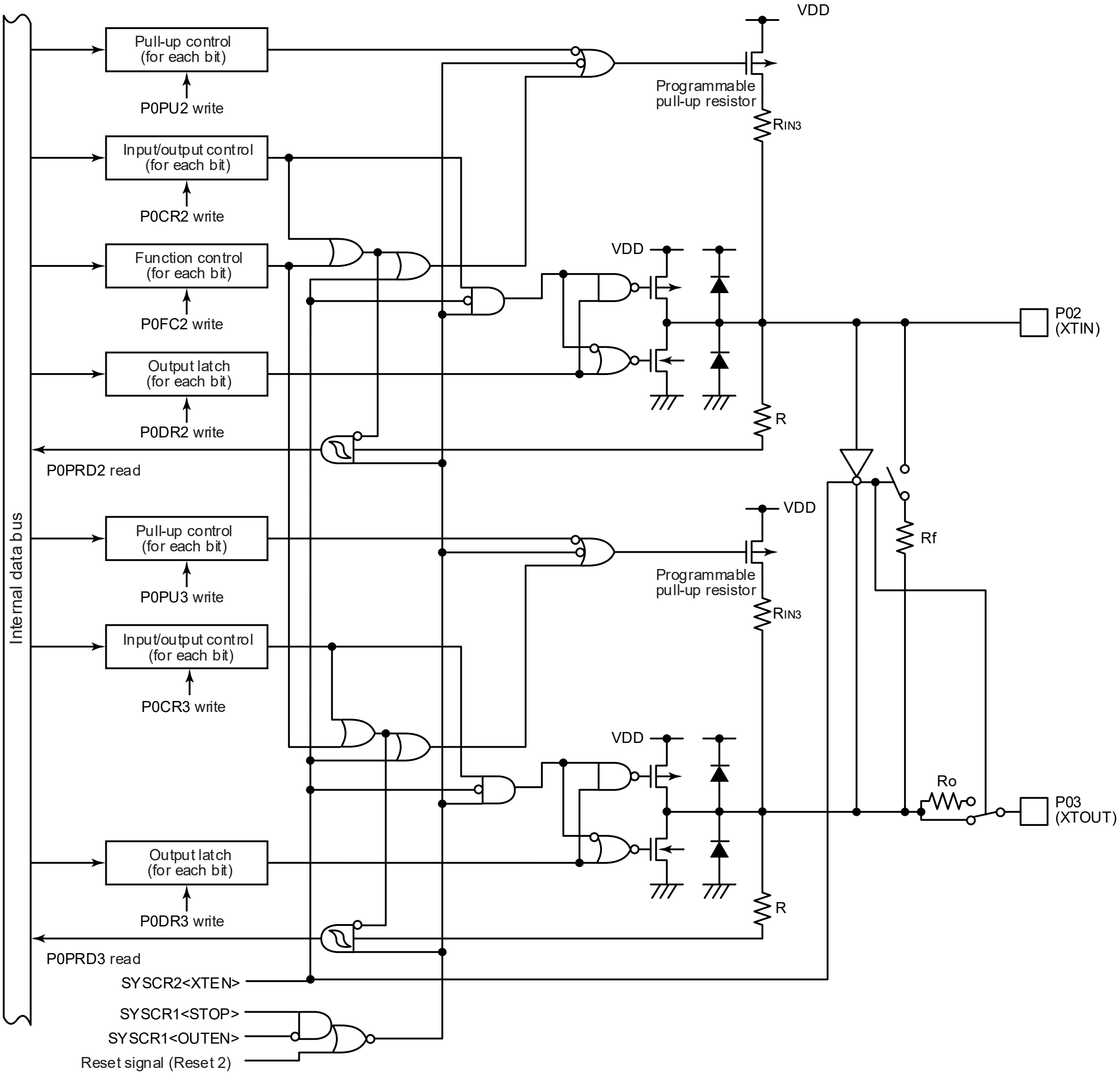
8.4.1.1. P0 Port for Each Product

P0 port for each product is shown in Table 8-5.

Table 8-5 P0 Port for Each Product

	TMP89FS60B	TMP89FS62B	TMP89FS63B
P03	A	A	A
P02	A	A	A

Note: A: Available, NA: Not available


VDD
Pull-up control (for each bit)
P0PU2 write
Programmable pull-up resistor
RIN3
Input/output control (for each bit)
P0CR2 write
Function control (for each bit)
P0FC2 write
VDD
P02 (XTIN)
Output latch (for each bit)
P0DR2 write
R
P0PRD2 read
VDD
Rf
Internal data bus
Pull-up control (for each bit)
P0PU3 write
Programmable pull-up resistor
RIN3
Input/output control (for each bit)
P0CR3 write
VDD
Ro
P03 (XTOUT)
Output latch (for each bit)
P0DR3 write
R
P0PRD3 read
SYSCR2<XTEN>
SYSCR1<STOP>
SYSCR1<OUTEN>
Reset signal (Reset 2)

Note 1: R = 100 [Ω] (typ.)

Note 2: Rf = 6 [MΩ] (typ.)

Note 3: Ro = 220 [kΩ] (typ.)

Note 4: R_{IN3} = 50 [kΩ] (typ.)

Figure 8-2 Port P0

8.4.1.2. Control

P0 port is controlled by the following registers.

Port P0 output latch

P0DR		7	6	5	4	3	2	1	0
(0x0000)	Bit Symbol	-	-	-	-	P03	P02	”0”	”0”
	Read/Write	R	R	R	R	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

			TMP89FS60B	TMP89FS62B	TMP89FS63B
P03	Output level in the output mode	0: 1:	“Low” level “High” level		
P02	Output level in the output mode	0: 1:	“Low” level “High” level		

Note: The bits 1 and 0 of P0DR must be clear to “0”.

Port P0 input/output control

P0CR		7	6	5	4	3	2	1	0
(0x0F1A)	Bit Symbol	-	-	-	-	P0CR3	P0CR2	”0”	”0”
	Read/Write	R	R	R	R	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

			TMP89FS60B	TMP89FS62B	TMP89FS63B
P0CR3	Selects input mode or output mode.	0:	Input mode		
		1:	Output mode		
P0CR2	Selects input mode or output mode.	0:	Input mode		
		1:	Output mode		

Note: The bits 1 and 0 of P0CR must be cleared to “0”.

Port P0 function control

P0FC		7	6	5	4	3	2	1	0
(0x0F34)	Bit Symbol	-	-	-	-	-	P0FC2	-	”1”
	Read/Write	R	R	R	R	R	R/W	R	R/W
	After reset	0	0	0	0	0	0	0	1

			TMP89FS60B	TMP89FS62B	TMP89FS63B
P0FC2	Port function	0: 1:	I/O port XTIN (I)		

Note 1: The bit 0 of P0FC must be set to “1”.

Note 2: Symbol "I" means secondary function input

Port P0 built-in pull-up resistor control

P0PU		7	6	5	4	3	2	1	0
(0x0F27)	Bit Symbol	-	-	-	-	P0PU3	P0PU2	”0”	”0”
	Read/Write	R	R	R	R	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

			TMP89FS60B	TMP89FS62B	TMP89FS63B
P0PU3	Selects connecting or disconnecting a built-in pull-up resister.	0: 1:	Refer to Table 8-6.		
P0PU2	Selects connecting or disconnecting a built-in pull-up resister.	0: 1:	Refer to Table 8-6.		

Note: The bits 1 and 0 of P0PU must be cleared to “0”.

Table 8-6 Built-in Pull-up resister for P03 and P02

Port function/ Secondary function	Connected or disconnected	Setting Condition			
		<P0PUj>	SYSCR2 <XTEN>	<P0CRj>	<P0FC2>
The connection pins for the low-frequency resonator	Disconnected	*	*	0	1
Port Input mode	Disconnected	0	0	0	0
	Connected	1			
Port Output mode	Disconnected	*		1	

Note 1: *: Don’t care

Note 2: j = 3 to 2

Port P0 input data

P0PRD		7	6	5	4	3	2	1	0
(0x000D)	Bit Symbol	-	-	-	-	P0PRD3	P0PRD2	-	-
	Read/Write	R	R	R	R	R	R	R	R
	After reset	0	0	0	0	*	*	*	*

		TMP89FS60B	TMP89FS62B	TMP89FS63B
P0PRD3	Input data from port	Refer to Table 8-7.		
P0PRD2	Input data from port	Refer to Table 8-7.		

Note: *: undefined

Table 8-7 Read Value of <P0PRDj>

Read value of <P0PRDj>	Setting Condition	
	<P0FC2>	<P0CRj>
0	*	1
	1	*
Level of the port pin	0	0

Note 1: * : Don’t care
Note 2: j = 3 to 2

8.4.2. Port P1

Port P1 is a 4-bit input/output port that can be set to input or output for each bit individually, and is also used as the external interrupt input, the STOP mode release signal input and the external reset input.

Port P1 contains a programmable pull-up resistor on the VDD pin side. This pull-up resistor can be used when the port is used in the input mode.

After reset, pin P10 is used as the external reset input. To use pin P10 as a port, refer to "2.4. Reset Control Circuit".

Table 8-8 Port P1

	-	-	-	-	P13	P12	P11	P10
Secondary function	-	-	-	-	INT1	$\overline{\text{INT0}}$	$\overline{\text{INT5}}$ $\overline{\text{STOP}}$	$\overline{\text{RESET}}$

8.4.2.1. P1 Port for Each Product

P1 port for each product is shown in Table 8-9.

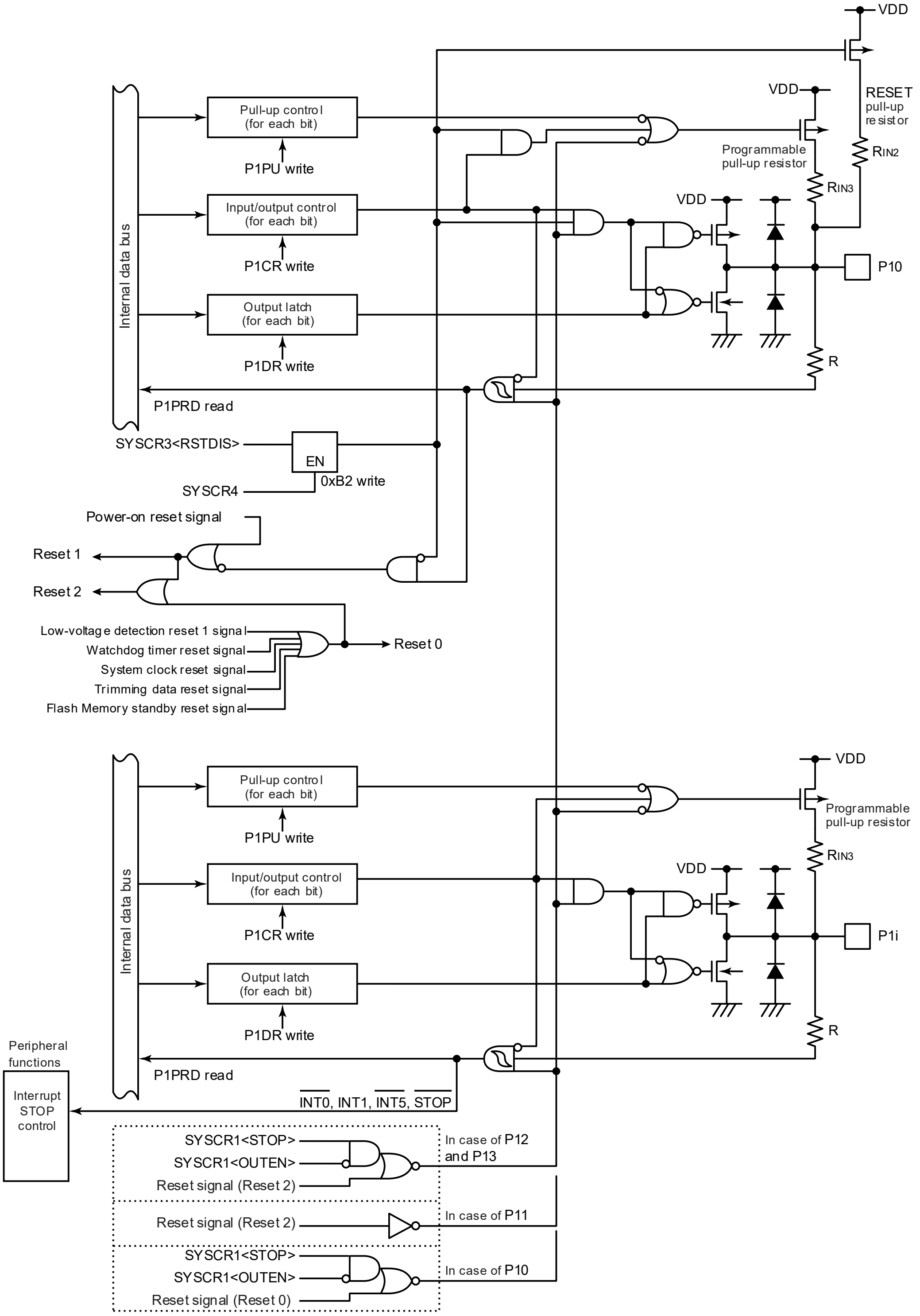
In regards to unavailable bits of the P1 port, the value which must be set or cleared to the corresponding bit of each register is shown in “8.4.2.2. Control”. These values must be set or cleared to corresponding bit of each register in the setup routine.

And also, when the value of the register which includes these bits is modified, don’t change the set or cleared value of these bit.

Table 8-9 P1 Port for Each Product

	TMP89FS60B	TMP89FS62B	TMP89FS63B
P13	A	NA	NA
P12	A	NA	NA
P11	A	A	A
P10	A	A	A

Note: A: Available, NA: Not available



Note 1: R = 100 [Ω] (typ.)

Note 2: RIN2 = 220 [kΩ] (typ.)

Note 3: RIN3 = 50 [kΩ] (typ.)

Note 4: TMP89FS60B: i = 3 to 1
TMP89FS62B, TMP89FS63B: i = 1

Figure 8-3 Port P1

8.4.2.2. Control

P1 port is controlled by the following registers.

Port P1 output latch

P1DR		7	6	5	4	3	2	1	0
(0x0001)	Bit Symbol	-	-	-	-	P13	P12	P11	P10
	Read/Write	R	R	R	R	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

			TMP89FS60B	TMP89FS62B	TMP89FS63B
P13	Output level in the output mode	0: 1:	“Low” level “High” level	Reserved (this bit symbol must be cleared to “0”.)	
P12	Output level in the output mode	0: 1:	“Low” level “High” level	Reserved (this bit symbol must be cleared to “0”.)	
P11	Output level in the output mode	0: 1:	“Low” level “High” level		
P10	Output level in the output mode	0: 1:	“Low” level “High” level		

Port P1 input/output control

P1CR		7	6	5	4	3	2	1	0
(0x0F1B)	Bit Symbol	-	-	-	-	P1CR3	P1CR2	P1CR1	P1CR0
	Read/Write	R	R	R	R	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

			TMP89FS60B	TMP89FS62B	TMP89FS63B
P1CR3	Selects input mode or output mode.	0:	Input mode INT1 (I)	Reserved (this bit symbol must be set to “1”.)	
		1:	Output mode		
P1CR2	Selects input mode or output mode.	0:	Input mode $\overline{\text{INT0}}$ (I)	Reserved (this bit symbol must be set to “1”.)	
		1:	Output mode		
P1CR1	Selects input mode or output mode.	0:	Input mode $\overline{\text{INT5}}$ (I) $\overline{\text{STOP}}$ (I)		
		1:	Output mode		
P1CR0	Selects input mode or output mode.	0:	Input mode		
		1:	Output mode (Note 2)		

Note 1: Symbol "I" means the secondary function input

Note 2: When P10 pin is used as $\overline{\text{RESET}}$ pin, the output buffer of P10 turns off regardless of the value of <P1CR0>.

Port P1 built-in pull-up resistor control

P1PU		7	6	5	4	3	2	1	0
(0x0F28)	Bit Symbol	-	-	-	-	P1PU3	P1PU2	P1PU1	P1PU0
	Read/Write	R	R	R	R	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

			TMP89FS60B	TMP89FS62B	TMP89FS63B
P1PU3	Selects connecting or disconnecting a built-in pull-up resister.	0: 1:	Refer to Table 8-11.	Reserved (this bit symbol must be cleared to “0”.)	
P1PU2	Selects connecting or disconnecting a built-in pull-up resister.	0: 1:	Refer to Table 8-11.	Reserved (this bit symbol must be cleared to “0”.)	
P1PU1	Selects connecting or disconnecting a built-in pull-up resister.	0: 1:	Refer to Table 8-11.		
P1PU0	Selects connecting or disconnecting a built-in pull-up resister.	0: 1:	Refer to Table 8-10.		

Table 8-10 P10 and Built-in Pull-up Resister

Port function/ Secondary function	Connected/ Disconnected	Setting Condition		
		<P1PU0>	SYSCR3 <RSTDIS>	<P1CR0>
Port Input mode	Disconnected	0	1	0
	Connected	1		
Port Output mode	Disconnected	*		1
$\overline{\text{RESET}}$ pin	Connected	1	0	0

Note: *: Don’t care

Table 8-11 P1i and Built-in Pull-up resister

Port function/ Secondary function	Connected/ Disconnected	Setting Condition	
		<P1PUi>	<P1CRi>
Port Input mode	Disconnected	0	0
	Connected	1	
Port Output mode	Disconnected	*	1

Note 1: *: Don’t care

Note 2: TMP89FS60B: i = 3 to 1
TMP89FS62B, TMP89FS63B: i = 1

Port P1 input data

P1PRD		7	6	5	4	3	2	1	0
(0x000E)	Bit Symbol	-	-	-	-	P1PRD3	P1PRD2	P1PRD1	P1PRD0
	Read/Write	R	R	R	R	R	R	R	R
	After reset	0	0	0	0	*	*	*	*

		TMP89FS60B	TMP89FS62B	TMP89FS63B
P1PRD3	Input data from port	Refer to Table 8-12.	The read data is “0”.	
P1PRD2	Input data from port	Refer to Table 8-12.	The read data is “0”.	
P1PRD1	Input data from port	Refer to Table 8-12.		
P1PRD0	Input data from port	Refer to Table 8-12.		

Table 8-12 Read Value of <P1PRDi>

Read value of <P0PRDj>	Setting Condition
	<P1CRi>
0	1
Level of the port pin	0

Note: TMP89FS60B: i = 3 to 0
TMP89FS62B, TMP89FS63B: i = 1 to 0

8.4.3. Port P2

Port P2 is an 8-bit input/output port that can be set to input or output for each bit individually, and it is also used as the serial bus interface input/output, the serial interface input/output, the UART input/output and the on-chip debug function.

The output circuit has the P-channel output control function and either the open-drain output or the CMOS output can be selected.

Port P2 contains a programmable pull-up resistor on the VDD pin side. This pull-up resistor can be used when the port is used in the input mode or as an open-drain output in the output mode.

When this port is used as the serial bus interface, the serial interface or the UART, setting for serial interface selecting function is also needed. For details, refer to "8.5. Serial Interface Selecting Function".

For the on-chip debug function, refer to "23. On-chip Debug Function (OCD)".

Table 8-13 Port P2

	P27	P26	P25	P24	P23	P22	P21	P20
Secondary function	-	-	SCLK0 (Note)	SCL0 SI0 (Note)	SDA0 SO0 (Note)	SCLK0	RXD0 TXD0 SI0 OCDDIO	TXD0 RXD0 SO0 OCDCK

Note: The TMP89FS62B does not have the serial interface input/output (SCLK0, SI0/SCL0 and SO0/SDA0).

8.4.3.1. P2 Port for Each Product

P2 port for each product is shown in Table 8-14.

In regards to unavailable bits of the P2 port, the value which must be set or cleared to the corresponding bit of each register is shown in “8.4.3.2. Control”.

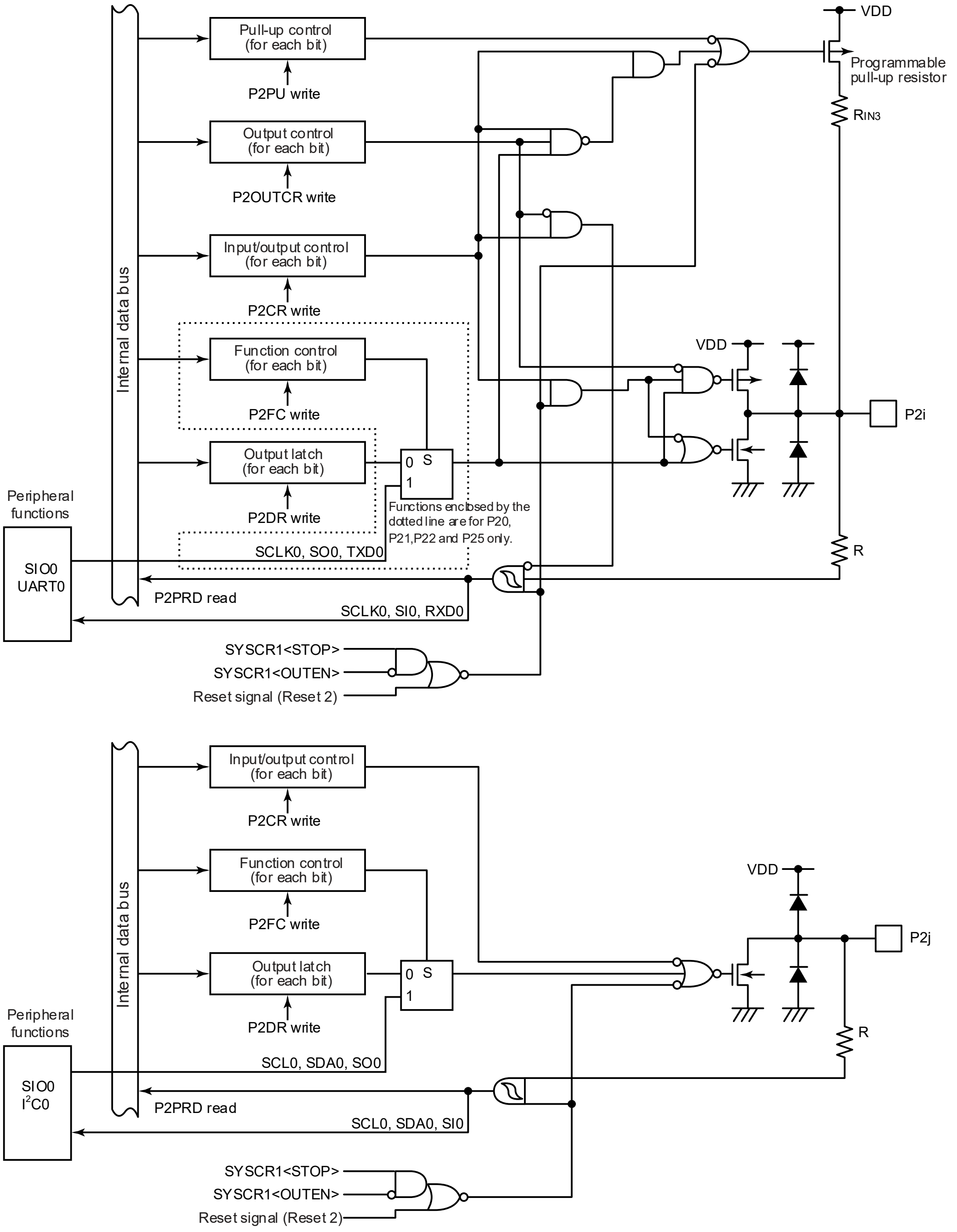
These values must be set or cleared to corresponding bit of each register in the setup routine.

And also, when the value of the register which includes these bits is modified, don’t change the set or cleared value of these bit.

Table 8-14 P2 Port for Each Product

	TMP89FS60B	TMP89FS62B	TMP89FS63B
P27	A	NA	NA
P26	A	NA	NA
P25	A	NA	A
P24	A	NA	A
P23	A	A	A
P22	A	A	A
P21	A	A	A
P20	A	A	A

Note: A: Available, NA: Not available



Note 1: R = 100 [Ω] (typ.)

Note 2: R_{IN3} = 50 [kΩ] (typ.)

Note 3: TMP89FS60B: i = 7 to 5, 2 to 0
TMP89FS62B: i = 2 to 0
TMP89FS63B: i = 5, 2 to 0

Note 4: TMP89FS60B, TMP89FS63B: j = 4 to 3
TMP89FS62B: j = 3

Figure 8-4 Port P2

8.4.3.2. Control

P2 port is controlled by the following registers.

Port P2 output latch

P2DR		7	6	5	4	3	2	1	0
(0x0002)	Bit Symbol	P27	P26	P25	P24	P23	P22	P21	P20
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

			TMP89FS60B	TMP89FS62B	TMP89FS63B
P27	Output level in the output mode	0: 1:	Refer to Table 8-15.	Reserved (this bit symbol must be cleared to “0”.)	
P26	Output level in the output mode	0: 1:	Refer to Table 8-15.	Reserved (this bit symbol must be cleared to “0”.)	
P25	Output level in the output mode	0: 1:	Refer to Table 8-15.	Reserved (this bit symbol must be cleared to “0”.)	Refer to Table 8-15.
P24	Output level in the output mode	0: 1:	“Low” level “Hi-Z”	Reserved (this bit symbol must be cleared to “0”.)	“Low” level “Hi-Z”
P23	Output level in the output mode	0: 1:	“Low” level “Hi-Z”		
P22	Output level in the output mode	0: 1:	Refer to Table 8-15.		
P21	Output level in the output mode	0: 1:	Refer to Table 8-15.		
P20	Output level in the output mode	0: 1:	Refer to Table 8-15.		

Table 8-15 Output Status of I/O Pin in the Output Mode

Port Function	Output status of I/O pin	Setting Condition		
		<P2i>	<P2OUTCRi>	<P2PUi>
Port Output mode CMOS output	“Low” level	0	0	*
	“High” level	1		
Port Output mode Open-drain output	“Low” level	0	1	0
	“Hi-Z”	1		
	“Low” level	0		1
	“Pull-up”	1		

Note 1: *: Don’t care

Note 2: TMP89FS60B: i = 7 to 5, 2 to 0
TMP89FS62B: i = 2 to 0
TMP89FS63B: i = 5, 2 to 0

Port P2 input/output control

P2CR		7	6	5	4	3	2	1	0
(0x0F1C)	Bit Symbol	P2CR7	P2CR6	P2CR5	P2CR4	P2CR3	P2CR2	P2CR1	P2CR0
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

			TMP89FS60B	TMP89FS62B	TMP89FS63B
P2CR7	Selects input mode or output mode.	0:	Input mode	Reserved (this bit symbol must be set to “1”).	
		1:	Output mode		
P2CR6	Selects input mode or output mode.	0:	Input mode	Reserved (this bit symbol must be set to “1”).	
		1:	Output mode		
P2CR5	Selects input mode or output mode.	0:	Input mode SCLK0 (I)	Reserved (this bit symbol must be set to “1”.	Input mode SCLK0 (I)
		1:	Output mode SCLK0 (O)		Output mode SCLK0 (O)
P2CR4	Selects input mode or output mode.	0:	Input mode SI0 (I)	Reserved (this bit symbol must be set to “1”.	Input mode SI0 (I)
		1:	Output mode SCL0 (I/O)		Output mode SCL0 (I/O)
P2CR3	Selects input mode or output mode.	0:	Input mode		
		1:	Output mode SDA0 (I/O) SO0 (O)		
P2CR2	Selects input mode or output mode.	0:	Input mode SCLK0 (I)		
		1:	Output mode SCLK0 (O)		
P2CR1	Selects input mode or output mode.	0:	Input mode RXD0 (I) SI0 (I)		
		1:	Output mode TXD0 (O)		
P2CR0	Selects input mode or output mode.	0:	Input mode RXD0 (I)		
		1:	Output mode TXD0 (O) SO0 (O)		

Note : Symbol "I" means the secondary function input, Symbol “O” means secondary function output, Symbol “I/O” means secondary function input/output

Port P2 function control

P2FC		7	6	5	4	3	2	1	0
(0x0F36)	Bit Symbol	-	-	P2FC5	P2FC4	P2FC3	P2FC2	P2FC1	P2FC0
	Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

			TMP89FS60B	TMP89FS62B	TMP89FS63B
P2FC5	Port function	0:	I/O port	Reserved (this bit symbol must be cleared to “0”.)	I/O port
		1:	SCLK0 (O)		SCLK0 (O)
P2FC4	Port function	0:	I/O port	Reserved (this bit symbol must be cleared to “0”.)	I/O port
		1:	SCL0 (I/O)		SCL0 (I/O)
P2FC3	Port function	0:	I/O port	Reserved (this bit symbol must be cleared to “0”.)	I/O port
		1:	SDA0 (I/O) SO0 (O)		SDA0 (I/O) SO0 (O)
P2FC2	Port function	0:	I/O port		
		1:	SCLK0 (O)		
P2FC1	Port function	0:	I/O port		
		1:	TXD0 (O)		
P2FC0	Port function	0:	I/O port		
		1:	TXD0 (O) SO0 (O)		

Note : Symbol “O” means secondary function output, Symbol “I/O” means secondary function input/output

Port P2 output control

P2OUTCR		7	6	5	4	3	2	1	0
(0x0F43)	Bit Symbol	P2OUT7	P2OUT6	P2OUT5	-	-	P2OUT2	P2OUT1	P2OUT0
	Read/Write	R/W	R/W	R/W	R	R	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

			TMP89FS60B	TMP89FS62B	TMP89FS63B
P2OUT7	Output circuit type of port	0: 1:	CMOS output Open-drain output	Reserved (this bit symbol must be cleared to “0”.)	
P2OUT6	Output circuit type of port	0: 1:	CMOS output Open-drain output	Reserved (this bit symbol must be cleared to “0”.)	
P2OUT5	Output circuit type of port	0: 1:	CMOS output Open-drain output	Reserved (this bit symbol must be cleared to “0”.)	CMOS output Open-drain output
P2OUT2	Output circuit type of port	0: 1:	CMOS output Open-drain output		
P2OUT1	Output circuit type of port	0: 1:	CMOS output Open-drain output		
P2OUT0	Output circuit type of port	0: 1:	CMOS output Open-drain output		

Port P2 built-in pull-up resistor control

P2PU		7	6	5	4	3	2	1	0
(0x0F29)	Bit Symbol	P2PU7	P2PU6	P2PU5	-	-	P2PU2	P2PU1	P2PU0
	Read/Write	R/W	R/W	R/W	R	R	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

			TMP89FS60B	TMP89FS62B	TMP89FS63B
P2PU7	Selects connecting or disconnecting a built-in pull-up resister.	0: 1:	Refer to Table 8-16.	Reserved (this bit symbol must be cleared to “0”.)	
P2PU6	Selects connecting or disconnecting a built-in pull-up resister.	0: 1:	Refer to Table 8-16.	Reserved (this bit symbol must be cleared to “0”.)	
P2PU5	Selects connecting or disconnecting a built-in pull-up resister.	0: 1:	Refer to Table 8-16.	Reserved (this bit symbol must be cleared to “0”.)	Refer to Table 8-16.
P2PU2	Selects connecting or disconnecting a built-in pull-up resister.	0: 1:	Refer to Table 8-16.		
P2PU1	Selects connecting or disconnecting a built-in pull-up resister.	0: 1:	Refer to Table 8-16.		
P2PU0	Selects connecting or disconnecting a built-in pull-up resister.	0: 1:	Refer to Table 8-16.		

Table 8-16 Port Function and Built-in Pull-up Resister

Port Function	Connected/ Disconnected	<P2PUi>	<P2CRi>	< P2OUTi>	<P2i>
Input mode	Not connected	0	0	*	*
	Connected	1			
Output mode CMOS output	Not connected	*	1	0	
Output mode Open-drain output	Not connected	0	1	1	0
	Connected	1			1

Note 1: *: Don’t care

Note 2: TMP89FS60B: i = 7 to 5, 2 to 0
TMP89FS62B: i = 2 to 0
TMP89FS63B: i = 5, 2 to 0

Port P2 input data

P2PRD		7	6	5	4	3	2	1	0
(0x000F)	Bit Symbol	P2PRD7	P2PRD6	P2PRD5	P2PRD4	P2PRD3	P2PRD2	P2PRD1	P2PRD0
	Read/Write	R	R	R	R	R	R	R	R
	After reset	*	*	*	*	*	*	*	*

		TMP89FS60B	TMP89FS62B	TMP89FS63B
P2PRD7	Input data from port	Refer to Table 8-17.	The read data is “0”.	
P2PRD6	Input data from port	Refer to Table 8-17.	The read data is “0”.	
P2PRD5	Input data from port	Refer to Table 8-17.	The read data is “0”.	Refer to Table 8-17.
P2PRD4	Input data from port	The level of port pin is read.	The read data is “0”.	The level of port pin is read.
P2PRD3	Input data from port	The level of port pin is read.		
P2PRD2	Input data from port	Refer to Table 8-17.		
P2PRD1	Input data from port	Refer to Table 8-17.		
P2PRD0	Input data from port	Refer to Table 8-17.		

Table 8-17 Read Value of <P2PRDi>

Read value of <P2PRDi>	Setting Condition	
	<P2CRi>	<P2OUTCRi>
Level of the port pin	0	*
0	1	0
Level of the port pin		1

Note 1: * : Don’t care

Note 2: TMP89FS60B: i = 7 to 5, 2 to 0
TMP89FS62B: i = 2 to 0
TMP89FS63B: i = 5, 2 to 0

8.4.4. Port P4

Port P4 is an 8-bit input/output port that can be set to input or output for each bit individually, and it is also used as the analog input of the AD convertor and the key-on wakeup input.

Port P4 contains a programmable pull-up resistor on the VDD pin side. This pull-up resistor can be used when the port is used in the input mode and is used as key-on wakeup input.

Table 8-18 Port P4

	P47	P46	P45	P44	P43	P42	P41	P40
Secondary function	AIN7 KWI7	AIN6 KWI6	AIN5 KWI5	AIN4 KWI4	AIN3 KWI3	AIN2 KWI2	AIN1 KWI1	AIN0 KWI0

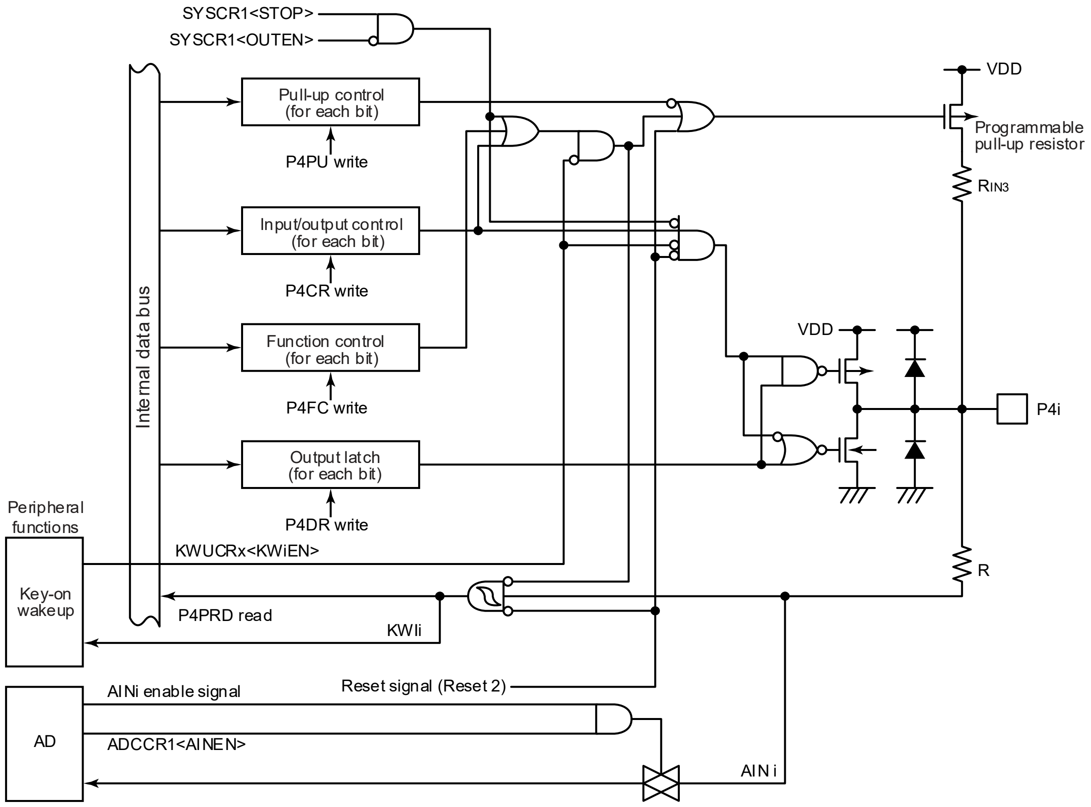
8.4.4.1. P4 Port for Each Product

P4 port for each product is shown in Table 8-19.
P4 port can be used on all products.

Table 8-19 P4 Port for Each Product

	TMP89FS60B	TMP89FS62B	TMP89FS63B
P47	A	A	A
P46	A	A	A
P45	A	A	A
P44	A	A	A
P43	A	A	A
P42	A	A	A
P41	A	A	A
P40	A	A	A

Note: A: Available



Note 1: R = 100 [Ω] (typ.)

Note 2: R_{IN3} = 50 [kΩ] (typ.)

Note 3: i = 7 to 0

Note 4: x = 1 to 0

Figure 8-5 Port P4

8.4.4.2. Control

P4 port is controlled by the following registers.

Port P4 output latch

P4DR		7	6	5	4	3	2	1	0
(0x0004)	Bit Symbol	P47	P46	P45	P44	P43	P42	P41	P40
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

			TMP89FS60B	TMP89FS62B	TMP89FS63B
P47	Output level in the output mode	0: 1:	“Low” level “High” level		
P46	Output level in the output mode	0: 1:	“Low” level “High” level		
P45	Output level in the output mode	0: 1:	“Low” level “High” level		
P44	Output level in the output mode	0: 1:	“Low” level “High” level		
P43	Output level in the output mode	0: 1:	“Low” level “High” level		
P42	Output level in the output mode	0: 1:	“Low” level “High” level		
P41	Output level in the output mode	0: 1:	“Low” level “High” level		
P40	Output level in the output mode	0: 1:	“Low” level “High” level		

Port P4 input/output control

P4CR (0x0F1E)		7	6	5	4	3	2	1	0
	Bit Symbol	P4CR7	P4CR6	P4CR5	P4CR4	P4CR3	P4CR2	P4CR1	P4CR0
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

			TMP89FS60B	TMP89FS62B	TMP89FS63B
P4CR7	Selects input mode or output mode.	0: 1:	Refer to Table 8-20.		
P4CR6	Selects input mode or output mode.	0: 1:	Refer to Table 8-20.		
P4CR5	Selects input mode or output mode.	0: 1:	Refer to Table 8-20.		
P4CR4	Selects input mode or output mode.	0: 1:	Refer to Table 8-20.		
P4CR3	Selects input mode or output mode.	0: 1:	Refer to Table 8-20.		
P4CR2	Selects input mode or output mode.	0: 1:	Refer to Table 8-20.		
P4CR1	Selects input mode or output mode.	0: 1:	Refer to Table 8-20.		
P4CR0	Selects input mode or output mode.	0: 1:	Refer to Table 8-20.		

Table 8-20 <P4CRi> and Input/output Mode of Each Pin

Port Function/ Shared Function	Setting Condition	
	<P4CRi>	KWUCRx<KWiEN>
Port Input mode/ AINi	0	0
Port Output mode	1	
Port Input mode Key-on wakeup input	*	1

Note 1: *: Don’t care

Note 2: i = 7 to 0

Note 3: x = 1 to 0

Port P4 function control

P4FC		7	6	5	4	3	2	1	0
(0x0F38)	Bit Symbol	P4FC7	P4FC6	P4FC5	P4FC4	P4FC3	P4FC2	P4FC1	P4FC0
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

			TMP89FS60B	TMP89FS62B	TMP89FS63B
P4FC7	Port function	0: 1:	I/O port AIN7 (I)		
P4FC6	Port function	0: 1:	I/O port AIN6 (I)		
P4FC5	Port function	0: 1:	I/O port AIN5 (I)		
P4FC4	Port function	0: 1:	I/O port AIN4 (I)		
P4FC3	Port function	0: 1:	I/O port AIN3 (I)		
P4FC2	Port function	0: 1:	I/O port AIN2 (I)		
P4FC1	Port function	0: 1:	I/O port AIN1 (I)		
P4FC0	Port function	0: 1:	I/O port AIN0 (I)		

Note 1: Symbol "I" means the secondary function input

Note 2: When P4i (i = 7 to 0) port is used as an analog input of the AD convertor (<P4FCi> = 1 (i = 7 to 0)), the input gate of I/O port is disabled to block the through current in the input gate.

Note 3: When P4i (i = 7 to 0) port is used as a key-on wakeup input (KWUCRx<KWiEN> = 1 (x = 1 to 0, i = 7 to 0)), the input gate of I/O port is enabled regardless of setting <P4FCi> (i = 7 to 0). Therefore, when P4i (i = 7 to 0) port is used as a key-on wakeup input and an analog input of the AD convertor, the through current may flow in the input gate.

Port P4 built-in pull-up resistor control

P4PU		7	6	5	4	3	2	1	0
(0x0F2B)	Bit Symbol	P4PU7	P4PU6	P4PU5	P4PU4	P4PU3	P4PU2	P4PU1	P4PU0
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

			TMP89FS60B	TMP89FS62B	TMP89FS63B
P4PU7	Selects connecting or disconnecting a built-in pull-up resister.	0: 1:	Refer to Table 8-21.		
P4PU6	Selects connecting or disconnecting a built-in pull-up resister.	0: 1:	Refer to Table 8-21.		
P4PU5	Selects connecting or disconnecting a built-in pull-up resister.	0: 1:	Refer to Table 8-21.		
P4PU4	Selects connecting or disconnecting a built-in pull-up resister.	0: 1:	Refer to Table 8-21.		
P4PU3	Selects connecting or disconnecting a built-in pull-up resister.	0: 1:	Refer to Table 8-21.		
P4PU2	Selects connecting or disconnecting a built-in pull-up resister.	0: 1:	Refer to Table 8-21.		
P4PU1	Selects connecting or disconnecting a built-in pull-up resister.	0: 1:	Refer to Table 8-21.		
P4PU0	Selects connecting or disconnecting a built-in pull-up resister.	0: 1:	Refer to Table 8-21.		

Table 8-21 Port and Secondary Function and Built-in Pull-up Resister

Port function/ Secondary function	Connected/ Disconnected	Setting Condition			
		<P4PUi>	KWUCRx <KWiEN>	<P4CRi>	<P4FCi>
Port Input mode	Not connected	0	0	0	0
Port Input mode	Connected	1			
AINi	Not connected	*			1
Port Output mode	Not connected	*		1	*
Key-on wakeup input	Not connected	0	1	*	*
Key-on wakeup input	Connected	1			

Note 1: *: Don’t care

Note 2: i = 7 to 0

Note 3: x = 1 to 0

Port P4 input data

P4PRD		7	6	5	4	3	2	1	0
(0x0011)	Bit Symbol	P4PRD7	P4PRD6	P4PRD5	P4PRD4	P4PRD3	P4PRD2	P4PRD1	P4PRD0
	Read/Write	R	R	R	R	R	R	R	R
	After reset	*	*	*	*	*	*	*	*

		TMP89FS60B	TMP89FS62B	TMP89FS63B
P4PRD7	Input data from port	Refer to Table 8-22.		
P4PRD6	Input data from port	Refer to Table 8-22.		
P4PRD5	Input data from port	Refer to Table 8-22.		
P4PRD4	Input data from port	Refer to Table 8-22.		
P4PRD3	Input data from port	Refer to Table 8-22.		
P4PRD2	Input data from port	Refer to Table 8-22.		
P4PRD1	Input data from port	Refer to Table 8-22.		
P4PRD0	Input data from port	Refer to Table 8-22.		

Table 8-22 Read Value of <P4PRDi>

Read Value of <P4PRDi>	Setting Condition		
	KWUCRx <KWiEN>	<P4CRi>	<P4FCi>
Level of the port pin	0	0	0
0		*	1
0		1	*
Level of the port pin	1	*	*

Note 1: *: Don’t care
Note 2: i = 7 to 0
Note 3: x = 1 to 0

8.4.5. Port P5

Port P5 is an 8-bit input/output port that can be set to input or output for each bit individually, and it is also used as the analog input of AD converter.

Table 8-23　Port P5

	P57	P56	P55	P54	P53	P52	P51	P50
Secondary function	AIN15 (Note)	AIN14 (Note)	AIN13 (Note)	AIN12	AIN11	AIN10	AIN9	AIN8

Note: The TMP89FS63B does not have the analog input (AIN15 to 13) of AD converter.

8.4.5.1. P5 Port for Each Product

P5 port for each product is shown in Table 8-24.

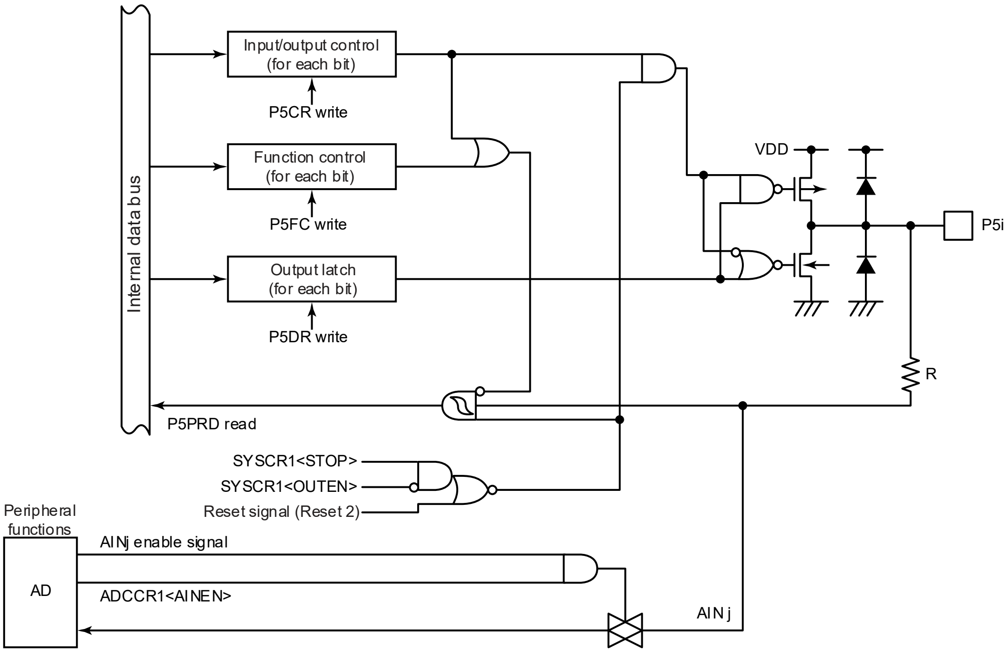
In regards to unavailable bits of the P5 port, the value which must be set or cleared to the corresponding bit of each register is shown in“8.4.5.2. Control”. These values must be set or cleared to corresponding bit of each register in the setup routine.

And also, when the value of the register which includes these bits is modified, don’t change the set or cleared value of these bit.

Table 8-24　P5 Port for Each Product

	TMP89FS60B	TMP89FS62B	TMP89FS63B
P57	A	NA	NA
P56	A	NA	NA
P55	A	NA	NA
P54	A	NA	A
P53	A	NA	A
P52	A	NA	A
P51	A	NA	A
P50	A	NA	A

Note: A: Available, NA: Not available



Note 1: R = 100 [Ω] (typ.)

Note 2: TMP89FS60B: i = 7 to 0
TMP89FS63B: i = 4 to 0

Note 3: TMP89FS60B: j = 15 to 8
TMP89FS63B: j = 12 to 8

Figure 8-6 Port P5

8.4.5.2. Control

P5 port is controlled by the following registers.

Port P5 output latch

P5DR		7	6	5	4	3	2	1	0
(0x0005)	Bit Symbol	P57	P56	P55	P54	P53	P52	P51	P50
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

			TMP89FS60B	TMP89FS62B	TMP89FS63B
P57	Output level in the output mode	0:	"Low" level	Reserved (this bit symbol must be cleared to "0".)	
		1:	"High" level		
P56	Output level in the output mode	0:	"Low" level	Reserved (this bit symbol must be cleared to "0".)	
		1:	"High" level		
P55	Output level in the output mode	0:	"Low" level	Reserved (this bit symbol must be cleared to "0".)	
		1:	"High" level		
P54	Output level in the output mode	0:	"Low" level	Reserved (this bit symbol must be cleared to "0".)	"Low" level
		1:	"High" level		"High" level
P53	Output level in the output mode	0:	"Low" level	Reserved (this bit symbol must be cleared to "0".)	"Low" level
		1:	"High" level		"High" level
P52	Output level in the output mode	0:	"Low" level	Reserved (this bit symbol must be cleared to "0".)	"Low" level
		1:	"High" level		"High" level
P51	Output level in the output mode	0:	"Low" level	Reserved (this bit symbol must be cleared to "0".)	"Low" level
		1:	"High" level		"High" level
P50	Output level in the output mode	0:	"Low" level	Reserved (this bit symbol must be cleared to "0".)	"Low" level
		1:	"High" level		"High" level

Port P5 input/output control

P5CR		7	6	5	4	3	2	1	0
(0x0F1F)	Bit Symbol	P5CR7	P5CR6	P5CR5	P5CR4	P5CR3	P5CR2	P5CR1	P5CR0
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

			TMP89FS60B	TMP89FS62B	TMP89FS63B
P5CR7	Selects input mode or output mode.	0:	Input mode AIN15 (I)	Reserved (this bit symbol must be set to “1”.)	
		1:	Output mode		
P5CR6	Selects input mode or output mode.	0:	Input mode AIN14 (I)	Reserved (this bit symbol must be set to “1”.)	
		1:	Output mode		
P5CR5	Selects input mode or output mode.	0:	Input mode AIN13 (I)	Reserved (this bit symbol must be set to “1”.)	
		1:	Output mode		
P5CR4	Selects input mode or output mode.	0:	Input mode AIN12 (I)	Reserved (this bit symbol must be set to “1”.)	Input mode AIN12 (I)
		1:	Output mode		Output mode
P5CR3	Selects input mode or output mode.	0:	Input mode AIN11 (I)	Reserved (this bit symbol must be set to “1”.)	Input mode AIN11 (I)
		1:	Output mode		Output mode
P5CR2	Selects input mode or output mode.	0:	Input mode AIN10 (I)	Reserved (this bit symbol must be set to “1”.)	Input mode AIN10 (I)
		1:	Output mode		Output mode
P5CR1	Selects input mode or output mode.	0:	Input mode AIN9 (I)	Reserved (this bit symbol must be set to “1”.)	Input mode AIN9 (I)
		1:	Output mode		Output mode
P5CR0	Selects input mode or output mode.	0:	Input mode AIN8 (I)	Reserved (this bit symbol must be set to “1”.)	Input mode AIN8 (I)
		1:	Output mode		Output mode

Note: Symbol "I" means the secondary function input

Port P5 function control

P5FC		7	6	5	4	3	2	1	0
(0x0F39)	Bit Symbol	P5FC7	P5FC6	P5FC5	P5FC4	P5FC3	P5FC2	P5FC1	P5FC0
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

			TMP89FS60B	TMP89FS62B	TMP89FS63B
P5FC7	Port function	0: 1:	I/O port AIN15 (I)	Reserved (this bit symbol must be cleared to "0".)	
P5FC6	Port function	0: 1:	I/O port AIN14 (I)	Reserved (this bit symbol must be cleared to "0".)	
P5FC5	Port function	0: 1:	I/O port AIN13 (I)	Reserved (this bit symbol must be cleared to "0".)	
P5FC4	Port function	0: 1:	I/O port AIN12 (I)	Reserved (this bit symbol must be cleared to "0".)	I/O port AIN12 (I)
P5FC3	Port function	0: 1:	I/O port AIN11 (I)	Reserved (this bit symbol must be cleared to "0".)	I/O port AIN11 (I)
P5FC2	Port function	0: 1:	I/O port AIN10 (I)	Reserved (this bit symbol must be cleared to "0".)	I/O port AIN10 (I)
P5FC1	Port function	0: 1:	I/O port AIN9 (I)	Reserved (this bit symbol must be cleared to "0".)	I/O port AIN9 (I)
P5FC0	Port function	0: 1:	I/O port AIN8 (I)	Reserved (this bit symbol must be cleared to "0".)	I/O port AIN8 (I)

Note 1: Symbol "I" means the secondary function input

Note 2: When P5i port is used as an analog input of the AD convertor (<P5FCi> = 1), the input gate of I/O port is disabled to block the through current in the input gate (TMP89FS60B: i = 7 to 0, TMP89FS63B: i = 4 to 0).

Port P5 input data

P5PRD		7	6	5	4	3	2	1	0
(0x0012)	Bit Symbol	P5PRD7	P5PRD6	P5PRD5	P5PRD4	P5PRD3	P5PRD2	P5PRD1	P5PRD0
	Read/Write	R	R	R	R	R	R	R	R
	After reset	*	*	*	*	*	*	*	*

		TMP89FS60B	TMP89FS62B	TMP89FS63B
P5PRD7	Input data from port	Refer to Table 8-25.	The read data is “0”.	
P5PRD6	Input data from port	Refer to Table 8-25.	The read data is “0”.	
P5PRD5	Input data from port	Refer to Table 8-25.	The read data is “0”.	
P5PRD4	Input data from port	Refer to Table 8-25.	The read data is “0”.	Refer to Table 8-25.
P5PRD3	Input data from port	Refer to Table 8-25.	The read data is “0”.	Refer to Table 8-25.
P5PRD2	Input data from port	Refer to Table 8-25.	The read data is “0”.	Refer to Table 8-25.
P5PRD1	Input data from port	Refer to Table 8-25.	The read data is “0”.	Refer to Table 8-25.
P5PRD0	Input data from port	Refer to Table 8-25.	The read data is “0”.	Refer to Table 8-25.

Table 8-25 Read Value of <P5PRDi>

Read Value of <P5PRDi>	Setting Condition	
	<P5CRi>	<P5FCi>
Level of the port pin	0	0
0	*	1
0	1	*

Note 1: *: Don’t care

Note 2: TMP89FS60B: i = 7 to 0
TMP89FS63B: i = 4 to 0

8.4.6. Port P7

Port P7 is an 8-bit input/output port that can be set to input or output for each bit individually, and it is also used as the external interrupt input, the divider output and the timer counter input/output.

Table 8-26 Port P7

	P77	P76	P75	P74	P73	P72	P71	P70
Secondary function	INT4	INT3	INT2	$\overline{\text{DVO}}$	TCA1 PPGA1	TCA0 PPGA0	TC01 PPG01 PWM01	TC00 PPG00 PWM00

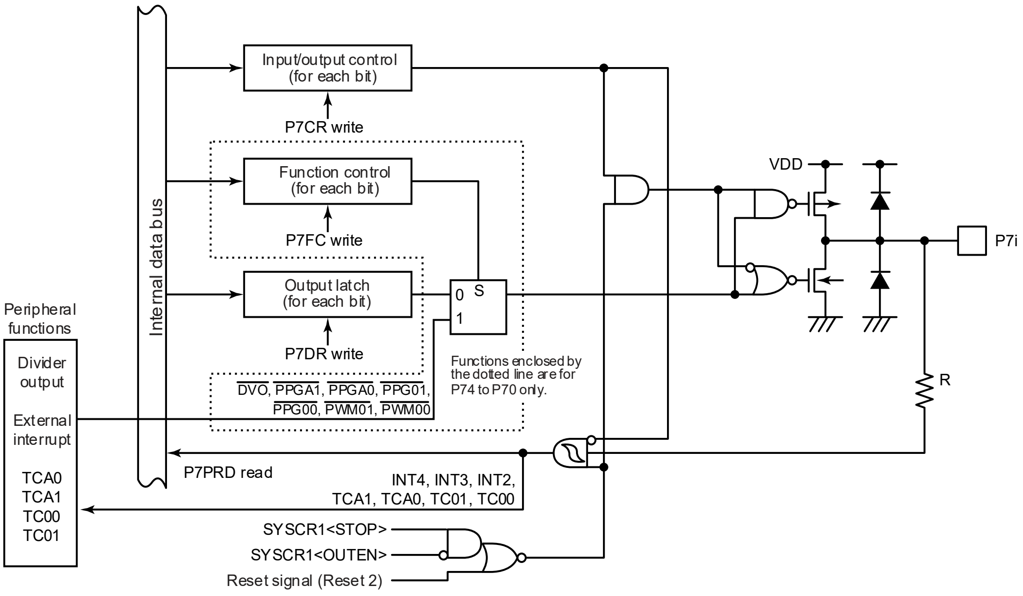
8.4.6.1. P7 Port for Each Product

P7 port for each product is shown in Table 8-27.
P7 port can be used on all products.

Table 8-27 P7 Port for Each Product

	TMP89FS60B	TMP89FS62B	TMP89FS63B
P77	A	A	A
P76	A	A	A
P75	A	A	A
P74	A	A	A
P73	A	A	A
P72	A	A	A
P71	A	A	A
P70	A	A	A

Note: A: Available


Internal data bus
Input/output control (for each bit)
P7CR write
Function control (for each bit)
P7FC write
Output latch (for each bit)
P7DR write
0
S
1
$\overline{\text{DVO}}$, $\overline{\text{PPGA1}}$, $\overline{\text{PPGA0}}$, $\overline{\text{PPG01}}$, $\overline{\text{PPG00}}$, $\overline{\text{PWM01}}$, $\overline{\text{PWM00}}$
Functions enclosed by the dotted line are for P74 to P70 only.
Peripheral functions
Divider output
External interrupt
TCA0
TCA1
TC00
TC01
P7PRD read
INT4, INT3, INT2, TCA1, TCA0, TC01, TC00
SYSCR1<STOP>
SYSCR1<OUTEN>
Reset signal (Reset 2)
VDD
P7i
R

Note 1: R = 100 [Ω] (typ.)

Note 2: i = 7 to 0

Figure 8-7 Port P7

8.4.6.2. Control

P7 port is controlled by the following registers.

Port P7 output latch

P7DR	7	6	5	4	3	2	1	0
(0x0007) Bit Symbol	P77	P76	P75	P74	P73	P72	P71	P70
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

			TMP89FS60B	TMP89FS62B	TMP89FS63B
P77	Output level in the output mode	0:	“Low” level		
		1:	“High” level		
P76	Output level in the output mode	0:	“Low” level		
		1:	“High” level		
P75	Output level in the output mode	0:	“Low” level		
		1:	“High” level		
P74	Output level in the output mode	0:	“Low” level		
		1:	“High” level		
P73	Output level in the output mode	0:	“Low” level		
		1:	“High” level		
P72	Output level in the output mode	0:	“Low” level		
		1:	“High” level		
P71	Output level in the output mode	0:	“Low” level		
		1:	“High” level		
P70	Output level in the output mode	0:	“Low” level		
		1:	“High” level		

Port P7 input/output control

P7CR		7	6	5	4	3	2	1	0
(0x0F21)	Bit Symbol	P7CR7	P7CR6	P7CR5	P7CR4	P7CR3	P7CR2	P7CR1	P7CR0
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

			TMP89FS60B	TMP89FS62B	TMP89FS63B
P7CR7	Selects input mode or output mode.	0: 1:	Input mode INT4 (I) Output mode		
P7CR6	Selects input mode or output mode.	0: 1:	Input mode INT3 (I) Output mode		
P7CR5	Selects input mode or output mode.	0: 1:	Input mode INT2 (I) Output mode		
P7CR4	Selects input mode or output mode.	0: 1:	Input mode Output mode $\overline{\text{DVO}}$ (O)		
P7CR3	Selects input mode or output mode.	0: 1:	Input mode TCA1 (I) Output mode $\overline{\text{PPGA1}}$ (O)		
P7CR2	Selects input mode or output mode.	0: 1:	Input mode TCA0 (I) Output mode $\overline{\text{PPGA0}}$ (O)		
P7CR1	Selects input mode or output mode.	0: 1:	Input mode TC01 (I) Output mode $\overline{\text{PPG01}}$ (O) $\overline{\text{PWM01}}$ (O)		
P7CR0	Selects input mode or output mode.	0: 1:	Input mode TC00 (I) Output mode $\overline{\text{PPG00}}$ (O) $\overline{\text{PWM00}}$ (O)		

Note: Symbol "I" means secondary function input. Symbol "O" means secondary function output.

Port P7 function control

P7FC		7	6	5	4	3	2	1	0
(0x0F3B)	Bit Symbol	-	-	-	P7FC4	P7FC3	P7FC2	P7FC1	P7FC0
	Read/Write	R	R	R	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

			TMP89FS60B	TMP89FS62B	TMP89FS63B
P7FC4	Port function	0:	I/O port		
		1:	$\overline{\text{DVO}}$ (O)		
P7FC3	Port function	0:	I/O port		
		1:	$\overline{\text{PPGA1}}$ (O)		
P7FC2	Port function	0:	I/O port		
		1:	$\overline{\text{PPGA0}}$ (O)		
P7FC1	Port function	0:	I/O port		
		1:	$\overline{\text{PPG01}}$ (O) $\overline{\text{PWM01}}$ (O)		
P7FC0	Port function	0:	I/O port		
		1:	$\overline{\text{PPG00}}$ (O) $\overline{\text{PWM00}}$ (O)		

Note: Symbol "O" means secondary function output.

Port P7 input data

P7PRD		7	6	5	4	3	2	1	0
(0x0014)	Bit Symbol	P7PRD7	P7PRD6	P7PRD5	P7PRD4	P7PRD3	P7PRD2	P7PRD1	P7PRD0
	Read/Write	R	R	R	R	R	R	R	R
	After reset	*	*	*	*	*	*	*	*

		TMP89FS60B	TMP89FS62B	TMP89FS63B
P7PRD7	Input data from port	Refer to Table 8-28.		
P7PRD6	Input data from port	Refer to Table 8-28.		
P7PRD5	Input data from port	Refer to Table 8-28.		
P7PRD4	Input data from port	Refer to Table 8-28.		
P7PRD3	Input data from port	Refer to Table 8-28.		
P7PRD2	Input data from port	Refer to Table 8-28.		
P7PRD1	Input data from port	Refer to Table 8-28.		
P7PRD0	Input data from port	Refer to Table 8-28.		

Table 8-28 Read Value of <P7PRDi>

Read Value of <P7PRDi>	Setting Condition
	<P7CRi>
Level of the port pin	0
0	1

Note: i= 7 to 0

8.4.7. Port P8

Port P8 is a 5-bit input/output port that can be set to input or output for each bit individually, and it is also used as the timer counter input/output.

Table 8-29 Port P8

	-	-	-	P84	P83	P82	P81	P80
Secondary function	-	-	-	-	-	-	TC03 $\overline{\text{PPG03}}$ $\overline{\text{PWM03}}$	TC02 $\overline{\text{PPG02}}$ $\overline{\text{PWM02}}$

8.4.7.1. P8 Port for Each Product

P8 port for each product is shown in Table 8-30.

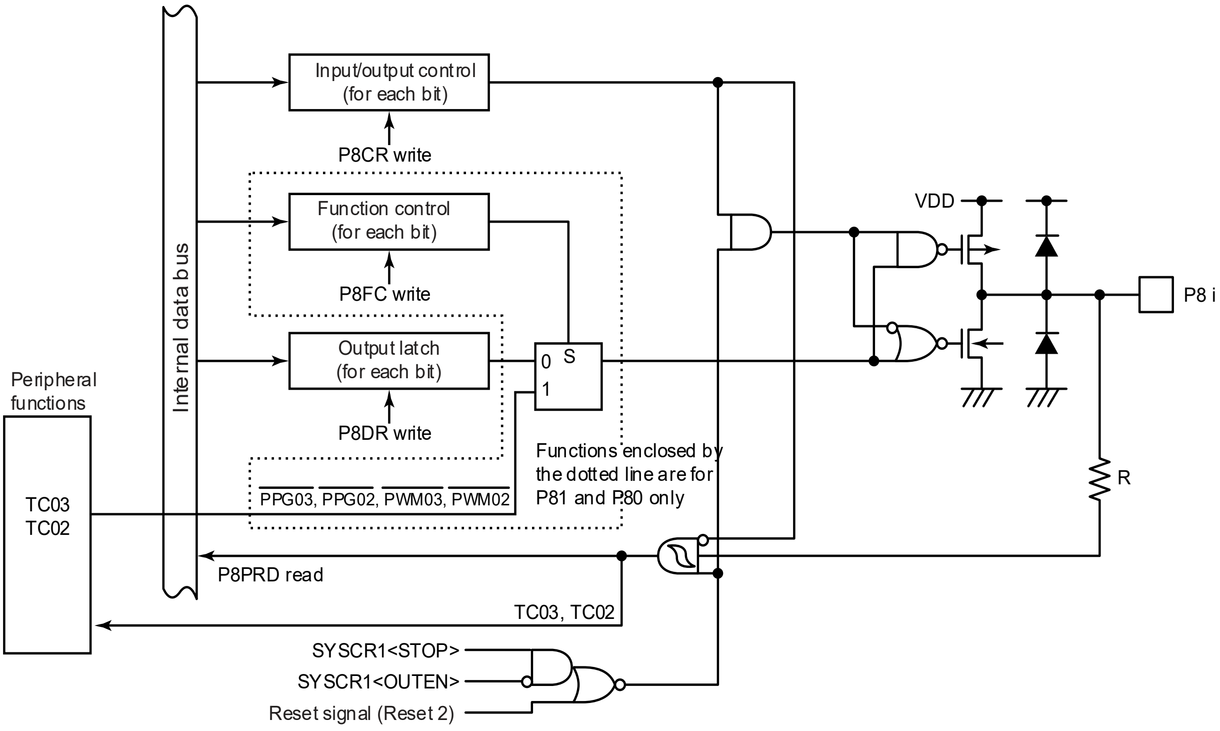
In regards to unavailable bits of the P8 port, the value which must be set or cleared to the corresponding bit of each register is shown in “8.4.7.2. Control” These values must be set or cleared to corresponding bit of each register in the setup routine.

And also, when the value of the register which includes these bits is modified, don’t change the set or cleared value of these bit.

Table 8-30 P8 Port for Each Product

	TMP89FS60B	TMP89FS62B	TMP89FS63B
P84	A	NA	NA
P83	A	NA	NA
P82	A	NA	NA
P81	A	A	A
P80	A	A	A

Note: A: Available, NA: Not available



Note 1: R = 100 [Ω] (typ.)

Note 2: TMP89FS60B: i = 4 to 0
TMP89FS62B, TMP89FS63B: i = 1 to 0

Figure 8-8 Port P8

8.4.7.2. Control

P8 port is controlled by the following registers.

Port P8 output latch

P8DR		7	6	5	4	3	2	1	0
(0x0008)	Bit Symbol	-	-	-	P84	P83	P82	P81	P80
	Read/Write	R	R	R	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

			TMP89FS60B	TMP89FS62B	TMP89FS63B
P84	Output level in the output mode	0:	“Low” level	Reserved (this bit symbol must be cleared to “0”.)	
		1:	“High” level		
P83	Output level in the output mode	0:	“Low” level	Reserved (this bit symbol must be cleared to “0”.)	
		1:	“High” level		
P82	Output level in the output mode	0:	“Low” level	Reserved (this bit symbol must be cleared to “0”.)	
		1:	“High” level		
P81	Output level in the output mode	0:	“Low” level		
		1:	“High” level		
P80	Output level in the output mode	0:	“Low” level		
		1:	“High” level		

Port P8 input/output control

P8CR		7	6	5	4	3	2	1	0
(0x0F22)	Bit Symbol	-	-	-	P8CR4	P8CR3	P8CR2	P8CR1	P8CR0
	Read/Write	R	R	R	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

			TMP89FS60B	TMP89FS62B	TMP89FS63B
P8CR4	Selects input mode or output mode.	0: 1:	Input mode Output mode	Reserved (this bit symbol must be set to “1”.)	
P8CR3	Selects input mode or output mode.	0: 1:	Input mode Output mode	Reserved (this bit symbol must be set to “1”.)	
P8CR2	Selects input mode or output mode.	0: 1:	Input mode Output mode	Reserved (this bit symbol must be set to “1”.)	
P8CR1	Selects input mode or output mode.	0: 1:	Input mode TC03 (I) Output mode $\overline{\mathrm{PPG03}}$ (O) $\overline{\mathrm{PWM03}}$ (O)		
P8CR0	Selects input mode or output mode.	0: 1:	Input mode TC02 (I) Output mode $\overline{\mathrm{PPG02}}$ (O) $\overline{\mathrm{PWM02}}$ (O)		

Note: Symbol "I" means secondary function input. Symbol "O" means secondary function output

Port P8 function control

P8FC		7	6	5	4	3	2	1	0
(0x0F3C)	Bit Symbol	-	-	-	-	-	-	P8FC1	P8FC0
	Read/Write	R	R	R	R	R	R	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

			TMP89FS60B	TMP89FS62B	TMP89FS63B
P8FC1	Port function	0:	I/O port		
		1:	$\overline{\text{PPG03}}$ (O) $\overline{\text{PWM03}}$ (O)		
P8FC0	Port function	0:	I/O port		
		1:	$\overline{\text{PPG02}}$ (O) $\overline{\text{PWM02}}$ (O)		

Note: Symbol "O" means secondary function output.

Port P8 input data

P8PRD		7	6	5	4	3	2	1	0
(0x0015)	Bit Symbol	-	-	-	P8PRD4	P8PRD3	P8PRD2	P8PRD1	P8PRD0
	Read/Write	R	R	R	R	R	R	R	R
	After reset	0	0	0	*	*	*	*	*

		TMP89FS60B	TMP89FS62B	TMP89FS63B
P8PRD4	Input data from port	Refer to Table 8-31.	The read data is “0”.	
P8PRD3	Input data from port	Refer to Table 8-31.	The read data is “0”.	
P8PRD2	Input data from port	Refer to Table 8-31.	The read data is “0”.	
P8PRD1	Input data from port	Refer to Table 8-31.		
P8PRD0	Input data from port	Refer to Table 8-31.		

Table 8-31 Read Value of <P8PRDi>

Read Value of <P8PRDi>	Setting Condition
	<P8CRi>
Level of the port pin	0
0	1

Note: TMP89FS60B: i = 4 to 0
TMP89FS62B, TMP89FS63B: i = 1 to 0

8.4.8. Port P9

Port P9 is a 5-bit input/output port that can be set to input or output for each bit individually, and it is also used as the serial interface and the UART.

The output circuit has the P-channel output control function and either the open-drain output or the CMOS output can be selected.

Port P9 contains a programmable pull-up resistor on the VDD pin side. This pull-up resistor can be used when the port is used in the input mode or as an open-drain output.

When this port is used as the serial interface or the UART, setting for the serial interface selecting function is also needed. For details, refer to "8.5. Serial Interface Selecting Function".

Table 8-32 Port P9

	-	-	-	P94	P93	P92	P91	P90
Secondary function	-	-	-	RXD2 TXD2	TXD2 RXD2	SCLK1	RXD1 TXD1 SI1	TXD1 RXD1 SO1

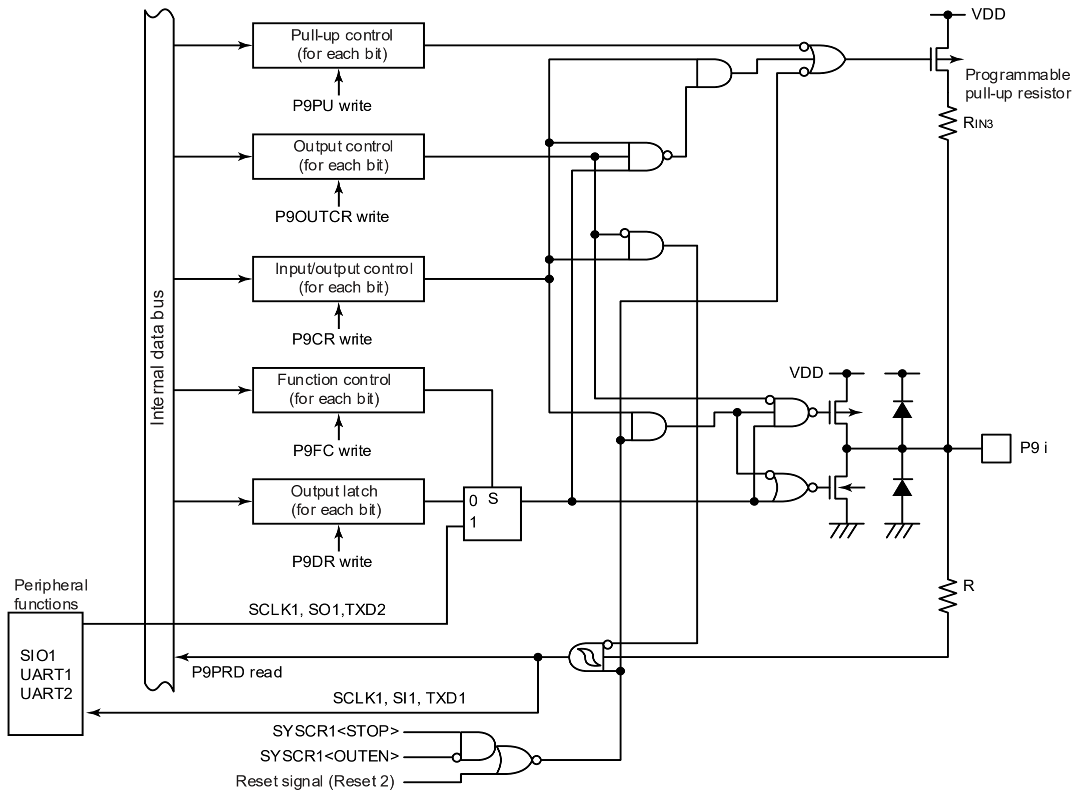
8.4.8.1. P9 Port for Each Product

P9 port for each product is shown in Table 8-33.
P9 port can be used on all products.

Table 8-33 P9 Port for Each Product

	TMP89FS60B	TMP89FS62B	TMP89FS63B
P94	A	A	A
P93	A	A	A
P92	A	A	A
P91	A	A	A
P90	A	A	A

Note: A: Available



Note 1: R = 100 [Ω] (typ.)

Note 2: R_{IN3} = 50 [kΩ] (typ.)

Note 3: i = 4 to 0

Figure 8-9 Port P9

8.4.8.2. Control

P9 port is controlled by the following registers.

Port P9output latch

P9DR		7	6	5	4	3	2	1	0
(0x0009)	Bit Symbol	-	-	-	P94	P93	P92	P91	P90
	Read/Write	R	R	R	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

			TMP89FS60B	TMP89FS62B	TMP89FS63B
P94	Output level in the output mode	0: 1:	Refer to Table 8-34.		
P93	Output level in the output mode	0: 1:	Refer to Table 8-34.		
P92	Output level in the output mode	0: 1:	Refer to Table 8-34.		
P91	Output level in the output mode	0: 1:	Refer to Table 8-34.		
P90	Output level in the output mode	0: 1:	Refer to Table 8-34.		

Table 8-34 Output Status of I/O Pin in the Output Mode

Port Function	Output status of I/O pin	Setting Condition		
		<P9i>	<P9OUTCRi>	<P9PUi>
Port Output mode CMOS Output	“Low” level	0	0	*
	“High” level	1		
Port Output mode Open-drain output	“Low” level	0	1	0
	“Hi-Z”	1		
	“Low” level	0		1
	“Pull-up”	1		

Note 1: *: Don’t care

Note 2: i = 4 to 0

Port P9 input/output control

		7	6	5	4	3	2	1	0
P9CR (0x0F23)	Bit Symbol	-	-	-	P9CR4	P9CR3	P9CR2	P9CR1	P9CR0
	Read/Write	R	R	R	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

			TMP89FS60B	TMP89FS62B	TMP89FS63B
P9CR4	Selects input mode or output mode.	0:	Input mode RXD2 (I)		
		1:	Output mode TXD2 (O)		
P9CR3	Selects input mode or output mode.	0:	Input mode RXD2 (I)		
		1:	Output mode TXD2 (O)		
P9CR2	Selects input mode or output mode.	0:	Input mode SCLK1 (I)		
		1:	Output mode SCLK1 (O)		
P9CR1	Selects input mode or output mode.	0:	Input mode RXD1 (I) SI1 (I)		
		1:	Output mode TXD1 (O)		
P9CR0	Selects input mode or output mode.	0:	Input mode RXD1 (I)		
		1:	Output mode TXD1 (O) SO1 (O)		

Note: Symbol "I" means secondary function input. Symbol "O" means secondary function output.

Port P9 function control

P9FC		7	6	5	4	3	2	1	0
(0x0F3D)	Bit Symbol	-	-	-	P9FC4	P9FC3	P9FC2	P9FC1	P9FC0
	Read/Write	R	R	R	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

			TMP89FS60B	TMP89FS62B	TMP89FS63B
P9FC4	Port function	0:	I/O port		
		1:	TXD2 (O)		
P9FC3	Port function	0:	I/O port		
		1:	TXD2 (O)		
P9FC2	Port function	0:	I/O port		
		1:	SCLK1 (O)		
P9FC1	Port function	0:	I/O port		
		1:	TXD1 (O)		
P9FC0	Port function	0:	I/O port		
		1:	TXD1 (O) SO1 (O)		

Note: Symbol "O" means secondary function output.

Port P9 output control

P9OUTCR		7	6	5	4	3	2	1	0
(0x0F4A)	Bit Symbol	-	-	-	P9OUT4	P9OUT3	P9OUT2	P9OUT1	P9OUT0
	Read/Write	R	R	R	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

			TMP89FS60B	TMP89FS62B	TMP89FS63B
P9OUT4	Output circuit type of port	0:	CMOS output		
		1:	Open-drain output		
P9OUT3	Output circuit type of port	0:	CMOS output		
		1:	Open-drain output		
P9OUT2	Output circuit type of port	0:	CMOS output		
		1:	Open-drain output		
P9OUT1	Output circuit type of port	0:	CMOS output		
		1:	Open-drain output		
P9OUT0	Output circuit type of port	0:	CMOS output		
		1:	Open-drain output		

Port P9 built-in pull-up resistor control

P9PU		7	6	5	4	3	2	1	0
(0x0F30)	Bit Symbol	-	-	-	P9PU4	P9PU3	P9PU2	P9PU1	P9PU0
	Read/Write	R	R	R	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

			TMP89FS60B	TMP89FS62B	TMP89FS63B
P9PU4	Selects connecting or disconnecting a built-in pull-up resister.	0: 1:	Refer to Table 8-35.		
P9PU3	Selects connecting or disconnecting a built-in pull-up resister.	0: 1:	Refer to Table 8-35.		
P9PU2	Selects connecting or disconnecting a built-in pull-up resister.	0: 1:	Refer to Table 8-35.		
P9PU1	Selects connecting or disconnecting a built-in pull-up resister.	0: 1:	Refer to Table 8-35.		
P9PU0	Selects connecting or disconnecting a built-in pull-up resister.	0: 1:	Refer to Table 8-35.		

Table 8-35 Port Function and Built-in Pull-up Resister

Port function	Connected or disconnected	<P9PUi>	<P9CRi>	< P9OUTi>	<P9i>
Input mode	Not connected	0	0	*	*
	Connected	1			
Output mode CMOS output	Not connected	*	1	0	
Output mode Open-drain output	Not connected	0		1	0
	Connected	1			1

Note 1: *: Don’t care

Note 2: i = 4 to 0

Port P9 input data

P9PRD		7	6	5	4	3	2	1	0
(0x0016)	Bit Symbol	-	-	-	P9PRD4	P9PRD3	P9PRD2	P9PRD1	P9PRD0
	Read/Write	R	R	R	R	R	R	R	R
	After reset	0	0	0	*	*	*	*	*

		TMP89FS60B	TMP89FS62B	TMP89FS63B
P9PRD4	Input data from port	Refer to Table 8-36.		
P9PRD3	Input data from port	Refer to Table 8-36.		
P9PRD2	Input data from port	Refer to Table 8-36.		
P9PRD1	Input data from port	Refer to Table 8-36.		
P9PRD0	Input data from port	Refer to Table 8-36.		

Table 8-36 Read Value of <P9PRDi>

Read value of <P9PRDi>	Setting Condition	
	<P9CRi>	<P9OUTCRi>
Level of the port pin	0	*
0	1	0
Level of the port pin		1

Note 1: * : Don’t care

Note 2: i = 4 to 0

8.4.9. Port PB

Port PB is an 8-bit input/output port that can be set to input or output for each bit individually, and it is also used as the serial interface input/output and the UART input/output.

The output circuit has the P-channel output control function and either the open-drain output or the CMOS output can be selected.

When this port is used as the serial interface or the UART, setting for serial interface selecting function is also needed. For details, refer to "8.5. Serial Interface Selecting Function".

Table 8-37 Port PB

	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
Secondary function	-	SCLK0	RXD0 TXD0 SI0	TXD0 RXD0 SO0	-	-	-	-

Note: The TMP89FS62B and the TMP89FS63B do not have the serial interface input/output (SCLK0, SI0 and SO0).

8.4.9.1. PB Port for Each Product

PB port for each product is shown in Table 8-38.

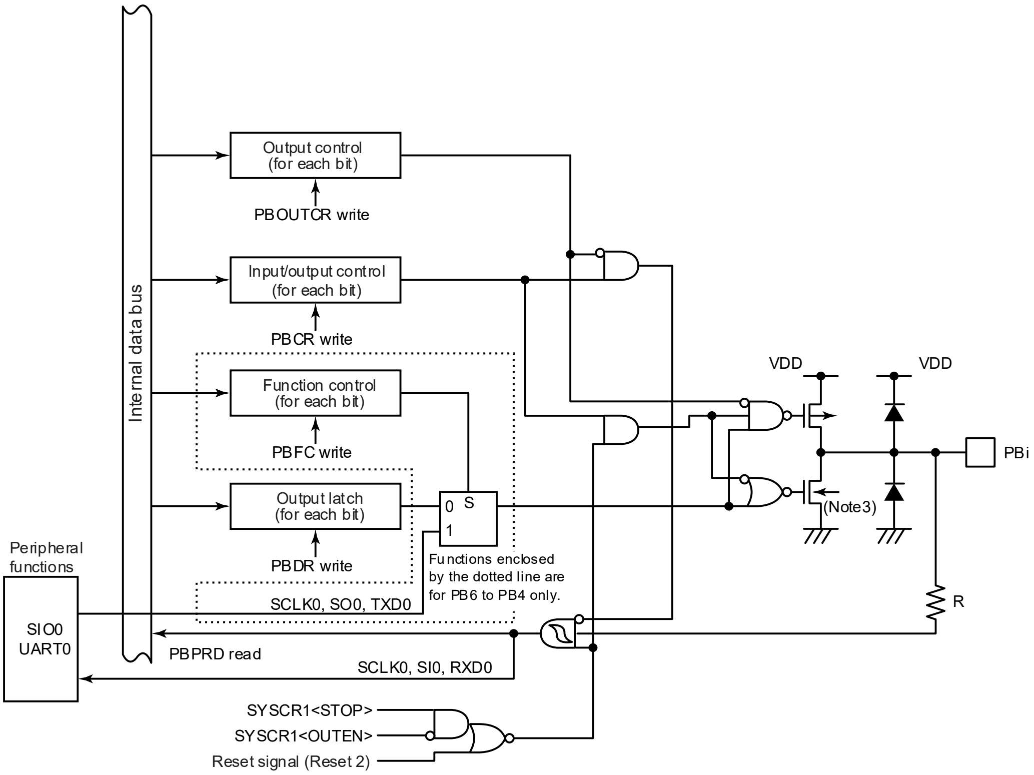
In regards to unavailable bits of the PB port, the value which must be set or cleared to the corresponding bit of each register is shown in “8.4.9.2. Control”. These values must be set or cleared to corresponding bit of each register in the setup routine.

And also, when the value of the register which includes these bits is modified, don’t change the set or cleared value of these bit.

Table 8-38 PB Port for Each Product

	TMP89FS60B	TMP89FS62B	TMP89FS63B
PB7	A	NA	NA
PB6	A	NA	NA
PB5	A	A	A
PB4	A	A	A
PB3	A	A	A
PB2	A	A	A
PB1	A	A	A
PB0	A	A	A

Note: A: Available, NA: Not available



Note 1: R = 100 [Ω] (typ.)

Note 2: i = 7 to 0

Note 3: Large Current Port (LIOL)

Figure 8-10 Port PB

8.4.9.2. Control

PB port is controlled by the following registers.

Port PB output latch

PBDR (0x000B)		7	6	5	4	3	2	1	0
	Bit Symbol	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

			TMP89FS60B	TMP89FS62B	TMP89FS63B
PB7	Output level in the output mode	0: 1:	Refer to Table 8-39.	Reserved (this bit symbol must be cleared to “0”.)	
PB6	Output level in the output mode	0: 1:	Refer to Table 8-39.	Reserved (this bit symbol must be cleared to “0”.)	
PB5	Output level in the output mode	0: 1:	Refer to Table 8-39.		
PB4	Output level in the output mode	0: 1:	Refer to Table 8-39.		
PB3	Output level in the output mode	0: 1:	Refer to Table 8-39.		
PB2	Output level in the output mode	0: 1:	Refer to Table 8-39.		
PB1	Output level in the output mode	0: 1:	Refer to Table 8-39.		
PB0	Output level in the output mode	0: 1:	Refer to Table 8-39.		

Table 8-39 Output Status of I/O Pin in the Output Mode

Port Function	Output status of I/O pin	Setting Condition	
		<PBi>	<PBOUTCRi>
Output mode CMOS output	“Low” level	0	0
	“High” level	1	
Output mode Open-drain output	“Low” level	0	1
	“Hi-Z”	1	

Note 1: *: Don’t care

Note 2: TMP89FS60B: i = 7 to 0
TMP89FS62B, TMP89FS63B: i = 5 to 0

Port PB input/output control

PBCR		7	6	5	4	3	2	1	0
(0x0F25)	Bit Symbol	PBCR7	PBCR6	PBCR5	PBCR4	PBCR3	PBCR2	PBCR1	PBCR0
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

			TMP89FS60B	TMP89FS62B	TMP89FS63B
PBCR7	Selects input mode or output mode.	0:	Input mode	Reserved (this bit symbol must be set to “1”.)	
		1:	Output mode		
PBCR6	Selects input mode or output mode.	0:	Input mode SCLK0 (I)	Reserved (this bit symbol must be set to “1”.)	
		1:	Output mode SCLK0 (O)		
PBCR5	Selects input mode or output mode.	0:	Input mode RXD0 (I) SI0 (I)	Input mode RXD0 (I)	
		1:	Output mode TXD0 (O)	Output mode TXD0 (O)	
PBCR4	Selects input mode or output mode.	0:	Input mode RXD0 (I)	Input mode RXD0 (I)	
		1:	Output mode TXD0 (O) SO0 (O)	Output mode TXD0 (O)	
PBCR3	Selects input mode or output mode.	0:	Input mode		
		1:	Output mode		
PBCR2	Selects input mode or output mode.	0:	Input mode		
		1:	Output mode		
PBCR1	Selects input mode or output mode.	0:	Input mode		
		1:	Output mode		
PBCR0	Selects input mode or output mode.	0:	Input mode		
		1:	Output mode		

Note: Symbol "I" means secondary function input. Symbol "O" means secondary function output.

Port PB function control

PBFC		7	6	5	4	3	2	1	0
(0x0F3F)	Bit Symbol	-	PBFC6	PBFC5	PBFC4	-	-	-	-
	Read/Write	R	R/W	R/W	R/W	R	R	R	R
	After reset	0	0	0	0	0	0	0	0

			TMP89FS60B	TMP89FS62B	TMP89FS63B
PBFC6	Port function	0: 1:	I/O port SCLK0 (O) (Note 2)	Reserved (this bit symbol must be cleared to “0”.)	
PBFC5	Port function	0: 1:	I/O port TXD0 (O)		
PBFC4	Port function	0: 1:	I/O port TXD0 (O) SO0 (O) (Note 2)		

Note 1: Symbol "O" means secondary function output.

Note 2: The SIO0 for the TMP89FS62B and TMP89FS63B is “Reserved”.

Port PB output control

PBOUTCR		7	6	5	4	3	2	1	0
(0x0F4C)	Bit Symbol	PBOUT7	PBOUT6	PBOUT5	PBOUT4	PBOUT3	PBOUT2	PBOUT1	PBOUT0
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

			TMP89FS60B	TMP89FS62B	TMP89FS63B
PBOUT7	Output circuit type of port	0: 1:	CMOS output Open-drain output	Reserved (this bit symbol must be cleared to “0”.)	
PBOUT6	Output circuit type of port	0: 1:	CMOS output Open-drain output	Reserved (this bit symbol must be cleared to “0”.)	
PBOUT5	Output circuit type of port	0: 1:	CMOS output Open-drain output		
PBOUT4	Output circuit type of port	0: 1:	CMOS output Open-drain output		
PBOUT3	Output circuit type of port	0: 1:	CMOS output Open-drain output		
PBOUT2	Output circuit type of port	0: 1:	CMOS output Open-drain output		
PBOUT1	Output circuit type of port	0: 1:	CMOS output Open-drain output		
PBOUT0	Output circuit type of port	0: 1:	CMOS output Open-drain output		

Port PB input data

PBPRD		7	6	5	4	3	2	1	0
(0x0018)	Bit Symbol	PBPRD7	PBPRD6	PBPRD5	PBPRD4	PBPRD3	PBPRD2	PBPRD1	PBPRD0
	Read/Write	R	R	R	R	R	R	R	R
	After reset	*	*	*	*	*	*	*	*

		TMP89FS60B	TMP89FS62B	TMP89FS63B
PBPRD7	Input data from port	Refer to Table 8-40.	The read data is “0”.	
PBPRD6	Input data from port	Refer to Table 8-40.	The read data is “0”.	
PBPRD5	Input data from port	Refer to Table 8-40.		
PBPRD4	Input data from port	Refer to Table 8-40.		
PBPRD3	Input data from port	Refer to Table 8-40.		
PBPRD2	Input data from port	Refer to Table 8-40.		
PBPRD1	Input data from port	Refer to Table 8-40.		
PBPRD0	Input data from port	Refer to Table 8-40.		

Table 8-40 Read Value of <PBPRDi>

Read value of <PBPRDi>	Setting Condition	
	<PBCRi>	<PBOUTCRi>
Level of the port pin	0	*
0	1	0
Level of the port pin		1

Note 1: * : Don’t care

Note 2: TMP89FS60B: i = 7 to 0
TMP89FS62B, TMP89FS63B: i = 5 to 0

8.5. Serial Interface Selecting Function

The TMP89FS60B/62B/63B can change the pins of the serial interfaces (SIO0/1, UART0/1/2 and I2C0) and interrupt source assignment and the input pin of the 16-bit timer counter (TCA0) by using the serial interface selecting function.

Two out of three functions, SIO0, UART0 and I2C0, can be used at the same time.

One of two functions, SIO1 and UART1, can be used.

Note: To use the serial interface and 16-bit timer counter, setting I/O port registers other than the registers of the serial interface selecting function is needed. Refer to " 8.4. I/O Port Registers".

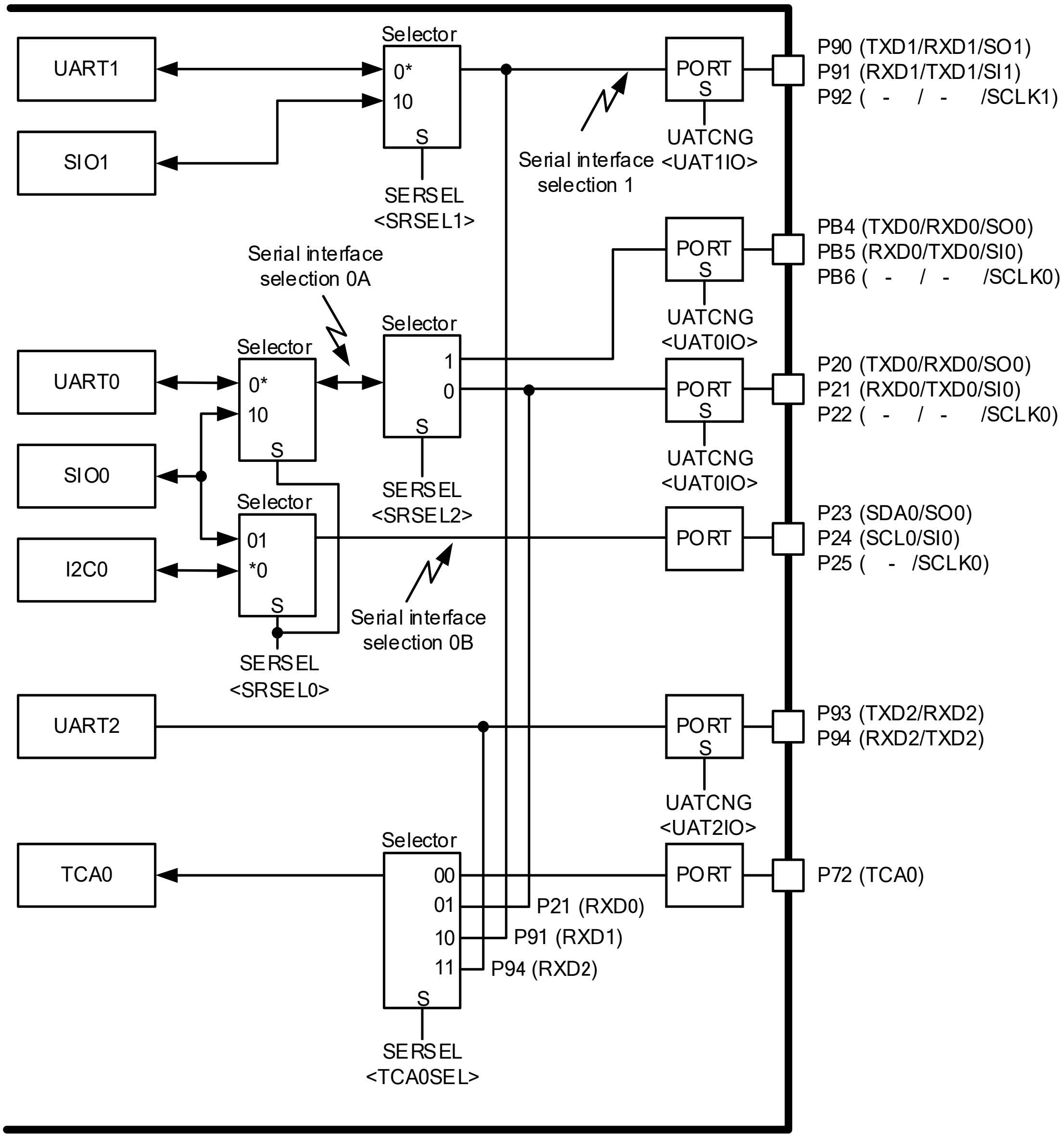


Figure 8-11 Serial Interface Selecting Function

Serial interface selection control register

SERSEL		7	6	5	4	3	2	1	0
(0x0FCB)	Bit Symbol	TCA0SEL		-	SRSEL2	SRSEL1		SRSEL0	
	Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

			TMP89FS60B	TMP89FS62B	TMP89FS63B
TCA0SEL	16-bit timer counter TCA0 input switching (Note 3)	00:	P72 (TCA0)		
		01:	Assigned to P21 (RXD0)		
		10:	Assigned to P91 (RXD1)		
		11:	Assigned to P94 (RXD2)		
SRSEL2	Select UART0/SIO0 input/output port	0:	P22, P21, P20	P22, P21, P20	P22, P21, P20
		1:	PB6, PB5, PB4	PB5, PB4 (Note 4)	PB6, PB5, PB4
SRSEL1	Serial interface selection 1 (Note 4)	00:	UART1		
		01:	UART1		
		10:	SIO1		
		11:	Reserved		
SRSEL0	Serial interface selection 0 (Note 6)	00:	0A: UART0 0B: I2C0	0A: UART0 0B: (Note 5)	0A: UART0 0B: I2C0
		01:	0A: UART0 0B: SIO0	0A: UART0 0B: (Note 5)	0A: UART0 0B: SIO0
		10:	0A: SIO0 0B: I2C0	0A: SIO0 0B: (Note 5)	0A: SIO0 0B: I2C0
		11:	Reserved	Reserved	Reserved

Note 1: The operation for changing SERSEL must be executed while the applicable serial interface and timer counter operations are stopped.

Note 2: The interrupt latch for the applicable serial interface must be cleared after changing SERSEL.

Note 3: The port assignment of the RXDx (x = 2 to 0) pin which is input to TCA0 input cannot be selected by UATCNG.

Note 4: The SIO0 cannot be selected as the serial interface selection 1.

Note 5: The serial interface which is assigned to the serial interface selection 0B cannot be used.

Note 6: "0A" means "Serial interface selection 0A" in Figure 8-11 and "0B" means "Serial interface selection 0B".

Note 7: When a read instruction is executed on SERSEL, bit 5 is read as "0".

UART input/output change control register

UATCNG		7	6	5	4	3	2	1	0
(0x0F57)	Bit Symbol	-	-	-	-	-	UAT2IO	UAT1IO	UAT0IO
	Read/Write	R	R	R	R	R	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

			RXDx pin		TXDx pin	
UAT2IO	Select UART2 input/ output port	0:	P94		P93	
		1:	P93		P94	
UAT1IO	Select UART1 input/ output port	0:	P91		P90	
		1:	P90		P91	
UAT0IO	Select UART0 input/ output port		SERSEL <SERSEL2> = 0	SERSEL <SERSEL2> = 1	SERSEL <SERSEL2> = 0	SERSEL <SERSEL2> = 1
		0:	P21	PB5	P20	PB4
		1:	P20	PB4	P21	PB5

Note 1: Set the corresponding bit of function register (PmFC (m = B, 9 and 2)) to "1", when the port is assigned to a TXDx (x = 2 to 0) pin.

Note 2: When a read instruction is executed on UATCNG, bits 7 to 3 are read as "0".

Note 3: The operation for changing UATCNG must be executed while the applicable UART operations are stopped.

Table 8-41 Select input/output port and interrupt

SERSEL <SRSEL0>	SERSEL <SRSEL2>	UATCNG <UAT0IO>	Port									Interrupt		
			UART0/SIO0						I2C0/SIO0					
			PB4	PB5	PB6	P20	P21	P22	P23	P24	P25	<IL7>	<IL6>	<IL15>
00	0	0	(Note2)	(Note2)	(Note2)	TXD0	RXD0	(Note2)	SDA0	SCL0	(Note2)	INTTXD0	INTRXD0	INTSBI0
		1				RXD0	TXD0							
	1	0	TXD0	RXD0	(Note2)	(Note2)	(Note2)	(Note2)						
		1	RXD0	TXD0										
01	0	0	(Note2)	(Note2)	(Note2)	TXD0	RXD0	(Note2)	SO0	SI0	SCLK0	INTTXD0	INTRXD0	INTSIO0
		1				RXD0	TXD0							
	1	0	TXD0	RXD0	(Note2)	(Note2)	(Note2)	(Note2)						
		1	RXD0	TXD0										
10	0	*	(Note2)	(Note2)	(Note2)	SO0	SI0	SCLK0	SDA0	SCL0	(Note2)	-	INTSIO0	INTSBI0
	1	*	SO0	SI0	SCLK0	(Note2)	(Note2)	(Note2)						
11	*	*	Reserved											

Note 1: *: Don't care

Note 2: This bit can be used as a port. Clear the corresponding bit of function register (PmFC (m = B and 2)) to "0".

Table 8-42 Select input/output port and interrupt

SERSEL <SRSEL1>	UATCNG <UAT1IO>	Port			Interrupt	
		UART1/SIO1				
		P90	P91	P92	<IL23>	<IL22>
00	0	TXD1	RXD1	(Note2)	INTTXD1	INTRXD1
	1	RXD1	TXD1			
01	0	TXD1	RXD1	(Note2)	INTTXD1	INTRXD1
	1	RXD1	TXD1			
10	*	SO1	SI1	SCLK1	-	INTSIO1
11	*	Reserved				

Note 1: *: Don't care

Note 2: This bit can be used as a port. Clear the corresponding bit of function register (P9FC) to "0".

9. Special Function Registers

The TMP89FS60B/62B/63B adopts the memory mapped I/O system, and all peripheral hardware data control and transfer operations are performed through the special function registers (SFR). SFR1 is mapped on addresses "0x0000" to "0x003F", SFR2 is mapped on addresses "0x0F00" to "0x0FFF", and SFR3 is mapped on addresses "0x0E40" to "0x0EBF".

9.1. SFR1 ("0x0000" to "0x003F")

Table 9-1 SFR1 ("0x0000" to "0x003F")

Address	Register Name
0x0000	P0DR
0x0001	P1DR
0x0002	P2DR
0x0003	Reserved
0x0004	P4DR
0x0005	P5DR
0x0006	Reserved
0x0007	P7DR
0x0008	P8DR
0x0009	P9DR
0x000A	Reserved
0x000B	PBDR
0x000C	Reserved
0x000D	P0PRD
0x000E	P1PRD
0x000F	P2PRD
0x0010	Reserved
0x0011	P4PRD
0x0012	P5PRD
0x0013	Reserved
0x0014	P7PRD
0x0015	P8PRD
0x0016	P9PRD
0x0017	Reserved
0x0018	PBPRD
0x0019	Reserved
0x001A	UART0CR1
0x001B	UART0CR2
0x001C	UART0DR
0x001D	UART0SR
0x001E	TD0BUF/RD0BUF
0x001F	SIO0CR
0x0020	SIO0SR
0x0021	SIO0BUF
0x0022	SBI0CR1
0x0023	SBI0CR2/SBI0SR2
0x0024	I2C0AR
0x0025	SBI0DBR
0x0026	T00REG
0x0027	T01REG
0x0028	T00PWM
0x0029	T01PWM
0x002A	T00MOD
0x002B	T01MOD
0x002C	T001CR
0x002D	TA0DRAL
0x002E	TA0DRAH
0x002F	TA0DRBL
0x0030	TA0DRBH
0x0031	TA0MOD
0x0032	TA0CR
0x0033	TA0SR
0x0034	ADCCR1
0x0035	ADCCR2
0x0036	ADCDRL
0x0037	ADCDRH
0x0038	DVOCR
0x0039	TBTCR
0x003A	EIRL
0x003B	EIRH
0x003C	EIRE
0x003D	EIRD
0x003E	Reserved
0x003F	PSW

Note: Don’t read from or write to the address of the SFR with “Reserved”.

9.2. SFR2 ("0x0F00" to "0x0FFF")

Table 9-2 SFR2 ("0x0F00" to "0x0F7F")

Address	Register Name	Address	Register Name	Address	Register Name	Address	Register Name
0x0F00	Reserved	0x0F20	Reserved	0x0F40	Reserved	0x0F60	Reserved
0x0F01	Reserved	0x0F21	P7CR	0x0F41	Reserved	0x0F61	Reserved
0x0F02	Reserved	0x0F22	P8CR	0x0F42	Reserved	0x0F62	Reserved
0x0F03	Reserved	0x0F23	P9CR	0x0F43	P2OUTCR	0x0F63	Reserved
0x0F04	Reserved	0x0F24	Reserved	0x0F44	Reserved	0x0F64	Reserved
0x0F05	Reserved	0x0F25	PBCR	0x0F45	Reserved	0x0F65	Reserved
0x0F06	Reserved	0x0F26	Reserved	0x0F46	Reserved	0x0F66	Reserved
0x0F07	Reserved	0x0F27	P0PU	0x0F47	Reserved	0x0F67	Reserved
0x0F08	Reserved	0x0F28	P1PU	0x0F48	Reserved	0x0F68	Reserved
0x0F09	Reserved	0x0F29	P2PU	0x0F49	Reserved	0x0F69	Reserved
0x0F0A	Reserved	0x0F2A	Reserved	0x0F4A	P9OUTCR	0x0F6A	Reserved
0x0F0B	Reserved	0x0F2B	P4PU	0x0F4B	Reserved	0x0F6B	Reserved
0x0F0C	Reserved	0x0F2C	Reserved	0x0F4C	PBOUTCR	0x0F6C	Reserved
0x0F0D	Reserved	0x0F2D	Reserved	0x0F4D	Reserved	0x0F6D	Reserved
0x0F0E	Reserved	0x0F2E	Reserved	0x0F4E	Reserved	0x0F6E	Reserved
0x0F0F	Reserved	0x0F2F	Reserved	0x0F4F	Reserved	0x0F6F	Reserved
0x0F10	Reserved	0x0F30	P9PU	0x0F50	Reserved	0x0F70	SIO1CR
0x0F11	Reserved	0x0F31	Reserved	0x0F51	Reserved	0x0F71	SIO1SR
0x0F12	Reserved	0x0F32	Reserved	0x0F52	Reserved	0x0F72	SIO1BUF
0x0F13	Reserved	0x0F33	Reserved	0x0F53	Reserved	0x0F73	Reserved
0x0F14	Reserved	0x0F34	P0FC	0x0F54	UART1CR1	0x0F74	POFFCR0
0x0F15	Reserved	0x0F35	Reserved	0x0F55	UART1CR2	0x0F75	POFFCR1
0x0F16	Reserved	0x0F36	P2FC	0x0F56	UART1DR	0x0F76	POFFCR2
0x0F17	Reserved	0x0F37	Reserved	0x0F57	UART1SR	0x0F77	POFFCR3
0x0F18	Reserved	0x0F38	P4FC	0x0F58	TD1BUF/RD1BUF	0x0F78	Reserved
0x0F19	Reserved	0x0F39	P5FC	0x0F59	Reserved	0x0F79	Reserved
0x0F1A	P0CR	0x0F3A	Reserved	0x0F5A	UART2CR1	0x0F7A	Reserved
0x0F1B	P1CR	0x0F3B	P7FC	0x0F5B	UART2CR2	0x0F7B	Reserved
0x0F1C	P2CR	0x0F3C	P8FC	0x0F5C	UART2DR	0x0F7C	Reserved
0x0F1D	Reserved	0x0F3D	P9FC	0x0F5D	UART2SR	0x0F7D	Reserved
0x0F1E	P4CR	0x0F3E	Reserved	0x0F5E	TD2BUF/RD2BUF	0x0F7E	Reserved
0x0F1F	P5CR	0x0F3F	PBFC	0x0F5F	Reserved	0x0F7F	Reserved

Note: Don’t read from or write to the address of the SFR with “Reserved”.

Table 9-3 SFR2 ("0x0F80" to "0x0FFF")

Address	Register Name
0x0F80	Reserved
0x0F81	Reserved
0x0F82	Reserved
0x0F83	Reserved
0x0F84	Reserved
0x0F85	Reserved
0x0F86	Reserved
0x0F87	Reserved
0x0F88	T02REG
0x0F89	T03REG
0x0F8A	T02PWM
0x0F8B	T03PWM
0x0F8C	T02MOD
0x0F8D	T03MOD
0x0F8E	T023CR
0x0F8F	Reserved
0x0F90	Reserved
0x0F91	Reserved
0x0F92	Reserved
0x0F93	Reserved
0x0F94	Reserved
0x0F95	Reserved
0x0F96	Reserved
0x0F97	Reserved
0x0F98	Reserved
0x0F99	Reserved
0x0F9A	Reserved
0x0F9B	Reserved
0x0F9C	Reserved
0x0F9D	Reserved
0x0F9E	Reserved
0x0F9F	Reserved
0x0FA0	Reserved
0x0FA1	Reserved
0x0FA2	Reserved
0x0FA3	Reserved
0x0FA4	Reserved
0x0FA5	Reserved
0x0FA6	Reserved
0x0FA7	Reserved
0x0FA8	TA1DRAL
0x0FA9	TA1DRAH
0x0FAA	TA1DRBL
0x0FAB	TA1DRBH
0x0FAC	TA1MOD
0x0FAD	TA1CR
0x0FAE	TA1SR
0x0FAF	Reserved
0x0FB0	Reserved
0x0FB1	Reserved
0x0FB2	Reserved
0x0FB3	Reserved
0x0FB4	Reserved
0x0FB5	Reserved
0x0FB6	Reserved
0x0FB7	Reserved
0x0FB8	Reserved
0x0FB9	Reserved
0x0FBA	Reserved
0x0FBB	Reserved
0x0FBC	Reserved
0x0FBD	Reserved
0x0FBE	Reserved
0x0FBF	Reserved
0x0FC0	Reserved
0x0FC1	Reserved
0x0FC2	Reserved
0x0FC3	Reserved
0x0FC4	KWUCR0
0x0FC5	KWUCR1
0x0FC6	VDCR1
0x0FC7	VDCR2
0x0FC8	RTCCR
0x0FC9	Reserved
0x0FCA	Reserved
0x0FCB	SERSEL
0x0FCC	IRSTSR
0x0FCD	WUCCR
0x0FCE	WUCDR
0x0FCF	CGCR
0x0FD0	FLSCR1
0x0FD1	FLSCR2/FLSCRM
0x0FD2	FLSSTB
0x0FD3	Reserved
0x0FD4	WDCTR
0x0FD5	WDCDR
0x0FD6	WDCNT
0x0FD7	WDST
0x0FD8	EINTCR1
0x0FD9	EINTCR2
0x0FDA	EINTCR3
0x0FDB	EINTCR4
0x0FDC	SYSCR1
0x0FDD	SYSCR2
0x0FDE	SYSCR3
0x0FDF	SYSCR4/SYSSR4
0x0FE0	ILL
0x0FE1	ILH
0x0FE2	ILE
0x0FE3	ILD
0x0FE4	Reserved
0x0FE5	Reserved
0x0FE6	Reserved
0x0FE7	Reserved
0x0FE8	Reserved
0x0FE9	Reserved
0x0FEA	Reserved
0x0FEB	Reserved
0x0FEC	Reserved
0x0FED	Reserved
0x0FEE	Reserved
0x0FEF	Reserved
0x0FF0	ILPRS1
0x0FF1	ILPRS2
0x0FF2	ILPRS3
0x0FF3	ILPRS4
0x0FF4	ILPRS5
0x0FF5	ILPRS6
0x0FF6	Reserved
0x0FF7	Reserved
0x0FF8	Reserved
0x0FF9	Reserved
0x0FFA	Reserved
0x0FFB	Reserved
0x0FFC	Reserved
0x0FFD	Reserved
0x0FFE	Reserved
0x0FFF	Reserved

Note: Don’t read from or write to the address of the SFR with “Reserved”.

9.3. SFR3 ("0x0E40" to "0x0EFF")

Table 9-4 SFR3 ("0x0E40" to "0x0EBF")

Address	Register Name	Address	Register Name	Address	Register Name	Address	Register Name
0x0E40	Reserved	0x0E60	Reserved	0x0E80	Reserved	0x0EA0	Reserved
0x0E41	Reserved	0x0E61	Reserved	0x0E81	Reserved	0x0EA1	Reserved
0x0E42	Reserved	0x0E62	Reserved	0x0E82	Reserved	0x0EA2	Reserved
0x0E43	Reserved	0x0E63	Reserved	0x0E83	Reserved	0x0EA3	Reserved
0x0E44	Reserved	0x0E64	Reserved	0x0E84	Reserved	0x0EA4	Reserved
0x0E45	Reserved	0x0E65	Reserved	0x0E85	Reserved	0x0EA5	Reserved
0x0E46	Reserved	0x0E66	Reserved	0x0E86	Reserved	0x0EA6	Reserved
0x0E47	Reserved	0x0E67	Reserved	0x0E87	Reserved	0x0EA7	Reserved
0x0E48	Reserved	0x0E68	Reserved	0x0E88	Reserved	0x0EA8	Reserved
0x0E49	Reserved	0x0E69	Reserved	0x0E89	Reserved	0x0EA9	Reserved
0x0E4A	Reserved	0x0E6A	Reserved	0x0E8A	Reserved	0x0EAA	Reserved
0x0E4B	Reserved	0x0E6B	Reserved	0x0E8B	Reserved	0x0EAB	Reserved
0x0E4C	Reserved	0x0E6C	Reserved	0x0E8C	Reserved	0x0EAC	Reserved
0x0E4D	Reserved	0x0E6D	Reserved	0x0E8D	Reserved	0x0EAD	Reserved
0x0E4E	Reserved	0x0E6E	Reserved	0x0E8E	Reserved	0x0EAE	Reserved
0x0E4F	Reserved	0x0E6F	Reserved	0x0E8F	Reserved	0x0EAF	Reserved
0x0E50	Reserved	0x0E70	Reserved	0x0E90	Reserved	0x0EB0	Reserved
0x0E51	Reserved	0x0E71	Reserved	0x0E91	Reserved	0x0EB1	Reserved
0x0E52	Reserved	0x0E72	Reserved	0x0E92	Reserved	0x0EB2	Reserved
0x0E53	Reserved	0x0E73	Reserved	0x0E93	Reserved	0x0EB3	Reserved
0x0E54	Reserved	0x0E74	Reserved	0x0E94	Reserved	0x0EB4	Reserved
0x0E55	Reserved	0x0E75	Reserved	0x0E95	Reserved	0x0EB5	Reserved
0x0E56	Reserved	0x0E76	Reserved	0x0E96	Reserved	0x0EB6	Reserved
0x0E57	UATCNG	0x0E77	Reserved	0x0E97	Reserved	0x0EB7	Reserved
0x0E58	Reserved	0x0E78	Reserved	0x0E98	Reserved	0x0EB8	Reserved
0x0E59	Reserved	0x0E79	Reserved	0x0E99	Reserved	0x0EB9	Reserved
0x0E5A	Reserved	0x0E7A	Reserved	0x0E9A	Reserved	0x0EBA	Reserved
0x0E5B	Reserved	0x0E7B	Reserved	0x0E9B	Reserved	0x0EBB	Reserved
0x0E5C	Reserved	0x0E7C	Reserved	0x0E9C	Reserved	0x0EBC	Reserved
0x0E5D	Reserved	0x0E7D	Reserved	0x0E9D	Reserved	0x0EBD	Reserved
0x0E5E	Reserved	0x0E7E	Reserved	0x0E9E	Reserved	0x0EBE	Reserved
0x0E5F	Reserved	0x0E7F	Reserved	0x0E9F	Reserved	0x0EBF	Reserved

Note: Don’t read from or write to the address of the SFR with “Reserved”.

Table 9-5 SFR3 ("0x0EC0" to "0x0EFF")

Address	Register Name	Address	Register Name	Address	Register Name	Address	Register Name
0x0EC0	Reserved	0x0ED0	Reserved	0x0EE0	Reserved	0x0EF0	Reserved
0x0EC1	Reserved	0x0ED1	Reserved	0x0EE1	Reserved	0x0EF1	Reserved
0x0EC2	Reserved	0x0ED2	Reserved	0x0EE2	Reserved	0x0EF2	Reserved
0x0EC3	Reserved	0x0ED3	Reserved	0x0EE3	Reserved	0x0EF3	Reserved
0x0EC4	Reserved	0x0ED4	Reserved	0x0EE4	Reserved	0x0EF4	Reserved
0x0EC5	Reserved	0x0ED5	Reserved	0x0EE5	Reserved	0x0EF5	Reserved
0x0EC6	Reserved	0x0ED6	Reserved	0x0EE6	Reserved	0x0EF6	Reserved
0x0EC7	Reserved	0x0ED7	Reserved	0x0EE7	Reserved	0x0EF7	Reserved
0x0EC8	Reserved	0x0ED8	Reserved	0x0EE8	Reserved	0x0EF8	Reserved
0x0EC9	Reserved	0x0ED9	Reserved	0x0EE9	Reserved	0x0EF9	Reserved
0x0ECA	Reserved	0x0EDA	Reserved	0x0EEA	Reserved	0x0EFA	Reserved
0x0ECB	Reserved	0x0EDB	Reserved	0x0EEB	Reserved	0x0EFB	Reserved
0x0ECC	Reserved	0x0EDC	Reserved	0x0EEC	Reserved	0x0EFC	Reserved
0x0ECD	Reserved	0x0EDD	Reserved	0x0EED	Reserved	0x0EFD	Reserved
0x0ECE	Reserved	0x0EDE	Reserved	0x0EEE	Reserved	0x0EFE	Reserved
0x0ECF	Reserved	0x0EDF	Reserved	0x0EEF	Reserved	0x0EFF	Reserved

Note: Don’t read from or write to the address of the SFR with “Reserved”.

10. Low Power Consumption Control for Peripherals

The TMP89FS60B/62B/63B has low power consumption registers (POFFCRn) that save power when specific peripheral functions are unused. Each bit of the low power consumption registers can be set to enable or disable each peripheral function. (n = 3 to 0)

The basic clock supply to each peripheral function is disabled for power saving, by setting the corresponding bit of the low power consumption registers (POFFCRn) to "0". (The disabled peripheral functions become unavailable.) The basic clock supply to each peripheral function is enabled and the function becomes available by setting the corresponding bit of the low power consumption registers (POFFCRn) to "1".

After reset, the low power consumption registers (POFFCRn) are initialized to "0", and thus the peripheral functions are unavailable. When each peripheral function is used for the first time, be sure to set the corresponding bit of the low power consumption registers (POFFCRn) to "1" in the initial settings of the program (before operating the control register for the peripheral function).

When a peripheral function is operating, the corresponding bit of the low power consumption registers (POFFCRn) must not be changed to "0". When it is changed, the peripheral function may operate unexpectedly.

10.1. Control

The Low Power Consumption Control is controlled by the low power consumption registers (POFFCRn). (n = 0, 1, 2, 3)

Low power consumption register 0

POFFCR0		7	6	5	4	3	2	1	0
(0x0F74)	Bit Symbol	-	-	TC023EN	TC001EN	-	-	TCA1EN	TCA0EN
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

TC023EN	TC02 and 03 control	0: 1:	Disable Enable
TC001EN	TC00 and 01 control	0: 1:	Disable Enable
TCA1EN	TCA1 control	0: 1:	Disable Enable
TCA0EN	TCA0 control	0: 1:	Disable Enable

Note: The written values are read from the bits 7 and 6, 3 and 2 of POFFCR0. These bits must be cleared to “0”.

Low power consumption register 1

POFFCR1		7	6	5	4	3	2	1	0
(0x0F75)	Bit Symbol	-	-	-	SBI0EN	-	UART2EN	UART1EN	UART0EN
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

SBI0EN (Note 2)	SBI0 control	0: 1:	Disable Enable
UART2EN	UART2 control	0: 1:	Disable Enable
UART1EN	UART1 control	0: 1:	Disable Enable
UART0EN	UART0 control	0: 1:	Disable Enable

Note 1: The written values are read from the bits 7 to 5 and 3 of POFFCR1. These bits must be cleared to “0”.

Note 2: <SBI0EN> is “Reserved” for the TMP89FS62B.

Low power consumption register 2

POFFCR2		7	6	5	4	3	2	1	0
(0x0F76)	Bit Symbol	-	-	RTCEN	-	-	-	SIO1EN	SIO0EN
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

RTCEN	RTC control	0: 1:	Disable Enable
SIO1EN	SIO1 control	0: 1:	Disable Enable
SIO0EN	SIO0 control	0: 1:	Disable Enable

Note: The written values are read from the bits 7 and 6, 4 to 2 of POFFCR2. These bits must be cleared to “0”.

Low power consumption register 3

POFFCR3		7	6	5	4	3	2	1	0
(0x0F77)	Bit Symbol	-	-	INT5EN	INT4EN	INT3EN	INT2EN	INT1EN	INT0EN
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

INT5EN	INT5 control	0: 1:	Disable Enable
INT4EN	INT4 control	0: 1:	Disable Enable
INT3EN	INT3 control	0: 1:	Disable Enable
INT2EN	INT2 control	0: 1:	Disable Enable
INT1EN (Note 2)	INT1 control	0: 1:	Disable Enable
INT0EN (Note 2)	INT0 control	0: 1:	Disable Enable

Note 1: The written values are read from the bits 7 and 6 of POFFCR3. These bits must be cleared to “0”.
Note 2: <INT0EN> and <INT1EN> are “Reserved” for the TMP89FS62B and TMP89FS63B.

11. Divider Output (DVO)

This function outputs approximately 50% duty pulses that can be used to drive the piezoelectric buzzer or other device.

11.1. Configuration

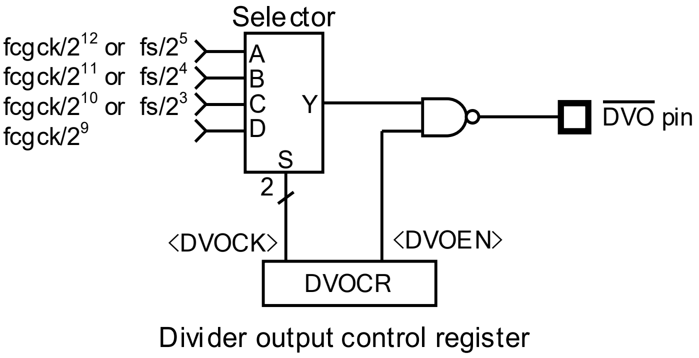


Figure 11-1 Divider Output

11.2. Control

The divider output is controlled by the divider output control register (DVOCR).

Divider output control register

DVOCR (0x0038)		7	6	5	4	3	2	1	0
	Bit Symbol	-	-	-	-	-	DVOEN	DVOCK	
	Read/Write	R	R	R	R	R	R/W	R/W	
	After reset	0	0	0	0	0	0	0	0

DVOEN	Enables/disables the divider output	0:	Disable the divider output		
		1:	Enable the divider output		
DVOCK	Selects the divider output frequency Unit: [Hz]		NORMAL1/2 or IDLE1/2 mode		SLOW1/2 or SLEEP1 mode
			SYSCR1<DV9CK> = 0	SYSCR1<DV9CK> = 1	
		00:	fcgck / 2^{12}	fs / 2^5	fs / 2^5
		01:	fcgck / 2^{11}	fs / 2^4	fs / 2^4
		10:	fcgck / 2^{10}	fs / 2^3	fs / 2^3
		11:	fcgck / 2^9	Reserved	Reserved

Note 1: fcgck: Gear clock [Hz], fs: Low-frequency clock [Hz]

Note 2: DVOCR<DVOEN> is cleared to "0" when the operation mode is switched to STOP, IDLE0 or SLEEP0 mode. At this time, DVOCR<DVOCK> holds the value.

Note 3: When SYSCR1<DV9CK> is "1" in the NORMAL1/2 or IDLE1/2 mode, the DVO frequency is subject to some fluctuations to synchronize fs and fcgck.

Note 4: When a read instruction is executed on DVOCR, bits 7 to 3 are read as "0".

11.3. Function

Select the divider output frequency at DVOCR<DVOCK>.

The divider output is enabled by setting DVOCR<DVOEN> to "1". Then, The rectangular wave selected by DVOCR<DVOCK> is output from $\overline{\text{DVO}}$ pin.

It is disabled by clearing DVOVR<DVOEN> to "0". And $\overline{\text{DVO}}$ pin keeps “High” level.

When the operation mode is changed to STOP or IDLE0/SLEEP0 mode, DVOCR<DVOEN> is cleared to "0" and the $\overline{\text{DVO}}$ pin outputs the “High” level.

The divider output source clock operates, regardless of the value of DVOCR<DVOEN>.

Therefore, the frequency of the first divider output after DVOCR<DVOEN> is set to "1" is not the frequency set at DVOCR<DVOCK>.

When DVOCR<DVOEN> is cleared to “0” by the software or the operation mode is changed to STOP, IDLE0 or SLEEP0 mode and DVOCR<DVOEN> is cleared to "0", the frequency of the divider output is not the frequency set at DVOCR<DVOCK>.

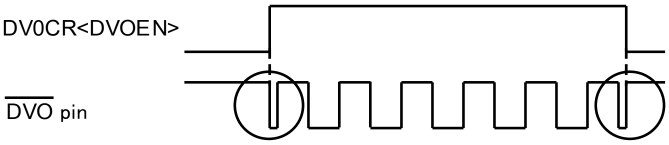


Figure 11-2 Divider Output Timing

When the operation mode is changed from NORMAL mode to SLOW mode or from SLOW mode to NORMAL mode, the divider output frequency does not reach the expected value due to synchronization of the gear clock (fcgck) and the low-frequency clock (fs).

Example: 2.441 [kHz] pulse outputs from $\overline{\text{DVO}}$ pin (fcgck = 10.0 [MHz])

```
LD    (DVOCR), 0x04                   ; <DVOCK> ← "00", <DVOEN> ← "1"
```

Table 11-1 Divider Output Frequency (Example: fcgck = 10.0 [MHz], fs = 32.768 [kHz])

<DVOCK>	Divider output frequency		
	NORMAL1/2 or IDLE1/2 mode		SLOW1/2 or SLEEP1 mode
	SYSCR1 <DV9CK> = 0	SYSCR1 <DV9CK> = 1	
00	2.441 [kHz]	1.024 [kHz]	1.024 [kHz]
01	4.883 [kHz]	2.048 [kHz]	2.048 [kHz]
10	9.766 [kHz]	4.096 [kHz]	4.096 [kHz]
11	19.531 [kHz]	Reserved	Reserved

12. Time Base Timer (TBT)

The time base timer generates the base time for key scanning, dynamic display and other processes. It also provides a time base timer interrupt request (INTTBT) in a certain cycle.

12.1. Configure

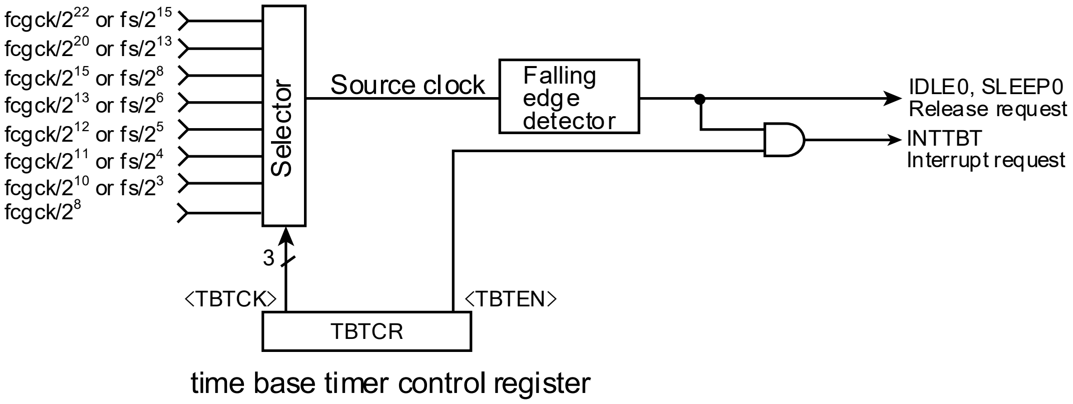


Figure 12-1 Time Base Timer Configuration

12.2. Control

The time base timer is controlled by the time base timer control register (TBTCR).

Time base timer control register

TBTCR (0x0039)		7	6	5	4	3	2	1	0
	Bit Symbol	-	-	-	-	TBTEN	TBTCK		
	Read/Write	R	R	R	R	R/W	R/W		
	After reset	0	0	0	0	0	0	0	0

TBTEN	Enables/disables the time base timer interrupt request	0: 1:	Disables generation of interrupt requests Enables generation of interrupt requests		
TBTCK	Selects the time base timer interrupt frequency Unit: [Hz]		NORMAL1/2 or IDLE1/2 mode		SLOW1/2 or SLEEP1 mode
			SYSCR1 <DV9CK> = 0	SYSCR1 <DV9CK> = 1	
		000:	fcgck / 2^{22}	fs / 2^{15}	fs / 2^{15}
		001:	fcgck / 2^{20}	fs / 2^{13}	fs / 2^{13}
		010:	fcgck / 2^{15}	fs / 2^{8}	Reserved
		011:	fcgck / 2^{13}	fs / 2^{6}	Reserved
		100:	fcgck / 2^{12}	fs / 2^{5}	Reserved
		101:	fcgck / 2^{11}	fs / 2^{4}	Reserved
		110:	fcgck / 2^{10}	fs / 2^{3}	Reserved
		111:	fcgck / 2^{8}	Reserved	Reserved

Note 1: fcgck: Gear clock [Hz], fs: Low-frequency clock [Hz]

Note 2: When the operation mode is changed to the STOP mode. At this time, TBTCR<TBTEN> is cleared to "0" and TBTCR<TBTCK> maintains the value.

Note 3: TBTCR<TBTCK> must be set when TBTCR<TBTEN> is "0".

Note 4: When SYSCR1<DV9CK> is "1" in the NORMAL1/2 or IDLE1/2 mode, the time of the interrupt request is subject to some fluctuations to synchronize fs and fcgck.

Note 5: When a read instruction is executed on TBTCR, bits 7 to 4 are read as "0".

12.3. Functions

Select the source clock frequency for the time base timer by TBTCR<TBTCK>. TBTCR<TBTCK> must be changed when TBTCR<TBTEN> is "0". Otherwise, the INTTBT interrupt request is generated at unexpected timing.

Setting TBTCR<TBTEN> to "1" causes interrupt request to occur at the falling edge of the source clock. When TBTCR<TBTEN> is cleared to "0", no interrupt request will occur.

When the operation mode is changed to the STOP mode, TBTCR<TBTEN> is cleared to "0".

The source clock of the time base timer operates regardless of the TBTCR<TBTEN> value.

A time base timer interrupt is generated at the first falling edge of the source clock after a time base timer interrupt request is enabled. Therefore, the period from when the time TBTCR<TBTEN> is set to "1" to the time when the first interrupt request occurs is shorter than the period of the frequency set at TBTCR<TBTCK>.

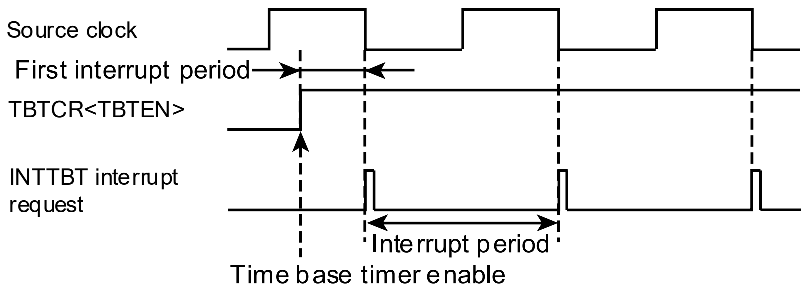


Figure 12-2 Time Base Timer Interrupt

When the operation mode is changed from NORMAL mode to SLOW mode or from SLOW mode to NORMAL mode, the interrupt request will not occur at the expected timing due to synchronization of the gear clock (fcgck) and the low-frequency clock (fs). It is recommended that the operation mode be changed when TBTCR<TBTCK> is "0".

Table 12-1 Time Base Timer Interrupt Frequency (Example: when fcgck = 10.0 [MHz] and fs = 32.768 [kHz])

<TBTCK>	Time base timer interrupt frequency		
	NORMAL1/2 or IDLE1/2 mode	NORMAL1/2 or IDLE1/2 mode	SLOW1/2 or SLEEP1 mode
	SYSCR1<DV9CK> = 0	SYSCR1<DV9CK> = 1	
000	2.38 [Hz]	1 [Hz]	1 [Hz]
001	9.54 [Hz]	4 [Hz]	4 [Hz]
010	305.18 [Hz]	128 [Hz]	Reserved
011	1220.70 [Hz]	512 [Hz]	Reserved
100	2441.41 [Hz]	1024 [Hz]	Reserved
101	4882.81 [Hz]	2048 [Hz]	Reserved
110	9765.63 [Hz]	4096 [Hz]	Reserved
111	39062.5 [Hz]	Reserved	Reserved

Example: Set the time base timer interrupt frequency to fcgck/2^{15} [Hz] and enable generation of interrupt request.

```
DI                                ; <IMF> ← "0"
SET   (EIRL).5                    ; Set the interrupt enable register
EI                                ; <IMF> ← "1"
LD    (TBTCR), 0x02               ; Set the interrupt frequency
LD    (TBTCR), 0x0A               ; Enable generation of interrupt request
```

13. 16-bit Timer Counter (TCA)

The TMP89FS60B/62B/63B contains the high-performance 16-bit timer counters (TCA).

This chapter describes the TCA0. For the TCA1, replace the SFR addresses and pin names, as shown in Table 13-1 and Table 13-2.

Table 13-1 SFR Address Assignment

	TAxDRAL (Address)	TAxDRAH (Address)	TAxDRBL (Address)	TAxDRBH (Address)	TAxMOD (Address)	TAxCR (Address)	TAxSR (Address)	Low power consumption register
TCA0	TA0DRAL (0x002D)	TA0DRAH (0x002E)	TA0DRBL (0x002F)	TA0DRBH (0x0030)	TA0MOD (0x0031)	TA0CR (0x0032)	TA0SR (0x0033)	POFFCR0 <TCA0EN>
TCA1	TA1DRAL (0x0FA8)	TA1DRAH (0x0FA9)	TA1DRBL (0x0FAA)	TA1DRBH (0x0FAB)	TA1MOD (0x0FAC)	TA1CR (0x0FAD)	TA1SR (0x0FAE)	POFFCR0 <TCA1EN>

Table 13-2 Pin Names

	Timer input pin	PPG output pin
TCA0	TCA0 pin	$\overline{\text{PPGA0}}$ pin
TCA1	TCA1 pin	$\overline{\text{PPGA1}}$ pin

13.1. 16-bit Timer Counters for Each Product

The 16-bit timer counters (TCA) for each product is shown Table 13-3.

In regards to unavailable channel of the TCA, the corresponding bit of the low power register must be cleared to “0”.

Table 13-3 16-bit Timer Counters (TCA) for Each Product

	TMP89FS60B	TMP89FS62B	TMP89FS63B
TCA0	A	A	A
TCA1	A	A	A

Note: A: Available

13.2. Configuration

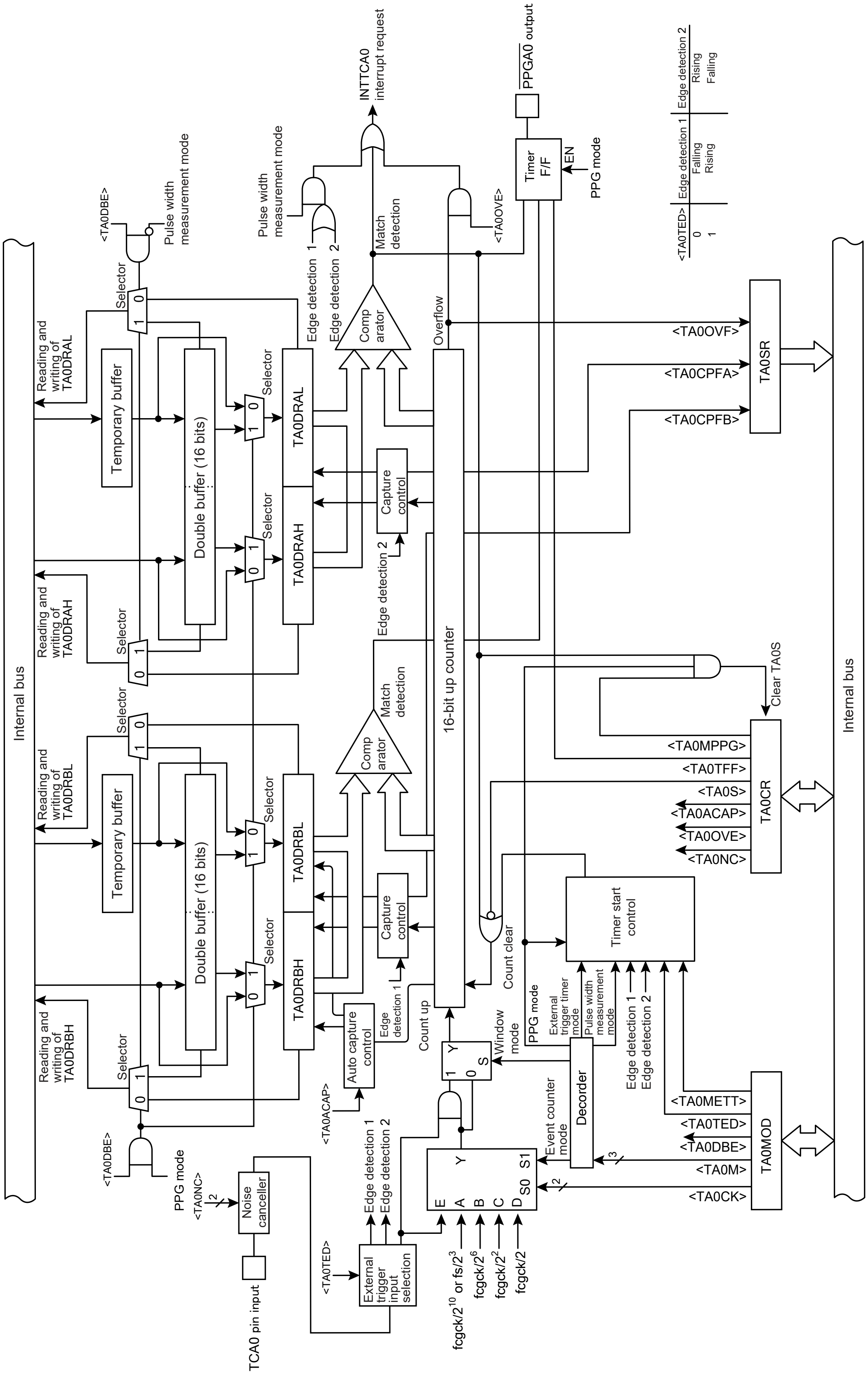


Figure 13-1 Timer Counter A0

13.3. Control

The TCA0 is controlled by the low power consumption register (POFFCR0), the timer counter A0 mode register (TA0MOD), the timer counter A0 control register (TA0CR) and two 16-bit timer A0 registers (TA0DRA and TA0DRB).

Low power consumption register 0

POFFCR0		7	6	5	4	3	2	1	0
(0x0F74)	Bit Symbol	-	-	(TC023EN)	(TC001EN)	-	-	TCA1EN	TCA0EN
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

TCA1EN	TCA1 control	0: 1:	Disable Enable
TCA0EN	TCA0 control	0: 1:	Disable Enable

Note: The written values are read from the bits 7 and 6, 3 and 2 of POFFCR0. These bits must be cleared to “0”.

Timer counter A0 mode register

TA0MOD		7	6	5	4	3	2	1	0
(0x0031)	Bit Symbol	TA0DBE	TA0TED	TA0MCAP TA0METT	TA0CK		TA0M		
	Read/Write	R/W	R/W	R/W	R/W		R/W		
	After reset	1	0	0	0	0	0	0	0

TA0DBE	Double buffer control	0:	Disable the double buffer		
		1:	Enable the double buffer		
TA0TED	External trigger input selection	0:	Rising edge/”High” level		
		1:	Falling edge/”Low” level		
TA0MCAP	Pulse width measurement mode control	0:	Both edge capture		
		1:	Single edge capture		
TA0METT	External trigger timer mode control	0:	Trigger start		
		1:	Trigger start & stop		
TA0CK	TCA0 source clock selection		NORMAL1/2 or IDLE1/2 mode		SLOW1/2 or SLEEP1 mode
			SYSCR1 <DV9CK> = 0	SYSCR1 <DV9CK> = 1	
		00:	fcgck / 2^{10}	fs / 2^{3}	fs / 2^{3}
		01:	fcgck / 2^{6}	fcgck / 2^{6}	-
		10:	fcgck / 2^{2}	fcgck / 2^{2}	-
		11:	fcgck / 2	fcgck / 2	-
TA0M	TCA0 operation mode selection	000:	Timer mode		
		001:	Timer mode		
		010:	Event counter mode		
		011:	PPG output mode (Software start)		
		100:	External trigger timer mode		
		101:	Window mode		
		110:	Pulse width measurement mode		
		111:	Reserved		

Note 1: fcgck: Gear clock [Hz], fs: Low-frequency clock [Hz]

Note 2: Set TA0MOD when the TCA0 is stopped (TA0CR<TA0S> = 0). Writing to TA0MOD is invalid during it is operated (TA0CR<TA0S> = 1).

Timer counter A0 control register

TA0CR		7	6	5	4	3	2	1	0
(0x0032)	Bit Symbol	TA0OVE	TA0TFF	TA0NC		-	-	TA0ACAP TA0MPPG	TA0S
	Read/Write	R/W	R/W	R/W		R	R	R/W	R/W
	After reset	0	1	0	0	0	0	0	0

TA0OVE	Overflow interrupt control	0: 1:	Generates no INTTCA0 interrupt request when the counter overflow occurs. Generates an INTTCA0 interrupt request when the counter overflow occurs.	
TA0TFF	Timer F/F control	0: 1:	Clear Set	
TA0NC	Noise canceller sampling interval setting		NORMAL1/2 or IDLE1/2 mode	SLOW1/2 or SLEEP1 mode
		00: 01: 10: 11:	No noise canceller fcgck / 2 fcgck / 2^2 fcgck / 2^8	No noise canceller - - fs / 2
TA0ACAP	Auto capture function	0: 1:	Disable the auto capture Enable the auto capture	
TA0MPPG	PPG output control	0: 1:	Continuous One-shot	
TA0S	TCA0 start control	0: 1:	Stop & clear counter Start	

Note 1: The auto capture can be used only in the timer, event counter, external trigger timer and window modes.

Note 2: Set <TA0TFF>, <TA0OVE> and <TA0NC> when the TCA0 is stopped (<TA0S> = 0). Writing to them is invalid during it is operated (<TA0S> = 1).

Note 3: When the STOP mode is started, the start control <TA0S> is automatically cleared to "0" and the TCA0 stops. Set <TA0S> again to use the TCA0 after the release of the STOP mode.

Note 4: When a read instruction is executed on TA0CR, the bits 3 and 2 are read as "0".

Note 5: Do not set <TA0NC> to "01" or "10" when the operation mode is in the SLOW1/2 or SLEEP1 mode. Setting <TA0NC> to "01" or "10" stops the noise canceller and no signal is input to the TCA0.

Timer counter A0 status register

TA0SR (0x0033)		7	6	5	4	3	2	1	0
	Bit Symbol	TA0OVF	-	-	-	-	-	TA0CPFA	TA0CPFB
	Read/Write	R	R	R	R	R	R	R	R
	After reset	0	0	0	0	0	0	0	0

TA0OVF	Overflow flag	0: 1:	No overflow has occurred. The overflow has occurred at least once.
TA0CPFA	Capture completion flag A	0: 1:	No capture operation has been executed. The pulse width capture has been executed in the both-edge capture mode at least once.
TA0CPFB	Capture completion flag B	0: 1:	No capture operation has been executed. The capture operation has been executed in the single-edge capture at least once. The pulse duty width capture has been executed in the both-edge capture at least once.

Note 1: <TA0OVF>, <TA0CPFA> and <TA0CPFB> are cleared to "0" automatically after TA0SR is read. Writing to TA0SR is invalid.

Note 2: When a read instruction is executed on TA0SR, the bits 6 to 2 are read as "0".

Timer counter A0 register AL

TA0DRAL		7	6	5	4	3	2	1	0
(0x002D)	Bit Symbol	TA0DRAL							
	Read/Write	R/W							
	After reset	1	1	1	1	1	1	1	1

Timer counter A0 register AH

TA0DRAH		15	14	13	12	11	10	9	8
(0x002E)	Bit Symbol	TA0DRAH							
	Read/Write	R/W							
	After reset	1	1	1	1	1	1	1	1

Timer counter A0 register BL

TA0DRBL		7	6	5	4	3	2	1	0
(0x002F)	Bit Symbol	TA0DRBL							
	Read/Write	R/W							
	After reset	1	1	1	1	1	1	1	1

Timer counter A0 register BH

TA0DRBH		15	14	13	12	11	10	9	8
(0x0030)	Bit Symbol	TA0DRBH							
	Read/Write	R/W							
	After reset	1	1	1	1	1	1	1	1

Note 1: When a write instruction is executed on TA0DRAL (TA0DRBL), the set value does not become effective immediately, but is temporarily stored in the temporary buffer. Subsequently, when a write instruction is executed on the high order register, TA0DRAH (TA0DRBH), the 16-bit value are collectively stored in the double buffer or TA0DRAL (TA0DRBL) and TA0DRAH (TA0DRBH) . When setting data to the registers of the TCA0, be sure to write the data into the lower order register and the higher order register in a sequence.

Note 2: The registers of the TCA0 cannot be written in the pulse width measurement mode.

13.4. Low Power Consumption Control

The TCA0 has the low power consumption register (POFFCR0) that saves power consumption when the TCA0 is not used.

Setting POFFCR0<TCA0EN> to "0" disables the basic clock supply to the TCA0 to save power. Note that this makes the TCA0 unusable. Setting POFFCR0<TCA0EN> to "1" enables the basic clock supply to the TCA0 and enables the TCA0 to operate.

After reset, POFFCR0<TCA0EN> is initialized to "0", and this makes the TCA0 unusable. When using the TCA0 for the first time, be sure to set POFFCR0<TCA0EN> to "1" in the initial setting of the program (before the TCA0 control register is modified).

Do not change POFFCR0<TCA0EN> to "0" during the TCA0 operation. Otherwise TCA0 may operate unexpectedly.

13.5. Timer Function

The TCA0 has six types of operation modes; timer, external trigger timer, event counter, window, pulse width measurement and programmable pulse generate (PPG) output modes.

13.5.1. Timer mode

In the timer mode, the up-counter counts up using the internal clock, and interrupt request can be generated regularly at specified times.

13.5.1.1. Setting

Setting the operation mode selection TA0MOD<TA0M> to "000" or "001" activates the timer mode. Selects the source clock at TA0MOD<TA0CK>.

Setting TA0CR<TA0S> to "1" starts the timer counter operation. After the timer counter operation is started, writing to TA0MOD and TA0CR<TA0OVE> is disabled. Be sure to complete the required settings to these registers before starting the timer counter operation.

Table 13-4 Resolution of Timer Mode and Maximum Time Setting

TA0MOD<TA0CK>	Source clock			Resolution		Maximum time setting	
	NORMAL1/2 or IDLE1/2 mode		SLOW1/2 or SLEEP1 mode	fcgck = 10 [MHz]	fs = 32.768 [kHz]	fcgck = 10 [MHz]	fs = 32.768 [kHz]
	SYSCR1 <DV9CK> = 0	SYSCR1 <DV9CK> = 1					
00	fcgck / 2^{10} [Hz]	fs / 2^3 [Hz]	fs / 2^3 [Hz]	102.4 [μs]	244.1 [μs]	6.7 [s]	16 [s]
01	fcgck / 2^6 [Hz]	fcgck / 2^6 [Hz]	-	6.4 [μs]	-	419.4 [ms]	-
10	fcgck / 2^2 [Hz]	fcgck / 2^2 [Hz]	-	400 [ns]	-	26.2 [ms]	-
11	fcgck / 2 [Hz]	fcgck / 2 [Hz]	-	200 [ns]	-	13.1 [ms]	-

13.5.1.2. Operation

Setting TA0CR<TA0S> to "1" enables the 16-bit up counter to increment based on the selected internal source clock. When a match between the up-counter value and the value set to timer register A (TA0DRA) is detected, an INTTCA0 interrupt request is generated and the up counter is cleared to "0x0000". After being cleared, the up counter continues counting. Setting TA0CR<TA0S> to "0" while the timer counter is operated causes that the up counter is stopped to count and the up counter is cleared to "0x0000".

13.5.1.3. Auto capture

The latest contents of the up counter can be taken into timer register B (TA0DRB) by setting TA0CR<TA0ACAP> to "1" (auto capture function). When TA0CR<TA0ACAP> is "1", the current contents of the up counter can be read by reading from TA0DRBL. TA0DRBH is loaded at the same time as TA0DRBL is read. Therefore, when reading the captured value, be sure to read TA0DRBL and TA0DRBH in this order.

The auto capture function can be used whether the timer counter is operating or stopped. When the timer counter is stopped, TA0DRBL is read as "0x00". TA0DRBH keeps the captured value after the timer counter stops, but it is cleared to "0x00" when TA0DRBL is read while the timer counter is stopped.

When the timer counter is started with TA0CR<TA0ACAP> which is set to "1", the auto capture is enabled immediately after the timer counter is started.

Note: The value set to TA0CR<TA0ACAP> cannot be changed at the same time as TA0CR<TA0S> is rewritten from "1" to "0". (This setting is ignored.)

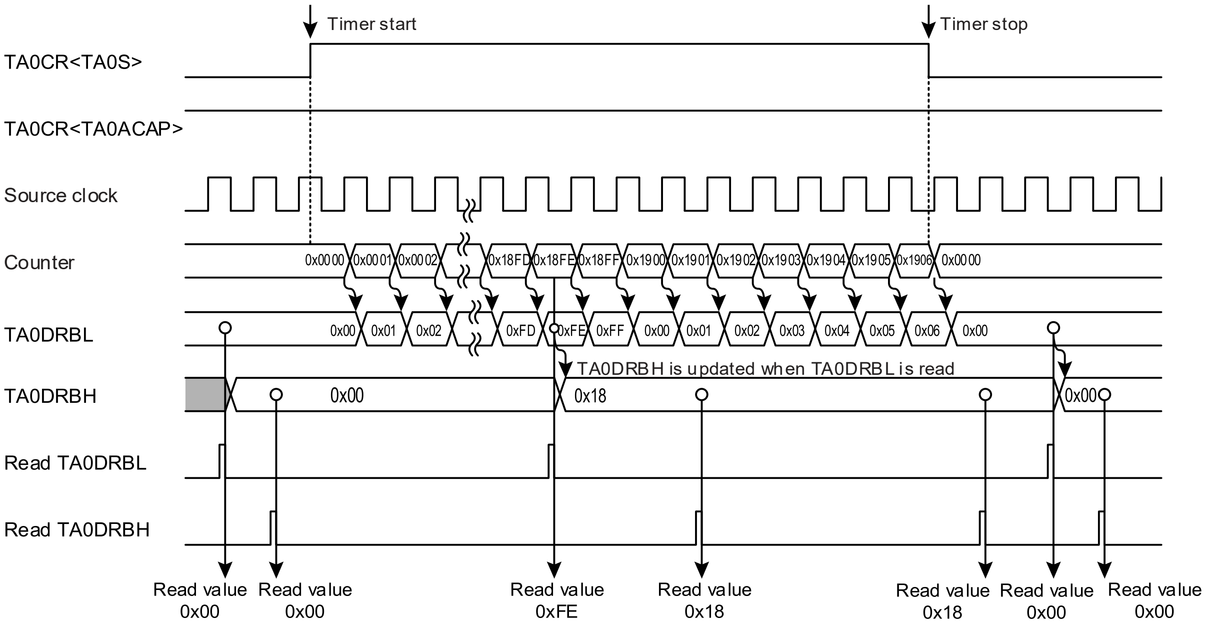


Figure 13-2 Timer Mode Timing Chart (Auto Capture)

13.5.1.4. Register buffer configuration

(1) Temporary buffer

The TMP89FS60B/62B/63B contains an 8-bit temporary buffer. When a write instruction to TA0DRAL is executed, the data is stored into this temporary buffer, whether the double buffer is enabled or disabled.

Subsequently, when a write instruction to TA0DRAH is executed, the written value is stored into the double buffer or TA0DRAH. At the same time, the written value in the temporary buffer is stored into the double buffer or TA0DRAL. (This structure is designed to enable the set values of the lower-level and higher-level registers simultaneously.) Therefore, when writing data to TA0DRA, be sure to write the data into TA0DRAL and TA0DRAH in this order.

See Figure 13-1 for the temporary buffer configuration.

(2) Double buffer

In the TMP89FS60B/62B/63B, the double buffer can be used by setting TA0MOD<TA0DBE>. Setting TA0MOD<TA0DBE> to "0" disables the double buffer. Setting TA0MOD<TA0DBE> to "1" enables the double buffer.

See Figure 13-1 for the double buffer configuration.

– When the double buffer is enabled

When a write instruction to TA0DRAH is executed while the timer counter is operated, the written value is stored into the double buffer. TA0DRAH and TA0DRAL are not updated.

TA0DRAH and TA0DRAL compare the up counter value to the last written values. When the values are matched, the INTTCA0 interrupt request is generated and the value set to double buffer is stored to TA0DRAH and TA0DRAL. Subsequently, the match detection is executed using a new stored value.

When a read instruction from TA0DRAH and TA0DRAL is executed, the value in the double buffer (the last written value) is read, not the values in TA0DRAH and TA0DRAL (the current effective values).

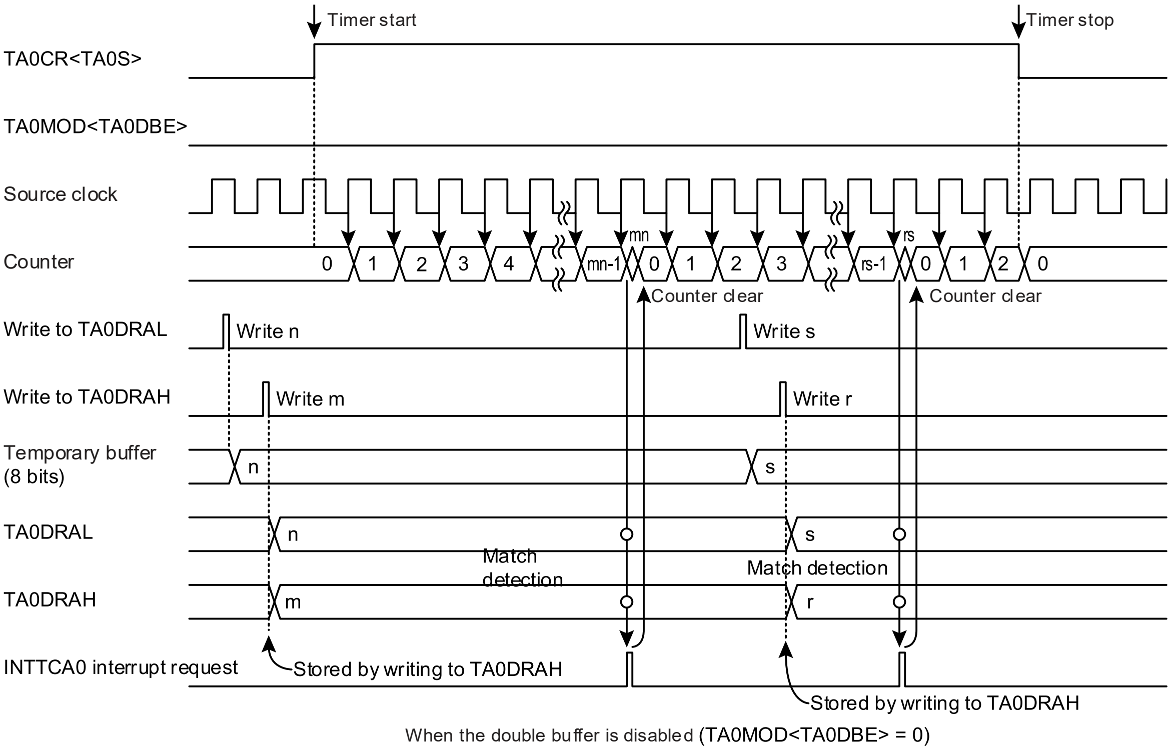
When a write instruction to TA0DRAH and TA0DRAL is executed while the timer counter is stopped, the written value is immediately stored into the double buffer, TA0DRAH and TA0DRAL.

– When the double buffer is disabled

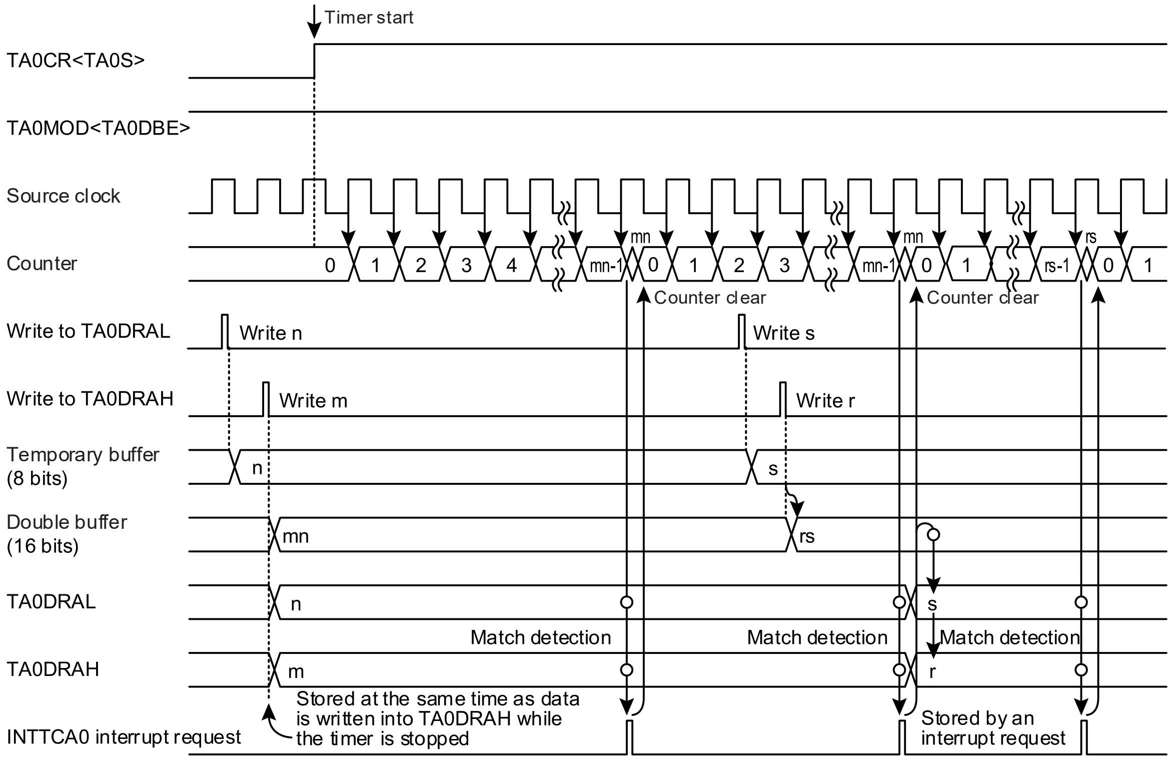
When a write instruction to TA0DRAH is executed while the timer counter is operated, the written value is stored into TA0DRAH and TA0DRAL. Subsequently, the match detection is executed using a new stored value.

When the values written to TA0DRAH and TA0DRAL are smaller than the up counter value, the match detection is executed using a new written values after the up counter is overflowed. Therefore, the interrupt request interval may be longer than the specified period. When this is a problem, enable the double buffer.

When a write instruction to TA0DRAH and TA0DRAL is executed while the timer counter is stopped, the set value is immediately stored into TA0DRAH and TA0DRAL.



When the double buffer is disabled (TA0MOD<TA0DBE> = 0)



When the double buffer is enabled (TA0MOD<TA0DBE> = 1)

Figure 13-3 Timer Mode Timing Chart

13.5.2. External trigger timer mode

In the external trigger timer mode, the up counter starts counting when TCA0 is triggered by the input to the TCA0 pin.

13.5.2.1. Setting

Setting the operation mode selection TA0MOD<TA0M> to “100" activates the external trigger timer mode. Select the source clock at TA0MOD<TA0CK>.

Select the trigger edge at the trigger edge input selection TA0MOD<TA0TED>. Clearing TA0MOD<TA0TED> to "0" selects the rising edge, and setting it to "1" selects the falling edge.

Note that this mode uses the TCA0 input pin, and the TCA0 pin must be set to the input mode beforehand in port settings.

The operation is started by setting TA0CR<TA0S> to "1". After the timer counter is started, writing to TA0MOD and TA0CR<TA0OVE> is disabled. Be sure to complete the required mode settings before starting the timer counter.

13.5.2.2. Operation

After the timer counter is started, when the selected trigger edge is input to the TCA0 pin, the up counter starts counting by the selected source clock. When a match between the up counter value and the value set to TA0DRA is detected, the INTTCA0 interrupt request is generated and the up counter is cleared to "0x0000". After being cleared, the up counter continues counting.

When TA0MOD<TA0METT> is "1" and the edge opposite to the selected trigger edge is detected, the up counter stops counting and is cleared to "0x0000". Subsequently, when the selected trigger edge is detected, the up counter restarts counting.

In this mode, the INTTCA0 interrupt request can be generated by detecting that the input pulse to TCA0 pin exceeds a certain pulse width. When TA0MOD<TA0METT> is "0", the selected edge and the opposite edge do not detect during the period from the detection of the specified trigger edge and the start of counting through until the match detection.

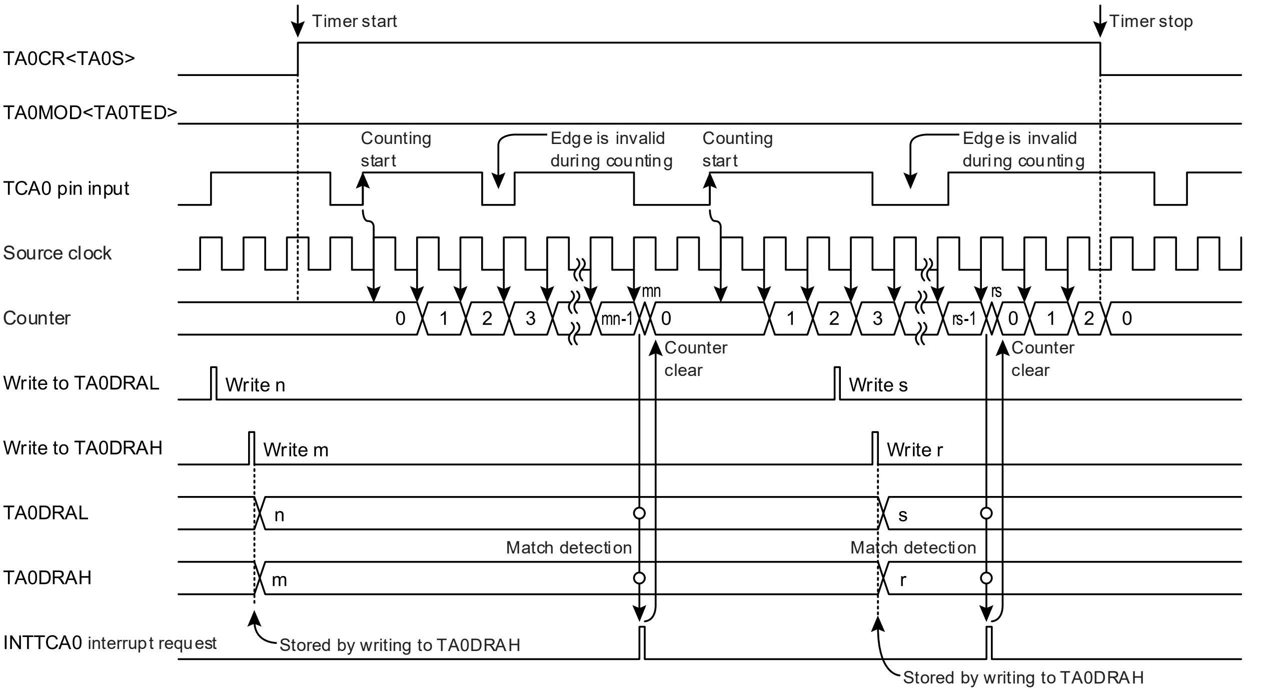
Setting TA0CR<TA0S> to "0" while the timer counter is operated causes the up counter to stop counting and be cleared to "0x0000".

13.5.2.3. Auto capture

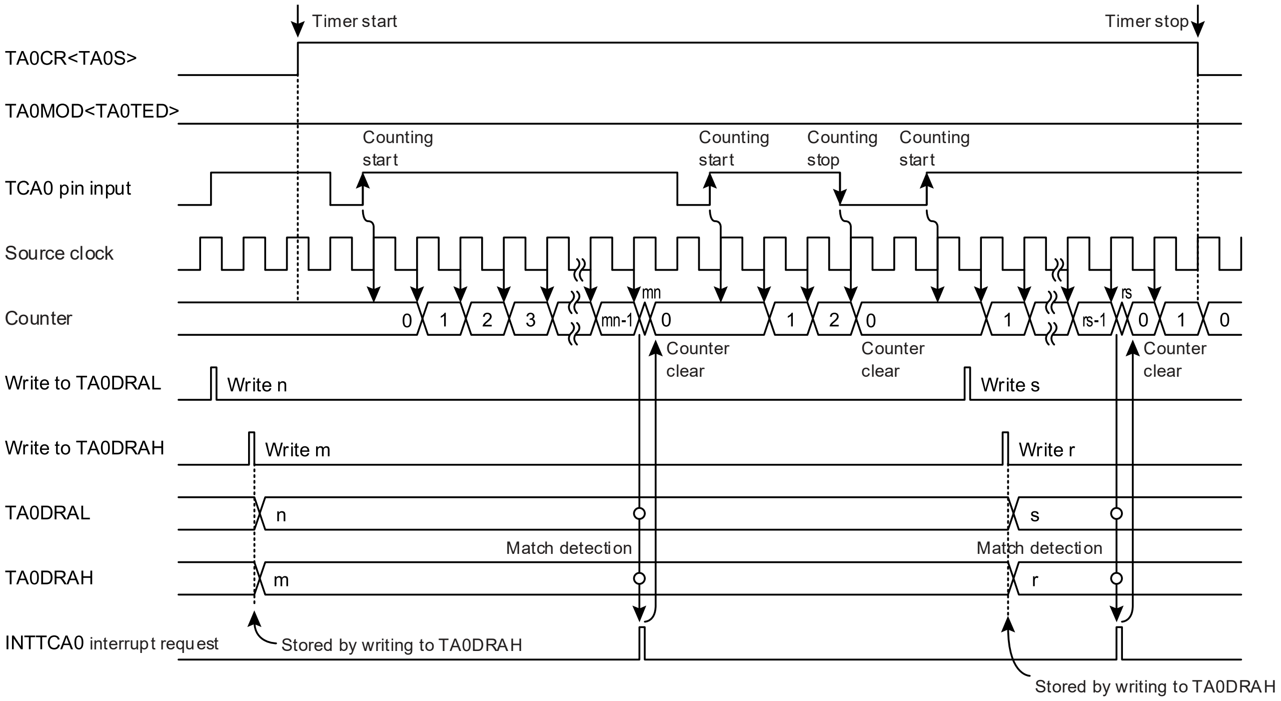
Refer to "13.5.1.3. Auto capture".

13.5.2.4. Register buffer configuration

Refer to "13.5.1.4. Register buffer configuration".



When the trigger is started (TA0MOD<TA0METT> = 0)



When the trigger is started and stopped (TA0MOD<TA0METT> = 1)

Figure 13-4 External Trigger Timer Timing Chart

13.5.3. Event counter mode

In the event counter mode, the up counter counts up at the edge of the input to the TCA0 pin.

13.5.3.1. Setting

Setting the operation mode selection TA0MOD<TA0M> to "010" activates the event counter mode. Set the edge of the input to the TCA0 pin at the external trigger input selection TA0MOD<TA0TED>. Clearing TA0MOD<TA0TED> to "0" selects the rising edge, and setting it to "1" selects the falling edge for counting up.

Note that this mode uses the TCA0 input pin, and the TCA0 pin must be set to the input mode beforehand in port settings.

The operation is started by setting TA0CR<TA0S> to "1". After the timer counter is started, writing to TA0MOD and TA0CR<TA0OVE> is disabled. Be sure to complete the required mode settings before starting the timer counter.

13.5.3.2. Operation

After the event counter mode is started, when the selected edge is input to the TCA0 pin, the up counter increments.

When a match between the up counter value and the value set to TA0DRA is detected, the INTTCA0 interrupt request is generated and the up counter is cleared to "0x0000". After being cleared, the up counter continues counting and counts up at each edge of the input to the TCA0 pin. Setting TA0CR<TA0S> to "0" while the timer counter is operated causes the up counter to stop counting and be cleared to "0x0000".

The maximum frequency which is supplied to TCA0 pin is fcgck / 2 [Hz] (in the NORMAL1/2 or IDLE1/2 mode) or fs / 2 [Hz] (in the SLOW1/2 or SLEEP1 mode), and a pulse width of two machine cycles or more is required at both the “High” and “Low” levels.

13.5.3.3. Auto capture

Refer to "13.5.1.3. Auto capture".

13.5.3.4. Register buffer configuration

Refer to "13.5.1.4. Register buffer configuration".

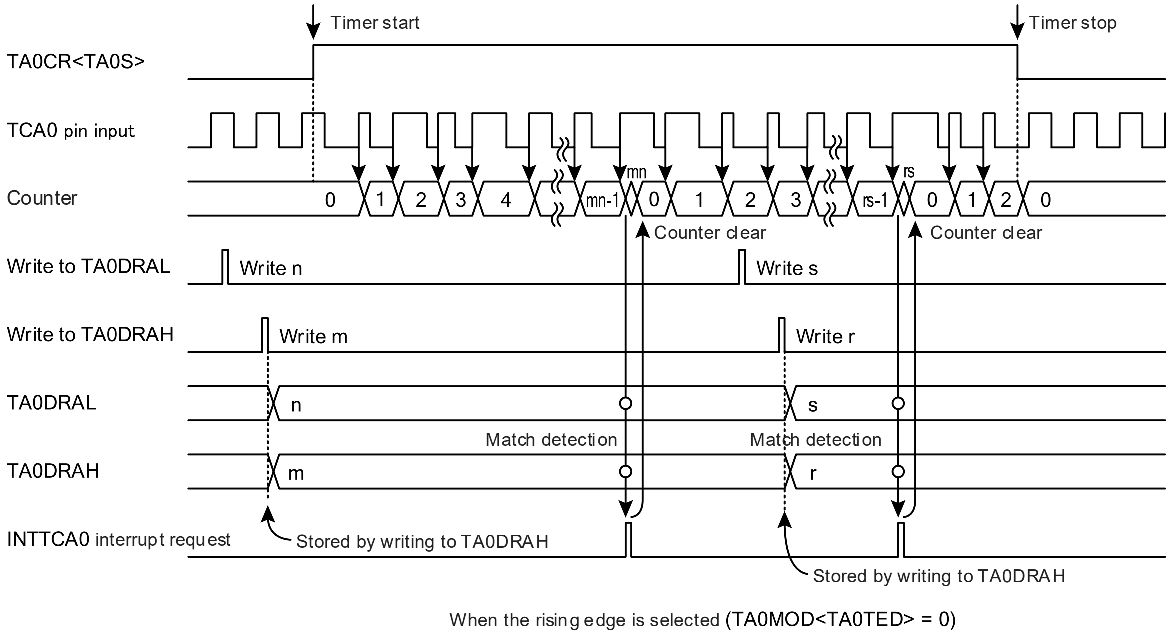


Figure 13-5 Event Count Mode Timing Chart

13.5.4. Window mode

In the window mode, the up counter counts up at the rising edge of the pulse that is logical ANDed the input pulse to the TCA0 pin (window pulse) and the source clock.

13.5.4.1. Setting

Setting the operation mode selection TA0MOD<TA0M> to "101" activates the window mode. Select the source clock at TA0MOD<TA0CK>.

Select the window pulse level at the trigger edge input selection TA0MOD<TA0TED>. Clearing TA0MOD<TA0TED> to "0" enables counting up as long as the window pulse is at the “High” level. Setting TA0MOD<TA0TED> to "1" enables counting up as long as the window pulse is at the “Low” level.

Note that this mode uses the TCA0 input pin, and the TCA0 pin must be set to the input mode beforehand in port settings.

The operation is started by setting TA0CR<TA0S> to "1". After the timer counter is started, writing to TA0MOD and TA0CR<TA0OVE> is disabled. Be sure to complete the required mode settings before starting the timer counter.

13.5.4.2. Operation

After the operation is started, when the level selected at TA0MOD<TA0TED> is input to the TCA0 pin, the up counter counts up according to the source clock selected at TA0MOD<TA0CK>. When a match between the up counter value and the value set to TA0DRA is detected, the INTTCA0 interrupt request is generated and the up counter is cleared to "0x0000". After being cleared, the up counter restarts counting.

The maximum frequency which is supplied to TCA0 pin must be slow enough for the program to analyze the count value. Define a frequency pulse that is sufficiently lower than the selected source clock.

Setting TA0CR<TA0S> to "0" while the timer counter is operated causes the up counter to stop counting and be cleared to "0x0000".

13.5.4.3. Auto capture

Refer to "13.5.1.3. Auto capture".

13.5.4.4. Register buffer configuration

Refer to "13.5.1.4. Register buffer configuration".

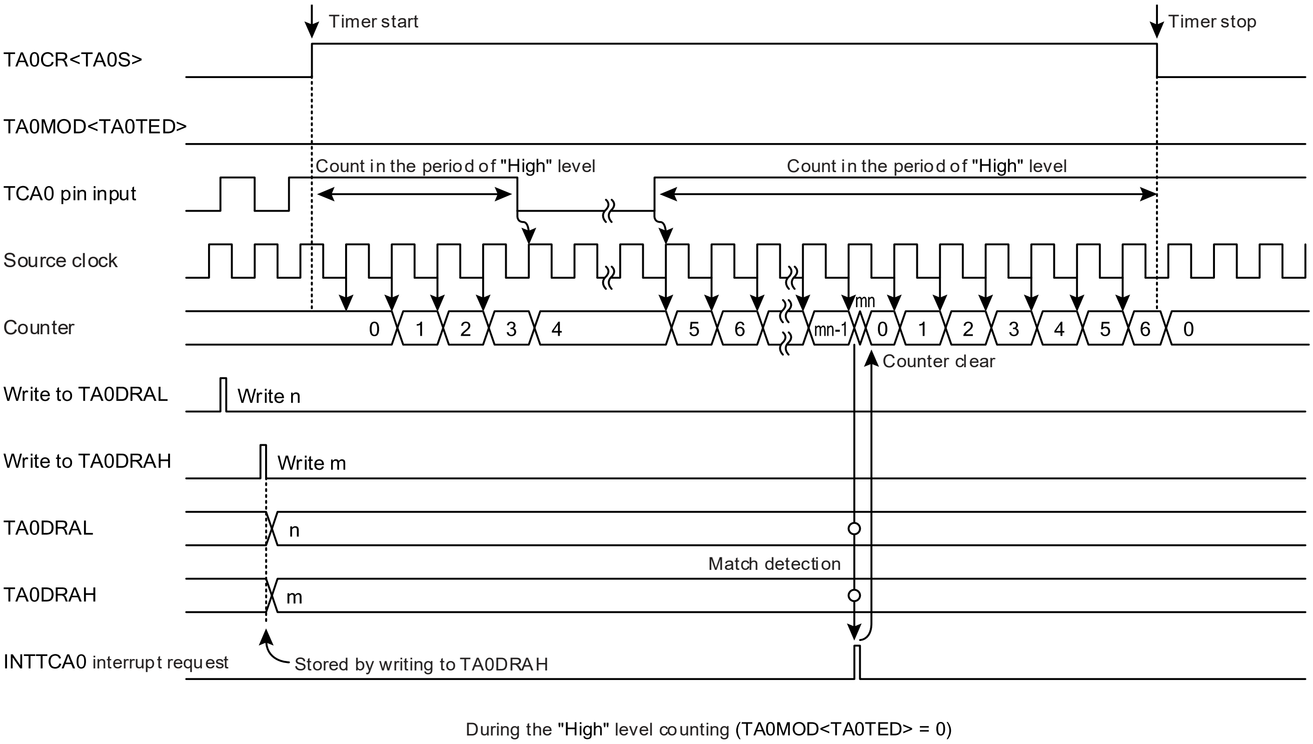


Figure 13-6 Window Mode Timing Chart

13.5.5. Pulse width measurement mode

In the pulse width measurement mode, the up counter starts counting at the rising or falling edge of the input to the TCA0 pin and measures the input pulse width based on the source clock.

13.5.5.1. Setting

Setting the operation mode selection TA0MOD<TA0M> to "110" activates the pulse width measurement mode. Select the source clock at TA0MOD<TA0CK>.

Select the trigger edge at the trigger edge input selection TA0MOD<TA0TED>. Clearing TA0MOD<TA0TED> to "0" selects the rising edge, and setting it to "1" selects the falling edge as a trigger to start the capture.

The operation after capturing is determined by the pulse width measurement mode control TA0MOD<TA0MCAP>. Clearing TA0MOD<TA0MCAP> to "0" selects the both edge capture. Setting TA0MOD<TA0MCAP> to "1" selects the single-edge capture.

The operation to be executed in case of an overflow of the up counter can be selected at the overflow interrupt control TA0CR<TA0OVE>. Setting <TA0OVE> to "1" makes the INTTCA0 interrupt request occur in case of an overflow. Setting <TA0OVE> to "0" makes no INTTCA0 interrupt request occur in case of an overflow.

Note that this mode uses the TCA0 input pin, and the TCA0 pin must be set to the input mode beforehand in port settings.

The operation is started by setting TA0CR<TA0S> to "1". In this time, TA0DRA and TA0DRB register are initialized to "0x0000". After the timer counter is started, writing to TA0MOD and TA0CR<TA0OVE> is disabled. Be sure to complete the required mode settings before starting the timer counter.

13.5.5.2. Operation

After the timer counter is started, when the selected trigger edge (start edge) is input to the TCA0 pin, the INTTCA0 interrupt request is generated, and then the up counter counts up according to the selected source clock. Subsequently, when the edge opposite to the selected edge is detected, the up counter value is captured into TA0DRB, the INTTCA0 interrupt request is generated, and TA0SR<TA0CPFB> is set to "1". Depending on the TA0MOD<TA0MCAP> setting, the operation differs as follows:

- Single-edge capture (When TA0MOD<TA0MCAP> is "1")

  The up counter stops counting up and is cleared to "0x0000" when the edge opposite to the selected edge is detected. Subsequently, when the start edge is input, the INTTCA0 interrupt request is generated, and then the up counter restarts increment.

- Both edge capture (When TA0MOD<TA0MCAP> is "0")

  The up counter continues counting up after the edge opposite to the selected edge is detected. Subsequently, when the selected trigger edge is input, the up counter value is captured into TA0DRA, the INTTCA0 interrupt request is generated, and TA0SR<TA0CPFA> is set to "1". At this time, the up counter is cleared to "0x0000".

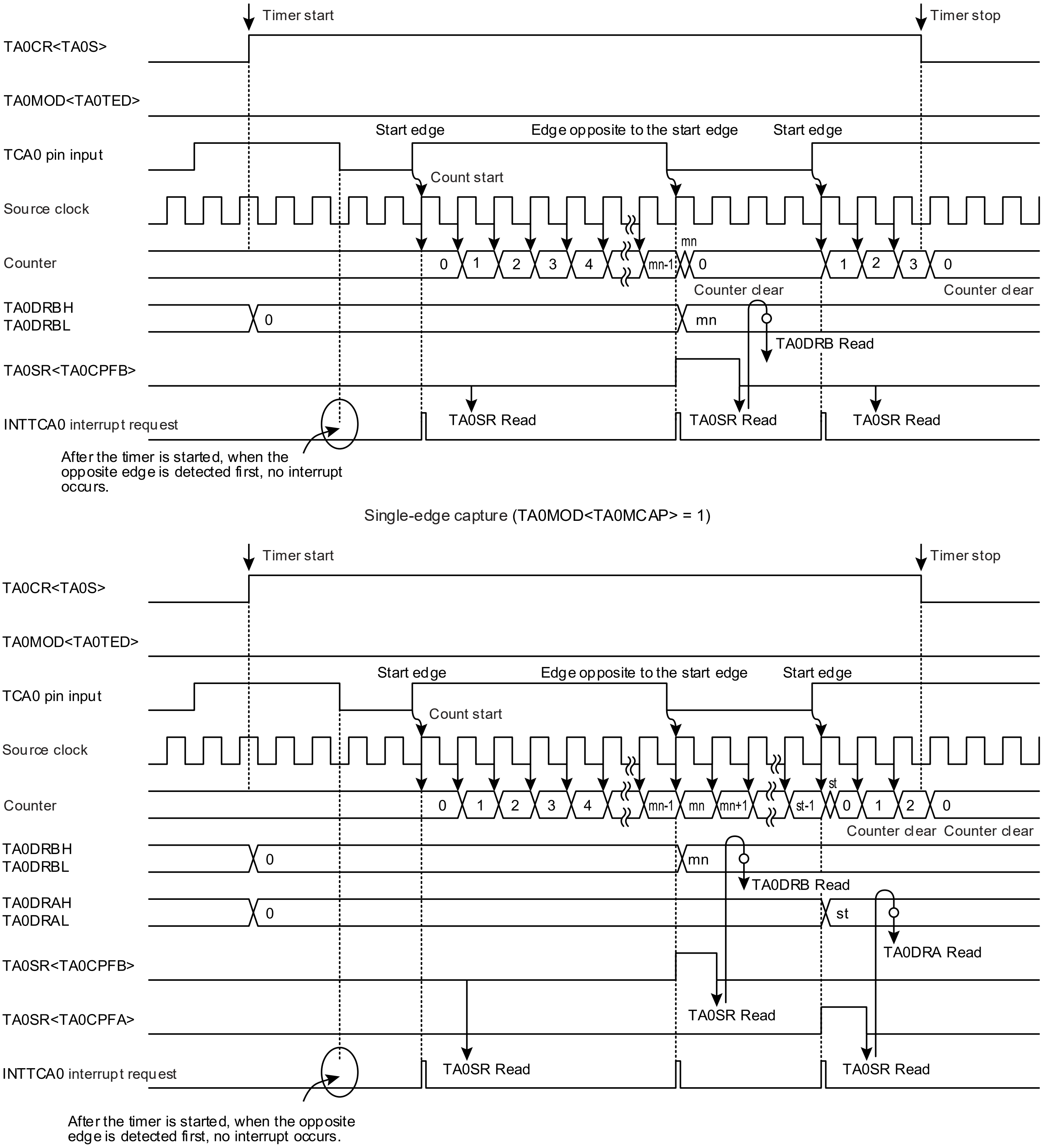
When the up counter overflows during capturing, the overflow flag TA0SR<TA0OVF> is set to "1". At this time, the INTTCA0 interrupt request occurs when the overflow interrupt control TA0CR<TA0OVE> is set to "1".

The capture completion flags (TA0SR<TA0CPFA>,TA0SR<TA0CPFB>) and the overflow flag (TA0SR<TA0OVF>) are cleared to "0" automatically when TA0SR is read.

The captured value must be read from TA0DRB (and also from TA0DRA for the both edge capture) before the next trigger edge is detected. When the captured value is not read, it becomes undefined. TA0DRA and TA0DRB must be read by using a 16-bit access instruction.

Clearing TA0CR<TA0S> to "0" while the timer counter is operated causes the up counter to stop counting and be cleared to "0x0000".

Note: After the timer counter is started, when the edge opposite to the selected trigger edge is detected first, no capture is executed and no INTTCA0 interrupt request occurs. In this case, the capture starts when the selected trigger edge is detected next.



After the timer is started, when the opposite edge is detected first, no interrupt occurs.

Single-edge capture (TA0MOD<TA0MCAP> = 1)

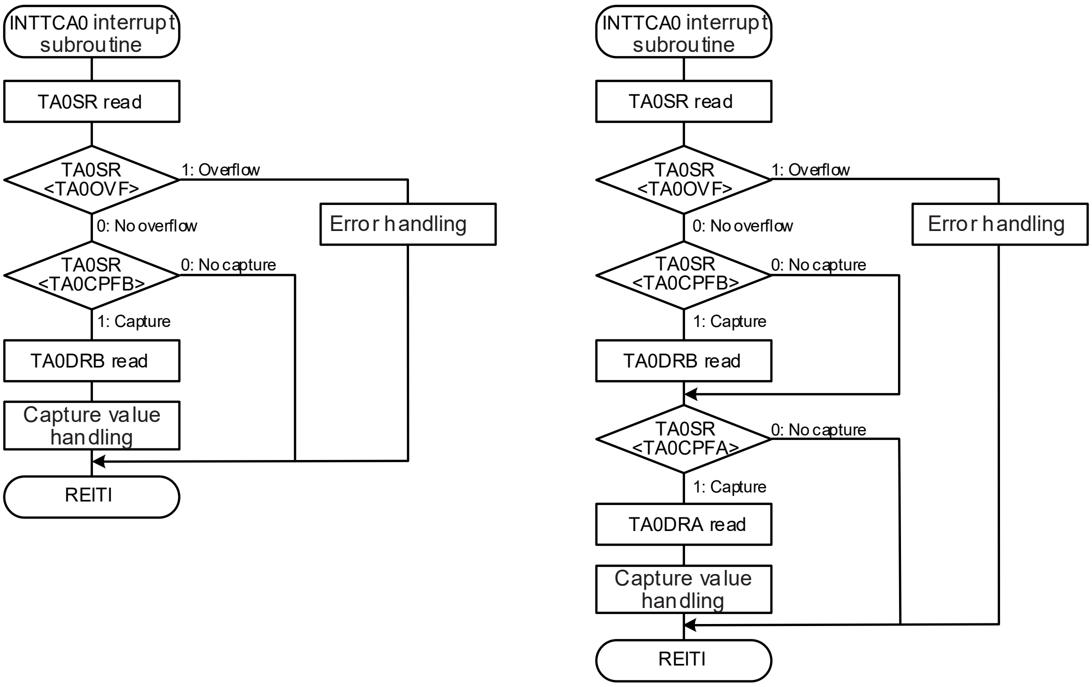
After the timer is started, when the opposite edge is detected first, no interrupt occurs.

Double-edge capture (TA0MOD<TA0MCAP> = 0)

Figure 13-7 Pulse Width Measurement Mode Timing Chart

13.5.5.3. Capture process

Figure 13-8 shows an example of the capture process for the INTTCA0 interrupt subroutine. The capture edge or overflow state can be easily judged by status register (TA0SR)



Interrupt process for single-edge capture

Interrupt process for double-edge capture

Figure 13-8 Example of capture process

13.5.6. Programmable pulse generate (PPG) mode

In the PPG output mode, an arbitrary duty pulse of an arbitrary cycle is output by two timer registers.

13.5.6.1. Setting

Setting the operation mode selection TA0MOD<TA0M> to "011" activates the PPG output mode. Select the source clock at TA0MOD<TA0CK>. Select continuous or one-shot PPG output at TA0CR<TA0MPPG>.

Set the PPG output cycle at TA0DRA and set the time until the output is reversed first at TA0DRB. Be sure to set register values so that TA0DRA is larger than TA0DRB.

Note that this mode uses the $\overline{\text{PPGA0}}$ pin. The $\overline{\text{PPGA0}}$ pin must be set to the output mode beforehand in port settings.

Set the initial state of the $\overline{\text{PPGA0}}$ pin at the timer flip-flop TA0CR<TA0TFF>. Setting TA0CR<TA0TFF> to "1" selects the “High” level as the initial state of the $\overline{\text{PPGA0}}$ pin. Clearing TA0CR<TA0TFF> to "0" selects the “Low” level as the initial state of the $\overline{\text{PPGA0}}$ pin.

The operation is started by setting TA0CR<TA0S> to "1". After the timer counter is started, writing to TA0MOD and TA0CR<TA0OVE> and <TA0TFF> is disabled. Be sure to complete the required mode settings before starting the timer counter.

13.5.6.2. Operation

After the timer counter is started, the up counter counts up.

When a match between the up counter value and the value set to tTA0DRB is detected, the $\overline{\text{PPGA0}}$ pin is changed to the “High” level when TA0CR<TA0TFF> is "0", or the $\overline{\text{PPGA0}}$ pin is changed to the “Low” level when TA0CR<TA0TFF> is "1".

Subsequently, the up counter continues counting. When a match between the up counter value and the value set to TA0DRA is detected, the $\overline{\text{PPGA0}}$ pin is changed to the “Low” level when TA0CR<TA0TEFF> is "0", or the $\overline{\text{PPGA0}}$ pin is changed to the “High” level when TA0CR<TA0TFF> is "1".

At this time, the INTTCA0 interrupt request occurs. When the PPG output control TA0CR<TA0MPPG> is set to "1" (one-shot), TA0CR<TA0S> is automatically cleared to "0" and the timer stops.

When TA0CR<TA0MPPG> is set to "0" (continuous), the up counter is cleared to "0x0000" and continues counting and PPG output. When TA0CR<TA0S> is cleared to "0" (including the auto stop by the one-shot operation) during the PPG output, the $\overline{\text{PPGA0}}$ pin returns to the level set in TA0CR<TA0TFF>.

TA0CR<TA0MPPG> can be changed during the operation. Changing TA0CR<TA0MPPG> from "1" to "0" during the operation cancels the one-shot operation and enables the continuous operation. Changing TA0CR<TA0MPPG> from "0" to "1" during the operation clears TA0CR<TA0S> to "0" and stops the up counter automatically after the current pulse output is completed.

The double buffer can be used for TA0DRA and TA0DRB. Setting TA0MOD<TA0DBE> to "1" enables the double buffer. When the values set to TA0DRA and TA0DRB are changed during the PPG output with the double buffer enabled, the writing to TA0DRA and TA0DRB will not immediately become effective but will become effective when a match between TA0DRA and the up counter is detected. When the double buffer is disabled, the writing to TA0DRA and TA0DRB will become effective immediately. When the written value is smaller than the up counter value, the up counter counts up from "0x0000" continuously after the up counter overflows. And then, the $\overline{\text{PPGA0}}$ pin is reversed when the up counter is matched with the written value.

13.5.6.3. Register buffer configuration

(1) Temporary buffer

The TMP89FS60B/62B/63B contains an 8-bit temporary buffer. When a write instruction to TA0DRAL is executed, the data is first stored into this temporary buffer, whether the double buffer is enabled or disabled.

Subsequently, when a write instruction to TA0DRAH is executed, the written value is stored into the double buffer or TA0DRAH. At the same time, the written value in the temporary buffer is stored into the double buffer or TA0DRAL . (This structure is designed to enable the set values of the lower-level register and the higher-level register simultaneously.) Therefore, when writing data to TA0DRA, be sure to write the data into TA0DRAL and TA0DRAH in this order.

The same operation is executed for TA0DRB. TA0DRA in the above explanation is replaced as TA0DRB for operation of TA0DRB.

See Figure 13-1 for the temporary buffer configuration.

(2) Double buffer

In the TMP89FS60B/62B/63B, the double buffer can be used by setting TA0MOD<TA0DBE>. Setting TA0MOD<TA0DBE> to "0" disables the double buffer. Setting TA0MOD<TA0DBE> to "1" enables the double buffer.

See Figure 13-1 for the double buffer configuration.

– When the double buffer is enabled

When a write instruction to TA0DRAH is executed while the timer counter is operated, the written value is stored into the double buffer. TA0DRAH and TA0DRAL are not updated.

TA0DRAH and TA0DRAL compare the up counter value to the last written values. When the values are matched, the INTTCA0 interrupt request is generated and the double buffer written value is stored into TA0DRAH and TA0DRAL. Subsequently, the match detection is executed using a new stored value.

When a read instruction from TA0DRAH and TA0DRAL is executed, the value in the double buffer (the last written value) is read, not the values in TA0DRAH and TA0DRAL (the current effective values).

When a write instruction to TA0DRAH and TA0DRAL is executed while the timer counter is stopped, the written value is immediately stored into the double buffer, and TA0DRAH and TA0DRAL.

The same operation is executed for TA0DRB. TA0DRA in the above explanation is replaced as TA0DRB for operation of TA0DRB.

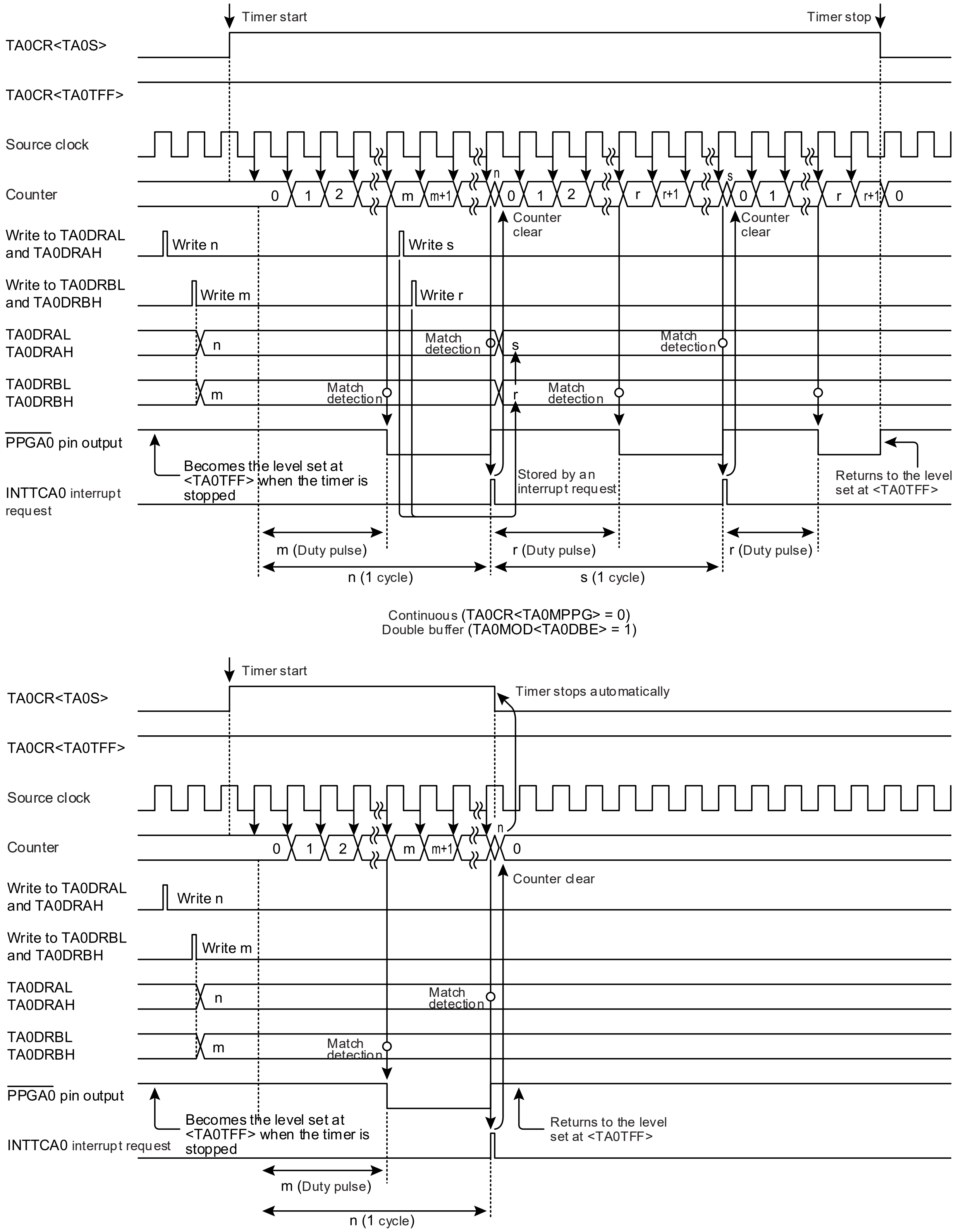
– When the double buffer is disabled

When a write instruction to TA0DRAH and TA0DRAL is executed while the timer counter is operated, the written value is stored into TA0DRAH and TA0DRAL. Subsequently, the match detection is executed using a new stored value.

When the values written to TA0DRAH and TA0DRAL are smaller than the up counter value, the up counter counts up from "0x0000" continuously after the up counter overflows. And then, the up counter is matched with the new written value. Therefore, the output pulse width may be longer than the written width. When that is a problem, enable the double buffer.

When a write instruction to TA0DRAH and TA0DRAL is executed while the timer counter is stopped, the set value is immediately stored into TA0DRAH and TA0DRAL.

The same operation is executed for TA0DRB. TA0DRA in the above explanation is replaced as TA0DRB for operation of TA0DRB.



Continuous (TA0CR<TA0MPPG> = 0)
Double buffer (TA0MOD<TA0DBE> = 1)

One-shot (TA0CR<TA0MPPG> = 1)

Figure 13-9 PPG Mode Timing Chart

13.6. Noise Canceller

The digital noise canceller can be used in the operation modes that use the TCA0 pin.

13.6.1. Setting

When the digital noise canceller is used, the input level is sampled at the sampling intervals set at TA0CR<TA0NC>. When the same level is detected three times consecutively, the level of the input to the timer is changed.

Setting TA0CR<TA0NC> to any values except "00" enables the noise canceller to start operation, regardless of the TA0CR<TA0S> value.

When the noise canceller is used, enable the timer to start after a period of time that is equal to four times or more the sampling interval after TA0CR<TA0NC> is set has elapsed. This stabilizes the input signal.

Set TA0CR<TA0NC> while the timer is stopped (TA0CR<TA0S> = 0). When TA0CR<TA0S> is "1", writing is ignored.

In the SLOW1/2 or SLEEP1 mode, setting TA0CR<TA0NC> to "11" selects fs / 2 as the source clock for the operation. Setting TA0CR<TA0NC> to "00" disables the noise canceller. Setting TA0CR<TA0NC> to "01" or "10" makes the TCA0 pin input invalid.

Table 13-5 Noise Cancel Time

	fcgck = 10 [MHz]		
<TA0NC>	**Sampling interval**	**Time canceled as noise**	**Time regarded as signal**
00	None	-	-
01	200 [ns] (2 / fcgck)	less than 600 [ns]	800 [ns] or more
10	400 [ns] (4 / fcgck)	less than 1.2 [μs]	1.6 [μs] or more
11	25.6 [μs] (256 / fcgck)	less than 76.8 [μs]	102.4 [μs] or more

14. 8-bit Timer Counter (TC0)

The TMP89FS60B/62B/63B contains the high-performance 8-bit timer counters (TC0). Each timer counter can be used for time measurement and pulse output with a prescribed width. When two TC0s are connected in a cascade, they also can be used as 16-bit timer counter.

This chapter describes the TC00 and TC01 when they are connected in an cascade.

For TC02 and TC03, replace the SFR addresses and pin names as shown in Table 14-1 and Table 14-2.

Table 14-1 SFR Address Assignment

	In the 16-bit timer mode	T0xREG (Address)	T0xPWM (Address)	T0xMOD (Address)	T0xxCR (Address)	Low power consumption register
TC00	Lower 8 bits	T00REG (0x0026)	T00PWM (0x0028)	T00MOD (0x002A)	T001CR (0x002C)	POFFCR0 <TC001EN>
TC01	Higher 8 bits	T01REG (0x0027)	T01PWM (0x0029)	T01MOD (0x002B)		
TC02	Lower 8 bits	T02REG (0x0F88)	T02PWM (0x0F8A)	T02MOD (0x0F8C)	T023CR (0x0F8E)	POFFCR0 <TC023EN>
TC03	Higher 8 bits	T03REG (0x0F89)	T03PWM (0x0F8B)	T03MOD (0x0F8D)		

Table 14-2 Pin Name

	Timer input pin	PWM output pin	PPG output pin
TC00	TC00 pin	$\overline{\text{PWM00}}$ pin	$\overline{\text{PPG00}}$ pin
TC01	TC01 pin	$\overline{\text{PWM01}}$ pin	$\overline{\text{PPG01}}$ pin
TC02	TC02 pin	$\overline{\text{PWM02}}$ pin	$\overline{\text{PPG02}}$ pin
TC03	TC03 pin	$\overline{\text{PWM03}}$ pin	$\overline{\text{PPG03}}$ pin

14.1. 8-bit Timer Counters for Each Product

The 8-bit timer counters (TC0) for each product is shown Table 14-3.

In regards to unavailable channel of the TC0, the corresponding bit of the low power register must be cleared to “0”.

Table 14-3 8-bit Timer Counters (TC0) for Each Product

	TMP89FS60B	TMP89FS62B	TMP89FS63B
TC00	A	A	A
TC01	A	A	A
TC02	A	A	A
TC03	A	A	A

Note: A: Available

14.2. Configuration

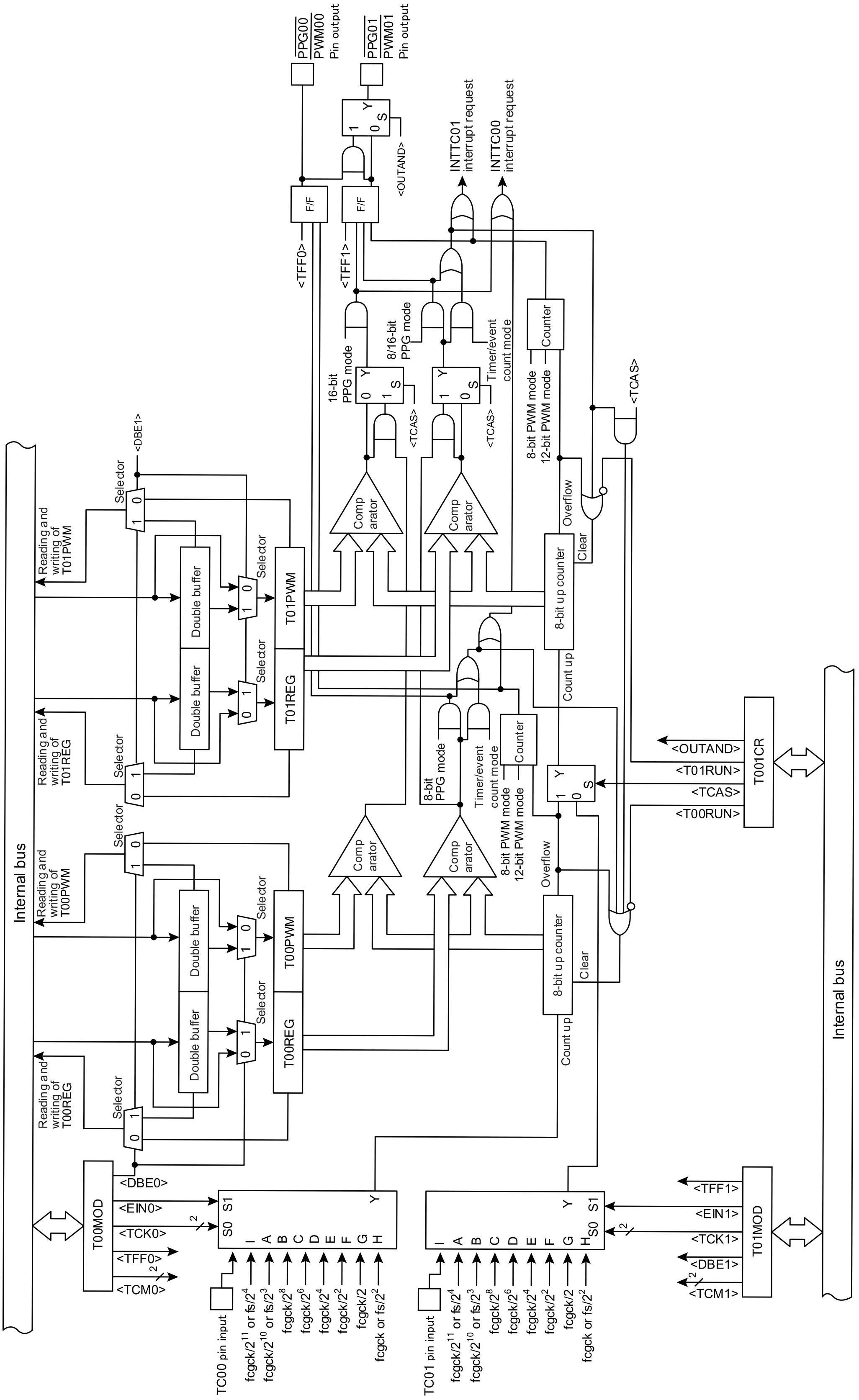


Figure 14-1   8-bit Timer Counters (TC00 and TC01)

14.3. Control

14.3.1. TC00

The TC00 is controlled by the timer counter 00 mode register (T00MOD) and two 8-bit timer registers (T00REG and T00PWM).

Timer register 00

T00REG		7	6	5	4	3	2	1	0
(0x0026)	Bit Symbol	T00REG							
	Read/Write	R/W							
	After reset	1	1	1	1	1	1	1	1

Timer register 00

T00PWM		7	6	5	4	3	2	1	0
(0x0028)	Bit Symbol	T00PWM							
	Read/Write	R/W							
	After reset	1	1	1	1	1	1	1	1

Note: For the configuration of T00PWM in the 8-bit and 12-bit PWM modes, refer to "14.5.3. 8-bit pulse width modulation (PWM) output mode" and "14.5.7. 12-bit pulse width modulation (PWM) output mode".

Timer counter 00 mode register

T00MOD		7	6	5	4	3	2	1	0
(0x002A)	Bit Symbol	TFF0	DBE0	TCK0			EIN0	TCM0	
	Read/Write	R/W	R/W	R/W			R/W	R/W	
	After reset	1	1	0	0	0	0	0	0

TFF0	Timer F/F0 control	0:	Clear		
		1:	Set		
DBE0	Double buffer control	0:	Disable the double buffer		
		1:	Enable the double buffer		
TCK0	Operation clock selection		NORMAL1/2 or IDLE1/2 mode		SLOW1/2 or SLEEP1 mode
			SYSCR1 <DV9CK> = 0	SYSCR1 <DV9CK> = 1	
		000:	fcgck / 2^{11}	fs / 2^{4}	fs / 2^{4}
		001:	fcgck / 2^{10}	fs / 2^{3}	fs / 2^{3}
		010:	fcgck / 2^{8}	fcgck / 2^{8}	-
		011:	fcgck / 2^{6}	fcgck / 2^{6}	-
		100:	fcgck / 2^{4}	fcgck / 2^{4}	-
		101:	fcgck / 2^{2}	fcgck / 2^{2}	-
		110:	fcgck / 2	fcgck / 2	-
		111:	fcgck	fcgck	fs / 2^{2}
EIN0	Selection for using external source clock	0:	Select the internal clock as the source clock.		
		1:	Select an external clock as the source clock. (the falling edge of the TC00 pin)		
TCM0	Operation mode selection	00:	8-bit timer/event counter modes		
		01:	8-bit timer/event counter modes		
		10:	8-bit pulse width modulation output (PWM) mode		
		11:	8-bit programmable pulse generate (PPG) mode		

Note 1: fcgck: Gear clock [Hz], fs: Low-frequency clock [Hz]

Note 2: Set T00MOD while the timer counter is stopped. Writing data into T00MOD is invalid while the timer counter is operated.

Note 3: In the 8-bit timer/event modes, the <TFF0> setting is invalid. In this mode, when the $\overline{\text{PWM00}}$ and $\overline{\text{PPG00}}$ pins are set as the function output pins in the port setting, the pins always output the “High” level.

Note 4: When <EIN0> is set to "1" and the external clock input is selected as the source clock, the <TCK0> setting is ignored.

Note 5: When the T001CR<TCAS> is "1", TC00 operates in the 16-bit mode. The T00MOD setting is invalid and TC00 cannot be used independently in this mode. When the $\overline{\text{PWM00}}$ and $\overline{\text{PPG00}}$ pins are set to the function output pins in the port setting, the pins always output the “High” level.

Note 6: When the 16-bit mode is selected at T001CR<TCAS>, the timer start is controlled at T001CR<T01RUN>. TC00 is not started by writing data into T001CR<T00RUN>.

14.3.2. TC01

The TC01 is controlled by timer counter 01 mode register (T01MOD) and two 8-bit timer registers (T01REG and T01PWM).

Timer register 01

T01REG		7	6	5	4	3	2	1	0
(0x0027)	Bit Symbol	T01REG							
	Read/Write	R/W							
	After reset	1	1	1	1	1	1	1	1

Timer register 01

T01PWM		7	6	5	4	3	2	1	0
(0x0029)	Bit Symbol	T01PWM							
	Read/Write	R/W							
	After reset	1	1	1	1	1	1	1	1

Note: For the configuration of T01PWM in the 8-bit and 12-bit PWM modes, refer to "14.5.3. 8-bit pulse width modulation (PWM) output mode" and "14.5.7. 12-bit pulse width modulation (PWM) output mode".

Timer counter 01 mode register

T01MOD		7	6	5	4	3	2	1	0
(0x002B)	Bit Symbol	TFF1	DBE1	TCK1			EIN1	TCM1	
	Read/Write	R/W	R/W	R/W			R/W	R/W	
	After reset	1	1	0	0	0	0	0	0

TFF1	Timer F/F1 control	0: 1:	Clear Set		
DBE1	Double buffer control	0: 1:	Disable the double buffer Enable the double buffer		
TCK1	Operation clock selection		NORMAL1/2 or IDLE1/2 mode		SLOW1/2 or SLEEP1 mode
			SYSCR1 <DV9CK> = 0	SYSCR1 <DV9CK> = 1	
		000:	fcgck / 2^{11}	fs / 2^4	fs / 2^4
		001:	fcgck / 2^{10}	fs / 2^3	fs / 2^3
		010:	fcgck / 2^8	fcgck / 2^8	-
		011:	fcgck / 2^6	fcgck / 2^6	-
		100:	fcgck / 2^4	fcgck / 2^4	-
		101:	fcgck / 2^2	fcgck / 2^2	-
		110:	fcgck / 2	fcgck / 2	-
		111:	fcgck	fcgck	fs / 2^2
EIN1	Selection for using external source clock	0: 1:	Select the internal clock as the source clock. Select an external clock as the source clock. (the falling edge of the TC01 pin)		
TCM1	Operation mode selection		T001CR<TCAS> = 0 (8-bit mode)	T001CR<TCAS> = 1 (16-bit mode)	
		00:	8-bit timer/event counter modes	16-bit timer/event counter modes	
		01:	8-bit timer/event counter modes	16-bit timer/event counter modes	
		10:	8-bit pulse width modulation output (PWM) mode	12-bit pulse width modulation output (PWM) mode	
		11:	8-bit programmable pulse generate (PPG) mode	16-bit programmable pulse generate (PPG) mode	

Note 1: fcgck: Gear clock [Hz], fs: Low-frequency clock [Hz]

Note 2: Set T01MOD while the timer counter is stopped. Writing data into T01MOD is invalid while the timer counter is operated.

Note 3: In the 8-bit timer/event counter modes, the <TFF1> setting is invalid. In this mode, when the $\overline{\text{PWM01}}$ and $\overline{\text{PPG01}}$ pins are set as the function output pins in the port setting, the pins always output the “High” level.

Note 4: When <EIN1> is set to "1" and the external clock input is selected as the source clock, the <TCK1> setting is ignored.

14.3.3. Common to TC00 and TC01

The TC00 and TC01 have the low power consumption register (POFFCR0) and timer 00 and 01 control registers in common.

Low power consumption register 0

POFFCR0		7	6	5	4	3	2	1	0
(0x0F74)	Bit Symbol	-	-	TC023EN	TC001EN	-	-	(TCA1EN)	(TCA0EN)
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

TC023EN	TC02 and TC03 control	0: 1:	Disable Enable
TC001EN	TC00 and TC01 control	0: 1:	Disable Enable

Note: The written values are read from the bits 7 and 6, 3 and 2 of POFFCR0. These bits must be cleared to “0”.

Timer counter 00 and 01 control register

T001CR		7	6	5	4	3	2	1	0
(0x002C)	Bit Symbol	-	-	-	-	OUTAND	TCAS	T01RUN	T00RUN
	Read/Write	R	R	R	R	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

OUTAND	TC00 and TC01 output control	0: 1:	Output the TC00 output from the $\overline{\text{PWM00}}$ and $\overline{\text{PPG00}}$ pins. Output a logical ANDed pulse of the outputs of TC00 and TC01 from the $\overline{\text{PWM01}}$ and $\overline{\text{PPG01}}$ pins.
TCAS	TC00 and TC01 cascade control	0: 1:	Use TC00 and TC01 independently (8-bit mode). Use TC00 and TC01 which are cascaded (16-bit mode).
T01RUN	TC01 control (8-bit mode) TC00 and TC01 control (16-bit mode)	0: 1:	Stop and clear the counter Start
T00RUN	TC00 control	0: 1:	Stop and clear the counter Start

Note 1: When STOP mode is started, <T00RUN> and <T01RUN> are cleared to "0" and the timer counters stop. Set T001CR again to use TC00 and TC01 after STOP mode is released.

Note 2: When a read instruction is executed on T001CR, bits 7 to 4 are read as "0".

Note 3: When <OUTAND> is "1", output is obtained from the $\overline{\text{PWM01}}$ and $\overline{\text{PPG01}}$ pins only. There is no timer output to the $\overline{\text{PWM00}}$ and $\overline{\text{PPG00}}$ pins. When the $\overline{\text{PWM00}}$ and $\overline{\text{PPG00}}$ pins are set as the function output pins in the port setting, the pins always output “High”.

Note 4: <OUTAND> and <TCAS> can be changed only when both <T01RUN> and <T00RUN> are "0". When either <T01RUN> or <T00RUN> is "1" or both are "1", the these values cannot be changed by executing write instructions to <OUTAND> and <TCAS>. <OUTAND> and <TCAS> can be changed at the same time as <T01RUN> and <T00RUN> are changed from "0" to "1".

14.3.4. Operation Modes and Usable Source Clocks

The operations modes of the TC0 and the usable source clocks are listed below.

Table 14-4 Operation Modes and Usable Source Clocks (in NORMAL1/2 or IDLE1/2 modes)

<TCK0>		000	001	010	011	100	101	110	111	TC0i Pin input
Operation mode		fcgck / 2^{11} or fs / 2^4	fcgck / 2^{10} or fs / 2^3	fcgck / 2^8	fcgck / 2^6	fcgck / 2^4	fcgck / 2^2	fcgck / 2	fcgck	
8-bit timer modes	8-bit timer	A	A	A	A	A	A	A	A	NA
	8-bit event counter	NA	NA	NA	NA	NA	NA	NA	NA	A
	8-bit PWM	A	A	A	A	A	A	A	A	NA
	8-bit PPG	A	A	A	A	A	A	A	A	NA
16-bit timer modes	16-bit timer	A	A	A	A	A	A	A	A	NA
	16-bit event counter	NA	NA	NA	NA	NA	NA	NA	NA	A
	12-bit PWM	A	A	A	A	A	A	A	A	A
	16-bit PPG	A	A	A	A	A	A	A	A	A

Note 1: A: Available, NA: Not available

Note 2: Set the source clock in the 16-bit modes on T01MOD<TCK1>.

Note 3: When the low-frequency clock fs is not oscillating, it must not be selected as the source clock. When fs is selected and is not oscillating, no source clock is supplied to the timer counter, and the timer counter remains stopped.

Note 4: i = 1 and 0 (Can be selected only “0” for i in the 16-bit modes)

Table 14-5 Operation Modes and Usable Source Clocks (SLOW1/2 and SLEEP1 modes)

<TCK0>		000	001	010	011	100	101	110	111	TC0i Pin input
動作モード		fs / 2^4	fs / 2^3	-	-	-	-	-	fs / 2^2	
8-bit timer modes	8-bit timer	A	A	NA	NA	NA	NA	NA	A	NA
	8-bit event counter	NA	NA	NA	NA	NA	NA	NA	NA	A
	8-bit PWM	A	A	NA	NA	NA	NA	NA	A	NA
	8-bit PPG	A	A	NA	NA	NA	NA	NA	A	NA
16-bit timer modes	16-bit timer	A	A	NA	NA	NA	NA	NA	A	NA
	16-bit event counter	NA	NA	NA	NA	NA	NA	NA	NA	A
	12-bit PWM	A	A	NA	NA	NA	NA	NA	A	A
	16-bit PPG	A	A	NA	NA	NA	NA	NA	A	A

Note 1: A: Available, NA: Not available

Note 2: Set the source clock in the 16-bit modes on T01MOD<TCK1>.

Note 3: i = 1 and 0 (Can be selected only “0” for i in the 16-bit modes)

14.4. Low Power Consumption Control

The TC00 and TC01 have the low power consumption registers (POFFCR0) that save power when the timer counters are not used.

Clearing POFFCR0<TC001EN> to "0" disables the basic clock supply to the TC00 and TC01 to save power. Note that this makes the TC00 and TC01 unusable. Setting POFFCR0<TC001EN> to "1" enables the basic clock supply to the TC00 and TC01 and enable the TC00 and TC01 to operate.

After reset, POFFCR0<TC001EN> are initialized to "0", and this makes the TC00 and TC01 unusable. When using the TC00 and TC01 for the first time, be sure to set POFFCR0<TC001EN> to "1" in the initial setting of the program (before the TC00 and TC01 control registers are modified).

Do not change POFFCR0<TC001EN> to "0" while the TC00 and TC01 are operated. Otherwise the TC00 and TC01 may operate unexpectedly.

14.5. Function

The TC00 and TC01 have 8-bit mode in which they are used independently and 16-bit mode in which they are cascaded.

The 8-bit mode includes four operation modes; the 8-bit timer mode, the 8-bit event counter mode, the 8-bit pulse width modulation output (PWM) mode and the 8-bit programmable pulse generated output (PPG) mode.

The 16-bit mode includes four operation modes; the 16-bit timer mode, the 16-bit event counter mode, the 12-bit PWM mode and the 16-bit PPG mode.

14.5.1. 8-bit timer mode

In the 8-bit timer mode, the up-counter counts up using the internal clock, and interrupt can be generated regularly at specified times.

14.5.1.1. Setting

TC00 is put into the 8-bit timer mode by setting T00MOD<TCM0> to "00" or "01", T001CR<TCAS> to "0" and T00MOD<EIN0> to "0". Select the source clock at T00MOD<TCK0>. Set the count value to be used for the match detection as an 8-bit value at the timer register T00REG.

Set T00MOD<DBE0> to "1" to use the double buffer.

Setting T001CR<T00RUN> to "1" starts the operation. After the timer counter is started, writing to T00MOD becomes invalid. Be sure to complete the required mode settings before starting the timer counter.

14.5.1.2. Operation

Setting T001CR<T00RUN> to "1" enables the 8-bit up counter to count up based on the selected internal source clock. When a match between the up counter value and the T00REG set value is detected, the INTTC00 interrupt request is generated and the up counter is cleared to "0x00". After being cleared, the up counter restarts counting. Setting T001CR<T00RUN> to "0" while the timer counter is operated makes the up counter stop counting and be cleared to "0x00".

14.5.1.3. Double buffer

In the TMP89FS60B/62B/63B, the double buffer for T00REG can be used by setting T00MOD<DBE0>. The double buffer is disabled by setting T00MOD<DBE0> to "0" or enabled by setting T00MOD<DBE0> to "1".

- When the double buffer is enabled

  When a write instruction to T00REG is executed while the timer counter is operated, the written value is stored in the double buffer, and T00REG is not updated. T00REG compares the previous written value with the up counter value. When the values match, the INTTC00 interrupt request is generated and the double buffer written value is stored in T00REG. Subsequently, the match detection is executed using a new stored value.

  When a write instruction to T00REG is executed while the timer counter is stopped, the set value is immediately stored in both the double buffer and T00REG.

- When the double buffer is disabled

  When a write instruction to T00REG is executed while the timer counter is operated, the written value is stored in T00REG. Subsequently, the match detection is executed using a new written value.

  When the value written to T00REG is smaller than the up counter value, the match detection is executed using a new written value after the up counter overflows. Therefore, the interrupt request interval may be longer than the selected time. When the value written to T00REG is equal to the up counter value, the match detection is executed immediately after data is stored T00REG. Therefore, the interrupt request interval may not be an integral multiple of the source clock (Figure 14-3). When these are problems, enable the double buffer. When a write instruction to T00REG is executed while the timer counter is stopped, the written value is immediately stored in T00REG.

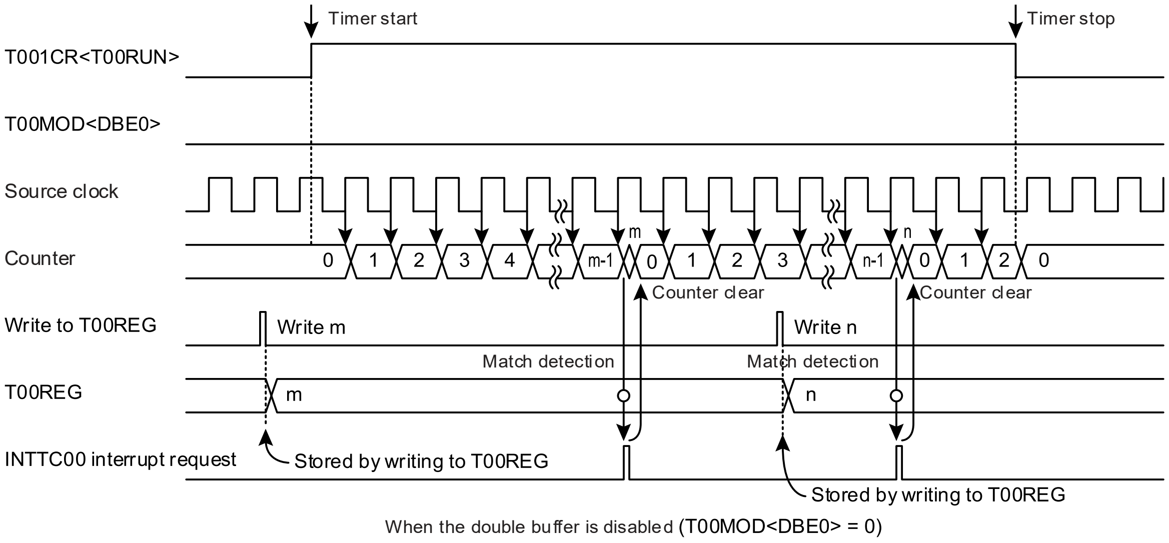
When a read instruction is executed on T00REG, the last value written into T00REG is read out, regardless of the T00MOD<DBE0> setting.

Table 14-6 8-bit Timer Mode Resolution and Maximum Time Setting

T00MOD <TCK0>	Source clock [Hz]			Resolution		Maximum time setting	
	NORMAL1/2 or IDLE1/2 mode		SLOW1/2 or SLEEP1 mode	fcgck = 10 [MHz]	fs = 32.768 [kHz]	fcgck = 10 [MHz]	fs = 32.768 [kHz]
	SYSCR1 <DV9CK> = 0	SYSCR1 <DV9CK> = 1					
000	fcgck / 2^{11}	fs / 2^4	fs / 2^4	204.8 [μs]	488.2 [μs]	52.2 [ms]	124.5 [ms]
001	fcgck / 2^{10}	fs / 2^3	fs / 2^3	102.4 [μs]	244.1 [μs]	26.1 [ms]	62.3 [ms]
010	fcgck / 2^8	fcgck / 2^8	-	25.6 [μs]	-	6.5 [ms]	-
011	fcgck / 2^6	fcgck / 2^6	-	6.4 [μs]	-	1.6 [ms]	-
100	fcgck / 2^4	fcgck / 2^4	-	1.6 [μs]	-	408 [μs]	-
101	fcgck / 2^2	fcgck / 2^2	-	400 [ns]	-	102 [μs]	-
110	fcgck / 2	fcgck / 2	-	200 [ns]	-	51 [μs]	-
111	fcgck	fcgck	fs / 2^2	100 [ns]	122.1 [μs]	25.5 [μs]	31.1 [ms]

Example: Operate TC00 in the 8-bit timer mode with the operation clock of fcgck / 2^2 [Hz] and generate interrupts at 64 [μs] intervals (fcgck = 10 [MHz])

```
LD      (POFFCR0), 0x10        ; Sets <TC001EN> to "1"
DI                             ; Sets the interrupt master enable flag to "disable"
SET     (EIRH).4               ; Sets the INTTC00 interrupt enable register to "1"
EI                             ; Sets the interrupt master enable flag to "enable"
LD      (T00MOD), 0xE8         ; Selects the 8-bit timer mode and fcgck / 2^2
LD      (T00REG), 0xA0         ; Sets the timer register (64 [μs] / (2^2 / fcgck) = 0xA0)
SET     (T001CR).0             ; Starts TC00
```



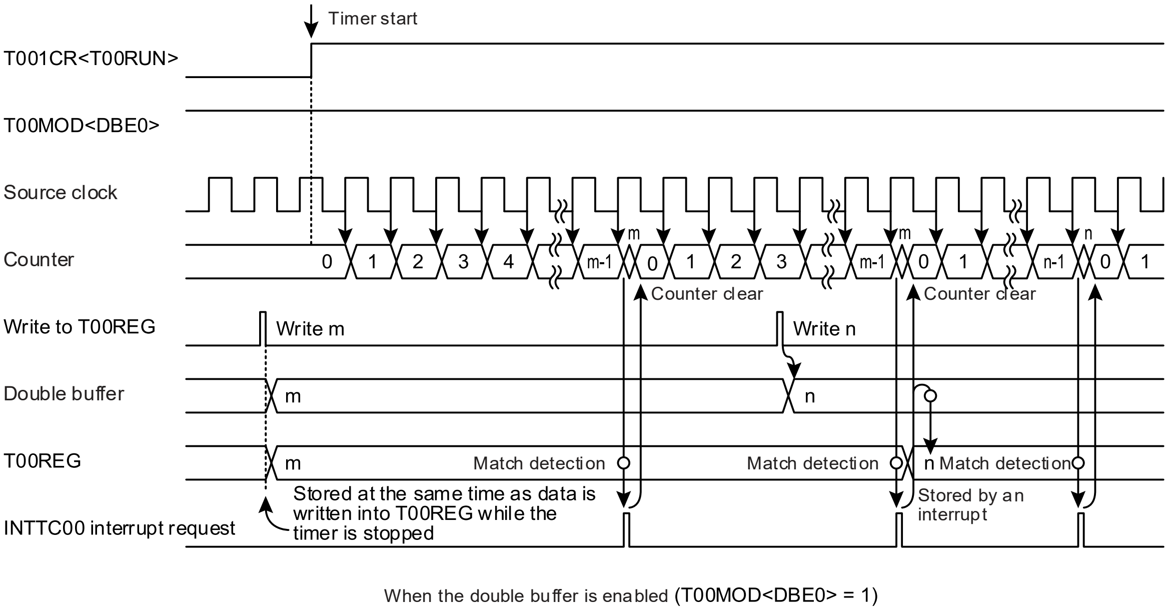


Figure 14-2 Timer Mode Timing Chart

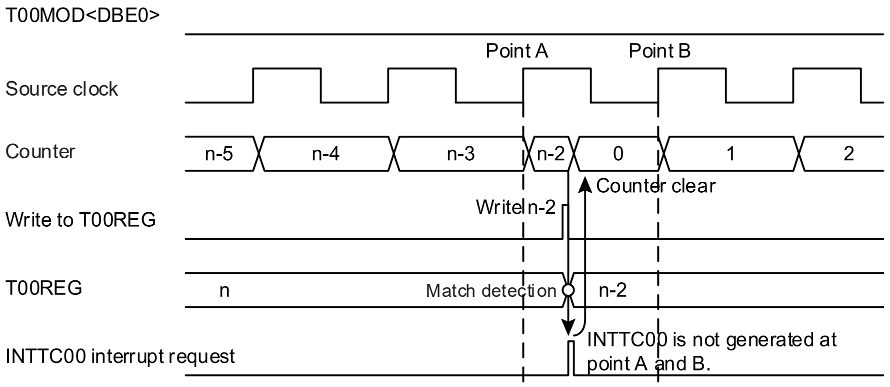


Figure 14-3 Operation When T00REG and the Up Counter Have the Same Value

14.5.2. 8-bit event counter mode

In the 8-bit event counter mode, the up counter counts up at the falling edge of the input to the TC00 pin.

14.5.2.1. Setting

TC00 is put into the 8-bit event counter mode by setting T00MOD<TCM0> to "00", T001CR<TCAS> to "0" and T00MOD<EIN0> to "1". Set the count value to be used for the match detection as an 8-bit value at the timer register T00REG.

Set T00MOD<DBE0> to "1" to use the double buffer.

Setting T001CR<T00RUN> to "1" starts the operation. After the timer counter is started, writing to T00MOD becomes invalid. Be sure to complete the required mode settings before starting the timer counter.

14.5.2.2. Operation

Setting T001CR<T00RUN> to "1" enables the 8-bit up counter to count up at the falling edge of the TC00 pin. When a match between the up-counter value and the T00REG set value is detected, the INTTC00 interrupt request is generated and the up counter is cleared to "0x00". After being cleared, the up counter restarts counting. Clearing T001CR<T00RUN> to "0" while the timer counter is operated makes the up counter stop counting and be cleared to "0x00".

The maximum frequency to be supplied is fcgck / 2^2 [Hz] (in NORMAL1/2 or IDLE1/2 mode) or fs / 2^4 [Hz] (in SLOW1/2 or SLEEP1 mode), and a pulse width of two machine cycles or more is required at both the “High” and “Low” levels.

14.5.2.3. Double buffer

Refer to "14.5.1.3. Double buffer".

Example: Operate TC00 in the 8-bit event counter mode and generate an interrupt each time 16 falling edges are detected at the TC00 pin.

```
LD      (POFFCR0), 0x10     ; Sets <TC001EN> to "1"
DI                          ; Sets the interrupt master enable flag to disable
SET     (EIRH).4            ; Sets the INTTC00 interrupt enable register to "1"
EI                          ; Sets the interrupt master enable flag to "enable"
LD      (T00MOD), 0xC4      ; Selects to the 8-bit event counter mode
                            ; Selects an external clock as the source clock.
LD      (T00REG), 0x10      ; Sets the timer register
SET     (T001CR).0          ; Starts TC00
```

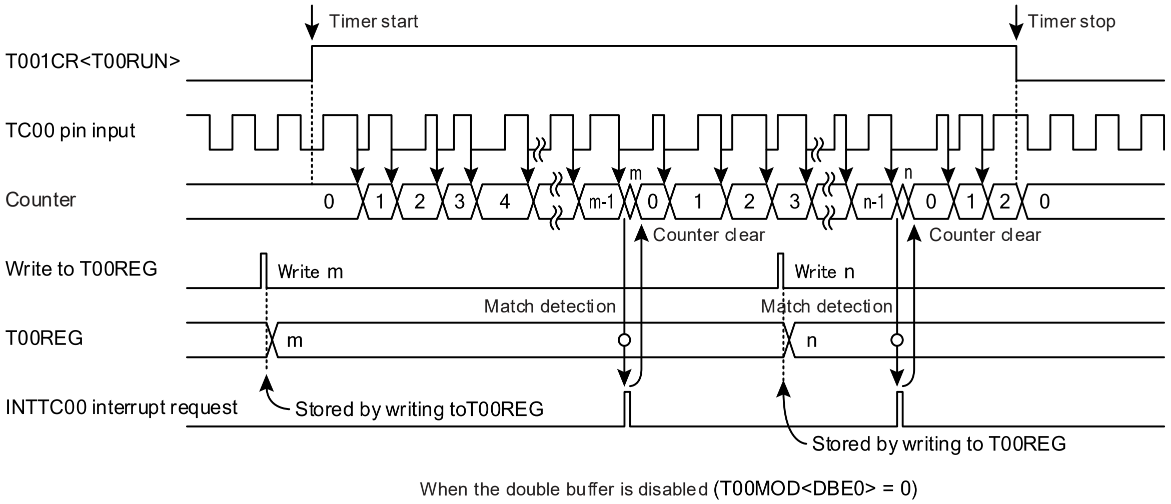


Figure 14-4 Event Counter Mode Timing Chart

14.5.3. 8-bit pulse width modulation (PWM) output mode

The pulse-width modulated pulses with a resolution of 7 bits are output in the 8-bit pulse width modulation (PWM) mode. An additional pulse can be added to the 2 × n-th (n = 1, 2, 3...) PWM pulse. This enables PWM output with a resolution nearly equivalent to 8 bits.

14.5.3.1. Setting

TC00 is put into the 8-bit PWM mode by setting T00MOD<TCM0> to "10" and T001CR<TCAS> to "0". Set T00MOD<EIN0> to "0" and select the clock at T00MOD<TCK0>. Set the count value to be used for the match detection and the additional pulse value at the PWM register T00PWM.

Set T00MOD<DBE0> to "1" to use the double buffer.

Setting T001CR<T00RUN> to "1" starts the operation. After the timer counter is started, writing to T00MOD becomes invalid. Be sure to complete the required mode settings before starting the timer counter.

In the 8-bit PWM mode, the T00PWM register is configured as follows:

Timer register 00

T00PWM		7	6	5	4	3	2	1	0
(0x0028)	Bit Symbol	PWMDUTY							PWMAD
	Read/Write	R/W							R/W
	After reset	1	1	1	1	1	1	1	1

Timer register 01

T01PWM		7	6	5	4	3	2	1	0
(0x0029)	Bit Symbol	PWMDUTY							PWMAD
	Read/Write	R/W							R/W
	After reset	1	1	1	1	1	1	1	1

<PWMDUTY> is a 7-bit register used to set the duty pulse width (the time before the first output change) in a cycle (128 counts of the source clock).

<PWMAD> is a register used to set the additional pulse. When <PWMAD> is "1", an additional pulse that corresponds to 1 count of the source clock is added to the (2 × n)-th (n = 1, 2, 3...) duty pulse. In other words, the (2 × n)-th duty pulse has the output of <PWMDUTY> + 1.

The additional pulse is not added when <PWMAD> is "0".

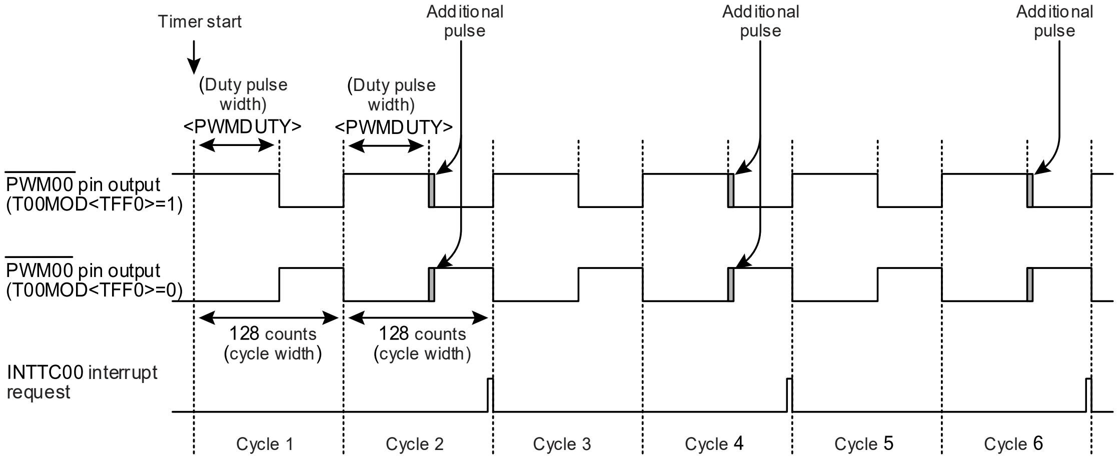

Timer start
Additional pulse
Additional pulse
Additional pulse
(Duty pulse width) <PWMDUTY>
(Duty pulse width) <PWMDUTY>
$\overline{\text{PWM00}}$ pin output (T00MOD<TFF0>=1)
$\overline{\text{PWM00}}$ pin output (T00MOD<TFF0>=0)
128 counts (cycle width)
128 counts (cycle width)
INTTC00 interrupt request
Cycle 1
Cycle 2
Cycle 3
Cycle 4
Cycle 5
Cycle 6

Figure 14-5 $\overline{\text{PWM00}}$ Pulse Output (<PWMAD> = 1)

Set the initial state of the $\overline{\text{PWM00}}$ pin at T00MOD<TFF0>. Setting T00MOD<TFF0> to "0" selects the “Low” level as the initial state of the $\overline{\text{PWM00}}$ pin. Setting T00MOD<TFF0> to "1" selects the “High” level as the initial state of the $\overline{\text{PWM00}}$ pin. When the $\overline{\text{PWM00}}$ pin is set as the function output pin in the port setting while the timer is stopped, the value of T00MOD<TFF0> is output to the $\overline{\text{PWM00}}$ pin. Table 14-7 shows the list of output levels of the $\overline{\text{PWM00}}$ pin.

Table 14-7 List of Output Levels of $\overline{\text{PWM00}}$ pin

<TFF0>	$\overline{\text{PWM00}}$ pin output level			
	Before the start of operation (initial state)	T00PWM <PWMDUTY> matched (after the additional pulse)	Overflow	Operation stopped (initial state)
0	“Low” level	“High” level	“Low” level	“Low” level
1	“High” level	“Low” level	“High” level	“High” level

And by setting "1" to T001CR<OUTAND>, a logical product (AND) pulse of TC00 and TC01 can be output to $\overline{\text{PWM00}}$ pin. By using this function, the remote-control waveform can be created easily.

14.5.3.2. Operations

Setting T001CR<T00RUN> to "1" enables the up counter to count up based on the selected source clock. When a match between the lower 7 bits of the up counter value and the value set to T00PWM<PWMDUTY> is detected, the output of the $\overline{\text{PWM00}}$ pin is reversed. When T00MOD<TFF0> is "0", the $\overline{\text{PWM00}}$ pin changes from the "Low" to "High" level. When T00MOD<TFF0> is "1", the $\overline{\text{PWM00}}$ pin changes from the "High" to "Low" level.

When T00PWM<PWMAD> is "1", an additional pulse that corresponds to 1 count of the source clock is added at the (2 × n)-th (n = 1, 2, 3...) match detection. In other words, the $\overline{\text{PWM00}}$ pin output is reversed at the timing of T00PWM<PWMDUTY>+1. When T00MOD<TFF0> is "0", the period of the "Low" level becomes longer than the value set to T00PWM<PWMDUTY> by 1 source clock. When T00MOD<TFF0> is "1", the period of the "High" level becomes longer than the value set to T00PWM<PWMDUTY> by 1 source clock. Two cycles of the 7-bit resolution PWM output can be used as 8-bit resolution by this function.

No additional pulse is inserted when T00PWM<PWMAD> is "0".

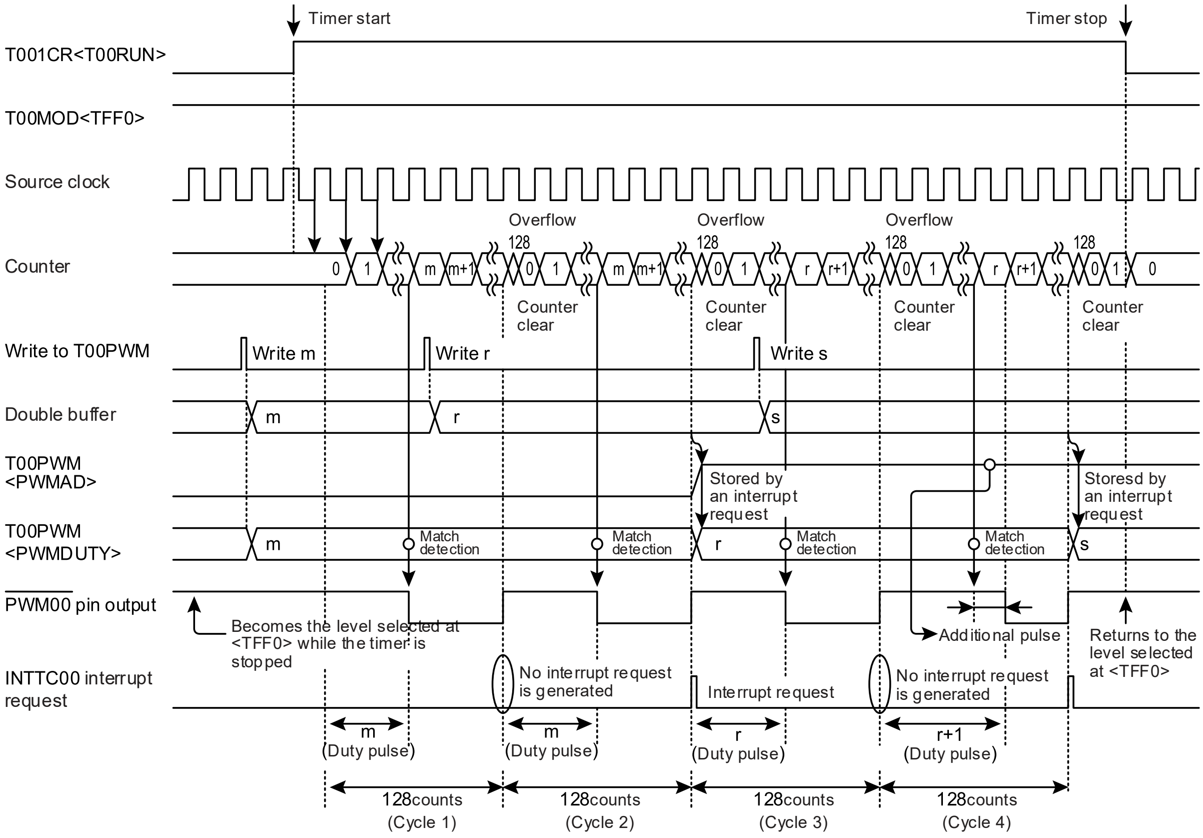
Subsequently, the up counter continues counting up. When the up counter value reaches 128, an overflow occurs and the up counter is cleared to "0x00". At the same time, the output of the PWM0 pin is reversed. When T00MOD<TFF0> is "0", the $\overline{\text{PWM00}}$ pin changes from the "High" to "Low" level. When T00MOD<TFF0> is "1", the $\overline{\text{PWM00}}$ pin changes from the "Low" to "High" level. When the (2 × n)-th overflow occurs at this time, the INTTC00 interrupt request is generated. (No interrupt request is generated at the (2 × n -1) -th overflow.)

Subsequently, the up counter continues counting up.

When T001CR<T00RUN> is set to "0" while the timer counter is operated, the up counter is stopped and cleared to "0x00". The $\overline{\text{PWM00}}$ pin returns to the level selected at T00MOD<TFF0>.

Example: Operate TC00 in the 8-bit PWM mode with the operation clock of fcgck / 2 and output a duty pulse nearly equivalent to 11.6 [μs] in 25.6 [μs] cycles (fcgck = 10 [MHz])
(Actually, output a total duty pulse of 23.2 [μs] in 2 cycles (51.2 [μs]))

```
SET     (P7FC).0            ; Sets <P7FC0> to "1"
SET     (P7CR).0            ; Sets <P7CR0> to "1"
LD      (POFFCR0), 0x10     ; Sets <TC001EN> to "1"
DI                          ; Sets the interrupt master enable flag to "disable"
SET     (EIRH).4            ; Sets the INTTC00 interrupt enable register to "1"
EI                          ; Sets the interrupt master enable flag to "enable"
LD      (T00MOD), 0xF2      ; Selects the 8-bit PWM mode and fcgck / 2
LD      (T00PWM), 0x74      ; Sets the timer register (duty pulse)
                            ; (11.6 [μs] × 2) / (2 / fcgck) = 0x74
SET     (T001CR).0          ; Starts TC00
```



When the double buffer is enabled (T00MOD<DBE0> = 1)

Figure 14-6 8-bit PWM Mode Timing Chart

14.5.3.3. Double buffer

In the TMP89FS60B/62B/63B, the double buffer for T00PWM can be used by setting T00MOD<DBE0>. The double buffer is disabled by setting T00MOD<DBE0> to "0" or enabled by setting T00MOD<DBE0> to "1".

– When the double buffer is enabled

When a write instruction to T00PWM is executed while the timer counter is operated, the written value is stored in the double buffer, and T00PWM is not updated. T00PWM compares the previous written value with the up counter value. When the (2 × n)-th overflow occurs, the INTTC00 interrupt request is generated and the double buffer written value is stored in T00PWM. Subsequently, the match detection is executed using a new stored value.

When a read instruction from T00PWM is executed, the value in the double buffer (the last set value) is read, not the T00PWM value (the currently effective value).

When a write instruction to T00PWM is executed while the timer counter is stopped, the set value is immediately stored in both the double buffer and T00PWM.

– When the double buffer is disabled

When a write instruction to T00PWM is executed while the timer counter is operated, the set value is immediately stored in T00PWM. Subsequently, the match detection is executed using a new stored value. When the value stored to T00PWM is smaller than the up counter value, the $\overline{\text{PWM00}}$ pin is not reversed until the up counter overflows and a match detection is executed using a new stored value. When the value stored to T00PWM is equal to the up counter value, the match detection is executed immediately after data is written into T00PWM. Therefore, the timing of changing the $\overline{\text{PWM00}}$ pin may not be an integral multiple of the source clock (Figure 14-7). Similarly, when T00PWM is set during the additional pulse output, the timing of changing the $\overline{\text{PWM00}}$ pin may not be an integral multiple of the source clock. When these are problems, enable the double buffer.

When a write instruction to T00PWM is executed while the timer counter is stopped, the set value is immediately stored in T00PWM.

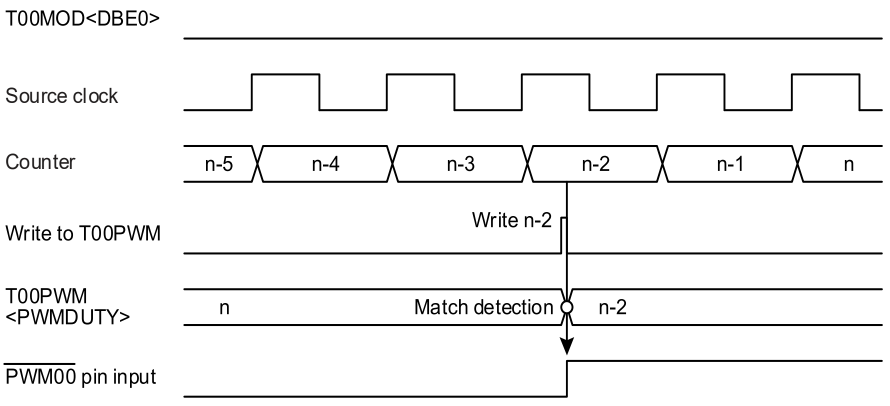


Figure 14-7 Example When the Value as same as an Up Counter's one is written to T00PWM

Table 14-8 Resolutions and Cycles in the 8-bit PWM Mode

T00MOD <TCK0>	Source clock [Hz]			Resolution		7-bit cycle (cycle × 2)	
	NORMAL1/2 or IDLE1/2 mode		SLOW1/2 or SLEEP1 mode	fcgck = 10 [MHz]	fs = 32.768 [kHz]	fcgck = 10 [MHz]	fs = 32.768 [kHz]
	SYSCR1 <DV9CK> = 0	SYSCR1 <DV9CK> = 1					
000	fcgck / 2^{11}	fs / 2^{4}	fs / 2^{4}	204.8 [μs]	488.2 [μs]	26.2 [ms] (52.4 [ms])	62.5 [ms] (125 [ms])
001	fcgck / 2^{10}	fs / 2^{3}	fs / 2^{3}	102.4 [μs]	244.1 [μs]	13.1 [ms] (26.2 [ms])	31.3 [ms] (62.5 [ms])
010	fcgck / 2^{8}	fcgck / 2^{8}	-	25.6 [μs]	-	3.3 [ms] (6.6 [ms])	-
011	fcgck / 2^{6}	fcgck / 2^{6}	-	6.4 [μs]	-	819.2 [μs] (1638.4 [μs])	-
100	fcgck / 2^{4}	fcgck / 2^{4}	-	1.6 [μs]	-	204.8 [μs] (409.6 [μs])	-
101	fcgck / 2^{2}	fcgck / 2^{2}	-	400 [ns]	-	51.2 [μs] (102.4 [μs])	-
110	fcgck / 2	fcgck / 2	-	200 [ns]	-	25.6 [μs] (51.2 [μs])	-
111	fcgck	fcgck	fs / 2^{2}	100 [ns]	122.1 [μs]	12.8 [μs] (25.6 [μs])	15.6 [ms] (31.3 [ms])

14.5.4. 8-bit programmable pulse generate (PPG) output mode

In the 8-bit PPG mode, the pulses with arbitrary duty and cycle are output by using the T00REG and T00PWM registers.

By setting T001CR<OUTAND>, output a logical ANDed pulse of the outputs of TC00 and TC01 from $\overline{\text{PPG01}}$ pin. This function facilitates the generation of remote-controlled waveforms, for example.

14.5.4.1. Setting

TC00 is put into the 8-bit PPG mode by setting T00MOD<TCM0> to "10" and T001CR<TCAS> to "0". Set T00MOD<EIN0> to "0" and select the clock at T00MOD<TCK0>. Set the duty pulse width at T00PWM and the cycle width at T00REG.

Set T00MOD<DBE0> to "1" to use the double buffer.

Setting T001CR<T00RUN> to "1" starts the operation. After the timer counter is started, writing to T00MOD becomes invalid. Be sure to complete the required mode settings before starting the timer counter.

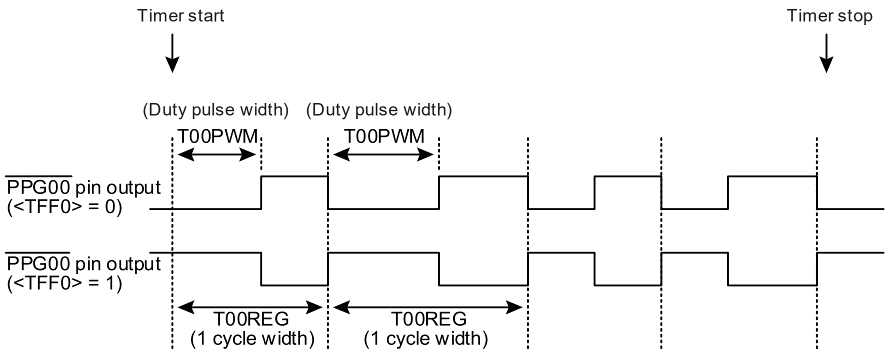


Figure 14-8 $\overline{\text{PPG00}}$ Pulse Output

Set the initial state of the $\overline{\text{PPG00}}$ pin at T00MOD<TFF0>. Clearing T00MOD<TFF0> to "0" selects the “Low” level as the initial state of the $\overline{\text{PPG00}}$ pin. Setting T00MOD<TFF0> to "1" selects the “High” level as the initial state of the $\overline{\text{PPG00}}$ pin. When the $\overline{\text{PPG00}}$ pin is set as the function output pin in the port setting while the timer counter is stopped, the value of T00MOD<TFF0> is output to the $\overline{\text{PPG00}}$ pin. Table 14-9 shows the list of output levels of the $\overline{\text{PPG00}}$ pin.

Table 14-9 List of Output Levels of $\overline{\text{PPG00}}$ pin

<TFF0>	$\overline{\text{PPG00}}$ pin output level			
	Before the start of operation (initial state)	T00PWM matched	T00REG matched	Operation stopped (initial state)
0	“Low” level	“High” level	“Low” level	“Low” level
1	“High” level	“Low” level	“High” level	“High” level

Setting the T001CR<OUTAND> to "1" enables the $\overline{\text{PPG01}}$ pin to output a pulse that is a logical ANDed.

14.5.4.2. Operation

Setting T001CR<T00RUN> to "1" enables the up counter to count up based on the selected source clock. When a match between the internal up counter value and the value set to T00PWM is detected, the output of the $\overline{\text{PPG00}}$ pin is reversed. When T00MOD<TFF0> is "0", the $\overline{\text{PPG00}}$ pin changes from the “Low” to “High” level. When T00MOD<TFF0> is "1", the $\overline{\text{PPG00}}$ pin changes from the “High” to “Low” level.

Subsequently, the up counter continues counting up. When a match between the up counter value and T00REG is detected, the output of the $\overline{\text{PPG00}}$ pin is reversed again. When T00MOD<TFF0> is "0", the $\overline{\text{PPG00}}$ pin changes from the “High” to “Low” level. When T00MOD<TFF0> is "1", the $\overline{\text{PPG00}}$ pin changes from the “Low” to “High” level. At this time, the INTTC00 interrupt request is generated.

When T001CR<T00RUN> is set to "0" during the operation, the up counter is stopped and cleared to "0x00". The $\overline{\text{PPG00}}$ pin returns to the level selected at T00MOD<TFF0>.

14.5.4.3. Double buffer

In the TMP89FS60B/62B/63B, the double buffer for T00PWM and T00REG can be used by setting T00MOD<DBE0>. The double buffer is disabled by setting T00MOD<DBE0> to "0" or enabled by setting T00MOD<DBE0> to "1".

- When the double buffer is enabled

  When a write instruction to T00PWM (T00REG) is executed while the timer counter is operated, the written value is stored in the double buffer, and T00PWM (T00REG) is not updated. T00PWM (T00REG) compares the previous written value with the up counter value.

  When an INTTC00 interrupt request is generated, the double buffer set value is stored in T00PWM (T00REG). Subsequently, the match detection is executed using a new stored value.

  When a read instruction is executed from T00PWM (T00REG), the value in the double buffer (the last set value) is read, not the T00PWM (T00REG) value (the currently effective value).

  When a write instruction to T00PWM (T00REG) is executed while the timer counter is stopped, the set value is immediately stored in both the double buffer and T00PWM (T00REG).

– When the double buffer is disabled

When a write instruction T00PWM (T00REG) is executed while the timer counter is operated, the set value is stored in T00PWM (T00REG). Subsequently, the match detection is executed using a new set value. When the value set to T00PWM (T00REG) is smaller than the up counter value, the $\overline{\text{PPG00}}$ pin is not reversed until the up counter overflows and a match detection is executed using a new set value. When the value set to T00PWM (T00REG) is equal to the up counter value, the match detection is executed immediately after data is written into T00PWM (T00REG). Therefore, the timing of changing the $\overline{\text{PPG00}}$ pin may not be an integral multiple of the source clock (Figure 14-9). When these are problems, enable the double buffer.

When a write instruction is executed on T00PWM (T00REG) while the timer counter is stopped, the set value is immediately stored in T00PWM (T00REG).

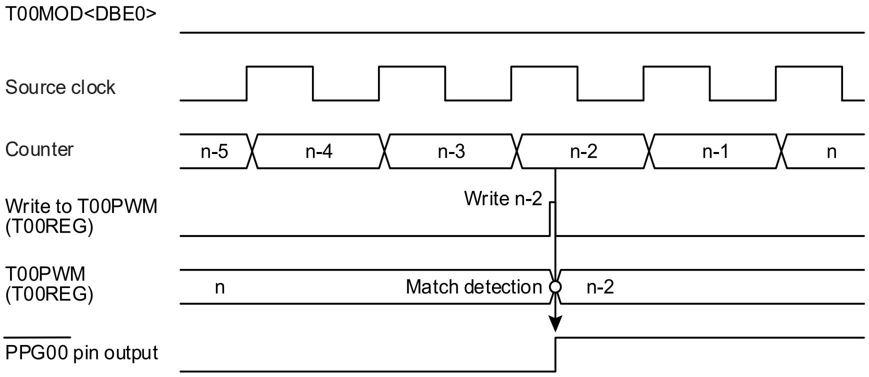
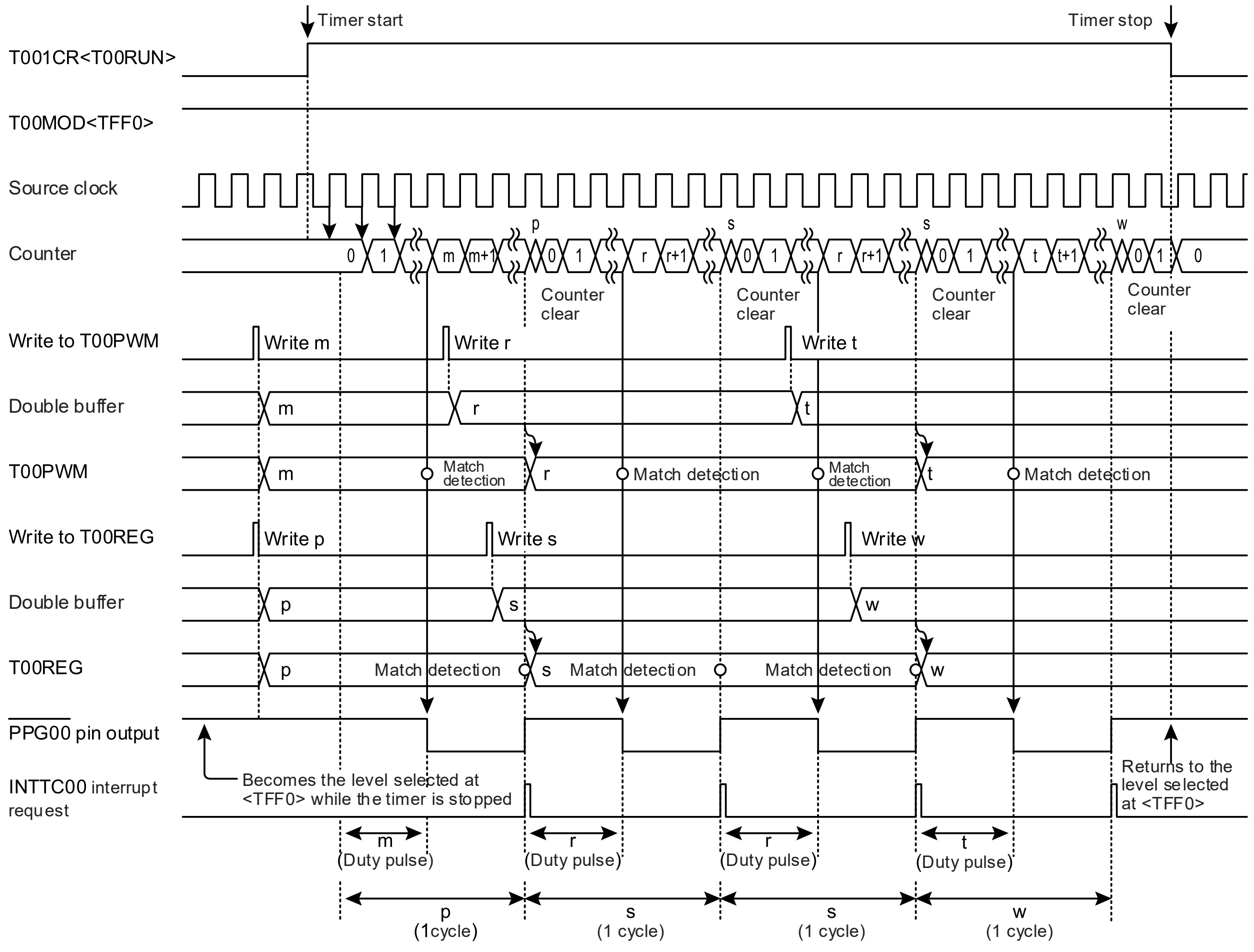


Figure 14-9 Example When the Value as same as an Up Counter's one is written to T00PWM (T00REG)

Example: Operate TC00 in the 8-bit PPG mode with the operation clock of fcgck / 2 and output the 8 [µs] duty pulse in 32 [µs] cycles (fcgck = 10 [MHz])

```
SET     (P7FC).0              ; Set <P7FC0> to "1"
SET     (P7CR).0              ; Set <P7CR0> to "1"
LD      (POFFCR0), 0x10       ; Set <TC001EN> to "1"
DI                            ; Set the interrupt master enable flag to "disable"
SET     (EIRH).4              ; Set the INTTC00 interrupt enable register to "1"
EI                            ; Set the interrupt master enable flag to "enable"
LD      (T00MOD), 0xF3        ; Select the 8-bit PPG mode and fcgck / 2
LD      (T00REG), 0xA0        ; Set the timer register (cycle)
                              ; 32 [µs] / (2 / fcgck) = 0xA0
LD      (T00PWM), 0x28        ; Set the timer register (duty pulse)
                              ; 8 [µs] / (2 / fcgck) = 0x28
SET     (T001CR).0            ; Start TC00
```



When the double buffer is enabled (T00MOD<DBE0> = 1)

Figure 14-10 8-bit PPG Mode Timing Chart

14.5.5. 16-bit timer mode

In the 16-bit timer mode, TC00 and TC01 are cascaded to form a 16-bit timer counter, which can measure a longer period than an 8-bit timer.

14.5.5.1. Setting

Setting T001CR<TCAS> to "1" connects TC00 and TC01 and activates the 16-bit mode. All the settings of TC00 are ignored and those of TC01 are effective in the 16-bit mode.

The 16-bit timer mode is activated by setting T01MOD<TCM1> to "00" or "01" and T01MOD<EIN1> to "0". Select the source clock at T01MOD<TCK1>.

Set the count value to be used for the match detection as a 16-bit value at the timer registers T00REG and T01REG. Set the lower 8 bits of the 16-bit value at T00REG and the higher 8 bits at T01REG. (Hereinafter, the 16-bit value specified by the combined setting of T01REG and T00REG is indicated as T01+00REG.) The timer register settings are stored on the double buffer or T01+00REG when a write instruction to T01REG is executed. Be sure to execute the write instructions on T00REG and T01REG in this order. (When data is written to the high-order register, the set values of the low-order and high-order registers become effective at the same time.)

Set T01MOD<DBE1> to "1" to use the double buffer.

Setting T001CR<T01RUN> to "1" starts the operation. After the timer counter is started, writing to T01MOD becomes invalid. Be sure to complete the required mode settings before starting the timer counter. (Make settings when T001CR<T00RUN> and <T01RUN> are "0".)

14.5.5.2. Operations

Setting T001CR<T01RUN> to "1" enables the 16-bit up counter to count up based on the selected internal source clock. When a match between the up counter value and the T01+00REG set value is detected, the INTTC01 interrupt request is generated and the up counter is cleared to "0x0000". After being cleared, the up counter restarts counting. Clearing T001CR<T01RUN> to "0" during the timer operation makes the up counter stop counting and be cleared to "0x0000".

14.5.5.3. Double buffer

In the TMP89FS60B/62B/63B, the double buffer for T01+00REG can be used by setting T01MOD<DBE1>. The double buffer is disabled by setting T01MOD<DBE1> to "0" or enabled by setting T01MOD<DBE1> to "1".

- When the double buffer is enabled

  When write instructions to T00REG and T01REG are executed in this order while the timer counter is operated, the written value is stored in the double buffer, and T01+00REG is not updated in a minute. T01+00REG compares the previous written value with the up counter value. When the values are matched, the INTTC01 interrupt request is generated and the double buffer set value is stored in T01+00REG. Subsequently, the match detection is executed using a new stored value.

  When write instructions to T00REG and T01REG are executed in this order while the timer counter is stopped, the set value is immediately stored in both the double buffer and T01+00REG.

- When the double buffer is disabled

  When write instructions to T00REG and T01REG are executed in this order while the timer counter is operated, the set value is stored in T01+00REG. Subsequently, the match detection is executed using a new set value.

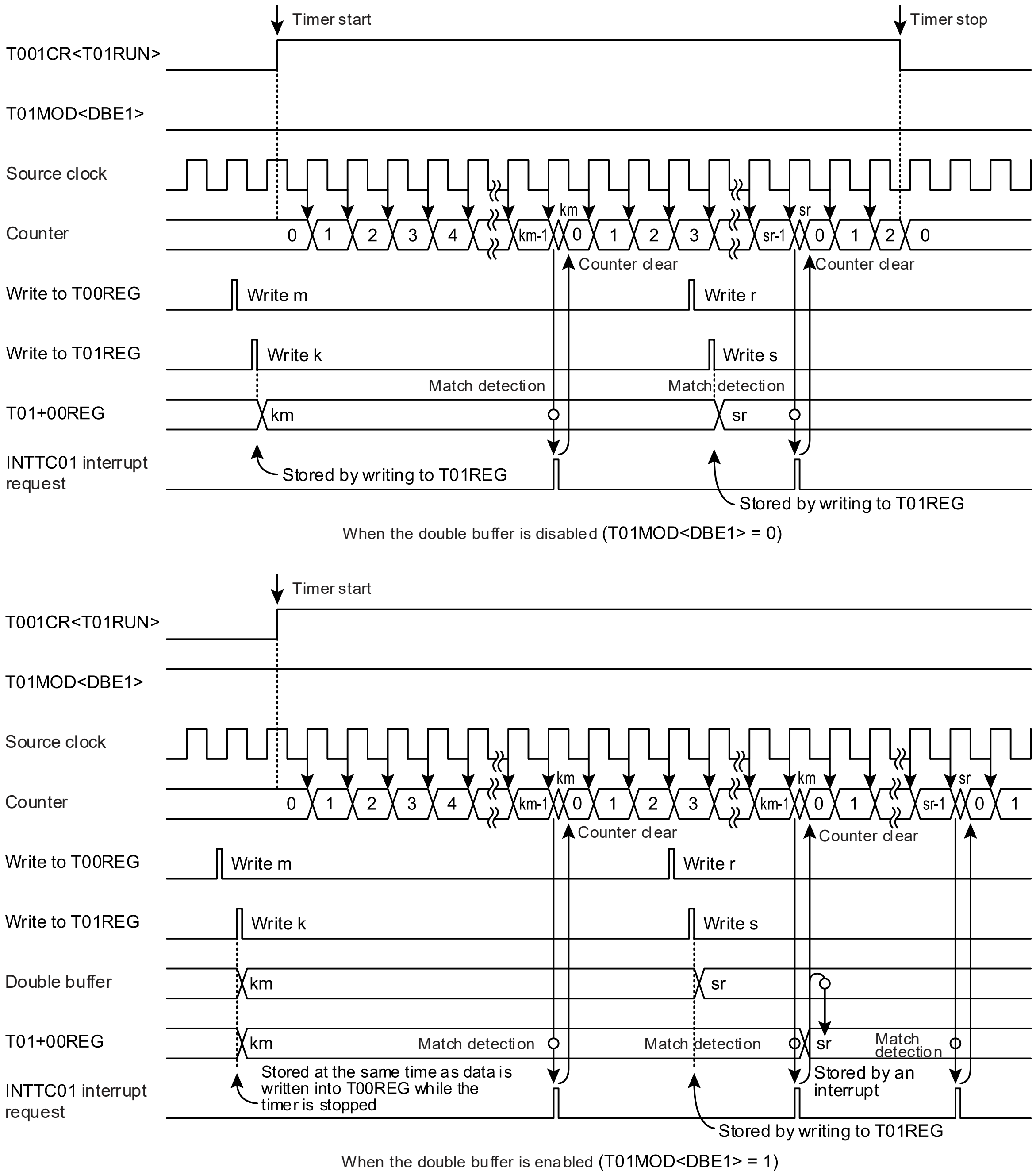
  When the value written to T01+00REG is smaller than the up counter value, the match detection is executed using a new written value after the up counter overflows. Therefore, the interrupt request interval may be longer than the selected time. When the value written to T01+00REG is equal to the up counter value, the match detection is executed immediately after data is written into T01+00REG. Therefore, the interrupt request interval may not be an integral multiple of the source clock. When these are problems, enable the double buffer.

  When write instructions are executed on T00REG and T01REG in this order while the timer is stopped, the set value is immediately stored in T01+00REG.

When a read instruction is executed on T01+00REG, the last value written into T01+00REG is read, regardless of the T00MOD<DBE1> setting.

Example: Operate TC00 and TC01 in the 16-bit timer mode with the operation clock of fcgck / 2 [Hz] and generate interrupts at 96 [μs] intervals (fcgck = 10 [MHz])

```
LD      (POFFCR0), 0x10     ; Sets <TC001EN> to "1"
DI                          ; Sets the interrupt master enable flag to "disable"
SET     (EIRH).4            ; Sets the INTTC00 interrupt enable register to "1"
EI                          ; Sets the interrupt master enable flag to "enable"
LD      (T01MOD), 0xF0      ; Selects the 16-bit timer mode and fcgck / 2
LD      (T00REG), 0xE0      ; Sets the timer register (96 [μs] / (2 / fcgck) = 0x1E0)
LD      (T01REG), 0x01      ; Sets the timer register
LD      (T001CR), 0x06      ; Starts TC00 and TC001 (16-bit mode)
```



When the double buffer is disabled (T01MOD<DBE1> = 0)

When the double buffer is enabled (T01MOD<DBE1> = 1)

Figure 14-11 16-bit Timer Counter Timing Chart

Table 14-10 16-bit Timer Mode Resolution and Maximum Time Setting

T01MOD <TCK1>	Source clock [Hz]			Resolution		Maximum time setting	
	NORMAL1/2 or IDLE1/2 mode		SLOW1/2 or SLEEP1 mode	fcgck = 10 [MHz]	fs = 32.768 [kHz]	fcgck = 10 [MHz]	fs = 32.768 [kHz]
	SYSCR1 <DV9CK> = 0	SYSCR1 <DV9CK> = 1					
000	fcgck / 2^{11}	fs / 2^{4}	fs / 2^{4}	204.8 [μs]	488.2 [μs]	13.4 [s]	32 [s]
001	fcgck / 2^{10}	fs / 2^{3}	fs / 2^{3}	102.4 [μs]	244.1 [μs]	6.7 [s]	16 [s]
010	fcgck / 2^{8}	fcgck / 2^{8}	-	25.6 [μs]	-	1.7 [s]	-
011	fcgck / 2^{6}	fcgck / 2^{6}	-	6.4 [μs]	-	419.4 [ms]	-
100	fcgck / 2^{4}	fcgck / 2^{4}	-	1.6 [μs]	-	104.9 [ms]	-
101	fcgck / 2^{2}	fcgck / 2^{2}	-	400 [ns]	-	26.2 [ms]	-
110	fcgck / 2	fcgck / 2	-	200 [ns]	-	13.1 [ms]	-
111	fcgck	fcgck	fs / 2^{2}	100 [ns]	122.1 [μs]	6.6 [ms]	8 [s]

14.5.6. 16-bit event counter mode

In the 16-bit event counter mode, the up counter counts up at the falling edge of the input to the TC00 pin. TC00 and TC01 are cascaded to form a 16-bit timer counter, which can measure a longer period than an 8-bit timer.

14.5.6.1. Setting

Setting T001CR<TCAS> to "1" connects TC00 and TC01 and activates the 16-bit timer mode. All the settings of TC00 are ignored and those of TC01 are effective in the 16-bit timer mode.

The 16-bit timer mode is activated by setting T01MOD<TCM1> to "00" or "01" and T01MOD<EIN0> to "1".

Set the count value to be used for the match detection as a 16-bit value at the timer registers T00REG and T01REG. Set the lower 8 bits of the 16-bit value at T00REG and set the higher 8 bits at T01REG. (Hereinafter, the 16-bit value specified by the combined setting of T01REG and T00REG is indicated as T01+00REG.) The timer register settings are stored on the double buffer or T01+00REG when a write instruction to T01REG is executed. Be sure to execute the write instructions to T00REG and T01REG in this order. (When data is written to the high-order register, the written values of the low-order and high-order registers become effective at the same time.)

Set T01MOD<DBE1> to "1" to use the double buffer.

Setting T001CR<T01RUN> to "1" starts the operation. After the timer counter is started, writing to T01MOD becomes invalid. Be sure to complete the required mode settings before starting the timer counter . (Make settings when T001CR<T00RUN> and <T01RUN> are "0".)

14.5.6.2. Operations

Setting T001CR<T01RUN> to "1" enables the 16-bit up counter to count up at the falling edge of the TC00 pin. When a match between the up counter value and the T01+00REG set value is detected, the INTTC01 interrupt request is generated and the up counter is cleared to "0x0000". After being cleared, the up counter restarts counting. Setting T001CR<T01RUN> to "0" during the timer operation makes the up counter stop counting and be cleared to "0x0000".

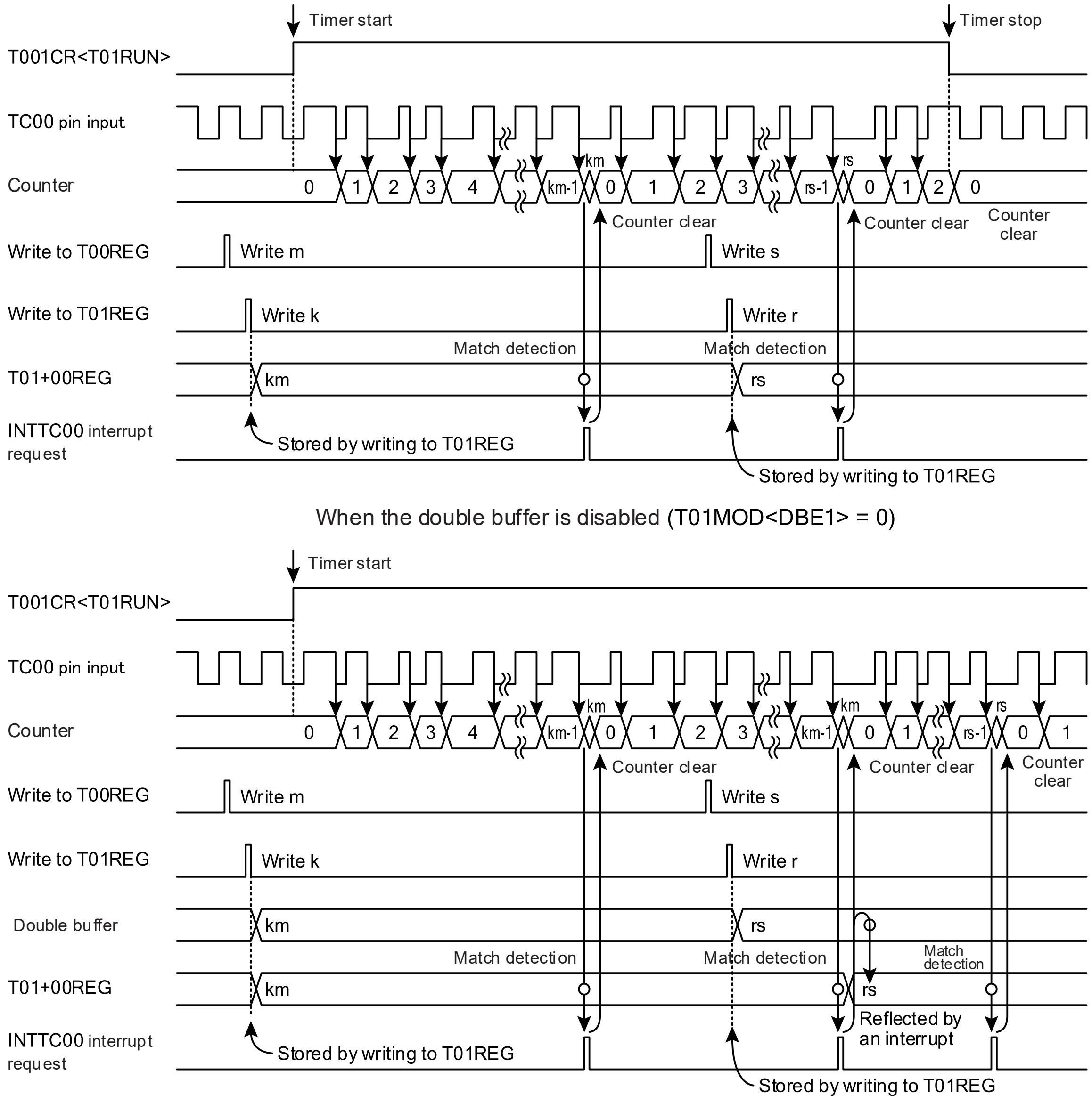
The maximum frequency to be supplied is fcgck / 2 [Hz] (in NORMAL1/2 or IDLE1/2 mode) or fs / 2^4 [Hz] (in SLOW1/2 or SLEEP1 mode), and a pulse width of two machine cycles or more is required at both the “High” and “Low” levels.

14.5.6.3. Double buffer

Refer to "14.5.5.3 Double buffer".

Example: Operate TC00 and TC01 in the 16-bit event counter mode and generate an interrupt each time the 384th falling edge is detected at the TC00 pin

```
LD     (POFFCR0), 0x10    ; Set <TC001EN> to "1"
DI                        ; Set the interrupt master enable flag to "disable"
SET    (EIRH).4           ; Set the INTTC00 interrupt enable register to "1"
EI                        ; Set the interrupt master enable flag to "enable"
LD     (T00MOD), 0xC4     ; Select the 16-bit event counter mode
LD     (T00REG), 0x80     ; Set the timer register
LD     (T01REG), 0x01     ; Set the timer register
LD     (T001CR), 0x06     ; Start TC00 and TC001 (16-bit mode)
```



When the double buffer is disabled (T01MOD<DBE1> = 0)

When the double buffer is enabled (T01MOD<DBE1> = 1)

Figure 14-12 16-bit Event Counter Mode Timing Chart

14.5.7. 12-bit pulse width modulation (PWM) output mode

In the 12-bit PWM output mode, TC00 and TC01 are cascaded to output the pulse-width modulated pulses with a resolution of 8 bits. An additional pulse of 4 bits can be inserted, which enables PWM output with a resolution nearly equivalent to 12 bits.

14.5.7.1. Setting

Setting T001CR<TCAS> to "1" connects TC00 and TC01 and activates the 16-bit timer mode. All the settings of TC00 are ignored and those of TC01 are effective in the 16-bit timer mode.

The 12-bit PWM mode is selected by setting T01MOD<TCM1> to "10". To use the internal clock as the source clock, set T01MOD<EIN1> to "0" and select the clock at T01MOD<TCK1>. To use an external clock as the source clock, set T01MOD<EIN1> to "1".

Set T01MOD<DBE1> to "1" to use the double buffer.

Setting T001CR<T01RUN> to "1" starts the operation. After the timer counter is started, writing to T01MOD becomes invalid. Be sure to complete the required mode settings before starting the timer counter. (Make settings when T001CR<T00RUN> and <T01RUN> are "0".)

Set the count value to be used for the match detection and the additional pulse value as a 12-bit value at the timer registers T00PWM and T01PWM. Set bits 11 to 8 of the 12-bit value at the lower 4 bits of T01PWM and set bits 7 to 0 at T00PWM. Refer to the following table for the register configuration. Hereinafter, the 12-bit value specified by the combined setting of T00PWM and T01PWM is indicated as T01+00PWM. The timer register settings are stored on the double buffer or T01+00PWM when a write instruction to T01PWM is executed. Be sure to execute the write instructions on T00PWM and T01PWM in this order. (When data is written to the high-order register, the written values of the low-order and high-order registers become effective at the same time.)

Timer register 00

T00PWM		7	6	5	4	3	2	1	0
(0x0028)	Bit Symbol	PWMDUTYL				PWMAD3	PWMAD2	PWMAD1	PWMAD0
	Read/Write	R/W				R/W	R/W	R/W	R/W
	After reset	1	1	1	1	1	1	1	1

Timer register 01

T01PWM		7	6	5	4	3	2	1	0
(0x0029)	Bit Symbol	-	-	-	-	PWMDUTYH			
	Read/Write	R/W	R/W	R/W	R/W	R/W			
	After reset	1	1	1	1	1	1	1	1

Bits 7 to 4 of T01PWM are not used in the 12-bit PWM mode. However, data can be written to these bits of T01PWM and the written values are read out as they are when the bits are read. Normally, set these bits to "0".

<PWMDUTYH> and <PWMDUTYL> are 4-bit registers. They are combined to set an 8-bit value of duty pulse width (time before the first change in the output) for one cycle (256 counts of the source clock). Hereinafter, an 8-bit value specified by the combined setting of <PWMDUTYH> and <PWMDUTYL> is indicated as <PWMDUTY>.

<PWMAD3 to 0> are the additional pulse setting register. Additional pulses can be inserted in specific cycles of the duty pulse by setting each bit to "1". The additional pulses are inserted in the positions listed in Table 14-11. <PWMAD3 to 0> can be combined to specify the number of times of inserting the additional pulses in 16 cycles to any number from 1 to 15. Examples of inserting additional pulses are shown in Figure 14-13.

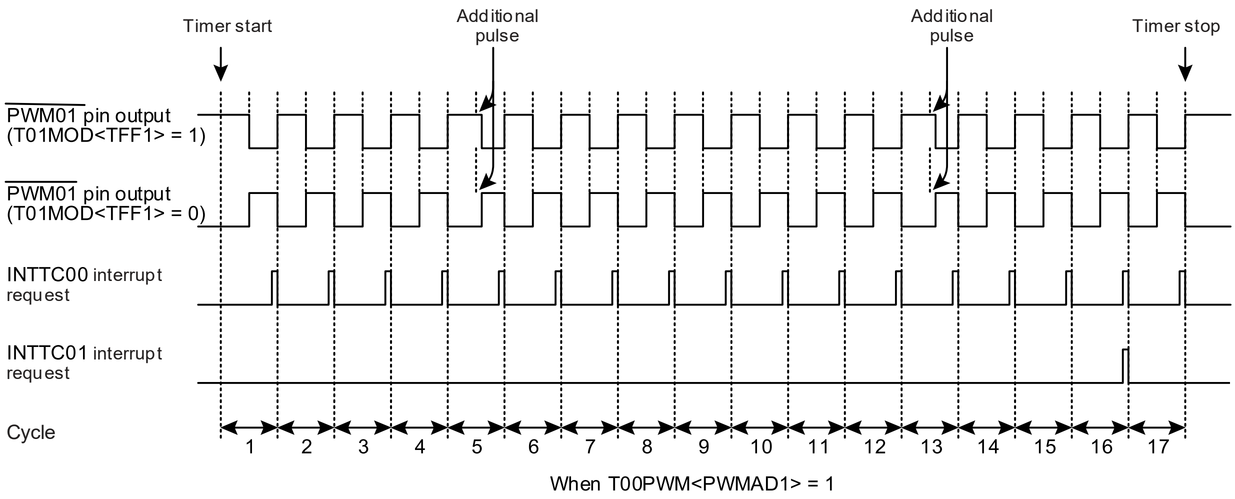
Table 14-11 Cycles in Which Additional Pulses Are Inserted

	Cycles in which additional pulses are inserted among cycles 1 to 16
<PWMAD0> = 1	9
<PWMAD1> = 1	5, 13
<PWMAD2> = 1	3, 7, 11, 15
<PWMAD3> = 1	2, 4, 6, 8, 10, 12, 14, 16

Set the initial state of the $\overline{\text{PWM01}}$ pin at T01MOD<TFF1>. Setting T01MOD<TFF1> to "0" selects the “Low” level as the initial state of the $\overline{\text{PWM01}}$ pin. Setting T01MOD<TFF1> to "1" selects the “High” level as the initial state of the $\overline{\text{PWM01}}$ pin. When the $\overline{\text{PWM01}}$ pin is set as the function output pin in the port setting while the timer is stopped, the value of T01MOD<TFF1> is output to the $\overline{\text{PWM01}}$ pin. Table 14-12 shows the list of output levels of the $\overline{\text{PWM01}}$ pin.

Table 14-12 List of Output Levels of $\overline{\text{PWM01}}$ Pin

<TFF1>	$\overline{\text{PWM01}}$ pin output level			
	Before the start of operation (initial state)	PWMDUTY matched (after the additional pulse)	Overflow	Operation stopped (initial state)
0	“Low” level	“High” level	“Low” level	“Low” level
1	“High” level	“Low” level	“High” level	“High” level


Timer start
Additional pulse
Additional pulse
Timer stop
PWM01 pin output (T01MOD<TFF1> = 1)
PWM01 pin output (T01MOD<TFF1> = 0)
INTTC00 interrupt request
INTTC01 interrupt request
Cycle
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
When T00PWM<PWMAD1> = 1

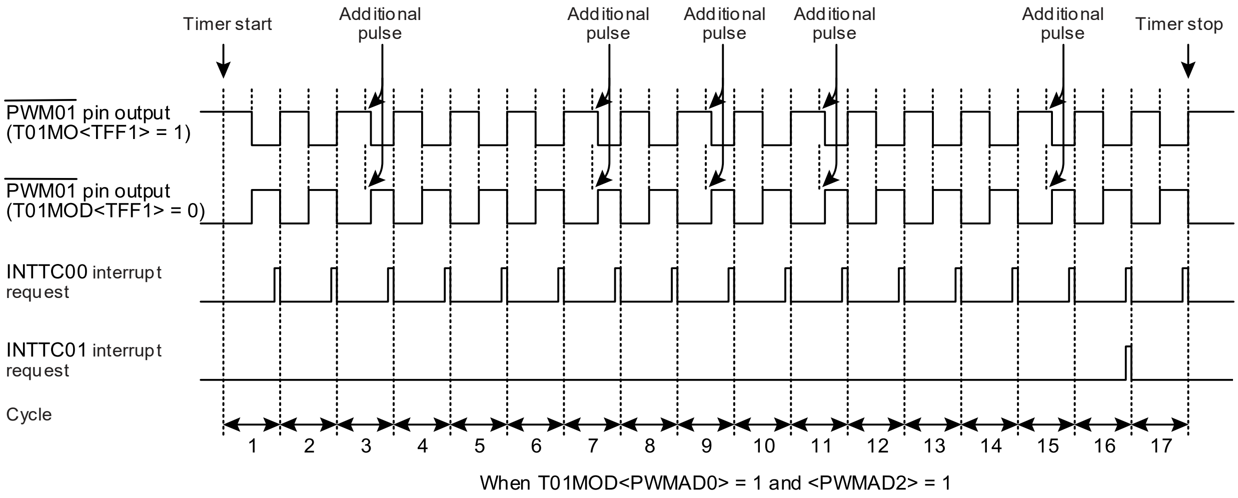

Timer start
Additional pulse
Additional pulse
Additional pulse
Additional pulse
Additional pulse
Timer stop
PWM01 pin output (T01MO<TFF1> = 1)
PWM01 pin output (T01MOD<TFF1> = 0)
INTTC00 interrupt request
INTTC01 interrupt request
Cycle
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
When T01MOD<PWMAD0> = 1 and <PWMAD2> = 1

Figure 14-13 Examples of Inserting Additional Pulses

14.5.7.2. Operations

Setting T001CR<T01RUN> to "1" enables the up counter to count up based on the selected source clock. When a match between the lower 8 bits of the up counter value and the value set to <PWMDUTY> is detected, the output of the $\overline{\text{PWM01}}$ pin is reversed. When T01MOD<TFF1> is "0", the $\overline{\text{PWM01}}$ pin changes from the “Low” to “High” level. When T01MOD<TFF1> is "1", the $\overline{\text{PWM01}}$ pin changes from the “High” to “Low” level.

When any of <PWMAD3 to 0> is "1", an additional pulse that corresponds to 1 count of the source clock is inserted in specific cycles of the duty pulse. In other words, the $\overline{\text{PWM01}}$ pin output is reversed at the timing of <PWMDUTY>+1. When T00MOD<TFF0> is "0", the period of the “Low” level becomes longer than the value set to <PWMDUTY> by 1 source clock. When T00MOD<TFF0> is "1", the period of the “High” level becomes longer than the value set to <PWMDUTY> by 1 source clock. This function allows 16 cycles of output pulses to be handled with a resolution nearly equivalent to 12 bits.

No additional pulse is inserted when <PWMAD3 to 0> are all "0".

Subsequently, the up counter continues counting up. When the up counter value reaches 256, an overflow occurs and the up counter is cleared to "0x00". At the same time, the output of the $\overline{\text{PWM01}}$ pin is reversed. When T01MOD<TFF1> is "0", the $\overline{\text{PWM01}}$ pin changes from the “High” to “Low” level. When T01MOD<TFF1> is "1", the $\overline{\text{PWM01}}$ pin changes from the “Low” to “High” level. At this time, the INTTC00 interrupt request is generated (the INTTC00 interrupt request is generated each time an overflow occurs.) An INTTC01 interrupt request is generated at the (16 × n)-th (n = 1, 2, 3...) overflow. Subsequently, the up counter continues counting up.

When T001CR<T01RUN> is set to "0" during the timer operation, the up counter is stopped and cleared to "0x00". The $\overline{\text{PWM01}}$ pin returns to the level selected at T01MOD<TFF1>.

When an external source clock is selected, input the clock at the TC00 pin. The maximum frequency to be supplied is fcgck / 2 [Hz] (in NORMAL1/2 or IDLE1/2 mode) or fs / 2^4 [Hz] (in SLOW1/2 or SLEEP1 mode), and a pulse width of two machine cycles or more is required at both the “High” and “Low” levels.

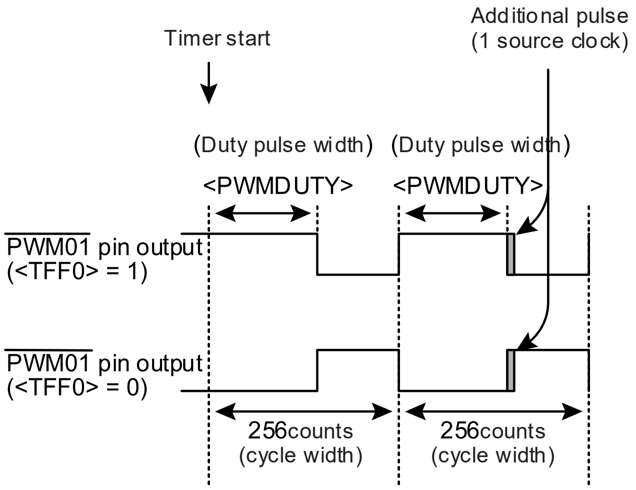


Figure 14-14 $\overline{\text{PWM01}}$ Pin Output

14.5.7.3. Double buffer

In the TMP89FS60B/62B/63B, the double buffer for T01+00PWM can be used by setting T01MOD<DBE1>. The double buffer is disabled by setting T01MOD<DBE1> to "0" or enabled by setting T01MOD<DBE1> to "1".

- When the double buffer is enabled

  When write instructions to T00PWM and T01PWM are executed in this order during the timer operation, the written value is stored in the double buffer, and T01+00PWM is not updated. T01+00PWM compares the previous written value with the up counter value. When the (16 × n)-th overflow occurs, the INTTC01 interrupt request is generated and the double buffer set value is stored in T01+00PWM. Subsequently, the match detection is executed using a new stored value.

  When a read instruction from T01+00PWM is executed, the value in the double buffer (the last set value) is read, not the T01+00PWM value (the currently effective value).

  When write instructions to T00PWM and T01PWM are executed in this order while the timer is stopped, the set value is immediately stored in both the double buffer and T01+00PWM.

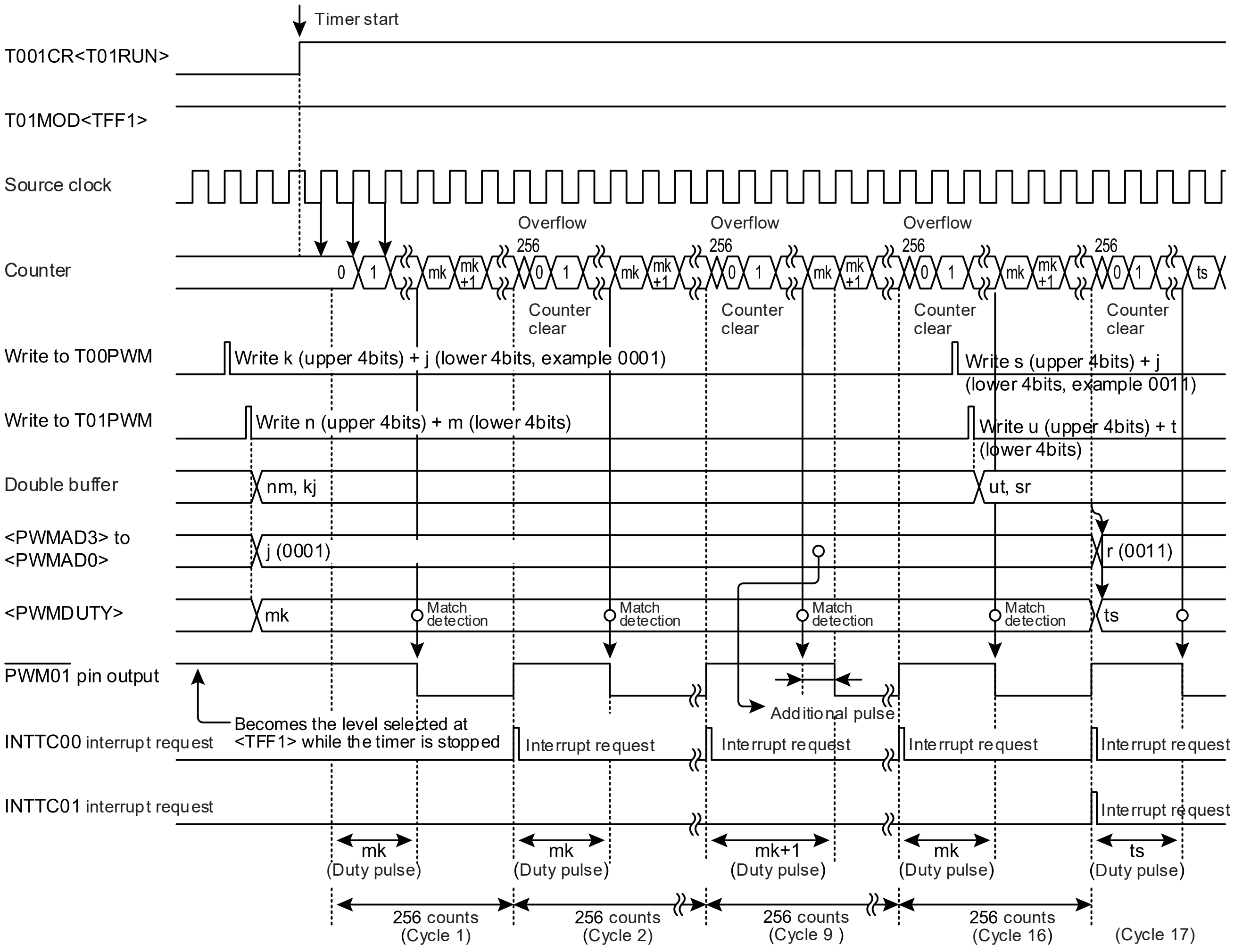
- When the double buffer is disabled

  When write instructions to T00PWM and T01PWM are executed in this order while the timer counter is operated, the set value is immediately stored in T01+00PWM. Subsequently, the match detection is executed using a new set value. When the value set to T01+00PWM is smaller than the up counter value, the $\overline{\text{PWM01}}$ pin is not reversed until the up counter overflows and a match detection is executed using a new set value. When the value set to T01+00PWM is equal to the up counter value, the match detection is executed immediately after data is written into T01+00PWM. Therefore, the timing of changing the $\overline{\text{PWM01}}$ pin may not be an integral multiple of the source clock. Similarly, when T01+00PWM is set during the additional pulse output, the timing of changing the $\overline{\text{PWM01}}$ pin may not be an integral multiple of the source clock. When these are problems, enable the double buffer.

  When write instructions to T00PWM and T01PWM are executed in this order while the timer counter is stopped, the set value is immediately stored in T01+00PWM.

Example: Operate TC00 and TC01 in the 12-bit PWM mode with the operation clock of fcgck / 2 and output a duty pulse nearly equivalent to 14.0625 [μs] in 51.2 [μs] cycles (fcgck = 10 [MHz])
(Actually, output a duty pulse of 225 [μs] in total in 16 cycles (819.2 [μs]))

```
SET     (P7FC).1            ; Set <P7FC1> to "1"
SET     (P7CR).1            ; Set <P7CR1> to "1"
LD      (POFFCR0), 0x10     ; Set <TC001EN> to "1"
DI                          ; Set the interrupt master enable flag to "disable"
SET     (EIRH).4            ; Set the INTTC00 interrupt enable register to "1"
EI                          ; Set the interrupt master enable flag to "enable"
LD      (T01MOD), 0xF2      ; Select the 12-bit PWM mode and fcgck / 2
LD      (T00PWM), 0x65      ; Set the timer register (duty pulse)
                            ; (14.0625 [μs] × 16) / (2 / fcgck) = 0x465
LD      (T01PWM), 0x04      ; Set the timer register (duty pulse)
LD      (T001CR), 0x06      ; Start TC00 and TC01
```



When the double buffer is enabled (T01MOD<DBE1> = 1)

Figure 14-15 12-bit PWM Mode Timing Chart

Table 14-13 Resolutions and Cycles in the 12-bit PWM Mode

T01MOD <TCK1>	Source clock [Hz]			Resolution		8-bit cycle (cycle × 16)	
	NORMAL1/2 or IDLE1/2 mode		SLOW1/2 or SLEEP1 mode	fcgck = 10 [MHz]	fs = 32.768 [kHz]	fcgck = 10 [MHz]	fs = 32.768 [kHz]
	SYSCR1 <DV9CK> = 0	SYSCR1 <DV9CK> = 1					
000	fcgck / 2^{11}	fs / 2^4	fs / 2^4	204.8 [μs]	488.2 [μs]	52.4 [ms] (838.9 [ms])	125 [ms] (2000 [ms])
001	fcgck / 2^{10}	fs / 2^3	fs / 2^3	102.4 [μs]	244.1 [μs]	26.2 [ms] (419.4 [ms])	62.5 [ms] (1000 [ms])
010	fcgck / 2^8	fcgck / 2^8	-	25.6 [μs]	-	6.6 [ms] (104.9 [ms])	-
011	fcgck / 2^6	fcgck / 2^6	-	6.4 [μs]	-	1.6 [ms] (26.2 [ms])	-
100	fcgck / 2^4	fcgck / 2^4	-	1.6 [μs]	-	409.6 [μs] (6.6 [ms])	-
101	fcgck / 2^2	fcgck / 2^2	-	400 [ns]	-	102.4 [μs] (1.6 [ms])	-
110	fcgck / 2	fcgck / 2	-	200 [ns]	-	51.2 [μs] (819.2 [μs])	-
111	fcgck	fcgck	fs / 2^2	100 [ns]	122.1 [μs]	25.6 [μs] (409.6 [μs])	31.3 [ms] (500 [ms])

14.5.8. 16-bit programmable pulse generate (PPG) output mode

In the 16-bit PPG mode, TC00 and TC01 are cascaded to output the pulses that have a resolution of 16 bits and arbitrary pulse width and duty. Two 16-bit registers, T01+00REG and T01+00PWM, are used to output the pulses. This enables output of longer pulses than an 8-bit timer.

14.5.8.1. Setting

Setting T001CR<TCAS> to "1" connects TC00 and TC01 and activates the 16-bit mode. All the settings of TC00 are ignored and those of TC01 are effective in the 16-bit mode.

The 16-bit PPG mode is selected by setting T01MOD<TCM1> to "11". To use the internal clock as the source clock, set T01MOD<EIN1> to "0" and select the clock at T01MOD<TCK1>. To use an external clock as the source clock, set T01MOD<EIN1> to "1".

Set T01MOD<DBE1> to "1" to use the double buffer.

Set the count value that corresponds to a cycle as a 16-bit value at the timer registers T01REG and T00REG. Set the count value that corresponds to a duty pulse as a 16-bit value at T01PWM and T00PWM (hereinafter, the 16-bit value specified by the combined setting of T01REG and T00REG is indicated as T01+00REG, and the 16-bit value specified by the combined setting of T01PWM and T00PWM is indicated as T01+00PWM). The timer register settings are stored on the double buffer or T01+00PWM and T01+00REG when a write instruction to T01PWM is executed. Be sure to execute the write instructions to T00REG, T01REG and T00PWM before executing a write instruction to T01PWM. (When data is written to T01PWM, the set values of the four timer registers become effective at the same time.)

Set the initial state of the $\overline{\text{PPG01}}$ pin at T01MOD<TFF1>. Setting T01MOD<TFF1> to "0" selects the “Low” level as the initial state of the $\overline{\text{PPG01}}$ pin. Setting T01MOD<TFF1> to "1" selects the “High” level as the initial state of the $\overline{\text{PPG01}}$ pin. When the $\overline{\text{PPG01}}$ pin is set as the function output pin in the port setting while the timer is stopped, the value of T01MOD<TFF1> is output to the $\overline{\text{PPG01}}$ pin. Table 14-14 shows the list of output levels of the $\overline{\text{PPG01}}$ pin.

Table 14-14 List of Output Levels of $\overline{\text{PPG01}}$ Pin

<TFF1>	$\overline{\text{PPG01}}$ pin output level			
	Before the start of operation (initial state)	T01+00PWM matched	T01+00REG matched	Operation stopped (initial state)
0	“Low” level	“High” level	“Low” level	“Low” level
1	“High” level	“Low” level	“High” level	“High” level

14.5.8.2. Operations

Setting T001CR<T01RUN> to "1" enables the up counter to count up based on the selected source clock. When a match between the up counter value and the value set to T01+00PWM is detected, the output of the $\overline{\text{PPG01}}$ pin is reversed. When T01MOD<TFF1> is "0", the $\overline{\text{PPG01}}$ pin changes from the “Low” to “High” level. When T01MOD<TFF1> is "1", the $\overline{\text{PPG01}}$ pin changes from the “High” to “Low” level. At this time, an INTTC00 interrupt request is generated.

The up counter continues counting up. When a match between the up counter value and the value set to T01+00REG is detected, the output of the $\overline{\text{PPG01}}$ pin is reversed again. When T01MOD<TFF1> is "0", the $\overline{\text{PPG01}}$ pin changes from the “High” to “Low” level. When T01MOD<TFF1> is "1", the $\overline{\text{PPG01}}$ pin changes from the “Low” to “High” level. At this time, an INTTC01 interrupt request is generated and the up counter is cleared to "0x0000".

When T001CR<T01RUN> is cleared to "0" during the timer operation, the up counter is stopped and cleared to "0x0000". The $\overline{\text{PPG01}}$ pin returns to the level selected at T01MOD<TFF1>.

When an external source clock is selected, input the clock at the TC00 pin. The maximum frequency to be supplied is fcgck / 2 [Hz] (in NORMAL1/2 or IDLE1/2 mode) or fs / 2^4 [Hz] (in SLOW1/2 or SLEEP1 mode), and a pulse width of two machine cycles or more is required at both the “High” and “Low” levels.

14.5.8.3. Double buffer

The double buffer for T01+00PWM and T01+00REG can be used by setting T01MOD<DBE1>. The double buffer is enabled by setting T01MOD<DBE1> to "0" or disabled by setting T01MOD<DBE1> to "1".

– When the double buffer is enabled

When a write instruction to T01PWM is executed after write instructions to T00REG, T01REG and T00PWM are executed, while the timer counter is operated, the set values are stored in the double buffer, and T01+00PWM and T01+00REG are not updated. T01+00PWM and T01+00REG compare the previous written values with the up counter value. When a match between the up counter value and the T01+00REG set value is detected, the INTTC01 interrupt request is generated and the double buffer set values are stored in T01+00PWM and T01+00REG. Subsequently, the match detection is executed using new stored values.

When a write instruction to T01PWM is executed after write instructions to T00REG, T01REG and T00PWM are executed while the timer counter is stopped, the set values are immediately stored in both the double buffer and T01+00PWM and T01+00REG.

– When the double buffer is disabled

When a write instruction to T01PWM is executed after write instructions to T00REG, T01REG and T00PWM are executed during the timer operation, the set values are immediately stored in T01+00PWM and T01+00REG. Subsequently, the match detection is executed using new set values.

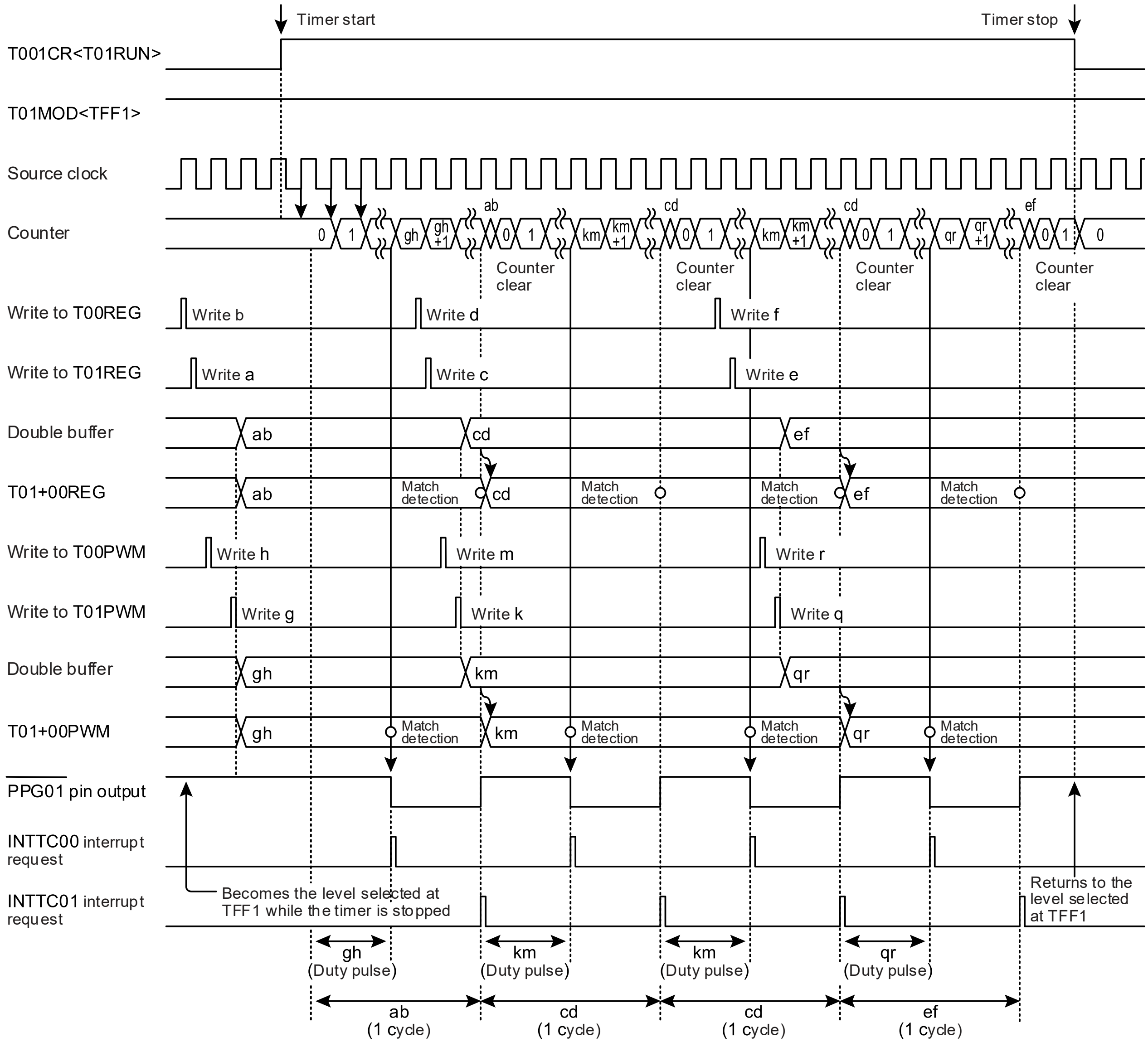
When the value written to T01+00PWM or T01+00REG is smaller than the up counter value, the $\overline{\text{PPG01}}$ pin is not reversed until the up counter overflows and a match detection is executed using a new written value. When the value set to T01+00PWM or T01+00REG is equal to the up counter value, the match detection is executed immediately after data is written into T01+00PWM and T01+00REG. Therefore, the timing of changing the $\overline{\text{PPG01}}$ pin may not be an integral multiple of the source clock. When these are problems, enable the double buffer.

When a write instruction to T01PWM is executed after write instructions to T00REG, T01REG and T00PWM are executed while the timer counter is stopped, the set values are immediately stored in T01+00PWM and T01+00REG.

When read instructions from T01+00PWM and T01+00REG are executed, the last values written into T01+00REG and T01+00PWN are read, regardless of the T00MOD<DBE1> setting.

Example: Operate TC00 and TC01 in the 16-bit PPG mode with the operation clock of fcgck / 2 and output the 68 [μs] duty pulse in 96 [μs] cycles (fcgck = 10 [MHz])

```
SET     (P7FC).1           ; Sets <P7FC1> to "1"
SET     (P7CR).1           ; Sets <P7CR1> to "1"
LD      (POFFCR0), 0x10    ; Sets <TC001EN> to "1"
DI                         ; Sets the interrupt master enable flag to "disable"
SET     (EIRH).4           ; Sets the INTTC00 interrupt enable register to "1"
EI                         ; Sets the interrupt master enable flag to "enable"
LD      (T01MOD), 0xF3     ; Selects the 16-bit PPG mode and fcgck/2
LD      (T00REG), 0xE0     ; Sets the timer register (cycle)
LD      (T01REG), 0x01     ; Sets the timer register (cycle)
                           ; 96 [μs] / (2 / fcgck) = 0x01E0
LD      (T00PWM), 0x54     ; Sets the timer register (duty pulse)
LD      (T01PWM), 0x01     ; Sets the timer register (duty pulse)
                           ; 68 [μs] / (2 / fcgck) = 0x0154
LD      (T001CR), 0x06     ; Starts TC00 and TC01
```



When the double buffer is enabled (T01MOD<DBE1> = 1)

Figure 14-16 16-bit PPG Output Mode Timing Chart

15. Real Time Clock (RTC)

The real time clock is a function that generates interrupt requests at certain intervals using the low-frequency clock.

The number of interrupts is counted by the software to realize the clock function.

The real time clock can be used only in the operation modes where the low-frequency clock oscillates, except for SLEEP0.

15.1. Configuration

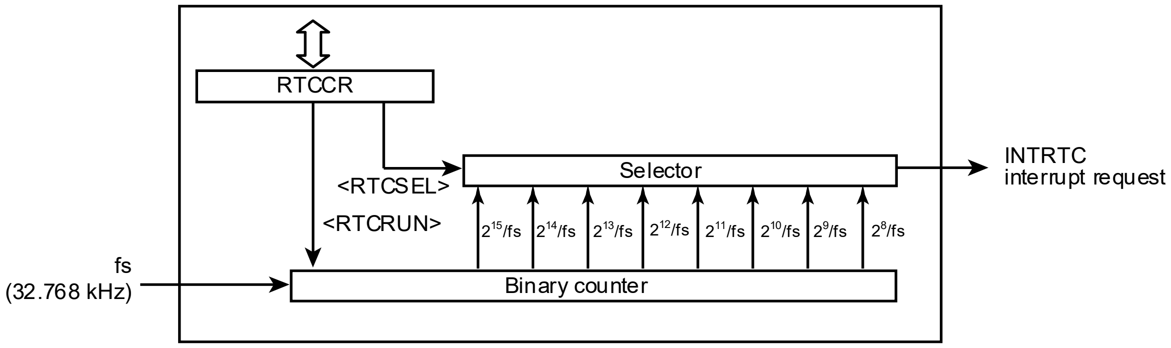


Figure 15-1 Real Time Clock

15.2. Control

The real time clock is controlled by following resisters.

Low power consumption register 2

POFFCR2		7	6	5	4	3	2	1	0
(0x0F76)	Bit Symbol	-	-	RTCEN	-	-	-	(SIO1EN)	(SIO0EN)
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

RTCEN	RTC control	0: 1:	Disable Enable

Note: The written values are read from the bits 7 and 6, 4 to 2 of POFFCR2. These bits must be cleared to “0”.

Real time clock control register

RTCCR		7	6	5	4	3	2	1	0
(0x0FC8)	Bit Symbol	-	-	-	-	RTCSEL			RTCRUN
	Read/Write	R	R	R	R	R/W			R/W
	After reset	0	0	0	0	0	0	0	0

RTCSEL	Selects the interrupt generation	000: 001: 010: 011: 100: 101: 110: 111:	2^{15} / fs (1.000 [s] @fs = 32.768 [kHz]) 2^{14} / fs (0.500 [s] @fs = 32.768 [kHz]) 2^{13} / fs (0.250 [s] @fs = 32.768 [kHz]) 2^{12} / fs (125.0 [ms] @fs = 32.768 [kHz]) 2^{11} / fs (62.50 [ms] @fs = 32.768 [kHz]) 2^{10} / fs (31.25 [ms] @fs = 32.768 [kHz]) 2^{9} / fs (15.62 [ms] @fs = 32.768 [kHz]) 2^{8} / fs (7.81 [ms] @fs = 32.768 [kHz])
RTCRUN	Enables/disables the real time clock operation	0: 1:	Disable Enable

Note 1: fs: Low-frequency clock [Hz]

Note 2: RTCCR<RTCSEL> can be modified only when RTCCR<RTCRUN> is "0". When data is written into RTCCR<RTCSEL> when RTCCR<RTCRUN> is "1", the existing data remains effective. RTCCR<RTCSEL> can be modified at the same time as enabling the real time clock, but it cannot be modified at the same time as disabling the real time clock.

Note 3: When the real time clock is enabled and when 1) SYSCR2<XTEN> is cleared to "0" to stop the low-frequency clock oscillation circuit or 2) the operation is changed to the STOP mode or the SLEEP0 mode, the data in RTCCR<RTCSEL> is maintained and RTCCR<RTCRUN> is cleared to "0".

15.3. Low Power Consumption Control

Real time clock has the low power consumption registers (POFFCR2) that save power when the real time clock is not being used.

Setting POFFCR2<RTCEN> to "0" disables the basic clock supply to real time clock to save power. Note that this makes the real time clock unusable. Setting POFFCR2<RTCEN> to "1" enables the basic clock supply to real time clock and enables the real time clock to operate.

After reset, POFFCR2<RTCEN> are initialized to "0", and this makes the real time clock unusable. When using the real time clock for the first time, be sure to set POFFCR2<RTCEN> to "1" in the initial setting of the program (before the real time clock control registers are operated).

Do not change POFFCR2<RTCEN> to "0" during the real time clock operation. Otherwise real time clock may operate unexpectedly.

15.4. Function

15.4.1. Enabling/disabling the real time clock operation

Setting RTCCR<RTCRUN> to "1" enables the real time clock operation. Setting RTCCR<RTCRUN> to "0" disables the real time clock operation.

RTCCR<RTCRUN> is cleared to "0" just after reset release.

15.4.2. Selecting the interrupt generation interval

The interrupt generation interval can be selected at RTCCR<RTCSEL>.

RTCCR<RTCSEL> can be rewritten only when RTCCR<RTCRUN> is "0".

When data is written into RTCCR<RTCSEL> when RTCCR<RTCRUN> is "1", the existing data remains effective.

RTCCR<RTCSEL> can be rewritten at the same time as enabling the real time clock operation, but it cannot be rewritten at the same time as disabling the real time clock operation.

15.5. Real Time Clock Operation

15.5.1. Enabling the real time clock operation

Set the interrupt generation interval to RTCCR<RTCSEL>, and at the same time, set RTCCR<RTCRUN> to "1".

When RTCCR<RTCRUN> is set to "1", the binary counter for the real time clock starts counting of the low-frequency clock.

When the interrupt generation interval selected at RTCCR<RTCSEL> is reached, a real time clock interrupt request (INTRTC) is generated and the counter continues counting.

15.5.2. Disabling the real time clock operation

Clear RTCCR<RTCRUN> to "0".

When RTCCR<RTCRUN> is cleared to "0", the binary counter for the real time clock is cleared to "0" and stops counting of the low-frequency clock.

16. Asynchronous Serial Interface (UART)

The TMP89FS60B/62B/63B contains the asynchronous serial interfaces (UART).

This chapter describes the UART0. For UART1 and UART2, replace the SFR addresses and pin names as shown in Table 16-1 and Table 16-2.

Table 16-1 SFR Address Assignment

	UARTxCR1 (address)	UARTxCR2 (address)	UARTxDR (address)	UARTxSR (address)	RDxBUF (address)	TDxBUF (address)	Low power consumption register
UART0	UART0CR1 (0x001A)	UART0CR2 (0x001B)	UART0DR (0x001C)	UART0SR (0x001D)	RD0BUF (0x001E)	TD0BUF (0x001E)	POFFCR1 <UART0EN>
UART1	UART1CR1 (0x0F54)	UART1CR2 (0x0F55)	UART1DR (0x0F56)	UART1SR (0x0F57)	RD1BUF (0x0F58)	TD1BUF (0x0F58)	POFFCR1 <UART1EN>
UART2	UART2CR1 (0x0F5A)	UART2CR2 (0x0F5B)	UART2DR (0x0F5C)	UART2SR (0x0F5D)	RD2BUF (0x0F5E)	TD2BUF (0x0F5E)	POFFCR1 <UART2EN>

Table 16-2 Pin Names

	Serial data input pin	Serial data output pin
UART0	RXD0 pin	TXD0 pin
UART1	RXD1 pin	TXD1 pin
UART2	RXD2 pin	TXD2 pin

16.1. Asynchronous Serial Interfaces (UART) for Each Product

The Asynchronous serial interfaces (UART) for each product is shown in Table 16-3.

In regards to unavailable channel of the UART, the corresponding bit of the low power register must be cleared to “0”.

Table 16-3 Asynchronous Serial Interface (UART) for Each Product

	TMP89FS60B	TMP89FS62B	TMP89FS63B
UART0	A	A	A
UART1	A	A	A
UART2	A	A	A

Note: A: Available

16.2. Configuration

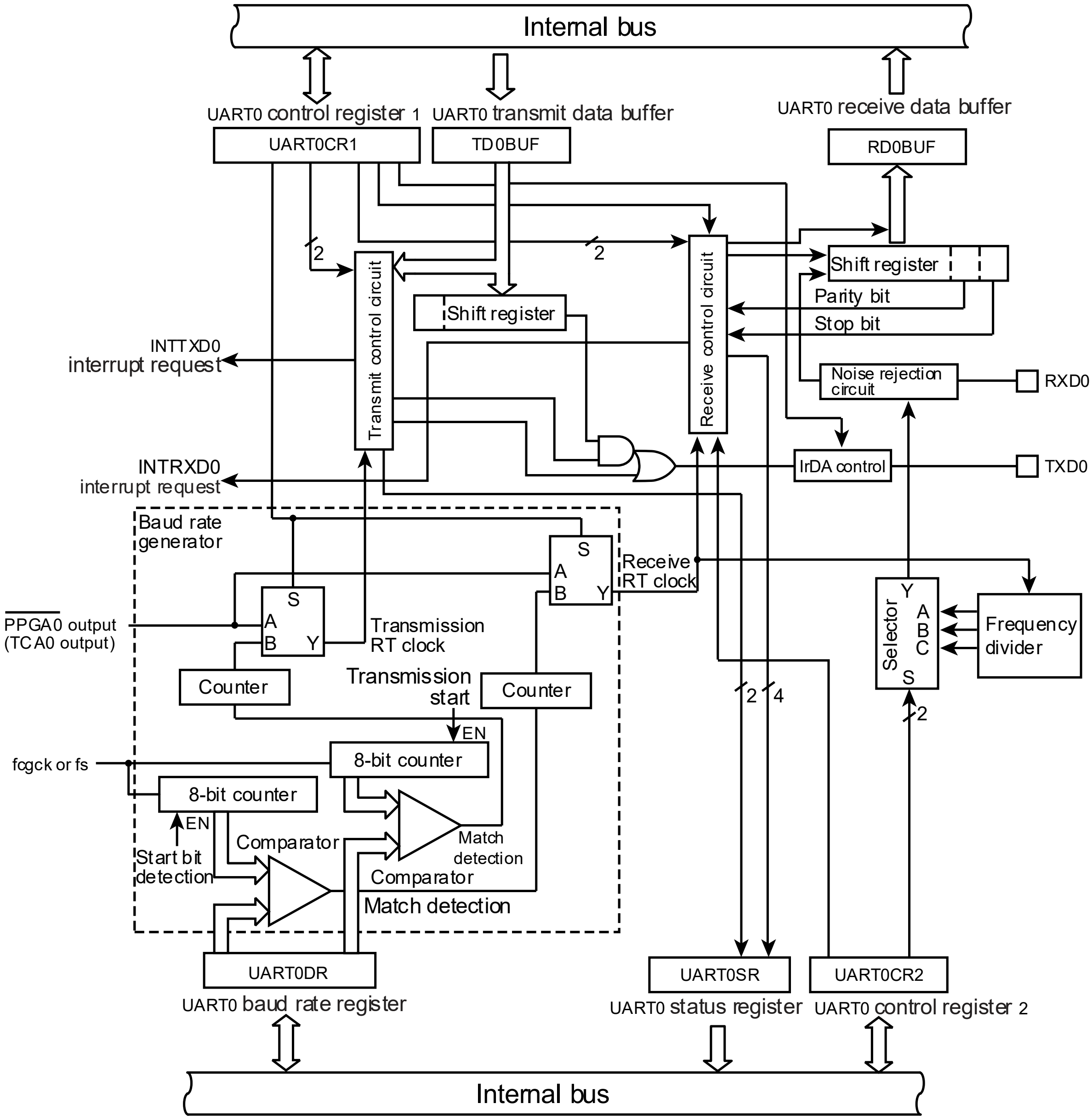


Figure 16-1 Asynchronous Serial Interface (UART)

Note: The UART input/output pins are also used as the I/O ports. The I/O port register settings are required to use these pins for the UART. For details, refer to "8. I/O Ports".

16.3. Control

UART0 is controlled by the low power consumption registers (POFFCR1), UART0 control registers 1 and 2 (UART0CR1 and UART0CR2) and the UART0 baud rate register (UART0DR). The operating status can be monitored using the UART status register (UART0SR).

Low power consumption register 1

POFFCR1		7	6	5	4	3	2	1	0
(0x0F75)	Bit Symbol	-	-	-	(SBI0EN)	-	UART2EN	UART1EN	UART0EN
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

UART2EN	UART2 control	0: 1:	Disable Enable
UART1EN	UART1 control	0: 1:	Disable Enable
UART0EN	UART0 control	0: 1:	Disable Enable

Note 1: The written values are read from the bits 7 to 5, 3 of POFFCR1. These bits must be cleared to “0”.

Note 2: <SBI0EN> is reserved for the TMP89FS62B.

UART0 control register 1

UART0CR1		7	6	5	4	3	2	1	0
(0x001A)	Bit Symbol	TXE	RXE	STOPBT	EVEN	PE	IRDASEL	BRG	-
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R
	After reset	0	0	0	0	0	0	0	0

TXE	Transmit operation	0:	Disable	
		1:	Enable	
RXE	Receive operation	0:	Disable	
		1:	Enable	
STOPBT	Transmit stop bit length	0:	1 bit	
		1:	2 bits	
EVEN	Parity selection	0:	Odd parity	
		1:	Even parity	
PE	Parity enable/ disable selection	0:	No parity	
		1:	With parity	
IRDASEL	TXD0 pin output selection	0:	UART output	
		1:	IrDA output	
BRG	Transfer base clock selection		SYSCR2<SYSCK> = 0	SYSCR2<SYSCK> = 1
		0:	fcgck	fs
		1:	TCA0 output	TCA0 output

Note 1: fcgck: Gear clock [Hz], fs: Low-frequency clock [Hz]

Note 2: When <TXE> or <RXE> is cleared to "0" during the transmission or receiving of data, the operation is not disabled until the data transfer is completed. At this time, the data stored in the transmit data buffer is discarded.

Note 3: <EVEN>, <PE> and <BRG> settings are common to transmission and receiving.

Note 4: Clear <RXE> and <TXE> to "0" before changing <BRG>.

Note 5: When <BRG >is set to the TCA0 output, the RT clock becomes asynchronous and the start bit of the transmitted/received data may get shorter by a maximum of (UART0DR+1) / (Transfer base clock frequency) [s]. When the pin is not used for the TCA0 output, control the TCA0 output by using the port function control register.

Note 6: To prevent <STOPBT>, <EVEN>, <PE>, <IRDASEL> and <BRG> from being changed accidentally during the UART communication, the register cannot be rewritten during the UART operation. For details, refer to "16.5. Protection to Prevent UART0CR1 and UART0CR2 Registers from Being Changed".

Note 7: When the STOP, IDLE0 or SLEEP0 mode is activated, <TXE> and <RXE> are cleared to "0" and the UART stops. Other bits keep their values.

UART0 control register 2

UART0CR2 (0x001B)		7	6	5	4	3	2	1	0
	Bit Symbol	-	-	RTSEL			RXDNC		STOPBR
	Read/Write	R	R	R/W			R/W		R/W
	After reset	0	0	0	0	0	0	0	0

			Odd bits of transfer frame	Even bits of transfer frame
RTSEL	Selects the number of RT clocks	000:	16 clocks	16 clocks
		001:	16 clocks	17 clocks
		010:	15 clocks	15 clocks
		011:	15 clocks	16 clocks
		100:	17 clocks	17 clocks
		101:	Reserved	Reserved
		110:	Reserved	Reserved
		111:	Reserved	Reserved
RXDNC	Selects the RXD0 input noise rejection time (Time of pulses to be removed as noise)	00:	No noise rejection	
		01:	1×(UART0DR+1) / (Transfer base clock frequency) [s]	
		10:	2×(UART0DR+1) / (Transfer base clock frequency) [s]	
		11:	4×(UART0DR+1) / (Transfer base clock frequency) [s]	
STOPBR	Receive stop bit length	0:	1 bit	
		1:	2 bits	

Note 1: When a read instruction from UART0CR2 is executed, bits 7 and 6 are read as "0".

Note 2: <RTSEL> can be set to two kinds of RT clocks for the even and odd bits of the transfer frame. For details, refer to "16.9.1. Transfer baud rate calculation method".

Note 3: For details of the <RXDNC> noise rejection time, refer to "16.11. Received Data Noise Rejection".

Note 4: When the STOP, IDLE0 or SLEEP0 mode is activated, the UART stops automatically but each bit value of UART0CR2 remains unchanged.

Note 5: When <STOPBR> is set to 2 bits, the first bit of the stop bits (during data receiving) is not checked for a framing error.

Note 6: To prevent <RTSEL>, <RXDNC> and <STOPBR> from being changed accidentally during the UART communication, the register cannot be rewritten during the UART operation. For details, refer to "16.5. Protection to Prevent UART0CR1 and UART0CR2 Registers from Being Changed".

UART0 baud rate register

UART0DR (0x001C)		7	6	5	4	3	2	1	0
	Bit Symbol	UART0DR7	UART0DR6	UART0DR5	UART0DR4	UART0DR3	UART0DR2	UART0DR1	UART0DR0
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

Note 1: UART0DR must be changed when UART0CR1<RXE> and <TXE> are "0". For the set values, refer to "16.9. Transfer Baud Rate".

Note 2: The board rate of the UART is determined by UART0DR only when UARTCR1<BRG> is “0” (the transfer base clock is the fcgck or the fs).

Note 3: When the STOP, IDLE0 or SLEEP0 mode is activated, the UART stops automatically but each bit value of UART0DR remains unchanged.

UART0 status register

UART0SR (0x001D)		7	6	5	4	3	2	1	0
	Bit Symbol	PERR	FERR	OERR	-	RBSY	RBFL	TBSY	TBFL
	Read/Write	R	R	R	R	R	R	R	R
	After reset	0	0	0	0	0	0	0	0

PERR	Parity error flag	0: 1:	No parity error occurs Parity error occurs
FERR	Framing error flag	0: 1:	No framing error occurs Framing error occurs
OERR	Overrun error flag	0: 1:	No overrun error occurs Overrun error occurs
RBSY	Receive busy flag	0: 1:	Before receiving or completion of receiving On receiving
RBFL	Receive buffer full flag	0: 1:	Receive buffer is empty Receive buffer is full
TBSY	Transmit busy flag	0: 1:	Before transmission or completion of transmission On transmitting
TBFL	Transmit buffer full flag	0: 1:	Transmit buffer is empty Transmit buffer is full. (Transmit data has been written.)

Note 1: <TBFL> is cleared to "0" automatically after an INTTXD0 interrupt request is generated, and is set to "1" when data is set to TD0BUF.

Note 2: When a read instruction from UART0SR is executed, bit 4 is read as "0".

Note 3: When the STOP, IDLE0 or SLEEP0 mode is activated, each bit of UART0SR is cleared to "0" and the UART stops.

UART0 receive data buffer

RD0BUF (0x001E)		7	6	5	4	3	2	1	0
	Bit Symbol	RD0DR7	RD0DR6	RD0DR5	RD0DR4	RD0DR3	RD0DR2	RD0DR1	RD0DR0
	Read/Write	R	R	R	R	R	R	R	R
	After reset	0	0	0	0	0	0	0	0

Note: When the STOP, IDLE0 or SLEEP0 mode is activated, the RD0BUF values become undefined. When received data is required, read it before activating these mode.

UART0 transmit data buffer

TD0BUF (0x001E)		7	6	5	4	3	2	1	0
	Bit Symbol	TD0DR7	TD0DR6	TD0DR5	TD0DR4	TD0DR3	TD0DR2	TD0DR1	TD0DR0
	Read/Write	W	W	W	W	W	W	W	W
	After reset	0	0	0	0	0	0	0	0

Note: When the STOP, IDLE0 or SLEEP0 mode is activated, the TD0BUF values become undefined.

16.4. Low Power Consumption Control

The UART0 has a low power consumption register (POFFCR1) that saves power consumption when the UART0 function is not used.

Setting POFFCR1<UART0EN> to "0" disables the basic clock supply to the UART0 to save power. Note that this makes the UART0 unusable. Setting POFFCR1<UART0EN> to "1" enables the basic clock supply to the UART0 and makes the UART0 usable.

After reset, POFFCR1<UART0EN> is initialized to "0", and this makes the UART0 unusable. When using the UART0 for the first time, be sure to set POFFCR1<UART0EN> to "1" in the initial setting of the program (before the UART0 control registers are modified).

Do not change POFFCR1<UART0EN> to "0" while the UART0 is operated, otherwise, UART0 may operate unexpectedly.

16.5. Protection to Prevent UART0CR1 and UART0CR2 Registers from Being Changed

The TMP89FS60B/62B/63B has a function that protects the registers from being changed so that the UART communication settings (for example, stop bit and parity) are not changed accidentally during the UART operation.

Specific bits of UART0CR1 and UART0CR2 can be changed only under the conditions shown in Table 16-4. When a write instruction to the register is executed while it is protected from being changed, the bits remain unchanged and keep their previous values.

Table 16-4 Changing of UART0CR1 and UART0CR2

Bit to be changed	Function	Conditions that allow the bit to be changed			
		UART0CR1<TXE>	UART0SR<TBSY>	UART0CR1<RXE>	UART0SR<RBSY>
UART0CR1<STOPBT>	Transmit stop bit length	Both of these bits are "0"		-	-
UART0CR1<EVEN>	Parity selection	All of these bits are "0"			
UART0CR1<PE>	Parity enable/disable				
UART0CR1<IRDASEL>	TXD0 pin output selection	Both of these bits are "0"		-	-
UART0CR1<BRG>	Transfer base clock selection	All of these bits are "0"			
UART0CR2<RTSEL>	Selection of number of RT clocks				
UART0CR2<RXDNC>	Selection of RXD0 pin input noise rejection time	-	-	Both of these bits are "0"	
UART0CR2<STOPBR>	Receive stop bit length				

16.6. Activation of STOP, IDLE0 or SLEEP0 Mode

16.6.1. Transition of register status

When the STOP, IDLE0 or SLEEP0 mode is activated, the UART stops automatically and each register becomes the status as shown in Table 16-5. For the registers that do not hold their values, make settings again as needed after the operation mode is recovered.

Table 16-5 Transition of Register Status

	7	6	5	4	3	2	1	0
UART0CR1	<TXE>	<RXE>	<STOPBT>	<EVEN>	<PE>	<IRDASEL>	<BRG>	-
	0	0	Hold the value	Hold the value	Hold the value	Hold the value	Hold the value	-
UART0CR2	-	-	<RTSEL>			<RXDNC>		<STOPBR>
	-	-	Hold the value	Hold the value	Hold the value	Hold the value	Hold the value	Hold the value
UART0SR	<PERR>	<FERR>	<OERR>	-	<RBSY>	<RBFL>	<TBSY>	<TBFL>
	0	0	0	-	0	0	0	0
UART0DR	<UART0DR7>	<UART0DR6>	<UART0DR5>	<UART0DR4>	<UART0DR3>	<UART0DR2>	<UART0DR1>	<UART0DR0>
	Hold the value	Hold the value	Hold the value	Hold the value	Hold the value	Hold the value	Hold the value	Hold the value
RD0BUF	<RD0DR7>	<RD0DR6>	<RD0DR5>	<RD0DR4>	<RD0DR3>	<RD0DR2>	<RD0DR1>	<RD0DR0>
	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
TD0BUF	<TD0DR7>	<TD0DR6>	<TD0DR5>	<TD0DR4>	<TD0DR3>	<TD0DR2>	<TD0DR1>	<TD0DR0>
	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

16.6.2. Transition of TXD0 pin status

When the IDLE0, SLEEP0 or STOP mode is activated, the TXD0 pin reverts to the status shown in Table 16-6, whether data is transmitted/received or the operation is stopped.

Table 16-6 TXD0 Pin Status When the STOP, IDLE0 or SLEEP0 Mode Is Activated

UART0CR1 <IRDASEL>	IDLE0 or SLEEP0 mode	STOP mode	
		SYSCR1<OUTEN> = 1	SYSCR1<OUTEN> = 0
0	“High” level	"High" level	“Hi-Z”
1	“Low” level	“Low” level	

16.7. Transfer Data Format

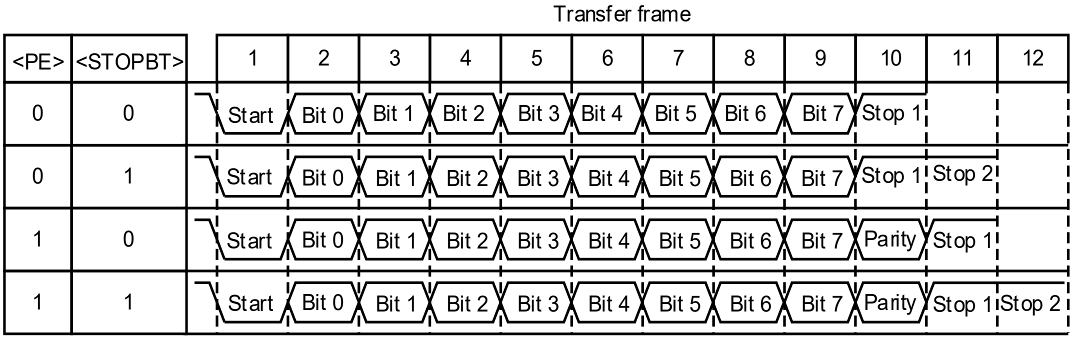
The UART transfers data composed of the following four elements. The data from the start bit to the stop bit is collectively defined as a "transfer frame".

The start bit consists of 1 bit (“Low” level) and the data consists of 8 bits. Parity bits are determined by UART0CR1<PE> that selects the presence or absence of parity and UART0CR1<EVEN> that selects even or odd parity.

The bit length of the stop bit can be selected at UART0CR1<STOPBT>.

Figure 16-2 shows the transfer data format.

- Start bit (1 bit)
- Data (8 bits)
- Parity bit (selectable from no parity or with parity. Selectable even or odd parity when with parity)
- Stop bit (selectable from 1 bit or 2 bits)



Transfer frame

<PE>	<STOPBT>	1	2	3	4	5	6	7	8	9	10	11	12
0	0	Start	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	Stop 1		
0	1	Start	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	Stop 1	Stop 2	
1	0	Start	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	Parity	Stop 1	
1	1	Start	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	Parity	Stop 1	Stop 2

Figure 16-2 Transfer Data Format

16.8. Infrared Data Format Transfer Mode

The TXD0 pin can output data in the infrared data format (IrDA) by the setting of the IrDA output control register. Setting UART0CR1<IRDASEL> to "1" enables the TXD0 pin to output data in the infrared data format.

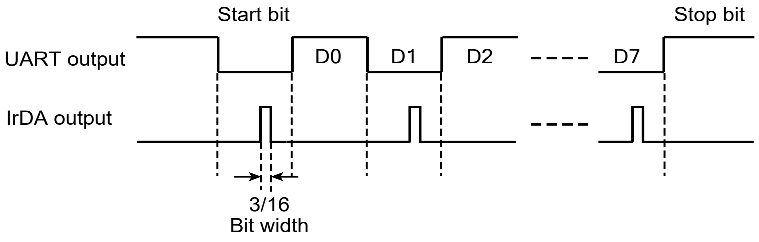


Figure 16-3 Example of Infrared Data Format
(Comparison between Normal Output and IrDA Output)

16.9. Transfer Baud Rate

The transfer baud rate of UART is set by UART0CR1<BRG>, UART0DR and UART0CR2<RTSEL>.

The settings of UART0DR and UART0CR2<RTSEL> and errors for basic baud rates and operating frequencies are shown below.

For calculation of transfer baud rates, refer to "16.9.1. Transfer baud rate calculation method".

Table 16-7 Set Values of UART0DR and UART0CR2<RTSEL> and errors for Transfer Baud Rates (fcgck = 10 to 1 [MHz], UART0CR2<RXDNC> = 00)

Basic baud rate [baud]	Register settings and errors	Operating frequency										
		10 [MHz]	8 [MHz]	7.3728 [MHz]	6.144 [MHz]	6 [MHz]	5 [MHz]	4.9152 [MHz]	4.19 [MHz]	4 [MHz]	2 [MHz]	1 [MHz]
128000	UART0DR	0x04	0x03	-	0x02	0x02	-	-	0x01	0x01	0x00	-
	<RTSEL>	011	011	-	000	011	-	-	001	011	011	-
	Errors	(+0.81%)	(+0.81%)	-	0%	(+0.81%)	-	-	(-0.80%)	(+0.81%)	(+0.81%)	-
115200	UART0DR	0x04	0x03	0x03	-	0x02	-	-	-	0x01	0x00	-
	<RTSEL>	100	100	000	-	100	-	-	-	100	100	-
	Errors	(+2.12%)	(+2.12%)	0%	-	(+2.12%)	-	-	-	(+2.12%)	(+2.12%)	-
76800	UART0DR	0x07	0x06	0x05	0x04	0x04	0x03	0x03	-	0x02	-	-
	<RTSEL>	001	010	000	000	011	001	000	-	100	-	-
	Errors	(-1.36%)	(-0.79%)	0%	0%	(+0.81%)	(-1.36%)	0%	-	(+2.12%)	-	-
62500	UART0DR	0x09	0x07	0x06	0x05	0x05	0x04	0x04	0x03	0x03	0x01	0x00
	<RTSEL>	000	000	100	001	000	000	011	100	000	000	000
	Errors	0%	0%	(-0.87%)	(-0.70%)	0%	0%	(+1.48%)	(-1.41%)	0%	0%	0%
57600	UART0DR	0x0A	0x08	0x07	0x06	0x06	0x04	0x04	-	0x03	0x01	0x00
	<RTSEL>	000	011	000	010	010	100	100	-	100	100	100
	Errors	(-1.36%)	(-0.44%)	0%	(+1.59%)	(-0.79%)	(+2.12%)	(+0.39%)	-	(+2.12%)	(+2.12%)	(+2.12%)
38400	UART0DR	0x10	0x0C	0x0B	0x09	0x09	0x07	0x07	0x06	0x06	0x02	-
	<RTSEL>	011	000	000	000	011	001	000	011	010	100	-
	Errors	(-1.17%)	(+0.16%)	0%	0%	(+0.81%)	(-1.36%)	0%	(+0.57%)	(-0.79%)	(+2.12%)	-
19200	UART0DR	0x22	0x19	0x17	0x13	0x12	0x10	0x0F	0x0D	0x0C	0x06	0x02
	<RTSEL>	010	000	000	000	001	011	000	011	000	010	100
	Errors	(-0.79%)	(+0.16%)	0%	0%	(-0.32%)	(-1.17%)	0%	(+0.57%)	(+0.16%)	(-0.79%)	(+2.12%)
9600	UART0DR	0x40	0x30	0x2F	0x27	0x26	0x22	0x1F	0x1C	0x19	0x0C	0x06
	<RTSEL>	000	100	000	000	000	010	000	010	000	000	010
	Errors	(+0.16%)	(+0.04%)	0%	0%	(+0.16%)	(-0.79%)	0%	(+0.34%)	(+0.16%)	(+0.16%)	(-0.79%)
4800	UART0DR	0x8A	0x64	0x5F	0x4F	0x4D	0x40	0x3F	0x34	0x30	0x19	0x0C
	<RTSEL>	010	01	000	000	000	000	000	001	100	000	000
	Errors	(-0.08%)	(+0.01%)	0%	0%	(+0.16%)	(+0.16%)	0%	(-0.18%)	(+0.04%)	(+0.16%)	(+0.16%)
2400	UART0DR	0xF4	0xC9	0xBF	0x9F	0x92	0x8A	0x7F	0x6C	0x64	0x30	0x19
	<RTSEL>	100	001	000	000	100	010	000	000	001	100	000
	Errors	(+0.04%)	(+0.01%)	0%	0%	(+0.04%)	(-0.08%)	0%	(+0.11%)	(+0.01%)	(+0.04%)	(+0.16%)
1200	UART0DR	-	-	-	-	-	0xF4	0xFF	0xE8	0xC9	0x64	0x30
	<RTSEL>	-	-	-	-	-	100	000	010	001	001	100
	Errors	-	-	-	-	-	(+0.04%)	(+0%)	(-0.10%)	(+0.01%)	(+0.01%)	(+0.04%)

Table 16-8 Set Values of UART0DR and UART0CR2<RTSEL> and errors for Transfer Baud Rates (fs = 32.768 [kHz], UART0CR2<RXDNC> = 00)

Basic baud rate [baud]	Register setting and errors	Operating frequency
		32.768 [kHz]
300	UART0DR	0x06
	<RTSEL>	011
	Errors	(+0.67%)
150	UART0DR	0x0D
	<RTSEL>	011
	Errors	(+0.67%)
134	UART0DR	0x0E
	<RTSEL>	001
	Errors	(-1.20%)
110	UART0DR	0x11
	<RTSEL>	001
	Errors	(+0.30%)
75	UART0DR	0x1C
	<RTSEL>	010
	Errors	(+0.44%)

Note: The overall error from the basic baud rate must be within ±3%. Even when the overall error is within ±3%, the communication may fail due to factors such as frequency errors in external controllers (for example, a personal computer) and oscillators and the load capacitance of the communication pin.

16.9.1. Transfer baud rate calculation method

16.9.1.1. Bit width adjustment using UART0CR2<RTSEL>

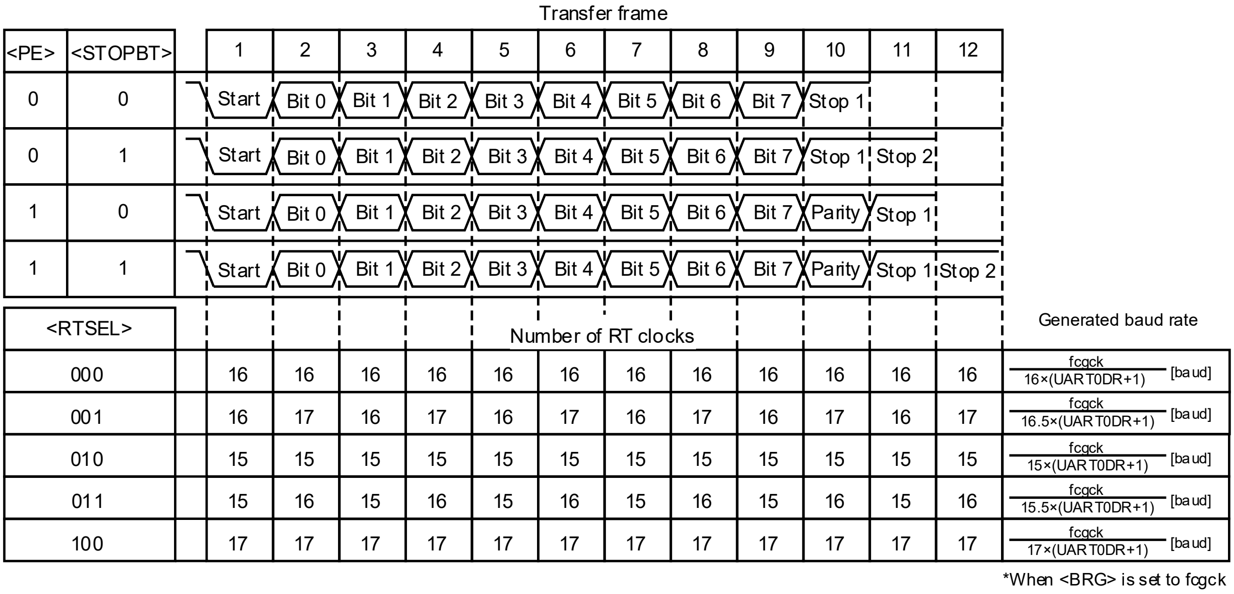
The bit width of transmitted/received data can be finely adjusted by changing UART0CR2<RTSEL>.

The number of RT clocks per bit can be changed in a range of 15 to 17 clocks by changing UART0CR2<RTSEL>. The RT clock is the transfer base clock, which is the pulses obtained by counting the clock selected at UART0CR1<BRG> the number of times of (UART0DR set value) + 1. Especially, when UART0CR2<RTSEL> is set to "001" or "011", two types of RT clocks alternate at each bit, so that the pseudo baud rates of RT × 15.5 clocks or RT × 16.5 clocks can be generated. The number of RT clocks per bit of transfer frame is shown in Figure 16-4.

For example, when fcgck is 4 [MHz], UART0CR2<RTSEL> is set to "0b000" and UART0DR is set to "0x19", the baud rate calculated using the formula in Figure 16-4 is expressed as:

fcgck / (16 × (UART0DR + 1) = 9615 [baud]

These settings generate a baud rate close to 9600 [baud] (+0.16%).



Transfer frame

<PE>	<STOPBT>	1	2	3	4	5	6	7	8	9	10	11	12
0	0	Start	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	Stop 1		
0	1	Start	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	Stop 1	Stop 2	
1	0	Start	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	Parity	Stop 1	
1	1	Start	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	Parity	Stop 1	Stop 2

Number of RT clocks

<RTSEL>	1	2	3	4	5	6	7	8	9	10	11	12	Generated baud rate
000	16	16	16	16	16	16	16	16	16	16	16	16	$\frac{fcgck}{16\times(UART0DR+1)}$ [baud]
001	16	17	16	17	16	17	16	17	16	17	16	17	$\frac{fcgck}{16.5\times(UART0DR+1)}$ [baud]
010	15	15	15	15	15	15	15	15	15	15	15	15	$\frac{fcgck}{15\times(UART0DR+1)}$ [baud]
011	15	16	15	16	15	16	15	16	15	16	15	16	$\frac{fcgck}{15.5\times(UART0DR+1)}$ [baud]
100	17	17	17	17	17	17	17	17	17	17	17	17	$\frac{fcgck}{17\times(UART0DR+1)}$ [baud]

*When <BRG> is set to fcgck

Figure 16-4 Fine Adjustment of Baud Rate Clock Using UART0CR2<RTSEL>

16.9.1.2. Calculation of set values of UART0CR2<RTSEL> and UART0DR

The set value of UART0DR for an operating frequency and baud rate can be calculated using the calculation formula shown in Table 16-9. For example, to generate a basic baud rate of 38400 [baud] with fcgck = 4 [MHz], calculate the set value of UART0DR for each setting of UART0CR2<RTSEL> and compensate the calculated value to an integral number to obtain the generated baud rate as shown in Table 16-10. Basically, select the set value of UART0CR2<RTSEL> that has the smallest baud rate error from among the generated baud rates. In Table 16-10, the setting of UART0CR2<RTSEL> = 010 has the smallest error among the calculated baud rates, and thus the generated baud rate is 38095 [baud] (-0.79%) against the basic baud rate of 38400 [baud].

Note: The error from the basic baud rate should be accurate to within ±3%. When the error is within ±3%, the communication may fail due to factors such as frequency errors of external controllers (for example, a personal computer) and oscillators and the load capacitance of the communication pin.

Table 16-9 UART0DR Calculation Method (When <BRG> is Set to fcgck)

<RTSEL>	UART0DR set value
000	$\text{UART0DR} = \frac{\text{fcgck [Hz]}}{16 \times \text{A [baud]}} - 1$
001	$\text{UART0DR} = \frac{\text{fcgck [Hz]}}{16.5 \times \text{A [baud]}} - 1$
010	$\text{UART0DR} = \frac{\text{fcgck [Hz]}}{15 \times \text{A [baud]}} - 1$
011	$\text{UART0DR} = \frac{\text{fcgck [Hz]}}{15.5 \times \text{A [baud]}} - 1$
100	$\text{UART0DR} = \frac{\text{fcgck [Hz]}}{17 \times \text{A [baud]}} - 1$

Table 16-10 Example of UART0DR Calculation (when fcgck = 4 [MHz])

<RTSEL>	UART0DR calculation	Generated baud rate
000	$\text{UART0DR} = \frac{4000000\text{ [Hz]}}{16 \times 38400\text{ [baud]}} - 1 \approx 6$	$\frac{4000000\text{ [Hz]}}{16 \times (6+1)} = 35714\text{ [baud] } (-6.99\%)$
001	$\text{UART0DR} = \frac{4000000\text{ [Hz]}}{16.5 \times 38400\text{ [baud]}} - 1 \approx 5$	$\frac{4000000\text{ [Hz]}}{16.5 \times (5+1)} = 40404\text{ [baud] } (+5.22\%)$
010	$\text{UART0DR} = \frac{4000000\text{ [Hz]}}{15 \times 38400\text{ [baud]}} - 1 \approx 6$	$\frac{4000000\text{ [Hz]}}{15 \times (6+1)} = 38095\text{ [baud] } (-0.79\%)$
011	$\text{UART0DR} = \frac{4000000\text{ [Hz]}}{15.5 \times 38400\text{ [baud]}} - 1 \approx 6$	$\frac{4000000\text{ [Hz]}}{15.5 \times (6+1)} = 36866\text{ [baud] } (-3.99\%)$
100	$\text{UART0DR} = \frac{4000000\text{ [Hz]}}{17 \times 38400\text{ [baud]}} - 1 \approx 5$	$\frac{4000000\text{ [Hz]}}{17 \times (5+1)} = 39216\text{ [baud] } (+2.12\%)$

16.10. Data Sampling Method

The UART receive control circuit starts RT clock counting when it detects a falling edge of the input pulses to the RXD0 pin. 15 to 17 RT clocks are counted per bit and each clock is expressed as RTn (n = 16 to 0). In a bit that has 17 RT clocks, RT16 to RT0 are counted. In a bit that has 16 RT clocks, RT15 to RT0 are counted. In a bit that has 15 RT clocks, RT14 to RT0 are counted (Decrement). During counting of RT8 to RT6, the UART receive control circuit samples the input pulses to the RXD0 pin to make a majority decision. The same level detected twice or more from among three samplings is processed as the data for the bit.

The number of RT clocks can be changed in a range of 15 to 17 by setting UART0CR2<RTSEL>. However, sampling is always executed in RT8 to RT6, when the number of RT clocks is changed (Figure 16-5).

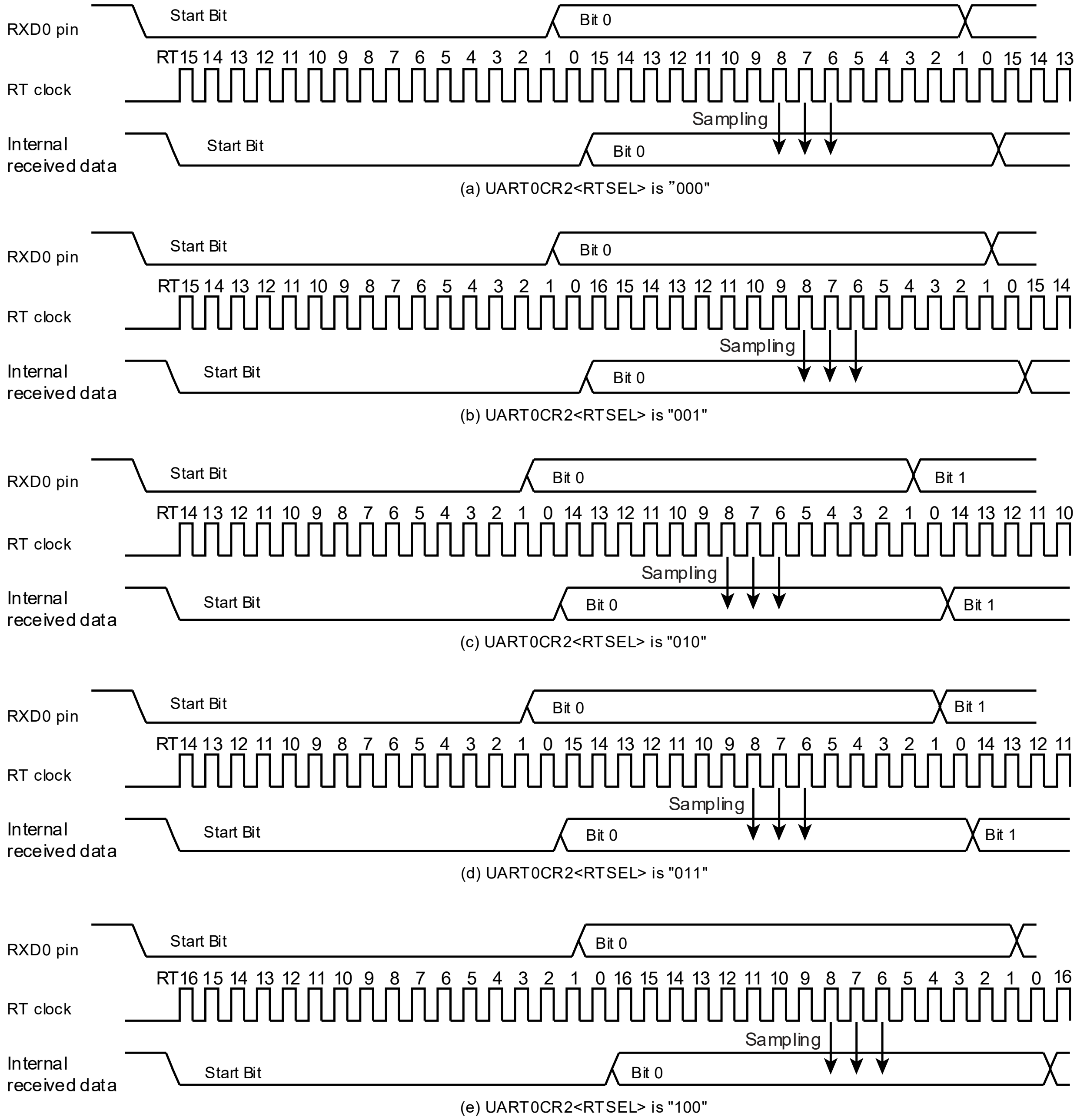

RXD0 pin
Start Bit
Bit 0
RT clock
RT 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 15 14 13
Sampling
Internal received data
Start Bit
Bit 0
(a) UART0CR2<RTSEL> is ”000"
RXD0 pin
Start Bit
Bit 0
RT clock
RT 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 15 14
Sampling
Internal received data
Start Bit
Bit 0
(b) UART0CR2<RTSEL> is "001"
RXD0 pin
Start Bit
Bit 0
Bit 1
RT clock
RT 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 14 13 12 11 10
Sampling
Internal received data
Start Bit
Bit 0
Bit 1
(c) UART0CR2<RTSEL> is "010"
RXD0 pin
Start Bit
Bit 0
Bit 1
RT clock
RT 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 14 13 12 11
Sampling
Internal received data
Start Bit
Bit 0
Bit 1
(d) UART0CR2<RTSEL> is "011"
RXD0 pin
Start Bit
Bit 0
RT clock
RT 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 16
Sampling
Internal received data
Start Bit
Bit 0
(e) UART0CR2<RTSEL> is "100"

Figure 16-5 Data Sampling in Each Case of UART0CR2<RTSEL>

The sampling (majority decision) is processed with Data bit, Parity bit, and STOP bit.

The sampling (majority decision) is not processed with Start bit. When the falling edge (“Low” level) is detected as the RXD0 pin input pulse, the RT clock continues to count, and starts the receive operation, regardless of the RXD0 pin state.

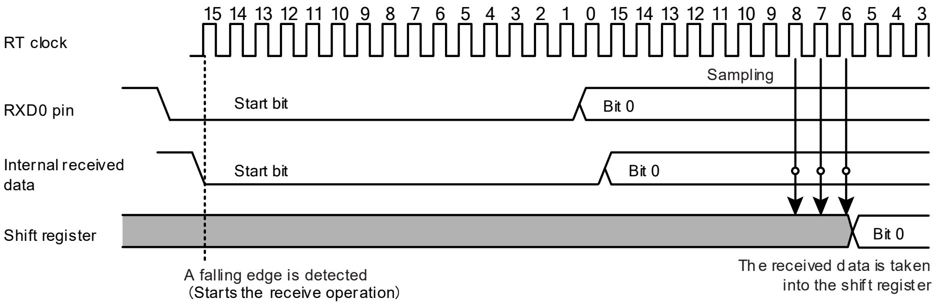


Figure 16-6 Start Bit Sampling

The noise rejection of the receive data should be performed so that the incorrect receive operation does not start by causes like noises. For details, refer to chapter "16.11. Received Data Noise Rejection". Additionally, for the method in case of receiving incorrect data, the process by software should be performed.

16.11. Received Data Noise Rejection

When noise rejection is enabled at UART0CR2<RXDNC>, the time of pulses to be regarded as signals is as shown in Table 16-11.

Table 16-11 Received Data Noise Rejection Time

<RXDNC>	Noise rejection time [s]	Time of pulses to be regarded as signals [s]
00	No noise rejection	-
01	(UART0DR+1) / (Transfer base clock frequency)	2 × (UART0DR+1) / (Transfer base clock frequency)
10	2 × (UART0DR+1) / (Transfer base clock frequency)	4 × (UART0DR+1) / (Transfer base clock frequency)
11	4 × (UART0DR+1) / (Transfer base clock frequency)	8 × (UART0DR+1) / (Transfer base clock frequency)

Note: The transfer base clock frequency is the clock frequency selected at UART0CR1<BRG>.

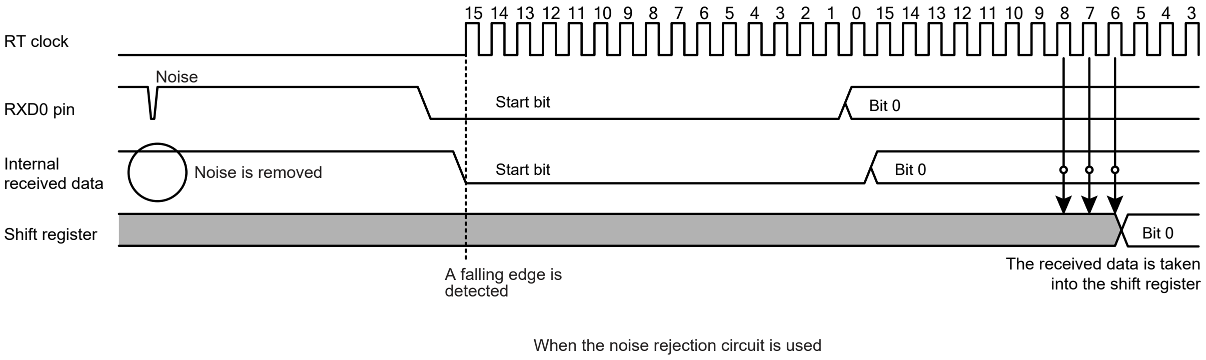


Figure 16-7 Received Data Noise Rejection

16.12. Transmit/Receive Operation

16.12.1. Data transmit operation

Set UART0CR1<TXE> to "1". Check UART0SR<TBFL> = 0, and then write data into TD0BUF (transmit data buffer). Writing data into TD0BUF sets UART0SR<TBFL> to "1", transfers the data to the transmit shift register, and outputs the data sequentially from the TXD0 pin. The data output includes a start bit, stop bits whose number is specified in UART0CR1<STOPBT> and a parity bit when additional parity is specified. Select the data transfer baud rate using UART0CR1<BRG>, UART0CR2<RTSEL> and UART0DR. When data transmission starts, the transmit buffer full flag UART0SR<TBFL> is cleared to "0" and the INTTXD0 interrupt request is generated.

Note 1: After data is written into TD0BUF, when new data is written into TD0BUF before the previous data is transferred to the shift register, the new data is written over the previous data and is transferred to the shift register.

Note 2: Under the conditions shown in Table 16-12, the TXD0 pin output is fixed at the “Low” or “High” level according to the setting of UART0CR1<IRDASEL>.

Table 16-12 TXD0 Pin Output

Condition	TXD0 pin output	
	UART0CR1 <IRDASEL> = 0	UART0CR1 <IRDASEL> = 1
When UART0CR1<TXE> is "0"	“High” level	“Low” level
Time from when "1" is written to UART0CR1<TXE> to when the transmitted data is written to TD0BUF		
When the STOP, IDLE0 or SLEEP0 mode is active		

16.12.2. Data receive operation

Set UART0CR1<RXE> to "1". When data is received via the RXD0 pin, the received data is stored to receive shift register. At this time, the transmitted data includes a start bit, stop bit(s) and a parity bit when additional parity is specified. When the stop bit(s) are received, data only is extracted and transferred to RD0BUF (receive data buffer). Then the receive buffer full flag UART0SR<RBFL> is set to “1” and the INTRXD0 interrupt request is generated.

Set the data transfer baud rate using UART0CR1<BRG>, UART0CR2<RTSEL> and UART0DR.

When an overrun error occurs at receiving of the last data, the last data is not transferred to RD0BUF (receive data buffer) but discarded; the data in the RD0BUF is not affected.

16.13. Status Flag

16.13.1. Parity error

When the parity determined using the received data bits differs from the received parity bit, the parity error flag UART0SR<PERR> is set to "1". At this time, the INTRXD0 interrupt request is generated.

When UART0SR<PERR> is "1" at reading from UART0SR, UART0SR<PERR> will be cleared to "0" when RD0BUF is read subsequently. (The RD0BUF read value becomes undefined.)

When UART0SR<PERR> is set to "1" after UART0SR is read, UART0SR<PERR> will not be cleared to "0" when RD0BUF is read subsequently. In this case, UART0SR<PERR> will be cleared to "0" when UART0SR is read again and RD0BUF is read.

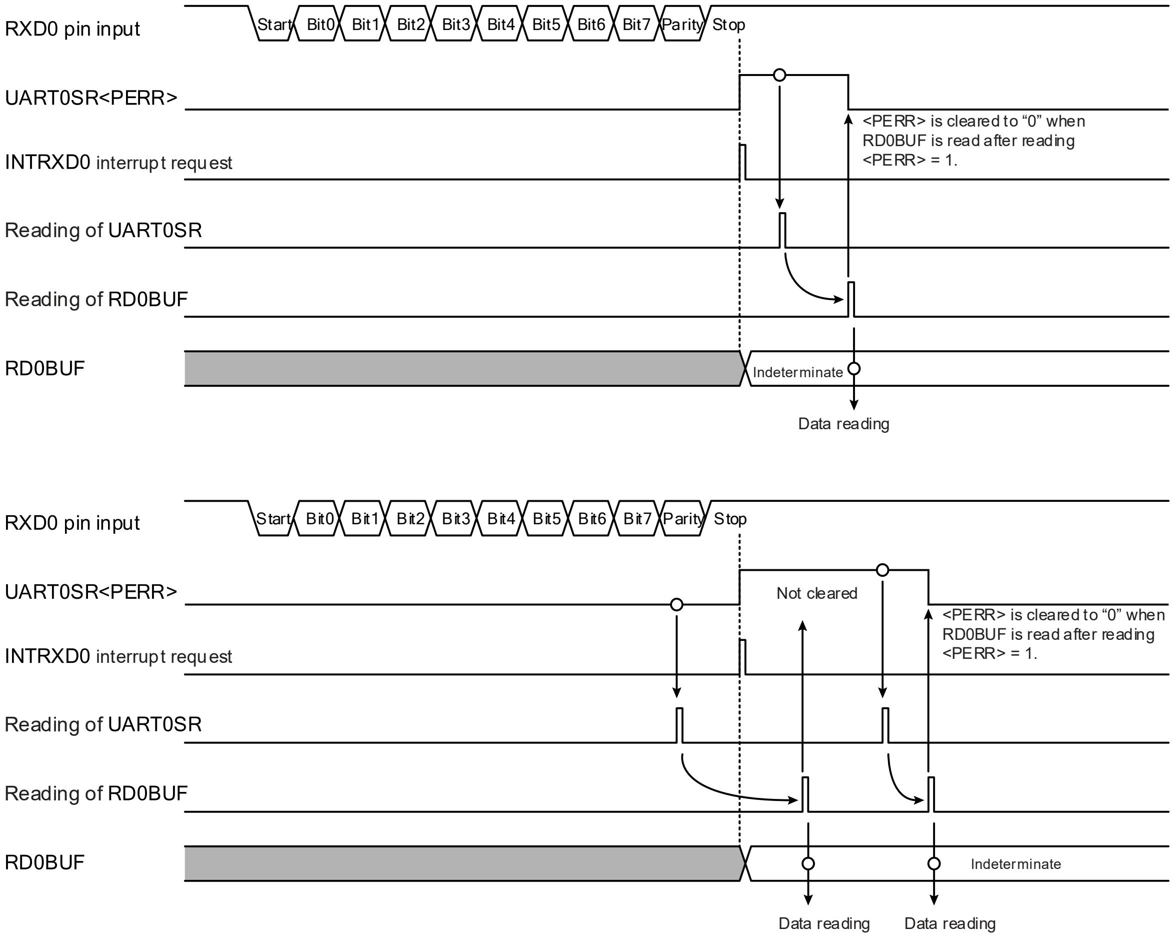


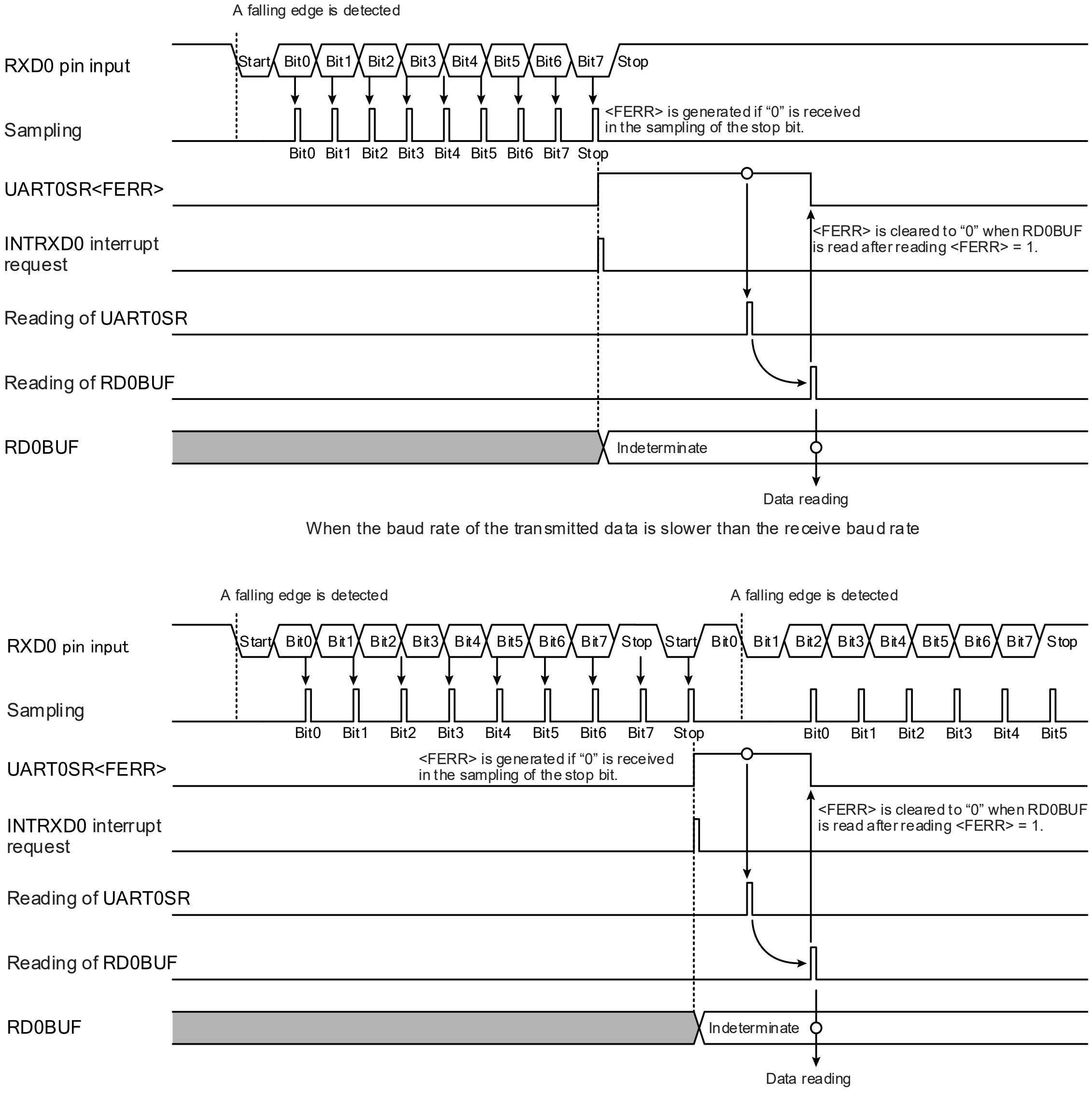
Figure 16-8 Occurrence of Parity Error

16.13.2. Framing Error

When the receive baud rate with the transmitted data's baud rate to RXD0 pin is differ or "0" is sampled as the stop bit of received data due to the influence of noise on the RXD0 pin, the framing error flag UART0SR<FERR> is set to "1". At this time, the INTRXD0 interrupt request is generated.

When UART0SR<FERR> is "1" at reading from UART0SR, UART0SR<FERR> will be cleared to "0" when RD0BUF is read subsequently.

When UART0SR<FERR> is set to "1" after UART0SR is read, UART0SR<FERR> will not be cleared to "0" when RD0BUF is read subsequently. In this case, UART0SR<FERR> will be cleared to "0" when UART0SR is read again and RD0BUF is read.


A falling edge is detected
RXD0 pin input
Start Bit0 Bit1 Bit2 Bit3 Bit4 Bit5 Bit6 Bit7 Stop
Sampling
<FERR> is generated if "0" is received in the sampling of the stop bit.
Bit0 Bit1 Bit2 Bit3 Bit4 Bit5 Bit6 Bit7 Stop
UART0SR<FERR>
INTRXD0 interrupt request
<FERR> is cleared to "0" when RD0BUF is read after reading <FERR> = 1.
Reading of UART0SR
Reading of RD0BUF
RD0BUF
Indeterminate
Data reading
A falling edge is detected
RXD0 pin input
Start Bit0 Bit1 Bit2 Bit3 Bit4 Bit5 Bit6 Bit7 Stop Start Bit0 Bit1 Bit2 Bit3 Bit4 Bit5 Bit6 Bit7 Stop
Sampling
Bit0 Bit1 Bit2 Bit3 Bit4 Bit5 Bit6 Bit7 Stop Bit0 Bit1 Bit2 Bit3 Bit4 Bit5
UART0SR<FERR>
<FERR> is generated if "0" is received in the sampling of the stop bit.
INTRXD0 interrupt request
<FERR> is cleared to "0" when RD0BUF is read after reading <FERR> = 1.
Reading of UART0SR
Reading of RD0BUF
RD0BUF
Indeterminate
Data reading

When the baud rate of the transmitted data is slower than the receive baud rate

When the baud rate of the transmitted data is faster than the receive baud rate

Figure 16-9 Occurrence of Framing Error

16.13.3. Overrun error

When receiving of all data bits is completed before the previous received data is read from RD0BUF, the overrun error flag UART0SR<OERR> is set to "1" and the INTRXD0 interrupt request is generated. The data received at the occurrence of the overrun error is discarded and the previous received data is maintained. Subsequently, when data is received while UART0SR<OERR> is still "1", no INTRXD0 interrupt request is generated, and the received data is discarded. (Figure 16-10)

Note that parity or framing errors in the discarded received data cannot be detected. (These error flags are not set.) That is to say, when these errors are detected together with an overrun error during the reading of UART0SR, they have occurred in the previous received data (the data stored in RD0BUF). (Figure 16-11)

When UART0SR<OERR> is "1" at reading from UART0SR, UART0SR<OERR> will be cleared to "0" when RD0BUF is read subsequently. (Figure 16-12)

When UART0SR<OERR> is set to "1" after UART0SR is read, UART0SR<OERR> will not be cleared to "0" when RD0BUF is read subsequently. In this case, UART0SR<OERR> will be cleared to "0" when UART0SR is read again and RD0BUF is read. (Figure 16-12)

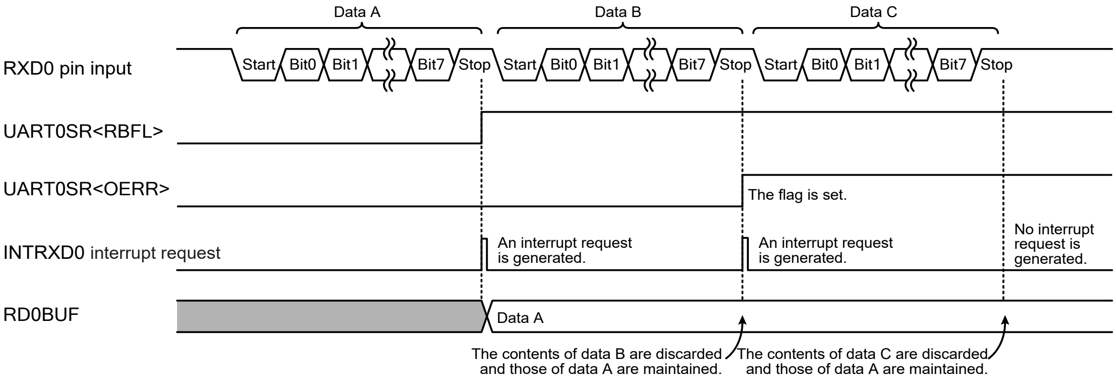
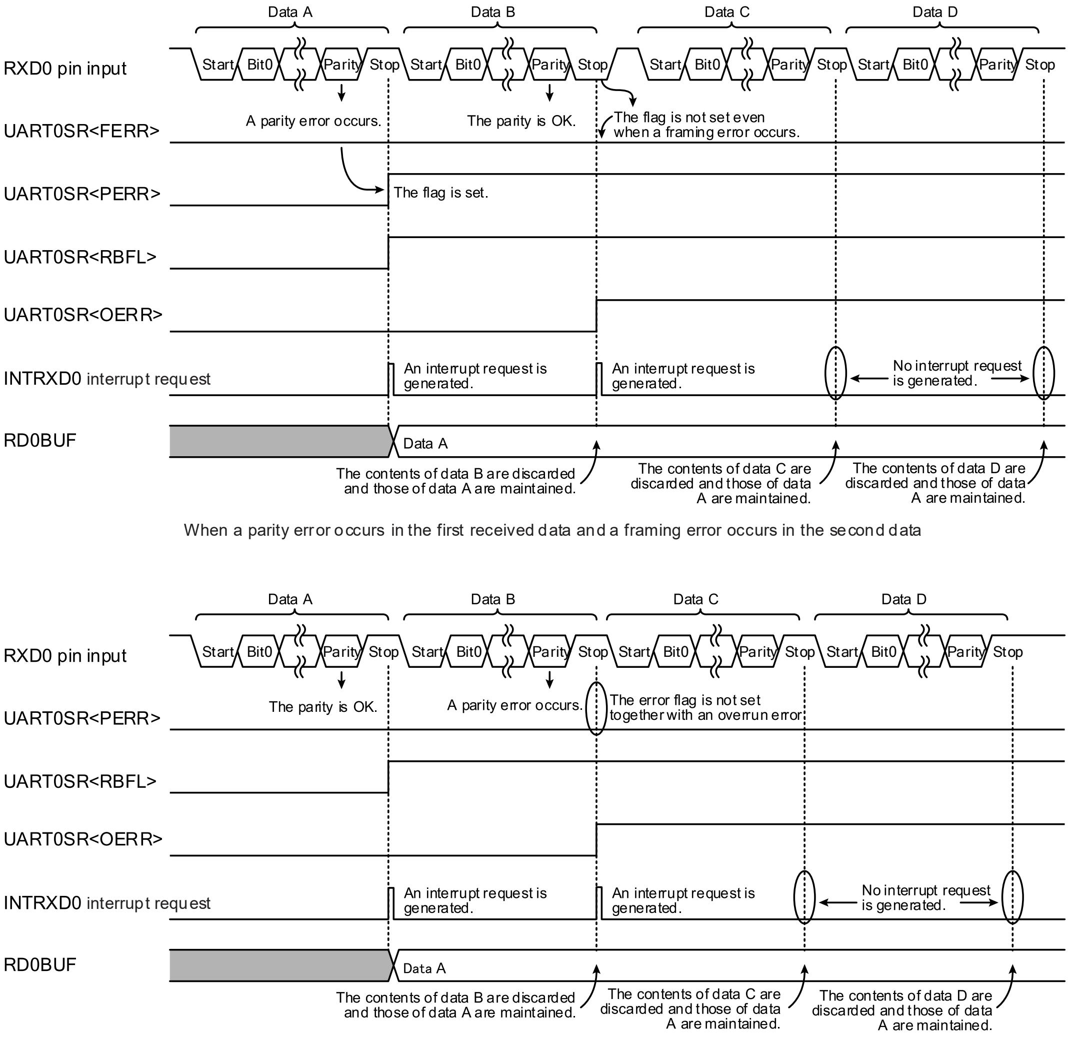

Data A
Data B
Data C
RXD0 pin input
Start
Bit0
Bit1
Bit7
Stop
UART0SR<RBFL>
UART0SR<OERR>
The flag is set.
INTRXD0 interrupt request
An interrupt request is generated.
An interrupt request is generated.
No interrupt request is generated.
RD0BUF
Data A
The contents of data B are discarded and those of data A are maintained.
The contents of data C are discarded and those of data A are maintained.

Figure 16-10 Occurrence of Overrun Error


Data A
Data B
Data C
Data D
RXD0 pin input
Start
Bit0
Parity
Stop
UART0SR<FERR>
A parity error occurs.
The parity is OK.
The flag is not set even when a framing error occurs.
UART0SR<PERR>
The flag is set.
UART0SR<RBFL>
UART0SR<OERR>
INTRXD0 interrupt request
An interrupt request is generated.
An interrupt request is generated.
No interrupt request is generated.
RD0BUF
Data A
The contents of data B are discarded and those of data A are maintained.
The contents of data C are discarded and those of data A are maintained.
The contents of data D are discarded and those of data A are maintained.
Data A
Data B
Data C
Data D
RXD0 pin input
Start
Bit0
Parity
Stop
UART0SR<PERR>
The parity is OK.
A parity error occurs.
The error flag is not set together with an overrun error.
UART0SR<RBFL>
UART0SR<OERR>
INTRXD0 interrupt request
An interrupt request is generated.
An interrupt request is generated.
No interrupt request is generated.
RD0BUF
Data A
The contents of data B are discarded and those of data A are maintained.
The contents of data C are discarded and those of data A are maintained.
The contents of data D are discarded and those of data A are maintained.

When a parity error occurs in the first received data and a framing error occurs in the second data

When a parity error occurs in the second received data

Figure 16-11 Framing/Parity Error Flags When an Overrun Error Occurs

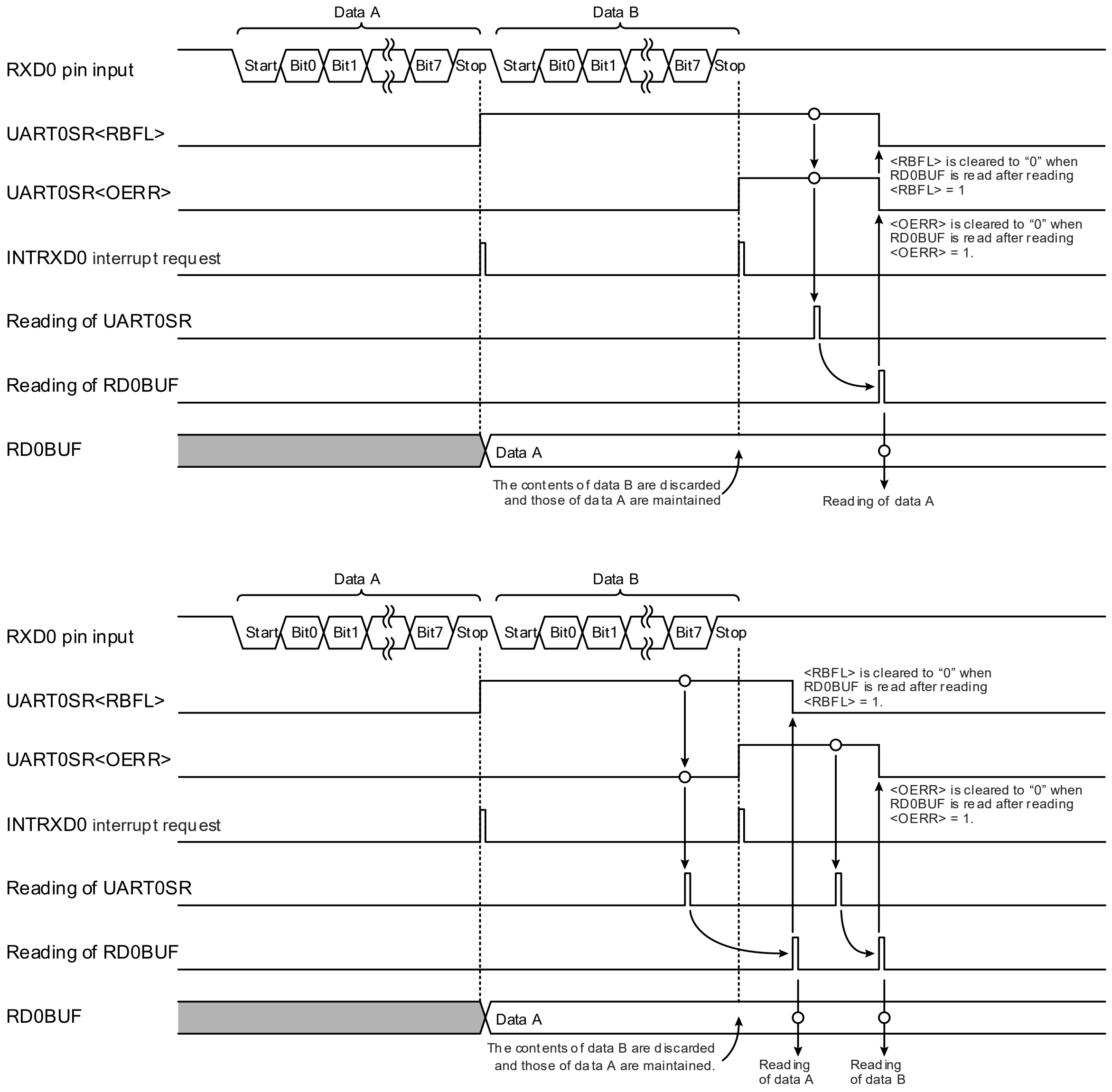

Data A
Data B
RXD0 pin input
Start
Bit0
Bit1
Bit7
Stop
UART0SR<RBFL>
UART0SR<OERR>
INTRXD0 interrupt request
Reading of UART0SR
Reading of RD0BUF
RD0BUF
Data A
<RBFL> is cleared to "0" when RD0BUF is read after reading <RBFL> = 1
<OERR> is cleared to "0" when RD0BUF is read after reading <OERR> = 1.
The contents of data B are discarded and those of data A are maintained
Reading of data A
<RBFL> is cleared to "0" when RD0BUF is read after reading <RBFL> = 1.
Reading of data A
Reading of data B

Figure 16-12　Clearance of Overrun Error Flag

16.13.4. Receive Data Buffer Full

Loading the received data in RD0BUF sets UART0SR<RBFL> to "1".

When UART0SR<RBFL> is "1" at reading from UART0SR, UART0SR<RBFL> will be cleared to "0" when RD0BUF is read subsequently.

When UART0SR<RBFL> is set to "1" after UART0SR is read, UART0SR<RBFL> will not be cleared to "0" when RD0BUF is read subsequently. In this case, UART0SR<RBFL> will be cleared to "0" when UART0SR is read again and RD0BUF is read.

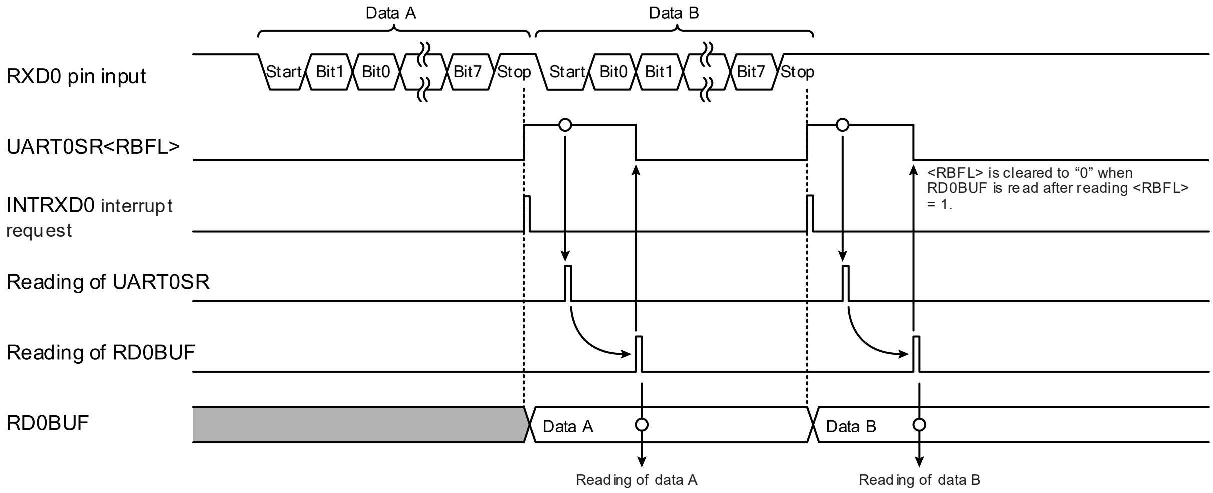


Figure 16-13 Occurrence of Receive Data Buffer Full

16.13.5. Transmit busy flag

When transmission is completed with no data in TD0BUF (when UART0SR<TBFL> = 0), UART0SR<TBSY> is cleared to "0". When transmission is restarted after data is written into TD0BUF, UART0SR<TBSY> is set to "1". At this time, an INTTXD0 interrupt request is generated.

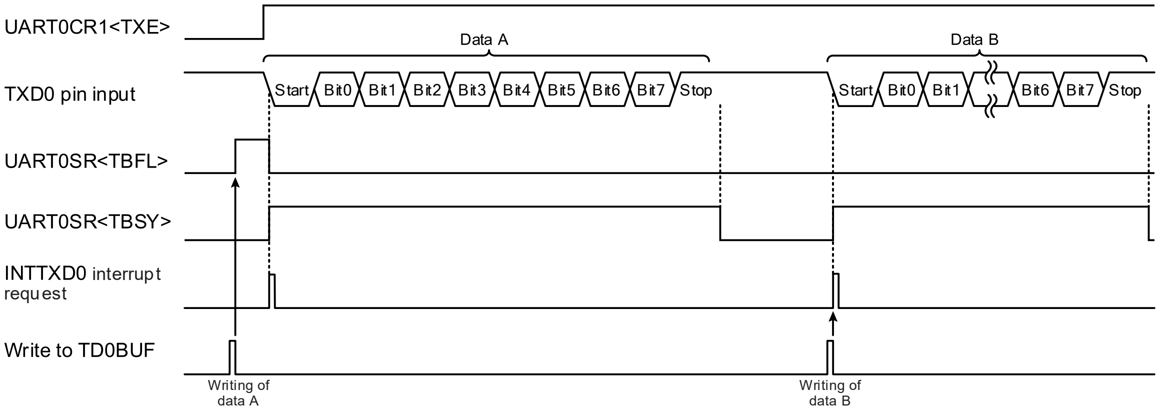


Figure 16-14 Transmit Busy Flag and Occurrence of Transmit Buffer Full

16.13.6. Transmit Buffer Full

When TD0BUF has no data, or when data in TD0BUF is transferred to the transmit shift register and transmission is started, UART0SR<TBFL> is cleared to "0". At this time, the INTTXD0 interrupt request is generated.

Writing data into TD0BUF sets UART0SR<TBFL> to "1".

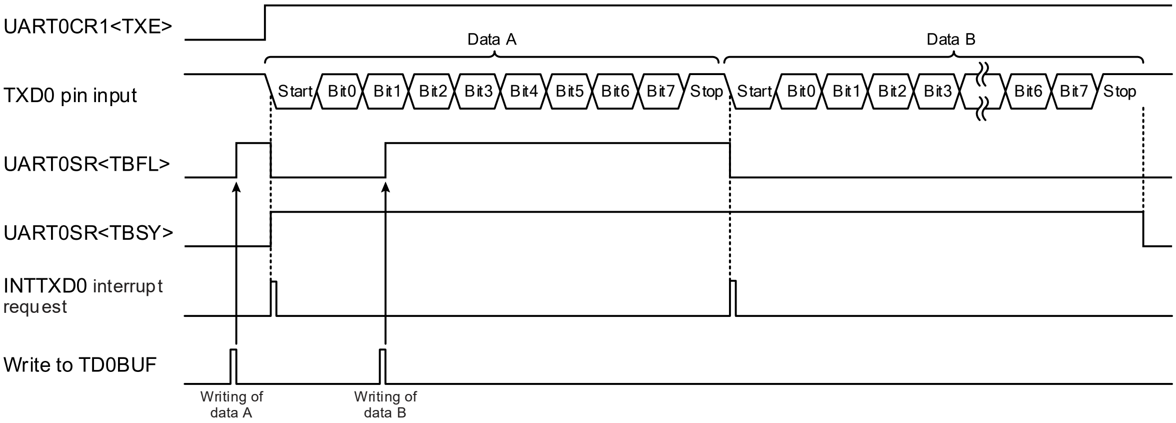


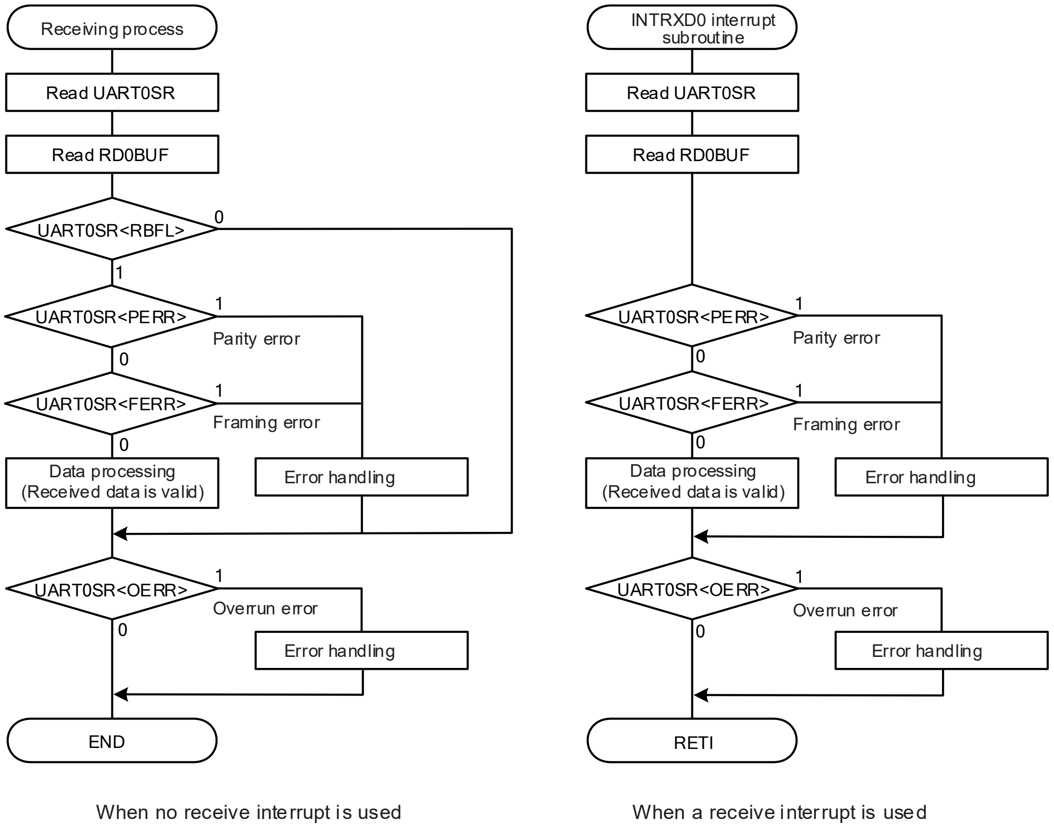
Figure 16-15 Occurrence of Transmit Buffer Full

16.14. Receiving Process

Figure 16-16 shows an example of the receiving process. Details of flag judgments in the processing are shown in Table 16-13 and Table 16-14.

When framing error or parity error is detected, the received data has erroneous value(s). Execute the error handling, for example, the discarding the received data read from RD0BUF and receiving the data again.

When overrun error is detected, the receiving of one or more numbers of data is unfinished. It is impossible to determine the number of data that could not be received. Execute the error handling, for example, by receiving data again from the beginning of the transfer. Generally, an overrun error occurs when the internal software processing cannot follow the data transfer speed. It is recommended to slow the transfer baud rate or modify the software to execute flow control.



Note: When multiple interrupts are used in the INTRXD0 interrupt subroutine, the interrupt should be enabled after reading UART0SR and RD0BUF.

Figure 16-16 Example of Receiving Process

Table 16-13   Flag Judgments When No Receive Interrupt Is Used

<RBFL>	<FERR>/<PERR>	<OERR>	State
0	*	0	Data has not been received yet.
0	*	1	Receiving of next data is completed in the period from when UART0SR is read to when RD0BUF is read.
1	0	0	Receiving has been completed properly.
1	0	1	Receiving has been completed properly, but some of data could not be received.
1	1	0	Received data has erroneous value(s).
1	1	1	Received data has erroneous value(s) and some of data could not be received.

Note: *: Don’t care

Table 16-14   Flag Judgments When a Receive Interrupt Is Used

<FERR>/<PERR>	<OERR>	State
0	0	Receiving has been completed properly.
0	1	Receiving has been completed properly, but some of data could not be received.
1	0	Received data has erroneous value(s).
1	1	Received data has erroneous value(s) and some of data could not be received.

16.15. AC Characteristics

16.15.1. IrDA Characteristics

(V_{SS} = 0 [V], V_{DD} = 4.5 to 5.5 [V], Topr = -40 to 85 [°C])

Parameter	Condition	Min	Typ.	Max	Unit
TXD0 output pulse time (RT clock × (3 / 16))	Transfer baud rate = 2400 [baud]	-	78.13	-	μs
	Transfer baud rate = 9600 [baud]	-	19.53	-	
	Transfer baud rate = 19200 [baud]	-	9.77	-	
	Transfer baud rate = 38400 [baud]	-	4.88	-	
	Transfer baud rate = 57600 [baud]	-	3.26	-	
	Transfer baud rate = 115200 [baud]	-	1.63	-	

17. Synchronous Serial Interface (SIO)

The TMP89FS60B/62B/63B contains the high-speed 8-bit synchronous serial interfaces (SIO). This chapter describes the SIO0. For the SIO1, replace the SFR addresses and pin names as shown in Table 17-1 and Table 17-2.

Table 17-1 SFR Address Assignment

	SIOxCR (address)	SIOxSR (address)	SIOxBUF (address)	Low power consumption register
SIO0	SIO0CR (0x001F)	SIO0SR (0x0020)	SIO0BUF (0x0021)	POFFCR2 <SIO0EN>
SIO1	SIO1CR (0x0F70)	SIO1SR (0x0F71)	SIO1BUF (0x0F72)	POFFCR2 <SIO1EN>

Table 17-2 Pin Names

	Serial clock input/output pin	Serial data input pin	Serial data output pin
SIO0	SCLK0 pin	SI0 pin	SO0 pin
SIO1	SCLK1 pin	SI1 pin	SO1 pin

17.1. Synchronous Serial Interfaces (SIO) for Each Product

The synchronous serial interfaces (SIO) for each product is shown in Table 17-3.

In regards to unavailable channel of the SIO, the corresponding bit of the low power register must be cleared to “0”.

Table 17-3 Synchronous Serial Interface (SIO) for Each Product

	TMP89FS60B	TMP89FS62B	TMP89FS63B
SIO0	A	A	A
SIO1	A	A	A

Note: A: Available

17.2. Configuration

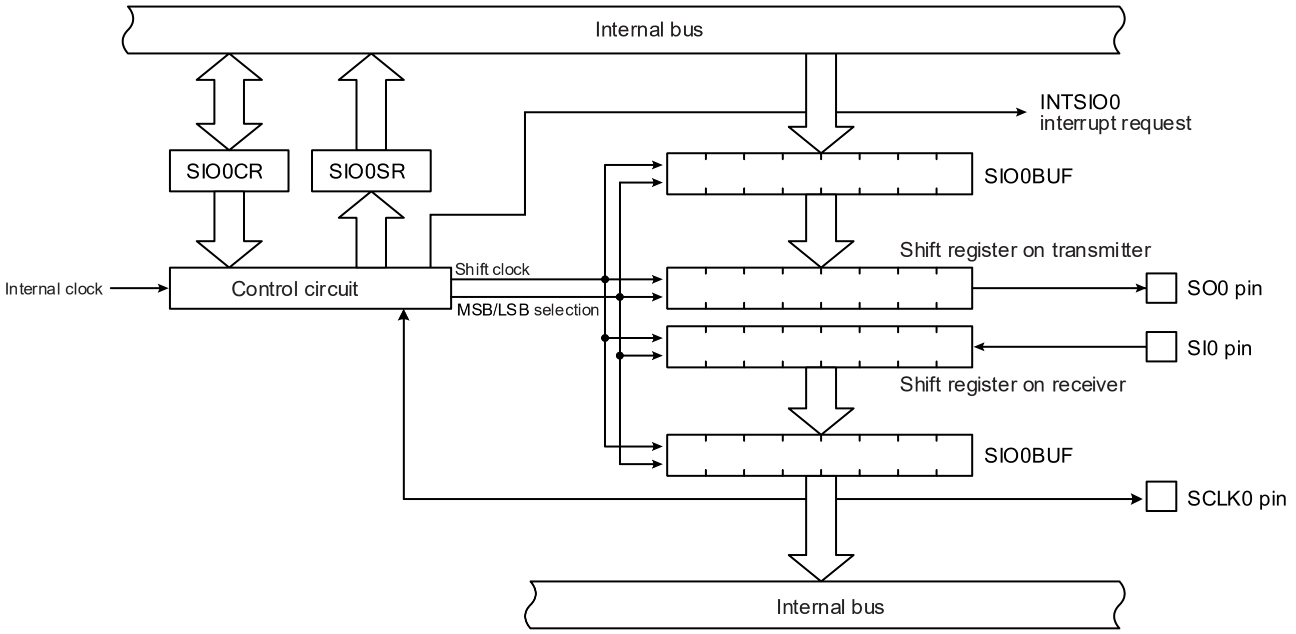


Figure 17-1 Serial Interface

Note: The SIO input/output pins are also used as the I/O ports. The I/O port register settings are required to use these pins for the SIO. For details, refer to "8. I/O Ports".

17.3. Control

The SIO0 is controlled by the low power consumption registers (POFFCR2), the serial interface data buffer register (SIO0BUF), the serial interface control register (SIO0CR) and the serial interface status register (SIO0SR).

Low power consumption register 2

POFFCR2 (0x0F76)

	7	6	5	4	3	2	1	0
Bit Symbol	-	-	(RTCEN)	-	-	-	SIO1EN	SIO0EN
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

SIO1EN	SIO1 control	0: 1:	Disable Enable
SIO0EN	SIO0 control	0: 1:	Disable Enable

Note: The written values are read from the bits 7 and 6, 4 to 2 of POFFCR2 . These bits must be cleared to “0”.

Serial interface buffer register

SIO0BUF (0x0021)

	7	6	5	4	3	2	1	0
Bit Symbol	SIO0BUF							
Read/Write	R							
After reset	0	0	0	0	0	0	0	0

Serial interface buffer register

SIO0BUF (0x0021)

	7	6	5	4	3	2	1	0
Bit Symbol	SIO0BUF							
Read/Write	W							
After reset	1	1	1	1	1	1	1	1

Note: SIO0BUF is the data buffer for both transmission and reception. The latest received data is read when SIO0BUF is read every time. When SIO0BUF has never received data, it is read as "0". When data is written into it, the data is treated as the transmit data.

Serial interface control register

SIO0CR		7	6	5	4	3	2	1	0
(0x001F)	Bit Symbol	SIOEDG	SIOCKS			SIODIR	SIOS	SIOM	
	Read/Write	R/W	R/W			R/W	R/W	R/W	
	After reset	0	0	0	0	0	0	0	0

SIOEDG	Transfer edge selection	0:	Receive data at a rising edge and transmit data at a falling edge	
		1:	Transmit data at a rising edge and receive data at a falling edge	
SIOCKS	Serial clock selection [Hz]		NORMAL1/2 or IDLE1/2 mode	SLOW1/2 or SLEEP1 mode
		000:	fcgck / 2^9	-
		001:	fcgck / 2^6	-
		010:	fcgck / 2^5	-
		011:	fcgck / 2^4	-
		100:	fcgck / 2^3	-
		101:	fcgck / 2^2	-
		110:	fcgck / 2	fs / 2^3
		111:	External clock input	
SIODIR	Transfer format (MSB/LSB) selection	0:	LSB first (transfer from bit 0)	
		1:	MSB first (transfer from bit 7)	
SIOS	Transfer operation start/stop instruction	0:	Operation stop (reserved stop)	
		1:	Operation start	
SIOM	Transfer mode selection and operation	00:	Operation stop (forced stop)	
		01:	8-bit transmit mode	
		10:	8-bit receive mode	
		11:	8-bit transmit and receive mode	

Note 1: fcgck: Gear clock [Hz], fs: Low-frequency clock [Hz]

Note 2: After the operation is started by writing "1" to SIO0CR<SIOS>, writing to <SIOEDG>, <SIOCKS> and <SIODIR> is invalid until SIO0SR<SIOF> becomes "0". (<SIOEDG>, <SIOCKS> and <SIODIR> can be changed at the same time as changing <SIOS> from "0" to "1".)

Note 3: After the operation is started by writing "1" to <SIOS>, no values other than"00" can be written to <SIOM> until <SIOF> becomes "0" (when a value from "01" to "11" is written to <SIOM>, it is ignored). The transfer mode cannot be changed during the operation.

Note 4: <SIOS> remains at "0" by writing "1" to <SIOS> when <SIOM> is "00" (operation stop).

Note 5: When SIO0 is used in SLOW1/2 or SLEEP1 mode, be sure to set <SIOCKS> to "110". When <SIOCKS> is set to any other value, SIO0 does not operate.

Note 6: When STOP, IDLE0 or SLEEP0 mode is activated, <SIOM> is automatically cleared to "00" and SIO0 stops the operation. At the same time, <SIOS> is cleared to "0". However, the values set for <SIOEDG>, <SIOCKS> and <SIODIR> are maintained.

Serial interface status register

SIO0SR (0x0020)		7	6	5	4	3	2	1	0
	Bit Symbol	SIOF	SEF	OERR	REND	UERR	TBFL	-	-
	Read/Write	R	R	R	R	R	R	R	R
	After reset	0	0	0	0	0	0	0	0

SIOF	Serial transfer operation status monitor	0: 1:	Transfer not in progress Transfer in progress
SEF	Shift operation status monitor	0: 1:	Shift operation not in progress Shift operation in progress
OERR	Receive overrun error flag	0: 1:	No overrun error has occurred At least one overrun error has occurred
REND	Receive completion flag	0: 1:	No data has been received since the last receive data was read At least one data receive operation has been executed
UERR	Transmit underrun error flag	0: 1:	No transmit underrun error has occurred At least one transmit underrun error has occurred
TBFL	Transmit buffer full flag	0: 1:	The transmit buffer is empty The transmit buffer has the data that has not yet been transmitted

Note 1: <OERR> and <UERR> are cleared by reading SIO0SR.

Note 2: <REND> is cleared by reading SIO0BUF.

Note 3: Writing "00" to SIO0CR<SIOM> clears all the bits of SIO0SR to "0", whether the SIO0 is operating or not. When STOP, IDLE0 or SLEEP0 mode is activated, <SIOM> is automatically cleared to "00" and all the bits of SIO0SR are cleared to "0".

Note 4: When a read instruction is executed on SIO0SR, bits 1 and 0 are read as "0".

17.4. Low Power Consumption Control

The SIO0 has the low power consumption registers (POFFCR2) that save power when the SIO0 is not being used.

Setting POFFCR2<SIO0EN> to "0" disables the basic clock supply to the SIO0 to save power. Note that this makes the SIO0 unusable. Setting POFFCR2<SIO0EN> to "1" enables the basic clock supply to the SIO0 and enables the SIO0 to operate.

After reset, POFFCR2<SIO0EN> are initialized to "0", and this makes the SIO0 unusable. When using the SIO0 for the first time, be sure to set POFFCR2<SIO0EN> to "1" in the initial setting of the program (before the SIO0 control registers are modified).

Do not change POFFCR2<SIO0EN> to "0" while the SIO0 is operated. Otherwise, the SIO0 may operate unexpectedly.

17.5. Functions

17.5.1. Transfer format

The transfer format can be set to either MSB or LSB first by using SIO0CR<SIODIR>. Setting SIO0CR<SIODIR> to "0" selects LSB first as the transfer format. In this case, the serial data is transferred in sequence from the least significant bit.

Setting SIO0CR<SIODIR> to "1" selects MSB first as the transfer format. In this case, the serial data is transferred in sequence from the most significant bit.

17.5.2. Serial clock

The serial clock can be selected by using SIO0CR<SIOCKS>.

Setting SIO0CR<SIOCKS> to "000" to "110" selects the internal clock as the serial clock. In this case, the serial clock is output from the SCLK0 pin. The serial data is transferred in synchronization with the edge of the SCLK0 pin output.

Setting SIO0CR<SIOCKS> to "111" selects an external clock as the serial clock. In this case, an external serial clock must be input to the SCLK0 pin. The serial data is transferred in synchronization with the edge of the external clock.

The serial data transfer edge can be selected for both the external and internal clocks. For details, refer to "17.5.3. Transfer edge selection".

Table 17-4 Transfer Speed

SIO0CR <SIOCKS>	Serial clock [Hz]		fcgck = 4 [MHz]		fcgck = 8 [MHz]		fcgck = 10 [MHz]		fs = 32.768 [kHz]	
	NORMAL1/2 or IDLE1/2 mode	SLOW1/2 or SLEEP1 mode	1-bit time	Transfer speed	1-bit time	Transfer speed	1-bit time	Transfer speed	1-bit time	Transfer speed
000	fcgck / 2^9	-	128 [μs]	7.813 [kbps]	64 [μs]	15.625 [kbps]	51.2 [μs]	19.531 [kbps]	-	-
001	fcgck / 2^6	-	16 [μs]	62.5 [kbps]	8 [μs]	125 [kbps]	6.4 [μs]	156.25 [kbps]	-	-
010	fcgck / 2^5	-	8 [μs]	125 [kbps]	4 [μs]	250 [kbps]	3.2 [μs]	312.5 [kbps]	-	-
011	fcgck / 2^4	-	4 [μs]	250 [kbps]	2 [μs]	500 [kbps]	1.6 [μs]	625 [kbps]	-	-
100	fcgck / 2^3	-	2 [μs]	500 [kbps]	1 [μs]	1 [Mbps]	0.8 [μs]	1.25 [Mbps]	-	-
101	fcgck / 2^2	-	1 [μs]	1 [Mbps]	0.5 [μs]	2 [Mbps]	0.4 [μs]	2.5 [Mbps]	-	-
110	fcgck / 2	fs / 2^3	0.5 [μs]	2 [Mbps]	0.25 [μs]	4 [Mbps]	0.2 [μs]	5 [Mbps]	244 [μs]	4 [kbps]

17.5.3. Transfer edge selection

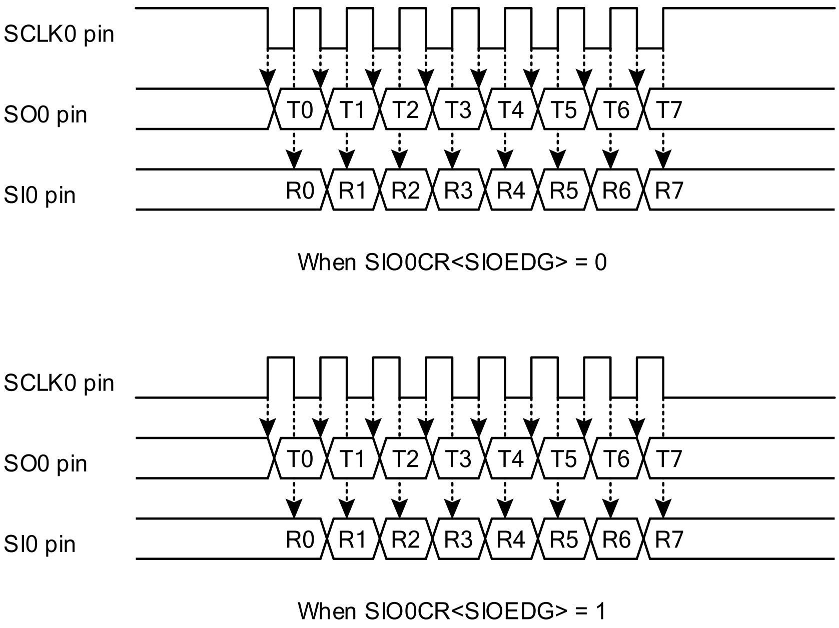
The serial data transfer edge can be selected by using SIO0CR<SIOEDG>.

Table 17-5 Transfer Edge Selection

SIO0CR<SIOEDG>	Data transmission	Data reception
0	Falling edge	Rising edge
1	Rising edge	Falling edge

When SIO0CR<SIOEDG> is "0", the data is transmitted in synchronization with the falling edge of the clock and the data is received in synchronization with the rising edge of the clock.

When SIO0CR<SIOEDG> is "1", the data is transmitted in synchronization with the rising edge of the clock and the data is received in synchronization with the falling edge of the clock.

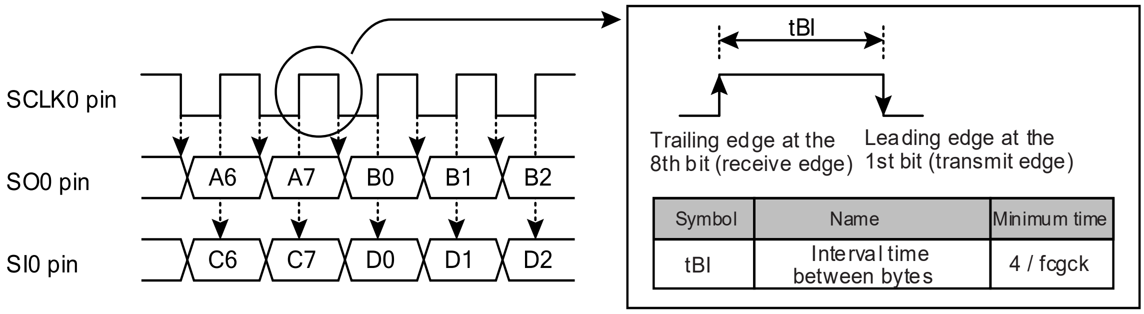


When SIO0CR<SIOEDG> = 0

When SIO0CR<SIOEDG> = 1

Figure 17-2 Transfer Edge

Note: When an external clock input is used, 4 / fcgck or longer is needed between the receive edge at the 8th bit and the transfer edge at the first bit of the next transfer.


SCLK0 pin
SO0 pin
SI0 pin
A6
A7
B0
B1
B2
C6
C7
D0
D1
D2
tBI
Trailing edge at the 8th bit (receive edge)
Leading edge at the 1st bit (transmit edge)

Symbol	Name	Minimum time
tBI	Interval time between bytes	4 / fcgck

Figure 17-3 Interval time between bytes

17.6. Transfer Modes

17.6.1. 8-bit transmit mode

The 8-bit transmit mode is selected by setting SIO0CR<SIOM> to "01".

17.6.1.1. Setting

Before starting the transmit operation, select the transfer edges at SIO0CR<SIOEDG>, a transfer format at SIO0CR<SIODIR> and a serial clock at SIO0CR<SIOCKS>. To use the internal clock as the serial clock, select an appropriate serial clock at SIO0CR<SIOCKS>. To use an external clock as the serial clock, set SIO0CR<SIOCKS> to "111".

The 8-bit transmit mode is selected by setting SIO0CR<SIOM> to "01".

The transmit operation is started by writing the first byte of transmit data to SIO0BUF and then setting SIO0CR<SIOS> to "1".

Writing data to SIO0CR<SIOEDG>, < SIOCKS> and <SIODIR> is invalid when the serial communication is in progress, or when SIO0SR<SIOF> is "1". Make these settings while the serial communication is stopped. While the serial communication is in progress (SIO0SR<SIOF> = 1), only writing "00" to SIO0CR<SIOM> or writing "0" to SIO0CR<SIOS> is valid.

17.6.1.2. Starting the transmit operation

The transmit operation is started by writing data to SIO0BUF and then setting SIO0CR<SIOS> to "1". The transmit data is transferred from SIO0BUF to the shift register, and then transmitted as the serial data from the SO0 pin according to the settings of SIO0CR<SIOEDG>, <SIOCKS> and <SIODIR>. The serial data becomes undefined when the transmit operation is started without writing any transmit data to SIO0BUF.

In the internal clock operation, the serial clock of the selected transfer speed is output from the SCLK0 pin. In the external clock operation, an external clock must be supplied to the SCLK0 pin.

By setting SIO0CR<SIOS> to "1", SIO0SR<SIOF> and < SEF> are automatically set to "1" and an INTSIO0 interrupt request is generated.

SIO0SR<SEF> is cleared to "0" when the 8th bit of the serial data is output.

17.6.1.3. Transmit buffer and shift operation

When data is written to SIO0BUF while the serial communication is in progress and the shift register is empty, the written data is transferred to the shift register immediately. At this time, SIO0SR<TBFL> remains at "0".

When data is written to SIO0BUF while some data remains in the shift register, SIO0SR<TBFL> is set to "1". When new data is written to SIO0BUF in this state, the contents of SIO0BUF are overwritten by the new value. Make sure that SIO0SR<TBFL> is "0" before writing data to SIO0BUF.

17.6.1.4. Transmit Operation on completion

The transmit operation on completion varies depending on the operating clock and the state of SIO0SR<TBFL>.

(1) When an internal clock is used and SIO0SR<TBFL> is "0"

When the transmit operation of the data is completed, the SCLK0 pin becomes the initial state and the SO0 pin becomes the “High” level. SIO0SR<SEF> remains at "0".

When the internal clock is used, the serial clock and data output is stopped until the next transmit data is written into SIO0BUF (automatic wait).

When the subsequent data is written into SIO0BUF, SIO0SR<SEF> is set to "1", the SCLK0 pin outputs the serial clock, and the transmit operation is restarted.

The INTSIO0 interrupt request is generated at the restart of the transmit operation.

(2) When an external clock is used and SIO0SR<TBFL> is "0"

When the transmit operation of the data is completed, the SO0 pin keeps last output value.

When a new transmit data to SIO0BUF after an external serial clock is input to the SCLK0 pin, a new data is output from the SO pin synchronous with an external serial clock.

When an external serial clock is input to the SCLK0 pin after the transmit operation of the data is completed, an underrun error occurs and the transmit underrun error flag SIO0SR<UERR> is set to "1". Then an undefined value is output from the SO pin.

When a transmit underrun error occurs, data must not be written to SIO0BUF during the transmission of an undefined value.

When a transmit underrun error occurs, the transmit operation should be finished by setting SIO0CR<SIOS> to "0" or by setting SIO0CR<SIOM> to "00".

The transmit underrun error flag SIO0SR<UERR> is cleared by reading SIO0SR.

(3) When an internal or external clock is used and SIO0SR<TBFL> is "1"

When the data transmission is completed, SIO0SR<TBFL> is cleared to "0". The data in SIO0BUF is transferred to the shift register and the transmission of subsequent data is started. At this time, SIO0SR<SEF> is set to "1" and the INTSIO0 interrupt request is generated.

17.6.1.5. Stopping the transmit operation

Clear SIO0CR<SIOS> to "0" to stop the transmit operation.

When SIO0SR<SEF> is "0", or when the shift operation is not in progress, the transmit operation is stopped immediately and the INTSIO0 interrupt request is generated. When SIO0SR<SEF> is "1", the transmit operation is stopped after all the data in the shift register is transmitted (reserved stop). At this time, the INTSIO0 interrupt request is generated.

When the transmit operation is completed, SIO0SR<SIOF>, <SEF> and <TBFL> are cleared to "0". Other SIO0SR registers keep their values.

When the internal clock has been used, the SO0 pin automatically returns to the “High” level. When an external clock has been used, the SO0 pin keeps the last output value. To return the SO0 pin to the “High” level, write "00" to SIO0CR<SIOM> when the transmit operation is stopped.

The transmit operation can be forced to stop by setting SIO0CR<SIOM> to "00" during the operation. By setting SIO0CR<SIOM> to "00", SIO0CR<SIOS> and SIO0SR are cleared to "0" and the SIO0 stops the operation, regardless of the SIO0SR<SEF> value. The SO0 pin becomes the “High” level. When the internal clock is selected, the SCLK0 pin returns to the initial level.

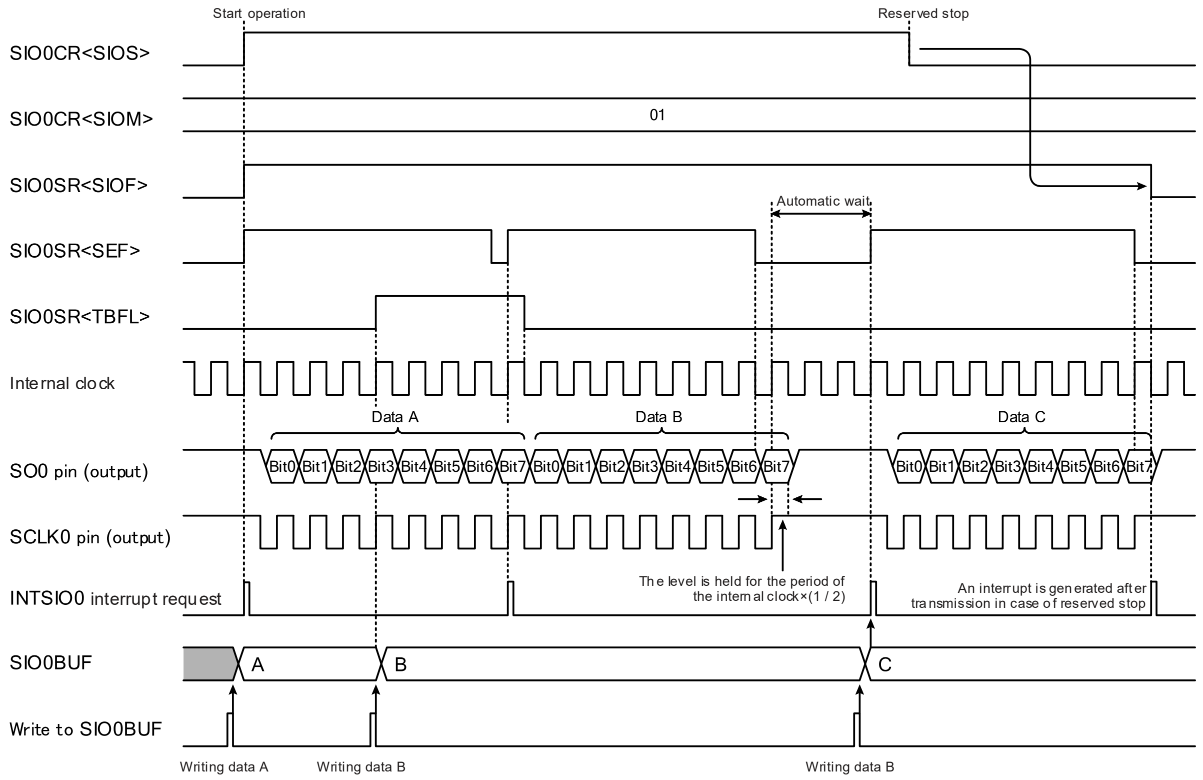


Figure 17-4 8-bit Transmit Mode (Internal Clock and Reserved Stop)

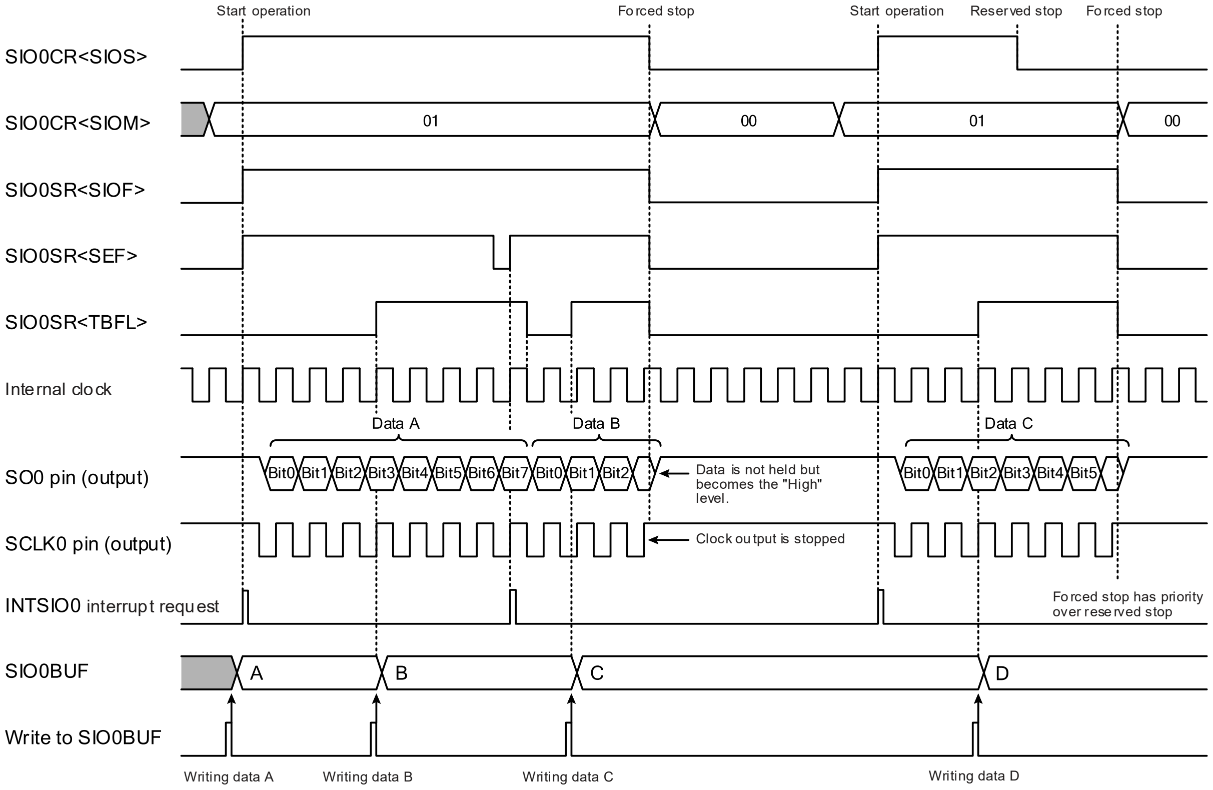


Figure 17-5 8-bit Transmit Mode (Internal Clock and Forced Stop)

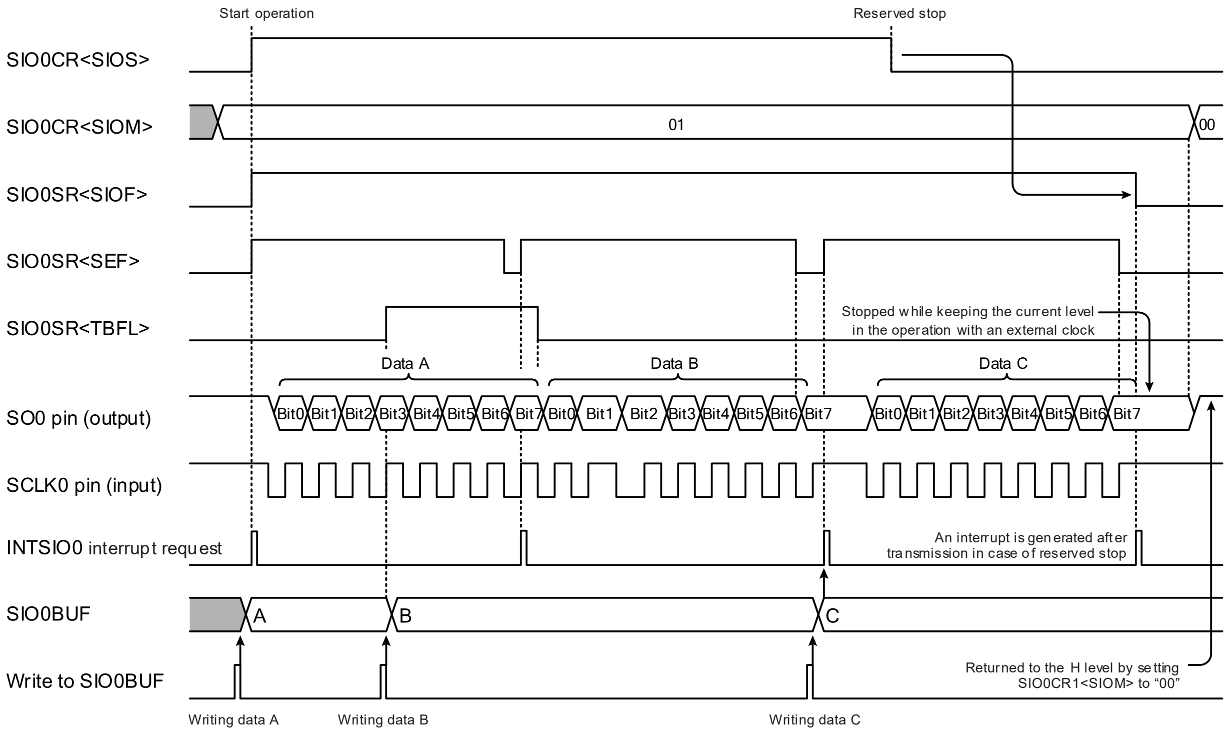

Start operation
Reserved stop
SIO0CR<SIOS>
SIO0CR<SIOM>
01
00
SIO0SR<SIOF>
SIO0SR<SEF>
SIO0SR<TBFL>
Stopped while keeping the current level in the operation with an external clock
Data A
Data B
Data C
SO0 pin (output)
Bit0 Bit1 Bit2 Bit3 Bit4 Bit5 Bit6 Bit7
SCLK0 pin (input)
INTSIO0 interrupt request
An interrupt is generated after transmission in case of reserved stop
SIO0BUF
A
B
C
Write to SIO0BUF
Returned to the H level by setting SIO0CR1<SIOM> to "00"
Writing data A
Writing data B
Writing data C

Figure 17-6 8-bit Transmit Mode (External Clock and Reserved Stop)

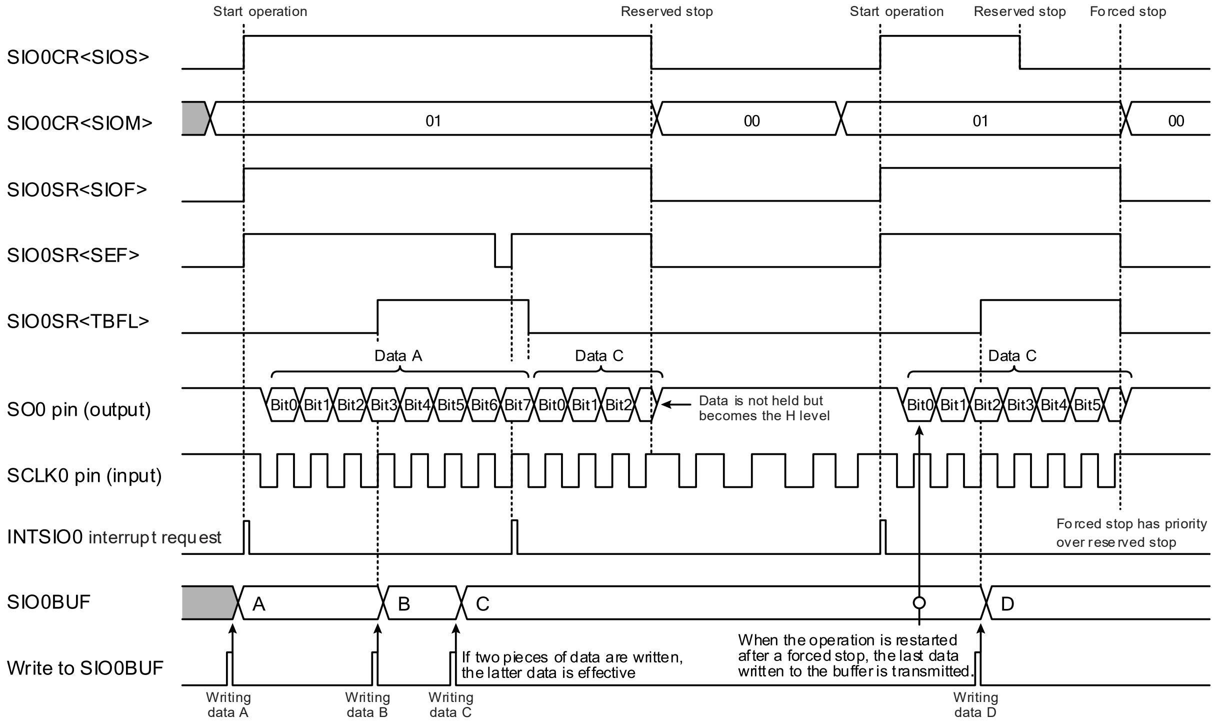

Start operation
Reserved stop
Start operation
Reserved stop
Forced stop
SIO0CR<SIOS>
SIO0CR<SIOM>
01
00
01
00
SIO0SR<SIOF>
SIO0SR<SEF>
SIO0SR<TBFL>
Data A
Data C
Data C
SO0 pin (output)
Bit0 Bit1 Bit2 Bit3 Bit4 Bit5 Bit6 Bit7
Data is not held but becomes the H level
SCLK0 pin (input)
INTSIO0 interrupt request
Forced stop has priority over reserved stop
SIO0BUF
A
B
C
D
Write to SIO0BUF
If two pieces of data are written, the latter data is effective
When the operation is restarted after a forced stop, the last data written to the buffer is transmitted.
Writing data A
Writing data B
Writing data C
Writing data D

Figure 17-7 8-bit Transmit Mode (External Clock and Forced Stop)

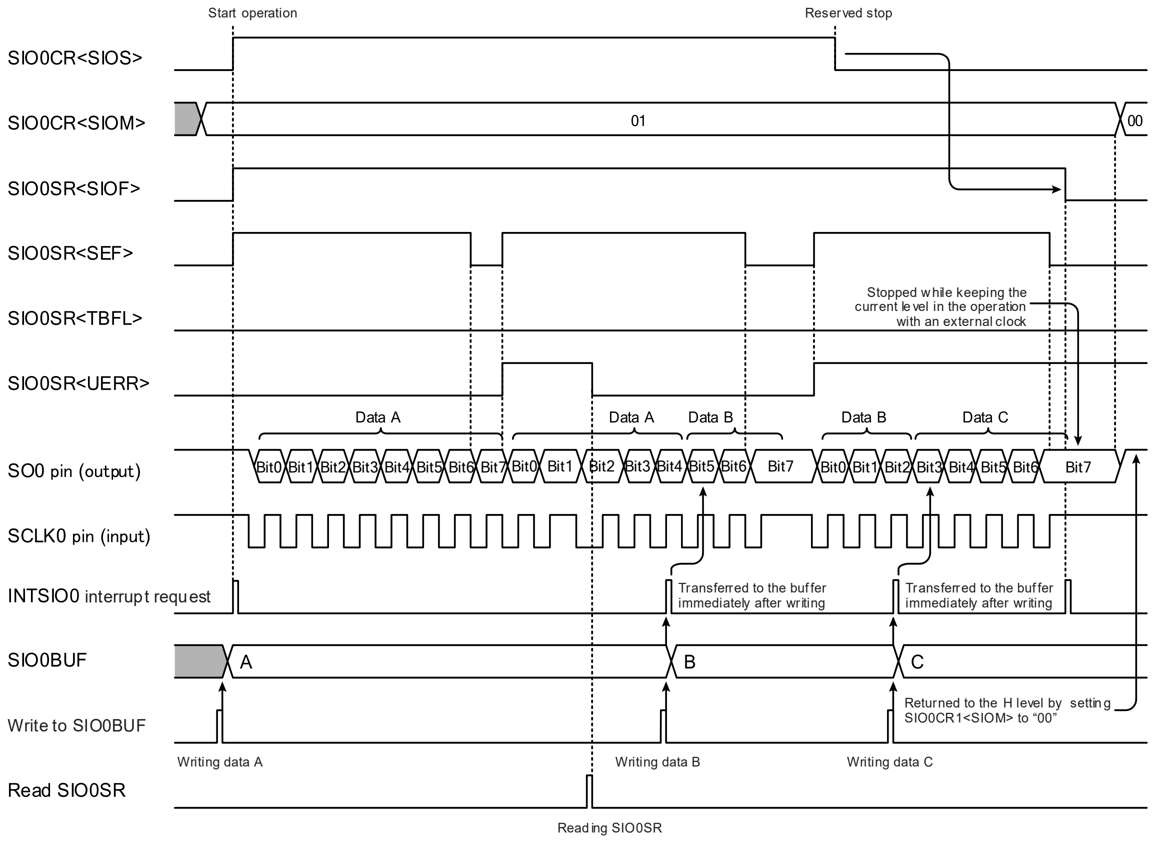

Start operation
Reserved stop
SIO0CR<SIOS>
SIO0CR<SIOM>
01
00
SIO0SR<SIOF>
SIO0SR<SEF>
SIO0SR<TBFL>
Stopped while keeping the current level in the operation with an external clock
SIO0SR<UERR>
Data A
Data A
Data B
Data B
Data C
SO0 pin (output)
Bit0 Bit1 Bit2 Bit3 Bit4 Bit5 Bit6 Bit7 Bit0 Bit1 Bit2 Bit3 Bit4 Bit5 Bit6 Bit7 Bit0 Bit1 Bit2 Bit3 Bit4 Bit5 Bit6 Bit7
SCLK0 pin (input)
INTSIO0 interrupt request
Transferred to the buffer immediately after writing
Transferred to the buffer immediately after writing
SIO0BUF
A
B
C
Write to SIO0BUF
Returned to the H level by setting SIO0CR1<SIOM> to "00"
Writing data A
Writing data B
Writing data C
Read SIO0SR
Reading SIO0SR

Figure 17-8 8-bit Transmit Mode (External Clock and Occurrence of Transmit Underrun Error)

17.6.2. 8-bit Receive Mode

The 8-bit receive mode is selected by setting SIO0CR<SIOM> to "10".

17.6.2.1. Setting

Before starting the receive operation, select the transfer edges at SIO0CR<SIOEDG>, a transfer format at SIO0CR<SIODIR> and a serial clock at SIO0CR<SIOCKS>. To use the internal clock as the serial clock, select an appropriate serial clock at SIO0CR<SIOCKS>. To use an external clock as the serial clock, set SIO0CR<SIOCKS> to "111".

The 8-bit receive mode is selected by setting SIO0CR<SIOM> to "10".

The receive operation is started by setting SIO0CR<SIOS> to "1".

Writing data to SIO0CR<SIOEDG>, <SIOCKS> and <SIODIR> is invalid when the serial communication is in progress, or when SIO0SR<SIOF> is "1". Make these settings while the serial communication is stopped. While the serial communication is in progress (SIO0SR<SIOF> = 1), only writing "00" to SIO0CR<SIOM> or writing "0" to SIO0CR<SIOS> is valid.

17.6.2.2. Starting the receive operation

The receive operation is started by setting SIO0CR<SIOS> to "1". External serial data is taken into the shift register from the SI0 pin according to the settings of SIO0CR<SIOEDG>, <SIOCKS> and <SIODIR>.

In the internal clock operation, the serial clock of the selected transfer speed is output from the SCLK0 pin. In the external clock operation, an external clock must be supplied to the SCLK0 pin.

By setting SIO0CR<SIOS> to "1", SIO0SR<SIOF> and <SEF> are automatically set to "1".

17.6.2.3. Operation on Completion of the receive operation

The receive operation of the 8-bit data is completed, the data from a shift register to SIO0BUF and the INTSIO0 interrupt request is generated. And the receive completion flag SIO0SR<REND> is set to “1” and SIO0SR<SEF> is cleared to “0”.

(1) When the internal clock is used

After the receive operation of the data is completed, the SCLK0 pin becomes the initial state.

When the internal clock is used, the serial clock output is stopped until the received data is read from SIO0BUF (automatic wait).

After the received data is read from SIO0BUF, SIO0SR<REND> is cleared to “0”, the serial clock output is restarted and received operation is continuous.

At this time, SIO0SR<SEF> is set to “1”

(2) When the external clock is used

When the external clock is used, the receive operation of the data can be continuous by inputting an external clock from the external before the received data is read from SIO0BUF. SIO0SR<SEF> is set to “1” after starting the receive operation.

When the received data is not read from SIO0BUF before the receive operation of the next data is completed, an overrun error is generated and an overrun error flag SIO0SR<OERR> is set to “1”.

The data received at the occurrence of an overrun error is discarded, and SIO0BUF holds the data value received before the occurrence of the overrun error. SIO0SR<OERR> is cleared after reading SIO0SR.

When an overrun error has occurred, set SIO0CR<SIOS> to “0” or set SIO0CR<SIOM> to "00" to abort the receive operation.

Before the received operation of the next data is completed, SIO0SR<REND> is cleared to “0” by reading the data from SIO0BUF.

17.6.2.4. Stopping the receive operation

Set SIO0CR<SIOS> to "0" to stop the receive operation.

When SIO0SR<SEF> is "0", or when the shift operation is not in progress, the operation is stopped immediately. Unlike the transmit mode, no INTSIO0 interrupt request is generated in this state.

When SIO0SR<SEF> is "1", the operation is stopped after the 8-bit data has been completely received (reserved stop). At this time, an INTSIO0 interrupt request is generated.

After the operation has stopped completely, SIO0SR<SIOF> and <SEF> are cleared to "0". Other SIO0SR registers keep their values.

The receive operation can be forced to stop by setting SIO0CR<SIOM> to "00" during the operation. By setting SIO0CR<SIOM> to "00", SIO0CR<SIOS> and SIO0SR are cleared to "0" and the SIO stops the operation, regardless of the SIO0SR<SEF> value. When the internal clock is selected, the SCLK0 pin returns to the initial level.

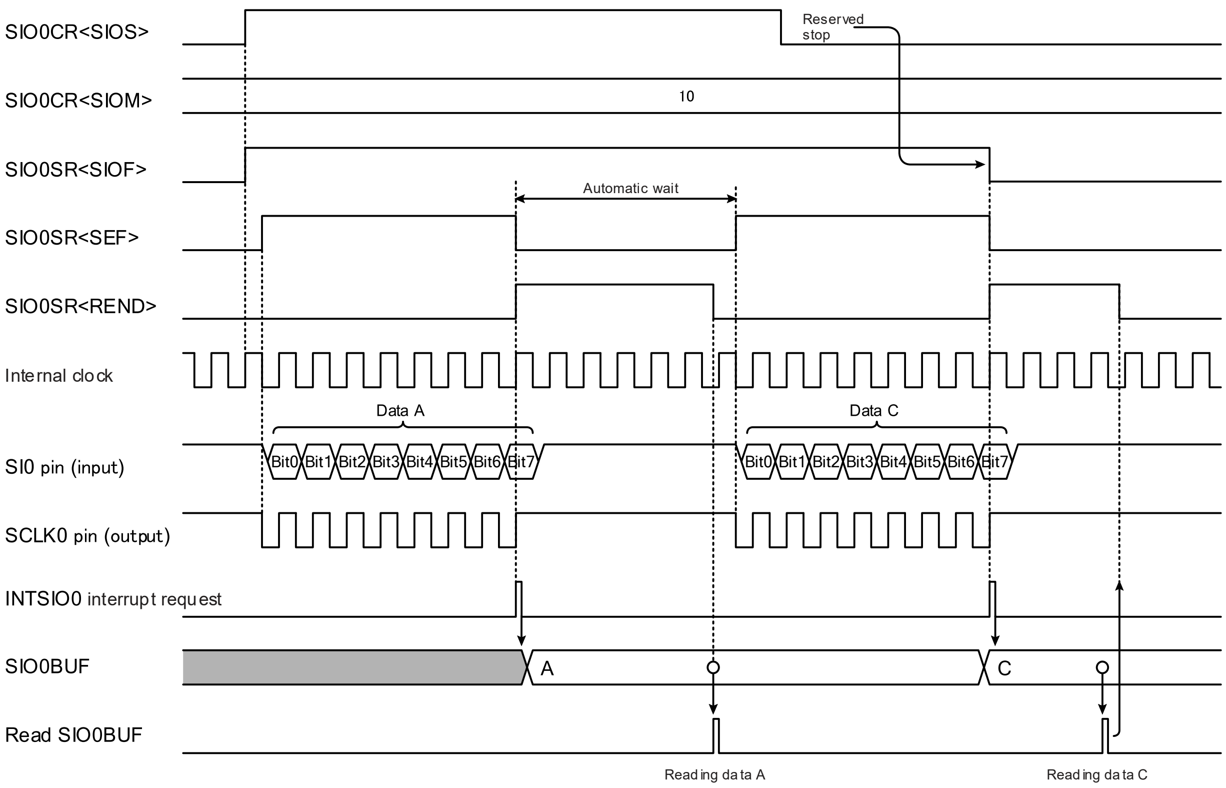


Figure 17-9 8-bit Receive Mode (Internal Clock and Reserved Stop)

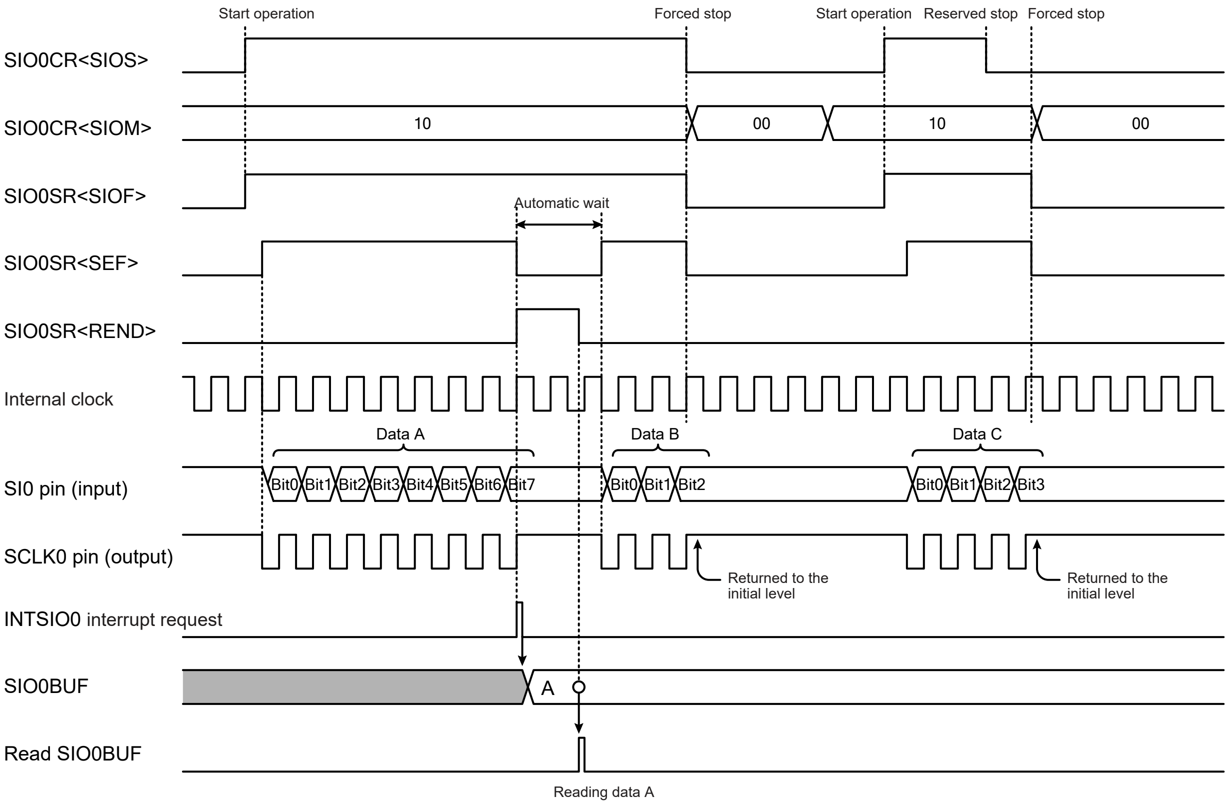


Figure 17-10 8-bit Receive Mode (Internal Clock and Forced Stop)

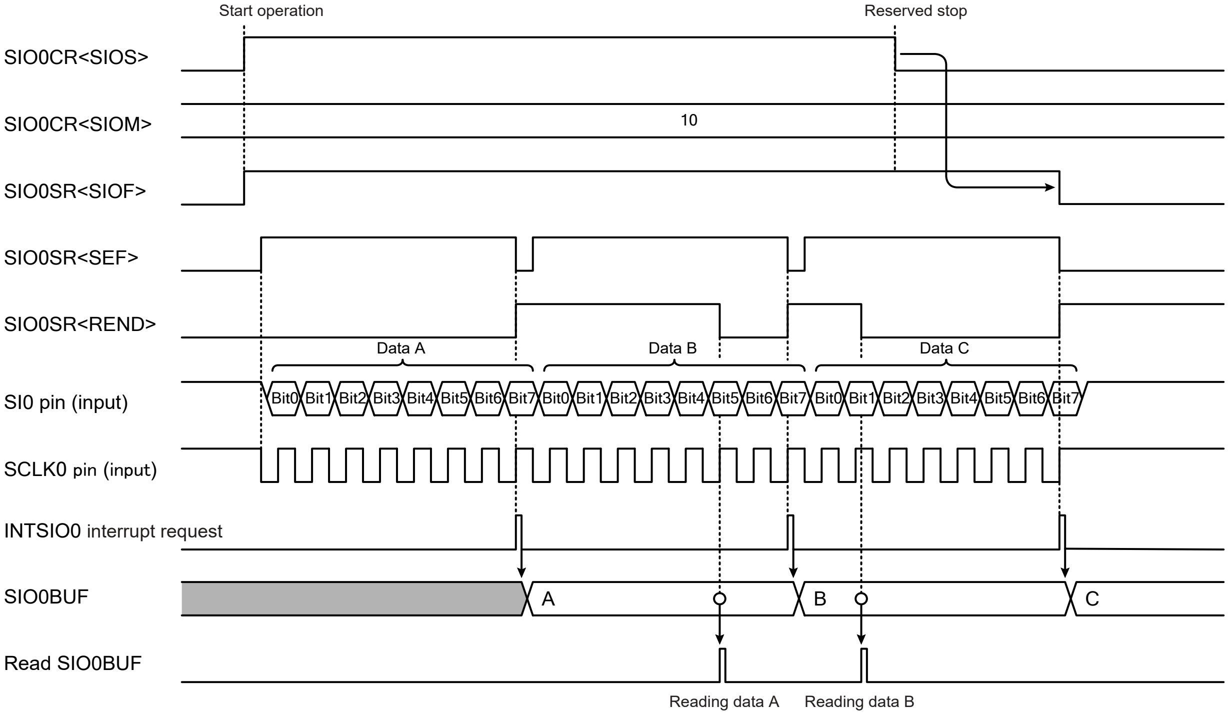


Figure 17-11 8-bit Receive Mode (External Clock and Reserved Stop)

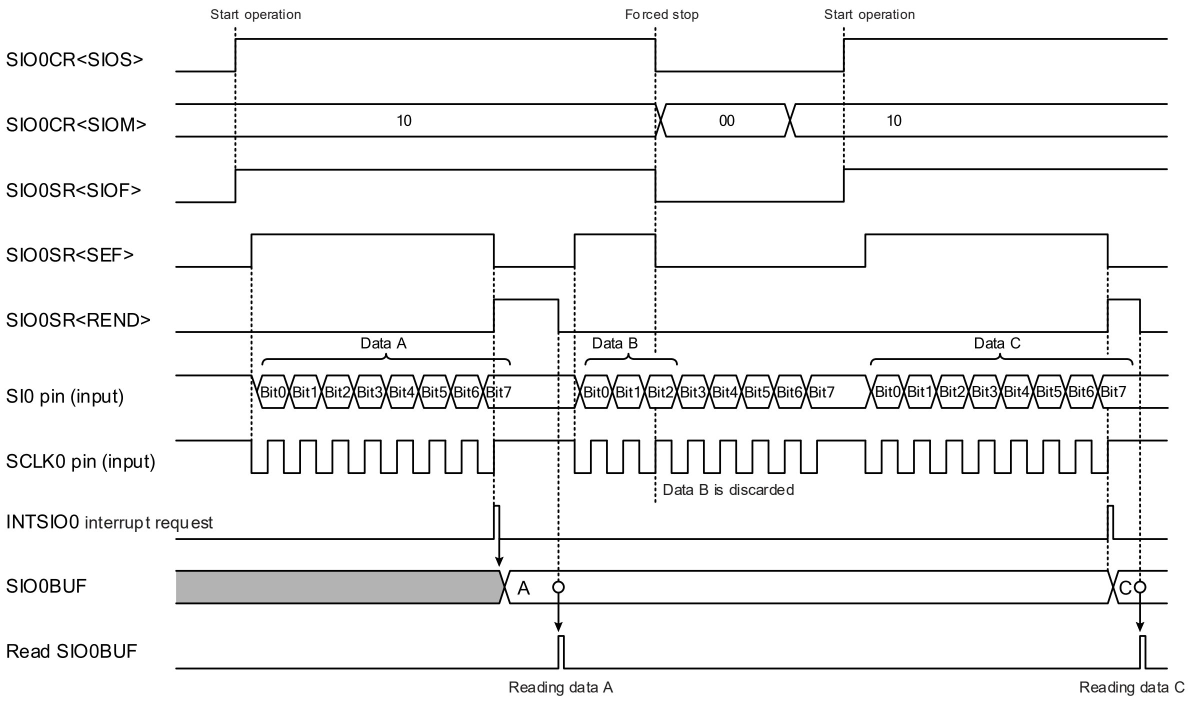


Figure 17-12 8-bit Receive Mode (External Clock and Forced Stop)

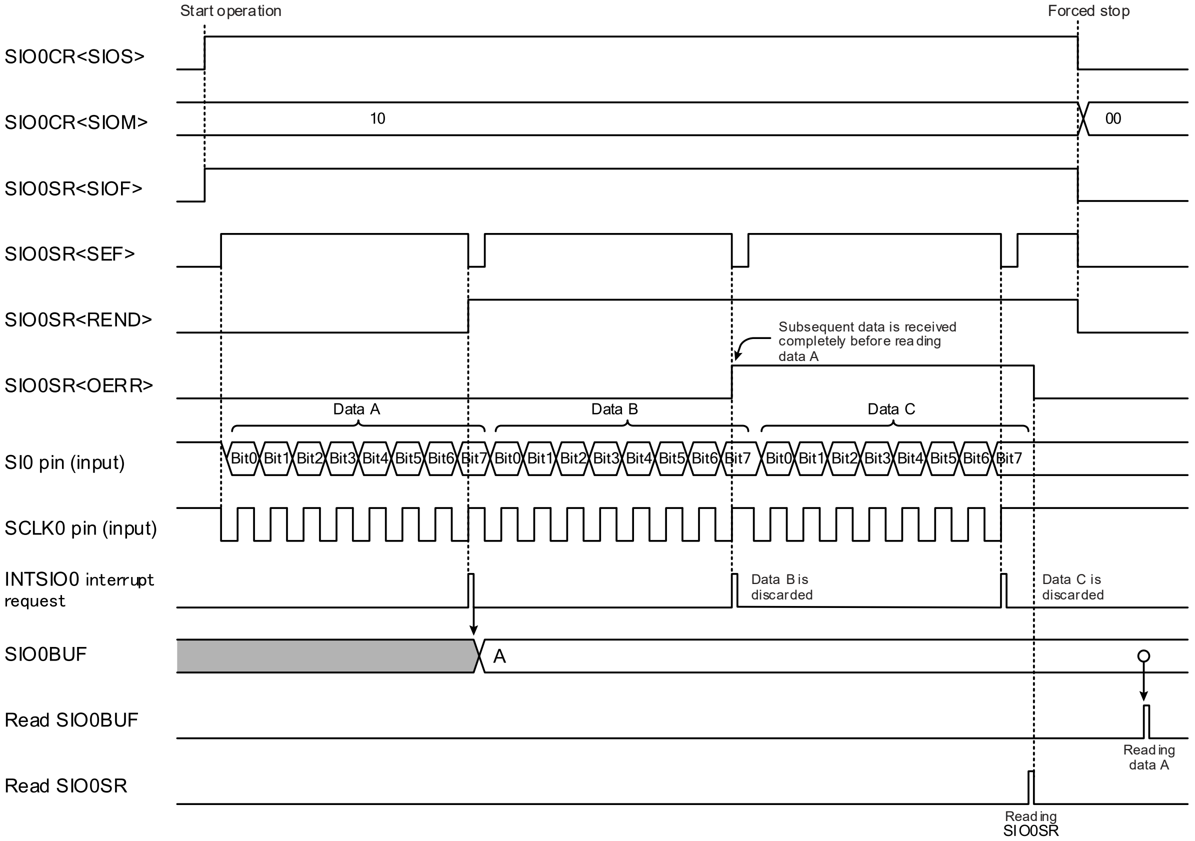


Figure 17-13 8-bit Receive Mode (External Clock and Occurrence of Overrun Error)

17.6.3. 8-bit transmit/receive mode

The 8-bit transmit/receive mode is selected by setting SIO0CR<SIOM> to "11".

17.6.3.1. Setting

Before starting the transmit/receive operation, select the transfer edges at SIO0CR<SIOEDG>, a transfer format at SIO0CR<SIODIR> and a serial clock at SIO0CR<SIOCKS>. To use the internal clock as the serial clock, select an appropriate serial clock at SIO0CR<SIOCKS>. To use an external clock as the serial clock, set SIO0CR<SIOCKS> to "111".

The 8-bit transmit/receive mode is selected by setting SIO0CR<SIOM> to "11".

The transmit/receive operation is started by writing the first byte of transmit data to SIO0BUF and then setting SIO0CR<SIOS> to "1".

Writing data to SIO0CR<SIOEDG>, <SIOCKS> and <SIODIR> is invalid when the serial communication is in progress, or when SIO0SR<SIOF> is "1". Make these settings while the serial communication is stopped. While the serial communication is in progress (SIO0SR<SIOF> = 1), only writing "00" to SIO0CR<SIOM> or writing "0" to SIO0CR<SIOS> is valid.

17.6.3.2. Starting the transmit/receive operation

The transmit/receive operation is started by writing data to SIO0BUF and then setting SIO0CR<SIOS> to "1". The transmit data is transferred from SIO0BUF to the shift register, and the serial data is transmitted from the SO0 pin according to the settings of SIO0CR<SIOEDG>, <SIOCKS> and <SIODIR>. At the same time, the serial data is received from the SI0 pin according to the settings of SIO0CR<SIOEDG>, <SIOCKS> and <SIODIR>.

In the internal clock operation, the serial clock of the selected transfer speed is output from the SCLK0 pin. In the external clock operation, an external clock must be supplied to the SCLK0 pin.

The transmit data becomes undefined when the transmit/receive operation is started without writing any transmit data to SIO0BUF.

By setting SIO0CR<SIOS> to "1", SIO0SR<SIOF> and <SEF> are automatically set to "1" and the INTSIO0 interrupt request is generated.

SIO0SR<SEF> is cleared to "0" when the 8th bit of data is received.

17.6.3.3. Transmit buffer and shift operation

When data is written to SIO0BUF while the serial communication is in progress and the shift register is empty, the written data is transferred to the shift register immediately. At this time, SIO0SR<TBFL> remains at "0".

When data is written to SIO0BUF while data remains in the shift register, SIO0SR<TBFL> is set to "1". When new data is written to SIO0BUF in this state, the contents of SIO0BUF are overwritten by the new value. Make sure that SIO0SR<TBFL> is "0" before writing data to SIO0BUF.

17.6.3.4. Transmit/Receive Operation on Completion

When the data transmit/receive operation is completed, SIO0SR<REND> is set to "1" and the INTSIO0 interrupt request is generated.

The operation varies depending on the operating clock and SIO0SR<TBFL>.

(1) When an internal clock is used and SIO0SR<TBFL> is “0”

When the transmit/receive operation of the data is completed, the SCLK0 pin becomes the initial state and the SO0 pin becomes the “High” level. SIO0SR<SEF> remains at “0”.

When the internal clock is used, the serial clock output is stopped until the next transmit data is written into SIO0BUF (automatic wait).

After the received data is read from SIO0BUF, SIO0SR<REND> is cleared to “0”.

When the new data is written into SIO0BUF, SIO0SR<SEF> is set to “1”, the SCLK0 pin outputs the serial clock, and the transmit/receive operation is restarted.

Therefore, the received data must be read from SIO0BUF before the new data is written to SIO0BUF.

(2) When an external clock is used and SIO0SR<TBFL> is “0”

When the transmit/receive operation of the data is completed, the SO0 pin keeps the last output value.

When a new transmit data to SIO0BUF after an external serial clock is input to the SCLK0 pin, a new data is output from the SO pin synchronous with an external serial clock.

At this time, because the new data is continuously received, the last data must be read from SIO0BUF before the new data transmit/receive operation is completed.

A transmit underrun error is generated and a transmit underrun error flag SIO0SR<UERR> is set to “1” when an external serial clock is input without writing a new data to SIO0BUF. At this time, the previous data which is set to SIO0BUF is transmitted again.

When the received data in SIO0BUF is not read until the receive operation of the next data is completed, an overrun error is generated and an overrun error flag SIO0SR<OERR> is set to “1”.

When a transmit underrun error or an overrun error is generated, clear SIO0CR<SIOS> to “0” or set SIO0VR<SIOM> to “00” to stop the transmit/receive operation.

(3) When an internal or external clock is used and SIO0SR<TBFL> is “1”

After the transmit/receive operation of the data is completed, the new data which is written SIO0BUF is transferred to shift register and the transmit/receive operation of the new data is started. At this time, SIO0SR<TBFL> is cleared to “0” and SIO0SR<SEF> is kept “1”.

Because the new transmit/receive operation is executed, the previous data must be read from SIO0BUF before the new transmit/receive operation is completed.

When the previous data is not read from SIO0BUF before the new transmit/receive operation is completed, an overrun error is generated and an overrun error flag SIO0SR<OERR> is set to “1”.

When a transmit underrun error or an overrun error is generated, clear SIO0CR<SIOS> to “0” or set SIO0CR<SIOM> to “00” to stop the transmit/receive operation.

17.6.3.5. Stopping the transmit/receive operation

Set SIO0CR<SIOS> to "0" to stop the transmit/receive operation. When SIO0SR<SEF> is "0", or when the shift operation is not in progress, the operation is stopped immediately. Unlike the transmit mode, no INTSIO0 interrupt request is generated in this state.

When SIO0SR<SEF> is "1", the operation is stopped after the 8-bit data is received completely. At this time, an INTSIO0 interrupt request is generated.

After the operation has stopped completely, SIO0SR<SIOF>, <SEF> and <TBFL> are cleared to "0". Other SIO0SR registers keep their values.

When an internal clock has been used, the SO0 pin automatically returns to the “High” level. When an external clock has been used, the SO0 pin keeps the last output value. To return the SO0 pin to the “High” level, write "00" to SIO0CR<SIOM> when the operation is stopped.

The transmit/receive operation can be forced to stop by setting SIO0CR<SIOM> to "00" during the operation. By setting SIO0CR<SIOM> to "00", SIO0CR<SIOS> and SIO0SR are cleared to "0" and the SIO stops the operation, regardless of the SIO0SR<SEF> value. The SO0 pin becomes the “High” level. When the internal clock is selected, the SCLK0 pin returns to the initial level.

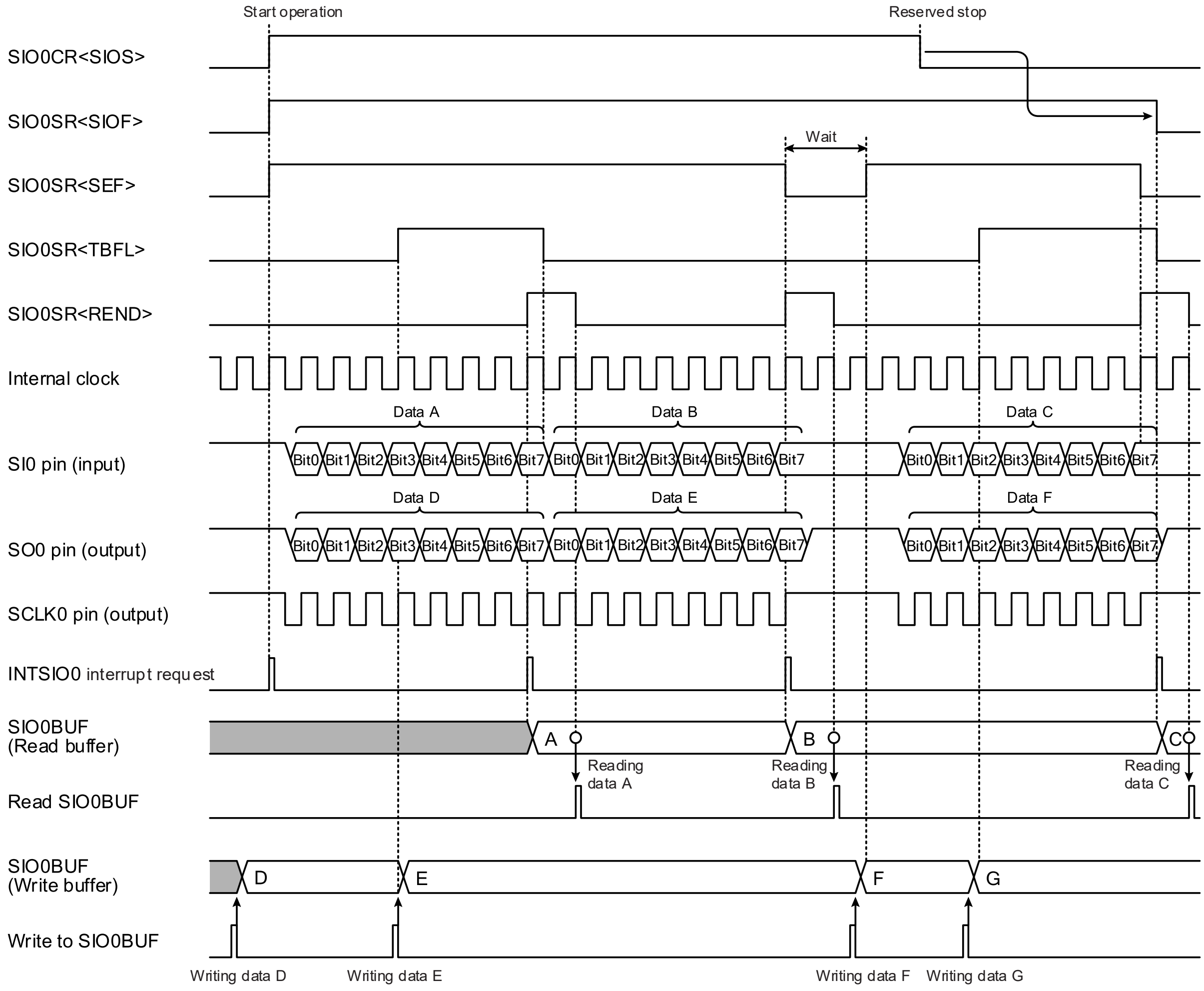


Figure 17-14 8-bit Transmit/Receive Mode (Internal Clock and Reserved Stop)

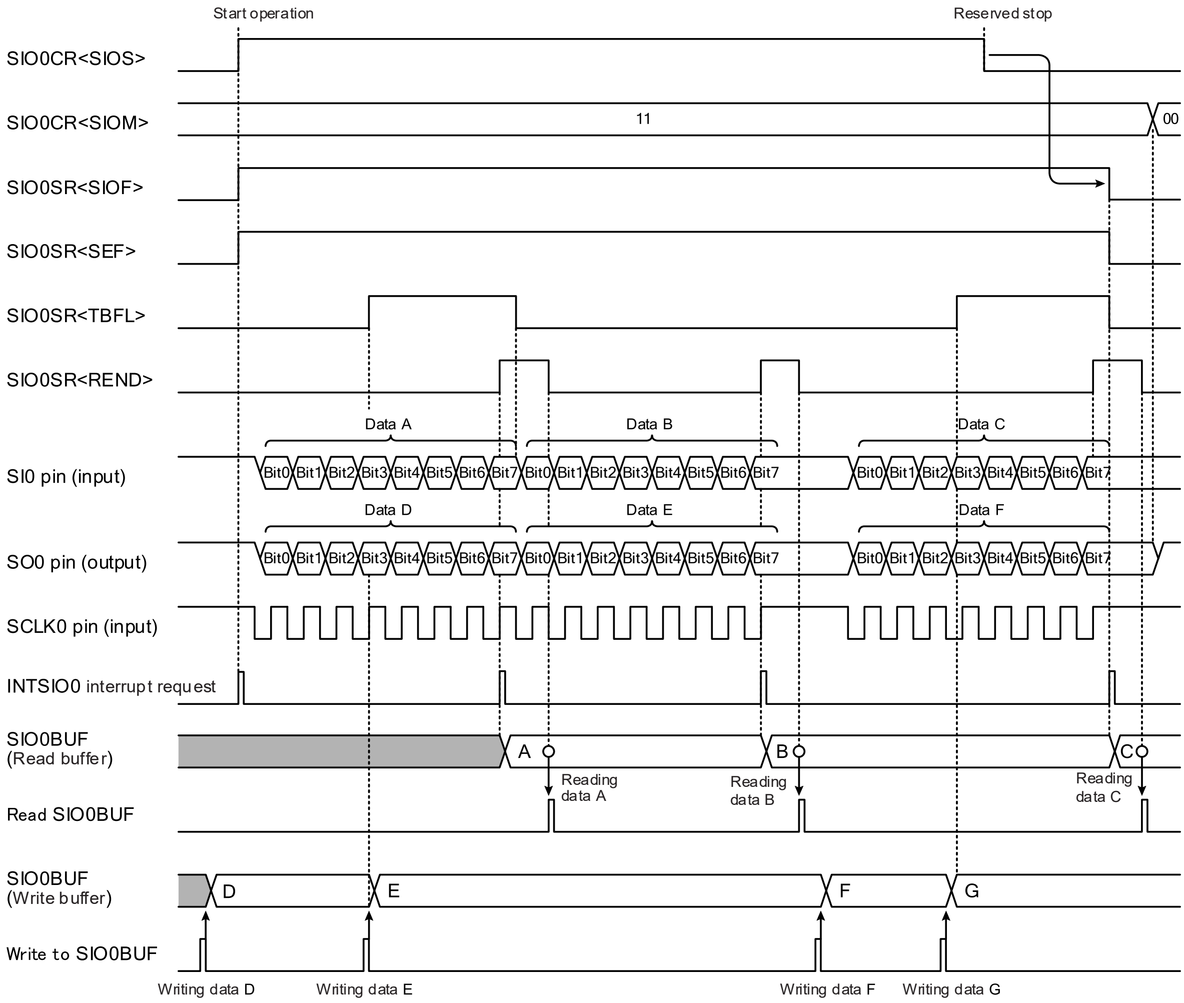


Figure 17-15 8-bit Transmit/Receive Mode (External Clock and Reserved Stop)

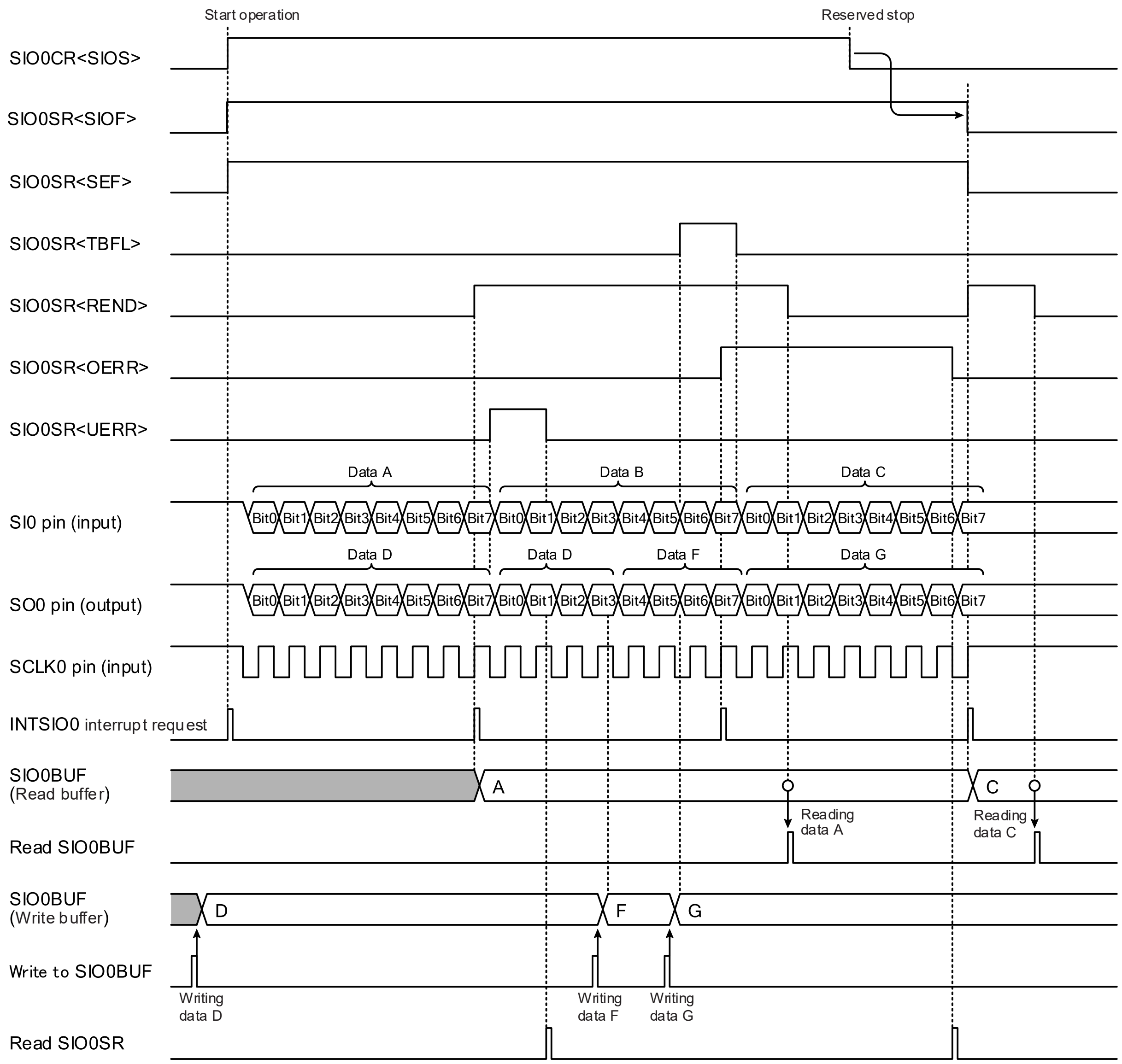


Figure 17-16 8-bit Transmit/Receive Mode (External Clock, Occurrence of Transmit Underrun Error and Occurrence of Overrun Error)

17.7. AC Characteristics

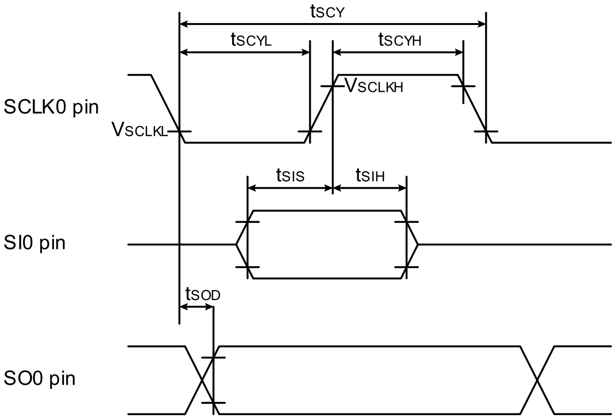


Figure 17-17 AC Characteristics

(V_{SS} = 0 [V], V_{DD} = 4.5 to 5.5 [V], Topr = -40 to 85 [°C])

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
SCLK0 pin cycle time	t_{SCY}	Internal clock operation SO0 pin and SCLK0 pin load capacity = 100 [pF]	2 / fcgck	-	-	ns
SCLK0 pin "Low" level pulse width	t_{SCYL}		1 / fcgck-25	-	-	
SCLK0 pin "High" level pulse width	t_{SCYH}		1 / fcgck-15	-	-	
SI0 pin input setup time	t_{SIS}		60	-	-	
SI0 pin input hold time	t_{SIH}		35	-	-	
SO0 pin output delay time	t_{SOD}		-50	-	50	
SCLK0 pin cycle time	t_{SCY}	External clock operation SO0 pin and SCLK0 pin load capacity = 100 [pF]	2 / fcgck	-	-	
SCLK0 pin "Low" level pulse width	t_{SCYL}		1 / fcgck	-	-	
SCLK0 pin "High" level pulse width	t_{SCYH}		1 / fcgck	-	-	
SI0 input setup time	t_{SIS}		50	-	-	
SI0 input hold time	t_{SIH}		50	-	-	
SO0 pin output delay time	t_{SOD}		0	-	60	
SCLK0 pin "Low" level input voltage	t_{SCLKL}	-	0	-	V_{DD} × 0.30	V
SCLK0 pin "High" level input voltage	t_{SCLKH}	-	V_{DD} × 0.70	-	V_{DD}	

18. Serial Bus Interface (SBI)

The TMP89FS60B/62B/63B contains the serial bus interfaces (SBI).

The SBI supports serial communication conforming to the I^2C bus standards. It has clock synchronization and arbitration functions, and supports the multi-master in which multiple masters are connected on a bus.

It also supports the unique free data format.

This chapter describes the SBI0.

Table 18-1 SFR Address Assignment

	SBIxCR1 (address)	SBIxCR2 (address)	SBIxSR2 (address)	I2CxAR (address)	SBIxDBR (address)	Low power consumption register
SBI0	SBI0CR1 (0x0022)	SBI0CR2 (0x0023)	SBI0SR2 (0x0023)	I2C0AR (0x0024)	SBI0DBR (0x0025)	POFFCR1 <SBI0EN>

Table 18-2 Pin Names

	Serial data input/output pin	Serial clock input/output pin
SBI0	SDA0 pin	SCL0 pin

18.1. Serial Bus Interface (SBI) for Each Product

The serial bus interfaces (SBI) for each product is shown in Table 18-3.

In regards to unavailable channel of the SBI0, the corresponding bit of the low power register must be cleared to “0”.

Table 18-3 Serial Bus Interface (SBI) for Each Product

	TMP89FS60B	TMP89FS62B	TMP89FS63B
SBI0	A	NA	A

Note: A: Available, NA: Not available

18.2. Communication Format

18.2.1. I²C Bus

The I²C bus is connected to devices via the SDA0 and SCL0 pins and can communicate with multiple devices.

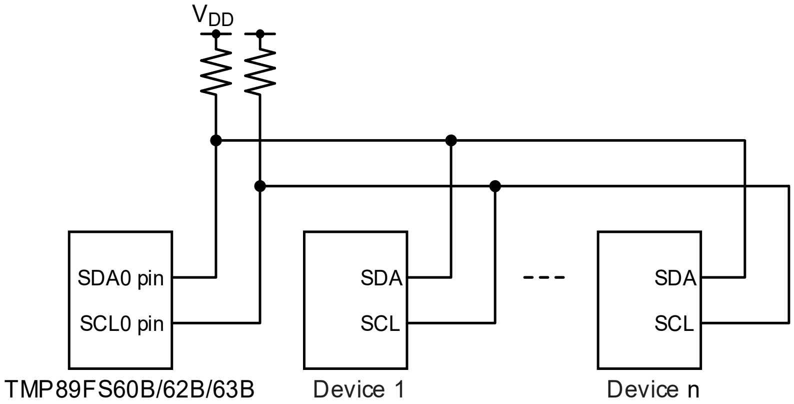


Figure 18-1 Device Connections

The Communications are implemented between masters and slaves.

The master transmits the start condition, the slave addresses, the direction bit and the stop condition to the slave(s) connected to the bus, and transmits and receives data.

The slave detects these conditions transmitted from the master by the hardware, and transmits and receives data.

The data format of the I²C bus that can communicate via the serial bus interface is shown in Figure 18-2.

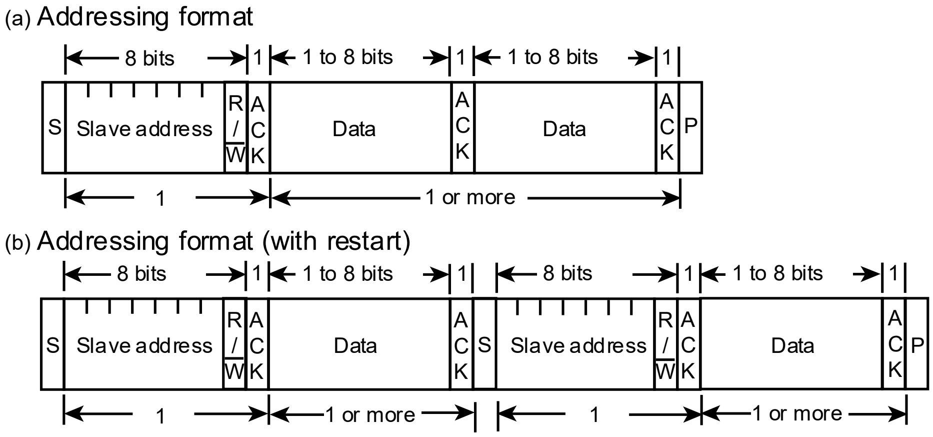
The SBI0 does not support the following functions among those specified by the I²C bus standards:

(1) Start byte

(2) 10-bit addressing

(3) SDA0 and SCL0 pins falling edge slope control

(4) The I/O pins of the serial bus interface do not avoid the transmission on the SDA and SCL line when the power supply (V_{DD}) is turned off.



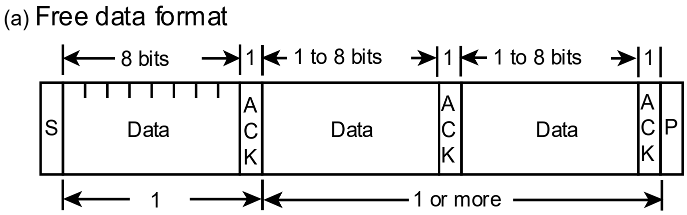
S : Start condition
R/$\overline{W}$: Direction bit
ACK : Acknowledge bit
P : Stop condition

Figure 18-2 Data Format of I²C Bus

18.2.2. Free data format

The free data format is for communication between a master and slave.

In the free data format, the slave address and the direction bit are processed as data.



S : Start condition
ACK : Acknowledge bit
P : Stop condition

Figure 18-3 Free data format

18.3. Configuration

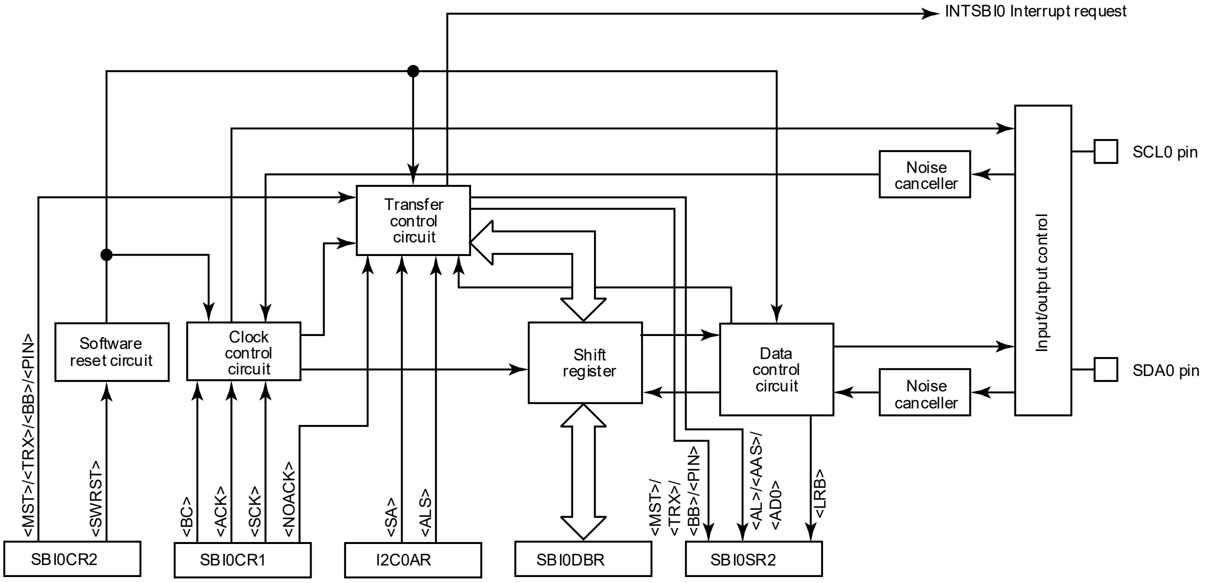


Figure 18-4 Serial Bus Interface 0 (SBI0)

18.4. Control

The SBI0 is controlled by the low power consumption registers (POFFCR1), Serial bus interface control register 1 (SBI0CR1), Serial bus interface control register 2 (SBI0CR2), Serial bus interface status register 2 (SBI0SR2), Serial bus interface data buffer register (SBI0DBR) and I^{2}C bus address register (I2C0AR).

Low power consumption register 1

POFFCR1		7	6	5	4	3	2	1	0
(0x0F75)	Bit Symbol	-	-	-	SBI0EN	-	(UART2EN)	(UART1EN)	(UART0EN)
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

SBI0EN (Note 3)	SBI0 control	0: 1:	Disable Enable

Note 1: When <SBI0EN> is cleared to "0", the clock supply to the SBI0 is stopped. At this time, the data written to the serial bus interface control registers is invalid. When the SBI0 is used, set <SBI0EN> to "1" and then write the data to the serial bus interface control registers.

Note 2: Note: The written values are read from the bits 7 to 5, 3 of POFFCR1 These bits must be cleared to “0”.

Note 3: <SBI0EN> is “Reserved” for the TMP89FS62B.

Serial bus interface control register 1

SBI0CR1		7	6	5	4	3	2	1	0
(0x0022)	Bit Symbol	BC			ACK	NOACK	SCK		
	Read/Write	R/W			R/W	R/W	R/W		
	After reset	0	0	0	0	0	0	0	0

BC	Select number of data bits		<ACK> = 0		<ACK> = 1	
			Number of clocks for data transfer	Number of data bits	Number of clocks for data transfer	Number of data bits
		000:	8	8	9	8
		001:	1	1	2	1
		010:	2	2	3	2
		011:	3	3	4	3
		100:	4	4	5	4
		101:	5	5	6	5
		110:	6	6	7	6
		111:	7	7	8	7
ACK	Generation and counting of the clocks for an acknowledge bit specification		Master mode		Slave mode	
		0:	Not generating the clocks for an acknowledge bit. Generate an INTSBI0 interrupt request when the data transfer is finished (non-acknowledgment mode)		Generate an INTSBI0 interrupt request when the data transfer is finished (non-acknowledgment mode)	
		1:	Generate the clocks for an acknowledge bit and an INTSBI0 interrupt request when the data transfer is finished (acknowledgment mode)		Count the clocks for an acknowledge bit and generate an INTSBI0 interrupt request when the data transfer is finished (acknowledgment mode)	
NOACK	Enables/disables the slave address match detection and the GENERAL CALL detection		Master mode		Slave mode	
		0:	Don’t Care		Enable the slave address match detection and the GENERAL CALL detection	
		1:	Don’t Care		Disable the slave address match detection and the GENERAL CALL detection	
SCK	Select “High” level and “Low” level period of SCL0 pin in the master mode And select time before the release of the SCL0 pin in the slave mode		t_{HIGH} (m / fcgck)	t_{LOW} (n / fcgck)	fscl@fcgck = 8 [MHz]	fscl@fcgck = 4 [MHz]
			m	n		
		000:	9	12	381 [kHz]	Reserved (Note 5)
		001:	11	14	320 [kHz]	Reserved (Note 5)
		010:	15	18	242 [kHz]	Reserved (Note 5)
		011:	23	26	163 [kHz]	82 [kHz]
		100:	39	42	99 [kHz]	49 [kHz]
		101:	71	74	55 [kHz]	28 [kHz]
		110:	135	138	29 [kHz]	15 [kHz]
		111:	263	266	15 [kHz]	8 [kHz]

Note 1: fcgck: Gear clock [Hz], fs: Low-frequency clock [Hz]

Note 2: Don't change the contents of the registers while the start condition is generated, the stop condition is generated or the data transfer is in progress. Write data to the registers before the start condition is generated or during the period from generating of the INTSBI0 interrupt request for stopping the data transfer to releasing it.

Note 3: After a software reset is generated, all the bits of SBI0CR2 except SBI0CR2<SBIM> and the SBI0CR1, I2C0AR and SBI0SR2 are initialized.

Note 4: When the operation mode is switched to STOP, IDLE0 or SLOW mode, the SBI0CR2, except SBI0CR2<SBIM>, and the SBI0CR1, I2C0AR and SBI0DBR are initialized.

Note 5: When fcgck is 4 [MHz], <SCK> must not be set to "000", "001" or "010" because it is not possible to satisfy the I^{2}C bus specification of fast mode.

Serial bus interface control register 2

SBI0CR2		7	6	5	4	3	2	1	0
(0x0023)	Bit Symbol	MST	TRX	BB	PIN	SBIM	-	SWRST	
	Read/Write	W	W	W	W	W	R	W	
	After reset	0	0	0	1	0	0	0	0

MST	Master/slave selection	0: 1:	Slave Master
TRX	Transmitter/receiver selection	0: 1:	Receiver Transmitter
BB	Start/stop generation	0: 1:	Generate the stop condition (when <MST>, <TRX> and <PIN> are "1") Generate the start condition (when <MST>, <TRX> and <PIN> are "1")
PIN	Cancel interrupt service request	0: 1:	- (Cannot clear this bit by the software) Release interrupt service request
SBIM	Serial bus interface operation mode register	0: 1:	Port mode Serial bus interface mode
SWRST	Software reset start bit	The software reset starts by first writing "10" and next writing "01"	

Note 1: When SBI0CR2<SBIM> is "0", no value can be written to SBI0CR2 except SBI0CR2<SBIM>. Before writing values to SBI0CR2, write "1" to SBI0CR2<SBIM> to activate the serial bus interface mode.

Note 2: Don't change the contents of the registers except SBI0CR2<SWRST> when the start condition is generated, the stop condition is generated or the data transfer is in progress. Write data to the registers before the start condition is generated or during the period from when an INTSBI0 interrupt request is generated for stopping the data transfer until it is released.

Note 3: Make sure that the port is in a high state before switching the port mode to the serial bus interface mode. Make sure that the bus is free before switching the serial bus interface mode to the port mode.

Note 4: SBI0CR2 is a write-only register, and must not be accessed by using a read-modify-write instruction, such as a bit operation.

Note 5: After a software reset is generated, all the bits of SBI0CR2 except SBI0CR2<SBIM>, SBI0CR1, I2C0AR and SBI0SR2 are initialized.

Note 6: When the operation is switched to STOP, IDLE0 or SLOW mode, all the bits of SBI0CR2 except SBI0CR2<SBIM>, SBI0CR1, I2C0AR and SBI0DBR are initialized.

Serial bus interface status register 2

SBI0SR2 (0x0023)		7	6	5	4	3	2	1	0
	Bit Symbol	MST	TRX	BB	PIN	AL	AAS	AD0	LRB
	Read/Write	R	R	R	R	R	R	R	R
	After reset	0	0	0	1	0	0	0	* (Note 1)

MST	Master/slave selection status monitor	0: 1:	Slave Master
TRX	Transmitter/receiver selection status monitor	0: 1:	Receiver Transmitter
BB	Bus status monitor	0: 1:	Bus free Bus busy
PIN	Interrupt service request status monitor	0: 1:	Requesting interrupt service Releasing interrupt service request
AL	Arbitration lost detection monitor	0: 1:	- Arbitration lost detected
AAS	Slave address match detection monitor	0: 1:	- Slave address match or GENERAL CALL detected
AD0	GENERAL CALL detection monitor	0: 1:	- GENERAL CALL detected
LRB	Last received bit monitor	0: 1:	Last received bit is "0" Last received bit is "1"

Note 1: * : Undefined

Note 2: When SBI0CR2<SBIM> becomes "0", SBI0SR is initialized.

Note 3: After a software reset is generated, all the bits of SBI0CR2 except SBI0CR2<SBIM> and SBI0CR1, I2C0AR and SBI0SR2 are initialized.

Note 4: When the operation mode is switched to STOP, IDLE0 or SLOW mode, all the bits of SBI0CR2 except SBI0CR2<SBIM>, SBI0CR1, I2C0AR and SBI0DBR are initialized.

I^{2}C bus address register

I2C0AR (0x0024)		7	6	5	4	3	2	1	0
	Bit Symbol	SA							ALS
	Read/Write	R/W							R/W
	After reset	0	0	0	0	0	0	0	0

SA	Slave address setting	Slave address in the slave mode	
ALS	Communication format selection	0: 1:	I^{2}C bus mode Free data format

Note 1: Don't set I2C0AR to "0x00". When it is set to "0x00", the slave address is deemed to be matched when the I^{2}C bus standard start byte "0x01" is received in the slave mode.

Note 2: Don't change the contents of the registers when the start condition is generated, the stop condition is generated or the data transfer is in progress. Write data to the registers before the start condition is generated or during the period from when an INTSBI0 interrupt request is generated for stopping the data transfer until it is released.

Note 3: After a software reset is generated, all the bits of the SBI0CR2 except SBI0CR2<SBIM> and SBI0CR1, I2C0AR and SBI0SR2 are initialized.

Note 4: When the operation mode is switched to STOP, IDLE0 or SLOW mode, all the bits of SBI0CR2 except SBI0CR2<SBIM>, SBI0CR1, I2C0AR and SBI0DBR are initialized.

Serial bus interface data buffer register

SBI0DBR		7	6	5	4	3	2	1	0
(0x0025)	Bit Symbol	SBI0DBR							
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0

Note 1: Write the transmit data beginning with the most significant bit (bit 7).

Note 2: SBI0DBR has individual writing and reading buffers, and written data cannot be read. Therefore, SBI0DBR must not be accessed by using a read-modify-write instruction, such as a bit operation.

Note 3: Don't change the contents of the registers when the start condition is generated, the stop condition is generated or the data transfer is in progress. Write data to the registers before the start condition is generated or during the period from when an INTSBI0 interrupt request is generated for stopping the data transfer until it is released.

Note 4: To set SBI0CR2<PIN> to "1" by writing the dummy data to SBI0DBR, write "0x00". Writing any data other than "0x00" causes an improper value in the subsequently received data.

Note 5: When the operation mode is switched to STOP, IDLE0 or SLOW mode, all the bits of SBI0CR2 except SBI0CR2<SBIM>, SBI0CR1, I2C0AR and SBI0DBR are initialized.

18.5. Low Power Consumption Control

The SBI0 has a low power consumption register (POFFCR1) that saves power when the SBI0 is not being used.

Setting POFFCR1<SBI0EN> to "0" disables the basic clock supply to the SBI0 to save power. Note that this makes the SBI0 unusable. Setting POFFCR1<SBI0EN> to "1" enables the basic clock supply to the SBI0 and enables the SBI0 to operate.

After reset, POFFCR1<SBI0EN> is initialized to "0", and this makes the SBI0 unusable. When using the SBI0 for the first time, be sure to set POFFCR1<SBI0EN> to "1" in the initial setting of the program (before the SBI0 control registers are modified).

Do not change POFFCR1<SBI0EN> to "0" during the SBI0 operation, otherwise SBI0 may operate unexpectedly.

18.6. Functions

18.6.1. Selecting the slave address match detection and the GENERAL CALL detection

SBI0CR1<NOACK> enables and disables the slave address match detection and the GENERAL CALL detection in the slave mode.

Clearing SBI0CR1<NOACK> to "0" enables the slave address match detection and the GENERAL CALL detection.

Setting SBI0CR1<NOACK> to "1" disables the subsequent slave address match and GENERAL CALL detections. The slave addresses and GENERAL CALL sent from the master are ignored. No acknowledgment is returned and no INTSBI0 interrupt request is generated.

In the master mode, SBI0CR1<NOACK> is ignored and has no influence on the operation.

Note: When SBI0CR1<NOACK> is cleared to "0" during data transfer in the slave mode, it remains at "1" and returns an acknowledge bit of data transfer.

18.6.2. Selecting the number of clocks for data transfer and selecting the acknowledgment mode or non-acknowledgment mode

1-word data transfer consists of data and an acknowledge bit. When the data transfer is finished, an INTSBI0 interrupt request is generated.

SBI0CR1<BC> is used to select the number of bits of data to be transmitted/received subsequently. The acknowledgment mode is activated by setting SBI0CR1<ACK> to "1".

The master device generates the clocks for an acknowledge bit and outputs an acknowledgment in the receiver mode. The slave device counts the clocks for an acknowledge bit and outputs an acknowledgment in the receiver mode.

The non-acknowledgment mode is activated by setting SBI0CR1<ACK> to "0".

The master device does not generate the clocks for an acknowledge bit. The slave device does not count the clocks for an acknowledge bit.

18.6.2.1. Number of clocks for data transfer

The number of clocks for data transfer is set by using SBI0CR1<BC> and <ACK>.
The number of clocks for data length is set by SBI0CR1<BC>.

The acknowledgment mode is activated by setting SBI0CR1<ACK> to "1".

In the acknowledgment mode, the master device generates the clocks that correspond to the number of data bits, generates the clocks for an acknowledge bit, and generates an INTSBI0 interrupt request.

The slave device counts the clocks that correspond to the data bits, counts the clocks for an acknowledge bit, and generates an INTSBI0 interrupt request.

The non-acknowledgment mode is activated by setting SBI0CR1<ACK> to "0".

In the non-acknowledgment mode, the master device generates the clocks that correspond to the number of data bits, and generates an INTSBI0 interrupt request.

The slave device counts the clocks that correspond to the data bits, and generates an INTSBI0 interrupt request.

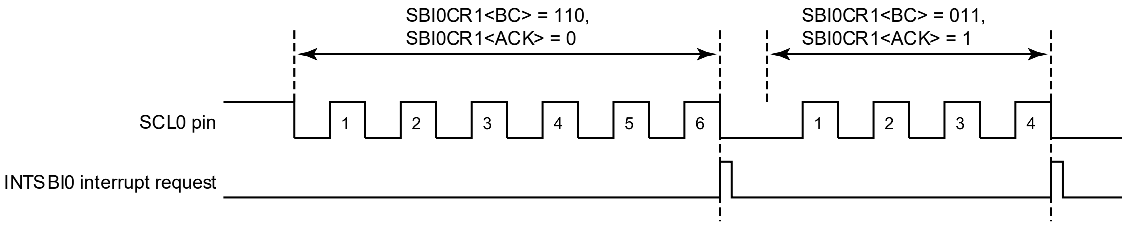


Figure 18-5 Number of Clocks for Data Transfer and SBI0CR1<BC> and <ACK>

The relationship between the number of clocks for data transfer and SBI0CR1<BC> and <ACK> is shown in Table 18-4.

Table 18-4 Relationship between the Number of Clocks for Data Transfer and SBI0CR1<BC> and <ACK>

<BC>	<ACK> = 0 (Non-acknowledgment mode)		<ACK> = 1 (Acknowledgment mode)	
	Number of clocks for data transfer	Number of data bits	Number of clocks for data transfer	Number of data bits
000	8	8	9	8
001	1	1	2	1
010	2	2	3	2
011	3	3	4	3
100	4	4	5	4
101	5	5	6	5
110	6	6	7	6
111	7	7	8	7

<BC> is cleared to "000" by the start condition.

Therefore, the slave address and the direction bit are always transferred in 8-bit units. In other cases, <BC> keeps the set value.

Note: SBI0CR1<ACK> must be set to "1" before transmitting or receiving a slave address. When SBI0CR1<ACK> is cleared to "0", the slave address match detection and the direction bit detection are not executed properly.

18.6.2.2. Output of an acknowledge bit

In the acknowledgment mode, the SDA0 pin changes as follows during the period of the clocks for an acknowledge bit.

- In the master mode

  In the transmitter mode, the SDA0 pin is released to receive an acknowledgment from the receiver during the period of the clocks for an acknowledge bit.

  In the receiver mode, the SDA0 pin is pulled down to the "Low" level and an acknowledgment is generated during the period of the clocks for an acknowledge bit.

- In the slave mode

  When a match between the received slave address and the slave address set to I2C0AR<SA> is detected or when the GENERAL CALL is received, the SDA0 pin is pulled down to the "Low" level and an acknowledgment is generated during the period of the clocks for an acknowledge bit.

  During the data transfer after the slave address match is detected or a "GENERAL CALL" is received in the transmitter mode, the SDA0 pin is released to receive an acknowledgment from the receiver during the period of the clocks for an acknowledge bit.

  In the receiver mode, the SDA0 pin is pulled down to the "Low" level and an acknowledgment is generated. Table 18-5 shows the states of the SCL0 and SDA0 pins in the acknowledgment mode.

Note: In the non-acknowledgment mode, the clocks for an acknowledge bit are not generated or counted, and thus no acknowledgment is output.

Table 18-5 States of the SCL0 and SDA0 Pins in the Acknowledgment Mode

Mode	Pin	Condition	Transmitter	Receiver
Master	SCL0	-	Add the clocks for an acknowledge bit.	Add the clocks for an acknowledge bit
	SDA0	-	Release the pin to receive an acknowledgment	Output "Low" level as an acknowledge bit to the pin
Slave	SCL0	-	Count the clocks for an acknowledge bit	Count the clocks for an acknowledge bit
	SDA0	When the slave address match is detected or a GENERAL CALL is received	-	Output "Low" level as an acknowledgment to the pin
		During transfer after the slave address match is detected or a GENERAL CALL is received	Release the pin to receive an acknowledgment	Output "Low" level as an acknowledgment to the pin

18.6.3. Serial clock

18.6.3.1. Clock source

SBI0CR1<SCK> is used to set the HIGH and LOW periods of the serial clock to be output in the master mode.

<SCK>	t_{HIGH} (m / fcgck)	t_{LOW} (n / fcgck)
	m	n
000	9	12
001	11	14
010	15	18
011	23	26
100	39	42
101	71	74
110	135	138
111	263	266

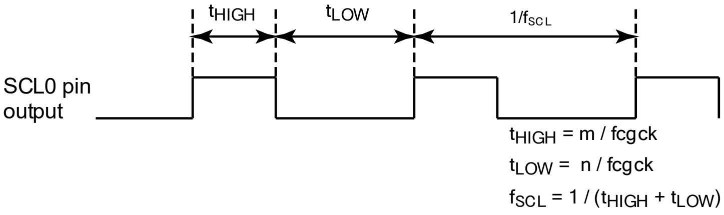


Figure 18-6 SCL0 Pin Output

Note: There are cases where the HIGH period differs from t_{HIGH} selected at SBI0CR1<SCK> when the rising edge of the SCL0 pin becomes blunt due to the load capacitor of the bus.

In the master mode, the hold time when the start condition is generated is t_{HIGH} [s] and the setup time when the stop condition is generated is t_{HIGH} [s].

When SBI0CR2<PIN> is set to "1" in the slave mode, the time that elapses before the release of the SCL pin is t_{LOW} [s].

In both the master and slave modes, the “High” level period must be 3 / fcgck [s] or longer and the “Low” level period must be 5 / fcgck [s] or longer for the externally input clock, regardless of the SBI0CR1<SCK> setting.

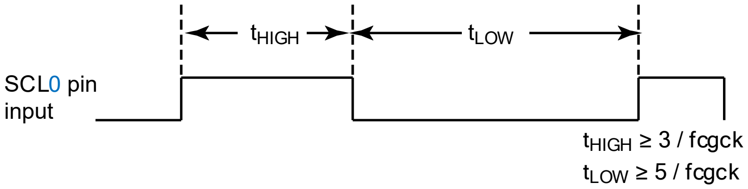


Figure 18-7 SCL0 Pin Input

18.6.3.2. Clock synchronization

In the I^{2}C bus, due to the structure of the pin, in order to drive a bus with a wired AND, a master device which pulls down a clock line to “Low” level will, in the first place, invalidate the clock of the other masters device which output “High” level clock.

Therefore, the master device which outputs “High” level clock must detects this to correspond to it.

The SBI0 has a clock synchronization function. This function ensures normal transfer when there are two or more masters on the same bus.

The example explains clock synchronization procedures when two masters simultaneously exist on a bus.

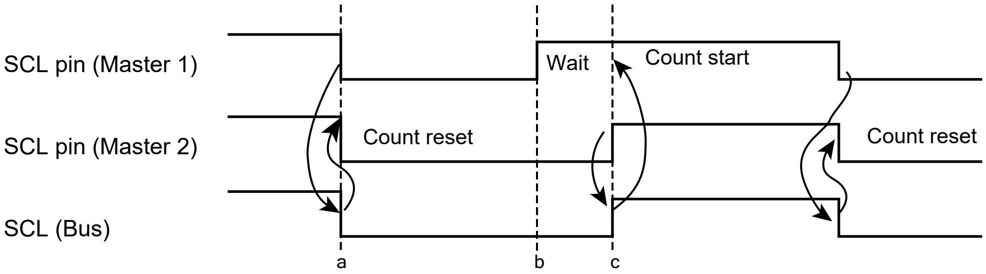


Figure 18-8 Example of Clock Synchronization

As Master 1 pulls down the SCL pin to the “Low” level at point "a", the SCL line of the bus becomes the “Low” level. After detecting this situation, Master 2 resets counting a clock width of “High” level and pulls down the SCL pin to the “Low” level.

Master 1 finishes counting a clock width of the “Low” level at point "b" and pulls up the SCL pin to the “High” level. Since Master 2 holds the SCL line of the bus at the “Low” level, Master 1 waits for counting a clock width of the “High” level. After Master 2 pulls up the SCL pin to the “High” level at point "c", Master 1 detects the SCL line of the bus as the “High” level, Master 1 starts counting a clock width of the “High” level. Then, Master 1 which has finished the counting a clock width of the “High” level pulls down the SCL pin to the “Low” level.

The clock pulse on the bus is determined by the master device with the shortest “High” level period and the master device with the longest “Low” level period from among those master devices connected to the bus.

18.6.4. Master/slave selection

To set a master device, SBI0CR2<MST> should be set to "1".

To set a slave device, SBI0CR2<MST> should be cleared to "0". When a stop condition on the bus or an arbitration lost is detected, SBI0CR2<MST> is cleared to "0" by the hardware.

18.6.5. Transmitter/receiver selection

To set the device as a transmitter, SBI0CR2<TRX> should be set to "1". To set the device as a receiver, SBI0CR2<TRX> should be cleared to "0".

For the I^{2}C bus data transfer in the slave mode, SBI0CR2<TRX> is set to "1" by the hardware when the direction bit (R/$\overline{\text{W}}$) sent from the master device is "1", and is cleared to "0" when the bit is "0".

In the master mode, after an acknowledgment is returned from the slave device, SBI0CR2<TRX> is cleared to "0" by hardware when a transmitted direction bit is "1", and is set to "1" by hardware when it is "0". When an acknowledgment is not returned, the current condition is maintained.

When a stop condition on the bus or an arbitration lost is detected, SBI0CR2<TRX> is cleared to "0" by the hardware. Table 18-6 shows SBI0CR2<TRX> changing conditions in each mode and SBI0CR2<TRX> value after changing.

Note: When SBI0CR1<NOACK> is "1", the slave address match detection and the GENERAL CALL detection are disabled, and thus SBI0CR2<TRX> remains unchanged.

Table 18-6 SBI0CR1<TRX> Operation in Each Mode

Mode	Direction bit (R/$\overline{\text{W}}$)	Change condition	<TRX> after changing
Slave mode	0	A received slave address is the same as the value set to I2C0AR<SA>	0
	1		1
Master mode	0	Acknowledgment is returned	1
	1		0

When the SBI0 operates in the free data format, a slave address and a direction bit (R/$\overline{\text{W}}$) are not recognized. They are handled as data just after generating the start condition. SBI0CR2<TRX> is not changed by the hardware.

18.6.6. Start/stop condition generation

When SBI0SR2<BB> is "0", a slave address and a direction bit which are set to the SBI0DBR beforehand are output on a bus after generating a start condition by writing "1" to SBI0CR2 <MST>, <TRX>, <BB> and <PIN>. It is necessary to set SBI0CR1<ACK> to "1" before generating the start condition.

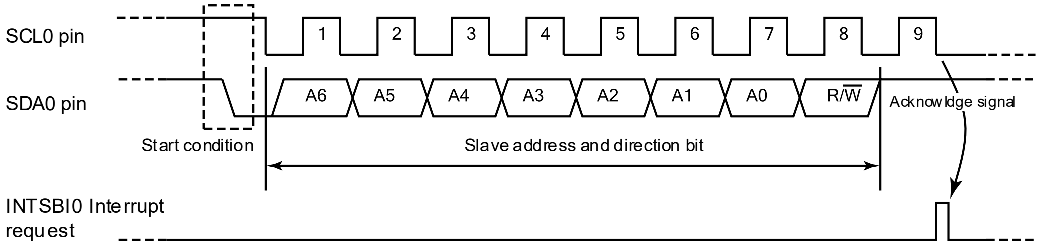


Figure 18-9 Generating the Start Condition and a Slave Address

When SBI0SR2<BB> is "1", the sequence of generating the stop condition on the bus is started by writing "1" to SBI0CR2<MST>, <TRX> and <PIN> and writing "0" to SBI0CR2<BB>.

When a stop condition is generated and the SCL line on a bus is pulled down to the “Low” level by other devices, a stop condition is generated after releasing the SCL line.

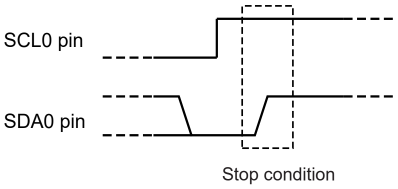


Figure 18-10 Stop Condition Generation

The bus condition can be indicated by reading the contents of SBI0SR2<BB>. SBI0SR2<BB> is set to "1" when the start condition on the bus is detected (Bus Busy State) and is cleared to "0" when the stop condition is detected (Bus Free State).

18.6.7. Interrupt service request and release

When a serial bus interface circuit is in the master mode and transferring a number of clocks set by SBI0CR1<BC> and <ACK> is complete, the INTSBI0 interrupt request is generated.

In the slave mode, the INTSBI0 interrupt request is generated when the above and following conditions are satisfied:

- At the end of the acknowledge bit when the received slave address matches to the value set by the I2C0AR<SA> with setting SBI0CR1<NOACK> to "0"
- At the end of the acknowledge bit when a GENERAL CALL is received with setting SBI0CR1<NOACK> to "0"
- At the end of transferring or receiving after matching of the slave address or receiving of GENERAL CALL

When the INTSBI0 interrupt request occurs, SBI0CR2<PIN> is cleared to "0". While the time that SBI0CR2<PIN> is "0", the SCL0 pin is pulled down to the “Low” level.

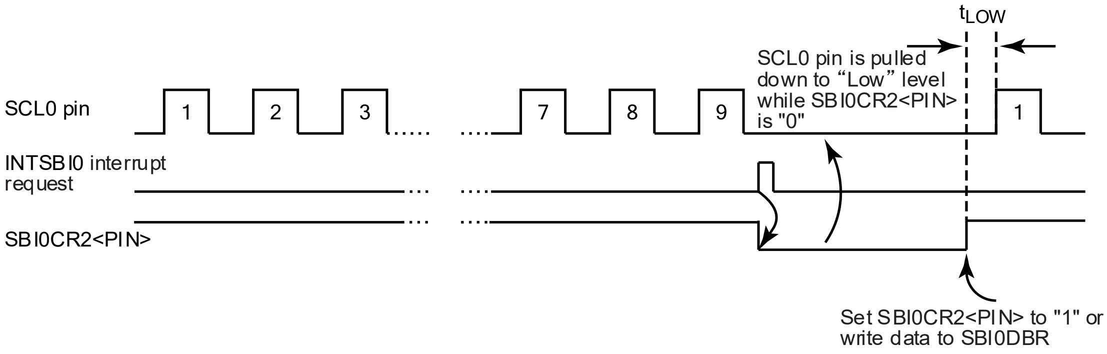

tLOW
SCL0 pin is pulled down to “Low” level while SBI0CR2<PIN> is "0"
SCL0 pin
1
2
3
7
8
9
1
INTSBI0 interrupt request
SBI0CR2<PIN>
Set SBI0CR2<PIN> to "1" or write data to SBI0DBR

Figure 18-11 SBI0CR2<PIN> and SCL0 Pin

Writing data to SBI0DBR sets SBI0CR2<PIN> to "1". The time from SBI0CR2<PIN> being set to "1" until the SCL0 pin is released takes t_{LOW}.

Although SBI0CR2<PIN> can be set to "1" by the software, SBI0CR2<PIN> cannot be cleared to "0" by the software.

18.6.8. Setting of serial bus interface mode

SBI0CR2<SBIM> is used to set the operation mode of the SBI0.

Setting SBI0CR2<SBIM> to "1" selects the serial bus interface mode. Setting it to "0" selects the port mode. Set SBI0CR2<SBIM> to "1" in order to set serial bus interface mode. Before setting of serial bus interface mode, confirm the SBI0 pins in a “High” level, and then, write "1" to SBI0CR2<SBIM>. And switch a port mode after confirming that a bus is free and set SBI0CR2<SBIM> to "0".

Note: When SBI0CR2<SBIM> is "0", no data can be written to SBI0CR2 except SBI0CR2<SBIM>. Before setting values to SBI0CR2, write "1" to SBI0CR2<SBIM> to activate the serial bus interface mode.

18.6.9. Software reset

The SBI0 has a software reset function that initializes the SBI0. When the SBI0 locks up, for example, due to noise, it can be initialized by using this function.

A software reset is generated by writing "10" and then "01" to SBI0CR2<SWRST>.

After a software reset is generated, the SBI0 is initialized and all the bits of SBI0CR2 register, except SBI0CR2<SBIM> and the SBI0CR1, I2C0AR and SBI0SR2, are initialized.

In addition a software reset, the bus should be initialized as needed.

18.6.10. Arbitration lost detection monitor

Since more than one master device can exist simultaneously on a bus, a bus arbitration procedure is implemented in order to guarantee the contents of transferred data.

Data on the SDA line is used for bus arbitration of the I^{2}C bus.

The following shows an example of a bus arbitration procedure when two master devices exist simultaneously on a bus. Master 1 and Master 2 output the same data until point "a". After that, when Master 1 outputs "1" and Master 2 outputs "0", since the SDA line of a bus is wired AND, the SDA line is pulled down to the “Low” level by Master 2. When the SCL line of a bus is pulled-up at point "b", the slave device reads data on the SDA line, that is data in Master 2. Data transmitted from Master 1 becomes invalid. The state in Master 1 is called "arbitration lost". A master device which loses arbitration releases the SDA pin and the SCL pin in order not to effect data transmitted from other masters which do not lose arbitration. When more than one master sends the same data at the first word, the bus arbitration continues after the second word.

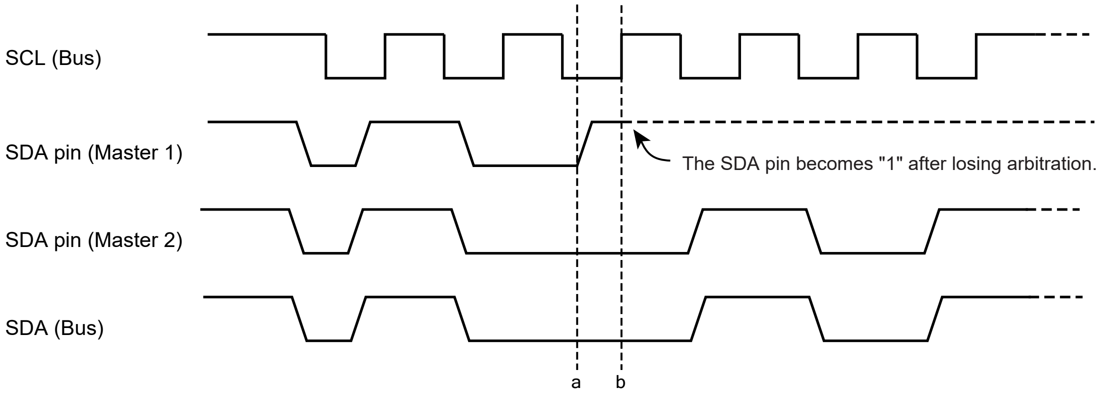


Figure 18-12 Arbitration Lost

The SBI0 compares levels of a SDA line on a bus with its SDA pin at the rising edge of the SCL line. When the levels are unmatched, arbitration is lost and SBI0SR2<AL> is set to "1".

When SBI0SR2<AL> is set to "1", SBI0CR2<MST> and <TRX> are cleared to "0" and the mode is switched to a slave receiver mode. Thus, the SBI0 stops outputting the clock pulses during data transfer after the SBI0SR2<AL> is set to "1". After the data transfer is completed, the INTSBI0 interrupt request occurs, SBI0CR2<PIN> is cleared to "0" and the SCL pin is pulled down to the low level.

SBI0SR2<AL> is cleared to "0" by writing data to the SBI0DBR, reading data from the SBI0DBR or writing data to the SBI0CR2.

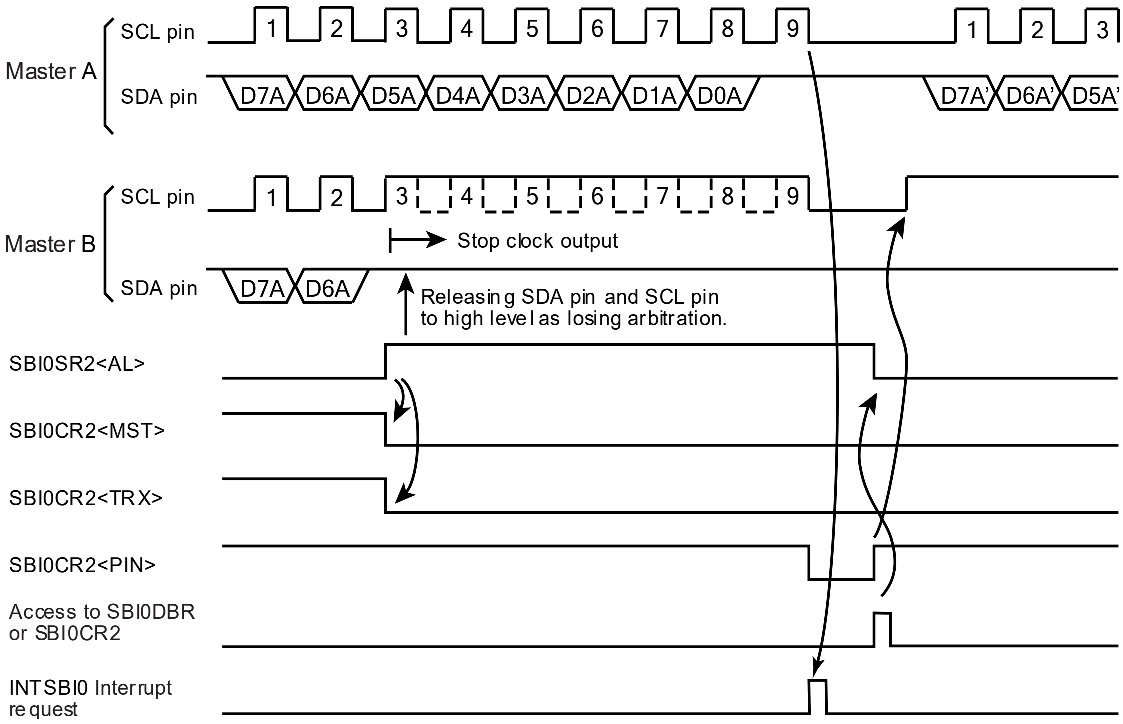


Figure 18-13 Example When Master B is a Serial Bus Interface Circuit

18.6.11. Slave address match detection monitor

In the slave mode, receiving GENERAL CALL or matching the received slave address with one setting by I2C0AR<SA> set SBI0SR2<AAS> to "1" when SBI0CR1<NOACK> is "0" and the I^{2}C bus mode is active (I2C0AR<ALS> = 0).

After setting SBI0CR1<NOACK> to "1" disables the subsequent slave address match and GENERAL CALL detections. Thus, SBI0SR2<AAS> remains at "0" when a GENERAL CALL is received or the same slave address as the I2C0AR<SA> set value is received.

When a SBI0 operates in the free data format (I2C0AR<ALS> = 1), SBI0SR2<AAS> is set to "1" after receiving the first 1-word of data.

SBI0SR2<AAS> is cleared to "0" by writing data to the SBI0DBR or reading data from the SBI0DBR.

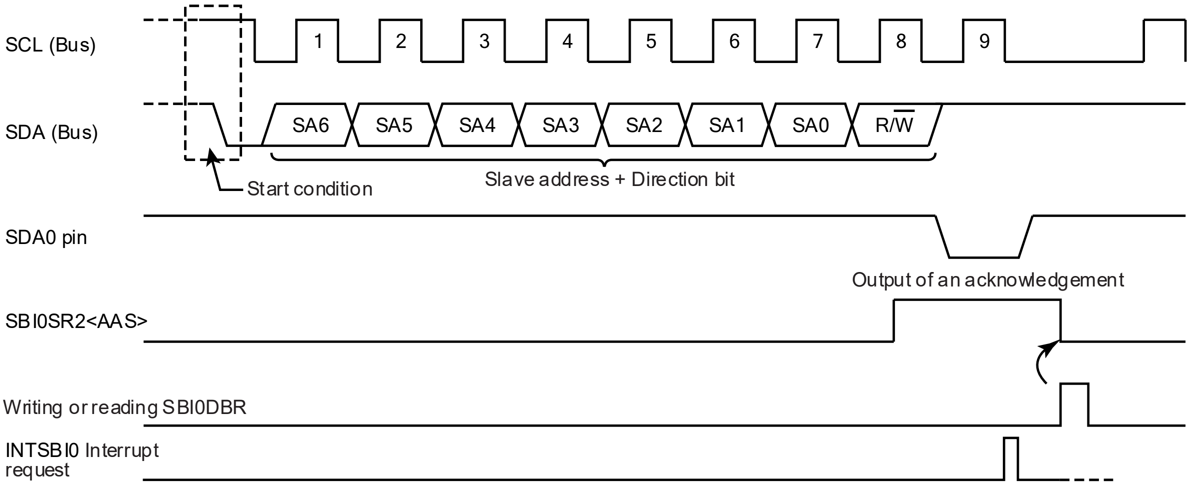

SCL (Bus)
1
2
3
4
5
6
7
8
9
SDA (Bus)
SA6
SA5
SA4
SA3
SA2
SA1
SA0
R/W̄
Start condition
Slave address + Direction bit
SDA0 pin
Output of an acknowledgement
SBI0SR2<AAS>
Writing or reading SBI0DBR
INTSBI0 Interrupt request

Figure 18-14 Changes in the Slave Address Match Detection Monitor

18.6.12. GENERAL CALL detection monitor

SBI0SR2<AD0> is set to "1" when SBI0CR1<NOACK> is "0" and GENERAL CALL (all 8-bit received data is "0" immediately after a start condition) in a slave mode.

Setting SBI0CR1<NOACK> to "1" disables the subsequent slave address match and GENERAL CALL detections. SBI0SR2<AD0> remains at "0" when a GENERAL CALL is received.

SBI0SR2<AD0> is cleared to "0" when a start or stop condition is detected on a bus.

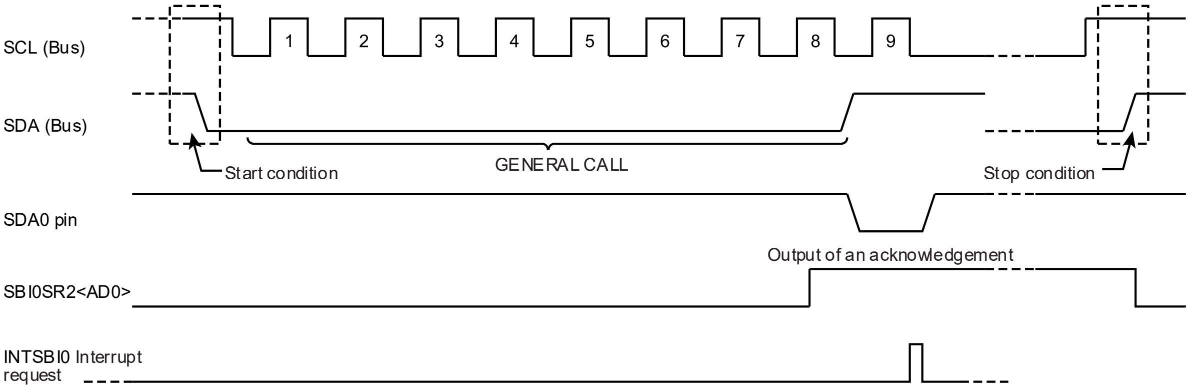


Figure 18-15 Changes in the GENERAL CALL Detection Monitor

18.6.13. Last received bit monitor

The SDA line value stored at the rising edge of the SCL line is set to SBI0SR2<LRB>.

In the acknowledge mode, immediately after an INTSBI0 interrupt request is generated, an acknowledgment is read by reading the contents of SBI0SR2<LRB>.

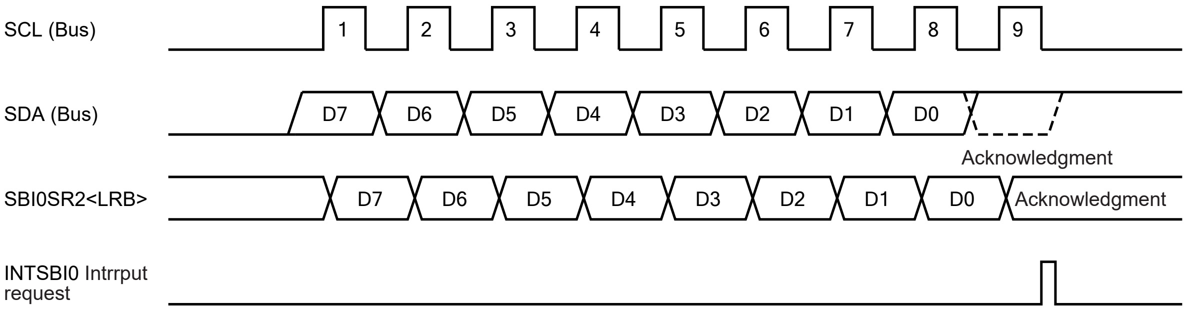


Figure 18-16 Changes in the Last Received Bit Monitor

18.6.14. Slave address and address recognition mode specification

When the SBI0 is used in the I^2C bus mode, clear I2C0AR<ALS> to "0", and set I2C0AR<SA> to the slave address.

When the serial bus interface circuit is used with a free data format not to recognize the slave address, set I2C0AR<ALS> to "1". With a free data format, the slave address and the direction bit are not recognized, and they are processed as data from immediately after the start condition.

18.7. Data Transfer of I²C Bus

18.7.1. Device initialization

Set POFFCR1<SBI0EN> to "1".

After confirming that the SBI0 pin is “High” level, set SBI0CR2<SBIM> to "1" to select the serial bus interface mode.

Set SBI0CR1<ACK> to "1", SBI0CR1<NOACK> to "0" and SBI0CR1<BC> to "000" to count the number of clocks for an acknowledge bit, to enable the slave address match detection and the GENERAL CALL detection, and set the data length to 8 bits. Set t_{HIGH} and t_{LOW} at SBI0CR1<SCK>.

Set a slave address at I2C0AR<SA> and set I2C0AR<ALS> to "0" to select the I²C bus mode.

Finally, set SBI0CR2<MST>, <TRX> and <BB> to "0", SBI0CR2<PIN> to "1" and SBI0CR2<SWRST> to "00" for specifying the default setting to a slave receiver mode.

Note: The initialization of the SBI0 must be complete within the time from all devices which are connected to a bus have initialized to and any device does not generate a start condition. When not, the data cannot be received correctly because the other device starts transferring before an end of the initialization of a serial bus interface circuit.

Example :Initialize a device

```
CHK_PORT:   LD      A, (P2PRD)           ; Check whether the SBI0 pin is at the “High” level
            AND     A, 0x18
            CMP     A, 0x18
            JR      NZ, CHK_PORT
            SET     (POFFCR1).SBI0EN     ; Set the SBI0 control to enable
            LD      (SBI0CR2), 0x18      ; Select the serial bus interface mode
            LD      (SBI0CR1), 0x16      ; Select the acknowledgment mode and sets SBI0CR1<SCK> to
                                           "110"
            LD      (I2C0AR), 0xA0       ; Set the slave address to "1010000" and selects the I²C bus mode
            LD      (SBI0CR2), 0x18      ; Select the slave receiver mode
```

18.7.2. Start condition and slave address generation

Confirm a bus free status (SBI0SR2<BB> = 0).

Set SBI0CR1<ACK> to "1" and specify a slave address and a direction bit to be transmitted to the SBI0DBR.

By writing "1" to SBI0CR2<MST>, <TRX>, <BB> and <PIN>, the start condition is generated on a bus and then, the slave address and the direction bit which are set to the SBI0DBR are output. The time from generating the START condition until the falling SCL0 pin takes t_{HIGH}.

The INTSBI0 interrupt request occurs at the 9th falling edge of a SCL clock, and SBI0CR2<PIN> is cleared to "0". The SCL0 pin is pulled down to the “Low” level while SBI0CR2<PIN> is "0". When the INTSBI0 interrupt request occurs, SBI0CR2<TRX> changes by the hardware according to the direction bit only when an acknowledgment is returned from the slave device.

Note 1: Do not write a slave address to the SBI0DBR while data is transferred. When data is written to the SBI0DBR, an unexpected data may be output..

Note 2: The bus free state must be confirmed by software within 98.0 [μs] (the shortest transmitting time according to the standard mode I²C bus standard) or 23.7 [μs] (the shortest transmitting time according to the fast mode I²C bus standard) after setting of the slave address and a direction bit to be output. Only when the bus free state is confirmed, set "1" to SBI0CR2<MST>, <TRX>, <BB> and <PIN> to generate the start conditions. When the writing of slave address and a direction bit and setting of SBI0CR2<MST>, <TRX>, <BB> and <PIN> doesn't finish within 98.0 [μs] or 23.7 [μs], the other masters may start the transferring and the slave address data written in SBI0DBR and a direction bit may be an unexpected values.

Example :Generate the start condition

```
CHK_BB:  TEST        (SBI0SR2).BB            ; Confirms that the bus is free
         JR          F, CHK_BB
         LD          (SBI0DBR), 0xCB         ; The transmission slave address "0x65" and the direction bit "1"
         LD          (SBI0CR2), 0xF8         ; Write "1" to SBI0CR2<MST>, <TRX>, <BB> and <PIN> to "1"
```

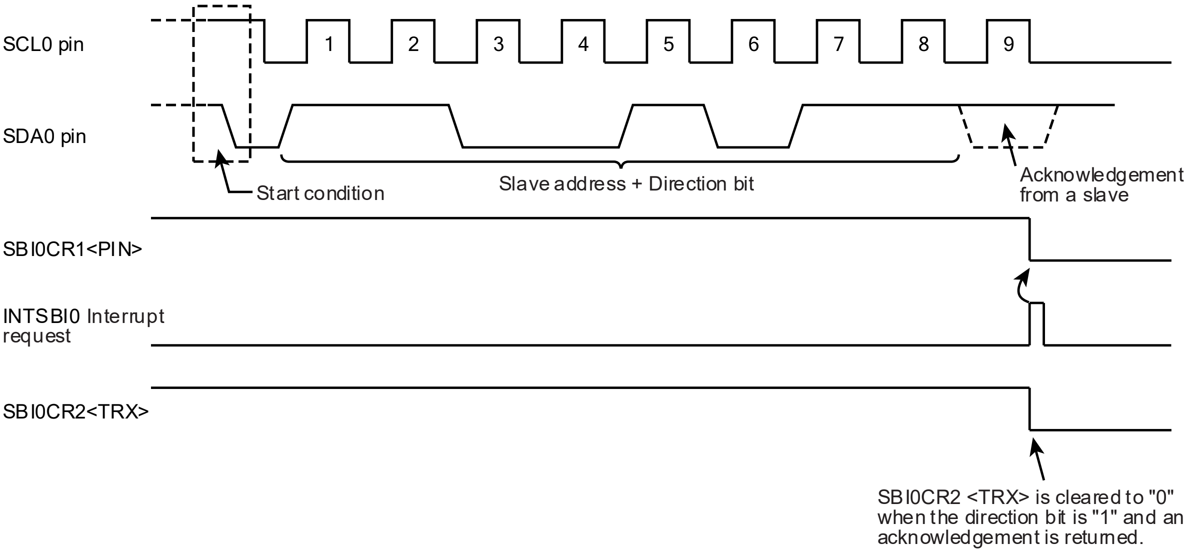

SCL0 pin
1
2
3
4
5
6
7
8
9
SDA0 pin
Start condition
Slave address + Direction bit
Acknowledgement from a slave
SBI0CR1<PIN>
INTSBI0 Interrupt request
SBI0CR2<TRX>
SBI0CR2 <TRX> is cleared to "0" when the direction bit is "1" and an acknowledgement is returned.

Figure 18-17 Generating the Start Condition and the Slave Address

18.7.3. 1-word data transfer

Check SBI0SR2<MST> in the interrupt process after a 1-word data transfer is completed, and determine whether the mode is a master or slave.

18.7.3.1. When SBI0SR2<MST> is "1" (Master mode)

Check SBI0SR2<TRX> and determine whether the mode is a transmitter or receiver.

(1) When SBI0SR2<TRX> is "1" (Transmitter mode)

Check SBI0SR2<LRB>. When SBI0SR2<LRB> is "1", a receiver does not request the next data. Implement the process to generate a stop condition and terminate data transfer.

When SBI0SR2<LRB> is "0", the receiver requests the next data. When the data to be transmitted subsequently is other than 8 bits, set SBI0CR1<BC>, set SBI0CR1<ACK> to "1", and write the transmit data to SBI0DBR.

After writing the data, SBI0CR2<PIN> becomes "1", the serial clock pulses are generated for transferring the next 1-word data from the SCL0 pin, and then the 1-word data is transmitted from the SDA0 pin.

After the data is transmitted, an INTSBI0 interrupt request occurs. SBI0CR2<PIN> become "0" and the SCL0 pin is pulled down to the “Low” level. When the data to be transferred is more than one word in length, repeat the procedure from the SBI0SR2<LRB> checking above.

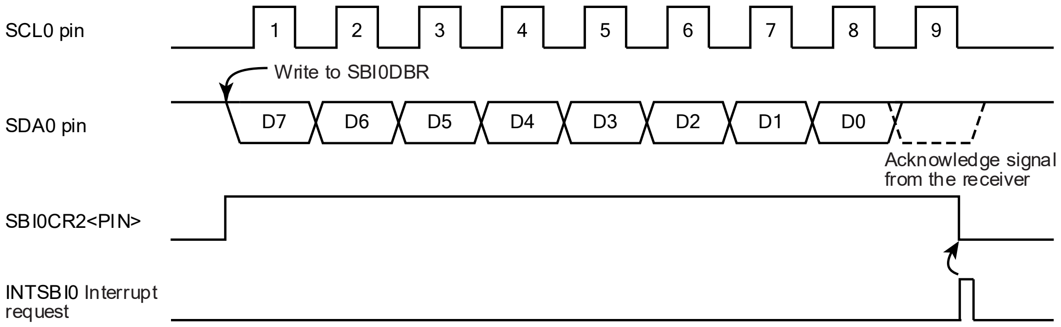


Figure 18-18 Example when SBI0CR1<BC> = 000 and SBI0CR1<ACK> = 1

(2) When SBI0SR2<TRX> is "0" (Receiver mode)

When the data to be transmitted subsequently is other than 8 bits, set SBI0CR1<BC>. Set SBI0CR1<ACK> to "1" and read the received data from the SBI0DBR (Reading data is undefined immediately after a slave address is sent).

After the data is read, SBI0CR2<PIN> becomes "1" by writing the dummy data "0x00" to the SBI0DBR. The SBI0 outputs a serial clock pulse to the SCL0 pin to transfer the next 1-word data and sets the SDA0 pin to “Low” level at the acknowledge bit timing.

The INTSBI0 interrupt request occurs and SBI0CR2<PIN> becomes "0".

Then the SBI0 outputs a clock pulse for 1-word data transfer and the acknowledge bit by writing dummy data “0x00” to the SBI0DBR or setting SBI0CR2<PIN> to "1" after reading the received data.

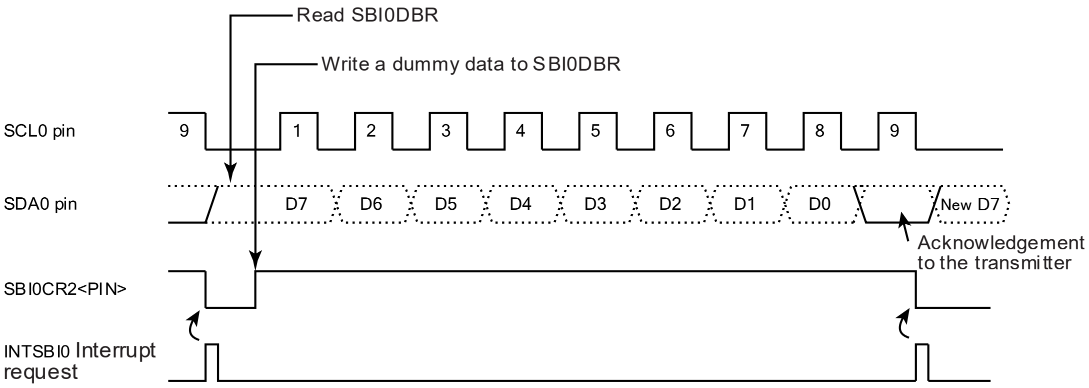

Read SBI0DBR
Write a dummy data to SBI0DBR
SCL0 pin
9
1
2
3
4
5
6
7
8
9
SDA0 pin
D7
D6
D5
D4
D3
D2
D1
D0
New D7
Acknowledgement to the transmitter
SBI0CR2<PIN>
INTSBI0 Interrupt request

Figure 18-19 Example when SBI0CR1<BC> = 000 and SBI0CR1<ACK> = 1

To make the transmitter terminate transmission, execute following procedure before receiving the last data.

1. Read the received data from SBI0DBR.
2. Clear SBI0CR1<ACK> to "0" and set SBI0CR1<BC> to "000".
3. To set SBI0CR2<PIN> to "1", write a dummy data "0x00" to SBI0DBR.

Transfer 1-word data in which no clock is generated for an acknowledge bit by setting SBI0CR2<PIN> to "1".

Next, execute following procedure.

1. Read the received data from SBI0DBR.
2. Clear SBI0CR1<ACK> to "0" and set SBI0CR1<BC> to "001".
3. To set SBI0CR2<PIN> to "1", write a dummy data "0x00" to SBI0DBR.

Transfer 1-bit data by setting SBI0CR2<PIN> to "1".

In this case, since the master device is a receiver, the SDA line on a bus keeps the “High” level. The transmitter receives the “High” level signal as a no acknowledgment. The receiver indicates to the transmitter that data transfer is complete.

After 1-bit data is received and an interrupt request has occurred, generate the stop condition to terminate data transfer.

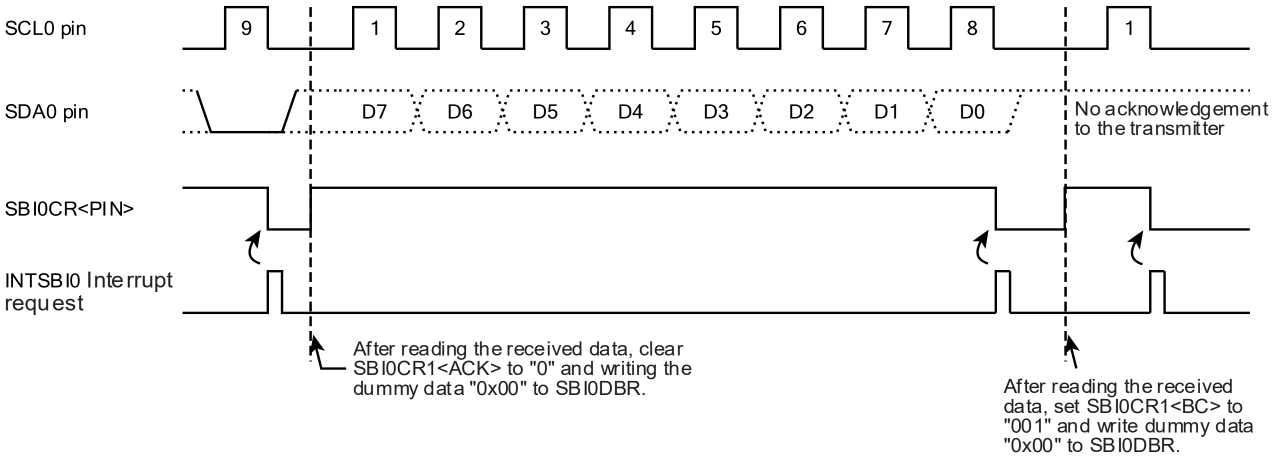

SCL0 pin
9
1
2
3
4
5
6
7
8
1
SDA0 pin
D7
D6
D5
D4
D3
D2
D1
D0
No acknowledgement to the transmitter
SBI0CR<PIN>
INTSBI0 Interrupt request
After reading the received data, clear SBI0CR1<ACK> to "0" and writing the dummy data "0x00" to SBI0DBR.
After reading the received data, set SBI0CR1<BC> to "001" and write dummy data "0x00" to SBI0DBR.

Figure 18-20 Termination of Data Transfer in the Master Receiver Mode

18.7.3.2. When SBI0SR2<MST> is "0" (Slave mode)

In the slave mode, the SBI0 operates either in the normal slave mode or in the slave mode after losing arbitration.

In the slave mode, the conditions of generating the INTSBI0 interrupt request are follows:

- At the end of the acknowledge bit when the received slave address matches the value set by the I2C0AR<SA> with SBI0CR1<NOACK> set at "0"
- At the end of the acknowledge bit when a "GENERAL CALL" is received with SBI0CR1<NOACK> set at "0"
- At the end of transferring or receiving after matching of slave address or receiving of GENERAL CALL

The serial bus interface circuit changes to the slave mode when arbitration is lost in the master mode. And an INTSBI0 interrupt request occurs when the word data transfer terminates after losing arbitration. The generation of the INTSBI0 interrupt request and the behavior of SBI0CR2<PIN> after losing arbitration are shown in Table 18-7.

Table 18-7 The Behavior of an INTSBI0 interrupt request and SBI0CR2<PIN> After Losing Arbitration

	When the Arbitration Lost Occurs during Transmission of Slave Address as a Master	When the Arbitration Lost Occurs during Transmission of Data as Master Transmitter
INTSBI0 interrupt request	An INTSBI0 interrupt request is generated at the termination of word-data transfer	
SBI0CR2<PIN>	SBI0CR2<PIN> is cleared to "0".	

When an INTSBI0 interrupt request occurs, SBI0CR2<PIN> is reset to "0", and the SCL0 pin is pulled down to “Low” level. Either writing data to the SBI0DBR or setting SBI0CR2<PIN> to "1" releases the SCL0 pin after taking t_{LOW}.

Check SBI0SR2<AL>, <TRX>, <AAS> and <AD0> and implement processes according to conditions in the slave mode listed in Table 18-8.

Table 18-8 Operation in the Slave Mode

SBI0SR2 <TRX>	SBI0SR2 <AL>	SBI0SR2 <AAS>	SBI0SR2 <AD0>	Condition	Process
1	1	1	0	The SBI0 loses arbitration during transmitting a slave address, and receives a slave address and a direction bit “1” which are sent by another master device and they are matched with SBI0’s them.	Set the number of bits in 1 word to SBI0CR1<BC> and write the transmit data to the SBI0DBR.
	0	1	0	In the slave receiver mode, the SBI0 receives a slave address and a direction bit “1” which are sent by another master device and they are matched with SBI0’s them.	
		0	0	In the slave transmitter mode, the SBI0 completes the transmission of 1-word data	Check SBI0SR2<LRB>. When it is set to "1", set SBI0CR2<PIN> to "1" since the receiver does not request subsequent data. Then, clear SBI0CR2<TRX> to "0" to release the bus. When SBI0SR2<LRB> is cleared to "0", set the number of bits in 1-word data to SBI0CR1<BC> and write the transmit data to SBI0DBR since the receiver requests next data.
0	1	1	1/0	The SBI0 loses arbitration during transmitting a slave address, and receives a slave address and a direction bit “0” which are sent by another master device and they are matched with SBI0’s them. Or, it receives GENERAL CALL.	Write the dummy data "0x00" to the SBI0DBR to set SBI0CR2<PIN> to "1". Or set SBI0CR2<PIN> to “1”.
		0	0	The SBI0 loses arbitration when transmitting a slave address or data. And the transferring 1-word data is completed.	The SBI0 is changed to the slave mode. Write the dummy data (0x00) to the SBI0DBR to clear SBI0SR2<AL> to "0" and set SBI0CR2<PIN> to "1". Or set SBI0CR2<PIN> to “1”.
	0	1	1/0	In the slave receiver mode, the SBI0 receives a slave address and a direction bit “0” which are sent by another master device and they are matched with SBI0’s them. Or, it receives GENERAL CALL.	Write the dummy data (0x00) to the SBI0DBR to set SBI0CR2<PIN> to "1". Or set SBI0CR2<PIN> to “1”.
		0	1/0	In the slave receiver mode, the SBI0 completes the reception of 1-word data	Set the number of bits of 1-word data to SBI0CR1<BC>, read the received data from the SBI0DBR and write the dummy data "0x00".

Note: Do not set I2C0AR to "0x00". In the slave mode, when the slave address set in I2C0AR<SA> is "0x00", a START Byte "0x01" in I^2C bus standard is received, the SBI0 detects the matching the received slave address and SBI0’s one and SBI0CR2<TRX> is set to "1".

18.7.4. Stop condition generation

When SBI0CR2<BB> is "1", a sequence of generating a stop condition is started by setting "1" to SBI0CR2<MST>, <TRX> and <PIN> and clearing SBI0CR2<BB> to "0". Do not modify the contents of SBI0CR2<MST>, <TRX>, <BB> and <PIN> until a stop condition is generated on a bus.

When a SCL line on a bus is pulled down by other devices, a SBI0 generates a stop condition after a SCL line is released.

The time from the releasing SCL line until the generating the STOP condition takes t_{HIGH}.

Example :Generate the stop condition

```
            LD (SBI0CR2), 0xD8       ; Sets SBI0CR2<MST>, <TRX> and <PIN> to "1" and SBI0CR2<BB> to "0"
CHK_BB:     TEST (SBI0SR2).BB        ; Waits until the bus is set free
            JR T, CHK_BB
```

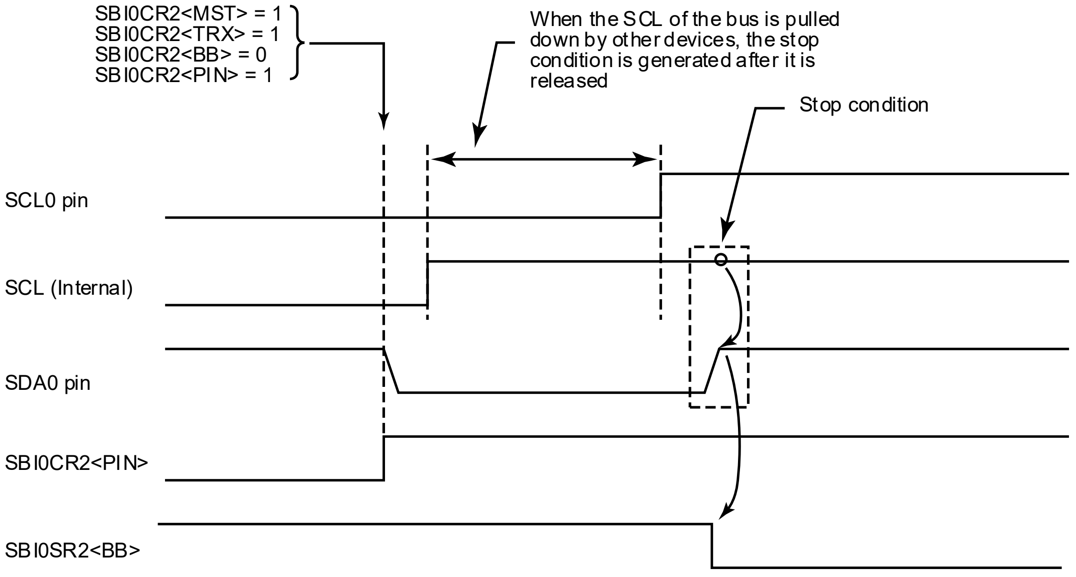


Figure 18-21 Stop Condition Generation

18.7.5. Restart

Restart is used to change the direction of data transfer between a master device and a slave device without completion of the transferring data.

The following explains how to restart by the SBI0.

Clear SBI0CR2<MST>, <TRX> and <BB> to "0" and set SBI0CR2 <PIN> to "1". The SDA0 pin retains the high level and the SCL0 pin is released.

Since this is not a stop condition, the bus is assumed to be in a busy state from other devices.

Check SBI0SR2<BB> until it becomes "0" to check that the SCL0 pin of the SBI0 is released.

Check SBI0SR2<LRB> until it becomes "1" to check that the SCL line on the bus is not pulled down to the “Low” level by other devices.

After confirming that the bus stays in a free state, generate a start condition in the procedure "18.7.2. Start condition and slave address generation".

In order to meet the setup time at a restart, take at least 4.7 [μs] of waiting time by the software in the standard mode in the I^2C bus standard or at least 0.6 [μs] of waiting time in the fast mode in the I^2C bus standard from the time of restarting to confirm that a bus is free until the time to generate a start condition.

Note: When the master is in the receiver mode, it is necessary to stop the data transmission from the slave device before the STOP condition is generated. To stop the transmission, the master device make the slave device receiving a no acknowledgment. Therefore, SBI0SR2<LRB> is "1" before generating the Restart and it cannot be confirmed that SCL line is not pulled down by other devices. Please confirm the SCL line state by reading the port.

Example :Generate a restart

```
            LD      (SBI0CR2), 0x18     ; Sets SBI0CR2<MST>, <TRX> and <BB> to "0" and SBI0CR2<PIN> to "1"
CHK_BB:     TEST    (SBI0SR2).BB        ; Waits until SBI0SR2<BB> becomes "0"
            JR      T, CHK_BB
CHK_LRB:    TEST    (SBI0SR2).LRB       ; Waits until SBI0SR2<LRB> becomes "1"
            JR      F, CHK_LRB
                    :
                    :                   ; Wait time process by the software
                    :
            LD      (SBI0CR2), 0xF8     ; Sets SBI0CR2<MST>, <TRX>, <BB> and <PIN> to "1"
```

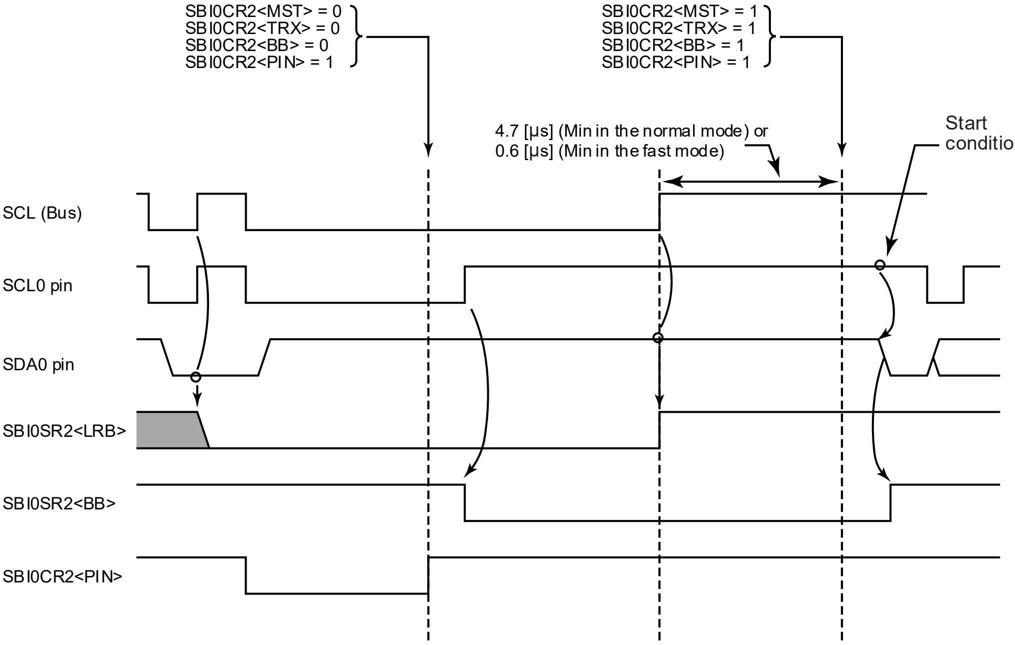


Figure 18-22 Timing Diagram When Restarting

18.8. AC Specifications

The AC specifications are as listed below.

The transfer mode (standard mode ore fast mode) should be selected suitable for frequency of fcgck. For these operating mode, refer to Table 18-9.

Table 18-9 AC Specifications (Output Timing)

Parameter	Symbol	Standard mode		Fast mode		Unit
		MIN	MAX	MIN	MAX	
SCL clock frequency	t_{SCL}	0	fcgck / (m+n)	0	fcgck / (m+n)	kHz
Hold time (re)start condition. This period is followed by generation of the first clock plus	$t_{HD;STA}$	m / fcgck	-	m / fcgck	-	μs
“Low” level period of SCL clock (output)	t_{LOW}	n / fcgck	-	n / fcgck	-	μs
“High” level period of SCL clock (output)	t_{HIGH}	m / fcgck	-	m / fcgck	-	μs
“Low” level period of SCL clock (input)	t_{LOW}	5 / fcgck	-	5 / fcgck	-	μs
“High” level period of SCL clock (input)	t_{HIGH}	3 / fcgck	-	3 / fcgck	-	μs
Restart condition setup time	$t_{SU;STA}$	Depends on the software	-	Depends on the software	-	μs
Data hold time	$t_{HD;DAT}$	0	5 / fcgck	0	5 / fcgck	μs
Data setup time	$t_{SU;DAT}$	250	-	100	-	ns
Rising time of SDA and SCL signals	t_r	-	1000	-	300	ns
Falling time of SDA and SCL signals	t_f	-	300	-	300	ns
Stop condition setup time	$t_{SU;STO}$	m / fcgck	-	m / fcgck	-	μs
Bus free time between the stop condition and the start condition	t_{BUF}	Depends on the software	-	Depends on the software	-	μs
Time before rising of SCL0 pin after SBI0CR2<PIN> is changed from "0" to "1"	$t_{SU;SCL}$	n / fcgck	-	n / fcgck	-	μs

Note: For m and n, refer to"18.6.3.1. Clock source".

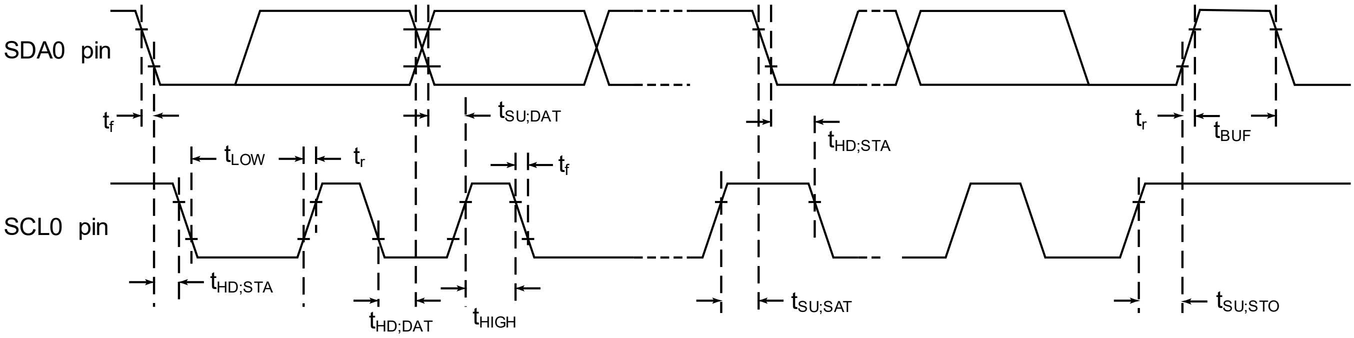


Figure 18-23 Definition of Timing (No. 1)

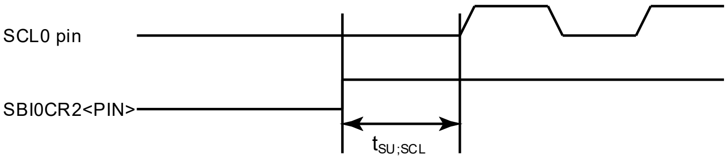


Figure 18-24 Definition of Timing (No. 2)

19. Key-on Wakeup (KWU)

The key-on wakeup is a function for releasing the STOP mode at the $\overline{\text{STOP}}$ pin or at pins KWI7 to KWI0.

19.1. Configuration

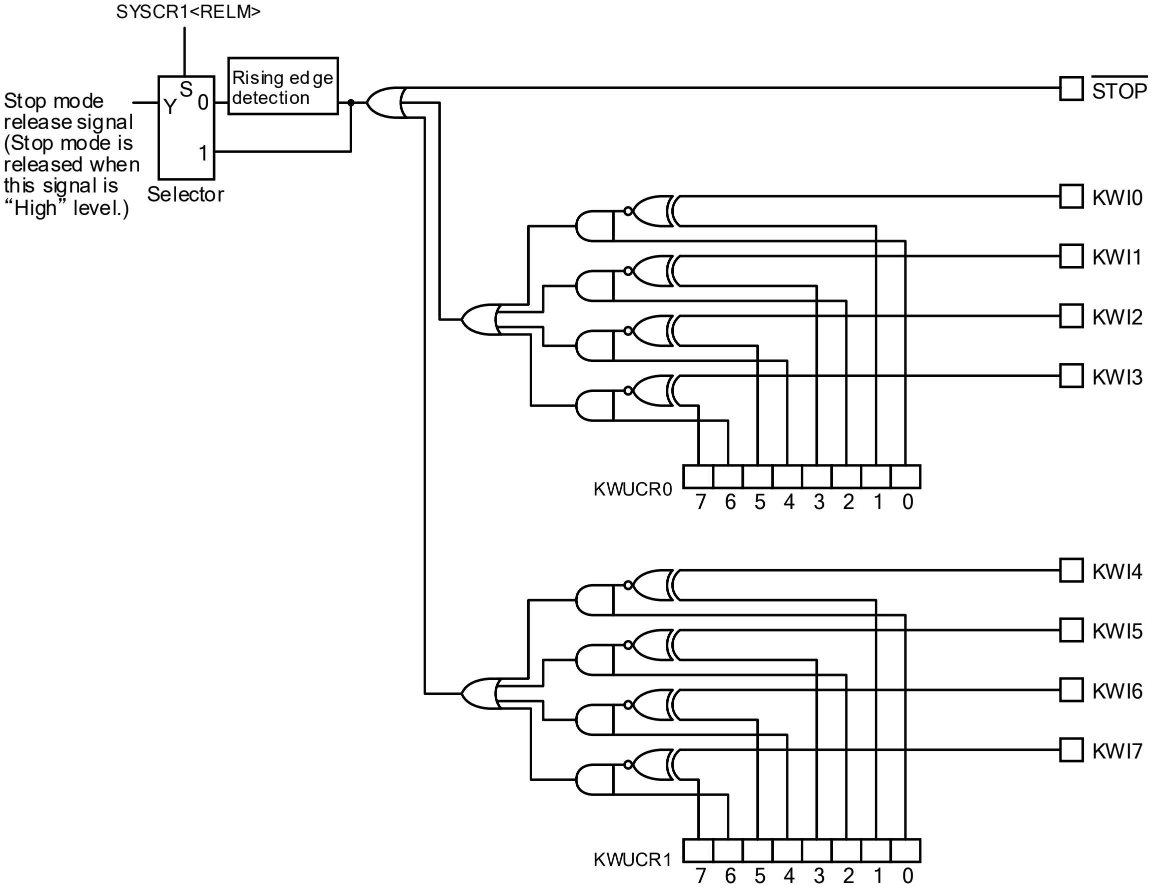


Figure 19-1 Key-on Wakeup Circuit

Note: The key-on wakeup pins KWI7 to KWI0 are also used as input/output ports P47 to P40. To operate them as key-on wakeup, set the registers of input/output ports. Refer to "8. I/O Ports".

19.2. Control

Key-on wakeup control registers (KWUCR0 and KWUCR1) can be configured to designate the key-on wakeup pins (KWI7 to KWI0) as STOP mode release pins and to specify the STOP mode release levels of each of these designated pins.

Key-on wakeup control register 0

KWUCR0		7	6	5	4	3	2	1	0
(0x0FC4)	Bit Symbol	KW3LE	KW3EN	KW2LE	KW2EN	KW1LE	KW1EN	KW0LE	KW0EN
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

KW3LE	STOP mode release level of KWI3 pin	0: 1:	“Low” level “High” level
KW3EN	Input enable/disable control of KWI3 pin	0: 1:	Disable Enable
KW2LE	STOP mode release level of KWI2 pin	0: 1:	“Low” level “High” level
KW2EN	Input enable/disable control of KWI2 pin	0: 1:	Disable Enable
KW1LE	STOP mode release level of KWI1	0: 1:	“Low” level “High” level
KW1EN	Input enable/disable control of KWI1 pin	0: 1:	Disable Enable
KW0LE	STOP mode release level of KWI0 pin	0: 1:	“Low” level “High” level
KW0EN	Input enable/disable control of KWI0 pin	0: 1:	Disable Enable

Key-on wakeup control register 1

KWUCR1		7	6	5	4	3	2	1	0
(0x0FC5)	Bit Symbol	KW7LE	KW7EN	KW6LE	KW6EN	KW5LE	KW5EN	KW4LE	KW4EN
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

KW7LE	STOP mode release level of KWI7 pin	0: 1:	“Low” level “High” level
KW7EN	Input enable/disable control of KWI7 pin	0: 1:	Disable Enable
KW6LE	STOP mode release level of KWI6 pin	0: 1:	“Low” level “High” level
KW6EN	Input enable/disable control of KWI6 pin	0: 1:	Disable Enable
KW5LE	STOP mode release level of KWI5	0: 1:	“Low” level “High” level
KW5EN	Input enable/disable control of KWI5 pin	0: 1:	Disable Enable
KW4LE	STOP mode release level of KWI4 pin	0: 1:	“Low” level “High” level
KW4EN	Input enable/disable control of KWI4 pin	0: 1:	Disable Enable

19.3. Functions

By using the key-on wakeup function, the STOP mode can be released at a $\overline{\text{STOP}}$ pin or at KWIm pin (m = 7 to 0).

After releasing reset, the STOP mode releasing function of the KWIm pins are disabled. Therefore, to assign the KWIm pin as a STOP mode release pin, it is necessary to set <KWmEN> (m = 7 to 0) in KWUCRn (n = 1 to 0) to “1”.

Because the $\overline{\text{STOP}}$ pin has no function for disabling, it is assigned as a STOP mode release pin, irrespective of whether the key-on wakeup function is used or not.

- Setting KWUCRn, P4PU and P4CR

  To assign key-on wakeup pin (KWIm) as a STOP mode release pin, set KWUCRn<KWmEN> to "1". Then KWIm pin whose KWUCRn<KWmEN> is set to "1" can be specified the STOP mode release level. When KWUCRn<KWmLE> is cleared to "0", the “Low” level to KWIm pin makes the STOP mode releasing. When it is set to “1”, the “High” level makes the STOP mode releasing.

  For example, the STOP mode wish to be released when the “High” level is input to KWI0 pin, Set KWUCR0<KW0EN> and KWUCR0<KW0LE> are set to “1”.

  Each KWIm pin can be connected to built-in pull-up resistors in the I/O port. Before connecting to built-in pull-up resistors, the corresponding bits in the pull-up control register (P4PU) at port P4 must be set to "1".

  Finally, to make each KWIm pin as input mode, the corresponding bits in the input/output control register (P4CR) at port P4 must be set to "0".

- Starting STOP mode

  To start the STOP mode, set SYSCR1<RELM> to "1" (level release mode), and SYSCR1<STOP> to "1".

  To use the key-on wakeup function, do not set SYSCR1<RELM> to "0" (edge release mode). When the key-on wakeup function is used in edge release mode, STOP mode cannot be released, although a rising edge is input into the $\overline{\text{STOP}}$ pin. This is because the KWIm pin enabling inputs to be received is at a release level after the STOP mode starts.

- Releasing STOP mode

  To release STOP mode, input a “High” level signal into the $\overline{\text{STOP}}$ pin or input a specific release level into the KWIm pin for which receipt of inputs is enabled. To release STOP mode not by $\overline{\text{STOP}}$ pin, but by the KWIm pin, continue inputting a “Low” level into the $\overline{\text{STOP}}$ pin throughout the period from when the STOP mode is started to when it is released.

  When the STOP mode starts and the $\overline{\text{STOP}}$ pin or KWIm pin is already at a release level, the following instruction will be executed without starting the STOP mode (with no warm-up performed).

Note: When an analog voltage is applied to KWIm pin for which receipt of inputs is enabled by the key-on wakeup control register (KWUCRn) setting, a through current will flow. In this case, the analog voltage should be not applied to this pin.

Table 19-1 STOP Mode Release

Pin name	SYSCR1<RELM> = 1 (level release mode)		SYSCR1<RELM> = 0 (edge release mode)
	KWUCRn<KWmLE> = 0	KWUCRn<KWmLE> = 1	
$\overline{\text{STOP}}$	“High” level		Rising edge
KWIm	“Low” level	“High” level	Don't use

Note: m = 7 to 0, n = 1 to 0

Example :A case in which STOP mode is started with the release level of the $\overline{\text{STOP}}$ pin set to a “High” level and the release level of KWI0 pin set to a “Low” level (connected to an internal pull-up resistor of the KWI0 pin)

```
DI                        ; <IMF> ← "0"
SET  (P4PU).0             ; KWI0 (P40) connected to a pull-up resistor
LD   (KWUCR0), 0x01       ; the KWI0 pin is set to enable inputs, and its release level is set to a “Low” level
LD   (SYSCR1), 0xA0       ; Starting in level release mode
```

20. 10-bit AD Converter (ADC)

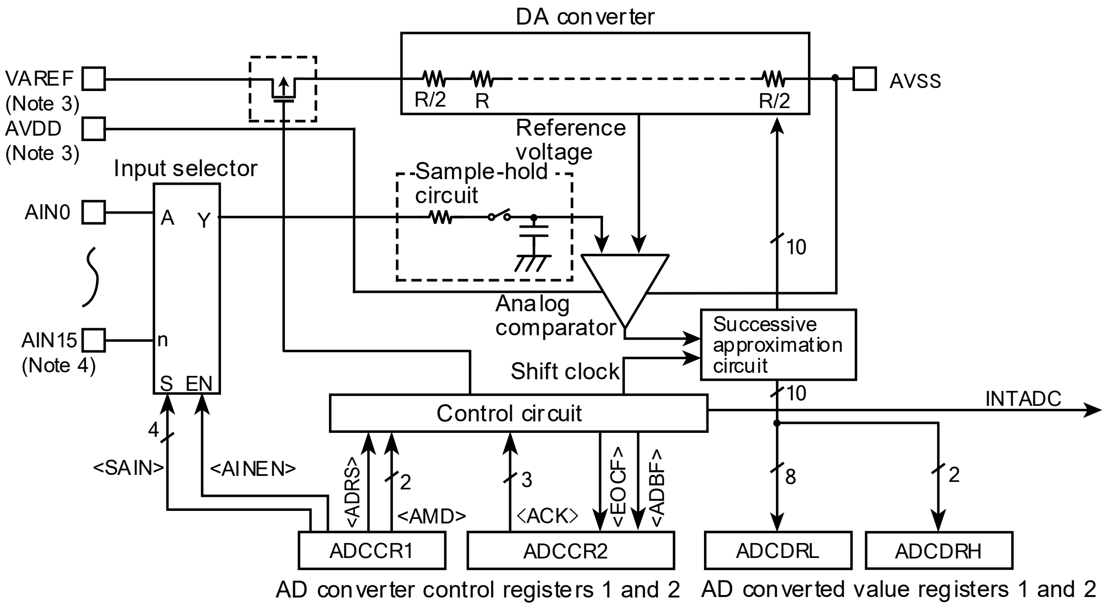
The TMP89FS60B/62B/63B has a 10-bit successive approximation type AD converter.

This chapter describes the ADC with VAREF pin and AVDD pin. The ADC for the TMP89FS62B, replace VAREF pin and AVDD pin as VAREF/AVDD pin.

20.1. Configuration

The circuit configuration of the 10-bit AD converter is shown in Figure 20-1.

It consists of control registers ADCCR1 and ADCCR2, converted value registers ADCDRL and ADCDRH, a DA converter, a sample-hold circuit, a comparator, a successive comparison circuit, etc.



Note 1: Before using the AD converter, set an appropriate value to the I/O port register which is also used as an analog input port. For details, refer to "8. I/O Ports".

Note 2: The DA converter current (I_{REF}) is automatically cut off at times other than during AD conversion.

Note 3: The same pin is assigned as VAREF pin and AVDD pin for the TMP89FS62B.

Note 4: The number of analog input pins is 8 (AIN7 to 0) for the TMP89FS62B. One is 13 (AIN12 to 0) for the TMP89FS63B.

Figure 20-1 10-bit AD Converter

20.2. Control

The AD converter consists of the following registers:

(1) AD converter control register 1 (ADCCR1)

This register selects an analog channel in which to perform AD conversion, selects an AD conversion operation mode, and controls the start of the AD converter.

(2) AD converter control register 2 (ADCCR2)

This register selects the AD conversion time, and monitors the operating status of the AD converter.

(3) AD converted value registers (ADCDRH and ADCDRL)

These registers store the digital values generated by the AD converter

AD converter control register 1

ADCCR1		7	6	5	4	3	2	1	0
(0x0034)	Bit Symbol	ADRS	AMD		AINEN	SAIN			
	Read/Write	R/W	R/W		R/W	R/W			
	After reset	0	0	0	0	0	0	0	0

ADRS	AD conversion start	0:	-		
		1:	AD conversion start		
AMD	AD operating mode	00:	AD operation disable, forcibly stop AD operation		
		01:	Single mode		
		10:	Reserved		
		11:	Repeat mode		
AINEN	Analog input control	0:	Analog input disable		
		1:	Analog input enable		
SAIN	Analog input channel select		TMP89FS60B	TMP89FS62B	TMP89FS63B
		0000:	AIN0	AIN0	AIN0
		0001:	AIN1	AIN1	AIN1
		0010:	AIN2	AIN2	AIN2
		0011:	AIN3	AIN3	AIN3
		0100:	AIN4	AIN4	AIN4
		0101:	AIN5	AIN5	AIN5
		0110:	AIN6	AIN6	AIN6
		0111:	AIN7	AIN7	AIN7
		1000:	AIN8	Reserved	AIN8
		1001:	AIN9	Reserved	AIN9
		1010:	AIN10	Reserved	AIN10
		1011:	AIN11	Reserved	AIN11
		1100:	AIN12	Reserved	AIN12
		1101:	AIN13	Reserved	Reserved
		1110:	AIN14	Reserved	Reserved
		1111:	AIN15	Reserved	Reserved

Note 1: Do not perform the following operations on the ADCCR1 register while AD conversion is being executed (ADCCR2<ADBF> = 1).

- Changing <SAIN>
- Setting <AINEN> to "0"
- Changing <AMD> (except a forced stop by setting <AMD> to "00")
- Setting <ADRS> to "1"

Note 2: When all analog input channels are disabled, set <AINEN> to "0".

Note 3: Although analog input pins are also used as input/output ports, to maintain the accuracy of AD conversion, do not execute input/output instructions during AD conversion. Additionally, do not input widely varying signals into the ports adjacent to analog input pins.

Note 4: When the operation mode is changed to STOP, IDLE0 or SLOW mode, <ADRS>, <AMD> and <AINEN> are initialized to "0". When the AD converter is used after returning to NORMAL mode, set <ADRS>, <AMD> and <AINEN> again.

Note 5: After the start of AD conversion, <ADRS> is automatically cleared to "0" ("0" is read).

AD converter control register 2

ADCCR2		7	6	5	4	3	2	1	0
(0x0035)	Bit Symbol	EOCF	ADBF	-	-	"0"	ACK		
	Read/Write	R	R	R	R	W	R/W		
	After reset	0	0	0	0	0	0	0	0

EOCF	AD conversion end flag	0: 1:	Before conversion or during conversion Conversion end
ADBF	AD conversion BUSY flag	0: 1:	AD conversion being halted AD conversion being executed
ACK	AD conversion time select ion (examples of AD conversion time are shown in Table 20-1)	000: 001: 010: 011: 100: 101: 110: 111:	39 / fcgck 78 / fcgck 156 / fcgck 312 / fcgck Reserved Reserved Reserved Reserved

Note 1: Modify the <ACK> when AD conversion is halted (ADCCR2<ADBF> = 0).

Note 2: The bit 3 of ADCCR2 must be cleared to “0”..

Note 3: When the operation mode changes to STOP, IDLE0 or SLOW mode, <EOCF> and <ADBF> are initialized to "0".

Note 4: When the AD converted value register (ADCDRH) is read, <EOCF> is cleared to "0". It is also cleared to "0" when AD conversion is started (ADCCR1<ADRS> = 1) without reading ADCDRH after completing AD conversion in single mode.

Note 5: When a read instruction is executed on ADCCR2, bits 5 to 3 are read as "0".

Table 20-1 <ACK> Settings and Conversion Times Relative to Frequencies

		Frequency (fcgck)						
Condition <ACK>	Conversion time	10 [MHz]	8 [MHz]	5 [MHz]	4 [MHz]	2.5 [MHz]	2 [MHz]	1 [MHz]
000	39 / fcgck	-	-	-	-	15.6 [μs]	19.5 [μs]	39.0 [μs]
001	78 / fcgck	-	-	15.6 [μs]	19.5 [μs]	31.2 [μs]	39.0 [μs]	-
010	156 / fcgck	15.6 [μs]	19.5 [μs]	31.2 [μs]	39.0 [μs]	-	-	-
011	312 / fcgck	31.2 [μs]	39.0 [μs]	-	-	-	-	-
1**	Reserved							

Note 1: fcgck: Gear clock [Hz]

Note 2: The condition shown by “-“ in the above table is prohibited.

Note 3: Above conversion times do not include the time shown below.

- Time from when ADCCR1<ADRS> is set to 1 to when AD conversion is started
- Time from when AD conversion is finished to when a converted value is stored in ADCDRL and ADCDRH.

These times are 10 / fcgck (s) (max) when ADCCR2<ACK> = 00* and are 32 / fcgck (s) (max) when ADCCR2<ACK> = 01*.

AD converted value register (lower side)

ADCDRL (0x0036)		7	6	5	4	3	2	1	0
	Bit Symbol	AD07	AD06	AD05	AD04	AD03	AD02	AD01	AD00
	Read/Write	R	R	R	R	R	R	R	R
	After reset	0	0	0	0	0	0	0	0

AD converted value register (upper side)

ADCDRH (0x0037)		7	6	5	4	3	2	1	0
	Bit Symbol	-	-	-	-	-	-	AD09	AD08
	Read/Write	R	R	R	R	R	R	R	R
	After reset	0	0	0	0	0	0	0	0

Note 1: ADCDRL or ADCDRH must be read after the INTADC interrupt is generated or after ADCCR2<EOCF> becomes "1".

Note 2: In single mode, do not read ADCDRL or ADCDRH during AD conversion (ADCCR2<ADBF> = 1). (When AD conversion is finished in the time from reading ADCDRL to reading ADCDRH, the INTADC interrupt request is canceled, and the conversion result is lost.)

Note 3: When the operation mode is changed to STOP, IDLE0 or SLOW mode, ADCDRL and ADCDRH are initialized to 0x00.

Note 4: When ADCCR1<AMD> is set to "00", ADCDRL and ADCDRH are initialized to 0x00.

Note 5: When a read instruction is executed on ADCDRH, bits 7 to 2 are read as "0".

Note 6: In the repeat mode, when AD conversion is finished in the time from reading ADCDRL to reading ADCDRH, the previous converted value is retained without overwriting to the AD converted value register. In this case, the INTADC interrupt request is canceled, and the conversion result without overwriting is lost.

20.3. Functions

The 10-bit AD converter operates in either single mode in which AD conversion is performed only once or repeat mode in which AD conversion is performed repeatedly.

20.3.1. Single mode

In single mode, the voltage at a designated analog input pin is AD converted only once.

Setting ADCCR1<ADRS> to "1" after setting ADCCR1<AMD> to "01" enables AD conversion to start. ADCCR1<ADRS> is automatically cleared after the completion of AD conversion. As AD conversion starts, ADCCR2<ADBF> is set to "1". It is cleared to "0" when AD conversion is completed or when AD conversion is forced to stop.

After AD conversion is completed, the conversion result is stored in the AD converted value registers (ADCDRL and ADCDRH), ADCCR2<EOCF> is set to "1", and the AD conversion finished interrupt (INTADC) is generated. The AD converted value registers (ADCDRL and ADCDRH) should be usually read according to the INTADC interrupt processing routine. When the upper side (ADCDRH) of the AD converted value register is read, ADCCR2<EOCF> is cleared to "0".

Note: Do not perform the following operations on the ADCCR1 register when AD conversion is being executed (ADCCR2<ADBF> = 1). When the following operations are performed, there is the possibility that AD conversion may not be executed properly.

- Changing the ADCCR1<SAIN> setting
- Setting ADCCR1<AINEN> to "0"
- Changing the ADCCR1<AMD> setting (except a forced stop by setting <AMD> to "00")
- Setting ADCCR1<ADRS> to "1"

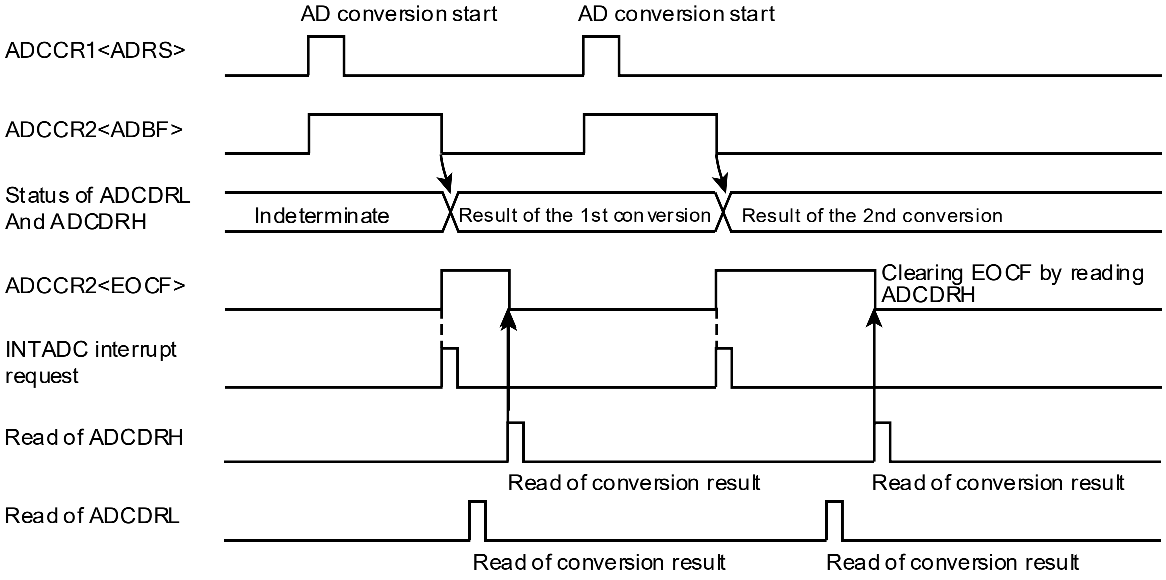

AD conversion start
AD conversion start
ADCCR1<ADRS>
ADCCR2<ADBF>
Status of ADCDRL And ADCDRH
Indeterminate
Result of the 1st conversion
Result of the 2nd conversion
ADCCR2<EOCF>
Clearing EOCF by reading ADCDRH
INTADC interrupt request
Read of ADCDRH
Read of conversion result
Read of conversion result
Read of ADCDRL
Read of conversion result
Read of conversion result

Figure 20-2 Single Mode

20.3.2. Repeat mode

In repeat mode, the voltage at an analog input pin designated at ADCCR1<SAIN> is AD converted repeatedly.

Setting ADCCR1<ADRS> to "1" after setting ADCCR1<AMD> to "11" enables AD conversion to start. After the completion of AD conversion, ADCCR1<ADRS> is automatically cleared. After the first AD conversion is finished, the conversion result is stored in the AD converted value registers (ADCDRL and ADCDRH), ADCCR2<EOCF> is set to "1", and the AD conversion finished interrupt (INTADC) is generated. After this interrupt is generated, the second AD conversion starts immediately.

The AD converted value registers (ADCDRL and ADDRH) should be read before the next AD conversion is completed. When the next AD conversion is completed in the time from reading ADCDRL to reading ADCDRH, the previous converted value is retained without overwriting the AD converted value registers (ADCDRL and ADCDRH). In this case, the INTADC interrupt request is not generated, and the conversion result without overwriting is lost. (See Figure 20-3)

To stop AD conversion, write "00"(AD operation disable) to ADCCR1<AMD>. As "00" is written to ADCCR1<AMD>, AD conversion stops immediately. In this case, the converted value is not stored in the AD converted value register. As AD conversion starts, ADCCR2<ADBF> is set to "1". It is cleared to "0" when "00" is written to <AMD>.

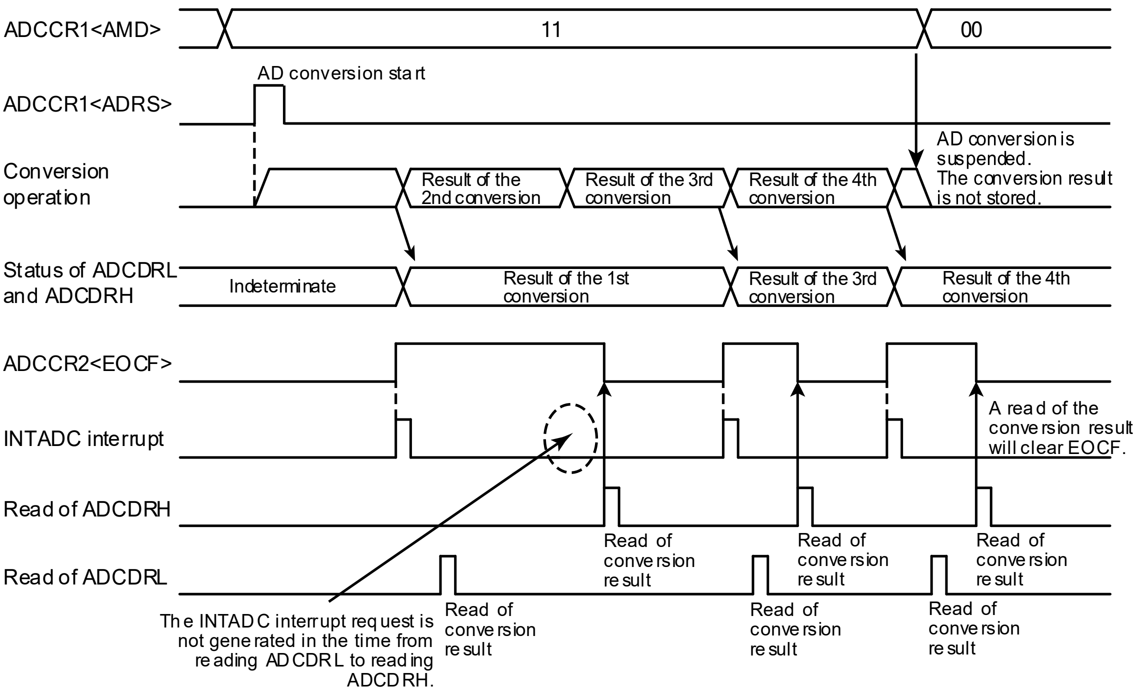

ADCCR1<AMD>
11
00
AD conversion start
ADCCR1<ADRS>
AD conversion is suspended. The conversion result is not stored.
Conversion operation
Result of the 2nd conversion
Result of the 3rd conversion
Result of the 4th conversion
Status of ADCDRL and ADCDRH
Indeterminate
Result of the 1st conversion
Result of the 3rd conversion
Result of the 4th conversion
ADCCR2<EOCF>
INTADC interrupt
A read of the conversion result will clear EOCF.
Read of ADCDRH
Read of ADCDRL
Read of conversion result
Read of conversion result
Read of conversion result
Read of conversion result
Read of conversion result
Read of conversion result
The INTADC interrupt request is not generated in the time from reading ADCDRL to reading ADCDRH.

Figure 20-3 Repeat Mode

20.3.3. AD operation disable and forced stop of AD operation

To force stopping the AD conversion, set ADCCR1<AMD> to "00" when AD conversion is operating in single mode or repeat mode.

When ADCCR1<AMD> is cleared to "00", ADCCR2<EOCF>, <ADBF>, ADCDRL and ADCDRH are initialized to "0".

20.4. Register Setting

(1) Set the AD converter control register 1 (ADCCR1) as described below:
From the AD input channel select (<SAIN>), select the channel in which AD conversion is to be performed. Set the analog input control (<AINEN>) to "Analog input enable". At <AMD>, specify the AD operating mode (single or repeat mode).

(2) Set the AD converter control register 2 (ADCCR2) as described below:
At the AD conversion time (<ACK>), specify the AD conversion time. For information on how to specify the conversion time, refer to the AD converter control register 2 and Table 20-1.

(3) After the above two steps are completed, set "1" on the AD conversion start (ADRS) of the AD converter control register 1 (ADCCR1), and AD conversion starts immediately when single mode is selected.

(4) As AD conversion is finished, the AD conversion end flag (<EOCF>) of the AD converter control register 2 (ADCCR2) is set to "1", the AD conversion result is stored in the AD converted value registers (ADCDRH and ADCDRL), and the INTADC interrupt request is generated.

(5) After the conversion result is read from the AD converted value register (ADCDRH), <EOCF> is cleared to "0". <EOCF> will also be cleared to "0" when AD conversion is performed once again before reading the AD converted value register (ADCDRH). In this case, the previous conversion result is retained until AD conversion is finished.

(6) In the repeat mode, the next conversion is started immediately after INTADC interrupt request occurs. The AD conversion result registers (ADCDRL and ADCDRH) must be read until the next AD conversion is completed.

Example: After selecting the conversion time 15.6 [[μs]] at 10 [MHz] and the analog input channel AIN3 pin, perform AD conversion once. After checking <EOCF>, store the conversion result in the HL register. The operation mode is single mode.

```
                  Port setting       ; Before setting AD converter registers, make an appropriate port register setting.
                                     ; (For details, refer to "8 I/O Ports".)
          LD      (ADCCR1), 0x33     ; Select AIN3 and operation mode
          LD      (ADCCR2), 0x02     ; Select conversion time (156 / fcgck)
          SET     (ADCCR1).7         ; <ADRS> ← "1" (AD conversion start)
SLOOP :   TEST    (ADCCR2).7         ; <EOCF> = 1 ?
          J       T, SLOOP
          LD      HL, (ADCDRL)       ; Read result data
```

20.5. Starting STOP/IDLE0/SLOW Modes

When the operation mode is changed to STOP/IDLE0/SLOW mode, ADCCR1<ADRS>, <AMD>, <AINEN>, ADCCR2<EOCF>, <ADBF>, ADCDRL and ADCDRH are initialized to "0". When the operation mode is changed to any of these modes during AD conversion, AD conversion is terminated and the AD converter stops (the registers are initialized). When the operation mode is returned from STOP/ IDLE0/SLOW mode, AD conversion is not automatically restarted. Therefore, the registers must be reconfigured as necessary.

When the operation mode is changed to STOP/IDLE0/SLOW mode during AD conversion, analog reference voltage is automatically disconnected. Therefore, The current does not flow into Analog reference voltage input pin for AD conversion .

20.6. Analog Input Voltage and AD Conversion Result

Analog input voltages correspond to AD-converted, 10-bit digital values, as shown in Figure 20-4.

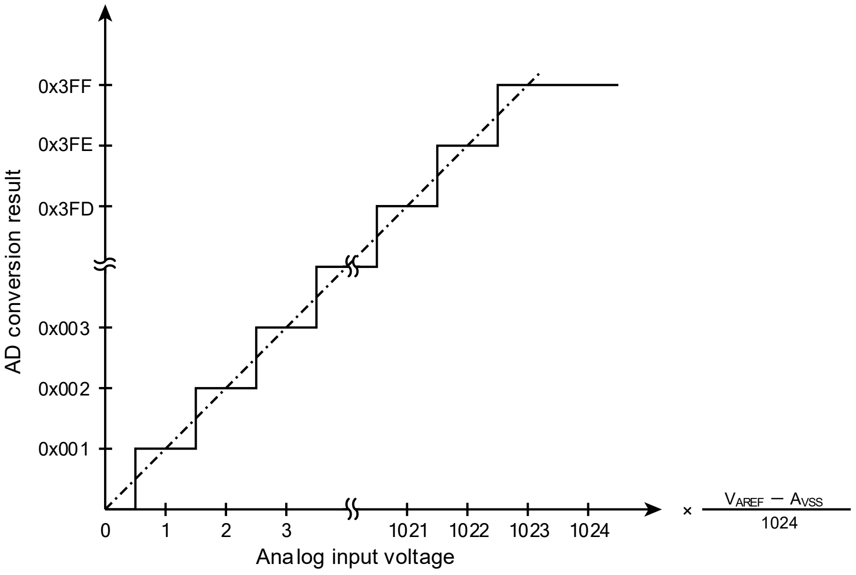


Figure 20-4 Relationships between Analog Input Voltages and AD-converted Values (Typ.)

20.7. Precautions about the AD Converter

20.7.1. Analog input pin voltage range

The voltage into the analog input pins must be used within a voltage from V_{AREF} to A_{VSS}. When any voltage outside this range is applied to one of the analog input pins, the converted value on that pin becomes undefined, and converted values on other pins will also be affected.

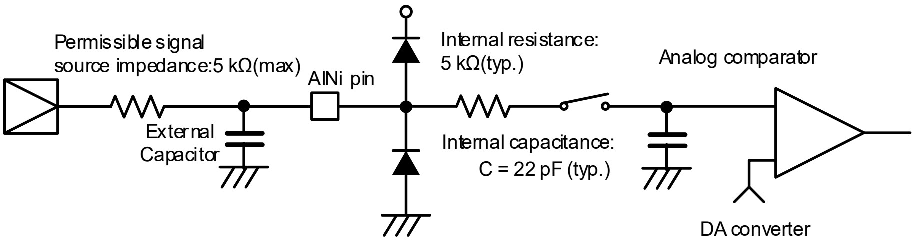
20.7.2. Analog input pins used as input/output ports

The analog input pins are shared with input/output ports. When the AD conversion is executed in using one of analog input pins, input/output instructions at all other ports must not be executed. When they are executed, there is the possibility that the accuracy of AD conversion may deteriorate. And the accuracy of AD conversion may deteriorate by the noise which is generated for input or output from/to the pins not used as the analog input.

20.7.3. Noise countermeasure

The internal equivalent circuit of the analog input pins is shown in Figure 20-5.

It is more susceptible by the noise when the output impedance of the analog input source is higher. Therefore, make sure the output impedance of the signal source in the design is 5 [kΩ] or less. It is recommended that a capacitor be attached to the outside of an analog input pin.



Note) TMP89FS60B: i = 15 to 0
TMP89FS62B: i = 7 to 0
TMP89FS63B: i = 12 to 0

Figure 20-5 Analog Input Equivalent Circuit and Example of Input Pin Processing

21. Flash Memory

The TMP89FS60B/62B/63B has Flash memory of 61440 bytes. A write and erase to be performed on Flash memory can be controlled in the following three modes:

- MCU mode

  In MCU mode, the Flash memory is accessed by the CPU control, and the Flash memory can be executed the erasing and writing without affecting the operations of a running application. Therefore, this mode is used for software debugging and firmware change after shipment of the TMP89FS60B/62B/63B.

- Serial PROM mode

  In the Serial PROM mode, the Flash memory is accessed by the CPU control. Use of the serial interface (UART and SIO) enables the Flash memory to be controlled by the small number of pins. The TMP89FS60B/62B/63B used in the Serial PROM mode supports on-board programming, which enables users to program Flash memory after the microcontroller is mounted on a user board.

- Parallel PROM mode

  In the Parallel PROM mode, the built-in Flash memory is accessed by the third party's program writer. High-speed access to the Flash memory is available by controlling address and data signals directly. To receive a support service for the program writer, please ask the our sales representative.

In the MCU and the Serial PROM modes, Flash memory control registers (FLSCR1 and FLSCR2) are used to control the Flash memory. This chapter describes how to access the Flash memory using the MCU and the Serial PROM modes.

21.1. Control

The Flash memory is controlled by the Flash memory control register 1 (FLSCR1), Flash memory control register 2 (FLSCR2), and Flash memory standby control register (FLSSTB).

Flash memory control register 1

FLSCR1		7	6	5	4	3	2	1	0
(0x0FD0)	Bit Symbol	FLSMD			BAREA	FAREA		-	-
	Read/Write	R/W			R/W	R/W		R/W	R/W
	After reset	0	1	0	0	0	0	0	0

FLSMD	Flash memory command sequence and toggle control	010:	Disable command sequence and toggle execution	
		101:	Enable command sequence and toggle execution	
		Others:	Reserved	
BAREA	BOOTROM mapping control		MCU mode	Serial PROM mode
		0:	Hide BOOTROM	-
		1:	Show BOOTROM	Show BOOTROM
FAREA	Flash memory area selection control	00:	Assign the data area "0x8000" to "0xFFFF" to the data area "0x8000" to "0xFFFF" (standard mapping).	
		01:	Assign the data area "0x1000" to "0x7FFF" to the data area "0x9000" to "0xFFFF".	
		10:	Assign the code area "0x8000" to "0xFFFF" to the data area "0x8000" to "0xFFFF".	
		11:	Assign the code area "0x1000" to "0x7FFF" to the data area "0x9000" to "0xFFFF".	

Note 1: It is prohibited to make a setting in "Reserved".

Note 2: The Flash memory control register 1 has a double-buffer structure comprised of the register FLSCR1 and a shift register. Writing "0xD5" to the register FLSCR2 allows a register setting to be stored and take effect in the shift register. This means that a register setting value does not take effect until "0xD5" is written to the register FLSCR2. The value of the shift register can be checked by reading the register FLSCRM.

Note 3: <FLSMD> must be set to either "010" or "101".

Flash memory control register 2

FLSCR2		7	6	5	4	3	2	1	0
(0x0FD1)	Bit Symbol	CR1EN							
	Read/Write	W							
	After reset	*	*	*	*	*	*	*	*

CR1EN	FLSCR1 register enable/disable control	0xD5: Others:	Enable a change in the FLSCR1 setting Reserved

Flash memory control register 1 monitor

FLSCRM
(0x0FD1)

	7	6	5	4	3	2	1	0
Bit Symbol	-	-	FLSMDM	BAREAM	FAREAM		ROMSELM	
Read/Write	R	R	R	R	R		R	
After reset	0	0	0	0	0	0	0	0

FLSMDM	Monitoring of FLSCR1<FLSMD> status	0: FLSCR1<FLSMD> = 101 setting disabled 1: FLSCR1<FLSMD> = 101 setting enabled
BAREAM	Monitoring of FLSCR1<BAREA> status	Value of currently enabled FLSCR1<BAREA>
FAREAM	Monitoring of FLSCR1<FAREA> status	Value of currently enabled FLSCR1<FAREA>
ROMSELM	Monitoring of FLSCR1<ROMSEL> status	Value of currently enabled FLSCR1<ROMSEL>

Note 1: FLSCRM is the register that checks the value of the shift register of the Flash memory control register 1.

Note 2: <FLSMDM> turns into "1" only when <FLSMD> = 101 becomes effective.

Note 3: When a read instruction is executed on FLSCRM, bits 7 and 6 are read as "0".

Note 4: In the Serial PROM mode, "1" is always read from <BAREAM>.

Flash memory standby reset control register

FLSSTB
(0x0FD2)

	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	-	-	-	-	FSTB
Read/Write	R	R	R	R	R	R	R	W
After reset	0	0	0	0	0	0	0	0

FSTB	Flash memory standby reset control	0: Disable Flash memory standby reset 1: Enable Flash memory standby reset

Note 1: <FSTB> can be modified only by a program in the RAM. When a value of <FSTB> is modified by a program in the Flash memory, it is not reflected.

Note 2: When <FSTB> is set to "1", do not execute instructions to fetch, read data from or write data to the Flash memory. When they are executed, a Flash memory standby reset will occur.

Note 3: When a read instruction is executed on FLSSTB, bits 7 to 1 are read as "0".

21.2. Functions

21.2.1. Flash memory command sequence execution and toggle control (FLSCR1<FLSMD>)

To prevent inadvertent writes to the Flash memory due to program error or TMP89FS60B/62B/63B malfunction, the execution of the Flash memory command sequence and the toggle operation can be disabled (the Flash memory can be write protected) by making an appropriate control register setting (write protect).

To enable the execution of the command sequence and the toggle operation, set FLSCR1<FLSMD> to "101", and then set "0xD5" on FLSCR2<CR1EN>. To disable the execution of the command sequence, set FLSCR1<FLSMD> to "010", and then set "0xD5" on FLSCR2<CR1EN>. When the command sequence or the toggle operation is executed with the execution of the command sequence and the toggle operation set to "disable", the executed command sequence or toggle operation takes no effect.

After a reset, FLSCR1<FLSMD> is initialized to "010" to disable the execution of the command sequence. FLSCR1<FLSMD> should normally be set to "010" except when a write or erase is to be performed on the Flash memory.

Note : When FLSCR1<FLSMD> is set to "disable", subsequent commands (write instructions) generated are rejected but a command sequence which is executed or executed from halfway is not initialized. To set FLSCR1<FLSMD> to "disable", set FLSCR1<FLSMD> to “101” after all command sequences are completed and verify that the Flash memory is ready to be read.

21.2.2. Flash memory area switching (FLSCR1<FAREA>)

To perform an erase or write on the Flash memory, a memory transfer instruction (command sequence) must be executed. When a memory transfer instruction is used to read or write data, a read or write can be performed only on the data area. To perform an erase or write on the code area, therefore, part of the code area must be temporarily switched to the data area. This switching between data and code areas is performed by making the appropriate FLSCR1<FAREA> setting.

Note: The program which sets FLSCR1<FAREA> must be executed in the RAM.

By setting "0xD5" on FLSCR2<CR1EN> after setting FLSCR1<FAREA> to "10", "0x8000" to "0xFFFF" (AREA C1) in the code area is mapped to "0x8000" to "0xFFFF" (AREA D1) in the data area.

By setting "0xD5" on FLSCR2<CR1EN> after setting FLSCR1<FAREA> to "11", "0x1000" to "0x7FFF" (AREA C0) in the code area is mapped to "0x9000" to "0xFFFF" (AREA D1) in the data area.

For example, to access "0x4000" in the code area, set "0xD5" on FLSCR2<CR1EN> after setting FLSCR1<FAREA> to "11", and then execute the memory transfer instruction on "0xC000".

To restore the Flash memory to the initial state of mapping, set FLSCR1<FAREA> to "00", and then set "0xD5" on FLSCR2<CR1EN>.

All Flash memory areas can be accessed by performing the appropriate steps described above and then executing the memory transfer instruction on "0x8000" to "0xFFFF" (AREA D1) in the data area.

"0x1000" to "0xFFFF" (AREA D1) in the data area and "0x1000" to "0xFFFF" (AREA C1) in the code area are mirror areas; these two areas refer to the same physical address in memory. Therefore, an erase or write must be performed on one of these two mirror areas. For example, when a write is performed on "0x8000" in the data area with FLSCR1<FAREA> set to "10" after performing a write on "0x8000" in the data area with FLSCR1<FAREA> set to "00", data is overwritten. To write data to the Flash memory that already has data written to it, existing data must first be erased from the Flash memory by performing a Sector Erase or Chip Erase, and then data must be written.

Additionally, access to areas to which memory is not assigned by executing an instruction, or specifying such an area by using jump or call instructions should be avoided.

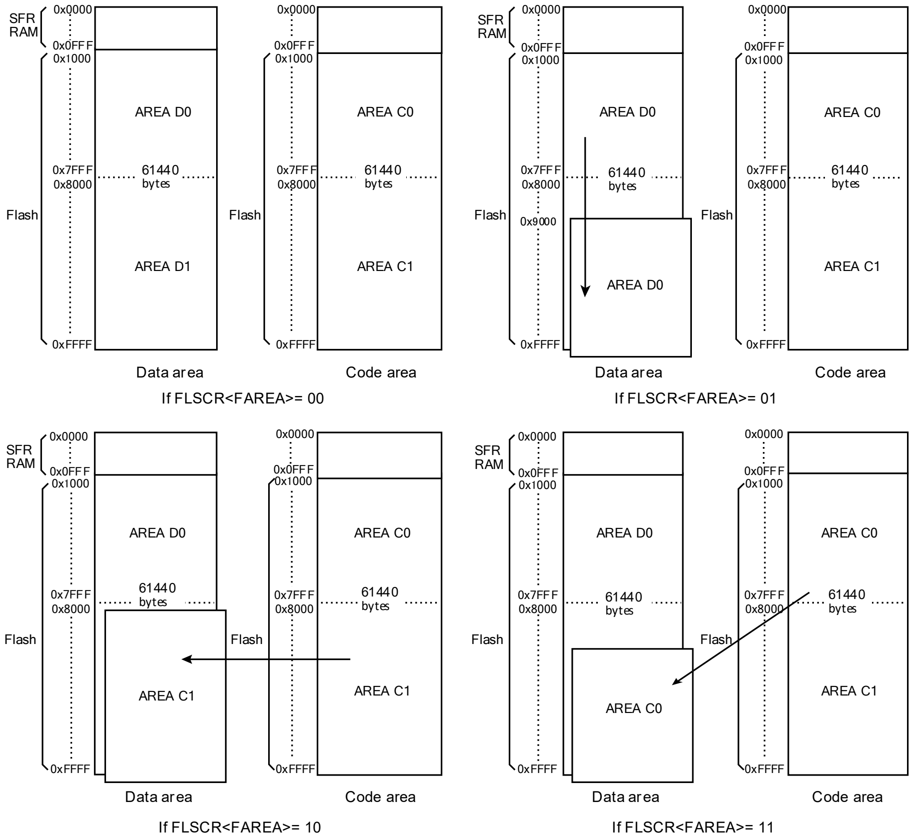


Figure 21-1 Area Switching Using the FLSCR1<FAREA> Setting

21.2.3. RAM area switching (SYSCR3<RAREA>)

When "0xD4" is set on SYSCR4<SYSCR4> after SYSCR3<RAREA> is set to "1" in MCU mode, RAM is mapped to the code area. To restore the RAM area to the initial state of mapping, set SYSCR3<RAREA> to "0", and then set "0xD4" on SYSCR4.

In the Serial PROM mode, RAM is mapped to the code area, irrespective of the SYSCR3<RAREA> setting.

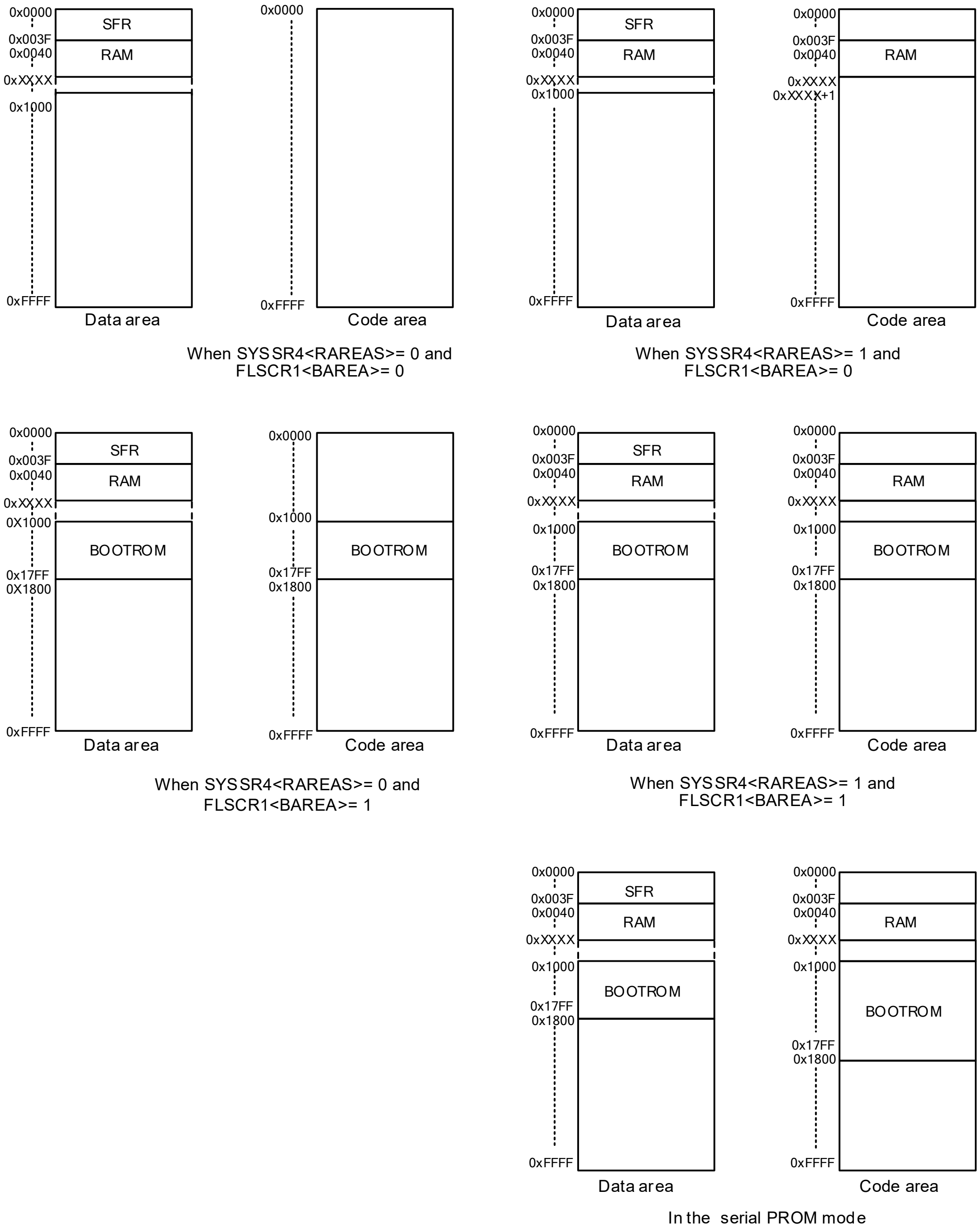
21.2.4. BOOTROM area switching (FLSCR1<BAREA>)

When "0xD5" is set on FLSCR2<CR1EN> after FLSCR1<BAREA> is set to "1" in MCU mode, "0x1000" to "0x17FF" in the code and data areas is masked by Flash memory, and 2K-byte (first half of 4KB) BOOTROM is mapped. Not to map BOOTROM, set "0xD5" on FLSCR2<CR1EN> after setting FLSCR1<BAREA> to "0".

A set of codes for programming Flash memory in the Serial PROM mode are built into BOOTROM, and a support program (API) for performing an erase or write on Flash memory in a simple manner is also built into one part in the BOOTROM area. Therefore, by calling a subroutine in the support program after BOOTROM is mapped, it is possible to erase, write and read Flash memory easily.

In the Serial PROM mode, BOOTROM is mapped to "0x1000" to "0x17FF" in the data area and "0x1000" to "0x1FFF" in the code area, irrespective of the FLSCR1<BAREA> setting. <BAREA> is always "1", and the set <BAREA> value remains unchanged, when data is written. "1" is always read from <BAREA>.

Note: Do not allocate the FLSCR1<FAREA> switching program to the code area "0x1000" to "0x1FFF". When it is allocated to that area, the software program may not function properly, and the TMP89FS60B/62B/63B may malfunction



When SYSSR4<RAREAS>= 0 and
FLSCR1<BAREA>= 0

When SYSSR4<RAREAS>= 1 and
FLSCR1<BAREA>= 0

When SYSSR4<RAREAS>= 0 and
FLSCR1<BAREA>= 1

When SYSSR4<RAREAS>= 1 and
FLSCR1<BAREA>= 1

In the serial PROM mode

Note: 0xXXXX means then end address of the RAM.

Figure 21-2 Show/Hide Switching for BOOTROM and RAM

21.2.5. Flash memory standby reset control

The Flash memory standby reset is used when a software for the TMP89FS60 is applied to TMP89FS60B/62B/63B.

About a detail of the Flash memory standby reset, refer to the explain of FLSSTB<FSTB>.

21.3. Command Sequence

In the MCU and the Serial PROM modes, the command sequence consists the command sequences shown below (JEDEC compatible). Table 21-1 shows the detail of command sequences.

Table 21-1 Command Sequence

	Command sequence	1st Bus Write Cycle		2nd Bus Write Cycle		3rd Bus Write Cycle		4th Bus Write Cycle		5th Bus Write Cycle		6th Bus Write Cycle	
		Address	Data	Address	Data	Address	Data	Address	Data	Address	Data	Address	Data
1	Byte Program	0x#555	0xAA	0x#AAA	0x55	0x#555	0xA0	BA (Note 1)	Data (Note 1)	-	-	-	-
2	Sector Erase (partial erase in units of 4KB)	0x#555	0xAA	0x#AAA	0x55	0x#555	0x80	0x#555	0xAA	0x#AAA	0x55	SA (Note 2)	0x30
3	Chip Erase (all erase)	0x#555	0xAA	0x#AAA	0x55	0x#555	0x80	0x#555	0xAA	0x#AAA	0x55	0x#555	0x10
4	Product ID Entry	0x#555	0xAA	0x#AAA	0x55	0x#555	0x90	-	-	-	-	-	-
5	Product ID Exit	0x#XXX	0xF0	-	-	-	-	-	-	-	-	-	-
6	Security Program	0x#555	0xAA	0x#AAA	0x55	0x#555	0xA5	0xFF7F	0x00	-	-	-	-

Note 1: Specify the address and data to be written (Refer to Table 21-2 about BA).

Note 2: The area to be erased is specified with the upper 4 bits of the address (Refer to Table 21-3 about SA).

Note 3: Do not start the STOP, IDLE0/1/2 or SLEEP0/1 mode while a command sequence is being executed or a task specified in a command sequence is being executed (write, erase or ID entry).

Note 4: #: "0x1" to "0xF" should be specified as the upper 4bits of the address. However, when FLSCRM<BAREAM> is "1", "0x2" or more should be specified. Usually, it is recommended that "0xF" be specified.

Note 5: XXX: Don’t care

21.3.1. Byte Program

This command writes the Flash memory in units of one byte. The address and data to be written are specified in the 4th bus write cycle. The range of addresses that can be specified is shown in Table 21-2. For example, to write data to "0x1000" in the data area, set FLSCR1<FAREA> to "01", set "0xD5" on FLSCR2<CR1EN>, and then specify "0x9000" as an address in the 4th bus write cycle. The time needed to write each byte is 3 [ms] maximum. The next command sequence cannot be executed when an ongoing write operation is not completed. To check the completion of the write operation, perform read operations twice on the same address in the Flash memory, and perform polling until the same data is read from the Flash memory. During the write operation, bit 6 is reversed each time a read is performed.

Note 1: To rewrite data to addresses in the Flash memory where data (including "0xFF") is already written, the data of address must be erased by performing a Sector Erase or Chip Erase before writing data.

Note 2: The data and code areas become mirror areas. As you access these areas, you are brought to the same physical address in memory. When performing a Byte Program, make sure that you write data to either of these two areas, not both.

Note 3: Do not perform a Byte Program on areas other than those shown in Table 21-2.

Table 21-2 Range of Addresses Specifiable (BA)

Write Area		FLSCR1 <FAREA>	Address specified by instruction
AREA D0 (Data area)	0x1000 to 0x7FFF	01	0x9000 to 0xFFFF
AREA D1 (Data area)	0x8000 to 0xFFFF	00	0x8000 to 0xFFFF
AREA C0 (Code area)	0x1000 to 0x7FFF	11	0x9000 to 0xFFFF
AREA C1 (Code area)	0x8000 to 0xFFFF	10	0x8000 to 0xFFFF

21.3.2. Sector Erase (4-Kbyte partial erase)

This command erases the Flash memory in units of 4 Kbytes. The Flash memory area to be erased is specified by the upper 4 bits of the 6th bus write cycle address. The range of addresses that can be specified is shown in Table 21-3. For example, to erase 4 Kbytes from "0x1000" to "0x1FFF" in the code area, set FLSCR1<FAREA> to "11", set "0xD5" on FLSCR2<CR1EN>, and then specify the address from "0x9000" to "0x9FFF" as the 6th bus write cycle.

The time needed to erase 4 Kbytes is 8 [ms] maximum. The next command sequence cannot be executed when an ongoing erase operation is not completed. To check the completion of the erase operation, perform read operations twice on the same address in the Flash memory, and perform polling until the same data is read from the Flash memory. During the erase operation, bit 6 is reversed each time a read is performed.

Data in the erased area is "0xFF".

Note 1: The data and code areas become mirror areas. As you access these areas, you are brought to the same physical address in memory. When performing a Sector Erase, make sure that you erase data from either of these two areas, not both.

Note 2: Do not perform a Sector Erase on areas other than those shown in Table 21-3.

Table 21-3 Range of Addresses Specifiable (SA)

Erase Area		FLSCR1<FAREA>	Address specified by instruction
AREA D0 (Data Area)	0x1000 to 0x1FFF	01	0x9000 to 0x9FFF
	0x2000 to 0x2FFF		0xA000 to 0xAFFF
	0x3000 to 0x3FFF		0xB000 to 0xBFFF
	0x4000 to 0x4FFF		0xC000 to 0xCFFF
	0x5000 to 0x5FFF		0xD000 to 0xDFFF
	0x6000 to 0x6FFF		0xE000 to 0xEFFF
	0x7000 to 0x7FFF		0xF000 to 0xFFFF
AREA D1 (Data Area)	0x8000 to 0x8FFF	00	0x8000 to 0x8FFF
	0x9000 to 0x9FFF		0x9000 to 0x9FFF
	0xA000 to 0xAFFF		0xA000 to 0xAFFF
	0xB000 to 0xBFFF		0xB000 to 0xBFFF
	0xC000 to 0xCFFF		0xC000 to 0xCFFF
	0xD000 to 0xDFFF		0xD000 to 0xDFFF
	0xE000 to 0xEFFF		0xE000 to 0xEFFF
	0xF000 to 0xFFFF		0xF000 to 0xFFFF
AREA C0 (Code Area)	0x1000 to 0x1FFF	11	0x9000 to 0x9FFF
	0x2000 to 0x2FFF		0xA000 to 0xAFFF
	0x3000 to 0x3FFF		0xB000 to 0xBFFF
	0x4000 to 0x4FFF		0xC000 to 0xCFFF
	0x5000 to 0x5FFF		0xD000 to 0xDFFF
	0x6000 to 0x6FFF		0xE000 to 0xEFFF
	0x7000 to 0x7FFF		0xF000 to 0xFFFF
AREA C1 (Code Area)	0x8000 to 0x8FFF	10	0x8000 to 0x8FFF
	0x9000 to 0x9FFF		0x9000 to 0x9FFF
	0xA000 to 0xAFFF		0xA000 to 0xAFFF
	0xB000 to 0xBFFF		0xB000 to 0xBFFF
	0xC000 to 0xCFFF		0xC000 to 0xCFFF
	0xD000 to 0xDFFF		0xD000 to 0xDFFF
	0xE000 to 0xEFFF		0xE000 to 0xEFFF
	0xF000 to 0xFFFF		0xF000 to 0xFFFF

21.3.3. Chip Erase (all erase)

This command erases the entire Flash memory.

The time needed to erase it is 136 [ms] maximum. The next command sequence cannot be executed when an ongoing erase operation is not completed. To check the completion of the erase operation, perform read operations twice on the same address in the Flash memory, and perform polling until the same data is read from the Flash memory. During the erase operation, bit 6 is reversed each time a read is performed.

Data in the erased area is "0xFF".

21.3.4. Product ID Entry

This command activates the product ID mode. When an instruction to read the Flash memory is executed in Product ID mode, the vendor ID, flash ID and security status can be read from the Flash memory.

Table 21-4 Values to Be Read in Product ID Mode

Address	Meaning	Read value
0xF000	Vendor ID	0x98
0xF001	Flash ID	0x4D
0xFF7F	Security status	0xFF: Security Program disabled Other than 0xFF: Security Program enabled

21.3.5. Product ID Exit

This command is used to exit the Product ID mode

21.3.6. Security Program

The Security Program is enabled, the Flash memory is protected from write and read operation. The Flash memory write command cannot be executed in the Serial PROM mode.

To disable the Security Program, the Chip Erase must be performed. To check whether the Security Program is enabled or disabled, read "0xFF7F" in product ID mode. Refer to Table 21-4 for further details. The time needed to enable or disable the Security Program is 3 [ms] maximum. The next command sequence cannot be executed until the Security Program setting is completed. To check the completion of the Security Program setting, perform read operations twice on the same address in the Flash memory, and perform polling until the same data is read. When the Security Program setting is being made, bit 6 is reversed each time a read is performed.

21.4. Toggle Bit (D6)

After the Byte Program, Sector Erase, Chip Erase, or Security Program command sequence are executed, the value of the 6th bit (D6) in data read by a read operation is reversed each time a read is performed. This bit reversal can be used as a software mechanism for checking the completion of each operation. Normally, perform read operations twice on the same address in the Flash memory, and perform polling until the same data is read from the Flash memory.

After the Flash memory write, the Chip Erase, and the Security Program command sequence are executed, the toggle bit read by the first read operation is always "1".

Note 1: When FLSCR1<FLSMD> is set to "disable", the toggle bit is not reversed.

Note 2: Do not read the toggle bit by using a 16-bit transfer instruction. When the toggle bit is read using a 16-bit transfer instruction, the toggle bit does not function properly.

21.5. Access to the Flash Memory Area

A read or a program fetch cannot be performed on the whole of the Flash memory area when data is being written to the Flash memory, when data in Flash memory is being erased or when a security setting is being made in the Flash memory. When performing these operation on the Flash memory area, the Flash memory cannot be directly accessed by using a program in the Flash memory; the Flash memory must be accessed using a program in the BOOTROM area or the RAM area.

Data can be written to and read from the Flash memory area in units of one byte. Data in the Flash memory can be erased in units of 4 Kbytes, and all data in the Flash memory can be erased at one stroke. A read can be performed using one memory transfer instruction. A write or erase, however, must be performed using more than one memory transfer instruction because the command sequence method is used. For information on the command sequence, refer to Table 21-1.

Note 1: To enable a program to resume control on the Flash memory area that is rewritten, it is recommended that the jump (or return) instruction be executed after verifying that the program has been written properly.

Note 2: Do not reset the MCU (including a reset generated due to internal factors) when data is being written to the Flash memory, data is being erased from the Flash memory or the security command is being executed. When a reset occurs, there is the possibility that data in the Flash memory may be rewritten to an unexpected value.

21.5.1. Flash memory control in the Serial PROM mode

The Serial PROM mode is used to access the Flash memory by using a control program provided in the BOOTROM area. Since almost all operations relating to access to the Flash memory can be controlled simply using data supplied via the serial interface (UART), it is not necessary to operate the control register for the user. For details of the Serial PROM mode, see "22. Serial PROM Mode".

21.5.2. Flash memory control in MCU mode

In MCU mode, a write can be performed on the Flash memory by executing a control program in the RAM or using a support program (API) provided inside BOOTROM.

21.5.2.1. How to write to the Flash memory by transferring a control program to the RAM area

This section describes how to execute a control program in then RAM in MCU mode. A control program to be executed in the RAM must be acquired and stored in the Flash memory or it must be imported from an outside source through a communication pin. (The following procedure assumes that a program copy is provided inside the Flash memory.)

Steps 1 to 5 and 11 shown below concern the control by a program in the Flash memory, and other steps concern the control by a program transferred to RAM. The following procedure is linked with a program example to be described later.

(1) Set the interrupt master enable flag to disable (DI) (<IMF> ← "0").

(2) Transfer the write control program to the RAM

(3) Establish the non-maskable interrupt vector in the RAM area.

(4) After setting both SYSCR3<RAREA> and <RVCTR> to "1", set "0xD4" on SYSCR4 to allocate the RAM to the code area, and to switch the vector area to the RAM.

(5) Invoke the erase processing program in the RAM by executing the CALL instruction.

(6) Set FLSCR1<FLSMD> to "101", and specify the area to be erased by making the appropriate FLSCR1<FAREA> setting. (Make the appropriate FLSCR1<ROMSEL> setting, as necessary.) Then set "0xD5" on FLSCR2<CR1EN>.

(7) Execute the erase command sequence.

(8) Perform a read on the same address in the Flash memory twice consecutively. (Repeat this step until the read values become the same.)

(9) After setting FLSCR1<FLSMD> to "010" and FLSCR1<FAREA> to "00", set "0xD5" on FLSCR2<CR1EN>. (This disables the execution of the command sequence and returns <FAREA> to the initial state of mapping.)

(10) Execute the RET instruction to return to the Flash memory.

(11) Invoke the write program in the RAM area by executing the CALL instruction.

(12) Set FLSCR1<FLSMD> to "101", and make the appropriate FLSCR1<FAREA> setting to specify the area (area erased by performing step 7 above) on which a write is to be performed. (Make the appropriate FLSCR1<ROMSEL> setting, as necessary.) Then set "0xD5" on FLSCR2<CR1EN>.

(13) Execute the write command sequence.

(14) Perform a read on the same address in the Flash memory twice consecutively. (Repeat this step until the read values become the same.)

(15) After setting FLSCR1<FLSMD> to "010" and FLSCR1<FAREA> to "00", set "0xD5" on FLSCR2<CR1EN>. (This disables the execution of the command sequence and returns <FAREA> to the initial state of mapping.)

(16) Execute the RET instruction to return to the Flash memory.

(17) After setting both SYSCR3<RAREA> and <RVCTR> to "0", set "0xD4" on SYSCR4 to release RAM allocation for the code area, and to switch the vector area to the flash area.

Note 1: Before writing data to the Flash memory from the RAM area in MCU mode, the vector area must be switched to the RAM area by using SYSCR3<RVCTR>, data must be written to the vector addresses (INTUNDEF and INTSWI: "0x01F8" to "0x01F9", INTWDT: "0x01FC" to "0x01FD") that correspond to non-maskable interrupts, and the interrupt subroutine (RAM area) must be defined. This enables to trap the errors that may occur due to an unexpected non-maskable interrupt during a write. When SYSCR3<RVCTR> is set in the Flash memory area and when an unexpected interrupt occurs during a write, a malfunction may occur because the vector area in the Flash memory cannot be read properly.

Note 2: Before using a certain interrupt in MCU mode, the vector address corresponding to that interrupt and the interrupt service routine must be established inside the RAM area. In this case, the non-maskable interrupt setting must be made, as explained in Note 1.

Note 3: Before jumping from the Flash memory to the RAM area, RAM must be allocated to the code area by making the appropriate SYSCR3<RAREA> setting (setting made in step 4 in the procedure described on the previous page).

Example: Case in which a program is transferred to RAM, a Sector Erase is performed on "0xE000" to "0xEFFF" in the code area, and then "0x3F" data is written to "0xE500".
When non-maskable interrupts (INTSWI, INTUNDEF or INTWDT) occur, system clock reset is generated.

```
                cRAMStartAdd equ 0x0200          ; RAM start address
main section code abs = 0xF000
                DI                               ; Disable interrupts (STEP 1)
; #### Transfer the program to RAM #### (STEP 2)
                LD      HL, cRAMStartAdd
                LD      IX, sRAMprogStart
sRAMLOOP:       LD      A, (IX)                  ; Transfer the program from sRAMprogStart to
                LD      (HL), A                  ; sRAMprogEnd into cRAMStartAdd
                INC     HL
                INC     IX
                CMP     IX, sRAMprogEnd
                J       NZ, sRAMLOOP
; #### Set a non-maskable interrupt vector inside the RAM area #### (STEP 3)
                LD      HL, 0x01FC               ; Set INTUNDEF and INTSWI interrupt vectors
                LDW     (HL), sINTSWI - sRAMprogStart + cRAMStartAdd
                LD      HL, 0x01F8               ; Set INTWDT interrupt vector
                LDW     (HL), sINTWDT - sRAMprogStart + cRAMStartAdd
; #### Allocate RAM to the code area. Switch the vector area to RAM #### (STEP 4)
                LD      (SYSCR3), 0x06           ; Set both <RAREA> and <RVCTR> to "1"
                LD      (SYSCR4), 0xD4           ; Write enabled code “0xD4” to SYSCR4
; #### Sector Erase and write process ####
                LD      HL, 0xF555               ; Variable for command sequence
                LD      DE, 0xFAAA               ; Variable for command sequence
; Sector Erase Process (STEP 5)
                LD      C, 0x00                  ; Set upper addresses
                LD      IX, 0xE000               ; Set middle and lower addresses
                CALL    sSectorErase - sRAMprogStart + cRAMStartAdd
                                                 ; Perform a Sector Erase (0xE000)
; Write process (STEP 11)
                LD      C, 0x00                  ; Set upper addresses
                LD      IX, 0xE500               ; Set middle and lower addresses
                LD      B, 0x3F                  ; Data to be written
                CALL    sByteProgram - sRAMprogStart + cRAMStartAdd
                                                 ; Write process (0xE500)
; #### Release allocating RAM to the code area. Switch the vector area to Flash memory #### (STEP 17)
                LD      (SYSCR3), 0x00           ; Set both <RAREA> and <RVCTR> to "0"
                LD      (SYSCR4), 0xD4           ; Write enable code “0xD4” to SYSCR4
;#### Execute the next main program ####
                        :                        ; Execute the main program
                J       XXXXX
; #### Program to be executed in RAM ####
sRAMprogStart:  NOP                              ; Fail-safe process
                NOP
                NOP
                NOP
                NOP
                LD      (SYSCR2), 0x10           ; Generate system clock reset
sSectorErase:   CALL    sAddConv - sRAMprogStart + cRAMStartAdd
                                                 ; Address conversion process
; Sector Erase process (STEP 7)
                LD      (HL), E                  ; 1st Bus Write Cycle (Note 1)
                LD      (DE), L                  ; 2nd Bus Write Cycle (Note 1)
                LD      (HL), 0x80               ; 3rd Bus Write Cycle (Note 1)
                LD      (HL), E                  ; 4th Bus Write Cycle (Note 1)
                LD      (DE), L                  ; 5th Bus Write Cycle (Note 1)
                LD      (IX), 0x30               ; 6th Bus Write Cycle (Note 1)
                J       sRAMopEnd
; Write process (STEP 13)
sByteProgram:   CALL    sAddConv - sRAMprogStart + cRAMStartAdd
                                                 ; Address conversion process
                LD      (HL), E                  ; 1st Bus Write Cycle (Note 1)
                LD      (DE), L                  ; 2nd Bus Write Cycle (Note 1)
                LD      (HL), 0xA0               ; 3rd Bus Write Cycle (Note 1)
                LD      (IX), B                  ; 4th Bus Write Cycle (Note 1)
; End process
```

```
sRAMopEnd:    NOP                               ; (Note 2)
              NOP                               ; (Note 2)
              NOP                               ; (Note 2)
sLOOP1:       LD      A, (IX)                   ; (STEP 8, 14)
              CMP     A, (IX)
              J       NZ, sLOOP1                ; Loop until the read values become the same
              LD      (FLSCR1), 0x40            ; Disable the execution of command sequence (STEP 9, 15)
              LD      (FLSCR2), 0xD5            ; Write enabled code “0xD5” to FLSCR2
              RET                               ; Return to Flash memory
; Address conversion process (STEP 6, 12)
sAddConv:     LD      WA, IX
              SWAP    C
              AND     C, 0x10
              SWAP    W
              AND     C, 0x08
              OR      C, W
              XOR     C, 0x08
              SHRC    C
              OR      C, 0xA0
              LD      (FLSCR1), C               ; Enable the execution of command sequence. Set <FAREA>.
              LD      (FLSCR2), 0xD5            ; Write enabled code “0xD5” to FLSCR2
              LD      WA,IX
              TEST    C.3
              J       Z, sAddConvEnd
              OR      W, 0x80
              LD      IX, WA
sAddConvEnd:  RET
; Interrupt subroutine
sINTWDT:
sINTSWI:      LD      IX, 0xF000
              LD      A, (IX)
              CMP     A, (IX)
              J       NZ, sINTWDT               ; Loop until the read values become the same
                                                ; (Wait the completion of erasing or writing)
              LD      (SYSCR2), 0x10            ; Generate system clock reset to restart TMP89FS60B/62B/63B
              RETN
sRAMprogEnd:  NOP
```

Note 1: In using a write instruction in the n-th Bus Write Cycle, use a write instruction of more than three machine cycles or arrange write instructions in such a way that they are generated at intervals of three or more machine cycles. When a 16-bit transfer instruction is used or when write instructions are executed at intervals of two machine cycles, the Flash memory command sequence will not be transmitted properly, and a malfunction may occur.

Note 2: When a read of the Flash memory (toggle operation) is to be performed after a write instruction is generated in the n-th Bus Write Cycle, instructions must be arranged in such a way that they are generated at intervals of three or more machine cycles; machine cycles are counted from when the last n-th bus write cycle is generated to when each instruction is generated. Generally three NOP instructions are used. When the interval between instructions is short, the toggle bit does not operate correctly.

21.5.2.2. How to write to the Flash memory by using a support program (API) of BOOTROM

TMP89FS60B/62B/63B has following support program (API) inside BOOTROM. The following shows how to perform an erase and a write on the Flash memory by using a support program (API) of BOOTROM in MCU mode. For details, please refer to "21.6. API (Application Programming Interface)".

Steps 1 to 15 shown below concern the control by a program in the Flash memory.

(1) Transfer the subroutine program of non-maskable interrupt (INTSWI, INTWDT) to the RAM.

(2) Establish the non-maskable interrupt vector in the RAM area.

(3) After setting both SYSCR3<RAREA> and <RVCTR> to "1", set "0xD4" on SYSCR4 to allocate the RAM to the code area, and to switch the vector area to the RAM.

(4) Set "0xD5" on FLSCR2<CR1EN> after setting FLSCR1<BAREA> to "1".

(5) Set the erasing address range of the sector to A register. For example, to erase area from "0xE000" to "0xEFFF", set "0x0E" to A register. To erase area from "0x1F000" to "0x1FFFF", set "0x1F" to A register.

(6) Set "0xD5" to C register as enable code.

(7) Call address "0x1012". (Sector Erase is performed. It is not necessary from 2 to 4 steps, when writing area is already erased beforehand.)

(8) Set "0x00" to C register.

(9) Set A[15:0] of address for programing to WA register.

(10) Set programing data to E register.

(11) Set "0xD5" to (SP-) as enable code.

(12) Call address (0x1010). (Byte Program is performed)

(13) When writing is continued for other address, return to step 8.

(14) Set "0xD5" to FLSCR2<CR1EN> after setting FLSCR1<BAREA> to "0".

(15) Set "0xD4" to SYSCR4 after setting both SYSCR3<RAREA> and <RVCTR> to "0".

Example: Case in which a Sector Erase is performed on "0xE000" to "0xEFFF" in the data area, and then data at "0x0100" to "0x01FF" in the RAM is written to "0xE000" to "0xE0FF" in the data area.
When non-maskable interrupts (INTSWI, INTUNDEF or INTWDT) occur, system clock reset is generated.

```
                    .BTWrite        equ 0x1010          ; Write data to the Flash memory
                    .BTEraseSec     equ 0x1012          ; Sector Erase
                    .BTEraseChip    equ 0x1014          ; Chip Erase
                    .BTGetRP        equ 0x1016          ; Check the status of the Security Program
                    .BTSetRP        equ 0x1018          ; Configure the Security Program

                    cRAMStartAdd    equ 0x0200          ; RAM start address
main section code abs = 0xF000
; #### Transfer the program to RAM ####
                    LD      HL, cRAMStartAdd
                    LD      IX, sRAMprogStart
sRAMLOOP:           LD      A, (IX)                     ; Transfer the program from sRAMprogStart to
                    LD      (HL), A                     ; sRAMprogEnd into cRAMStartAdd
                    INC     HL
                    INC     IX
                    CMP     IX, sRAMprogEnd
                    J       NZ, sRAMLOOP
; #### Set a non-maskable interrupt vector inside the RAM area ####
                    LD      HL, 0x01FC                  ; Set INTUNDEF and INTSWI interrupt vectors
                    LDW     (HL), sINTSWI - sRAMprogStart + cRAMStartAdd
                    LD      HL, 0x01F8                  ; Set INTWDT interrupt vector
                    LDW     (HL), sINTWDT - sRAMprogStart + cRAMStartAdd
; #### Allocate RAM to the code area. Switch the vector area to RAM ####
                    LD      (SYSCR3), 0x06              ; Set both <RAREA> and <RVCTR> to "1"
                    LD      (SYSCR4), 0xD4              ; Write enable code “0xD4” to SYSCR4
; #### Allocate BOOTROM to the data/code area ####
                    LD      (FLSCR1), 0x50              ; Set <BAREA> to "1" (Note 1)
                    LD      (FLSCR2), 0xD5              ; Write enable code “0xD5” to FLSCR2
; #### Sector Erase process (API) ####
                    LD      A, 0x0E                     ; Specify the area to be erased (0xE000 to 0xEFFF)
                    LD      C, 0xD5                     ; Enable Code
                    CALL    (.BTEraseSec)               ; Execute Sector Erase
; #### Write process ####
                    LD      HL, 0xE000                  ; Flash start address (address where data is written)
                    LD      IY, 0x0100                  ; RAM start address
sLOOP1:
                    LD      C, 0x00                     ; Address where data is written (bit 16)
                    LD      WA, HL                      ; Address where data is written (bits 15 to 0)
                    LD      E, (IY)                     ; Data to be written
                    LD      (SP-), 0xD5                 ; Enable Code
                    CALL    (.BTWrite)                  ; Write data to the Flash memory (1 byte)
                    INC     IY                          ; Increment flash address
                    INC     HL                          ; Increment RAM address
                    CMP     L, 0x00                     ; Finish 256-byte write?
                    J       NZ, sLOOP1                  ; Return to sLOOP1 when the number of bytes is less than 256
```

```
; #### End process ####
            LD      (FLSCR1), 0x40          ; Set <BAREA> to "0"
            LD      (FLSCR2), 0xD5          ; Write enable code 0xD5 to FLSCR2
            LD      (SYSCR3), 0x00          ; Set both <RAREA> and <RVCTR> to "0"
            LD      (SYSCR4), 0xD4          ; Write enable code “0xD4” to SYSCR4
                    :                       ; Main program processing
            J       XXXX
; #### Program to be executed in RAM ####
sRAMprogStart:
; Interrupt subroutine
sINTWDT:
sINTSWI:    LD      IX, 0xF000
            LD      A, (IX)
            CMP     A, (IX)
            J       NZ, sINTWDT             ; Loop until the read values become the same
                                            ; (Wait the completion of erasing or writing)
            LD      (SYSCR2), 0x10          ; Generate system clock reset to restart TMP89FS60B/62B/63B
            RETN
sRAMprogEnd:
            NOP
```

Note 1: Do not allocate the above program to "0x1000" to "0x17FF" in the code area of the Flash memory. When <BAREA> is set “1”, this area changes from the Flash memory area to the BOOTROM area so that the program will not function properly and the TMP89FS60B/62B/63B may malfunction.

Note 2: It is not necessary to add DI instruction for above example program, because the support program include it. However, the support program does not include EI instruction. Therefore, when an interrupt process is used, enable <IMF> after finishing all above process.

21.5.2.3. How to set the Security Program by using a support program (API) of BOOTROM

(1) Transfer the subroutine program of non-maskable interrupt (INTSWI, INTWDT) to the RAM.

(2) Establish the non-maskable interrupt vector in the RAM area.

(3) After setting both SYSCR3<RAREA> and <RVCTR> to "1", set "0xD4" on SYSCR4. Then allocate RAM to the code area, and switch the vector area to the RAM.

(4) Set FLSCR1<BAREA> to "1".

(5) Set "0xD5" to A register as enable code.

(6) Set "0x00" to C register.

(7) Call address "0x1016". (After processing, Security Program state returns to A register.)

(8) When A register is not "0xFF", jump to sSKIP(12) because Security Program is already set.

(9) Set "0xD5" to A register as enable code.

(10) Set "0x00" to C register.

(11) Call address "0x1018". (Security Program is performed.)

(12) Set "0xD5" to FLSCR2 after setting FLSCR1<BAREA> to "0".

(13) Set "0xD4" to SYSCR4 after setting both SYSCR3<RAREA> and <RVCTR> to "0".

Example: Whether the Security Program is enabled or disabled is checked. When it is disabled, it is enabled.

```
    .BTWrite         equ 0x1010          ; Write data to the Flash memory
    .BTEraseSec      equ 0x1012          ; Sector Erase
    .BTEraseChip     equ 0x1014          ; Chip Erase
    .BTGetRP         equ 0x1016          ; Check the status of the Security Program
    .BTSetRP         equ 0x1018          ; Configure the Security Program

    cRAMStartAdd     equ 0x0200          ; RAM start address
main section code abs = 0xF000
; #### Transfer the program to RAM ####
           LD      HL, cRAMStartAdd
           LD      IX, sRAMprogStart
sRAMLOOP:  LD      A, (IX)               ; Transfer the program from sRAMprogStart to
           LD      (HL), A               ; sRAMprogEnd into cRAMStartAdd
           INC     HL
           INC     IX
           CMP     IX, sRAMprogEnd
           J       NZ, sRAMLOOP
;#### Set a non-maskable interrupt vector inside the RAM area ####
           LD      HL, 0x01FC            ; Set INTUNDEF and INTSWI interrupt vectors
           LDW     (HL), sINTSWI - sRAMprogStart + cRAMStartAdd
           LD      HL, 0x01F8            ; Set INTWDT interrupt vector
           LDW     (HL), sINTWDT - sRAMprogStart + cRAMStartAdd
;#### Allocate RAM to the code area. Switch the vector area to RAM ####
           LD      (SYSCR3), 0x06        ; Set both <RAREA> and <RVCTR> to "1"
           LD      (SYSCR4), 0xD4        ; Write enable code “0xD4” to SYSCR4
;#### Allocate BOOTROM to the data/code area ####
           LD      (FLSCR1), 0x50        ; Set <BAREA> to "1" (Note 2)
           LD      (FLSCR2), 0xD5        ; Reflect the FLSCR1 setting
;#### Check the status of the Security Program ####
           LD      A, 0xD5               ; Enable Code
           LD      C, 0x00               ; Set 0x00 (Note 1)
           CALL    (.BTGetRP)            ; Check the status of the Security Program
           CMP     A, 0xFF               ; The status of the security is released when the return value A is 0xFF.
           J       NZ, sSKIP             ; Go to sSKIP when the Security Program is enabled
;#### Security Program enable process (API) ####
           LD      A, 0xD5               ; Enable Code
           LD      C, 0x00               ; Set 0x00 (Note 1)
           CALL    (.BTSetRP)            ; Enable the Security Program
sSKIP:     LD      (FLSCR1), 0x40        ; Set <BAREA> to "0"
           LD      (FLSCR2), 0xD5        ; Enable Code
           LD      (SYSCR3), 0x00        ; Set <RAREA> and <RVCTR> to "0"
           LD      (SYSCR4), 0xD4        ; Enable Code
                   :
           J       XXXX
```

```
;#### Program to be executed in RAM ####
sRAMprogStart:
; Interrupt subroutine
sINTWDT:
sINTSWI:    LD      IX, 0xF000
            LD      A, (IX)
            CMP     A, (IX)
            J       NZ, sINTWDT         ; Loop until the read values become the same
            LD      (SYSCR2), 0x10      ; Generate system clock reset
            RETN
sRAMprogEnd:
            NOP
```

Note 1: Make sure that you set the C register to "0x00".

Note 2: Do not allocate the above program to "0x1000" to "0x17FF" in the code area in the Flash memory. When this area is set to <BAREA> = 1, it changes from the Flash memory area to the BOOTROM area so that the program will not function properly and the microcontroller may malfunction.

Note 3: It is not necessary to add DI instruction for above example program, because the support program include it. However, the support program does not include EI instruction. Therefore, when interrupt process is used, enable <IMF> after finishing all above process.

21.5.2.4. How to read data from Flash memory

To read data from Flash memory, execute transfer instruction for memory. It is possible to read the corresponding individual data (include data of code area) to each address in Flash memory, when FLSCR1<FAREA> and FLSCR2<CR1EN> is selected properly.

Example: Case in which data is read from "0xF000" in the code area and stored at "0x98" in RAM

```
LD      (FLSCR1), 0xA8      ; Select AREA C1
LD      (FLSCR2), 0xD5      ; Write enable code “0xD5” to FLSCR2
LD      A, (0xF000)         ; Read data from "0xF000"
LD      (0x98), A           ; Store data at "0x98"
LD      (FLSCR1), 0x40      ; Select AREA D1
LD      (FLSCR2), 0xD5      ; Write enable code “0xD5” to FLSCR2
```

21.6. API (Application Programming Interface)

The BOOTROM has a support program (API) which contains a special subroutine for erasing or writing on the Flash memory. After mapping of the BOOTROM, it allows easy erasing or writing on the Flash memory by only calling the subroutine in BOOTROM. The Table 21-5 shows the list of API.

Table 21-5 List of API

Address	Contents	Using Stack (Note 2)	Working Register (Note 1)	Argument		Return value		
				Register	Setting Value	Register	Contents	
0x1010 (.BTWrite)	Writing the data of one byte to specified address.	7 bytes	WA BC DE IX	WA	Specify the address to be written.	-	-	-
				C	0x00			
				E	Specify the data to be written.			
				(SP-)	0xD5 (Enable Code)			
0x1012 (.BTEraseSec)	Erasing the specified one sector.	4 bytes	WA BC DE IX	A	Specify the sector to be erased.	-	-	-
				C	0xD5 (Enable Code)			
0x1014 (.BTEraseChip)	Executing the Chip Erase.	6 bytes	WA BC DE IX	A	0xD5 (Enable Code)	-	-	-
				C	0x00			
0x1016 (.BTGetRP)	Getting the status of Security Program.	6 bytes	WA BC DE IX	A	0xD5 (Enable Code)	A	0xFF	Security Program disabled.
				C	0x00		Others	Security Program enabled.
0x1018 (.BTSetRP)	Setting the Security Program.	6 bytes	WA BC DE IX	A	0xD5 (Enable Code)	-	-	-
				C	0x00			
0x101E (.BTCalcUART)	Calculating the setting for UART (Baud rate) from the captured value by timer counter.	4 bytes	WA BC DE IX IY	WA	Captured value by timer counter	W	Setting value for <RTSEL>	
				C	The number of bit for calculation.	A	Setting value for UARTxDR	

Note 1: Because working registers (general-purpose registers) are rewritten in the support program, the contents of these registers should be saved before calling the support program.

Note 2: While the support program is executed, a maximum 7 bytes are used as stack which doesn’t include the stack used by interrupts. Therefore, be sure to reserve a stack area beforehand.

Note 3: Each API works properly without the setting Enable Code "0xD5" as argument. However, it is recommended to set the Enable Code "0xD5" to keep compatibility in the family products.

21.6.1. .BTWrite

Data in E register is written into the address specified by WA register. C register should be written "0x00" and (SP-) should be written Enable Code "0xD5" before calling the API.

21.6.2. .BTEraseSec

The sector specified by A register is erased. C register should be written Enable Code "0xD5" before calling the API. Table 21-6 shows the erased sector by this API.

Table 21-6 Erased Sector by setting of A register

A register	Erased area
0x01	0x1000 to 0x1FFF
0x02	0x2000 to 0x2FFF
0x03	0x3000 to 0x3FFF
0x04	0x4000 to 0x4FFF
0x05	0x5000 to 0x5FFF
0x06	0x6000 to 0x6FFF
0x07	0x7000 to 0x7FFF
0x08	0x8000 to 0x8FFF
0x09	0x9000 to 0x9FFF
0x0A	0xA000 to 0xAFFF
0x0B	0xB000 to 0xBFFF
0x0C	0xC000 to 0xCFFF
0x0D	0xD000 to 0xDFFF
0x0E	0xE000 to 0xEFFF
0x0F	0xF000 to 0xFFFF
Others	Reserved

Note: Reserved : Do not set the reserved data into A register. Unexpected sector might be erased by the reserved data setting.

21.6.3. .BTEraseChip

All Flash memory area is erased. C register should be written "0x00" and A register should be written Enable Code "0xD5" before calling the API.

21.6.4. .BTGetRP

The security status of Flash memory can be read out. C register should be written "0x00" and A register should be written Enable Code "0xD5" before calling the API.

After completion of the execution, API returns A register with the contents of "0xFF7F" (security status) in Product ID as return value.

21.6.5. .BTSetRP

The setting of Security Program can be executed by this API. C register should be written "0x00" and A register should be written Enable Code "0xD5" before calling the API.

21.6.6. .BTCalcUART

This API calculates the proper setting for baud rate of UART from the value of C and WA register.

WA register is set the capture value which is captured the external UART signal by the pulse width measurement mode of the 16-bit timer counter. The fcgck / 2 is selected as source clock of the 16-bit timer counter is selected and the pulse for 8 bits of the UART must be captured. C register must be written “0x08”

RXDx pin can be used as TCA0 pin. To capture the value, please select the pin as TCA0 pin by SERSEL<TCA0SEL> temporarily. After capturing, be sure to resume the pin to RXDx pin.

The possible value for WA register as argument is from "0x0020" to "0x3BFF". When the contents of WA register is out of the area "0x0020" to "0x3BFF", the API returns WA register with "0xFFFF" as return value.

In the return value of W register, bits 5, 4 and 3 are suitable value for UARTxCR2<RTSEL> and the return value of A register is a suitable data for UARTxDR. The API sets bits 7, 6, 2, 1 and 0 to "0" as return value. Therefore, set the proper value for UARTxCR2<RXDNC> and <STOPBR>.

Note 1: x = 2 to 0

Note 2: When the captured value of WA register is little (The baud rate is less than one generated by the UART), the proper setting may not be gotten.

The following procedure shows example how to calculate the baud rate for UART in MCU mode by using support program.

(1) By serial interface selection control register SERSEL<TCA0SEL>, assign RXDx pin to TCA0 pin.

(2) Set 16-bit timer counter to pulse width measurement mode. And set falling edge as an external trigger and select fcgck / 2 as the source clock.

(3) Receive data "0x80" via RXDx pin and capture it by 16-bit timer counter. In this case, enabling of UART is no need.

(4) Write the captured value into WA register. Because general-purpose registers (DE, BC, IX and IY) are rewritten in the support program, the contents of these registers should be saved before calling the support program.

(5) Set the interrupt master enable flag to "disable (DI)" (<IMF> ← "0").

(6) Set "0xD5" on FLSCR2<CR1EN> after setting FLSCR1<BAREA> to "1".

(7) Set "0x08" to C register as the number of bit.

(8) Call .BTCalcUART (0x101E).

(9) Set bits 5, 4 and 3 of W register into UARTxCR2<RTSEL> and set the contents of A register to UARTxDR. When the value of WA register is "0xFFFF" which indicates an error of calculation, retry the execution from receiving data 0x80 for adjusting UARTxCR2<RTSEL> and UARTxDR.

(10) Set "0xD5" to FLSCR2<CR1EN> after setting FLSCR1<BAREA> to "0".

Note 1 x = 2 to 0

Note 2: When general-purpose registers (WA, BC, DE, IX, IY) are used in non-maskable interrupt subroutine, occurring of non-maskable interrupt may cause unexpected result.

Note 3: With success of calculation, this API returns "0" into bits 7, 6, 2, 1 and 0 of W resister as return value. Therefore, set proper value for these bits to set UARTxCR2<RXDNC> and <STOPBR>.

Example: Captures the low width of 8 bit value via RXD0 pin by 16-bit timer counter (TCA0) which is set to the pulse width measurement mode. And calculates setting baud rate for UART from the captured value.

```
                .BTCalcUART     equ 0x101E      ; Calculating the setting for UART (Baud rate)

CalcUART section code abs = 0xF000
; #### Receive data (0x80) from a master device ####
                LD      (SERSEL), 0x40          ; Assign RXDx pin to TCA0 pin
; #### Receive data (0x80) from a master device ####
                LD      (TA0MOD), 0x5E          ; Set the pulse width measurement mode and select falling
                                                ; edge as external trigger. Select fcgck / 2 as the source clock
sTimerStart:    LD      (TA0CR), 0x01           ; Timer start
                        :
```

Receives data "0x80" as the adjustment data via TCA0 pin

```
                        :
                LD      WA, (TA0DRBL)           ; Write the captured data into WA register
                LD      (TA0CR), 0x00           ; Timer stop
                DI
; #### Allocate BOOTROM to the data/code area ####
                LD      (FLSCR1), 0x50          ; Set <BAREA> to "1"
                LD      (FLSCR2), 0xD5          ; Write enable code 0xD5 to FLSCR2
; #### Calculation for UART setting (API) ####
                LD      C, 0x08                 ; The number of bit length (8 bit)
                CALL    (.BTCalcUART)           ; Calculate UART setting
                CMP     W, 0xFF
                J       Z, sTimerStart          ; Return to sTimerStart when W register equals "0xFF"
; #### Setting the calculated result to UART registers ####
                LD      (UART0CR2), W           ; SET <RTSEL>
                LD      (UART0DR), A            ; Set UART0DR
; #### End process ####
                LD      (FLSCR1), 0x40          ; Set <BAREA> to "0"
                LD      (FLSCR2), 0xD5          ; Write enable code 0xD5 to FLSCR2
```

22. Serial PROM Mode

22.1. Outline

The TMP89FS60B/62B/63B has a 4K-byte BOOTROM (Mask ROM) for programming to Flash memory. BOOTROM is available in the Serial PROM mode. The Serial PROM mode is controlled by RXD0/SI0/P21 pin, TXD0/SO0/P20 pin, MODE pin, and $\overline{\text{RESET}}$ pin. In the Serial PROM mode, communication is performed via the UART.

Table 22-1 Operating Range in Serial PROM Mode

Parameter	Min	Max	Unit
Power supply voltage	4.5	5.5	V
High frequency (fc)	4	10	MHz

22.2. Security

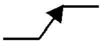
In the Serial PROM mode, two security functions are provided to prevent illegal memory access attempts by a third party: Password and Security Program functions. For more security-related information, refer to "22.12. Security".

22.3. Serial PROM Mode Setting

22.3.1. Serial PROM mode control pins

To execute on-board programming, activate the Serial PROM mode. Table 22-2 shows the pin setting used to activate the Serial PROM mode.

Table 22-2 Serial PROM Mode Setting

Pin	Setting
RXD0/SI0/P21 pin	“High” level
TXD0/SO0/P20 pin	“High” level
MODE, $\overline{\text{RESET}}$ pin	

Note: Before you activate the Serial PROM mode, you must set the RXD0/SI0/P21 and TXD0/SO0/P20 pins to “High” level by using a pull-up resistor.

Table 22-3 Pin Functions in the Serial PROM Mode

Pin name (in the Serial PROM mode)	Input/ output	Function		Pin name (in MCU mode)
TXD0	Output	Serial PROM mode control/serial data output	(Note)	TXD0/SO0/P20
RXD0	Input	Serial PROM mode control/serial data input		RXD0/SI0/P21
$\overline{\text{RESET}}$	Input	Serial PROM mode control		$\overline{\text{RESET}}$
MODE	Input	Serial PROM mode control		MODE
VDD	Power supply	4.5 to 5.5 [V]		
AVDD	Power supply	Connect to VDD.		
VSS	Power supply	0 [V]		
AVSS	Power supply	Connect to VSS.		
VAREF	Power supply	Apply reference voltage.		
Input/output port other than RXD0 and TXD0	Input/ output	These ports are in the high-impedance state in the Serial PROM mode. The port input is physically fixed to a specified input level in order to prevent a through current (the port input is disabled).		
XIN	Input	Connect a resonator to make these pins self-oscillate.		
XOUT	Output			

Note : When other parts are mounted on a user board, they may interfere with data being communicated through these communication pins during on-board programming. It is recommended that these parts be somehow isolated to prevent the pins from being affected.

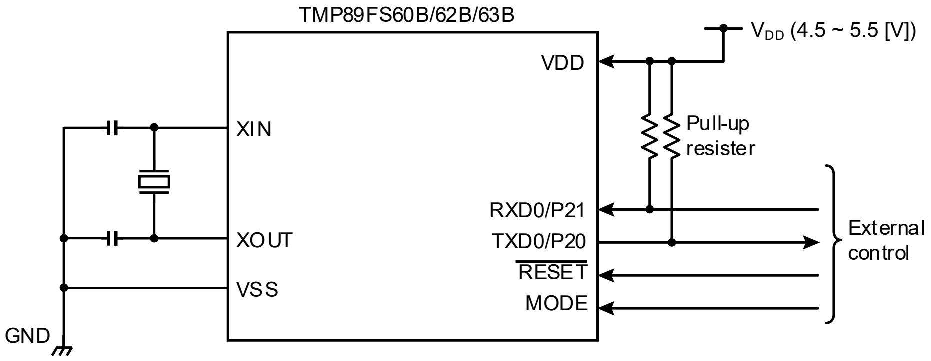


Figure 22-1 Serial PROM Mode Pin Setting

Note: For information on other pin settings, refer to "Table 22-3".

22.4. Example Connection for On-board Writing

Figure 22-2 shows example connections to perform on-board writing.

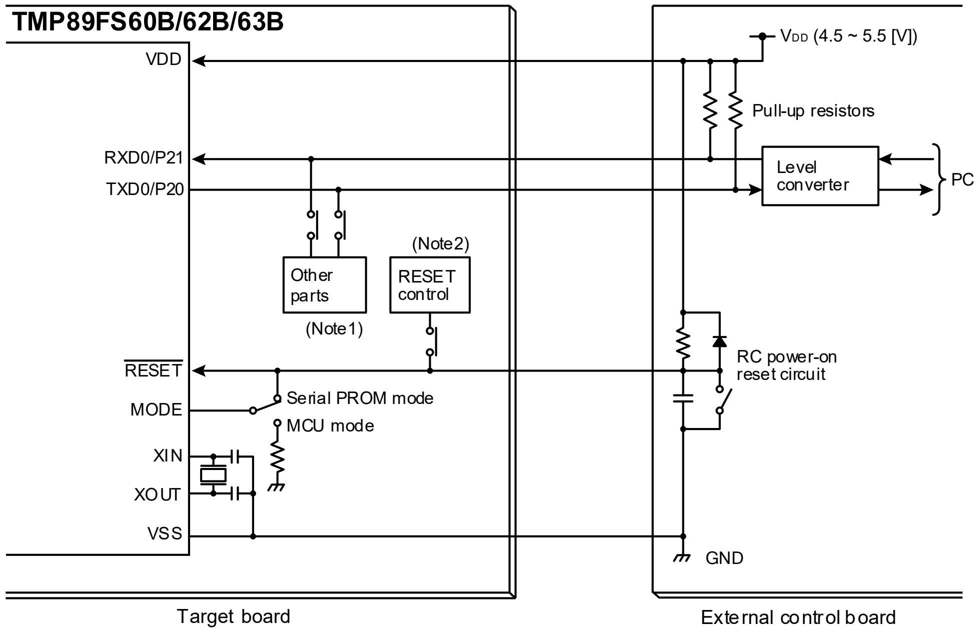


Figure 22-2 Example Connections for On-board Writing

Note 1: When other parts on a target board interfere with the UART communication in the Serial PROM mode, disconnect these pins by using a jumper or switch.

Note 2: When the reset control circuit on a target board interferes with the startup of the Serial PROM mode, disconnect the circuit by using a jumper, etc.

Note 3: For information on other pin settings, refer to "Table 22-3".

22.5. Activating the Serial PROM Mode

Activate the Serial PROM mode by performing the following procedure. For information on the detailed timing, refer to "22.14.1. Reset timing".

(1) Supply power to the VDD pin.

(2) Set the $\overline{\text{RESET}}$ and MODE pins to the “Low” level.

(3) Set the RXD0/SI0/P21 and TXD0/SO0/P20 pins to the “High” level.

(4) Wait until the power supply and clock oscillation stabilize.

(5) Set the $\overline{\text{RESET}}$ and MODE pins from the “Low” level to the “High” level.

(6) Input the matching data "0x86" to the RXD0/P21 pins after the setup period has elapsed.

22.6. Interface Specifications

The Serial PROM mode supports UART as communication method.

To execute an on-board program, the communication format of the external controller (personal computer, microcontroller, etc.) must be set as described below.

22.6.1. UART communication

The transfer format of the UART communication is shown below;

- Baud rate: 9600 to 128000 [baud] (automatic detection)
- Data length: 8 bits (LSB first)
- Parity bit: None
- STOP bit: 1 bit

When the TMP89FS60B/62B/63B receives serial data "0x86" after a reset, it starts the UART communication. It also measures the pulse width of the received data "0x86", and automatically establishes the reference baud rate. In all subsequent data communication transactions, this reference baud rate is used. For information on the communication timings of each operation command, refer to "22.14. AC Characteristics (UART)".

Usable baud rates differ depending on the operating frequency and are shown in Table 22-4. However, there is the possibility of data communication not working properly, even when a baud rate shown in Table 22-4 is used, because data communication is affected by frequency errors of a resonator of the external controller (personal computer, etc.), the load capacity of a communication pin, and various other factors.

Table 22-4 Usable Baud Rates as a General Guideline

fc [MHz]	9600 [baud]	19200 [baud]	38400 [baud]	57600 [baud]	115200 [baud]	128000 [baud]
10 [MHz]	A	A	A	A	A	A
8 [MHz]	A	A	A	A	A	A
7.3728 [MHz]	A	A	A	A	A	NA
6.144 [MHz]	A	A	A	NA	NA	A
6 [MHz]	A	A	A	A	A	A
5 [MHz]	A	A	A	NA	NA	NA
4.9152 [MHz]	A	A	A	A	NA	NA
4.19 [MHz]	A	A	A	NA	NA	A
4 [MHz]	A	A	A	A	A	A

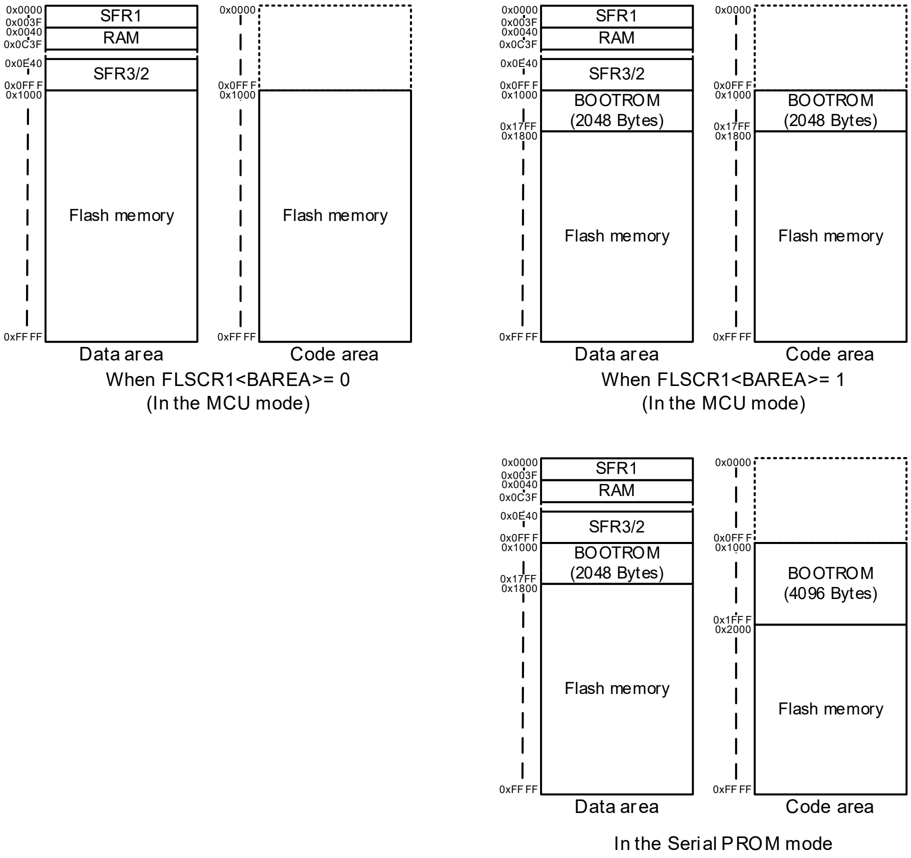
Note: A: Available. NA: Not available

22.7. Memory Mapping

Figure 22-3 shows memory maps in the Serial PROM and MCU modes.

In the Serial PROM mode, the BOOTROM (Mask ROM) is mapped to the "0x1000" to "0x17FF" in the data area and "0x1000" to "0x1FFF" in the code area respectively.

When the command to write data to Flash memory (hereafter called the "0x30" command) or the command to erase data from Flash memory (hereafter called the "0xF0" command) is executed, BOOTROM automatically converts addresses. Therefore, as the address of Flash memory, specify an address equivalent to that specified in MCU mode (when FLSCR1<BAREA> = 0), namely, "0x1000" to "0xFFFF".



When FLSCR1<BAREA>= 0
(In the MCU mode)

When FLSCR1<BAREA>= 1
(In the MCU mode)

In the Serial PROM mode

Figure 22-3 Memory Mapping

22.8. Operation Commands

In the Serial PROM mode, the commands shown in Table 22-5 are used. After a reset is released, the TMP89FS60B/62B/63B goes into a standby state and awaits the arrival of matching data 1 ("0x86").

Table 22-5 Operation Command in Serial PROM Mode

Command data	Operation command	Description
0x86, 0x79	Setup (matching data 1, 2)	After a reset is released, the Serial PROM mode always starts operation with this command.
0xF0	Flash memory erase	Data in the Flash memory area (address "0x1000" to "0xFFFF") can be erased.
0x30	Flash memory write	Data can be written to the Flash memory area (address "0x1000" to "0xFFFF").
0x40	Flash memory read	Data can be read from the Flash memory area (address "0x1000" to "0xFFFF").
0x90	Flash memory SUM output	0xFF check data and 2-byte checksums of the entire Flash memory area (address "0x1000" to "0xFFFF") are output in descending order (from upper to lower bytes).
0xC0	Product ID code output	Product ID codes are output.
0xC3	Flash memory status output	The Security Program status and other status codes are output.
0xFA	Flash memory security setting	The Security Program setting is enabled.

Each command is outlined below. For detailed information on how each command works, refer to "22.8.1. Flash memory erase command (operation command: 0xF0)" and subsequent sections.

(1) Flash memory erase command

Either Chip Erase (erase of entire Flash memory) or Sector Erase (erase of Flash memory in 4K-byte units) can be used to erase the data in the Flash memory. Data in the erased area is "0xFF".

When the Security Program is enabled or when the option code EPFC_OP is set to "0xFF", the Flash memory erase command of the Chip Erase can be executed but the Flash memory erase command of Sector Erase cannot be executed.

To disable the Security Program setting, execute the Chip Erase of the Flash memory erase command.

The TMP89FS60B/62B/63B performs password authentication in the Flash memory erase command except when it is a blank product or setting the option code EPFC_OP is "0xFF". When a password is not authenticated, the Flash memory erase command is pretermitted and the erasing operation is not executed.

(2) Flash memory write command

Data can be written in single-byte units to a specified address in the Flash memory. Provision the external controller so that it transmits data to write as binary data in the Intel Hex format. When the end record completes to be transmitted, the TMP89FS60B/62B/63B calculates checksums in the entire Flash memory area ("0x1000" to "0xFFFF"), and returns the calculation results.

When the Security Program is enabled, the Flash memory write command cannot be executed.

To disable the Security Program setting, execute the Chip Erase of the Flash memory erase command.

The TMP89FS60B/62B/63B performs password authentication in the Flash memory write command except when it is a blank product. When a password is not authenticated, the Flash memory write command is pretermitted and the writing operation is not executed.

(3) Flash memory read command

Data can be read from a specified address in the Flash memory in single-byte units. Provision the external controller so that it transmits the address in memory where a read starts, as well as the number of bytes. After the data of specified bytes output, the TMP89FS60B/62B/63B calculates the checksums of the output data, and returns the calculation results.

When the Security Program is enabled, the Flash memory read command cannot be executed.

To disable the Security Program setting, execute the Chip Erase of the Flash memory erase command.

The TMP89FS60B/62B/63B performs password authentication in the Flash memory read command except when it is a blank product. When a password is not authenticated, the Flash memory read command is pretermitted and the reading operation is not executed.

(4) Flash memory SUM output command

Checksums in the entire Flash memory area ("0x1000" to "0xFFFF") are calculated, and the calculation results are returned.

(5) Product ID code output command

A code to identify a product is output. The output code consists of information on the ROM area and in the RAM area respectively. The external controller reads this code to identify the product to which data is to be written.

(6) Flash memory status output command

The status of "0xFFE0" to "0xFFFF" and that of the Security Program are output. The external controller reads this code to identify the status of Flash memory.

(7) Flash memory security setting command

This command is used to set the Security Program.

When the Security Program is set, the operation of the TMP89FS60B/62B/63B is shown below;

– Parallel mode

  Prohibit the reading and writing of data.

– Serial PROM mode

  Prohibit the Flash memory write command and the Flash memory read command.

– On-chip debug (OCD) function

  Skipping the Security Program check and only authenticating a password, or executing the Security Program check is selected by setting of option code DAFC_OP. In case of checking the Security Program, the OCD function cannot be activated when the Security Program is enabled, and it can be activated with authenticating a password.

To disable the Security Program setting, execute the Chip Erase of the Flash memory erase command.

22.8.1. Flash memory erase command (operation command: 0xF0)

Table 22-6 shows the Flash memory erase commands.

Table 22-6 Flash Memory Erase Commands

	Transfer byte	Transfer data from the external controller to TMP89FS60B/62B/63B	Baud rate	Transfer data from TMP89FS60B/62B/63B to the external controller
BOOT ROM	1st byte 2nd byte	Matching data 1 "0x86" -	Automatic adjustment Baud rate after adjustment	- (Automatic baud rate adjustment) OK: Echo back data "0x86" Error: No data transmitted
	3rd byte 4th byte	Matching data 2 "0x79" -	Baud rate after adjustment Baud rate after adjustment	- OK: Echo back data "0x79" Error: No data transmitted
	5th byte 6th byte	Operation command data "0xF0" -	Baud rate after adjustment Baud rate after adjustment	- OK: Echo back data "0xF0" Error: "0xA1" × 3, "0xA3" × 3, "0x63" × 3 (Note 1)
	7th byte 8th byte	Password count storage address bits 23 to 16	Baud rate after adjustment Baud rate after adjustment	- OK: No data transmitted Error: No data transmitted
	9th byte 10th byte	Password count storage address bits 15 to 8	Baud rate after adjustment Baud rate after adjustment	- OK: No data transmitted Error: No data transmitted
	11th byte 12th byte	Password count storage address bits 7 to 0	Baud rate after adjustment Baud rate after adjustment	- OK: No data transmitted Error: No data transmitted
	13th byte 14th byte	Password comparison start address bits 23 to 16	Baud rate after adjustment Baud rate after adjustment	- OK: No data transmitted Error: No data transmitted
	15th byte 16th byte	Password comparison start address bits 15 to 8	Baud rate after adjustment Baud rate after adjustment	- OK: No data transmitted Error: No data transmitted
	17th byte 18th byte	Password comparison start address bits 7 to 0	Baud rate after adjustment Baud rate after adjustment	- OK: No data transmitted Error: No data transmitted
	19th byte : m-th byte	Password string (Note 3) - -	Baud rate after adjustment Baud rate after adjustment	- OK: No data transmitted Error: No data transmitted
	(n-2)-th byte	Erase sector start address	Baud rate after adjustment	-
	(n-1)-th byte	-	Baud rate after adjustment	OK: Checksum (higher) Error: No data transmitted
	n-th byte	-	Baud rate after adjustment	OK: Checksum (lower) Error: No data transmitted
	(n+1)-th byte	(Wait for the next operation command data)	Baud rate after adjustment	-

Note 1: "0x**" × 3 means that the TMP89FS60B/62B/63B stops communication after transmitting 3 bytes of "0x**". For further information, refer to Table 22-18.

Note 2: For information on the erase area specification, refer to "22.8.1.1. Specifying the erase area". For information on checksums, refer to "22.10. Checksum (SUM)". For information on passwords, refer to "22.12.1. Passwords".

Note 3: Do not transmit a password string when the TMP89FS60B/62B/63B is a blank product or when setting data of option code EPFC_OP is “0xFF”. (However, the password count storage address and the password comparison start address must be transmitted.)

Note 4: When the setting data of option code EPFC_OP is “0xFF” and the Sector Erase is executed (a value less than "0x20" is transmitted at the (n-2)-th byte), the TMP89FS60B/62B/63B stops communication.

Note 5: When a password error occurs, the TMP89FS60B/62B/63B stops communication. Therefore, when a password error occurs, initialize the TMP89FS60B/62B/63B by using the $\overline{\text{RESET}}$ pin, and restart the Serial PROM mode.

Note 6: When a communication error occurs during the transfer of a password address or a password string, the TMP89FS60B/62B/63B stops communication. Therefore, when a communication error occurs, initialize the TMP89FS60B/62B/63B by using the $\overline{\text{RESET}}$ pin, and restart the Serial PROM mode.

22.8.1.1. Specifying the erase area

The Flash memory erase command is used to specify an area in the Flash memory to be erased at (n-2)-th byte.

The ERASEC is used to specify the erase sector start address.

When data of less than "0x20" is specified, the Sector Erase (erasing the Flash memory in 4K-byte units) is executed is disabled.

When data of "0x20" or more is specified, the Chip Erase (erasing the entire Flash memory) is executed, and the Security Program of the Flash memory.

Executing the Sector Erase with data of the option code EPFC_OP which is set to "0xFF" or with the Security Program which is enabled will cause the TMP89FS60B/62B/63B goes into an endless loop and stops communication.

Erase area specification data (data at (n-2)-th byte)

7	6	5	4	3	2	1	0
ERASEC							

ERASEC	Erase sector start address	0x00:	Reserved
		0x01:	0x1000 to 0x1FFF
		0x02:	0x2000 to 0x2FFF
		0x03:	0x3000 to 0x3FFF
		0x04:	0x4000 to 0x4FFF
		0x05:	0x5000 to 0x5FFF
		0x06:	0x6000 to 0x6FFF
		0x07:	0x7000 to 0x7FFF
		0x08:	0x8000 to 0x8FFF
		0x09:	0x9000 to 0x9FFF
		0x0A:	0xA000 to 0xAFFF
		0x0B:	0xB000 to 0xBFFF
		0x0C:	0xC000 to 0xCFFF
		0x0D:	0xD000 to 0xDFFF
		0x0E:	0xE000 to 0xEFFF
		0x0F:	0xF000 to 0xFFFF
		0x10:	Reserved
		0x11:	Reserved
		0x12:	Reserved
		0x13:	Reserved
		0x14:	Reserved
		0x15:	Reserved
		0x16:	Reserved
		0x17:	Reserved
		0x18:	Reserved
		0x19:	Reserved
		0x1A:	Reserved
		0x1B:	Reserved
		0x1C:	Reserved
		0x1D:	Reserved
		0x1E:	Reserved
		0x1F:	Reserved
		0x20 or more	Chip Erase (entire Flash memory)

Note 1: When Sector Erase is performed on an area where Flash memory does not exist, the TMP89FS60B/62B/63B stops communication.

Note 2: When Reserved data is transmitted, the TMP89FS60B/62B/63B stops communication.

22.8.2. Flash memory write command (operation command: 0x30)

Table 22-7 shows the transfer formats of Flash memory write commands.

Table 22-7 Transfer Formats of Flash Memory Write Commands

	Transfer byte	Transfer data from the external controller to TMP89FS60B/62B/63B	Baud rate	Transfer data from TMP89FS60B/62B/63B to the external controller
BOOT ROM	1st byte 2nd byte	Matching data 1 "0x86" -	Automatic adjustment Baud rate after adjustment	- (Automatic baud rate adjustment) OK: Echo back data "0x86" Error: No data transmitted
	3rd byte 4th byte	Matching data 2 "0x79" -	Baud rate after adjustment Baud rate after adjustment	- OK: Echo back data "0x79" Error: No data transmitted
	5th byte 6th byte	Operation command data "0x30" -	Baud rate after adjustment Baud rate after adjustment	- OK: Echo back data "0x30" Error: "0xA1" × 3, "0xA3" × 3, "0x63" × 3 (Note 1)
	7th byte 8th byte	Password count storage address bits 23 to 16	Baud rate after adjustment Baud rate after adjustment	- OK: No data transmitted Error: No data transmitted
	9th byte 10th byte	Password count storage address bits 15 to 8	Baud rate after adjustment Baud rate after adjustment	- OK: No data transmitted Error: No data transmitted
	11th byte 12th byte	Password count storage address bits 7 to 0	Baud rate after adjustment Baud rate after adjustment	- OK: No data transmitted Error: No data transmitted
	13th byte 14th byte	Password comparison start address bits 23 to 16	Baud rate after adjustment Baud rate after adjustment	- OK: No data transmitted Error: No data transmitted
	15th byte 16th byte	Password comparison start address bits 15 to 8	Baud rate after adjustment Baud rate after adjustment	- OK: No data transmitted Error: No data transmitted
	17th byte 18th byte	Password comparison start address bits 7 to 0	Baud rate after adjustment Baud rate after adjustment	- OK: No data transmitted Error: No data transmitted
	19th byte : m-th byte	Password string (Note 3) -	Baud rate after adjustment Baud rate after adjustment	- OK: No data transmitted Error: No data transmitted
	m-th + 1byte : (n-3)-th byte	Intel Hex format (binary) (Note 7) Intel Hex format (binary) (Note 7)	Baud rate after adjustment Baud rate after adjustment	- -
	(n-2)-th byte	-	Baud rate after adjustment	Transmit 0x55
	(n-1)-th byte	-	Baud rate after adjustment	OK: Checksum (higher) Error: No data transmitted
	n-th byte	-	Baud rate after adjustment	OK: Checksum (lower) Error: No data transmitted
	(n+1)-th byte	(Wait for the next operation command data)	Baud rate after adjustment	-

Note 1: "0x**" × 3 means that the TMP89FS60B/62B/63Bdevice stops communication after transmitting 3 bytes of "0x**". For further information, refer to Table 22-18.

Note 2: For information on the Intel Hex format, refer to "22.11. Intel Hex Format (Binary)". For information on checksums, refer to "22.10. Checksum (SUM)". For information on passwords, refer to "22.12.1. Passwords".

Note 3: When the area "0xFFE0" to "0xFFFF" is all "0xFF", password authentication is not performed and, therefore, the password string need not be transmitted. The password count storage address and password comparison start address, however, must be specified, even in this case. when the password count storage address and/or password comparison start address is/are incorrect, a password error occurs, the TMP89FS60B/62B/63B stops communication. Therefore, when a password error occurs, initialize the TMP89FS60B/62B/63B by using the $\overline{\text{RESET}}$ pin, and restart the Serial PROM mode.

Note 4: When the Security Program is enabled in Flash memory or when a password error occurs, the TMP89FS60B/62B/63B stops communication. Therefore, when a password error occurs, initialize the TMP89FS60B/62B/63B by using the $\overline{\text{RESET}}$ pin, and restart the Serial PROM mode.

Note 5: When a communication error occurs during the transfer of a password address or a password string, the TMP89FS60B/62B/63B stops communication. Therefore, when a communication error occurs, initialize the TMP89FS60B/62B/63B by using the $\overline{\text{RESET}}$ pin, and restart the Serial PROM mode.

Note 6: When all data in Flash memory are the same data, do not write data to the address "0xFFE0" to "0xFFFF". When data is written to this address, a password error occurs

Note 7:The period which is 3 [ms] or more is required after a completion of the transmitting data record. For details, refer to Figure 22-8.

Note 8:When the data is rewritten to the address already written a data (including 0xFF), that address must be erased beforehand by the Sector Erase or Chip Erase command.

22.8.3. Flash memory read command (operation command: 0x40)

Table 22-8 and Table 22-9 show the transfer formats of the Flash memory read command.

Table 22-8 Transfer Formats of the Flash Memory Read Command

	Transfer byte	Transfer data from the external controller to TMP89FS60B/62B/63B	Baud rate	Transfer data from TMP89FS60B/62B/63B to the external controller
BOOT ROM	1st byte 2nd byte	Matching data 1 "0x86" -	Automatic adjustment Baud rate after adjustment	- (Automatic baud rate adjustment) OK: Echo back data "0x86" Error: No data transmitted
	3rd byte 4th byte	Matching data 2 "0x79" -	Baud rate after adjustment Baud rate after adjustment	- OK: Echo back data "0x79" Error: No data transmitted
	5th byte 6th byte	Operation command data "0x40" -	Baud rate after adjustment Baud rate after adjustment	- OK: Echo back data "0x40" Error: "0xA1" × 3, "0xA3" × 3, "0x63" × 3 (Note 1)
	7th byte 8th byte	Password count storage address bits 23 to 16	Baud rate after adjustment Baud rate after adjustment	- OK: No data transmitted Error: No data transmitted
	9th byte 10th byte	Password count storage address bits 15 to 8	Baud rate after adjustment Baud rate after adjustment	- OK: No data transmitted Error: No data transmitted
	11th byte 12th byte	Password count storage address bits 7 to 0	Baud rate after adjustment Baud rate after adjustment	- OK: No data transmitted Error: No data transmitted
	13th byte 14th byte	Password comparison start address bits 23 to 16	Baud rate after adjustment Baud rate after adjustment	- OK: No data transmitted Error: No data transmitted
	15th byte 16th byte	Password comparison start address bits 15 to 8	Baud rate after adjustment Baud rate after adjustment	- OK: No data transmitted Error: No data transmitted
	17th byte 18th byte	Password comparison start address bits 07 to 0	Baud rate after adjustment Baud rate after adjustment	- OK: No data transmitted Error: No data transmitted
	19th byte : m-th byte	Password string (Note 3) -	Baud rate after adjustment Baud rate after adjustment	- OK: No data transmitted Error: No data transmitted
	(m+1)-th byte (m+2)-th byte	Read start address 23 to 16	Baud rate after adjustment Baud rate after adjustment	- OK: No data transmitted Error: No data transmitted
	(m+3)-th byte (m+4)-th byte	Read start address 15 to 8	Baud rate after adjustment Baud rate after adjustment	- OK: No data transmitted Error: No data transmitted
	(m+5)-th byte (m+6)-th byte	Read start address 7 to 0	Baud rate after adjustment Baud rate after adjustment	- OK: No data transmitted Error: No data transmitted
	(m+7)-th byte (m+8)-th byte	Number of bytes to read 23 to 16	Baud rate after adjustment Baud rate after adjustment	- OK: No data transmitted Error: No data transmitted
	(m+9)-th byte (m+10)-th byte	Number of bytes to read 15 to 8	Baud rate after adjustment Baud rate after adjustment	- OK: No data transmitted Error: No data transmitted

Table 22-9 Transfer Formats of the Flash Memory Read Command

	Transfer byte	Transfer data from the external controller to TMP89FS60B/62B/63B	Baud rate	Transfer data from TMP89FS60B/62B/63B to the external controller
BOOT ROM	(m+11)-th byte (m+12)-th byte	Number of bytes to read 7 to 0	Baud rate after adjustment Baud rate after adjustment	- OK: No data transmitted Error: No data transmitted
	(m+13)-th byte : (n-2)-th byte		Baud rate after adjustment Baud rate after adjustment	Memory data Memory data
	(n-1)-th byte	-	Baud rate after adjustment	OK: Checksum (higher) Error: No data transmitted
	n-th byte	-	Baud rate after adjustment	OK: Checksum (lower) Error: No data transmitted
	(n+1)-th byte	(Wait for the next operation command data)	Baud rate after adjustment	-

Note 1: "0x**" × 3 means that the TMP89FS60B/62B/63B stops communication after transmitting 3 bytes of "0x**". For further information, refer to Table 22-18.

Note 2: For information on checksums, refer to "22.10. Checksum (SUM)". For information on passwords, refer to "22.12.1. Passwords".

Note 3: When the area "0xFFE0" to "0xFFFF" is all "0xFF", password authentication is not performed and, therefore, the password string need not be transmitted. The password count storage address and password comparison start address, however, must be specified, even in this case. when the password count storage address and/or password comparison start address is/are incorrect, a password error occurs, the TMP89FS60B/62B/63B stops communication. Therefore, when a password error occurs, initialize the TMP89FS60B/62B/63B by using the $\overline{\text{RESET}}$ pin, and restart the Serial PROM mode.

Note 4: When the Security Program is enabled in Flash memory or when a password error occurs, the TMP89FS60B/62B/63B stops communication. Therefore, when a password error occurs, initialize the TMP89FS60B/62B/63B by using the $\overline{\text{RESET}}$ pin, and restart the Serial PROM mode.

Note 5: When a communication error occurs during the transfer of a password address or a password string, the TMP89FS60B/62B/63B stops communication. Therefore, when a communication error occurs, initialize the TMP89FS60B/62B/63B by using the $\overline{\text{RESET}}$ pin, and restart the Serial PROM mode.

Note 6: When the number of bytes received at the (m+7)-th byte to (m+12)-th byte is “0x000000” or more than the size of internal memory, the TMP89FS60B/62B/63B stops communication.

22.8.4. Flash memory SUM output command (operation command: 0x90)

Table 22-10 shows the transfer formats of the Flash memory SUM output command.

Table 22-10 Transfer Formats of the Flash Memory SUM Output Command

	Transfer byte	Transfer data from the external controller to TMP89FS60B/62B/63B	Baud rate	Transfer data from TMP89FS60B/62B/63B to the external controller
BOOT ROM	1st byte 2nd byte	Matching data 1 "0x86" -	Automatic adjustment Baud rate after adjustment	- (Automatic baud rate adjustment) OK: Echo back data "0x86" Error: No data transmitted
	3rd byte 4th byte	Matching data 2 "0x79" -	Baud rate after adjustment Baud rate after adjustment	- OK: Echo back data "0x79" Error: No data transmitted
	5th byte 6th byte	Operation command data "0x90" -	Baud rate after adjustment Baud rate after adjustment	- OK: Echo back data "0x90" Error: "0xA1" × 3, "0xA3" × 3, "0x63" × 3 (Note 1)
	7th byte	-	Baud rate after adjustment	0x55: - 0xAA: All data are "0xFF". (Note 2)
	8th byte	-	Baud rate after adjustment	OK: Checksum (higher) (Note 3) Error: No data transmitted
	9th byte	-	Baud rate after adjustment	OK: Checksum (lower) (Note 3) Error: No data transmitted
	10th byte	(Wait for the next operation command data)	Baud rate after adjustment	-

Note 1: "0x**" × 3 means that the TMP89FS60B/62B/63B stops communication after transmitting 3 bytes of "0x**". For further information, refer to Table 22-18.

Note 2: When data to be included in the checksum are all "0xFF", the 7th byte becomes "0xAA". When any one piece of data to be included in the checksum is other than "0xFF", the 7th byte becomes "0x55".

Note 3: For information on checksums, refer to "22.10. Checksum (SUM)".

22.8.5. Product ID code output command (operation command: 0xC0)

Table 22-11 shows the transfer formats of the product ID code output command.

Table 22-11 Transfer Formats of the Product ID Code Output Command

	Transfer byte	Transfer data from the external controller to TMP89FS60B/62B/63B	Baud rate	Transfer data from TMP89FS60B/62B/63B to the external controller	
BOOT ROM	1st byte 2nd byte	Matching data 1 "0x86" -	Automatic adjustment Baud rate after adjustment	- (Automatic baud rate adjustment) OK: Echo back data "0x86" Error: No data transmitted	
	3rd byte 4th byte	Matching data 2 "0x79" -	Baud rate after adjustment Baud rate after adjustment	- OK: Echo back data "0x79" Error: No data transmitted	
	5th byte 6th byte	Operation command data "0xC0" -	Baud rate after adjustment Baud rate after adjustment	- OK: Echo back data "0xC0" Error: "0xA1" × 3, "0xA3" × 3, "0x63" × 3 (Note 1)	
	7th byte		Baud rate after adjustment	0x3A	Start mark
	8th byte		Baud rate after adjustment	0x13	Number of transfer data (from 9th to 27th bytes)
	9th byte		Baud rate after adjustment	0x03	Length of address (3 bytes)
	10th byte		Baud rate after adjustment	0xFD	Reserved
	11th byte		Baud rate after adjustment	0x08	Reserved
	12th byte		Baud rate after adjustment	0x00	Reserved
	13th byte		Baud rate after adjustment	0x02	Reserved
	14th byte		Baud rate after adjustment	0xF0	Read the value as shown on the left.
	15th byte		Baud rate after adjustment	0x01	Read the value as shown on the left.
	16th byte		Baud rate after adjustment	0x00	Read the value as shown on the left.
	17th byte		Baud rate after adjustment	0x10	Read the value as shown on the left.
	18th byte		Baud rate after adjustment	0x00	Read the value as shown on the left.
	19th byte		Baud rate after adjustment	0x00	Read the value as shown on the left.
	20th byte		Baud rate after adjustment	0xFF	Read the value as shown on the left.
	21st byte		Baud rate after adjustment	0xFF	Read the value as shown on the left.
	22nd byte		Baud rate after adjustment	0x00	Read the value as shown on the left.
	23rd byte		Baud rate after adjustment	0x00	Read the value as shown on the left.
	24th byte		Baud rate after adjustment	0x60	Read the value as shown on the left.
	25th byte		Baud rate after adjustment	0x00	Read the value as shown on the left.
	26th byte		Baud rate after adjustment	0x0C	Read the value as shown on the left.
	27th byte		Baud rate after adjustment	0x3F	Read the value as shown on the left.
	28th byte		Baud rate after adjustment	0xYY	0xYY: Checksum of transfer data (complement of 2 of the sum total from 9th to 27th bytes)
	29th byte	(Wait for the next operation command data)	Baud rate after adjustment	-	

Note : "0x**" × 3 means that the TMP89FS60B/62B/63B stops communication after transmitting 3 bytes of "0x**". For further information, refer to Table 22-18.

22.8.6. Flash memory status output command (operation command: 0xC3)

Table 22-12 shows the Flash memory status output commands.

Table 22-12 Flash Memory Status Output Commands

	Transfer byte	Transfer data from the external controller to TMP89FS60B/62B/63B	Baud rate	Transfer data from TMP89FS60B/62B/63B to the external controller	
BOOT ROM	1st byte 2nd byte	Matching data 1 "0x86" -	Automatic adjustment Baud rate after adjustment	- (Automatic baud rate adjustment) OK: Echo back data "0x86" Error: No data transmitted	
	3rd byte 4th byte	Matching data 2 "0x79" -	Baud rate after adjustment Baud rate after adjustment	- OK: Echo back data "0x79" Error: No data transmitted	
	5th byte 6th byte	Operation command data "0xC3" -	Baud rate after adjustment Baud rate after adjustment	- OK: Echo back data "0xC3" Error: "0xA1" × 3, "0xA3" × 3, "0x63" × 3 (Note 1)	
	7th byte		Baud rate after adjustment	0x3A	Start mark
	8th byte		Baud rate after adjustment	0x04	Byte count (from 9th to 12th bytes)
	9th byte		Baud rate after adjustment	0x00 to 0x7F	Status code 1 (Note 2)
	10th byte		Baud rate after adjustment	0x00	Reserved
	11th byte		Baud rate after adjustment	0x00	Reserved
	12th byte		Baud rate after adjustment	0x00	Reserved
	13th byte		Baud rate after adjustment	Checksum (complement of 2 of the sum total from 9th to 12th byte)	
	14th byte	(Wait for the next operation command data)	Baud rate after adjustment	-	

Note 1: "0x**" × 3" means that the TMP89FS60B/62B/63B stops communication after transmitting 3 bytes of "0x**". For further information, refer to Table 22-18.

Note 2: For detailed information on the Status code 1, refer to "22.8.6.1. Flash memory status code".

22.8.6.1. Flash memory status code

The Flash memory status code is 7-byte data. The 3rd byte: Status code 1 shows the status of setting data of option code EPFC_OP and DAFC_OP, that of the Security Program and that of the address from "0xFFE0" to "0xFFFF".

Table 22-13 Flash Memory Status Code

Data	Description	when TMP89FS60B/62B/63B	
1st byte	Start mark	0x3A	
2nd byte	Number of transfer data (4 bytes from 3rd to 6th bytes)	0x04	
3rd byte	Status code 1	0x00 to 0x7F (Note 2)	
4th byte	Reserved	0x00	
5th byte	Reserved	0x00	
6th byte	Reserved	0x00	
7th byte	Checksum of transfer data (complement of 2 of the sum total of 3rd to 6th bytes)	3rd byte data	Checksum
		0x00	0x00
		0x01	0xFF
		0x02	0xFE
		:	:
		0x7F	0x81

Status code 1

	7	6	5	4	3	2	1	0
	-	-	-	-	EPFC	DAFC	RPENA	BLANK
Read/Write	R	R	R	R	R	R	R	R
After reset	0	*	*	*	*	*	*	*

EPFC	Password string judgment when the Flash memory erase command is executed	0:	Skips the authentication of a password string (to judge PNSA and PCSA only) When data of option code EPFC_OP is “0xFF”, <EPFC> is cleared to “0”.
		1:	Authenticates a password string and to judge PNSA and PCSA When data of option code EPFC_OP is not “0xFF”. <EPFC> is set to “1”.
DAFC	Security Program check at start of the on-chip debugging function (OCD)	0:	Skips the Security Program check at the start of OCD function When data of option code DAFC_OP is “0xFF”, <DAFC> is cleared to “0”.
		1:	Performs the Security Program check at the start of OCD function When data of option code DAFC_OP is not “0xFF”, <DAFC> is set to “1”.
RPENA	Status of Security Program	0:	Status in which the Security Program is disabled
		1:	Status in which the Security Program is enabled
BLANK	Status data from "0xFFE0" to "0xFFFF" in the Flash memory	0:	When data in the area "0xFFE0" to "0xFFFF" are all "0xFF"
		1:	When data in the area "0xFFE0" to "0xFFFF" are other than "0xFF"

Note 1: *: Undefined

Note 2: The bits 6 to 4 are read as undefined.

Restrictions are placed on the execution of some operation commands, depending on the contents of the status code 1. Detailed information on this is shown in the table below.

When the Security Program is enabled (RPENA = 1), 3 commands cannot be executed: Sector Erase of “Flash memory erase command”, "Flash memory write command" and “Flash memory read command". To execute these commands, the Security Program must be disabled (RPENA = 0) by executing Chip Erase of “Flash memory erase command”.

Table 22-14 Status Code 1 and Operation Command (1/3)

<RPENA>	<BLANK>	<EPFC>	<DAFC>	Flash memory erase		Flash memory write	Flash memory read
				Chip erase	Sector erase		
0	0	0	0	Executable	Not executable	Executable	Executable
1	0	0	0	Executable	Not executable	Not executable	Not executable
0	1	0	*	Executable	Not executable	Password	Password
		1	*	Password		Password	Password
1	1	0	*	Executable	Not executable	Not executable	Not executable
		1	*	Password	Not executable	Not executable	Not executable

Table 22-15 Status Code 1 and Operation Command (2/3)

<RPENA>	<BLANK>	<EPFC>	<DAFC>	Flash memory sum output	Flash memory ID code output
0	0	0	0	Executable	Executable
1	0	0	0	Executable	Executable
0	1	0	*	Executable	Executable
		1	*	Executable	Executable
1	1	0	*	Executable	Executable
		1	*	Executable	Executable

Table 22-16 Status Code 1 and Operation Command (3/3)

<RPENA>	<BLANK>	<EPFC>	<DAFC>	Flash memory status output	Flash memory security setting
0	0	0	0	Executable	Not executable
1	0	0	0	Executable	Not executable
0	1	0	*	Executable	Password
		1	*	Executable	Password
1	1	0	*	Executable	Password
		1	*	Executable	Password

Note 1: *: Don’t care

Note: Executable: A command can be executed.
Password: A password authentication is required to execute a command.
Not executable: A command cannot be executed. (After a command is echoed back, the TMP89FS60B/62B/63B stops communication.)

22.8.7. Flash memory security setting command (operation command: 0xFA)

Table 22-17 shows the Flash memory security setting command.

Table 22-17 Flash Memory Security Setting Command

	Transfer byte	Transfer data from the external controller to TMP89FS60B/62B/63B	Baud rate	Transfer data from TMP89FS60B/62B/63B to the external controller
BOOT ROM	1st byte 2nd byte	Matching data 1 "0x86" -	Automatic adjustment Baud rate after adjustment	- (Automatic baud rate adjustment) OK: Echo back data "0x86" Error: No data transmitted
	3rd byte 4th byte	Matching data 2 "0x79" -	Baud rate after adjustment Baud rate after adjustment	- OK: Echo back data "0x79" Error: No data transmitted
	5th byte 6th byte	Operation command data "0xFA" -	Baud rate after adjustment Baud rate after adjustment	- OK: Echo back data "0xFA" Error: "0xA1" × 3, "0xA3" × 3, "0x63" × 3 (Note 1)
	7th byte 8th byte	Password count storage address bits 23 to 16	Baud rate after adjustment Baud rate after adjustment	- OK: No data transmitted Error: No data transmitted
	9th byte 10th byte	Password count storage address bits 15 to 8	Baud rate after adjustment Baud rate after adjustment	- OK: No data transmitted Error: No data transmitted
	11th byte 12th byte	Password count storage address bits 7 to 0	Baud rate after adjustment Baud rate after adjustment	- OK: No data transmitted Error: No data transmitted
	13th byte 14th byte	Password comparison start address bits 23 to 16	Baud rate after adjustment Baud rate after adjustment	- OK: No data transmitted Error: No data transmitted
	15th byte 16th byte	Password comparison start address bits 15 to 8	Baud rate after adjustment Baud rate after adjustment	- OK: No data transmitted Error: No data transmitted
	17th byte 18th byte	Password comparison start address bits 7 to 0	Baud rate after adjustment Baud rate after adjustment	- OK: No data transmitted Error: No data transmitted
	19th byte : m-th byte	Password string (Note 3) -	Baud rate after adjustment Baud rate after adjustment	- OK: No data transmitted Error: No data transmitted
	n-th byte	-	Baud rate after adjustment	OK: 0xFB (Note 3) Error: No data transmitted
	n-th+1byte	(Wait for the next operation command data)	Baud rate after adjustment	-

Note 1: "0x**" × 3" means that the TMP89FS60B/62B/63B stops communication after transmitting 3 bytes of "0x**". For further information, refer to Table 22-18.

Note 2: For information on passwords, refer to "22.12.1. Passwords".

Note 3: When the Flash memory security setting command is executed for a blank product or when a password error occurs for a nonblank product by executing the Flash memory security command, the TMP89FS60B/62B/63B stops communication. Therefore, in these case, initialize the TMP89FS60B/62B/63B by using the $\overline{\text{RESET}}$ pin, and restart the Serial PROM mode.

Note 4: When a communication error occurs during the transfer of a password address or password string, the TMP89FS60B/62B/63B stops communication. Therefore, when a communication error occurs, initialize the TMP89FS60B/62B/63B by using the $\overline{\text{RESET}}$ pin, and restart the Serial PROM mode.

Note 5: When the Flash memory security is not enabled, it becomes possible to read the Flash memory data in the parallel PROM mode. the Security Program must be enabled in mass production.

22.9. Error Code

Table 22-18 shows the error codes that the TMP89FS60B/62B/63B transmits when it detects errors.

Table 22-18 Error Codes

Data transmitted	Meaning of error data
0x63, 0x63, 0x63	Operation command error
0xA1, 0xA1, 0xA1	Framing error in the received data
0xA3, 0xA3, 0xA3	Overrun error in the received data

Note: When a password error occurs, the TMP89FS60B/62B/63B does not transmit an error code.

22.10. Checksum (SUM)

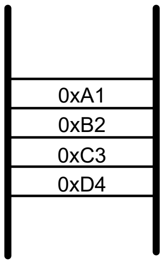
For the following operation commands, a checksum is returned to verify the appropriateness of the result of command execution:

- Flash memory erase command "0xF0"
- Flash memory write command "0x30"
- Flash memory read command "0x40"
- Flash memory SUM output command "0x90"
- Product ID code output command "0xC0"
- Flash memory status output command "0xC3"

22.10.1. Calculation method

The checksum (SUM) is calculated as a word with the sum of all data which is read as a byte in a calculated address area.

Example:



0xA1
0xB2
0xC3
0xD4

When the data to be calculated consists of four bytes as shown on the left, the checksum of them is as follows:

0xA1 + 0xB2 + 0xC3 + 0xD4 = 0x02EA
SUM (HIGH)= 0x02
SUM (LOW)= 0xEA

In the case of the product ID code output command and Flash memory status output command, however, a different calculation method is used. For more information, refer to Table 22-19.

22.10.2. Calculation data

Table 22-19 shows the data for which a checksum is calculated for each command.

Table 22-19 Data for which a Checksum Is Calculated

Operation command	Calculation data	Description
Flash memory erase command	All data in the erased area of Flash memory	In the case of the Chip Erase (entire Flash memory is erased), an entire area of the Flash memory is used. When the Sector Erase is executed, only the erased area is used to calculate the checksum.
Flash memory write command	Data in the entire area of Flash memory	When a part of the Flash memory is written, the checksum of the entire Flash memory area ("0x1000" to "0xFFFF") is calculated. The data length, address, record type and checksum in Intel Hex format are not included in the checksum.
Flash memory read command	Data in the read area of Flash memory	-
Flash memory SUM output command	Data in the entire area of Flash memory	The checksum of the entire Flash memory area ("0x1000" to "0xFFFF") is calculated. The data length, address, record type and checksum in Intel Hex format are not included in the checksum.
Product ID code output command	9th to 27th bytes of transferred data	For details, refer to "22.8.5. Product ID code output command (operation command: 0xC0)"
Flash memory status output command	9th to 12th bytes of transferred data	For details, refer to “22.8.6. Flash memory status output command (operation command: 0xC3)”

22.11. Intel Hex Format (Binary)

For the following two commands, the Intel Hex format is used in part of the transfer format:

- Flash memory write command "0x30"

For information on the definition of the Intel Hex format, refer to Table 22-20.
Data is in binary form. The start mark ":" must be transmitted as binary data of "0x3A".

(1) After receiving the checksum of each data record, the TMP89FS60B/62B/63B waits for the arrival of the start mark (0x3A ":") of the next data record. Although the external controller transmits data other than "0x3A" between records, the TMP89FS60B/62B/63B ignores such data.

(2) The external controller must be provisioned so that after it transmits the checksum of end record, it does not transmit any data until the arrival of 3-byte data (the data started calculating checksum, upper and lower bytes of the checksum). (3-byte data is used when the Flash memory write command is used.)

(3) When a receiving error or the Intel Hex format error occurs, the TMP89FS60B/62B/63B stops communication without returning an error code to the external controller. The Intel Hex format error occurs in the following cases:

- When the record type is other than "0x00", "0x01", or "0x02"
- When a checksum error of the Intel Hex format occurs
- When the data length of an extended record (record type = 0x02) is not "0x02"
- When the TMP89FS60B/62B/63B receives the data record after receiving an extended record (record type = 0x02) whose segment address is more than "0x2000"
- When the data length of the end record (record type = 0x01) is not "0x00"
- When the offset address of an extended record (record type = 0x02) is not "0x0000"

Table 22-20 Definition of the Intel Hex Format

	(1)	(2)	(3)	(4)	(5)	(6)
	Start mark	Data length (1 byte)	Offset address (2 bytes)	Record type (1 byte)	Data	Checksum (1 byte)
Data record (record type = 0x00)	0x3A	Number of data in a data field	Starting byte storage address Specified using big-endian	0x00	Data (1 to 255 bytes) (Note)	(2) Data length (3) Offset address (4) Record type (5) Complement of 2 of the sum total of data
End record (record type = 0x01)	0x3A	0x00	0x0000	0x01	None	(2) Data length (3) Offset address (4) Complement of 2 of the sum total of a record type
Extended record (record type = 0x02)	0x3A	0x02	0x0000	0x02	Segment address (2 bytes) Specified using big-endian	(2) Data length (3) Offset address (4) Record type (5) Complement of 2 of the sum total of a segment address

Note: The data length of a data record for the TMP89FS60B/62B/63B must be 64 bytes.

22.12. Security

In the Serial PROM mode, two security functions are provided to prohibit illegal memory access attempts by a third party: Password and Security Program functions.

22.12.1. Passwords

A password is one of the security functions, and can be used when the TMP89FS60B/62B/63B operates in the Serial PROM mode or when the on-chip debugging function (hereafter called OCD function) is used. Specifically, a password can be established by using data (part of user memory) in the Flash memory. When a password is established, a password authentication process must be performed to execute the Flash memory read command, Flash memory write command, and other operation commands. In the case of the OCD function, the password authentication process is required prior to the start of the system.

In parallel PROM mode, there are no access-related restrictions using a password. To establish the access-related restrictions that work in both serial and parallel PROM modes, the Security Program must be set to an appropriate setting.

22.12.1.1. How a password can be specified

With the TMP89FS60B/62B/63B, any piece of data in the Flash memory (8 or more consecutive bytes) can be specified as a password. A password thus specified is authenticated by comparing a password string transmitted by the external controller with the memory data string of MCU where the password is specified.

The area where a password can be specified is "0x1000" to "0xFEFF" in the Flash memory.

22.12.1.2. Password configuration

A password consists of three components: PNSA, PCSA, and a password string. Figure 22-4 shows the password structure (example of a transmitted password).

- PNSA (password count storage address)

  A 3-byte address is specified in the area "0x1000" to "0xFEFF". The memory data of a specified address is the number of bytes of a password string. When the memory data is less than "0x07" or when an address is outside the address range, a password error occurs.

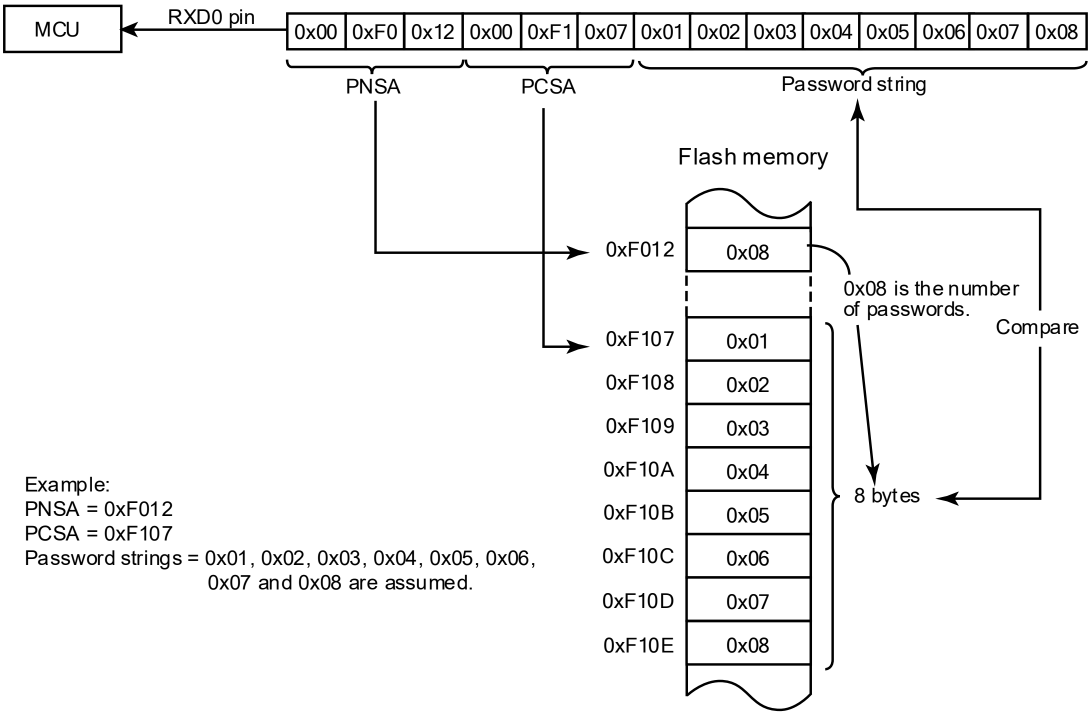
  The memory data specified here is defined as N.

- PCSA (password comparison start address)

  A 3-byte address is specified in the area ("0x1000" to "0xFF00 - N"). An address thus specified is the starting address to be used to compare with a password string. When an address is outside the address range, a password error occurs.

- Password string

  Data of 8 bytes to 255 bytes (= N) must be specified as a password string. Memory data and a password string are compared by a specified number "N" of bytes; a comparison starts at an address specified by PCSA. When there is a mismatch as a result of this comparison or when same data of 3 or more consecutive bytes is specified, a password error occurs, and the TMP89FS60B/62B/63B stops communication. After this, external devices cannot communicate with the TMP89FS60B/62B/63B. To resume communication, the TMP89FS60B/62B/63B must be restarted in the Serial PROM mode by using the $\overline{\text{RESET}}$ pin.



Example:
PNSA = 0xF012
PCSA = 0xF107
Password strings = 0x01, 0x02, 0x03, 0x04, 0x05, 0x06,
0x07 and 0x08 are assumed.

Figure 22-4 Password Structure (Example of a Password Transmitted)

22.12.1.3. Password setting, cancellation and authentication

- Password setting

  Because a password is created by using part of a user program, a special password setting process is unnecessary. A password can be set by simply writing a program to Flash memory.

- Password cancellation

  To cancel a password, Chip Erase (erase the entire area of Flash memory) must be performed on Flash memory. A password is canceled when Flash memory is all initialized to "0xFF".

- Password authentication

  When there is data other than "0xFF" in any one byte of data written to the address "0xFFE0" to "0xFFFF" of the TMP89FS60B/62B/63B, a product is considered a non-blank product, and password authentication is required to execute an operation command. In this password authentication process, PNSA, PCSA and a password string are used. An operation command is executed only when a password has been successfully authenticated. When a password is unsuccessfully authenticated, the TMP89FS60B/62B/63B stops communication.

  When all data written to the address "0xFFE0" to "0xFFFF" are "0xFF", a product is considered blank, and no password authentication is performed. To execute some special operation commands, however, PNSA and PCSA are still required (a password string is not required) when a product is blank. In this case, the addresses defined in Table 22-21 must be selected as PNSA and PCSA.

  Whether a product is blank or non-blank can be confirmed by executing the status output command.

  The operation commands that require PNSA and PCSA (password string) for them to be executed are as follows:

- Flash memory erase command "0xF0"
- Flash memory write command "0x30"
- Flash memory read command "0x40"
- Flash memory security setting command "0xFA"

22.12.1.4. Password values and setting range

A password must be set in accordance with the conditions shown in Table 22-21. When a password created without meeting these conditions is used, a password error occurs. In this case, the TMP89FS60B/62B/63B stops communication.

Table 22-21 Password Values and Setting Range

Password	Blank product (Note 2)	Non-blank product
PNSA (password count storage address)	0x1000 ≤ PNSA ≤ 0xFEFF	0x1000 ≤ PNSA ≤ 0xFEFF
PCSA (password comparison start address)	0x1000 ≤ PCSA ≤ 0xFEFF	0x1000 ≤ PCSA ≤ 0xFF00 - N
N (password count)	*	8 ≤ N
Password string	Not required (Notes 4 and 5)	Required (Note 3)

Note 1: *: Don’t care.

Note 2: When addresses from "0xFFE0" to "0xFFFF" are filled with "0xFF", the product is recognized as a blank product.

Note 3: The data including the same consecutive data (three or more bytes) cannot be used as a password. (A password error occurs during password authentication. The TMP89FS60B/62B/63B does not transmit any data and goes into an idle state.)

Note 4: When Flash memory writing command is executed, the blank product receives the Intel Hex format data immediately after receiving PCSA; it does not receive password strings. In this case, the subsequent processing is performed correctly because the TMP89FS60B/62B/63B keeps ignoring incoming data until the start mark (0x3A ":") in the Intel Hex format is detected, when the external controller transmits the dummy password string. However, when the dummy password string contains "0x3A", it is detected as the start mark erroneously, and the TMP89FS60B/62B/63B stops communication. When this causes a problem, do not transmit the dummy password strings.

Note 5: In executing the Flash memory erase command, do not transmit a password string to a blank product.

22.12.2. Security Program

The Security Program can be used in the parallel and the Serial PROM modes and for OCD function. TMP89FS60B/62B/63B has a special memory for protection, and a special command is required to make this protection setting.

When the Security Program is enabled, the reading or writing of the Flash memory in parallel PROM mode is prohibited.

In the Serial PROM mode, the read and write of Flash memory and other operation commands cannot be used.

In using OCD function, two options about system startup are provided: prohibiting the system startup by using an option code and starting the system by password authentication.

Because protection-related information is written to this specially-designed memory, no user memory resource are required.

22.12.2.1. Enabling or disabling the Security Program

- Enabling the Security Program

  To enable the Security Program, execute "the Flash memory security setting command".

- Disabling the Security Program

  To disable the Security Program, execute Chip Erase of "the Flash memory erase command".

22.12.3. Option codes

When a specified option code is placed at a specified address inside the interrupt vector area, whether password string authentication is performed or not when executing the Flash memory erase command and whether the Security Program is checked or not when starting OCD function can be designated.

- Erase password free code EPFC_OP: address "0xFFFA"

  When changes are frequently made to a program during software development, there are cases in which a password may get lost. In this case, you can cancel the password string authentication of the Flash memory erase command "0xF0" by setting the erase password free code (EPFC_OP).

  EPFC_OP is assigned to "0xFFFA" in the vector area. Allocate "0xFF" to this EPFC_OP to cancel the password string of the Flash memory erase command "0xF0".

  It is recommended that the password string authentication of the Flash memory erase command "0xF0" be enabled during mass production by allocating data other than "0xFF" to EPFC_OP.

  Only Chip Erase can cancel the password string authentication by using the Flash memory erase command. When Sector Erase is executed with EPFC_OP set to "0xFF", the TMP89FS60B/62B/63B stops communication. Commands other than the Flash memory erase command cannot cancel the password string authentication.

- OCD Security Program free code DAFC_OP: address "0xFFFB"

  With the TMP89FS60B/62B/63B, you can enable the Security Program to prevent illegal access attempts by a third party. When the Security Program is enabled, restrictions are imposed on operation commands related to memory access, and the startup of OCD function.

  The Security Program should be usually enabled at the time of shipment. When there is the possibility that the OCD function may be used by keeping the contents of memory intact, it is possible to directly start the OCD function by setting DAFC_OP and thereby skipping the Security Program check (the password authentication, however, is still required).

  DAFC_OP is assigned to "0xFFFB" in the vector area. To skip the Security Program check at the startup of the OCD function, assign "0xFF" to DAFC_OP. In this case, the Security Program check is not performed, and the OCD function can be started by performing only the password authentication.

  When DAFC_OP is not "0xFF", whether the OCD function can be used or not is determined by the status of the Security Program. When the OCD function is started with the Security Program enabled, the TMP89FS60B/62B/63B stops communication. To use the OCD function when the TMP89FS60B/62B/63B is in this idle state, Chip Erase must be executed for the Flash memory by using the Flash memory erase command "0xF0". When the Security Program is disabled, the OCD function can be started by performing only the password string authentication.

Table 22-22 Option Codes

Symbol	Function	Address	Set value
EPFC_OP	Password string authentication when the Flash memory erase command is executed	0xFFFA	0xFF: The password string authentication is skipped (only PNSA and PCSA are authenticated). Other than 0xFF: The password string, PNSA, and PCSA are authenticated.
DAFC_OP	Security Program check when the OCD is started	0xFFFB	0xFF: The Security Program check is skipped. Other than 0xFF: The Security Program check is performed.

Example: Case in which the password authentication and OCD Security Program authentication are disabled

```
vector section romdata abs = 0xFFFA
        DB      0xFF        ; Disable the password string authentication at the Flash memory command (EPFC_OP)
        DB      0xFF        ;  Skip the Security Program when the OCD function is started (DAFC_OP)
```

22.12.4. Recommended settings

Table 22-23 shows the option codes and recommended Security Program settings.

Table 22-23 Option Codes and Recommended Security Program Settings

	Device status			Serial PROM mode		Parallel PROM mode		OCD function
	EPFC_OP (0xFFFA)	**DAFC_OP (0xFFFB)**	**Security Program**	**Flash memory read**	**Flash Memory erase**	**Flash memory read**	**Flash Memory erase**	
At the time of debugging during software development	0xFF	0xFF	Disable	Password string required	Possible	Possible	Possible (note 1)	Can be used
During mass production	0xFF	0xFF	Enable	Impossible	Possible	Impossible	Possible (Note 1)	Can be used
		Other than 0xFF						Cannot be used
	Other than 0xFF	0xFF			Password string required			Can be used
		Other than 0xFF						Cannot be used

Note 1: In parallel PROM mode, Chip Erase can be performed irrespective of the option code setting.

Note 2: When the Security Program is not enabled in parallel PROM mode, the Flash memory data can be read with no restrictions. Make sure that in parallel PROM mode, always enable the Security Program to protect the Flash memory data in mass production.

22.13. Flowchart

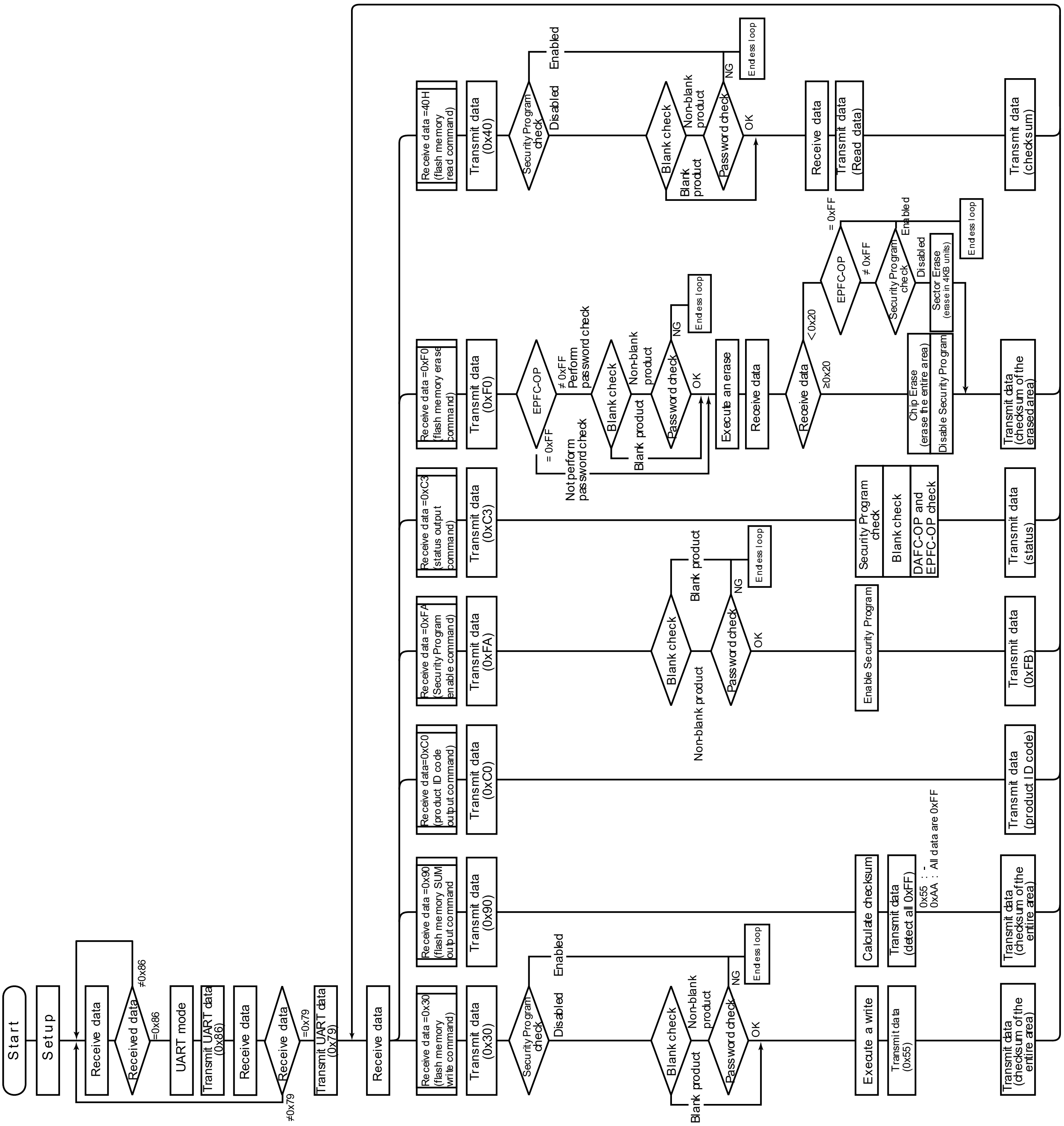


Figure 22-5 Flowchart

22.14. AC Characteristics (UART)

Table 22-24 UART Timing-1

Parameter	Symbol	Number of clock (fc)	Minimum required time	
			At fc = 4 MHz	At fc = 10 MHz
Time from when MCU receives "0x86" to when it echoes back	CMeb1	Approx. 660	165 [μs]	66 [μs]
Time from when MCU receives "0x79" to when it echoes back	CMeb2	Approx. 540	135 [μs]	54 [μs]
Time from when MCU receives an operation command to when it echoes back	CMeb3	Approx. 300	75 [μs]	30 [μs]
Time required to calculate the checksum (except the time of writing to the Flash memory)	CMfsm	Approx. 2800000 (60KB)	0.7 [s]	280 [ms]
Time when MCU receives Intel Hex data to when it transmits the started calculating checksum.	CMwr	Approx. 200	50 [μs]	20 [μs]
Time from when MCU receives data (number of read bytes) to when it transmits memory data	CMrd	Approx. 430	107.5 [μs]	43 [μs]
Time required to enable the Security Program	CMrp	Approx. 1080	270 [μs]	108 [μs]

Table 22-25 UART Timing-2

Parameter	Symbol	Number of clock (fc)	Minimum required time	
			At fc = 4 MHz	At fc = 10 MHz
Time required to keep MODE and $\overline{\text{RESET}}$ pins at “Low” level after power-on	RSsup	-	10 [ms]	
Time from when MODE and $\overline{\text{RESET}}$ pins are set to “High” level to the acceptance of RXD0 pin	RXsup	-	20 [ms]	
Time from when MCU echoes back "0x86" to the acceptance of RXD0 pin	CMtr1	Approx. 140	35 [μs]	14 [μs]
Time from when MCU echoes back "0x79" to the acceptance of RXD0 pin	CMtr2	Approx. 90	22.5 [μs]	9 [μs]
Time from when MCU echoes back an operation command to the acceptance of RXD0 pin	CMtr3	Approx. 270	67.5 [μs]	27 [μs]
Time from when the execution of a current command is completed to the acceptance of the next operation command from RXD0 pin	CMnx	Approx. 1100	275 [μs]	110 [μs]
Wait time after the completion of the receiving data record	CMer		3 [ms]	

22.14.1. Reset timing

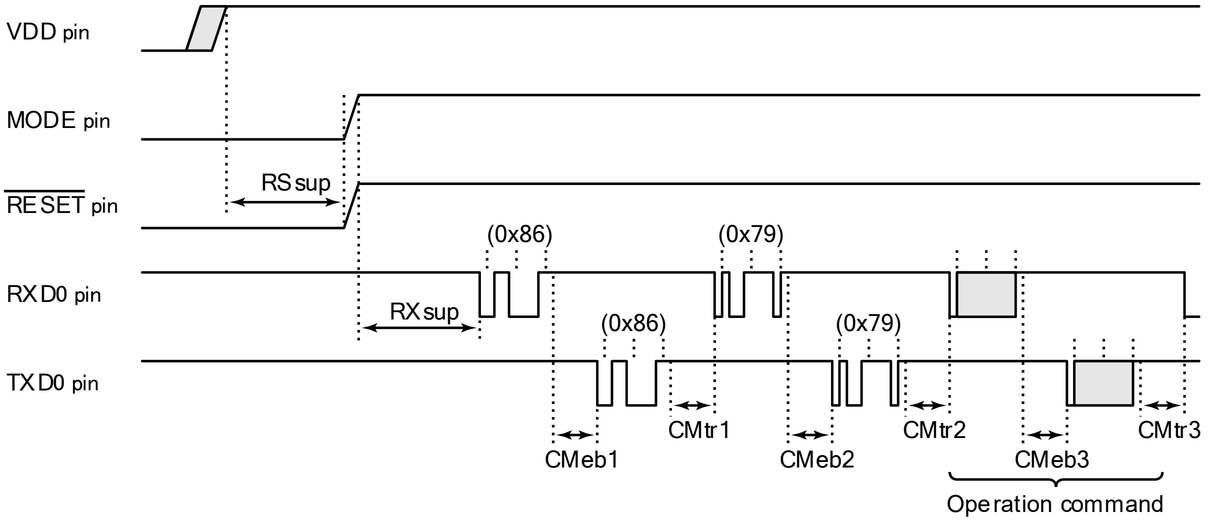


Figure 22-6 Reset Timing

22.14.2. Flash memory erase command (0xF0)

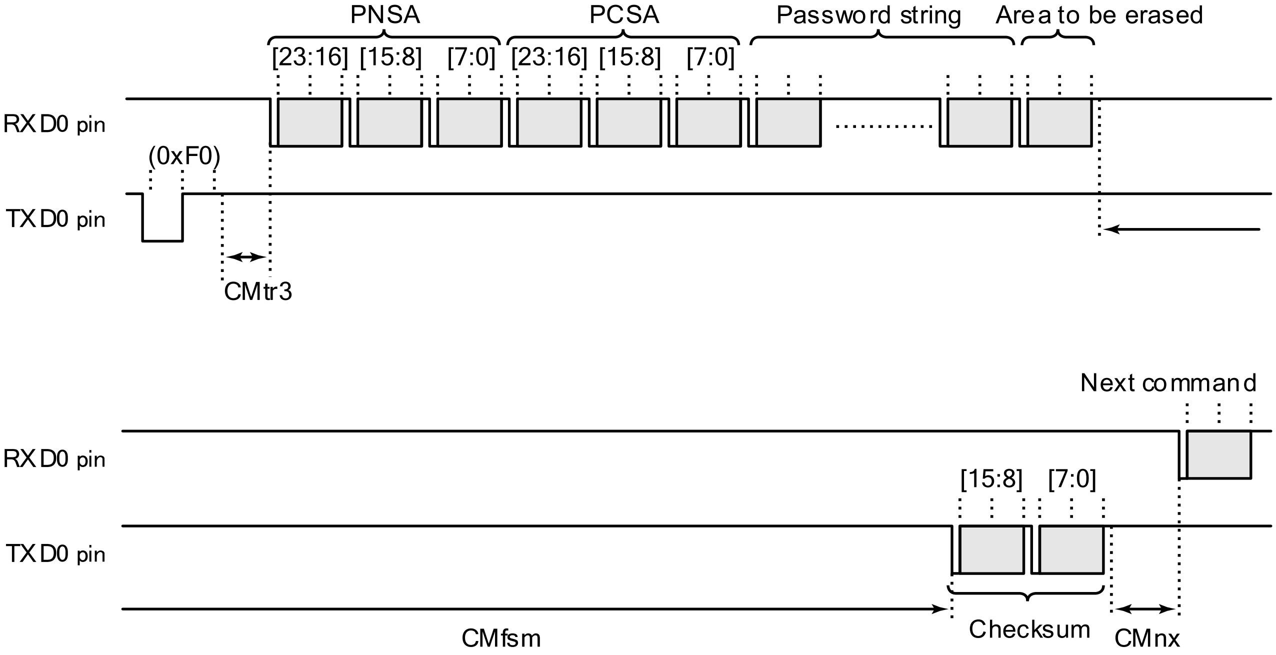


Figure 22-7 Flash Memory Erase Command

22.14.3. Flash memory write command (0x30)

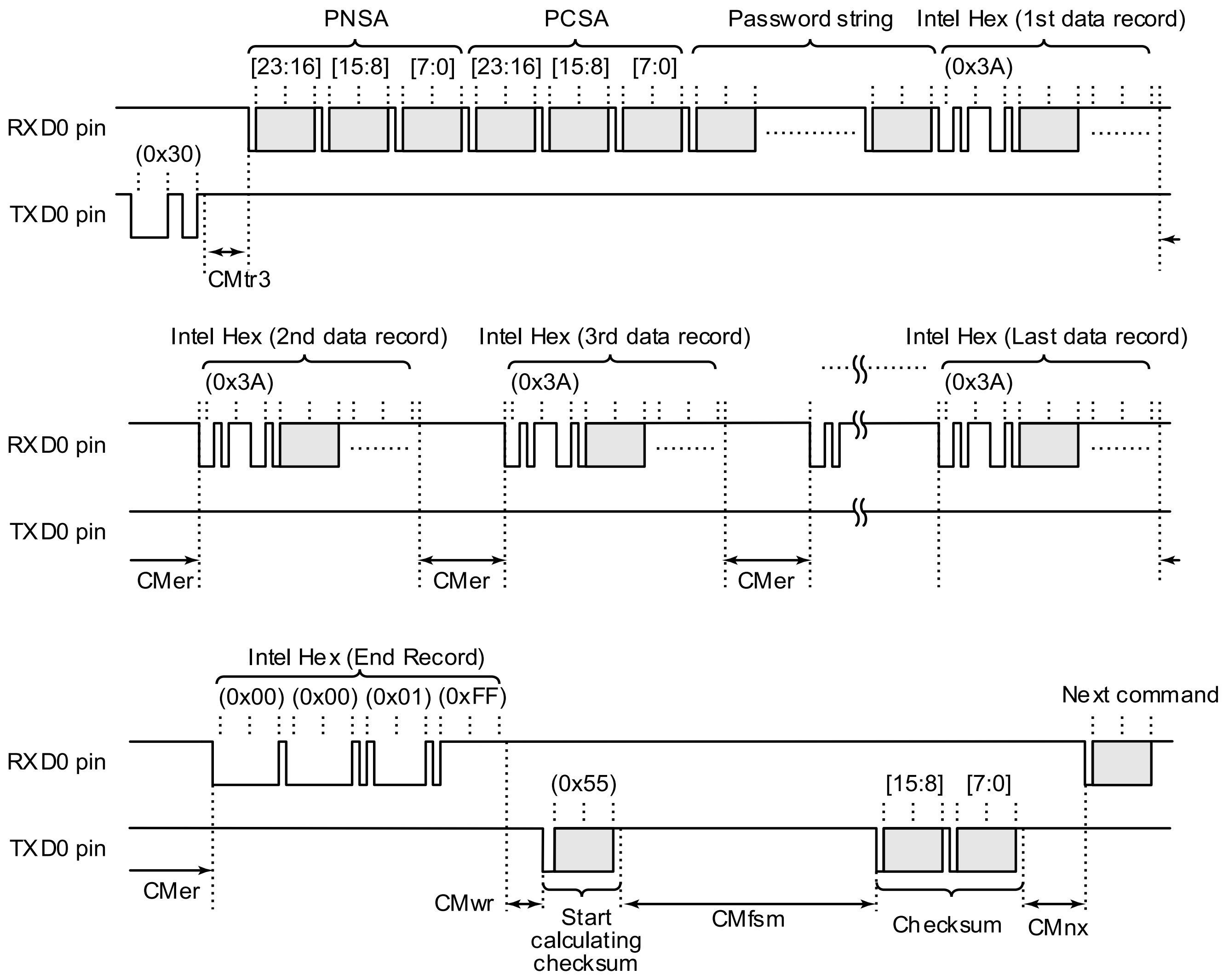


Figure 22-8 Flash Memory Write Command

22.14.4. Flash memory read command (0x40)

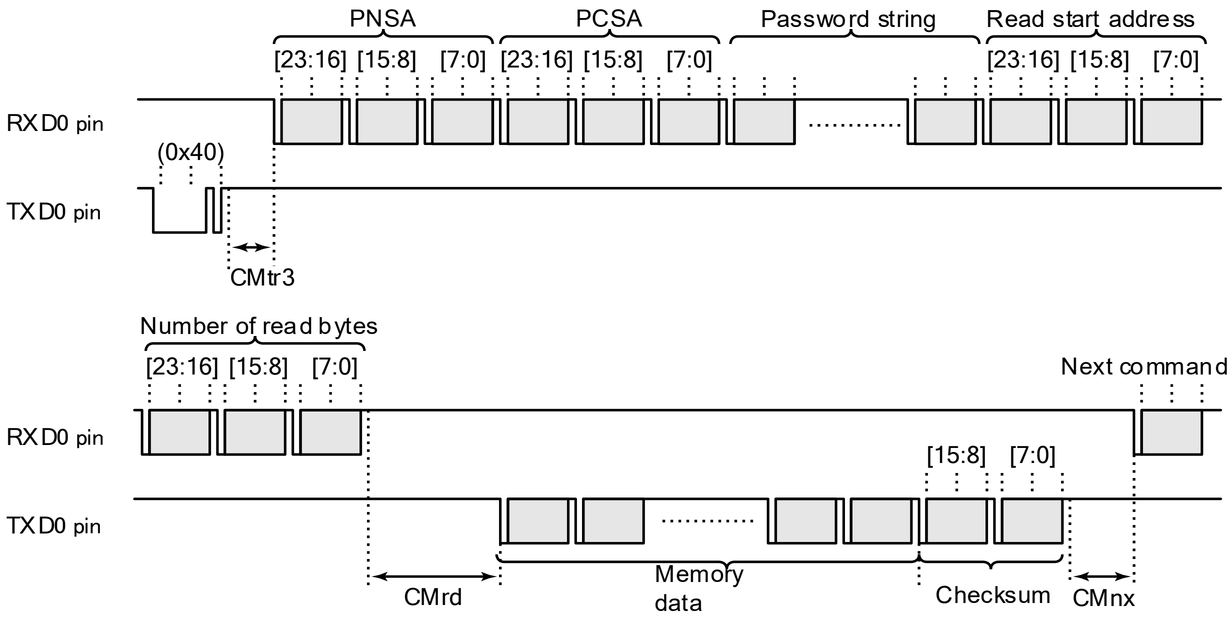


Figure 22-9 Flash Memory Read Command

22.14.5. Flash memory SUM output command (0x90)

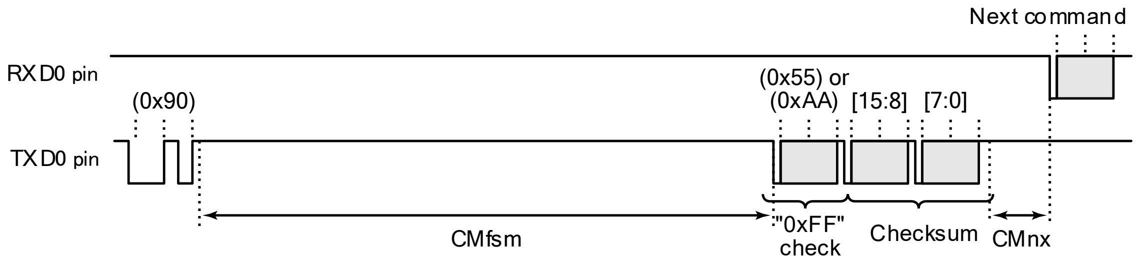


Figure 22-10 Flash Memory SUM Output Command

22.14.6. Product ID code output command (0xC0)

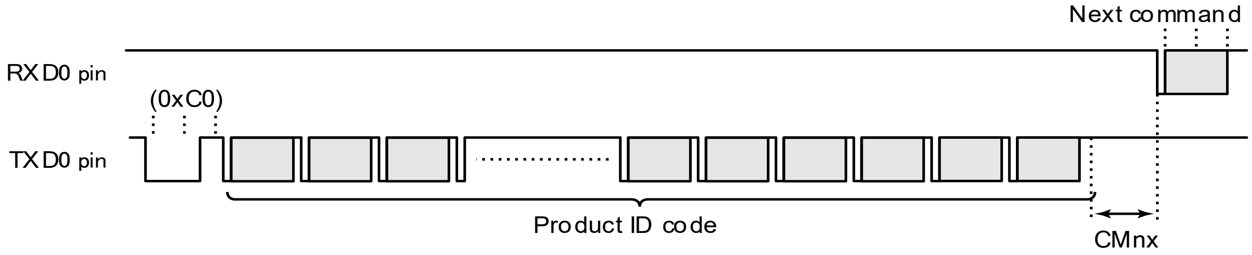


Figure 22-11 Product ID Code Output Command

22.14.7. Flash memory status output command (0xC3)

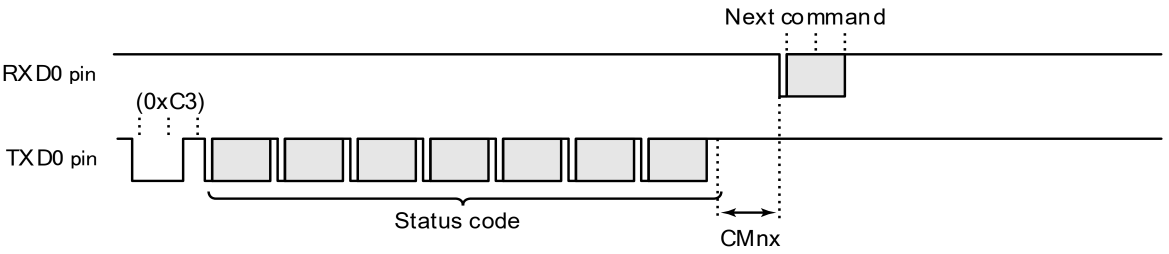


Figure 22-12 Flash Memory Status Output Command

22.14.8. Flash memory security setting command (0xFA)

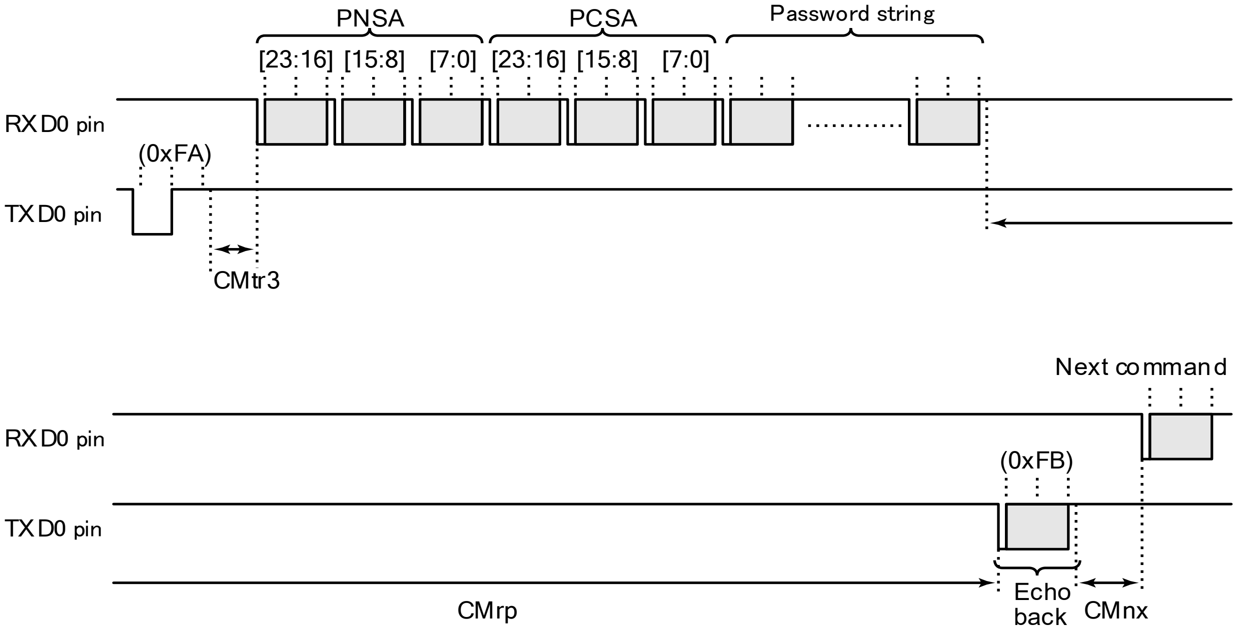


Figure 22-13 Flash Memory Security Setting Command

23. On-chip Debug Function (OCD)

The TMP89FS60B/62B/63B has an on-chip debug function. Using a combination of this function and an our on-chip debug emulator RTE870/C1, the user is able to perform software debugging in the on-board environment.

This emulator can be operated from a debugger installed on a PC so that the emulation and debugging functions of an application program can be used to modify a program or for other purposes.

This chapter describes the control pins needed to use the on-chip debug function and how a target system is connected to the on-chip debug function. For more detailed information on how to use the on-chip debug emulator RTE870/C1, refer to the emulator operating manual.

23.1. Features

The on-chip debug function of the TMP89FS60B/62B/63B has the following features:

- Debugging can be performed on the PCB which is mounted MCU.
- The debugging function can be realized using two communication control pins.
- Useful on-chip debug functions include the following:
  - 8 breaks function are provided (one of which can also be used as an event function).
  - A trace function that allows the newest two branch instructions to be stored in real time is provided.
  - Functions to display active memory and to overwrite active memory are provided.
- Built-in Flash memory can be erased and written.

23.2. Control Pins

The on-chip debug function uses two pins for communication and four pins for power supply, reset and mode control. The pins used for the on-chip debug function are shown in Table 23-1.

Ports P20 and P21 are used as communication control pins of the on-chip debug function. When the RTE870/C1 On-Chip Debug Emulator is used, therefore, Ports P20 and P21 cannot be debugged as port pins or UART0 and SIO0 pins. However, because the UART0 and SIO0 functions can be assigned to other ports by using SERSEL<SRSEL2>, these communication functions can also be used during on-chip debug operation. For details, refer to "8. I/O Ports".

Table 23-1 Pins Used for the On-chip Debug Function

Pin name (during on-chip debugging)	Input/output	Function	Pin name (in MCU mode)
OCDCK	Input	Communication control pin (clock control)	P20/TXD0/SO0
OCDIO	I/O	Communication control pin (data control)	P21/RXD0/SI0
$\overline{\text{RESET}}$	Input	Reset control pin	$\overline{\text{RESET}}$
MODE	Input	Mode control pin	MODE
VDD	Power supply	4.5 to 5.5 [V] (Note)	
VSS	Power supply	0 [V]	
Input and output ports other than P20 and P21	I/O	Can be used for an application in a target system	
XIN	Input	To be connected to a resonator to put these pins in a state of self-oscillation	
XOUT	Output		

Note: To use all on-chip debug functions, the power supply voltage must be within the range 4.5 to 5.5 V. When it is within the range 4.2 to less than 4.5 [V], functional limitations occur with some of the debug functions. For more detailed information, refer to the emulator operating manual.

23.3. How to Connect the On-chip Debug Emulator to a Target System

To use the on-chip debug function, the specific pins on a target system must be connected to an external debugging system.

The on-chip debug emulator RTE870/C1 can be connected to a target system via an interface control cable.

Our company provides a connector for this interface control cable as an accessory tool. Mounting this connector on a target system will make it easier to use the on-chip debug function.

The connection between the on-chip debug emulator RTE870/C1 and a target system is shown in Figure 23-1.

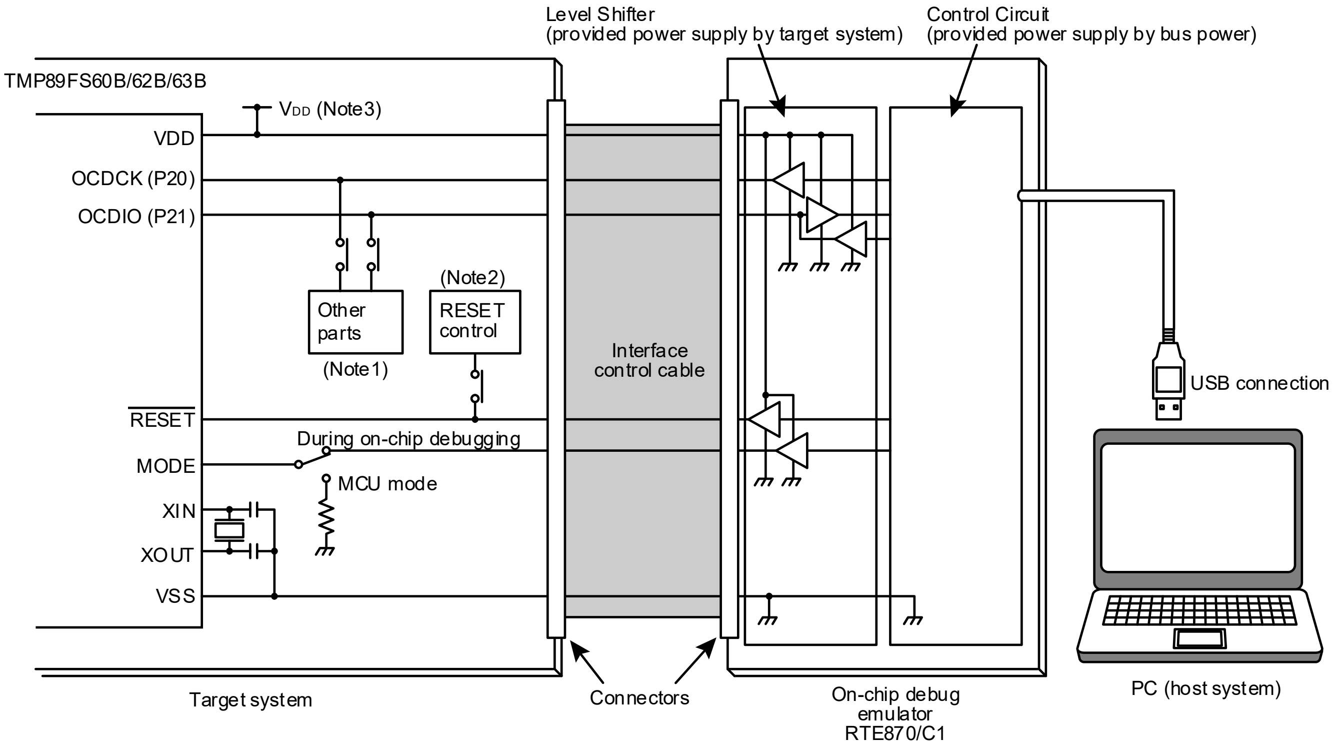


Figure 23-1 The Connection Between the On-chip Debug Emulator RTE870/C1 and a Target System

Note 1: Ports P20 and P21 are used as communication control pins of the on-chip debug function. When the on-chip debug emulator RTE870/C1 is used, therefore, the port functions and the functions of UART0 and SIO0, which are also used as ports, cannot be debugged. When the emulator is disconnected to be used as a single MCU, the functions of ports P20 and P21 can be used. To use the on-chip debug function, however, P20 and P21 should be disconnected using a jumper, switch, etc. when there is the possibility of other parts affecting the communication control.

Note 2: When the reset control circuit on an application board affects the control of the on-chip debug function, it must be disconnected using a jumper, switch, etc.

Note 3: The power supply voltage V_{DD} must be provided by a target system. The VDD pin is connected to the emulator so that the level of voltage appropriate for driving communication pins can be obtained by using the power supply of a target system. The connection of the VDD pin is for receiving the power supply voltage, not for supplying it from the emulator side to a target system.

23.4. Security

The TMP89FS60B/62B/63B provides two security functions to prevent the on-chip debug function from being used through illegal memory access attempted by a third person: a “Password” function and a “Security Program” function. When a “Password” is set for the TMP89FS60B/62B/63B, it is necessary to authenticate the “Password” for using the on-chip debug function. By setting both a “Password” and the “Security Program” for the TMP89FS60B/62B/63B, it is possible to prohibit the use of all on-chip debug functions. Furthermore, by using the “Option Code”, the on-chip debug function only can be used when the “Security Program” is enabled. However, to use the on-chip debug function in this setting, a “Password” authentication process is required.

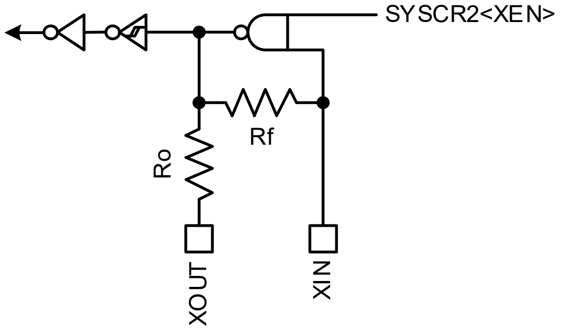
For information on how to set a "Password" and to enable the “Security Program” and "Option Code", refer to "22. Serial PROM Mode".

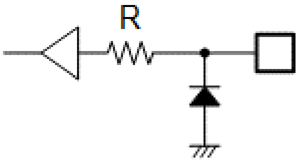
24. Input/output Circuit

24.1. Control Pins

The input/output circuitries of the TMP89FS60B/62B/63B control pins are shown below.

Control pin	I/O	Circuitry	Remarks

XIN
XOUT
	Input
	Output
			Rf = 1.2 [MΩ] (typ.)
			Ro = 0.5 [kΩ] (typ.)

XTIN XTOUT	Input Output	Refer to the P0 ports in “8. I/O Ports”.	
$\overline{\text{RESET}}$	Input	Refer to the P1 ports in “8. I/O Ports”.	
MODE	Input		R = 100 [Ω] (typ.)

25. Electrical Characteristics

25.1. Absolute Maximum Ratings

The absolute maximum ratings are rated values which must not be exceeded during operation, even for an instant. Any one of the ratings must not be exceeded. If any absolute maximum rating is exceeded, a device may break down or its performance may be degraded, causing it to catch fire or explode resulting in injury to the user. Thus, when designing products which include this device, ensure that no absolute maximum rating value will ever be exceeded.

(V_{SS} = 0 [V])

Parameter	Symbol	Pins	Ratings	Unit
Supply voltage	V_{DD}		-0.3 to 6.0	V
Input voltage	V_{IN1}	P0, P1, P2, P4, P5, P7, P8, P9, PB	-0.3 to V_{DD} + 0.3	V
	V_{IN2}	AIN15 to AIN0 (analog input voltage)	-0.3 to AV_{DD} + 0.3	
Output voltage	V_{OUT1}	All ports	-0.3 to V_{DD} + 0.3	V
Output current (per pin)	I_{OUT1}	P0, P1, P2 (except P23 and P24), P4, P5, P7, P8, P9, PB (tri-state port)	-1.8	mA
	I_{OUT2}	P0, P1, P2 (except P23 and P24), P4, P9 (pull-up resistor)	-0.4	
	I_{OUT3}	P0, P1, P2, P4, P5, P7, P8, P9	3.2	
	I_{OUT4}	PB (large current port)	30	
Output current (total)	$\sum I_{OUT1}$	P0, P1, P2 (except P23 and P24), P4, P5, P7, P8, P9, PB (tri-state port)	-30	
	$\sum I_{OUT2}$	P0, P1, P2 (except P23 and P24), P4, P9 (pull-up resistor)	-4	
	$\sum I_{OUT3}$	P0, P1, P2, P4, P5, P7, P8, P9	60	
	$\sum I_{OUT}4$	PB (large current port)	120	
Power dissipation (Topr = 85 [°C])	P_D		250	mW
Soldering temperature (time)	Tsld		260 (10s)	°C
Storage temperature	Tstg		-55 to 125	
Operating temperature	Topr		-40 to 85	

Note: The number of port pins is different depend on the products. For the pins of each port, Refer to “8. I/O Ports”.

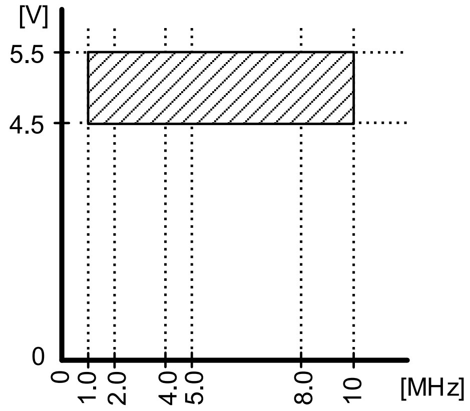
25.2. Operating Conditions

The operating conditions for a device are operating conditions under which it can be guaranteed that the device will operate as specified. When the device is used under operating conditions other than the operating conditions (supply voltage, operating temperature range, specified AC/DC values etc.), malfunction may occur. Thus, when designing products which include this device, ensure that the operating conditions for the device are always adhered to.

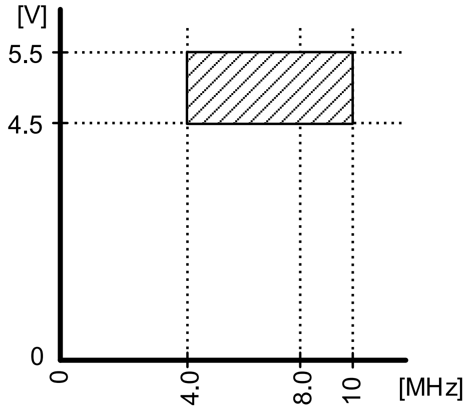
25.2.1. MCU mode (Flash Memory Programming or Erasing)

(V_{SS} = 0 [V], Topr = -40 to 85 [°C])

Parameter	Symbol	Pins	Condition		Min	Max	Unit
Supply voltage	V_{DD}		fc = 4.0 to 10 [MHz]	NORMAL1/2 modes	4.5	5.5	V
			fcgck = 1.0 to 10 [MHz]				
Clock frequency	fc	XIN, XOUT	V_{DD} = 4.5 to 5.5 [V]		4.0	10	MHz
	fcgck				1.0	10	



Gear clock (fcgck) frequency range



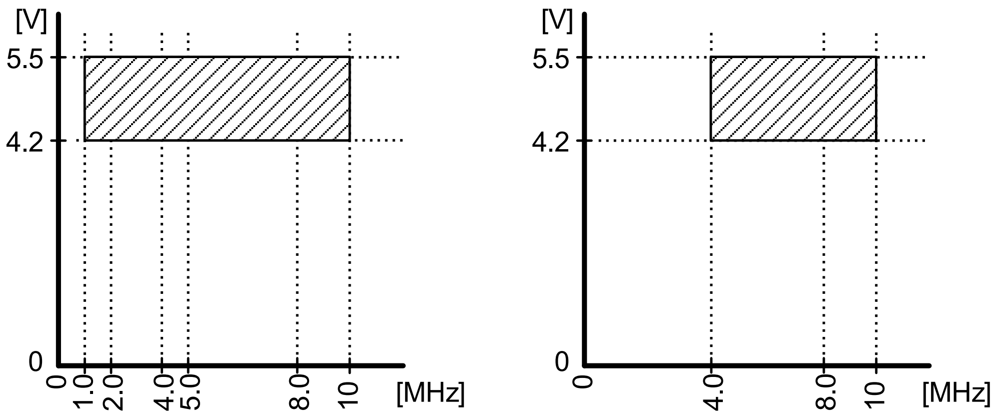
High-frequency clock (fc) frequency range

Figure 25-1 Clock gear (fcgck) and High-frequency clock (fc)

25.2.2. MCU mode (Except Flash Memory Programming or Erasing)

($V_{SS} = 0$ [V], Topr = -40 to 85 [°C])

Parameter	Symbol	Pins	Condition		Min	Max	Unit
Supply voltage	V_{DD}		fc = 4.0 to 10 [MHz]	NORMAL1 mode IDLE0/1 modes	4.2	5.5	V
			fcgck = 1.0 to 10 [MHz]				
			fc = 4.0 to 10 [MHz]	NORMAL2 mode IDLE2 mode SLOW2 mode			
			fcgck = 1.0 to 10 [MHz]				
			fs = 32.768 [kHz]				
			fs = 32.768 [kHz]	SLOW1 mode SLEEP0/1 modes			
			STOP Mode				
Clock frequency	fc	XIN, XOUT	V_{DD} = 4.2 to 5.5 [V]		4.0	10	MHz
	fcgck				1.0	10	
	fs	XTIN, XTOUT			30	34	kHz



Gear clock (fcgck) frequency range

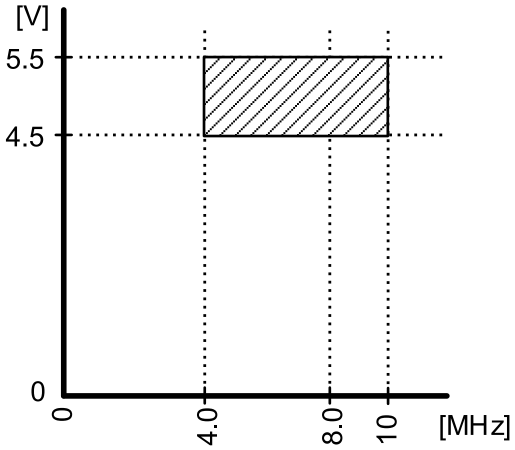
High-frequency clock (fc) frequency range

Figure 25-2 Clock gear (fcgck) and High-frequency clock (fc)

25.2.3. Serial PROM mode

(V_{SS} = 0 [V], Topr = -40 to 85 [°C])

Parameter	Symbol	Pins	Condition		Min	Max	Unit
Supply voltage	V_{DD}		fc = 4.0 to 10 [MHz]	NORMAL1 mode	4.5	5.5	V
Clock frequency	fc	XIN, XOUT	V_{DD} = 4.5 to 5.5 [V]		4.0	10	MHz



High-frequency clock (fc) frequency range

Figure 25-3 High-frequency clock (fc)

25.3. DC Characteristics

(V_{DD} = 4.5 to 5.5 [V], V_{SS} = 0 [V], Topr = -40 to 85 [°C])

Parameter	Symbol	Pins	Condition	Min	Typ.	Max	Unit
Hysteresis voltage	V_{HS}	Hysteresis input	V_{DD} = 5.5V V_{IN} = V_{MODE} = 5.5V/0V	-	0.9	-	V
Input current	I_{IN1}	MODE		-	-	±2	μA
	I_{IN2}	P0, P1, P2, P4, P5 (Note 4), P7, P8, P9, PB					
	I_{IN3}	$\overline{\text{RESET}}$, $\overline{\text{STOP}}$					
Input resistance	R_{IN2}	$\overline{\text{RESET}}$ pull-up	V_{DD} = 5.5V, V_{IN} = V_{MODE} = 0V	100	220	500	kΩ
	R_{IN3}	P0, P1, P2 (except P23 and P24), P4, P9 pull-up		30	50	100	
Output leakage current	I_{LO1}	P0, P1, P2, P4, P5, P7, P8, P9, PB	V_{DD} = 5.5V, V_{OUT} = 5.5V/0V	-	-	±2	μA
Input high level	V_{IH1}	MODE pin	V_{DD} ≥ 4.5 [V]	V_{DD} × 0.70	-	V_{DD}	V
	V_{HI2}	Hysteresis input		V_{DD} × 0.75	-		
Input low level	V_{IL1}	MODE pin	V_{DD} ≥ 4.5 [V]	0	-	V_{DD} × 0.30	
	V_{IL2}	Hysteresis input			-	V_{DD} × 0.25	
Output high level voltage	V_{OH}	Except P23 and P24, XOUT, XTOUT	V_{DD} = 4.5V, I_{OH} = -0.7mA	4.1	-	-	V
Output low level voltage	V_{OL}	Except XOUT, XTOUT	V_{DD} = 4.5V, I_{OL} = 1.6mA	-	-	0.4	
Output low level current	I_{OL}	PB (large current port)	V_{DD} = 4.5V, V_{OL} = 1.0V	-	20	-	mA

Note 1: Typical values show those at Topr = 25°C and V_{DD} = 5.0 V.

Note 2: Input current I_{IN3} : The current through pull-up resistor is not included.

Note 3: V_{IN} : The input voltage on the pin except MODE pin, V_{MODE} : The input voltage on the MODE pin

Note 4: The number of port pins is different depend on the products. For the pins of each port, Refer to “8. I/O Ports.

(V_{SS} = 0 [V], Topr = -40 to 85 [°C])

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Supply current in NORMAL1/2 modes	I_{DD}	V_{DD} = 5.5 [V] V_{IN} = 5.3 [V]/0.2 [V] V_{MODE} = 5.3 [V]/0.1 [V] fcgck = 10.0 [MHz] fs = 32.768 [kHz]	-	3.7	7.0	mA
Supply current in IDLE0/1/2 modes			-	3.3	6.0	
Supply current in SLOW1 mode			-	170	320	μA
Supply current in SLEEP1 mode			-	170	320	
Supply current in SLEEP0 mode			-	170	320	
Supply current in STOP mode			-	20	170	
Current for writing to Flash memory, erasing and Security Program	I_{DDEW}	V_{DD} = 5.5 [V] V_{IN} = 5.3 [V]/0.2 [V] V_{MODE} = 5.3 [V]/0.1 [V]	-	3.7	-	mA

Note 1: Typical values shown are Topr = 25 [°C] and V_{DD} = 5.0 [V], unless otherwise specified.

Note 2: I_{DD} does not include I_{REF}. It is the electrical current in the state in which the peripheral circuitry has been operated.

Note 3: V_{IN}: The input voltage on the pin except MODE pin, V_{MODE}: The input voltage on the MODE pin

Note 4: Each supply current in SLOW2 mode is equivalent to that in IDLE0, IDLE1 and IDLE2 modes.

25.4. AD Conversion Characteristics

25.4.1. AD Conversion Characteristics of TMP89FS60B

(V_{DD} = 4.5 to 5.5 [V], V_{SS} = AV_{SS} = 0.0 [V], Topr = -40 to 85 [°C])

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Analog reference voltage	V_{AREF}		4.5	-	A_{VDD}	V
Power supply voltage of analog control circuit	A_{VDD}		V_{DD}			
Analog reference voltage range (Note 3)	ΔV_{AREF}		4.5	-	5.5	
Analog input voltage range	V_{AIN}		A_{VSS}	-	V_{AREF}	
Power supply current of analog reference voltage	I_{REF}	V_{DD} = A_{VDD} = V_{AREF} = 5.5 [V]	-	0.4	1.0	mA
		V_{DD} = A_{VDD} = V_{AREF} = 4.5 [V]	-	0.3	0.8	
Non-linearity error		V_{DD} = A_{VDD} = 4.5 to 5.5 [V] V_{AREF} = 4.5 to 5.5 [V]	-	-	±2	LSB
Zero point error			-	-	±2	
Full scale error			-	-	±2	
Total error			-	-	±2	

Note 1: The total error includes all errors except a quantization error, and is defined as the maximum deviation from the ideal conversion line.

Note 2: The voltage to be input to the AIN input pin must be within the range V_{AREF} to A_{VSS}. When a voltage outside this range is input, converted values will become undefined, and converted values of other channels will be affected.

Note 3: Analog reference voltage range: ΔV_{AREF} = V_{AREF} - A_{VSS}

Note 4: When the AD converter is not used, fix the AVDD and VAREF pins to the V_{DD} level.

25.4.2. AD Conversion Characteristics of TMP89FS62B

(V_{DD} = 4.5 to 5.5 [V], V_{SS} = A_{VSS} = 0.0 [V], Topr = -40 to 85 [°C])

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Analog reference voltage/ Power supply voltage of analog control circuit	V_{AREF}/A_{VDD}		V_{DD}			V
Analog input voltage range	V_{AIN}		A_{VSS}	-	V_{AREF}/A_{VDD}	
Power supply current of analog reference voltage (Note 4)	I_{REF}	V_{DD} = V_{AREF}/A_{VDD} = 5.5 [V]	-	0.4	1.0	mA
		V_{DD} = V_{AREF}/A_{VDD} = 4.5 [V]	-	0.3	0.8	
Non-linearity error		V_{DD} = V_{AREF}/A_{VDD} = 4.5 to 5.5 [V]	-	-	±3.5	LSB
Zero point error			-	-	±3.5	
Full scale error			-	-	±3.5	
Total error			-	-	±3.5	

Note 1: The total error includes all errors except a quantization error, and is defined as the maximum deviation from the ideal conversion line.

Note 2: The voltage to be input to the AIN input pin must be within the range V_{AREF}/A_{VDD} to A_{VSS}. When a voltage outside this range is input, converted values will become undefined, and converted values of other channels will be affected.

Note 3: When the AD converter is not used, fix the VAREF/AVDD pins to the V_{DD} level.

Note 4: The power supply current of analog reference voltage includes the current to AVDD pin.

25.4.3. AD Conversion Characteristics of TMP89FS63B

(V_{DD} = 4.5 to 5.5 [V], V_{SS} = A_{VSS} = 0.0 [V], Topr = -40 to 85 [°C])

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Analog reference voltage	V_{AREF}		4.5	-	A_{VDD}	V
Power supply voltage of analog control circuit	A_{VDD}		V_{DD}			
Analog reference voltage range (Note 3)	ΔV_{AREF}		4.5	-	5.5	
Analog input voltage range	V_{AIN}		A_{VSS}	-	V_{AREF}	
Power supply current of analog reference voltage	I_{REF}	V_{DD} = A_{VDD} = V_{AREF} = 5.5 [V]	-	0.4	1.0	mA
		V_{DD} = A_{VDD} = V_{AREF} = 4.5 [V]	-	0.3	0.8	
Non-linearity error		V_{DD} = A_{VDD} = 4.5 to 5.5 [V] V_{AREF} = 4.5 to 5.5 [V]	-	-	±2	LSB
Zero point error			-	-	±2	
Full scale error			-	-	±2	
Total error			-	-	±2	

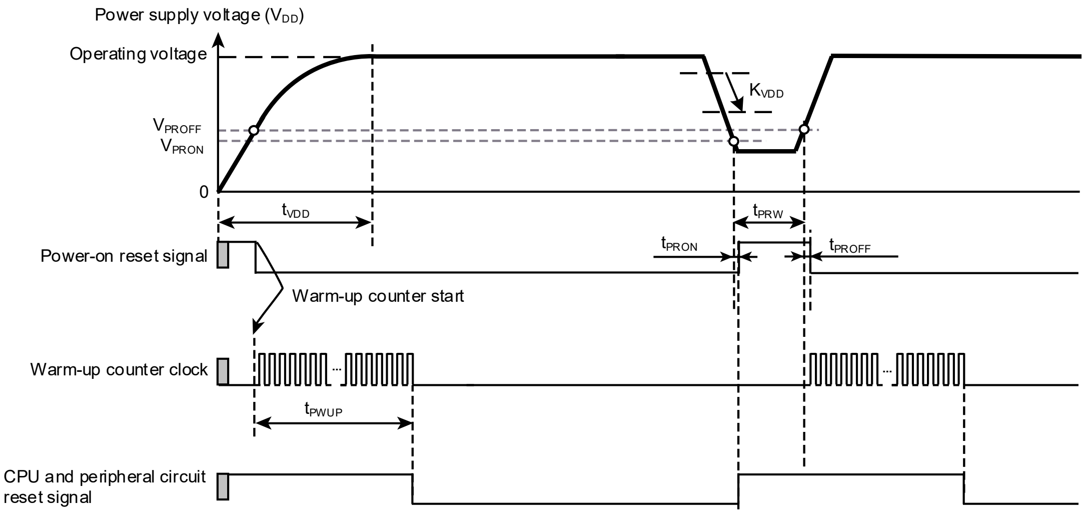
Note 1: The total error includes all errors except a quantization error, and is defined as the maximum deviation from the ideal conversion line.

Note 2: The voltage to be input to the AIN input pin must be within the range V_{AREF} to A_{VSS}. When a voltage outside this range is input, converted values will become undefined, and converted values of other channels will be affected.

Note 3: Analog reference voltage range: ΔV_{AREF} = V_{AREF} - A_{VSS}

Note 4: When the AD converter is not used, fix the AVDD and VAREF pins to the V_{DD} level.

25.5. Power-on Reset Circuit Characteristics



Note: Care must be taken in system designing by referring to “25. Electrical Characteristics” since the power-on reset circuit may not fulfill its functions due to the fluctuations in the power supply voltage (V_{DD}).

Figure 25-4 Power-on Reset Operation Timing

(V_{SS} = 0 [V], Topr = -40 to 85 [°C])

Symbol	Parameter	Min	Typ.	Max	Unit
V_{PROFF}	Power-on reset releasing voltage (Note 2)	4.16	4.26	4.36	V
V_{PRON}	Power-on reset detecting voltage (Note 2)	4.15	4.25	4.35	
t_{PROFF}	Power-on reset releasing response time	-	0.01	0.1	ms
t_{PRON}	Power-on reset detecting response time	-	0.01	0.1	
t_{PRW}	Power-on reset minimum pulse width	1.0	-	-	
t_{PWUP}	Warming-up time after a reset is cleared	-	102×2^9 / fc	-	s
t_{VDD}	Power supply rise time	0.1	-	5	ms
K_{VDD}	Power supply falling gradient	1	-	-	ms/V

Note 1: fc: High-frequency clock

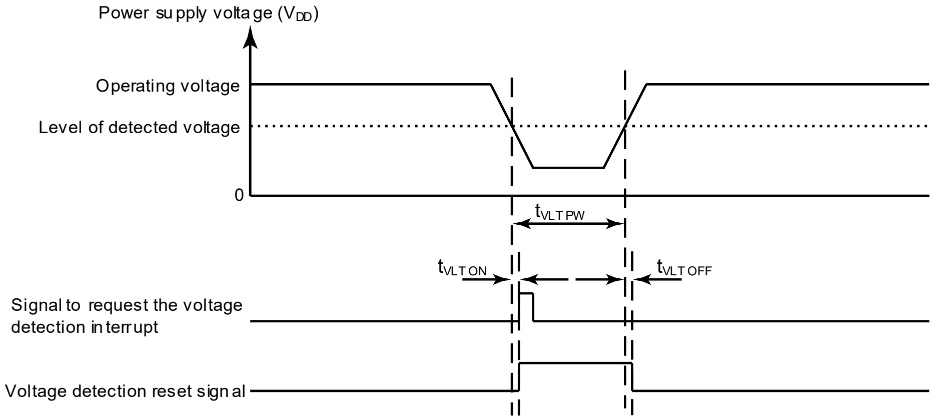
Note 2: Because the power-on reset releasing voltage and detecting voltage change relative to one another, the releasing voltage and the detecting voltage will never become inverted.

Note 3: Because the power-on reset releasing voltage, detecting voltage and the detection voltage of the VLTD change relative to one another, they will never become inverted.

Note 3: A clock output by an oscillating circuit is used as the input clock for a warming-up counter. Because the oscillation frequency does not stabilize until an oscillating circuit stabilizes, some errors may be included in the warming-up time.

Note 4: Boost the power supply voltage such that t_{VDD} becomes smaller than t_{PWUP}.

25.6. Voltage Detecting Circuit Characteristics


Power supply voltage (V_{DD})
Operating voltage
Level of detected voltage
0
t_{VLTPW}
t_{VLTON}
t_{VLTOFF}
Signal to request the voltage detection interrupt
Voltage detection reset signal

Note: Care must be taken in system designing by referring to “25. Electrical Characteristics” since the voltage detecting circuit may not fulfill its functions in the power supply voltage (V_{DD}).

Figure 25-5 Operation Timing of the Voltage Detecting Circuit

(V_{SS} = 0 [V], Topr = -40 to 85 [°C])

Symbol	Parameter	Min	Typ.	Max	Unit
t_{VLTOFF}	Voltage detection releasing response time	-	0.01	0.1	ms
t_{VLTON}	Voltage detecting detection response time	-	0.01	0.1	
t_{VLTPW}	Voltage detecting minimum pulse width	1.0	-	-	

25.7. AC Characteristics

25.7.1. MCU mode (Flash Memory Programming or Erasing)

(V_{SS} = 0 [V], V_{DD} = 4.5 to 5.5 [V], Topr = -40 to 85 [°C])

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Machine cycle time	t_{cy}	NORMAL1/2 modes	0.1	-	1	μs
High-level clock pulse width	t_{WCH}	For external clock operation (XIN input) fc = 10.0 [MHz]	-	50.0	-	ns
Low-level clock pulse width	t_{WCL}					

Note: fc: High-frequency clock [Hz]

25.7.2. MCU mode (Except Flash Memory Programming or Erasing)

(V_{SS} = 0 [V], V_{DD} = 4.2 to 5.5 [V], Topr = -40 to 85 [°C])

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Machine cycle time	t_{cy}	NORMAL1/2 modes	0.1	-	1	μs
		IDLE0/1/2 modes				
		SLOW1/2 modes	117.6	-	133.3	
		SLEEP0/1 modes				
High-level clock pulse width	t_{WCH}	For external clock operation (XIN input) fc = 10.0 [MHz]	-	50.0	-	ns
Low-level clock pulse width	t_{WCL}					
High-level clock pulse width	t_{WSH}	For external clock operation (XTIN input) fs = 32.768 [kHz]	-	15.26	-	μs
Low-level clock pulse width	t_{WSL}					

Note: fc: High-frequency clock [Hz], fs: Low-frequency clock [Hz]

25.7.3. Serial PROM Mode

(V_{SS} = 0 [V], V_{DD} = 4.5 to 5.5 [V], Topr = -40 to 85 [°C])

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Machine cycle time	t_{cy}	NORMAL1 modes	0.1	-	0.25	μs
High-level clock pulse width	t_{WCH}	For external clock operation (XIN input) fc = 10.0 [MHz]	-	50.0	-	ns
Low-level clock pulse width	t_{WCL}					

Note: fc: High-frequency clock [Hz]

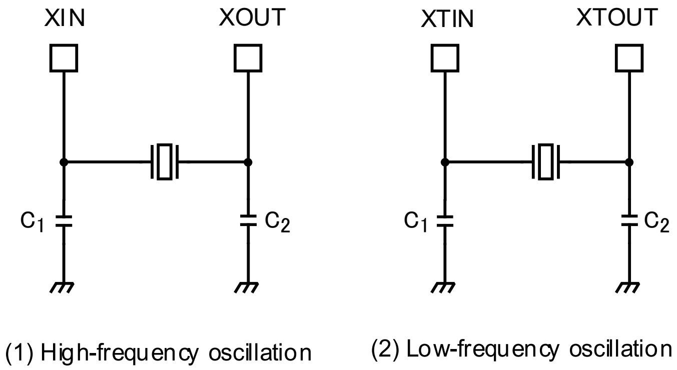
25.8. Flash Characteristics

25.8.1. Erasing and writing characteristics

(V_{SS} = 0 [V], V_{DD} = 4.5 to 5.5 [V], Topr = -40 to 85 [°C])

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Number of erasing and writing to Flash memory	n_{EW}		-	-	1000	Times
Flash memory write time	t_W		-	-	3	ms
Flash memory erase time	t_{CE}	Chip Erase	-	-	136	
	t_{SE}	Sector Erase	-	-	8	

25.9. Oscillating Condition



Note 1: To ensure stable oscillation, the resonator position, load capacitance, etc. must be appropriate. Because these factors are greatly affected by board patterns, please be sure to evaluate operation on the board on which the device will actually be mounted.

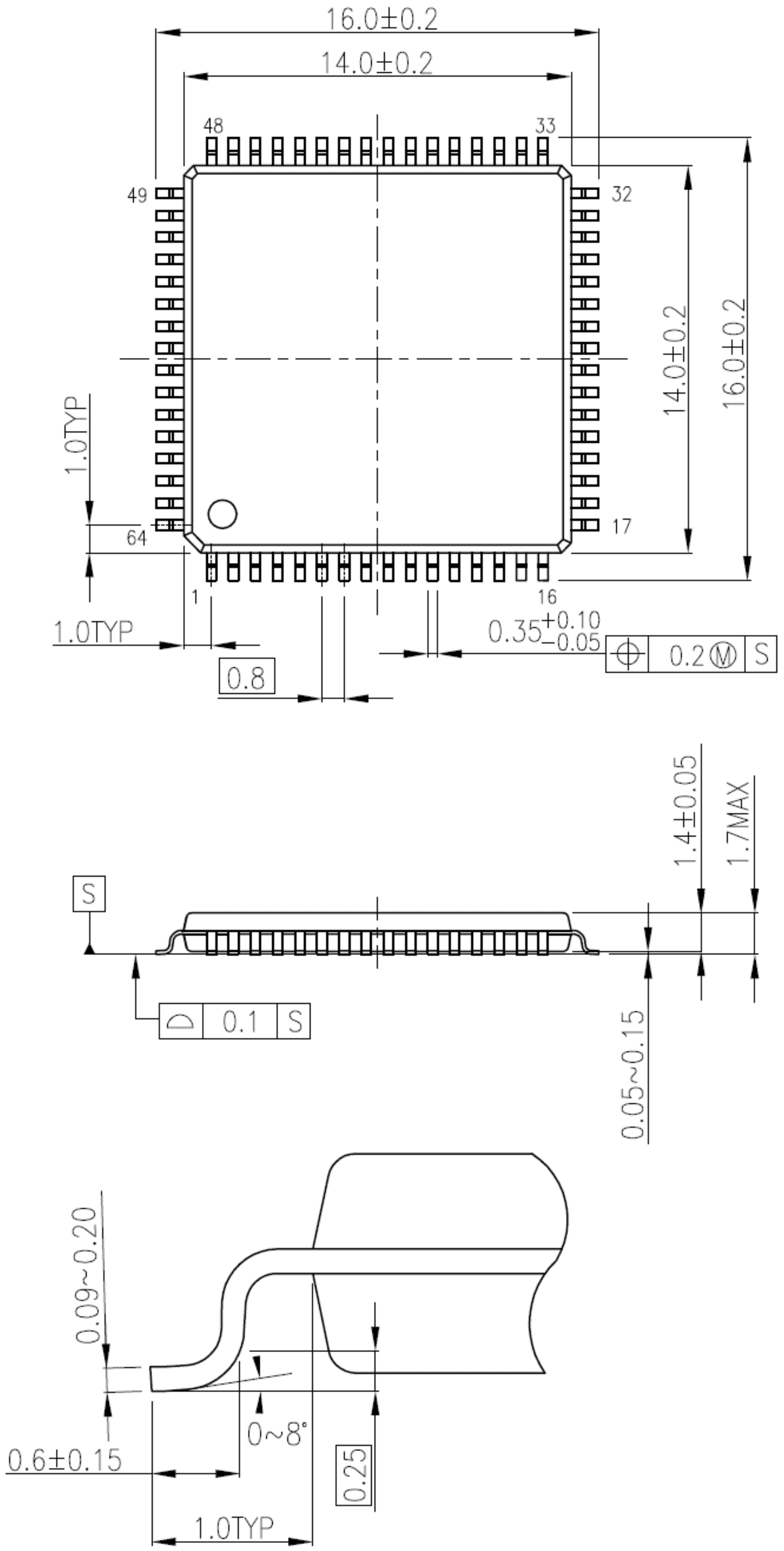
Note 2: The resonator of Murata Manufacturing Co., Ltd is recommended for the high-frequency clock oscillation circuit of this product.
Please refer to the Murata Website for details.

26. Package Dimensions

26.1. TMP89FS60BFG

P-LQFP64-1414-0.80-002

Unit: mm

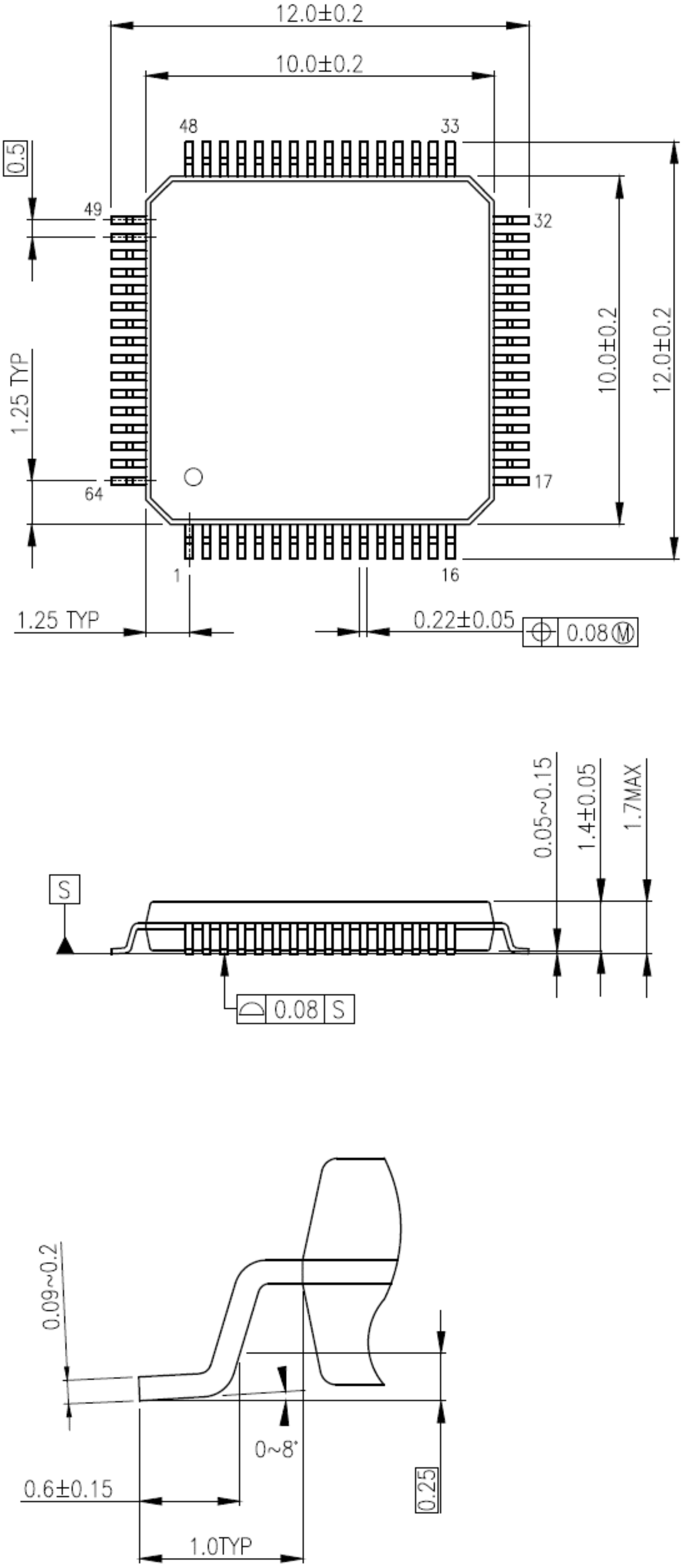


Weight: 0.67 g (typ.)

26.2. TMP89FS60BUG

P-LQFP64-1010-0.50 -003

Unit: mm

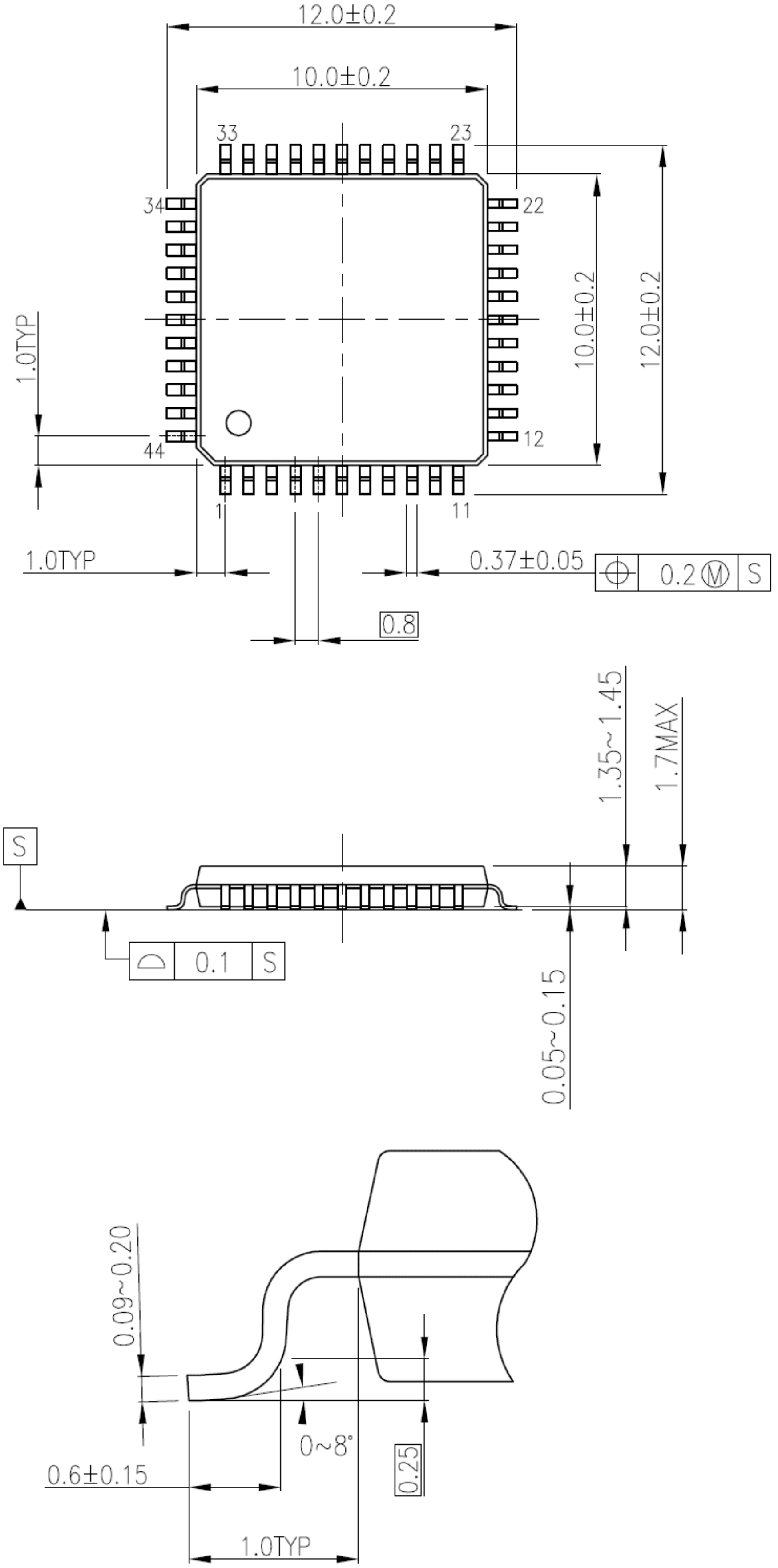


Weight: 0.37 g (typ.)

26.3. TMP89FS62BUG

P-LQFP44-1010-0.80-003

Unit: mm

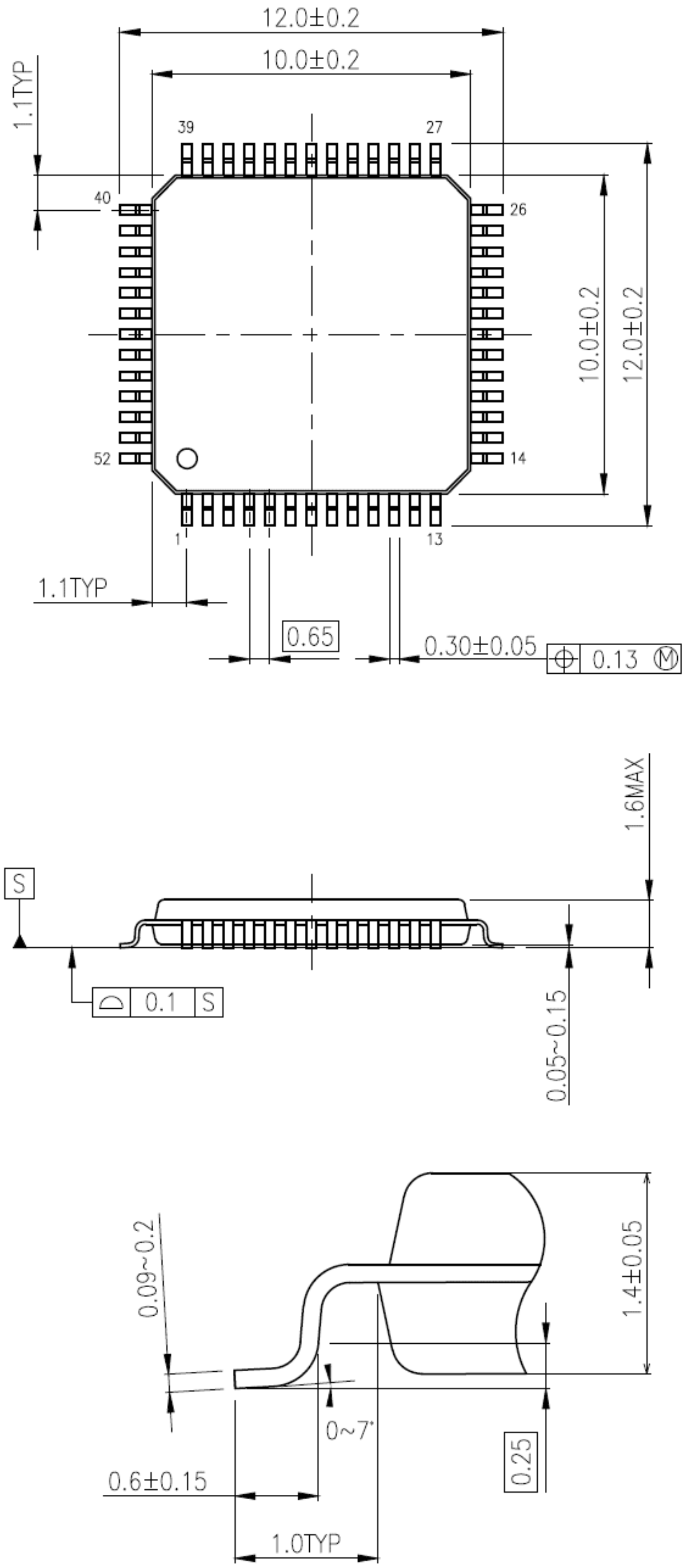


Weight: 0.37 g (typ.)

26.4. TMP89FS63BUG

P-LQFP52-1010-0.65-002

Unit: mm



Weight: 0.37 g (typ.)

RESTRICTIONS ON PRODUCT USE

Toshiba Corporation and its subsidiaries and affiliates are collectively referred to as “TOSHIBA”.
Hardware, software and systems described in this document are collectively referred to as “Product”.

- TOSHIBA reserves the right to make changes to the information in this document and related Product without notice.
- This document and any information herein may not be reproduced without prior written permission from TOSHIBA. Even with TOSHIBA's written permission, reproduction is permissible only if reproduction is without alteration/omission.
- Though TOSHIBA works continually to improve Product's quality and reliability, Product can malfunction or fail. Customers are responsible for complying with safety standards and for providing adequate designs and safeguards for their hardware, software and systems which minimize risk and avoid situations in which a malfunction or failure of Product could cause loss of human life, bodily injury or damage to property, including data loss or corruption. Before customers use the Product, create designs including the Product, or incorporate the Product into their own applications, customers must also refer to and comply with (a) the latest versions of all relevant TOSHIBA information, including without limitation, this document, the specifications, the data sheets and application notes for Product and the precautions and conditions set forth in the "TOSHIBA Semiconductor Reliability Handbook" and (b) the instructions for the application with which the Product will be used with or for. Customers are solely responsible for all aspects of their own product design or applications, including but not limited to (a) determining the appropriateness of the use of this Product in such design or applications; (b) evaluating and determining the applicability of any information contained in this document, or in charts, diagrams, programs, algorithms, sample application circuits, or any other referenced documents; and (c) validating all operating parameters for such designs and applications. **TOSHIBA ASSUMES NO LIABILITY FOR CUSTOMERS' PRODUCT DESIGN OR APPLICATIONS.**
- **PRODUCT IS NEITHER INTENDED NOR WARRANTED FOR USE IN EQUIPMENTS OR SYSTEMS THAT REQUIRE EXTRAORDINARILY HIGH LEVELS OF QUALITY AND/OR RELIABILITY, AND/OR A MALFUNCTION OR FAILURE OF WHICH MAY CAUSE LOSS OF HUMAN LIFE, BODILY INJURY, SERIOUS PROPERTY DAMAGE AND/OR SERIOUS PUBLIC IMPACT ("UNINTENDED USE").** Except for specific applications as expressly stated in this document, Unintended Use includes, without limitation, equipment used in nuclear facilities, equipment used in the aerospace industry, lifesaving and/or life supporting medical equipment, equipment used for automobiles, trains, ships and other transportation, traffic signaling equipment, equipment used to control combustions or explosions, safety devices, elevators and escalators, and devices related to power plant. **IF YOU USE PRODUCT FOR UNINTENDED USE, TOSHIBA ASSUMES NO LIABILITY FOR PRODUCT.** For details, please contact your TOSHIBA sales representative or contact us via our website.
- Do not disassemble, analyze, reverse-engineer, alter, modify, translate or copy Product, whether in whole or in part.
- Product shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable laws or regulations.
- The information contained herein is presented only as guidance for Product use. No responsibility is assumed by TOSHIBA for any infringement of patents or any other intellectual property rights of third parties that may result from the use of Product. No license to any intellectual property right is granted by this document, whether express or implied, by estoppel or otherwise.
- **ABSENT A WRITTEN SIGNED AGREEMENT, EXCEPT AS PROVIDED IN THE RELEVANT TERMS AND CONDITIONS OF SALE FOR PRODUCT, AND TO THE MAXIMUM EXTENT ALLOWABLE BY LAW, TOSHIBA (1) ASSUMES NO LIABILITY WHATSOEVER, INCLUDING WITHOUT LIMITATION, INDIRECT, CONSEQUENTIAL, SPECIAL, OR INCIDENTAL DAMAGES OR LOSS, INCLUDING WITHOUT LIMITATION, LOSS OF PROFITS, LOSS OF OPPORTUNITIES, BUSINESS INTERRUPTION AND LOSS OF DATA, AND (2) DISCLAIMS ANY AND ALL EXPRESS OR IMPLIED WARRANTIES AND CONDITIONS RELATED TO SALE, USE OF PRODUCT, OR INFORMATION, INCLUDING WARRANTIES OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, ACCURACY OF INFORMATION, OR NONINFRINGEMENT.**
- Do not use or otherwise make available Product or related software or technology for any military purposes, including without limitation, for the design, development, use, stockpiling or manufacturing of nuclear, chemical, or biological weapons or missile technology products (mass destruction weapons). Product and related software and technology may be controlled under the applicable export laws and regulations including, without limitation, the Japanese Foreign Exchange and Foreign Trade Law and the U.S. Export Administration Regulations. Export and re-export of Product or related software or technology are strictly prohibited except in compliance with all applicable export laws and regulations.
- Please contact your TOSHIBA sales representative for details as to environmental matters such as the RoHS compatibility of Product. Please use Product in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. **TOSHIBA ASSUMES NO LIABILITY FOR DAMAGES OR LOSSES OCCURRING AS A RESULT OF NONCOMPLIANCE WITH APPLICABLE LAWS AND REGULATIONS.**

TOSHIBA ELECTRONIC DEVICES & STORAGE CORPORATION
https://toshiba.semicon-storage.com/