

TOSHIBA

8 Bit Microcontroller
TLCS-870/C Series

TMP86FS64FG

TOSHIBA CORPORATION

The information contained herein is subject to change without notice. 021023 _ D

TOSHIBA is continually working to improve the quality and reliability of its products. Nevertheless, semiconductor devices in general can malfunction or fail due to their inherent electrical sensitivity and vulnerability to physical stress.

It is the responsibility of the buyer, when utilizing TOSHIBA products, to comply with the standards of safety in making a safe design for the entire system, and to avoid situations in which a malfunction or failure of such TOSHIBA products could cause loss of human life, bodily injury or damage to property.

In developing your designs, please ensure that TOSHIBA products are used within specified operating ranges as set forth in the most recent TOSHIBA products specifications.

Also, please keep in mind the precautions and conditions set forth in the "Handling Guide for Semiconductor Devices," or "TOSHIBA Semiconductor Reliability Handbook" etc. 021023_A

The Toshiba products listed in this document are intended for usage in general electronics applications (computer, personal equipment, office equipment, measuring equipment, industrial robotics, domestic appliances, etc.).

These Toshiba products are neither intended nor warranted for usage in equipment that requires extraordinarily high quality and/or reliability or a malfunction or failure of which may cause loss of human life or bodily injury ("Unintended Usage"). Unintended Usage include atomic energy control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, combustion control instruments, medical instruments, all types of safety devices, etc. Unintended Usage of Toshiba products listed in this document shall be made at the customer's own risk. 021023_B

The products described in this document shall not be used or embedded to any downstream products of which manufacture, use and/or sale are prohibited under any applicable laws and regulations. 060106_Q

The information contained herein is presented only as a guide for the applications of our products. No responsibility is assumed by TOSHIBA for any infringements of patents or other rights of the third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of TOSHIBA or others. 021023_C

The products described in this document may include products subject to the foreign exchange and foreign trade laws. 021023_F

For a discussion of how the reliability of microcontrollers can be predicted, please refer to Section 1.3 of the chapter entitled Quality and Reliability Assurance/Handling Precautions. 030619_S

Revision History

Date	Revision	
2005/12/27	1	First Release
2006/2/6	2	Contents Revised
2006/6/29	3	Periodical updating.No change in contents.
2006/8/22	4	Contents Revised

Table of Contents

TMP86FS64FG

1.1	Features	1
1.2	Pin Assignment	3
1.3	Block Diagram	4
1.4	Pin Names and Functions	6

2. Operational Descriptions

2.1	CPU Core Function	11
2.1.1	Memory Address Map	11
2.1.2	Program Memory (Flash)	12
2.1.3	Data Memory (RAM)	12
2.2	System Clock Controller	13
2.2.1	Clock Generator	13
2.2.2	Timing Generator	14
2.2.2.1	Timing Generator Configuration	
2.2.2.2	Machine Cycle	
2.2.3	Operating Modes	16
2.2.3.1	Single-Clock Mode	
2.2.3.2	Dual-Clock Mode	
2.2.3.3	STOP Mode	
2.2.3.4	Operation Mode Transition	
2.2.4	Operating Mode Control	21
2.2.4.1	STOP Mode	
2.2.4.2	IDLE1/2 and SLEEP1/2 Modes	
2.2.4.3	IDLE0 and SLEEP0 Modes	
2.2.4.4	SLOW Mode	
2.3	Reset Circuit	36
2.3.1	External Reset Input	36
2.3.2	Address-Trap-Reset	37
2.3.3	Watchdog Timer Reset	37
2.3.4	System Clock Reset	37

3. Interrupt Control Circuit

3.1	Interrupt latches (IL15 to IL2)	39
3.2	Interrupt enable register (EIR)	40
3.2.1	Interrupt master enable flag (IMF)	40
3.2.2	Individual interrupt enable flags (EF15 to EF4)	41
3.3	Interrupt Source Selector (INTSEL)	43
3.4	Interrupt Sequence	43
3.4.1	Interrupt acceptance processing is packaged as follows	43
3.4.2	Saving/restoring general-purpose registers	44
3.4.2.1	Using PUSH and POP instructions	
3.4.2.2	Using data transfer instructions	
3.4.3	Interrupt return	46
3.5	Software Interrupt (INTSW)	47
3.5.1	Address error detection	47

3.5.2	Debugging	47
3.6	Undefined Instruction Interrupt (INTUNDEF).....	47
3.7	Address Trap Interrupt (INTATRAP).....	47
3.8	External Interrupts	47

4. Special Function Register (SFR)

4.1	SFR	51
4.2	DBR	53

5. I/O Ports

5.1	Port P0 (P07 to P00)	58
5.2	Port P1 (P17 to P10)	59
5.3	Port P2 (P22 to P20)	60
5.4	Port P3 (P37 to P30)	61
5.5	Port P4 (P47 to P40)	62
5.6	Port P5 (P57 to P50)	64
5.7	Port P6 (P67to P60)	65
5.8	Port P7 (P77 to P70)	67
5.9	Port P8 (P87 to P80)	69
5.10	Port P9 (P97 to P90)	70
5.11	Port PA (PA7 to PA0)	71
5.12	Port PB (PB7 to PB0)	72

6. Watchdog Timer (WDT)

6.1	Watchdog Timer Configuration	73
6.2	Watchdog Timer Control	74
6.2.1	Malfunction Detection Methods Using the Watchdog Timer	74
6.2.2	Watchdog Timer Enable	75
6.2.3	Watchdog Timer Disable	76
6.2.4	Watchdog Timer Interrupt (INTWDT).....	76
6.2.5	Watchdog Timer Reset	77
6.3	Address Trap	78
6.3.1	Selection of Address Trap in Internal RAM (ATAS).....	78
6.3.2	Selection of Operation at Address Trap (ATOUT).....	78
6.3.3	Address Trap Interrupt (INTATRAP).....	78
6.3.4	Address Trap Reset.....	79

7. Time Base Timer (TBT)

7.1	Time Base Timer	81
7.1.1	Configuration	81
7.1.2	Control	81
7.1.3	Function	82
7.2	Divider Output (DVO)	83
7.2.1	Configuration	83
7.2.2	Control	83

8. 16-Bit TimerCounter 1 (TC1)

8.1	Configuration	85
8.2	TimerCounter Control	86
8.3	Function	88
8.3.1	Timer mode.....	88
8.3.2	External Trigger Timer Mode	90
8.3.3	Event Counter Mode.....	92
8.3.4	Window Mode	93
8.3.5	Pulse Width Measurement Mode.....	94
8.3.6	Programmable Pulse Generate (PPG) Output Mode	97

9. 16-Bit Timer/Counter2 (TC2)

9.1	Configuration	101
9.2	Control	102
9.3	Function	103
9.3.1	Timer mode.....	103
9.3.2	Event counter mode.....	105
9.3.3	Window mode	105

10. 8-Bit TimerCounter 3 (TC3)

10.1	Configuration	107
10.2	TimerCounter Control	108
10.3	Function	109
10.3.1	Timer mode.....	109
Figure 10-3	111
10.3.3	Capture Mode	112

11. 8-Bit TimerCounter 4 (TC4)

11.1	Configuration	113
11.2	TimerCounter Control	114
11.3	Function	116
11.3.1	Timer Mode.....	116
11.3.2	Event Counter Mode.....	117
11.3.3	Programmable Divider Output (PDO) Mode	118
11.3.4	Pulse Width Modulation (PWM) Output Mode	119

12. 8-Bit TimerCounter 5 (TC5)

12.1	Configuration	121
12.2	TimerCounter Control	122
12.3	Function	124
12.3.1	Timer Mode.....	124
12.3.2	Event Counter Mode.....	125
12.3.3	Programmable Divider Output (PDO) Mode	126
12.3.4	Pulse Width Modulation (PWM) Output Mode	127

13. 8-Bit TimerCounter 6 (TC6)

13.1	Configuration	129
13.2	TimerCounter Control	130
13.3	Function	132
13.3.1	Timer Mode.....	132
13.3.2	Event Counter Mode.....	133
13.3.3	Programmable Divider Output (PDO) Mode	134
13.3.4	Pulse Width Modulation (PWM) Output Mode	135

14. Asynchronous Serial interface (UART)

14.1	Configuration	137
14.2	Control	138
14.3	Transfer Data Format	140
14.4	Infrared (IrDA) Data Format Transfer Mode	141
14.5	Transfer Rate	142
14.6	Data Sampling Method	142
14.7	STOP Bit Length	143
14.8	Parity	143
14.9	Transmit/Receive Operation	143
14.9.1	Data Transmit Operation	143
14.9.2	Data Receive Operation	143
14.10	Status Flag	144
14.10.1	Parity Error.....	144
14.10.2	Framing Error.....	144
14.10.3	Overrun Error.....	144
14.10.4	Receive Data Buffer Full.....	145
14.10.5	Transmit Data Buffer Empty	145
14.10.6	Transmit End Flag	146

15. Synchronous Serial Interface (SIO1)

15.1	Configuration	147
15.2	Control	148
15.3	Serial clock	149
15.3.1	Clock source	149
15.3.1.1	Internal clock	
15.3.1.2	External clock	
15.3.2	Shift edge.....	151
15.3.2.1	Leading edge	
15.3.2.2	Trailing edge	
15.4	Number of bits to transfer	151
15.5	Number of words to transfer	151
15.6	Transfer Mode	152
15.6.1	4-bit and 8-bit transfer modes.....	152
15.6.2	4-bit and 8-bit receive modes	154
15.6.3	8-bit transfer / receive mode	155

16. Synchronous Serial Interface (SIO2)

16.1	Configuration	159
------	---------------------	-----

16.2	Control	160
16.3	Serial clock	161
16.3.1	Clock source	161
16.3.1.1	Internal clock	
16.3.1.2	External clock	
16.3.2	Shift edge	163
16.3.2.1	Leading edge	
16.3.2.2	Trailing edge	
16.4	Number of bits to transfer	163
16.5	Number of words to transfer	163
16.6	Transfer Mode	164
16.6.1	4-bit and 8-bit transfer modes	164
16.6.2	4-bit and 8-bit receive modes	166
16.6.3	8-bit transfer / receive mode	167

17. 10-bit AD Converter (ADC)

17.1	Configuration	171
17.2	Register configuration	172
17.3	Function	175
17.3.1	Software Start Mode	175
17.3.2	Repeat Mode	175
17.3.3	Register Setting	176
17.4	STOP/SLOW Modes during AD Conversion	177
17.5	Analog Input Voltage and AD Conversion Result	178
17.6	Precautions about AD Converter	179
17.6.1	Analog input pin voltage range	179
17.6.2	Analog input shared pins	179
17.6.3	Noise Countermeasure	179

18. Key-on Wakeup (KWU)

18.1	Configuration	181
18.2	Control	181
18.3	Function	181

19. Flash Memory

19.1	Flash Memory Control	184
19.1.1	Flash Memory Command Sequence Execution Control (FLSCR<FLSMD>)	184
19.1.2	Flash Memory Bank Select Control (FLSCR<BANKSEL>)	184
19.2	Command Sequence	185
19.2.1	Byte Program	185
19.2.2	Sector Erase (4-kbyte Erase)	185
19.2.3	Chip Erase (All Erase)	186
19.2.4	Product ID Entry	186
19.2.5	Product ID Exit	186
19.2.6	Read Protect	186
19.3	Toggle Bit (D6)	187
19.4	Access to the Flash Memory Area	188
19.4.1	Flash Memory Control in the Serial PROM Mode	188
19.4.1.1	How to write to the flash memory by executing the control program in the RAM area (in the RAM loader mode within the serial PROM mode)	
19.4.2	Flash Memory Control in the MCU mode	190
19.4.2.1	How to write to the flash memory by executing a user write control program in the RAM area (in the MCU mode)	

20. Serial PROM Mode

20.1	Outline	193
20.2	Memory Mapping	193
20.3	Serial PROM Mode Setting	194
20.3.1	Serial PROM Mode Control Pins	194
20.3.2	Pin Function	194
20.3.3	Example Connection for On-Board Writing	195
20.3.4	Activating the Serial PROM Mode	196
20.4	Interface Specifications for UART	197
20.5	Operation Command	199
20.6	Operation Mode	199
20.6.1	Flash Memory Erasing Mode (Operating command: F0H)	201
20.6.2	Flash Memory Writing Mode (Operation command: 30H)	203
20.6.3	RAM Loader Mode (Operation Command: 60H)	206
20.6.4	Flash Memory SUM Output Mode (Operation Command: 90H)	208
20.6.5	Product ID Code Output Mode (Operation Command: C0H)	209
20.6.6	Flash Memory Status Output Mode (Operation Command: C3H)	211
20.6.7	Flash Memory Read Protection Setting Mode (Operation Command: FAH)	212
20.7	Error Code	214
20.8	Checksum (SUM)	214
20.8.1	Calculation Method	214
20.8.2	Calculation data	215
20.9	Intel Hex Format (Binary)	216
20.10	Passwords	216
20.10.1	Password String	217
20.10.2	Handling of Password Error	217
20.10.3	Password Management during Program Development	217
20.11	Product ID Code	218
20.12	Flash Memory Status Code	218
20.13	Specifying the Erasure Area	220
20.14	Port Input Control Register	220
20.15	Flowchart	222
20.16	UART Timing	223

21. Input/Output Circuitry

21.1	Control Pins	225
21.2	Input/Output Ports	226

22. Electrical Characteristics

22.1	Absolute Maximum Ratings	227
22.2	Recommended Operating Conditions	228
22.2.1	MCU mode (Flash Programming or erasing)	228
22.2.2	MCU mode (Except Flash Programming or erasing)	228
22.2.3	Serial PROM mode	229
22.3	DC Characteristics	230
22.4	AD Characteristics	232
22.5	AC Characteristics	234
22.6	Flash Characteristics	234
22.6.1	Write/Retention Characteristics	234

22.7	Recommended Oscillating Conditions	235
22.8	Handling Precaution	235

23. Package Dimension

This is a technical document that describes the operating functions and electrical specifications of the 8-bit microcontroller series TLCS-870/C (LSI).



CMOS 8-Bit Microcontroller
TMP86FS64FG

The TMP86FS64FG is a single-chip 8-bit high-speed and high-functionality microcomputer incorporating 61440 bytes of Flash Memory. It is pin-compatible with the TMP86CS64AFG (Mask ROM version). The TMP86FS64FG can realize operations equivalent to those of the TMP86CS64AFG by programming the on-chip Flash Memory.

Product No.	ROM (FLASH)	RAM	Package	MASK ROM MCU	Emulation Chip
TMP86FS64FG	61440 bytes	2048 bytes	P-QFP100-1420-0.65A	TMP86CS64AFG	TMP86C964XB

1.1 Features

1. 8-bit single chip microcomputer TLCS-870/C series
 - Instruction execution time :
 - 0.25 μs (at 16 MHz)
 - 122 μs (at 32.768 kHz)
 - 132 types & 731 basic instructions
2. 21 interrupt sources (External : 6 Internal : 15)
3. Input / Output ports (91 pins)
 - Large current output: 16pins (Typ. 20mA), LED direct drive
4. Watchdog Timer
5. Prescaler
 - Time base timer
 - Divider output function
6. 16-bit timer counter: 1 ch
 - Timer, External trigger, Window, Pulse width measurement, Event counter, Programmable pulse generate (PPG) modes
7. 16-bit timer counter: 1 ch
 - Timer, Event counter, Window modes
8. 8-bit timer counter : 1 ch

This product uses the Super Flash® technology under the licence of Silicon Storage Technology, Inc. Super Flash® is registered trademark of Silicon Storage Technology, Inc.

060116EBP

- The information contained herein is subject to change without notice. 021023_D
- TOSHIBA is continually working to improve the quality and reliability of its products. Nevertheless, semiconductor devices in general can malfunction or fail due to their inherent electrical sensitivity and vulnerability to physical stress. It is the responsibility of the buyer, when utilizing TOSHIBA products, to comply with the standards of safety in making a safe design for the entire system, and to avoid situations in which a malfunction or failure of such TOSHIBA products could cause loss of human life, bodily injury or damage to property. In developing your designs, please ensure that TOSHIBA products are used within specified operating ranges as set forth in the most recent TOSHIBA products specifications. Also, please keep in mind the precautions and conditions set forth in the "Handling Guide for Semiconductor Devices," or "TOSHIBA Semiconductor Reliability Handbook" etc. 021023_A
- The TOSHIBA products listed in this document are intended for usage in general electronics applications (computer, personal equipment, office equipment, measuring equipment, industrial robotics, domestic appliances, etc.). These TOSHIBA products are neither intended nor warranted for usage in equipment that requires extraordinarily high quality and/or reliability or a malfunction or failure of which may cause loss of human life or bodily injury ("Unintended Usage"). Unintended Usage include atomic energy control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, combustion control instruments, medical instruments, all types of safety devices, etc. Unintended Usage of TOSHIBA products listed in this document shall be made at the customer's own risk. 021023_B
- The products described in this document shall not be used or embedded to any downstream products of which manufacture, use and/or sale are prohibited under any applicable laws and regulations. 060106_Q
- The information contained herein is presented only as a guide for the applications of our products. No responsibility is assumed by TOSHIBA for any infringements of patents or other rights of the third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of TOSHIBA or others. 021023_C
- The products described in this document are subject to the foreign exchange and foreign trade laws. 021023_E
- For a discussion of how the reliability of microcontrollers can be predicted, please refer to Section 1.3 of the chapter entitled Quality and Reliability Assurance/Handling Precautions. 030619_S

-
- Timer, Event counter, Capture modes
9. 8-bit timer counter : 3 ch
 - Timer, Event counter, Pulse width modulation (PWM) output, Programmable divider output (PDO) modes
 10. 8-bit UART : 1 ch
 11. 8-bit SIO: 2 ch
 12. 10-bit successive approximation type AD converter
 - Analog input: 16 ch
 13. Key-on wakeup : 4 ch
 14. Clock operation
 - Single clock mode
 - Dual clock mode
 15. Low power consumption operation
 - STOP mode: Oscillation stops. (Battery/Capacitor back-up.)
 - SLOW1 mode: Low power consumption operation using low-frequency clock.(High-frequency clock stop.)
 - SLOW2 mode: Low power consumption operation using low-frequency clock.(High-frequency clock oscillate.)
 - IDLE0 mode: CPU stops, and only the Time-Based-Timer(TBT) on peripherals operate using high frequency clock. Release by falling edge of the source clock which is set by TBTCR<TBTCK>.
 - IDLE1 mode: CPU stops and peripherals operate using high frequency clock. Release by interrupts(CPU restarts).
 - IDLE2 mode: CPU stops and peripherals operate using high and low frequency clock. Release by interrupts. (CPU restarts).
 - SLEEP0 mode: CPU stops, and only the Time-Based-Timer(TBT) on peripherals operate using low frequency clock.Release by falling edge of the source clock which is set by TBTCR<TBTCK>.
 - SLEEP1 mode: CPU stops, and peripherals operate using low frequency clock. Release by interrupt.(CPU restarts).
 - SLEEP2 mode: CPU stops and peripherals operate using high and low frequency clock. Release by interrupt.
 16. Wide operation voltage:
 - 4.5 V to 5.5 V at 16MHz /32.768 kHz
 - 2.7 V to 5.5 V at 8 MHz /32.768 kHz

1.2 Pin Assignment

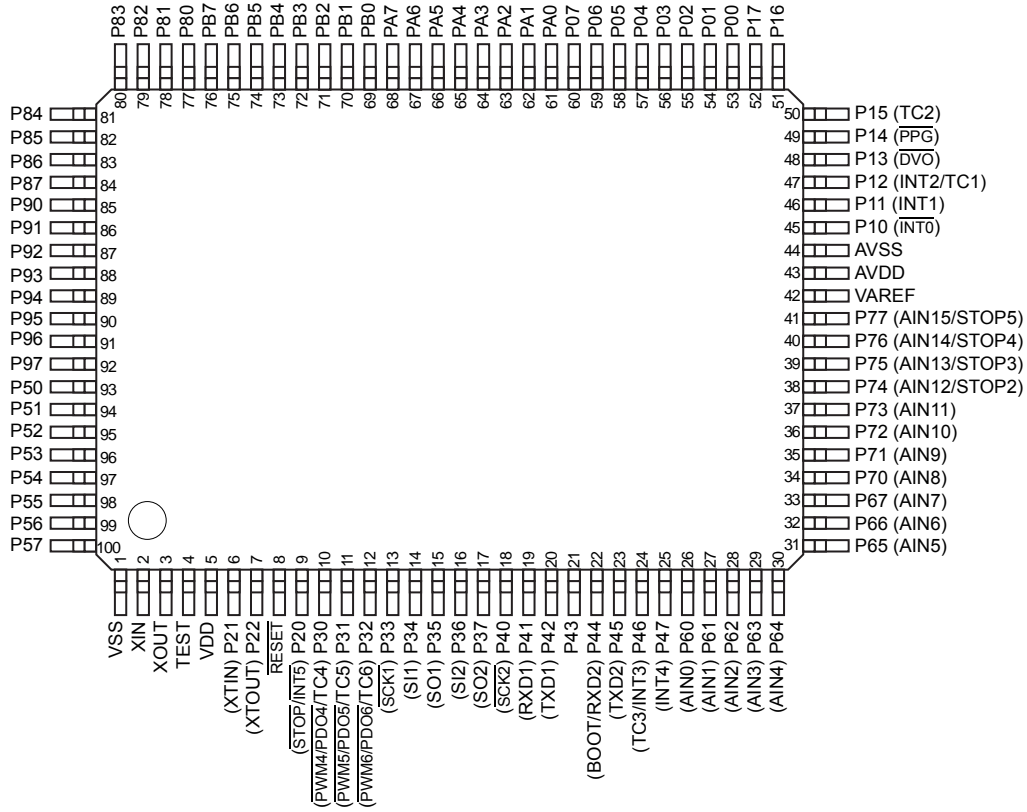


Figure 1-1 Pin Assignment

1.3 Block Diagram

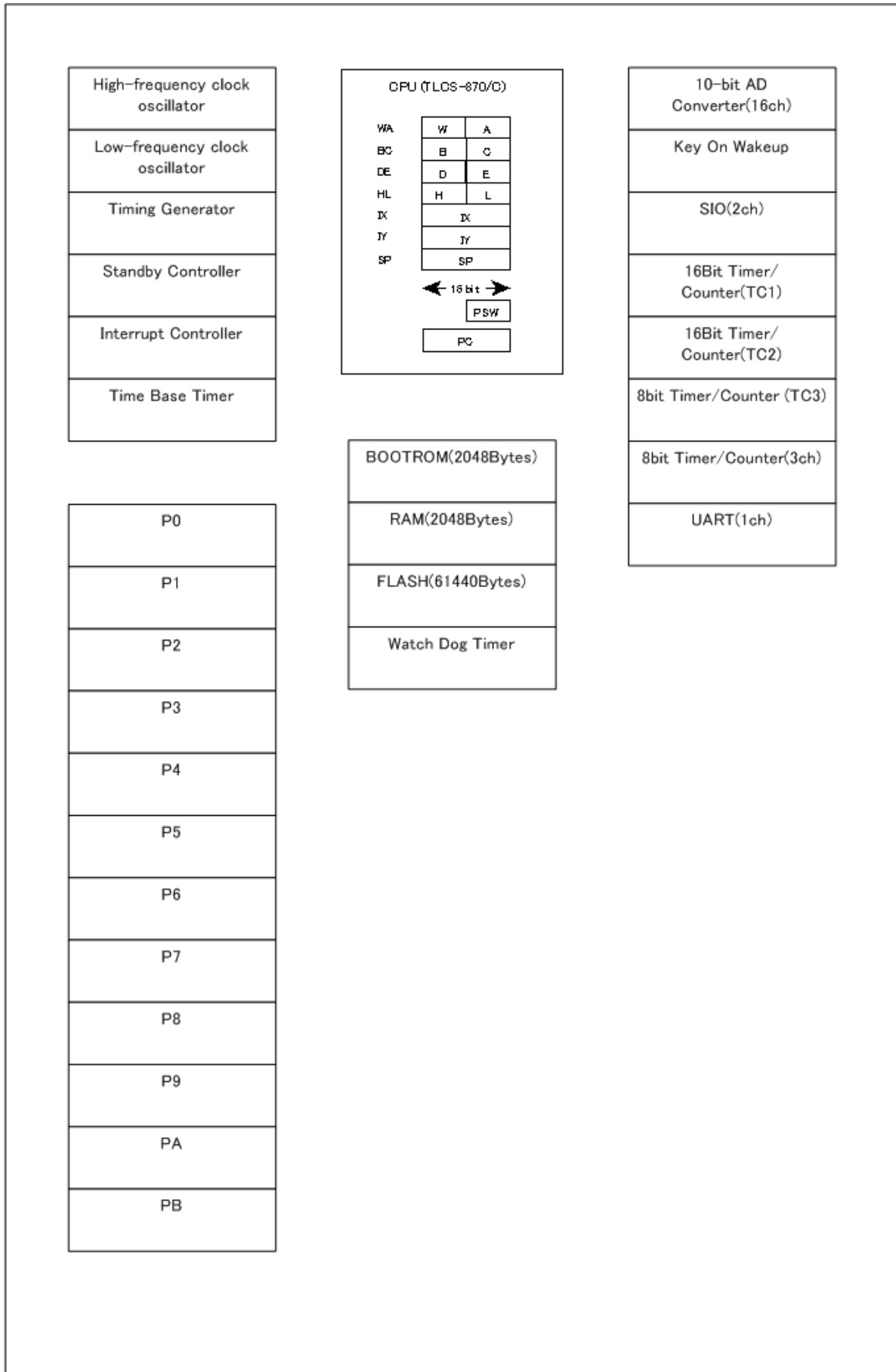


Figure 1-2 Block Diagram



1.4 Pin Names and Functions

The TMP86FS64FG has MCU mode, parallel PROM mode, and serial PROM mode. Table 1-1 shows the pin functions in MCU mode. The serial PROM mode is explained later in a separate chapter.

Table 1-1 Pin Names and Functions(1/4)

Pin Name	Pin Number	Input/Output	Functions
P07	60	IO	PORT07
P06	59	IO	PORT06
P05	58	IO	PORT05
P04	57	IO	PORT04
P03	56	IO	PORT03
P02	55	IO	PORT02
P01	54	IO	PORT01
P00	53	IO	PORT00
P17	52	IO	PORT17
P16	51	IO	PORT16
P15 TC2	50	IO I	PORT15 TC2 input
P14 PPG	49	IO O	PORT14 PPG output
P13 DVO	48	IO O	PORT13 Divider Output
P12 INT2 TC1	47	IO I I	PORT12 External interrupt 2 input TC1 input
P11 INT1	46	IO I	PORT11 External interrupt 1 input
P10 INT0	45	IO I	PORT10 External interrupt 0 input
P22 XTOUT	7	IO O	PORT22 Low frequency OSC output pin
P21 XTIN	6	IO I	PORT21 Low frequency OSC input pin
P20 INT5 STOP	9	IO I I	PORT20 External interrupt 5 input STOP mode release input
P37 SO2	17	IO O	PORT37 Serial Data Output 2
P36 SI2	16	IO I	PORT36 Serial Data Input 2
P35 SO1	15	IO O	PORT35 Serial Data Output 1
P34 SI1	14	IO I	PORT34 Serial Data Input 1
P33 SCK1	13	IO IO	PORT33 Serial Clock I/O 1

Table 1-1 Pin Names and Functions(2/4)

Pin Name	Pin Number	Input/Output	Functions
P32 TC6 PWM6/PDO6	12	IO I O	PORT32 TC6 input PWM6/PDO6 output
P31 TC5 PWM5/PDO5	11	IO I O	PORT31 TC5 input PWM5/PDO5 output
P30 TC4 PWM4/PDO4	10	IO I O	PORT30 TC4 input PWM4/PDO4 output
P47 INT4	25	IO I	PORT47 External interrupt 4 input
P46 INT3 TC3	24	IO I I	PORT46 External interrupt 3 input TC3 pin input
P45 TXD2	23	IO O	PORT45 UART data output 2
P44 RXD2 BOOT	22	IO I I	PORT44 UART data input 2 Serial PROM mode control input
P43	21	IO	PORT43
P42 TXD1	20	IO O	PORT42 UART data output 1
P41 RXD1	19	IO I	PORT41 UART data input 1
P40 SCK2	18	IO IO	PORT40 Serial Clock I/O 2
P57	100	IO	PORT57
P56	99	IO	PORT56
P55	98	IO	PORT55
P54	97	IO	PORT54
P53	96	IO	PORT53
P52	95	IO	PORT52
P51	94	IO	PORT51
P50	93	IO	PORT50
P67 AIN7	33	IO I	PORT67 Analog Input7
P66 AIN6	32	IO I	PORT66 Analog Input6
P65 AIN5	31	IO I	PORT65 Analog Input5
P64 AIN4	30	IO I	PORT64 Analog Input4
P63 AIN3	29	IO I	PORT63 Analog Input3
P62 AIN2	28	IO I	PORT62 Analog Input2

Table 1-1 Pin Names and Functions(3/4)

Pin Name	Pin Number	Input/Output	Functions
P61 AIN1	27	IO I	PORT61 Analog Input1
P60 AIN0	26	IO I	PORT60 Analog Input0
P77 AIN15 STOP5	41	IO I I	PORT77 Analog Input15 STOP5 input
P76 AIN14 STOP4	40	IO I I	PORT76 Analog Input14 STOP4 input
P75 AIN13 STOP3	39	IO I I	PORT75 Analog Input13 STOP3 input
P74 AIN12 STOP2	38	IO I I	PORT74 Analog Input12 STOP2 input
P73 AIN11	37	IO I	PORT73 Analog Input11
P72 AIN10	36	IO I	PORT72 Analog Input10
P71 AIN9	35	IO I	PORT71 Analog Input9
P70 AIN8	34	IO I	PORT70 Analog Input8
P87	84	IO	PORT87
P86	83	IO	PORT86
P85	82	IO	PORT85
P84	81	IO	PORT84
P83	80	IO	PORT83
P82	79	IO	PORT82
P81	78	IO	PORT81
P80	77	IO	PORT80
P97	92	IO	PORT97
P96	91	IO	PORT96
P95	90	IO	PORT95
P94	89	IO	PORT94
P93	88	IO	PORT93
P92	87	IO	PORT92
P91	86	IO	PORT91
P90	85	IO	PORT90
PA7	68	IO	PORTA7
PA6	67	IO	PORTA6

Table 1-1 Pin Names and Functions(4/4)

Pin Name	Pin Number	Input/Output	Functions
PA5	66	IO	PORTA5
PA4	65	IO	PORTA4
PA3	64	IO	PORTA3
PA2	63	IO	PORTA2
PA1	62	IO	PORTA1
PA0	61	IO	PORTA0
PB7	76	IO	PORTB7
PB6	75	IO	PORTB6
PB5	74	IO	PORTB5
PB4	73	IO	PORTB4
PB3	72	IO	PORTB3
PB2	71	IO	PORTB2
PB1	70	IO	PORTB1
PB0	69	IO	PORTB0
XIN	2	I	High frequency OSC input pin
XOUT	3	I	High frequency OSC output pin
RESET	8	I	RESET input
TEST	4	I	TEST pin (Fix to Low level)
VAREF	42	I	Analog Base Voltage Input Pin for A/D Conversion
AVDD	43	I	Analog Power Supply
AVSS	44	I	Analog Power Supply
VDD	5	I	VDD pin
VSS	1	I	GND pin

2. Operational Descriptions

2.1 CPU Core Function

The CPU core consists of a CPU, a system clock controller and an interrupt controller.

This chapter provides descriptions of the CPU core, the program memory, the data memory and the reset circuit.

2.1.1 Memory Address Map

TMP86FS64FG memory consists of RAM and Special Function Register (SFR), which are mapped to a 64 Kbyte address space.

The TMP86FS64FG memory consists of Flash, RAM, Special Function Register (SFR) and Data Buffer Register (DBR), which are mapped to a 64 kbyte address space.

Figure 2-1 shows the TMP86FS64FG memory address map.

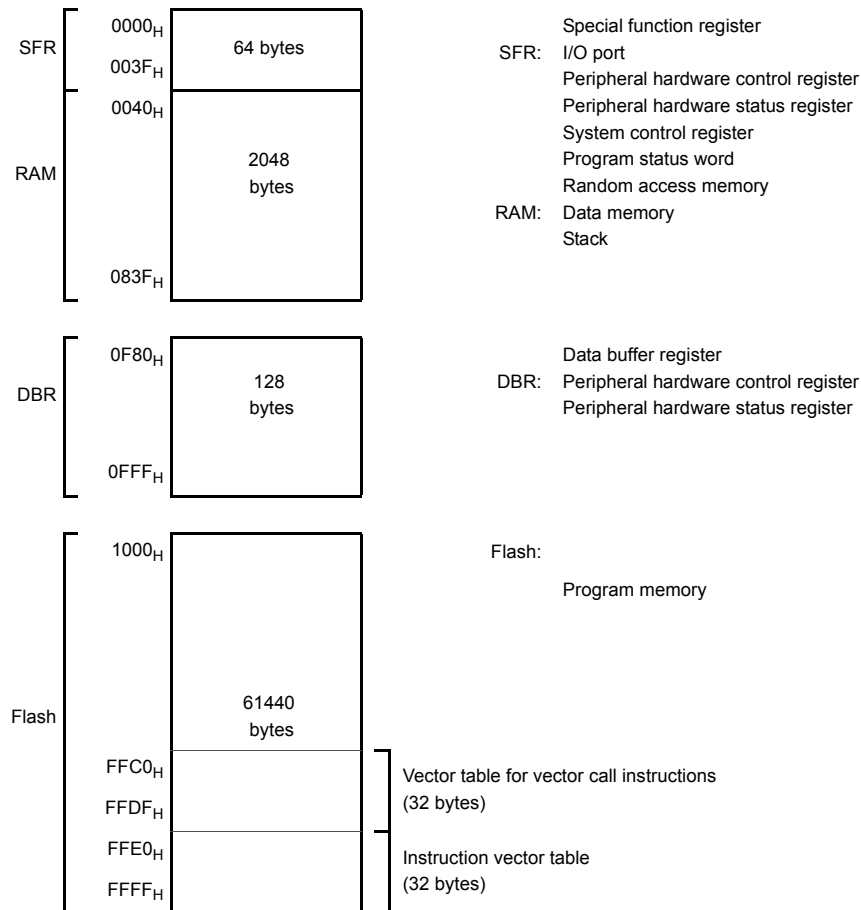


Figure 2-1 Memory Address Map

2.1.2 Program Memory (Flash)

TMP86FS64FG incorporates the 61440-byte (addresses from 1000H through FFFFH) program memory (Flash).

2.1.3 Data Memory (RAM)

TMP86FS64FG incorporates the 2048-byte (addresses from 0040H through 083FH) RAM. Since the address space from 0040H through 00FFH within the on-chip RAM can be accessed directly, it can be accessed by instructions to shorten the processing time.

Perform initial setting through an initialize routine since the contents of the data memory become don't cares at power-up.

Example :Clearing RAM of TMP86FS64FG

	LD	HL, 0040H	: Sets the start address
	LD	A, H	: Sets the initialization data (00H)
	LD	BC,07FFH	: Sets the number of bytes (-1)
SRAMCLR:	LD	(HL), A	
	INC	HL	
	DEC	BC	
	JRS	F, SRAMCLR	

2.2 System Clock Controller

The system clock controller consists of a clock generator, a timing generator and an operating mode controller.

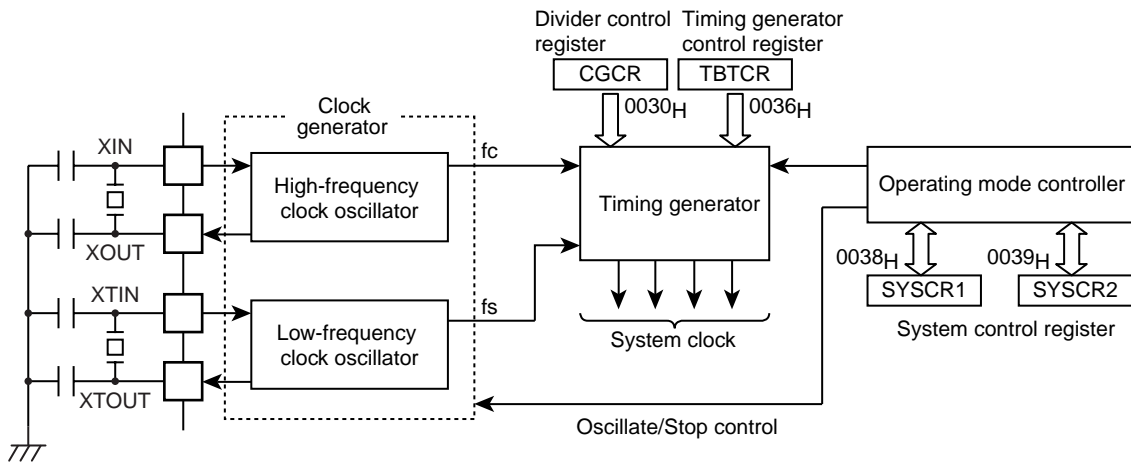


Figure 2-2 System Clock Controller

2.2.1 Clock Generator

The clock generator generates the basic clock which provides the system clocks to be supplied to the CPU core and peripheral hardware. The clock generator contains two oscillators used for the high- and low-frequency clocks. Power consumption can be reduced by the low-speed operation with the low-frequency clock, which is switched by the operating mode controller.

The high-frequency clock (fc) or low-frequency clock (fs) can be obtained easily by connecting a resonator between the XIN and XOUT pins, or XTIN and XTOUT pins, respectively. The clock can be supplied from an external oscillator. In this case, supply the clock via the XIN or XTIN pin, and leave the XOUT or XTOUT pins unconnected.

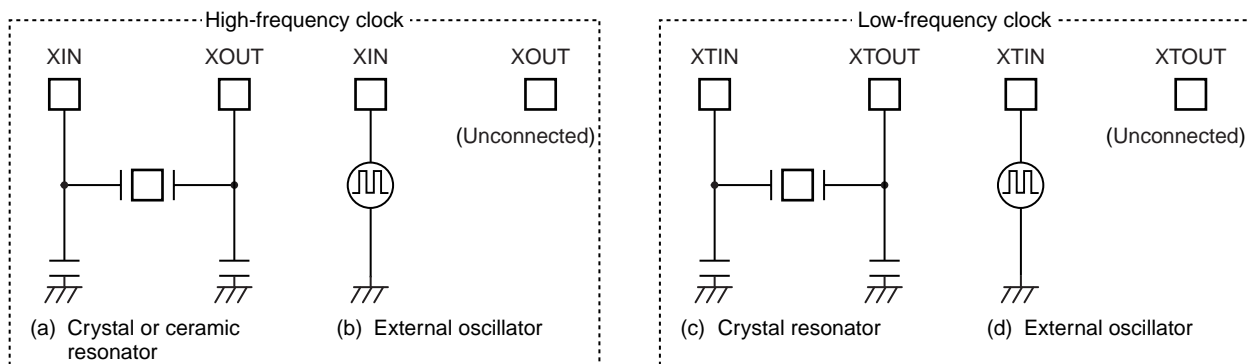


Figure 2-3 Example Resonator Connection

Note: The hardware feature does not provide the function to monitor externally the basic clock directly. However, with disabling all interrupts and watchdog timers, the oscillation frequency can be adjusted by programming to output a fixed-frequency pulse (i.e., clock output) to a port and monitoring the pulse. For the system to require the adjustment of the oscillation frequency, the adjustment program must be created beforehand.

2.2.2 Timing Generator

The timing generator generates various types of system clocks which are supplied to the CPU core or peripheral hardware from the basic clock (f_c or f_s). The timing generator provides the following functions.

1. Generating the main system clock
2. Generating the divider output (\overline{DVO}) pulses
3. Generating the source clocks for the time base timer
4. Generating the source clocks for the watchdog timer
5. Generating the internal source clocks for the TimerCounter
6. Generating the warm-up clocks upon exit from the STOP mode

2.2.2.1 Timing Generator Configuration

The timing generator consists of a 3-stage prescaler, a 21-stage divider, a main system clock generator and a machine cycle counter.

Either the clock $f_c/4$ output from the 2nd stage or the clock $f_c/8$ output from the 3rd stage can be selected as the clock input to the 1st stage of the divider by $CGCR\langle DV1CK \rangle$. This function enables to operate the peripheral circuits without program change by inputting $f_c/8$ to the 1st stage of the divider when the operation clock is multiplied by 2. (ex., 8 MHz to 16 MHz)

The input clock to the 7th stage of the divider depends on $SYSCR2\langle SYSCK \rangle$, $TBTCR\langle DV7CK \rangle$ and $CGCR\langle DV1CK \rangle$ settings, as shown in Table 2-2.

The prescaler and divider are cleared to 0 upon reset and entry to/exit from the STOP mode.

Note: $TBTCR\langle DV7CK \rangle$ indicates the bit 4 ($DV7CK$) of the timing generator ($TBTCR$). Hereafter, this notational convention is used for each functional bit of the register.

Table 2-1 Divider Output

Divider Output									
$DV1CK = 0$					$DV1CK = 1$				
DV1G	DV2G	DV3G	DV4	DV5	DV1G	DV2G	DV3G	DV4	DV5
$f_c/2^3$	$f_c/2^4$	$f_c/2^5$	$f_c/2^6$	$f_c/2^7$	$f_c/2^4$	$f_c/2^5$	$f_c/2^6$	$f_c/2^7$	$f_c/2^8$

Table 2-2 Input Clock to 7th Stage of the Divider [Hz]

NORMAL1, IDLE1 mode		NORMAL2, IDLE2 mode ($SYSCK=0$)				SLOW1/2, SLEEP1/2 mode ($SYSCK = 1$)
$DV7CK = 0$		$DV7CK = 0$		$DV7CK = 1$		
$DV1CK = 0$	$DV1CK = 1$	$DV1CK = 0$	$DV1CK = 1$			
$f_c/2^8$	$f_c/2^9$	$f_c/2^8$	$f_c/2^9$	f_s	f_s	

Note 1: Do not set $TBTCR\langle DV7CK \rangle$ to 1 during the NORMAL1 or IDLE1 mode.

Note 2: Since the input clock to the 1st stage of the divider is stopped in the SLOW1/2 or SLEEP1/2 mode, output from the 1st to 7th stages of the divider is also stopped.

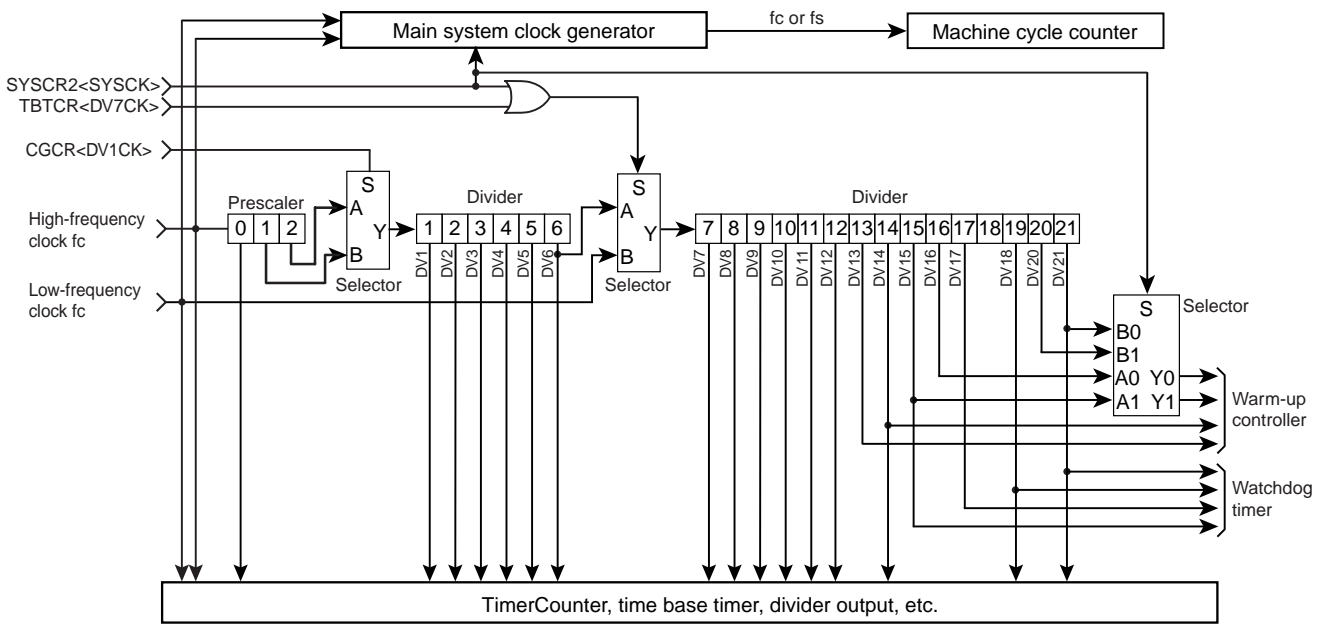


Figure 2-4 Timing Generator Configuration

Table 2-3 Division Ratio of the Divider

	DV7CK = 0		DV7CK = 1			DV7CK = 0		DV7CK = 1	
	DV1CK = 0	DV1CK = 1	DV1CK = 0	DV1CK = 1		DV1CK = 0	DV1CK = 1	DV1CK = 0	DV1CK = 1
DV1	$f_c/2^3$	$f_c/2^4$	$f_c/2^3$	$f_c/2^4$	DV12	$f_c/2^{14}$	$f_c/2^{15}$	$f_s/2^6$	
DV2	$f_c/2^4$	$f_c/2^5$	$f_c/2^4$	$f_c/2^5$	DV13	$f_c/2^{15}$	$f_c/2^{16}$	$f_s/2^7$	
DV3	$f_c/2^5$	$f_c/2^6$	$f_c/2^5$	$f_c/2^6$	DV14	$f_c/2^{16}$	$f_c/2^{17}$	$f_s/2^8$	
DV4	$f_c/2^6$	$f_c/2^7$	$f_c/2^6$	$f_c/2^7$	DV15	$f_c/2^{17}$	$f_c/2^{18}$	$f_s/2^9$	
DV5	$f_c/2^7$	$f_c/2^8$	$f_c/2^7$	$f_c/2^8$	DV16	$f_c/2^{18}$	$f_c/2^{19}$	$f_s/2^{10}$	
DV6	$f_c/2^8$	$f_c/2^9$	$f_c/2^8$	$f_c/2^9$	DV17	$f_c/2^{19}$	$f_c/2^{20}$	$f_s/2^{11}$	
DV7	$f_c/2^9$	$f_c/2^{10}$	$f_s/2$		DV18	$f_c/2^{20}$	$f_c/2^{21}$	$f_s/2^{12}$	
DV8	$f_c/2^{10}$	$f_c/2^{11}$	$f_s/2^2$		DV19	$f_c/2^{21}$	$f_c/2^{22}$	$f_s/2^{13}$	
DV9	$f_c/2^{11}$	$f_c/2^{12}$	$f_s/2^3$		DV20	$f_c/2^{22}$	$f_c/2^{23}$	$f_s/2^{14}$	
DV10	$f_c/2^{12}$	$f_c/2^{13}$	$f_s/2^4$		DV21	$f_c/2^{23}$	$f_c/2^{24}$	$f_s/2^{15}$	
DV11	$f_c/2^{13}$	$f_c/2^{14}$	$f_s/2^5$						

Divider Control Register

CGCR (0030H)	7	6	5	4	3	2	1	0	
	"0"	"0"	DV1CK	"0"	"0"	"0"	"0"	"0"	(Initial value: **0* ****)

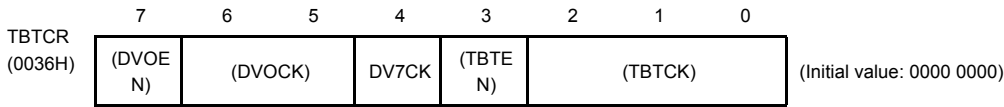
DV1CK	Selection of the input clock to the 1st stage of the divider [Hz]	0: $f_c/4$ 1: $f_c/8$	R/W
-------	---	--------------------------	-----

Note 1: f_c : High-frequency clock [Hz], *: Don't care

Note 2: The bit 4 and 3 are read as a don't care when the read instruction is executed to CGCR.

Note 3: 0 must be written to the bit 7, 6, 4 through 0 of CGCR.

Timing Generator Control Register



Note 1: Do not set DV7CK to 1 in the single-clock mode.

Note 2: Do not set DV7CK to 1 until the low-frequency clock oscillation is stabilized.

Note 3: fc: High-frequency clock [Hz], fs: Low-frequency clock [Hz], *: Don't care

Note 4: In the SLOW 1/2 or SLEEP1/2 mode, fs is input to the 7th stage of the divider regardless of DV7CK setting.

Note 5: When the STOP mode is entered from the NORMAL1/2 mode, the output of the 6th stage of the divider is input to the 7th stage of the divider during warm up after exiting from the STOP mode regardless of DV7CK setting.

2.2.2.2 Machine Cycle

The instruction execution and peripheral hardware operation are synchronized with the system clock. The minimum instruction execution unit is called a "machine cycle". There are 10 types of instructions for TLCS-870/C Series, which are 1-cycle instructions to be executed within 1-cycle through 10-cycle instructions to be executed within ten cycles.

A machine cycle consists of 4 states (S0 to S3), and each state consists of one main system clock.

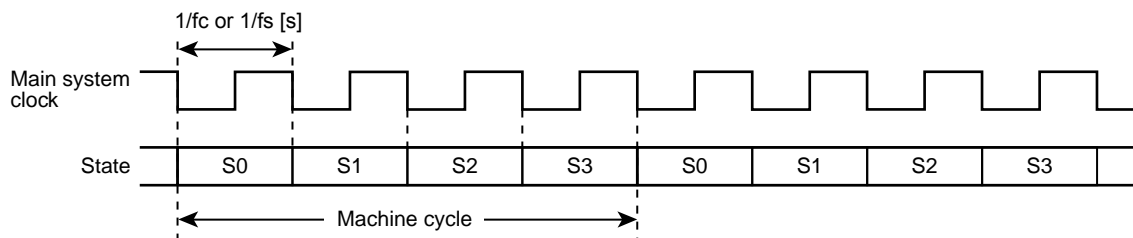


Figure 2-5 Machine Cycle

2.2.3 Operating Modes

The operating mode controller starts and stops the oscillators for the high-frequency and low-frequency clocks, and switches the main system clock. The device has the single-clock, dual-clock and STOP modes, which can be controlled by the system control registers (SYSCR1 and SYSCR2). Figure 2-6 shows the operating mode transition.

2.2.3.1 Single-Clock Mode

In the single-clock mode, only the oscillator for high-frequency clock is used. The P21(XTIN) and P22(XTOUT) pins for the low-frequency clock can be used as usual I/O ports. Since the main system clock is generated from the high-frequency clock, the machine cycle time becomes $4/f_c$ [s] in the single-clock mode.

(1) NORMAL1 mode

In the NORMAL1 mode, the CPU core and on-chip peripherals operate using the high-frequency clock. After reset is released, NORMAL1 mode is entered.

(2) IDLE1 mode

In the IDLE1 mode, the CPU and watchdog timer are halted, and on-chip peripherals are clocked by the high-frequency clock. To enter the IDLE1 mode, set IDEL in the system control register 2 (SYSCR2) to 1. The IDLE1 mode is exited by the interrupt from the on-chip peripherals or external interrupts, and returned to the NORMAL1 mode. When the IMF (interrupt master enable flag) is set to 1 (interrupt enable), the normal operation is performed after the interrupt processing is completed. When the IMF is set to 0 (interrupt disable), program execution resumes with the instruction immediately following the instruction that activated the IDLE1 mode.

(3) IDLE0 mode

In the IDLE0 mode, the CPU and on-chip peripherals are halted except oscillator and TBT. The IDEL0 mode is entered by setting the system control register SYSCR2<TGHALT> to 1 in the NORMAL1 mode. When the IDLE0 mode is entered, the CPU is halted and the timing generator stops clocking to the peripherals except TBT. When detecting the falling edge of the source clock set in TBTCR<TBTCCK>, the timing generator starts clocking to all on-chip peripherals.

When the IDLE0 mode is exited, the CPU restarts operation and returns to the NORMAL1 mode. The IDLE0 mode is entered and returned to the NORMAL1 mode regardless of setting in TBTCR<TBTEN>. Interrupt processing is performed when IMF = 1, EF8 (TBT interrupt enable flag) = 1, and TBTCR<TBTEN> = 1.

When the IDLE0 mode is entered with TBTCR<TBTEN> = 1, INTTBT interrupt latch is set after returning to the NORMAL mode.

2.2.3.2 Dual-Clock Mode

In the dual-clock mode, two oscillators for high-frequency and low-frequency are used. The P21(XTIN) and P22(XTOUT) pins are used for the low-frequency clock pins. (In the dual-clock mode, these pins can not be used as I/O ports.) The main system clock is generated by the high-frequency clock in the NORMAL2 and IDLE2 modes, and the low-frequency clock in the SLOW1/2 and SLEEP1/2 modes. Therefore, the machine cycle time is $4/f_c$ [s] in the NORMAL2 and IDLE2 modes, and $4/f_s$ [s] ($122 \mu\text{s}$ @ $f_s = 32.768 \text{ kHz}$) in the SLOW and SLEEP modes.

The TLCS-870/C series is put in the single-clock mode during reset. To use the dual-clock mode, oscillate the low-frequency clock at the top of the program.

(1) NORMAL2 Mode

The CPU core operates with high-frequency clock. On-chip peripherals operate with high- and low-frequency clocks.

(2) SLOW2 Mode

The CPU core operates with low-frequency clock. Switching from NORMAL2 to SLOW2, and vice-versa is programmed in SYSCR2<SYSCK>. Do not clear XTEN to 0 in the SLOW2 mode.

(3) SLOW1 Mode

Power dissipation can be reduced by stopping high-frequency clock oscillation, and operating the CPU core and on-chip peripherals with low-frequency clock.

Switching from SLOW1 to SLOW2, and vice-versa is programmed in SYSCR2<XEN>. In the SLOW1 and SLEEP1 modes, output from the 1st to 6th stages is stopped.

(4) IDLE2 Mode

The CPU and watchdog timer are halted, and on-chip peripherals are operated with the high- and low-frequency clocks. Entering and exiting the IDLE2 mode is the same as for the IDLE1 mode. After exiting the IDLE2 mode, the CPU returns to the NORMAL2 mode.

(5) SLEEP1 Mode

The CPU and watchdog timer are halted, and on-chip peripherals are operated with the low-frequency clock. Entering and exiting the SLEEP1 mode is the same as for the IDLE1 mode. After exiting the SLEEP1 mode, the CPU returns to the SLOW1 mode. High-frequency clock oscillation is stopped. In the SLOW1 and SLEEP1 modes, output from the 1st to 6th stages is stopped.

(6) SLEEP2 Mode

The SLEEP2 mode is the idle mode corresponding to the SLOW2 mode. The SLEEP2 mode is the same as the SLOW2 mode except that high-frequency clock is activated.

(7) SLEEP0 Mode

The CPU and on-chip peripherals are halted except oscillator and TBT. The SLEEP0 mode is entered by setting the system control register SYSCR2<TGHALT> to 1 in the SLOW1 mode. When the SLEEP0 mode is entered, the CPU is halted and the timing generator stops clocking to the peripherals except TBT. When detecting the falling edge of the source clock set in TBTCR<TBTCK>, the timing generator starts the clocking operation to all on-chip peripherals.

When the SLEEP0 mode is exited, the CPU restarts operation and returns to the SLOW1 mode. the CPU enters to the SLEEP0 mode and returns to the SLOW1 mode regardless of setting in TBTCR<TBTEN>. Interrupt processing is performed when IMF = 1, EF8 (TBT interrupt enable flag) = 1, and TBTCR<TBTEN> = 1.

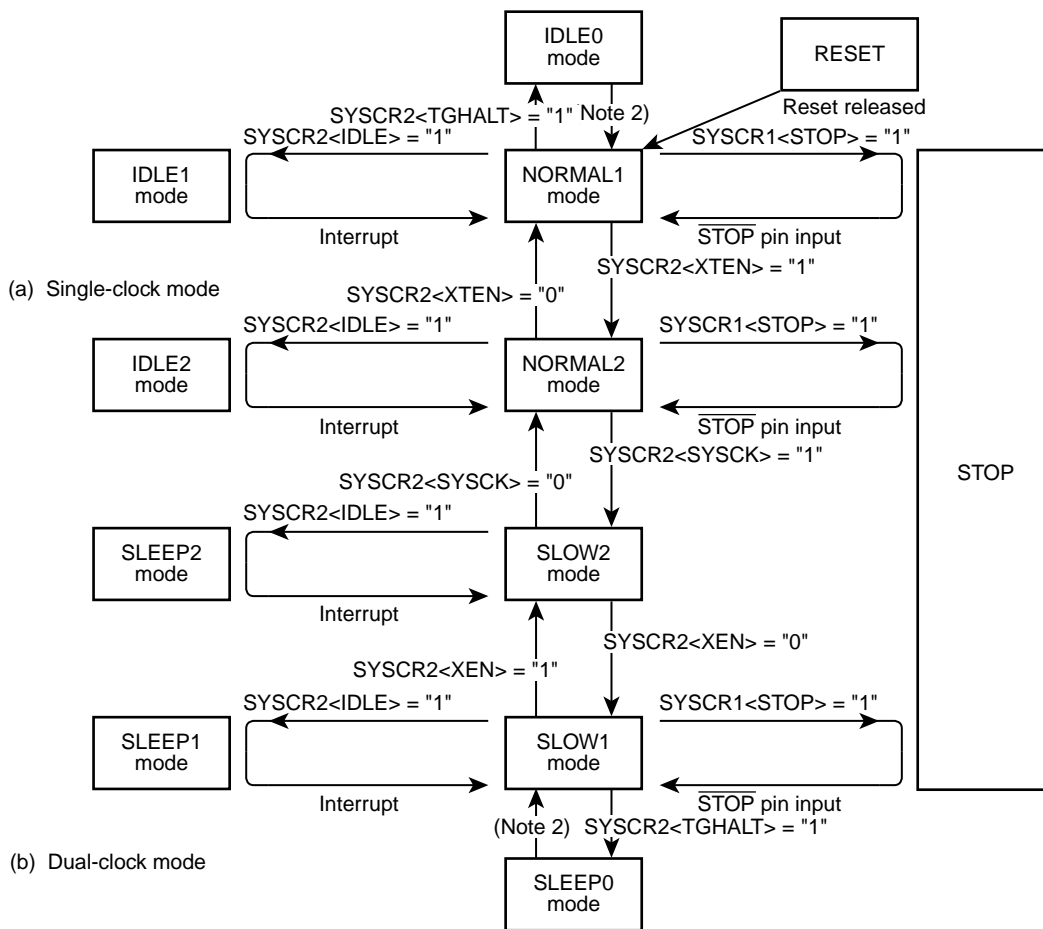
When the SLEEP0 mode is entered with TBTCR<TBTEN> = 1, INTTBT interrupt latch is set after returning to the SLOW1 mode.

2.2.3.3 STOP Mode

In the STOP mode, all system operations including oscillators are halted, and the internal conditions immediately before the halt are retained with low-power dissipation.

The STOP mode is entered by setting the system control register 1, and exited with the $\overline{\text{STOP}}$ pin input. After the warm-up period time has expired, the CPU returns to the mode it was before entering the STOP mode, and program execution resumes with the instruction immediately following the instruction that activated the STOP mode.

2.2.3.4 Operation Mode Transition



Note 1: NORMAL1 and NORMAL2 modes are generically called NORMAL mode: SLOW1 and SLOW2 are called SLOW mode: IDLE0 and IDLE1 and IDLE2 are called IDLE mode: SLEEP0, SLEEP1 and SLEEP2 are called SLEEP mode.

Note 2: This mode is exited at the falling edge of the source clock selected in TBTCR<TBTCCK>.

Figure 2-6 Operating Mode Transition

Table 2-4 Operating Mode and Conditions

Operating Mode		Oscillator		CPU Core	TBT	Other Peripherals	Machine Cycle Time
		High-freq.	Low-freq.				
Single-Clock	RESET	Oscillation	Stop	Reset	Reset	Reset	4/fc [s]
	NORMAL1			Operate	Operate	Operate	
	IDLE1			Halt		Halt	
	IDLE0						
	STOP	Stop	Halt	-			
Dual-Clock	NORMAL2	Oscillation	Oscillation	Operate with High-freq.	Operate	Operate	4/fs [s]
	IDLE2			Halt			
	SLOW2			Operate with Low-freq.			
	SLEEP2			Halt			
	SLOW1	Stop	Stop	Operate with Low-freq.	Operate	Halt	-
	SLEEP1			Halt			
	SLEEP0						
	STOP			Stop			

2.2.4 Operating Mode Control

System Control Register 1

	7	6	5	4	3	2	1	0	
SYSCR1 (0038H)	STOP	RELM	RETM	OUTEN	WUT	"0"			(Initial value: 0000 00**)

STOP	STOP mode enter	0: CPU core and peripherals operate 1: CPU core and peripherals halt (Enter STOP mode)			R/W	
RELM	STOP mode exit method	0: Edge-sensitive (Exit at the rising edge of $\overline{\text{STOP}}$ pin) 1: Level-sensitive (Exit at the high level of $\overline{\text{STOP}}$ pin)			R/W	
RETM	Operating mode after STOP mode	0: Return to NORMAL 1/2 mode 1: Return to SLOW1 mode			R/W	
OUTEN	Port output during STOP mode	0: High impedance 1: Output retained			R/W	
WUT	Warm-up time on exiting STOP mode [ns]		Return to NORMAL 1/2 mode		Return to SLOW1 mode	R/W
			DV1CK=0	DV1CK=1		
		00	$3 \times 2^{16}/f_c$	$3 \times 2^{17}/f_c$	$3 \times 2^{13}/f_s$	
		01	$2^{16}/f_c$	$2^{17}/f_c$	$2^{13}/f_s$	
10	$3 \times 2^{14}/f_c$	$3 \times 2^{15}/f_c$	$3 \times 2^6/f_s$			
11	$2^{14}/f_c$	$2^{15}/f_c$	$2^6/f_s$			

Note 1: To transit from the NOMAL mode to the STOP mode, set RETM to 0. To transit from the STOP mode to the NOMAL mode, set RETM to 1.

Note 2: When exiting the STOP mode with the RESET pin input, the CPU returns to the NORMAL1 mode regardless of the RETM value.

Note 3: fc: High-frequency clock [Hz], fs: Low-frequency clock [Hz], *: Don't care

Note 4: Bit 1 and 0 in SYSCR1 are read as don't cares.

Note 5: When entering the STOP mode with OUTEN = 0, input value is fixed to 0. That may cause an external interrupt request to be set on falling edge.

Note 6: To use the Key on wake-up is used, set RELM to 1.

Note 7: The P20 pin is shared with the STOP pin. When the STOP mode is entered, output assumes the high-impedance state regardless of the OUTEN state.

Note 8: Select the warm-up period time depending on the feature of the resonator to be used.

System Control Register 2

	7	6	5	4	3	2	1	0	
SYSCR2 (0039H)	XEN	XTEN	SYSCCK	IDLE		TGHAL T			(Initial value: 1000 *0**)

XEN	High-frequency oscillator control	0: Stop oscillation 1: Continue or start oscillation	R/W
XTEN	Low-frequency oscillator control	0: Stop oscillation 1: Continue or start oscillation	
SYSCCK	System clock select (write)/monitor (read)	0: High-frequency clock (NORMAL1/NORMAL2/IDLE1/IDLE2) 1: Low-frequency clock (SLOW/SLEEP)	
IDLE	CPU and WDT control (IDLE1/2, SLEEP1/2 mode)	0: CPU, WDT enabled 1: CPU, WDT disabled (Enter IDLE1/2, SLEEP1/2 mode)	R/W
TGHALT	TG control (IDLE0, SLEEP0 mode)	0: Clocking operation to all peripherals from TG 1: Stop the clocking operation to peripherals except TBT from TG (Enter IDLE0, SLEEP0 mode)	R/W

Note 1: Reset is performed when both XEN and XTEN are cleared to 0, XEN is cleared to 0 with SYSCCK = 0, or XTEN is cleared to 0 with SYSCCK = 1.

Note 2: WDT: watchdog timer, TG: timing generator, *: Don't care

Note 3: When the bit 3, 1 or 0 of SYSCR2 is read, a don't care is read.

Note 4: Do not set IDLE and TGHALT to 1 simultaneously.

Note 5: Since the IDLE0/SLEEP0 mode is returned to the NORMAL1/SLOW1 mode by the asynchronous internal source clock specified in TBTCR<TBTCK>, the time to return to the NORMAL1/SLOW1 mode from the IDLE0/SLEEP0 mode is shorter than the period time specified in TBTCR<TBTCK>.

Note 6: Upon exit from the IDLE1/2 or SLEEP1/2 mode, IDLE is automatically cleared to 0.

Note 7: Upon exit from the IDLE0 or SLEEP0 mode, TGHALT is automatically cleared to 0.

Note 8: When setting TGHAL to 1, stop functions of on-chip peripherals beforehand. If not stopped, an interrupt latch to the peripherals may be set immediately after the IDLE0 or SLEEP0 mode is exited.

2.2.4.1 STOP Mode

The STOP mode is controlled by the system control register 1 (SYSCR1), $\overline{\text{STOP}}$ pin input and STOP5 to STOP2. The $\overline{\text{STOP}}$ pin is used as the P20 port and $\overline{\text{INT5}}$ pin (external interrupt input 5). The STOP mode is entered by setting SYSCR1<STOP> to 1, and the following status is held in the STOP mode.

1. Both high-frequency and low-frequency oscillations are stopped, and all internal behaviors are stopped.
2. The data memory, registers, and program status words and port output latches hold the status before the STOP mode is entered.
3. The prescaler and divider of the timing generator are cleared to 0.
4. The program counter holds the address of the instruction after next to the instruction (e.g., [SET(SYSCR1).7]) by which the STOP mode is entered.

The STOP mode contains the level-sensitive and edge-sensitive exit modes which can be selected in SYSCR1<RELM>. In the case of the edge-sensitive exit mode, STOP5 to STOP2 must be disabled.

Note 1: Unlike the key-on wake-up input pin, the $\overline{\text{STOP}}$ pin does not have the function to disable input. To use the STOP mode, the $\overline{\text{STOP}}$ pin must be used to exit the STOP mode.

Note 2: During STOP period (from the start of the STOP mode to the end of warm-up period time), interrupt latches are set to 1 due to external interrupt signal changes, and interrupts may be accepted immediately after the STOP mode is exited. Therefore, disable interrupts before entering the STOP mode. Before enabling interrupts after the STOP mode is exited, clear unnecessary interrupt latches beforehand.

(1) Level-sensitive exit mode (RELM = 1)

In this mode, the STOP mode is exited by setting the $\overline{\text{STOP}}$ pin to high or STOP5 to STOP2 (can be specified to each bit in STOPPCR) to low. This mode is used for capacitor back-up when the main power supply is cut off and long term battery back-up.

When the $\overline{\text{STOP}}$ pin input is set to high or STOP5 to STOP2 is set to low, executing an instruction to enter the STOP mode does not enter the STOP mode, but immediately starts the exit sequence (warm-up). When the STOP mode is entered in the level-sensitive exit mode, it is required to check that the $\overline{\text{STOP}}$ pin input is programmed to low and the STOP5 to STOP2 pin input is programmed to high by the following methods.

1. Testing the port condition.
2. Using the INT5 interrupt (an interrupt is generated at the falling edge of the $\overline{\text{INT5}}$ pin input)

Example 1 :Entering the STOP mode from the NORMAL mode by testing a port P20

	LD	(SYSCR1), 01010000B	: Sets the level-sensitive exit mode.
SSTOPH:	TEST	(P2PRD) . 0	: Wait state until the $\overline{\text{STOP}}$ pin input becomes low.
	JRS	F, SSTOPH	
	DI		: IMF'0
	SET	(SYSCR1) . 7	: Enters the STOP mode.

Example 2 :Entering the STOP mode from the NORMAL mode by the INT5 interrupt

```

PINT5:      TEST      (P2PRD) . 0           : To eliminate spurious noise, the STOP mode is not entered if the
                                                P20 port input is set to high.
            JRS       F, SINT5             : Sets the level-sensitive exit mode.
            LD        (SYSCR1), 01010000B
            DI        : IMF'0
            SET      (SYSCR1) . 7         : Enters the STOP mode.
SINT5:      RETI
    
```

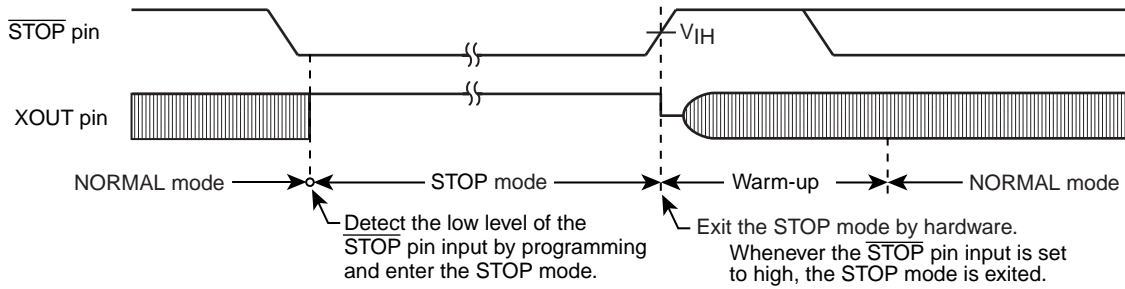


Figure 2-7 Level-Sensitive Exit Mode

Note 1: After a warm-up period starts, the STOP mode is not reentered if the $\overline{\text{STOP}}$ pin input becomes low or STOP5 to STOP2 becomes high again.

Note 2: To return to the level-sensitive exit mode after setting up the edge-sensitive exit mode, the exit mode is not switched until the rising edge of the $\overline{\text{STOP}}$ pin input is detected.

(2) Edge-sensitive exit mode (RELM = 0)

In this mode, the STOP mode is exited at the rising edge of the $\overline{\text{STOP}}$ pin input. This mode is used in applications where a relatively short program is run repeatedly at periodic intervals. This periodic signal (i.e., a clock from a low-power consumption oscillator) is input input to the $\overline{\text{STOP}}$ pin. In the edge-sensitive exit mode, the STOP mode is entered even if the $\overline{\text{STOP}}$ pin input is high. Disable the STOP5 to STOP2 pin input with the key-on wake-up control register (STOPPCR).

Example :Entering the STOP mode from the NORMAL mode

```

DI        : IMF'0
LD        (SYSCR1), 10010000B : Sets the edge-sensitive exit mode to enter the STOP mode
    
```

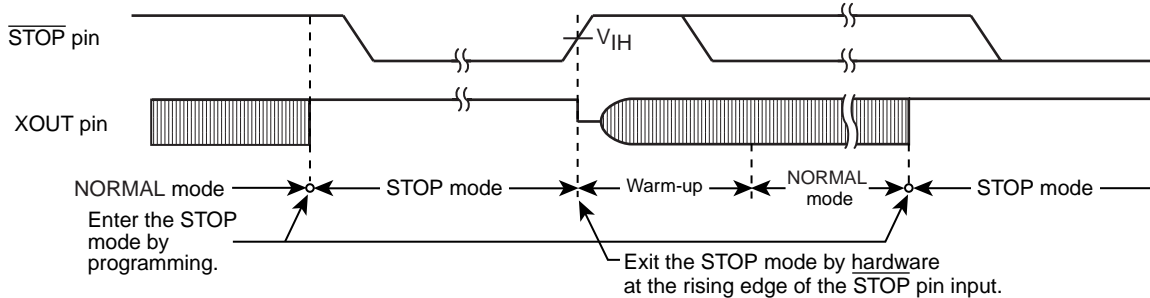


Figure 2-8 Edge-Sensitive Exit Mode

The STOP mode is exited in the edge-sensitive exit mode by the following sequence.

1. Oscillations start. In the dual-clock mode, both high-frequency and low-frequency oscillators start to return to the NORMAL2 mode, and only the low-frequency oscillator starts to return to the SLOW mode. In the single-clock mode, only the high-frequency oscillator starts.
2. The warm-up period time is inserted to allow sufficient time for the oscillator to stabilize. During warm-up, internal operations remain halted. 4 types of warming-up period time can be selected in SYSCR1<WUT> depending on the characteristics of the oscillator.
3. After the warm-up period time, program execution resumes with the instruction immediately following the instruction that activated the STOP mode.

Note 1: When the STOP mode is exited, the prescaler and divider of the timing generator are cleared to 0.

Note 2: The STOP mode is exited by setting the $\overline{\text{RESET}}$ pin to low, that immediately performs the normal reset operation.

Note 3: To exit the STOP mode with a low hold voltage, the following cautions must be observed. The power supply voltage must be at the operating voltage level before exiting the STOP mode. The $\overline{\text{RESET}}$ pin must also be high, rising together with the power supply voltage. In this case, if an external time constant circuit is connected, the $\overline{\text{RESET}}$ pin input voltage increases at a slower pace than the power supply voltage. At this time, there is a danger that a reset may occur if the input voltage level of the RESET pin drops below the non-inverting high-level input voltage (hysteresis input).

Table 2-5 Warm-up Time (fc = 16.0 MHz, fs = 32.768 kHz)

WUT	Warm-up time [ms]		
	Return to the NORMAL mode		Return to the SLOW mode
	DV1CK=0	DV1CK=1	
00	12.288	24.576	750
01	4.096	8.192	250
10	3.072	6.144	5.85
11	1.024	2.048	1.95

Note 1: Since the warm-up period time is obtained by dividing the basic clock by the divider, any frequency fluctuations will lead to small warm-up period time error. The warm-up period time should be considered as an approximate value.

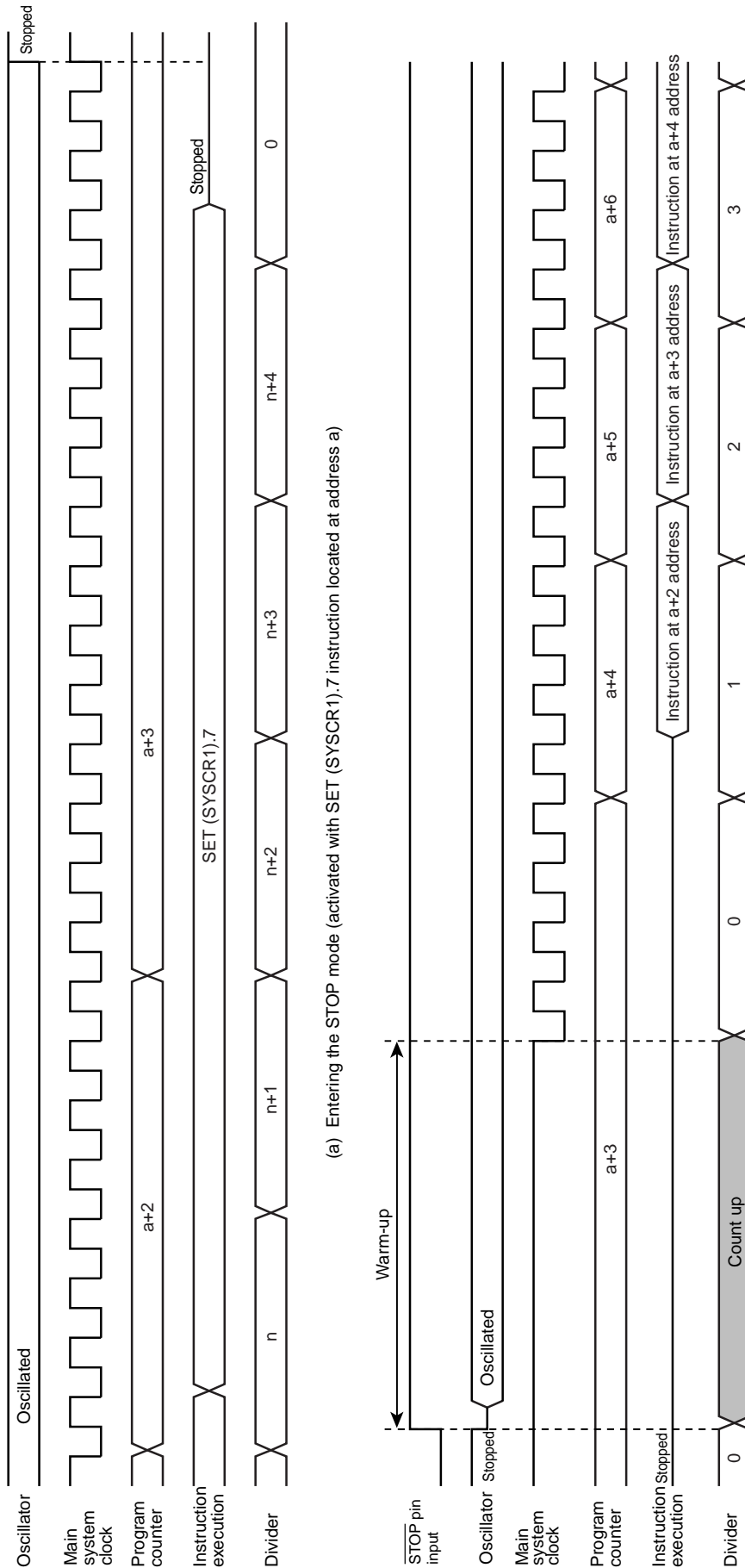


Figure 2-9 Entering and Exiting the STOP Mode

2.2.4.2 IDLE1/2 and SLEEP1/2 Modes

The IDLE1/2 and SLEEP1/2 modes controlled by the system control register 2 (SYSCR2) and maskable interrupts. The following status is held during the IDLE1/2 or SLEEP1/2 mode.

1. The CPU and watchdog timer are halted. On-chip peripherals continue operation.
2. The data memory, registers, program status words, port output latches hold the status that activated the IDLE1/2 or SLEEP1/2 mode.
3. The program counter holds the address of the instruction after next to the instruction to activate IDLE1/2 or SLEEP1/2 mode.

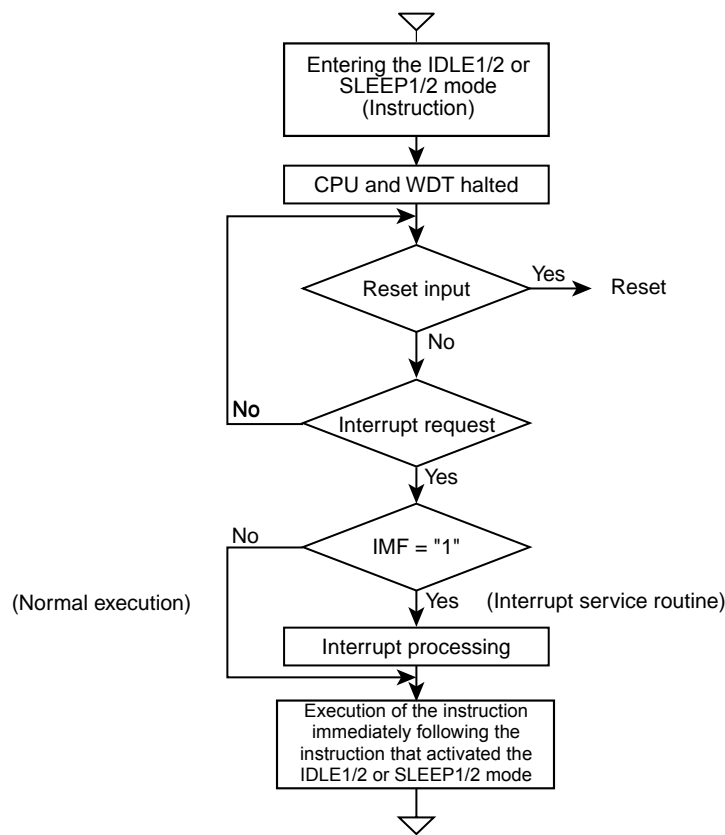


Figure 2-10 IDLE1/2 and SLEEP1/2 Modes

- Entering IDLE1/2 or SLEEP1/2 mode

After clearing the interrupt master enable flag (IMF) to 0, set the individual interrupt enable flag used to exit the IDLE1/2 or SLEEP1/2 mode to 1.

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

- Exiting IDLE1/2 or SLEEP1/2 Mode

Upon return from the IDLE1/2 or SLEEP1/2 mode, the interrupt master enable flag (IMF) determines the action taken after exiting the IDLE1/2 or SLEEP1/2 mode; i.e., whether execution resumes with an interrupt service routine. When exiting the IDLE1/2 or SLEEP1/2 mode, SYSCR2<IDLE> is automatically cleared to 0, and the operating mode is returned to the mode before entering the IDLE1/2 or SLEEP1/2 mode.

The IDLE1/2 or SLEEP1/2 mode is exited by setting the $\overline{\text{RESET}}$ pin to low. In this case, the NORMAL1 mode is activated after exiting the IDLE1/2 or SLEEP1/2.

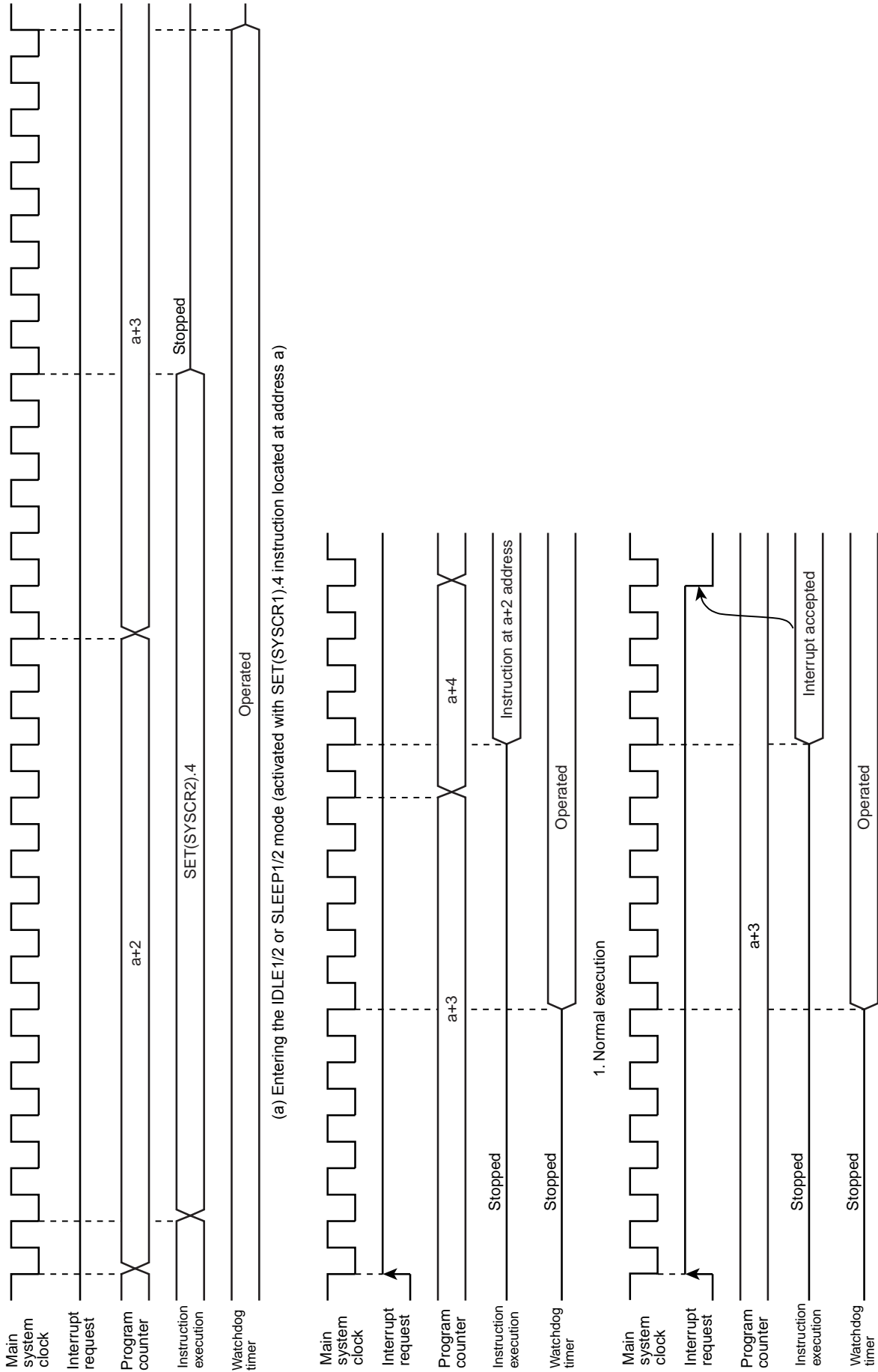
- (1) Program execution resuming with the instruction (IMF = 0)

The IDLE1/2 or SLEEP1/2 mode is exited by the individual interrupt enable flag (EF). Program execution resumes with the instruction immediately following the instruction that activated the IDLE1/2 or SLEEP1/2 mode. Normally the instruction latches (IL) of the interrupt source used to exit the IDLE1/2 or SLEEP1/2 mode must be cleared to 0 by the load instruction.

- (2) Program execution resuming with the interrupt service routine (IMF = 1)

The IDLE1/2 or SLEEP1/2 mode is exited by an interrupt source enabled by the individual interrupt enable flag (EF). Execution resumes with the interrupt service routine. Upon completion of the interrupt service routine, program execution resumes with the instruction immediately following the instruction that activated the IDLE1/2 or SLEEP1/2 mode.

Note: When a watchdog timer interrupt is generated immediately before entering the IDLE1/2 or SLEEP1/2 mode, the watchdog timer interrupt is processed without entering the IDLE1/2 or SLEEP1/2 mode.



(a) Entering the IDLE1/2 or SLEEP1/2 mode (activated with SET(SYSCR1),4 instruction located at address a)

1. Normal execution

2. Interrupt service routine (b) Exiting the IDLE1/2 or SLEEP1/2 mode

Figure 2-11 Entering and Exiting the IDLE1/2 or SLEEP1/2 Mode

2.2.4.3 IDLE0 and SLEEP0 Modes

The IDLE0 mode is controlled by the system control register 2 (SYSCR2) and time base timer. The following status is held during the IDLE0 mode.

- The timing generator stops the clock distribution to the on-chip peripherals except the time base timer.
- The data memory, registers, program status words and port output latches hold the status that activated the IDLE0 or SLEEP0 mode.
- The program counter holds the address of the instruction after next to the instruction to activate the IDLE0 or SLEEP0 mode.

Note: Before entering the IDLE0 or SLEEP0 mode, the on-chip peripherals must be disabled.

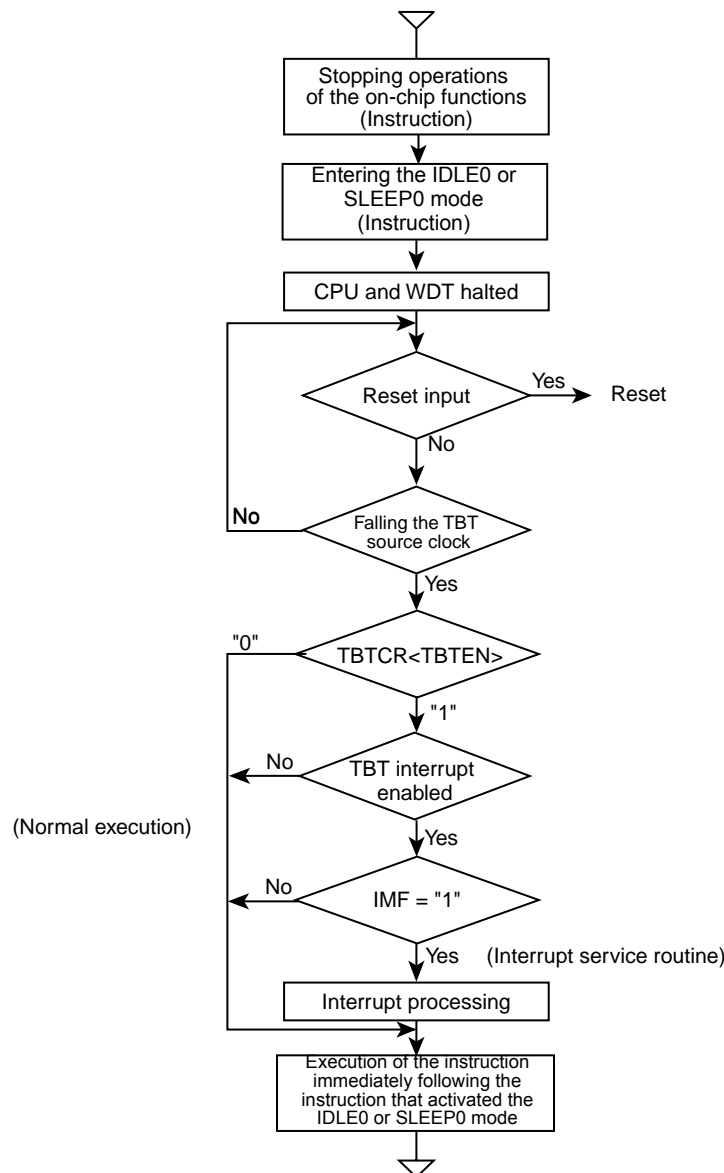


Figure 2-12 IDLE0 or SLEEP0 Mode

- Entering IDLE0 or SLEEP0 mode

Disable on-chip peripherals such as a timer counter. To enter the IDLE0 or SLEEP0 mode, set SYSCR2<TGHALT> to 1.

- Exiting IDLE0 or SLEEP0 Mode

Upon return from the IDLE0 or SLEEP0 mode, the interrupt master enable flag (IMF) determines the action taken after exiting the IDLE0 or SLEEP0 mode; i.e., whether execution resumes with an interrupt service routine. When exiting the IDLE0 or SLEEP0 mode, SYSCR2<TGHALT> is automatically cleared to 0, and the operating mode is returned to the mode before entering the IDLE0 or SLEEP0 mode. When TBTCR<TBTEN> is set to 1 at this time, the INTTBT interrupt latch is set.

The IDLE0 or SLEEP0 mode is exited by setting the $\overline{\text{RESET}}$ pin to low. In this case, the NORMAL1 mode is activated after exiting the IDLE1/2 or SLEEP1/2.

Note: The IDLE0 or SLEEP0 mode is entered and exited regardless of TBTCR<TBTEN> setting.

- (1) Program execution resuming with the instruction (IMF, EF8, TBTCR<TBTEN> = 0)

When detecting the falling edge of the source clock set in TBTCR<TBTK>, the IDLE0 or SLEEP0 mode is exited. When the IDLE0 or SLEEP0 mode is exited, program execution resumes with the instruction immediately following the instruction that activated the IDLE0 or SLEEP0 mode. When TBTCR<TBTEN> is set to 1, the time base timer interrupt latch is set.

- (2) Program execution resuming with the interrupt service routine (IMF, EF8, TBTCR<TBTEN> = 1)

When detecting the falling edge of the source clock set in TBTCR<TBTK>, the IDLE0 or SLEEP0 mode is exited, and then INTTBT interrupt processing is performed.

Note 1: The IDLE0 or SLEEP0 mode is returned to the NORMAL1 or SLEEP1 mode by the asynchronous internal clock specified in TBTCR<TBTK>, the period time of IDLE0 or SLEEP0 mode is shorter than the period set in TBTCR<TBTK>.

Note 2: When a watchdog timer interrupt is generated immediately before entering the IDLE1/2 or SLEEP1/2 mode, the watchdog timer interrupt is processed, without entering the IDLE1/2 or SLEEP1/2 mode.

Note 3: When IL8ER in interrupt source selector (INTSEL) is set to "1", the program execution resumes with the instruction immediately following the instruction that activated the IDLE0 or SLEEP0 mode even though all of IMF, EF8 and TBTCR<TBTEN> are set to "1".

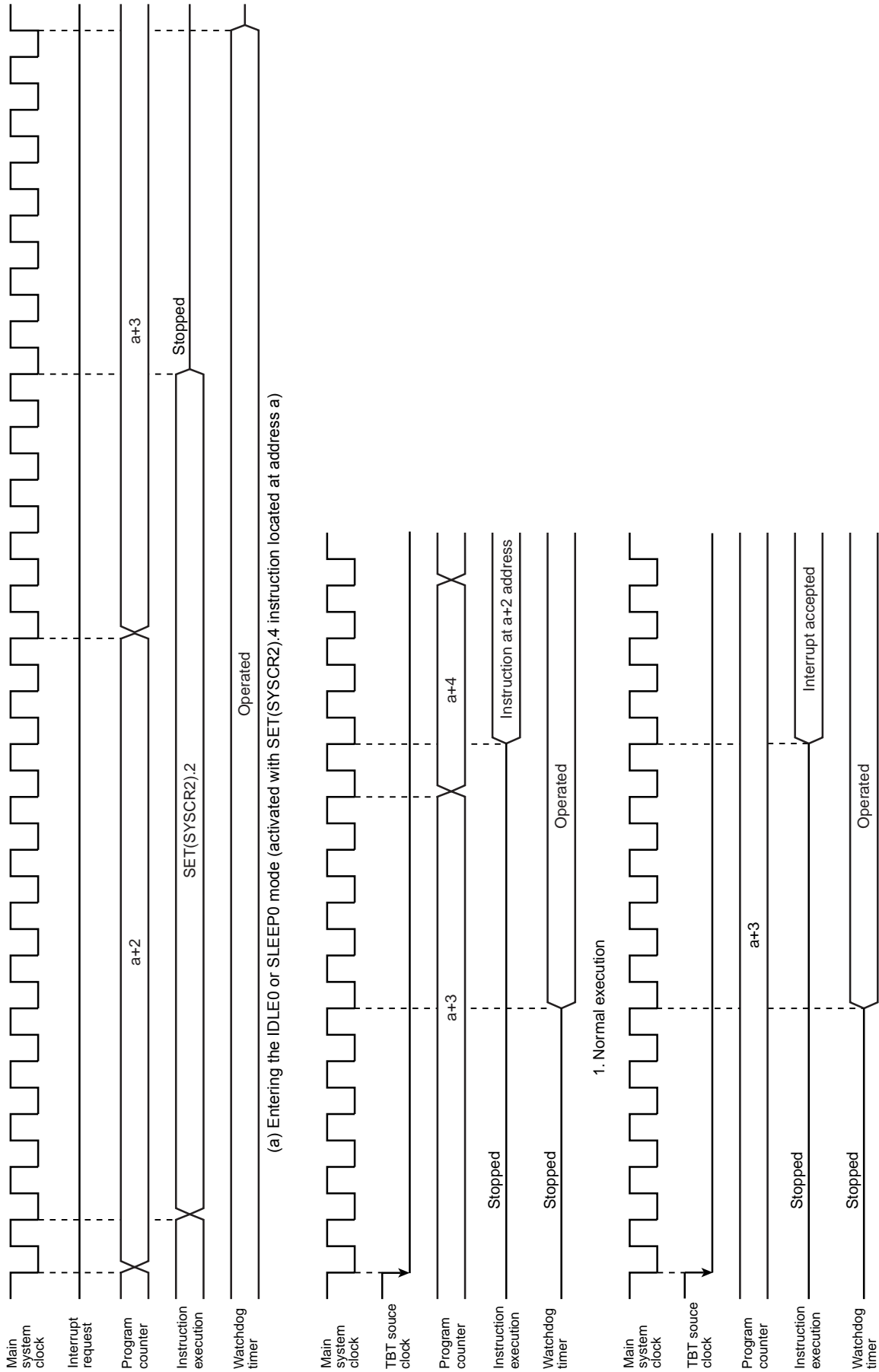


Figure 2-13 Entering and Exiting the IDLE0 or SLEEP0 Mode

2.2.4.4 SLOW Mode

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

(1) Switching the NORMAL2 mode to SLOW mode

Write 1 to SYSCR2<SYSCK> to switch the main system clock to the low-frequency clock. Clear SYSCR2<XEN> to 0 to stop the high-frequency oscillator.

Note: The high-frequency clock oscillation can be continued to return quickly to the NORMAL2 mode. To enter the STOP mode from the SLOW mode, the high-frequency clock must be stopped.

When the low-frequency clock oscillation is unstable, wait until the oscillation is stabilized before performing the above operation. The TimerCounter (TC2) is convenient to check the low-frequency clock oscillation stability.)

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

	SET	(SYSCR2). 5	: SYSCR2<SYSCK>=1 : (switches the system clock to the low-frequency clock for the SLOW2 mode.)
	CLR	(SYSCR2). 7	: SYSCR2<XEN>=0(stops the high-frequency oscillation.)

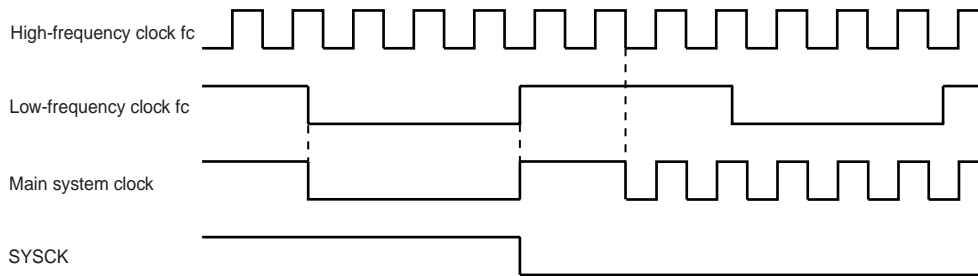
Example 2 :Switching to the SLOW1 mode after checking the low-frequency clock oscillation stability with TC2

	SET	(SYSCR2). 6	: SYSCR2<XTEN>=1 : (starts low-frequency oscillation.)
	LD	(TC2CR), 14H	: sets the mode for TC2.
	LDW	(TC2DRL), 8000H	:sets the warm-up time. : (determines the time depending on the resonator.)
	DI		: IMF=0
	SET	(EIRH), 4	: enables the INTTC2.
	EI		: IMF=1
	SET	(TC2CR). 5	: starts INTTC2.
PINTTC2:	CLR	(TC2CR). 5	: stops INTTC2.
	SET	(SYSCR2). 5	: SYSCR2<SYSCK>=1 : (switches the system clock to the low-frequency clock.)
	CLR	(SYSCR2). 7	: SYSCR2<XEN>=0(stops the high-frequency clock.)
	RETI		
VINTTC2:	DW	PINTTC2	: INTTC2 vector table

(2) Switching from the SLOW1 mode to NORMAL2 mode

First, set SYSCR2<XEN> to 1 to oscillate the high-frequency clock. After the warm-up period time required to assure oscillation stability with the TimerCounter (TC2) has elapsed, clear SYSCR2<SYSCK> to 0 to switch the system clock to the high-frequency clock. The SLOW mode is also exited by setting the $\overline{\text{RESET}}$ pin to low, which immediately performs normal reset operation. The NORMAL1 mode is entered after a reset release.

Note: After SYSCK is cleared to 0, instructions are executed continuously by the low-frequency clock during synchronization period for high-frequency and low-frequency clocks.



Example :Switching from SLOW1 mode to NORMAL2 mode with TC2
($f_c = 16 \text{ MHz}$, warm-up time = 4.0 ms)

	SET	(SYSCR2). 7	: SYSCR2<XEN>=1 :(starts high-frequency oscillation.)
	LD	(TC2CR), 10H	: sets the TC2 mode.
	LD	(TC2DRH), 0F8H	: sets the warm-up time. :(determines the time depending on the frequency and resonator.)
	DI		: IMF=0
	SET	(EIRH), 4	: enables INTTC2 interrupt.
	EI		: IMF=1
	SET	(TC2CR), 5	: starts TC2.
PINTTC2	CLR	(TC2CR), 5	: stops TC2.
	CLR	(SYSCR2). 5	: SYSCR2<SYSCK>=0 :(switches the system clock to the high-frequency clock.)
	RETI		
VINTTC2:	DW	PINTTC2	: INTTC2 vector table

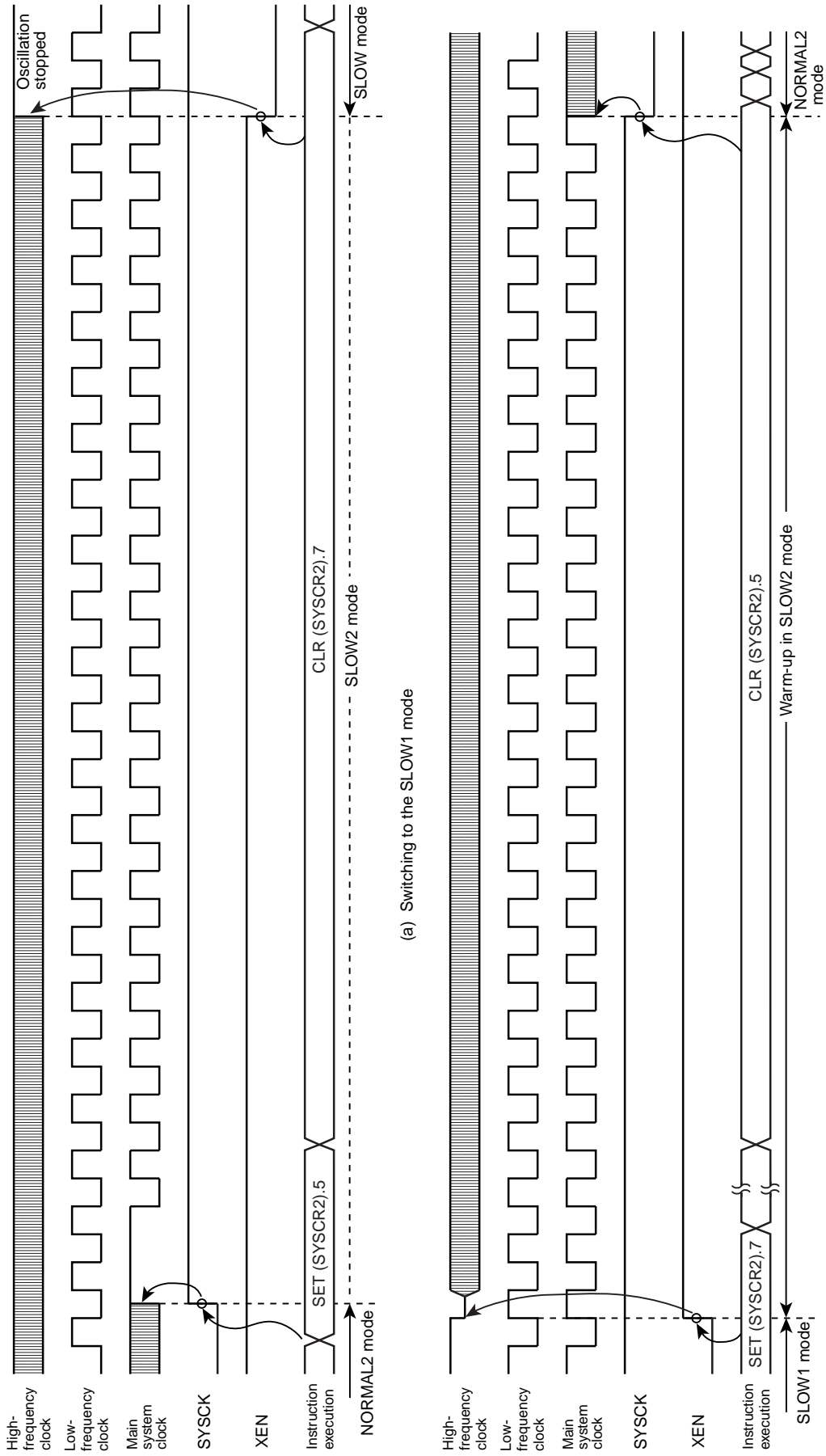


Figure 2-14 Switching between SLOW and NORMAL2 Modes

2.3 Reset Circuit

TMP86FS64FG has four types of reset, that are an external reset, address trap reset, watchdog timer reset and system clock reset.

An address trap reset, watchdog timer reset and system clock reset are internal factor resets. When detecting these reset requests, TMP86FS64FG is in the reset state during a maximum of $24/f_c$ [s].

(During a flash reset, the $\overline{\text{RESET}}$ pin is held high.)

Since the internal factor reset circuits that are watchdog timer reset, address trap reset and system clock reset are not initialized upon power-up, a maximum reset time may become $24/f_c$ [s] ($1.5 \mu\text{s}$ @ 16.0 MHz).

Table 2-6 shows the on-chip hardware initialization by reset operation.

Table 2-6 On-Chip Hardware Initialization by Reset Operation

On-Chip Hardware	Initial Value	On-Chip Hardware	Initial Value
Program counter (PC)	(FFFEH)	Prescaler and divider of the timing generator	0
Stack pointer (SP)	Not initialized		
General-purpose register (W, A, B, C, D, E, H, L, IX, IY)	Not initialized		
Jump status flag (JF)	Not initialized	Watchdog timer	Enabled
Zero flag (ZF)	Not initialized	Output latch of I/O port	Refer to description of each I/O port
Carry flag (CF)	Not initialized		
Half carry flag (HF)	Not initialized		
Sign flag (SF)	Not initialized		
Overflow flag (VF)	Not initialized		
Interrupt master enable flag (IMF)	0		
Interrupt individual enable flag (EF)	0	Control register	Refer to description of each register
Interrupt latch (IL)	0		
		RAM	Not initialized

2.3.1 External Reset Input

The $\overline{\text{RESET}}$ pin is the hysteresis input with pull-up resistance. When the $\overline{\text{RESET}}$ pin is held low for a minimum of 3 machine cycles ($12/f_c$ [s]) with the power supply voltage within the operating voltage range and stable oscillation, a reset is triggered and internal state is initialized.

When the $\overline{\text{RESET}}$ pin input goes high, the reset operation is released, and program execution starts at the vector address stored at addresses FFFE to FFFH.

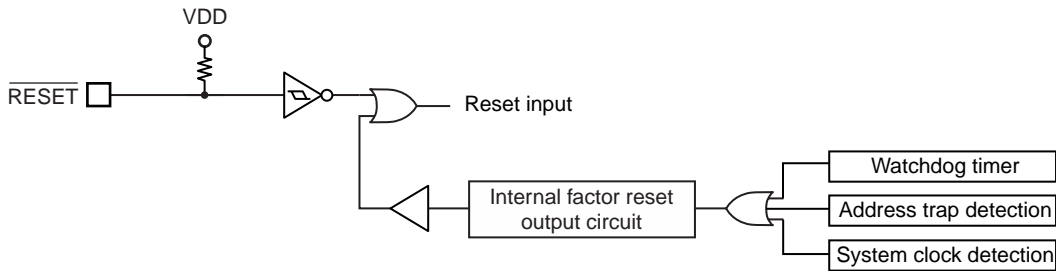


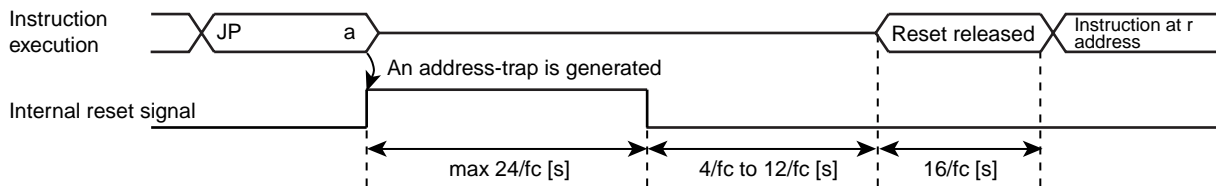
Figure 2-15 Reset Circuit

2.3.2 Address-Trap-Reset

If the CPU runs away due to spurious noises and attempts to fetch an instruction from the on-chip RAM (WDTCR1<ATA> = 1) or the SFR area, an address-trap-reset is generated. The reset time is a maximum of $24/f_c$ [s] ($1.5 \mu\text{s}$ @16.0 MHz).

If the CPU runs away due to spurious noises and attempts to fetch an instruction from the on-chip RAM (WDTCR1<ATAS> = 1), the DBR or the SFR area, an address-trap-reset is generated. The reset time is a maximum of $24/f_c$ [s] ($1.5 \mu\text{s}$ @16.0 MHz).

Note: Either a reset or an interrupt can be selected for an address-trap. An address-trap area can be specified.



Note 1: "a" is the address in on-chip RAM (WDTCR1<ATAS>=1), SFR or DBR area.

Note 2: During the reset release process, the reset vector "r" is read out, and an instruction at the address "r" is fetched and decoded.

Figure 2-16 Address Trap Reset

2.3.3 Watchdog Timer Reset

Refer to "Watchdog Timer".

2.3.4 System Clock Reset

Either one of the following conditions is met, a system clock reset is generated automatically to prevent the CPU to be in the deadlock condition. (Oscillation is continued.)

- SYSCR2<XEN> and SYSCR2<XTEN> are cleared to 0.
- SYSCR2<XEN> is cleared to 0 when SYSCR2<SYSCK> = 0.
- SYSCR2<XTEN> is cleared to 0 when SYSCR2<SYSCK> = 1.

The reset time is a maximum of $24/f_c$ [s] ($1.5 \mu\text{s}$ @16.0 MHz).

3. Interrupt Control Circuit

The TMP86FS64FG has a total of 21 interrupt sources excluding reset, of which 5 source levels are multiplexed. Interrupts can be nested with priorities. Four of the internal interrupt sources are non-maskable while the rest are maskable.

Interrupt sources are provided with interrupt latches (IL), which hold interrupt requests, and independent vectors. The interrupt latch is set to “1” by the generation of its interrupt request which requests the CPU to accept its interrupts. Interrupts are enabled or disabled by software using the interrupt master enable flag (IMF) and interrupt enable flag (EF). If more than one interrupts are generated simultaneously, interrupts are accepted in order which is dominated by hardware. However, there are no prioritized interrupt factors among non-maskable interrupts.

Interrupt Factors		Enable Condition	Interrupt Latch	Vector Address	Priority
Internal/External	(Reset)	Non-maskable	–	FFFE	1
Internal	INTSWI (Software interrupt)	Non-maskable	–	FFFC	2
Internal	INTUNDEF (Executed the undefined instruction interrupt)	Non-maskable	–	FFFC	2
Internal	INTATRAP (Address trap interrupt)	Non-maskable	IL2	FFFA	2
Internal	INTWDT (Watchdog timer interrupt)	Non-maskable	IL3	FFF8	2
External	$\overline{INT0}$	IMF• EF4 = 1, INTOEN = 1	IL4	FFF6	5
External	INT1	IMF• EF5 = 1	IL5	FFF4	6
Internal	INTTC4	IMF• EF6 = 1	IL6	FFF2	7
Internal	INTTC5	IMF• EF7 = 1	IL7	FFF0	8
Internal	INTTBT	IMF• EF8 = 1, IL8ER = 0	IL8	FFEE	9
External	INT2	IMF• EF8 = 1, IL8ER = 1			
Internal	INTTC1	IMF• EF9 = 1, IL9ER = 0	IL9	FFEC	10
External	INT3	IMF• EF9 = 1, IL9ER = 1			
Internal	INTTC3	IMF• EF10 = 1	IL10	FFEA	11
Internal	INTTC6	IMF• EF11 = 1	IL11	FFE8	12
Internal	INTTC2	IMF• EF12 = 1	IL12	FFE6	13
Internal	INTSIO1	IMF• EF13 = 1, IL13ER = 0	IL13	FFE4	14
External	INT4	IMF• EF13 = 1, IL13ER = 1			
Internal	INTTRX	IMF• EF14 = 1, IL14ER = 0	IL14	FFE2	15
External	$\overline{INT5}$	IMF• EF14 = 1, IL14ER = 1			
Internal	INTADC	IMF• EF15 = 1, IL15ER = 0	IL15	FFE0	16
Internal	INTSIO2	IMF• EF15 = 1, IL15ER = 1			

Note 1: The INTSEL register is used to select the interrupt source to be enabled for each multiplexed source level (see 3.3 Interrupt Source Selector (INTSEL)).

Note 2: To use the address trap interrupt (INTATRAP), clear WDTCR1<ATOUT> to “0” (It is set for the “reset request” after reset is cancelled). For details, see “Address Trap”.

Note 3: To use the watchdog timer interrupt (INTWDT), clear WDTCR1<WDTOUT> to “0” (It is set for the “Reset request” after reset is released). For details, see “Watchdog Timer”.

3.1 Interrupt latches (IL15 to IL2)

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

The interrupt latches are located on address 003CH and 003DH in SFR area. Each latch can be cleared to "0" individually by instruction. However, IL2 and IL3 should not be cleared to "0" by software. For clearing the interrupt latch, load instruction should be used and then IL2 and IL3 should be set to "1". If the read-modify-write instructions such as bit manipulation or operation instructions are used, interrupt request would be cleared inadequately if interrupt is requested while such instructions are executed.

Interrupt latches are not set to "1" by an instruction.

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

Note: In main program, before manipulating the interrupt enable flag (EF) or the interrupt latch (IL), be sure to clear IMF to "0" (Disable interrupt by DI instruction). Then set IMF newly again as required after operating on the EF or IL (Enable interrupt by EI instruction)

In interrupt service routine, because the IMF becomes "0" automatically, clearing IMF need not execute normally on interrupt service routine. However, if using multiple interrupt on interrupt service routine, manipulating EF or IL should be executed before setting IMF="1".

Example 1 :Clears interrupt latches

```
DI                ; IMF ← 0
LDW              (ILL), 1110100000111111B ; IL12, IL10 to IL6 ← 0
EI                ; IMF ← 1
```

Example 2 :Reads interrupt latches

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

Example 3 :Tests interrupt latches

```
TEST           (ILL), 7 ; if IL7 = 1 then jump
JR            F, SSET
```

3.2 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, address trap interrupt and watchdog interrupt). Non-maskable interrupt is accepted regardless of the contents of the EIR.

The EIR consists of an interrupt master enable flag (IMF) and the individual interrupt enable flags (EF). These registers are located on address 003AH and 003BH in SFR area, and they can be read and written by an instructions (Including read-modify-write instructions such as bit manipulation or operation instructions).

3.2.1 Interrupt master enable flag (IMF)

The interrupt enable register (IMF) enables and disables the acceptance of the whole maskable interrupt. While IMF = "0", all maskable interrupts are not accepted regardless of the status on each individual interrupt enable flag (EF). By setting IMF to "1", the interrupt becomes acceptable if the individuals are enabled. When an interrupt is accepted, IMF is cleared to "0" after the latest status on IMF is stacked. Thus the maskable interrupts which follow are disabled. By executing return interrupt instruction [RETI/RETN], the stacked data, which was the status before interrupt acceptance, is loaded on IMF again.

The IMF is located on bit0 in EIRL (Address: 003AH in SFR), and can be read and written by an instruction. The IMF is normally set and cleared by [EI] and [DI] instruction respectively. During reset, the IMF is initialized to "0".

3.2.2 Individual interrupt enable flags (EF15 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 (EF15 to EF4) are initialized to “0” and all maskable interrupts are not accepted until they are set to “1”.

Note: In main program, before manipulating the interrupt enable flag (EF) or the interrupt latch (IL), be sure to clear IMF to "0" (Disable interrupt by DI instruction). Then set IMF newly again as required after operating on the EF or IL (Enable interrupt by EI instruction)
 In interrupt service routine, because the IMF becomes "0" automatically, clearing IMF need not execute normally on interrupt service routine. However, if using multiple interrupt on interrupt service routine, manipulating EF or IL should be executed before setting IMF="1".

Example 1 :Enables interrupts individually and sets IMF

```

DI                                     ; IMF ← 0

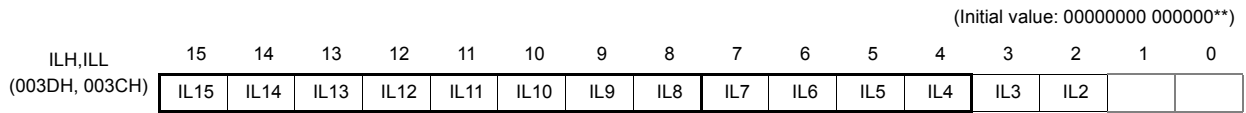
LDW      (EIRL), 1110100010100000B    ; EF15 to EF13, EF11, EF7, EF5 ← 1
:                                         Note: IMF should not be set.
:
EI                                     ; IMF ← 1
    
```

Example 2 :C compiler description example

```

unsigned int _io (3AH) EIRL;           /* 3AH shows EIRL address */
_DI();
EIRL = 10100000B;
:
_EI();
    
```

Interrupt Latches



ILH (003DH)

ILL (003CH)

IL15 to IL2	Interrupt latches	at RD 0: No interrupt request 1: Interrupt request	at WR 0: Clears the interrupt request 1: (Interrupt latch is not set.)	R/W
-------------	-------------------	--	--	-----

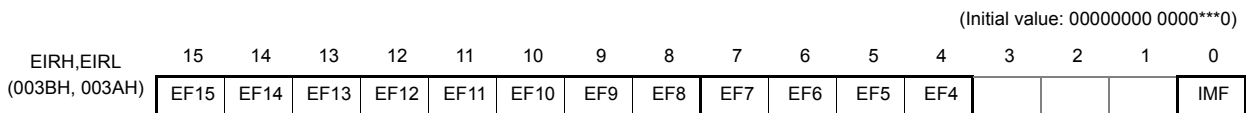
Note 1: To clear any one of bits IL7 to IL4, be sure to write "1" into IL2 and IL3.

Note 2: In main program, before manipulating the interrupt enable flag (EF) or the interrupt latch (IL), be sure to clear IMF to "0" (Disable interrupt by DI instruction). Then set IMF newly again as required after operating on the EF or IL (Enable interrupt by EI instruction)

In interrupt service routine, because the IMF becomes "0" automatically, clearing IMF need not execute normally on interrupt service routine. However, if using multiple interrupt on interrupt service routine, manipulating EF or IL should be executed before setting IMF="1".

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

Interrupt Enable Registers



EIRH (003BH)

EIRL (003AH)

EF15 to EF4	Individual-interrupt enable flag (Specified for each bit)	0: Disables the acceptance of each maskable interrupt. 1: Enables the acceptance of each maskable interrupt.	R/W
IMF	Interrupt master enable flag	0: Disables the acceptance of all maskable interrupts 1: Enables the acceptance of all maskable interrupts	

Note 1: *: Don't care

Note 2: Do not set IMF and the interrupt enable flag (EF15 to EF4) to "1" at the same time.

Note 3: In main program, before manipulating the interrupt enable flag (EF) or the interrupt latch (IL), be sure to clear IMF to "0" (Disable interrupt by DI instruction). Then set IMF newly again as required after operating on the EF or IL (Enable interrupt by EI instruction)

In interrupt service routine, because the IMF becomes "0" automatically, clearing IMF need not execute normally on interrupt service routine. However, if using multiple interrupt on interrupt service routine, manipulating EF or IL should be executed before setting IMF="1".

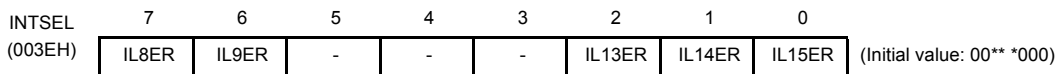
3.3 Interrupt Source Selector (INTSEL)

Each interrupt source that shares the interrupt source level with another interrupt source is allowed to enable the interrupt latch only when it is selected in the INTSEL register. The interrupt controller does not hold interrupt requests corresponding to interrupt sources that are not selected in the INTSEL register. Therefore, the INTSEL register must be set appropriately before interrupt requests are generated.

The following interrupt sources share their interrupt source level; the source is selected on the register INTSEL.

1. INTTBT and INT2 share the interrupt source level whose priority is 9.
2. INTTC1 and INT3 share the interrupt source level whose priority is 10.
3. INTSIO1 and INT4 share the interrupt source level whose priority is 14.
4. INTTRX and $\overline{\text{INT5}}$ share the interrupt source level whose priority is 15.
5. INTADC and INTSIO2 share the interrupt source level whose priority is 16.

Interrupt source selector



IL8ER	Selects INTTBT or INT2	0: INTTBT 1: INT2	R/W
IL9ER	Selects INTTC1 or INT3	0: INTTC1 1: INT3	R/W
IL13ER	Selects INTSIO1 or INT4	0: INTSIO1 1: INT4	R/W
IL14ER	Selects INTTRX or $\overline{\text{INT5}}$	0: INTTRX 1: $\overline{\text{INT5}}$	R/W
IL15ER	Selects INTADC or INTSIO2	0: INTADC 1: INTSIO2	R/W

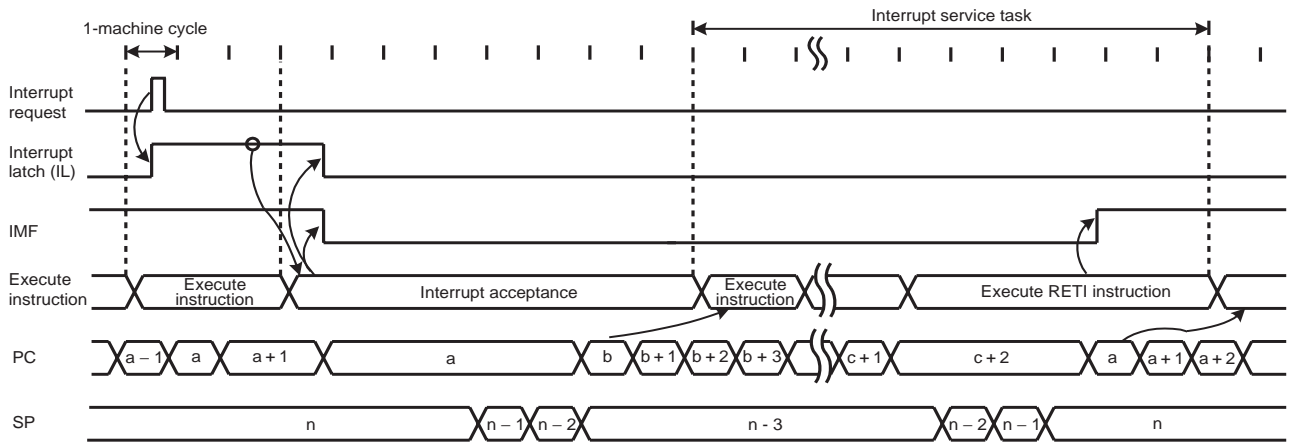
3.4 Interrupt Sequence

An interrupt request, which raised interrupt latch, is held, until interrupt is accepted or interrupt latch is cleared to “0” by resetting or an instruction. Interrupt acceptance sequence requires 8 machine cycles (2 μs @16 MHz) 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). Figure 3-1 shows the timing chart of interrupt acceptance processing.

3.4.1 Interrupt acceptance processing is packaged as follows.

- a. The interrupt master enable flag (IMF) is cleared to “0” in order to disable the acceptance of any following interrupt.
- b. The interrupt latch (IL) for the interrupt source accepted is cleared to “0”.
- c. The contents of the program counter (PC) and the program status word, including the interrupt master enable flag (IMF), are saved (Pushed) on the stack in sequence of PSW + IMF, PCH, PCL. Meanwhile, the stack pointer (SP) is decremented by 3.
- d. The entry address (Interrupt vector) of the corresponding interrupt service program, loaded on the vector table, is transferred to the program counter.
- e. The instruction stored at the entry address of the interrupt service program is executed.

Note: When the contents of PSW are saved on the stack, the contents of IMF are also saved.



Note 1: a: Return address entry address, b: Entry address, c: Address which RETI instruction is stored
 Note 2: On condition that interrupt is enabled, it takes $38/f_c$ [s] or $38/f_s$ [s] at maximum (If the interrupt latch is set at the first machine cycle on 10 cycle instruction) to start interrupt acceptance processing since its interrupt latch is set.

Figure 3-1 Timing Chart of Interrupt Acceptance/Return Interrupt Instruction

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

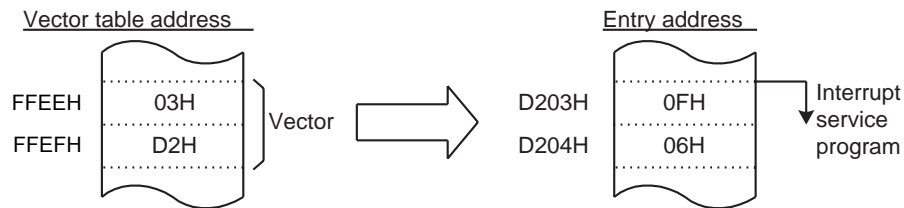


Figure 3-2 Vector table address,Entry address

A maskable interrupt is not accepted until the IMF is set to “1” even if the maskable interrupt higher than the level of current servicing interrupt is requested.

In order to utilize nested interrupt service, the IMF is set to “1” in the interrupt service program. In this case, acceptable interrupt sources are selectively enabled by the individual interrupt enable flags.

To avoid overloaded nesting, clear the individual interrupt enable flag whose interrupt is currently serviced, before setting IMF to “1”. As for non-maskable interrupt, keep interrupt service shorten compared with length between interrupt requests; otherwise the status cannot be recovered as non-maskable interrupt would simply nested.

3.4.2 Saving/restoring general-purpose registers

During interrupt acceptance processing, the program counter (PC) and the program status word (PSW, includes IMF) are automatically saved on the stack, but the accumulator and others are not. These registers are saved by software if necessary. When multiple interrupt services are nested, it is also necessary to avoid using the same data memory area for saving registers. The following methods are used to save/restore the general-purpose registers.

3.4.2.1 Using PUSH and POP instructions

If only a specific register is saved or interrupts of the same source are nested, general-purpose registers can be saved/restored using the PUSH/POP instructions.

Example :Save/store register using PUSH and POP instructions

```
PINTxx:    PUSH    WA           ; Save WA register
           (interrupt processing)
           POP     WA           ; Restore WA register
           RETI    ; RETURN
```

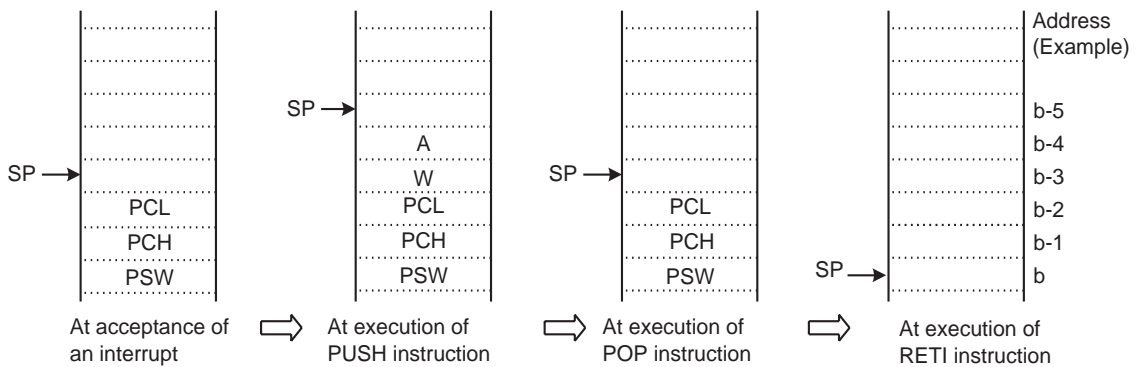


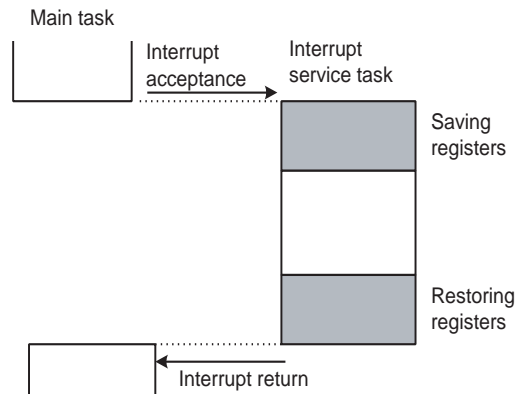
Figure 3-3 Save/store register using PUSH and POP instructions

3.4.2.2 Using data transfer instructions

To save only a specific register without nested interrupts, data transfer instructions are available.

Example :Save/store register using data transfer instructions

```
PINTxx:    LD      (GSAVA), A   ; Save A register
           (interrupt processing)
           LD      A, (GSAVA)   ; Restore A register
           RETI    ; RETURN
```



Saving/Restoring general-purpose registers using PUSH/POP data transfer instruction

Figure 3-4 Saving/Restoring General-purpose Registers under Interrupt Processing

3.4.3 Interrupt return

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

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

As for address trap interrupt (INTATRAP), it is required to alter stacked data for program counter (PC) to restarting address, during interrupt service program.

Note: If [RETN] is executed with the above data unaltered, the program returns to the address trap area and INTATRAP occurs again. When interrupt acceptance processing has completed, stacked data for PCL and PCH are located on address (SP + 1) and (SP + 2) respectively.

Example 1 :Returning from address trap interrupt (INTATRAP) service program

```

PINTxx:      POP      WA          ; Recover SP by 2
              LD       WA, Return Address ;
              PUSH    WA          ; Alter stacked data
              (interrupt processing)
              RETN   ; RETURN
    
```

Example 2 :Restarting without returning interrupt

(In this case, PSW (Includes IMF) before interrupt acceptance is discarded.)

```

PINTxx:      INC      SP          ; Recover SP by 3
              INC      SP          ;
              INC      SP          ;
              (interrupt processing)
              LD       EIRL, data    ; Set IMF to "1" or clear it to "0"
              JP       Restart Address ; Jump into restarting address
    
```

Interrupt requests are sampled during the final cycle of the instruction being executed. Thus, the next interrupt can be accepted immediately after the interrupt return instruction is executed.

Note 1: It is recommended that stack pointer be return to rate before INTATRAP (Increment 3 times), if return interrupt instruction [RETN] is not utilized during interrupt service program under INTATRAP (such as Example 2).

Note 2: When the interrupt processing time is longer than the interrupt request generation time, the interrupt service task is performed but not the main task.

3.5 Software Interrupt (INTSW)

Executing the SWI instruction generates a software interrupt and immediately starts interrupt processing (INTSW is highest prioritized interrupt).

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

3.5.1 Address error detection

FFH is read if for some cause such as noise the CPU attempts to fetch an instruction from a non-existent memory address during single chip mode. Code FFH is the 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 FFH to unused areas of the program memory. Address trap reset is generated in case that an instruction is fetched from RAM, DBR or SFR areas.

3.5.2 Debugging

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

3.6 Undefined Instruction Interrupt (INTUNDEF)

Taking code which is not defined as authorized instruction for instruction causes INTUNDEF. INTUNDEF is generated when the CPU fetches such a code and tries to execute it. INTUNDEF is accepted even if non-maskable interrupt is in process. Contemporary process is broken and INTUNDEF interrupt process starts, soon after it is requested.

Note: The undefined instruction interrupt (INTUNDEF) forces CPU to jump into vector address, as software interrupt (SWI) does.

3.7 Address Trap Interrupt (INTATRAP)

Fetching instruction from unauthorized area for instructions (Address trapped area) causes reset output or address trap interrupt (INTATRAP). INTATRAP is accepted even if non-maskable interrupt is in process. Contemporary process is broken and INTATRAP interrupt process starts, soon after it is requested.

Note: The operating mode under address trapped, whether to be reset output or interrupt processing, is selected on watchdog timer control register (WDTCR).

3.8 External Interrupts

The TMP86FS64FG has 6 external interrupt inputs. These inputs are equipped with digital noise reject circuits (Pulse inputs of less than a certain time are eliminated as noise).

Edge selection is also possible with INT1 to INT4. The $\overline{\text{INT0}}$ /P10 pin can be configured as either an external interrupt input pin or an input/output port, and is configured as an input port during reset.

Edge selection, noise reject control and $\overline{\text{INT0}}$ /P10 pin function selection are performed by the external interrupt control register (EINTCR).

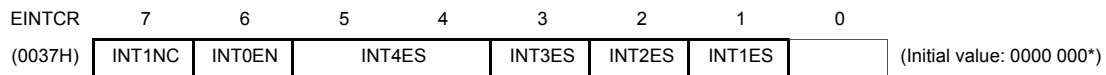
Source	Pin	Enable Conditions	Release Edge (level)	Digital Noise Reject
INT0	$\overline{\text{INT0}}$	IMF • EF4 • INTOEN=1	Falling edge	Pulses of less than 2/fc [s] are eliminated as noise. Pulses of 7/fc [s] or more are considered to be signals. In the SLOW or the SLEEP mode, pulses of less than 1/fs [s] are eliminated as noise. Pulses of 3.5/fs [s] or more are considered to be signals.
INT1	INT1	IMF • EF5 = 1	Falling edge or Rising edge	Pulses of less than 15/fc or 63/fc [s] are eliminated as noise. Pulses of 49/fc or 193/fc [s] or more are considered to be signals.(at CGCR<DV1CK>=0). In the SLOW or the SLEEP mode, pulses of less than 1/fs [s] are eliminated as noise. Pulses of 3.5/fs [s] or more are considered to be signals.
INT2	INT2	IMF • EF8 = 1 and IL8ER=1	Falling edge or Rising edge	Pulses of less than 7/fc [s] are eliminated as noise. Pulses of 25/fc [s] or more are considered to be signals.(at CGCR<DV1CK>=0). In the SLOW or the SLEEP mode, pulses of less than 1/fs [s] are eliminated as noise. Pulses of 3.5/fs [s] or more are considered to be signals.
INT3	INT3	IMF • EF9 = 1 and IL9ER=1	Falling edge or Rising edge	Pulses of less than 7/fc [s] are eliminated as noise. Pulses of 25/fc [s] or more are considered to be signals.(at CGCR<DV1CK>=0). In the SLOW or the SLEEP mode, pulses of less than 1/fs [s] are eliminated as noise. Pulses of 3.5/fs [s] or more are considered to be signals.
INT4	INT4	IMF • EF13 = 1 and IL13ER=1	Falling edge, Rising edge, Falling and Rising edge or H level	Pulses of less than 7/fc [s] are eliminated as noise. Pulses of 25/fc [s] or more are considered to be signals.(at CGCR<DV1CK>=0). In the SLOW or the SLEEP mode, pulses of less than 1/fs [s] are eliminated as noise. Pulses of 3.5/fs [s] or more are considered to be signals.
INT5	$\overline{\text{INT5}}$	IMF • EF14 = 1 and IL14ER=1	Falling edge	Pulses of less than 2/fc [s] are eliminated as noise. Pulses of 7/fc [s] or more are considered to be signals. In the SLOW or the SLEEP mode, pulses of less than 1/fs [s] are eliminated as noise. Pulses of 3.5/fs [s] or more are considered to be signals.

Note 1: In NORMAL1/2 or IDLE1/2 mode, if a signal with no noise is input on an external interrupt pin, it takes a maximum of "signal establishment time + 6/fs[s]" from the input signal's edge to set the interrupt latch.

Note 2: When INTOEN = "0", IL4 is not set even if a falling edge is detected on the $\overline{\text{INT0}}$ pin input.

Note 3: When a pin with more than one function is used as an output and a change occurs in data or input/output status, an interrupt request signal is generated in a pseudo manner. In this case, it is necessary to perform appropriate processing such as disabling the interrupt enable flag.

External Interrupt Control Register



INT1NC	Noise reject time select	0: Pulses of less than $63/f_c$ [s] are eliminated as noise 1: Pulses of less than $15/f_c$ [s] are eliminated as noise	R/W
INT0EN	P10/ $\overline{\text{INT0}}$ pin configuration	0: P10 input/output port 1: $\overline{\text{INT0}}$ pin (Port P10 should be set to an input mode)	R/W
INT4 ES	INT4 edge select	00: Rising edge 01: Falling edge 10: Rising edge and Falling edge 11: H level	R/W
INT3 ES	INT3 edge select	0: Rising edge 1: Falling edge	R/W
INT2 ES	INT2 edge select	0: Rising edge 1: Falling edge	R/W
INT1 ES	INT1 edge select	0: Rising edge 1: Falling edge	R/W

Note 1: f_c : High-frequency clock [Hz], *: Don't care

Note 2: When the system clock frequency is switched between high and low or when the external interrupt control register (EINTCR) is overwritten, the noise canceller may not operate normally. It is recommended that external interrupts are disabled using the interrupt enable register (EIR).

Note 3: The maximum time from modifying INT1NC until a noise reject time is changed is $2^6/f_c$.

Note 4: In case $\overline{\text{RESET}}$ pin is released while the state of INT4 pin keeps "H" level, the external interrupt 4 request is not generated even if the INT4 edge select is specified as "H" level. The rising edge is needed after $\overline{\text{RESET}}$ pin is released.

4. Special Function Register (SFR)

The TMP86FS64FG adopts the memory mapped I/O system, and all peripheral control and data transfers are performed through the special function register (SFR) or the data buffer register (DBR). The SFR is mapped on address 0000H to 003FH, DBR is mapped on address 0F80H to 0FFFH.

This chapter shows the arrangement of the special function register (SFR) and data buffer register (DBR) for TMP86FS64FG.

4.1 SFR

Address	Read	Write
0000H		P0DR
0001H		P1DR
0002H		P2DR
0003H		P3DR
0004H		P4DR
0005H		P5DR
0006H		P6DR
0007H		P7DR
0008H		P0CR
0009H		P1CR
000AH	P4PRD	-
000BH		P3CR
000CH		P4CR
000DH		P5CR
000EH		ADCCR1
000FH		ADCCR2
0010H		TC3DRA
0011H	TC3DRB	-
0012H		TC3CR
0013H		TC2CR
0014H		TC4CR
0015H		TC5CR
0016H		TC6CR
0017H		TC6DR
0018H		TC4DR
0019H		TC5DR
001AH		IRDACR
001BH	UARTSR	UARTCR1
001CH	-	UARTCR2
001DH	RDBUF	TDBUF
001EH		Reserved
001FH		Reserved
0020H		TC1DRAL
0021H		TC1DRAH
0022H		TC1DRBL
0023H		TC1DRBH
0024H		TC2DRL
0025H		TC2DRH

Address	Read	Write
0026H	ADCDR2	-
0027H	ADCDR1	-
0028H	-	SIO1CR1
0029H	SIO1SR	SIO1CR2
002AH	SCISEL	
002BH	Reserved	
002CH	P2PRD	-
002DH	P4OED	
002EH	P6CR	
002FH	P7CR	
0030H	CGCR	
0031H	-	STOPCR
0032H	TC1CR	
0033H	Reserved	
0034H	-	WDTCR1
0035H	-	WDTCR2
0036H	TBTCR	
0037H	EINTCR	
0038H	SYSCR1	
0039H	SYSCR2	
003AH	EIRL	
003BH	EIRH	
003CH	ILL	
003DH	ILH	
003EH	INTSEL	
003FH	PSW	

Note 1: Do not access reserved areas by the program.

Note 2: - ; Cannot be accessed.

Note 3: Write-only registers and interrupt latches cannot use the read-modify-write instructions (Bit manipulation instructions such as SET, CLR, etc. and logical operation instructions such as AND, OR, etc.).

4.2 DBR

Address	Read	Write
0F80H		Reserved
0F81H		Reserved
0F82H		Reserved
0F83H		Reserved
0F84H		Reserved
0F85H		Reserved
0F86H		Reserved
0F87H		Reserved
0F88H		Reserved
0F89H		Reserved
0F8AH		Reserved
0F8BH		Reserved
0F8CH		Reserved
0F8DH		Reserved
0F8EH		Reserved
0F8FH		Reserved
0F90H		SIO1BR0
0F91H		SIO1BR1
0F92H		SIO1BR2
0F93H		SIO1BR3
0F94H		SIO1BR4
0F95H		SIO1BR5
0F96H		SIO1BR6
0F97H		SIO1BR7
0F98H		SIO2BR0
0F99H		SIO2BR1
0F9AH		SIO2BR2
0F9BH		SIO2BR3
0F9CH		SIO2BR4
0F9DH		SIO2BR5
0F9EH		SIO2BR6
0F9FH		SIO2BR7

Address	Read	Write
0FA0H		Reserved
0FA1H		Reserved
0FA2H		Reserved
0FA3H		Reserved
0FA4H		Reserved
0FA5H		Reserved
0FA6H		Reserved
0FA7H		Reserved
0FA8H		Reserved
0FA9H		Reserved
0FAAH		Reserved
0FABH		Reserved
0FACH		Reserved
0FADH		Reserved
0FAEH		Reserved
0FAFH		Reserved
0FB0H		P8DR
0FB1H		P9DR
0FB2H		P8CR
0FB3H		P9CR
0FB4H	-	SIO2CR1
0FB5H	SIO2SR	SIO2CR2
0FB6H		PADR
0FB7H		PBDR
0FB8H		PACR
0FB9H		PBCR
0FBAH		PAPU
0FBBH		PBPU
0FBCH		P6PU
0FBDH		P7PU
0FBEH		Reserved
0FBFH		Reserved

Address	Read	Write
0FC0H		Reserved
::		::
0FDFH		Reserved

Address	Read	Write
0FE0H		Reserved
0FE1H		Reserved
0FE2H		Reserved
0FE3H		Reserved
0FE4H		Reserved
0FE5H		Reserved
0FE6H		Reserved
0FE7H		Reserved
0FE8H		Reserved
0FE9H		Reserved
0FEAH		SPCR
0FEBH		Reserved
0FECH		Reserved
0FEDH		Reserved
0FEEH		Reserved
0FEFH		Reserved
0FF0H		Reserved
0FF1H		Reserved
0FF2H		Reserved
0FF3H		Reserved
0FF4H		Reserved
0FF5H		Reserved
0FF6H		Reserved
0FF7H		Reserved
0FF8H		Reserved
0FF9H		Reserved
0FFAH		Reserved
0FFBH		Reserved
0FFCH		Reserved
0FFDH		Reserved
0FFEH		Reserved
0FFFH		FLSCR

Note 1: Do not access reserved areas by the program.

Note 2: – ; Cannot be accessed.

Note 3: Write-only registers and interrupt latches cannot use the read-modify-write instructions (Bit manipulation instructions such as SET, CLR, etc. and logical operation instructions such as AND, OR, etc.).

5. I/O Ports

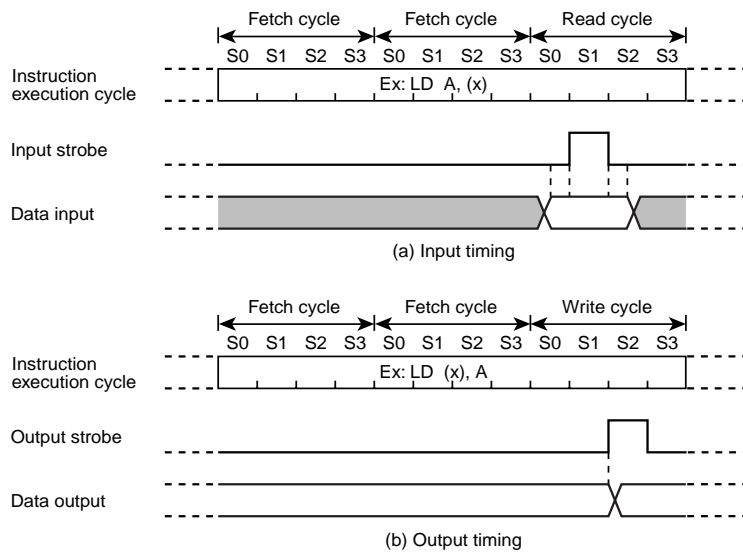
The TMP86FS64FG have 12 parallel input/output ports (91 pins) as follows.

1. Port P0 (8-bit I/O port)
2. Port P1 (8-bit I/O port)
 - External interrupt input, timer/counter input, divider output.
3. Port P2 (3-bit I/O port)
 - External interrupt input, STOP mode release signal input.
4. Port P3 (8-bit I/O port)
 - Timer/counter input, serial interface input/output.
5. Port P4 (8-bit I/O port)
 - Timer/counter input, serial interface input/output, external interrupt input.
6. Port P5 (8-bit I/O port)
7. Port P6 (8-bit I/O port)
 - Analog input.
8. Port P7 (8-bit I/O port)
 - Analog input, STOP mode release signal input.
9. Port P8 (8-bit I/O port)
10. Port P9 (8-bit I/O port)
11. Port PA (8-bit I/O port)
12. Port PB (8-bit I/O port)

Each output port contains a latch, which holds the output data. All input ports do not have latches, so the external input data should be externally held until the input data is read from outside or reading should be performed several timer before processing. Figure 5-1 shows input/output timing examples.

External data is read from an I/O port in the S1 state of 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.

Output data changes in the S2 state of the write cycle during execution of the instruction which writes to an I/O port.

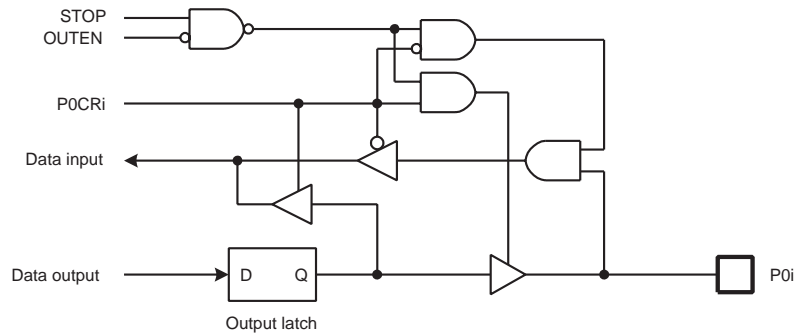


Note: The positions of the read and write cycles may vary, depending on the instruction.

Figure 5-1 Input/Output Timing (Example)

5.1 Port P0 (P07 to P00)

Port P0 is an 8-bit input/output port, which can be configured individually as an input or an output under software control. Input/output mode is specified by the corresponding bit in the port P0 input/output control register (P0CR). During reset, the P0CR is initialized to “0”, which configures port P0 as an input. The P0 output latches are also initialized to “0”.



Note 1: i = 7 to 0

Note 2: STOP: bit 7 in SYSCR1, OUTEN: bit 4 in SYSCR1

Figure 5-2 Port P0

P0DR (0000H)	7	6	5	4	3	2	1	0	(Initial value: 0000 0000)
	P07	P06	P05	P04	P03	P02	P01	P00	

P0CR (0008H)	7	6	5	4	3	2	1	0	(Initial value: 0000 0000)
	P0CR7	P0CR6	P0CR5	P0CR4	P0CR3	P0CR2	P0CR1	P0CR0	

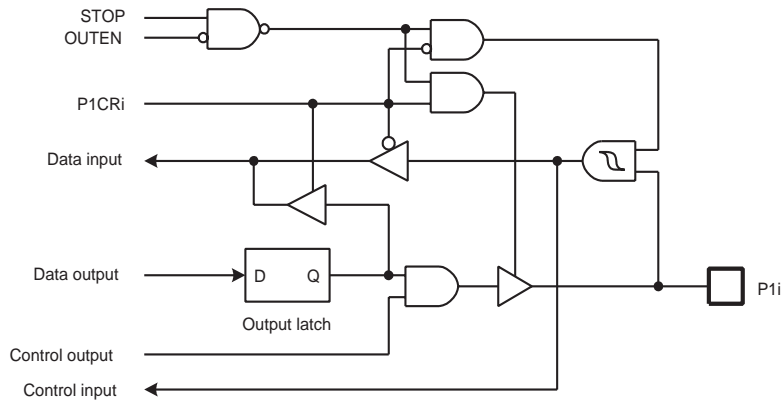
P0CR	I/O control for port P0 (specified for each bit)	0: Input mode 1: Output mode	R/W
------	---	---------------------------------	-----

Note: When used as an input mode, read-modify-write instructions such as bit manipulate instructions cannot be used. Read-modify-write instruction writes the all data of 8-bit after read and modified. Because a bit setting input mode reads data of terminal, the output latch is changed by these instructions.

5.2 Port P1 (P17 to P10)

Port P1 is an 8-bit input/output port, which can be configured individually as an input or an output under software control. Input/output mode is specified by the corresponding bit in the port P1 input/output control register (P1CR). During reset, the P1CR is initialized to “0”, which configures port P1 as an input mode. The P1 output latches are also initialized to “0”.

It is also used as $\overline{INT0}$, INT1, INT2/TC1, TC2, \overline{DVO} and \overline{PPG} . When used as secondary function pin, the input pins ($\overline{INT0}$, INT1, INT2, TC1, TC2) should be set to the input mode and the output pins (\overline{DVO} , \overline{PPG}) should be set to the output mode beforehand the output latch should be set to “1”.



Note 1: i = 7 to 0

Note 2: STOP: bit 7 in SYSCR1, OUTEN: bit 4 in SYSCR1

Figure 5-3 Port P1

	7	6	5	4	3	2	1	0	
P1DR (0001H)	P17	P16	P15 TC2	P14 \overline{PPG}	P13 \overline{DVO}	P12 INT2 TC1	P11 INT1	P10 $\overline{INT0}$	(Initial value: 0000 0000)

	7	6	5	4	3	2	1	0	
P1CR (0009H)	P1CR7	P1CR6	P1CR5	P1CR4	P1CR3	P1CR2	P1CR1	P1CR0	(Initial value: 0000 0000)

P1CR	I/O control for port P1 (specified for each bit)	0: Input mode 1: Output mode	R/W
------	---	---------------------------------	-----

Note: When used as an input mode, read-modify-write instructions such as bit manipulate instructions cannot be used. Read-modify-write instruction writes the all data of 8-bit after read and modified. Because a bit setting input mode reads data of terminal, the output latch is changed by these instructions.

5.3 Port P2 (P22 to P20)

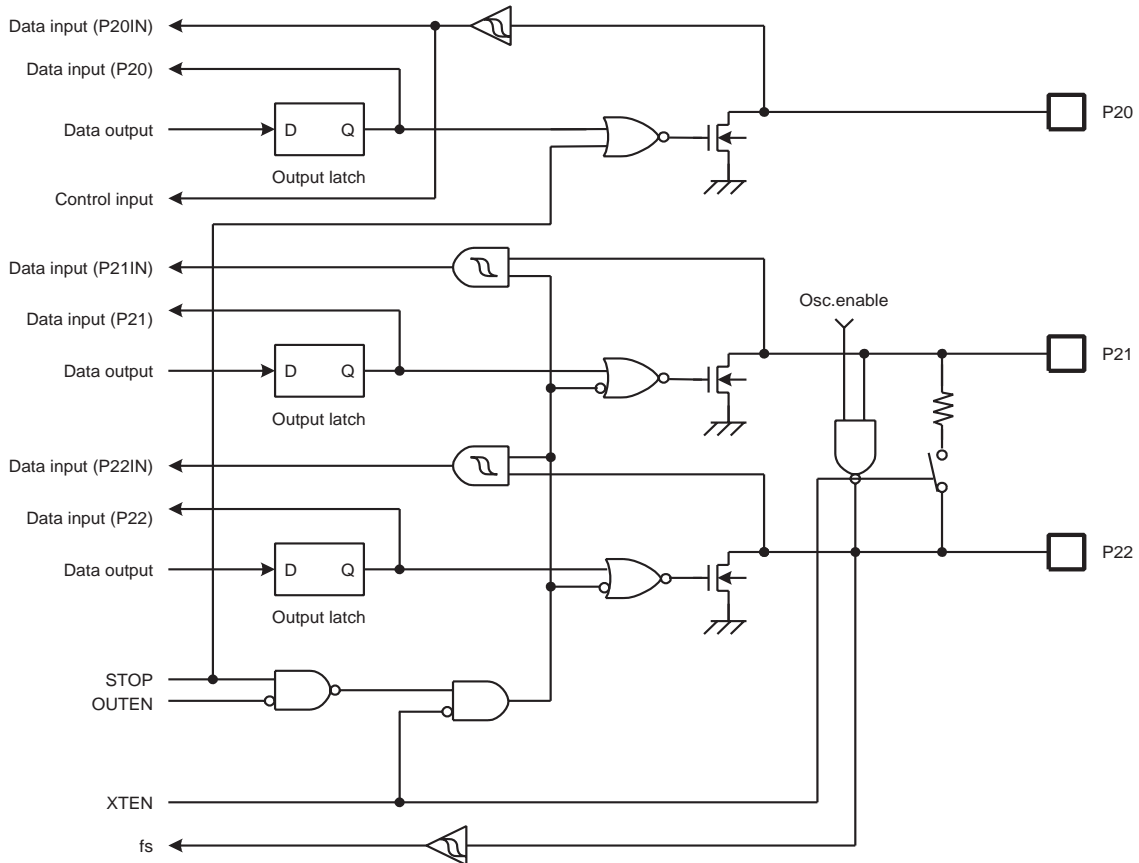
Port P2 is a 3-bit input/output port. During reset, the P2DR is initialized to “1”.

It is also used as $\overline{\text{INT5}}/\text{STOP1}$. When used as secondary function pin or an input pin, the output latch should be set to “1”.

In the dual-clock mode, the low-frequency oscillator (32.768 kHz) is connected to P21 (XTIN) and P22 (XTOUT) pins.

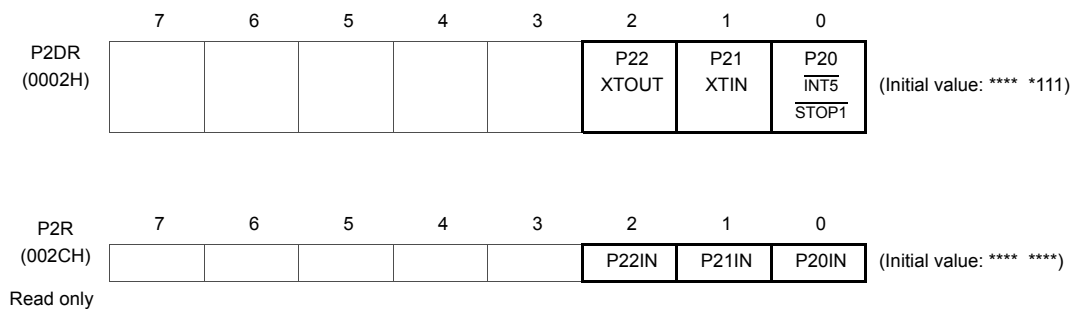
P2 port output latch (P2DR) and P2 port terminal input (P2R) are located on their respective address.

When a read instruction is executed for port P2, read data of bits 7 to 3 are unstable.



Note: STOP: bit 7 in SYSCR1, OUTEN: bit 4 in SYSCR1, XTEN: bit 6 in SYSCR2

Figure 5-4 Port P2



Note 1: Port P20 is used as $\overline{\text{STOP1}}$ pin. Therefore, when stop mode is started, however SYSCR1<OUTEN> is set to “1”, port P20 becomes High-Z (input mode).

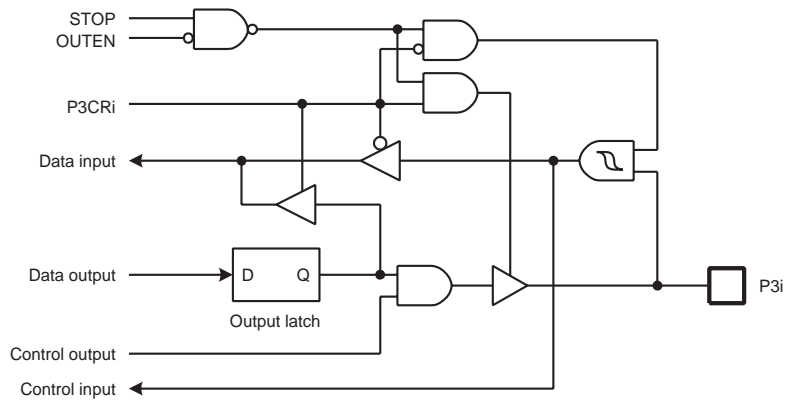
Note 2: Each terminal has a protect diode. Please refer to section “Input/Output Circuitry; (2) Input/Output ports”.

5.4 Port P3 (P37 to P30)

Port P3 is an 8-bit input/output port, which can be configured individually as an input or an output under software control. Input/output mode is specified by the corresponding bit in the port P3 input/output control register (P3CR). During reset, the P3CR is initialized to “0”, which configures port P3 as an input mode. The P3 output latches (P3DR) are also initialized to “1”.

Port P30, P31 and P32 are also used as TC4/ $\overline{\text{PWM4/PDO4}}$, TC5/ $\overline{\text{PWM5/PDO5}}$ and TC6/ $\overline{\text{PWM6/PDO6}}$. When used as secondary function pin, the input pins (TC4, TC5, TC6) should be set to the input mode and the output pins ($\overline{\text{PWM4/PDO4}}$, $\overline{\text{PWM5/PDO5}}$, $\overline{\text{PWM6/PDO6}}$) should be set to the output mode.

Port P33, P34, P35, P36 and P37 are also used as $\overline{\text{SCK1}}$, SI1, SO1, SI2 and SO2. When used as secondary function pin, $\overline{\text{SCK1}}$ should be set to the input or output mode, SI1 and SI2 should be set to the input mode, SO1 and SO2 should be set to the output mode.



Note 1: i = 7 to 0

Note 2: STOP: bit 7 in SYSCR1, OUTEN: bit 4 in SYSCR1

Figure 5-5 Port P3

	7	6	5	4	3	2	1	0	
P3DR (0003H)	P37 SO2	P36 SI2	P35 SO1	P34 SI1	P33 $\overline{\text{SCK1}}$	P32 TC6 $\overline{\text{PWM6/PDO6}}$	P31 TC5 $\overline{\text{PWM5/PDO5}}$	P30 TC4 $\overline{\text{PWM4/PDO4}}$	(Initial value: 1111 1111)

	7	6	5	4	3	2	1	0	
P3CR (000BH)	P3CR7	P3CR6	P3CR5	P3CR4	P3CR3	P3CR2	P3CR1	P3CR0	(Initial value: 0000 0000)

P3CR	I/O control for port P3 (specified for each bit)	0: Input mode 1: Output mode	R/W
------	---	---------------------------------	-----

Note: When used as an input mode, read-modify-write instructions such as bit manipulate instructions cannot be used. Read-modify-write instruction writes the all data of 8-bit after read and modified. Because a bit setting input mode reads data of terminal, the output latch is changed by these instructions.

5.5 Port P4 (P47 to P40)

Port P4 is an 8-bit input/output port, which can be configured individually as an input or an output under software control. Input/output mode is specified by the corresponding bit in the port P4 input/output control register (P4CR). During reset, the P4CR is initialized to “0”, which configures port P4 as an input mode. The P4 output latches are also initialized to “1”.

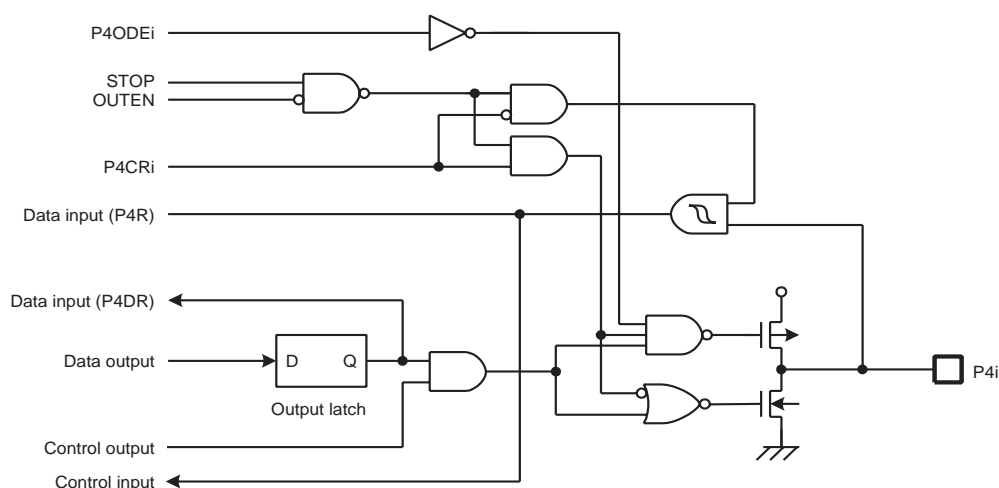
It is also used as $\overline{\text{INT0}}$, INT1, INT2/TC1, TC2, $\overline{\text{DVO}}$ and $\overline{\text{PPG}}$. When used as secondary function pin, the input pins ($\overline{\text{INT0}}$, INT1, INT2, TC1, TC2) should be set to the input mode and the output pins ($\overline{\text{DVO}}$, $\overline{\text{PPG}}$) should be set to the output mode beforehand the output latch should be set to “1”.

Port P4 can be configured individually as a tri-state output or sink open drain output under software control. It is specified by the corresponding bit in the P4ODE. During reset, the P4ODE is initialized to “0”, and then P4CR is set to “1”, the tri-state output is configured.

P4 port output latch (P4DR) and P4 port terminal input (P4R) are located on their respective address.

When the input mode and output mode are configured simultaneously, even if the bit manipulate instruction is executed, the data of the output latch of the terminal set as the input mode is not influenced of the terminal input.

Port P40, P41, P42, P44, P45, P46 and P47 are also used as SCK2, RXD1, TXD1, RXD2, TXD2, INT3/TC3 and INT4. When used as secondary function pin, the $\overline{\text{SCK1}}$ pin should be set to the input or output mode, the input pins (RXD1, RXD2, INT3/TC3, INT4) should be set to the input mode and the output pins (TXD1, TXD2) should be set to the output.



Note 1: $i = 7$ to 0

Note 2: STOP: bit 7 in SYSCR1, OUTEN: bit 4 in SYSCR1

Figure 5-6 Port P4

	7	6	5	4	3	2	1	0	
P4DR (0004H)	P47 INT4	P46 INT3 TC3	P45 TXD2	P44 RXD2	P43	P42 TXD1	P41 RXD1	P40 SCK2	(Initial value: 1111 1111)

	7	6	5	4	3	2	1	0	
P4R (000AH)	P47IN	P46IN	P45IN	P44IN	P43IN	P42IN	P41IN	P40IN	(Initial value: **** *)

Read only

	7	6	5	4	3	2	1	0	
P4CR (000CH)	P4CR7	P4CR6	P4CR5	P4CR4	P4CR3	P4CR2	P4CR1	P4CR0	(Initial value: 0000 0000)

P4CR	I/O control for port P4 (specified for each bit)	0: Input mode 1: Output mode	R/W
------	---	---------------------------------	-----

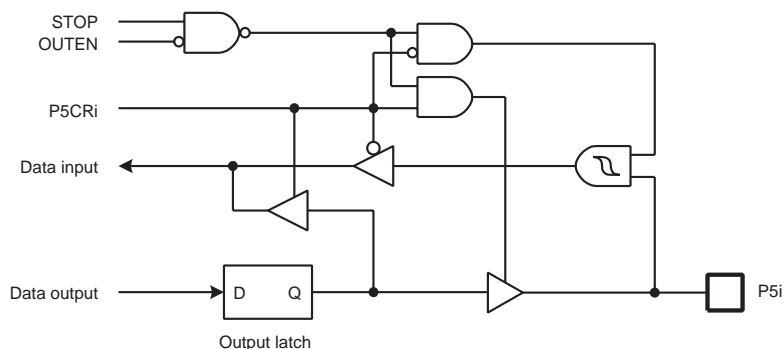
	7	6	5	4	3	2	1	0	
P4ODE (002DH)	P4ODE7	P4ODE6	P4ODE5	P4ODE4	P4ODE3	P4ODE2	P4ODE1	P4ODE0	(Initial value: 0000 0000)

P4ODE	P4 open drain control register (Specified for each bit)	0: Tri-state output 1: Sink open drain output	R/W
-------	--	--	-----

Note: Regardless of P4ODE setting, each terminal has a protect diode. Please refer to section "Input/Output Circuitry; (2) Input/Output ports".

5.6 Port P5 (P57 to P50)

Port P5 is an 8-bit input/output port, which can be configured individually as an input or an output under software control. Input/output mode is specified by the corresponding bit in the port P5 input/output control register (P5CR). During reset, the P5CR is initialized to “0”, which configures port P5 as an input mode. The P5 output latches are also initialized to “0”.



Note 1: $i = 7$ to 0

Note 2: STOP: bit 7 in SYSCR1, OUTEN: bit 4 in SYSCR1

Figure 5-7 Port P5

P5DR (0005H)	7	6	5	4	3	2	1	0	(Initial value: 0000 0000)
	P57	P56	P55	P54	P53	P52	P51	P50	

P5CR (000DH)	7	6	5	4	3	2	1	0	(Initial value: 0000 0000)
	P5CR7	P5CR6	P5CR5	P5CR4	P5CR3	P5CR2	P5CR1	P5CR0	

P5CR	I/O control for port P5 (specified for each bit)	0: Input mode 1: Output mode	R/W
------	---	---------------------------------	-----

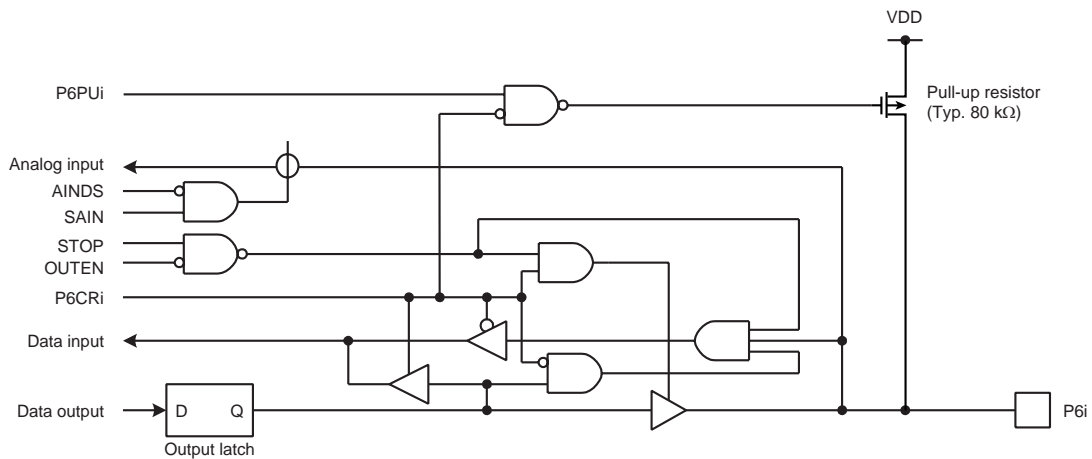
Note: When used as an input mode, read-modify-write instructions such as bit manipulate instructions cannot be used. Read-modify-write instruction writes the all data of 8-bit after read and modified. Because a bit setting input mode reads data of terminal, the output latch is changed by these instructions.

5.7 Port P6 (P67to P60)

Port P6 is an 8-bit input/output port, which can be configured individually as an input or an output under software control. Port P6 is also used as an analog input. Input/output mode is specified by the corresponding bit in the port P6 input/output control register (P6CR), P6 output latch (P6DR) and ADCCR1<AINDS>. During reset, P6CR and P6DR are initialized to “0” and ADCCR1<AINDS> is set to “1”. At the same time, the input data of P67 to P60 are fixed to “0” level. When port P6 is used as input port, the corresponding bit in P6CR and P6DR should be set to input mode (P6CR = “0”, P6DR = “1”). When used as output port, the corresponding bit in P6CR should be set to “1”. When used as analog input port, the corresponding bit in P6CR and P6DR should be set to analog input mode (P6CR = “0”, P6DR = “0”) and ADCCR1<AINDS> is set to “0”, then the AD conversion is started. Setting P6DR to “0” is necessary to prevent generating the penetration electric current. So the output latch of the port used as analog input should be set to “0” beforehand. Actually selection of the conversion input channels is specified by ADCCR1<SAIN>.

Pins used for analog input can be used as I/O port. During AD conversion, output instructions should not be executed to keep a precision. In addition, a variable signal should not be input to a port adjacent to the analog input during AD conversion.

When the AD converter is in use (P6DR = “0”), bits mentioned above are read as “0” by executing input instructions.



Note 1: i = 7 to 0

Note 2: STOP: bit 7 in SYSCR1, OUTEN: bit 4 in SYSCR1

Note 3: SAIN: AD input channel select signal

Figure 5-8 Port P6

	7	6	5	4	3	2	1	0	
P6DR (0006H)	P67 AIN7	P66 AIN6	P65 AIN5	P64 AIN4	P63 AIN3	P62 AIN2	P61 AIN1	P60 AIN0	(Initial value: 0000 0000)

	7	6	5	4	3	2	1	0	
P6CR (002EH)	P6CR7	P6CR6	P6CR5	P6CR4	P6CR3	P6CR2	P6CR1	P6CR0	(Initial value: 0000 0000)

P6CR	I/O control for port P6 (specified for each bit)		AINDS = 1 (AD unused)		AINDS = 0 (AD used)		R/W
			P6DR = "0"	P6DR = "1"	P6DR = "0"	P6DR = "1"	
		0	Input "0" fixed #1	Input mode	AD input #2	Input mode	
		1	Output mode				

#1 Input data to a pin whose input is fixed to "0" is always "0" regardless of the pin state and whether or not a programmable pull-up resistor is added.

#2 When a read instruction for port P6 is executed, the bit of analog input mode becomes read data "0".

Note 1: Don't set output mode to pin, which is used for an analog input.

Note 2: When used for input mode (include analog input mode), read-modify-write instruction such as bit manipulate instructions cannot be used.

Read-modify-write instruction writes the all data of 8-bit after data is read and modified. Because a bit setting input mode read data of terminal, the output latch is changed by these instructions. So P6 port cannot input data.

	7	6	5	4	3	2	1	0	
P6PU (0FBCH)	P6PU7	P6PU6	P6PU5	P6PU4	P6PU3	P6PU2	P6PU1	P6PU0	(Initial value: 0000 0000)

P6PU	Port P6 pull up control register (specified for each bit)	0: Non pull-up 1: Pull-up	R/W
------	--	------------------------------	-----

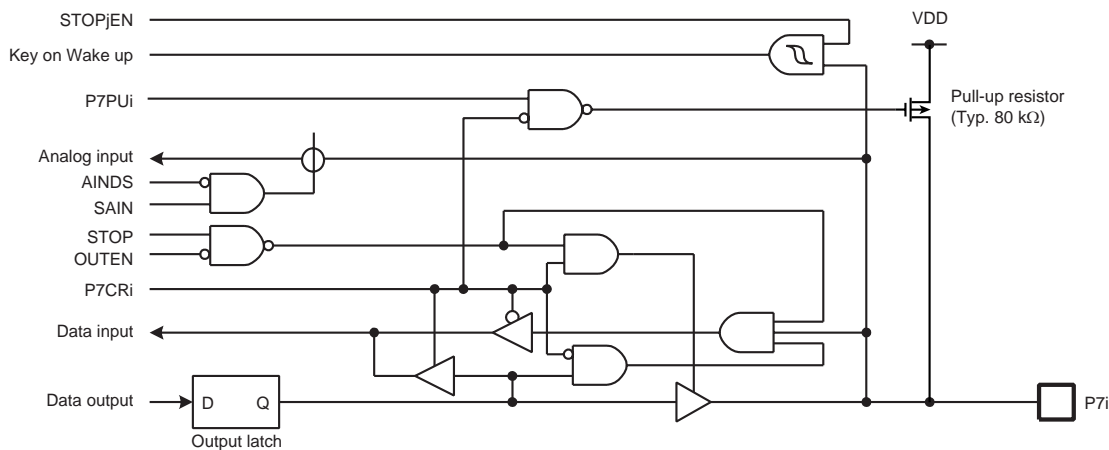
Note: However the P6PU is set to "1" (pull-up), the port configured output is not set up pull-up resistor.

5.8 Port P7 (P77 to P70)

Port P7 is an 8-bit input/output port, which can be configured individually as an input or an output under software control. Port P7 is also used as an analog input and Key on Wake up input. Input/output mode is specified by the corresponding bit in the port P7 input/output control register (P7CR), P7 output latch (P7DR) and ADCCR1<AINDS>. During reset, P7CR and P7DR are initialized to “0” and ADCCR1<AINDS> is set to “1”. At the same time, the input data of P77 to P70 are fixed to “0” level. When port P7 is used as input port, the corresponding bit in P7CR and P7DR should be set to input mode (P7CR = “0”, P7DR = “1”). When used as output port, the corresponding bit in P7CR should be set to “1”. When used as analog input port, the corresponding bit in P7CR and P7DR should be set to analog input mode (P7CR = “0”, P7DR = “0”) and ADCCR1<AINDS> is set to “0”, then the AD conversion is started. Setting P7DR to “0” is necessary to prevent generating the penetration electric current. So the output latch of the port used as analog input should be set to “0” beforehand. Actually selection of the conversion input channels is specified by ADCCR1<SAIN>.

Pins used for analog input can be used as I/O port. During AD conversion, output instructions should not be executed to keep a precision. In addition, a variable signal should not be input to a port adjacent to the analog input during AD conversion.

When the AD converter is in use (P7DR = “0”), bits mentioned above are read as “0” by executing input instructions.



- Note 1: i = 7 to 0, j = 5 to 2
- Note 2: STOP: bit 7 in SYSCR1, OUTEN: bit 4 in SYSCR1
- Note 3: SAIN: bit 0 to 3 in ADCCRA
- Note 4: SOTpjEN: bit 4 to 7 in STOPCR

Figure 5-9 Port P7

	7	6	5	4	3	2	1	0	
P7DR (0007H)	P77 AIN15 STOP5	P76 AIN14 STOP4	P75 AIN13 STOP3	P74 AIN12 STOP2	P73 AIN11	P72 AIN10	P71 AIN9	P70 AIN8	(Initial value: 0000 0000)

	7	6	5	4	3	2	1	0	
P7CR (002FH)	P7CR7	P7CR6	P7CR5	P7CR4	P7CR3	P7CR2	P7CR1	P7CR0	(Initial value: 0000 0000)

P7CR	I/O control for port P7 (specified for each bit)		AINDS = 1 (AD unused)		AINDS = 0 (AD used)		R/W
			P7DR = "0"	P7DR = "1"	P7DR = "0"	P7DR = "1"	
		0	Input "0" fixed #1	Input mode	AD input #2	Input mode	
		1	Output mode				

#1 Input data to a pin whose input is fixed to "0" is always "0" regardless of the pin state and whether or not a programmable pull-up resistor is added.

#2 When a read instruction for port P7 is executed, the bit of analog input mode becomes read data "0".

Note 1: Don't set output mode to pin, which is used for an analog input.

Note 2: When used for input mode (include analog input mode), read-modify-write instruction such as bit manipulate instructions cannot be used.

Read-modify-write instruction writes the all data of 8-bit after data is read and modified. Because a bit setting input mode read data of terminal, the output latch is changed by these instructions. So P7 port cannot input data.

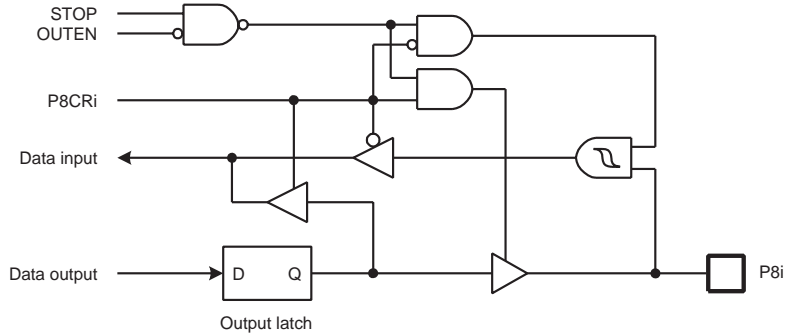
	7	6	5	4	3	2	1	0	
P7PU (0FBDH)	P7PU7	P7PU6	P7PU5	P7PU4	P7PU3	P7PU2	P7PU1	P7PU0	(Initial value: 0000 0000)

P7PU	Port P7 pull up control register (specified for each bit)	0: Non pull-up 1: Pull-up	R/W
------	--	------------------------------	-----

Note: However the P7PU is set to "1" (pull-up), the port configured output is not set up pull-up resistor.

5.9 Port P8 (P87 to P80)

Port P8 is an 8-bit input/output port, which can be configured individually as an input or an output under software control. Input/output mode is specified by the corresponding bit in the port P8 input/output control register (P8CR). During reset, the P8CR is initialized to “0”, which configures port P8 as an input. The P8 output latches are also initialized to “0”.



Note 1: i = 7 to 0

Note 2: STOP: bit 7 in SYSCR1, OUTEN: bit 4 in SYSCR1

Figure 5-10 Port P8

P8DR (0FB0H)	7	6	5	4	3	2	1	0	(Initial value: 0000 0000)
	P87	P86	P85	P84	P83	P82	P81	P80	

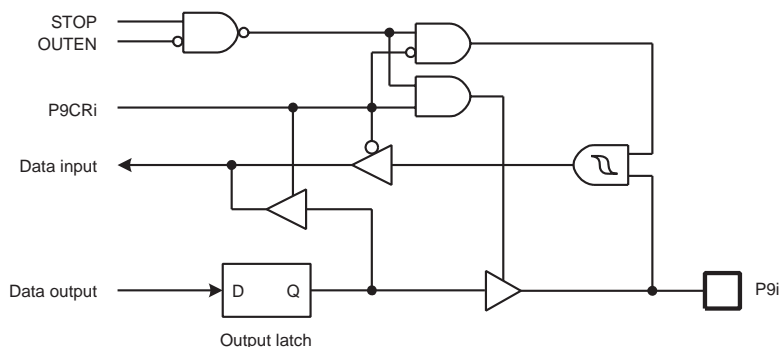
P8CR (0FB2H)	7	6	5	4	3	2	1	0	(Initial value: 0000 0000)
	P8CR7	P8CR6	P8CR5	P8CR4	P8CR3	P8CR2	P8CR1	P8CR0	

P8CR	I/O control for port P8 (specified for each bit)	0: Input mode 1: Output mode	R/W
------	---	---------------------------------	-----

Note: When used as an input mode, read-modify-write instructions such as bit manipulate instructions cannot be used. Read-modify-write instruction writes the all data of 8-bit after read and modified. Because a bit setting input mode reads data of terminal, the output latch is changed by these instructions.

5.10 Port P9 (P97 to P90)

Port P9 is an 8-bit input/output port, which can be configured individually as an input or an output under software control. Input/output mode is specified by the corresponding bit in the port P9 input/output control register (P9CR). During reset, the P9CR is initialized to “0”, which configures port P9 as an input. The P9 output latches are also initialized to “0”.



Note 1: $i = 7$ to 0

Note 2: STOP: bit 7 in SYSCR1, OUTEN: bit 4 in SYSCR1

Figure 5-11 Port P9

P9DR	7	6	5	4	3	2	1	0	
(0FB1H)	P97	P96	P95	P94	P93	P92	P91	P90	(Initial value: 0000 0000)

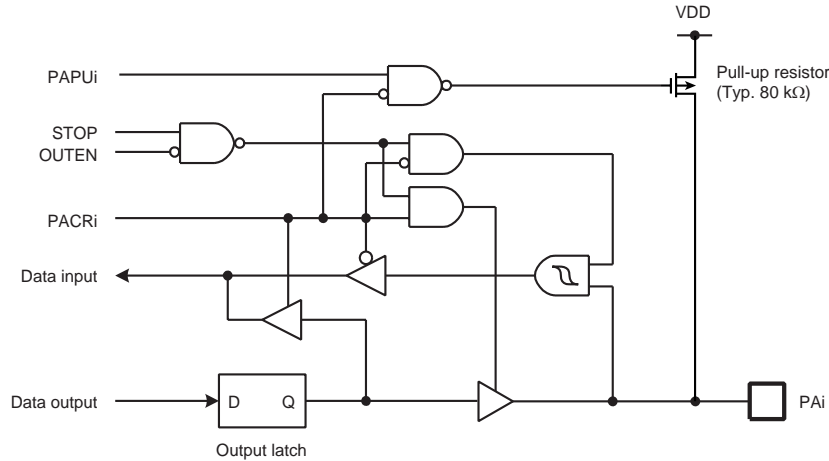
P9CR	7	6	5	4	3	2	1	0	
(0FB3H)	P9CR7	P9CR6	P9CR5	P9CR4	P9CR3	P9CR2	P9CR1	P9CR0	(Initial value: 0000 0000)

P9CR	I/O control for port P9 (specified for each bit)	0: Input mode 1: Output mode	R/W

Note: When used as an input mode, read-modify-write instructions such as bit manipulate instructions cannot be used. Read-modify-write instruction writes the all data of 8-bit after read and modified. Because a bit setting input mode reads data of terminal, the output latch is changed by these instructions.

5.11 Port PA (PA7 to PA0)

Port PA is an 8-bit input/output port, which can be configured individually as an input or an output under software control. Input/output mode is specified by the corresponding bit in the port PA input/output control register (PACR). During reset, the PACR is initialized to “0”, which configures port PA as an input. The PA output latches are also initialized to “0”.



Note 1: i = 7 to 0

Note 2: STOP: bit 7 in SYSCR1, OUTEN: bit 4 in SYSCR1

Figure 5-12 Port PA

PADR (0FB6H)	7	6	5	4	3	2	1	0	(Initial value: 0000 0000)
	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	

PACR (0FB8H)	7	6	5	4	3	2	1	0	(Initial value: 0000 0000)
	PACR7	PACR6	PACR5	PACR4	PACR3	PACR2	PACR1	PACR0	

PACR	I/O control for port PA (specified for each bit)	0: Input mode 1: Output mode	R/W
------	---	---------------------------------	-----

Note: When used as an input mode, read-modify-write instructions such as bit manipulate instructions cannot be used. Read-modify-write instruction writes the all data of 8-bit after read and modified. Because a bit setting input mode reads data of terminal, the output latch is changed by these instructions.

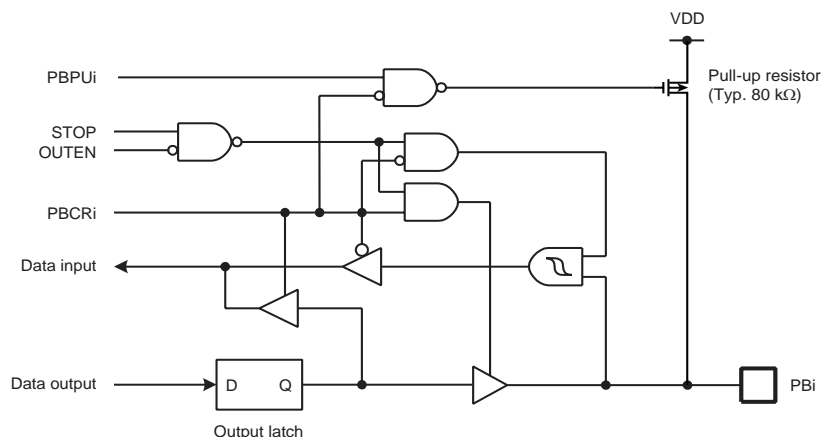
PAPU (0FBAH)	7	6	5	4	3	2	1	0	(Initial value: 0000 0000)
	PAPU7	PAPU6	PAPU5	PAPU4	PAPU3	PAPU2	PAPU1	PAPU0	

PAPU	Port PA pull up control register (specified for each bit)	0: Non pull-up 1: Pull-up	R/W
------	--	------------------------------	-----

Note: However the PAPU is set to “1” (pull-up), the port configured output is not set up pull-up resistor.

5.12 Port PB (PB7 to PB0)

Port PB is an 8-bit input/output port, which can be configured individually as an input or an output under software control. Input/output mode is specified by the corresponding bit in the port PB input/output control register (PBCR). During reset, the PBCR is initialized to “0” which configures port PB as an input. The PB output latches are also initialized to “0”.



Note 1: $i = 7$ to 0

Note 2: STOP: bit 7 in SYSCR1, OUTEN: bit 4 in SYSCR1

Figure 5-13 Port PB and PBCR

PBDR (0FB7H)	7	6	5	4	3	2	1	0	
	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	(Initial value: 0000 0000)

PBCR (0FB9H)	7	6	5	4	3	2	1	0	
	PBCR7	PBCR6	PBCR5	PBCR4	PBCR3	PBCR2	PBCR1	PBCR0	(Initial value: 0000 0000)

PBCR	I/O control for port PB (specified for each bit)	0: Input mode 1: Output mode	R/W
------	---	---------------------------------	-----

Note: When used as an input mode, read-modify-write instructions such as bit manipulate instructions cannot be used. Read-modify-write instruction writes the all data of 8-bit after read and modified. Because a bit setting input mode reads data of terminal, the output latch is changed by these instructions.

PBPU (0FBBH)	7	6	5	4	3	2	1	0	
	PBPU7	PBPU6	PBPU5	PBPU4	PBPU3	PBPU2	PBPU1	PBPU0	(Initial value: 0000 0000)

PBPU	Port PB pull up control register (specified for each bit)	0: Non pull-up 1: Pull-up	R/W
------	--	------------------------------	-----

Note: However the PBPU is set to “1” (pull-up), the port configured output is not set up pull-up resistor.

6. Watchdog Timer (WDT)

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

The watchdog timer signal for detecting malfunctions can be programmed only once as “reset request” or “interrupt request”. Upon the reset release, this signal is initialized to “reset request”.

When the watchdog timer is not used to detect malfunctions, it can be used as the timer to provide a periodic interrupt.

Note: Care must be taken in system design since the watchdog timer functions are not be operated completely due to effect of disturbing noise.

6.1 Watchdog Timer Configuration

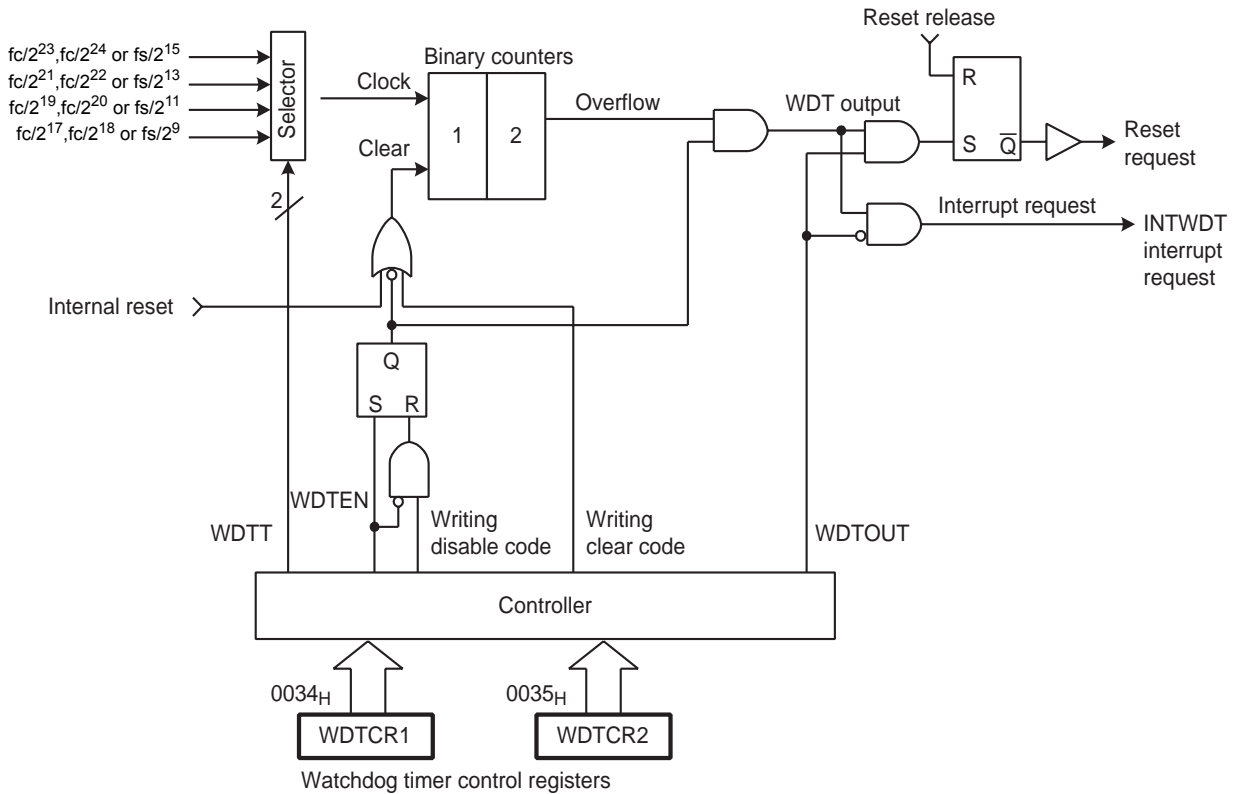


Figure 6-1 Watchdog Timer Configuration

6.2 Watchdog Timer Control

The watchdog timer is controlled by the watchdog timer control registers (WDTCR1 and WDTCR2). The watchdog timer is automatically enabled after the reset release.

6.2.1 Malfunction Detection Methods Using the Watchdog Timer

The CPU malfunction is detected, as shown below.

1. Set the detection time, select the output, and clear the binary counter.
2. Clear the binary counter repeatedly within the specified detection time.

If the CPU malfunctions such as endless loops or the deadlock conditions occur for some reason, the watchdog timer output is activated by the binary-counter overflow unless the binary counters are cleared. When WDTCR1<WDTOUT> is set to “1” at this time, the reset request is generated and then internal hardware is initialized. When WDTCR1<WDTOUT> is set to “0”, a watchdog timer interrupt (INTWDT) is generated.

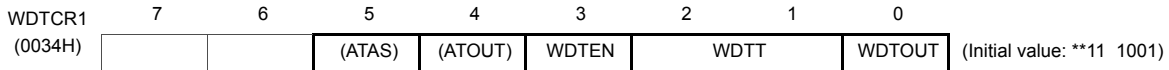
The watchdog timer temporarily stops counting in the STOP mode including the warm-up or IDLE/SLEEP mode, and automatically restarts (continues counting) when the STOP/IDLE/SLEEP mode is inactivated.

Note: The watchdog timer consists of an internal divider and a two-stage binary counter. When the clear code 4EH is written, only the binary counter is cleared, but not the internal divider. The minimum binary-counter overflow time, that depends on the timing at which the clear code (4EH) is written to the WDTCR2 register, may be 3/4 of the time set in WDTCR1<WDTT>. Therefore, write the clear code using a cycle shorter than 3/4 of the time set to WDTCR1<WDTT>.

Example :Setting the watchdog timer detection time to $2^{21}/f_c$ [s], and resetting the CPU malfunction detection

	LD	(WDTCR2), 4EH	: Clears the binary counters.
	LD	(WDTCR1), 00001101B	: WDTT ← 10, WDTOUT ← 1
Within 3/4 of WDT detection time	┌	LD	(WDTCR2), 4EH : Clears the binary counters (always clears immediately before and after changing WDTT).
		:	
		:	
Within 3/4 of WDT detection time	└	LD	(WDTCR2), 4EH : Clears the binary counters.
		:	
		LD	(WDTCR2), 4EH : Clears the binary counters.

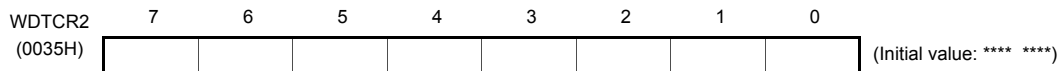
Watchdog Timer Control Register 1



WDTEN	Watchdog timer enable/disable	0: Disable (Writing the disable code to WDTCR2 is required.) 1: Enable	Write only	
WDTT	Watchdog timer detection time [s]	NORMAL 1/2 mode	SLOW 1/2 mode Write only	
		DV7CK = 0		DV7CK = 1
		DV1CK=0		DV1CK=1
		00		$2^{25}/fc$ $2^{26}/fc$ $2^{17}/fs$ $2^{17}/fs$ $2^{17}/fs$
		01		$2^{23}/fc$ $2^{24}/fc$ $2^{15}/fs$ $2^{15}/fs$ $2^{15}/fs$
10	$2^{21}/fc$ $2^{22}/fc$ $2^{13}/fs$ $2^{13}/fs$ $2^{13}/fs$			
11	$2^{19}/fc$ $2^{20}/fc$ $2^{11}/fs$ $2^{11}/fs$ $2^{11}/fs$			
WDTOUT	Watchdog timer output select	0: Interrupt request 1: Reset request	Write only	

- Note 1: After clearing WDTOUT to “0”, the program cannot set it to “1”.
- Note 2: fc: High-frequency clock [Hz], fs: Low-frequency clock [Hz], *: Don't care
- Note 3: WDTCR1 is a write-only register and must not be used with any of read-modify-write instructions. If WDTCR1 is read, a don't care is read.
- Note 4: To activate the STOP mode, disable the watchdog timer or clear the counter immediately before entering the STOP mode. After clearing the counter, clear the counter again immediately after the STOP mode is inactivated.
- Note 5: To clear WDTCR1, set the register in accordance with the procedures shown in “1.2.3 Watchdog Timer Disable”.

Watchdog Timer Control Register 2



WDTCR2	Write Watchdog timer control code	4EH: Clear the watchdog timer binary counter (Clear code) B1H: Disable the watchdog timer (Disable code) D2H: Enable assigning address trap area Others: Invalid	Write only
--------	-----------------------------------	---	------------

- Note 1: The disable code is valid only when WDTCR1<WDTEN> = 0.
- Note 2: *: Don't care
- Note 3: The binary counter of the watchdog timer must not be cleared by the interrupt task.
- Note 4: Write the clear code 4EH using a cycle shorter than 3/4 of the time set in WDTCR1<WDTT>.

6.2.2 Watchdog Timer Enable

Setting WDTCR1<WDTEN> to “1” enables the watchdog timer. Since WDTCR1<WDTEN> is initialized to “1” during reset, the watchdog timer is enabled automatically after the reset release.

6.2.3 Watchdog Timer Disable

To disable the watchdog timer, set the register in accordance with the following procedures. Setting the register in other procedures causes a malfunction of the microcontroller.

1. Set the interrupt master flag (IMF) to “0”.
2. Set WDTCR2 to the clear code (4EH).
3. Set WDTCR1<WDTEN> to “0”.
4. Set WDTCR2 to the disable code (B1H).

Note: While the watchdog timer is disabled, the binary counters of the watchdog timer are cleared.

Example :Disabling the watchdog timer

```
DI                : IMF ← 0
LD                (WDTCR2), 04EH    : Clears the binary coutner
LDW              (WDTCR1), 0B101H   : WDTEN ← 0, WDTCR2 ← Disable code
```

Table 6-1 Watchdog Timer Detection Time (Example: fc = 16.0 MHz, fs = 32.768 kHz)

WDTT	Watchdog Timer Detection Time[s]				
	NORMAL1/2 mode				SLOW mode
	DV7CK = 0		DV7CK = 1		
	DV1CK = 0	DV1CK = 1	DV1CK = 0	DV1CK = 1	
00	2.097	4.194	4	4	4
01	524.288 m	1.049	1	1	1
10	131.072 m	262.144 m	250 m	250 m	250 m
11	32.768 m	65.536 m	62.5 m	62.5 m	62.5 m

6.2.4 Watchdog Timer Interrupt (INTWDT)

When WDTCR1<WDTOUT> is cleared to “0”, a watchdog timer interrupt request (INTWDT) is generated by the binary-counter overflow.

A watchdog timer interrupt is the non-maskable interrupt which can be accepted regardless of the interrupt master flag (IMF).

When a watchdog timer interrupt is generated while the other interrupt including a watchdog timer interrupt is already accepted, the new watchdog timer interrupt is processed immediately and the previous interrupt is held pending. Therefore, if watchdog timer interrupts are generated continuously without execution of the RETN instruction, too many levels of nesting may cause a malfunction of the microcontroller.

To generate a watchdog timer interrupt, set the stack pointer before setting WDTCR1<WDTOUT>.

Example :Setting watchdog timer interrupt

```
LD                SP, 083FH        : Sets the stack pointer
LD                (WDTCR1), 00001000B : WDTOUT ← 0
```


6.2.5 Watchdog Timer Reset

When a binary-counter overflow occurs while WDTCR1<WDTOUT> is set to “1”, a watchdog timer reset request is generated. When a watchdog timer reset request is generated, the internal hardware is reset. The reset time is maximum $24/fc$ [s] ($1.5 \mu s$ @ $fc = 16.0$ MHz).

Note: When a watchdog timer reset is generated in the SLOW1 mode, the reset time is maximum $24/fc$ (high-frequency clock) since the high-frequency clock oscillator is restarted. However, when crystals have inaccuracies upon start of the high-frequency clock oscillator, the reset time should be considered as an approximate value because it has slight errors.

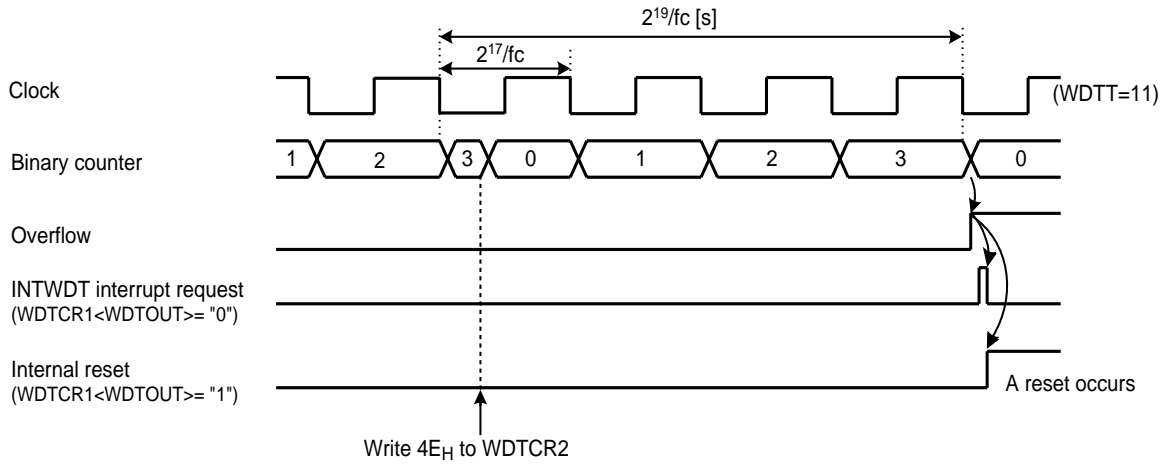
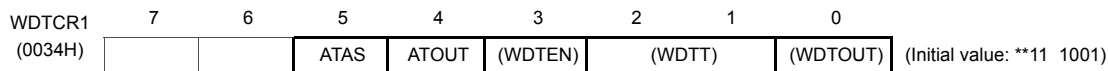


Figure 6-2 Watchdog Timer Interrupt

6.3 Address Trap

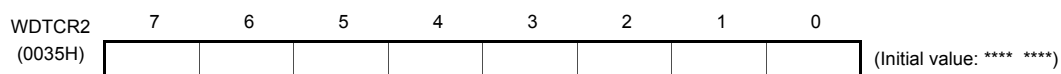
The Watchdog Timer Control Register 1 and 2 share the addresses with the control registers to generate address traps.

Watchdog Timer Control Register 1



ATAS	Select address trap generation in the internal RAM area	0: Generate no address trap 1: Generate address traps (After setting ATAS to "1", writing the control code D2H to WDTCR2 is required)	Write only
ATOUT	Select operation at address trap	0: Interrupt request 1: Reset request	

Watchdog Timer Control Register 2



WDTCR2	Write Watchdog timer control code and address trap area control code	D2H: Enable address trap area selection (ATRAP control code) 4EH: Clear the watchdog timer binary counter (WDT clear code) B1H: Disable the watchdog timer (WDT disable code) Others: Invalid	Write only
--------	--	--	------------

6.3.1 Selection of Address Trap in Internal RAM (ATAS)

WDTCR1<ATAS> specifies whether or not to generate address traps in the internal RAM area. To execute an instruction in the internal RAM area, clear WDTCR1<ATAS> to "0". To enable the WDTCR1<ATAS> setting, set WDTCR1<ATAS> and then write D2H to WDTCR2.

Executing an instruction in the SFR or DBR area generates an address trap unconditionally regardless of the setting in WDTCR1<ATAS>.

6.3.2 Selection of Operation at Address Trap (ATOUT)

When an address trap is generated, either the interrupt request or the reset request can be selected by WDTCR1<ATOUT>.

6.3.3 Address Trap Interrupt (INTATRAP)

While WDTCR1<ATOUT> is "0", if the CPU should start looping for some cause such as noise and an attempt be made to fetch an instruction from the on-chip RAM (while WDTCR1<ATAS> is "1"), DBR or the SFR area, address trap interrupt (INTATRAP) will be generated.

An address trap interrupt is a non-maskable interrupt which can be accepted regardless of the interrupt master flag (IMF).

When an address trap interrupt is generated while the other interrupt including a watchdog timer interrupt is already accepted, the new address trap is processed immediately and the previous interrupt is held pending. Therefore, if address trap interrupts are generated continuously without execution of the RETN instruction, too many levels of nesting may cause a malfunction of the microcontroller.

To generate address trap interrupts, set the stack pointer beforehand.

6.3.4 Address Trap Reset

While WDTCR1<ATOOUT> is “1”, if the CPU should start looping for some cause such as noise and an attempt be made to fetch an instruction from the on-chip RAM (while WDTCR1<ATAS> is “1”), DBR or the SFR area, address trap reset will be generated.

When an address trap reset request is generated, the internal hardware is reset. The reset time is maximum $24/f_c$ [s] ($1.5 \mu\text{s}$ @ $f_c = 16.0 \text{ MHz}$).

Note: When an address trap reset is generated in the SLOW1 mode, the reset time is maximum $24/f_c$ (high-frequency clock) since the high-frequency clock oscillator is restarted. However, when crystals have inaccuracies upon start of the high-frequency clock oscillator, the reset time should be considered as an approximate value because it has slight errors.

7. Time Base Timer (TBT)

The time base timer generates time base for key scanning, dynamic displaying, etc. It also provides a time base timer interrupt (INTTBT).

7.1 Time Base Timer

7.1.1 Configuration

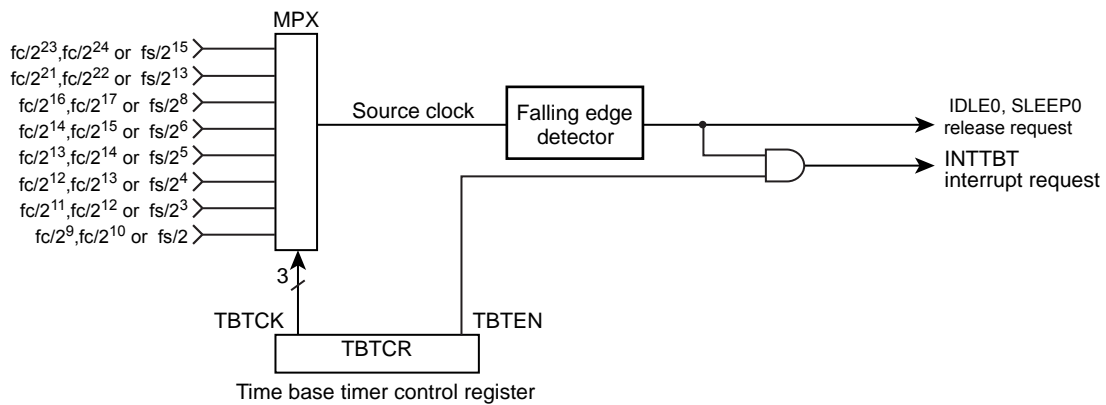


Figure 7-1 Time Base Timer configuration

7.1.2 Control

Time Base Timer is controlled by Time Base Timer control register (TBTCR).

Time Base Timer Control Register

	7	6	5	4	3	2	1	0	
TBTCR (0036H)	(DVOEN)	(DVOCK)	(DV7CK)	TBTEN	TBTC				(Initial Value: 0000 0000)

TBTEN	Time Base Timer Enable / Disable	0: Disable 1: Enable						
TBTC	Time Base Timer interrupt Frequency select : [Hz]	NORMAL1/2, IDLE1/2 Mode				SLOW1/2 SLEEP1/2 Mode	R/W	
		DV7CK = 0		DV7CK = 1				
		DV1CK=0	DV1CK=1	DV1CK=0	DV1CK=1			
		000	fc/2 ²³	fc/2 ²⁴	fs/2 ¹⁵	fs/2 ¹⁵		fs/2 ¹⁵
		001	fc/2 ²¹	fc/2 ²²	fs/2 ¹³	fs/2 ¹³		fs/2 ¹³
		010	fc/2 ¹⁶	fc/2 ¹⁷	fs/2 ⁸	fs/2 ⁸		—
		011	fc/2 ¹⁴	fc/2 ¹⁵	fs/2 ⁶	fs/2 ⁶		—
		100	fc/2 ¹³	fc/2 ¹⁴	fs/2 ⁵	fs/2 ⁵		—
		101	fc/2 ¹²	fc/2 ¹³	fs/2 ⁴	fs/2 ⁴		—
110	fc/2 ¹¹	fc/2 ¹²	fs/2 ³	fs/2 ³	—			
111	fc/2 ⁹	fc/2 ¹⁰	fs/2	fs/2	—			

Note 1: fc; High-frequency clock [Hz], fs; Low-frequency clock [Hz], *; Don't care

Note 2: The interrupt frequency (TBTCK) must be selected with the time base timer disabled (TBTEN="0"). (The interrupt frequency must not be changed with the disable from the enable state.) Both frequency selection and enabling can be performed simultaneously.

Example :Set the time base timer frequency to $fc/2^{16}$ [Hz] and enable an INTTBT interrupt.

```
LD      (TBTCR) , 00000010B      ; TBTCK ← 010
LD      (TBTCR) , 00001010B      ; TBTEN ← 1
DI                               ; IMF ← 0
SET     (EIRH) . 0
```

Table 7-1 Time Base Timer Interrupt Frequency (Example : $fc = 16.0$ MHz, $fs = 32.768$ kHz)

TBTCK	Time Base Timer Interrupt Frequency [Hz]				
	NORMAL1/2, IDLE1/2 Mode		NORMAL1/2, IDLE1/2 Mode		SLOW1/2, SLEEP1/2 Mode
	DV7CK = 0		DV7CK = 1		
	DV1CK = 0	DV1CK = 1	DV1CK = 0	DV1CK = 1	
000	1.91	0.95	1	1	1
001	7.63	3.81	4	4	4
010	244.14	122.07	128	128	-
011	976.56	488.28	512	512	-
100	1953.13	976.56	1024	1024	-
101	3906.25	1953.13	2048	2048	-
110	7812.5	3906.25	4096	4096	-
111	31250	15625	16384	16384	-

7.1.3 Function

An INTTBT (Time Base Timer Interrupt) is generated on the first falling edge of source clock (The divider output of the timing generato which is selected by TBTCK.) after time base timer has been enabled.

The divider is not cleared by the program; therefore, only the first interrupt may be generated ahead of the set interrupt period (Figure 7-2).

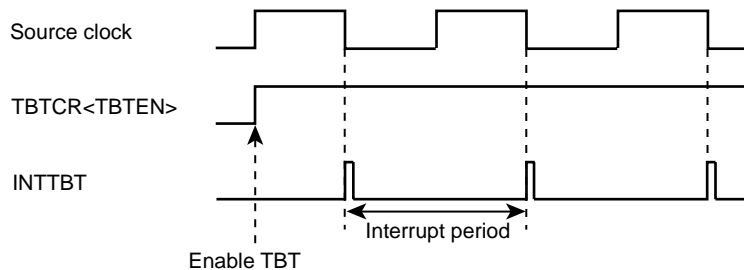


Figure 7-2 Time Base Timer Interrupt

7.2 Divider Output (\overline{DVO})

Approximately 50% duty pulse can be output using the divider output circuit, which is useful for piezoelectric buzzer drive. Divider output is from \overline{DVO} pin.

7.2.1 Configuration

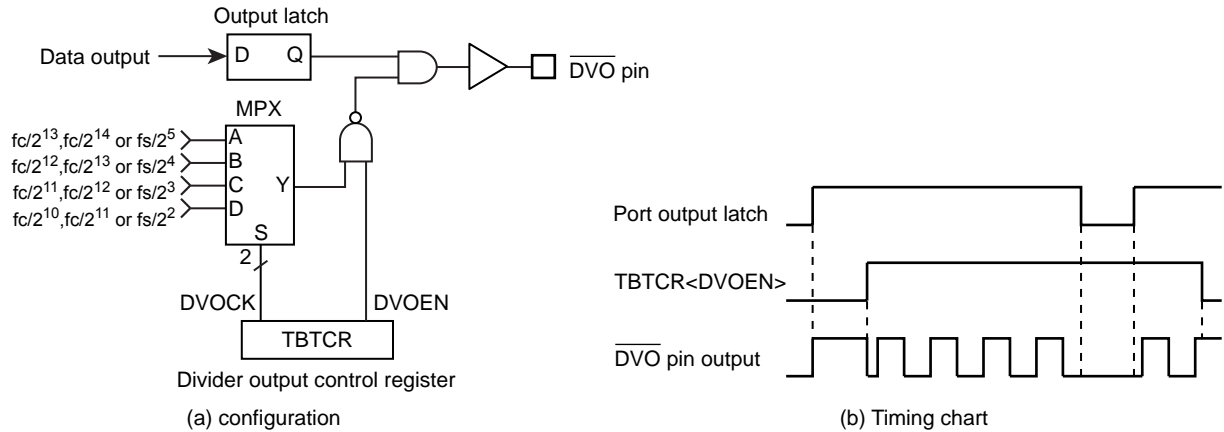
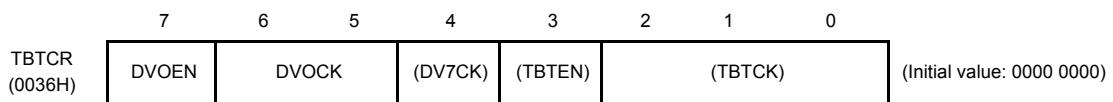


Figure 7-3 Divider Output

7.2.2 Control

The Divider Output is controlled by the Time Base Timer Control Register.

Time Base Timer Control Register



DVOEN	Divider output enable / disable	0: Disable 1: Enable				R/W	
		NORMAL 1/2, IDLE 1/2 Mode					
DVOCK	Divider Output (\overline{DVO}) frequency selection: [Hz]	DV7CK=0		DV7CK=1		SLOW 1/2 SLEEP 1/2 Mode	
		DV1CK = 0	DV1CK = 1	DV1CK = 0	DV1CK = 1		
		00	$fc/2^{13}$	$fc/2^{14}$	$fs/2^5$	$fs/2^5$	$fs/2^5$
		01	$fc/2^{12}$	$fc/2^{13}$	$fs/2^4$	$fs/2^4$	$fs/2^4$
		10	$fc/2^{11}$	$fc/2^{12}$	$fs/2^3$	$fs/2^3$	$fs/2^3$
11	$fc/2^{10}$	$fc/2^{11}$	$fs/2^2$	$fs/2^2$	$fs/2^2$		

Note: Selection of divider output frequency (DVOCK) must be made while divider output is disabled (DVOEN="0"). Also, in other words, when changing the state of the divider output frequency from enabled (DVOEN="1") to disabled(DVOEN="0"), do not change the setting of the divider output frequency.

Example : 1.95 kHz pulse output (fc = 16.0 MHz)

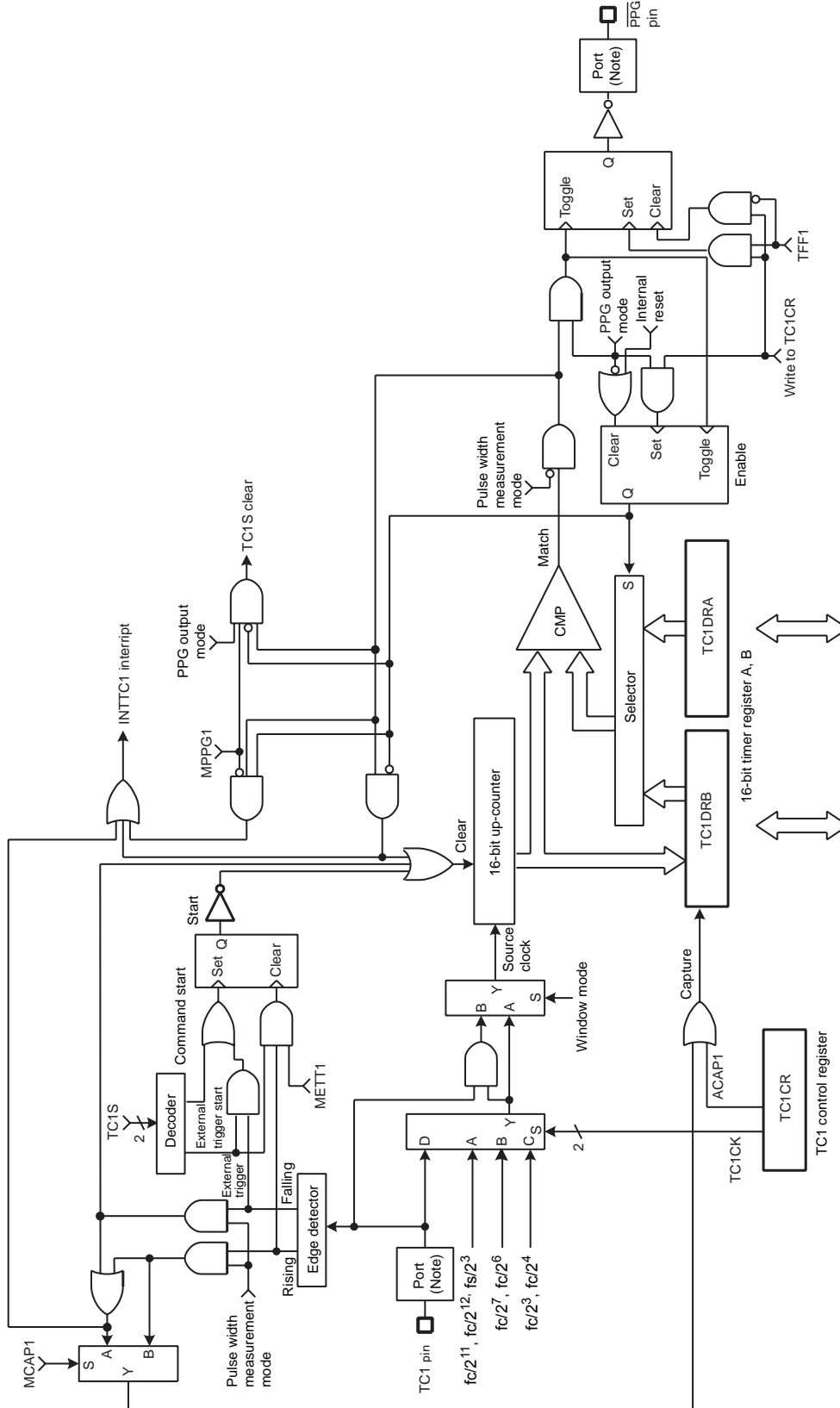
```
LD      (TBTCR), 00000000B      ; DVOCK ← "00"
LD      (TBTCR), 10000000B      ; DVOEN ← "1"
```

Table 7-2 Divider Output Frequency (Example : fc = 16.0 MHz, fs = 32.768 kHz)

DVOCK	Divider Output Frequency [Hz]				
	NORMAL1/2, IDLE1/2 Mode				SLOW1/2, SLEEP1/2 Mode
	DV7CK = 0		DV7CK = 1		
	DV1CK=0	DV1CK=1	DV1CK=0	DV1CK=1	
00	1.953 k	976.6	1.024 k	1.024 k	1.024 k
01	3.906 k	1.953 k	2.048 k	2.048 k	2.048 k
10	7.813 k	3.906 k	4.096 k	4.096 k	4.096 k
11	15.625 k	7.813 k	8.192 k	8.192 k	8.192 k

8. 16-Bit TimerCounter 1 (TC1)

8.1 Configuration



Note: Function I/O may not operate depending on I/O port setting. For more details, see the chapter "I/O Port".

Figure 8-1 TimerCounter 1 (TC1)

8.2 TimerCounter Control

The TimerCounter 1 is controlled by the TimerCounter 1 control register (TC1CR) and two 16-bit timer registers (TC1DRA and TC1DRB).

Timer Register

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TC1DRA (0021H, 0020H)	TC1DRAH (0021H)								TC1DRAL (0020H)							
	(Initial value: 1111 1111 1111 1111)								Read/Write							
TC1DRB (0023H, 0022H)	TC1DRBH (0023H)								TC1DRBL (0022H)							
	(Initial value: 1111 1111 1111 1111)								Read/Write (Write enabled only in the PPG output mode)							

TimerCounter 1 Control Register

	7	6	5	4	3	2	1	0
TC1CR (0032H)	TFF1	ACAP1 MCAP1 METT1 MPPG1	TC1S	TC1CK	TC1M	Read/Write (Initial value: 0000 0000)		

TFF1	Timer F/F1 control	0: Clear	1: Set					R/W		
ACAP1	Auto capture control	0:Auto-capture disable	1:Auto-capture enable					R/W		
MCAP1	Pulse width measurement mode control	0:Double edge capture	1:Single edge capture							
METT1	External trigger timer mode control	0:Trigger start	1:Trigger start and stop							
MPPG1	PPG output control	0:Continuous pulse generation	1:One-shot							
TC1S	TC1 start control			Timer	Extrig-ger	Event	Win-dow		Pulse	PPG
		00: Stop and counter clear		0	0	0	0	0	0	
		01: Command start		0	-	-	-	-	0	
		10: Rising edge start (Ex-trigger/Pulse/PPG) Rising edge count (Event) Positive logic count (Window)		-	0	0	0	0	0	
		11: Falling edge start (Ex-trigger/Pulse/PPG) Falling edge count (Event) Negative logic count (Window)		-	0	0	0	0	0	
TC1CK	TC1 source clock select [Hz]	NORMAL1/2, IDLE1/2 mode						Divider	SLOW, SLEEP mode	
		DV7CK = 0			DV7CK = 1					
			DV1CK = 0	DV1CK = 1	DV1CK = 0	DV1CK = 1				
		00	$fc/2^{11}$	$fc/2^{12}$	$fs/2^3$	$fs/2^3$	DV9	$fs/2^3$		
		01	$fc/2^7$	$fc/2^8$	$fc/2^7$	$fc/2^8$	DV5	-		
10	$fc/2^3$	$fc/2^4$	$fc/2^3$	$fc/2^4$	DV1	-				
		External clock (TC1 pin input)								
TC1M	TC1 operating mode select	00: Timer/external trigger timer/event counter mode 01: Window mode 10: Pulse width measurement mode 11: PPG (Programmable pulse generate) output mode						R/W		

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

Note 2: The timer register consists of two shift registers. A value set in the timer register becomes valid at the rising edge of the first source clock pulse that occurs after the upper byte (TC1DRAH and TC1DRBH) is written. Therefore, write the lower byte and the upper byte in this order (it is recommended to write the register with a 16-bit access instruction). Writing only the lower byte (TC1DRAL and TC1DRBL) does not enable the setting of the timer register.

- Note 3: To set the mode, source clock, PPG output control and timer F/F control, write to TC1CR1 during TC1S=00. Set the timer F/F1 control until the first timer start after setting the PPG mode.
- Note 4: Auto-capture can be used only in the timer, event counter, and window modes.
- Note 5: To set the timer registers, the following relationship must be satisfied.
TC1DRA > TC1DRB > 1 (PPG output mode), TC1DRA > 1 (other modes)
- Note 6: Set TFF1 to "0" in the mode except PPG output mode.
- Note 7: Set TC1DRB after setting TC1M to the PPG output mode.
- Note 8: When the STOP mode is entered, the start control (TC1S) is cleared to "00" automatically, and the timer stops. After the STOP mode is exited, set the TC1S to use the timer counter again.
- Note 9: Use the auto-capture function in the operative condition of TC1. A captured value may not be fixed if it's read after the execution of the timer stop or auto-capture disable. Read the capture value in a capture enabled condition.
- Note 10: Since the up-counter value is captured into TC1DRB by the source clock of up-counter after setting TC1CR<ACAP1> to "1". Therefore, to read the captured value, wait at least one cycle of the internal source clock before reading TC1DRB for the first time.

8.3 Function

TimerCounter 1 has six types of operating modes: timer, external trigger timer, event counter, window, pulse width measurement, programmable pulse generator output modes.

8.3.1 Timer mode

In the timer mode, the up-counter counts up using the internal clock. When a match between the up-counter and the timer register 1A (TC1DRA) value is detected, an INTTC1 interrupt is generated and the up-counter is cleared. After being cleared, the up-counter restarts counting. Setting TC1CR<ACAP1> to "1" captures the up-counter value into the timer register 1B (TC1DRB) with the auto-capture function. Use the auto-capture function in the operative condition of TC1. A captured value may not be fixed if it's read after the execution of the timer stop or auto-capture disable. Read the capture value in a capture enabled condition. Since the up-counter value is captured into TC1DRB by the source clock of up-counter after setting TC1CR<ACAP1> to "1". Therefore, to read the captured value, wait at least one cycle of the internal source clock before reading TC1DRB for the first time.

Table 8-1 Internal Source Clock for TimerCounter 1 (Example: fc = 16 MHz, fs = 32.768 kHz)

TC1CK	NORMAL1/2, IDLE1/2 mode								SLOW, SLEEP mode	
	DV7CK = 0				DV7CK = 1					
	DV1CK = 0		DV1CK = 1		DV1CK = 0		DV1CK = 1		Resolution [μs]	Maximum Time Setting [s]
	Resolution [μs]	Maximum Time Setting [s]	Resolution [μs]	Maximum Time Setting [s]	Resolution [μs]	Maximum Time Setting [s]	Resolution [μs]	Maximum Time Setting [s]		
00	128	8.39	256	16.78	244.14	16.0	244.14	16.0	244.14	16.0
01	8.0	0.524	16	1.05	8.0	0.524	16.0	0.838	–	–
10	0.5	32.77 m	1	65.53 m	0.5	32.77 m	1.0	52.42 m	–	–

Example 1 :Setting the timer mode with source clock $fc/2^{11}$ [Hz] and generating an interrupt 1 second later (fc = 16 MHz, TBTCR<DV7CK> = "0", CGCR<DV1CK> = "0")

```
LDW      (TC1DRA), 1E84H      ; Sets the timer register (1 s ÷ 211/fc = 1E84H)
DI                                              ; IMF= "0"
SET      (EIRH), 1           ; Enables INTTC1
EI                                              ; IMF= "1"
LD       (TC1CR), 00000000B    ; Selects the source clock and mode
LD       (TC1CR), 00010000B    ; Starts TC1
```

Example 2 :Auto-capture

```
LD       (TC1CR), 01010000B    ; ACAP1 ← 1
:       :
LD       WA, (TC1DRB)         ; Reads the capture value
```

Note: Since the up-counter value is captured into TC1DRB by the source clock of up-counter after setting TC1CR<ACAP1> to "1". Therefore, to read the captured value, wait at least one cycle of the internal source clock before reading TC1DRB for the first time.

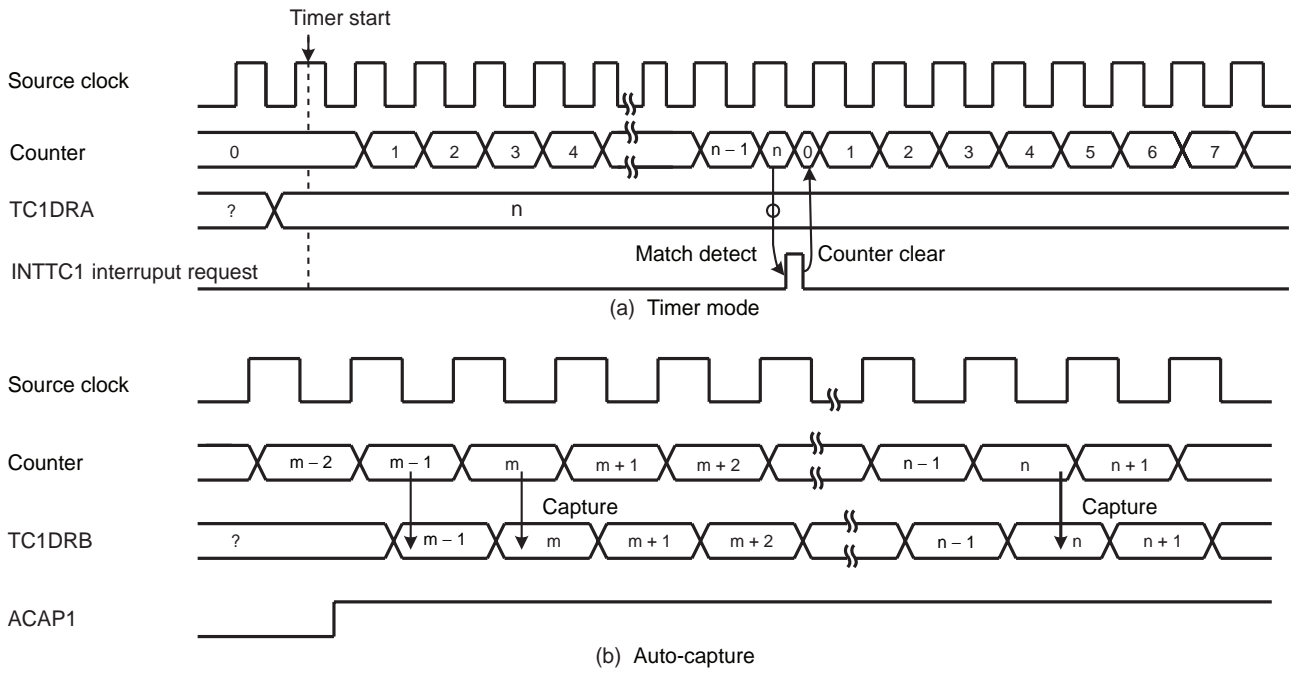


Figure 8-2 Timer Mode Timing Chart

8.3.2 External Trigger Timer Mode

In the external trigger timer mode, the up-counter starts counting by the input pulse triggering of the TC1 pin, and counts up at the edge of the internal clock. For the trigger edge used to start counting, either the rising or falling edge is defined in TC1CR<TCIS>.

- When TC1CR<METT1> is set to “1” (trigger start and stop)

When a match between the up-counter and the TC1DRA value is detected after the timer starts, the up-counter is cleared and halted and an INTTC1 interrupt request is generated.

If the edge opposite to trigger edge is detected before detecting a match between the up-counter and the TC1DRA, the up-counter is cleared and halted without generating an interrupt request. Therefore, this mode can be used to detect exceeding the specified pulse by interrupt.

After being halted, the up-counter restarts counting when the trigger edge is detected.

- When TC1CR<METT1> is set to “0” (trigger start)

When a match between the up-counter and the TC1DRA value is detected after the timer starts, the up-counter is cleared and halted and an INTTC1 interrupt request is generated.

The edge opposite to the trigger edge has no effect in count up. The trigger edge for the next counting is ignored if detecting it before detecting a match between the up-counter and the TC1DRA.

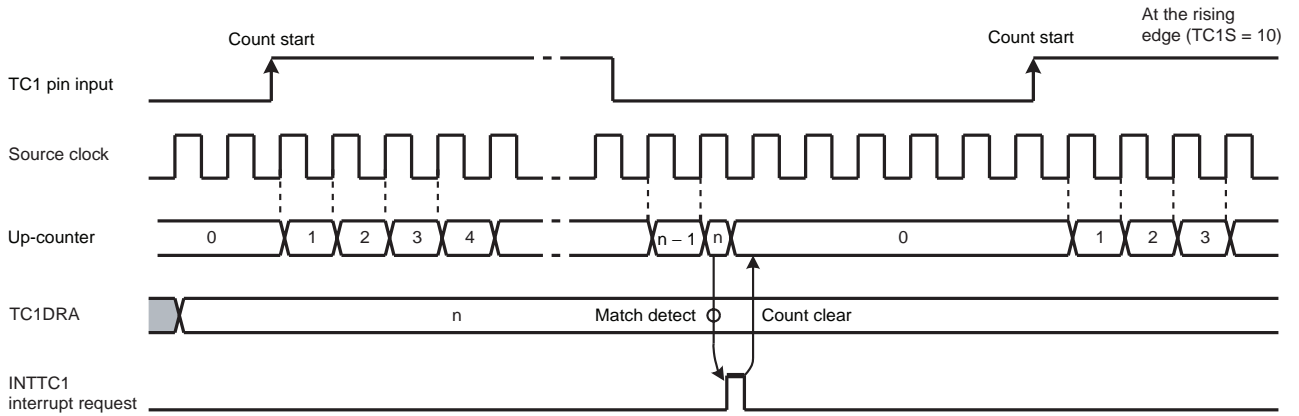
Since the TC1 pin input has the noise rejection, pulses of $4/f_c$ [s] or less are rejected as noise. A pulse width of $12/f_c$ [s] or more is required to ensure edge detection. The rejection circuit is turned off in the SLOW1/2 or SLEEP1/2 mode, but a pulse width of one machine cycle or more is required.

Example 1 :Generating an interrupt 1 ms after the rising edge of the input pulse to the TC1 pin
($f_c = 16$ MHz, CGCR<DV1CK> = “0”)

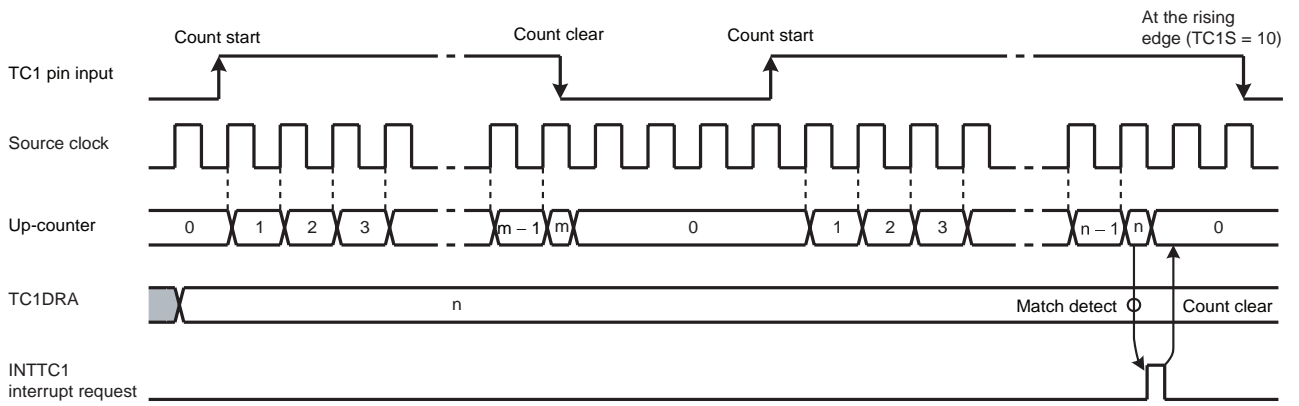
```
LDW      (TC1DRA), 007DH      ; 1ms ÷ 27/fc = 7DH
DI                          ; IMF= “0”
SET      (EIRH), 1           ; Enables INTTC1 interrupt
EI                          ; IMF= “1”
LD       (TC1CR), 00000100B   ; Selects the source clock and mode
LD       (TC1CR), 00100100B   ; Starts TC1 external trigger, METT1 = 0
```

Example 2 :Generating an interrupt when the low-level pulse with 4 ms or more width is input to the TC1 pin
($f_c = 16$ MHz, CGCR<DV1CK> = “0”)

```
LDW      (TC1DRA), 01F4H      ; 4 ms ÷ 27/fc = 1F4H
DI                          ; IMF= “0”
SET      (EIRH), 1           ; Enables INTTC1 interrupt
EI                          ; IMF= “1”
LD       (TC1CR), 00000100B   ; Selects the source clock and mode
LD       (TC1CR), 01110100B   ; Starts TC1 external trigger, METT1 = 0
```



(a) Trigger start (METT1 = 0)



Note: $m < n$

(b) Trigger start and stop (METT1 = 1)

Figure 8-3 External Trigger Timer Mode Timing Chart

8.3.3 Event Counter Mode

In the event counter mode, the up-counter counts up at the edge of the input pulse to the TC1 pin. Either the rising or falling edge of the input pulse is selected as the count up edge in TC1CR<TC1S>.

When a match between the up-counter and the TC1DRA value is detected, an INTTC1 interrupt is generated and the up-counter is cleared. After being cleared, the up-counter restarts counting at each edge of the input pulse to the TC1 pin. Since a match between the up-counter and the value set to TC1DRA is detected at the edge opposite to the selected edge, an INTTC1 interrupt request is generated after a match of the value at the edge opposite to the selected edge.

Two or more machine cycles are required for the low-or high-level pulse input to the TC1 pin.

Setting TC1CR<ACAP1> to "1" captures the up-counter value into TC1DRB with the auto capture function. Use the auto-capture function in the operative condition of TC1. A captured value may not be fixed if it's read after the execution of the timer stop or auto-capture disable. Read the capture value in a capture enabled condition. Since the up-counter value is captured into TC1DRB by the source clock of up-counter after setting TC1CR<ACAP1> to "1". Therefore, to read the captured value, wait at least one cycle of the internal source clock before reading TC1DRB for the first time.

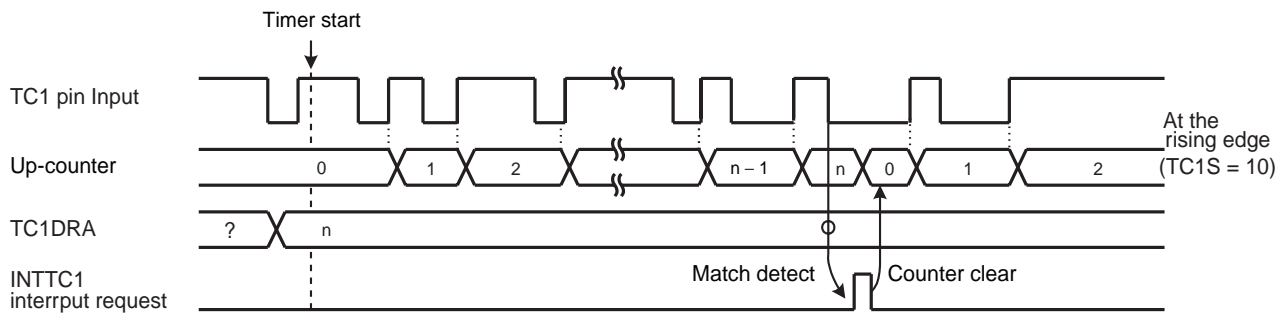


Figure 8-4 Event Counter Mode Timing Chart

Table 8-2 Input Pulse Width to TC1 Pin

	Minimum Pulse Width [s]	
	NORMAL1/2, IDLE1/2 Mode	SLOW1/2, SLEEP1/2 Mode
High-going	$2^3/f_c$	$2^3/f_s$
Low-going	$2^3/f_c$	$2^3/f_s$

8.3.4 Window Mode

In the window mode, the up-counter counts up at the rising edge of the pulse that is logical ANDed product of the input pulse to the TC1 pin (window pulse) and the internal source clock. Either the positive logic (count up during high-going pulse) or negative logic (count up during low-going pulse) can be selected.

When a match between the up-counter and the TC1DRA value is detected, an INTTC1 interrupt is generated and the up-counter is cleared.

Define the window pulse to the frequency which is sufficiently lower than the internal source clock programmed with TC1CR<TC1CK>.

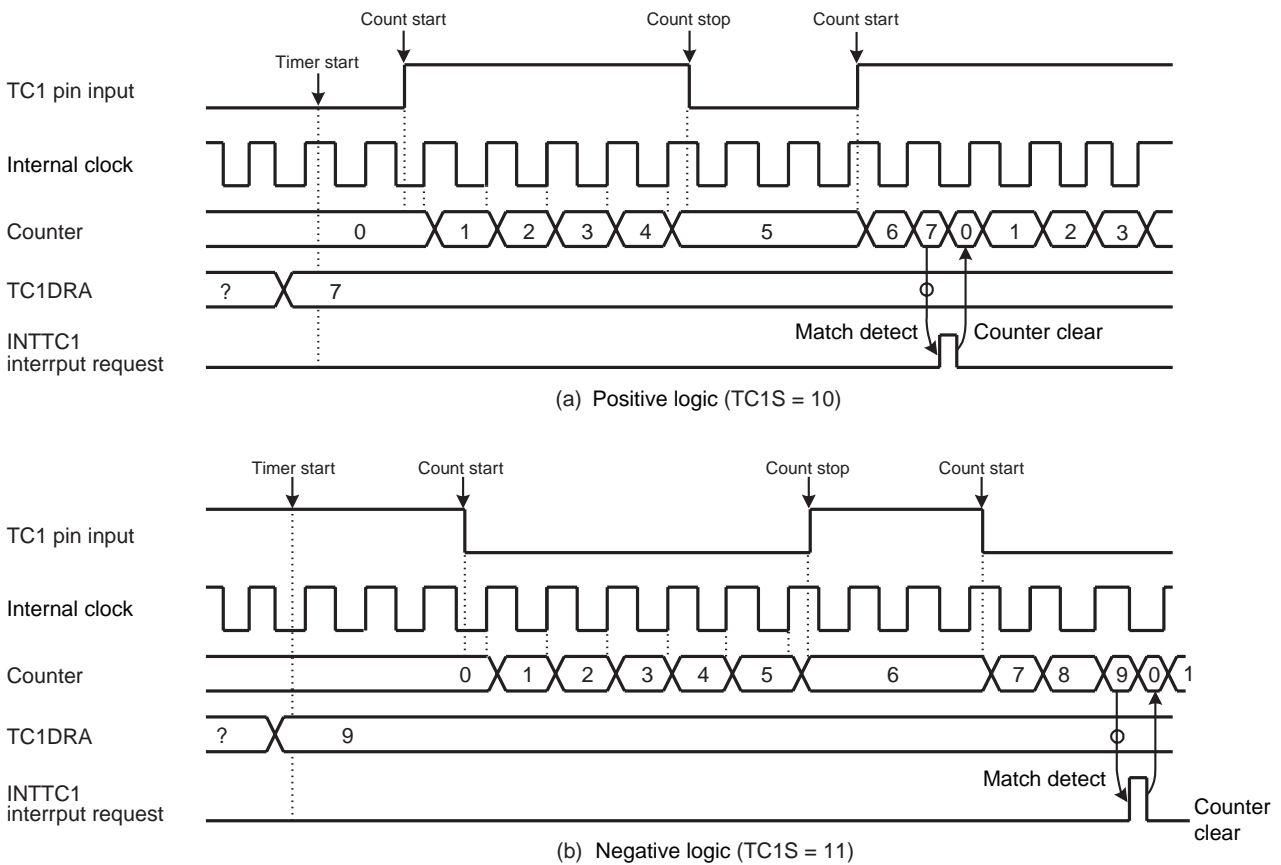


Figure 8-5 Window Mode Timing Chart

8.3.5 Pulse Width Measurement Mode

In the pulse width measurement mode, the up-counter starts counting by the input pulse triggering of the TC1 pin, and counts up at the edge of the internal clock. Either the rising or falling edge of the internal clock is selected as the trigger edge in TC1CR<TC1S>. Either the single- or double-edge capture is selected as the trigger edge in TC1CR<MCAP1>.

- When TC1CR<MCAP1> is set to “1” (single-edge capture)

Either high- or low-level input pulse width can be measured. To measure the high-level input pulse width, set the rising edge to TC1CR<TC1S>. To measure the low-level input pulse width, set the falling edge to TC1CR<TC1S>.

When detecting the edge opposite to the trigger edge used to start counting after the timer starts, the up-counter captures the up-counter value into TC1DRB and generates an INTTC1 interrupt request. The up-counter is cleared at this time, and then restarts counting when detecting the trigger edge used to start counting.

- When TC1CR<MCAP1> is set to “0” (double-edge capture)

The cycle starting with either the high- or low-going input pulse can be measured. To measure the cycle starting with the high-going pulse, set the rising edge to TC1CR<TC1S>. To measure the cycle starting with the low-going pulse, set the falling edge to TC1CR<TC1S>.

When detecting the edge opposite to the trigger edge used to start counting after the timer starts, the up-counter captures the up-counter value into TC1DRB and generates an INTTC1 interrupt request. The up-counter continues counting up, and captures the up-counter value into TC1DRB and generates an INTTC1 interrupt request when detecting the trigger edge used to start counting. The up-counter is cleared at this time, and then continues counting.

Note 1: The captured value must be read from TC1DRB until the next trigger edge is detected. If not read, the captured value becomes a don't care. It is recommended to use a 16-bit access instruction to read the captured value from TC1DRB.

Note 2: For the single-edge capture, the counter after capturing the value stops at “1” until detecting the next edge. Therefore, the second captured value is “1” larger than the captured value immediately after counting starts.

Note 3: The first captured value after the timer starts may be read incorrectly, therefore, ignore the first captured value.

Example :Duty measurement (resolution $fc/2^7$ [Hz], CGCR<DV1CK> = "0")

```

CLR      (INTTC1SW). 0      ; INTTC1 service switch initial setting
                          Address set to convert INTTC1SW at each INTTC1

LD       (TC1CR), 00000110B ; Sets the TC1 mode and source clock

DI       ; IMF= "0"

SET      (EIRH). 1         ; Enables INTTC1

EI       ; IMF= "1"

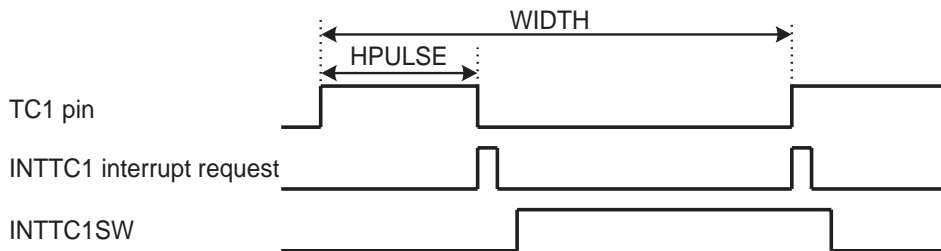
LD       (TC1CR), 00100110B ; Starts TC1 with an external trigger at MCAP1 = 0

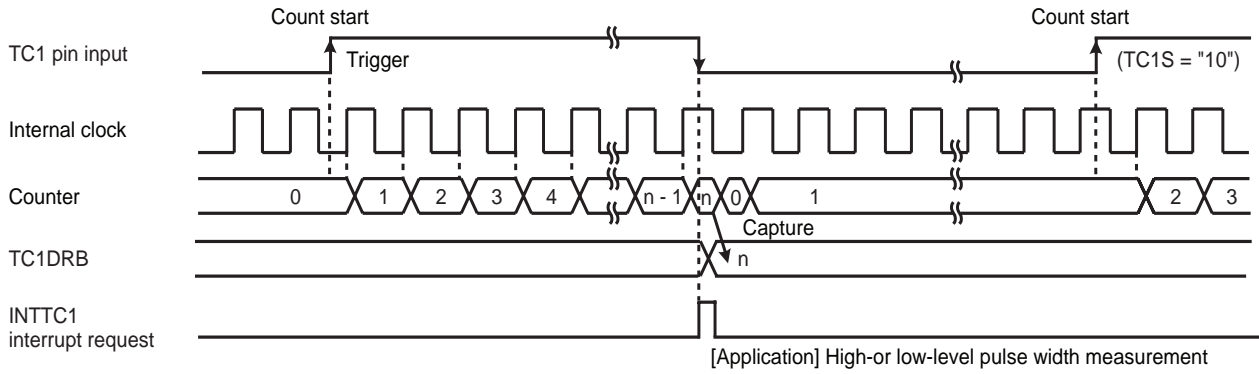
:

PINTTC1: CPL      (INTTC1SW). 0      ; INTTC1 interrupt, inverts and tests INTTC1 service switch
          JRS      F, SINTTC1
          LD       A, (TC1DRBL)      ; Reads TC1DRB (High-level pulse width)
          LD       W,(TC1DRBH)
          LD       (HPULSE), WA      ; Stores high-level pulse width in RAM
          RETI

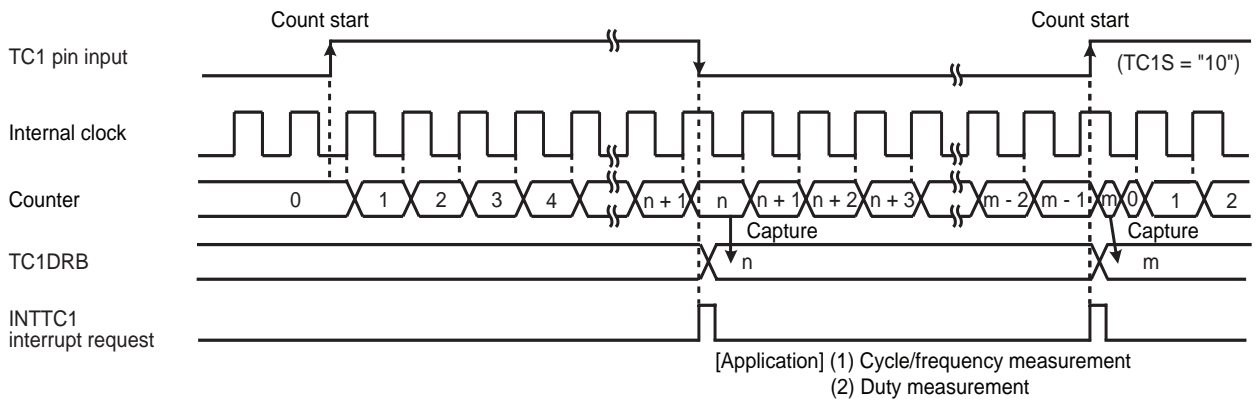
SINTTC1: LD       A, (TC1DRBL)      ; Reads TC1DRB (Cycle)
          LD       W,(TC1DRBH)
          LD       (WIDTH), WA      ; Stores cycle in RAM
          :
          RETI      ; Duty calculation
          :

VINTTC1: DW       PINTTC1          ; INTTC1 Interrupt vector
    
```





(a) Single-edge capture (MCAP1 = "1")



(b) Double-edge capture (MCAP1 = "0")

Figure 8-6 Pulse Width Measurement Mode

8.3.6 Programmable Pulse Generate (PPG) Output Mode

In the programmable pulse generation (PPG) mode, an arbitrary duty pulse is generated by counting performed in the internal clock. To start the timer, TC1CR<TC1S> specifies either the edge of the input pulse to the TC1 pin or the command start. TC1CR<MPPG1> specifies whether a duty pulse is produced continuously or not (one-shot pulse).

- When TC1CR<MPPG1> is set to “0” (Continuous pulse generation)

When a match between the up-counter and the TC1DRB value is detected after the timer starts, the level of the $\overline{\text{PPG}}$ pin is inverted and an INTTC1 interrupt request is generated. The up-counter continues counting. When a match between the up-counter and the TC1DRA value is detected, the level of the $\overline{\text{PPG}}$ pin is inverted and an INTTC1 interrupt request is generated. The up-counter is cleared at this time, and then continues counting and pulse generation.

When TC1S is cleared to “00” during PPG output, the $\overline{\text{PPG}}$ pin retains the level immediately before the counter stops.

- When TC1CR<MPPG1> is set to “1” (One-shot pulse generation)

When a match between the up-counter and the TC1DRB value is detected after the timer starts, the level of the $\overline{\text{PPG}}$ pin is inverted and an INTTC1 interrupt request is generated. The up-counter continues counting. When a match between the up-counter and the TC1DRA value is detected, the level of the $\overline{\text{PPG}}$ pin is inverted and an INTTC1 interrupt request is generated. TC1CR<TC1S> is cleared to “00” automatically at this time, and the timer stops. The pulse generated by PPG retains the same level as that when the timer stops.

Since the output level of the $\overline{\text{PPG}}$ pin can be set with TC1CR<TFF1> when the timer starts, a positive or negative pulse can be generated. Since the inverted level of the timer F/F1 output level is output to the $\overline{\text{PPG}}$ pin, specify TC1CR<TFF1> to “0” to set the high level to the $\overline{\text{PPG}}$ pin, and “1” to set the low level to the $\overline{\text{PPG}}$ pin. Upon reset, the timer F/F1 is initialized to “0”.

Note 1: To change TC1DRA or TC1DRB during a run of the timer, set a value sufficiently larger than the count value of the counter. Setting a value smaller than the count value of the counter during a run of the timer may generate a pulse different from that specified.

Note 2: Do not change TC1CR<TFF1> during a run of the timer. TC1CR<TFF1> can be set correctly only at initialization (after reset). When the timer stops during PPG, TC1CR<TFF1> can not be set correctly from this point onward if the PPG output has the level which is inverted of the level when the timer starts. (Setting TC1CR<TFF1> specifies the timer F/F1 to the level inverted of the programmed value.) Therefore, the timer F/F1 needs to be initialized to ensure an arbitrary level of the PPG output. To initialize the timer F/F1, change TC1CR<TC1M> to the timer mode (it is not required to start the timer mode), and then set the PPG mode. Set TC1CR<TFF1> at this time.

Note 3: In the PPG mode, the following relationship must be satisfied.
TC1DRA > TC1DRB

Note 4: Set TC1DRB after changing the mode of TC1M to the PPG mode.

Example :Generating a pulse which is high-going for 800 μ s and low-going for 200 μ s
($f_c = 16$ MHz, $CGCR\langle DV1CK \rangle = "0"$)

```

Setting port
LD      (TC1CR), 10000111B    ; Sets the PPG mode, selects the source clock
LDW     (TC1DRA), 007DH       ; Sets the cycle ( $1\text{ ms} \div 2^7/f_c\text{ ms} = 007DH$ )
LDW     (TC1DRB), 0019H       ; Sets the low-level pulse width ( $200\ \mu\text{s} \div 2^7/f_c = 0019H$ )
LD      (TC1CR), 10010111B    ; Starts the timer
    
```

Example :After stopping PPG, setting the PPG pin to a high-level to restart PPG
($f_c = 16$ MHz, $CGCR\langle DV1CK \rangle = "0"$)

```

Setting port
LD      (TC1CR), 10000111B    ; Sets the PPG mode, selects the source clock
LDW     (TC1DRA), 007DH       ; Sets the cycle ( $1\text{ ms} \div 2^7/f_c\ \mu\text{s} = 007DH$ )
LDW     (TC1DRB), 0019H       ; Sets the low-level pulse width ( $200\ \mu\text{s} \div 2^7/f_c = 0019H$ )
LD      (TC1CR), 10010111B    ; Starts the timer
:      :
LD      (TC1CR), 10000111B    ; Stops the timer
LD      (TC1CR), 10000100B    ; Sets the timer mode
LD      (TC1CR), 00000111B    ; Sets the PPG mode,  $TFF1 = 0$ 
LD      (TC1CR), 00010111B    ; Starts the timer
    
```

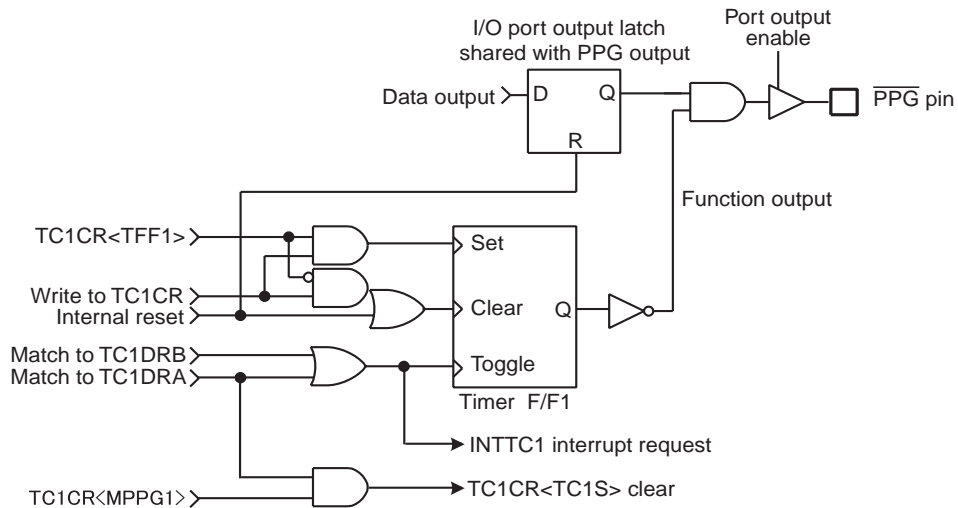
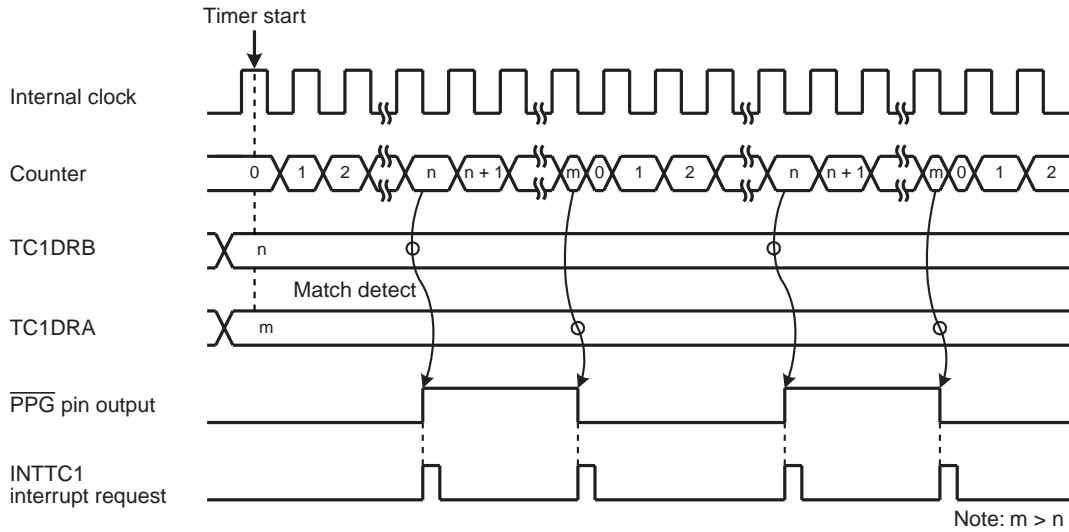
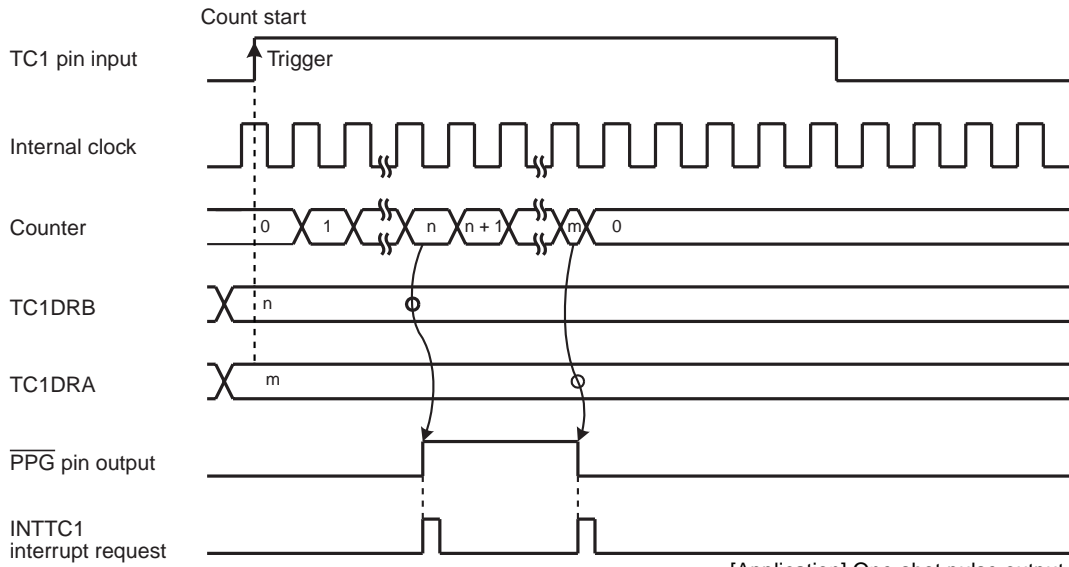


Figure 8-7 \overline{PPG} Output



(a) Continuous pulse generation (TC1S = 01)

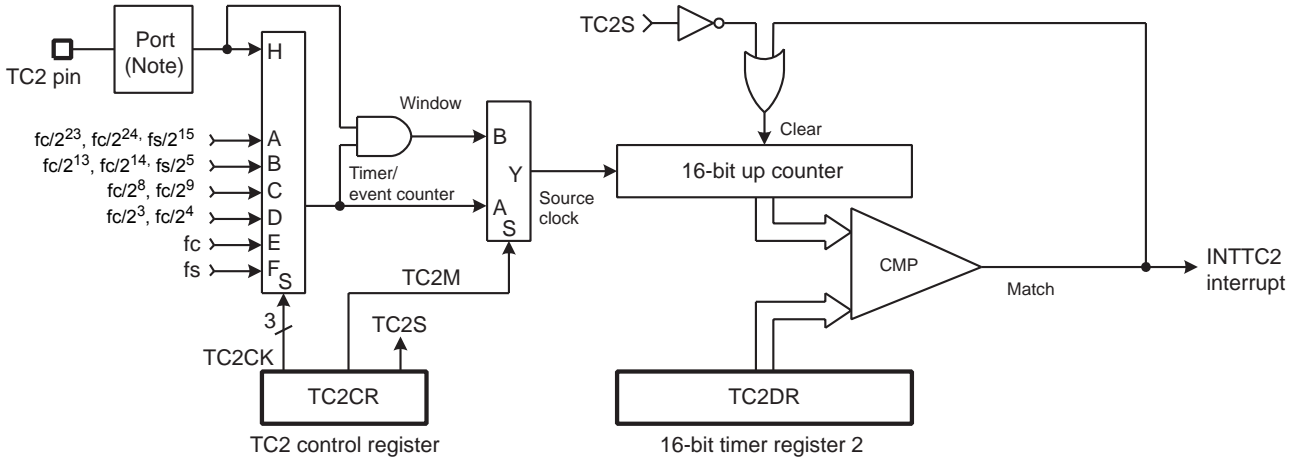


(b) One-shot pulse generation (TC1S = 10)

Figure 8-8 PPG Mode Timing Chart

9. 16-Bit Timer/Counter2 (TC2)

9.1 Configuration

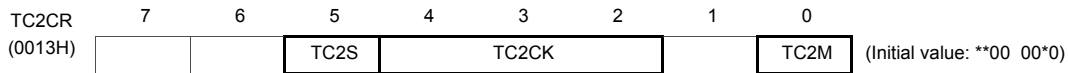
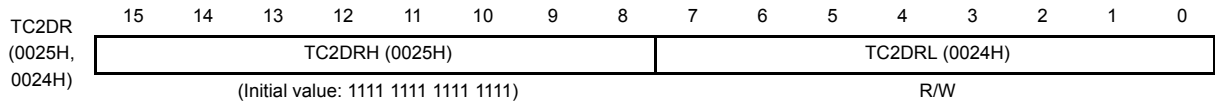


Note: When control input/output is used, I/O port setting should be set correctly. For details, refer to the section "I/O ports".

Figure 9-1 Timer/Counter2 (TC2)

9.2 Control

The timer/counter 2 is controlled by a timer/counter 2 control register (TC2CR) and a 16-bit timer register 2 (TC2DR).



TC2S	TC2 start control	0: Stop and counter clear 1: Start						R/W	
TC2CK	TC2 source clock select Unit : [Hz]	NORMAL1/2, IDLE1/2 mode				Divider	SLOW1/2 mode	SLEEP1/2 mode	
		DV7CK = 0		DV7CK = 1					
		DV1CK = 0	DV1CK = 1	DV1CK = 0	DV1CK = 1				
		000	$fc/2^{23}$	$fc/2^{24}$	$fs/2^{15}$	$fs/2^{15}$	DV21	$fs/2^{15}$	$fs/2^{15}$
		001	$fc/2^{13}$	$fc/2^{14}$	$fs/2^5$	$fs/2^5$	DV11	$fs/2^5$	$fs/2^5$
		010	$fc/2^8$	$fc/2^9$	$fc/2^8$	$fc/2^9$	DV6	-	-
		011	$fc/2^3$	$fc/2^4$	$fc/2^3$	$fc/2^4$	DV1	-	-
		100	-	-	-	-	-	fc (Note7)	-
		101	fs	fs	fs	fs	-	-	-
110	Reserved						R/W		
111	External clock (TC2 pin input)								
TC2M	TC2 operating mode select	0: Timer/event counter mode 1: Window mode						R/W	

Note 1: fc: High-frequency clock [Hz], fs: Low-frequency clock [Hz], *: Don't care

Note 2: When writing to the Timer Register 2 (TC2DR), always write to the lower side (TC2DRL) and then the upper side (TC2DRH) in that order. Writing to only the lower side (TC2DRL) or the upper side (TC2DRH) has no effect.

Note 3: The timer register 2 (TC2DR) uses the value previously set in it for coincidence detection until data is written to the upper side (TC2DRH) after writing data to the lower side (TC2DRL).

Note 4: Set the mode and source clock when the TC2 stops (TC2S = 0).

Note 5: Values to be loaded to the timer register must satisfy the following condition.
 $TC2DR > 1$ ($TC2DR_{15}$ to $TC2DR_{11} > 1$ at warm up)

Note 6: If a read instruction is executed for TC2CR, read data of bit 7, 6 and 1 are unstable.

Note 7: The high-frequency clock (fc) can be selected only when the time mode at SLOW2 mode is selected.

Note 8: On entering STOP mode, the TC2 start control (TC2S) is cleared to "0" automatically. So, the timer stops. Once the STOP mode has been released, to start using the timer counter, set TC2S again.

9.3 Function

The timer/counter 2 has three operating modes: timer, event counter and window modes.

And if fc or fs is selected as the source clock in timer mode, when switching the timer mode from SLOW1 to NORMAL2, the timer/counter2 can generate warm-up time until the oscillator is stable.

9.3.1 Timer mode

In this mode, the internal clock is used for counting up. The contents of TC2DR are compared with the contents of up counter. If a match is found, a timer/counter 2 interrupt (INTTC2) is generated, and the counter is cleared. Counting up is resumed after the counter is cleared.

When fc is selected for source clock at SLOW2 mode, lower 11-bits of TC2DR are ignored and generated a interrupt by matching upper 5-bits only. Though, in this situation, it is necessary to set TC2DRH only.

Table 9-1 Source Clock (Internal clock) for Timer/Counter2 (at fc = 16 MHz, DV7CK=0)

TC2CK	NORMAL1/2, IDLE1/2 mode								SLOW1/2 mode		SLEEP1/2 mode	
	DV7CK = 0				DV7CK = 1							
	DV1CK = 0		DV1CK = 1		DV1CK = 0		DV1CK = 1		Resolution	Maximum Time Setting	Resolution	Maximum Time Setting
	Resolution	Maximum Time Setting	Resolution	Maximum Time Setting	Resolution	Maximum Time Setting	Resolution	Maximum Time Setting				
000	524.29 [ms]	9.54 [h]	1.05 [s]	19.1 [h]	1 [s]	18.2 [h]	1 [s]	18.2 [h]	1 [s]	18.2 [h]	1 [s]	18.2 [h]
001	512.0 [μs]	33.55 [s]	1.02 [ms]	1.12 [min]	0.98 [ms]	1.07 [min]	0.98 [ms]	1.07 [min]	0.98 [ms]	1.07 [min]	0.98 [ms]	1.07 [min]
010	16.0 [μs]	1.05 [s]	32 [μs]	2.09 [s]	16.0 [μs]	1.05 [s]	32.0 [μs]	2.10 [s]	-	-	-	-
011	0.5 [μs]	32.77 [ms]	1.0 [μs]	65.5 [ms]	0.5 [μs]	32.77 [ms]	1.0 [μs]	65.5 [ms]	-	-	-	-
100	-	-	-	-	-	-	-	-	62.5 [ns]	-	-	-
101	30.52 [μs]	2 [s]	30.52 [μs]	2 [s]	30.52 [μs]	2 [s]	30.52 [μs]	2 [s]	-	-	-	-

Note: When fc is selected as the source clock in timer mode, it is used at warm-up for switching from SLOW1 mode to NORMAL2 mode.

Example :Sets the timer mode with source clock $fc/2^3$ [Hz] and generates an interrupt every 25 ms (at fc = 16 MHz, CGCR<DV1CK> = "0")

```
LDW      (TC2DR), 061AH      ; Sets TC2DR (25 ms * 28/fc = 061AH)
DI                               ; IMF= "0"
SET      (EIRH), 4           ; Enables INTTC2 interrupt
EI                               ; IMF= "1"
LD       (TC2CR), 00001000B   ; Source clock / mode select
LD       (TC2CR), 00101000B   ; Starts Timer
```

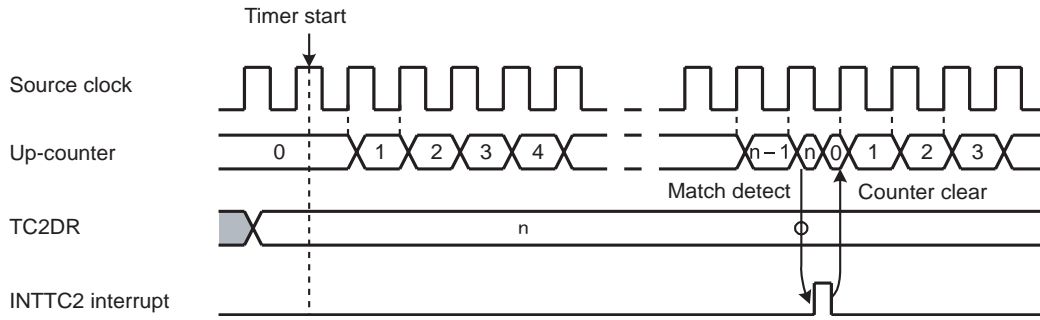


Figure 9-2 Timer Mode Timing Chart

9.3.2 Event counter mode

In this mode, events are counted on the rising edge of the TC2 pin input. The contents of TC2DR are compared with the contents of the up counter. If a match is found, an INTTC2 interrupt is generated, and the counter is cleared. Counting up is resumed every the rising edge of the TC2 pin input after the up counter is cleared.

Match detect is executed on the falling edge of the TC2 pin. Therefore, an INTTC2 interrupt is generated at the falling edge after the match of TC2DR and up counter.

The minimum input pulse width of TC2 pin is shown in Table 9-2. Two or more machine cycles are required for both the “H” and “L” levels of the pulse width.

Example :Sets the event counter mode and generates an INTTC2 interrupt 640 counts later.

```
LDW      (TC2DR), 640      ; Sets TC2DR
DI                          ; IMF= "0"
SET      (EIRH), 4        ; Enables INTTC2 interrupt
EI                          ; IMF= "1"
LD       (TC2CR), 00011100B ; TC2 source vclock / mode select
LD       (TC2CR), 00111100B ; Starts TC2
```

Table 9-2 Timer/Counter 2 External Input Clock Pulse Width

	Minimum Input Pulse Width [s]	
	NORMAL 1/2, IDLE 1/2 mode	SLOW 1/2, SLEEP 1/2 mode
“H” width	$2^3/f_c$	$2^3/f_s$
“L” width	$2^3/f_c$	$2^3/f_s$

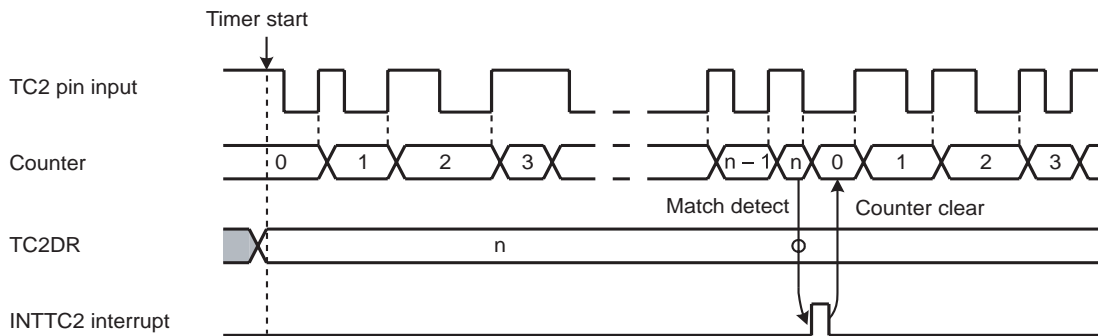


Figure 9-3 Event Counter Mode Timing Chart

9.3.3 Window mode

In this mode, counting up performed on the rising edge of an internal clock during TC2 external pin input (Window pulse) is “H” level. The contents of TC2DR are compared with the contents of up counter. If a match found, an INTTC2 interrupt is generated, and the up-counter is cleared.

The maximum applied frequency (TC2 input) must be considerably slower than the selected internal clock by the TC2CR<TC2CK>.

Note: It is not available window mode in the SLOW/SLEEP mode. Therefore, at the window mode in NORMAL mode, the timer should be halted by setting TC2CR<TC2S> to "0" before the SLOW/SLEEP mode is entered.

Example :Generates an interrupt, inputting “H” level pulse width of 120 ms or more. (at $f_c = 16 \text{ MHz}$, $\text{TBTCR}\langle\text{DV7CK}\rangle = \text{“0”}$, $\text{CGCR}\langle\text{DV1CK}\rangle = \text{“0”}$)

```
LDW      (TC2DR), 00EAH      ; Sets TC2DR ( $120 \text{ ms} \cdot 2^{13}/f_c = 00EAH$ )
DI       ; IMF= “0”
SET      (EIRH), 4          ; Enables INTTC2 interrupt
EI       ; IMF= “1”
LD       (TC2CR), 00000101B ; TC2sorce clock / mode select
LD       (TC2CR), 00100101B ; Starts TC2
```

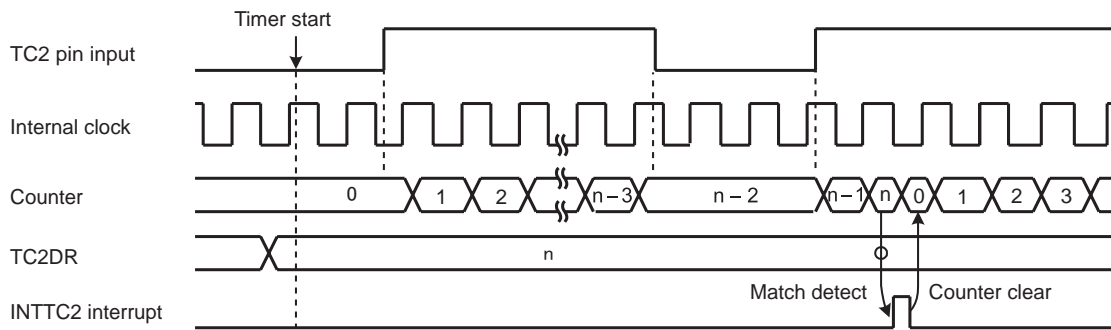
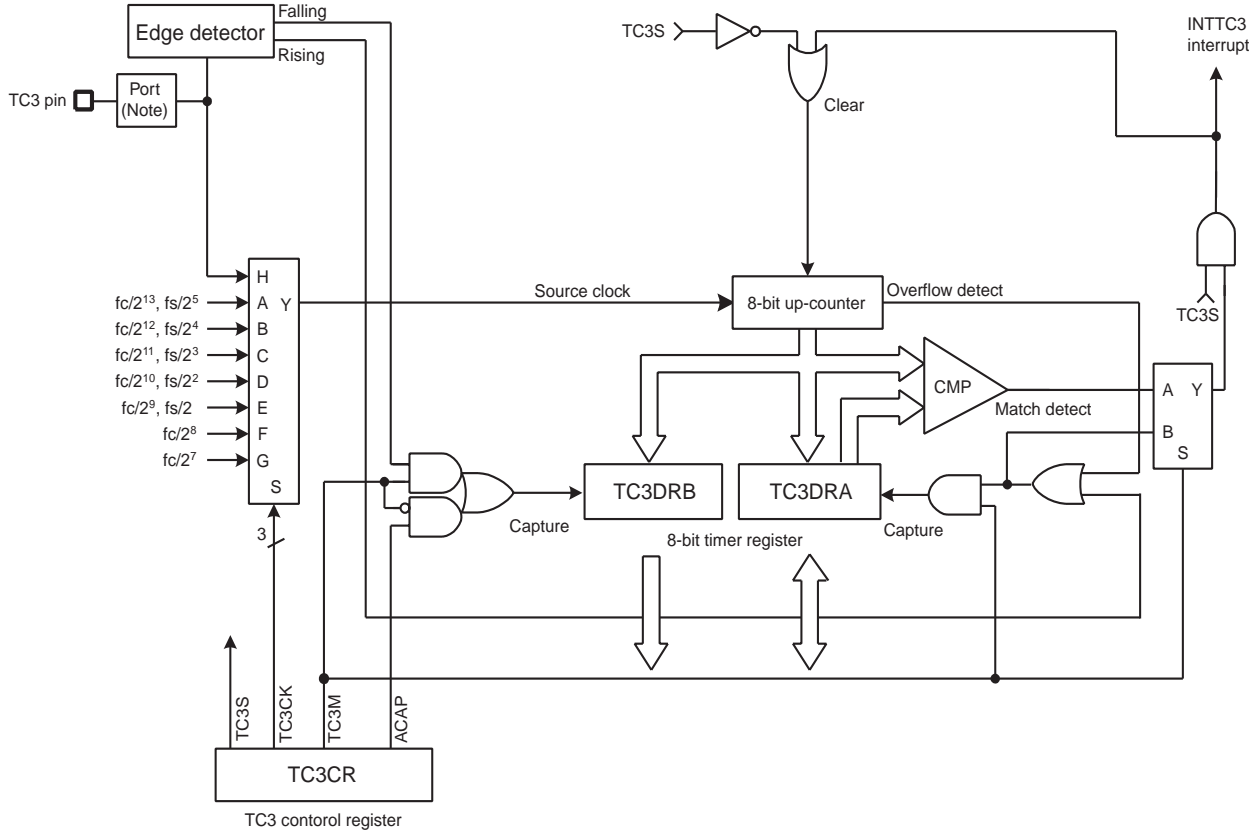


Figure 9-4 Window Mode Timing Chart

10. 8-Bit TimerCounter 3 (TC3)

10.1 Configuration



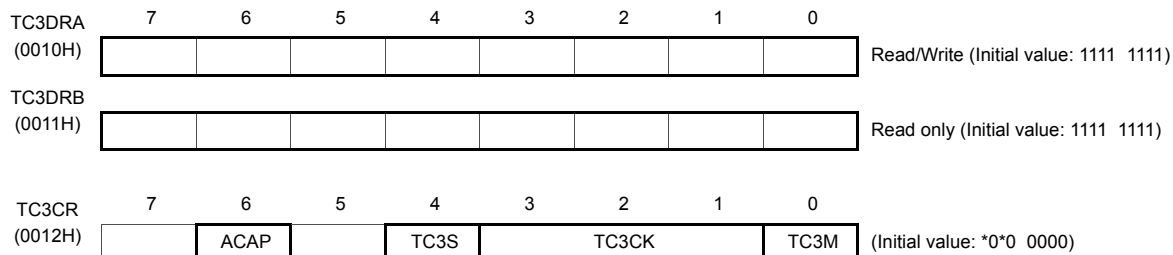
Note: Function input may not operate depending on I/O port setting. For more details, see the chapter "I/O Port".

Figure 10-1 TimerCounter 3 (TC3)

10.2 TimerCounter Control

The TimerCounter 3 is controlled by the TimerCounter 3 control register (TC3CR) and two 8-bit timer registers (TC3DRA and TC3DRB).

Timer Register and Control Register



ACAP	Auto capture control	0: – 1: Auto capture	R/W						
TC3S	TC3 start control	0: Stop and counter clear 1: Start	R/W						
TC3CK	TC3 source clock select [Hz]	NORMAL1/2, IDLE1/2 mode	R/W						
		DV7CK = 0		Divider					
		DV7CK = 1							
		DV1CK=0		DV1CK=1	SLOW1/2, SLEEP1/2 mode				
		DV1CK=0		DV1CK=1					
		000		$fc/2^{13}$	$fc/2^{14}$	$fs/2^5$	$fs/2^5$	DV11	$fs/2^5$
		001		$fc/2^{12}$	$fc/2^{13}$	$fs/2^4$	$fs/2^4$	DV10	$fs/2^4$
		010		$fc/2^{11}$	$fc/2^{12}$	$fs/2^3$	$fs/2^3$	DV9	$fs/2^3$
011	$fc/2^{10}$	$fc/2^{11}$	$fs/2^2$	$fs/2^2$	DV8	$fs/2^2$			
100	$fc/2^9$	$fc/2^{10}$	$fs/2$	$fs/2$	DV7	$fs/2$			
101	$fc/2^8$	$fc/2^9$	$fc/2^8$	$fc/2^9$	DV6	–			
110	$fc/2^7$	$fc/2^8$	$fc/2^7$	$fc/2^8$	DV5	–			
111	External clock (TC3 pin input)								
TC3M	TC3 operating mode select	0: Timer/event counter mode 1: Capture mode	R/W						

Note 1: fc: High-frequency clock [Hz], fs: Low-frequency clock [Hz], *: Don't care

Note 2: Set the operating mode and source clock when TimerCounter stops (TC3S = 0).

Note 3: To set the timer registers, the following relationship must be satisfied.
TC3DRA > 1 (Timer/event counter mode)

Note 4: Auto-capture (ACAP) can be used only in the timer and event counter modes.

Note 5: When the read instruction is executed to TC3CR, the bit 5 and 7 are read as a don't care.

Note 6: Do not program TC3DRA when the timer is running (TC3S = 1).

Note 7: When the STOP mode is entered, the start control (TC3S) is cleared to 0 automatically, and the timer stops. After the STOP mode is exited, TC3S must be set again to use the timer counter.

10.3 Function

TimerCounter 3 has three types of operating modes: timer, event counter and capture modes.

10.3.1 Timer mode

In the timer mode, the up-counter counts up using the internal clock. When a match between the up-counter and the timer register 3A (TC3DRA) value is detected, an INTTC3 interrupt is generated and the up-counter is cleared. After being cleared, the up-counter restarts counting. Setting TC3CR<ACAP> to 1 captures the up-counter value into the timer register B (TC3DRB) with the auto-capture function. The count value during timer operation can be checked by executing the read instruction to TC3DRB.

Note: 00H which is stored in the up-counter immediately after detection of a match is not captured into TC3DRB. (Figure 10-2)

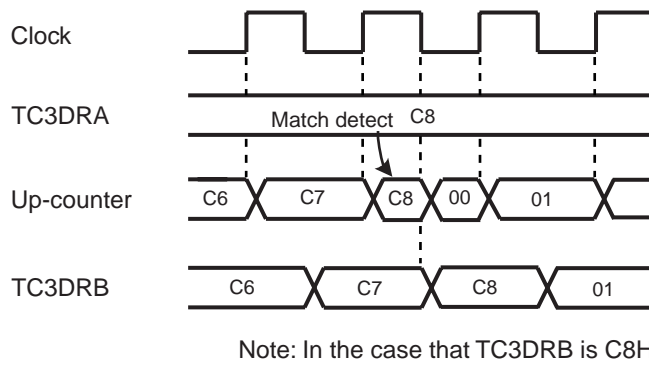
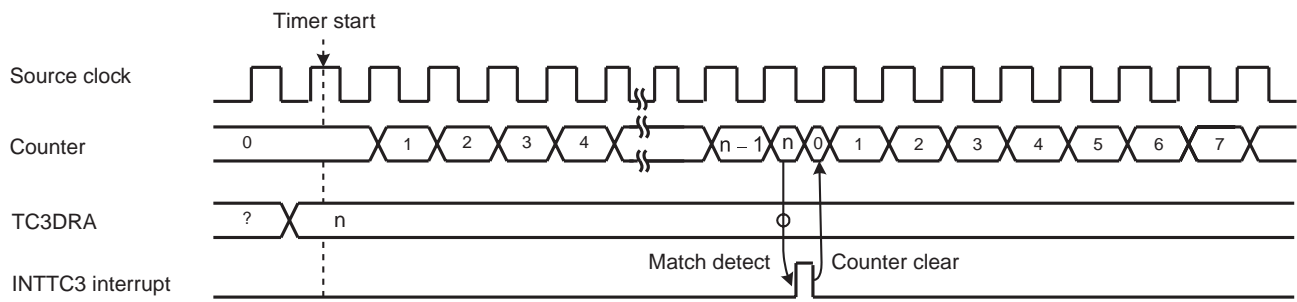


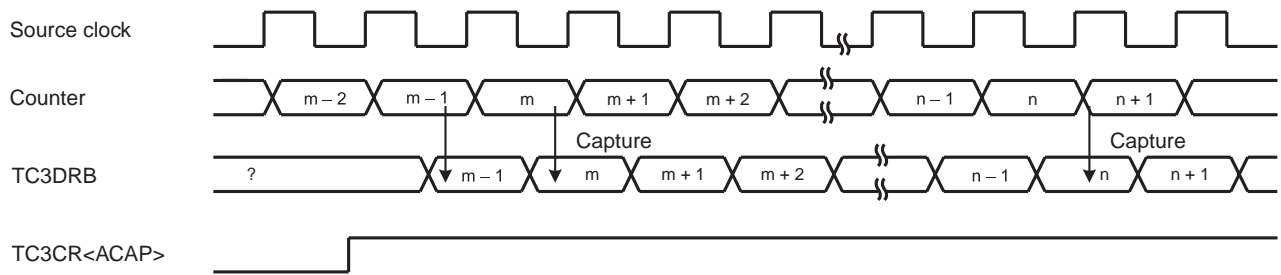
Figure 10-2 Auto-Capture Function

Table 10-1 Source Clock for TimerCounter 3 (Example: fc = 16 MHz, fs = 32.768 kHz)

TC3CK	NORMAL1/2, IDLE1/2 mode								SLOW1/2, SLEEP1/2 mode	
	DV7CK = 0				DV7CK = 1					
	DV1CK = 0		DV1CK = 1		DV1CK = 0		DV1CK = 1			
	Resolution [μs]	Maximum Time Setting [ms]	Resolution [μs]	Maximum Time Setting [ms]	Resolution [μs]	Maximum Time Setting [ms]	Resolution [μs]	Maximum Time Setting [ms]		
000	512	130.6	1024	261.1	976.56	249.0	976.56	249.0	976.56	249.0
001	256	65.3	512	130.6	488.28	124.5	488.28	124.5	488.28	124.5
010	128	32.6	256	65.3	244.14	62.3	244.14	62.3	244.14	62.3
011	64	16.3	128	32.6	122.07	31.1	122.07	31.1	122.07	31.1
100	32	8.2	64	16.3	61.01	15.6	61.01	15.6	61.01	15.6
101	16	4.1	32	8.2	16.0	4.1	32.0	8.2	–	–
110	8	2.0	16	4.1	8.0	2.0	16.0	4.1	–	–



(a) Timer mode



(b) Auto capture

Figure 10-3 Timer Mode Timing Chart

10.3.2 Event Counter Mode

In the event counter mode, the up-counter counts up at the rising edge of the input pulse to the TC3 pin.

When a match between the up-counter and TC3DRA value is detected, an INTTC3 interrupt is generated and up-counter is cleared. After being cleared, the up-counter restarts counting at each rising edge of the input pulse to the TC3 pin. Since a match is detected at the falling edge of the input pulse to TC3 pin, an INTTC3 interrupt request is generated at the falling edge immediately after the up-counter reaches the value set in TC3DRA.

The maximum applied frequencies are shown in Table 10-2. The pulse width larger than one machine cycle is required for high-going and low-going pulses.

Setting TC3CR<ACAP> to 1 captures the up-counter value into TC3DRB with the auto-capture function. The count value during a timer operation can be checked by the read instruction to TC3DRB.

Note: 00H which is stored in the up-counter immediately after detection of a match is not captured into TC3DRB. (Figure 10-2)

Example :Inputting 50 Hz pulse to TC3, and generating interrupts every 0.5 s

- LD (TC3CR), 00001110B : Sets the clock mode
- LD (TC3DRA), 19H : $0.5 \text{ s} \div 1/50 = 25 = 19\text{H}$
- LD (TC3CR), 00011110B : Starts TC3.

Table 10-2 Maximum Frequencies Applied to TC3

	Minimum Pulse Width	
	NORMAL1/2, IDLE1/2 mode	SLOW1/2, SLEEP1/2 mode
High-going	$2^2/f_c$	$2^2/f_s$
Low-going	$2^2/f_c$	$2^2/f_s$

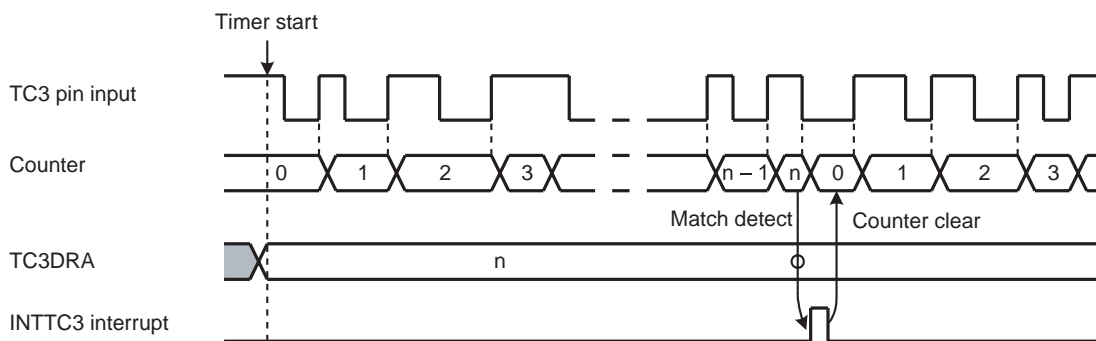


Figure 10-4 Event Counter Mode Timing Chart

10.3.3 Capture Mode

In the capture mode, the pulse width, frequency and duty cycle of the pulse input to the TC3 pin are measured with the internal clock. The capture mode is used to decode remote control signals, and identify AC50/60 Hz.

When the falling edge of the TC3 input is detected after the timer starts, the up-counter value is captured into TC3DRB. Hereafter, whenever the rising edge is detected, the up-counter value is captured into TC3DRA and the INTTC3 interrupt request is generated. The up-counter is cleared at this time. Generally, read TC3DRB and TC3DRA during INTTC3 interrupt processing. After the up-counter is cleared, counting is continued and the next up-counter value is captured into TC3DRB.

When the rising edge is detected immediately after the timer starts, the up-counter value is captured into TC3DRA only, but not into TC3DRB. The INTTC3 interrupt request is generated. When the read instruction is executed to TC3DRB at this time, the value at the completion of the last capture (FF immediately after a reset) is read.

The minimum input pulse width must be larger than one cycle width of the source clock programmed in TC3CR<TC3CK>.

The INTTC3 interrupt request is generated if the up-counter overflow (FFH) occurs during capture operation before the edge is detected. TC3DRA is set to FFH and the up-counter is cleared. Counting is continued by the up-counter, but capture operation and overflow detection are stopped until TC3DRA is read. Generally, read TC3DRB first because capture operation and overflow detection resume by reading TC3DRA.

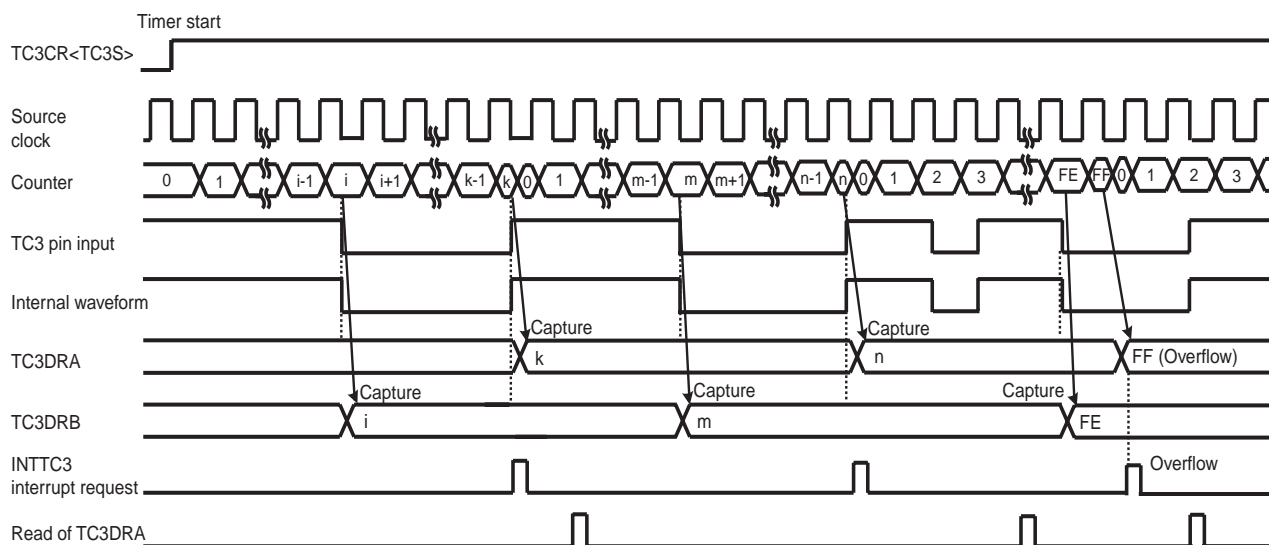
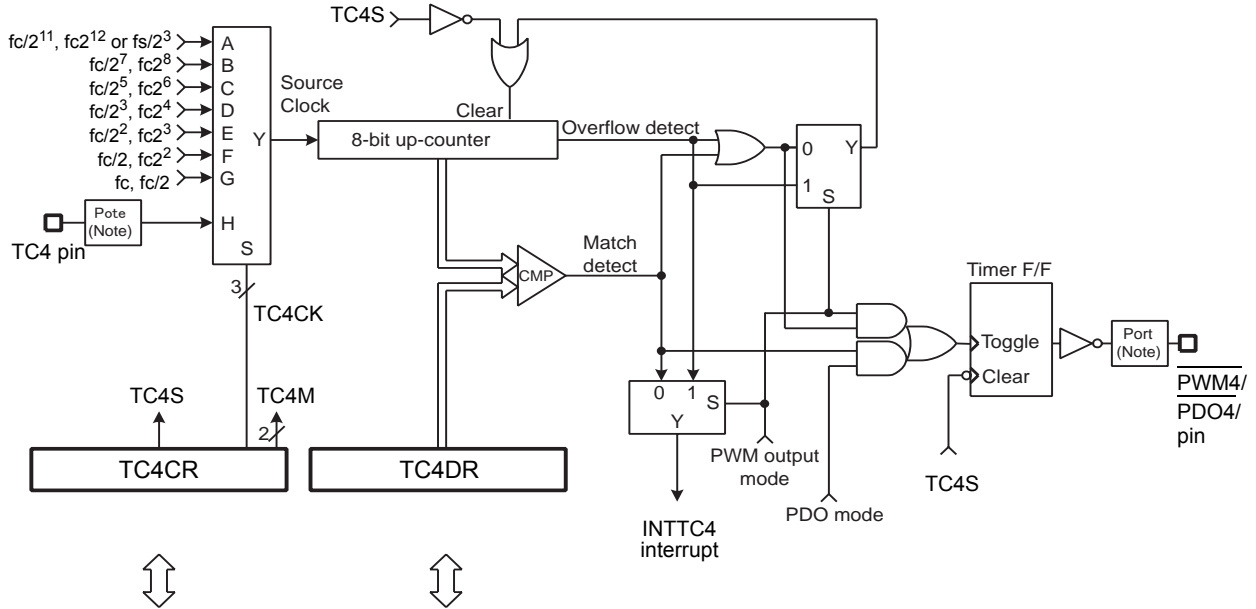


Figure 10-5 Capture Mode Timing Chart

11. 8-Bit TimerCounter 4 (TC4)

11.1 Configuration



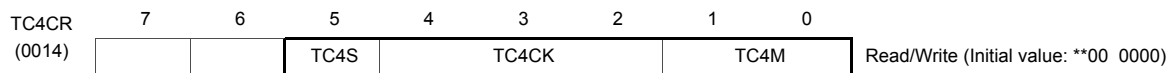
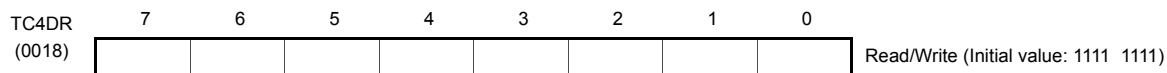
Note: Function I/O may not operate depending on I/O port setting. For more details, see the chapter "I/O Port".

Figure 11-1 TimerCounter 4 (TC4)

11.2 TimerCounter Control

The TimerCounter 4 is controlled by the TimerCounter 4 control register (TC4CR) and timer registers 4 (TC4DR).

Timer Register and Control Register



TC4S	TC4 start control	0: Stop and counter clear 1: Start	R/W						
TC4CK	TC4 source clock select [Hz]	NORMAL1/2, IDLE1/2 mode	R/W						
		DV7CK = 0		DV7CK = 1	Divider	SLOW1/2, SLEEP1/2 mode			
		DV1CK = 0		DV1CK = 1					
		000		$fc/2^{11}$	$fc/2^{12}$	$fs/2^3$	$fs/2^3$	DV9	$fs/2^3$
		001		$fc/2^7$	$fc/2^8$	$fc/2^7$	$fc/2^8$	DV5	-
		010		$fc/2^5$	$fc/2^6$	$fc/2^5$	$fc/2^6$	DV3	-
		011		$fc/2^3$	$fc/2^4$	$fc/2^3$	$fc/2^4$	DV1	-
		100		$fc/2^2$	$fc/2^3$	$fc/2^2$	$fc/2^3$	-	-
		101		$fc/2$	$fc/2^2$	$fc/2$	$fc/2^2$	-	-
110	fc	$fc/2$	fc	$fc/2$	-	-			
111	External clock (TC4 pin input)					-	-		
TC4M	TC4 operating mode select	00: Timer/event counter mode 01: Reserved 10: Programmable divider output (PDO) mode 11: Pulse width modulation (PWM) output mode	R/W						

Note 1: fc: High-frequency clock [Hz], fs: Low-frequency clock [Hz], *: Don't care

Note 2: To set the timer registers, the following relationship must be satisfied.
 $1 \leq TC4DR \leq 255$

Note 3: To start timer operation (TC4S = 0 → 1) or disable timer operation (TC4S = 1 → 0), do not change the TC4CR<TC4M, TC4CK> setting. During timer operation (TC4S = 1 → 1), do not change it, either. If the setting is programmed during timer operation, counting is not performed correctly.

Note 4: The event counter and PWM output modes are used only in the NOMAL1/2 and IDLE1/2 modes.

Note 5: When the STOP mode is entered, the start control (TC4S) is cleared to "0" automatically.

Note 6: The bit 6 and 7 of TC4CR are read as a don't care when these bits are read.

Note 7: In the timer, event counter and PDO modes, do not change the TC4DR setting when the timer is running.

Note 8: When the high-frequency clock fc exceeds 10 MHz, do not select the source clock of TC4CK = 110.

Note 9: The operating clock fs can not be used in NORMAL1 or IDEL1 mode (when low-frequency oscillation is stopped.)

Note 10: For available source clocks depending on the operation mode, refer to the following table.

TC4CK		Timer Mode	Event Counter Mode	PDO Mode	PWM Mode
		000	O	-	O
001	O	-	O	-	
010	O	-	O	-	
011	O	-	-	O	
100	-	-	-	O	
101	-	-	-	O	
110	-	-	-	O	
111	-	O	-	x	

Note: O : Available source clock



11.3 Function

TimerCounter 4 has four types of operating modes: timer, event counter, programmable divider output (PDO), and pulse width modulation (PWM) output modes.

11.3.1 Timer Mode

In the timer mode, the up-counter counts up using the internal clock. When a match between the up-counter and the TC4DR value is detected, an INTTC4 interrupt is generated and the up-counter is cleared. After being cleared, the up-counter restarts counting.

Table 11-1 Source Clock for TimerCounter 4 (Example: $f_c = 16$ MHz, $f_s = 32.768$ kHz)

TC4CK	NORMAL1/2, IDLE1/2 Mode								SLOW1/2, SLEEP1/2 Mode	
	DV7CK = 0				DV7CK = 1					
	DV1CK = 0		DV1CK = 1		DV1CK = 0		DV1CK = 1		Resolution [μs]	Maximum Time Setting [ms]
	Resolution [μs]	Maximum Time Setting [ms]	Resolution [μs]	Maximum Time Setting [ms]	Resolution [μs]	Maximum Time Setting [ms]	Resolution [μs]	Maximum Time Setting [ms]		
000	128.0	32.6	256.0	65.3	244.14	62.2	244.14	62.2	244.14	62.2
001	8.0	2.0	16.0	4.1	8.0	2.0	16.0	4.1	–	–
010	2.0	0.510	4.0	1.0	2.0	0.510	4.0	1.0	–	–
011	0.5	0.128	1.0	0.255	0.5	0.128	1.0	0.255	–	–

11.3.2 Event Counter Mode

In the event counter mode, the up-counter counts up at the rising edge of the input pulse to the TC4 pin.

When a match between the up-counter and the TC4DR value is detected, an INTTC4 interrupt is generated and the up-counter is cleared. After being cleared, the up-counter restarts counting at rising edge of the TC4 pin. Since a match is detected at the falling edge of the input pulse to the TC4 pin, the INTTC4 interrupt request is generated at the falling edge immediately after the up-counter reaches the value set in TC4DR.

The minimum pulse width applied to the TC4 pin are shown in Table 11-2. The pulse width larger than two machine cycles is required for high- and low-going pulses.

Note: The event counter mode can not used in the SLOW1/2 and SLEEP1/2 modes since the external clock is not supplied in these modes.

Table 11-2 External Source Clock for TimerCounter 4

	Minimum Pulse Width
	NORMAL1/2, IDLE1/2 mode
High-going	$2^3/f_c$
Low-going	$2^3/f_c$

11.3.3 Programmable Divider Output (PDO) Mode

The programmable divider output (PDO) mode is used to generate a pulse with a 50% duty cycle by counting with the internal clock.

When a match between the up-counter and the TC4DR value is detected, the logic level output from the PDO4 pin is switched to the opposite state and INTTC4 interrupt request is generated. The up-counter is cleared at this time and then counting is continued. When a match between the up-counter and the TC4DR value is detected, the logic level output from the PDO4 pin is switched to the opposite state again and INTTC4 interrupt request is generated. The up-counter is cleared at this time, and then counting and PDO are continued.

When the timer is stopped, the PDO4 pin is high. Therefore, if the timer is stopped when the PDO4 pin is low, the duty pulse may be shorter than the programmed value.

Example :Generating 1024 Hz pulse ($f_c = 16.0 \text{ Mhz}$)

LD	(TC4CR), 00000110B	: Sets the PDO mode. (TC4M = 10, TC4CK = 001)
LD	(TC4DR), 3DH	: $1/1024 \div 2^7/f_c \div 2$ (half cycle period) = 3DH
LD	(TC4CR), 00100110B	: Start TC4

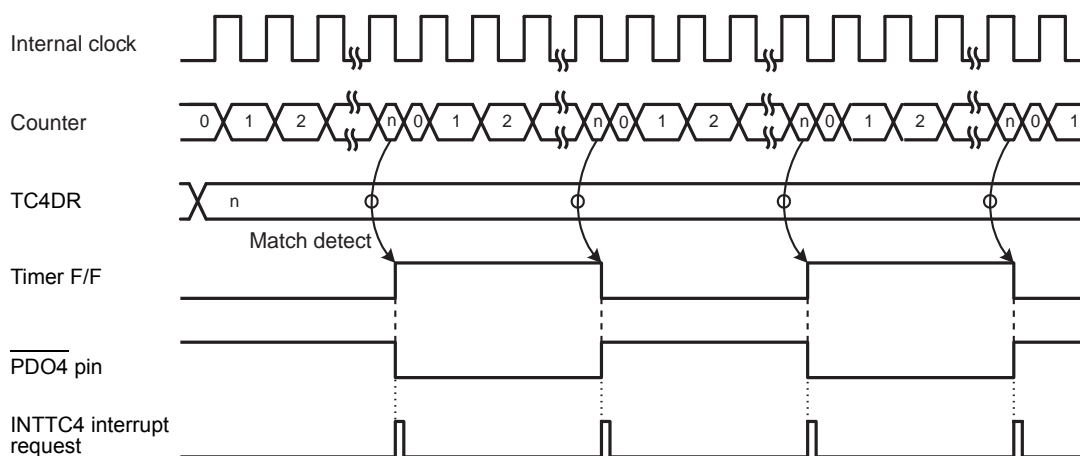


Figure 11-2 PDO Mode Timing Chart

11.3.4 Pulse Width Modulation (PWM) Output Mode

The pulse width modulation (PWM) output mode is used to generate the PWM pulse with up to 8 bits of resolution by an internal clock.

When a match between the up-counter and the TC4DR value is detected, the logic level output from the PWM4 pin becomes low. The up-counter continues counting. When the up-counter overflow occurs, the PWM4 pin becomes high. The INTTC4 interrupt request is generated at this time.

When the timer is stopped, the PWM4 pin is high. Therefore, if the timer is stopped when the PWM4 pin is low, one PWM cycle may be shorter than the programmed value.

TC4DR is serially connected to the shift register. If TC4DR is programmed during PWM output, the data set to TC4DR is not shifted until one PWM cycle is completed. Therefore, a pulse can be modulated periodically. For the first time, the data written to TC4DR is shifted when the timer is started by setting TC4CR<TC4S> to 1.

Note 1: The PWM output mode can be used only in the NORMAL1/2 and IDEL 1/2 modes.

Note 2: In the PWM output mode, program TC4DR immediately after the INTTC4 interrupt request is generated (typically in the INTTC4 interrupt service routine.) When the programming of TC4DR and the INTTC4 interrupt occur at the same time, an unstable value is shifted, that may result in generation of pulse different from the programmed value until the next INTTC4 interrupt request is issued.

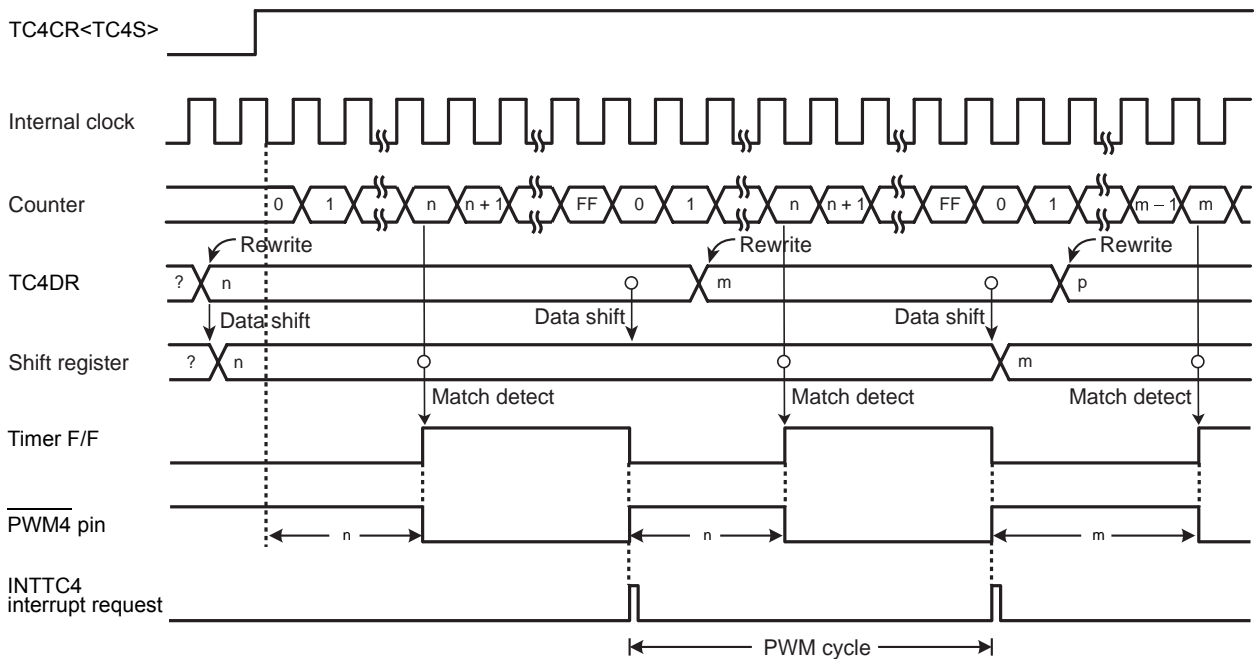


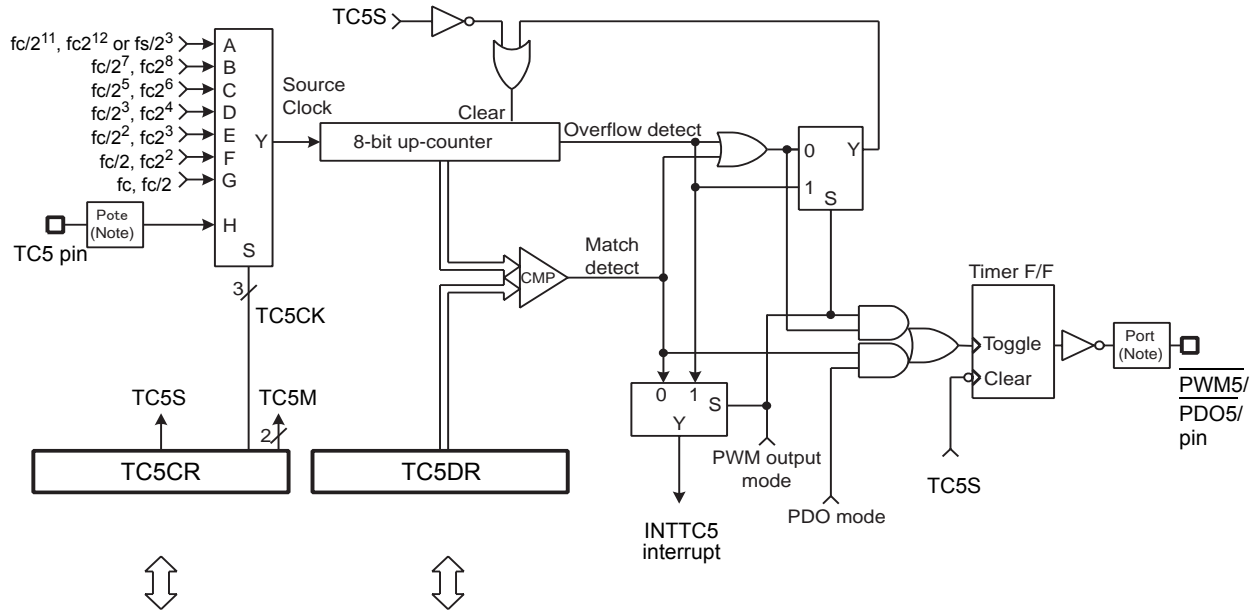
Figure 11-3 PWM output Mode Timing Chart (TC4)

Table 11-3 PWM Mode (Example: $f_c = 16$ MHz)

TC4CK	NORMAL1/2, IDLE1/2 Mode							
	DV7CK = 0				DV7CK = 1			
	DV1CK = 0		DV1CK = 1		DV1CK = 0		DV1CK = 1	
	Resolution [ns]	Cycle [μ s]	Resolution [ns]	Cycle [μ s]	Resolution [ns]	Cycle [μ s]	Resolution [ns]	Cycle [μ s]
000	–	–	–	–	–	–	–	–
001	–	–	–	–	–	–	–	–
010	–	–	–	–	–	–	–	–
011	500	128	1000	256	500	128	1000	256
100	250	64	500	128	250	64	500	128
101	125	32	250	64	125	32	250	64
110	–	–	–	–	–	–	–	–

12. 8-Bit TimerCounter 5 (TC5)

12.1 Configuration



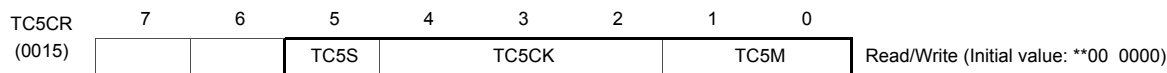
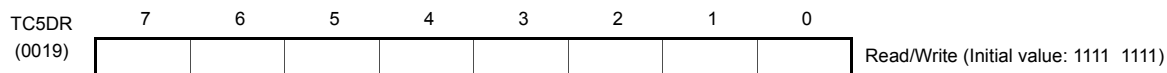
Note: Function I/O may not operate depending on I/O port setting. For more details, see the chapter "I/O Port".

Figure 12-1 TimerCounter 5 (TC5)

12.2 TimerCounter Control

The TimerCounter 5 is controlled by the TimerCounter 5 control register (TC5CR) and timer registers 5 (TC5DR).

Timer Register and Control Register



TC5S	TC5 start control	0: Stop and counter clear 1: Start	R/W																																																																						
TC5CK	TC5 source clock select [Hz]	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="4" style="text-align: center;">NORMAL1/2, IDLE1/2 mode</th> <th rowspan="3" style="text-align: center;">Divider</th> <th rowspan="3" style="text-align: center;">SLOW1/2, SLEEP1/2 mode</th> </tr> <tr> <th colspan="2" style="text-align: center;">DV7CK = 0</th> <th colspan="2" style="text-align: center;">DV7CK = 1</th> </tr> <tr> <th style="text-align: center;">DV1CK = 0</th> <th style="text-align: center;">DV1CK = 1</th> <th style="text-align: center;">DV1CK = 0</th> <th style="text-align: center;">DV1CK = 1</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">000</td> <td style="text-align: center;">$fc/2^{11}$</td> <td style="text-align: center;">$fc/2^{12}$</td> <td style="text-align: center;">$fs/2^3$</td> <td style="text-align: center;">$fs/2^3$</td> <td style="text-align: center;">DV9</td> <td style="text-align: center;">$fs/2^3$</td> </tr> <tr> <td style="text-align: center;">001</td> <td style="text-align: center;">$fc/2^7$</td> <td style="text-align: center;">$fc/2^8$</td> <td style="text-align: center;">$fc/2^7$</td> <td style="text-align: center;">$fc/2^8$</td> <td style="text-align: center;">DV5</td> <td style="text-align: center;">-</td> </tr> <tr> <td style="text-align: center;">010</td> <td style="text-align: center;">$fc/2^5$</td> <td style="text-align: center;">$fc/2^6$</td> <td style="text-align: center;">$fc/2^5$</td> <td style="text-align: center;">$fc/2^6$</td> <td style="text-align: center;">DV3</td> <td style="text-align: center;">-</td> </tr> <tr> <td style="text-align: center;">011</td> <td style="text-align: center;">$fc/2^3$</td> <td style="text-align: center;">$fc/2^4$</td> <td style="text-align: center;">$fc/2^3$</td> <td style="text-align: center;">$fc/2^4$</td> <td style="text-align: center;">DV1</td> <td style="text-align: center;">-</td> </tr> <tr> <td style="text-align: center;">100</td> <td style="text-align: center;">$fc/2^2$</td> <td style="text-align: center;">$fc/2^3$</td> <td style="text-align: center;">$fc/2^2$</td> <td style="text-align: center;">$fc/2^3$</td> <td style="text-align: center;">-</td> <td style="text-align: center;">-</td> </tr> <tr> <td style="text-align: center;">101</td> <td style="text-align: center;">$fc/2$</td> <td style="text-align: center;">$fc/2^2$</td> <td style="text-align: center;">$fc/2$</td> <td style="text-align: center;">$fc/2^2$</td> <td style="text-align: center;">-</td> <td style="text-align: center;">-</td> </tr> <tr> <td style="text-align: center;">110</td> <td style="text-align: center;">fc</td> <td style="text-align: center;">$fc/2$</td> <td style="text-align: center;">fc</td> <td style="text-align: center;">$fc/2$</td> <td style="text-align: center;">-</td> <td style="text-align: center;">-</td> </tr> <tr> <td style="text-align: center;">111</td> <td colspan="4" style="text-align: center;">External clock (TC5 pin input)</td> <td style="text-align: center;">-</td> <td style="text-align: center;">-</td> </tr> </tbody> </table>	NORMAL1/2, IDLE1/2 mode				Divider	SLOW1/2, SLEEP1/2 mode	DV7CK = 0		DV7CK = 1		DV1CK = 0	DV1CK = 1	DV1CK = 0	DV1CK = 1	000	$fc/2^{11}$	$fc/2^{12}$	$fs/2^3$	$fs/2^3$	DV9	$fs/2^3$	001	$fc/2^7$	$fc/2^8$	$fc/2^7$	$fc/2^8$	DV5	-	010	$fc/2^5$	$fc/2^6$	$fc/2^5$	$fc/2^6$	DV3	-	011	$fc/2^3$	$fc/2^4$	$fc/2^3$	$fc/2^4$	DV1	-	100	$fc/2^2$	$fc/2^3$	$fc/2^2$	$fc/2^3$	-	-	101	$fc/2$	$fc/2^2$	$fc/2$	$fc/2^2$	-	-	110	fc	$fc/2$	fc	$fc/2$	-	-	111	External clock (TC5 pin input)				-	-	R/W
		NORMAL1/2, IDLE1/2 mode				Divider			SLOW1/2, SLEEP1/2 mode																																																																
		DV7CK = 0		DV7CK = 1																																																																					
		DV1CK = 0	DV1CK = 1	DV1CK = 0	DV1CK = 1																																																																				
		000	$fc/2^{11}$	$fc/2^{12}$	$fs/2^3$	$fs/2^3$	DV9	$fs/2^3$																																																																	
		001	$fc/2^7$	$fc/2^8$	$fc/2^7$	$fc/2^8$	DV5	-																																																																	
		010	$fc/2^5$	$fc/2^6$	$fc/2^5$	$fc/2^6$	DV3	-																																																																	
		011	$fc/2^3$	$fc/2^4$	$fc/2^3$	$fc/2^4$	DV1	-																																																																	
		100	$fc/2^2$	$fc/2^3$	$fc/2^2$	$fc/2^3$	-	-																																																																	
101	$fc/2$	$fc/2^2$	$fc/2$	$fc/2^2$	-	-																																																																			
110	fc	$fc/2$	fc	$fc/2$	-	-																																																																			
111	External clock (TC5 pin input)				-	-																																																																			
TC5M	TC5 operating mode select	00: Timer/event counter mode 01: Reserved 10: Programmable divider output (PDO) mode 11: Pulse width modulation (PWM) output mode	R/W																																																																						

Note 1: fc: High-frequency clock [Hz], fs: Low-frequency clock [Hz], *: Don't care

Note 2: To set the timer registers, the following relationship must be satisfied.
 $1 \leq TC5DR \leq 255$

Note 3: To start timer operation (TC5S = 0 → 1) or disable timer operation (TC5S = 1 → 0), do not change the TC5CR<TC5M, TC5CK> setting. During timer operation (TC5S = 1 → 1), do not change it, either. If the setting is programmed during timer operation, counting is not performed correctly.

Note 4: The event counter and PWM output modes are used only in the NOMAL1/2 and IDLE1/2 modes.

Note 5: When the STOP mode is entered, the start control (TC5S) is cleared to "0" automatically.

Note 6: The bit 6 and 7 of TC5CR are read as a don't care when these bits are read.

Note 7: In the timer, event counter and PDO modes, do not change the TC5DR setting when the timer is running.

Note 8: When the high-frequency clock fc exceeds 10 MHz, do not select the source clock of TC5CK = 110.

Note 9: The operating clock fs can not be used in NORMAL1 or IDEL1 mode (when low-frequency oscillation is stopped.)

Note 10: For available source clocks depending on the operation mode, refer to the following table.

		Timer Mode	Event Counter Mode	PDO Mode	PWM Mode
		000	O	-	O
TC5CK	001	O	-	O	-
	010	O	-	O	-
	011	O	-	-	O
	100	-	-	-	O
	101	-	-	-	O
	110	-	-	-	O
	111	-	O	-	x

Note: O : Available source clock



12.3 Function

TimerCounter 5 has four types of operating modes: timer, event counter, programmable divider output (PDO), and pulse width modulation (PWM) output modes.

12.3.1 Timer Mode

In the timer mode, the up-counter counts up using the internal clock. When a match between the up-counter and the TC5DR value is detected, an INTTC5 interrupt is generated and the up-counter is cleared. After being cleared, the up-counter restarts counting.

Table 12-1 Source Clock for TimerCounter 5 (Example: $f_c = 16$ MHz, $f_s = 32.768$ kHz)

TC5CK	NORMAL1/2, IDLE1/2 Mode								SLOW1/2, SLEEP1/2 Mode	
	DV7CK = 0				DV7CK = 1					
	DV1CK = 0		DV1CK = 1		DV1CK = 0		DV1CK = 1			
	Resolution [μs]	Maximum Time Setting [ms]	Resolution [μs]	Maximum Time Setting [ms]	Resolution [μs]	Maximum Time Setting [ms]	Resolution [μs]	Maximum Time Setting [ms]	Resolution [μs]	Maximum Time Setting [ms]
000	128.0	32.6	256.0	65.3	244.14	62.2	244.14	62.2	244.14	62.2
001	8.0	2.0	16.0	4.1	8.0	2.0	16.0	4.1	–	–
010	2.0	0.510	4.0	1.0	2.0	0.510	4.0	1.0	–	–
011	0.5	0.128	1.0	0.255	0.5	0.128	1.0	0.255	–	–

12.3.2 Event Counter Mode

In the event counter mode, the up-counter counts up at the rising edge of the input pulse to the TC5 pin.

When a match between the up-counter and the TC5DR value is detected, an INTTC5 interrupt is generated and the up-counter is cleared. After being cleared, the up-counter restarts counting at rising edge of the TC5 pin. Since a match is detected at the falling edge of the input pulse to the TC5 pin, the INTTC5 interrupt request is generated at the falling edge immediately after the up-counter reaches the value set in TC5DR.

The minimum pulse width applied to the TC5 pin are shown in Table 12-2. The pulse width larger than two machine cycles is required for high- and low-going pulses.

Note: The event counter mode can not used in the SLOW1/2 and SLEEP1/2 modes since the external clock is not supplied in these modes.

Table 12-2 External Source Clock for TimerCounter 5

	Minimum Pulse Width
	NORMAL1/2, IDLE1/2 mode
High-going	$2^3/f_c$
Low-going	$2^3/f_c$

12.3.3 Programmable Divider Output (PDO) Mode

The programmable divider output (PDO) mode is used to generate a pulse with a 50% duty cycle by counting with the internal clock.

When a match between the up-counter and the TC5DR value is detected, the logic level output from the PDO5 pin is switched to the opposite state and INTTC5 interrupt request is generated. The up-counter is cleared at this time and then counting is continued. When a match between the up-counter and the TC5DR value is detected, the logic level output from the PDO5 pin is switched to the opposite state again and INTTC5 interrupt request is generated. The up-counter is cleared at this time, and then counting and PDO are continued.

When the timer is stopped, the PDO5 pin is high. Therefore, if the timer is stopped when the PDO5 pin is low, the duty pulse may be shorter than the programmed value.

Example :Generating 1024 Hz pulse (fc = 16.0 Mhz)

LD	(TC5CR), 00000110B	: Sets the PDO mode. (TC5M = 10, TC5CK = 001)
LD	(TC5DR), 3DH	: $1/1024 \div 2^7/fc \div 2$ (half cycle period) = 3DH
LD	(TC5CR), 00100110B	: Start TC5

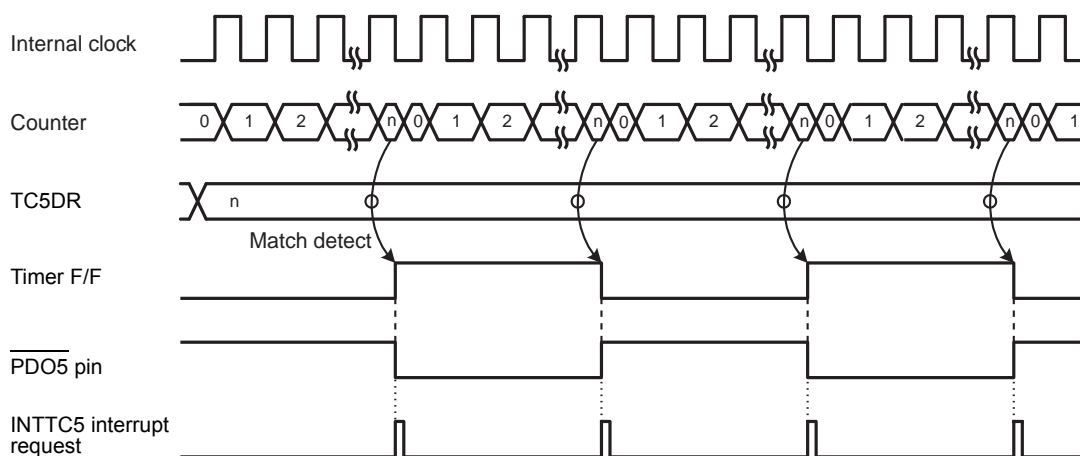


Figure 12-2 PDO Mode Timing Chart

12.3.4 Pulse Width Modulation (PWM) Output Mode

The pulse width modulation (PWM) output mode is used to generate the PWM pulse with up to 8 bits of resolution by an internal clock.

When a match between the up-counter and the TC5DR value is detected, the logic level output from the PWM5 pin becomes low. The up-counter continues counting. When the up-counter overflow occurs, the PWM5 pin becomes high. The INTTC5 interrupt request is generated at this time.

When the timer is stopped, the PWM5 pin is high. Therefore, if the timer is stopped when the PWM5 pin is low, one PWM cycle may be shorter than the programmed value.

TC5DR is serially connected to the shift register. If TC5DR is programmed during PWM output, the data set to TC5DR is not shifted until one PWM cycle is completed. Therefore, a pulse can be modulated periodically. For the first time, the data written to TC5DR is shifted when the timer is started by setting TC5CR<TC5S> to 1.

Note 1: The PWM output mode can be used only in the NORMAL1/2 and IDEL 1/2 modes.

Note 2: In the PWM output mode, program TC5DR immediately after the INTTC5 interrupt request is generated (typically in the INTTC5 interrupt service routine.) When the programming of TC5DR and the INTTC5 interrupt occur at the same time, an unstable value is shifted, that may result in generation of pulse different from the programmed value until the next INTTC5 interrupt request is issued.

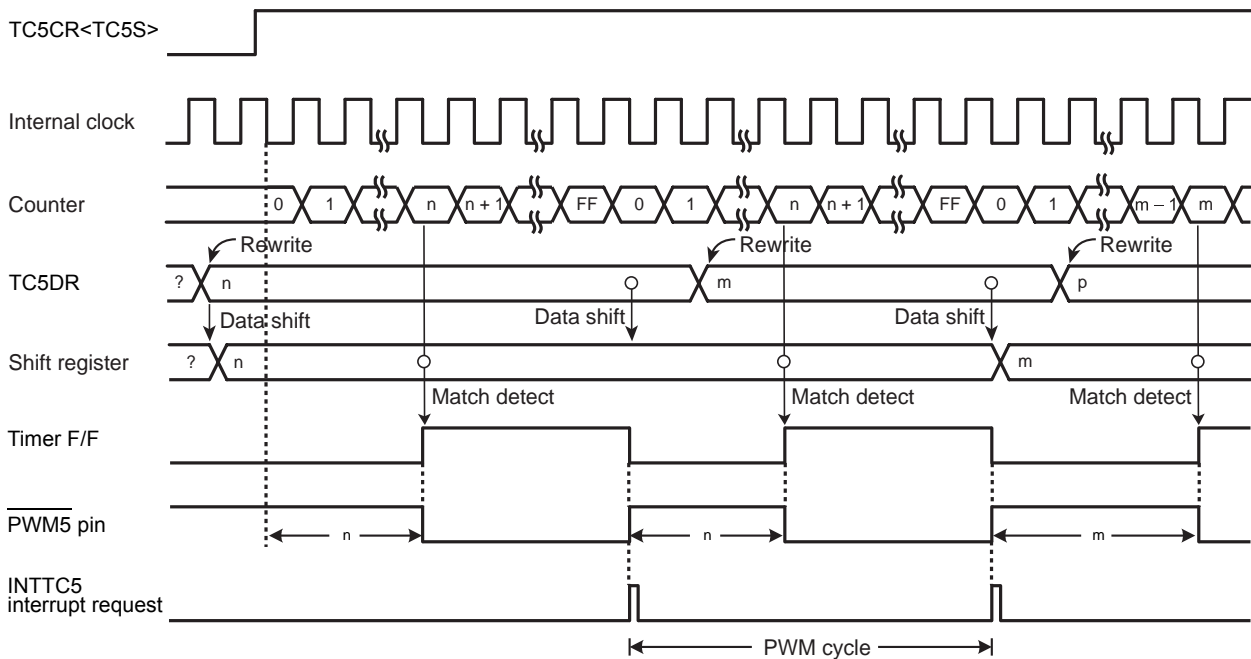


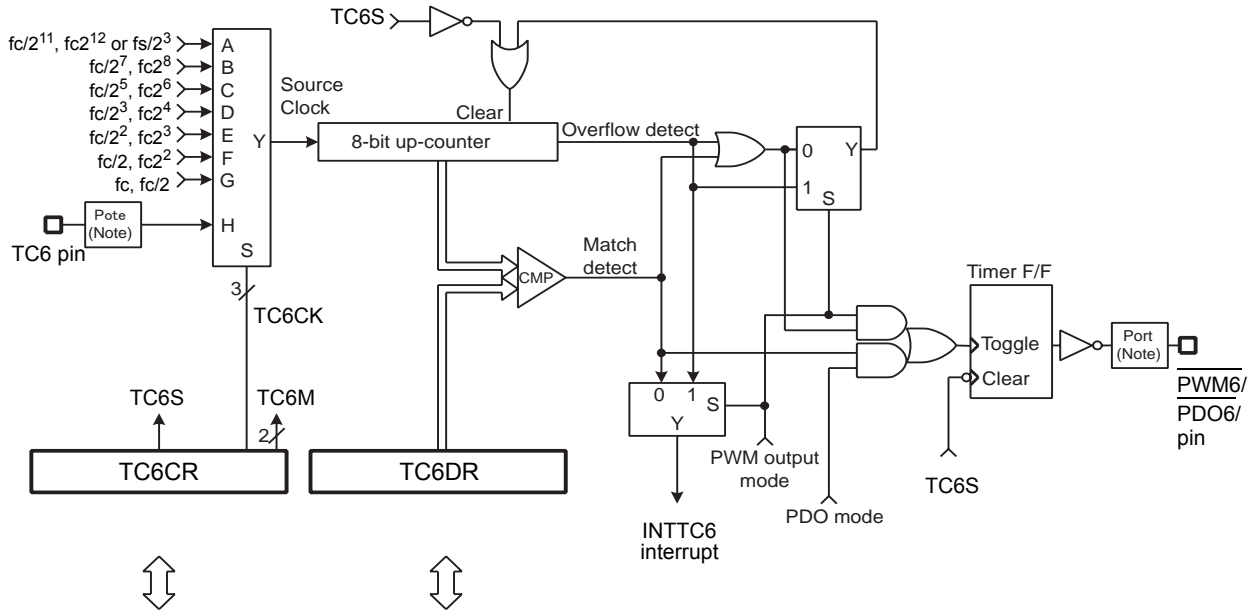
Figure 12-3 PWM output Mode Timing Chart (TC5)

Table 12-3 PWM Mode (Example: $f_c = 16$ MHz)

TC5CK	NORMAL1/2, IDLE1/2 Mode							
	DV7CK = 0				DV7CK = 1			
	DV1CK = 0		DV1CK = 1		DV1CK = 0		DV1CK = 1	
	Resolution [ns]	Cycle [μ s]	Resolution [ns]	Cycle [μ s]	Resolution [ns]	Cycle [μ s]	Resolution [ns]	Cycle [μ s]
000	-	-	-	-	-	-	-	-
001	-	-	-	-	-	-	-	-
010	-	-	-	-	-	-	-	-
011	500	128	1000	256	500	128	1000	256
100	250	64	500	128	250	64	500	128
101	125	32	250	64	125	32	250	64
110	-	-	-	-	-	-	-	-

13. 8-Bit TimerCounter 6 (TC6)

13.1 Configuration



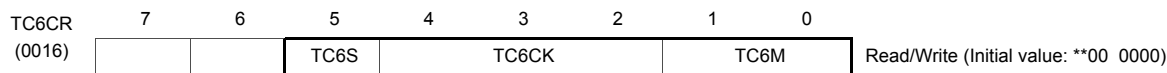
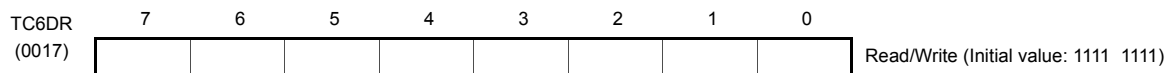
Note: Function I/O may not operate depending on I/O port setting. For more details, see the chapter "I/O Port".

Figure 13-1 TimerCounter 6 (TC6)

13.2 TimerCounter Control

The TimerCounter 6 is controlled by the TimerCounter 6 control register (TC6CR) and timer registers 6 (TC6DR).

Timer Register and Control Register



TC6S	TC6 start control	0: Stop and counter clear 1: Start						R/W	
TC6CK	TC6 source clock select [Hz]	NORMAL1/2, IDLE1/2 mode				Divider	SLOW1/2, SLEEP1/2 mode	R/W	
		DV7CK = 0		DV7CK = 1					
		DV1CK = 0	DV1CK = 1	DV1CK = 0	DV1CK = 1				
		000	$fc/2^{11}$	$fc/2^{12}$	$fs/2^3$	$fs/2^3$	DV9		$fs/2^3$
		001	$fc/2^7$	$fc/2^8$	$fc/2^7$	$fc/2^8$	DV5		–
		010	$fc/2^5$	$fc/2^6$	$fc/2^5$	$fc/2^6$	DV3		–
		011	$fc/2^3$	$fc/2^4$	$fc/2^3$	$fc/2^4$	DV1		–
		100	$fc/2^2$	$fc/2^3$	$fc/2^2$	$fc/2^3$	–		–
		101	$fc/2$	$fc/2^2$	$fc/2$	$fc/2^2$	–		–
110	fc	$fc/2$	fc	$fc/2$	–	–			
111	External clock (TC6 pin input)								
TC6M	TC6 operating mode select	00: Timer/event counter mode 01: Reserved 10: Programmable divider output (PDO) mode 11: Pulse width modulation (PWM) output mode						R/W	

Note 1: fc: High-frequency clock [Hz], fs: Low-frequency clock [Hz], *: Don't care

Note 2: To set the timer registers, the following relationship must be satisfied.
 $1 \leq TC6DR \leq 255$

Note 3: To start timer operation (TC6S = 0 → 1) or disable timer operation (TC6S = 1 → 0), do not change the TC6CR<TC6M, TC6CK> setting. During timer operation (TC6S = 1 → 1), do not change it, either. If the setting is programmed during timer operation, counting is not performed correctly.

Note 4: The event counter and PWM output modes are used only in the NOMAL1/2 and IDLE1/2 modes.

Note 5: When the STOP mode is entered, the start control (TC6S) is cleared to "0" automatically.

Note 6: The bit 6 and 7 of TC6CR are read as a don't care when these bits are read.

Note 7: In the timer, event counter and PDO modes, do not change the TC6DR setting when the timer is running.

Note 8: When the high-frequency clock fc exceeds 10 MHz, do not select the source clock of TC6CK = 110.

Note 9: The operating clock fs can not be used in NORMAL1 or IDEL1 mode (when low-frequency oscillation is stopped.)

Note 10: For available source clocks depending on the operation mode, refer to the following table.

		Timer Mode	Event Counter Mode	PDO Mode	PWM Mode
		TC6CK	000	O	–
	001	O	–	O	–
	010	O	–	O	–
	011	O	–	–	O
	100	–	–	–	O
	101	–	–	–	O
	110	–	–	–	O
	111	–	O	–	×

Note: O : Available source clock



13.3 Function

TimerCounter 6 has four types of operating modes: timer, event counter, programmable divider output (PDO), and pulse width modulation (PWM) output modes.

13.3.1 Timer Mode

In the timer mode, the up-counter counts up using the internal clock. When a match between the up-counter and the TC6DR value is detected, an INTTC6 interrupt is generated and the up-counter is cleared. After being cleared, the up-counter restarts counting.

Table 13-1 Source Clock for TimerCounter 6 (Example: $f_c = 16$ MHz, $f_s = 32.768$ kHz)

TC6CK	NORMAL1/2, IDLE1/2 Mode								SLOW1/2, SLEEP1/2 Mode	
	DV7CK = 0				DV7CK = 1					
	DV1CK = 0		DV1CK = 1		DV1CK = 0		DV1CK = 1		Resolution [μs]	Maximum Time Setting [ms]
	Resolution [μs]	Maximum Time Setting [ms]	Resolution [μs]	Maximum Time Setting [ms]	Resolution [μs]	Maximum Time Setting [ms]	Resolution [μs]	Maximum Time Setting [ms]		
000	128.0	32.6	256.0	65.3	244.14	62.2	244.14	62.2	244.14	62.2
001	8.0	2.0	16.0	4.1	8.0	2.0	16.0	4.1	–	–
010	2.0	0.510	4.0	1.0	2.0	0.510	4.0	1.0	–	–
011	0.5	0.128	1.0	0.255	0.5	0.128	1.0	0.255	–	–

13.3.2 Event Counter Mode

In the event counter mode, the up-counter counts up at the rising edge of the input pulse to the TC6 pin.

When a match between the up-counter and the TC6DR value is detected, an INTTC6 interrupt is generated and the up-counter is cleared. After being cleared, the up-counter restarts counting at rising edge of the TC6 pin. Since a match is detected at the falling edge of the input pulse to the TC6 pin, the INTTC6 interrupt request is generated at the falling edge immediately after the up-counter reaches the value set in TC6DR.

The minimum pulse width applied to the TC6 pin are shown in Table 13-2. The pulse width larger than two machine cycles is required for high- and low-going pulses.

Note: The event counter mode can not used in the SLOW1/2 and SLEEP1/2 modes since the external clock is not supplied in these modes.

Table 13-2 External Source Clock for TimerCounter 6

	Minimum Pulse Width
	NORMAL1/2, IDLE1/2 mode
High-going	$2^3/f_c$
Low-going	$2^3/f_c$

13.3.3 Programmable Divider Output (PDO) Mode

The programmable divider output (PDO) mode is used to generated a pulse with a 50% duty cycle by counting with the internal clock.

When a match between the up-counter and the TC6DR value is detected, the logic level output from the PDO6 pin is switched to the opposite state and INTTC6 interrupt request is generated. The up-counter is cleared at this time and then counting is continued. When a match between the up-counter and the TC6DR value is detected, the logic level output from the PDO6 pin is switched to the opposite state again and INTTC6 interrupt request is generated. The up-counter is cleared at this time, and then counting and PDO are continued.

When the timer is stopped, the PDO6 pin is high. Therefore, if the timer is stopped when the PDO6 pin is low, the duty pulse may be shorter than the programmed value.

Example :Generating 1024 Hz pulse (fc = 16.0 Mhz)

LD	(TC6CR), 00000110B	: Sets the PDO mode. (TC6M = 10, TC6CK = 001)
LD	(TC6DR), 3DH	: $1/1024 \div 2^7/fc \div 2$ (half cycle period) = 3DH
LD	(TC6CR), 00100110B	: Start TC6

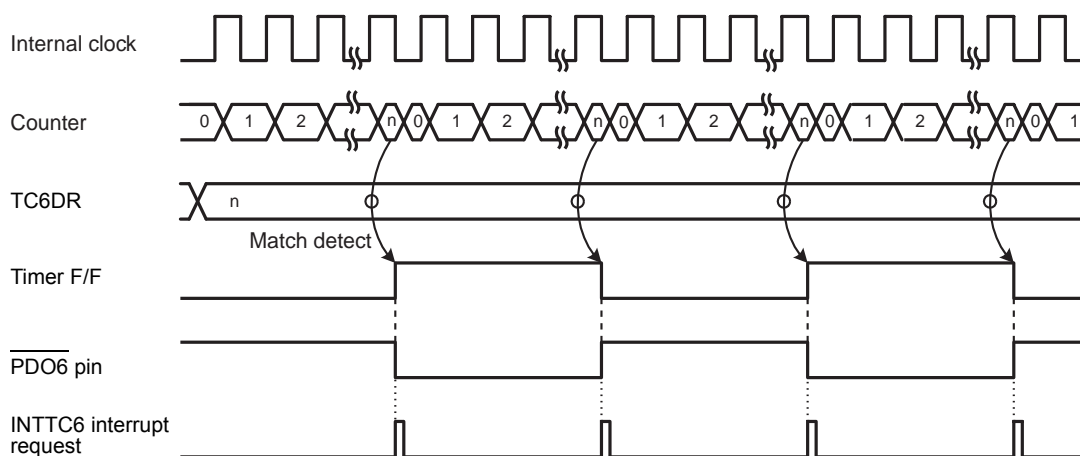


Figure 13-2 PDO Mode Timing Chart

13.3.4 Pulse Width Modulation (PWM) Output Mode

The pulse width modulation (PWM) output mode is used to generate the PWM pulse with up to 8 bits of resolution by an internal clock.

When a match between the up-counter and the TC6DR value is detected, the logic level output from the PWM6 pin becomes low. The up-counter continues counting. When the up-counter overflow occurs, the PWM6 pin becomes high. The INTTC6 interrupt request is generated at this time.

When the timer is stopped, the PWM6 pin is high. Therefore, if the timer is stopped when the PWM6 pin is low, one PMW cycle may be shorter than the programmed value.

TC6DR is serially connected to the shift register. If TC6DR is programmed during PWM output, the data set to TC6DR is not shifted until one PWM cycle is completed. Therefore, a pulse can be modulated periodically. For the first time, the data written to TC6DR is shifted when the timer is started by setting TC6CR<TC6S> to 1.

Note 1: The PWM output mode can be used only in the NORMAL1/2 and IDEL 1/2 modes.

Note 2: In the PWM output mode, program TC6DR immediately after the INTTC6 interrupt request is generated (typically in the INTTC6 interrupt service routine.) When the programming of TC6DR and the INTTC6 interrupt occur at the same time, an unstable value is shifted, that may result in generation of pulse different from the programmed value until the next INTTC6 interrupt request is issued.

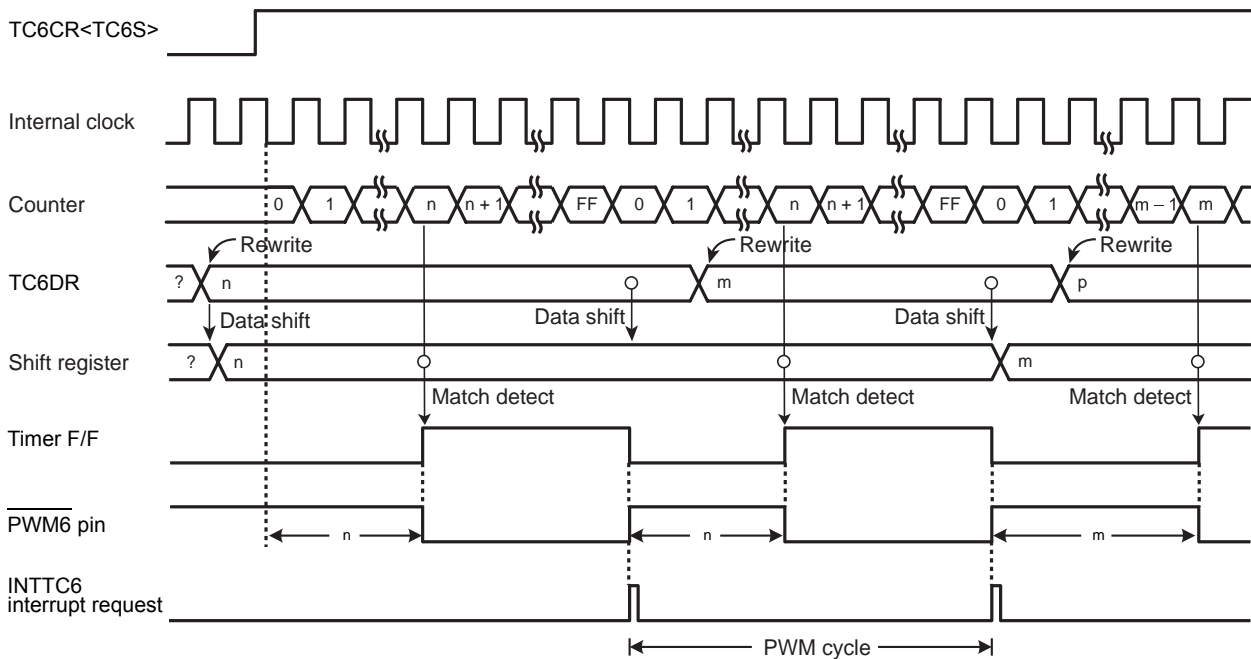


Figure 13-3 PWM output Mode Timing Chart (TC6)

Table 13-3 PWM Mode (Example: $f_c = 16$ MHz)

TC6CK	NORMAL1/2, IDLE1/2 Mode							
	DV7CK = 0				DV7CK = 1			
	DV1CK = 0		DV1CK = 1		DV1CK = 0		DV1CK = 1	
	Resolution [ns]	Cycle [μ s]	Resolution [ns]	Cycle [μ s]	Resolution [ns]	Cycle [μ s]	Resolution [ns]	Cycle [μ s]
000	–	–	–	–	–	–	–	–
001	–	–	–	–	–	–	–	–
010	–	–	–	–	–	–	–	–
011	500	128	1000	256	500	128	1000	256
100	250	64	500	128	250	64	500	128
101	125	32	250	64	125	32	250	64
110	–	–	–	–	–	–	–	–

14. Asynchronous Serial interface (UART)

14.1 Configuration

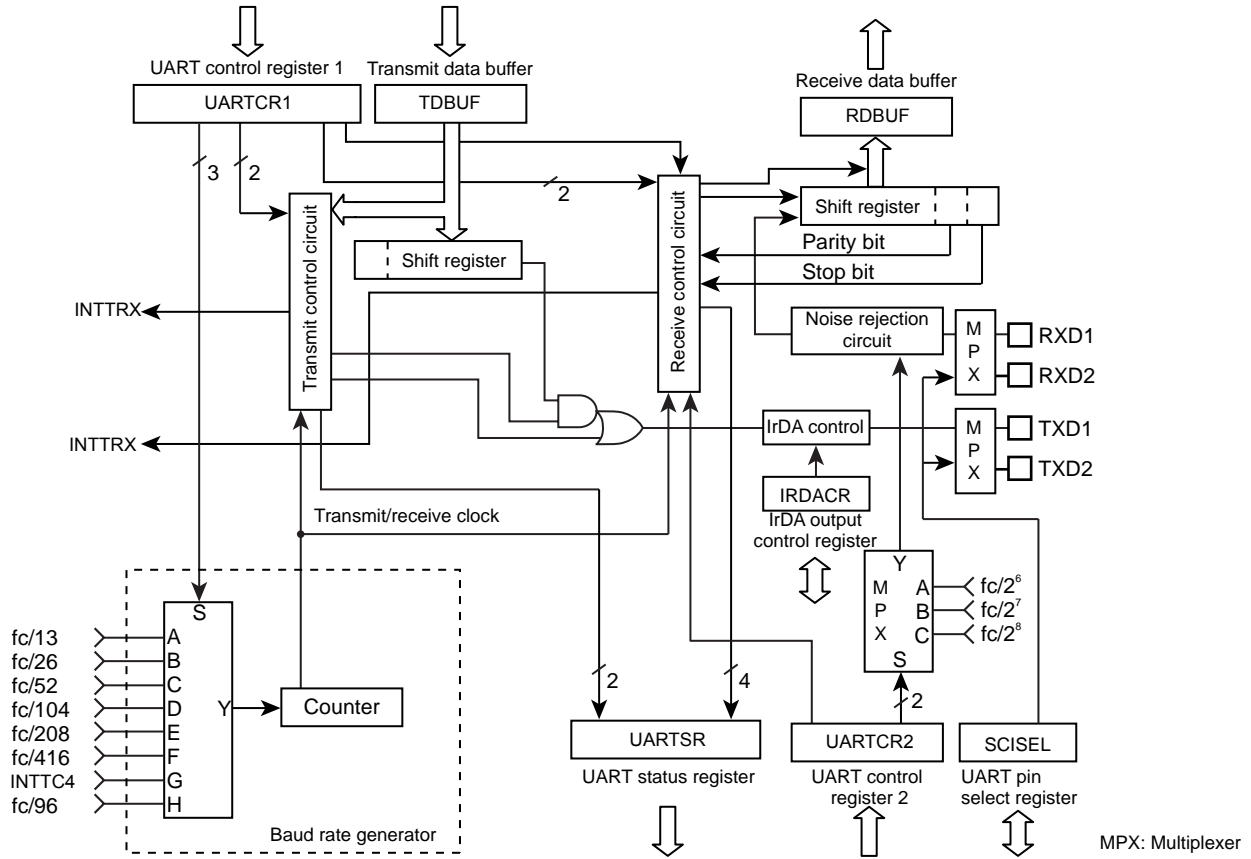


Figure 14-1 UART (Asynchronous Serial Interface)

14.2 Control

UART is controlled by the UART Control Registers (UARTCR1, UARTCR2). The operating status can be monitored using the UART status register (UARTSR).

TXD1 pin and RXD1 pin can be selected a port assignment by UART Pin Select Register (SCISEL). And Infrared data format (IrDA) output is available by setting IrDA output control register (IRDACR) through TXD1 pin.

UART Control Register1

UARTCR1 (001BH)	7	6	5	4	3	2	1	0	
	TXE	RXE	STBT	EVEN	PE	BRG			(Initial value: 0000 0000)

TXE	Transfer operation	0: Disable 1: Enable	Write only
RXE	Receive operation	0: Disable 1: Enable	
STBT	Transmit stop bit length	0: 1 bit 1: 2 bits	
EVEN	Even-numbered parity	0: Odd-numbered parity 1: Even-numbered parity	
PE	Parity addition	0: No parity 1: Parity	
BRG	Transmit clock select	000: fc/13 [Hz] 001: fc/26 010: fc/52 011: fc/104 100: fc/208 101: fc/416 110: TC4 (Input INTTC4) 111: fc/96	

Note 1: When operations are disabled by setting TXE and RXE bit to "0", the setting becomes valid when data transmit or receive complete. When the transmit data is stored in the transmit data buffer, the data are not transmitted. Even if data transmit is enabled, until new data are written to the transmit data buffer, the current data are not transmitted.

Note 2: The transmit clock and the parity are common to transmit and receive.

Note 3: UARTCR1<RXE> and UARTCR1<TXE> should be set to "0" before UARTCR1<BRG> is changed.

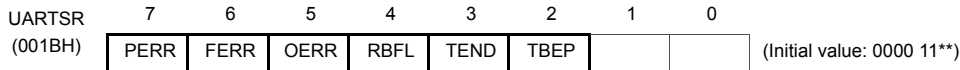
UART Control Register2

UARTCR2 (001CH)	7	6	5	4	3	2	1	0	
						RXDNC	STOPBR		(Initial value: **** *000)

RXDNC	Selection of RXD input noise rejectio time	00: No noise rejection (Hysteresis input) 01: Rejects pulses shorter than 31/fc [s] as noise 10: Rejects pulses shorter than 63/fc [s] as noise 11: Rejects pulses shorter than 127/fc [s] as noise	Write only
STOPBR	Receive stop bit length	0: 1 bit 1: 2 bits	

Note: When UARTCR2<RXDNC> = "01", pulses longer than 96/fc [s] are always regarded as signals; when UARTCR2<RXDNC> = "10", longer than 192/fc [s]; and when UARTCR2<RXDNC> = "11", longer than 384/fc [s].

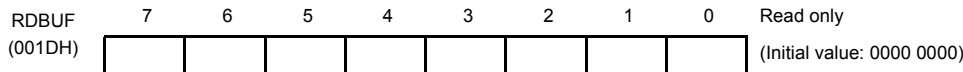
UART Status Register



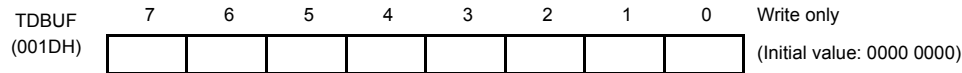
PERR	Parity error flag	0: No parity error 1: Parity error	Read only
FERR	Framing error flag	0: No framing error 1: Framing error	
OERR	Overrun error flag	0: No overrun error 1: Overrun error	
RBFL	Receive data buffer full flag	0: Receive data buffer empty 1: Receive data buffer full	
TEND	Transmit end flag	0: On transmitting 1: Transmit end	
TBEP	Transmit data buffer empty flag	0: Transmit data buffer full (Transmit data writing is finished) 1: Transmit data buffer empty	

Note: When an INTTXD is generated, TBEP flag is set to "1" automatically.

UART Receive Data Buffer



UART Transmit Data Buffer



UART Pin Select Register



TXDESEL	TXD connect pin select	0: P41 1: P44	R/W
RXDESEL	RXD connect pin select	0: P42 1: P45	

Note 1: Do not change SCISEL register during UART operation.

Note 2: Set SCISEL register before performing the setting terminal of a I/O port when changing a terminal.

14.3 Transfer Data Format

In UART, an one-bit start bit (Low level), stop bit (Bit length selectable at high level, by UARTCR1<STBT>), and parity (Select parity in UARTCR1<PE>; even- or odd-numbered parity by UARTCR1<EVEN>) are added to the transfer data. The transfer data formats are shown as follows.

PE	STBT	Frame Length									
		1	2	3		8	9	10	11	12	
0	0										
0	1										
1	0										
1	1										

Figure 14-2 Transfer Data Format

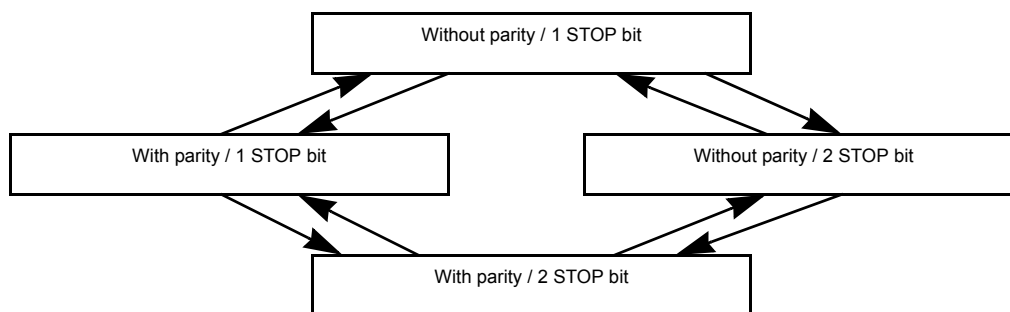


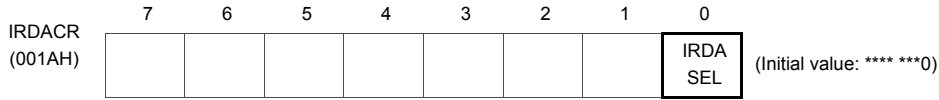
Figure 14-3 Caution on Changing Transfer Data Format

Note: In order to switch the transfer data format, perform transmit operations in the above Figure 14-3 sequence except for the initial setting.

14.4 Infrared (IrDA) Data Format Transfer Mode

Infrared data format (IrDA) output is available by setting IrDA output control register (IRDACR) through TXD1 pin.

IrDA Output Control Register



IRDASEL	IrDA output / UART output select	0: UART output 1: IrDA output	R/W
---------	----------------------------------	----------------------------------	-----

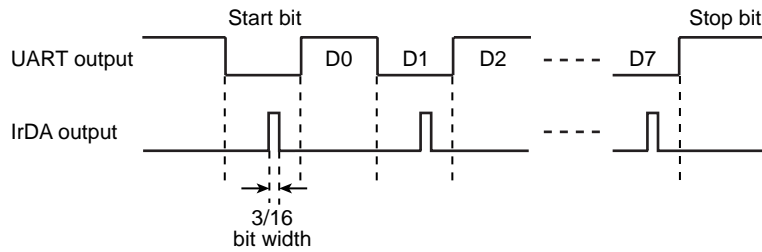


Figure 14-4 Example of Infrared Data Format (Comparison of Normal output and IrDA output)

14.5 Transfer Rate

The baud rate of UART is set of UARTCR1<BRG>. The example of the baud rate are shown as follows.

Table 14-1 Transfer Rate (Example)

BRG	Source Clock		
	16 MHz	8 MHz	4 MHz
000	76800 [baud]	38400 [baud]	19200 [baud]
001	38400	19200	9600
010	19200	9600	4800
011	9600	4800	2400
100	4800	2400	1200
101	2400	1200	600

When TC4 is used as the UART transfer rate (when UARTCR1<BRG> = “110”), the transfer clock and transfer rate are determined as follows:

$$\text{Transfer clock [Hz]} = \text{TC4 source clock [Hz]} / \text{TTREG4 setting value}$$

$$\text{Transfer Rate [baud]} = \text{Transfer clock [Hz]} / 16$$

14.6 Data Sampling Method

The UART receiver keeps sampling input using the clock selected by UARTCR1<BRG> until a start bit is detected in RXD1 pin input. RT clock starts detecting “L” level of the RXD1 pin. Once a start bit is detected, the start bit, data bits, stop bit(s), and parity bit are sampled at three times of RT7, RT8, and RT9 during one receiver clock interval (RT clock). (RT0 is the position where the bit supposedly starts.) Bit is determined according to majority rule (The data are the same twice or more out of three samplings).

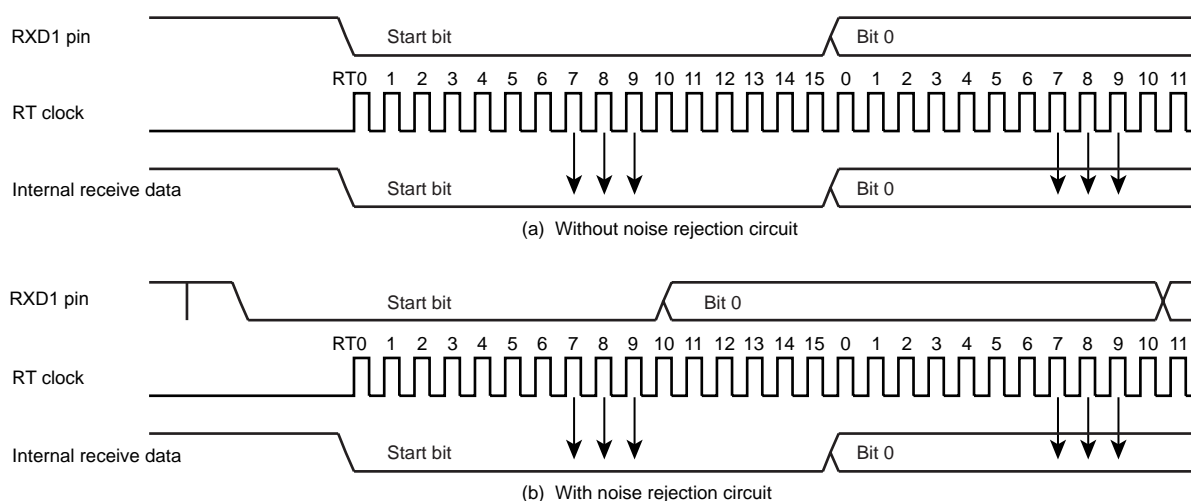


Figure 14-5 Data Sampling Method

14.7 STOP Bit Length

Select a transmit stop bit length (1 bit or 2 bits) by UARTCR1<STBT>.

14.8 Parity

Set parity / no parity by UARTCR1<PE> and set parity type (Odd- or Even-numbered) by UARTCR1<EVEN>.

14.9 Transmit/Receive Operation

14.9.1 Data Transmit Operation

Set UARTCR1<TXE> to “1”. Read UARTSR to check UARTSR<TBEP> = “1”, then write data in TDBUF (Transmit data buffer). Writing data in TDBUF zero-clears UARTSR<TBEP>, transfers the data to the transmit shift register and the data are sequentially output from the TXD1 pin. The data output include a one-bit start bit, stop bits whose number is specified in UARTCR1<STBT> and a parity bit if parity addition is specified. Select the data transfer baud rate using UARTCR1<BRG>. When data transmit starts, transmit buffer empty flag UARTSR<TBEP> is set to “1” and an INTTRX interrupt is generated.

While UARTCR1<TXE> = “0” and from when “1” is written to UARTCR1<TXE> to when send data are written to TDBUF, the TXD1 pin is fixed at high level.

When transmitting data, first read UARTSR, then write data in TDBUF. Otherwise, UARTSR<TBEP> is not zero-cleared and transmit does not start.

14.9.2 Data Receive Operation

Set UARTCR1<RXE> to “1”. When data are received via the RXD1 pin, the receive data are transferred to RDBUF (Receive data buffer). At this time, the data transmitted includes a start bit and stop bit(s) and a parity bit if parity addition is specified. When stop bit(s) are received, data only are extracted and transferred to RDBUF (Receive data buffer). Then the receive buffer full flag UARTSR<RBFL> is set and an INTTRX interrupt is generated. Select the data transfer baud rate using UARTCR1<BRG>.

If an overrun error (OERR) occurs when data are received, the data are not transferred to RDBUF (Receive data buffer) but discarded; data in the RDBUF are not affected.

Note: When a receive operation is disabled by setting UARTCR1<RXE> bit to “0”, the setting becomes valid when data receive is completed. However, if a framing error occurs in data receive, the receive-disabling setting may not become valid. If a framing error occurs, be sure to perform a re-receive operation.

14.10 Status Flag

14.10.1 Parity Error

When parity determined using the receive data bits differs from the received parity bit, the parity error flag UARTSR<PERR> is set to “1”. The UARTSR<PERR> is cleared to “0” when the RDBUF is read after reading the UARTSR.

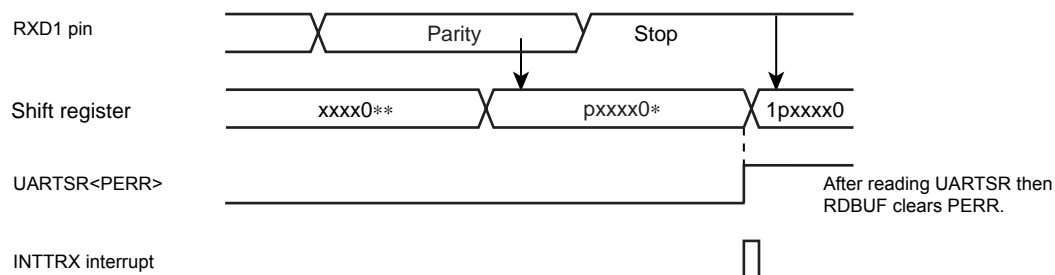


Figure 14-6 Generation of Parity Error

14.10.2 Framing Error

When “0” is sampled as the stop bit in the receive data, framing error flag UARTSR<FERR> is set to “1”. The UARTSR<FERR> is cleared to “0” when the RDBUF is read after reading the UARTSR.

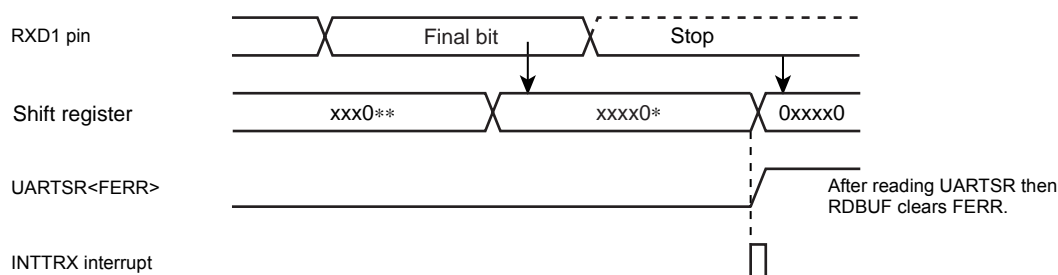


Figure 14-7 Generation of Framing Error

14.10.3 Overrun Error

When all bits in the next data are received while unread data are still in RDBUF, overrun error flag UARTSR<OERR> is set to “1”. In this case, the receive data is discarded; data in RDBUF are not affected. The UARTSR<OERR> is cleared to “0” when the RDBUF is read after reading the UARTSR.

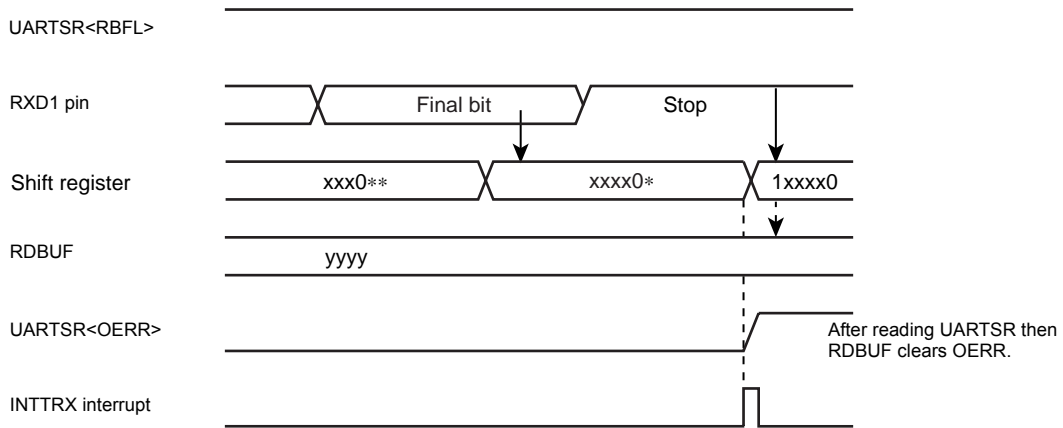


Figure 14-8 Generation of Overrun Error

Note: Receive operations are disabled until the overrun error flag UARTSR<OERR> is cleared.

14.10.4 Receive Data Buffer Full

Loading the received data in RDBUF sets receive data buffer full flag UARTSR<RBFL> to "1". The UARTSR<RBFL> is cleared to "0" when the RDBUF is read after reading the UARTSR.

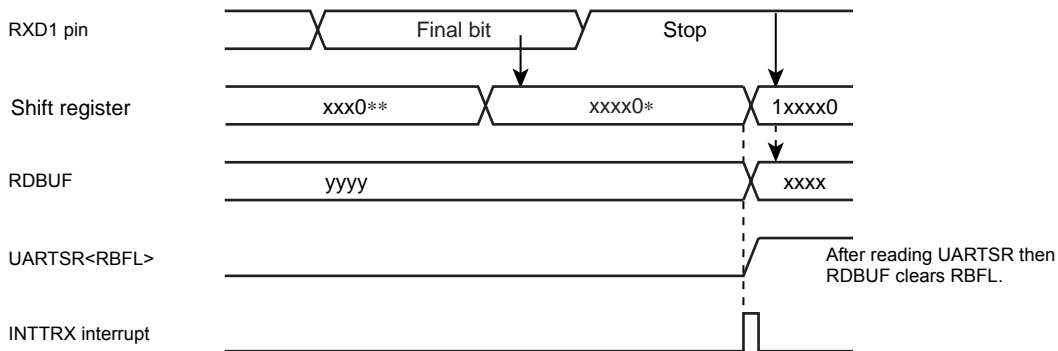


Figure 14-9 Generation of Receive Data Buffer Full

Note: If the overrun error flag UARTSR<OERR> is set during the period between reading the UARTSR and reading the RDBUF, it cannot be cleared by only reading the RDBUF. Therefore, after reading the RDBUF, read the UARTSR again to check whether or not the overrun error flag which should have been cleared still remains set.

14.10.5 Transmit Data Buffer Empty

When no data is in the transmit buffer TDBUF, UARTSR<TBEP> is set to "1", that is, when data in TDBUF are transferred to the transmit shift register and data transmit starts, transmit data buffer empty flag UARTSR<TBEP> is set to "1". The UARTSR<TBEP> is cleared to "0" when the TDBUF is written after reading the UARTSR.

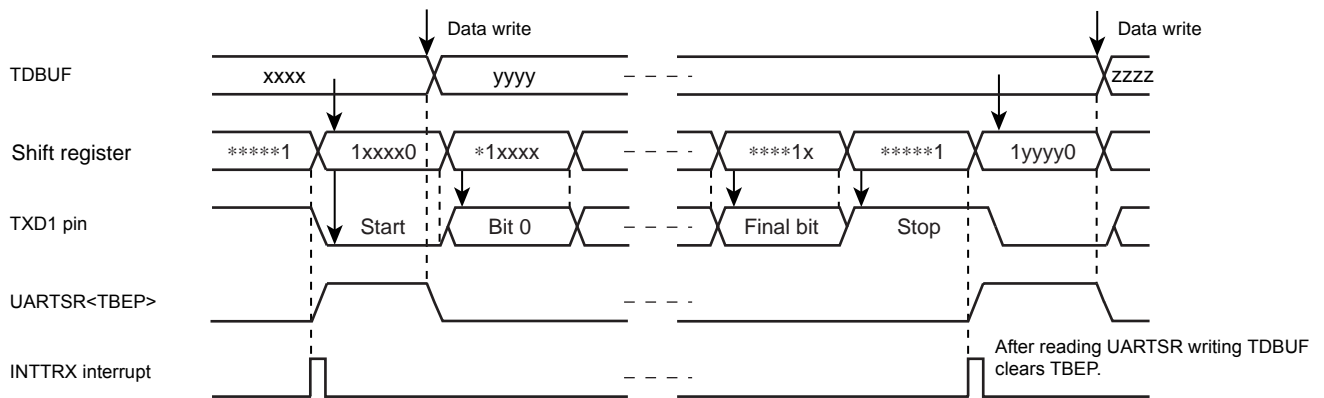


Figure 14-10 Generation of Transmit Data Buffer Empty

14.10.6 Transmit End Flag

When data are transmitted and no data is in TDBUF (UARTSR<TBEP> = “1”), transmit end flag UARTSR<TEND> is set to “1”. The UARTSR<TEND> is cleared to “0” when the data transmit is started after writing the TDBUF.

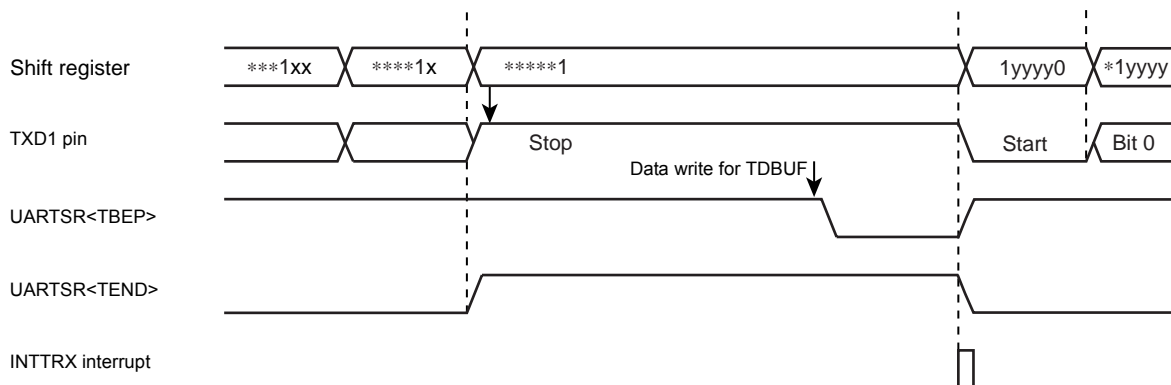


Figure 14-11 Generation of Transmit End Flag and Transmit Data Buffer Empty

15. Synchronous Serial Interface (SIO1)

The TMP86FS64FG has a clocked-synchronous 8-bit serial interface. Serial interface has an 8-byte transmit and receive data buffer that can automatically and continuously transfer up to 64 bits of data.

Serial interface is connected to outside peripheral devices via SO1, SI1, SCK1 port.

15.1 Configuration

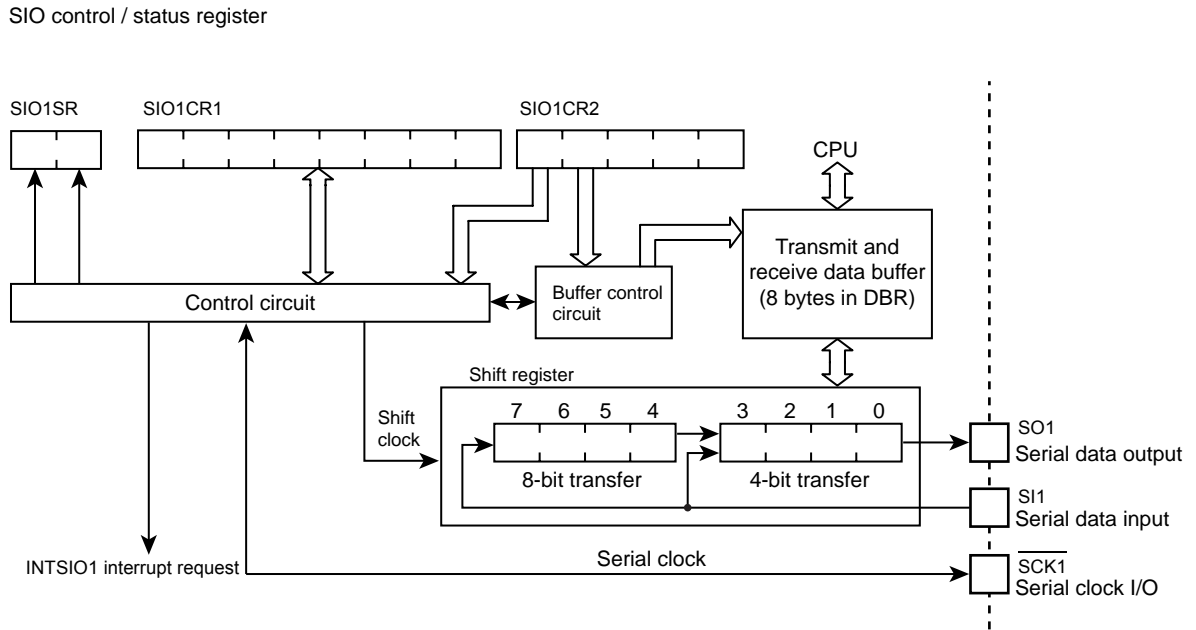


Figure 15-1 Serial Interface

15.2 Control

The serial interface is controlled by SIO control registers (SIO1CR1/SIO1CR2). The serial interface status can be determined by reading SIO status register (SIO1SR).

The transmit and receive data buffer is controlled by the SIO1CR2<BUF>. The data buffer is assigned to address 0F90H to 0F97H for SIO in the DBR area, and can continuously transfer up to 8 words (bytes or nibbles) at one time. When the specified number of words has been transferred, a buffer empty (in the transmit mode) or a buffer full (in the receive mode or transmit/receive mode) interrupt (INTSIO1) is generated.

When the internal clock is used as the serial clock in the 8-bit receive mode and the 8-bit transmit/receive mode, a fixed interval wait can be applied to the serial clock for each word transferred. Four different wait times can be selected with SIO1CR2<WAIT>.

SIO Control Register 1

SIO1CR1	7	6	5	4	3	2	1	0	
(0028H)	SIOS	SIOINH	SIOM			SCK			(Initial value: 0000 0000)

SIOS	Indicate transfer start / stop	0: Stop 1: Start					Write only	
SIOINH	Continue / abort transfer	0: Continuously transfer 1: Abort transfer (Automatically cleared after abort)						
SIOM	Transfer mode select	000: 8-bit transmit mode 010: 4-bit transmit mode 100: 8-bit transmit / receive mode 101: 8-bit receive mode 110: 4-bit receive mode Except the above: Reserved					Write only	
SCK	Serial clock select		NORMAL1/2, IDLE1/2 mode					SLOW1/2 SLEEP1/2 mode
			DV7CK = 0		DV7CK = 1			
			DV1CK = 0	DV1CK = 1	DV1CK = 1	DV1CK = 1		
			000	$fc/2^{13}$	$fc/2^{14}$	$fs/2^5$	$fs/2^5$	$fs/2^5$
			001	$fc/2^8$	$fc/2^9$	$fc/2^8$	$fc/2^9$	-
			010	$fc/2^7$	$fc/2^8$	$fc/2^7$	$fc/2^8$	-
			011	$fc/2^6$	$fc/2^7$	$fc/2^6$	$fc/2^7$	-
			100	$fc/2^5$	$fc/2^6$	$fc/2^5$	$fc/2^6$	-
101	$fc/2^4$	$fc/2^5$	$fc/2^4$	$fc/2^5$	-			
110	Reserved							
111	External clock (Input from SCK1 pin)							

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

Note 2: Set SIOS to "0" and SIOINH to "1" when setting the transfer mode or serial clock.

Note 3: SIO1CR1 is write-only register, which cannot access any of in read-modify-write instruction such as bit operate, etc.

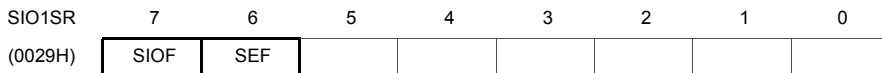
SIO Control Register 2

SIO1CR2	7	6	5	4	3	2	1	0	
(0029H)				WAIT		BUF			(Initial value: ***0 0000)

WAIT	Wait control	Always sets "00" except 8-bit transmit / receive mode. 00: $T_f = T_D$ (Non wait) 01: $T_f = 2T_D$ (Wait) 10: $T_f = 4T_D$ (Wait) 11: $T_f = 8T_D$ (Wait)	Write only
BUF	Number of transfer words (Buffer address in use)	000: 1 word transfer 0F90H 001: 2 words transfer 0F90H ~ 0F91H 010: 3 words transfer 0F90H ~ 0F92H 011: 4 words transfer 0F90H ~ 0F93H 100: 5 words transfer 0F90H ~ 0F94H 101: 6 words transfer 0F90H ~ 0F95H 110: 7 words transfer 0F90H ~ 0F96H 111: 8 words transfer 0F90H ~ 0F97H	

- Note 1: The lower 4 bits of each buffer are used during 4-bit transfers. Zeros (0) are stored to the upper 4bits when receiving.
- Note 2: Transmitting starts at the lowest address. Received data are also stored starting from the lowest address to the highest address. (The first buffer address transmitted is 0F90H).
- Note 3: The value to be loaded to BUF is held after transfer is completed.
- Note 4: SIO1CR2 must be set when the serial interface is stopped (SIOF = 0).
- Note 5: *: Don't care
- Note 6: SIO1CR2 is write-only register, which cannot access any of in read-modify-write instruction such as bit operate, etc.

SIO Status Register



SIOF	Serial transfer operating status monitor	0: Transfer terminated 1: Transfer in process	Read only
SEF	Shift operating status monitor	0: Shift operation terminated 1: Shift operation in process	

- Note 1: T_f : Frame time, T_D : Data transfer time
- Note 2: After SIOS is cleared to "0", SIOF is cleared to "0" at the termination of transfer or the setting of SIOINH to "1".

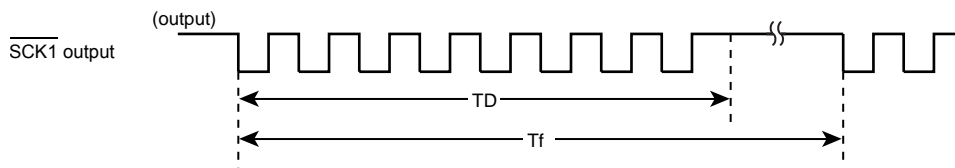


Figure 15-2 Frame time (T_f) and Data transfer time (T_D)

15.3 Serial clock

15.3.1 Clock source

Internal clock or external clock for the source clock is selected by SIO1CR1<SCK>.

15.3.1.1 Internal clock

Any of six frequencies can be selected. The serial clock is output to the outside on the SCK1 pin. The SCK1 pin goes high when transfer starts.

When data writing (in the transmit mode) or reading (in the receive mode or the transmit/receive mode) cannot keep up with the serial clock rate, there is a wait function that automatically stops the serial clock and holds the next shift operation until the read/write processing is completed.

Table 15-1 Serial Clock Rate

SCK	NORMAL 1/2, IDLE 1/2 mode								SLOW 1/2, SLEEP 1/2 mode	
	DV7CK = 0				DV7CK = 1					
	DV1CK = 0		DV1CK = 1		DV1CK = 0		DV1CK = 1			
	Clock	Baud Rate	Clock	Baud Rate	Clock	Baud Rate	Clock	Baud Rate		
000	$fc/2^{13}$	1.91 Kbps	$fc/2^{14}$	0.95 Kbps	$fs/2^5$	1024 bps	$fs/2^5$	1024 bps	$fs/2^5$	1024 bps
001	$fc/2^8$	61.04 Kbps	$fc/2^9$	30.52 Kbps	$fc/2^8$	61.04 Kbps	$fc/2^9$	30.52 Kbps	-	-
010	$fc/2^7$	122.07 Kbps	$fc/2^8$	61.04 Kbps	$fc/2^7$	122.07 Kbps	$fc/2^8$	61.04 Kbps	-	-
011	$fc/2^6$	244.14 Kbps	$fc/2^7$	122.07 Kbps	$fc/2^6$	244.14 Kbps	$fc/2^7$	122.07 Kbps	-	-
100	$fc/2^5$	488.28 Kbps	$fc/2^6$	244.14 Kbps	$fc/2^5$	488.28 Kbps	$fc/2^6$	244.14 Kbps	-	-
101	$fc/2^4$	976.56 Kbps	$fc/2^5$	488.28 Kbps	$fc/2^4$	976.56 Kbps	$fc/2^5$	488.28 Kbps	-	-
110	-	-	-	-	-	-	-	-	-	-
111	External	External	External	External	External	External	External	External	External	External

Note: 1 Kbit = 1024 bit ($fc = 16$ MHz, $fs = 32.768$ kHz)

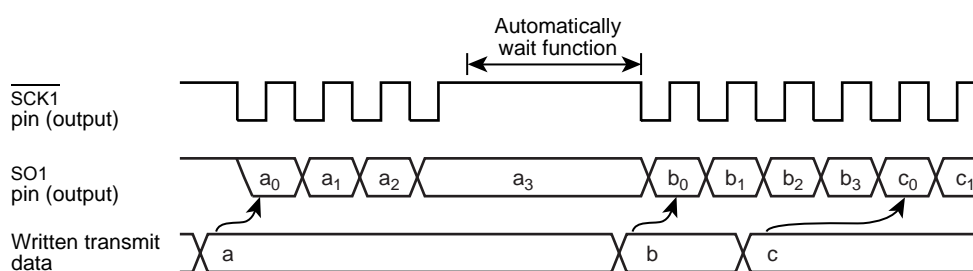


Figure 15-3 Automatic Wait Function (at 4-bit transmit mode)

15.3.1.2 External clock

An external clock connected to the $\overline{SCK1}$ pin is used as the serial clock. In this case, output latch of this port should be set to "1". To ensure shifting, a pulse width of at least 4 machine cycles is required. This pulse is needed for the shift operation to execute certainly. Actually, there is necessary processing time for interrupting, writing, and reading. The minimum pulse is determined by setting the mode and the program. Therefore, maximum transfer frequency will be 488.3K bit/sec (at $fc=16$ MHz).

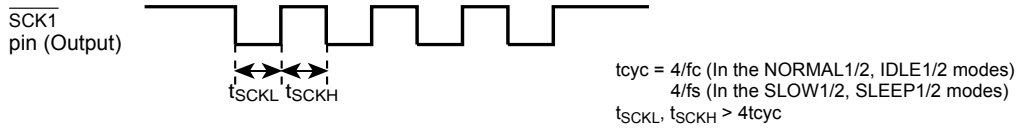


Figure 15-4 External clock pulse width

15.3.2 Shift edge

The leading edge is used to transmit, and the trailing edge is used to receive.

15.3.2.1 Leading edge

Transmitted data are shifted on the leading edge of the serial clock (falling edge of the $\overline{\text{SCK1}}$ pin input/output).

15.3.2.2 Trailing edge

Received data are shifted on the trailing edge of the serial clock (rising edge of the $\overline{\text{SCK1}}$ pin input/output).

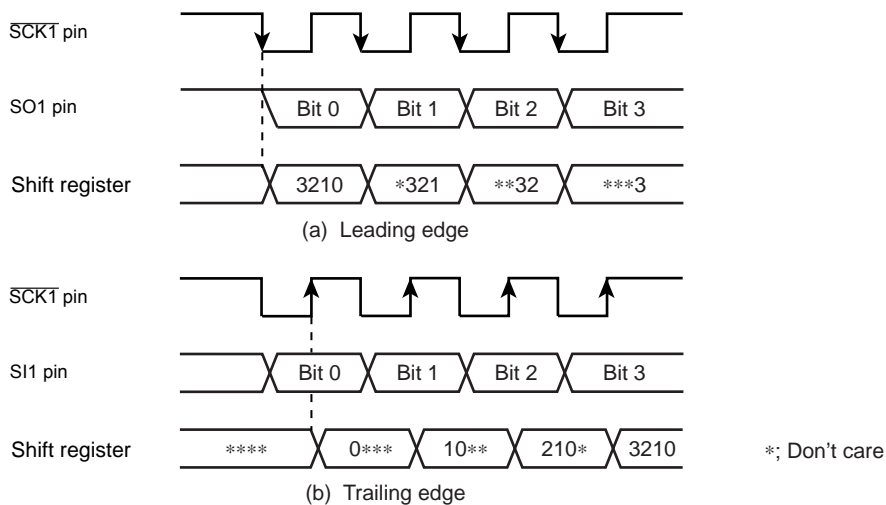


Figure 15-5 Shift edge

15.4 Number of bits to transfer

Either 4-bit or 8-bit serial transfer can be selected. When 4-bit serial transfer is selected, only the lower 4 bits of the transmit/receive data buffer register are used. The upper 4 bits are cleared to “0” when receiving. The data is transferred in sequence starting at the least significant bit (LSB).

15.5 Number of words to transfer

Up to 8 words consisting of 4 bits of data (4-bit serial transfer) or 8 bits (8-bit serial transfer) of data can be transferred continuously. The number of words to be transferred can be selected by SIO1CR2<BUF>.

An INTSIO1 interrupt is generated when the specified number of words has been transferred. If the number of words is to be changed during transfer, the serial interface must be stopped before making the change. The number of words can be changed during automatic-wait operation of an internal clock. In this case, the serial interface is not required to be stopped.

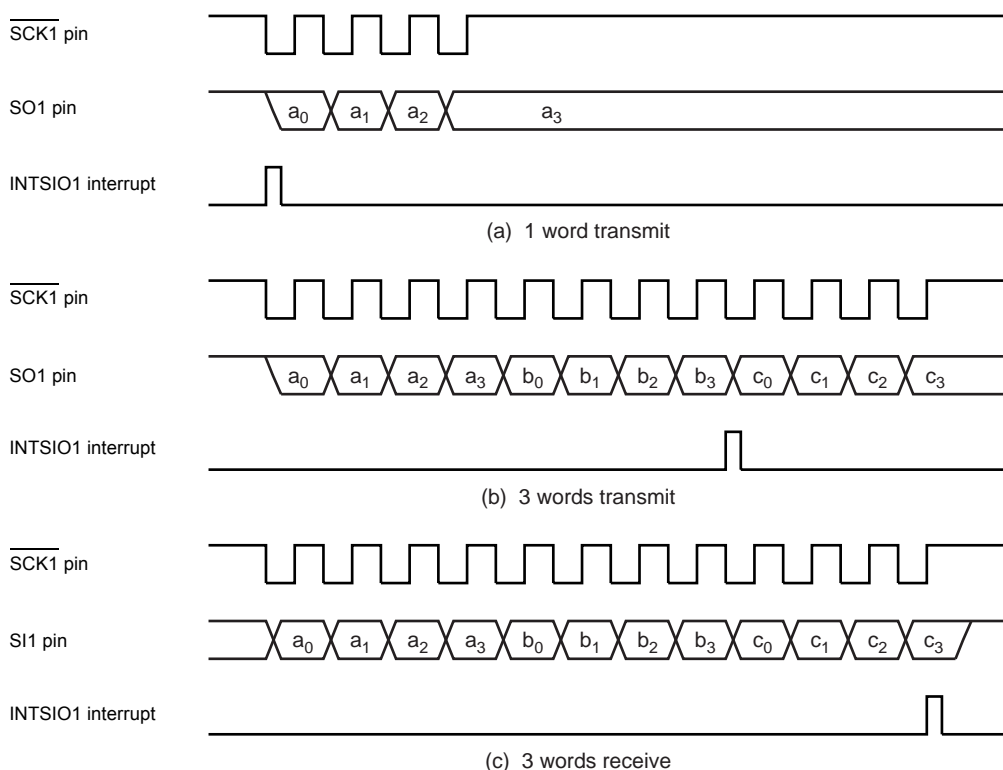


Figure 15-6 Number of words to transfer (Example: 1 word = 4bit)

15.6 Transfer Mode

SIO1CR1<SIOM> is used to select the transmit, receive, or transmit/receive mode.

15.6.1 4-bit and 8-bit transfer modes

In these modes, firstly set the SIO control register to the transmit mode, and then write first transmit data (number of transfer words to be transferred) to the data buffer registers (DBR).

After the data are written, the transmission is started by setting SIO1CR1<SIOS> to “1”. The data are then output sequentially to the SO pin in synchronous with the serial clock, starting with the least significant bit (LSB). As soon as the LSB has been output, the data are transferred from the data buffer register to the shift register. When the final data bit has been transferred and the data buffer register is empty, an INTSIO1 (Buffer empty) interrupt is generated to request the next transmitted data.

When the internal clock is used, the serial clock will stop and an automatic-wait will be initiated if the next transmitted data are not loaded to the data buffer register by the time the number of data words specified with the SIO1CR2<BUF> has been transmitted. Writing even one word of data cancels the automatic-wait; therefore, when transmitting two or more words, always write the next word before transmission of the previous word is completed.

Note: Automatic waits are also canceled by writing to a DBR not being used as a transmit data buffer register; therefore, during SIO do not use such DBR for other applications. For example, when 3 words are transmitted, do not use the DBR of the remained 5 words.

When an external clock is used, the data must be written to the data buffer register before shifting next data. Thus, the transfer speed is determined by the maximum delay time from the generation of the interrupt request to writing of the data to the data buffer register by the interrupt service program.

The transmission is ended by clearing SIO1CR1<SIOS> to “0” or setting SIO1CR1<SIOINH> to “1” in buffer empty interrupt service program.

SIO1CR1<SIOS> is cleared, the operation will end after all bits of words are transmitted.

That the transmission has ended can be determined from the status of SIO1SR<SIOF> because SIO1SR<SIOF> is cleared to “0” when a transfer is completed.

When SIO1CR1<SIOINH> is set, the transmission is immediately ended and SIO1SR<SIOF> is cleared to “0”.

When an external clock is used, it is also necessary to clear SIO1CR1<SIOS> to “0” before shifting the next data; If SIO1CR1<SIOS> is not cleared before shift out, dummy data will be transmitted and the operation will end.

If it is necessary to change the number of words, SIO1CR1<SIOS> should be cleared to “0”, then SIO1CR2<BUF> must be rewritten after confirming that SIO1SR<SIOF> has been cleared to “0”.

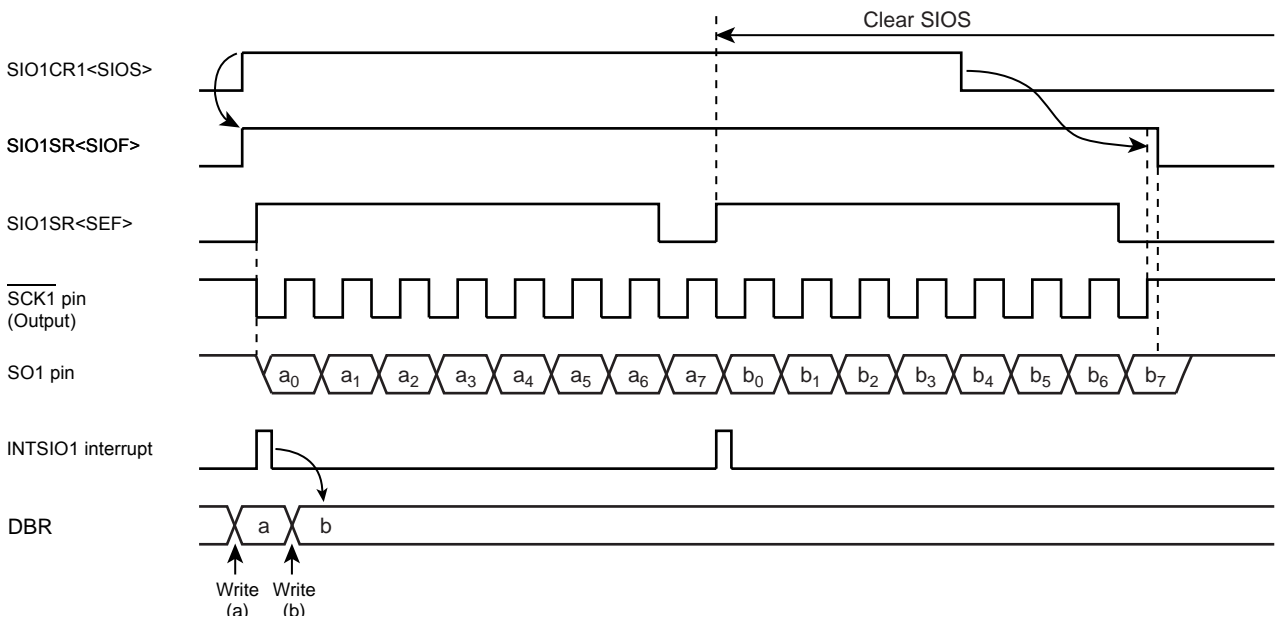


Figure 15-7 Transfer Mode (Example: 8bit, 1word transfer, Internal clock)

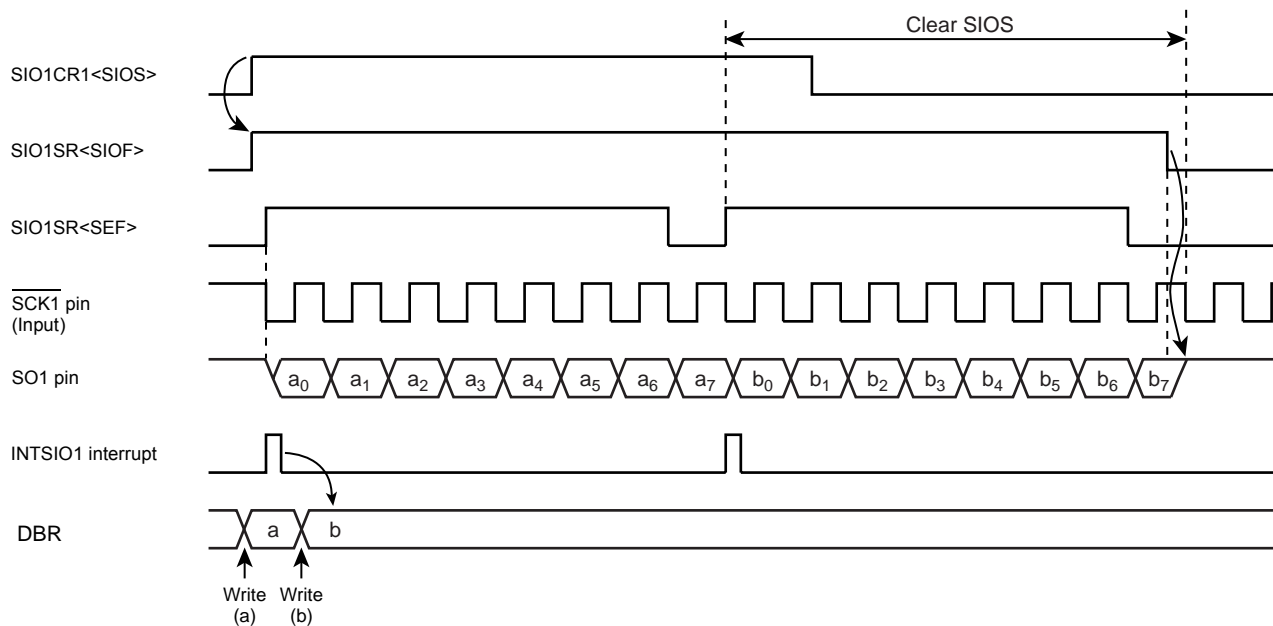


Figure 15-8 Transfer Mode (Example: 8bit, 1word transfer, External clock)

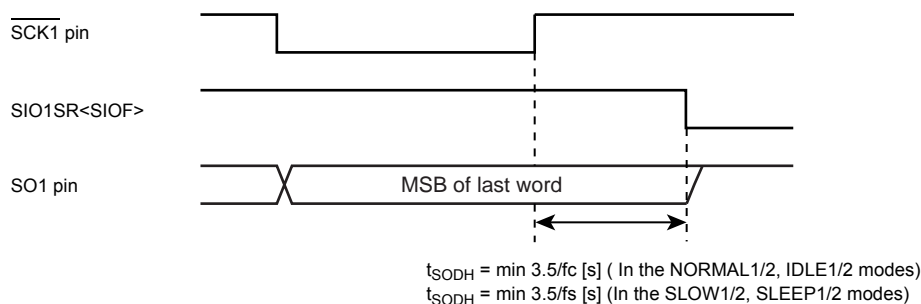


Figure 15-9 Transmitted Data Hold Time at End of Transfer

15.6.2 4-bit and 8-bit receive modes

After setting the control registers to the receive mode, set SIO1CR1<SIOS> to “1” to enable receiving. The data are then transferred to the shift register via the SI pin in synchronous with the serial clock. When one word of data has been received, it is transferred from the shift register to the data buffer register (DBR). When the number of words specified with the SIO1CR2<BUF> has been received, an INTSIO1 (Buffer full) interrupt is generated to request that these data be read out. The data are then read from the data buffer registers by the interrupt service program.

When the internal clock is used, and the previous data are not read from the data buffer register before the next data are received, the serial clock will stop and an automatic-wait will be initiated until the data are read. A wait will not be initiated if even one data word has been read.

Note: Waits are also canceled by reading a DBR not being used as a received data buffer register is read; therefore, during SIO1 do not use such DBR for other applications.

When an external clock is used, the shift operation is synchronized with the external clock; therefore, the previous data are read before the next data are transferred to the data buffer register. If the previous data have not been read, the next data will not be transferred to the data buffer register and the receiving of any more data will be canceled. When an external clock is used, the maximum transfer speed is determined by the delay between the time when the interrupt request is generated and when the data received have been read.

The receiving is ended by clearing SIO1CR1<SIOS> to “0” or setting SIO1CR1<SIOINH> to “1” in buffer full interrupt service program.

When SIO1CR1<SIOS> is cleared, the current data are transferred to the buffer. After SIO1CR1<SIOS> cleared, the receiving is ended at the time that the final bit of the data has been received. That the receiving has ended can be determined from the status of SIO1SR<SIOF>. SIO1SR<SIOF> is cleared to “0” when the receiving is ended. After confirmed the receiving termination, the final receiving data is read. When SIO1CR1<SIOINH> is set, the receiving is immediately ended and SIO1SR<SIOF> is cleared to “0”. (The received data is ignored, and it is not required to be read out.)

If it is necessary to change the number of words in external clock operation, SIO1CR1<SIOS> should be cleared to “0” then SIO1CR2<BUF> must be rewritten after confirming that SIO1SR<SIOF> has been cleared to “0”. If it is necessary to change the number of words in internal clock, during automatic-wait operation which occurs after completion of data receiving, SIO1CR2<BUF> must be rewritten before the received data is read out.

Note: The buffer contents are lost when the transfer mode is switched. If it should become necessary to switch the transfer mode, end receiving by clearing SIO1CR1<SIOS> to “0”, read the last data and then switch the transfer mode.

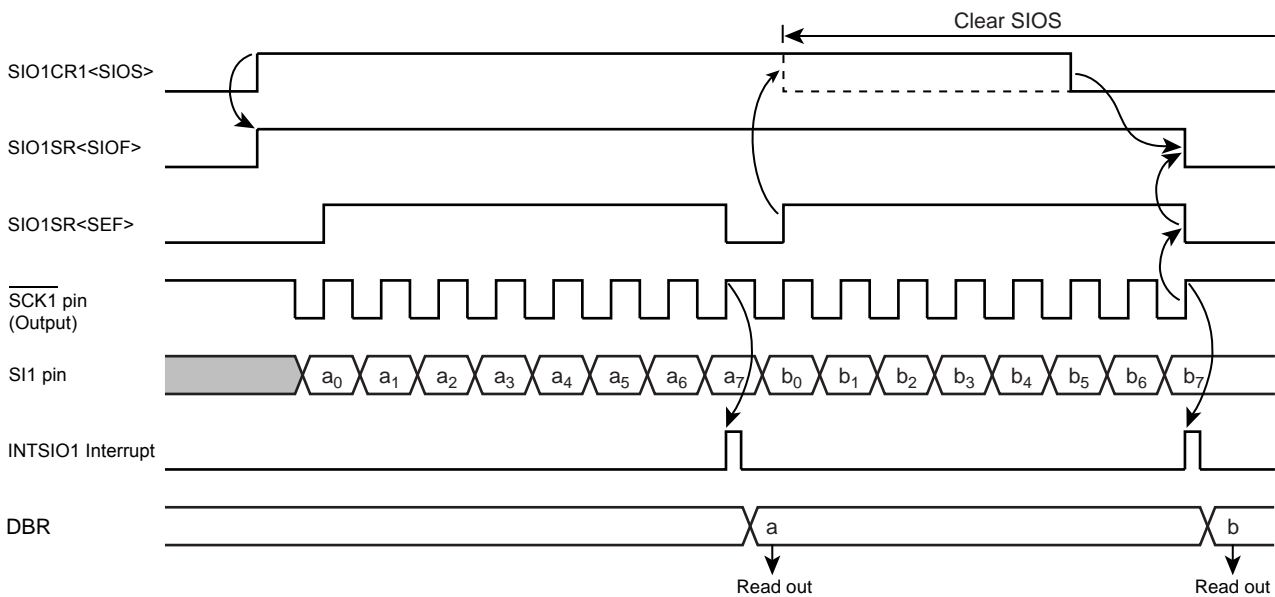


Figure 15-10 Receive Mode (Example: 8bit, 1word transfer, Internal clock)

15.6.3 8-bit transfer / receive mode

After setting the SIO control register to the 8-bit transmit/receive mode, write the data to be transmitted first to the data buffer registers (DBR). After that, enable the transmit/receive by setting SIO1CR1<SIOS> to “1”. When transmitting, the data are output from the SO1 pin at leading edges of the serial clock. When receiving, the data are input to the S11 pin at the trailing edges of the serial clock. When the all receive is enabled, 8-bit data are transferred from the shift register to the data buffer register. An INTSIO1 interrupt is generated when the number of data words specified with the SIO1CR2<BUF> has been transferred. Usually, read the receive data from the buffer register in the interrupt service. The data buffer register is used for both transmitting and receiving; therefore, always write the data to be transmitted after reading the all received data.

When the internal clock is used, a wait is initiated until the received data are read and the next transfer data are written. A wait will not be initiated if even one transfer data word has been written.

When an external clock is used, the shift operation is synchronized with the external clock; therefore, it is necessary to read the received data and write the data to be transmitted next before starting the next shift operation. When an external clock is used, the transfer speed is determined by the maximum delay between generation of an interrupt request and the received data are read and the data to be transmitted next are written.

The transmit/receive operation is ended by clearing SIO1CR1<SIOS> to “0” or setting SIO1CR1<SIOINH> to “1” in INTSIO1 interrupt service program.

When SIO1CR1<SIOS> is cleared, the current data are transferred to the buffer. After SIO1CR1<SIOS> cleared, the transmitting/receiving is ended at the time that the final bit of the data has been transmitted.

That the transmitting/receiving has ended can be determined from the status of SIO1SR<SIOF>. SIO1SR<SIOF> is cleared to “0” when the transmitting/receiving is ended.

When SIO1CR1<SIOINH> is set, the transmit/receive operation is immediately ended and SIO1SR<SIOF> is cleared to “0”.

If it is necessary to change the number of words in external clock operation, SIO1CR1<SIOS> should be cleared to “0”, then SIO1CR2<BUF> must be rewritten after confirming that SIO1SR<SIOF> has been cleared to “0”.

If it is necessary to change the number of words in internal clock, during automatic-wait operation which occurs after completion of transmit/receive operation, SIO1CR2<BUF> must be rewritten before reading and writing of the receive/transmit data.

Note: The buffer contents are lost when the transfer mode is switched. If it should become necessary to switch the transfer mode, end receiving by clearing SIO1CR1<SIOS> to “0”, read the last data and then switch the transfer mode.

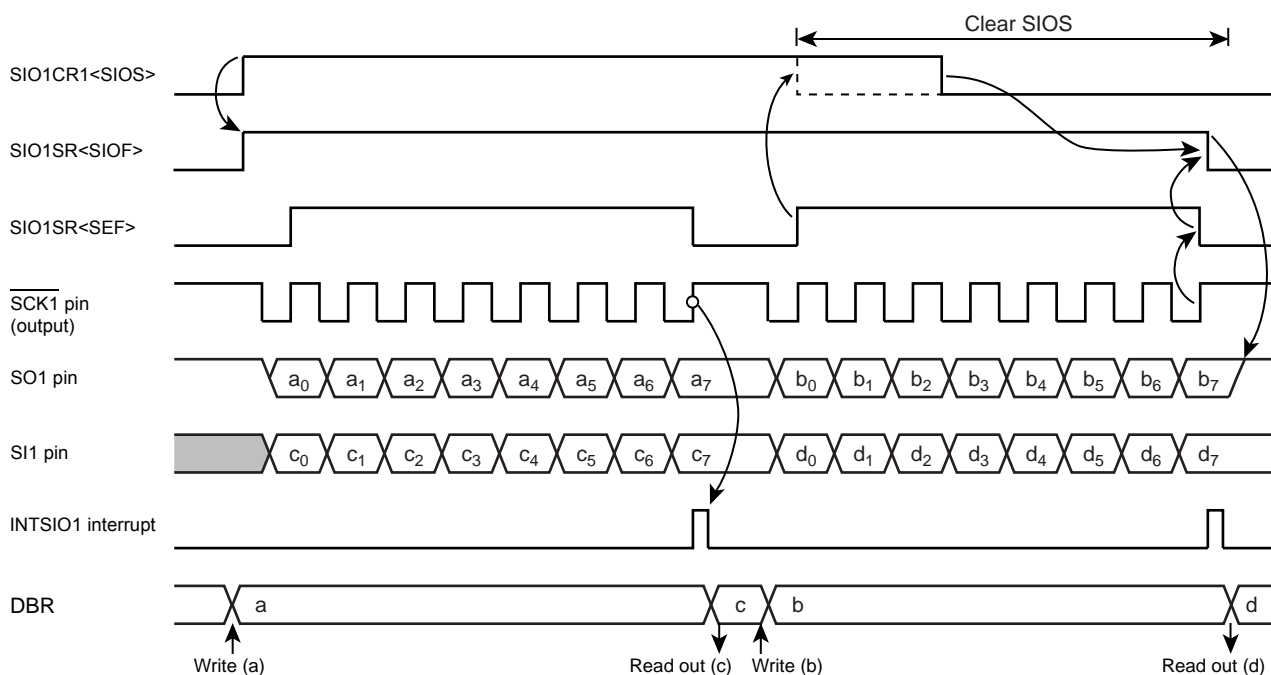


Figure 15-11 Transfer / Receive Mode (Example: 8bit, 1word transfer, Internal clock)

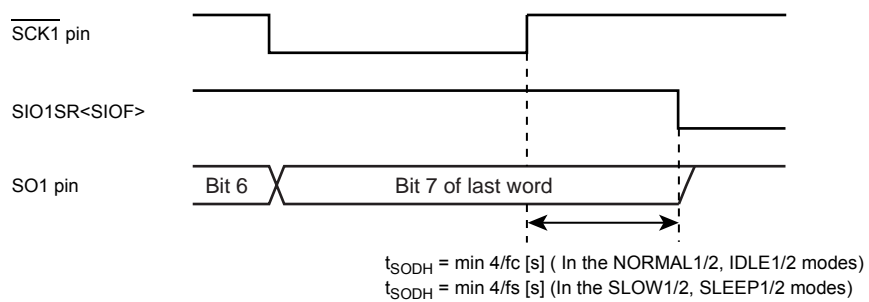


Figure 15-12 Transmitted Data Hold Time at End of Transfer / Receive

16. Synchronous Serial Interface (SIO2)

The TMP86FS64FG has a clocked-synchronous 8-bit serial interface. Serial interface has an 8-byte transmit and receive data buffer that can automatically and continuously transfer up to 64 bits of data.

Serial interface is connected to outside peripheral devices via SO2, SI2, SCK2 port.

16.1 Configuration

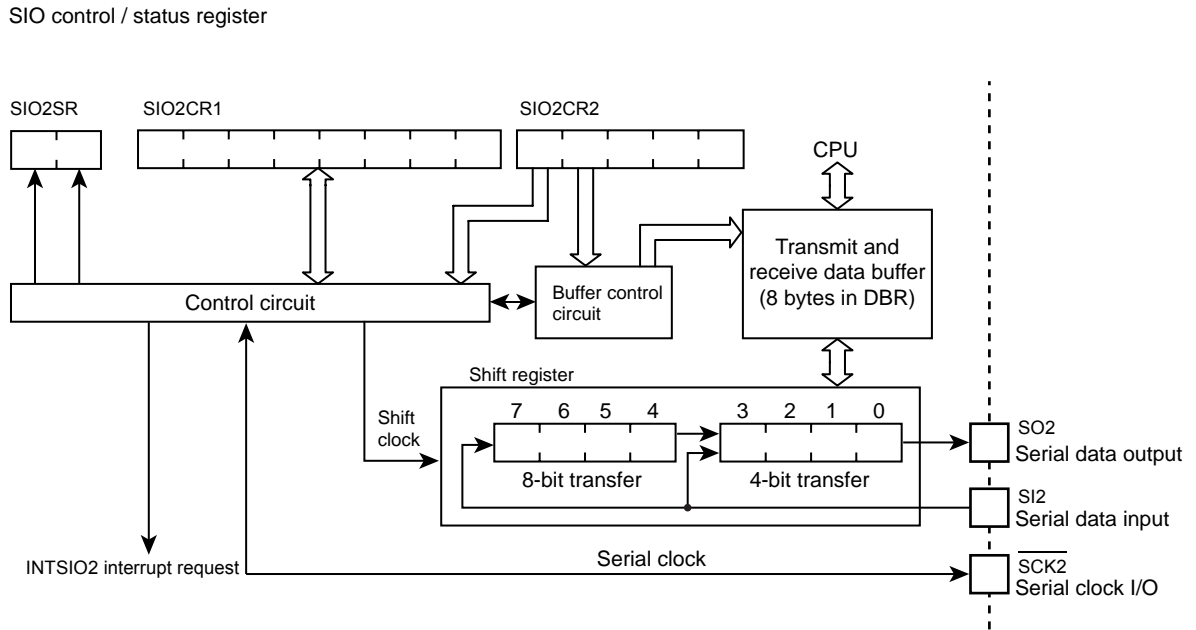


Figure 16-1 Serial Interface

16.2 Control

The serial interface is controlled by SIO control registers (SIO2CR1/SIO2CR2). The serial interface status can be determined by reading SIO status register (SIO2SR).

The transmit and receive data buffer is controlled by the SIO2CR2<BUF>. The data buffer is assigned to address 0F98H to 0F9FH for SIO in the DBR area, and can continuously transfer up to 8 words (bytes or nibbles) at one time. When the specified number of words has been transferred, a buffer empty (in the transmit mode) or a buffer full (in the receive mode or transmit/receive mode) interrupt (INTSIO2) is generated.

When the internal clock is used as the serial clock in the 8-bit receive mode and the 8-bit transmit/receive mode, a fixed interval wait can be applied to the serial clock for each word transferred. Four different wait times can be selected with SIO2CR2<WAIT>.

SIO Control Register 1

SIO2CR1 (0FB4H)	7	6	5	4	3	2	1	0	
	SIOS	SIOINH	SIOM			SCK			(Initial value: 0000 0000)

SIOS	Indicate transfer start / stop	0: Stop 1: Start					Write only		
SIOINH	Continue / abort transfer	0: Continuously transfer 1: Abort transfer (Automatically cleared after abort)							
SIOM	Transfer mode select	000: 8-bit transmit mode 010: 4-bit transmit mode 100: 8-bit transmit / receive mode 101: 8-bit receive mode 110: 4-bit receive mode Except the above: Reserved							
SCK	Serial clock select		NORMAL1/2, IDLE1/2 mode				SLOW1/2 SLEEP1/2 mode	Write only	
			DV7CK = 0		DV7CK = 1				
			DV1CK = 0	DV1CK = 1	DV1CK = 1	DV1CK = 1			
			000	$fc/2^{15}$	$fc/2^{16}$	$fs/2^7$	$fs/2^7$		$fs/2^7$
			001	$fc/2^8$	$fc/2^9$	$fc/2^8$	$fc/2^9$		-
			010	$fc/2^7$	$fc/2^8$	$fc/2^7$	$fc/2^8$		-
			011	$fc/2^6$	$fc/2^7$	$fc/2^6$	$fc/2^7$		-
			100	$fc/2^5$	$fc/2^6$	$fc/2^5$	$fc/2^6$		-
101	$fc/2^4$	$fc/2^5$	$fc/2^4$	$fc/2^5$	-				
110	Reserved								
111	External clock (Input from SCK2 pin)								

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

Note 2: Set SIOS to "0" and SIOINH to "1" when setting the transfer mode or serial clock.

Note 3: SIO2CR1 is write-only register, which cannot access any of in read-modify-write instruction such as bit operate, etc.

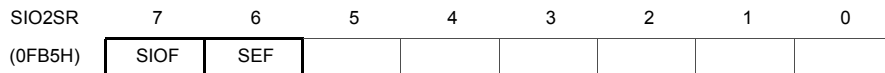
SIO Control Register 2

SIO2CR2 (0FB5H)	7	6	5	4	3	2	1	0	
				WAIT		BUF			(Initial value: ***0 0000)

WAIT	Wait control	Always sets "00" except 8-bit transmit / receive mode. 00: $T_f = T_D$ (Non wait) 01: $T_f = 2T_D$ (Wait) 10: $T_f = 4T_D$ (Wait) 11: $T_f = 8T_D$ (Wait)	Write only
BUF	Number of transfer words (Buffer address in use)	000: 1 word transfer 0F98H 001: 2 words transfer 0F98H ~ 0F99H 010: 3 words transfer 0F98H ~ 0F9AH 011: 4 words transfer 0F98H ~ 0F9BH 100: 5 words transfer 0F98H ~ 0F9CH 101: 6 words transfer 0F98H ~ 0F9DH 110: 7 words transfer 0F98H ~ 0F9EH 111: 8 words transfer 0F98H ~ 0F9FH	

- Note 1: The lower 4 bits of each buffer are used during 4-bit transfers. Zeros (0) are stored to the upper 4bits when receiving.
- Note 2: Transmitting starts at the lowest address. Received data are also stored starting from the lowest address to the highest address. (The first buffer address transmitted is 0F98H).
- Note 3: The value to be loaded to BUF is held after transfer is completed.
- Note 4: SIO2CR2 must be set when the serial interface is stopped (SIOF = 0).
- Note 5: *: Don't care
- Note 6: SIO2CR2 is write-only register, which cannot access any of in read-modify-write instruction such as bit operate, etc.

SIO Status Register



SIOF	Serial transfer operating status monitor	0: Transfer terminated 1: Transfer in process	Read only
SEF	Shift operating status monitor	0: Shift operation terminated 1: Shift operation in process	

- Note 1: T_f : Frame time, T_D : Data transfer time
- Note 2: After SIOS is cleared to "0", SIOF is cleared to "0" at the termination of transfer or the setting of SIOINH to "1".

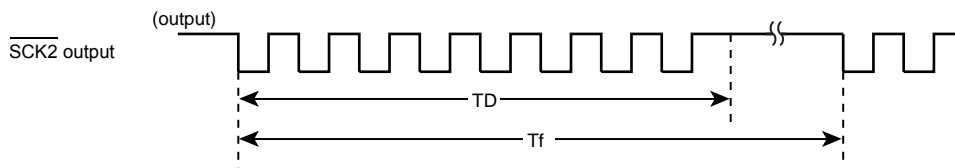


Figure 16-2 Frame time (T_f) and Data transfer time (T_D)

16.3 Serial clock

16.3.1 Clock source

Internal clock or external clock for the source clock is selected by SIO2CR1<SCK>.

16.3.1.1 Internal clock

Any of six frequencies can be selected. The serial clock is output to the outside on the SCK2 pin. The SCK2 pin goes high when transfer starts.

When data writing (in the transmit mode) or reading (in the receive mode or the transmit/receive mode) cannot keep up with the serial clock rate, there is a wait function that automatically stops the serial clock and holds the next shift operation until the read/write processing is completed.

Table 16-1 Serial Clock Rate

SCK	NORMAL 1/2, IDLE 1/2 mode								SLOW 1/2, SLEEP 1/2 mode	
	DV7CK = 0				DV7CK = 1					
	DV1CK = 0		DV1CK = 1		DV1CK = 0		DV1CK = 1			
	Clock	Baud Rate	Clock	Baud Rate	Clock	Baud Rate	Clock	Baud Rate		
000	$fc/2^{15}$	0.48 Kbps	$fc/2^{16}$	0.24 Kbps	$fs/2^7$	256 bps	$fs/2^7$	256 bps	$fs/2^7$	256 bps
001	$fc/2^8$	61.04 Kbps	$fc/2^9$	30.52 Kbps	$fc/2^8$	61.04 Kbps	$fc/2^9$	30.52 Kbps	-	-
010	$fc/2^7$	122.07 Kbps	$fc/2^8$	61.04 Kbps	$fc/2^7$	122.07 Kbps	$fc/2^8$	61.04 Kbps	-	-
011	$fc/2^6$	244.14 Kbps	$fc/2^7$	122.07 Kbps	$fc/2^6$	244.14 Kbps	$fc/2^7$	122.07 Kbps	-	-
100	$fc/2^5$	488.28 Kbps	$fc/2^6$	244.14 Kbps	$fc/2^5$	488.28 Kbps	$fc/2^6$	244.14 Kbps	-	-
101	$fc/2^4$	976.56 Kbps	$fc/2^5$	488.28 Kbps	$fc/2^4$	976.56 Kbps	$fc/2^5$	488.28 Kbps	-	-
110	-	-	-	-	-	-	-	-	-	-
111	External	External	External	External	External	External	External	External	External	External

Note: 1 Kbit = 1024 bit ($fc = 16$ MHz, $fs = 32.768$ kHz)

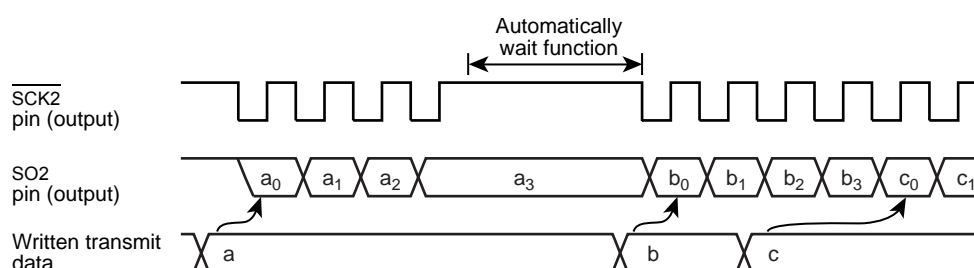


Figure 16-3 Automatic Wait Function (at 4-bit transmit mode)

16.3.1.2 External clock

An external clock connected to the $\overline{SCK2}$ pin is used as the serial clock. In this case, output latch of this port should be set to "1". To ensure shifting, a pulse width of at least 4 machine cycles is required. This pulse is needed for the shift operation to execute certainly. Actually, there is necessary processing time for interrupting, writing, and reading. The minimum pulse is determined by setting the mode and the program. Therefore, maximum transfer frequency will be 488.3K bit/sec (at $fc=16$ MHz).

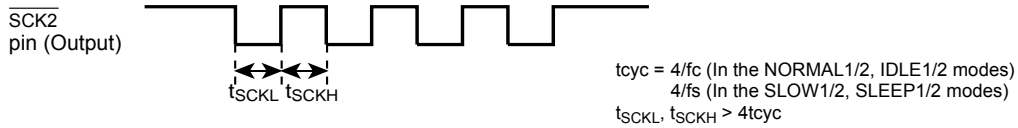


Figure 16-4 External clock pulse width

16.3.2 Shift edge

The leading edge is used to transmit, and the trailing edge is used to receive.

16.3.2.1 Leading edge

Transmitted data are shifted on the leading edge of the serial clock (falling edge of the $\overline{\text{SCK2}}$ pin input/output).

16.3.2.2 Trailing edge

Received data are shifted on the trailing edge of the serial clock (rising edge of the $\overline{\text{SCK2}}$ pin input/output).

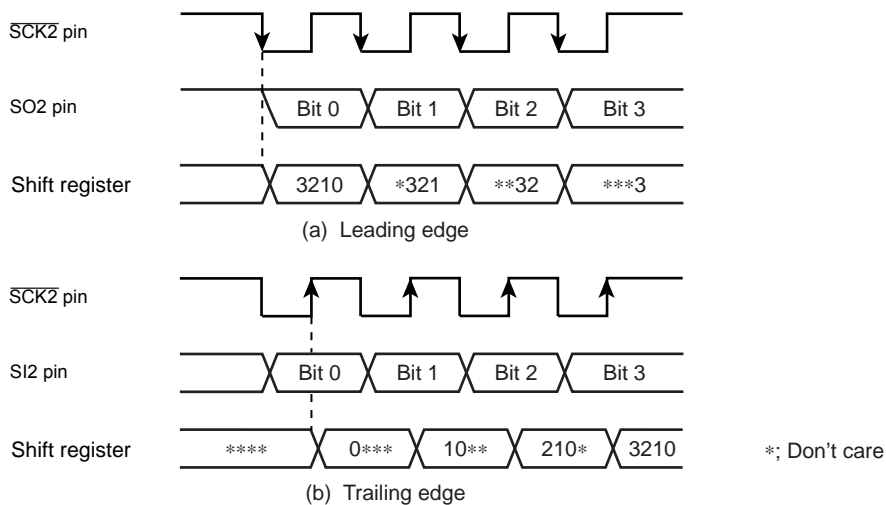


Figure 16-5 Shift edge

16.4 Number of bits to transfer

Either 4-bit or 8-bit serial transfer can be selected. When 4-bit serial transfer is selected, only the lower 4 bits of the transmit/receive data buffer register are used. The upper 4 bits are cleared to “0” when receiving. The data is transferred in sequence starting at the least significant bit (LSB).

16.5 Number of words to transfer

Up to 8 words consisting of 4 bits of data (4-bit serial transfer) or 8 bits (8-bit serial transfer) of data can be transferred continuously. The number of words to be transferred can be selected by $\text{SIO2CR2}<\text{BUF}>$.

An INTSIO2 interrupt is generated when the specified number of words has been transferred. If the number of words is to be changed during transfer, the serial interface must be stopped before making the change. The number of words can be changed during automatic-wait operation of an internal clock. In this case, the serial interface is not required to be stopped.

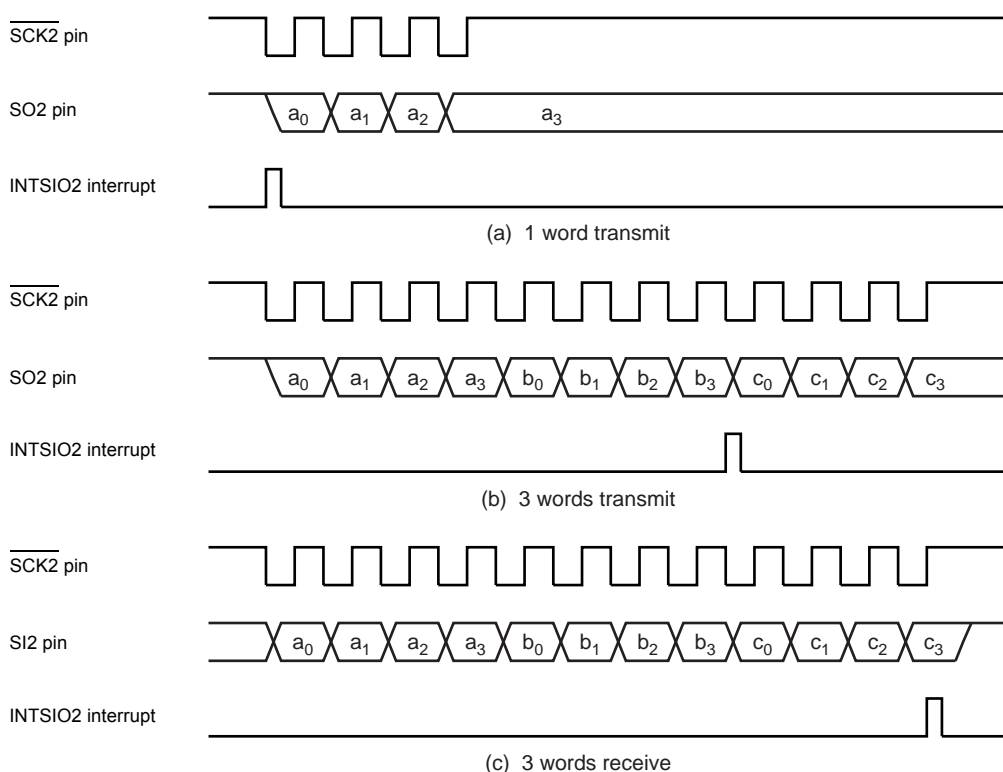


Figure 16-6 Number of words to transfer (Example: 1 word = 4bit)

16.6 Transfer Mode

SIO2CR1<SIOM> is used to select the transmit, receive, or transmit/receive mode.

16.6.1 4-bit and 8-bit transfer modes

In these modes, firstly set the SIO control register to the transmit mode, and then write first transmit data (number of transfer words to be transferred) to the data buffer registers (DBR).

After the data are written, the transmission is started by setting SIO2CR1<SIOS> to “1”. The data are then output sequentially to the SO pin in synchronous with the serial clock, starting with the least significant bit (LSB). As soon as the LSB has been output, the data are transferred from the data buffer register to the shift register. When the final data bit has been transferred and the data buffer register is empty, an INTSIO2 (Buffer empty) interrupt is generated to request the next transmitted data.

When the internal clock is used, the serial clock will stop and an automatic-wait will be initiated if the next transmitted data are not loaded to the data buffer register by the time the number of data words specified with the SIO2CR2<BUF> has been transmitted. Writing even one word of data cancels the automatic-wait; therefore, when transmitting two or more words, always write the next word before transmission of the previous word is completed.

Note: Automatic waits are also canceled by writing to a DBR not being used as a transmit data buffer register; therefore, during SIO do not use such DBR for other applications. For example, when 3 words are transmitted, do not use the DBR of the remained 5 words.

When an external clock is used, the data must be written to the data buffer register before shifting next data. Thus, the transfer speed is determined by the maximum delay time from the generation of the interrupt request to writing of the data to the data buffer register by the interrupt service program.

The transmission is ended by clearing SIO2CR1<SIOS> to “0” or setting SIO2CR1<SIOINH> to “1” in buffer empty interrupt service program.

SIO2CR1<SIOS> is cleared, the operation will end after all bits of words are transmitted.

That the transmission has ended can be determined from the status of SIO2SR<SIOF> because SIO2SR<SIOF> is cleared to “0” when a transfer is completed.

When SIO2CR1<SIOINH> is set, the transmission is immediately ended and SIO2SR<SIOF> is cleared to “0”.

When an external clock is used, it is also necessary to clear SIO2CR1<SIOS> to “0” before shifting the next data; If SIO2CR1<SIOS> is not cleared before shift out, dummy data will be transmitted and the operation will end.

If it is necessary to change the number of words, SIO2CR1<SIOS> should be cleared to “0”, then SIO2CR2<BUF> must be rewritten after confirming that SIO2SR<SIOF> has been cleared to “0”.

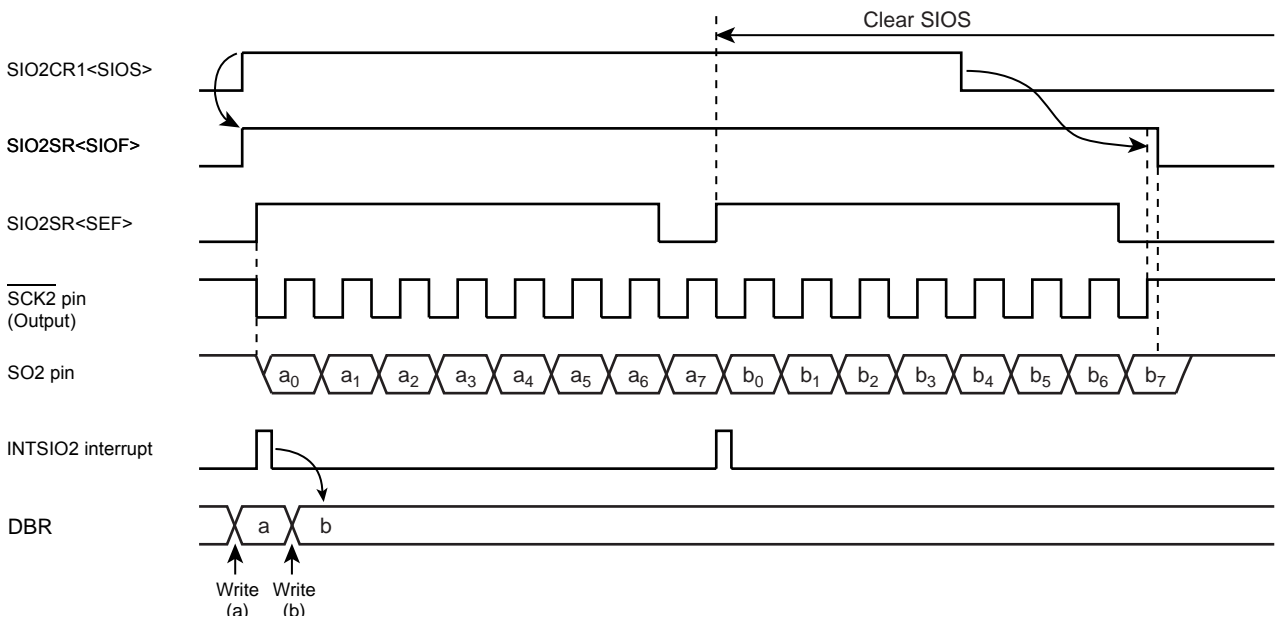


Figure 16-7 Transfer Mode (Example: 8bit, 1word transfer, Internal clock)

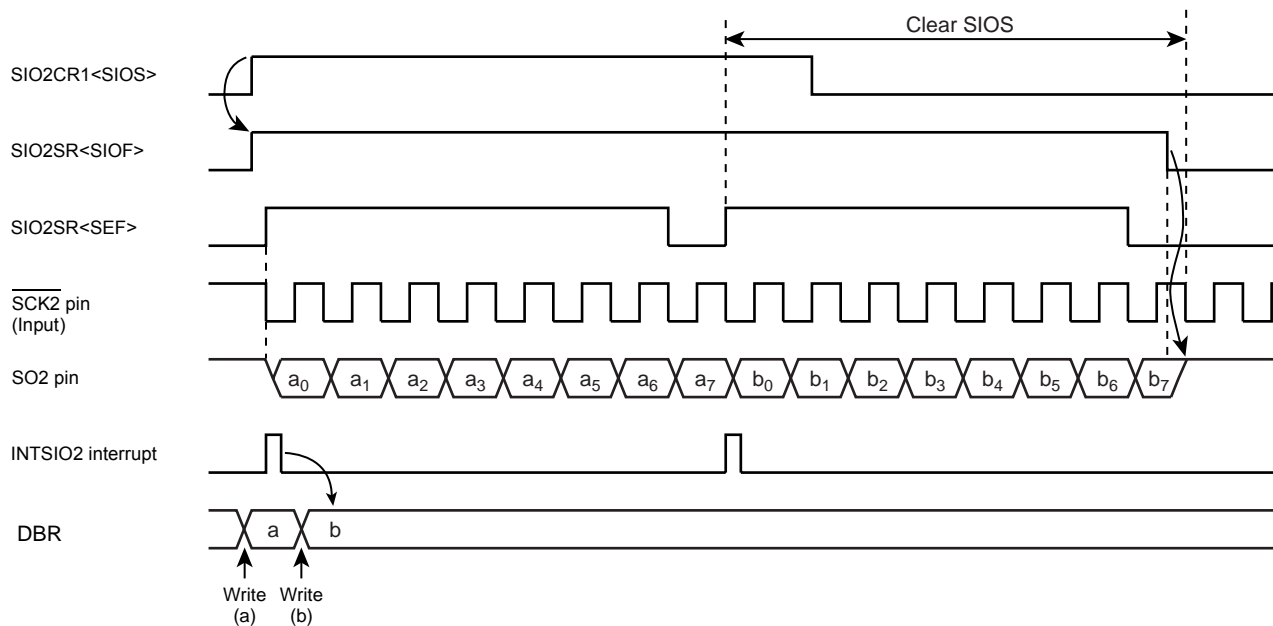


Figure 16-8 Transfer Mode (Example: 8bit, 1word transfer, External clock)

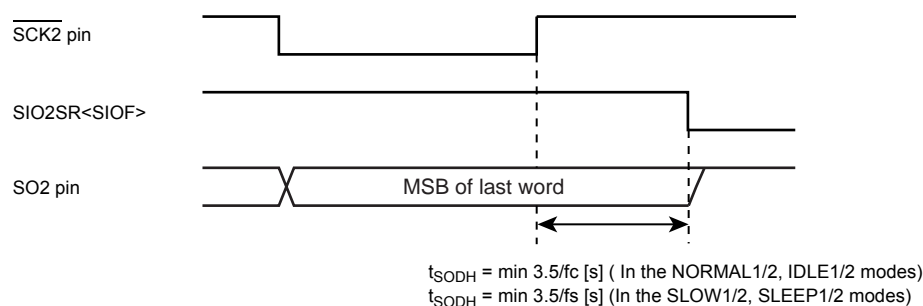


Figure 16-9 Transmitted Data Hold Time at End of Transfer

16.6.2 4-bit and 8-bit receive modes

After setting the control registers to the receive mode, set SIO2CR1<SIOS> to “1” to enable receiving. The data are then transferred to the shift register via the SI pin in synchronous with the serial clock. When one word of data has been received, it is transferred from the shift register to the data buffer register (DBR). When the number of words specified with the SIO2CR2<BUF> has been received, an INTSIO2 (Buffer full) interrupt is generated to request that these data be read out. The data are then read from the data buffer registers by the interrupt service program.

When the internal clock is used, and the previous data are not read from the data buffer register before the next data are received, the serial clock will stop and an automatic-wait will be initiated until the data are read. A wait will not be initiated if even one data word has been read.

Note: Waits are also canceled by reading a DBR not being used as a received data buffer register is read; therefore, during SIO2 do not use such DBR for other applications.

When an external clock is used, the shift operation is synchronized with the external clock; therefore, the previous data are read before the next data are transferred to the data buffer register. If the previous data have not been read, the next data will not be transferred to the data buffer register and the receiving of any more data will be canceled. When an external clock is used, the maximum transfer speed is determined by the delay between the time when the interrupt request is generated and when the data received have been read.

The receiving is ended by clearing SIO2CR1<SIOS> to “0” or setting SIO2CR1<SIOINH> to “1” in buffer full interrupt service program.

When SIO2CR1<SIOS> is cleared, the current data are transferred to the buffer. After SIO2CR1<SIOS> cleared, the receiving is ended at the time that the final bit of the data has been received. That the receiving has ended can be determined from the status of SIO2SR<SIOF>. SIO2SR<SIOF> is cleared to “0” when the receiving is ended. After confirmed the receiving termination, the final receiving data is read. When SIO2CR1<SIOINH> is set, the receiving is immediately ended and SIO2SR<SIOF> is cleared to “0”. (The received data is ignored, and it is not required to be read out.)

If it is necessary to change the number of words in external clock operation, SIO2CR1<SIOS> should be cleared to “0” then SIO2CR2<BUF> must be rewritten after confirming that SIO2SR<SIOF> has been cleared to “0”. If it is necessary to change the number of words in internal clock, during automatic-wait operation which occurs after completion of data receiving, SIO2CR2<BUF> must be rewritten before the received data is read out.

Note: The buffer contents are lost when the transfer mode is switched. If it should become necessary to switch the transfer mode, end receiving by clearing SIO2CR1<SIOS> to “0”, read the last data and then switch the transfer mode.

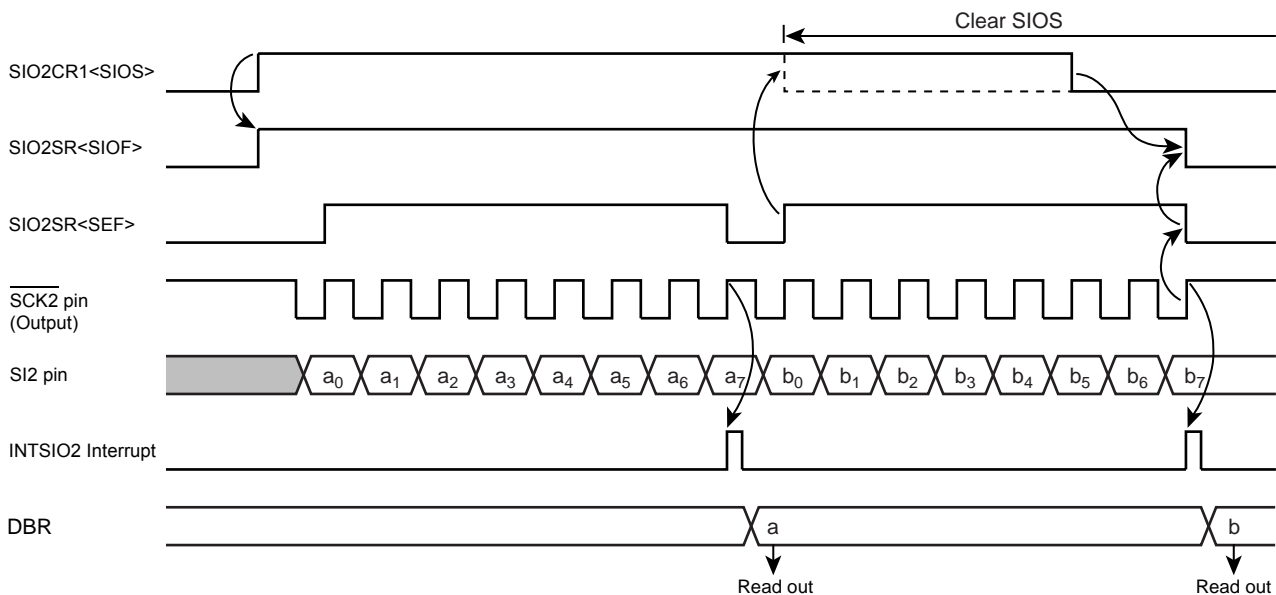


Figure 16-10 Receive Mode (Example: 8bit, 1word transfer, Internal clock)

16.6.3 8-bit transfer / receive mode

After setting the SIO control register to the 8-bit transmit/receive mode, write the data to be transmitted first to the data buffer registers (DBR). After that, enable the transmit/receive by setting SIO2CR1<SIOS> to “1”. When transmitting, the data are output from the SO2 pin at leading edges of the serial clock. When receiving, the data are input to the SI2 pin at the trailing edges of the serial clock. When the all receive is enabled, 8-bit data are transferred from the shift register to the data buffer register. An INTSIO2 interrupt is generated when the number of data words specified with the SIO2CR2<BUF> has been transferred. Usually, read the receive data from the buffer register in the interrupt service. The data buffer register is used for both transmitting and receiving; therefore, always write the data to be transmitted after reading the all received data.

When the internal clock is used, a wait is initiated until the received data are read and the next transfer data are written. A wait will not be initiated if even one transfer data word has been written.

When an external clock is used, the shift operation is synchronized with the external clock; therefore, it is necessary to read the received data and write the data to be transmitted next before starting the next shift operation. When an external clock is used, the transfer speed is determined by the maximum delay between generation of an interrupt request and the received data are read and the data to be transmitted next are written.

The transmit/receive operation is ended by clearing SIO2CR1<SIOS> to “0” or setting SIO2CR1<SIOINH> to “1” in INTSIO2 interrupt service program.

When SIO2CR1<SIOS> is cleared, the current data are transferred to the buffer. After SIO2CR1<SIOS> cleared, the transmitting/receiving is ended at the time that the final bit of the data has been transmitted.

That the transmitting/receiving has ended can be determined from the status of SIO2SR<SIOF>. SIO2SR<SIOF> is cleared to “0” when the transmitting/receiving is ended.

When SIO2CR1<SIOINH> is set, the transmit/receive operation is immediately ended and SIO2SR<SIOF> is cleared to “0”.

If it is necessary to change the number of words in external clock operation, SIO2CR1<SIOS> should be cleared to “0”, then SIO2CR2<BUF> must be rewritten after confirming that SIO2SR<SIOF> has been cleared to “0”.

If it is necessary to change the number of words in internal clock, during automatic-wait operation which occurs after completion of transmit/receive operation, SIO2CR2<BUF> must be rewritten before reading and writing of the receive/transmit data.

Note: The buffer contents are lost when the transfer mode is switched. If it should become necessary to switch the transfer mode, end receiving by clearing SIO2CR1<SIOS> to “0”, read the last data and then switch the transfer mode.

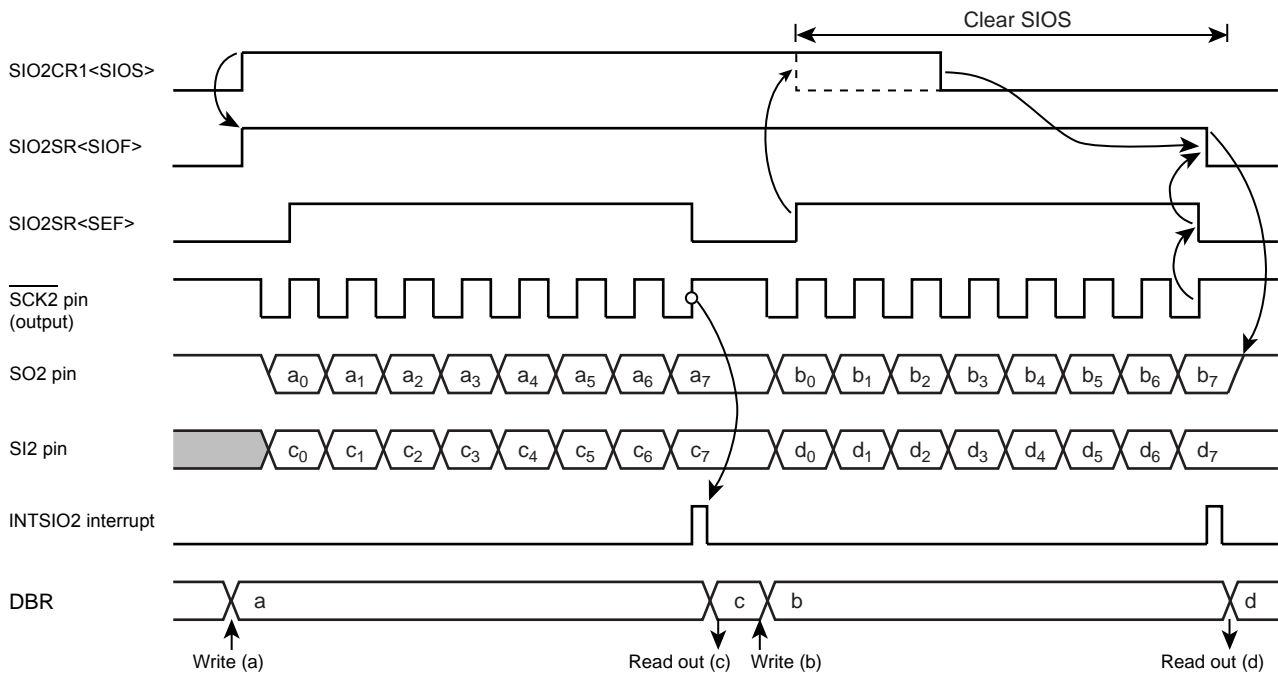


Figure 16-11 Transfer / Receive Mode (Example: 8bit, 1word transfer, Internal clock)

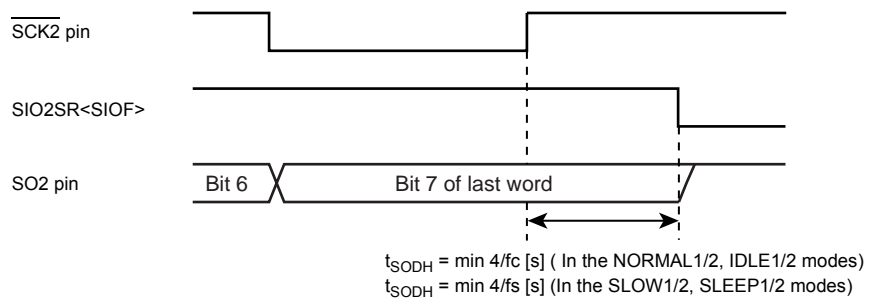


Figure 16-12 Transmitted Data Hold Time at End of Transfer / Receive

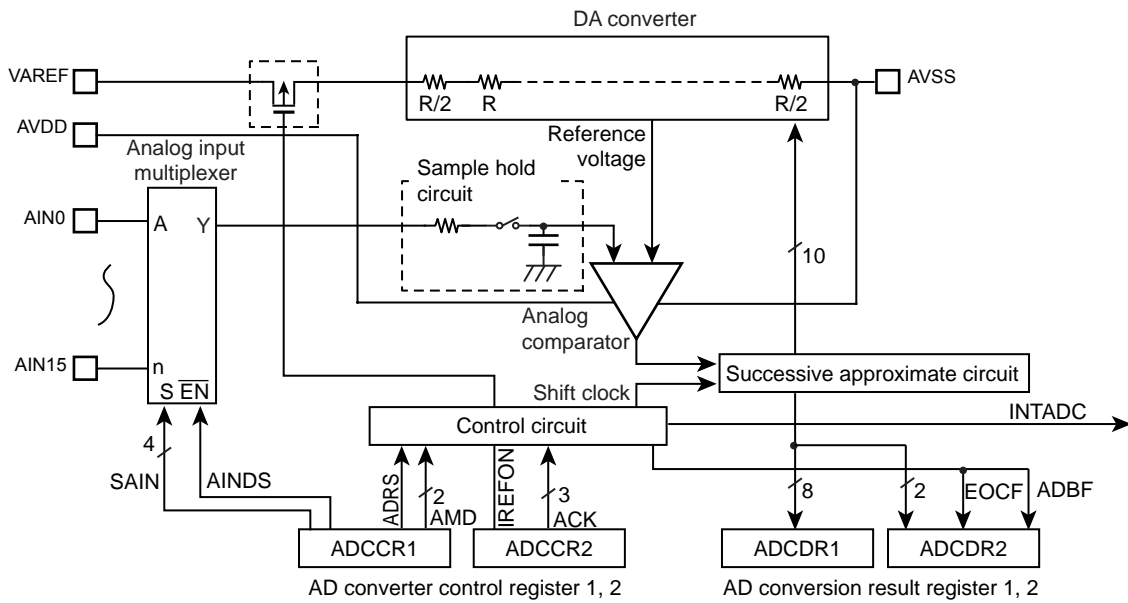
17. 10-bit AD Converter (ADC)

The TMP86FS64FG have a 10-bit successive approximation type AD converter.

17.1 Configuration

The circuit configuration of the 10-bit AD converter is shown in Figure 17-1.

It consists of control register ADCCR1 and ADCCR2, converted value register ADCDR1 and ADCDR2, a DA converter, a sample-hold circuit, a comparator, and a successive comparison circuit.



Note: Before using AD converter, set appropriate value to I/O port register combining a analog input port. For details, see the section on "I/O ports".

Figure 17-1 10-bit AD Converter

17.2 Register configuration

The AD converter consists of the following four registers:

1. AD converter control register 1 (ADCCR1)

This register selects the analog channels and operation mode (Software start or repeat) in which to perform AD conversion and controls the AD converter as it starts operating.

2. AD converter control register 2 (ADCCR2)

This register selects the AD conversion time and controls the connection of the DA converter (Ladder resistor network).

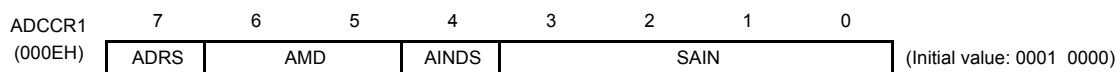
3. AD converted value register 1 (ADCDR1)

This register used to store the digital value after being converted by the AD converter.

4. AD converted value register 2 (ADCDR2)

This register monitors the operating status of the AD converter.

AD Converter Control Register 1



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

Note 1: Select analog input channel during AD converter stops (ADCDR2<ADBF> = "0").

Note 2: When the analog input channel is all use disabling, the ADCCR1<AINDS> should be set to "1".

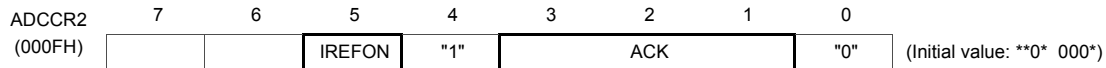
Note 3: During conversion, Do not perform port output instruction to maintain a precision for all of the pins because analog input port use as general input port. And for port near to analog input, Do not input intense signaling of change.

Note 4: The ADCCR1<ADRS> is automatically cleared to "0" after starting conversion.

Note 5: Do not set ADCCR1<ADRS> newly again during AD conversion. Before setting ADCCR1<ADRS> newly again, check ADCDR2<EOCF> to see that the conversion is completed or wait until the interrupt signal (INTADC) is generated (e.g., interrupt handling routine).

Note 6: After STOP or SLOW/SLEEP mode are started, AD converter control register1 (ADCCR1) is all initialized and no data can be written in this register. Therefore, to use AD converter again, set the ADCCR1 newly after returning to NORMAL1 or NORMAL2 mode.

AD Converter Control Register 2



IREFON	DA converter (Ladder resistor) connection control	0: Connected only during AD conversion 1: Always connected	
ACK	AD conversion time select (Refer to the following table about the conversion time)	000: 39/fc 001: Reserved 010: 78/fc 011: 156/fc 100: 312/fc 101: 624/fc 110: 1248/fc 111: Reserved	R/W

Note 1: Always set bit0 in ADCCR2 to "0" and set bit4 in ADCCR2 to "1".

Note 2: When a read instruction for ADCCR2, bit6 to 7 in ADCCR2 read in as undefined data.

Note 3: After STOP or SLOW/SLEEP mode are started, AD converter control register2 (ADCCR2) is all initialized and no data can be written in this register. Therefore, to use AD converter again, set the ADCCR2 newly after returning to NORMAL1 or NORMAL2 mode.

Table 17-1 ACK setting and Conversion time (at CGCR<DV1CK>="0")

Condition ACK	Conversion time	16 MHz	8 MHz	4 MHz	2 MHz	10 MHz	5 MHz	2.5 MHz
000	39/fc	-	-	-	19.5 μs	-	-	15.6 μs
001	Reserved							
010	78/fc	-	-	19.5 μs	39.0 μs	-	15.6 μs	31.2 μs
011	156/fc	-	19.5 μs	39.0 μs	78.0 μs	15.6 μs	31.2 μs	62.4 μs
100	312/fc	19.5 μs	39.0 μs	78.0 μs	156.0 μs	31.2 μs	62.4 μs	124.8 μs
101	624/fc	39.0 μs	78.0 μs	156.0 μs	-	62.4 μs	124.8 μs	-
110	1248/fc	78.0 μs	156.0 μs	-	-	124.8 μs	-	-
111	Reserved							

Table 17-2 ACK setting and Conversion time (at CGCR<DV1CK>="1")

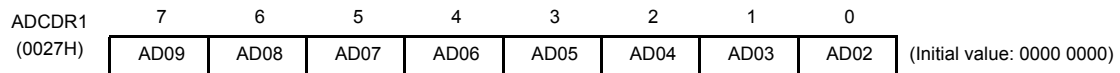
Condition ACK	Conversion time	16 MHz	8 MHz	4 MHz	2 MHz	10 MHz	5 MHz	2.5 MHz
000	39/fc	-	-	-	19.5 μs	-	-	15.6 μs
001	Reserved							
010	156/fc	-	19.5 μs	39.0 μs	78.0 μs	15.6 μs	31.2 μs	62.4 μs
011	312/fc	19.5 μs	39.0 μs	78.0 μs	156.0 μs	31.2 μs	62.4 μs	124.8 μs
100	624/fc	39.0 μs	78.0 μs	156.0 μs	-	62.4 μs	124.8 μs	-
101	1248/fc	78.0 μs	156.0 μs	-	-	124.8 μs	-	-
110	2096/fc	156.0 μs	-	-	-	-	-	-
111	Reserved							

Note 1: Setting for "-" in the above table are inhibited. fc: High Frequency oscillation clock [Hz]

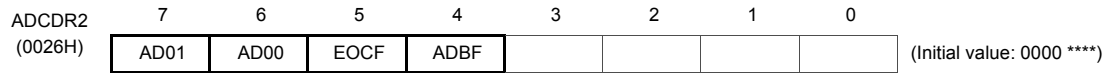
Note 2: Set conversion time setting should be kept more than the following time by Analog reference voltage (VAREF) .

- VAREF = 4.5 to 5.5 V 15.6 μs and more
- VAREF = 2.7 to 5.5 V 31.2 μs and more

AD Converted value Register 1



AD Converted value Register 2



EOCF	AD conversion end flag	0: Before or during conversion 1: Conversion completed	Read only
ADBF	AD conversion BUSY flag	0: During stop of AD conversion 1: During AD conversion	

Note 1: The ADCDR2<EOCF> is cleared to "0" when reading the ADCDR1. Therefore, the AD conversion result should be read to ADCDR2 more first than ADCDR1.

Note 2: The ADCDR2<ADBF> is set to "1" when AD conversion starts, and cleared to "0" when AD conversion finished. It also is cleared upon entering STOP mode or SLOW mode .

Note 3: If a read instruction is executed for ADCDR2, read data of bit3 to bit0 are unstable.

17.3 Function

17.3.1 Software Start Mode

After setting ADCCR1<AMD> to “01” (software start mode), set ADCCR1<ADRS> to “1”. AD conversion of the voltage at the analog input pin specified by ADCCR1<SAIN> is thereby started.

After completion of the AD conversion, the conversion result is stored in AD converted value registers (ADCDR1, ADCDR2) and at the same time ADCDR2<EOCF> is set to 1, the AD conversion finished interrupt (INTADC) is generated.

ADRS is automatically cleared after AD conversion has started. Do not set ADCCR1<ADRS> newly again (Restart) during AD conversion. Before setting ADRS newly again, check ADCDR2<EOCF> to see that the conversion is completed or wait until the interrupt signal (INTADC) is generated (e.g., interrupt handling routine).

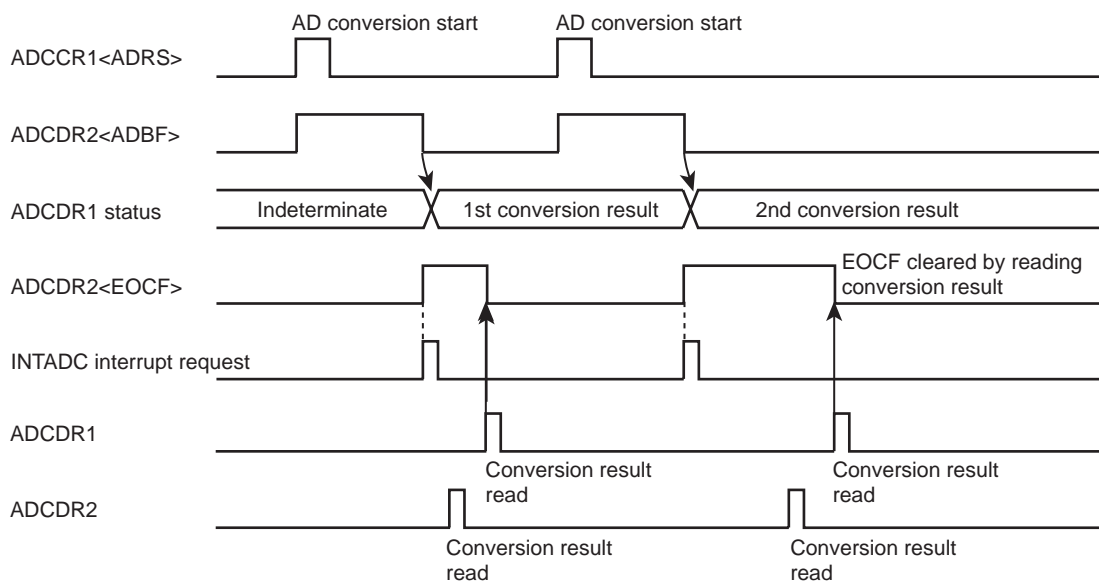


Figure 17-2 Software Start Mode

17.3.2 Repeat Mode

AD conversion of the voltage at the analog input pin specified by ADCCR1<SAIN> is performed repeatedly. In this mode, AD conversion is started by setting ADCCR1<ADRS> to “1” after setting ADCCR1<AMD> to “11” (Repeat mode).

After completion of the AD conversion, the conversion result is stored in AD converted value registers (ADCDR1, ADCDR2) and at the same time ADCDR2<EOCF> is set to 1, the AD conversion finished interrupt (INTADC) is generated.

In repeat mode, each time one AD conversion is completed, the next AD conversion is started. To stop AD conversion, set ADCCR1<AMD> to “00” (Disable mode) by writing 0s. The AD convert operation is stopped immediately. The converted value at this time is not stored in the AD converted value register.

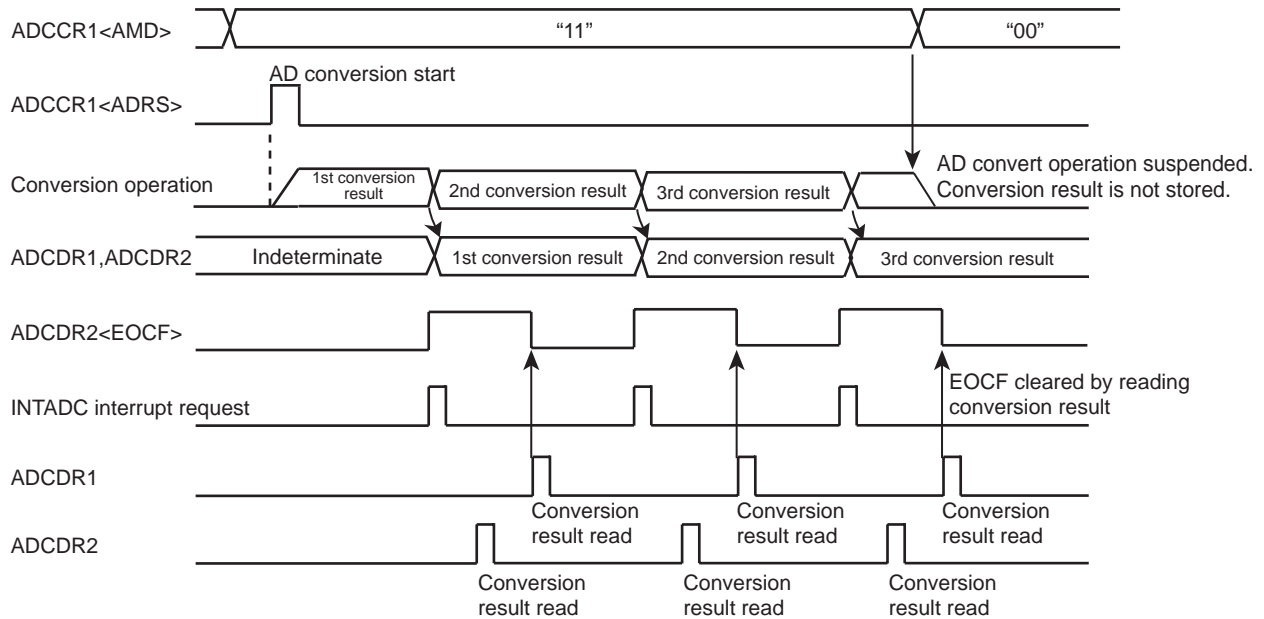


Figure 17-3 Repeat Mode

17.3.3 Register Setting

1. Set up the AD converter control register 1 (ADCCR1) as follows:
 - Choose the channel to AD convert using AD input channel select (SAIN).
 - Specify analog input enable for analog input control (AINDS).
 - Specify AMD for the AD converter control operation mode (software or repeat mode).
2. Set up the AD converter control register 2 (ADCCR2) as follows:
 - Set the AD conversion time using AD conversion time (ACK). For details on how to set the conversion time, refer to Figure 17-1, Figure 17-2 and AD converter control register 2.
 - Choose IREFON for DA converter control.
3. After setting up (1) and (2) above, set AD conversion start (ADRS) of AD converter control register 1 (ADCCR1) to "1". If software start mode has been selected, AD conversion starts immediately.
4. After an elapse of the specified AD conversion time, the AD converted value is stored in AD converted value register 1 (ADCDR1) and the AD conversion finished flag (EOCF) of AD converted value register 2 (ADCDR2) is set to "1", upon which time AD conversion interrupt INTADC is generated.
5. EOCF is cleared to "0" by a read of the conversion result. However, if reconverted before a register read, although EOCF is cleared the previous conversion result is retained until the next conversion is completed.

Example :After selecting the conversion time 19.5 μ s at 16 MHz and the analog input channel AIN3 pin, perform AD conversion once. After checking EOCF, read the converted value, store the lower 2 bits in address 0009EH and store the upper 8 bits in address 0009FH in RAM. The operation mode is software start mode.

```

: (port setting)      :                               ;Set port register appropriately before setting AD
:                               ; converter registers.
:                               ;
:                               ; (Refer to section I/O port in details)
LD      (ADCCR1) , 00100011B    ; Select AIN3
LD      (ADCCR2) , 11011000B    ;Select conversion time(312/fc) and operation
                                ; mode
SLOOP : SET      (ADCCR1) . 7      ; ADRS = 1(AD conversion start)
        TEST     (ADCCR2) . 5      ; EOCF= 1 ?
        JRS      T, SLOOP
        LD      A , (ADCDR2)       ; Read result data
        LD      (9EH) , A
        LD      A , (ADCDR1)       ; Read result data
        LD      (9FH), A
    
```

17.4 STOP/SLOW Modes during AD Conversion

When standby mode (STOP or SLOW mode) is entered forcibly during AD conversion, the AD convert operation is suspended and the AD converter is initialized (ADCCR1 and ADCCR2 are initialized to initial value). Also, the conversion result is indeterminate. (Conversion results up to the previous operation are cleared, so be sure to read the conversion results before entering standby mode (STOP or SLOW mode).) When restored from standby mode (STOP or SLOW mode), AD conversion is not automatically restarted, so it is necessary to restart AD conversion. Note that since the analog reference voltage is automatically disconnected, there is no possibility of current flowing into the analog reference voltage.

17.5 Analog Input Voltage and AD Conversion Result

The analog input voltage is corresponded to the 10-bit digital value converted by the AD as shown in Figure 17-4.

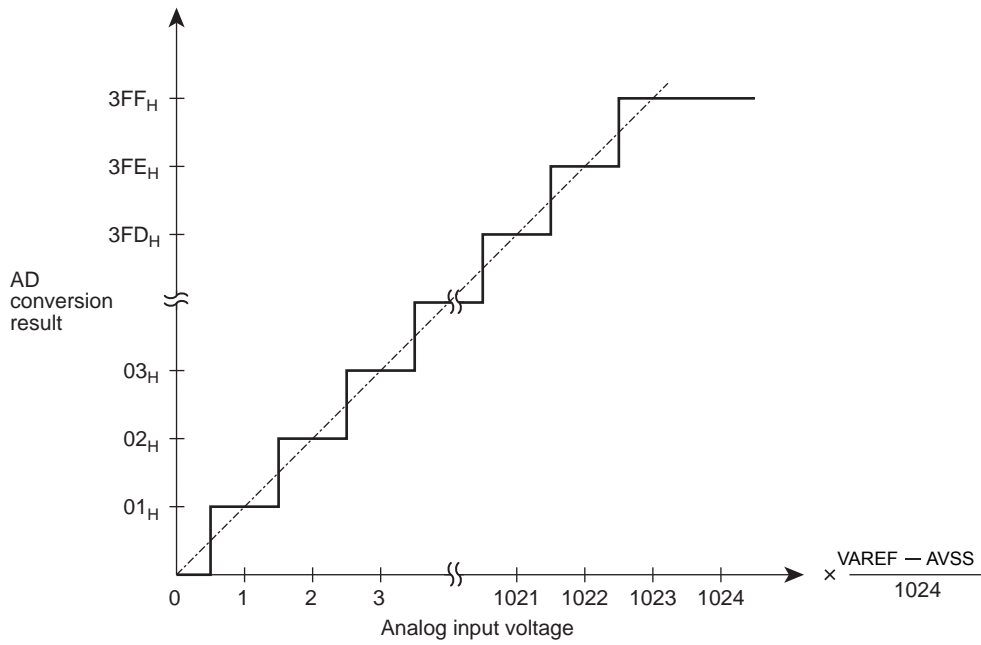


Figure 17-4 Analog Input Voltage and AD Conversion Result (Typ.)

17.6 Precautions about AD Converter

17.6.1 Analog input pin voltage range

Make sure the analog input pins (AIN0 to AIN15) are used at voltages within VAREF to AVSS. If any voltage outside this range is applied to one of the analog input pins, the converted value on that pin becomes uncertain. The other analog input pins also are affected by that.

17.6.2 Analog input shared pins

The analog input pins (AIN0 to AIN15) are shared with input/output ports. When using any of the analog inputs to execute AD conversion, do not execute input/output instructions for all other ports. This is necessary to prevent the accuracy of AD conversion from degrading. Not only these analog input shared pins, some other pins may also be affected by noise arising from input/output to and from adjacent pins.

17.6.3 Noise Countermeasure

The internal equivalent circuit of the analog input pins is shown in Figure 17-5. The higher the output impedance of the analog input source, more easily they are susceptible to noise. Therefore, make sure the output impedance of the signal source in your design is 5 kΩ or less. Toshiba also recommends attaching a capacitor external to the chip.

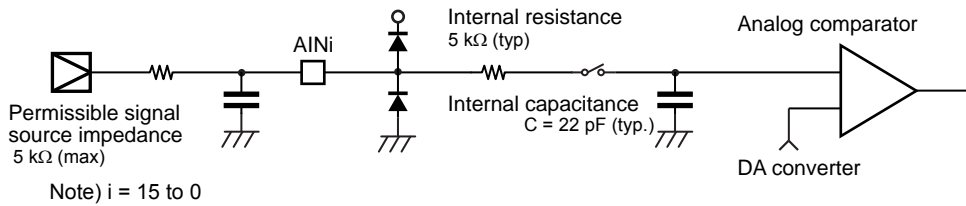


Figure 17-5 Analog Input Equivalent Circuit and Example of Input Pin Processing

18. Key-on Wakeup (KWU)

In the TMP86FS64FG, the STOP mode is released by not only P20($\overline{\text{INT5}}/\overline{\text{STOP}}$) pin but also four (STOP2 to STOP5) pins.

When the STOP mode is released by STOP2 to STOP5 pins, the $\overline{\text{STOP}}$ pin needs to be used. In details, refer to the following section " 18.2 Control ".

18.1 Configuration

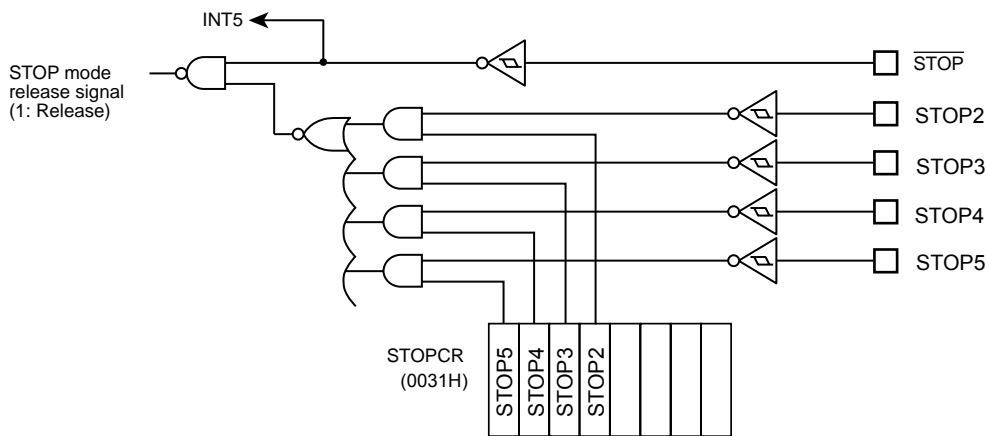


Figure 18-1 Key-on Wakeup Circuit

18.2 Control

STOP2 to STOP5 pins can be controlled by Key-on Wakeup Control Register (STOPCR). It can be configured as enable/disable in 1-bit unit. When those pins are used for STOP mode release, configure corresponding I/O pins to input mode by I/O port register beforehand.

Key-on Wakeup Control Register

STOPCR	7	6	5	4	3	2	1	0	
(0031H)	STOP5	STOP4	STOP3	STOP2					(Initial value: 0000 ****)

STOP5	STOP mode released by STOP5	0:Disable 1:Enable	Write only
STOP4	STOP mode released by STOP4	0:Disable 1:Enable	Write only
STOP3	STOP mode released by STOP3	0:Disable 1:Enable	Write only
STOP2	STOP mode released by STOP2	0:Disable 1:Enable	Write only

18.3 Function

Stop mode can be entered by setting up the System Control Register (SYSCR1), and can be exited by detecting the "L" level on STOP2 to STOP5 pins, which are enabled by STOPCR, for releasing STOP mode (Note1).

Also, each level of the STOP2 to STOP5 pins can be confirmed by reading corresponding I/O port data register, check all STOP2 to STOP5 pins "H" that is enabled by STOPPCR before the STOP mode is started (Note2,3).

Note 1: When the STOP mode is released by the edge release mode (SYSCR1<RELM> = "0"), inhibit input from STOP2 to STOP5 pins by Key-on Wakeup Control Register (STOPPCR) or must be set "H" level into STOP2 to STOP5 pins that are available input during STOP mode.

Note 2: When the $\overline{\text{STOP}}$ pin input is high or STOP2 to STOP5 pins input which is enabled by STOPPCR is low, executing an instruction which starts STOP mode will not place in STOP mode but instead will immediately start the release sequence (Warm up).

Note 3: The input circuit of Key-on Wakeup input and Port input is separated. Also each input voltage threshold value is different. Therefore, a value comes from port input before STOP mode starts may be different from a value which is detected by Key-on Wakeup input (Figure 18-2).

Note 4: $\overline{\text{STOP}}$ pin doesn't have the control register such as STOPPCR, so when STOP mode is released by STOP2 to STOP5 pins, $\overline{\text{STOP}}$ pin also should be used as STOP mode release function.

Note 5: In STOP mode, Key-on Wakeup pin which is enabled as input mode (for releasing STOP mode) by Key-on Wakeup Control Register (STOPPCR) may generate the penetration current, so the said pin must be disabled as AD conversion input (analog voltage input).

Note 6: When the STOP mode is released by STOP2 to STOP5 pins, the level of $\overline{\text{STOP}}$ pin should hold "L" level (Figure 18-3).

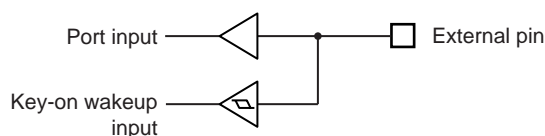


Figure 18-2 Key-on Wakeup Input and Port Input

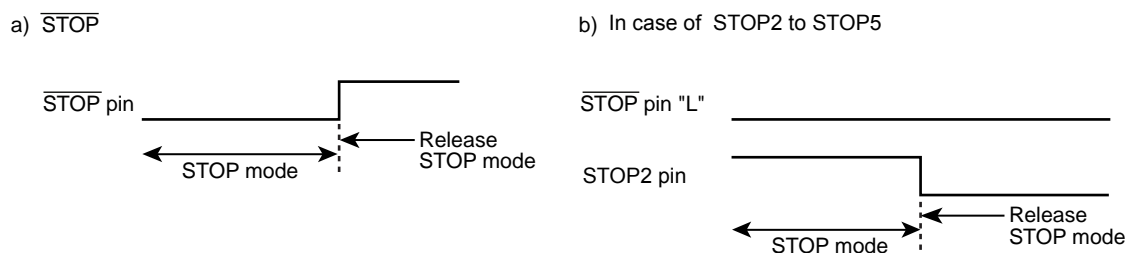


Figure 18-3 Priority of STOP pin and STOP2 to STOP5 pins

Table 18-1 Release level (edge) of STOP mode

Pin name	Release level (edge)	
	SYSCR1<RELM>="1" (Note2)	SYSCR1<RELM>="0"
$\overline{\text{STOP}}$	"H" level	Rising edge
STOP2	"L" level	Don't use (Note1)
STOP3	"L" level	Don't use (Note1)
STOP4	"L" level	Don't use (Note1)
STOP5	"L" level	Don't use (Note1)

19. Flash Memory

TMP86FS64FG has 61440byte flash memory (address: 1000H to FFFFH). The write and erase operations to the flash memory are controlled in the following three types of mode.

- MCU mode

The flash memory is accessed by the CPU control in the MCU mode. This mode is used for software bug correction and firmware change after shipment of the device since the write operation to the flash memory is available by retaining the application behavior.

- Serial PROM mode

The flash memory is accessed by the CPU control in the serial PROM mode. Use of the serial interface (UART) enables the flash memory to be controlled by the small number of pins. TMP86FS64FG 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

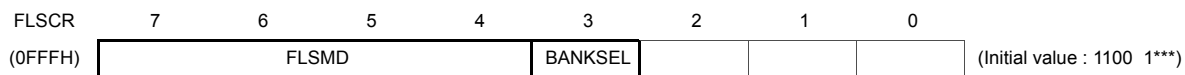
The parallel PROM mode allows the flash memory to be accessed as a stand-alone flash memory by the program writer provided by the third party. High-speed access to the flash memory is available by controlling address and data signals directly. For the support of the program writer, please ask Toshiba sales representative.

In the MCU and serial PROM modes, the flash memory control register (FLSCR) is used for flash memory control. This chapter describes how to access the flash memory using the flash memory control register (FLSCR) in the MCU and serial PROM modes.

19.1 Flash Memory Control

The flash memory is controlled via the flash memory control register (FLSCR).

Flash Memory Control Register



FLSMD	Flash memory command sequence execution control	1100: Disable command sequence execution 0011: Enable command sequence execution Others: Reserved	R/W
BANKSEL	Flash memory bank select control (Serial PROM mode only)	0: Select BANK0 1: Select BANK1	R/W

- Note 1: The command sequence of the flash memory can be executed only when FLSMD="0011B". In other cases, any attempts to execute the command sequence are ineffective.
- Note 2: FLSMD must be set to either "1100B" or "0011B".
- Note 3: BANKSEL is effective only in the serial PROM mode. In the MCU mode, the flash memory is always accessed with actual addresses (1000-FFFFH) regardless of BANKSEL.
- Note 4: Bits 2 through 0 in FLSCR are always read as don't care.

19.1.1 Flash Memory Command Sequence Execution Control (FLSCR<FLSMD>)

The flash memory can be protected from inadvertent write due to program error or microcontroller misoperation. This write protection feature is realized by disabling flash memory command sequence execution via the flash memory control register (write protect). To enable command sequence execution, set FLSCR<FLSMD> to "0011B". To disable command sequence execution, set FLSCR<FLSMD> to "1100B". After reset, FLSCR<FLSMD> is initialized to "1100B" to disable command sequence execution. Normally, FLSCR<FLSMD> should be set to "1100B" except when the flash memory needs to be written or erased.

19.1.2 Flash Memory Bank Select Control (FLSCR<BANKSEL>)

In the serial PROM mode, a 2-kbyte BOOTROM is mapped to addresses 7800H-7FFFH and the flash memory is mapped to 2 banks at 8000H-FFFFH. Flash memory addresses 1000H-7FFFH are mapped to 9000H-FFFFH as BANK0, and flash memory addresses 8000H-FFFFH are mapped to 8000H-FFFFH as BANK1. FLSCR<BANKSEL> is used to switch between these banks. For example, to access the flash memory address 7000H, set FLSCR<BANKSEL> to "0" and then access F000H. To access the flash memory address 9000H, set FLSCR<BANKSEL> to "1" and then access 9000H.

In the MCU mode, the flash memory is accessed with actual addresses at 1000H-FFFFH. In this case, FLSCR<BANKSEL> is ineffective (i.e., its value has no effect on other operations).

Table 19-1 Flash Memory Access

Operating Mode	FLSCR <BANKSEL>	Access Area	Specified Address
MCU mode	Don't care	1000H-FFFFH	
Serial PROM mode	0 (BANK0)	1000H-7FFFH	9000H-FFFFH
	1 (BANK1)	8000H-FFFFH	

19.2 Command Sequence

The command sequence in the MCU and the serial PROM modes consists of six commands (JEDEC compatible), as shown in Table 19-2. Addresses specified in the command sequence are recognized with the lower 12 bits (excluding BA, SA, and FF7FH used for read protection). The upper 4 bits are used to specify the flash memory area, as shown in Table 19-3.

Table 19-2 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	555H	AAH	AAAH	55H	555H	A0H	BA (Note 1)	Data (Note 1)	-	-	-	-
2	Sector Erase (4-kbyte Erase)	555H	AAH	AAAH	55H	555H	80H	555H	AAH	AAAH	55H	SA (Note 2)	30H
3	Chip Erase (All Erase)	555H	AAH	AAAH	55H	555H	80H	555H	AAH	AAAH	55H	555H	10H
4	Product ID Entry	555H	AAH	AAAH	55H	555H	90H	-	-	-	-	-	-
5	Product ID Exit	XXH	F0H	-	-	-	-	-	-	-	-	-	-
	Product ID Exit	555H	AAH	AAAH	55H	555H	F0H	-	-	-	-	-	-
6	Read Protect	555H	AAH	AAAH	55H	555H	A5H	FF7FH	00H	-	-	-	-

Note 1: Set the address and data to be written.

Note 2: The area to be erased is specified with the upper 4 bits of the address.

Table 19-3 Address Specification in the Command Sequence

Operating Mode	FLSCR <BANKSEL>	Specified Address
MCU mode	Don't care	1***H-F***H
Serial PROM mode	0 (BANK0)	9***H-F***H
	1 (BANK1)	8***H-F***H

19.2.1 Byte Program

This command writes the flash memory for each byte unit. The addresses and data to be written are specified in the 4th bus write cycle. Each byte can be programmed in a maximum of 40 μs. The next command sequence cannot be executed until the write operation is completed. To check the completion of the write operation, perform read operations repeatedly until the same data is read twice from the same address in the flash memory. During the write operation, any consecutive attempts to read from the same address is reversed bit 6 of the data (toggling between 0 and 1).

Note: To rewrite data to Flash memory addresses at which data (including FFH) is already written, make sure to erase the existing data by "sector erase" or "chip erase" before rewriting data.

19.2.2 Sector Erase (4-kbyte 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. For example, in the MCU mode, to erase 4 kbytes from 7000H to 7FFFH, specify one of the addresses in 7000H-7FFFH as the 6th bus write cycle. In the serial PROM mode, to erase 4 kbytes from 7000H to 7FFFH, set FLSCR<BANKSEL> to "0" and then specify one of the addresses in F000H-FFFFH as the 6th bus write cycle. The sector erase command is effective only in the MCU and serial PROM modes, and it cannot be used in the parallel PROM mode.

A maximum of 30 ms is required to erase 4 kbytes. The next command sequence cannot be executed until the erase operation is completed. To check the completion of the erase operation, perform read operations repeatedly for data polling until the same data is read twice from the same address in the flash memory. During the erase operation, any consecutive attempts to read from the same address is reversed bit 6 of the data (toggling between 0 and 1).

19.2.3 Chip Erase (All Erase)

This command erases the entire flash memory in approximately 30 ms. The next command sequence cannot be executed until the erase operation is completed. To check the completion of the erase operation, perform read operations repeatedly for data polling until the same data is read twice from the same address in the flash memory. During the erase operation, any consecutive attempts to read from the same address is reversed bit 6 of the data (toggling between 0 and 1). After the chip is erased, all bytes contain FFH.

19.2.4 Product ID Entry

This command activates the Product ID mode. In the Product ID mode, the vendor ID, the flash ID, and the read protection status can be read from the flash memory.

Table 19-4 Values To Be Read in the Product ID Mode

Address	Meaning	Read Value
F000H	Vendor ID	98H
F001H	Flash macro ID	41H
F002H	Flash size	0EH: 60 kbytes 0BH: 48 kbytes 07H: 32 kbytes 05H: 24 kbytes 03H: 16 kbytes 01H: 8 kbytes 00H: 4 kbytes
FF7FH	Read protection status	FFH: Read protection disabled Other than FFH: Read protection enabled

Note: The value at address F002H (flash size) depends on the size of flash memory incorporated in each product. For example, if the product has 60-kbyte flash memory, "0EH" is read from address F002H.

19.2.5 Product ID Exit

This command is used to exit the Product ID mode.

19.2.6 Read Protect

This command enables the read protection setting in the flash memory. When the read protection is enabled, the flash memory cannot be read in the parallel PROM mode. In the serial PROM mode, the flash write and RAM loader commands cannot be executed.

To enable the read protection setting in the serial PROM mode, set FLSCR<BANKSEL> to "1" before executing the read protect command sequence. To disable the read protection setting, it is necessary to execute the chip erase command sequence. Whether or not the read protection is enabled can be checked by reading FF7FH in the Product ID mode. For details, see Table 19-4.

It takes a maximum of 40 μ s to set read protection in the flash memory. The next command sequence cannot be executed until this operation is completed. To check the completion of the read protect operation, perform read operations repeatedly for data polling until the same data is read twice from the same address in the flash memory. During the read protect operation, any attempts to read from the same address is reversed bit 6 of the data (toggling between 0 and 1).

19.3 Toggle Bit (D6)

After the byte program, chip erase, and read protect command sequence is executed, any consecutive attempts to read from the same address is reversed bit 6 (D6) of the data (toggling between 0 and 1) until the operation is completed. Therefore, this toggle bit provides a software mechanism to check the completion of each operation. Usually perform read operations repeatedly for data polling until the same data is read twice from the same address in the flash memory. After the byte program, chip erase, or read protect command sequence is executed, the initial read of the toggle bit always produces a "1".

19.4 Access to the Flash Memory Area

When the write, erase and read protections are set in the flash memory, read and fetch operations cannot be performed in the entire flash memory area. Therefore, to perform these operations in the entire flash memory area, access to the flash memory area by the control program in the BOOTROM or RAM area. (The flash memory program cannot write to the flash memory.) The serial PROM or MCU mode is used to run the control program in the BOOTROM or RAM area.

Note 1: The flash memory can be written or read for each byte unit. Erase operations can be performed either in the entire area or in units of 4 kbytes, whereas read operations can be performed by an one transfer instruction. However, the command sequence method is adopted for write and erase operations, requiring several-byte transfer instructions for each operation.

Note 2: To rewrite data to Flash memory addresses at which data (including FFH) is already written, make sure to erase the existing data by "sector erase" or "chip erase" before rewriting data.

19.4.1 Flash Memory Control in the Serial PROM Mode

The serial PROM mode is used to access to the flash memory by the control program provided in the BOOTROM area. Since almost of all operations relating to access to the flash memory can be controlled simply by the communication data of the serial interface (UART), these functions are transparent to the user. For the details of the serial PROM mode, see "Serial PROM Mode."

To access to the flash memory by using peripheral functions in the serial PROM mode, run the RAM loader command to execute the control program in the RAM area. The procedures to execute the control program in the RAM area is shown in "19.4.1.1 How to write to the flash memory by executing the control program in the RAM area (in the RAM loader mode within the serial PROM mode)".

19.4.1.1 How to write to the flash memory by executing the control program in the RAM area (in the RAM loader mode within the serial PROM mode)

(Steps 1 and 2 are controlled by the BOOTROM, and steps 3 through 10 are controlled by the control program executed in the RAM area.)

1. Transfer the write control program to the RAM area in the RAM loader mode.
2. Jump to the RAM area.
3. Disable (DI) the interrupt master enable flag (IMF←"0").
4. Set FLSCR<FLSMD> to "0011B" (to enable command sequence execution).
5. Execute the erase command sequence.
6. Read the same flash memory address twice.
(Repeat step 6 until the same data is read by two consecutive reads operations.)
7. Specify the bank to be written in FLSCR<BANKSEL>.
8. Execute the write command sequence.
9. Read the same flash memory address twice.
(Repeat step 9 until the same data is read by two consecutive reads operations.)
10. Set FLSCR<FLSMD> to "1100B" (to disable command sequence execution).

Note 1: Before writing to the flash memory in the RAM area, disable interrupts by setting the interrupt master enable flag (IMF) to "0". Usually disable interrupts by executing the DI instruction at the head of the write control program in the RAM area.

Note 2: Since the watchdog timer is disabled by the BOOTROM in the RAM loader mode, it is not required to disable the watchdog timer by the RAM loader program.

Example :After chip erasure, the program in the RAM area writes data 3FH to address F000H.

```

DI                                     : Disable interrupts (IMF←"0")
LD      (FLSCR),0011_1000B           : Enable command sequence execution.
LD      IX,0F555H
LD      IY,0FAAAH
LD      HL,0F000H

; #### Flash Memory Chip erase Process ####
LD      (IX),0AAH                     : 1st bus write cycle
LD      (IY),55H                      : 2nd bus write cycle
LD      (IX),80H                      : 3rd bus write cycle
LD      (IX),0AAH                     : 4th bus write cycle
LD      (IY),55H                      : 5th bus write cycle
LD      (IX),10H                      : 6th bus write cycle
sLOOP1: LD      W,(IX)
      CMP     W,(IX)
      JR      NZ,sLOOP1                : Loop until the same value is read.
      SET    (FLSCR),3                : Set BANK1.

; #### Flash Memory Write Process ####
LD      (IX),0AAH                     : 1st bus write cycle
LD      (IY),55H                      : 2nd bus write cycle
LD      (IX),0A0H                     : 3rd bus write cycle
LD      (HL),3FH                      : 4th bus write cycle, (F000H)=3FH
sLOOP2: LD      W,(HL)
      CMP     W,(HL)
      JR      NZ,sLOOP2                : Loop until the same value is read.
      LD      (FLSCR),1100_1000B      : Disable command sequence execution.
sLOOP3: JP      sLOOP3

```

19.4.2 Flash Memory Control in the MCU mode

In the MCU mode, write operations are performed by executing the control program in the RAM area. Before execution of the control program, copy the control program into the RAM area or obtain it from the external using the communication pin. The procedures to execute the control program in the RAM area in the MCU mode are described below.

19.4.2.1 How to write to the flash memory by executing a user write control program in the RAM area (in the MCU mode)

(Steps 1 and 2 are controlled by the program in the flash memory, and steps 3 through 11 are controlled by the control program in the RAM area.)

1. Transfer the write control program to the RAM area.
2. Jump to the RAM area.
3. Disable (DI) the interrupt master enable flag (IMF←"0").
4. Disable the watchdog timer, if it is used.
5. Set FLSCR<FLSMD> to "0011B" (to enable command sequence execution).
6. Execute the erase command sequence.
7. Read the same flash memory address twice.
(Repeat step 7 until the same data is read by two consecutive read operations.)
8. Execute the write command sequence. (It is not required to specify the bank to be written.)
9. Read the same flash memory address twice.
(Repeat step 9 until the same data is read by two consecutive read operations.)
10. Set FLSCR<FLSMD> to "1100B" (to disable command sequence execution).
11. Jump to the flash memory area.

Note 1: Before writing to the flash memory in the RAM area, disable interrupts by setting the interrupt master enable flag (IMF) to "0". Usually disable interrupts by executing the DI instruction at the head of the write control program in the RAM area.

Note 2: When writing to the flash memory, do not intentionally use non-maskable interrupts (the watchdog timer must be disabled if it is used). If a non-maskable interrupt occurs while the flash memory is being written, unexpected data is read from the flash memory (interrupt vector), resulting in malfunction of the microcontroller.

Example :After sector erasure (E000H-EFFFH), the program in the RAM area writes data 3FH to address E000H.

```

DI          : Disable interrupts (IMF←"0")

LD          (WDTCR2),4EH      : Clear the WDT binary counter.

LDW        (WDTCR1),0B101H   : Disable the WDT.

LD          (FLSCR),0011_1000B : Enable command sequence execution.

LD          IX,0F555H

LD          IY,0FAAAH

LD          HL,0E000H

; ##### Flash Memory Sector Erase Process #####

LD          (IX),0AAH        : 1st bus write cycle

LD          (IY),55H         : 2nd bus write cycle

LD          (IX),80H         : 3rd bus write cycle

LD          (IX),0AAH        : 4th bus write cycle

LD          (IY),55H         : 5th bus write cycle

LD          (HL),30H         : 6th bus write cycle

sLOOP1:    LD          W,(IX)

CMP        W,(IX)

JR         NZ,sLOOP1        : Loop until the same value is read.

; ##### Flash Memory Write Process #####

LD          (IX),0AAH        : 1st bus write cycle

LD          (IY),55H         : 2nd bus write cycle

LD          (IX),0A0H        : 3rd bus write cycle

LD          (HL),3FH         : 4th bus write cycle, (1000H)=3FH

sLOOP2:    LD          W,(HL)

CMP        W,(HL)

JR         NZ,sLOOP2        : Loop until the same value is read.

LD          (FLSCR),1100_1000B : Disable command sequence execution.

JP         XXXXH            : Jump to the flash memory area.

```

Example :This write control program reads data from address F000H and stores it to 98H in the RAM area.

```

LD          A,(0F000H)      : Read data from address F000H.

LD          (98H),A         : Store data to address 98H.

```


20. Serial PROM Mode

20.1 Outline

The TMP86FS64FG has a 2048 byte BOOTROM (Mask ROM) for programming to flash memory. The BOOTROM is available in the serial PROM mode, and controlled by TEST, BOOT and RESET pins. Communication is performed via UART. The serial PROM mode has seven types of operating mode: Flash memory writing, RAM loader, Flash memory SUM output, Product ID code output, Flash memory status output, Flash memory erasing and Flash memory read protection setting. Memory address mapping in the serial PROM mode differs from that in the MCU mode. Figure 20-1 shows memory address mapping in the serial PROM mode.

Table 20-1 Operating Range in the Serial PROM Mode

Parameter	Min	Max	Unit
Power supply	4.5	5.5	V
High frequency (Note)	2	16	MHz

Note: Though included in above operating range, some of high frequencies are not supported in the serial PROM mode. For details, refer to "Table 20-5".

20.2 Memory Mapping

The Figure 20-1 shows memory mapping in the Serial PROM mode and MCU mode.

In the serial PROM mode, the BOOTROM (Mask ROM) is mapped in addresses from 7800H to 7FFFH. The flash memory is divided into two banks for mapping. Therefore, when the RAM loader mode (60H) is used, it is required to specify the flash memory address according to Figure 20-1 (For detail of banks and control register, refer to the chapter of "Flash Memory Control Register".)

To use the Flash memory writing command (30H), specify the flash memory addresses from 1000H to FFFFH, that is the same addresses in the MCU mode, because the BOOTROM changes the flash memory address.

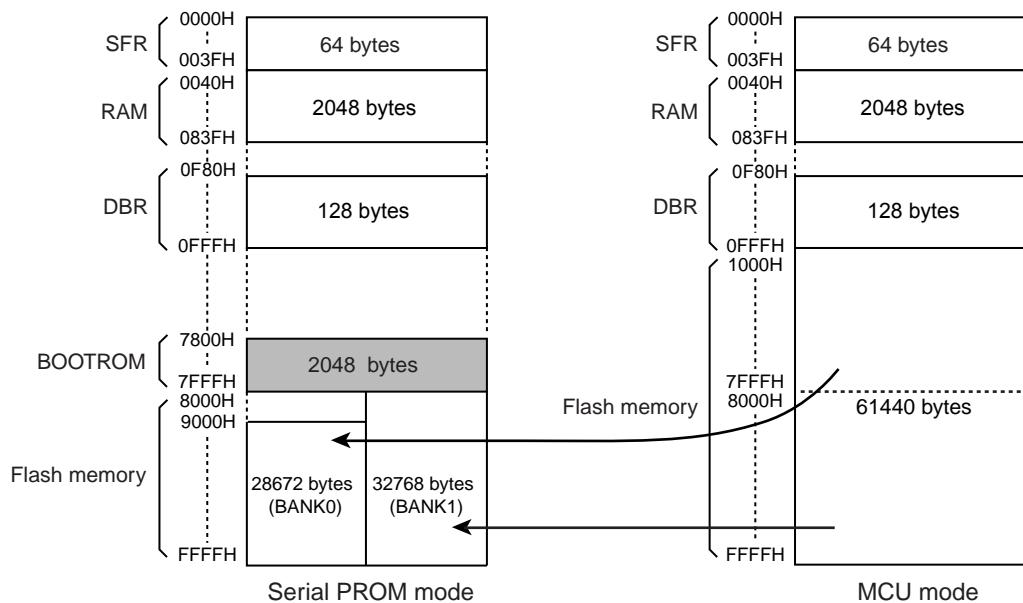



Figure 20-1 Memory Address Maps

20.3 Serial PROM Mode Setting

20.3.1 Serial PROM Mode Control Pins

To execute on-board programming, activate the serial PROM mode. Table 20-2 shows pin setting to activate the serial PROM mode.

Table 20-2 Serial PROM Mode Setting

Pin	Setting
TEST pin	High
BOOT/RXD2 pin	High
$\overline{\text{RESET}}$ pin	

Note: The BOOT pin is shared with the UART communication pin (RXD2 pin) in the serial PROM mode. This pin is used as UART communication pin after activating serial PROM mode

20.3.2 Pin Function

In the serial PROM mode, TXD2 (P45) and RXD2 (P44) are used as a serial interface pin.

Table 20-3 Pin Function in the Serial PROM Mode

Pin Name (Serial PROM Mode)	Input/ Output	Function	Pin Name (MCU Mode)
TXD2	Output	Serial data output	P45
BOOT/RXD2	Input/Input	Serial PROM mode control/Serial data input	P44
$\overline{\text{RESET}}$	Input	Serial PROM mode control	$\overline{\text{RESET}}$
TEST	Input	Fixed to high	TEST
VDD, AVDD	Power supply	4.5 to 5.5 V	
VSS, AVSS	Power supply	0 V	
VAREF	Power supply	Leave open or apply input reference voltage.	
I/O ports except P45, P44	I/O	These ports are in the high-impedance state in the serial PROM mode. The input level is fixed to the port inputs with a hardware feature to prevent overlap current. (The port inputs are invalid.) To make the port inputs valid, set the pin of the SPCR register to "1" by the RAM loader control program.	
XIN	Input	Self-oscillate with an oscillator.	(Note 2)
XOUT	Output		

Note 1: During on-board programming with other parts mounted on a user board, be careful no to affect these communication control pins.

Note 2: Operating range of high frequency in serial PROM mode is 2 MHz to 16 MHz.

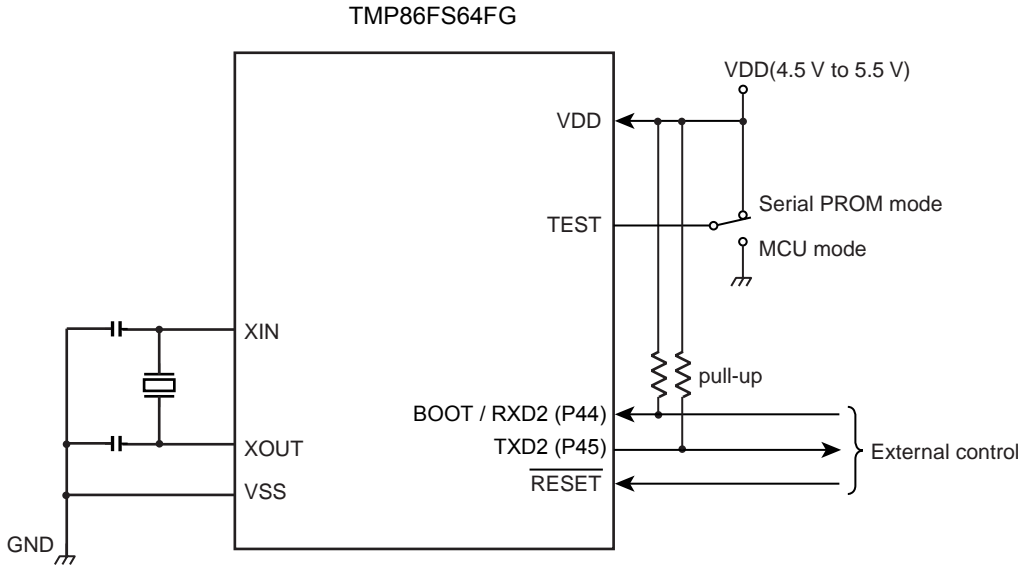


Figure 20-2 Serial PROM Mode Pin Setting

Note 1: For connection of other pins, refer to " Table 20-3 Pin Function in the Serial PROM Mode ".

20.3.3 Example Connection for On-Board Writing

Figure 20-3 shows an example connection to perform on-board writing.

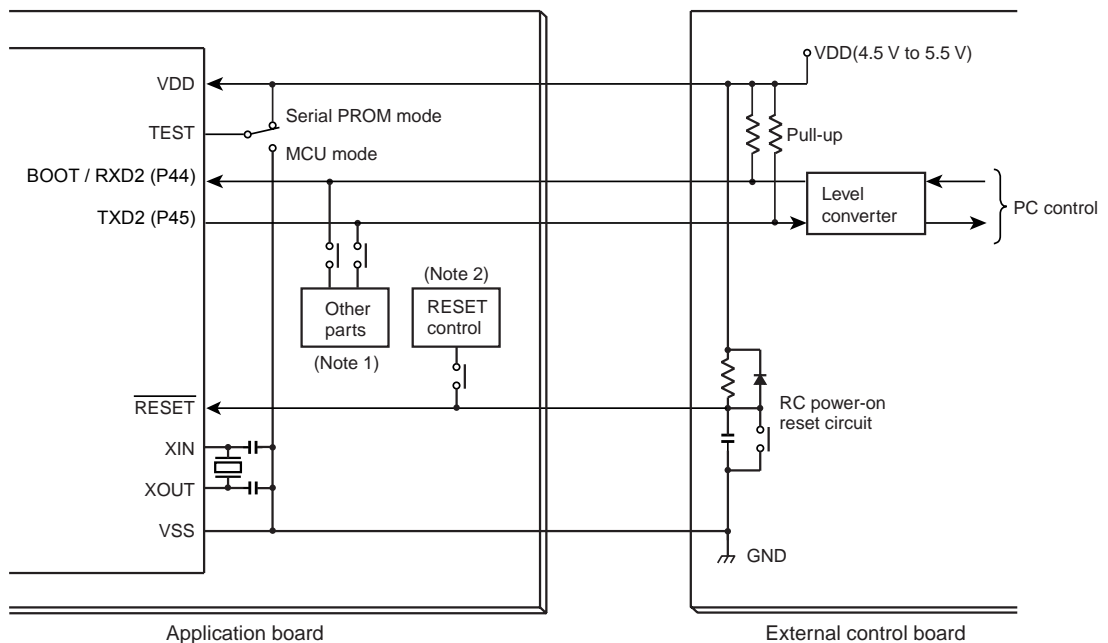


Figure 20-3 Example Connection for On-Board Writing

Note 1: When other parts on the application board effect the UART communication in the serial PROM mode, isolate these pins by a jumper or switch.

Note 2: When the reset control circuit on the application board effects activation of the serial PROM mode, isolate the pin by a jumper or switch.

Note 3: For connection of other pins, refer to " Table 20-3 Pin Function in the Serial PROM Mode ".

20.3.4 Activating the Serial PROM Mode

The following is a procedure to activate the serial PROM mode. " Figure 20-4 Serial PROM Mode Timing " shows a serial PROM mode timing.

1. Supply power to the VDD pin.
2. Set the $\overline{\text{RESET}}$ pin to low.
3. Set the TEST pin and BOOT/RXD2 pins to high.
4. Wait until the power supply and clock oscillation stabilize.
5. Set the $\overline{\text{RESET}}$ pin to high.
6. Input the matching data (5AH) to the BOOT/RXD2 pin after setup sequence. For details of the setup timing, refer to " 20.16 UART Timing ".

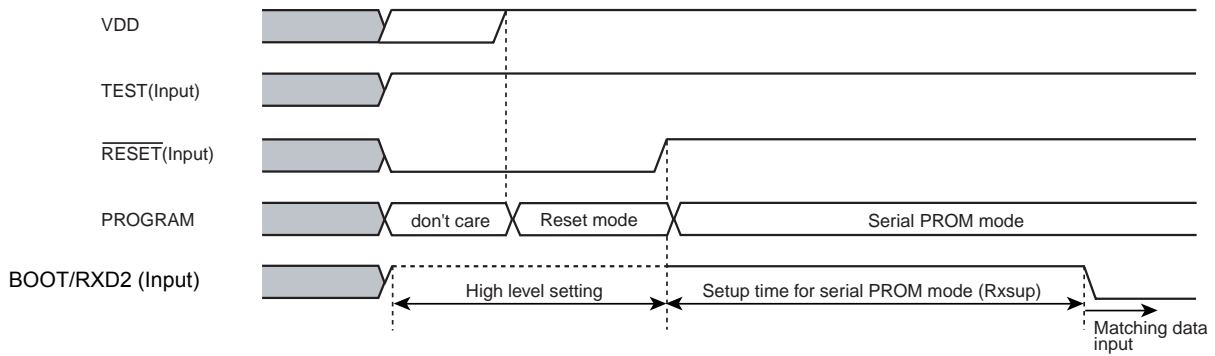


Figure 20-4 Serial PROM Mode Timing

20.4 Interface Specifications for UART

The following shows the UART communication format used in the serial PROM mode.

To perform on-board programming, the communication format of the write controller must also be set in the same manner.

The default baud rate is 9600 bps regardless of operating frequency of the microcontroller. The baud rate can be modified by transmitting the baud rate modification data shown in Table 1-4 to TMP86FS64FG. The Table 20-5 shows an operating frequency and baud rate. The frequencies which are not described in Table 20-5 can not be used.

- Baud rate (Default): 9600 bps
- Data length: 8 bits
- Parity addition: None
- Stop bit: 1 bit

Table 20-4 Baud Rate Modification Data

Baud rate modification data	04H	05H	06H	07H	0AH	18H	28H
Baud rate (bps)	76800	62500	57600	38400	31250	19200	9600

Table 20-5 Operating Frequency and Baud Rate in the Serial PROM Mode

(Note 3)	Reference Baud Rate (bps)		76800		62500		57600		38400		31250		19200		9600		
	Baud Rate Modification Data		04H		05H		06H		07H		0AH		18H		28H		
	Ref. Frequency (MHz)	Rating (MHz)	Baud rate (bps)	(%)	(bps)	(%)	(bps)	(%)	(bps)	(%)	(bps)	(%)	(bps)	(%)	(bps)	(%)	
1	2	1.91 to 2.10	-	-	-	-	-	-	-	-	-	-	-	-	-	9615	+0.16
2	4	3.82 to 4.19	-	-	-	-	-	-	-	-	31250	0.00	19231	+0.16	9615	+0.16	
	4.19	3.82 to 4.19	-	-	-	-	-	-	-	-	32734	+4.75	20144	+4.92	10072	+4.92	
3	4.9152	4.70 to 5.16	-	-	-	-	-	-	38400	0.00	-	-	19200	0.00	9600	0.00	
	5	4.70 to 5.16	-	-	-	-	-	-	39063	+1.73	-	-	19531	+1.73	9766	+1.73	
4	6	5.87 to 6.45	-	-	-	-	-	-	-	-	-	-	-	-	9375	-2.34	
	6.144	5.87 to 6.45	-	-	-	-	-	-	-	-	-	-	-	-	9600	0.00	
5	7.3728	7.05 to 7.74	-	-	-	-	57600	0.00	-	-	-	-	19200	0.00	9600	0.00	
6	8	7.64 to 8.39	-	-	62500	0.00	-	-	38462	+0.16	31250	0.00	19231	+0.16	9615	+0.16	
7	9.8304	9.40 to 10.32	76800	0.00	-	-	-	-	38400	0.00	-	-	19200	0.00	9600	0.00	
	10	9.40 to 10.32	78125	+1.73	-	-	-	-	39063	+1.73	-	-	19531	+1.73	9766	+1.73	
8	12	11.75 to 12.90	-	-	-	-	57692	+0.16	-	-	31250	0.00	18750	-2.34	9375	-2.34	
	12.288	11.75 to 12.90	-	-	-	-	59077	+2.56	-	-	32000	+2.40	19200	0.00	9600	0.00	
	12.5	11.75 to 12.90	-	-	60096	-3.85	60096	+4.33	-	-	30048	-3.85	19531	+1.73	9766	+1.73	
9	14.7456	14.10 to 15.48	-	-	-	-	57600	0.00	38400	0.00	-	-	19200	0.00	9600	0.00	
10	16	15.27 to 16.77	76923	+0.16	62500	0.00	-	-	38462	+0.16	31250	0.00	19231	+0.16	9615	+0.16	

Note 1: "Ref. Frequency" and "Rating" show frequencies available in the serial PROM mode. Though the frequency is supported in the serial PROM mode, the serial PROM mode may not be activated correctly due to the frequency difference in the external controller (such as personal computer) and oscillator, and load capacitance of communication pins.

Note 2: It is recommended that the total frequency difference is within $\pm 3\%$ so that auto detection is performed correctly by the reference frequency.

Note 3: The external controller must transmit the matching data (5AH) repeatedly till the auto detection of baud rate is performed. This number indicates the number of times the matching data is transmitted for each frequency.

20.5 Operation Command

The eight commands shown in Table 20-6 are used in the serial PROM mode. After reset release, the TMP86FS64FG waits for the matching data (5AH).

Table 20-6 Operation Command in the Serial PROM Mode

Command Data	Operating Mode	Description
5AH	Setup	Matching data. Execute this command after releasing the reset.
F0H	Flash memory erasing	Erases the flash memory area (address 1000H to FFFFH).
30H	Flash memory writing	Writes to the flash memory area (address 1000H to FFFFH).
60H	RAM loader	Writes to the specified RAM area (address 0050H to 083FH).
90H	Flash memory SUM output	Outputs the 2-byte checksum upper byte and lower byte in this order for the entire area of the flash memory (address 1000H to FFFFH).
C0H	Product ID code output	Outputs the product ID code (13-byte data).
C3H	Flash memory status output	Outputs the status code (7-byte data) such as the read protection condition.
FAH	Flash memory read protection setting	Enables the read protection.

20.6 Operation Mode

The serial PROM mode has seven types of modes, that are (1) Flash memory erasing, (2) Flash memory writing, (3) RAM loader, (4) Flash memory SUM output, (5) Product ID code output, (6) Flash memory status output and (7) Flash memory read protection setting modes. Description of each mode is shown below.

1. Flash memory erasing mode

The flash memory is erased by the chip erase (erasing an entire flash area) or sector erase (erasing sectors in 4-kbyte units). The erased area is filled with FFH. When the read protection is enabled, the sector erase in the flash erasing mode can not be performed. To disable the read protection, perform the chip erase. Before erasing the flash memory, TMP86FS64FG checks the passwords except a blank product. If the password is not matched, the flash memory erasing mode is not activated.

2. Flash memory writing mode

Data is written to the specified flash memory address for each byte unit. The external controller must transmit the write data in the Intel Hex format (Binary). If no error is encountered till the end record, TMP86FS64FG calculates the checksum for the entire flash memory area (1000H to FFFFH), and returns the obtained result to the external controller. When the read protection is enabled, the flash memory writing mode is not activated. In this case, perform the chip erase command beforehand in the flash memory erasing mode. Before activating the flash memory writing mode, TMP86FS64FG checks the password except a blank product. If the password is not matched, flash memory writing mode is not activated.

3. RAM loader mode

The RAM loader transfers the data in Intel Hex format sent from the external controller to the internal RAM. When the transfer is completed normally, the RAM loader calculates the checksum. After transmitting the results, the RAM loader jumps to the RAM address specified with the first data record in order to execute the user program. When the read protection is enabled, the RAM loader mode is not activated. In this case, perform the chip erase beforehand in the flash memory erasing mode. Before activating the RAM loader mode, TMP86FS64FG checks the password except a blank product. If the password is not matched, flash RAM loader mode is not activated.

4. Flash memory SUM output mode

The checksum is calculated for the entire flash memory area (1000H to FFFFH), and the result is returned to the external controller. Since the BOOTROM does not support the operation command to read the flash memory, use this checksum to identify programs when managing revisions of application programs.

5. Product ID code output

The code used to identify the product is output. The code to be output consists of 13-byte data, which includes the information indicating the area of the ROM incorporated in the product. The external controller reads this code, and recognizes the product to write.

(In the case of TMP86FS64FG, the addresses from 1000H to FFFFH become the ROM area.)

6. Flash memory status output mode
The status of the area from FFE0H to FFFFH, and the read protection condition are output as 7-byte code. The external controller reads this code to recognize the flash memory status.
7. Flash memory read protection setting mode
This mode disables reading the flash memory data in parallel PROM mode. In the serial PROM mode, the flash memory writing and RAM loader modes are disabled. To disable the flash memory read protection, perform the chip erase in the flash memory erasing mode.

20.6.1 Flash Memory Erasing Mode (Operating command: F0H)

Table 20-7 shows the flash memory erasing mode.

Table 20-7 Flash Memory Erasing Mode

	Transfer Byte	Transfer Data from the External Controller to TMP86FS64FG	Baud Rate	Transfer Data from TMP86FS64FG to the External Controller
BOOT ROM	1st byte 2nd byte	Matching data (5AH) -	9600 bps 9600 bps	- (Automatic baud rate adjustment) OK: Echo back data (5AH) Error: No data transmitted
	3rd byte 4th byte	Baud rate change data (Table 20-4) -	9600 bps 9600 bps	- OK: Echo back data Error: A1H × 3, A3H × 3, 62H × 3 (Note 1)
	5th byte 6th byte	Operation command data (F0H) -	Modified baud rate Modified baud rate	- OK: Echo back data (F0H) Error: A1H × 3, A3H × 3, 63H × 3 (Note 1)
	7th byte 8th byte	Password count storage address bit 15 to 08 (Note 4, 5)	Modified baud rate Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	9th byte 10th byte	Password count storage address bit 07 to 00 (Note 4, 5)	Modified baud rate Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	11th byte 12th byte	Password comparison start address bit 15 to 08 (Note 4, 5)	Modified baud rate Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	13th byte 14th byte	Password comparison start address bit 07 to 00 (Note 4, 5)	Modified baud rate Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	15th byte : m'th byte	Password string (Note 4, 5) -	Modified baud rate Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	n'th - 2 byte	Erase area specification (Note 2)	Modified baud rate	-
	n'th - 1 byte	-	Modified baud rate	OK: Checksum (Upper byte) (Note 3) Error: Nothing transmitted
	n'th byte	-	Modified baud rate	OK: Checksum (Lower byte) (Note 3) Error: Nothing transmitted
	n'th + 1 byte	(Wait for the next operation command data)	Modified baud rate	-

Note 1: "xxH × 3" indicates that the device enters the halt condition after transmitting 3 bytes of xxh.

Note 2: Refer to " 20.13 Specifying the Erasure Area ".

Note 3: Refer to " 20.8 Checksum (SUM) ".

Note 4: Refer to " 20.10 Passwords ".

Note 5: Do not transmit the password string for a blank product.

Note 6: When a password error occurs, TMP86FS64FG stops UART communication and enters the halt mode. Therefore, when a password error occurs, initialize TMP86FS64FG by the $\overline{\text{RESET}}$ pin and reactivate the serial PROM mode.

Note 7: If an error occurs during transfer of a password address or a password string, TMP86FS64FG stops UART communication and enters the halt condition. Therefore, when a password error occurs, initialize TMP86FS64FG by the $\overline{\text{RESET}}$ pin and reactivate the serial PROM mode.

Description of the flash memory erasing mode

1. The 1st through 4th bytes of the transmitted and received data contain the same data as in the flash memory writing mode.

2. The 5th byte of the received data contains the command data in the flash memory erasing mode (F0H).
3. When the 5th byte of the received data contains the operation command data shown in Table 20-6, the device echoes back the value which is the same data in the 6th byte position of the received data (in this case, F0H). If the 5th byte of the received data does not contain the operation command data, the device enters the halt condition after sending 3 bytes of the operation command error code (63H).
4. The 7th through m'th bytes of the transmitted and received data contain the same data as in the flash memory writing mode. In the case of a blank product, do not transmit a password string. (Do not transmit a dummy password string.)
5. The n'th - 2 byte contains the erasure area specification data. The upper 4 bits and lower 4 bits specify the start address and end address of the erasure area, respectively. For the detailed description, see "1.13 Specifying the Erasure Area".
6. The n'th - 1 byte and n'th byte contain the upper and lower bytes of the checksum, respectively. For how to calculate the checksum, refer to "1.8 Checksum (SUM)". Checksum is calculated unless a receiving error or Intel Hex format error occurs. After sending the end record, the external controller judges whether the transmission is completed correctly by receiving the checksum sent by the device.
7. After sending the checksum, the device waits for the next operation command data.

20.6.2 Flash Memory Writing Mode (Operation command: 30H)

Table 20-8 shows flash memory writing mode process.

Table 20-8 Flash Memory Writing Mode Process

	Transfer Byte	Transfer Data from External Controller to TMP86FS64FG	Baud Rate	Transfer Data from TMP86FS64FG to External Controller
BOOT ROM	1st byte 2nd byte	Matching data (5Ah) -	9600 bps 9600 bps	- (Automatic baud rate adjustment) OK: Echo back data (5AH) Error: Nothing transmitted
	3rd byte 4th byte	Baud rate modification data (See Table 20-4) -	9600 bps 9600 bps	- OK: Echo back data Error: A1H × 3, A3H × 3, 62H × 3 (Note 1)
	5th byte 6th byte	Operation command data (30H) -	Modified baud rate Modified baud rate	- OK: Echo back data (30H) Error: A1H × 3, A3H × 3, 63H × 3 (Note 1)
	7th byte 8th byte	Password count storage address bit 15 to 08 (Note 4)	Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	9th byte 10th byte	Password count storage address bit 07 to 00 (Note 4)	Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	11th byte 12th byte	Password comparison start address bit 15 to 08 (Note 4)	Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	13th byte 14th byte	Password comparison start address bit 07 to 00 (Note 4)	Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	15th byte : m'th byte	Password string (Note 5) -	Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	m'th + 1 byte : n'th - 2 byte	Intel Hex format (binary) (Note 2)	Modified baud rate	- -
	n'th - 1 byte	-	Modified baud rate	OK: SUM (Upper byte) (Note 3) Error: Nothing transmitted
	n'th byte	-	Modified baud rate	OK: SUM (Lower byte) (Note 3) Error: Nothing transmitted
	n'th + 1 byte	(Wait state for the next operation command data)	Modified baud rate	-

Note 1: "xxH × 3" indicates that the device enters the halt condition after sending 3 bytes of xxH. For details, refer to " 20.7 Error Code ".

Note 2: Refer to " 20.9 Intel Hex Format (Binary) ".

Note 3: Refer to " 20.8 Checksum (SUM) ".

Note 4: Refer to " 20.10 Passwords ".

Note 5: If addresses from FFE0H to FFFFH are filled with "FFH", the passwords are not compared because the device is considered as a blank product. Transmitting a password string is not required. Even in the case of a blank product, it is required to specify the password count storage address and the password comparison start address. Transmit these data from the external controller. If a password error occurs due to incorrect password count storage address or password comparison start address, TMP86FS64FG stops UART communication and enters the halt condition. Therefore, when a password error occurs, initialize TMP86FS64FG by the $\overline{\text{RESET}}$ pin and reactivate the serial ROM mode.

Note 6: If the read protection is enabled or a password error occurs, TMP86FS64FG stops UART communication and enters the halt condition. In this case, initialize TMP86FS64FG by the $\overline{\text{RESET}}$ pin and reactivate the serial ROM mode.

Note 7: If an error occurs during the reception of a password address or a password string, TMP86FS64FG stops UART communication and enters the halt condition. In this case, initialize TMP86FS64FG by the $\overline{\text{RESET}}$ pin and reactivate the serial PROM mode.

Description of the flash memory writing mode

1. The 1st byte of the received data contains the matching data. When the serial PROM mode is activated, TMP86FS64FG (hereafter called device), waits to receive the matching data (5AH). Upon reception of the matching data, the device automatically adjusts the UART's initial baud rate to 9600 bps.
2. When receiving the matching data (5AH), the device transmits an echo back data (5AH) as the second byte data to the external controller. If the device can not recognize the matching data, it does not transmit the echo back data and waits for the matching data again with automatic baud rate adjustment. Therefore, the external controller should transmit the matching data repeatedly till the device transmits an echo back data. The transmission repetition count varies depending on the frequency of device. For details, refer to Table 20-5.
3. The 3rd byte of the received data contains the baud rate modification data. The five types of baud rate modification data shown in Table 20-4 are available. Even if baud rate is not modified, the external controller should transmit the initial baud rate data (28H: 9600 bps).
4. Only when the 3rd byte of the received data contains the baud rate modification data corresponding to the device's operating frequency, the device echoes back data the value which is the same data in the 4th byte position of the received data. After the echo back data is transmitted, baud rate modification becomes effective. If the 3rd byte of the received data does not contain the baud rate modification data, the device enters the halts condition after sending 3 bytes of baud rate modification error code (62H).
5. The 5th byte of the received data contains the command data (30H) to write the flash memory.
6. When the 5th byte of the received data contains the operation command data shown in Table 1-6, the device echoes back the value which is the same data in the 6th byte position of the received data (in this case, 30H). If the 5th byte of the received data does not contain the operation command data, the device enters the halt condition after sending 3 bytes of the operation command error code (63H).
7. The 7th byte contains the data for 15 to 8 bits of the password count storage address. When the data received with the 7th byte has no receiving error, the device does not send any data. If a receiving error or password error occurs, the device does not send any data and enters the halt condition.
8. The 9th byte contains the data for 7 to 0 bits of the password count storage address. When the data received with the 9th byte has no receiving error, the device does not send any data. If a receiving error or password error occurs, the device does not send any data and enters the halt condition.
9. The 11th byte contains the data for 15 to 8 bits of the password comparison start address. When the data received with the 11th byte has no receiving error, the device does not send any data. If a receiving error or password error occurs, the device does not send any data and enters the halt condition.
10. The 13th byte contains the data for 7 to 0 bits of the password comparison start address. When the data received with the 13th byte has no receiving error, the device does not send any data. If a receiving error or password error occurs, the device does not send any data and enters the halt condition.
11. The 15th through m'th bytes contain the password data. The number of passwords becomes the data (N) stored in the password count storage address. The external password data is compared with N-byte data from the address specified by the password comparison start address. The external controller should send N-byte password data to the device. If the passwords do not match, the device enters the halt condition without returning an error code to the external controller. If the addresses from FFE0H to FFFFH are filled with "FFH", the passwords are not compared because the device is considered as a blank product.
12. The m'th + 1 through n'th - 2 bytes of the received data contain the binary data in the Intel Hex format. No received data is echoed back to the external controller. After receiving the start mark (3AH for ":" in the Intel Hex format), the device starts data record reception. Therefore, the received data except 3AH is ignored until the start mark is received. After receiving the start mark, the device receives the data record, that consists of data length, address, record type, write data and checksum. Since the device starts checksum calculation after receiving an end record, the external controller should wait for the checksum after sending the end record. If a receiving error or Intel Hex format error occurs, the device enters the halts condition without returning an error code to the external controller.
13. The n'th - 1 and n'th bytes contain the checksum upper and lower bytes. For details on how to calculate the SUM, refer to " 20.8 Checksum (SUM) ". The checksum is calculated only when the end record is detected and no receiving error or Intel Hex format error occurs. After sending the end

record, the external controller judges whether the transmission is completed correctly by receiving the checksum sent by the device.

14. After transmitting the checksum, the device waits for the next operation command data.

Note 1: Do not write only the address from FFE0H to FFFFH when all flash memory data is the same. If only these area are written, the subsequent operation can not be executed due to password error.

Note 2: To rewrite data to Flash memory addresses at which data (including FFH) is already written, make sure to erase the existing data by "sector erase" or "chip erase" before rewriting data.

20.6.3 RAM Loader Mode (Operation Command: 60H)

Table 20-9 shows RAM loader mode process.

Table 20-9 RAM Loader Mode Process

	Transfer Bytes	Transfer Data from External Controller to TMP86FS64FG	Baud Rate	Transfer Data from TMP86FS64FG to External Controller	
BOOT ROM	1st byte 2nd byte	Matching data (5AH) -	9600 bps 9600 bps	- (Automatic baud rate adjustment) OK: Echo back data (5AH) Error: Nothing transmitted	
	3rd byte 4th byte	Baud rate modification data (See Table 20-4) -	9600 bps 9600 bps	- OK: Echo back data Error: A1H × 3, A3H × 3, 62H × 3 (Note 1)	
	5th byte 6th byte	Operation command data (60H) -	Modified baud rate Modified baud rate	- OK: Echo back data (60H) Error: A1H × 3, A3H × 3, 63H × 3 (Note 1)	
	7th byte 8th byte	Password count storage address bit 15 to 08 (Note 4)	Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted	
	9th byte 10th byte	Password count storage address bit 07 to 00 (Note 4)	Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted	
	11th byte 12th byte	Password comparison start address bit 15 to 08 (Note 4)	Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted	
	13th byte 14th byte	Password comparison start address bit 07 to 00 (Note 4)	Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted	
	15th byte : m'th byte	Password string (Note 5) -	Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted	
	m'th + 1 byte : n'th - 2 byte	Intel Hex format (Binary) (Note 2)	Modified baud rate Modified baud rate	- -	
	n'th - 1 byte	-	Modified baud rate	OK: SUM (Upper byte) (Note 3) Error: Nothing transmitted	
	n'th byte	-	Modified baud rate	OK: SUM (Lower byte) (Note 3) Error: Nothing transmitted	
	RAM	-	The program jumps to the start address of RAM in which the first transferred data is written.		

Note 1: "xxH × 3" indicates that the device enters the halt condition after sending 3 bytes of xxH. For details, refer to " 20.7 Error Code ".

Note 2: Refer to " 20.9 Intel Hex Format (Binary) ".

Note 3: Refer to " 20.8 Checksum (SUM) ".

Note 4: Refer to " 20.10 Passwords ".

Note 5: If addresses from FFE0H to FFFFH are filled with "FFH", the passwords are not compared because the device is considered as a blank product. Transmitting a password string is not required. Even in the case of a blank product, it is required to specify the password count storage address and the password comparison start address. Transmit these data from the external controller. If a password error occurs due to incorrect password count storage address or password comparison start address, TMP86FS64FG stops UART communication and enters the halt condition. Therefore, when a password error occurs, initialize TMP86FS64FG by the $\overline{\text{RESET}}$ pin and reactivate the serial ROM mode.

Note 6: After transmitting a password string, the external controller must not transmit only an end record. If receiving an end record after a password string, the device may not operate correctly.

Note 7: If the read protection is enabled or a password error occurs, TMP86FS64FG stops UART communication and enters the halt condition. In this case, initialize TMP86FS64FG by the $\overline{\text{RESET}}$ pin and reactivate the serial PROM mode.

Note 8: If an error occurs during the reception of a password address or a password string, TMP86FS64FG stops UART communication and enters the halt condition. In this case, initialize TMP86FS64FG by the RESET pin and reactivate the serial PROM mode.

Description of RAM loader mode

1. The 1st through 4th bytes of the transmitted and received data contains the same data as in the flash memory writing mode.
2. In the 5th byte of the received data contains the RAM loader command data (60H).
3. When the 5th byte of the received data contains the operation command data shown in Table 1-6, the device echoes back the value which is the same data in the 6th byte position (in this case, 60H). If the 5th byte does not contain the operation command data, the device enters the halt condition after sending 3 bytes of operation command error code (63H).
4. The 7th through m'th bytes of the transmitted and received data contain the same data as in the flash memory writing mode.
5. The m'th + 1 through n'th - 2 bytes of the received data contain the binary data in the Intel Hex format. No received data is echoed back to the external controller. After receiving the start mark (3AH for ":" in the Intel Hex format), the device starts data record reception. Therefore, the received data except 3AH is ignored until the start mark is received. After receiving the start mark, the device receives the data record, that consists of data length, address, record type, write data and checksum. The writing data of the data record is written into RAM specified by address. Since the device starts checksum calculation after receiving an end record, the external controller should wait for the checksum after sending the end record. If a receiving error or Intel Hex format error occurs, the device enters the halts condition without returning an error code to the external controller.
6. The n'th - 1 and n'th bytes contain the checksum upper and lower bytes. For details on how to calculate the SUM, refer to " 20.8 Checksum (SUM) ". The checksum is calculated only when the end record is detected and no receiving error or Intel Hex format error occurs. After sending the end record, the external controller judges whether the transmission is completed correctly by receiving the checksum sent by the device.
7. After transmitting the checksum to the external controller, the boot program jumps to the RAM address that is specified by the first received data record.

Note 1: To rewrite data to Flash memory addresses at which data (including FFH) is already written, make sure to erase the existing data by "sector erase" or "chip erase" before rewriting data.

20.6.4 Flash Memory SUM Output Mode (Operation Command: 90H)

Table 20-10 shows flash memory SUM output mode process.

Table 20-10 Flash Memory SUM Output Process

	Transfer Bytes	Transfer Data from External Controller to TMP86FS64FG	Baud Rate	Transfer Data from TMP86FS64FG to External Controller
BOOT ROM	1st byte 2nd byte	Matching data (5AH) -	9600 bps 9600 bps	- (Automatic baud rate adjustment) OK: Echo back data (5AH) Error: Nothing transmitted
	3rd byte 4th byte	Baud rate modification data (See Table 20-4) -	9600 bps 9600 bps	- OK: Echo back data Error: A1H × 3, A3H × 3, 62H × 3 (Note 1)
	5th byte 6th byte	Operation command data (90H) -	Modified baud rate Modified baud rate	- OK: Echo back data (90H) Error: A1H × 3, A3H × 3, 63H × 3 (Note 1)
	7th byte	-	Modified baud rate	OK: SUM (Upper byte) (Note 2) Error: Nothing transmitted
	8th byte	-	Modified baud rate	OK: SUM (Lower byte) (Note 2) Error: Nothing transmitted
	9th byte	(Wait for the next operation command data)	Modified baud rate	-

Note 1: "xxH × 3" indicates that the device enters the halt condition after sending 3 bytes of xxH. For details, refer to " 20.7 Error Code ".

Note 2: Refer to " 20.8 Checksum (SUM) ".

Description of the flash memory SUM output mode

1. The 1st through 4th bytes of the transmitted and received data contains the same data as in the flash memory writing mode.
2. The 5th byte of the received data contains the command data in the flash memory SUM output mode (90H).
3. When the 5th byte of the received data contains the operation command data shown in Table 1-6, the device echoes back the value which is the same data in the 6th byte position of the received data (in this case, 90H). If the 5th byte of the received data does not contain the operation command data, the device enters the halt condition after transmitting 3 bytes of operation command error code (63H).
4. The 7th and the 8th bytes contain the upper and lower bits of the checksum, respectively. For how to calculate the checksum, refer to " 20.8 Checksum (SUM) ".
5. After sending the checksum, the device waits for the next operation command data.

20.6.5 Product ID Code Output Mode (Operation Command: C0H)

Table 20-11 shows product ID code output mode process.

Table 20-11 Product ID Code Output Process

	Transfer Bytes	Transfer Data from External Controller to TMP86FS64FG	Baud Rate	Transfer Data from TMP86FS64FG to External Controller	
BOOT ROM	1st byte 2nd byte	Matching data (5AH) -	9600 bps 9600 bps	- (Automatic baud rate adjustment) OK: Echo back data (5AH) Error: Nothing transmitted	
	3rd byte 4th byte	Baud rate modification data (See Table 20-4) -	9600 bps 9600 bps	- OK: Echo back data Error: A1H × 3, A3H × 3, 62H × 3 (Note 1)	
	5th byte 6th byte	Operation command data (C0H) -	Modified baud rate Modified baud rate	- OK: Echo back data (C0H) Error: A1H × 3, A3H × 3, 63H × 3 (Note 1)	
	7th byte		Modified baud rate	3AH	Start mark
	8th byte		Modified baud rate	0AH	The number of transfer data (from 9th to 18th bytes)
	9th byte		Modified baud rate	02H	Length of address (2 bytes)
	10th byte		Modified baud rate	1DH	Reserved data
	11th byte		Modified baud rate	00H	Reserved data
	12th byte		Modified baud rate	00H	Reserved data
	13th byte		Modified baud rate	00H	Reserved data
	14th byte		Modified baud rate	01H	ROM block count (1 block)
	15th byte		Modified baud rate	10H	First address of ROM (Upper byte)
	16th byte		Modified baud rate	00H	First address of ROM (Lower byte)
	17th byte		Modified baud rate	FFH	End address of ROM (Upper byte)
	18th byte		Modified baud rate	FFH	End address of ROM (Lower byte)
	19th byte		Modified baud rate	D2H	Checksum of transferred data (9th through 18th byte)
	20th byte	(Wait for the next operation command data)	Modified baud rate	-	

Note: "xxH × 3" indicates that the device enters the halt condition after sending 3 bytes of xxH. For details, refer to " 20.7 Error Code ".

Description of Product ID code output mode

1. The 1st through 4th bytes of the transmitted and received data contain the same data as in the flash memory writing mode.
2. The 5th byte of the received data contains the product ID code output mode command data (C0H).
3. When the 5th byte contains the operation command data shown in Table 20-6, the device echoes back the value which is the same data in the 6th byte position of the received data (in this case, C0H). If the 5th byte data does not contain the operation command data, the device enters the halt condition after sending 3 bytes of operation command error code (63H).
4. The 9th through 19th bytes contain the product ID code. For details, refer to " 20.11 Product ID Code ".

5. After sending the checksum, the device waits for the next operation command data.

20.6.6 Flash Memory Status Output Mode (Operation Command: C3H)

Table 20-12 shows Flash memory status output mode process.

Table 20-12 Flash Memory Status Output Mode Process

	Transfer Bytes	Transfer Data from External Controller to TMP86FS64FG	Baud Rate	Transfer Data from TMP86FS64FG to External Controller	
BOOT ROM	1st byte 2nd byte	Matching data (5AH) -	9600 bps 9600 bps	- (Automatic baud rate adjustment) OK: Echo back data (5AH) Error: Nothing transmitted	
	3rd byte 4th byte	Baud rate modification data (See Table 20-4) -	9600 bps 9600 bps	- OK: Echo back data Error: A1H × 3, A3H × 3, 62H × 3 (Note 1)	
	5th byte 6th byte	Operation command data (C3H) -	Modified baud rate Modified baud rate	- OK: Echo back data (C3H) Error: A1H × 3, A3H × 3, 63H × 3 (Note 1)	
	7th byte		Modified baud rate	3AH	Start mark
	8th byte		Modified baud rate	04H	Byte count (from 9th to 12th byte)
	9th byte		Modified baud rate	00H to 03H	Status code 1
	10th byte		Modified baud rate	00H	Reserved data
	11th byte		Modified baud rate	00H	Reserved data
	12th byte		Modified baud rate	00H	Reserved data
	13th byte		Modified baud rate	Checksum 2's complement for the sum of 9th through 12th bytes 9th byte Checksum 00H: 00H 01H: FFH 02H: FEH 03H: FDH	
	14th byte	(Wait for the next operation command data)	Modified baud rate	-	

Note 1: "xxH × 3" indicates that the device enters the halt condition after sending 3 bytes of xxH. For details, refer to " 20.7 Error Code ".

Note 2: For the details on status code 1, refer to " 20.12 Flash Memory Status Code ".

Description of Flash memory status output mode

1. The 1st through 4th bytes of the transmitted and received data contain the same data as in the Flash memory writing mode.
2. The 5th byte of the received data contains the flash memory status output mode command data (C3H).
3. When the 5th byte contains the operation command data shown in Table 20-6, the device echoes back the value which is the same data in the 6th byte position of the received data (in this case, C3H). If the 5th byte does not contain the operation command data, the device enters the halt condition after sending 3 bytes of operation command error code (63H).
4. The 9th through 13th bytes contain the status code. For details on the status code, refer to " 20.12 Flash Memory Status Code ".
5. After sending the status code, the device waits for the next operation command data.

20.6.7 Flash Memory Read Protection Setting Mode (Operation Command: FAH)

Table 20-13 shows Flash memory read protection setting mode process.

Table 20-13 Flash Memory Read Protection Setting Mode Process

	Transfer Bytes	Transfer Data from External Controller to TMP86FS64FG	Baud Rate	Transfer Data from TMP86FS64FG to External Controller
BOOT ROM	1st byte 2nd byte	Matching data (5AH) -	9600 bps 9600 bps	- (Automatic baud rate adjustment) OK: Echo back data (5AH) Error: Nothing transmitted
	3rd byte 4th byte	Baud rate modification data (See Table 20-4) -	9600 bps 9600 bps	- OK: Echo back data Error: A1H × 3, A3H × 3, 62H × 3 (Note 1)
	5th byte 6th byte	Operation command data (FAH) -	Modified baud rate Modified baud rate	- OK: Echo back data (FAH) Error: A1H × 3, A3H × 3, 63H × 3 (Note 1)
	7th byte 8th byte	Password count storage address 15 to 08 (Note 2)	Modified baud rate Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	9th byte 10th byte	Password count storage address 07 to 00 (Note 2)	Modified baud rate Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	11th byte 12th byte	Password comparison start address 15 to 08 (Note 2)	Modified baud rate Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	13th byte 14th byte	Password comparison start address 07 to 00 (Note 2)	Modified baud rate Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	15th byte : m'th byte	Password string (Note 2) -	Modified baud rate Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	n'th byte	-	Modified baud rate	OK: FBH (Note 3) Error: Nothing transmitted
	n'+1th byte	(Wait for the next operation command data)	Modified baud rate	-

Note 1: "xxH × 3" indicates that the device enters the halt condition after sending 3 bytes of xxH. For details, refer to " 20.7 Error Code ".

Note 2: Refer to " 20.10 Passwords ".

Note 3: If the read protection is enabled for a blank product or a password error occurs for a non-blank product, TMP86FS64FG stops UART communication and enters the halt mode. In this case, initialize TMP86FS64FG by the $\overline{\text{RESET}}$ pin and reactivate the serial PROM mode.

Note 4: If an error occurs during reception of a password address or a password string, TMP86FS64FG stops UART communication and enters the halt mode. In this case, initialize TMP86FS64FG by the $\overline{\text{RESET}}$ pin and reactivate the serial PROM mode.

Description of the Flash memory read protection setting mode

1. The 1st through 4th bytes of the transmitted and received data contain the same data as in the Flash memory writing mode.
2. The 5th byte of the received data contains the command data in the flash memory status output mode (FAH).
3. When the 5th byte of the received data contains the operation command data shown in Table 1-6, the device echoes back the value which is the same data in the 6th byte position of the received data (in

this case, FAH). If the 5th byte does not contain the operation command data, the device enters the halt condition after transmitting 3 bytes of operation command error code (63H).

4. The 7th through m'th bytes of the transmitted and received data contain the same data as in the flash memory writing mode.
5. The n'th byte contains the status to be transmitted to the external controller in the case of the successful read protection.

20.7 Error Code

When detecting an error, the device transmits the error code to the external controller, as shown in Table 20-14.

Table 20-14 Error Code

Transmit Data	Meaning of Error Data
62H, 62H, 62H	Baud rate modification error.
63H, 63H, 63H	Operation command error.
A1H, A1H, A1H	Framing error in the received data.
A3H, A3H, A3H	Overrun error in the received data.

Note: If a password error occurs, TMP86FS64FG does not transmit an error code.

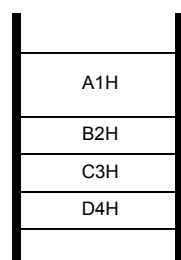
20.8 Checksum (SUM)

20.8.1 Calculation Method

The checksum (SUM) is calculated with the sum of all bytes, and the obtained result is returned as a word.

The data is read for each byte unit and the calculated result is returned as a word.

Example:



If the data to be calculated consists of the four bytes, the checksum of the data is as shown below.

$$\begin{aligned}
 A1H + B2H + C3H + D4H &= 02EAH \\
 \text{SUM (HIGH)} &= 02H \\
 \text{SUM (LOW)} &= EAH
 \end{aligned}$$

The checksum which is transmitted by executing the flash memory write command, RAM loader command, or flash memory SUM output command is calculated in the manner, as shown above.

20.8.2 Calculation data

The data used to calculate the checksum is listed in Table 20-15.

Table 20-15 Checksum Calculation Data

Operating Mode	Calculation Data	Description
Flash memory writing mode	Data in the entire area of the flash memory	Even when a part of the flash memory is written, the checksum of the entire flash memory area (1000H to FFFH) is calculated. The data length, address, record type and checksum in Intel Hex format are not included in the checksum.
Flash memory SUM output mode		
RAM loader mode	RAM data written in the first received RAM address through the last received RAM address	The length of data, address, record type and checksum in Intel Hex format are not included in the checksum.
Product ID Code Output mode	9th through 18th bytes of the transferred data	For details, refer to " 20.11 Product ID Code ".
Flash Memory Status Output mode	9th through 12th bytes of the transferred data	For details, refer to " 20.12 Flash Memory Status Code "
Flash Memory Erasing mode	All data in the erased area of the flash memory (the whole or part of the flash memory)	When the sector erase is executed, only the erased area is used to calculate the checksum. In the case of the chip erase, an entire area of the flash memory is used.

20.9 Intel Hex Format (Binary)

1. After receiving the checksum of a data record, the device waits for the start mark (3AH “:”) of the next data record. After receiving the checksum of a data record, the device ignores the data except 3AH transmitted by the external controller.
2. After transmitting the checksum of end record, the external controller must transmit nothing, and wait for the 2-byte receive data (upper and lower bytes of the checksum).
3. If a receiving error or Intel Hex format error occurs, the device enters the halt condition without returning an error code to the external controller. The Intel Hex format error occurs in the following case:

When the record type is not 00H, 01H, or 02H

When a checksum error occurs

When the data length of an extended record (record type = 02H) is not 02H

When the device receives the data record after receiving an extended record (record type = 02H) with extended address of 1000H or larger.

When the data length of the end record (record type = 01H) is not 00H

20.10 Passwords

The consecutive eight or more-byte data in the flash memory area can be specified to the password. TMP86FS64FG compares the data string specified to the password with the password string transmitted from the external controller. The area in which passwords can be specified is located at addresses 1000H to FF9FH. The area from FFA0H to FFFFH can not be specified as the passwords area.

If addresses from FFE0H through FFFFH are filled with “FFH”, the passwords are not compared because the product is considered as a blank product. Even in this case, the password count storage addresses and password comparison start address must be specified. Table 20-16 shows the password setting in the blank product and non-blank product.

Table 20-16 Password Setting in the Blank Product and Non-Blank Product

Password	Blank Product (Note 1)	Non-Blank Product
PNSA (Password count storage address)	$1000H \leq PNSA \leq FF9FH$	$1000H \leq PNSA \leq FF9FH$
PCSA (Password comparison start address)	$1000H \leq PCSA \leq FF9FH$	$1000H \leq PCSA \leq FFA0 - N$
N (Password count)	*	$8 \leq N$
Password string setting	Not required (Note 5)	Required (Note 2)

Note 1: When addresses from FFE0H through FFFFH are filled with “FFH”, the product is recognized as a blank product.

Note 2: The data including the same consecutive data (three or more bytes) can not be used as a password. (This causes a password error data. TMP86FS64FG transmits no data and enters the halt condition.)

Note 3: *: Don't care.

Note 4: When the above condition is not met, a password error occurs. If a password error occurs, the device enters the halt condition without returning the error code.

Note 5: In the flash memory writing mode or RAM loader mode, the blank product receives the Intel Hex format data immediately after receiving PCSA without receiving password strings. In this case, the subsequent processing is performed correctly because the blank product ignores the data except the start mark (3AH “:”) as the Intel Hex format data, even if the external controller transmits the dummy password string. However, if the dummy password string contains “3AH”, it is detected as the start mark erroneously. The microcontroller enters the halt mode. If this causes the problem, do not transmit the dummy password strings.

Note 6: In the flash memory erasing mode, the external controller must not transmit the password string for the blank product.

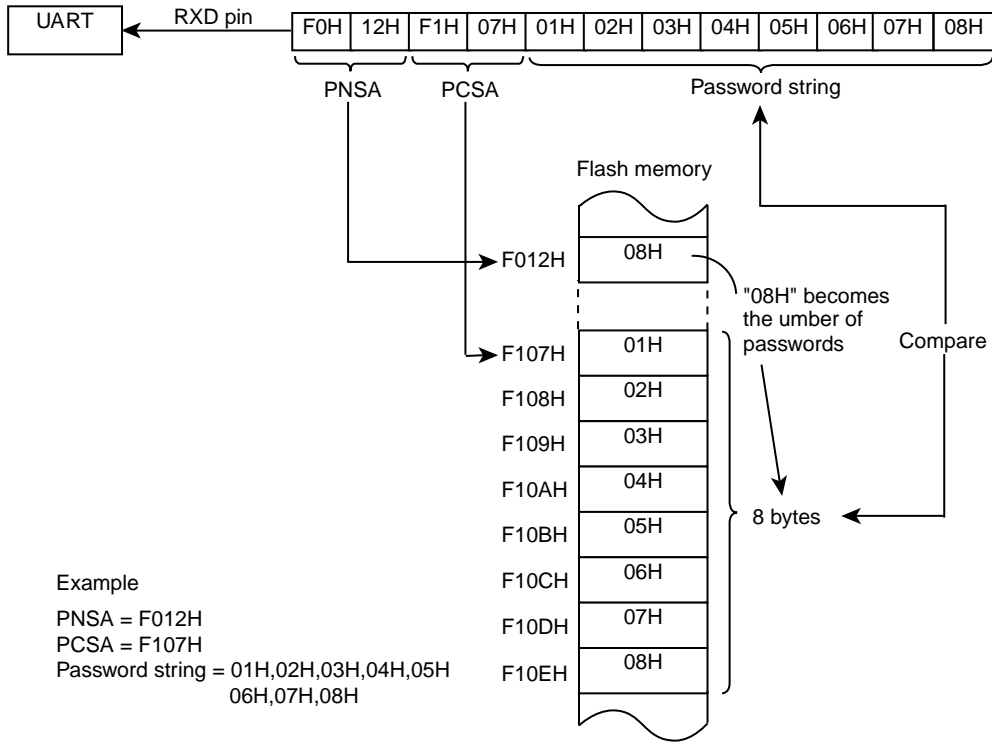


Figure 20-5 Password Comparison

20.10.1 Password String

The password string transmitted from the external controller is compared with the specified data in the flash memory. When the password string is not matched to the data in the flash memory, the device enters the halt condition due to the password error.

20.10.2 Handling of Password Error

If a password error occurs, the device enters the halt condition. In this case, reset the device to reactivate the serial PROM mode.

20.10.3 Password Management during Program Development

If a program is modified many times in the development stage, confusion may arise as to the password. Therefore, it is recommended to use a fixed password in the program development stage.

Example :Specify PNSA to F000H, and the password string to 8 bytes from address F001H (PCSA becomes F001H.)

Password Section code abs = 0F000H

DB	08H	: PNSA definition
DB	"CODE1234"	: Password string definition

20.11 Product ID Code

The product ID code is the 13-byte data containing the start address and the end address of ROM. Table 20-17 shows the product ID code format.

Table 20-17 Product ID Code Format

Data	Description	In the Case of TMP86FS64FG
1st	Start Mark (3AH)	3AH
2nd	The number of transfer data (10 bytes from 3rd to 12th byte)	0AH
3rd	Address length (2 bytes)	02H
4th	Reserved data	1DH
5th	Reserved data	00H
6th	Reserved data	00H
7th	Reserved data	00H
8th	ROM block count	01H
9th	The first address of ROM (Upper byte)	10H
10th	The first address of ROM (Lower byte)	00H
11th	The end address of ROM (Upper byte)	FFH
12th	The end address of ROM (Lower byte)	FFH
13th	Checksum of the transferred data (2's compliment for the sum of 3rd through 12th bytes)	D2H

20.12 Flash Memory Status Code

The flash memory status code is the 7-byte data including the read protection status and the status of the data from FFE0H to FFFFH. Table 20-18 shows the flash memory status code.

Table 20-18 Flash Memory Status Code

Data	Description	In the Case of TMP86FS64FG										
1st	Start mark	3AH										
2nd	Transferred data count (3rd through 6th byte)	04H										
3rd	Status code	00H to 03H (See figure below)										
4th	Reserved data	00H										
5th	Reserved data	00H										
6th	Reserved data	00H										
7th	Checksum of the transferred data (2's compliment for the sum of 3rd through 6th data)	<table border="0"> <tr> <td>3rd byte</td> <td>checksum</td> </tr> <tr> <td>00H</td> <td>00H</td> </tr> <tr> <td>01H</td> <td>FFH</td> </tr> <tr> <td>02H</td> <td>FEH</td> </tr> <tr> <td>03H</td> <td>FDH</td> </tr> </table>	3rd byte	checksum	00H	00H	01H	FFH	02H	FEH	03H	FDH
3rd byte	checksum											
00H	00H											
01H	FFH											
02H	FEH											
03H	FDH											

Status Code 1

7	6	5	4	3	2	1	0	
						RPENA	BLANK	(Initial Value: 0000 00**)

RPENA	Flash memory read protection status	0: Read protection is disabled. 1: Read protection is enabled.
BLANK	The status from FFE0H to FFFFH.	0: All data is FFH in the area from FFE0H to FFFFH. 1: The value except FFH is included in the area from FFE0H to FFFFH.

Some operation commands are limited by the flash memory status code 1. If the read protection is enabled, flash memory writing mode command and RAM loader mode command can not be executed. Erase all flash memory before executing these command.

RPENA	BLANK	Flash Memory Writing Mode	RAM Loader Mode	Flash memory SUM Output Mode	Product ID Code Output Mode	Flash Memory Status Output Mode	Flash Memory Erasing Mode		Read Protection Setting Mode
							Chip Erase	Sector Erase	
0	0	m	m	m	m	m	m		×
0	1	Pass	Pass	m	m	m	Pass		Pass
1	0	×	×	m	m	m	m	×	×
1	1	×	×	m	m	m	Pass	×	Pass

Note: m: The command can be executed.

Pass: The command can be executed with a password.

×: The command can not be executed.

(After echoing the command back to the external controller, TMP86FS64FG stops UART communication and enters the halt condition.)

20.13 Specifying the Erasure Area

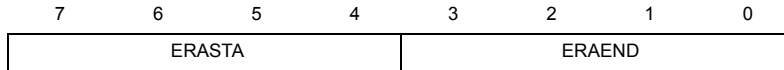
In the flash memory erasing mode, the erasure area of the flash memory is specified by n–2 byte data.

The start address of an erasure area is specified by ERASTA, and the end address is specified by ERAEND.

If ERASTA is equal to or smaller than ERAEND, the sector erase (erasure in 4 kbyte units) is executed. Executing the sector erase while the read protection is enabled results in an infinite loop.

If ERASTA is larger than ERAEND, the chip erase (erasure of an entire flash memory area) is executed and the read protection is disabled. Therefore, execute the chip erase (not sector erase) to disable the read protection.

Erasure Area Specification Data (n–2 byte data)



ERASTA	The start address of the erasure area	0000: from 0000H 0001: from 1000H 0010: from 2000H 0011: from 3000H 0100: from 4000H 0101: from 5000H 0110: from 6000H 0111: from 7000H 1000: from 8000H 1001: from 9000H 1010: from A000H 1011: from B000H 1100: from C000H 1101: from D000H 1110: from E000H 1111: from F000H
ERAEND	The end address of the erasure area	0000: to 0FFFH 0001: to 1FFFH 0010: to 2FFFH 0011: to 3FFFH 0100: to 4FFFH 0101: to 5FFFH 0110: to 6FFFH 0111: to 7FFFH 1000: to 8FFFH 1001: to 9FFFH 1010: to AFFFH 1011: to BFFFH 1100: to CFFFH 1101: to DFFFH 1110: to EFFFH 1111: to FFFFH

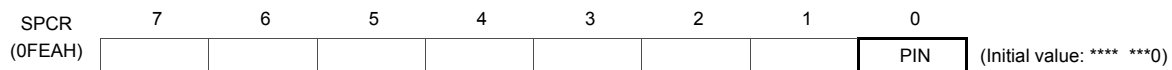
Note: When the sector erase is executed for the area containing no flash cell, TMP86FS64FG stops the UART communication and enters the halt condition.

20.14 Port Input Control Register

In the serial PROM mode, the input level is fixed to the all ports except P45 and P44 ports with a hardware feature to prevent overlap current to unused ports. (All port inputs and peripheral function inputs shared with the ports become invalid.) Therefore, to access to the flash memory in the RAM loader mode without UART communication, port inputs must be valid. To make port inputs valid, set the pin of the port input control register (SPCR) to “1”.

The SPCR register is not operated in the MCU mode.

Port Input Control Register

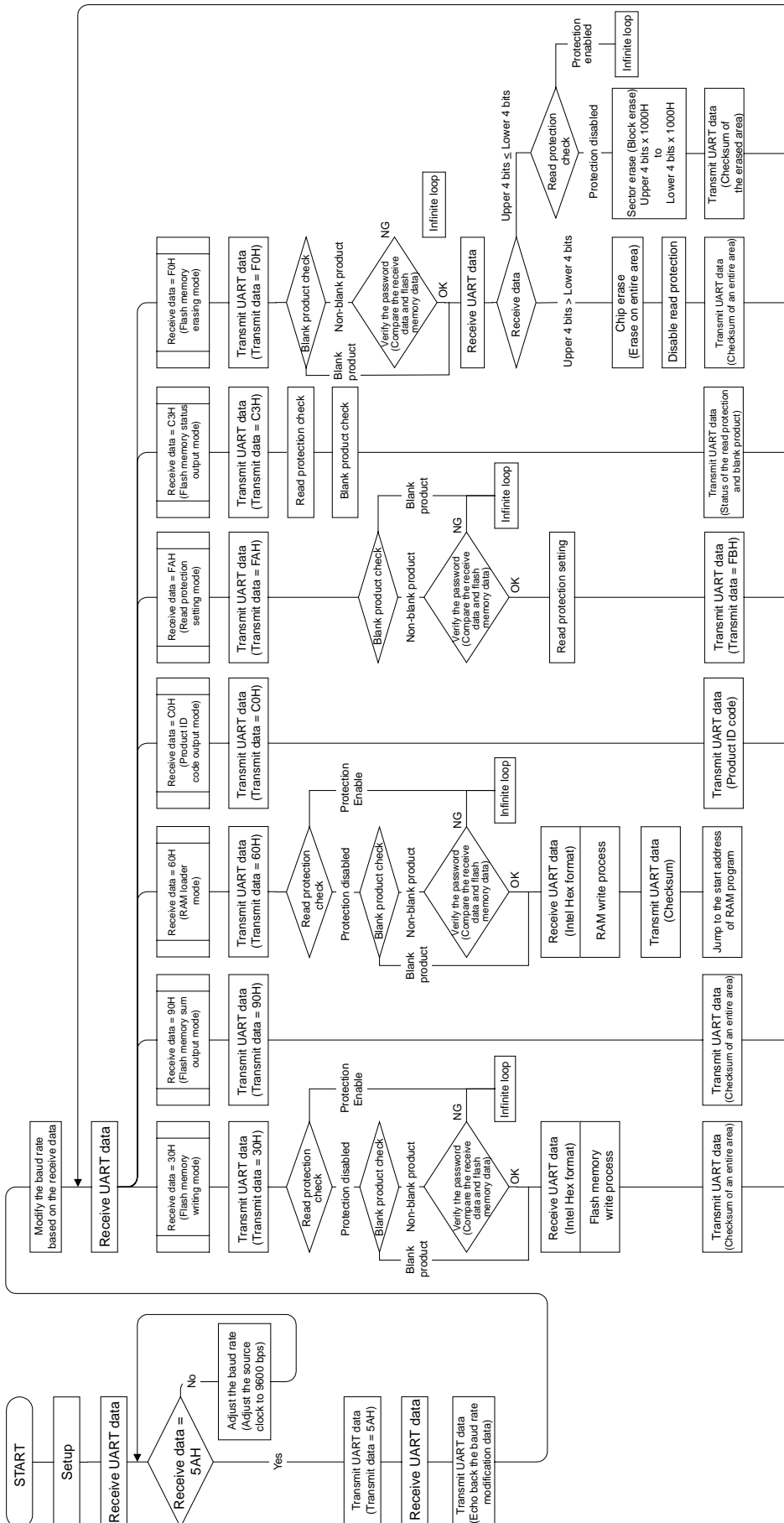


PIN	Port input control in the serial PROM mode	0 : Invalid port inputs (The input level is fixed with a hardware feature.) 1 : Valid port inputs	R/W
-----	--	--	-----

Note 1: The SPCR register can be read or written only in the serial PROM mode. When the write instruction is executed to the SPCR register in the MCU mode, the port input control can not be performed. When the read instruction is executed for the SPCR register in the MCU mode, read data of bit7 to 1 are unstable.

Note 2: All I/O ports except P45 and P44 ports are controlled by the SPCR register.

20.15Flowchart



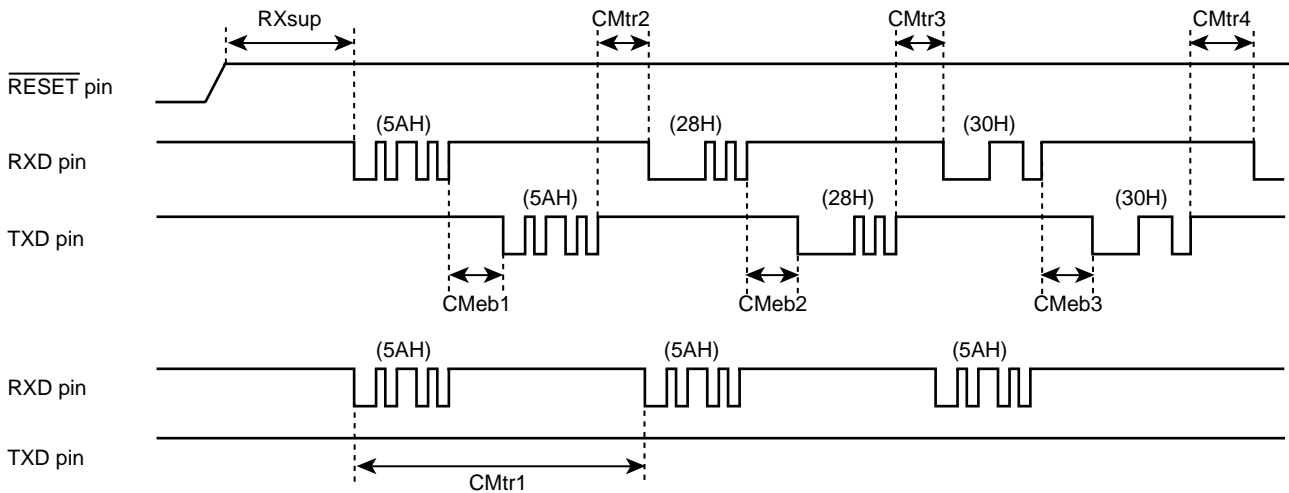
20.16UART Timing

Table 20-19 UART Timing-1 (VDD = 4.5 to 5.5 V, fc = 2 to 16 MHz, Topr = -10 to 40°C)

Parameter	Symbol	Clock Frequency (fc)	Minimum Required Time	
			At fc = 2 MHz	At fc = 16 MHz
Time from matching data reception to the echo back	CMeb1	Approx. 930	465 μs	58.1 μs
Time from baud rate modification data reception to the echo back	CMeb2	Approx. 980	490 μs	61.3 μs
Time from operation command reception to the echo back	CMeb3	Approx. 800	400 μs	50 μs
Checksum calculation time	CKsm	Approx. 7864500	3.93 s	491.5 μs
Erase time of an entire flash memory	CEall	-	30 ms	30 ms
Erase time for a sector of a flash memory (in 4-kbyte units)	CEsec	-	15 ms	15 ms

Table 20-20 UART Timing-2 (VDD = 4.5 to 5.5 V, fc = 2 to 16 MHz, Topr = -10 to 40°C)

Parameter	Symbol	Clock Frequency (fc)	Minimum Required Time	
			At fc = 2 MHz	At fc = 16 MHz
Time from the reset release to the acceptance of start bit of RXD pin	RXsup	2100	1.05 ms	131.3 ms
Matching data transmission interval	CMtr1	28500	14.2 ms	1.78 ms
Time from the echo back of matching data to the acceptance of baud rate modification data	CMtr2	380	190 μs	23.8 μs
Time from the echo back of baud rate modification data to the acceptance of an operation command	CMtr3	650	325 μs	40.6 μs
Time from the echo back of operation command to the acceptance of password count storage addresses (Upper byte)	CMtr4	800	400 μs	50 μs



21. Input/Output Circuitry

21.1 Control Pins

The input/output circuitries of the TMP86FS64FG control pins are shown below.

Control Pin	I/O	Input/Output Circuitry	Remarks
XIN XOUT	Input Output		Resonator connecting pins (high-frequency) $R_f = 1.2\text{ M}\Omega$ (typ.) $R_o = 1.5\text{ k}\Omega$ (typ.)
XTIN XTOUT	Input Output		Resonator connecting pins (Low-frequency) $R_f = 6\text{ M}\Omega$ (typ.) $R_o = 220\text{ k}\Omega$ (typ.)
$\overline{\text{RESET}}$	Input		Hysteresis input Pull-up resistor $R_{IN} = 220\text{ k}\Omega$ (typ.) $R = 1\text{ k}\Omega$ (typ.)
TEST	Input		Without pull-down resistor $R = 1\text{ k}\Omega$ (typ.) Fix the TEST pin at low-level in MCU mode.

Note: The TEST pin of the TMP86PS64 does not have a pull-down resistor. Fix the TEST pin at low-level in MCU mode.

21.2 Input/Output Ports

Port	I/O	Input/Output Circuitry	Remarks
P0	I/O	<p>Initial "High-Z"</p> <p>Data output</p> <p>Disable</p> <p>Pin input</p> <p>VDD</p> <p>R</p>	Tri-state I/O R = 100 Ω (typ.)
P1 P3 P5 P8 P9	I/O	<p>Initial "High-Z"</p> <p>Data output</p> <p>Disable</p> <p>Pin input</p> <p>VDD</p> <p>R</p>	Tri-state I/O Hysteresis input High current output (N-ch)(P5, P9) R = 100 Ω (typ.)
P6 P7	I/O	<p>Initial "High-Z"</p> <p>Data output</p> <p>Disable</p> <p>Pin input</p> <p>VDD</p> <p>R</p> <p>R_{IN}</p>	Tri-state I/O Programmable pull-up R _{IN} = 80 k Ω (typ.) R = 100 Ω (typ.)
PA PB	I/O	<p>Initial "High-Z"</p> <p>Data output</p> <p>Disable</p> <p>Pin input</p> <p>VDD</p> <p>R</p> <p>R_{IN}</p>	Tri-state I/O Hysteresis input Programmable pull-up R _{IN} = 80 k Ω (typ.) R = 100 Ω (typ.)
P2	I/O	<p>Initial "High-Z"</p> <p>Data output</p> <p>Pin input</p> <p>VDD</p> <p>R</p>	Sink open drain output Hysteresis input R = 100 Ω (typ.)
P4	I/O	<p>Initial "High-Z"</p> <p>Pch control</p> <p>Data output</p> <p>Data input</p> <p>Disable</p> <p>Pin input</p> <p>VDD</p> <p>R</p>	Sink open drain I/O or Tri-state I/O Hysteresis input R = 100 Ω (typ.)

22. Electrical Characteristics

22.1 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.

(VSS = 0 V)

Parameter	Symbol	Pins	Ratings	Unit
Supply voltage	V_{DD}		-0.3 to 6.5	V
Input voltage	V_{IN}		-0.3 to $V_{DD} + 0.3$	V
Output voltage	V_{OUT}		-0.3 to $V_{DD} + 0.3$	V
Output current (Per 1 pin)	I_{OUTH}	Except Nch Open Drain Port	-3.2	mA
	I_{OUT1}	Except P5 and P9 ports	3.2	
	I_{OUT2}	P5 and P9 ports	30	
Output current (Total)	ΣI_{OUT1}	Except P5 and P9 ports	60	
	ΣI_{OUT2}	P5 and P9 ports	80	
Power dissipation [$T_{opr} = 85\text{ }^{\circ}\text{C}$]	P_D		250	mW
Soldering temperature (time)	T_{sld}		260 (10 s)	$^{\circ}\text{C}$
Storage temperature	T_{stg}		-55 to 125	
Operating temperature	T_{opr}		-40 to 85	

22.2 Recommended Operating Conditions

The recommended operating conditions for a device are operating conditions under which it can be guaranteed that the device will operate as specified. If the device is used under operating conditions other than the recommended 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 recommended operating conditions for the device are always adhered to.

22.2.1 MCU mode (Flash Programming or erasing)

(V_{SS} = 0 V, T_{opr} = -10 to 40°C)

Parameter	Symbol	Pins	Ratings	Min	Max	Unit
Supply voltage	V _{DD}		NORMAL1, 2 modes	4.5	5.5	V
Input high level	V _{IH1}	Except hysteresis input	V _{DD} ≥ 4.5 V	V _{DD} × 0.70	V _{DD}	
	V _{IH2}	Hysteresis input		V _{DD} × 0.75		
Input low level	V _{IL1}	Except hysteresis input	V _{DD} ≥ 4.5 V	0	V _{DD} × 0.30	
	V _{IL2}	Hysteresis input			V _{DD} × 0.25	
Clock frequency	f _c	XIN, XOUT		1.0	16.0	

22.2.2 MCU mode (Except Flash Programming or erasing)

(V_{SS} = 0 V, T_{opr} = -40 to 85°C)

Parameter	Symbol	Pins	Ratings	Min	Max	Unit
Supply voltage (Condition 1)	V _{DD}		f _c = 16 MHz	4.5	5.5	V
			NORMAL1, 2 modes IDLE0, 1, 2 modes			
			f _c = 8 MHz	3.0		
			NORMAL1, 2 modes IDLE0, 1, 2 modes			
fs = 32.768 KHz	SLOW1, 2 modes SLEEP0, 1, 2 modes					
STOP mode						
Supply voltage (Condition 2) (Note)	V _{DD}		f _c = 8 MHz	2.7	3.0	V
			NORMAL1, 2 modes IDLE0, 1, 2 modes			
			fs = 32.768 KHz	SLOW1, 2 modes SLEEP0, 1, 2 modes		
STOP mode						
Input high level	V _{IH1}	Except hysteresis input	V _{DD} ≥ 4.5 V	V _{DD} × 0.70	V _{DD}	V
	V _{IH2}	Hysteresis input		V _{DD} × 0.75		
	V _{IH3}			V _{DD} < 4.5 V		
Input low level	V _{IL1}	Except hysteresis input	V _{DD} ≥ 4.5 V	0	V _{DD} × 0.30	
	V _{IL2}	Hysteresis input			V _{DD} × 0.25	
	V _{IL3}				V _{DD} < 4.5 V	
Clock frequency	f _c	XIN, XOUT	V _{DD} = 2.7 to 3.0V (Note)	1.0	8.0	MHz
			V _{DD} = 3.0 to 5.5V			
			V _{DD} = 4.5 to 5.5V		16.0	
	fs	XTIN, XTOUT	V _{DD} = 2.7 to 3.0V (Note)	30.0	34.0	
V _{DD} = 3.0 to 5.5V						

Note: When the supply voltage V_{DD} is less than 3.0 V, the operating temperature (T_{opr}) must be in a range of -20°C to 85°C.

22.2.3 Serial PROM mode

(V_{SS} = 0 V, T_{opr} = -10 to 40 °C)

Parameter	Symbol	Pins	Condition	Min	Max	Unit
Supply voltage	V _{DD}		NORMAL 1, 2 modes	4.5	5.5	V
Input high voltage	V _{IH1}	Except hysteresis input	V _{DD} ≥ 4.5 V	V _{DD} × 0.70	V _{DD}	
	V _{IH2}	Hysteresis input		V _{DD} × 0.75		
Input low voltage	V _{IL1}	Except hysteresis input	V _{DD} ≥ 4.5 V	0	V _{DD} × 0.30	
	V _{IL2}	Hysteresis input		V _{DD} × 0.25		
Clock frequency	f _c	XIN, XOUT		2.0	16.0	MHz

22.3 DC Characteristics

($V_{SS} = 0\text{ V}$, $T_{opr} = -40\text{ to }85\text{ }^{\circ}\text{C}$)

Parameter	Symbol	Pins	Condition	Min	Typ.	Max	Unit
Hysteresis voltage	V_{HS}	Hysteresis input		–	0.9	–	V
Input current	I_{IN1}	TEST	$V_{DD} = 5.5\text{ V}$, $V_{IN} = 5.5\text{ V}/0\text{ V}$	–	–	± 2	μA
	I_{IN2}	Sink open drain, tri-state port					
	I_{IN3}	$\overline{\text{RESET}}$, $\overline{\text{STOP}}$					
Input resistance	R_{IN2}	$\overline{\text{RESET}}$ pull-up	$V_{DD} = 5.5\text{ V}$, $V_{IN} = 0\text{ V}$	100	220	450	$\text{k}\Omega$
	R_{IN3}	Programmable pull-up (P6, P7, PA, PB ports)	$V_{DD} = 5.5\text{ V}$	40	80	200	
Feedback resistance	R_{fx}	XIN-XOUT		–	1.2	–	$\text{M}\Omega$
	R_{ftx}	XTIN-XTOUT		–	6	–	
Output leakage current	I_{LO1}	Sink open drain port	$V_{DD} = 5.5\text{ V}$, $V_{OUT} = 5.5\text{ V}$	–	–	2	μA
	I_{LO2}	Tri-state port	$V_{DD} = 5.5\text{ V}$, $V_{OUT} = 5.5\text{ V}/0\text{ V}$	–	–	± 2	
Output high voltage	V_{OH}	Tri-state port	$V_{DD} = 4.5\text{ V}$, $I_{OH} = -0.7\text{ mA}$	4.1	–	–	V
Output low voltage	V_{OL}	Except XOUT, P5, P9	$V_{DD} = 4.5\text{ V}$, $I_{OL} = 1.6\text{ mA}$	–	–	0.4	
Output low current	I_{OL}	High current port (P5, P9 Port)	$V_{DD} = 4.5\text{ V}$, $V_{OL} = 1.0\text{ V}$	–	20	–	mA
Supply current in NORMAL1, 2 modes	I_{DD}		$V_{DD} = 5.5\text{ V}$ $V_{IN} = 5.3\text{ V}/0.2\text{ V}$ $f_c = 16\text{ MHz}$ $f_s = 32.768\text{ kHz}$	–	13	16	mA
Supply current in IDLE 0, 1, 2 modes							
Supply current in SLOW1 mode			$V_{DD} = 3.0\text{ V}$ $V_{IN} = 2.8\text{ V}/0.2\text{ V}$ $f_s = 32.768\text{ kHz}$	–	40	260	μA
Supply current in SLEEP1 mode							
Supply current in SLEEPS1 mode			$V_{DD} = 5.5\text{ V}$ $V_{IN} = 5.3\text{ V}/0.2\text{ V}$	–	12	24	μA
Supply current in SLEEPS0 mode							
Supply current in STOP mode			–	–	0.5	10	
Peak current for SLOW1 mode (Note5,6)	I_{DDP-P}		$V_{DD} = 5.5\text{ V}$	–	10	–	mA
			$V_{DD} = 3.0\text{ V}$	–	2	–	

Note 1: Typical values show those at $T_{opr} = 25^{\circ}\text{C}$ and $V_{DD} = 5\text{ V}$.

Note 2: Input current (I_{IN3}): The current through pull-up resistor is not included.

Note 3: I_{DD} does not include I_{REF} .

Note 4: The supply currents of SLOW2 and SLEEP2 modes are equivalent to those of IDLE0, IDLE1 and IDLE2 modes.

Note 5: When a program is executing in the flash memory or when data is being read from the flash memory, the flash memory operates in an intermittent manner, causing peak currents in the operation current, as shown in Figure 22-1. In this case, the supply current I_{DD} (in NORMAL1, NORMAL2 and SLOW1 modes) is defined as the sum of the average peak current and MCU current.

Note 6: When designing the power supply, make sure that peak currents can be supplied. In SLOW1 mode, the difference between the peak current and the average current becomes large.

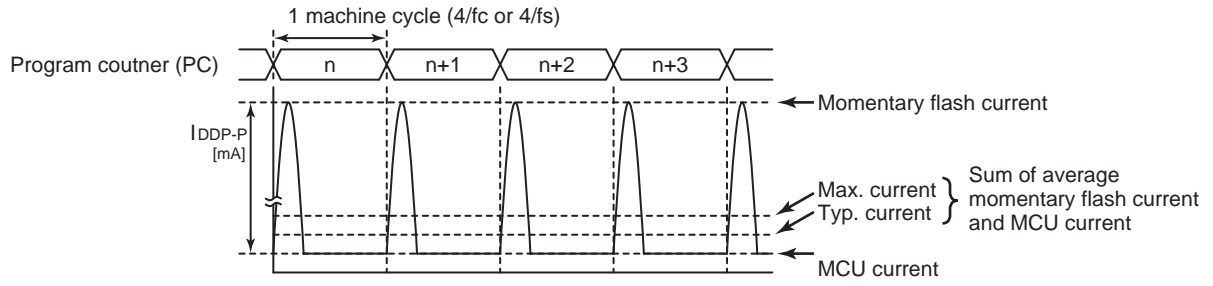


Figure 22-1 Intermittent Operation of Flash Memory

22.4 AD Characteristics

($V_{SS} = 0.0\text{ V}$, $4.5\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $T_{opr} = -40\text{ to }85\text{ }^\circ\text{C}$)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Analog reference voltage	V_{AREF}		$A_{VDD} - 1.0$	–	A_{VDD}	V
Power supply voltage of analog control circuit	A_{VDD}		V_{DD}			
	A_{VSS}		V_{SS}			
Analog reference voltage range (Note 4)	ΔV_{AREF}		3.5	–	–	
Analog input voltage	V_{AIN}		V_{SS}	–	V_{AREF}	
Power supply current of analog reference voltage	I_{REF}	$V_{DD} = A_{VDD} = V_{AREF} = 5.5\text{ V}$ $V_{SS} = A_{VSS} = 0.0\text{ V}$	–	0.6	1.0	mA
Non linearity error		$V_{DD} = A_{VDD} = 5.0\text{ V}$, $V_{SS} = A_{VSS} = 0.0\text{ V}$ $V_{AREF} = 5.0\text{ V}$	–	–	± 2	LSB
Zero point error			–	–	± 2	
Full scale error			–	–	± 2	
Total error			–	–	± 4	

($V_{SS} = 0\text{ V}$, $3.0\text{ V} \leq V_{DD} < 4.5\text{ V}$, $T_{opr} = -40\text{ to }85\text{ }^\circ\text{C}$)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Analog reference voltage	V_{AREF}		$A_{VDD} - 1.0$	–	A_{VDD}	V
Power supply voltage of analog control circuit	A_{VDD}		V_{DD}			
	A_{VSS}		V_{SS}			
Analog reference voltage range (Note 4)	ΔV_{AREF}		2.5	–	–	
Analog input voltage	V_{AIN}		V_{SS}	–	V_{AREF}	
Power supply current of analog reference voltage	I_{REF}	$V_{DD} = A_{VDD} = V_{AREF} = 4.5\text{ V}$ $V_{SS} = A_{VSS} = 0.0\text{ V}$	–	0.5	0.8	mA
Non linearity error		$V_{DD} = A_{VDD} = 3.0\text{ V}$ $V_{SS} = A_{VSS} = 0.0\text{ V}$ $V_{AREF} = 3.0\text{ V}$	–	–	± 2	LSB
Zero point error			–	–	± 2	
Full scale error			–	–	± 2	
Total error			–	–	± 4	

($V_{SS} = 0\text{ V}$, $2.7\text{ V} \leq V_{DD} < 3.0\text{ V}$, $T_{opr} = -20\text{ to }85\text{ }^\circ\text{C}$) (Note6)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Analog reference voltage	V_{AREF}		$A_{VDD} - 0.5$	–	A_{VDD}	V
Power supply voltage of analog control circuit	A_{VDD}		V_{DD}			
	A_{VSS}		V_{SS}			
Analog reference voltage range (Note 4)	ΔV_{AREF}		2.5	–	–	
Analog input voltage	V_{AIN}		V_{SS}	–	V_{AREF}	
Power supply current of analog reference voltage	I_{REF}	$V_{DD} = A_{VDD} = V_{AREF} = 2.7\text{ V}$ $V_{SS} = A_{VSS} = 0.0\text{ V}$	–	0.3	0.5	mA
Non linearity error		$V_{DD} = A_{VDD} = 2.7\text{ V}$ $V_{SS} = A_{VSS} = 0.0\text{ V}$ $V_{AREF} = 2.7\text{ V}$	–	–	± 2	LSB
Zero point error			–	–	± 2	
Full scale error			–	–	± 2	
Total error			–	–	± 4	

Note 1: The total error includes all errors except a quantization error, and is defined as a maximum deviation from the ideal conversion line.

Note 2: Conversion time is different in recommended value by power supply voltage.

Note 3: The voltage to be input on the AIN input pin must not exceed the range between V_{AREF} and V_{SS} . If a voltage outside this range is input, conversion values will become unstable and conversion values of other channels will also be affected.

Note 4: Analog reference voltage range: $\Delta V_{AREF} = V_{AREF} - V_{SS}$

Note 5: When AD converter is not used, fix the A_{VDD} and V_{AREF} pin on the V_{DD} level.

Note 6: When the supply voltage V_{DD} is less than 3.0 V, the operating temperature T_{opr} must be in a range of -20°C to 85°C.



22.5 AC Characteristics

($V_{SS} = 0\text{ V}$, $4.5\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $T_{opr} = -40\text{ to }85^\circ\text{C}$)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Machine cycle time	t_{cy}	NORMAL1, 2 modes	0.25	-	4	μs
		IDLE0, 1, 2 modes				
		SLOW1, 2 modes	117.6	-	133.3	
		SLEEP0, 1, 2 modes				
High-level clock pulse width	t_{WCH}	For external clock operation (XIN input) $f_c = 16\text{ MHz}$	-	31.25	-	ns
Low-level clock pulse width	t_{WCL}					
High-level clock pulse width	t_{WSH}	For external clock operation (XTIN input) $f_s = 32.768\text{ kHz}$	-	15.26	-	μs
Low-level clock pulse width	t_{WSL}					

($V_{SS} = 0\text{ V}$, $3.0\text{ V} \leq V_{DD} < 4.5\text{ V}$, $T_{opr} = -40\text{ to }85^\circ\text{C}$)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Machine cycle time	t_{cy}	NORMAL1, 2 modes	0.5	-	4	μs
		IDLE0, 1, 2 modes				
		SLOW1, 2 modes	117.6	-	133.3	
		SLEEP0, 1, 2 modes				
High-level clock pulse width	t_{WCH}	For external clock operation (XIN input) $f_c = 8\text{ MHz}$	-	62.5	-	ns
Low-level clock pulse width	t_{WCL}					
High-level clock pulse width	t_{WSH}	For external clock operation (XTIN input) $f_s = 32.768\text{ kHz}$	-	15.26	-	μs
Low-level clock pulse width	t_{WSL}					

($V_{SS} = 0\text{ V}$, $2.7\text{ V} \leq V_{DD} < 3.0\text{ V}$, $T_{opr} = -20\text{ to }85^\circ\text{C}$) (Note1)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Machine cycle time	t_{cy}	NORMAL1, 2 modes	0.5	-	4	μs
		IDLE0, 1, 2 modes				
		SLOW1, 2 modes	117.6	-	133.3	
		SLEEP0, 1, 2 modes				
High-level clock pulse width	t_{WCH}	For external clock operation (XIN input) $f_c = 8\text{ MHz}$	-	62.5	-	ns
Low-level clock pulse width	t_{WCL}					
High-level clock pulse width	t_{WSH}	For external clock operation (XTIN input) $f_s = 32.768\text{ kHz}$	-	15.26	-	μs
Low-level clock pulse width	t_{WSL}					

Note 1: When the supply voltage V_{DD} is less than 3.0 V, the operating temperature T_{opr} must be in a range of -20°C to 85°C .

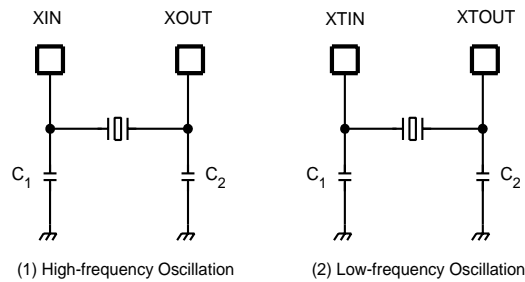
22.6 Flash Characteristics

22.6.1 Write/Retention Characteristics

($V_{SS} = 0\text{ V}$)

Parameter	Condition	Min	Typ.	Max.	Unit
Number of guaranteed writes to flash memory	$V_{SS} = 0\text{ V}$, $T_{opr} = -10\text{ to }40^\circ\text{C}$	-	-	100	Times

22.7 Recommended Oscillating Conditions



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: When using the device (oscillator) in places exposed to high electric fields such as cathode-ray tubes, we recommend electrically shielding the package in order to maintain normal operating condition.

Note 3: The product numbers and specifications of the resonators by Murata Manufacturing Co., Ltd. are subject to change. For up-to-date information, please refer to the following URL:
<http://www.murata.com>

22.8 Handling Precaution

- The solderability test conditions for lead-free products (indicated by the suffix G in product name) are shown below.
 1. When using the Sn-37Pb solder bath
 - Solder bath temperature = 230 °C
 - Dipping time = 5 seconds
 - Number of times = once
 - R-type flux used
 2. When using the Sn-3.0Ag-0.5Cu solder bath
 - Solder bath temperature = 245 °C
 - Dipping time = 5 seconds
 - Number of times = once
 - R-type flux used

Note: The pass criterion of the above test is as follows:

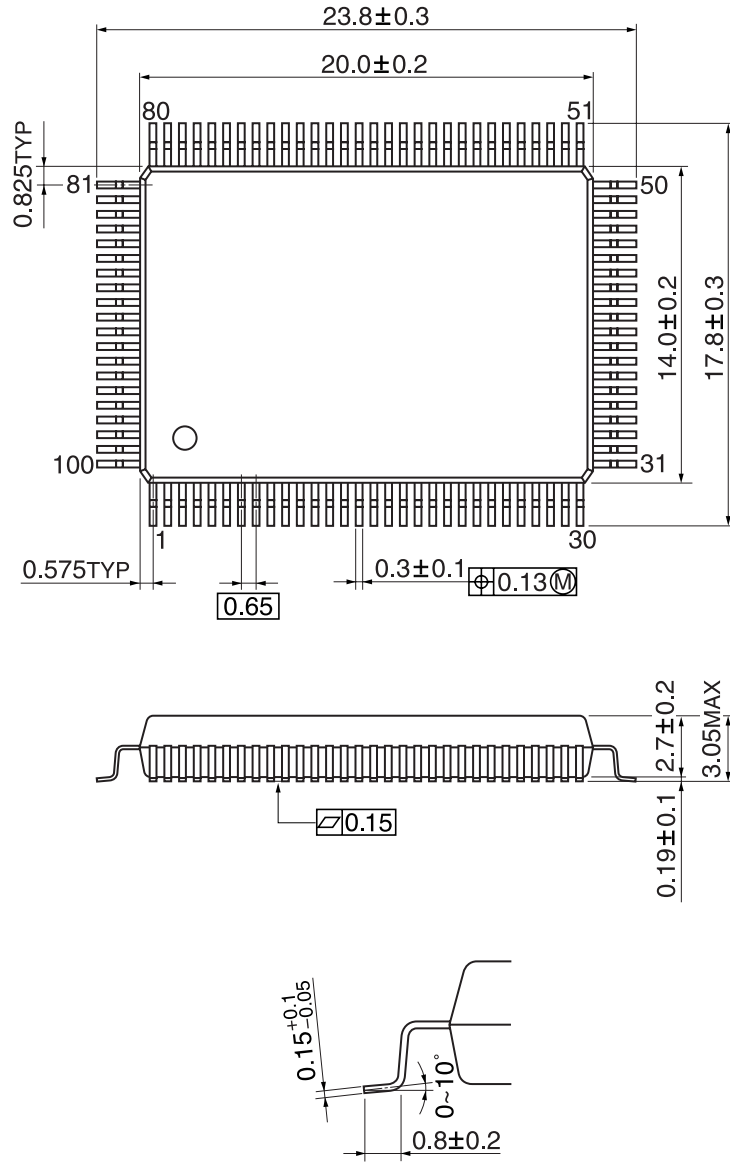
Solderability rate until forming $\geq 95\%$

- When using the device (oscillator) in places exposed to high electric fields such as cathode-ray tubes, we recommend electrically shielding the package in order to maintain normal operating condition.

23. Package Dimension

P-QFP100-1420-0.65A

Unit: mm



This is a technical document that describes the operating functions and electrical specifications of the 8-bit microcontroller series TLCS-870/C (LSI).

Toshiba provides a variety of development tools and basic software to enable efficient software development.

These development tools have specifications that support advances in microcomputer hardware (LSI) and can be used extensively. Both the hardware and software are supported continuously with version updates.

The recent advances in CMOS LSI production technology have been phenomenal and microcomputer systems for LSI design are constantly being improved. The products described in this document may also be revised in the future. Be sure to check the latest specifications before using.

Toshiba is developing highly integrated, high-performance microcomputers using advanced MOS production technology and especially well proven CMOS technology.

We are prepared to meet the requests for custom packaging for a variety of application areas.

We are confident that our products can satisfy your application needs now and in the future.

