

**TOSHIBA**

8 Bit Microcontroller  
TLCS-870/C Series

**TMP86FS23UG**

**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	
2005/9/12	1	First Release
2005/12/8	2	Contents Revised
2006/8/28	3	Contents Revised



# Table of Contents

---

---

## TMP86FS23UG

---

---

1.1	Features	1
1.2	Pin Assignment	3
1.3	Block Diagram	4
1.4	Pin Names and Functions	5

---

---

## 2. Operational Description

---

---

2.1	CPU Core Functions	9
2.1.1	Memory Address Map	9
2.1.2	Program Memory (Flash)	9
2.1.3	Data Memory (RAM)	9
2.2	System Clock Controller	10
2.2.1	Clock Generator	10
2.2.2	Timing Generator	12
2.2.2.1	Configuration of timing generator	
2.2.2.2	Machine cycle	
2.2.3	Operation Mode Control Circuit	13
2.2.3.1	Single-clock mode	
2.2.3.2	Dual-clock mode	
2.2.3.3	STOP mode	
2.2.4	Operating Mode Control	18
2.2.4.1	STOP mode	
2.2.4.2	IDLE1/2 mode and SLEEP1/2 mode	
2.2.4.3	IDLE0 and SLEEP0 modes (IDLE0, SLEEP0)	
2.2.4.4	SLOW mode	
2.3	Reset Circuit	31
2.3.1	External Reset Input	31
2.3.2	Address trap reset	32
2.3.3	Watchdog timer reset	32
2.3.4	System clock reset	32

---

---

## 3. Interrupt Control Circuit

---

---

3.1	Interrupt latches (IL19 to IL2)	35
3.2	Interrupt enable register (EIR)	36
3.2.1	Interrupt master enable flag (IMF)	36
3.2.2	Individual interrupt enable flags (EF19 to EF4)	37
Note 3:		38
3.3	Interrupt Sequence	39
3.3.1	Interrupt acceptance processing is packaged as follows	39
3.3.2	Saving/restoring general-purpose registers	40
3.3.2.1	Using PUSH and POP instructions	
3.3.2.2	Using data transfer instructions	
3.3.3	Interrupt return	41
3.4	Software Interrupt (INTSW)	42
3.4.1	Address error detection	42
3.4.2	Debugging	42

3.5	Undefined Instruction Interrupt (INTUNDEF) . . . . .	42
3.6	Address Trap Interrupt (INTATRAP) . . . . .	42
3.7	External Interrupts . . . . .	43

---

#### 4. Special Function Register (SFR)

---

4.1	SFR . . . . .	45
4.2	DBR . . . . .	47

---

#### 5. I/O Ports

---

5.1	Port P1 (P17 to P10) . . . . .	52
5.2	Port P2 (P22 to P20) . . . . .	54
5.3	Port P3 (P37 to P30) . . . . .	55
5.4	Port P5 (P57 to P50) . . . . .	57
5.5	Port P6 (P67 to P60) . . . . .	59
5.6	Port P7 (P77 to P70) . . . . .	62
5.7	Port P8 (P87 to P80) . . . . .	64

---

#### 6. Time Base Timer (TBT)

---

6.1	Time Base Timer . . . . .	67
6.1.1	Configuration . . . . .	67
6.1.2	Control . . . . .	67
6.1.3	Function . . . . .	68
6.2	Divider Output (DVO) . . . . .	69
6.2.1	Configuration . . . . .	69
6.2.2	Control . . . . .	69

---

#### 7. Watchdog Timer (WDT)

---

7.1	Watchdog Timer Configuration . . . . .	71
7.2	Watchdog Timer Control . . . . .	72
7.2.1	Malfunction Detection Methods Using the Watchdog Timer . . . . .	72
7.2.2	Watchdog Timer Enable . . . . .	73
7.2.3	Watchdog Timer Disable . . . . .	74
7.2.4	Watchdog Timer Interrupt (INTWDT) . . . . .	74
7.2.5	Watchdog Timer Reset . . . . .	75
7.3	Address Trap . . . . .	76
7.3.1	Selection of Address Trap in Internal RAM (ATAS) . . . . .	76
7.3.2	Selection of Operation at Address Trap (ATOUT) . . . . .	76
7.3.3	Address Trap Interrupt (INTATRAP) . . . . .	76
7.3.4	Address Trap Reset . . . . .	77

---

#### 8. 18-Bit Timer/Counter (TC1)

---

8.1	Configuration . . . . .	79
8.2	Control . . . . .	80
8.3	Function . . . . .	83

8.3.1	Timer mode.....	83
8.3.2	Event Counter mode.....	84
8.3.3	Pulse Width Measurement mode.....	85
8.3.4	Frequency Measurement mode.....	86

---

## 9. 8-Bit TimerCounter (TC3, TC4)

---

9.1	Configuration .....	89
9.2	TimerCounter Control .....	90
9.3	Function .....	95
9.3.1	8-Bit Timer Mode (TC3 and 4).....	95
9.3.2	8-Bit Event Counter Mode (TC3, 4).....	96
9.3.3	8-Bit Programmable Divider Output (PDO) Mode (TC3, 4).....	96
9.3.4	8-Bit Pulse Width Modulation (PWM) Output Mode (TC3, 4).....	99
9.3.5	16-Bit Timer Mode (TC3 and 4).....	101
9.3.6	16-Bit Event Counter Mode (TC3 and 4).....	102
9.3.7	16-Bit Pulse Width Modulation (PWM) Output Mode (TC3 and 4).....	102
9.3.8	16-Bit Programmable Pulse Generate (PPG) Output Mode (TC3 and 4).....	105
9.3.9	Warm-Up Counter Mode.....	107
9.3.9.1	Low-Frequency Warm-up Counter Mode (NORMAL1 → NORMAL2 → SLOW2 → SLOW1)	
9.3.9.2	High-Frequency Warm-Up Counter Mode (SLOW1 → SLOW2 → NORMAL2 → NORMAL1)	

---

## 10. 8-Bit TimerCounter (TC5, TC6)

---

10.1	Configuration .....	109
10.2	TimerCounter Control .....	110
10.3	Function .....	115
10.3.1	8-Bit Timer Mode (TC5 and 6).....	115
10.3.2	8-Bit Event Counter Mode (TC5, 6).....	116
10.3.3	8-Bit Programmable Divider Output (PDO) Mode (TC5, 6).....	116
10.3.4	8-Bit Pulse Width Modulation (PWM) Output Mode (TC5, 6).....	119
10.3.5	16-Bit Timer Mode (TC5 and 6).....	121
10.3.6	16-Bit Event Counter Mode (TC5 and 6).....	122
10.3.7	16-Bit Pulse Width Modulation (PWM) Output Mode (TC5 and 6).....	122
10.3.8	16-Bit Programmable Pulse Generate (PPG) Output Mode (TC5 and 6).....	125
10.3.9	Warm-Up Counter Mode.....	127
10.3.9.1	Low-Frequency Warm-up Counter Mode (NORMAL1 → NORMAL2 → SLOW2 → SLOW1)	
10.3.9.2	High-Frequency Warm-Up Counter Mode (SLOW1 → SLOW2 → NORMAL2 → NORMAL1)	

---

## 11. Asynchronous Serial interface (UART )

---

11.1	Configuration .....	129
11.2	Control .....	130
11.3	Transfer Data Format .....	132
11.4	Transfer Rate.....	133
11.5	Data Sampling Method .....	133
11.6	STOP Bit Length .....	134
11.7	Parity .....	134
11.8	Transmit/Receive Operation .....	134
11.8.1	Data Transmit Operation .....	134
11.8.2	Data Receive Operation .....	134
11.9	Status Flag .....	135

11.9.1	Parity Error.....	135
11.9.2	Framing Error.....	135
11.9.3	Overrun Error.....	135
11.9.4	Receive Data Buffer Full.....	136
11.9.5	Transmit Data Buffer Empty.....	136
11.9.6	Transmit End Flag.....	137

---



---

## 12. Synchronous Serial Interface (SIO)

---

12.1	Configuration.....	139
12.2	Control.....	140
12.3	Serial clock.....	141
12.3.1	Clock source.....	141
12.3.1.1	Internal clock.....	
12.3.1.2	External clock.....	
12.3.2	Shift edge.....	143
12.3.2.1	Leading edge.....	
12.3.2.2	Trailing edge.....	
12.4	Number of bits to transfer.....	143
12.5	Number of words to transfer.....	143
12.6	Transfer Mode.....	144
12.6.1	4-bit and 8-bit transfer modes.....	144
12.6.2	4-bit and 8-bit receive modes.....	146
12.6.3	8-bit transfer / receive mode.....	147

---



---

## 13. 10-bit AD Converter (ADC)

---

13.1	Configuration.....	149
13.2	Register configuration.....	150
13.3	Function.....	153
13.3.1	Software Start Mode.....	153
13.3.2	Repeat Mode.....	153
13.3.3	Register Setting.....	154
13.4	STOP/SLOW Modes during AD Conversion.....	155
13.5	Analog Input Voltage and AD Conversion Result.....	156
13.6	Precautions about AD Converter.....	157
13.6.1	Restrictions for AD Conversion interrupt (INTADC) usage.....	157
13.6.2	Analog input pin voltage range.....	157
13.6.3	Analog input shared pins.....	157
13.6.4	Noise Countermeasure.....	157

---



---

## 14. Key-on Wakeup (KWU)

---

14.1	Configuration.....	159
14.2	Control.....	159
14.3	Function.....	159

---



---

## 15. LCD Driver

---

15.1	Configuration.....	161
15.2	Control.....	162
15.2.1	LCD driving methods.....	163
15.2.2	Frame frequency.....	164



15.2.3	LCD drive voltage .....	165
15.2.4	Adjusting the LCD panel drive capability .....	165
<b>15.3</b>	<b>LCD Display Operation .....</b>	<b>166</b>
15.3.1	Display data setting .....	166
15.3.2	Blanking .....	166
<b>15.4</b>	<b>Control Method of LCD Driver .....</b>	<b>167</b>
15.4.1	Initial setting .....	167
15.4.2	Store of display data .....	167
15.4.3	Example of LCD driver output .....	169

---



---

## 16. Real-Time Clock

---

16.1	Configuration .....	175
16.2	Control of the RTC .....	175
16.3	Function .....	176

---



---

## 17. Multiply-Accumulate (MAC) Unit

---

17.1	Configuration .....	177
17.2	Registers .....	177
17.2.1	Command Register .....	177
17.2.2	Status Register .....	178
17.2.3	Multiplier data Register .....	178
17.2.4	Multiplicand data Register .....	178
17.2.5	Result Register .....	178
17.2.6	Addend Register .....	178
17.3	Control .....	178
17.4	Register Description .....	180
17.4.1	EMAC .....	180
17.4.2	CMOD .....	180
17.4.3	RCLR .....	180
17.5	Arithmetic Modes .....	181
17.5.1	Unsigned Multiply Mode .....	181
17.5.2	Signed Multiply Mode .....	181
17.5.3	Unsigned Multiply-Accumulate Mode .....	181
17.5.4	Signed Multiply-Accumulate Mode .....	182
17.5.5	Valid Numerical Ranges .....	182
17.6	Status Flags .....	182
17.6.1	Operation Status Flag (CALC) .....	183
17.6.2	Overflow Flag (OVRF) .....	183
17.6.3	Carry Flag (CARF) .....	183
17.6.4	Sign Flag (SIGN) .....	183
17.6.5	Zero Flag (ZERF) .....	183
17.7	Example of Software Processing .....	183

---



---

## 18. Flash Memory

---

18.1	Flash Memory Control .....	186
18.1.1	Flash Memory Command Sequence Execution Control (FLSCR<FLSMD>) .....	186
18.1.2	Flash Memory Bank Select Control (FLSCR<BANKSEL>) .....	186
18.1.3	Flash Memory Standby Control (FLSSTB<FSTB>) .....	187
18.2	Command Sequence .....	188
18.2.1	Byte Program .....	188
18.2.2	Sector Erase (4-kbyte Erase) .....	188
18.2.3	Chip Erase (All Erase) .....	189

18.2.4	Product ID Entry .....	189
18.2.5	Product ID Exit .....	189
18.2.6	Read Protect .....	189
<b>18.3</b>	<b>Toggle Bit (D6) .....</b>	<b>190</b>
<b>18.4</b>	<b>Access to the Flash Memory Area .....</b>	<b>191</b>
18.4.1	Flash Memory Control in the Serial PROM Mode.....	191
18.4.1.1	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)	
18.4.2	Flash Memory Control in the MCU mode.....	193
18.4.2.1	How to write to the flash memory by executing a user write control program in the RAM area (in the MCU mode)	

---



---

## 19. Serial PROM Mode

---

19.1	Outline .....	195
19.2	Memory Mapping .....	195
19.3	Serial PROM Mode Setting .....	196
19.3.1	Serial PROM Mode Control Pins .....	196
19.3.2	Pin Function .....	196
19.3.3	Example Connection for On-Board Writing .....	197
19.3.4	Activating the Serial PROM Mode .....	198
19.4	Interface Specifications for UART .....	199
19.5	Operation Command .....	201
19.6	Operation Mode .....	201
19.6.1	Flash Memory Erasing Mode (Operating command: F0H) .....	203
19.6.2	Flash Memory Writing Mode (Operation command: 30H) .....	205
19.6.3	RAM Loader Mode (Operation Command: 60H) .....	208
19.6.4	Flash Memory SUM Output Mode (Operation Command: 90H) .....	210
19.6.5	Product ID Code Output Mode (Operation Command: C0H) .....	211
19.6.6	Flash Memory Status Output Mode (Operation Command: C3H) .....	213
19.6.7	Flash Memory Read Protection Setting Mode (Operation Command: FAH) .....	214
19.7	Error Code .....	216
19.8	Checksum (SUM) .....	216
19.8.1	Calculation Method .....	216
19.8.2	Calculation data .....	217
19.9	Intel Hex Format (Binary) .....	218
19.10	Passwords .....	218
19.10.1	Password String .....	219
19.10.2	Handling of Password Error .....	219
19.10.3	Password Management during Program Development .....	219
19.11	Product ID Code .....	220
19.12	Flash Memory Status Code .....	220
19.13	Specifying the Erasure Area .....	222
19.14	Port Input Control Register .....	222
19.15	Flowchart .....	224
19.16	UART Timing .....	225

---



---

## 20. Input/Output Circuitry

---

20.1	Control Pins .....	227
20.2	Input/Output Ports .....	228

---



---

## 21. Electrical Characteristics

---

21.1	Absolute Maximum Ratings .....	229
------	--------------------------------	-----

---

---

21.2	Recommended Operating Condition . . . . .	230
21.2.1	When Programming Flash memory in MCU mode . . . . .	230
21.2.2	When Not Programming Flash Memory in MCU Mode . . . . .	230
21.2.3	Serial PROM mode . . . . .	231
21.3	DC Characteristics . . . . .	232
21.4	AD Conversion Characteristics . . . . .	234
21.5	AC Characteristics . . . . .	235
21.6	Timer Counter 1 input (ECIN) Characteristics . . . . .	235
21.7	Flash Characteristics . . . . .	235
21.7.1	Write/Retention Characteristics . . . . .	235
21.8	Recommended Oscillating Conditions . . . . .	236
21.9	Handling Precaution . . . . .	236

---

---

## 22. 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  
**TMP86FS23UG**

The TMP86FS23UG is a single-chip 8-bit high-speed and high-functionality microcomputer incorporating 61440 bytes of Flash Memory. It is pin-compatible with the TMP86CM23/CP23AUG (Mask ROM version). The TMP86FS23UG can realize operations equivalent to those of the TMP86CM23/CP23AUG by programming the on-chip Flash Memory.

Product No.	ROM (FLASH)	RAM	Package	MASK ROM MCU	Emulation Chip
TMP86FS23UG	61440 bytes	2048 bytes	P-LQFP64-1010-0.50D	TMP86CM23/CP23AUG	TMP86C923XB

**1.1 Features**

1. 8-bit single chip microcomputer TLCS-870/C series
  - Instruction execution time :
    - 0.25  $\mu$ s (at 16 MHz)
    - 122  $\mu$ s (at 32.768 kHz)
  - 132 types & 731 basic instructions
2. 20interrupt sources (External : 5 Internal : 15)
3. Input / Output ports (I/O : 48 pins Output : 3 pins)
  - Large current output: 5pins (Typ. 20mA), LED direct drive
4. Prescaler
  - Time base timer
  - Divider output function
5. Watchdog Timer
6. 18-bit Timer/Counter : 1ch
  - Timer Mode
  - Event Counter Mode
  - Pulse Width Measurement Mode
  - Frequency Measurement Mode

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

- 
7. 8-bit timer counter : 4 ch
    - Timer, Event counter, Programmable divider output (PDO),  
Pulse width modulation (PWM) output,  
Programmable pulse generation (PPG) modes
  8. 8-bit UART : 1 ch
  9. 8-bit SIO: 1 ch
  10. 10-bit successive approximation type AD converter
    - Analog input: 8 ch
  11. Key-on wakeup : 4 ch
  12. LCD driver/controller
    - LCD direct drive capability (MAX 32 seg × 4 com)
    - 1/4,1/3,1/2duties or static drive are programmably selectable
  13. Multiply accumulate unit (MAC)
    - Multiply or MAC mode are selectable
    - Signed or unsigned operation are selectable
  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:
    - 3.5 V to 5.5 V at 16MHz /32.768 kHz
    - 2.7 V to 5.5 V at 8 MHz/32.768 kHz
-



### 1.3 Block Diagram

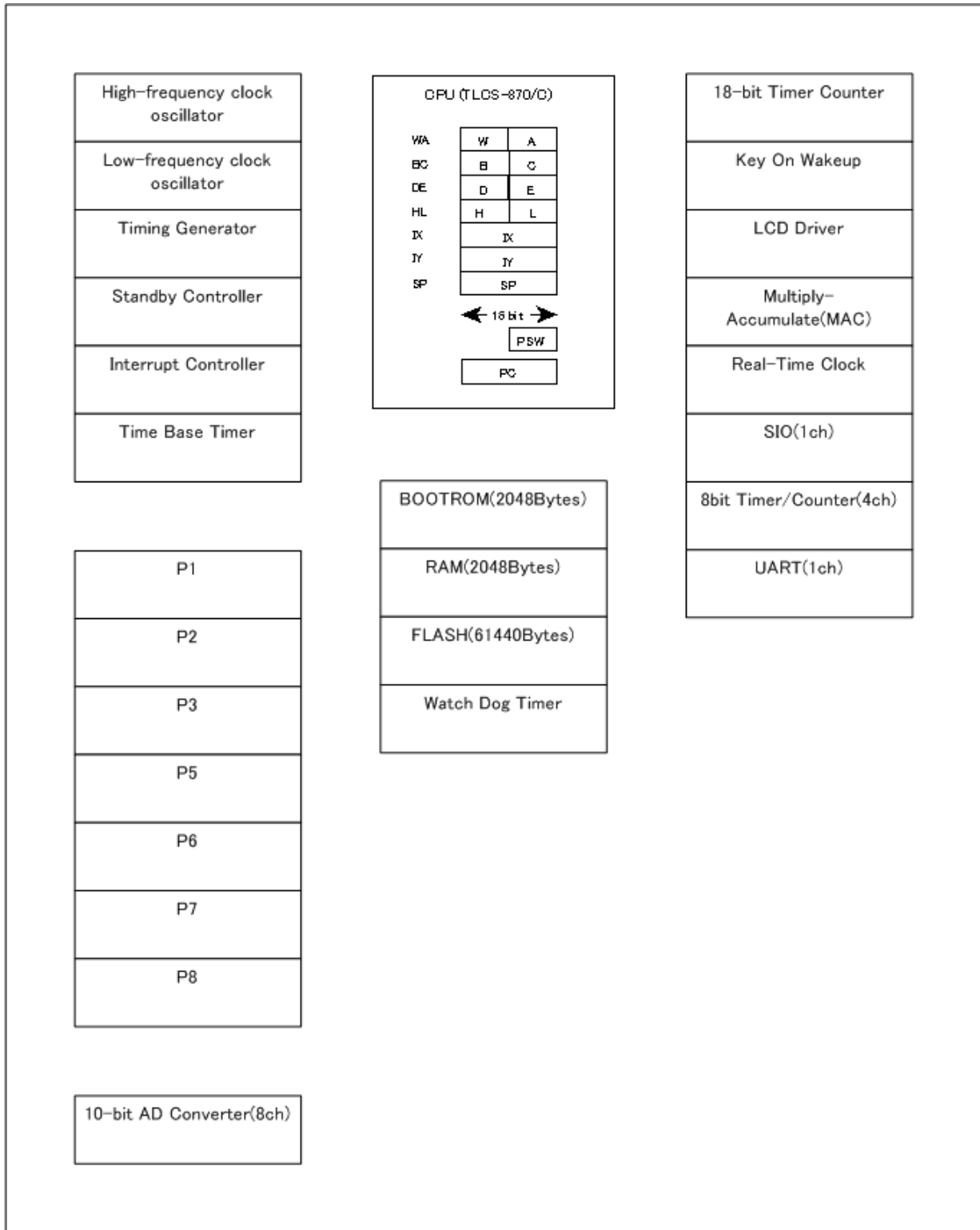


Figure 1-2 Block Diagram



## 1.4 Pin Names and Functions

The TMP86FS23UG 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(1/3)

Pin Name	Pin Number	Input/Output	Functions
P17 SEG24	27	IO O	PORT17 LCD segment output 24
P16 SEG25	26	IO O	PORT16 LCD segment output 25
P15 SEG26	25	IO O	PORT15 LCD segment output 26
P14 SEG27 INT3	24	IO O I	PORT14 LCD segment output 27 External interrupt 3 input
P13 SEG28 INT2	23	IO O I	PORT13 LCD segment output 28 External interrupt 2 input
P12 SEG29 INT1	22	IO O I	PORT12 LCD segment output 29 External interrupt 1 input
P11 SEG30 TXD	21	IO O O	PORT11 LCD segment output 30 UART data output
P10 SEG31 RXD	20	IO O I	PORT10 LCD segment output 31 UART data input
P22 XTOUT	7	IO O	PORT22 Resonator connecting pins(32.768kHz) for inputting external clock
P21 XTIN	6	IO I	PORT21 Resonator connecting pins(32.768kHz) for inputting external clock
P20 $\overline{\text{STOP}}$ INT5	9	IO I I	PORT20 STOP mode release signal input External interrupt 5 input
P37 $\overline{\text{DVO}}$	64	O O	PORT37 Divider Output
P36 $\overline{\text{PDO3/PWM3}}$	63	O O	PORT36 PDO3/PWM3 output
P35 $\overline{\text{PDO4/PWM4/PPG4}}$	62	O O	PORT35 PDO4/PWM4/PPG4 output
P34 $\overline{\text{PDO5/PWM5}}$ TC5	61	IO O I	PORT34 PDO5/PWM5 output TC5 input
P33 $\overline{\text{PDO6/PWM6/PPG6}}$ TC6	60	IO O I	PORT33 PDO6/PWM6/PPG6 output TC6 input
P32 $\overline{\text{SCK}}$	59	IO IO	PORT32 Serial Clock I/O

Table 1-1 Pin Names and Functions(2/3)

Pin Name	Pin Number	Input/Output	Functions
P31 SO TC3	58	IO O I	PORT31 Serial Data Output TC3 input
P30 SI TC4	57	IO I I	PORT30 Serial Data Input TC4 input
P57 SEG16	35	IO O	PORT57 LCD segment output 16
P56 SEG17	34	IO O	PORT56 LCD segment output 17
P55 SEG18	33	IO O	PORT55 LCD segment output 18
P54 SEG19	32	IO O	PORT54 LCD segment output 19
P53 SEG20	31	IO O	PORT53 LCD segment output 20
P52 SEG21	30	IO O	PORT52 LCD segment output 21
P51 SEG22	29	IO O	PORT51 LCD segment output 22
P50 SEG23	28	IO O	PORT50 LCD segment output 23
P67 AIN7 STOP5	19	IO I I	PORT67 Analog Input7 STOP5 input
P66 AIN6 STOP4	18	IO I I	PORT66 Analog Input6 STOP4 input
P65 AIN5 STOP3	17	IO I I	PORT65 Analog Input5 STOP3 input
P64 AIN4 STOP2	16	IO I I	PORT64 Analog Input4 STOP2 input
P63 AIN3 INT0	15	IO I I	PORT63 Analog Input3 External interrupt 0 input
P62 AIN2 ECNT	14	IO I I	PORT62 Analog Input2 ECNT input
P61 AIN1 ECIN	13	IO I I	PORT61 Analog Input1 ECIN input
P60 AIN0	12	IO I	PORT60 Analog Input0
P77 SEG8	43	IO O	PORT77 LCD segment output 8
P76 SEG9	42	IO O	PORT76 LCD segment output 9

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

Pin Name	Pin Number	Input/Output	Functions
P75 SEG10	41	IO O	PORT75 LCD segment output 10
P74 SEG11	40	IO O	PORT74 LCD segment output 11
P73 SEG12	39	IO O	PORT73 LCD segment output 12
P72 SEG13	38	IO O	PORT72 LCD segment output 13
P71 SEG14	37	IO O	PORT71 LCD segment output 14
P70 SEG15	36	IO O	PORT70 LCD segment output 15
P87 SEG0	51	IO O	PORT87 LCD segment output 0
P86 SEG1	50	IO O	PORT86 LCD segment output 1
P85 SEG2	49	IO O	PORT85 LCD segment output 2
P84 SEG3	48	IO O	PORT84 LCD segment output 3
P83 SEG4	47	IO O	PORT83 LCD segment output 4
P82 SEG5	46	IO O	PORT82 LCD segment output 5
P81 SEG6	45	IO O	PORT81 LCD segment output 6
P80 SEG7	44	IO O	PORT80 LCD segment output 7
COM3	52	O	LCD common output 3
COM2	53	O	LCD common output 2
COM1	54	O	LCD common output 1
COM0	55	O	LCD common output 0
XIN	2	I	Resonator connecting pins for high-frequency clock
XOUT	3	O	Resonator connecting pins for high-frequency clock
RESET	8	I	Reset signal
TEST	4	I	Test pin for out-going test. Normally, be fixed to low.
VAREF	11	I	Analog Base Voltage Input Pin for A/D Conversion
AVDD	10	I	Analog Power Supply
VDD	5	I	+5V
VSS	1	I	0(GND)



## 2. Operational Description

### 2.1 CPU Core Functions

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

This section provides a description of the CPU core, the program memory, the data memory, and the reset circuit.

#### 2.1.1 Memory Address Map

The TMP86FS23UG memory is composed Flash, RAM, DBR(Data buffer register) and SFR(Special function register). They are all mapped in 64-Kbyte address space. Figure 2-1 shows the TMP86FS23UG memory address map.

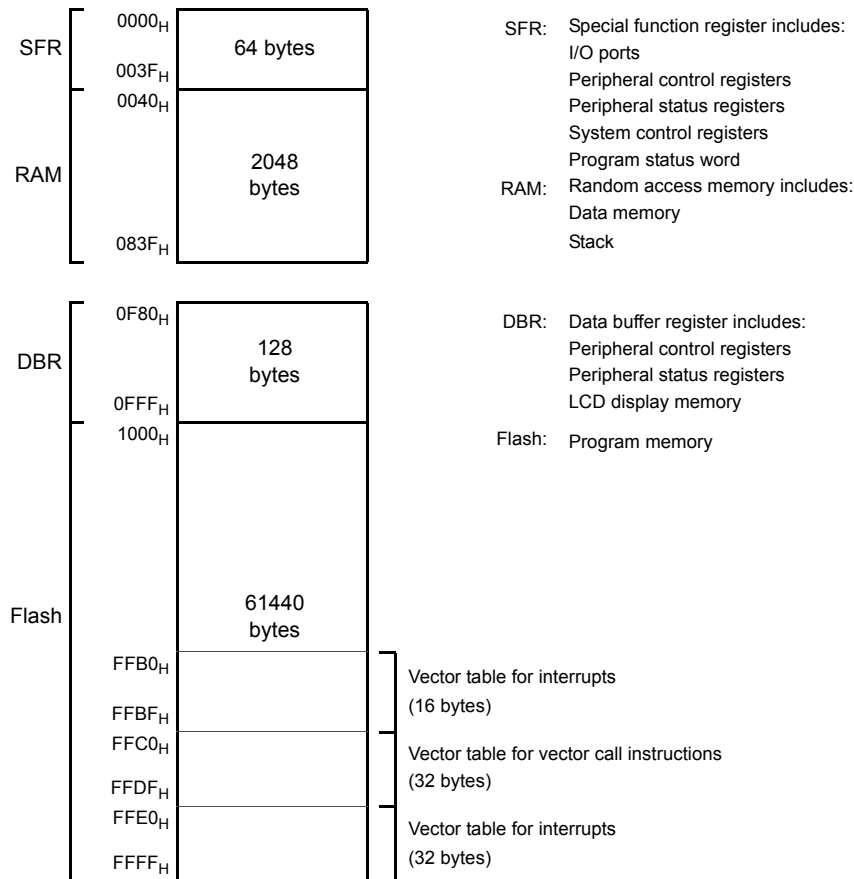


Figure 2-1 Memory Address Map

#### 2.1.2 Program Memory (Flash)

The TMP86FS23UG has a 61440 bytes (Address 1000H to FFFFH) of program memory (Flash).

#### 2.1.3 Data Memory (RAM)

The TMP86FS23UG has 2048 bytes (Address 0040H to 083FH) of internal RAM. The first 192 bytes (0040H to 00FFH) of the internal RAM are located in the direct area; instructions with shorten operations are available against such an area.

The data memory contents become unstable when the power supply is turned on; therefore, the data memory should be initialized by an initialization routine.

Example :Clears RAM to “00H”. (TMP86FS23UG)

```

LD      HL, 0040H      ; Start address setup
LD      A, H          ; Initial value (00H) setup
LD      BC, 07FFH
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 a standby controller.

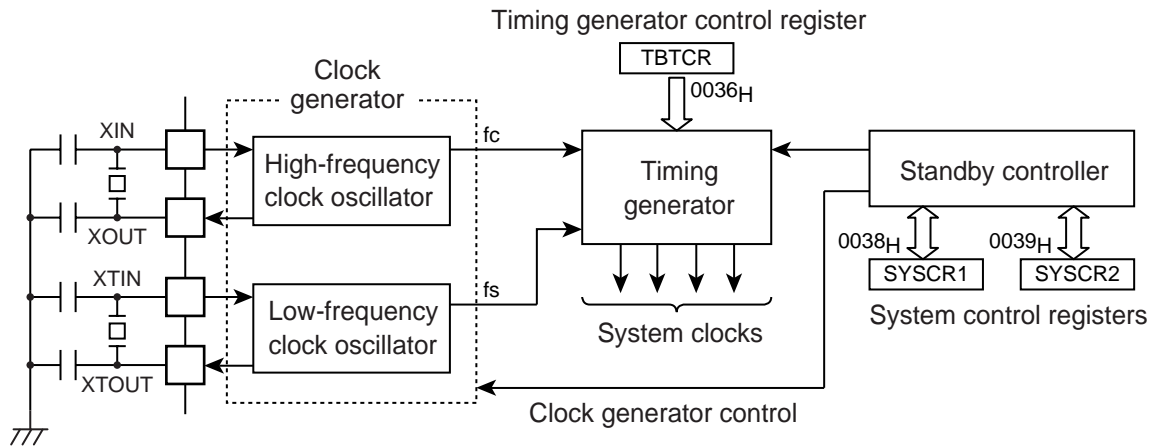


Figure 2-2 System Colck Control

### 2.2.1 Clock Generator

The clock generator generates the basic clock which provides the system clocks supplied to the CPU core and peripheral hardware. It contains two oscillation circuits: One for the high-frequency clock and one for the low-frequency clock. Power consumption can be reduced by switching of the standby controller to low-power operation based on the low-frequency clock.

The high-frequency (fc) clock and low-frequency (fs) clock can easily be obtained by connecting a resonator between the XIN/XOUT and XTIN/XTOUT pins respectively. Clock input from an external oscillator is also possible. In this case, external clock is applied to XIN/XTIN pin with XOUT/XTOUT pin not connected.

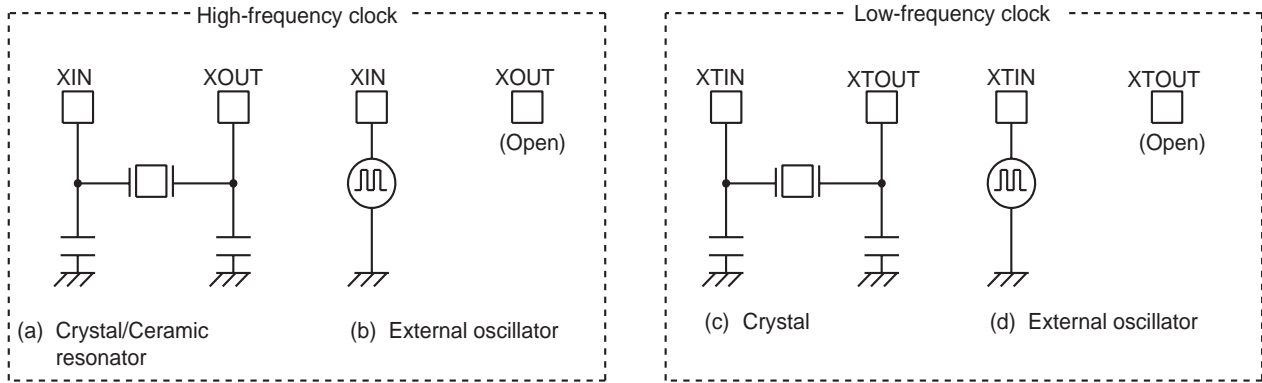


Figure 2-3 Examples of Resonator Connection

Note: The function to monitor the basic clock directly at external is not provided for hardware, however, with disabling all interrupts and watchdog timers, the oscillation frequency can be adjusted by monitoring the pulse which the fixed frequency is outputted to the port by the program. The system to require the adjustment of the oscillation frequency should create the program for the adjustment in advance.

## 2.2.2 Timing Generator

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

1. Generation of main system clock
2. Generation of divider output ( $\overline{DV0}$ ) pulses
3. Generation of source clocks for time base timer
4. Generation of source clocks for watchdog timer
5. Generation of internal source clocks for timer/counters
6. Generation of warm-up clocks for releasing STOP mode
7. LCD

### 2.2.2.1 Configuration of timing generator

The timing generator consists of a 2-stage prescaler, a 21-stage divider, a main system clock generator, and machine cycle counters.

An input clock to the 7th stage of the divider depends on the operating mode,  $SYSCR2<SYSCK>$  and  $TBTCR<DV7CK>$ , that is shown in Figure 2-4. As reset and STOP mode started/canceled, the prescaler and the divider are cleared to “0”.

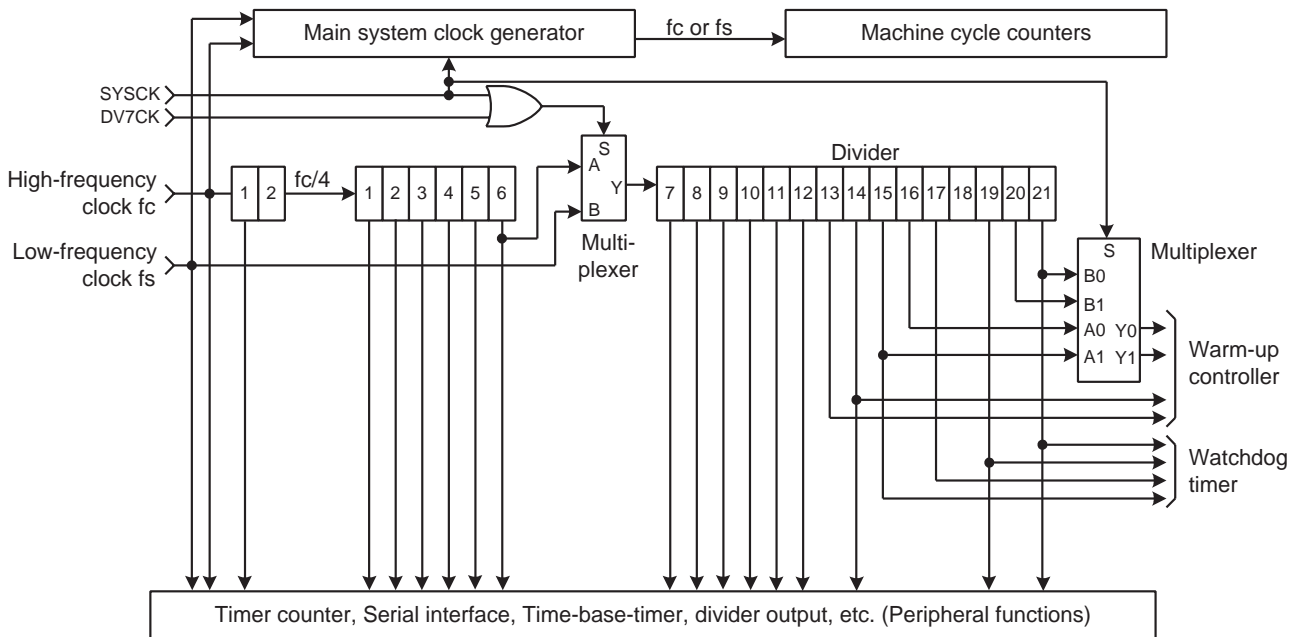
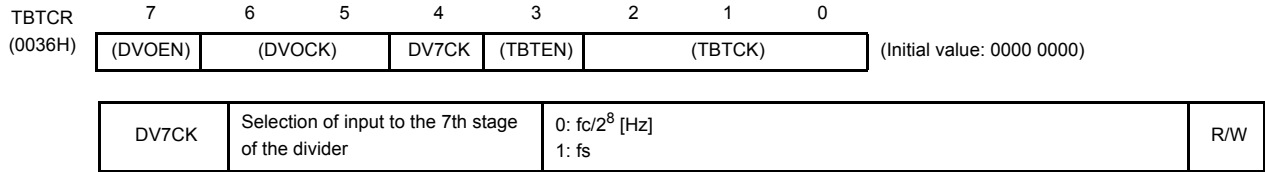


Figure 2-4 Configuration of Timing Generator



Timing Generator Control Register



- Note 1: In single clock mode, do not set DV7CK to "1".
- Note 2: Do not set "1" on DV7CK while the low-frequency clock is not operated stably.
- Note 3:  $fc$ : High-frequency clock [Hz],  $fs$ : Low-frequency clock [Hz], \*: Don't care
- Note 4: In SLOW1/2 and SLEEP1/2 modes, the DV7CK setting is ineffective, and  $fs$  is input to the 7th stage of the divider.
- Note 5: When STOP mode is entered from NORMAL1/2 mode, the DV7CK setting is ineffective during the warm-up period after release of STOP mode, and the 6th stage of the divider is input to the 7th stage during this period.

2.2.2.2 Machine cycle

Instruction execution and peripheral hardware operation are synchronized with the main system clock.

The minimum instruction execution unit is called an "machine cycle". There are a total of 10 different types of instructions for the TLCS-870/C Series: Ranging from 1-cycle instructions which require one machine cycle for execution to 10-cycle instructions which require 10 machine cycles for execution. A machine cycle consists of 4 states (S0 to S3), and each state consists of one main system clock.

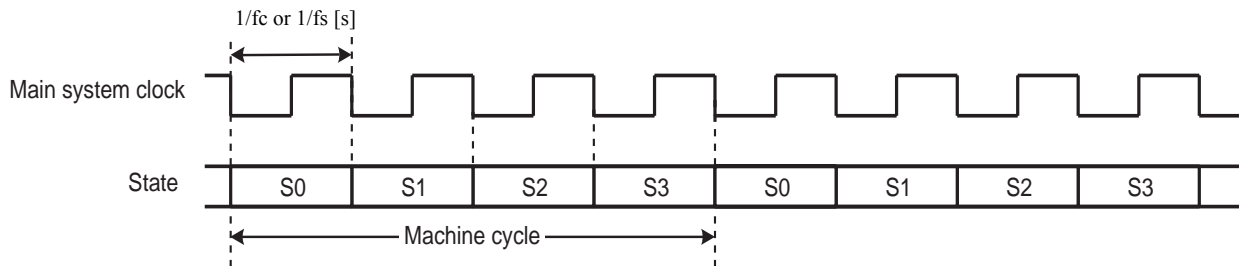


Figure 2-5 Machine Cycle

2.2.3 Operation Mode Control Circuit

The operation mode control circuit starts and stops the oscillation circuits for the high-frequency and low-frequency clocks, and switches the main system clock. There are three operating modes: Single clock mode, dual clock mode and STOP mode. These modes are controlled by the system control registers (SYSCR1 and SYSCR2). Figure 2-6 shows the operating mode transition diagram.

2.2.3.1 Single-clock mode

Only the oscillation circuit for the high-frequency clock is used, and P21 (XTIN) and P22 (XTOUT) pins are used as input/output ports. The main-system clock is obtained from the high-frequency clock. In the single-clock mode, the machine cycle time is  $4/fc$  [s].

(1) NORMAL1 mode

In this mode, both the CPU core and on-chip peripherals operate using the high-frequency clock. The TMP86FS23UG is placed in this mode after reset.

## (2) IDLE1 mode

In this mode, the internal oscillation circuit remains active. The CPU and the watchdog timer are halted; however on-chip peripherals remain active (Operate using the high-frequency clock).

IDLE1 mode is started by `SYSCR2<IDLE> = "1"`, and IDLE1 mode is released to NORMAL1 mode by an interrupt request from the on-chip peripherals or external interrupt inputs. When the IMF (Interrupt master enable flag) is "1" (Interrupt enable), the execution will resume with the acceptance of the interrupt, and the operation will return to normal after the interrupt service is completed. When the IMF is "0" (Interrupt disable), the execution will resume with the instruction which follows the IDLE1 mode start instruction.

## (3) IDLE0 mode

In this mode, all the circuit, except oscillator and the timer-base-timer, stops operation.

This mode is enabled by `SYSCR2<TGHALT> = "1"`.

When IDLE0 mode starts, the CPU stops and the timing generator stops feeding the clock to the peripheral circuits other than TBT. Then, upon detecting the falling edge of the source clock selected with `TBTCR<TBTCK>`, the timing generator starts feeding the clock to all peripheral circuits.

When returned from IDLE0 mode, the CPU restarts operating, entering NORMAL1 mode back again. IDLE0 mode is entered and returned regardless of how `TBTCR<TBTEN>` is set. When `IMF = "1"`, `EF6` (TBT interrupt individual enable flag) = "1", and `TBTCR<TBTEN> = "1"`, interrupt processing is performed. When IDLE0 mode is entered while `TBTCR<TBTEN> = "1"`, the `INTTBT` interrupt latch is set after returning to NORMAL1 mode.

### 2.2.3.2 Dual-clock mode

Both the high-frequency and low-frequency oscillation circuits are used in this mode. P21 (XTIN) and P22 (XTOUT) pins cannot be used as input/output ports. The main system clock is obtained from the high-frequency clock in NORMAL2 and IDLE2 modes, and is obtained from the low-frequency clock in SLOW and SLEEP modes. The machine cycle time is  $4/f_c$  [s] in the NORMAL2 and IDLE2 modes, and  $4/f_s$  [s] (122  $\mu$ s at  $f_s = 32.768$  kHz) in the SLOW and SLEEP modes.

The TLCS-870/C is placed in the signal-clock mode during reset. To use the dual-clock mode, the low-frequency oscillator should be turned on at the start of a program.

## (1) NORMAL2 mode

In this mode, the CPU core operates with the high-frequency clock. On-chip peripherals operate using the high-frequency clock and/or low-frequency clock.

## (2) SLOW2 mode

In this mode, the CPU core operates with the low-frequency clock, while both the high-frequency clock and the low-frequency clock are operated. As the `SYSCR2<SYSCK>` becomes "1", the hardware changes into SLOW2 mode. As the `SYSCR2<SYSCK>` becomes "0", the hardware changes into NORMAL2 mode. As the `SYSCR2<XEN>` becomes "0", the hardware changes into SLOW1 mode. Do not clear `SYSCR2<XTEN>` to "0" during SLOW2 mode.

## (3) SLOW1 mode

This mode can be used to reduce power-consumption by turning off oscillation of the high-frequency clock. The CPU core and on-chip peripherals operate using the low-frequency clock.

Switching back and forth between SLOW1 and SLOW2 modes are performed by SYSCR2<XEN>. In SLOW1 and SLEEP modes, the input clock to the 1st stage of the divider is stopped; output from the 1st to 6th stages is also stopped.

#### (4) IDLE2 mode

In this mode, the internal oscillation circuit remain active. The CPU and the watchdog timer are halted; however, on-chip peripherals remain active (Operate using the high-frequency clock and/or the low-frequency clock). Starting and releasing of IDLE2 mode are the same as for IDLE1 mode, except that operation returns to NORMAL2 mode.

#### (5) SLEEP1 mode

In this mode, the internal oscillation circuit of the low-frequency clock remains active. The CPU, the watchdog timer, and the internal oscillation circuit of the high-frequency clock are halted; however, on-chip peripherals remain active (Operate using the low-frequency clock). Starting and releasing of SLEEP mode are the same as for IDLE1 mode, except that operation returns to SLOW1 mode. In SLOW1 and SLEEP1 modes, the input clock to the 1st stage of the divider is stopped; output from the 1st to 6th stages is also stopped.

#### (6) SLEEP2 mode

The SLEEP2 mode is the idle mode corresponding to the SLOW2 mode. The status under the SLEEP2 mode is same as that under the SLEEP1 mode, except for the oscillation circuit of the high-frequency clock.

#### (7) SLEEP0 mode

In this mode, all the circuit, except oscillator and the timer-base-timer, stops operation. This mode is enabled by setting “1” on bit SYSCR2<TGHALT>.

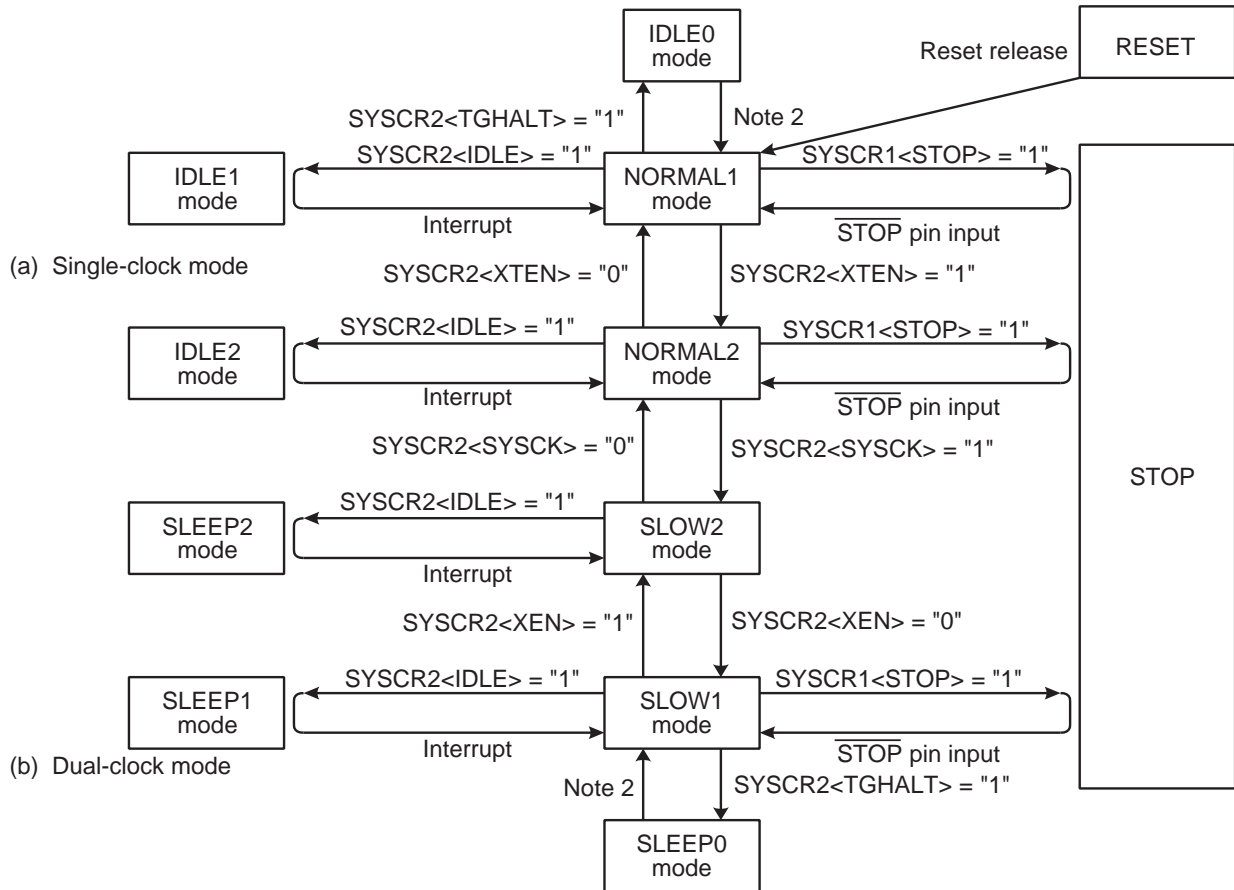
When SLEEP0 mode starts, the CPU stops and the timing generator stops feeding the clock to the peripheral circuits other than TBT. Then, upon detecting the falling edge of the source clock selected with TBTCR<TBTCK>, the timing generator starts feeding the clock to all peripheral circuits.

When returned from SLEEP0 mode, the CPU restarts operating, entering SLOW1 mode back again. SLEEP0 mode is entered and returned regardless of how TBTCR<TBTEN> is set. When IMF = “1”, EF6 (TBT interrupt individual enable flag) = “1”, and TBTCR<TBTEN> = “1”, interrupt processing is performed. When SLEEP0 mode is entered while TBTCR<TBTEN> = “1”, the INTTBT interrupt latch is set after returning to SLOW1 mode.

### 2.2.3.3 STOP mode

In this mode, the internal oscillation circuit is turned off, causing all system operations to be halted. The internal status immediately prior to the halt is held with a lowest power consumption during STOP mode.

STOP mode is started by the system control register 1 (SYSCR1), and STOP mode is released by a inputting (Either level-sensitive or edge-sensitive can be programmably selected) to the  $\overline{\text{STOP}}$  pin. After the warm-up period is completed, the execution resumes with the instruction which follows the STOP mode start instruction.



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

Note 2: The mode is released by falling edge of TBTCR<TBTCk> setting.

Figure 2-6 Operating Mode Transition Diagram

Table 2-1 Operating Mode and Conditions

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

**System Control Register 1**

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

STOP	STOP mode start	0: CPU core and peripherals remain active 1: CPU core and peripherals are halted (Start STOP mode)		R/W	
RELM	Release method for STOP mode	0: Edge-sensitive release 1: Level-sensitive release		R/W	
RETM	Operating mode after STOP mode	0: Return to NORMAL1/2 mode 1: Return to SLOW1 mode		R/W	
OUTEN	Port output during STOP mode	0: High impedance 1: Output kept		R/W	
WUT	Warm-up time at releasing STOP mode		Return to NORMAL mode	Return to SLOW mode	R/W
		00	$3 \times 2^{16}/fc$	$3 \times 2^{13}/fs$	
		01	$2^{16}/fc$	$2^{13}/fs$	
		10	$3 \times 2^{14}/fc$	$3 \times 2^6/fs$	
		11	$2^{14}/fc$	$2^6/fs$	

- Note 1: Always set RETM to "0" when transitioning from NORMAL mode to STOP mode. Always set RETM to "1" when transitioning from SLOW mode to STOP mode.
- Note 2: When STOP mode is released with  $\overline{\text{RESET}}$  pin input, a return is made to NORMAL1 regardless of the RETM contents.
- Note 3: fc: High-frequency clock [Hz], fs: Low-frequency clock [Hz], \*, Don't care
- Note 4: Bits 1 and 0 in SYSCR1 are read as undefined data when a read instruction is executed.
- Note 5: As the hardware becomes STOP mode under OUTEN = "0", input value is fixed to "0"; therefore it may cause external interrupt request on account of falling edge.
- Note 6: When the key-on wakeup is used, RELM should be set to "1".
- Note 7: Port P20 is used as  $\overline{\text{STOP}}$  pin. Therefore, when stop mode is started, OUTEN does not affect to P20, and P20 becomes High-Z mode.
- Note 8: The warmig-up time should be set correctly for using oscillator.

**System Control Register 2**

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

XEN	High-frequency oscillator control	0: Turn off oscillation 1: Turn on oscillation		R/W
XTEN	Low-frequency oscillator control	0: Turn off oscillation 1: Turn on oscillation		
SYSCK	Main system clock select (Write)/main system clock monitor (Read)	0: High-frequency clock (NORMAL1/NORMAL2/IDLE1/IDLE2) 1: Low-frequency clock (SLOW1/SLOW2/SLEEP1/SLEEP2)		
IDLE	CPU and watchdog timer control (IDLE1/2 and SLEEP1/2 modes)	0: CPU and watchdog timer remain active 1: CPU and watchdog timer are stopped (Start IDLE1/2 and SLEEP1/2 modes)		R/W
TGHALT	TG control (IDLE0 and SLEEP0 modes)	0: Feeding clock to all peripherals from TG 1: Stop feeding clock to peripherals except TBT from TG. (Start IDLE0 and SLEEP0 modes)		

- Note 1: A reset is applied if both XEN and XTEN are cleared to "0", XEN is cleared to "0" when SYSCK = "0", or XTEN is cleared to "0" when SYSCK = "1".
- Note 2: \*: Don't care, TG: Timing generator, \*, Don't care
- Note 3: Bits 3, 1 and 0 in SYSCR2 are always read as undefined value.
- Note 4: Do not set IDLE and TGHALT to "1" simultaneously.
- Note 5: Because returning from IDLE0/SLEEP0 to NORMAL1/SLOW1 is executed by the asynchronous internal clock, the period of IDLE0/SLEEP0 mode might be shorter than the period setting by  $\text{TBTCR} < \text{TBTCCK} >$ .
- Note 6: When IDLE1/2 or SLEEP1/2 mode is released, IDLE is automatically cleared to "0".
- Note 7: When IDLE0 or SLEEP0 mode is released, TGHALT is automatically cleared to "0".
- Note 8: Before setting TGHALT to "1", be sure to stop peripherals. If peripherals are not stopped, the interrupt latch of peripherals may be set after IDLE0 or SLEEP0 mode is released.

## 2.2.4 Operating Mode Control

### 2.2.4.1 STOP mode

STOP mode is controlled by the system control register 1, the  $\overline{\text{STOP}}$  pin input and key-on wakeup input (STOP5 to STOP2) which is controlled by the STOP mode release control register (STOPCR). The  $\overline{\text{STOP}}$  pin is also used both as a port P20 and an  $\overline{\text{INT5}}$  (external interrupt input 5) pin. STOP mode is started by setting SYSCR1<STOP> to “1”. During STOP mode, the following status is maintained.

1. Oscillations are turned off, and all internal operations are halted.
2. The data memory, registers, the program status word and port output latches are all held in the status in effect before STOP mode was entered.
3. The prescaler and the divider of the timing generator are cleared to “0”.
4. The program counter holds the address 2 ahead of the instruction (e.g., [SET (SYSCR1).7]) which started STOP mode.

STOP mode includes a level-sensitive mode and an edge-sensitive mode, either of which can be selected with the SYSCR1<RELM>. Do not use any key-on wakeup input (STOP5 to STOP2) for releasing STOP mode in edge-sensitive mode.

Note 1: The STOP mode can be released by either the STOP or key-on wakeup pin (STOP5 to STOP2). However, because the STOP pin is different from the key-on wakeup and can not inhibit the release input, the STOP pin must be used for releasing STOP mode.

Note 2: During STOP period (from start of STOP mode to end of warm up), due to changes in the external interrupt pin signal, interrupt latches may be set to “1” and interrupts may be accepted immediately after STOP mode is released. Before starting STOP mode, therefore, disable interrupts. Also, before enabling interrupts after STOP mode is released, clear unnecessary interrupt latches.

#### (1) Level-sensitive release mode (RELM = “1”)

In this mode, STOP mode is released by setting the  $\overline{\text{STOP}}$  pin high or setting the STOP5 to STOP2 pin input which is enabled by STOPCR. This mode is used for capacitor backup when the main power supply is cut off and long term battery backup.

Even if an instruction for starting STOP mode is executed while  $\overline{\text{STOP}}$  pin input is high or STOP5 to STOP2 input is low, STOP mode does not start but instead the warm-up sequence starts immediately. Thus, to start STOP mode in the level-sensitive release mode, it is necessary for the program to first confirm that the  $\overline{\text{STOP}}$  pin input is low or STOP5 to STOP2 input is high. The following two methods can be used for confirmation.

1. Testing a port.
2. Using an external interrupt input  $\overline{\text{INT5}}$  ( $\overline{\text{INT5}}$  is a falling edge-sensitive input).

Example 1 :Starting STOP mode from NORMAL mode by testing a port P20.

```
LD          (SYSCR1), 01010000B    ; Sets up the level-sensitive release mode
SSTOPH:    TEST      (P2PRD), 0      ; Wait until the  $\overline{\text{STOP}}$  pin input goes low level
JRS        F, SSTOPH
DI          ; IMF ← 0
SET        (SYSCR1), 7             ; Starts STOP mode
```

Example 2 :Starting STOP mode from NORMAL mode with an INT5 interrupt.

```

PINT5:      TEST      (P2PRD). 0           ; To reject noise, STOP mode does not start if
           JRS       F, SINT5             port P20 is at high
           LD        (SYSCR1), 01010000B  ; Sets up the level-sensitive release mode.
           DI                          ; IMF ← 0
           SET      (SYSCR1). 7           ; Starts STOP mode

SINT5:      RETI
    
```

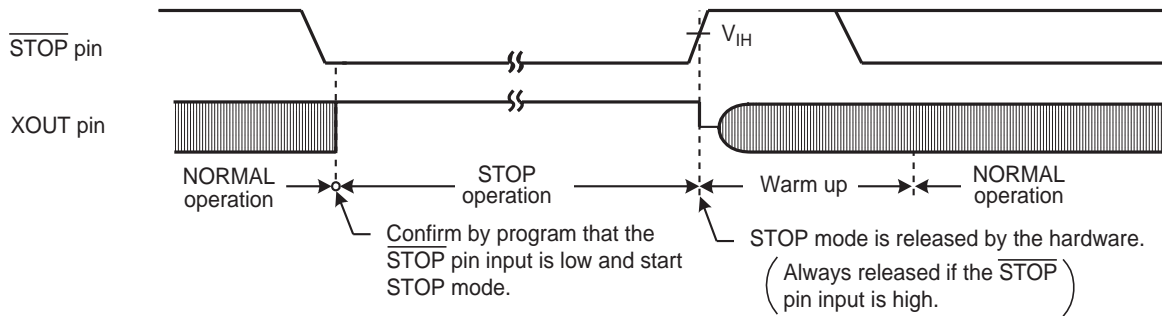


Figure 2-7 Level-sensitive Release Mode

- Note 1: Even if the  $\overline{\text{STOP}}$  pin input is low after warm-up start, the STOP mode is not restarted.
- Note 2: In this case of changing to the level-sensitive mode from the edge-sensitive mode, the release mode is not switched until a rising edge of the  $\overline{\text{STOP}}$  pin input is detected.

(2) Edge-sensitive release mode (RELM = "0")

In this mode, STOP mode is released by a rising edge of the  $\overline{\text{STOP}}$  pin input. This is used in applications where a relatively short program is executed repeatedly at periodic intervals. This periodic signal (for example, a clock from a low-power consumption oscillator) is input to the  $\overline{\text{STOP}}$  pin. In the edge-sensitive release mode, STOP mode is started even when the  $\overline{\text{STOP}}$  pin input is high level. Do not use any STOP5 to STOP2 pin input for releasing STOP mode in edge-sensitive release mode.

Example :Starting STOP mode from NORMAL mode

```

DI          ; IMF ← 0
LD          (SYSCR1), 10010000B  ; Starts after specified to the edge-sensitive release mode
    
```

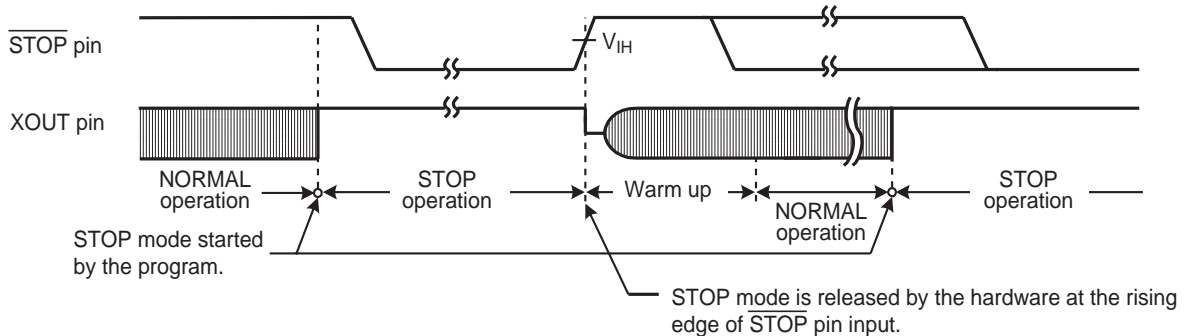


Figure 2-8 Edge-sensitive Release Mode

STOP mode is released by the following sequence.

1. In the dual-clock mode, when returning to NORMAL2, both the high-frequency and low-frequency clock oscillators are turned on; when returning to SLOW1 mode, only the low-frequency clock oscillator is turned on. In the single-clock mode, only the high-frequency clock oscillator is turned on.
2. A warm-up period is inserted to allow oscillation time to stabilize. During warm up, all internal operations remain halted. Four different warm-up times can be selected with the SYSCR1<WUT> in accordance with the resonator characteristics.
3. When the warm-up time has elapsed, normal operation resumes with the instruction following the STOP mode start instruction.

Note 1: When the STOP mode is released, the start is made after the prescaler and the divider of the timing generator are cleared to "0".

Note 2: STOP mode can also be released by inputting low level on the  $\overline{\text{RESET}}$  pin, which immediately performs the normal reset operation.

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

Table 2-2 Warm-up Time Example (at  $f_c = 16.0$  MHz,  $f_s = 32.768$  kHz)

WUT	Warm-up Time [ms]	
	Return to NORMAL Mode	Return to SLOW Mode
00	12.288	750
01	4.096	250
10	3.072	5.85
11	1.024	1.95

Note 1: The warm-up time is obtained by dividing the basic clock by the divider. Therefore, the warm-up time may include a certain amount of error if there is any fluctuation of the oscillation frequency when STOP mode is released. Thus, the warm-up time must be considered as an approximate value.



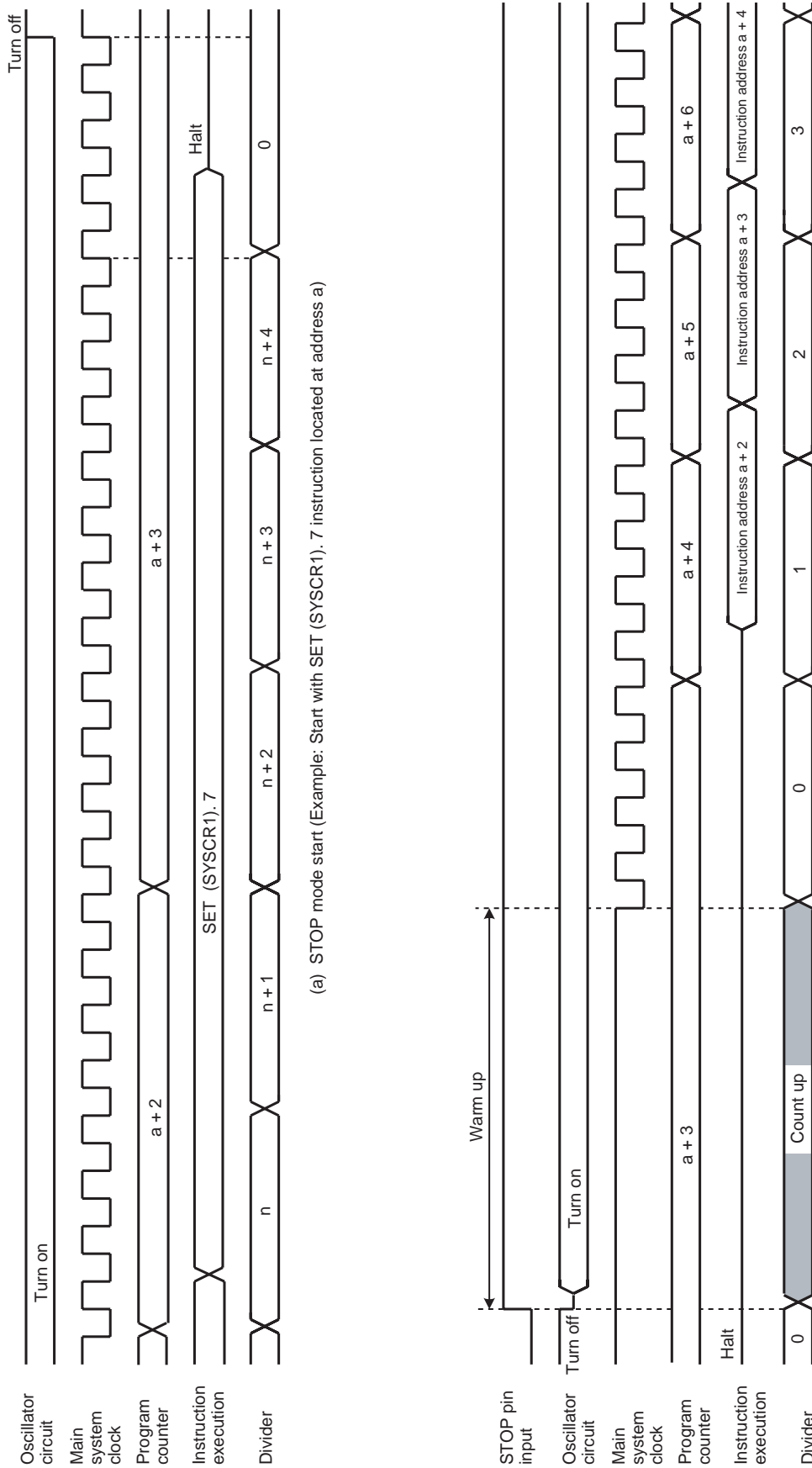


Figure 2-9 STOP Mode Start/Release

### 2.2.4.2 IDLE1/2 mode and SLEEP1/2 mode

IDLE1/2 and SLEEP1/2 modes are controlled by the system control register 2 (SYSCR2) and maskable interrupts. The following status is maintained during these modes.

1. Operation of the CPU and watchdog timer (WDT) is halted. On-chip peripherals continue to operate.
2. The data memory, CPU registers, program status word and port output latches are all held in the status in effect before these modes were entered.
3. The program counter holds the address 2 ahead of the instruction which starts these modes.

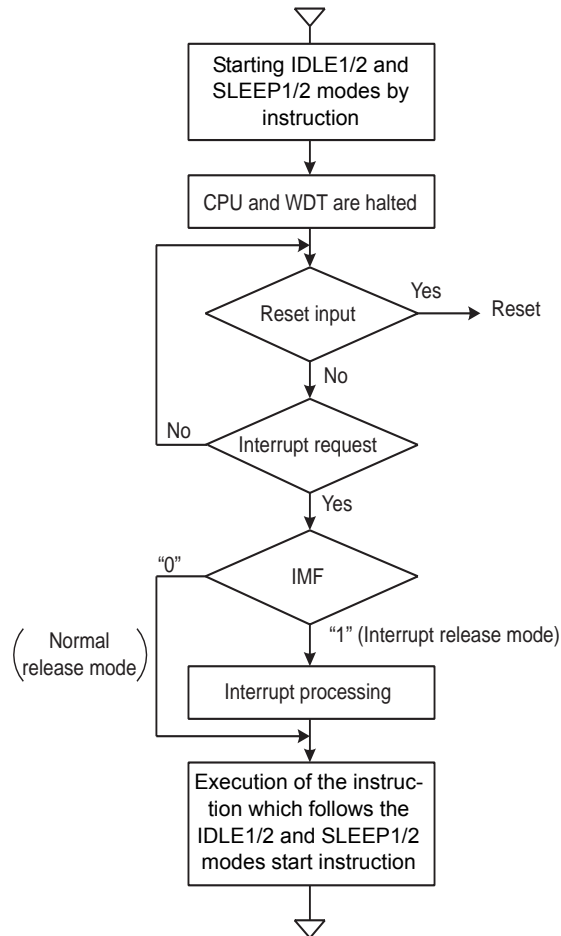


Figure 2-10 IDLE1/2 and SLEEP1/2 Modes

- Start the IDLE1/2 and SLEEP1/2 modes

After IMF is set to "0", set the individual interrupt enable flag (EF) which releases IDLE1/2 and SLEEP1/2 modes. To start IDLE1/2 and SLEEP1/2 modes, set SYSCR2<IDLE> to "1".

- Release the IDLE1/2 and SLEEP1/2 modes

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

IDLE1/2 and SLEEP1/2 modes can also be released by inputting low level on the  $\overline{\text{RESET}}$  pin. After releasing reset, the operation mode is started from NORMAL1 mode.

(1) Normal release mode (IMF = "0")

IDLE1/2 and SLEEP1/2 modes are released by any interrupt source enabled by the individual interrupt enable flag (EF). After the interrupt is generated, the program operation is resumed from the instruction following the IDLE1/2 and SLEEP1/2 modes start instruction. Normally, the interrupt latches (IL) of the interrupt source used for releasing must be cleared to "0" by load instructions.

(2) Interrupt release mode (IMF = "1")

IDLE1/2 and SLEEP1/2 modes are released by any interrupt source enabled with the individual interrupt enable flag (EF) and the interrupt processing is started. After the interrupt is processed, the program operation is resumed from the instruction following the instruction, which starts IDLE1/2 and SLEEP1/2 modes.

Note: When a watchdog timer interrupts is generated immediately before IDLE1/2 and SLEEP1/2 modes are started, the watchdog timer interrupt will be processed but IDLE1/2 and SLEEP1/2 modes will not be started.

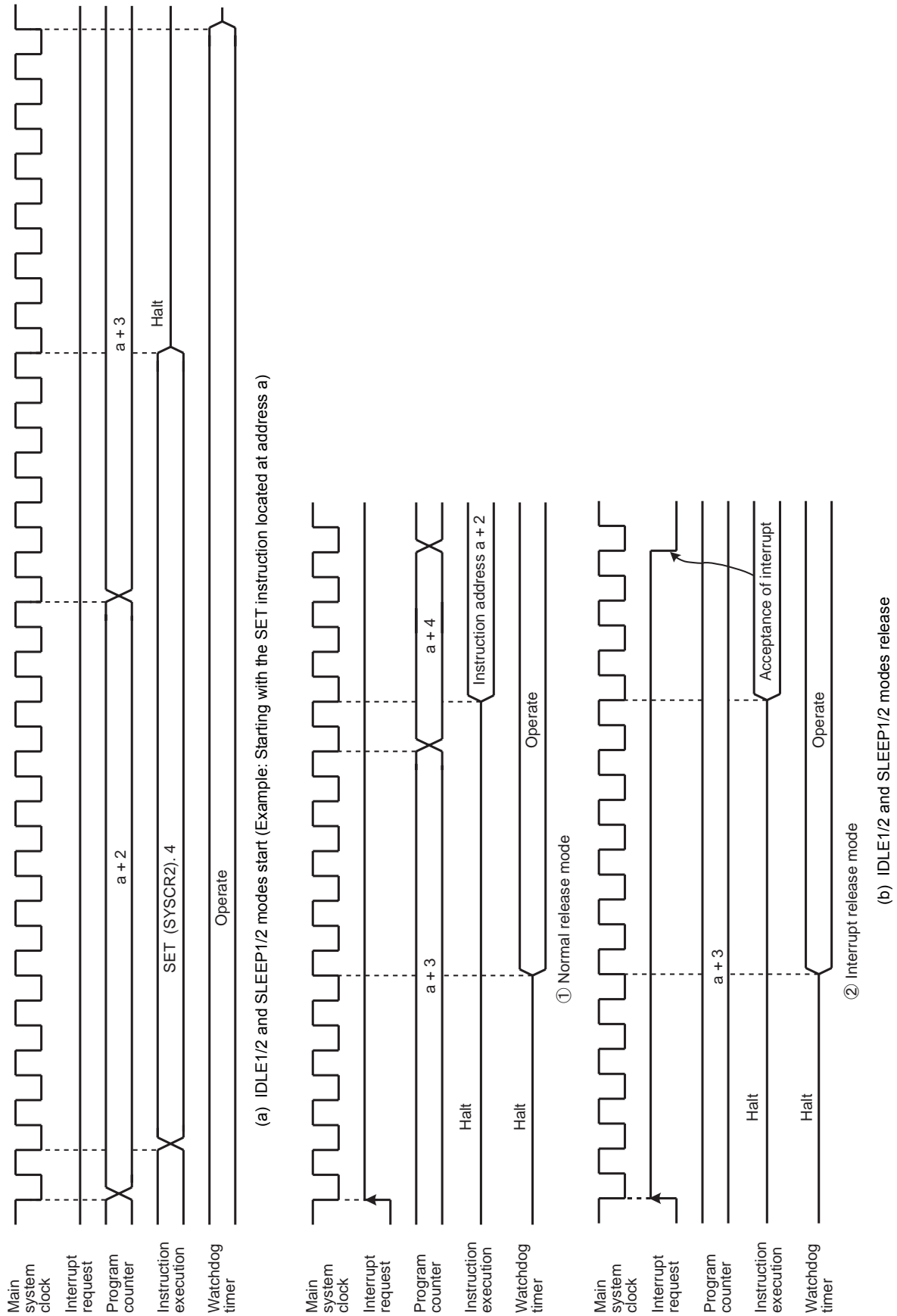


Figure 2-11 IDLE1/2 and SLEEP1/2 Modes Start/Release



- Start the IDLE0 and SLEEP0 modes

Stop (Disable) peripherals such as a timer counter.

To start IDLE0 and SLEEP0 modes, set SYSCR2<TGHALT> to “1”.

- Release the IDLE0 and SLEEP0 modes

IDLE0 and SLEEP0 modes include a normal release mode and an interrupt release mode.

These modes are selected by interrupt master flag (IMF), the individual interrupt enable flag of TBT and TBTCR<TBTEN>.

After releasing IDLE0 and SLEEP0 modes, the SYSCR2<TGHALT> is automatically cleared to “0” and the operation mode is returned to the mode preceding IDLE0 and SLEEP0 modes. Before starting the IDLE0 or SLEEP0 mode, when the TBTCR<TBTEN> is set to “1”, INTTBT interrupt latch is set to “1”.

IDLE0 and SLEEP0 modes can also be released by inputting low level on the  $\overline{\text{RESET}}$  pin. After releasing reset, the operation mode is started from NORMAL1 mode.

Note: IDLE0 and SLEEP0 modes start/release without reference to TBTCR<TBTEN> setting.

- (1) Normal release mode (IMF•EF6•TBTCR<TBTEN> = “0”)

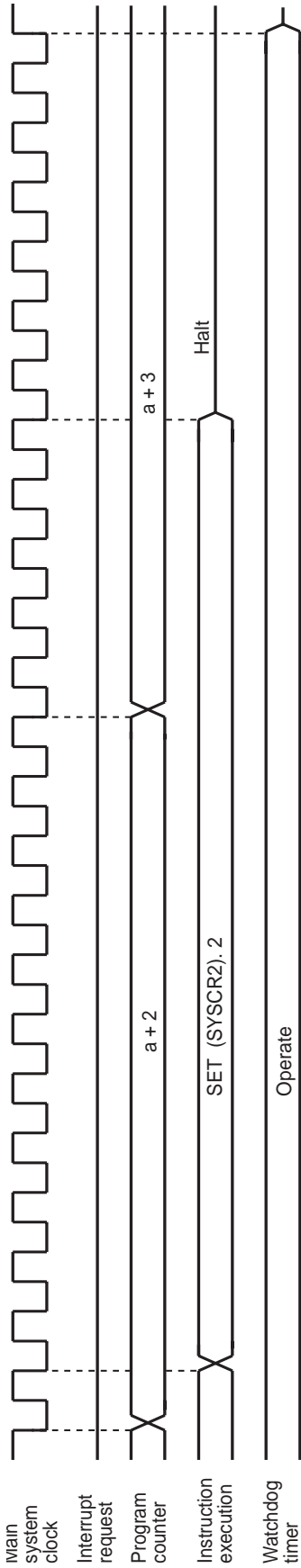
IDLE0 and SLEEP0 modes are released by the source clock falling edge, which is setting by the TBTCR<TBTCK>. After the falling edge is detected, the program operation is resumed from the instruction following the IDLE0 and SLEEP0 modes start instruction. Before starting the IDLE0 or SLEEP0 mode, when the TBTCR<TBTEN> is set to “1”, INTTBT interrupt latch is set to “1”.

- (2) Interrupt release mode (IMF•EF6•TBTCR<TBTEN> = “1”)

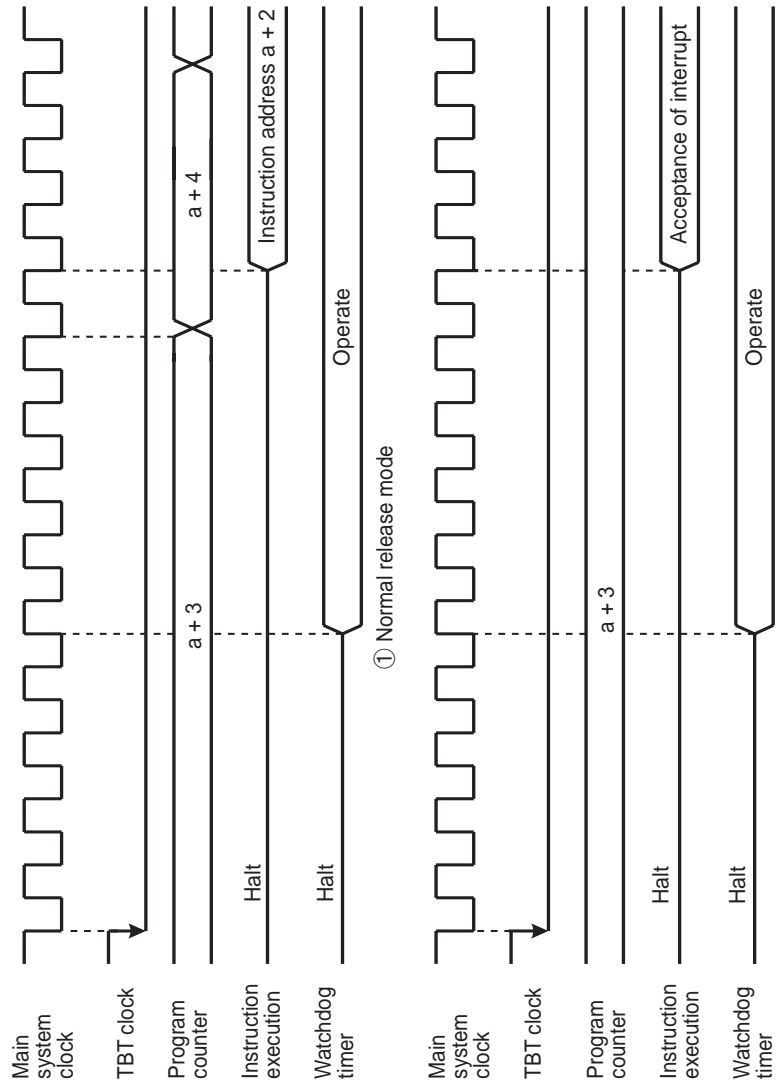
IDLE0 and SLEEP0 modes are released by the source clock falling edge, which is setting by the TBTCR<TBTCK> and INTTBT interrupt processing is started.

Note 1: Because returning from IDLE0, SLEEP0 to NORMAL1, SLOW1 is executed by the asynchronous internal clock, the period of IDLE0, SLEEP0 mode might be the shorter than the period setting by TBTCR<TBTCK>.

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



(a) IDLE0 and SLEEP0 modes start (Example: Starting with the SET instruction located at address a



(b) IDLE and SLEEP0 modes release

Figure 2-13 IDLE0 and SLEEP0 Modes Start/Release

### 2.2.4.4 SLOW mode

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

The following is the methods to switch the mode with the warm-up counter.

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

First, set SYSCR2<SYSCK> to switch the main system clock to the low-frequency clock for SLOW2 mode. Next, clear SYSCR2<XEN> to turn off high-frequency oscillation.

Note: The high-frequency clock can be continued oscillation in order to return to NORMAL2 mode from SLOW mode quickly. Always turn off oscillation of high-frequency clock when switching from SLOW mode to stop mode.

Example 1 :Switching from NORMAL2 mode to SLOW1 mode.

```

SET      (SYSCR2). 5      ; SYSCR2<SYSCK> ← 1
                               (Switches the main system clock to the low-frequency
                               clock for SLOW2)

CLR      (SYSCR2). 7      ; SYSCR2<XEN> ← 0
                               (Turns off high-frequency oscillation)

```

Example 2 :Switching to the SLOW1 mode after low-frequency clock has stabilized.

```

SET      (SYSCR2). 6      ; SYSCR2<XTEN> ← 1

LD       (TC3CR), 43H     ; Sets mode for TC4, 3 (16-bit mode, fs for source)

LD       (TC4CR), 05H     ; Sets warming-up counter mode

LDW     (TTREG3), 8000H   ; Sets warm-up time (Depend on oscillator accompanied)

DI                               ; IMF ← 0

SET      (EIRH). 4        ; Enables INTTC4

EI                               ; IMF ← 1

SET      (TC4CR). 3       ; Starts TC4, 3

:

PINTTC4: CLR      (TC4CR). 3       ; Stops TC4, 3

SET      (SYSCR2). 5      ; SYSCR2<SYSCK> ← 1
                               (Switches the main system clock to the low-frequency clock)

CLR      (SYSCR2). 7      ; SYSCR2<XEN> ← 0
                               (Turns off high-frequency oscillation)

RETI

:

VINTTC4: DW       PINTTC4       ; INTTC4 vector table

```

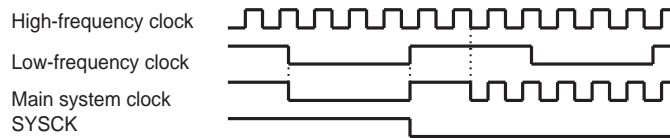


## (2) Switching from SLOW1 mode to NORMAL2 mode

First, set SYSCR2<XEN> to turn on the high-frequency oscillation. When time for stabilization (Warm up) has been taken by the timer/counter (TC4,TC3), clear SYSCR2<SYSCK> to switch the main system clock to the high-frequency clock.

SLOW mode can also be released by inputting low level on the  $\overline{\text{RESET}}$  pin. After releasing reset, the operation mode is started from NORMAL1 mode.

Note: After SYSCK is cleared to "0", executing the instructions is continued by the low-frequency clock for the period synchronized with low-frequency and high-frequency clocks.



Example :Switching from the SLOW1 mode to the NORMAL2 mode (fc = 16 MHz, warm-up time is 4.0 ms).

```

SET      (SYSCR2). 7      ; SYSCR2<XEN> ← 1 (Starts high-frequency oscillation)

LD       (TC3CR), 63H    ; Sets mode for TC4, 3 (16-bit mode, fc for source)

LD       (TC4CR), 05H    ; Sets warming-up counter mode

LD       (TTREG4), 0F8H  ; Sets warm-up time

DI       ; IMF ← 0

SET      (EIRH). 4      ; Enables INTTC4

EI       ; IMF ← 1

SET      (TC4CR). 3      ; Starts TC4, 3

:

PINTTC4: CLR      (TC4CR). 3      ; Stops TC4, 3

CLR      (SYSCR2). 5      ; SYSCR2<SYSCK> ← 0
                          ; (Switches the main system clock to the high-frequency clock)

RETI

:

VINTTC4: DW       PINTTC4      ; INTTC4 vector table
    
```

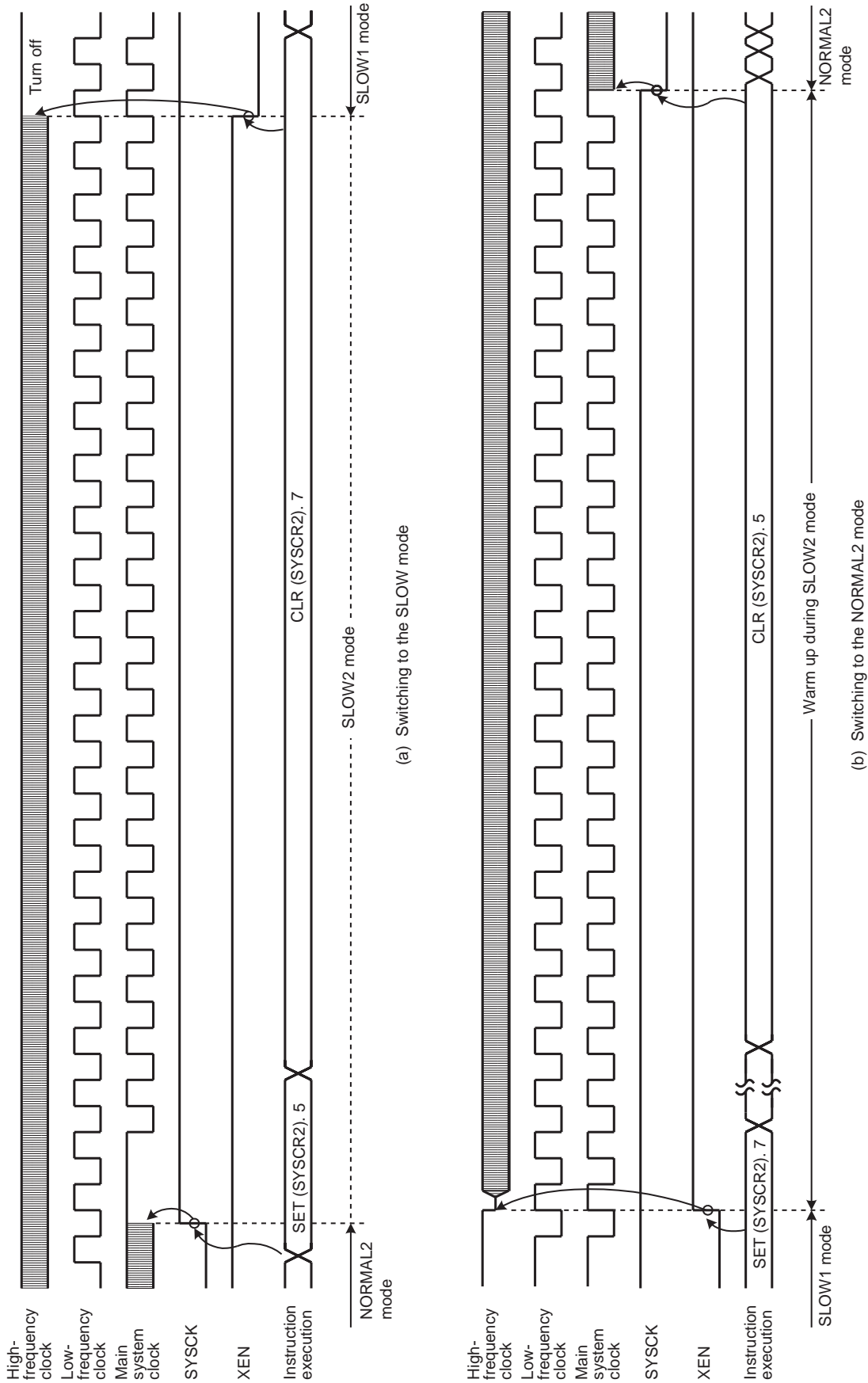


Figure 2-14 Switching between the NORMAL2 and SLOW Modes

## 2.3 Reset Circuit

The TMP86FS23UG has four types of reset generation procedures: An external reset input, an address trap reset, a watchdog timer reset and a system clock reset. Of these reset, the address trap reset, the watchdog timer and the system clock reset are a malfunction reset. When the malfunction reset request is detected, reset occurs during the maximum  $24/f_c[s]$ .

The malfunction reset circuit such as watchdog timer reset, address trap reset and system clock reset is not initialized when power is turned on. Therefore, reset may occur during maximum  $24/f_c[s]$  ( $1.5\mu s$  at 16.0 MHz) when power is turned on.

Table 2-3 shows on-chip hardware initialization by reset action.

Table 2-3 Initializing Internal Status by Reset Action

On-chip Hardware	Initial Value	On-chip Hardware	Initial Value
Program counter (PC)	(FFFEH)	Prescaler and divider of timing generator	0
Stack pointer (SP)	Not initialized		
General-purpose registers (W, A, B, C, D, E, H, L, IX, IY)	Not initialized		
Jump status flag (JF)	Not initialized	Watchdog timer	Enable
Zero flag (ZF)	Not initialized	Output latches of I/O ports	Refer to I/O port circuitry
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 flags (EF)	0	Control registers	Refer to each of control register
Interrupt latches (IL)	0		
		LCD data buffer	Not initialized
		RAM	Not initialized

### 2.3.1 External Reset Input

The  $\overline{\text{RESET}}$  pin contains a Schmitt trigger (Hysteresis) with an internal pull-up resistor.

When the  $\overline{\text{RESET}}$  pin is held at “L” level for at least 3 machine cycles ( $12/f_c [s]$ ) with the power supply voltage within the operating voltage range and oscillation stable, a reset is applied and the internal state is initialized.

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

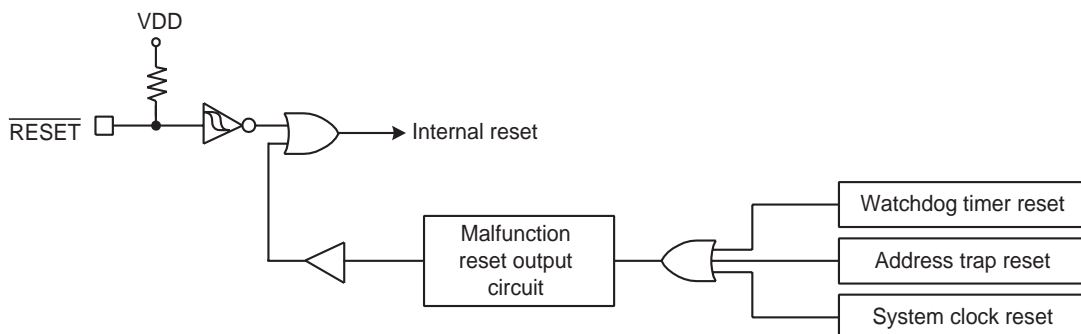
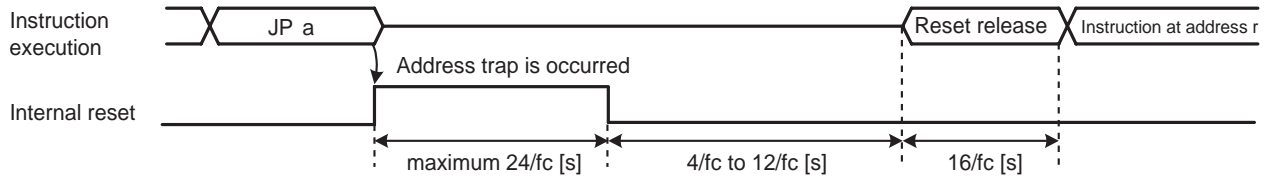


Figure 2-15 Reset Circuit

### 2.3.2 Address trap reset

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 (when WDTTCR1<ATAS> is set to “1”), DBR or the SFR area, address trap reset will be generated. The reset time is maximum  $24/f_c$  [s] ( $1.5\mu\text{s}$  at 16.0 MHz).

Note: The operating mode under address trapped is alternative of reset or interrupt. The address trap area is alternative.



Note 1: Address “a” is in the SFR, DBR or on-chip RAM (WDTTCR1<ATAS> = “1”) space.

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

Figure 2-16 Address Trap Reset

### 2.3.3 Watchdog timer reset

Refer to Section “Watchdog Timer”.

### 2.3.4 System clock reset

If the condition as follows is detected, the system clock reset occurs automatically to prevent dead lock of the CPU. (The oscillation is continued without stopping.)

- In case of clearing SYSCR2<XEN> and SYSCR2<XTEN> simultaneously to “0”.
- In case of clearing SYSCR2<XEN> to “0”, when the SYSCR2<SYSCK> is “0”.
- In case of clearing SYSCR2<XTEN> to “0”, when the SYSCR2<SYSCK> is “1”.

The reset time is maximum  $24/f_c$  ( $1.5\mu\text{s}$  at 16.0 MHz).





### 3. Interrupt Control Circuit

The TMP86FS23UG has a total of 20 interrupt sources excluding reset. 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	INTTBT	IMF• EF6 = 1	IL6	FFF2	7
Internal	INTTC1	IMF• EF7 = 1	IL7	FFF0	8
Internal	INTSIO	IMF• EF8 = 1	IL8	FFEE	9
External	INT2	IMF• EF9 = 1	IL9	FFEC	10
Internal	INTRXD	IMF• EF10 = 1	IL10	FFEA	11
Internal	INTTXD	IMF• EF11 = 1	IL11	FFE8	12
Internal	INTTC4	IMF• EF12 = 1	IL12	FFE6	13
Internal	INTTC6	IMF• EF13 = 1	IL13	FFE4	14
Internal	INTRTC	IMF• EF14 = 1	IL14	FFE2	15
Internal	INTADC	IMF• EF15 = 1	IL15	FFE0	16
Internal	INTTC3	IMF• EF16 = 1	IL16	FFBE	17
External	INT3	IMF• EF17 = 1	IL17	FFBC	18
Internal	INTTC5	IMF• EF18 = 1	IL18	FFBA	19
External	$\overline{INT5}$	IMF• EF19 = 1	IL19	FFB8	20
-	Reserved	IMF• EF20 = 1	IL20	FFB6	21
-	Reserved	IMF• EF21 = 1	IL21	FFB4	22
-	Reserved	IMF• EF22 = 1	IL22	FFB2	23
-	Reserved	IMF• EF23 = 1	IL23	FFB0	24

Note 1: To use the address trap interrupt (INTATRAP), clear WDTTCR1<ATOUT> to “0” (It is set for the “reset request” after reset is cancelled). For details, see “Address Trap”.

Note 2: To use the watchdog timer interrupt (INTWDT), clear WDTTCR1<WDTOUT> to “0” (It is set for the “Reset request” after reset is released). For details, see “Watchdog Timer”.

Note 3: If an INTADC interrupt request is generated while an interrupt with priority lower than the interrupt latch IL15 (INTADC) is being accepted, the INTADC interrupt latch may be cleared without the INTADC interrupt being processed. For details, refer to the corresponding notes in the chapter on the AD converter.

#### 3.1 Interrupt latches (IL19 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 002EH, 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 002CH, 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 (EF19 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 (EF19 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

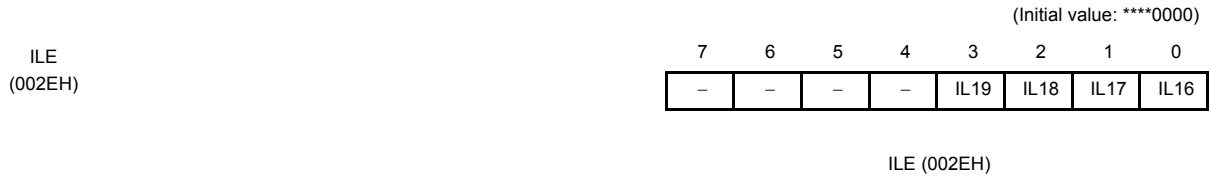
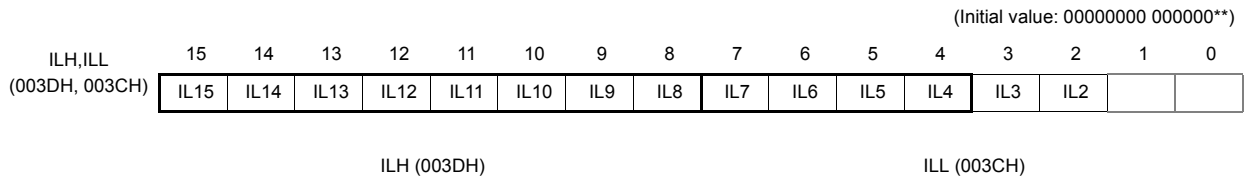
LDW      (EIRL), 1110100010100000B    ; EF15 to EF13, EF11, EF7, EF5 ← 1
:                                         Note: IMF should not be set.
:
EI                                     ; IMF ← 1
    
```

Example 2 :C compiler description example

```

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

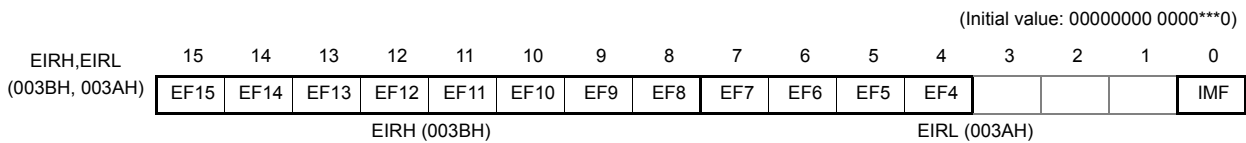
Interrupt Latches



IL19 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



EF19 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".

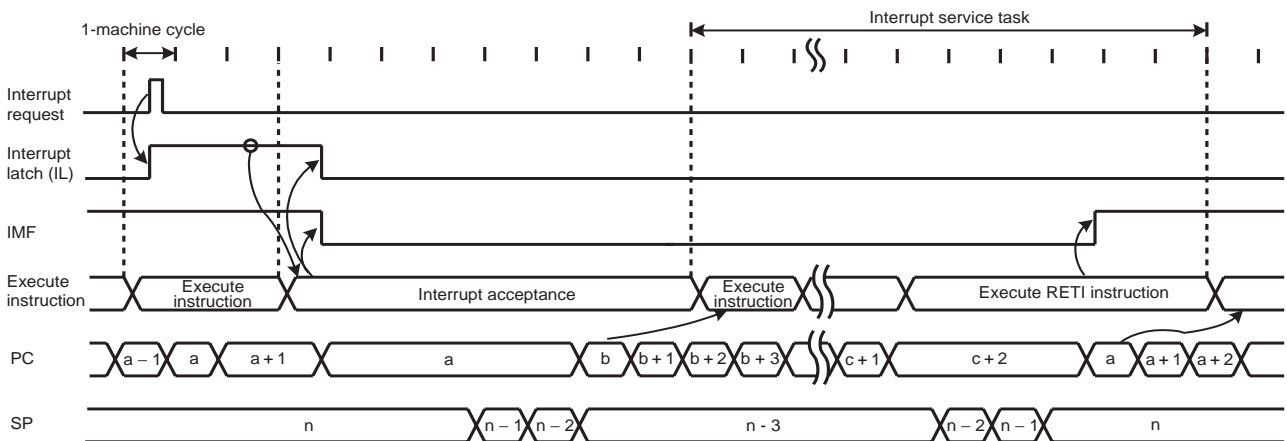
### 3.3 Interrupt Sequence

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.

#### 3.3.1 Interrupt acceptance processing is packaged as follows.

- The interrupt master enable flag (IMF) is cleared to “0” in order to disable the acceptance of any following interrupt.
- The interrupt latch (IL) for the interrupt source accepted is cleared to “0”.
- 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.
- The entry address (Interrupt vector) of the corresponding interrupt service program, loaded on the vector table, is transferred to the program counter.
- 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/fc [s] or 38/fs [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

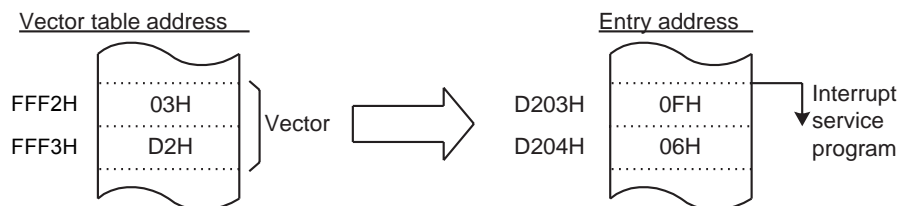


Figure 3-2 Vector table address,Entry address

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.

### 3.3.2 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.

#### 3.3.2.1 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
    
```

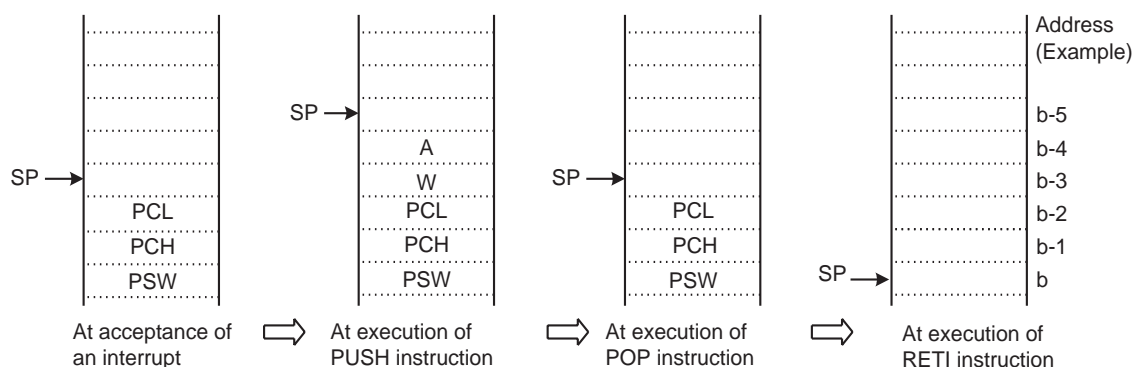


Figure 3-3 Save/store register using PUSH and POP instructions

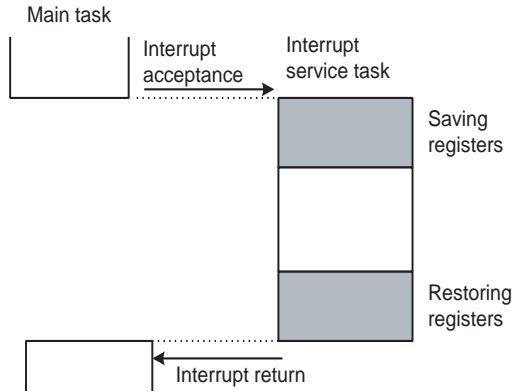
#### 3.3.2.2 Using data transfer instructions

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

### 3.3.3 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.

## 3.4 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.

### 3.4.1 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.

### 3.4.2 Debugging

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

## 3.5 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.

## 3.6 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 (WDTCSR).

### 3.7 External Interrupts

The TMP86FS23UG has 5 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 INT3. The  $\overline{\text{INT0}}$ /P63 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  $\overline{\text{INT0}}$ /P63 pin function selection are performed by the external interrupt control register (EINTCR).

Source	Pin	Enable Conditions	Release Edge	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. 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 • EF9 = 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. 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 • EF17 = 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. 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 • EF19 = 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 NORMAL 1/2 or IDLE 1/2 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

EINTCR	7	6	5	4	3	2	1	0	
(0037H)	INT1NC	INT0EN	-	-	INT3ES	INT2ES	INT1ES		(Initial value: 00** 000*)

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	P63/ $\overline{\text{INT0}}$ pin configuration	0: P63 input/output port 1: $\overline{\text{INT0}}$ pin (Port P63 should be set to an input mode)	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  $2^6/f_c$ .



## 4. Special Function Register (SFR)

The TMP86FS23UG 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 TMP86FS23UG.

### 4.1 SFR

Address	Read	Write
0000H	Reserved	
0001H	P1DR	
0002H	P2DR	
0003H	P3DR	
0004H	P3OUTCR	
0005H	P5DR	
0006H	P6DR	
0007H	P7DR	
0008H	P8DR	
0009H	P1CR	
000AH	P5CR	
000BH	P6CR1	
000CH	P6CR2	
000DH	P7CR	
000EH	ADCCR1	
000FH	ADCCR2	
0010H	TREG1AL	
0011H	TREG1AM	
0012H	TREG1AH	
0013H	TREG1B	
0014H	TC1CR1	TC1CR
0015H	TC1CR2	
0016H	TC1SR	-
0017H	RTCCR	
0018H	TC3CR	
0019H	TC4CR	
001AH	TC5CR	
001BH	TC6CR	
001CH	TTREG3	
001DH	TTREG4	
001EH	TTREG5	
001FH	TTREG6	
0020H	ADCDR2	-
0021H	ADCDR1	-
0022H	Reserved	
0023H	Reserved	
0024H	P8CR	
0025H	UARTSR	UARTCR1

Address	Read	Write
0026H	-	UARTCR2
0027H		LDCDR
0028H		PWREG3
0029H		PWREG4
002AH		PWREG5
002BH		PWREG6
002CH		EIRE
002DH		Reserved
002EH		ILE
002FH		Reserved
0030H		Reserved
0031H		Reserved
0032H		Reserved
0033H		Reserved
0034H	-	WDTCR1
0035H	-	WDTCR2
0036H		TBTCR
0037H		EINTCR
0038H		SYSCR1
0039H		SYSCR2
003AH		EIRL
003BH		EIRH
003CH		ILL
003DH		ILH
003EH		Reserved
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		SEG1/0
0F81H		SEG3/2
0F82H		SEG5/4
0F83H		SEG7/6
0F84H		SEG9/8
0F85H		SEG11/10
0F86H		SEG13/12
0F87H		SEG15/14
0F88H		SEG17/16
0F89H		SEG19/18
0F8AH		SEG21/20
0F8BH		SEG23/22
0F8CH		SEG25/24
0F8DH		SEG27/26
0F8EH		SEG29/28
0F8FH		SEG31/30
0F90H		SIOBR0
0F91H		SIOBR1
0F92H		SIOBR2
0F93H		SIOBR3
0F94H		SIOBR4
0F95H		SIOBR5
0F96H		SIOBR6
0F97H		SIOBR7
0F98H	-	SIOCR1
0F99H	SIOSR	SIOCR2
0F9AH	-	STOPCR
0F9BH	RDBUF	TDBUF
0F9CH	P2PRD	-
0F9DH	P3PRD	-
0F9EH		P1LCR
0F9FH		P5LCR

Address	Read	Write
0FA0H		P7LCR
0FA1H		P8LCR
0FA2H		Reserved
0FA3H		Reserved
0FA4H		MACCR
0FA5H	MACSR	-
0FA6H		MPLDRL
0FA7H		MPLDRH
0FA8H		MPCDRL
0FA9H		MPCDRH
0FAAH	RCALDR1	MADDR1
0FABH	RCALDR2	MADDR2
0FACH	RCALDR3	MADDR3
0FADH	RCALDR4	MADDR4
0FAEH		Reserved
0FAFH		Reserved
0FB0H		Reserved
0FB1H		Reserved
0FB2H		Reserved
0FB3H		Reserved
0FB4H		Reserved
0FB5H		Reserved
0FB6H		Reserved
0FB7H		Reserved
0FB8H		Reserved
0FB9H		Reserved
0FBAH		Reserved
0FBBH		Reserved
0FBCH		Reserved
0FBDH		Reserved
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	-	FLSSTB
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

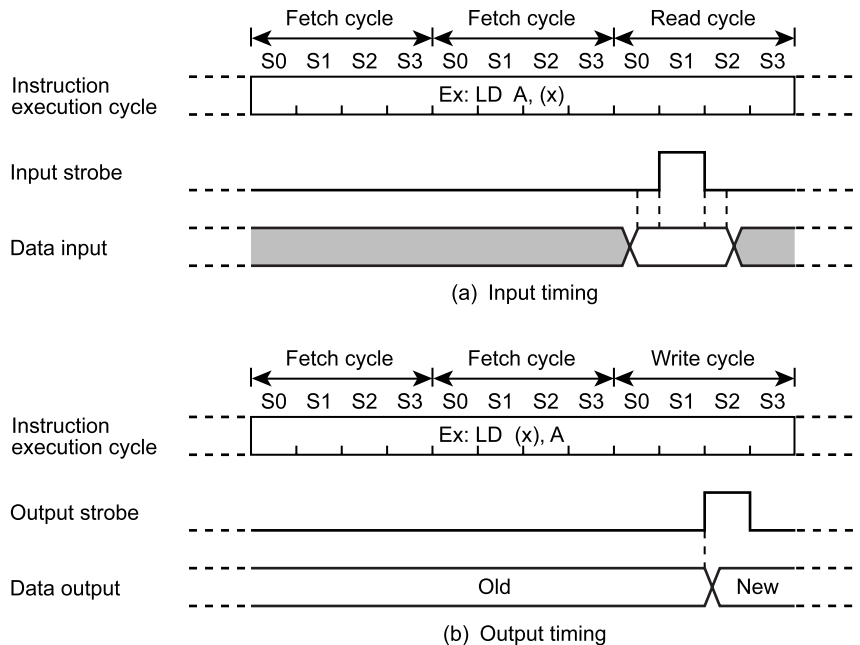
The TMP86FS23UG has 7 parallel input/output ports (48 pins) and output ports (3 pins) as follows.

	Primary Function	Secondary Functions
Port P1	8-bit I/O port	External interrupt input, UART input/output, Serial PROM mode control input and segment output.
Port P2	3-bit I/O port	Low-frequency resonator connections, external interrupt input, STOP mode release signal input.
Port P3	5-bit I/O port	Timer/counter input/output serial interface input/output and divider output.
	3-bit I/O port	Timer/counter input/output.
Port P5	8-bit I/O port	LCD segment output.
Port P6	8-bit I/O port	Analog input, external interrupt input, timer/counter input and STOP mode release signal input.
Port P7	8-bit I/O port	LCD segment output.
Port P8	8-bit I/O port	LCD segment output.

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 P1 (P17 to P10)

Port P1 is an 8-bit input/output port which can be configured as an input or an output in one-bit unit. Port P1 is also used as a UART input/output, an external interrupt input, a serial PROM mode control input and segment output of LCD. Input/output mode is specified by the P1 control register (P1CR).

When used as an input port or a secondary function input pins (UART input or external interrupt input), the corresponding bit of P1CR and P1LCR should be cleared to “0”.

When used as an output port, the corresponding bit of P1CR should be set to “1”, and the respective P1LCR bit should be cleared to “0”. When used as a UART output pin, the corresponding bit of P1CR and the output latch (P1DR) should be set to “1”, and the respective P1LCR bit should be cleared to “0”.

When used as a segment pins of LCD, the respective bit of P1LCR should be set to “1”.

During reset, the P1DR, P1CR and P1LCR are initialized to “0”.

When the bit of P1CR and P1LCR is “0”, the corresponding bit data by read instruction is a terminal input data.

When the bit of P1CR is “0” and that of P1LCR is “1”, the corresponding bit data by read instruction is always “0”.

When the bit of P1CR is “1”, the corresponding bit data by read instruction is the value of P1DR.

Table 5-1 Register Programming for Multi-function Ports

Function	Programmed Value		
	P1DR	P1CR	P1LCR
Port input, UART input, and external interrupt input	*	“0”	“0”
Port “0” output	“0”	“1”	“0”
Port “1” output and UART output	“1”	“1”	“0”
LCD segment output	*	*	“1”

Note: Asterisk (\*) indicates “1” or “0” either of which can be selected.

Table 5-2 Values Read from P1DR and Register Programming

Conditions		Values Read from P1DR
P1CR	P1LCR	
“0”	“0”	Terminal input data
“0”	“1”	“0”
“1”	“0”	Output latch contents
	“1”	



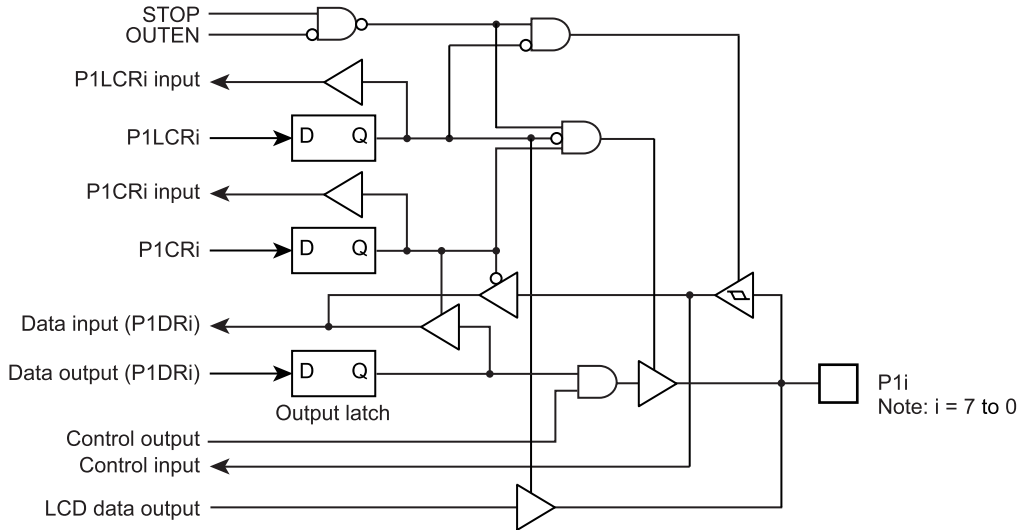


Figure 5-2 Port P1

	7	6	5	4	3	2	1	0	
P1DR (0001H) R/W	P17 SEG24	P16 SEG25	P15 SEG26	P14 SEG27 INT3	P13 SEG28 INT2	P12 SEG29 INT1	P11 SEG30 TXD	P10 SEG31 RXD BOOT	(Initial value: 0000 0000)
P1LCR (0F9EH)									(Initial value: 0000 0000)
P1LCR	Port P1/segment output control (set for each bit individually)						0: P1 input/output port or secondary function (expect for segment) 1: LCD segment output		R/W
P1CR (0009H)									(Initial value: 0000 0000)
P1CR	P1 port input/output control (set for each bit individually)						0: Input mode 1: Output mode		R/W

Note: The port placed in input mode reads the pin input state. Therefore, when the input and output modes are used together, the output latch contents for the port in input mode might be changed by executing a bit manipulation instruction.



### 5.3 Port P3 (P37 to P30)

Port P3 is a 3-bit output and a 5-bit input/output port.

It is also used as a timer/counter input/output, serial interface input/output or divider output.

When used as a timer/counter output, serial interface output or divider output, respective output latch (P3DR) should be set to "1".

It can be selected whether output circuit of P30 to P34 port is C-MOS output or a sink open drain individually, by setting P3OUTCR. When a corresponding bit of P3OUTCR is "0", the output circuit is selected to a sink open drain and when a corresponding bit of P3OUTCR is "1", the output circuit is selected to a C-MOS output. When used as an input port, serial interface input or timer/counter input, respective output control (P3OUTCR) should be set to "0" after P3DR is set to "1". During reset, the P3DR is initialized to "1", and the P3OUTCR is initialized to "0".

P3 port output latch (P3DR) and P3 port terminal input (P3PRD) are located on their respective address.

When read the output latch data, the P3DR should be read and when read the terminal input data, the P3PRD register should be read. If a read instruction is executed for the P3PRD and the P3OUTCR, read data of bits 7 to 5 are unstable.

Table 5-3 Register Programming for Multi-function ports (P34 to P30)

Function	Programmed Value	
	P3DR	P3OUTCR
Port input, serial interface input, or timer counter input	"1"	"0"
Port "0" output	"0"	Programming for each applications
Port "1" output, serial interface output, or timer counter output	"1"	

Table 5-4 Register Programming for Multi-function (P37 to P35)

Function	Programmed Value
	P3DR
Port "0" output	"0"
Port "1" output, timer counter output, or divider output	"1"

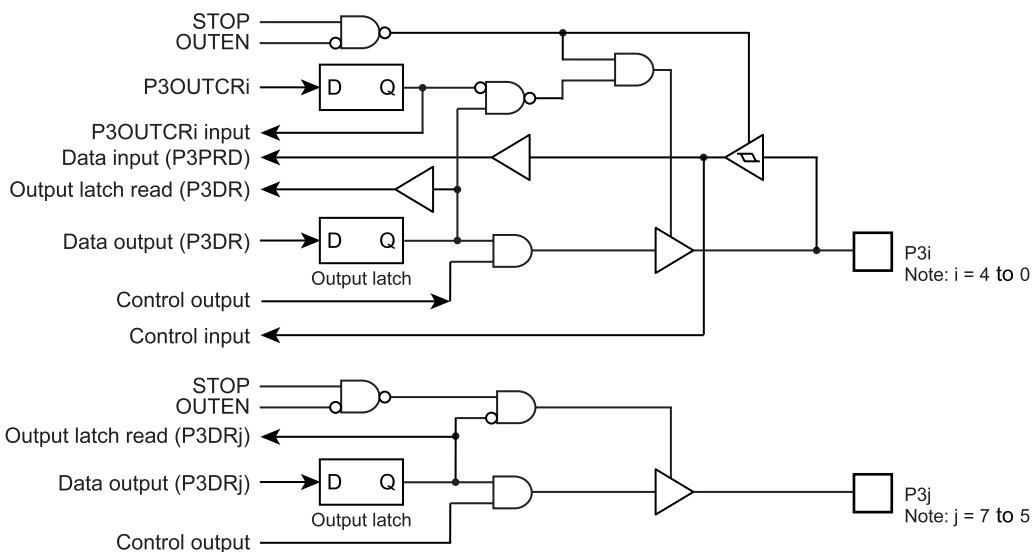


Figure 5-4 Port P3

	7	6	5	4	3	2	1	0	
P3DR (0003H) R/W	P37 DVO	P36 PWM3 PDO3	P35 PWM4 PDO4 PPG4	P34 PWM5 PDO5 TC5	P33 PWM6 PDO6 PPG6 TC6	P32 SCK	P31 SO TC3	P30 SI TC4	(Initial value: 1111 111)

P3OUTCR (0004H)	7	6	5	4	3	2	1	0	(Initial value: ***0 0000)

P3OUTCR	Port P3 output circuit control (set for each bit individually)	0: Sink open-drain output 1: C-MOS output	R/W
---------	---	--	-----

P3PRD (0F9DH) Read only	7	6	5	4	3	2	1	0
				P34	P33	P32	P31	P30

### 5.4 Port P5 (P57 to P50)

Port P5 is an 8-bit input/output port which can be configured as an input or an output in one-bit unit. Port P5 is also used as a segment output of LCD. Input/output mode is specified by the P5 control register (P5CR). When used as an input port, the corresponding bit of P5CR and P5LCR should be cleared to "0". When used as an output port, the corresponding bit of P5CR should be set to "1", and the respective P5LCR bit should be cleared to "0".

When used as a segment pins of LCD, the respective bit of P5LCR should be set to "1". During reset, the output latch (P5DR), P5CR and P5LCR are initialized to "0". When the bit of P5CR and P5LCR is "0", the corresponding bit data by read instruction is a terminal input data. When the bit of P5CR is "0" and that of P5LCR is "1", the corresponding bit data by read instruction is always "0". When the bit of P5CR is "1", the corresponding bit data by read instruction is the value of P5DR.

Table 5-5 Register Programming for Multi-function Ports

Function	Programmed Value		
	P5DR	P5CR	P5LCR
Port input	*	"0"	"0"
Port "0" output	"0"	"1"	"0"
Port "1" output	"1"	"1"	"0"
LCD segment output	*	*	"1"

Note: Asterisk (\*) indicates "1" or "0" either of which can be selected.

Table 5-6 Values Read from P5DR and Register Programming

Conditions		Values Read from P5DR
P5CR	P5LCR	
"0"	"0"	Terminal input data
"0"	"1"	"0"
"1"	"0"	Output latch contents
	"1"	

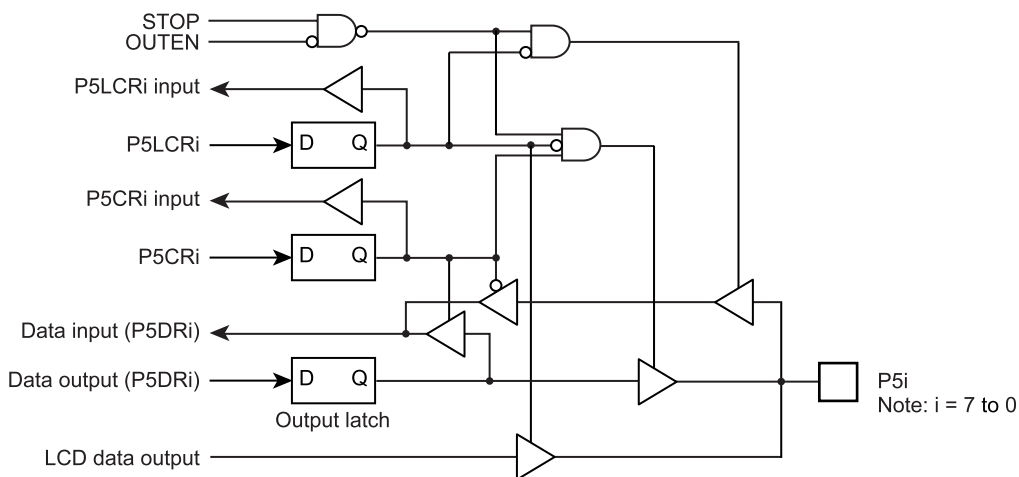


Figure 5-5 Port P5

	7	6	5	4	3	2	1	0	
P5DR (0005H) R/W	P57 SEG16	P56 SEG17	P55 SEG18	P54 SEG19	P53 SEG20	P52 SEG21	P51 SEG22	P50 SEG23	(Initial value: 0000 0000)

	7	6	5	4	3	2	1	0	
P5LCR (0F9FH)									(Initial value: 0000 0000)

P5LCR	Port P5/segment output control (Set for each bit individually)	0: P5 input/output port 1: LCD segment output	R/W
-------	---	--	-----

	7	6	5	4	3	2	1	0	
P5CR (000AH)									(Initial value: 0000 0000)

P5CR	P5 port input/output control (Set for each bit individually)	0: Input mode 1: Output mode	R/W
------	---	---------------------------------	-----

Note: The port placed in input mode reads the pin input state. Therefore, when the input and output modes are used together, the output latch contents for the port in input mode might be changed by executing a bit manipulation instruction.

## 5.5 Port P6 (P67 to P60)

Port P6 is an 8-bit input/output port which can be configured as an input or an output in one-bit unit. Port P6 is also used as an analog input, Key on Wake up input, timer/counter input and external interrupt input. Input/output mode is specified by the P6 control register (P6CR1) and input control register (P6CR2).

When used as an output port, the corresponding bit of P6CR1 should be set to “1”.

When used as an input port, timer/counter input or an external interrupt input, the corresponding bit of P6CR1 should be cleared to “0”, and then, the corresponding bit of P6CR2 should be set to “1”.

When used as an analog input or key on wake up input, the corresponding bit of P6CR1 should be cleared to “0”, and then, the corresponding bit of P6CR2 should be cleared to “0” .

The output latch of each digital input port with multiple functions should be set to “0” to prevent flow-through current. Therefore, the output latch of each port to be used for analog input should be preprogrammed to “0”. The conversion input channel to be used is actually selected by ADCCR1<SAIN>.

During reset, the output latch (P6DR) and P6CR1 are initialized to 0<sup>c</sup>, P6CR2 is initialized to “1”.

When the bit of P6CR1 and P6CR2 is “0”, the corresponding bit data by read instruction is always “0”.

When the bit of P6CR1 is “0” and that of P6CR2 is “1”, the corresponding bit data by read instruction is a terminal input data.

When the bit of P6CR1 is “1”, the corresponding bit data by read instruction is the value of P6DR.

Table 5-7 Register Programming for Multi-function Ports

Function	Programmed Value		
	P6DR	P6CR1	P6CR2
Port input external interrupt input or timer counter input	*	“0”	“1”
Analog input or key-on wake-up input	*	“0”	“0”
Port “0” output	“0”	“1”	*
Port “1” output	“1”	“1”	*

Note: Asterisk (\*) indicates “1” or “0” either of which can be selected.

Table 5-8 Values Read from P6DR and Register Programming

Conditions		Values Read from P6DR
P6CR1	P6CR2	
“0”	“0”	“0”
“0”	“1”	Terminal input data
“1”	“0”	Output latch contents
	“1”	

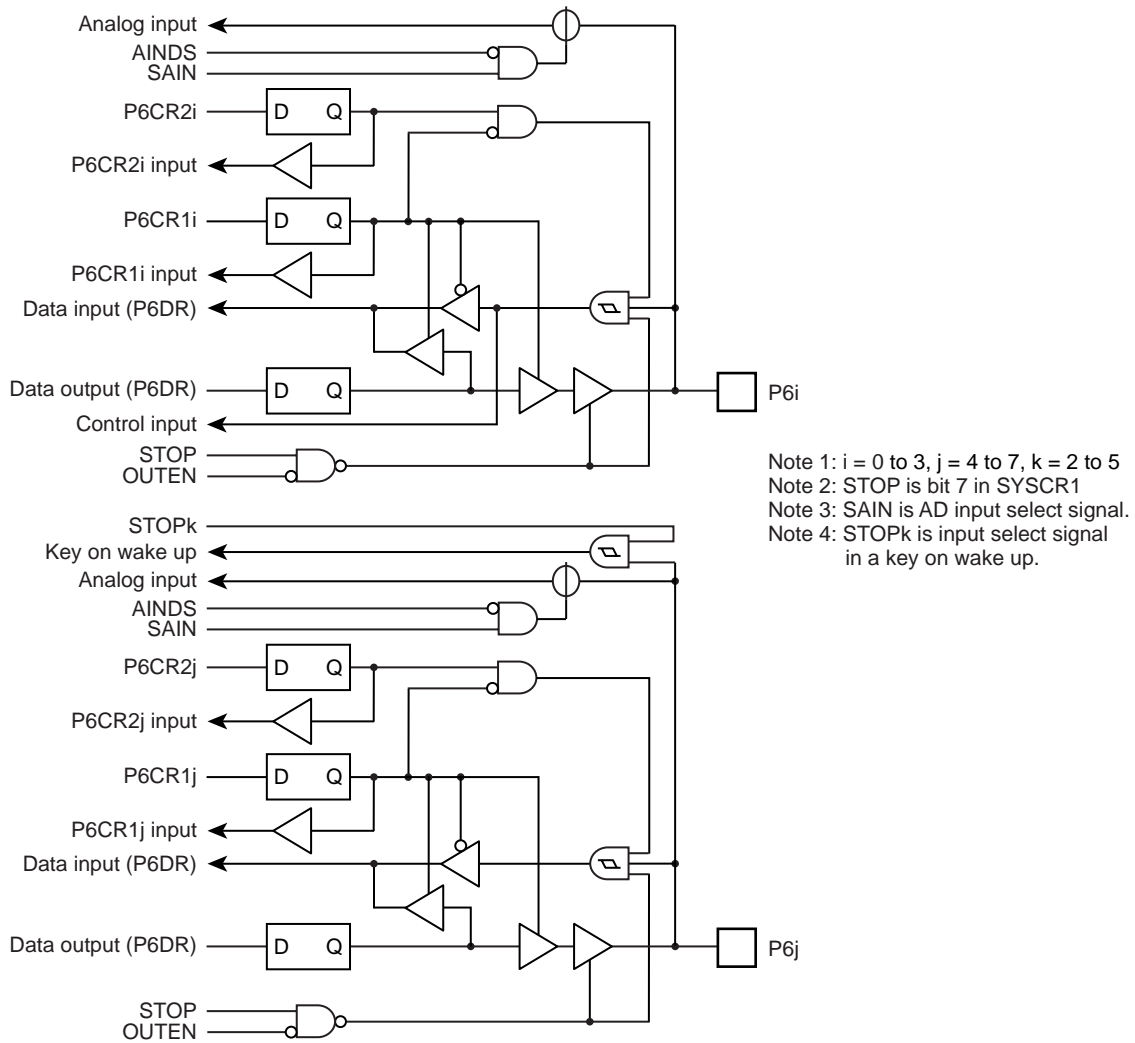


Figure 5-6 Port P6

Note 1: The port placed in input mode reads the pin input state. Therefore, when the input and output modes are used together, the output latch contents for the port in input mode might be changed by executing a bit manipulation instruction.

Note 2: When used as an analog inport, be sure to clear the corresponding bit of P6CR2 to disable the port input.

Note 3: Do not set the output mode (P6CR1 = "1" ) for the pin used as a analog input pin.

Note 4: Pins not used for analog input can be used as I/O ports. 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.



	7	6	5	4	3	2	1	0	
P6DR (0006H) R/W	P67 AIN7 STOP5	P66 AIN6 STOP4	P65 AIN5 STOP3	P64 AIN4 STOP2	P63 AIN3 INT0	P62 AIN2 ECNT	P61 AIN1 ECIN	P60 AIN0	(Initial value: 0000 0000)

P6CR1 (000BH)	7	6	5	4	3	2	1	0	(Initial value: 0000 0000)

P6CR1	I/O control for port P6 (Specified for each bit)	0: Port input, Key on wake up input, Analog input, external interrupt input or timer counter input 1: Port output	R/W
-------	---	---	-----

P6CR2 (000CH)	7	6	5	4	3	2	1	0	(Initial value: 1111 1111)

P6CR2	P6 port input control (Specified for each bit)	0: Analog input or Key on wake up input 1: Port input , external interrupt input or timer counter input	R/W
-------	---	--	-----

## 5.6 Port P7 (P77 to P70)

Port P7 is an 8-bit input/output port which can be configured as an input or an output in one-bit unit. Port P7 is also used as a segment output of LCD. Input/output mode is specified by the P7 control register (P7CR).

When used as an input port, the corresponding bit of P7CR and P7LCR should be cleared to “0”.

When used as an output port, the corresponding bit of P7CR should be set to “1”, and the respective P7LCR bit should be cleared to “0”.

When used as a segment pins of LCD, the respective bit of P7LCR should be set to “1”.

During reset, the output latch (P7DR), P7CR and P7LCR are initialized to “0”.

When the bit of P7CR and P7LCR is “0”, the corresponding bit data by read instruction is a terminal input data.

When the bit of P7CR is “0” and that of P7LCR is “1”, the corresponding bit data by read instruction is always “0”.

When the bit of P7CR is “1”, the corresponding bit data by read instruction is the value of P7DR.

Table 5-9 Register Programming for Multi-function Ports

Function	Programmed Value		
	P7DR	P7CR	P7LCR
Port input	*	“0”	“0”
Port “0” output	“0”	“1”	“0”
Port “1” output	“1”	“1”	“0”
LCD segment output	*	*	“1”

Note: Asterisk (\*) indicates “1” or “0” either of which can be selected.

Table 5-10 Values Read from P7DR and Register Programming

Conditions		Values Read from P7DR
P7CR	P7LCR	
“0”	“0”	Terminal input data
“0”	“1”	“0”
“1”	“0”	Output latch contents
	“1”	

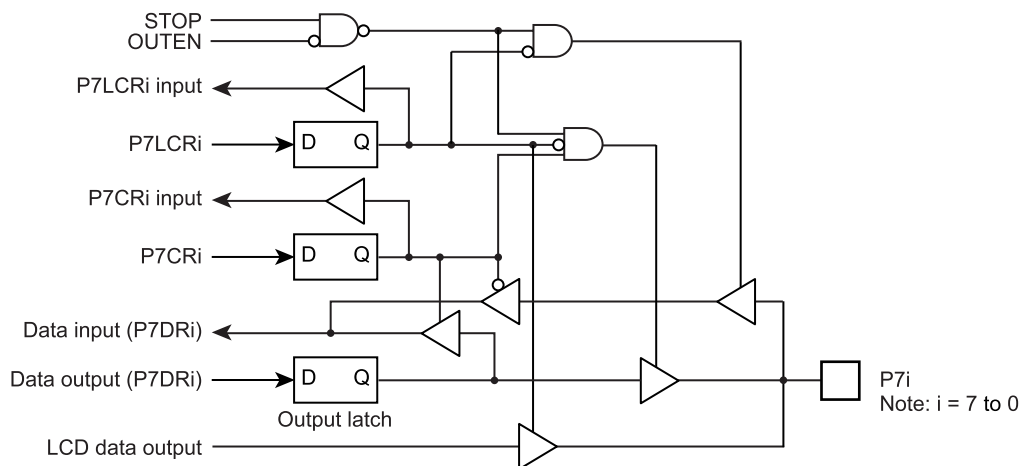


Figure 5-7 Port P7

P7DR (0007H) R/W	7	6	5	4	3	2	1	0	
	P77 SEG8	P76 SEG9	P75 SEG10	P74 SEG11	P73 SEG12	P72 SEG13	P71 SEG14	P70 SEG15	(Initial value: 0000 0000)

P7LCR (0FA0H)	7	6	5	4	3	2	1	0	
									(Initial value: 0000 0000)

P7LCR	Port P7/segment output control (set for each bit individually)	0: P7 input/output port 1: Segment output	R/W
-------	---	--	-----

P7CR (000DH)	7	6	5	4	3	2	1	0	
									(Initial value: 0000 0000)

P7CR	P7 port input/output control (set for each bit individually)	0: Input mode 1: Output mode	R/W
------	---	---------------------------------	-----

Note: The port placed in input mode reads the pin input state. Therefore, when the input and output modes are used together, the output latch contents for the port in input mode might be changed by executing a bit manipulation instruction.

## 5.7 Port P8 (P87 to P80)

Port P8 is an 8-bit input/output port which can be configured as an input or an output in one-bit unit. Port P8 is also used as a segment output of LCD. Input/output mode is specified by the P8 control register (P8CR).

When used as an input port, the corresponding bit of P8CR and P8LCR should be cleared to “0”.

When used as an output port, the corresponding bit of P8CR should be set to “1”, and the respective P8LCR bit should be cleared to “0”.

When used as a segment pins of LCD, the respective bit of P8LCR should be set to “1”.

During reset, the output latch (P8DR), P8CR and P8LCR are initialized to “0”.

When the bit of P8CR and P8LCR is “0”, the corresponding bit data by read instruction is a terminal input data.

When the bit of P8CR is “0” and that of P8LCR is “1”, the corresponding bit data by read instruction is always “0”.

When the bit of P8CR is “1”, the corresponding bit data by read instruction is the value of P8DR.

Table 5-11 Register Programming for Multi-function ports

Function	Port Input		
	P8DR	P8CR	P8LCR
Port input	*	“0”	“0”
Port “0” output	“0”	“1”	“0”
Port “1” output	“1”	“1”	“0”
LCD segment output	*	*	“1”

Note: Asterisk (\*) indicates “1” or “0” either of which can be selected.

Table 5-12 Values Read from P8DR and Register Programming

Conditions		Values Read from P8DR
P8CR	P8LCR	
“0”	“0”	Terminal input data
“0”	“1”	“0”
“1”	“0”	Output latch contents
	“1”	

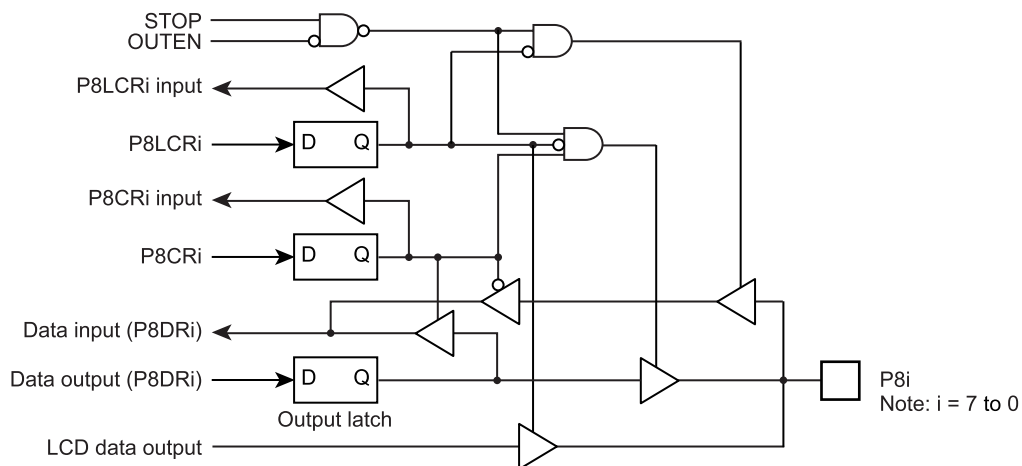


Figure 5-8 Port P8

P8DR (0008H) R/W	7	6	5	4	3	2	1	0	
	P87 SEG0	P86 SEG1	P85 SEG2	P84 SEG3	P83 SEG4	P82 SEG5	P81 SEG6	P80 SEG7	(Initial value: 0000 0000)

P8LCR (0FA1H)	7	6	5	4	3	2	1	0	
									(Initial value: 0000 0000)

P8LCR	P8 port segment output control (Specified for each bit)	0: Input/Output port 1: LCD segment output	R/W
-------	--	---	-----

P8CR (0024H)	7	6	5	4	3	2	1	0	
									(Initial value: 0000 0000)

P8CR	P8 port input/output control (Specified for each bit)	0: Input mode 1: Output mode	R/W
------	--	---------------------------------	-----

Note: The port placed in input mode reads the pin input state. Therefore, when the input and output modes are used together, the output latch contents for the port in input mode might be changed by executing a bit manipulation instruction.



## 6. 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).

### 6.1 Time Base Timer

#### 6.1.1 Configuration

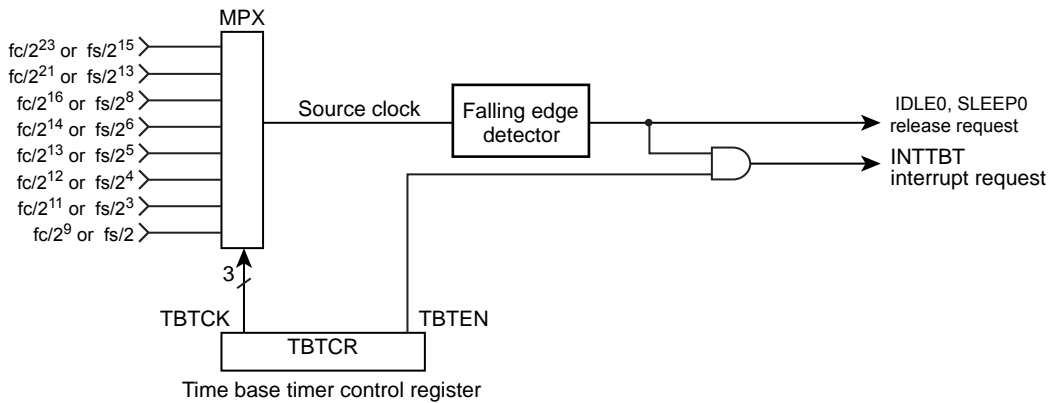


Figure 6-1 Time Base Timer configuration

#### 6.1.2 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	TBTCCK				(Initial Value: 0000 0000)

TBTCCK	Time Base Timer interrupt Frequency select : [Hz]	0: Disable 1: Enable			R/W
		NORMAL 1/2, IDLE 1/2 Mode		SLOW 1/2 SLEEP 1/2 Mode	
		DV7CK = 0	DV7CK = 1		
000	$fc/2^{23}$	$fs/2^{15}$	$fs/2^{15}$		
001	$fc/2^{21}$	$fs/2^{13}$	$fs/2^{13}$		
010	$fc/2^{16}$	$fs/2^8$	–		
011	$fc/2^{14}$	$fs/2^6$	–		
100	$fc/2^{13}$	$fs/2^5$	–		
101	$fc/2^{12}$	$fs/2^4$	–		
110	$fc/2^{11}$	$fs/2^3$	–		
111	$fc/2^9$	$fs/2$	–		

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      (TBTCK) , 00000010B      ; TBTCK ← 010
LD      (TBTCK) , 00001010B      ; TBTEN ← 1
DI      ; IMF ← 0
SET     (EIRL) . 6
```

Table 6-1 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	
000	1.91	1	1
001	7.63	4	4
010	244.14	128	-
011	976.56	512	-
100	1953.13	1024	-
101	3906.25	2048	-
110	7812.5	4096	-
111	31250	16384	-

### 6.1.3 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 6-2 ).

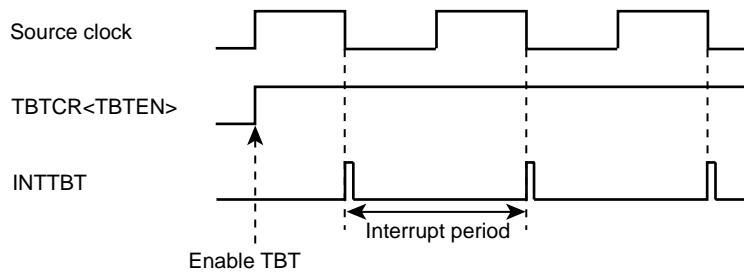


Figure 6-2 Time Base Timer Interrupt



## 6.2 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.

### 6.2.1 Configuration

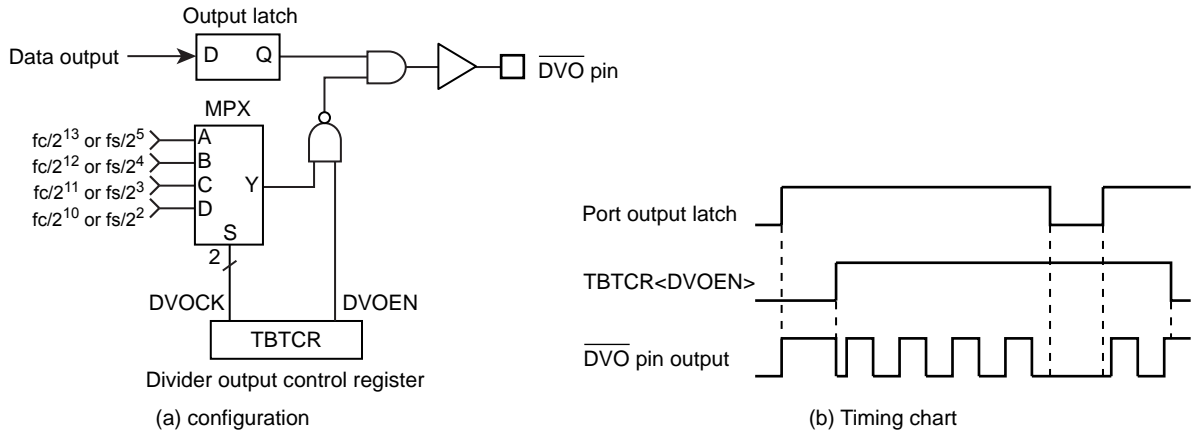
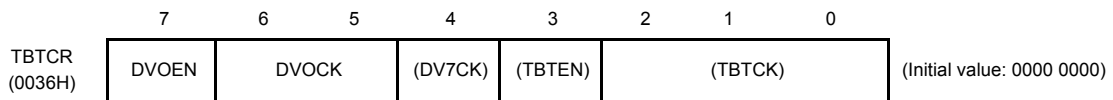


Figure 6-3 Divider Output

### 6.2.2 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	
DVOCK	Divider Output ( $\overline{DVO}$ ) frequency selection: [Hz]	NORMAL 1/2, IDLE 1/2 Mode		SLOW 1/2 SLEEP 1/2 Mode	R/W	
		DV7CK = 0	DV7CK = 1			
		00	$fc/2^{13}$	$fs/2^5$		$fs/2^5$
		01	$fc/2^{12}$	$fs/2^4$		$fs/2^4$
		10	$fc/2^{11}$	$fs/2^3$		$fs/2^3$
11	$fc/2^{10}$	$fs/2^2$	$fs/2^2$			

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 6-2 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	
00	1.953 k	1.024 k	1.024 k
01	3.906 k	2.048 k	2.048 k
10	7.813 k	4.096 k	4.096 k
11	15.625 k	8.192 k	8.192 k

## 7. 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.

### 7.1 Watchdog Timer Configuration

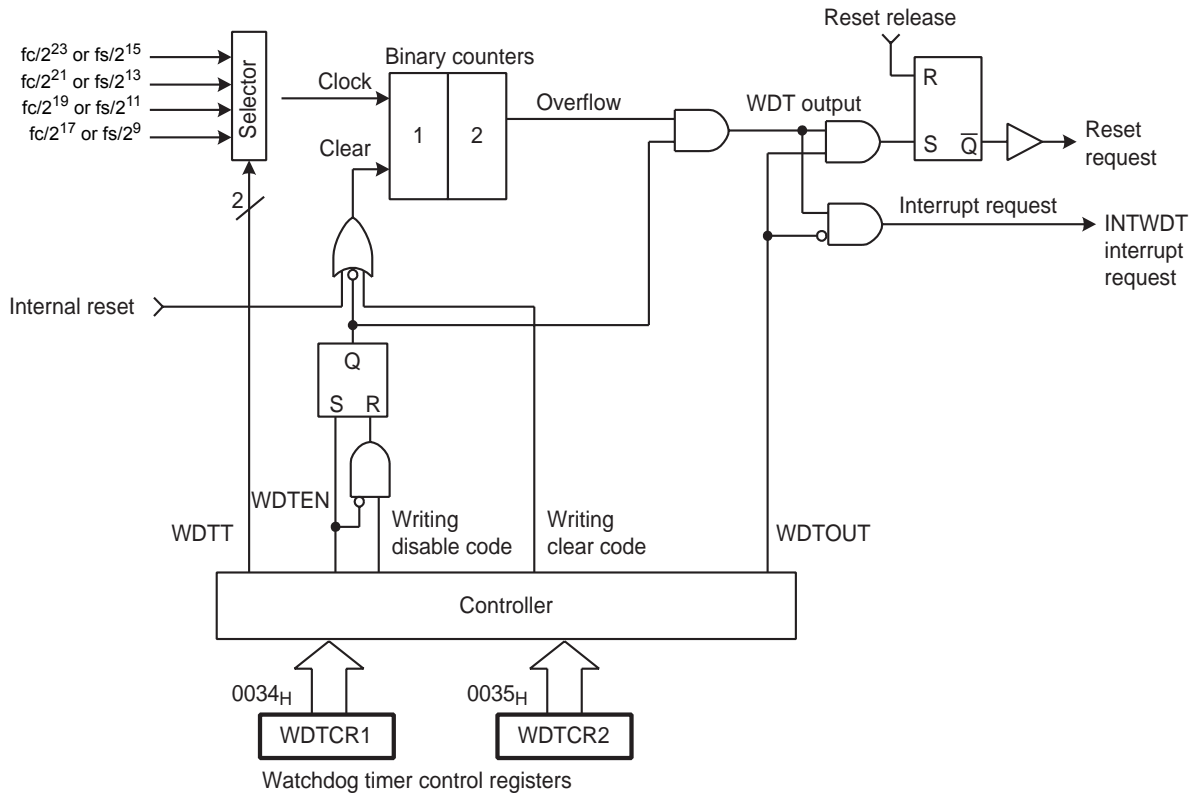


Figure 7-1 Watchdog Timer Configuration

## 7.2 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.

### 7.2.1 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.



### 7.2.3 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 7-1 Watchdog Timer Detection Time (Example: fc = 16.0 MHz, fs = 32.768 kHz)

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

### 7.2.4 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
```

### 7.2.5 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.

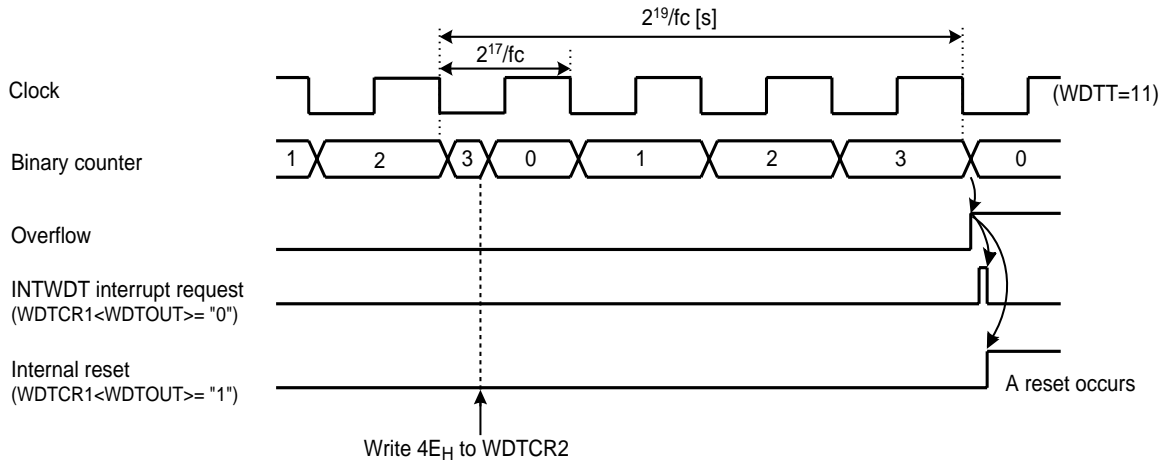


Figure 7-2 Watchdog Timer Interrupt

## 7.3 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

WDTCR1 (0034H)	7	6	5	4	3	2	1	0	
			ATAS	ATOUT	(WDTEN)	(WDTT)	(WDTOUT)		(Initial value: **11 1001)

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 (0035H)	7	6	5	4	3	2	1	0	
									(Initial value: **** ***)

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
--------	--	--	------------

#### 7.3.1 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>.

#### 7.3.2 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>.

#### 7.3.3 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.



### 7.3.4 Address Trap Reset

While WDTCR1<ATOUT> 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  $24/fc$  [s] ( $1.5 \mu\text{s}$  @  $fc = 16.0 \text{ MHz}$ ).

Note: When an address trap 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.



# 8. 18-Bit Timer/Counter (TC1)

## 8.1 Configuration

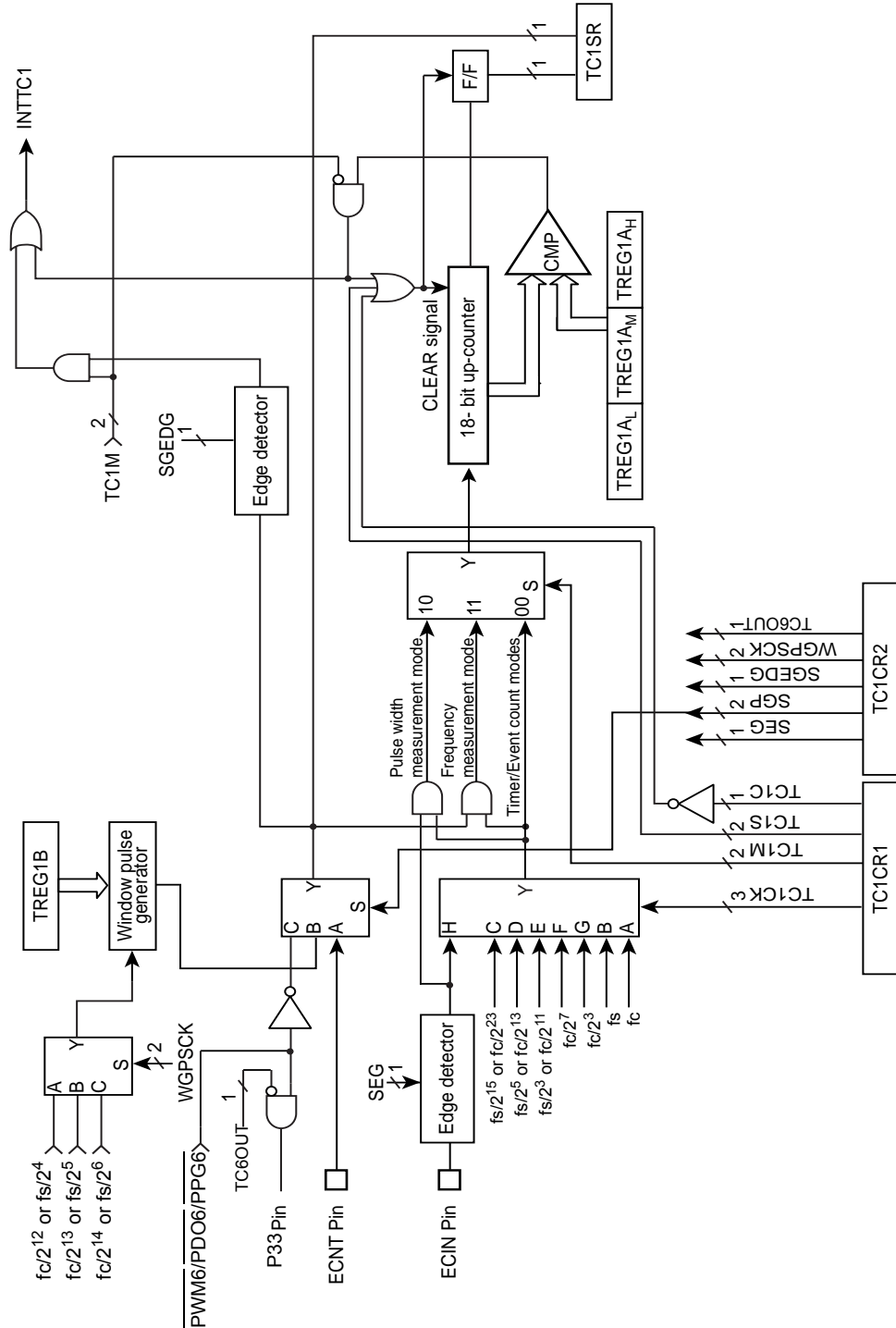
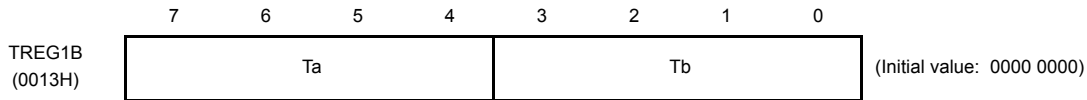
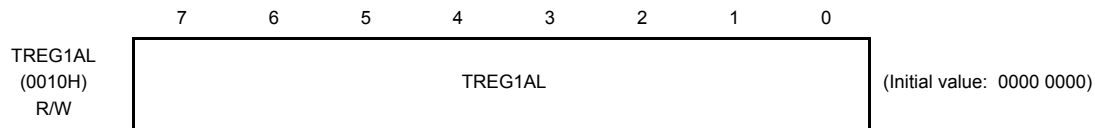
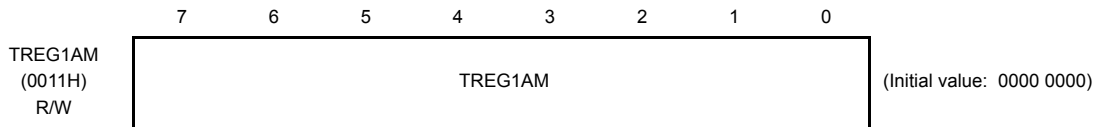
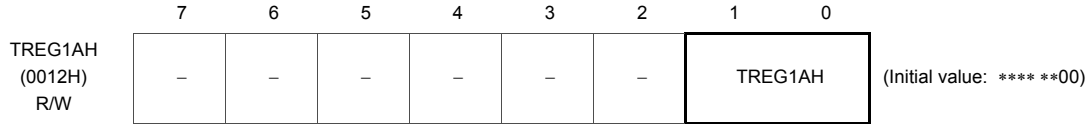


Figure 8-1 Timer/Counter1

## 8.2 Control

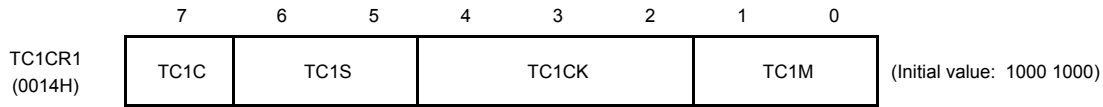
The Timer/counter 1 is controlled by timer/counter 1 control registers (TC1CR1/TC1CR2), an 18-bit timer register (TREG1A), and an 8-bit internal window gate pulse setting register (TREG1B).

### Timer register



		WGPSCK	NORMAL1/2, IDLE1/2 modes		SLOW1/2, SLEEP1/2 modes	R/W
			DV7CK=0	DV7CK=1		
Ta	Setting "H" level period of the window gate pulse	00	$(16 - Ta) \times 2^{12}/f_c$	$(16 - Ta) \times 2^4/f_s$	$(16 - Ta) \times 2^4/f_s$	
		01	$(16 - Ta) \times 2^{13}/f_c$	$(16 - Ta) \times 2^5/f_s$	$(16 - Ta) \times 2^5/f_s$	
		10	$(16 - Ta) \times 2^{14}/f_c$	$(16 - Ta) \times 2^6/f_s$	$(16 - Ta) \times 2^6/f_s$	
Tb	Setting "L" level period of the window gate pulse	00	$(16 - Tb) \times 2^{12}/f_c$	$(16 - Tb) \times 2^4/f_s$	$(16 - Tb) \times 2^4/f_s$	
		01	$(16 - Tb) \times 2^{13}/f_c$	$(16 - Tb) \times 2^5/f_s$	$(16 - Tb) \times 2^5/f_s$	
		10	$(16 - Tb) \times 2^{14}/f_c$	$(16 - Tb) \times 2^6/f_s$	$(16 - Tb) \times 2^6/f_s$	

### Timer/counter 1 control register 1



TC1C	Counter/overflow flag control	0:	Clear Counter/overflow flag ("1" is automatically set after clearing.)			R/W	
		1:	Not clear Counter/overflow flag				
TC1S	TC1 start control	00:	Stop and counter clear and overflow flag clear			R/W	
		10:	Start				
		*1:	Reserved				
TC1CK	TC1 source clock select		NORMAL 1/2, IDLE 1/2 modes		SLOW 1/2 mode	SLEEP 1/2 mode	
			DV7CK="0"	DV7CK="1"			
		000:	fc	fc	fc	fc	R/W
		001:	fs	fs	-	-	
		010:	$fc/2^{23}$	$fs/2^{15}$	$fs/2^{15}$	$fs/2^{15}$	
		011:	$fc/2^{13}$	$fs/2^5$	$fs/2^5$	$fs/2^5$	
		100:	$fc/2^{11}$	$fs/2^3$	$fs/2^3$	$fs/2^3$	
		101:	$fc/2^7$	$fc/2^7$	-	-	
110:	$fc/2^3$	$fc/2^3$	-	-			
111:	External clock (ECIN pin input)						
TC1M	TC1 mode select	00:	Timer/Event counter mode			R/W	
		01:	Reserved				
		10:	Pulse width measurement mode				
		11:	Frequency measurement mode				

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

Note 2: Writing to the low-byte of the timer register 1A (TREG1AL, TREG1AM), the compare function is inhibited until the high-byte (TREG1AH) is written.

Note 3: Set the mode and source clock, and edge (selection) when the TC1 stops (TC1S=00).

Note 4: "fc" can be selected as the source clock only in the timer mode during SLOW mode and in the pulse width measurement mode during NORMAL 1/2 or IDLE 1/2 mode.

Note 5: When a read instruction is executed to the timer register (TREG1A), the counter immediate value, not the register set value, is read out. Therefore it is impossible to read out the written value of TREG1A. To read the counter value, the read instruction should be executed when the counter stops to avoid reading unstable value.

Note 6: Set the timer register (TREG1A) to  $\geq 1$ .

Note 7: When using the timer mode and pulse width measurement mode, set TC1CK (TC1 source clock select) to internal clock.

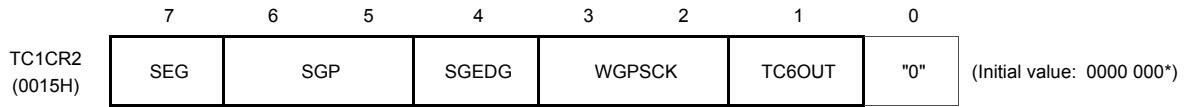
Note 8: When using the event counter mode, set TC1CK (TC1 source clock select) to external clock.

Note 9: Because the read value is different from the written value, do not use read-modify-write instructions to TREG1A.

Note 10:  $fc/2^7$ ,  $fc/2^3$  can not be used as source clock in SLOW/SLEEP mode.

Note 11: The read data of bits 7 to 2 in TREG1AH are always "0". (Data "1" can not be written.)

Timer/Counter 1 control register 2



SEG	External input clock (ECIN) edge select	0: Counts at the falling edge 1: Counts at the both (falling/rising) edges	R/W				
SGP	Window gate pulse select	00: ECNT input 01: Internal window gate pulse (TREG1B) 10: PWM6/PDO6/PPG6 (TC6)output 11: Reserved	R/W				
SGEDG	Window gate pulse interrupt edge select	0: Interrupts at the falling edge 1: Interrupts at the falling/rising edges					
WGPSCK	Window gate pulse source clock select						
			NORMAL1/2, IDLE1/2 modes	SLOW1/2 mode	SLEEP1/2 mode	R/W	
			DV7CK="0"	DV7CK="1"			
		00:	$2^{12}/fc$	$2^4/fs$	$2^4/fs$		$2^4/fs$
		01:	$2^{13}/fc$	$2^5/fs$	$2^5/fs$		$2^5/fs$
10:	$2^{14}/fc$	$2^6/fs$	$2^6/fs$	$2^6/fs$			
11:	Reserved	Reserved	Reserved	Reserved			
TC6OUT	TC6 output ( $\overline{PWM6/PDO6/PPG6}$ ) external output select	0: Output to P33 1: No output to P33	R/W				

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

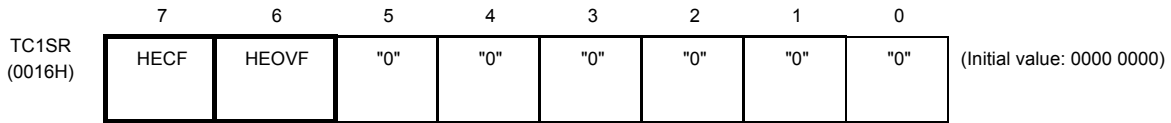
Note 2: Set the mode, source clock, and edge (selection) when the TC1 stops (TC1S = 00).

Note 3: If there is no need to use  $\overline{PWM6/PDO6/PPG6}$  as window gate pulse of TC1 always write "0" to TC6OUT.

Note 4: Make sure to write TC1CR2 "0" to bit 0 in TC1CR2.

Note 5: When using the event counter mode or pulse width measurement mode, set SEG to "0".

TC1 status register



HECF	Operating Status monitor	0: Stop (during Tb) or disable 1: Under counting (during Ta)	Read only
HEOVF	Counter overflow monitor	0: No overflow 1: Overflow status	

**8.3 Function**

TC1 has four operating modes. The timer mode of the TC1 is used at warm-up when switching from SLOW mode to NORMAL2 mode.

**8.3.1 Timer mode**

In this mode, counting up is performed using the internal clock. The contents of TREGIA are compared with the contents of up-counter. If a match is found, an INTTC1 interrupt is generated, and the counter is cleared. Counting up resumes after the counter is cleared.

Table 8-1 Source clock (internal clock) of Timer/Counter 1

Source Clock				Resolution		Maximum Time Setting	
NORMAL1/2, IDLE1/2 Mode		SLOW Mode	SLEEP Mode	fc = 16 MHz	fs =32.768 kHz	fc = 16 MHz	fs =32.768 kHz
DV7CK = 0	DV7CK = 1						
$fc/2^{23}$ [Hz]	$fs/2^{15}$ [Hz]	$fs/2^{15}$ [Hz]	$fs/2^{15}$ [Hz]	0.52 s	1 s	38.2 h	72.8 h
$fc/2^{13}$	$fs/2^5$	$fs/2^5$	$fs/2^5$	512 ms	0.98 ms	2.2 min	4.3 min
$fc/2^{11}$	$fs/2^3$	$fs/2^3$	$fs/2^3$	128 ms	244 ms	0.6 min	1.07 min
$fc/2^7$	$fc/2^7$	-	-	8 ms	-	2.1 s	-
$fc/2^3$	$fc/2^3$	-	-	0.5 ms	-	131.1 ms	-
fc	fc	fc (Note)	-	62.5 ns	-	16.4 ms	-
fs	fs	-	-	-	30.5 ms	-	8 s

Note: When fc is selected for the source clock in SLOW mode, the lower bits 11 of TREG1A is invalid, and a match of the upper bits 7 makes interrupts.

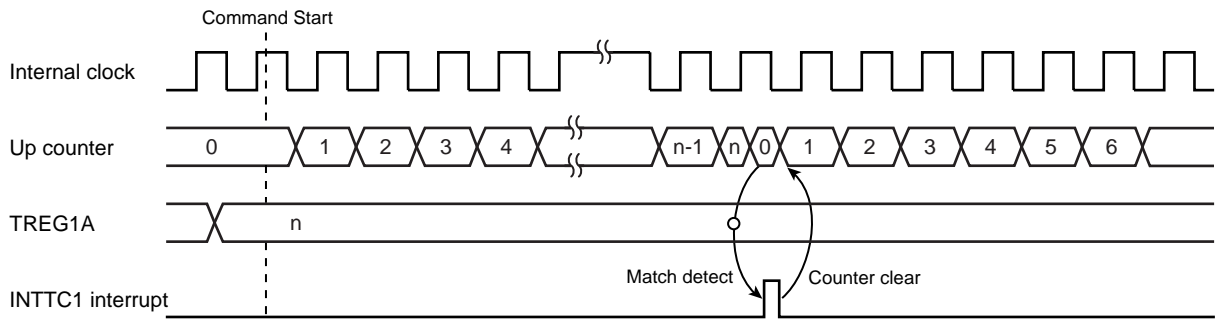


Figure 8-2 Timing chart for timer mode

### 8.3.2 Event Counter mode

It is a mode to count up at the falling edge of the ECIN pin input. When using this mode, set TC1CR1<TC1CK> to the external clock and then set TC1CR2<SEG> to "0" (Both edges can not be used).

The countents of TREG1A are compared with the contents of up-counter. If a match is found, an INTTC1 interrupt is generated, and the counter is cleared. Counting up resumes for ECIN pin input edge each after the counter is cleared.

The maximum applied frequency is  $f_c/2^4$  [Hz] in NORMAL 1/2 or IDLE 1/2 mode and  $f_s/2^4$  [Hz] in SLOW or SLEEP mode. Two or more machine cycles are required for both the "H" and "L" levels of the pulse width.

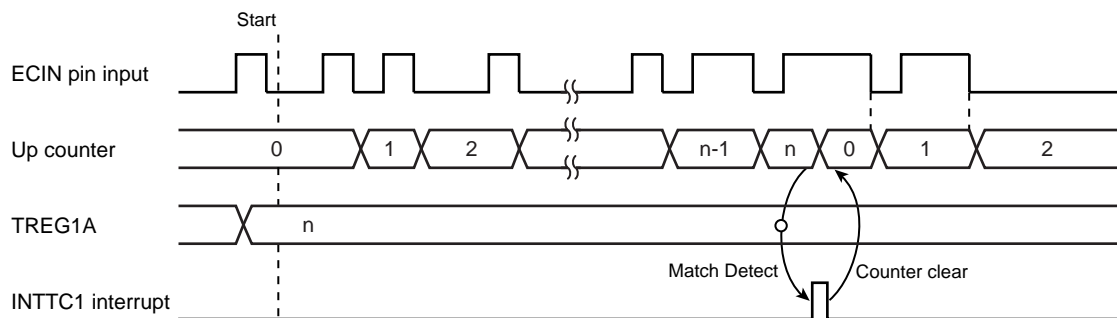


Figure 8-3 Event counter mode timing chart



### 8.3.3 Pulse Width Measurement mode

In this mode, pulse widths are counted on the falling edge of logical AND-ed pulse between ECIN pin input (window pulse) and the internal clock. When using this mode, set TC1CR1<TC1CK> to suitable internal clock and then set TC1CR2<SEG> to "0" (Both edges can not be used).

An INTTC1 interrupt is generated when the ECIN input detects the falling edge of the window pulse or both rising and falling edges of the window pulse, that can be selected by TC1CR2<SGEDG>.

The contents of TREG1A should be read while the count is stopped (ECIN pin is low), then clear the counter using TC1CR1<TC1C> (Normally, execute these process in the interrupt program).

When the counter is not cleared by TC1CR1<TC1C>, counting-up resumes from previous stopping value. When up counter is counted up from 3FFFFH to 00000H, an overflow occurs. At that time, TC1SR<HEOVF> is set to "1". TC1SR<HEOVF> remains the previous data until the counter is required to be cleared by TC1CR1<TC1C>.

Note: In pulse width measurement mode, if TC1CR1<TC1S> is written to "00" while ECIN input is "1", INTTC1 interrupt occurs. According to the following step, when timer counter is stopped, INTTC1 interrupt latch should be cleared to "0".

Example :

```

TC1STOP :
    |          |
    DI                      ; Clear IMF
    CLR          (EIRL). 7    ; Clear bit7 of EIRL
    LD           (TC1CR1), 00011010B ; Stop timer couer 1
    LD           (ILL), 01111111B   ; Clear bit7 of ILL
    SET          (EIRL). 7    ; Set bit7 of EIRL
    EI                      ; Set IMF
    |          |
    
```

Note 1: When SGEDG (window gate pulse interrupt edge select) is set to both edges and ECIN pin input is "1" in the pulse width measurement mode, an INTTC1 interrupt is generated by setting TC1S (TC1 start control) to "10" (start).

Note 2: In the pulse width measurement mode, HECF (operating status monitor) cannot used.

Note 3: Because the up counter is counted on the falling edge of logical AND-ed pulse (between ECIN pin input and the internal clock), if ECIN input becomes falling edge while internal source clock is "H" level, the up counter stops plus "1".

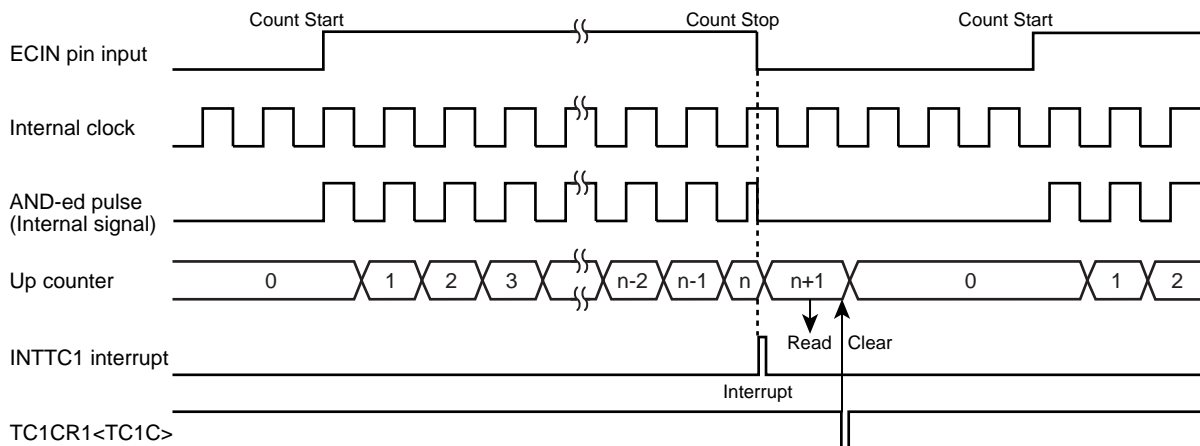


Figure 8-4 Pulse width measurement mode timing chart

### 8.3.4 Frequency Measurement mode

In this mode, the frequency of ECIN pin input pulse is measured. When using this mode, set TC1CR1<TC1CK> to the external clock.

The edge of the ECIN input pulse is counted during “H” level of the window gate pulse selected by TC1CR2<SGP>. To use ECNT input as a window gate pulse, TC1CR2<SGP> should be set to “00”.

An INTTC1 interrupt is generated on the falling edge or both the rising/falling edges of the window gate pulse, that can be selected by TC1CR2<SGEDG>. In the interrupt service program, read the contents of TREG1A while the count is stopped (window gate pulse is low), then clear the counter using TC1CR1<TC1C>. When the counter is not cleared, counting up resumes from previous stopping value.

The window pulse status can be monitored by TC1SR<HECF>.

When up counter is counted up from 3FFFFH to 00000H, an overflow occurs. At that time, TC1SR<HEOVF> is set to “1”. TC1SR<HEOVF> remains the previous data until the counter is required to be cleared by TC1CR1<TC1C>.

Using TC6 output ( $\overline{\text{PWM6/PDO6/PPG6}}$ ) for the window gate pulse, external output of  $\overline{\text{PWM6/PDO6/PPG6}}$  to P33 can be controlled using TC1CR2<TC6OUT>. Zero-clearing TC1CR2<TC6OUT> outputs  $\overline{\text{PWM6/PDO6/PPG6}}$  to P33; setting 1 in TC1CR2<TC6OUT> does not output  $\overline{\text{PWM6/PDO6/PPG6}}$  to P33. (TC1CR2<TC6OUT> is used to control output to P33 only. Thus, use the timer counter 6 control register to operate/stop  $\overline{\text{PWM6/PDO6/PPG6}}$ .)

When the internal window gate pulse is selected, the window gate pulse is set as follows.

Table 8-2 Internal window gate pulse setting time

		WGPSCK	NORMAL1/2, IDLE1/2 modes		SLOW1/2, SLEEP1/2 modes	R/W
			DV7CK=0	DV7CK=1		
Ta	Setting "H" level period of the window gate pulse	00	$(16 - Ta) \times 2^{12}/f_c$	$(16 - Ta) \times 2^4/f_s$	$(16 - Ta) \times 2^4/f_s$	
		01	$(16 - Ta) \times 2^{13}/f_c$	$(16 - Ta) \times 2^5/f_s$	$(16 - Ta) \times 2^5/f_s$	
		10	$(16 - Ta) \times 2^{14}/f_c$	$(16 - Ta) \times 2^6/f_s$	$(16 - Ta) \times 2^6/f_s$	
Tb	Setting "L" level period of the window gate pulse	00	$(16 - Tb) \times 2^{12}/f_c$	$(16 - Tb) \times 2^4/f_s$	$(16 - Tb) \times 2^4/f_s$	
		01	$(16 - Tb) \times 2^{13}/f_c$	$(16 - Tb) \times 2^5/f_s$	$(16 - Tb) \times 2^5/f_s$	
		10	$(16 - Tb) \times 2^{14}/f_c$	$(16 - Tb) \times 2^6/f_s$	$(16 - Tb) \times 2^6/f_s$	

The internal window gate pulse consists of “H” level period (Ta) that is counting time and “L” level period (Tb) that is counting stop time. Ta or Tb can be individually set by TREG1B. One cycle contains Ta + Tb.

Note 1: Because the internal window gate pulse is generated in synchronization with the internal divider, it may be delayed for a maximum of one cycle of the source clock (WGPSCK) immediately after start of the timer.

Note 2: Set the internal window gate pulse when the timer counter is not operating or during the Tb period. When Tb is overwritten during the Tb period, the update is valid from the next Tb period.

Note 3: In case of TC1CR2<SEG> = "1", if window gate pulse becomes falling edge, the up counter stops plus "1" regardless of ECIN input level. Therefore, if ECIN is always "H" or "L" level, count value becomes "1".

Note 4: In case of TC1CR2<SEG> = "0", because the up counter is counted on the falling edge of logical AND-ed pulse (between ECIN pin input and window gate pulse), if window gate pulse becomes falling edge while ECIN input is "H" level, the up counter stops plus "1". Therefore, if ECIN input is always "H" level, count value becomes "1".

Table 8-3 Table Setting Ta and Tb (WGPSCK = 10, fc = 16 MHz)

Setting Value	Setting time	Setting Value	Setting time
0	16.38ms	8	8.19ms
1	15.36ms	9	7.17ms
2	14.34ms	A	6.14ms
3	13.31ms	B	5.12ms
4	12.29ms	C	4.10ms
5	11.26ms	D	3.07ms
6	10.24ms	E	2.05ms
7	9.22ms	F	1.02ms

Table 8-4 Table Setting Ta and Tb (WGPSCK = 10, fs = 32.768 kHz)

Setting Valuen	Setting time	Setting Value	Setting time
0	31.25ms	8	15.63ms
1	29.30ms	9	13.67ms
2	27.34ms	A	11.72ms
3	25.39ms	B	9.77ms
4	23.44ms	C	7.81ms
5	21.48ms	D	5.86ms
6	19.53ms	E	3.91ms
7	17.58ms	F	1.95ms

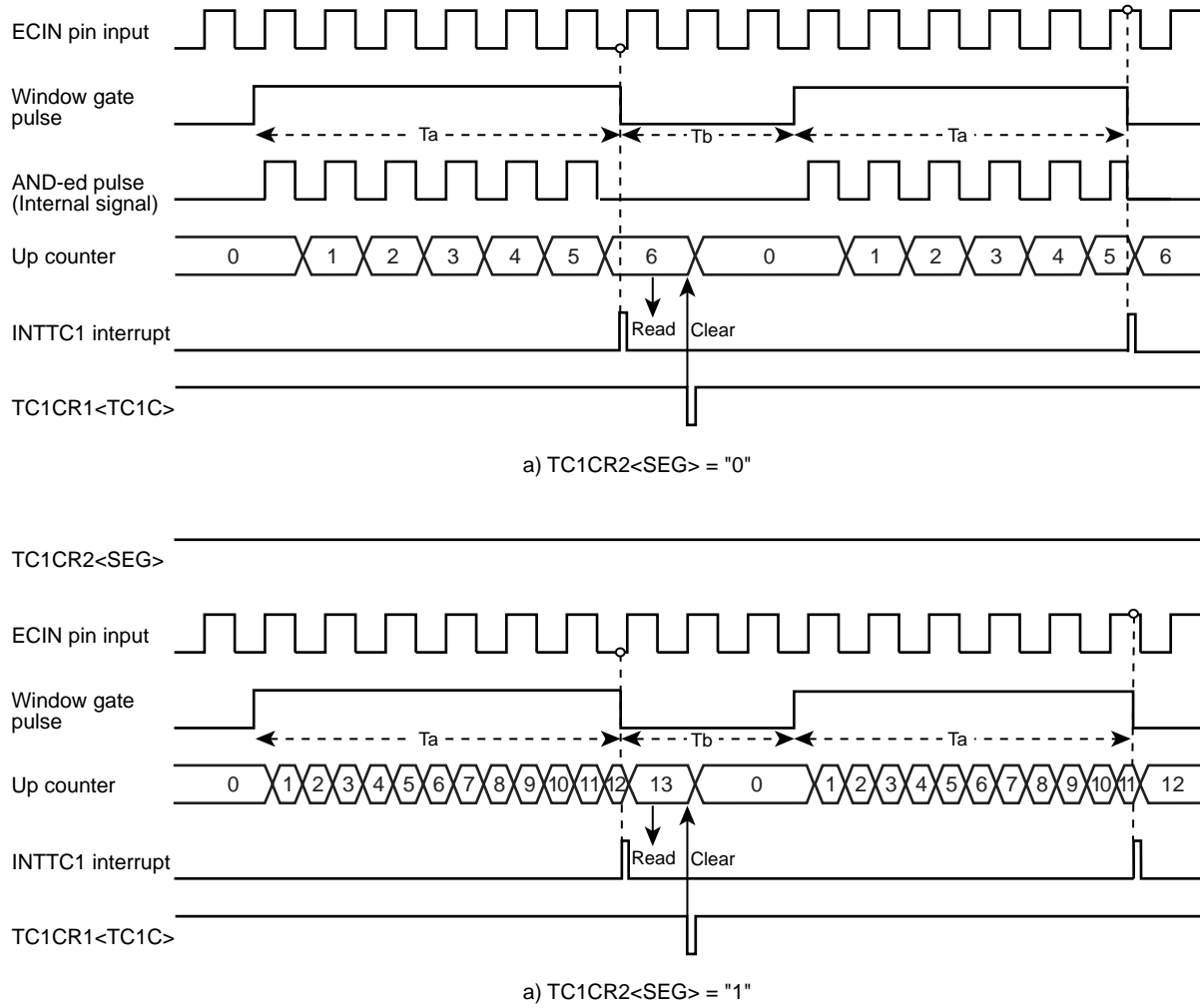


Figure 8-5 Timing chart for the frequency measurement mode (Window gate pulse falling interrupt)

# 9. 8-Bit TimerCounter (TC3, TC4)

## 9.1 Configuration

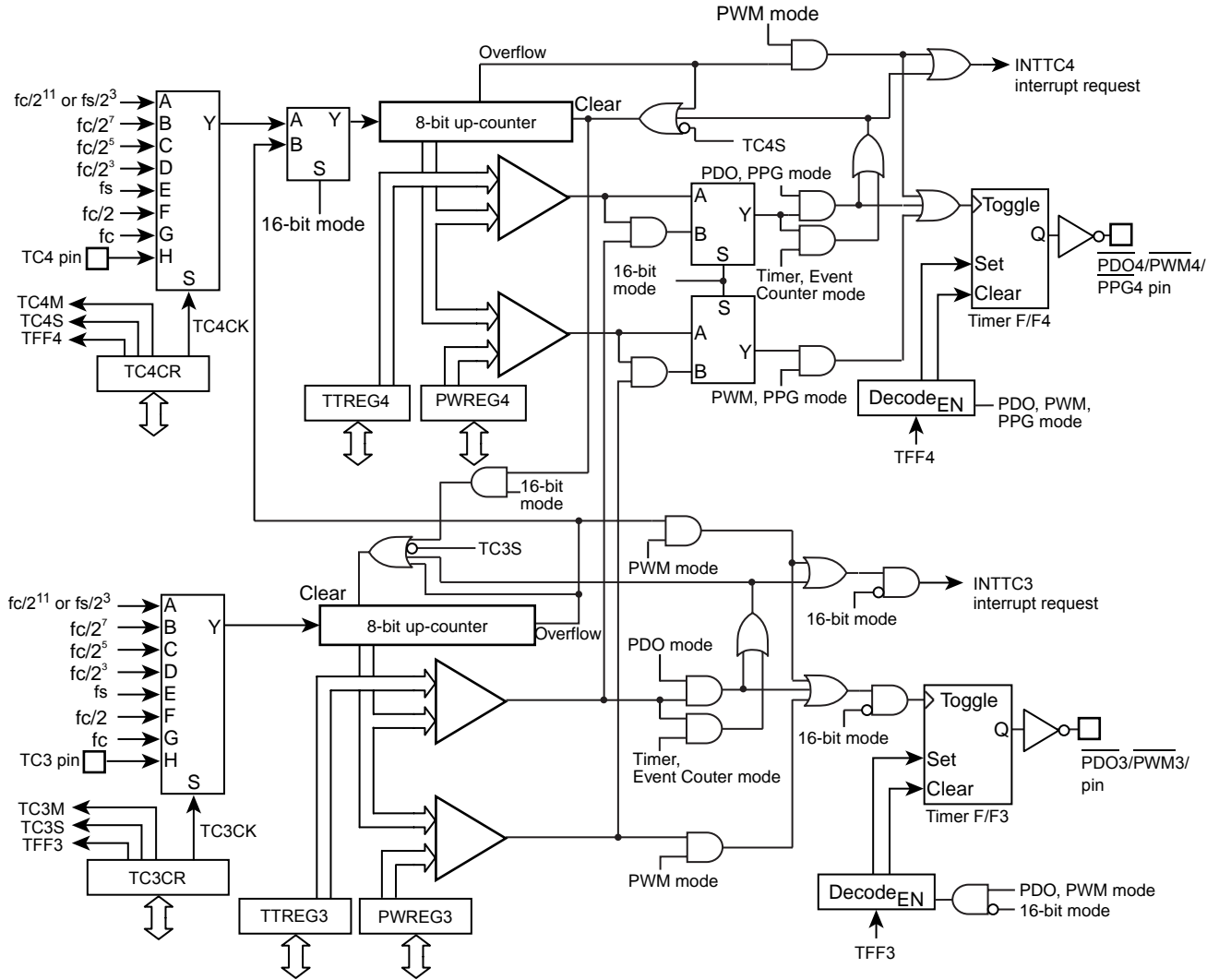
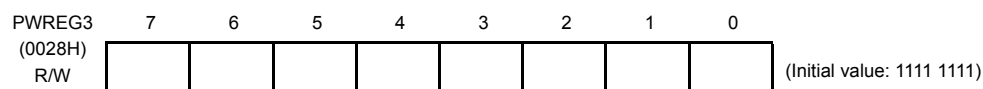
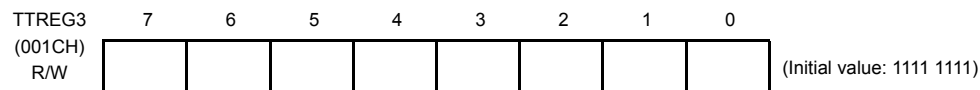


Figure 9-1 8-Bit TimerCounter 3, 4

## 9.2 TimerCounter Control

The TimerCounter 3 is controlled by the TimerCounter 3 control register (TC3CR) and two 8-bit timer registers (TTREG3, PWREG3).

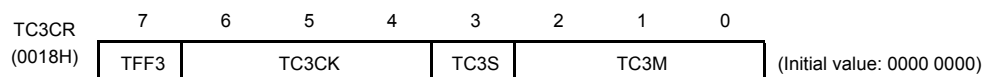
### TimerCounter 3 Timer Register



Note 1: Do not change the timer register (TTREG3) setting while the timer is running.

Note 2: Do not change the timer register (PWREG3) setting in the operating mode except the 8-bit and 16-bit PWM modes while the timer is running.

### TimerCounter 3 Control Register



TFF3	Time F/F3 control	0: Clear 1: Set			R/W	
TC3CK	Operating clock selection [Hz]	NORMAL1/2, IDLE1/2 mode		SLOW1/2 SLEEP1/2 mode	R/W	
		DV7CK = 0	DV7CK = 1			
		000	$fc/2^{11}$	$fs/2^3$		$fs/2^3$
		001	$fc/2^7$	$fc/2^7$		–
		010	$fc/2^5$	$fc/2^5$		–
		011	$fc/2^3$	$fc/2^3$		–
		100	fs	fs		fs
		101	$fc/2$	$fc/2$		–
110	fc	fc	fc (Note 8)			
111	TC3 pin input					
TC3S	TC3 start control	0: Operation stop and counter clear 1: Operation start			R/W	
TC3M	TC3M operating mode select	000: 8-bit timer/event counter mode 001: 8-bit programmable divider output (PDO) mode 010: 8-bit pulse width modulation (PWM) output mode 011: 16-bit mode (Each mode is selectable with TC4M.) 1**: Reserved			R/W	

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

Note 2: Do not change the TC3M, TC3CK and TFF3 settings while the timer is running.

Note 3: To stop the timer operation (TC3S= 1 → 0), do not change the TC3M, TC3CK and TFF3 settings. To start the timer operation (TC3S= 0 → 1), TC3M, TC3CK and TFF3 can be programmed.

Note 4: To use the TimerCounter in the 16-bit mode, set the operating mode by programming TC4CR<TC4M>, where TC3M must be fixed to 011.

Note 5: To use the TimerCounter in the 16-bit mode, select the source clock by programming TC3CK. Set the timer start control and timer F/F control by programming TC4CR<TC4S> and TC4CR<TFF4>, respectively.

Note 6: The operating clock settings are limited depending on the timer operating mode. For the detailed descriptions, see Table 9-1 and Table 9-2.

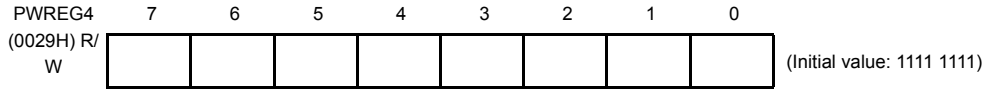
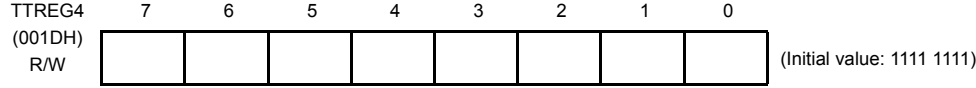
Note 7: The timer register settings are limited depending on the timer operating mode. For the detailed descriptions, see Table 9-3.

Note 8: The operating clock  $f_c$  in the SLOW or SLEEP mode can be used only as the high-frequency warm-up mode.



The TimerCounter 4 is controlled by the TimerCounter 4 control register (TC4CR) and two 8-bit timer registers (TTREG4 and PWREG4).

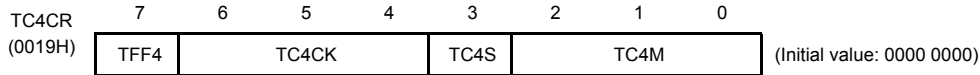
TimerCounter 4 Timer Register



Note 1: Do not change the timer register (TTREG4) setting while the timer is running.

Note 2: Do not change the timer register (PWREG4) setting in the operating mode except the 8-bit and 16-bit PWM modes while the timer is running.

TimerCounter 4 Control Register



TFF4	Timer F/F4 control	0: Clear 1: Set			R/W	
TC4CK	Operating clock selection [Hz]	NORMAL 1/2, IDLE 1/2 mode		SLOW 1/2 SLEEP 1/2 mode	R/W	
		DV7CK = 0	DV7CK = 1			
		000	$fc/2^{11}$	$fs/2^3$		$fs/2^3$
		001	$fc/2^7$	$fc/2^7$		–
		010	$fc/2^5$	$fc/2^5$		–
		011	$fc/2^3$	$fc/2^3$		–
		100	fs	fs		fs
		101	$fc/2$	$fc/2$		–
110	fc	fc	–			
111	TC4 pin input					
TC4S	TC4 start control	0: Operation stop and counter clear 1: Operation start			R/W	
TC4M	TC4M operating mode select	000: 8-bit timer/event counter mode 001: 8-bit programmable divider output (PDO) mode 010: 8-bit pulse width modulation (PWM) output mode 011: Reserved 100: 16-bit timer/event counter mode 101: Warm-up counter mode 110: 16-bit pulse width modulation (PWM) output mode 111: 16-bit PPG mode			R/W	

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

Note 2: Do not change the TC4M, TC4CK and TFF4 settings while the timer is running.

Note 3: To stop the timer operation (TC4S= 1 → 0), do not change the TC4M, TC4CK and TFF4 settings. To start the timer operation (TC4S= 0 → 1), TC4M, TC4CK and TFF4 can be programmed.

Note 4: When TC4M= 1\*\* (upper byte in the 16-bit mode), the source clock becomes the TC4 overflow signal regardless of the TC3CK setting.

Note 5: To use the TimerCounter in the 16-bit mode, select the operating mode by programming TC4M, where TC3CR<TC3 M> must be set to 011.



Note 6: To the TimerCounter in the 16-bit mode, select the source clock by programming TC3CR<TC3CK>. Set the timer start control and timer F/F control by programming TC4S and TFF4, respectively.

Note 7: The operating clock settings are limited depending on the timer operating mode. For the detailed descriptions, see Table 9-1 and Table 9-2.

Note 8: The timer register settings are limited depending on the timer operating mode. For the detailed descriptions, see Table 9-3.

Table 9-1 Operating Mode and Selectable Source Clock (NORMAL1/2 and IDLE1/2 Modes)

Operating mode	fc/2 <sup>11</sup> or fs/2 <sup>3</sup>	fc/2 <sup>7</sup>	fc/2 <sup>5</sup>	fc/2 <sup>3</sup>	fs	fc/2	fc	TC3 pin input	TC4 pin input
8-bit timer	○	○	○	○	–	–	–	–	–
8-bit event counter	–	–	–	–	–	–	–	○	○
8-bit PDO	○	○	○	○	–	–	–	–	–
8-bit PWM	○	○	○	○	○	○	○	–	–
16-bit timer	○	○	○	○	–	–	–	–	–
16-bit event counter	–	–	–	–	–	–	–	○	–
Warm-up counter	–	–	–	–	○	–	–	–	–
16-bit PWM	○	○	○	○	○	○	○	○	–
16-bit PPG	○	○	○	○	–	–	–	○	–

Note 1: For 16-bit operations (16-bit timer/event counter, warm-up counter, 16-bit PWM and 16-bit PPG), set its source clock on lower bit (TC3CK).

Note 2: ○ : Available source clock

Table 9-2 Operating Mode and Selectable Source Clock (SLOW1/2 and SLEEP1/2 Modes)

Operating mode	fc/2 <sup>11</sup> or fs/2 <sup>3</sup>	fc/2 <sup>7</sup>	fc/2 <sup>5</sup>	fc/2 <sup>3</sup>	fs	fc/2	fc	TC3 pin input	TC4 pin input
8-bit timer	○	–	–	–	–	–	–	–	–
8-bit event counter	–	–	–	–	–	–	–	○	○
8-bit PDO	○	–	–	–	–	–	–	–	–
8-bit PWM	○	–	–	–	○	–	–	–	–
16-bit timer	○	–	–	–	–	–	–	–	–
16-bit event counter	–	–	–	–	–	–	–	○	–
Warm-up counter	–	–	–	–	–	–	○	–	–
16-bit PWM	○	–	–	–	○	–	–	○	–
16-bit PPG	○	–	–	–	–	–	–	○	–

Note1: For 16-bit operations (16-bit timer/event counter, warm-up counter, 16-bit PWM and 16-bit PPG), set its source clock on lower bit (TC3CK).

Note2: ○ : Available source clock

Table 9-3 Constraints on Register Values Being Compared

Operating mode	Register Value
8-bit timer/event counter	$1 \leq (TTREGn) \leq 255$
8-bit PDO	$1 \leq (TTREGn) \leq 255$
8-bit PWM	$2 \leq (PWREGn) \leq 254$
16-bit timer/event counter	$1 \leq (TTREG4, 3) \leq 65535$
Warm-up counter	$256 \leq (TTREG4, 3) \leq 65535$
16-bit PWM	$2 \leq (PWREG4, 3) \leq 65534$
16-bit PPG	$1 \leq (PWREG4, 3) < (TTREG4, 3) \leq 65535$ and $(PWREG4, 3) + 1 < (TTREG4, 3)$

Note: n = 3 to 4

### 9.3 Function

The TimerCounter 3 and 4 have the 8-bit timer, 8-bit event counter, 8-bit programmable divider output (PDO), 8-bit pulse width modulation (PWM) output modes. The TimerCounter 3 and 4 (TC3, 4) are cascadable to form a 16-bit timer. The 16-bit timer has the operating modes such as the 16-bit timer, 16-bit event counter, warm-up counter, 16-bit pulse width modulation (PWM) output and 16-bit programmable pulse generation (PPG) modes.

#### 9.3.1 8-Bit Timer Mode (TC3 and 4)

In the timer mode, the up-counter counts up using the internal clock. When a match between the up-counter and the timer register  $j$  (TTREG $j$ ) value is detected, an INTTC $j$  interrupt is generated and the up-counter is cleared. After being cleared, the up-counter restarts counting.

Note 1: In the timer mode, fix TCjCR<TFFj> to 0. If not fixed, the  $\overline{PDOj}$ ,  $\overline{PWMj}$  and  $\overline{PPGj}$  pins may output pulses.

Note 2: In the timer mode, do not change the TTREG $j$  setting while the timer is running. Since TTREG $j$  is not in the shift register configuration in the timer mode, the new value programmed in TTREG $j$  is in effect immediately after the programming. Therefore, if TTREG $i$  is changed while the timer is running, an expected operation may not be obtained.

Note 3:  $j = 3, 4$

Table 9-4 Source Clock for TimerCounter 3, 4 (Internal Clock)

Source Clock		SLOW1/2, SLEEP1/2 mode	Resolution		Repeated Cycle	
NORMAL1/2, IDLE1/2 mode			$f_c = 16 \text{ MHz}$	$f_s = 32.768 \text{ kHz}$	$f_c = 16 \text{ MHz}$	$f_s = 32.768 \text{ kHz}$
DV7CK = 0	DV7CK = 1					
$f_c/2^{11}$ [Hz]	$f_s/2^3$ [Hz]	$f_s/2^3$ [Hz]	128 $\mu\text{s}$	244.14 $\mu\text{s}$	32.6 ms	62.3 ms
$f_c/2^7$	$f_c/2^7$	–	8 $\mu\text{s}$	–	2.0 ms	–
$f_c/2^5$	$f_c/2^5$	–	2 $\mu\text{s}$	–	510 $\mu\text{s}$	–
$f_c/2^3$	$f_c/2^3$	–	500 ns	–	127.5 $\mu\text{s}$	–

Example :Setting the timer mode with source clock  $f_c/2^7$  Hz and generating an interrupt 80  $\mu\text{s}$  later (TimerCounter4,  $f_c = 16.0 \text{ MHz}$ )

- LD (TTREG4), 0AH : Sets the timer register ( $80 \mu\text{s} \div 2^7 / f_c = 0\text{AH}$ ).
- DI
- SET (EIRH). 4 : Enables INTTC4 interrupt.
- EI
- LD (TC4CR), 0001000B : Sets the operating clock to  $f_c/2^7$ , and 8-bit timer mode.
- LD (TC4CR), 00011000B : Starts TC4.

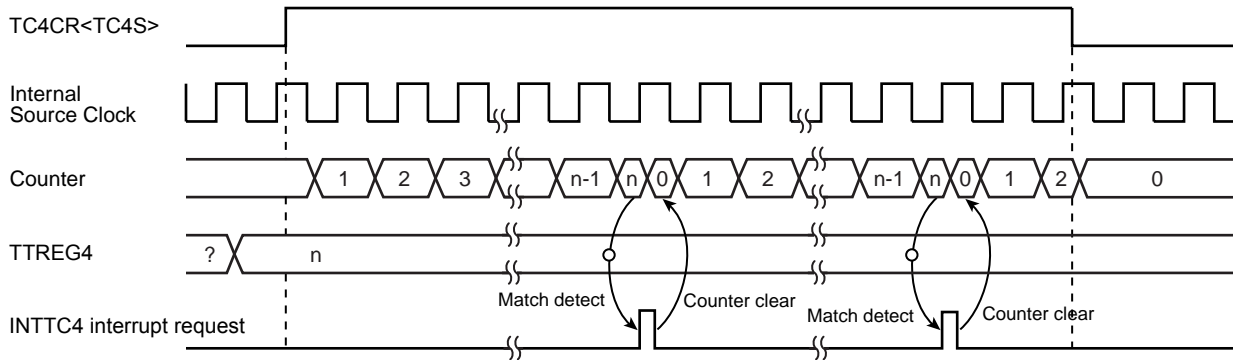


Figure 9-2 8-Bit Timer Mode Timing Chart (TC4)

### 9.3.2 8-Bit Event Counter Mode (TC3, 4)

In the 8-bit event counter mode, the up-counter counts up at the falling edge of the input pulse to the TCj pin. When a match between the up-counter and the TTREGj value is detected, an INTTCj interrupt is generated and the up-counter is cleared. After being cleared, the up-counter restarts counting at the falling edge of the input pulse to the TCj pin. Two machine cycles are required for the low- or high-level pulse input to the TCj pin. Therefore, a maximum frequency to be supplied is  $f_c/2^4$  Hz in the NORMAL1/2 or IDLE1/2 mode, and  $f_s/2^4$  Hz in the SLOW1/2 or SLEEP1/2 mode.

Note 1: In the event counter mode, fix TCjCR<TFFj> to 0. If not fixed, the  $\overline{PDOj}$ ,  $\overline{PWMj}$  and  $\overline{PPGj}$  pins may output pulses.

Note 2: In the event counter mode, do not change the TTREGj setting while the timer is running. Since TTREGj is not in the shift register configuration in the event counter mode, the new value programmed in TTREGj is in effect immediately after the programming. Therefore, if TTREGj is changed while the timer is running, an expected operation may not be obtained.

Note 3: j = 3, 4

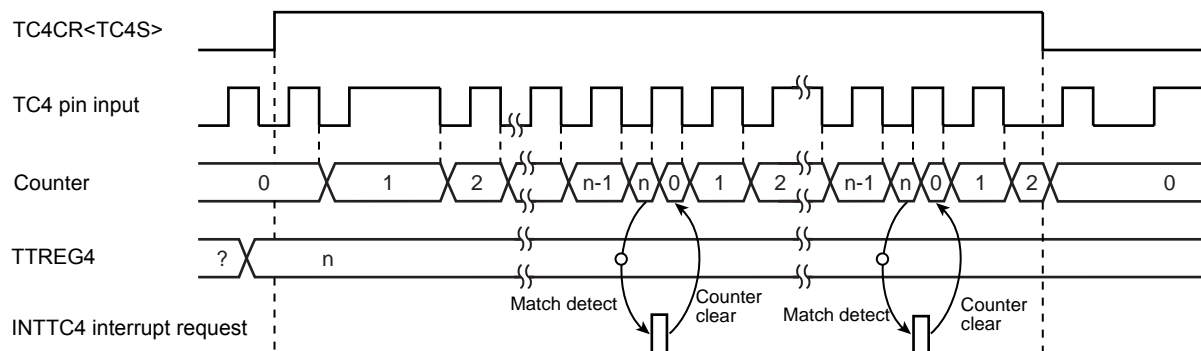


Figure 9-3 8-Bit Event Counter Mode Timing Chart (TC4)

### 9.3.3 8-Bit Programmable Divider Output (PDO) Mode (TC3, 4)

This mode is used to generate a pulse with a 50% duty cycle from the  $\overline{PDOj}$  pin.

In the PDO mode, the up-counter counts up using the internal clock. When a match between the up-counter and the TTREGj value is detected, the logic level output from the  $\overline{PDOj}$  pin is switched to the opposite state and the up-counter is cleared. The INTTCj interrupt request is generated at the time. The logic state opposite to the timer F/Fj logic level is output from the  $\overline{PDOj}$  pin. An arbitrary value can be set to the timer F/Fj by TCjCR<TFFj>. Upon reset, the timer F/Fj value is initialized to 0.

To use the programmable divider output, set the output latch of the I/O port to 1.

Example :Generating 1024 Hz pulse using TC4 ( $f_c = 16.0$  MHz)

Setting port		
LD	(TTREG4), 3DH	: $1/1024 \div 2^7 / f_c \div 2 = 3DH$
LD	(TC4CR), 00010001B	: Sets the operating clock to $f_c/2^7$ , and 8-bit PDO mode.
LD	(TC4CR), 00011001B	: Starts TC4.

Note 1: In the programmable divider output mode, do not change the TTREGj setting while the timer is running. Since TTREGj is not in the shift register configuration in the programmable divider output mode, the new value programmed in TTREGj is in effect immediately after programming. Therefore, if TTREGi is changed while the timer is running, an expected operation may not be obtained.

Note 2: When the timer is stopped during PDO output, the  $\overline{PDOj}$  pin holds the output status when the timer is stopped. To change the output status, program TCjCR<TFFj> after the timer is stopped. Do not change the TCjCR<TFFj> setting upon stopping of the timer.

Example: Fixing the  $\overline{PDOj}$  pin to the high level when the TimerCounter is stopped

CLR (TCjCR).3: Stops the timer.

CLR (TCjCR).7: Sets the  $\overline{PDOj}$  pin to the high level.

Note 3: j = 3, 4



### 9.3.4 8-Bit Pulse Width Modulation (PWM) Output Mode (TC3, 4)

This mode is used to generate a pulse-width modulated (PWM) signals with up to 8 bits of resolution. The up-counter counts up using the internal clock.

When a match between the up-counter and the PWREGj value is detected, the logic level output from the timer F/Fj is switched to the opposite state. The counter continues counting. The logic level output from the timer F/Fj is switched to the opposite state again by the up-counter overflow, and the counter is cleared. The INTTCj interrupt request is generated at this time.

Since the initial value can be set to the timer F/Fj by TCjCR<TFFj>, positive and negative pulses can be generated. Upon reset, the timer F/Fj is cleared to 0.

(The logic level output from the  $\overline{PWMj}$  pin is the opposite to the timer F/Fj logic level.)

Since PWREGj in the PWM mode is serially connected to the shift register, the value set to PWREGj can be changed while the timer is running. The value set to PWREGj during a run of the timer is shifted by the INTTCj interrupt request and loaded into PWREGj. While the timer is stopped, the value is shifted immediately after the programming of PWREGj. If executing the read instruction to PWREGj during PWM output, the value in the shift register is read, but not the value set in PWREGj. Therefore, after writing to PWREGj, the reading data of PWREGj is previous value until INTTCj is generated.

For the pin used for PWM output, the output latch of the I/O port must be set to 1.

Note 1: In the PWM mode, program the timer register PWREGj immediately after the INTTCj interrupt request is generated (normally in the INTTCj interrupt service routine.) If the programming of PWREGj and the interrupt request occur at the same time, an unstable value is shifted, that may result in generation of the pulse different from the programmed value until the next INTTCj interrupt request is generated.

Note 2: When the timer is stopped during PWM output, the  $\overline{PWMj}$  pin holds the output status when the timer is stopped. To change the output status, program TCjCR<TFFj> after the timer is stopped. Do not change the TCjCR<TFFj> upon stopping of the timer.

Example: Fixing the  $\overline{PWMj}$  pin to the high level when the TimerCounter is stopped

CLR (TCjCR).3: Stops the timer.

CLR (TCjCR).7: Sets the  $\overline{PWMj}$  pin to the high level.

Note 3: To enter the STOP mode during PWM output, stop the timer and then enter the STOP mode. If the STOP mode is entered without stopping the timer when fc, fc/2 or fs is selected as the source clock, a pulse is output from the  $\overline{PWMj}$  pin during the warm-up period time after exiting the STOP mode.

Note 4: j = 3, 4

Table 9-5 PWM Output Mode

Source Clock		SLOW1/2, SLEEP1/2 mode	Resolution		Repeated Cycle	
NORMAL1/2, IDLE1/2 mode			fc = 16 MHz	fs = 32.768 kHz	fc = 16 MHz	fs = 32.768 kHz
DV7CK = 0	DV7CK = 1					
$fc/2^{11}$ [Hz]	$fs/2^3$ [Hz]	$fs/2^3$ [Hz]	128 $\mu$ s	244.14 $\mu$ s	32.8 ms	62.5 ms
$fc/2^7$	$fc/2^7$	–	8 $\mu$ s	–	2.05 ms	–
$fc/2^5$	$fc/2^5$	–	2 $\mu$ s	–	512 $\mu$ s	–
$fc/2^3$	$fc/2^3$	–	500 ns	–	128 $\mu$ s	–
fs	fs	fs	30.5 $\mu$ s	30.5 $\mu$ s	7.81 ms	7.81 ms
fc/2	fc/2	–	125 ns	–	32 $\mu$ s	–
fc	fc	–	62.5 ns	–	16 $\mu$ s	–

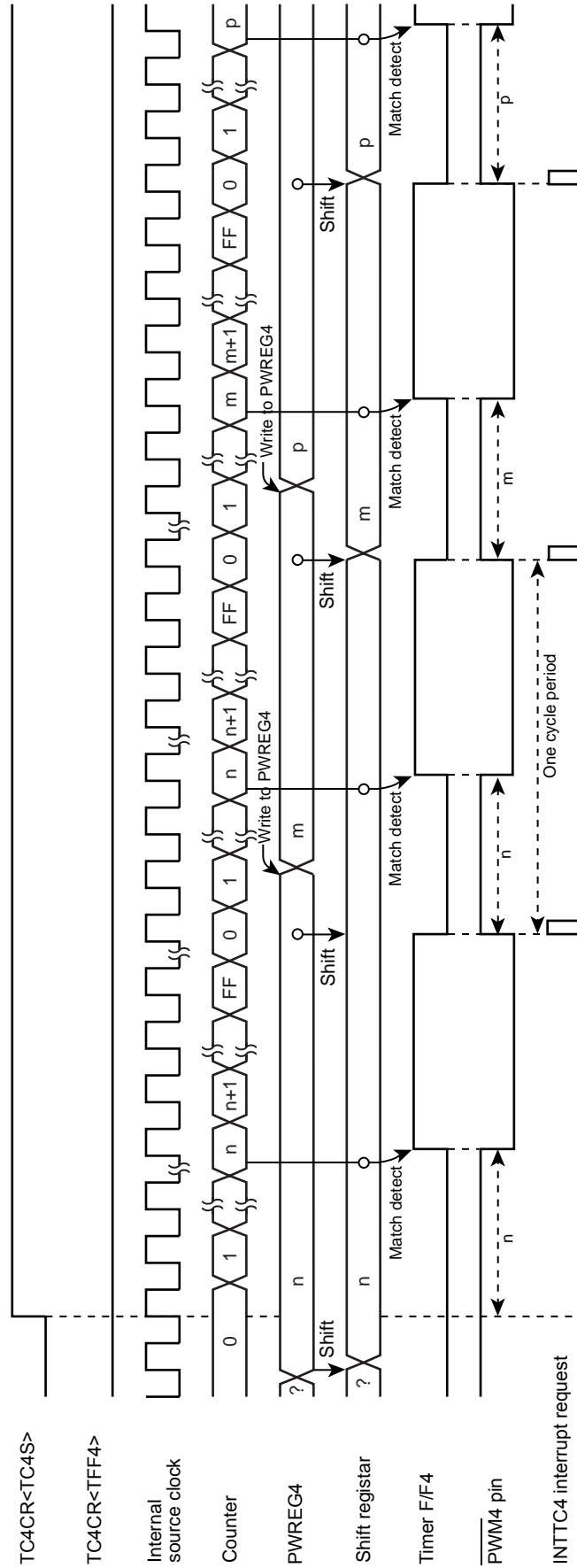


Figure 9-5 8-Bit PWM Mode Timing Chart (TC4)



### 9.3.5 16-Bit Timer Mode (TC3 and 4)

In the timer mode, the up-counter counts up using the internal clock. The TimerCounter 3 and 4 are cascadable to form a 16-bit timer.

When a match between the up-counter and the timer register (TTREG3, TTREG4) value is detected after the timer is started by setting TC4CR<TC4S> to 1, an INTTC4 interrupt is generated and the up-counter is cleared. After being cleared, the up-counter continues counting. Program the upper byte and lower byte in this order in the timer register. (Programming only the upper or lower byte should not be attempted.)

Note 1: In the timer mode, fix TCjCR<TFFj> to 0. If not fixed, the  $\overline{PDOj}$ ,  $\overline{PWMj}$ , and  $\overline{PPGj}$  pins may output a pulse.

Note 2: In the timer mode, do not change the TTREGj setting while the timer is running. Since TTREGj is not in the shift register configuration in the timer mode, the new value programmed in TTREGj is in effect immediately after programming of TTREGj. Therefore, if TTREGj is changed while the timer is running, an expected operation may not be obtained.

Note 3: j = 3, 4

Table 9-6 Source Clock for 16-Bit Timer Mode

Source Clock		SLOW1/2, SLEEP1/2 mode	Resolution		Repeated Cycle	
NORMAL1/2, IDLE1/2 mode	DV7CK = 0		DV7CK = 1	fc = 16 MHz	fs = 32.768 kHz	fc = 16 MHz
fc/2 <sup>11</sup>	fs/2 <sup>3</sup>	fs/2 <sup>3</sup>	128 μs	244.14 μs	8.39 s	16 s
fc/2 <sup>7</sup>	fc/2 <sup>7</sup>	–	8 μs	–	524.3 ms	–
fc/2 <sup>5</sup>	fc/2 <sup>5</sup>	–	2 μs	–	131.1 ms	–
fc/2 <sup>3</sup>	fc/2 <sup>3</sup>	–	500 ns	–	32.8 ms	–

Example :Setting the timer mode with source clock  $fc/2^7$  Hz, and generating an interrupt 300 ms later  
(fc = 16.0 MHz)

- LDW (TTREG3), 927CH : Sets the timer register (300 ms =  $2^7/fc = 927CH$ ).
- DI
- SET (EIRH), 4 : Enables INTTC4 interrupt.
- EI
- LD (TC3CR), 13H : Sets the operating clock to  $fc/2^7$ , and 16-bit timer mode (lower byte).
- LD (TC4CR), 04H : Sets the 16-bit timer mode (upper byte).
- LD (TC4CR), 0CH : Starts the timer.

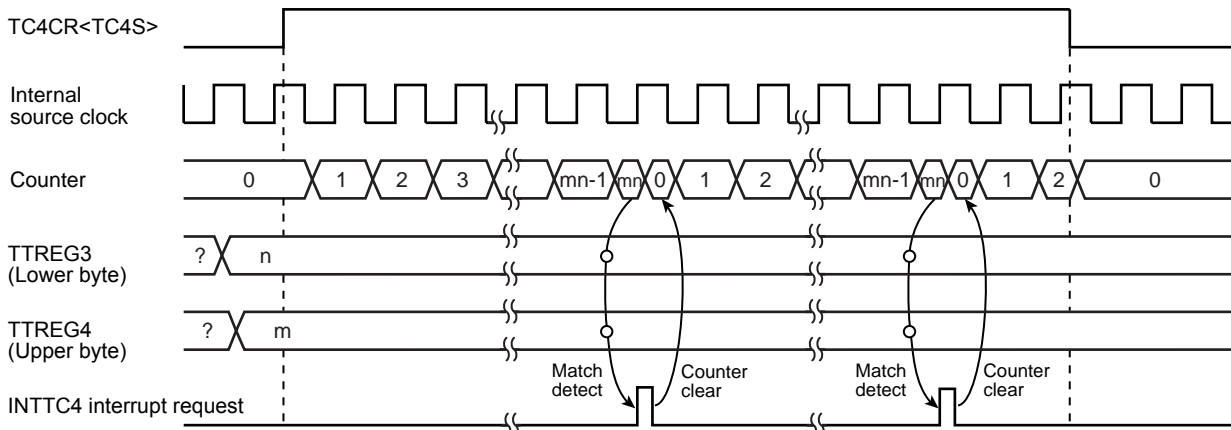


Figure 9-6 16-Bit Timer Mode Timing Chart (TC3 and TC4)

### 9.3.6 16-Bit Event Counter Mode (TC3 and 4)

In the event counter mode, the up-counter counts up at the falling edge to the TC3 pin. The TimerCounter 3 and 4 are cascadable to form a 16-bit event counter.

When a match between the up-counter and the timer register (TTREG3, TTREG4) value is detected after the timer is started by setting TC4CR<TC4S> to 1, an INTTC4 interrupt is generated and the up-counter is cleared.

After being cleared, the up-counter restarts counting at the falling edge of the input pulse to the TC3 pin. Two machine cycles are required for the low- or high-level pulse input to the TC3 pin.

Therefore, a maximum frequency to be supplied is  $f_c/2^4$  Hz in the NORMAL1 or IDLE1 mode, and  $f_s/2^4$  in the SLOW1/2 or SLEEP1/2 mode. Program the lower byte (TTREG3), and upper byte (TTREG4) in this order in the timer register. (Programming only the upper or lower byte should not be attempted.)

Note 1: In the event counter mode, fix TCjCR<TFFj> to 0. If not fixed, the  $\overline{PDOj}$ ,  $\overline{PWMj}$  and  $\overline{PPGj}$  pins may output pulses.

Note 2: In the event counter mode, do not change the TTREGj setting while the timer is running. Since TTREGj is not in the shift register configuration in the event counter mode, the new value programmed in TTREGj is in effect immediately after the programming. Therefore, if TTREGj is changed while the timer is running, an expected operation may not be obtained.

Note 3: j = 3, 4

### 9.3.7 16-Bit Pulse Width Modulation (PWM) Output Mode (TC3 and 4)

This mode is used to generate a pulse-width modulated (PWM) signals with up to 16 bits of resolution. The TimerCounter 3 and 4 are cascadable to form the 16-bit PWM signal generator.

The counter counts up using the internal clock or external clock.

When a match between the up-counter and the timer register (PWREG3, PWREG4) value is detected, the logic level output from the timer F/F4 is switched to the opposite state. The counter continues counting. The logic level output from the timer F/F4 is switched to the opposite state again by the counter overflow, and the counter is cleared. The INTTC4 interrupt is generated at this time.

Two machine cycles are required for the high- or low-level pulse input to the TC3 pin. Therefore, a maximum frequency to be supplied is  $f_c/2^4$  Hz in the NORMAL1 or IDLE1 mode, and  $f_s/2^4$  to in the SLOW1/2 or SLEEP1/2 mode.

Since the initial value can be set to the timer F/F4 by TC4CR<TFF4>, positive and negative pulses can be generated. Upon reset, the timer F/F4 is cleared to 0.

(The logic level output from the  $\overline{PWM4}$  pin is the opposite to the timer F/F4 logic level.)

Since PWREG4 and 3 in the PWM mode are serially connected to the shift register, the values set to PWREG4 and 3 can be changed while the timer is running. The values set to PWREG4 and 3 during a run of the timer are shifted by the INTTCj interrupt request and loaded into PWREG4 and 3. While the timer is stopped, the values are shifted immediately after the programming of PWREG4 and 3. Set the lower byte (PWREG3) and upper byte (PWREG3) in this order to program PWREG4 and 3. (Programming only the lower or upper byte of the register should not be attempted.)

If executing the read instruction to PWREG4 and 3 during PWM output, the values set in the shift register is read, but not the values set in PWREG4 and 3. Therefore, after writing to the PWREG4 and 3, reading data of PWREG4 and 3 is previous value until INTTC4 is generated.

For the pin used for PWM output, the output latch of the I/O port must be set to 1.

Note 1: In the PWM mode, program the timer register PWREG4 and 3 immediately after the INTTC4 interrupt request is generated (normally in the INTTC4 interrupt service routine.) If the programming of PWREGj and the interrupt request 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 generated.

Note 2: When the timer is stopped during PWM output, the  $\overline{PWM4}$  pin holds the output status when the timer is stopped. To change the output status, program TC4CR<TFF4> after the timer is stopped. Do not program TC4CR<TFF4> upon stopping of the timer.

Example: Fixing the  $\overline{PWM4}$  pin to the high level when the TimerCounter is stopped

CLR (TC4CR).3: Stops the timer.  
 CLR (TC4CR).7 : Sets the  $\overline{PWM4}$  pin to the high level.

Note 3: To enter the STOP mode, stop the timer and then enter the STOP mode. If the STOP mode is entered without stopping of the timer when  $f_c$ ,  $f_c/2$  or  $f_s$  is selected as the source clock, a pulse is output from the  $\overline{PWM4}$  pin during the warm-up period time after exiting the STOP mode.

Table 9-7 16-Bit PWM Output Mode

Source Clock		SLOW1/2, SLEEP1/2 mode	Resolution		Repeated Cycle	
NORMAL1/2, IDLE1/2 mode			$f_c = 16 \text{ MHz}$	$f_s = 32.768 \text{ kHz}$	$f_c = 16 \text{ MHz}$	$f_s = 32.768 \text{ kHz}$
DV7CK = 0	DV7CK = 1					
$f_c/2^{11}$	$f_s/2^3 \text{ [Hz]}$	$f_s/2^3 \text{ [Hz]}$	128 $\mu\text{s}$	244.14 $\mu\text{s}$	8.39 s	16 s
$f_c/2^7$	$f_c/2^7$	–	8 $\mu\text{s}$	–	524.3 ms	–
$f_c/2^5$	$f_c/2^5$	–	2 $\mu\text{s}$	–	131.1 ms	–
$f_c/2^3$	$f_c/2^3$	–	500ns	–	32.8 ms	–
$f_s$	$f_s$	$f_s$	30.5 $\mu\text{s}$	30.5 $\mu\text{s}$	2 s	2 s
$f_c/2$	$f_c/2$	–	125 ns	–	8.2 ms	–
$f_c$	$f_c$	–	62.5 ns	–	4.1 ms	–

Example :Generating a pulse with 1-ms high-level width and a period of 32.768 ms ( $f_c = 16.0 \text{ MHz}$ )

Setting ports

- LDW (PWREG3), 07D0H : Sets the pulse width.
- LD (TC3CR), 33H : Sets the operating clock to  $f_c/2^3$ , and 16-bit PWM output mode (lower byte).
- LD (TC4CR), 056H : Sets TFF4 to the initial value 0, and 16-bit PWM signal generation mode (upper byte).
- LD (TC4CR), 05EH : Starts the timer.

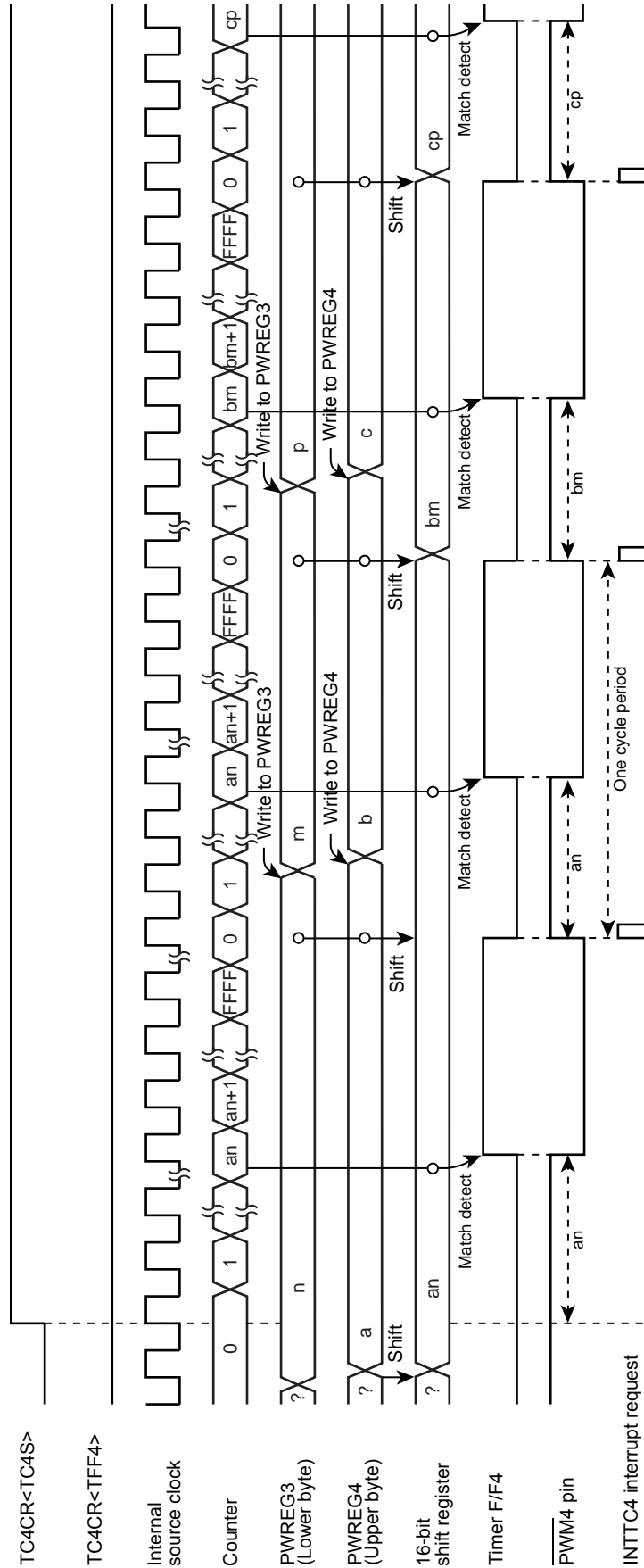


Figure 9-7 16-Bit PWM Mode Timing Chart (TC3 and TC4)

### 9.3.8 16-Bit Programmable Pulse Generate (PPG) Output Mode (TC3 and 4)

This mode is used to generate pulses with up to 16-bits of resolution. The timer counter 3 and 4 are cascaded to enter the 16-bit PPG mode.

The counter counts up using the internal clock or external clock. When a match between the up-counter and the timer register (PWREG3, PWREG4) value is detected, the logic level output from the timer F/F4 is switched to the opposite state. The counter continues counting. The logic level output from the timer F/F4 is switched to the opposite state again when a match between the up-counter and the timer register (TTREG3, TTREG4) value is detected, and the counter is cleared. The INTTC4 interrupt is generated at this time.

Two machine cycles are required for the high- or low-level pulse input to the TC3 pin. Therefore, a maximum frequency to be supplied is  $fc/2^4$  Hz in the NORMAL1 or IDLE1 mode, and  $fc/2^4$  to in the SLOW1/2 or SLEEP1/2 mode.

Since the initial value can be set to the timer F/F4 by TC4CR<TFF4>, positive and negative pulses can be generated. Upon reset, the timer F/F4 is cleared to 0.

(The logic level output from the  $\overline{PPG4}$  pin is the opposite to the timer F/F4.)

Set the lower byte and upper byte in this order to program the timer register. (TTREG3 → TTREG4, PWREG3 → PWREG4) (Programming only the upper or lower byte should not be attempted.)

For PPG output, set the output latch of the I/O port to 1.

Example :Generating a pulse with 1-ms high-level width and a period of 16.385 ms ( $fc = 16.0$  MHz)

Setting ports		
LDW	(PWREG3), 07D0H	: Sets the pulse width.
LDW	(TTREG3), 8002H	: Sets the cycle period.
LD	(TC3CR), 33H	: Sets the operating clock to $fc/2^3$ , and 16-bit PPG mode (lower byte).
LD	(TC4CR), 057H	: Sets TFF4 to the initial value 0, and 16-bit PPG mode (upper byte).
LD	(TC4CR), 05FH	: Starts the timer.

Note 1: In the PPG mode, do not change the PWREG<sub>i</sub> and TTREG<sub>i</sub> settings while the timer is running. Since PWREG<sub>i</sub> and TTREG<sub>i</sub> are not in the shift register configuration in the PPG mode, the new values programmed in PWREG<sub>i</sub> and TTREG<sub>i</sub> are in effect immediately after programming PWREG<sub>i</sub> and TTREG<sub>i</sub>. Therefore, if PWREG<sub>i</sub> and TTREG<sub>i</sub> are changed while the timer is running, an expected operation may not be obtained.

Note 2: When the timer is stopped during PPG output, the  $\overline{PPG4}$  pin holds the output status when the timer is stopped. To change the output status, program TC4CR<TFF4> after the timer is stopped. Do not change TC4CR<TFF4> upon stopping of the timer.

Example: Fixing the  $\overline{PPG4}$  pin to the high level when the TimerCounter is stopped

CLR (TC4CR).3: Stops the timer

CLR (TC4CR).7: Sets the  $\overline{PPG4}$  pin to the high level

Note 3: i = 3, 4

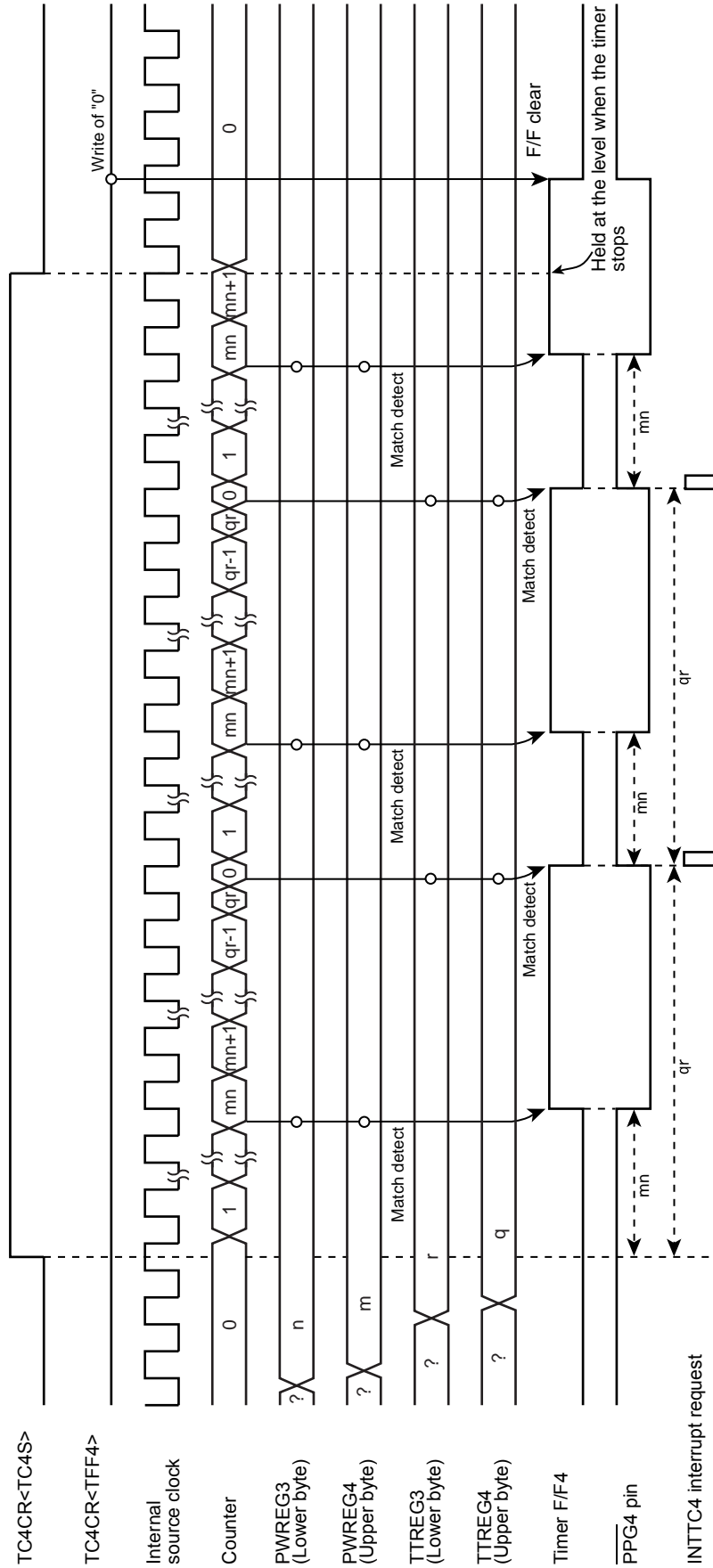


Figure 9-8 16-Bit PPG Mode Timing Chart (TC3 and TC40)

### 9.3.9 Warm-Up Counter Mode

In this mode, the warm-up period time is obtained to assure oscillation stability when the system clocking is switched between the high-frequency and low-frequency. The timer counter 3 and 4 are cascadable to form a 16-bit TimerCounter. The warm-up counter mode has two types of mode; switching from the high-frequency to low-frequency, and vice-versa.

Note 1: In the warm-up counter mode, fix TCiCR<TFFi> to 0. If not fixed, the  $\overline{P\text{DO}}_i$ ,  $\overline{P\text{WM}}_i$  and  $\overline{P\text{PG}}_i$  pins may output pulses.

Note 2: In the warm-up counter mode, only upper 8 bits of the timer register TTREG4 and 3 are used for match detection and lower 8 bits are not used.

Note 3: i = 3, 4

#### 9.3.9.1 Low-Frequency Warm-up Counter Mode (NORMAL1 → NORMAL2 → SLOW2 → SLOW1)

In this mode, the warm-up period time from a stop of the low-frequency clock fs to oscillation stability is obtained. Before starting the timer, set SYSCR2<XTEN> to 1 to oscillate the low-frequency clock. When a match between the up-counter and the timer register (TTREG4, 3) value is detected after the timer is started by setting TC4CR<TC4S> to 1, the counter is cleared by generating the INTTC4 interrupt request. After stopping the timer in the INTTC4 interrupt service routine, set SYSCR2<SYSCK> to 1 to switch the system clock from the high-frequency to low-frequency, and then clear of SYSCR2<XTEN> to 0 to stop the high-frequency clock.

Table 9-8 Setting Time of Low-Frequency Warm-Up Counter Mode (fs = 32.768 kHz)

Maximum Time Setting (TTREG4, 3 = 0100H)	Maximum Time Setting (TTREG4, 3 = FF00H)
7.81 ms	1.99 s

Example :After checking low-frequency clock oscillation stability with TC4 and 3, switching to the SLOW1 mode

```

SET      (SYSCR2).6      : SYSCR2<XTEN> ← 1
LD       (TC3CR), 43H    : Sets TFF3=0, source clock fs, and 16-bit mode.
LD       (TC4CR), 05H    : Sets TFF4=0, and warm-up counter mode.
LD       (TTREG3), 8000H : Sets the warm-up time.
                               (The warm-up time depends on the oscillator characteristic.)
DI       : IMF ← 0
SET      (EIRH), 4       : Enables the INTTC4.
EI       : IMF ← 1
SET      (TC4CR).3      : Starts TC4 and 3.
:        :
PINTTC4: CLR      (TC4CR).3 : Stops TC4 and 3.
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
:        :
VINTTC4: DW       PINTTC4 : INTTC4 vector table
    
```

### 9.3.9.2 High-Frequency Warm-Up Counter Mode (SLOW1 → SLOW2 → NORMAL2 → NORMAL1)

In this mode, the warm-up period time from a stop of the high-frequency clock  $f_c$  to the oscillation stability is obtained. Before starting the timer, set SYSCR2<XEN> to 1 to oscillate the high-frequency clock. When a match between the up-counter and the timer register (TTREG4, 3) value is detected after the timer is started by setting TC4CR<TC4S> to 1, the counter is cleared by generating the INTTC4 interrupt request. After stopping the timer in the INTTC4 interrupt service routine, clear SYSCR2<SYSCK> to 0 to switch the system clock from the low-frequency to high-frequency, and then SYSCR2<XTEN> to 0 to stop the low-frequency clock.

Table 9-9 Setting Time in High-Frequency Warm-Up Counter Mode

Minimum time (TTREG4, 3 = 0100H)	Maximum time (TTREG4, 3 = FF00H)
16 $\mu$ s	4.08 ms

Example :After checking high-frequency clock oscillation stability with TC4 and 3, switching to the NORMAL1 mode

```

SET      (SYSCR2).7      : SYSCR2<XEN> ← 1
LD       (TC3CR), 63H    : Sets TFF3=0, source clock  $f_s$ , and 16-bit mode.
LD       (TC4CR), 05H    : Sets TFF4=0, and warm-up counter mode.
LD       (TTREG3), 0F800H : Sets the warm-up time.
                                     (The warm-up time depends on the oscillator characteristic.)

DI       : IMF ← 0
SET      (EIRH). 4      : Enables the INTTC4.
EI       : IMF ← 1
SET      (TC4CR).3      : Starts the TC4 and 3.
:        :
PINTTC4: CLR      (TC4CR).3 : Stops the TC4 and 3.
CLR      (SYSCR2).5      : SYSCR2<SYSCK> ← 0
                                     (Switches the system clock to the high-frequency clock.)
CLR      (SYSCR2).6      : SYSCR2<XTEN> ← 0
                                     (Stops the low-frequency clock.)

RETI
:        :
VINTTC4: DW       PINTTC4 : INTTC4 vector table

```



# 10. 8-Bit TimerCounter (TC5, TC6)

## 10.1 Configuration

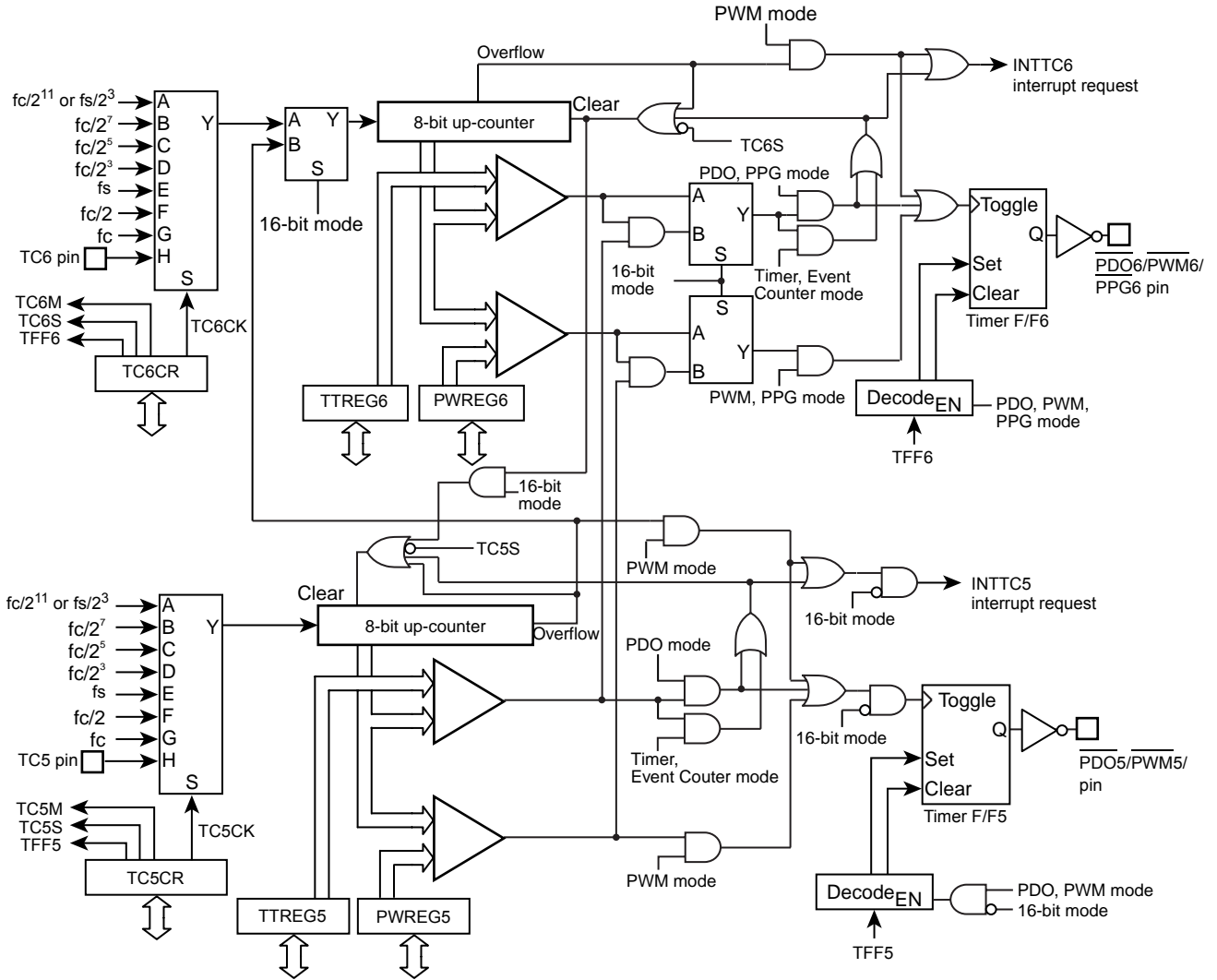
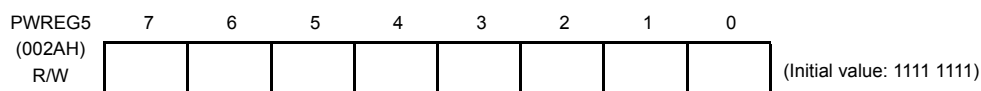
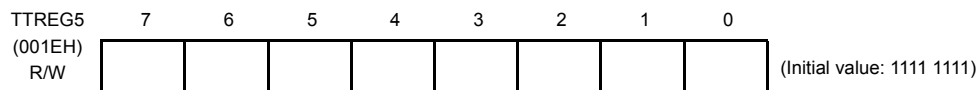


Figure 10-1 8-Bit TimerCounter 5, 6

## 10.2 TimerCounter Control

The TimerCounter 5 is controlled by the TimerCounter 5 control register (TC5CR) and two 8-bit timer registers (TTREG5, PWREG5).

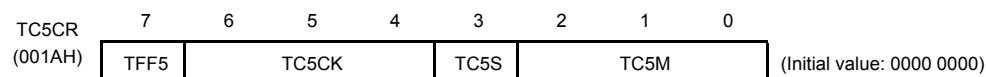
### TimerCounter 5 Timer Register



Note 1: Do not change the timer register (TTREG5) setting while the timer is running.

Note 2: Do not change the timer register (PWREG5) setting in the operating mode except the 8-bit and 16-bit PWM modes while the timer is running.

### TimerCounter 5 Control Register



TFF5	Time F/F5 control	0: Clear 1: Set			R/W	
TC5CK	Operating clock selection [Hz]	NORMAL1/2, IDLE1/2 mode		SLOW1/2 SLEEP1/2 mode	R/W	
		DV7CK = 0	DV7CK = 1			
		000	$fc/2^{11}$	$fs/2^3$		$fs/2^3$
		001	$fc/2^7$	$fc/2^7$		–
		010	$fc/2^5$	$fc/2^5$		–
		011	$fc/2^3$	$fc/2^3$		–
		100	fs	fs		fs
		101	$fc/2$	$fc/2$		–
110	fc	fc	fc (Note 8)			
111	TC5 pin input					
TC5S	TC5 start control	0: Operation stop and counter clear 1: Operation start			R/W	
TC5M	TC5M operating mode select	000: 8-bit timer/event counter mode 001: 8-bit programmable divider output (PDO) mode 010: 8-bit pulse width modulation (PWM) output mode 011: 16-bit mode (Each mode is selectable with TC6M.) 1**: Reserved			R/W	

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

Note 2: Do not change the TC5M, TC5CK and TFF5 settings while the timer is running.

Note 3: To stop the timer operation (TC5S= 1 → 0), do not change the TC5M, TC5CK and TFF5 settings. To start the timer operation (TC5S= 0 → 1), TC5M, TC5CK and TFF5 can be programmed.

Note 4: To use the TimerCounter in the 16-bit mode, set the operating mode by programming TC6CR<TC6M>, where TC5M must be fixed to 011.

Note 5: To use the TimerCounter in the 16-bit mode, select the source clock by programming TC5CK. Set the timer start control and timer F/F control by programming TC6CR<TC6S> and TC6CR<TFF6>, respectively.

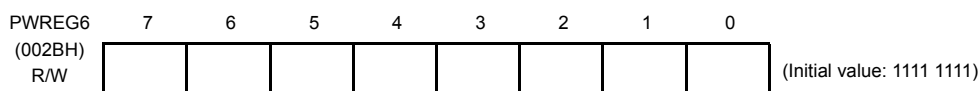
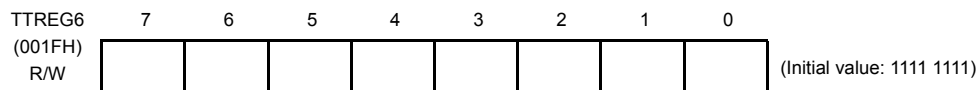
Note 6: The operating clock settings are limited depending on the timer operating mode. For the detailed descriptions, see Table 10-1 and Table 10-2.

Note 7: The timer register settings are limited depending on the timer operating mode. For the detailed descriptions, see Table 10-3.

Note 8: The operating clock  $f_c$  in the SLOW or SLEEP mode can be used only as the high-frequency warm-up mode.

The TimerCounter 6 is controlled by the TimerCounter 6 control register (TC6CR) and two 8-bit timer registers (TTREG6 and PWREG6).

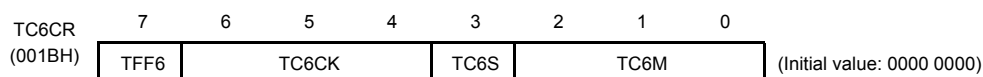
### TimerCounter 6 Timer Register



Note 1: Do not change the timer register (TTREG6) setting while the timer is running.

Note 2: Do not change the timer register (PWREG6) setting in the operating mode except the 8-bit and 16-bit PWM modes while the timer is running.

### TimerCounter 6 Control Register



TFF6	Timer F/F6 control	0: Clear 1: Set			R/W	
TC6CK	Operating clock selection [Hz]	NORMAL 1/2, IDLE 1/2 mode		SLOW 1/2 SLEEP 1/2 mode	R/W	
		DV7CK = 0	DV7CK = 1			
		000	$fc/2^{11}$	$fs/2^3$		$fs/2^3$
		001	$fc/2^7$	$fc/2^7$		–
		010	$fc/2^5$	$fc/2^5$		–
		011	$fc/2^3$	$fc/2^3$		–
		100	fs	fs		fs
		101	$fc/2$	$fc/2$		–
110	fc	fc	–			
111	TC6 pin input					
TC6S	TC6 start control	0: Operation stop and counter clear 1: Operation start			R/W	
TC6M	TC6M operating mode select	000: 8-bit timer/event counter mode 001: 8-bit programmable divider output (PDO) mode 010: 8-bit pulse width modulation (PWM) output mode 011: Reserved 100: 16-bit timer/event counter mode 101: Warm-up counter mode 110: 16-bit pulse width modulation (PWM) output mode 111: 16-bit PPG mode			R/W	

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

Note 2: Do not change the TC6M, TC6CK and TFF6 settings while the timer is running.

Note 3: To stop the timer operation (TC6S= 1 → 0), do not change the TC6M, TC6CK and TFF6 settings. To start the timer operation (TC6S= 0 → 1), TC6M, TC6CK and TFF6 can be programmed.

Note 4: When TC6M= 1\*\* (upper byte in the 16-bit mode), the source clock becomes the TC6 overflow signal regardless of the TC5CK setting.

Note 5: To use the TimerCounter in the 16-bit mode, select the operating mode by programming TC6M, where TC5CR < TC5 M must be set to 011.

Note 6: To the TimerCounter in the 16-bit mode, select the source clock by programming TC5CR<TC5CK>. Set the timer start control and timer F/F control by programming TC6S and TFF6, respectively.

Note 7: The operating clock settings are limited depending on the timer operating mode. For the detailed descriptions, see Table 10-1 and Table 10-2.

Note 8: The timer register settings are limited depending on the timer operating mode. For the detailed descriptions, see Table 10-3.

Note 9: To use the PDO, PWM or PPG mode, a pulse is not output from the timer output pin when TC1CR2<TC6OUT> is set to 1. To output a pulse from the timer output pin, clear TC1CR2<TC6OUT> to 0.

Table 10-1 Operating Mode and Selectable Source Clock (NORMAL1/2 and IDLE1/2 Modes)

Operating mode	fc/2 <sup>11</sup> or fs/2 <sup>3</sup>	fc/2 <sup>7</sup>	fc/2 <sup>5</sup>	fc/2 <sup>3</sup>	fs	fc/2	fc	TC5 pin input	TC6 pin input
8-bit timer	○	○	○	○	–	–	–	–	–
8-bit event counter	–	–	–	–	–	–	–	○	○
8-bit PDO	○	○	○	○	–	–	–	–	–
8-bit PWM	○	○	○	○	○	○	○	–	–
16-bit timer	○	○	○	○	–	–	–	–	–
16-bit event counter	–	–	–	–	–	–	–	○	–
Warm-up counter	–	–	–	–	○	–	–	–	–
16-bit PWM	○	○	○	○	○	○	○	○	–
16-bit PPG	○	○	○	○	–	–	–	○	–

Note 1: For 16-bit operations (16-bit timer/event counter, warm-up counter, 16-bit PWM and 16-bit PPG), set its source clock on lower bit (TC5CK).

Note 2: ○ : Available source clock

Table 10-2 Operating Mode and Selectable Source Clock (SLOW1/2 and SLEEP1/2 Modes)

Operating mode	fc/2 <sup>11</sup> or fs/2 <sup>3</sup>	fc/2 <sup>7</sup>	fc/2 <sup>5</sup>	fc/2 <sup>3</sup>	fs	fc/2	fc	TC5 pin input	TC6 pin input
8-bit timer	○	–	–	–	–	–	–	–	–
8-bit event counter	–	–	–	–	–	–	–	○	○
8-bit PDO	○	–	–	–	–	–	–	–	–
8-bit PWM	○	–	–	–	○	–	–	–	–
16-bit timer	○	–	–	–	–	–	–	–	–
16-bit event counter	–	–	–	–	–	–	–	○	–
Warm-up counter	–	–	–	–	–	–	○	–	–
16-bit PWM	○	–	–	–	○	–	–	○	–
16-bit PPG	○	–	–	–	–	–	–	○	–

Note1: For 16-bit operations (16-bit timer/event counter, warm-up counter, 16-bit PWM and 16-bit PPG), set its source clock on lower bit (TC5CK).

Note2: ○ : Available source clock

Table 10-3 Constraints on Register Values Being Compared

Operating mode	Register Value
8-bit timer/event counter	$1 \leq (TTREGn) \leq 255$
8-bit PDO	$1 \leq (TTREGn) \leq 255$
8-bit PWM	$2 \leq (PWREGn) \leq 254$
16-bit timer/event counter	$1 \leq (TTREG6, 5) \leq 65535$
Warm-up counter	$256 \leq (TTREG6, 5) \leq 65535$
16-bit PWM	$2 \leq (PWREG6, 5) \leq 65534$
16-bit PPG	$1 \leq (PWREG6, 5) < (TTREG6, 5) \leq 65535$ and $(PWREG6, 5) + 1 < (TTREG6, 5)$

Note: n = 5 to 6

### 10.3 Function

The TimerCounter 5 and 6 have the 8-bit timer, 8-bit event counter, 8-bit programmable divider output (PDO), 8-bit pulse width modulation (PWM) output modes. The TimerCounter 5 and 6 (TC5, 6) are cascadable to form a 16-bit timer. The 16-bit timer has the operating modes such as the 16-bit timer, 16-bit event counter, warm-up counter, 16-bit pulse width modulation (PWM) output and 16-bit programmable pulse generation (PPG) modes.

#### 10.3.1 8-Bit Timer Mode (TC5 and 6)

In the timer mode, the up-counter counts up using the internal clock. When a match between the up-counter and the timer register j (TTREGj) value is detected, an INTTCj interrupt is generated and the up-counter is cleared. After being cleared, the up-counter restarts counting.

Note 1: In the timer mode, fix TCjCR<TFFj> to 0. If not fixed, the  $\overline{PDOj}$ ,  $\overline{PWMj}$  and  $\overline{PPGj}$  pins may output pulses.

Note 2: In the timer mode, do not change the TTREGj setting while the timer is running. Since TTREGj is not in the shift register configuration in the timer mode, the new value programmed in TTREGj is in effect immediately after the programming. Therefore, if TTREGi is changed while the timer is running, an expected operation may not be obtained.

Note 3: j = 5, 6

Table 10-4 Source Clock for TimerCounter 5, 6 (Internal Clock)

Source Clock		SLOW1/2, SLEEP1/2 mode	Resolution		Repeated Cycle	
NORMAL1/2, IDLE1/2 mode			fc = 16 MHz	fs = 32.768 kHz	fc = 16 MHz	fs = 32.768 kHz
DV7CK = 0	DV7CK = 1					
$fc/2^{11}$ [Hz]	$fs/2^3$ [Hz]	$fs/2^3$ [Hz]	128 $\mu$ s	244.14 $\mu$ s	32.6 ms	62.3 ms
$fc/2^7$	$fc/2^7$	–	8 $\mu$ s	–	2.0 ms	–
$fc/2^5$	$fc/2^5$	–	2 $\mu$ s	–	510 $\mu$ s	–
$fc/2^3$	$fc/2^3$	–	500 ns	–	127.5 $\mu$ s	–

Example :Setting the timer mode with source clock  $fc/2^7$  Hz and generating an interrupt 80  $\mu$ s later (TimerCounter6,  $fc = 16.0$  MHz)

```
LD      (TTREG6), 0AH      : Sets the timer register (80  $\mu$ s =  $2^7$ /fc = 0AH).
DI
SET     (EIRH). 5         : Enables INTTC6 interrupt.
EI
LD      (TC6CR), 00010000B : Sets the operating clock to  $fc/2^7$ , and 8-bit timer mode.
LD      (TC6CR), 00011000B : Starts TC6.
```

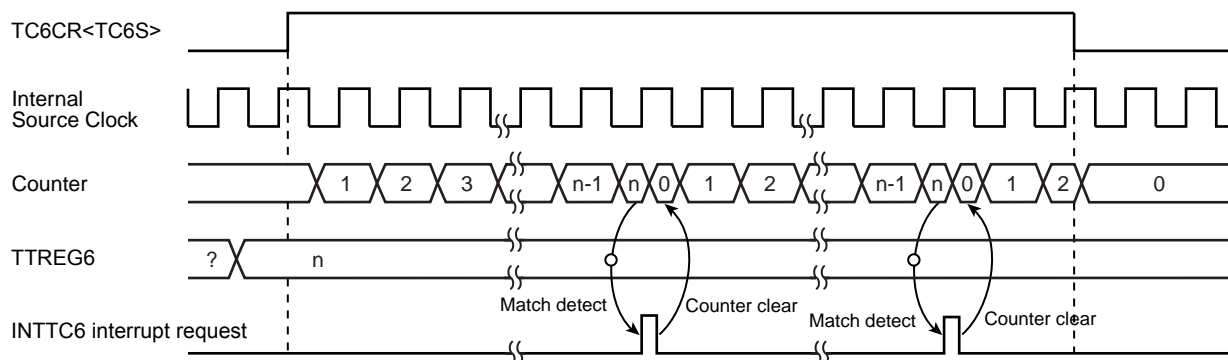


Figure 10-2 8-Bit Timer Mode Timing Chart (TC6)

### 10.3.2 8-Bit Event Counter Mode (TC5, 6)

In the 8-bit event counter mode, the up-counter counts up at the falling edge of the input pulse to the TCj pin. When a match between the up-counter and the TTREGj value is detected, an INTTCj interrupt is generated and the up-counter is cleared. After being cleared, the up-counter restarts counting at the falling edge of the input pulse to the TCj pin. Two machine cycles are required for the low- or high-level pulse input to the TCj pin. Therefore, a maximum frequency to be supplied is  $f_c/2^4$  Hz in the NORMAL1/2 or IDLE1/2 mode, and  $f_s/2^4$  Hz in the SLOW1/2 or SLEEP1/2 mode.

Note 1: In the event counter mode, fix TCjCR<TFFj> to 0. If not fixed, the  $\overline{PDOj}$ ,  $\overline{PWMj}$  and  $\overline{PPGj}$  pins may output pulses.

Note 2: In the event counter mode, do not change the TTREGj setting while the timer is running. Since TTREGj is not in the shift register configuration in the event counter mode, the new value programmed in TTREGj is in effect immediately after the programming. Therefore, if TTREGj is changed while the timer is running, an expected operation may not be obtained.

Note 3: j = 5, 6

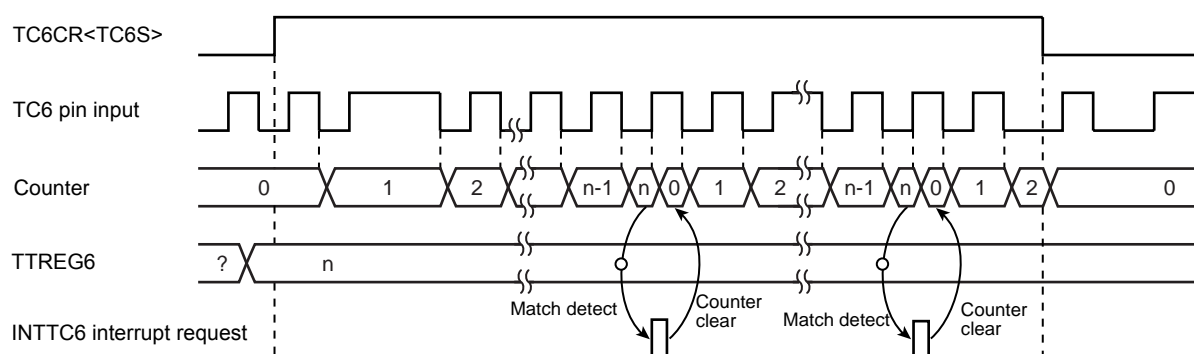


Figure 10-3 8-Bit Event Counter Mode Timing Chart (TC6)

### 10.3.3 8-Bit Programmable Divider Output (PDO) Mode (TC5, 6)

This mode is used to generate a pulse with a 50% duty cycle from the  $\overline{PDOj}$  pin.

In the PDO mode, the up-counter counts up using the internal clock. When a match between the up-counter and the TTREGj value is detected, the logic level output from the  $\overline{PDOj}$  pin is switched to the opposite state and the up-counter is cleared. The INTTCj interrupt request is generated at the time. The logic state opposite to the timer F/Fj logic level is output from the  $\overline{PDOj}$  pin. An arbitrary value can be set to the timer F/Fj by TCjCR<TFFj>. Upon reset, the timer F/Fj value is initialized to 0.

To use the programmable divider output, set the output latch of the I/O port to 1.



Example :Generating 1024 Hz pulse using TC6 (fc = 16.0 MHz)

Setting port		
LD	(TTREG6), 3DH	: $1/1024 \div 2^7 / fc \div 2 = 3DH$
LD	(TC6CR), 00010001B	: Sets the operating clock to $fc/2^7$ , and 8-bit PDO mode.
LD	(TC6CR), 00011001B	: Starts TC6.

Note 1: In the programmable divider output mode, do not change the TTREGj setting while the timer is running. Since TTREGj is not in the shift register configuration in the programmable divider output mode, the new value programmed in TTREGj is in effect immediately after programming. Therefore, if TTREGi is changed while the timer is running, an expected operation may not be obtained.

Note 2: When the timer is stopped during PDO output, the  $\overline{PDOj}$  pin holds the output status when the timer is stopped. To change the output status, program TCjCR<TFFj> after the timer is stopped. Do not change the TCjCR<TFFj> setting upon stopping of the timer.

Example: Fixing the  $\overline{PDOj}$  pin to the high level when the TimerCounter is stopped

CLR (TCjCR).3: Stops the timer.

CLR (TCjCR).7: Sets the  $\overline{PDOj}$  pin to the high level.

Note 3: j = 5, 6

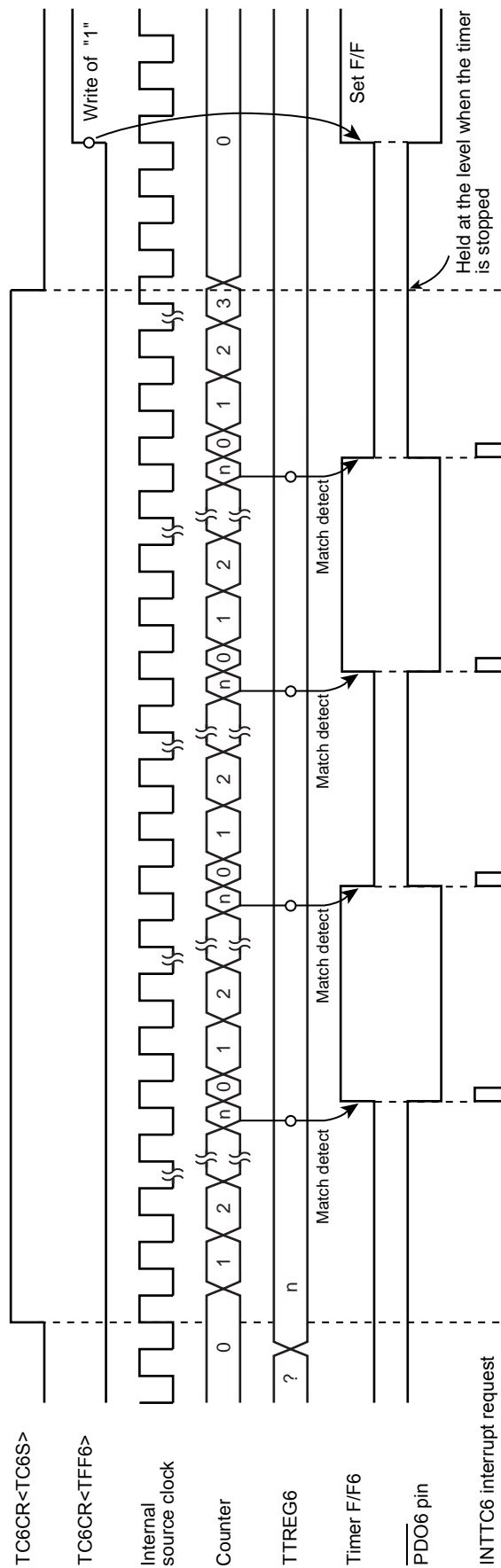


Figure 10-4 8-Bit PDO Mode Timing Chart (TC6)

### 10.3.4 8-Bit Pulse Width Modulation (PWM) Output Mode (TC5, 6)

This mode is used to generate a pulse-width modulated (PWM) signals with up to 8 bits of resolution. The up-counter counts up using the internal clock.

When a match between the up-counter and the PWREGj value is detected, the logic level output from the timer F/Fj is switched to the opposite state. The counter continues counting. The logic level output from the timer F/Fj is switched to the opposite state again by the up-counter overflow, and the counter is cleared. The INTTCj interrupt request is generated at this time.

Since the initial value can be set to the timer F/Fj by TCjCR<TFFj>, positive and negative pulses can be generated. Upon reset, the timer F/Fj is cleared to 0.

(The logic level output from the  $\overline{\text{PWMj}}$  pin is the opposite to the timer F/Fj logic level.)

Since PWREGj in the PWM mode is serially connected to the shift register, the value set to PWREGj can be changed while the timer is running. The value set to PWREGj during a run of the timer is shifted by the INTTCj interrupt request and loaded into PWREGj. While the timer is stopped, the value is shifted immediately after the programming of PWREGj. If executing the read instruction to PWREGj during PWM output, the value in the shift register is read, but not the value set in PWREGj. Therefore, after writing to PWREGj, the reading data of PWREGj is previous value until INTTCj is generated.

For the pin used for PWM output, the output latch of the I/O port must be set to 1.

Note 1: In the PWM mode, program the timer register PWREGj immediately after the INTTCj interrupt request is generated (normally in the INTTCj interrupt service routine.) If the programming of PWREGj and the interrupt request occur at the same time, an unstable value is shifted, that may result in generation of the pulse different from the programmed value until the next INTTCj interrupt request is generated.

Note 2: When the timer is stopped during PWM output, the  $\overline{\text{PWMj}}$  pin holds the output status when the timer is stopped. To change the output status, program TCjCR<TFFj> after the timer is stopped. Do not change the TCjCR<TFFj> upon stopping of the timer.

Example: Fixing the  $\overline{\text{PWMj}}$  pin to the high level when the TimerCounter is stopped

CLR (TCjCR).3: Stops the timer.

CLR (TCjCR).7: Sets the  $\overline{\text{PWMj}}$  pin to the high level.

Note 3: To enter the STOP mode during PWM output, stop the timer and then enter the STOP mode. If the STOP mode is entered without stopping the timer when fc, fc/2 or fs is selected as the source clock, a pulse is output from the  $\overline{\text{PWMj}}$  pin during the warm-up period time after exiting the STOP mode.

Note 4: j = 5, 6

Table 10-5 PWM Output Mode

Source Clock		SLOW1/2, SLEEP1/2 mode	Resolution		Repeated Cycle	
NORMAL1/2, IDLE1/2 mode			fc = 16 MHz	fs = 32.768 kHz	fc = 16 MHz	fs = 32.768 kHz
DV7CK = 0	DV7CK = 1					
$fc/2^{11}$ [Hz]	$fs/2^3$ [Hz]	$fs/2^3$ [Hz]	128 $\mu$ s	244.14 $\mu$ s	32.8 ms	62.5 ms
$fc/2^7$	$fc/2^7$	–	8 $\mu$ s	–	2.05 ms	–
$fc/2^5$	$fc/2^5$	–	2 $\mu$ s	–	512 $\mu$ s	–
$fc/2^3$	$fc/2^3$	–	500 ns	–	128 $\mu$ s	–
fs	fs	fs	30.5 $\mu$ s	30.5 $\mu$ s	7.81 ms	7.81 ms
$fc/2$	$fc/2$	–	125 ns	–	32 $\mu$ s	–
fc	fc	–	62.5 ns	–	16 $\mu$ s	–

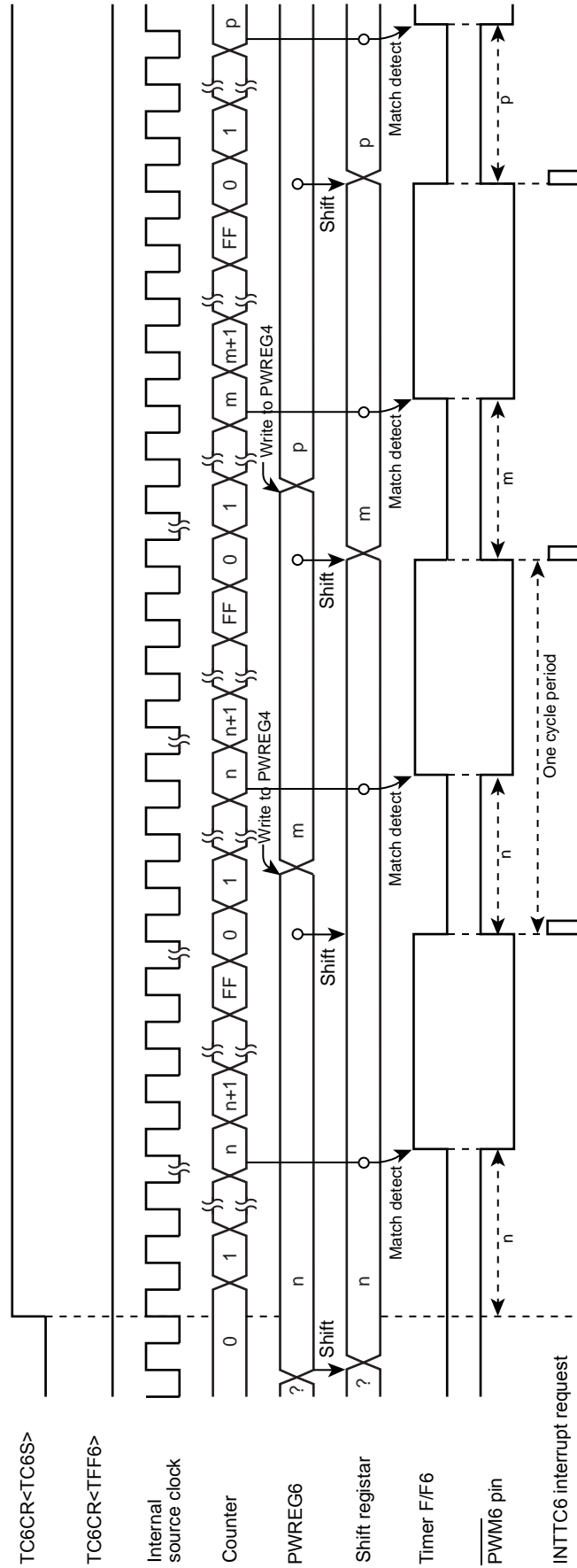


Figure 10-5 8-Bit PWM Mode Timing Chart (TC6)

### 10.3.5 16-Bit Timer Mode (TC5 and 6)

In the timer mode, the up-counter counts up using the internal clock. The TimerCounter 5 and 6 are cascadable to form a 16-bit timer.

When a match between the up-counter and the timer register (TTREG5, TTREG6) value is detected after the timer is started by setting TC6CR<TC6S> to 1, an INTTC6 interrupt is generated and the up-counter is cleared. After being cleared, the up-counter continues counting. Program the upper byte and lower byte in this order in the timer register. (Programming only the upper or lower byte should not be attempted.)

Note 1: In the timer mode, fix TCjCR<TFFj> to 0. If not fixed, the  $\overline{P\bar{D}O_j}$ ,  $\overline{P\bar{W}M_j}$ , and  $\overline{P\bar{P}G_j}$  pins may output a pulse.

Note 2: In the timer mode, do not change the TTREGj setting while the timer is running. Since TTREGj is not in the shift register configuration in the timer mode, the new value programmed in TTREGj is in effect immediately after programming of TTREGj. Therefore, if TTREGj is changed while the timer is running, an expected operation may not be obtained.

Note 3: j = 5, 6

Table 10-6 Source Clock for 16-Bit Timer Mode

Source Clock		SLOW1/2, SLEEP1/2 mode	Resolution		Repeated Cycle	
NORMAL1/2, IDLE1/2 mode	DV7CK = 0		DV7CK = 1	fc = 16 MHz	fs = 32.768 kHz	fc = 16 MHz
fc/2 <sup>11</sup>	fs/2 <sup>3</sup>	fs/2 <sup>3</sup>	128 μs	244.14 μs	8.39 s	16 s
fc/2 <sup>7</sup>	fc/2 <sup>7</sup>	–	8 μs	–	524.3 ms	–
fc/2 <sup>5</sup>	fc/2 <sup>5</sup>	–	2 μs	–	131.1 ms	–
fc/2 <sup>3</sup>	fc/2 <sup>3</sup>	–	500 ns	–	32.8 ms	–

Example :Setting the timer mode with source clock  $fc/2^7$  Hz, and generating an interrupt 300 ms later  
(fc = 16.0 MHz)

- LDW (TTREG5), 927CH : Sets the timer register (300 ms =  $2^7 / fc = 927CH$ ).
- DI
- SET (EIRH), 5 : Enables INTTC6 interrupt.
- EI
- LD (TC5CR), 13H : Sets the operating clock to  $fc/2^7$ , and 16-bit timer mode (lower byte).
- LD (TC6CR), 04H : Sets the 16-bit timer mode (upper byte).
- LD (TC6CR), 0CH : Starts the timer.

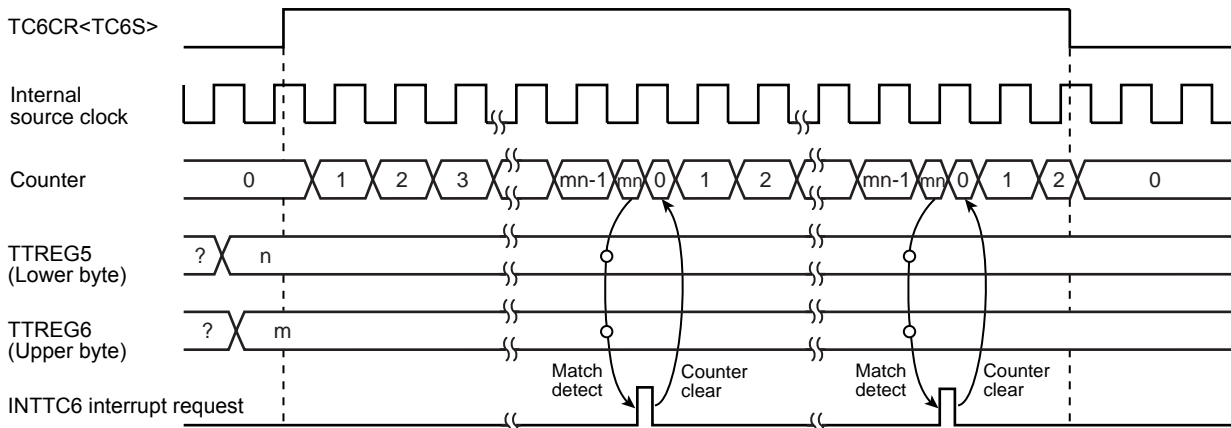


Figure 10-6 16-Bit Timer Mode Timing Chart (TC5 and TC6)

### 10.3.6 16-Bit Event Counter Mode (TC5 and 6)

In the event counter mode, the up-counter counts up at the falling edge to the TC5 pin. The TimerCounter 5 and 6 are cascadable to form a 16-bit event counter.

When a match between the up-counter and the timer register (TTREG5, TTREG6) value is detected after the timer is started by setting TC6CR<TC6S> to 1, an INTTC6 interrupt is generated and the up-counter is cleared.

After being cleared, the up-counter restarts counting at the falling edge of the input pulse to the TC5 pin. Two machine cycles are required for the low- or high-level pulse input to the TC5 pin.

Therefore, a maximum frequency to be supplied is  $f_c/2^4$  Hz in the NORMAL1 or IDLE1 mode, and  $f_s/2^4$  in the SLOW1/2 or SLEEP1/2 mode. Program the lower byte (TTREG5), and upper byte (TTREG6) in this order in the timer register. (Programming only the upper or lower byte should not be attempted.)

Note 1: In the event counter mode, fix TCjCR<TFFj> to 0. If not fixed, the  $\overline{PDOj}$ ,  $\overline{PWMj}$  and  $\overline{PPGj}$  pins may output pulses.

Note 2: In the event counter mode, do not change the TTREGj setting while the timer is running. Since TTREGj is not in the shift register configuration in the event counter mode, the new value programmed in TTREGj is in effect immediately after the programming. Therefore, if TTREGj is changed while the timer is running, an expected operation may not be obtained.

Note 3: j = 5, 6

### 10.3.7 16-Bit Pulse Width Modulation (PWM) Output Mode (TC5 and 6)

This mode is used to generate a pulse-width modulated (PWM) signals with up to 16 bits of resolution. The TimerCounter 5 and 6 are cascadable to form the 16-bit PWM signal generator.

The counter counts up using the internal clock or external clock.

When a match between the up-counter and the timer register (PWREG5, PWREG6) value is detected, the logic level output from the timer F/F6 is switched to the opposite state. The counter continues counting. The logic level output from the timer F/F6 is switched to the opposite state again by the counter overflow, and the counter is cleared. The INTTC6 interrupt is generated at this time.

Two machine cycles are required for the high- or low-level pulse input to the TC5 pin. Therefore, a maximum frequency to be supplied is  $f_c/2^4$  Hz in the NORMAL1 or IDLE1 mode, and  $f_s/2^4$  to in the SLOW1/2 or SLEEP1/2 mode.

Since the initial value can be set to the timer F/F6 by TC6CR<TFF6>, positive and negative pulses can be generated. Upon reset, the timer F/F6 is cleared to 0.

(The logic level output from the  $\overline{PWM6}$  pin is the opposite to the timer F/F6 logic level.)

Since PWREG6 and 5 in the PWM mode are serially connected to the shift register, the values set to PWREG6 and 5 can be changed while the timer is running. The values set to PWREG6 and 5 during a run of the timer are shifted by the INTTCj interrupt request and loaded into PWREG6 and 5. While the timer is stopped, the values are shifted immediately after the programming of PWREG6 and 5. Set the lower byte (PWREG5) and upper byte (PWREG5) in this order to program PWREG6 and 5. (Programming only the lower or upper byte of the register should not be attempted.)

If executing the read instruction to PWREG6 and 5 during PWM output, the values set in the shift register is read, but not the values set in PWREG6 and 5. Therefore, after writing to the PWREG6 and 5, reading data of PWREG6 and 5 is previous value until INTTC6 is generated.

For the pin used for PWM output, the output latch of the I/O port must be set to 1.

Note 1: In the PWM mode, program the timer register PWREG6 and 5 immediately after the INTTC6 interrupt request is generated (normally in the INTTC6 interrupt service routine.) If the programming of PWREGj and the interrupt request 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 generated.

Note 2: When the timer is stopped during PWM output, the  $\overline{PWM6}$  pin holds the output status when the timer is stopped. To change the output status, program TC6CR<TFF6> after the timer is stopped. Do not program TC6CR<TFF6> upon stopping of the timer.

Example: Fixing the  $\overline{PWM6}$  pin to the high level when the TimerCounter is stopped

CLR (TC6CR).3: Stops the timer.  
 CLR (TC6CR).7 : Sets the  $\overline{PWM6}$  pin to the high level.

Note 3: To enter the STOP mode, stop the timer and then enter the STOP mode. If the STOP mode is entered without stopping of the timer when  $f_c$ ,  $f_c/2$  or  $f_s$  is selected as the source clock, a pulse is output from the  $\overline{PWM6}$  pin during the warm-up period time after exiting the STOP mode.

Table 10-7 16-Bit PWM Output Mode

Source Clock		SLOW1/2, SLEEP1/2 mode	Resolution		Repeated Cycle	
NORMAL1/2, IDLE1/2 mode	DV7CK = 0		DV7CK = 1	$f_c = 16 \text{ MHz}$	$f_s = 32.768 \text{ kHz}$	$f_c = 16 \text{ MHz}$
$f_c/2^{11}$	$f_s/2^3 \text{ [Hz]}$	$f_s/2^3 \text{ [Hz]}$	128 $\mu\text{s}$	244.14 $\mu\text{s}$	8.39 s	16 s
$f_c/2^7$	$f_c/2^7$	–	8 $\mu\text{s}$	–	524.3 ms	–
$f_c/2^5$	$f_c/2^5$	–	2 $\mu\text{s}$	–	131.1 ms	–
$f_c/2^3$	$f_c/2^3$	–	500ns	–	32.8 ms	–
$f_s$	$f_s$	$f_s$	30.5 $\mu\text{s}$	30.5 $\mu\text{s}$	2 s	2 s
$f_c/2$	$f_c/2$	–	125 ns	–	8.2 ms	–
$f_c$	$f_c$	–	62.5 ns	–	4.1 ms	–

Example :Generating a pulse with 1-ms high-level width and a period of 32.768 ms ( $f_c = 16.0 \text{ MHz}$ )

Setting ports

- LDW (PWREG5), 07D0H : Sets the pulse width.
- LD (TC5CR), 33H : Sets the operating clock to  $f_c/2^3$ , and 16-bit PWM output mode (lower byte).
- LD (TC6CR), 056H : Sets TFF6 to the initial value 0, and 16-bit PWM signal generation mode (upper byte).
- LD (TC6CR), 05EH : Starts the timer.

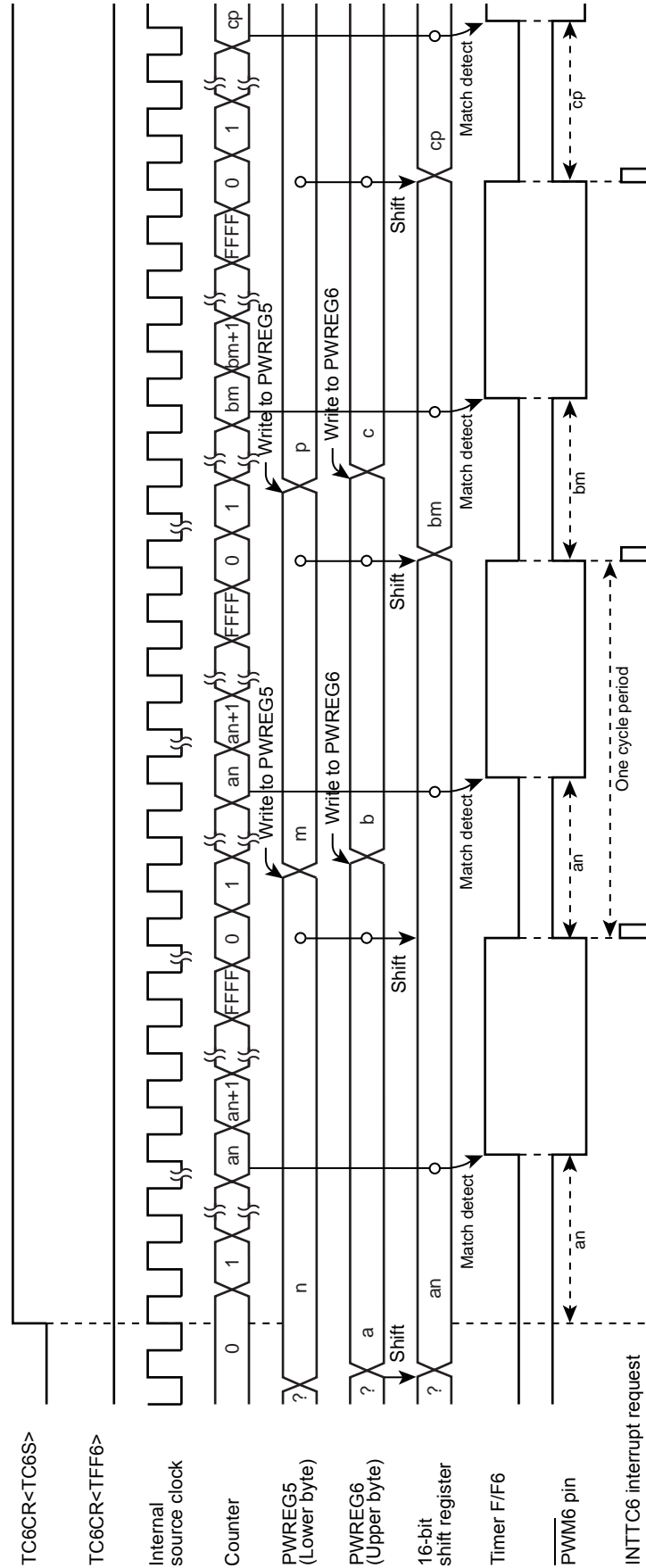


Figure 10-7 16-Bit PWM Mode Timing Chart (TC5 and TC6)



### 10.3.8 16-Bit Programmable Pulse Generate (PPG) Output Mode (TC5 and 6)

This mode is used to generate pulses with up to 16-bits of resolution. The timer counter 5 and 6 are cascaded to enter the 16-bit PPG mode.

The counter counts up using the internal clock or external clock. When a match between the up-counter and the timer register (PWREG5, PWREG6) value is detected, the logic level output from the timer F/F6 is switched to the opposite state. The counter continues counting. The logic level output from the timer F/F6 is switched to the opposite state again when a match between the up-counter and the timer register (TTREG5, TTREG6) value is detected, and the counter is cleared. The INTTC6 interrupt is generated at this time.

Two machine cycles are required for the high- or low-level pulse input to the TC5 pin. Therefore, a maximum frequency to be supplied is  $fc/2^4$  Hz in the NORMAL1 or IDLE1 mode, and  $fc/2^4$  to in the SLOW1/2 or SLEEP1/2 mode.

Since the initial value can be set to the timer F/F6 by TC6CR<TFF6>, positive and negative pulses can be generated. Upon reset, the timer F/F6 is cleared to 0.

(The logic level output from the  $\overline{PPG6}$  pin is the opposite to the timer F/F6.)

Set the lower byte and upper byte in this order to program the timer register. (TTREG5 → TTREG6, PWREG5 → PWREG6) (Programming only the upper or lower byte should not be attempted.)

For PPG output, set the output latch of the I/O port to 1.

Example :Generating a pulse with 1-ms high-level width and a period of 16.385 ms ( $fc = 16.0$  MHz)

Setting ports		
LDW	(PWREG5), 07D0H	: Sets the pulse width.
LDW	(TTREG5), 8002H	: Sets the cycle period.
LD	(TC5CR), 33H	: Sets the operating clock to $fc/2^3$ , and 16-bit PPG mode (lower byte).
LD	(TC6CR), 057H	: Sets TFF6 to the initial value 0, and 16-bit PPG mode (upper byte).
LD	(TC6CR), 05FH	: Starts the timer.

Note 1: In the PPG mode, do not change the PWREGi and TTREGi settings while the timer is running. Since PWREGi and TTREGi are not in the shift register configuration in the PPG mode, the new values programmed in PWREGi and TTREGi are in effect immediately after programming PWREGi and TTREGi. Therefore, if PWREGi and TTREGi are changed while the timer is running, an expected operation may not be obtained.

Note 2: When the timer is stopped during PPG output, the  $\overline{PPG6}$  pin holds the output status when the timer is stopped. To change the output status, program TC6CR<TFF6> after the timer is stopped. Do not change TC6CR<TFF6> upon stopping of the timer.

Example: Fixing the  $\overline{PPG6}$  pin to the high level when the TimerCounter is stopped

CLR (TC6CR).3: Stops the timer

CLR (TC6CR).7: Sets the  $\overline{PPG6}$  pin to the high level

Note 3: i = 5, 6



### 10.3.9 Warm-Up Counter Mode

In this mode, the warm-up period time is obtained to assure oscillation stability when the system clocking is switched between the high-frequency and low-frequency. The timer counter 5 and 6 are cascadable to form a 16-bit TimerCounter. The warm-up counter mode has two types of mode; switching from the high-frequency to low-frequency, and vice-versa.

Note 1: In the warm-up counter mode, fix TCiCR<TFFi> to 0. If not fixed, the  $\overline{P\text{DO}}_i$ ,  $\overline{P\text{WM}}_i$  and  $\overline{P\text{PG}}_i$  pins may output pulses.

Note 2: In the warm-up counter mode, only upper 8 bits of the timer register TTREG6 and 5 are used for match detection and lower 8 bits are not used.

Note 3: i = 5, 6

#### 10.3.9.1 Low-Frequency Warm-up Counter Mode (NORMAL1 → NORMAL2 → SLOW2 → SLOW1)

In this mode, the warm-up period time from a stop of the low-frequency clock fs to oscillation stability is obtained. Before starting the timer, set SYSCR2<XTEN> to 1 to oscillate the low-frequency clock. When a match between the up-counter and the timer register (TTREG6, 5) value is detected after the timer is started by setting TC6CR<TC6S> to 1, the counter is cleared by generating the INTTC6 interrupt request. After stopping the timer in the INTTC6 interrupt service routine, set SYSCR2<SYSCK> to 1 to switch the system clock from the high-frequency to low-frequency, and then clear of SYSCR2<XTEN> to 0 to stop the high-frequency clock.

Table 10-8 Setting Time of Low-Frequency Warm-Up Counter Mode (fs = 32.768 kHz)

Maximum Time Setting (TTREG6, 5 = 0100H)	Maximum Time Setting (TTREG6, 5 = FF00H)
7.81 ms	1.99 s

Example :After checking low-frequency clock oscillation stability with TC6 and 5, switching to the SLOW1 mode

```

SET      (SYSCR2).6      : SYSCR2<XTEN> ← 1
LD       (TC5CR), 43H    : Sets TFF5=0, source clock fs, and 16-bit mode.
LD       (TC6CR), 05H    : Sets TFF6=0, and warm-up counter mode.
LD       (TTREG5), 8000H : Sets the warm-up time.
                               (The warm-up time depends on the oscillator characteristic.)
DI       : IMF ← 0
SET      (EIRH). 5      : Enables the INTTC6.
EI       : IMF ← 1
SET      (TC6CR).3     : Starts TC6 and 5.
:       :
PINTTC6: CLR      (TC6CR).3 : Stops TC6 and 5.
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
:       :
VINTTC6: DW       PINTTC6 : INTTC6 vector table
    
```

### 10.3.9.2 High-Frequency Warm-Up Counter Mode (SLOW1 → SLOW2 → NORMAL2 → NORMAL1)

In this mode, the warm-up period time from a stop of the high-frequency clock  $f_c$  to the oscillation stability is obtained. Before starting the timer, set SYSCR2<XEN> to 1 to oscillate the high-frequency clock. When a match between the up-counter and the timer register (TTREG6, 5) value is detected after the timer is started by setting TC6CR<TC6S> to 1, the counter is cleared by generating the INTTC6 interrupt request. After stopping the timer in the INTTC6 interrupt service routine, clear SYSCR2<SYSCK> to 0 to switch the system clock from the low-frequency to high-frequency, and then SYSCR2<XTEN> to 0 to stop the low-frequency clock.

Table 10-9 Setting Time in High-Frequency Warm-Up Counter Mode

Minimum time (TTREG6, 5 = 0100H)	Maximum time (TTREG6, 5 = FF00H)
16 $\mu$ s	4.08 ms

Example :After checking high-frequency clock oscillation stability with TC6 and 5, switching to the NORMAL1 mode

```

SET      (SYSCR2).7      : SYSCR2<XEN> ← 1
LD       (TC5CR), 63H    : Sets TFF5=0, source clock  $f_s$ , and 16-bit mode.
LD       (TC6CR), 05H    : Sets TFF6=0, and warm-up counter mode.
LD       (TTREG5), 0F800H : Sets the warm-up time.
                                     (The warm-up time depends on the oscillator characteristic.)

DI                                              : IMF ← 0
SET      (EIRH). 5      : Enables the INTTC6.
EI                                              : IMF ← 1
SET      (TC6CR).3      : Starts the TC6 and 5.
:      :
PINTTC6: CLR      (TC6CR).3      : Stops the TC6 and 5.
CLR      (SYSCR2).5      : SYSCR2<SYSCK> ← 0
                                     (Switches the system clock to the high-frequency clock.)
CLR      (SYSCR2).6      : SYSCR2<XTEN> ← 0
                                     (Stops the low-frequency clock.)

RETI
:      :
VINTTC6: DW       PINTTC6      : INTTC6 vector table

```

# 11. Asynchronous Serial interface (UART )

## 11.1 Configuration

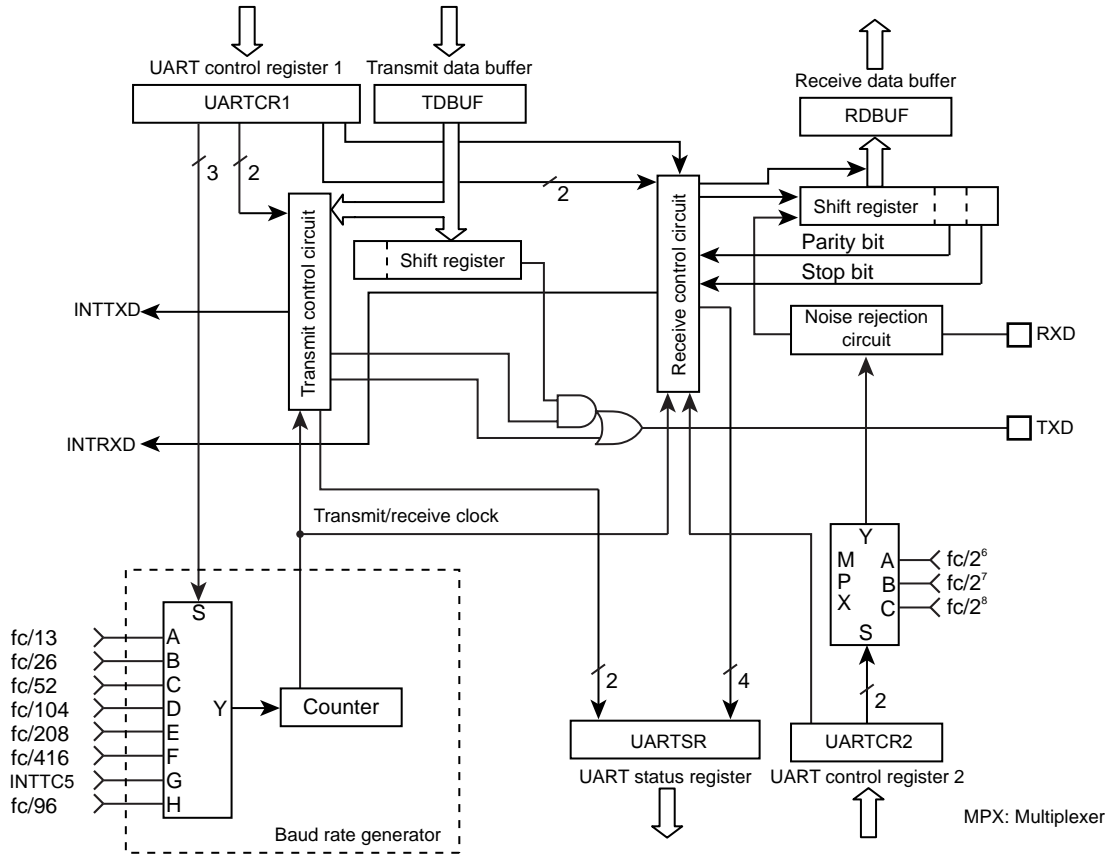


Figure 11-1 UART (Asynchronous Serial Interface)

## 11.2 Control

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

### UART Control Register1

UARTCR1 (0025H)	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: TC5 ( Input INTTC5) 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.

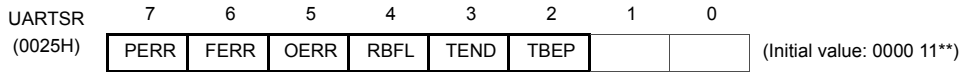
### UART Control Register2

UARTCR2 (0026H)	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].

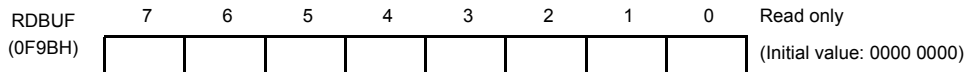
UART Status Register



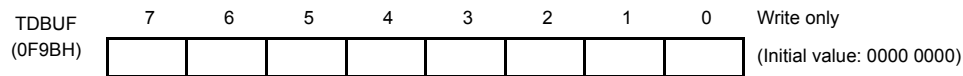
PERR	Parity error flag	0: No parity error 1: Parity error	Read only
FERR	Framing error flag	0: No framing error 1: Framing error	
OERR	Overrun error flag	0: No overrun error 1: Overrun error	
RBFL	Receive data buffer full flag	0: Receive data buffer empty 1: Receive data buffer full	
TEND	Transmit end 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



### 11.3 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	4	5	6	7	8	9	10
0	0										
0	1										
1	0										
1	1										

Figure 11-2 Transfer Data Format

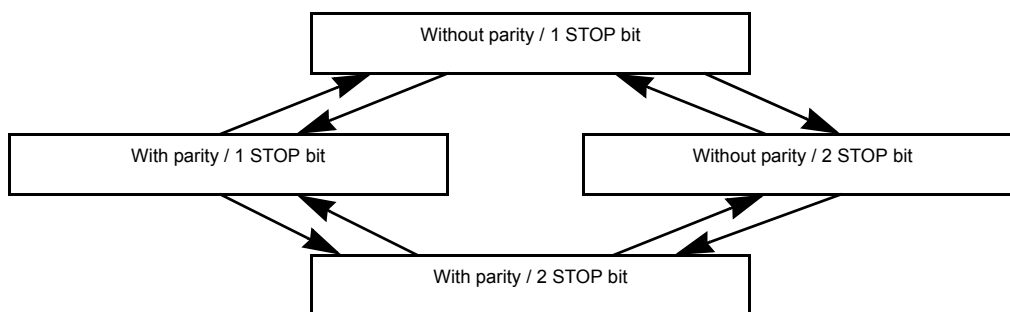


Figure 11-3 Caution on Changing Transfer Data Format

Note: In order to switch the transfer data format, perform transmit operations in the above Figure 11-3 sequence except for the initial setting.



### 11.4 Transfer Rate

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

Table 11-1 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 TC5 is used as the UART transfer rate (when UARTCR1<BRG> = “110”), the transfer clock and transfer rate are determined as follows:

$$\text{Transfer clock [Hz]} = \text{TC5 source clock [Hz]} / \text{TTREG5 setting value}$$

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

### 11.5 Data Sampling Method

The UART receiver keeps sampling input using the clock selected by UARTCR1<BRG> until a start bit is detected in RXD pin input. RT clock starts detecting “L” level of the RXD 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).

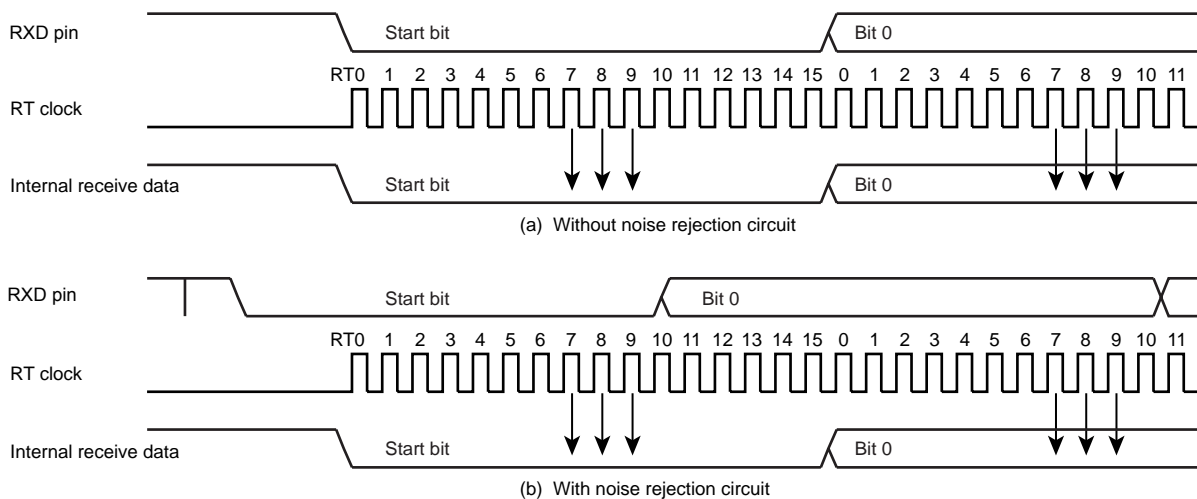


Figure 11-4 Data Sampling Method

---

## 11.6 STOP Bit Length

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

## 11.7 Parity

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

## 11.8 Transmit/Receive Operation

### 11.8.1 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 TXD 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 INTTXD 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 TXD 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.

### 11.8.2 Data Receive Operation

Set UARTCR1<RXE> to “1”. When data are received via the RXD 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 INTRXD 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.

## 11.9 Status Flag

### 11.9.1 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.

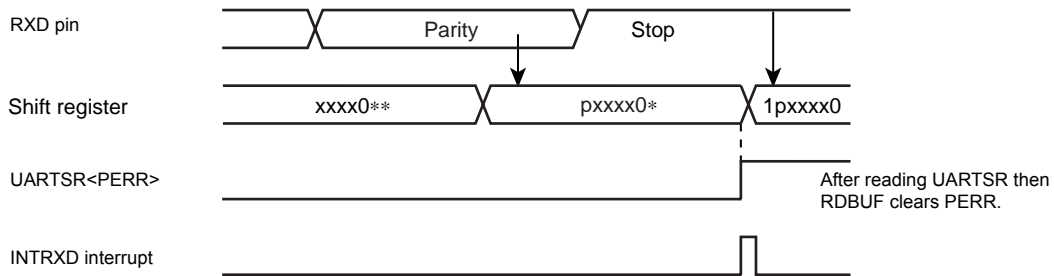


Figure 11-5 Generation of Parity Error

### 11.9.2 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.

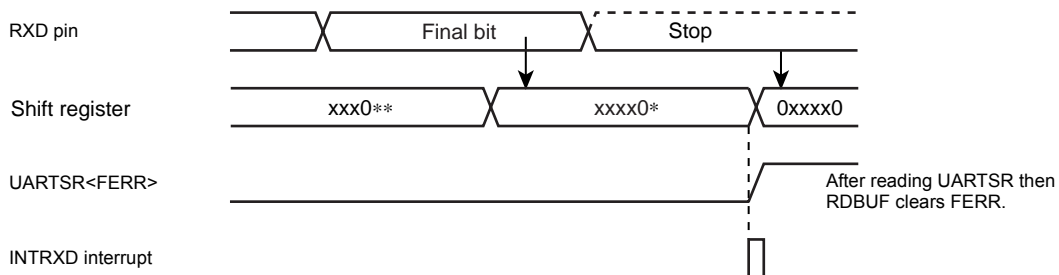


Figure 11-6 Generation of Framing Error

### 11.9.3 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.

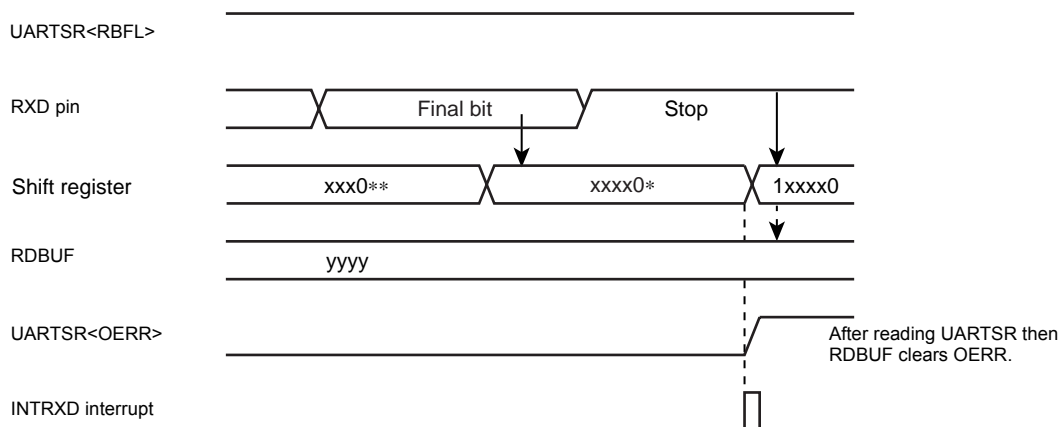


Figure 11-7 Generation of Overrun Error

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

#### 11.9.4 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.

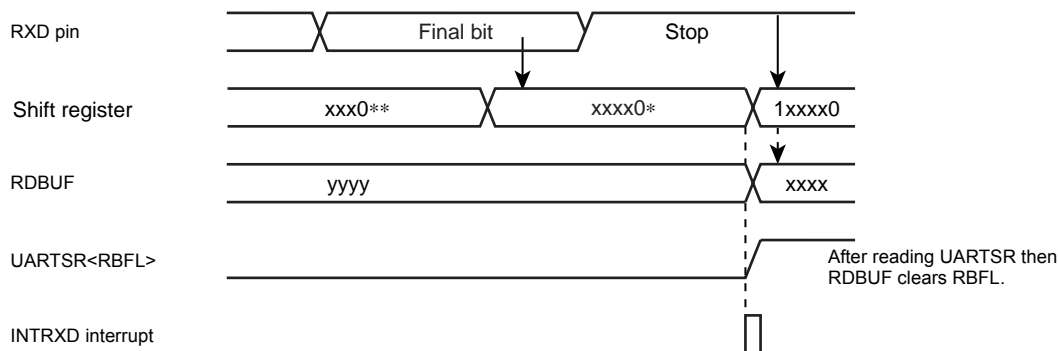


Figure 11-8 Generation of Receive Data Buffer Full

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.

#### 11.9.5 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.

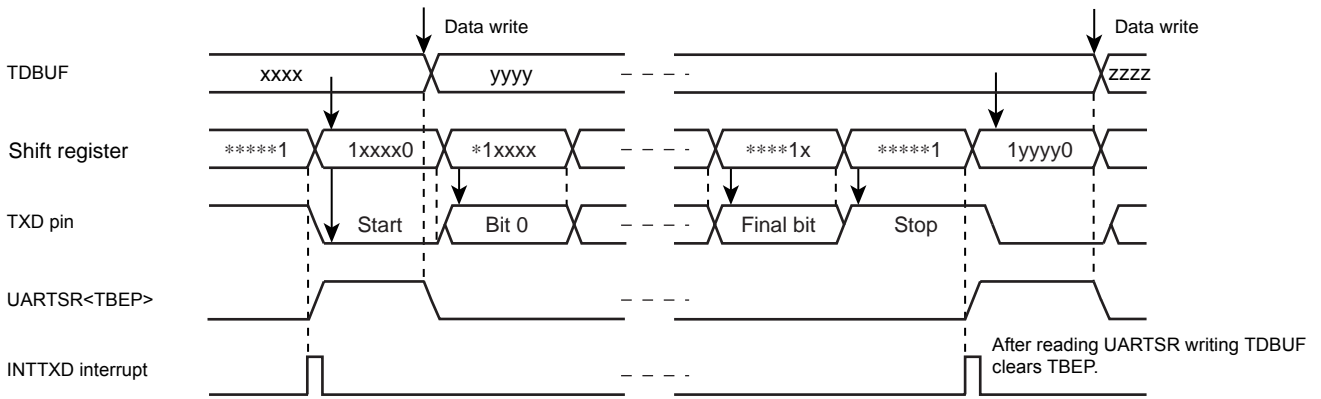


Figure 11-9 Generation of Transmit Data Buffer Empty

### 11.9.6 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.

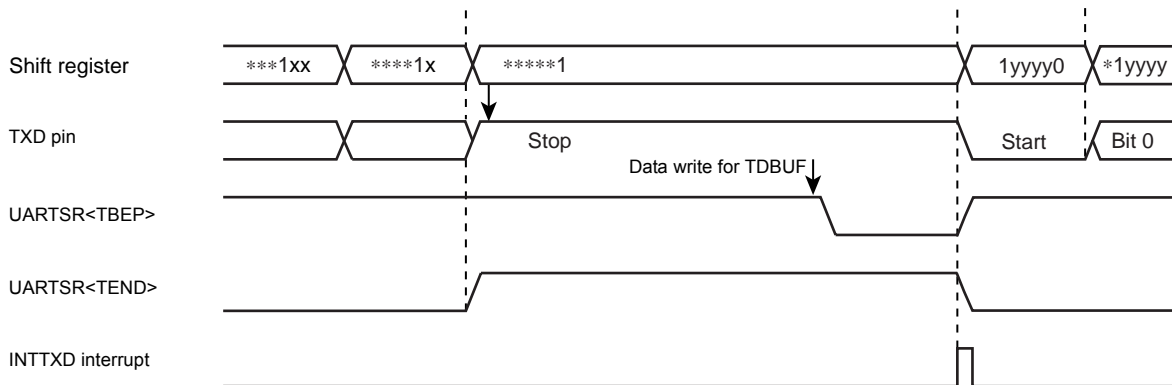


Figure 11-10 Generation of Transmit End Flag and Transmit Data Buffer Empty



## 12. Synchronous Serial Interface (SIO)

The TMP86FS23UG 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 SO, SI, SCK port.

### 12.1 Configuration

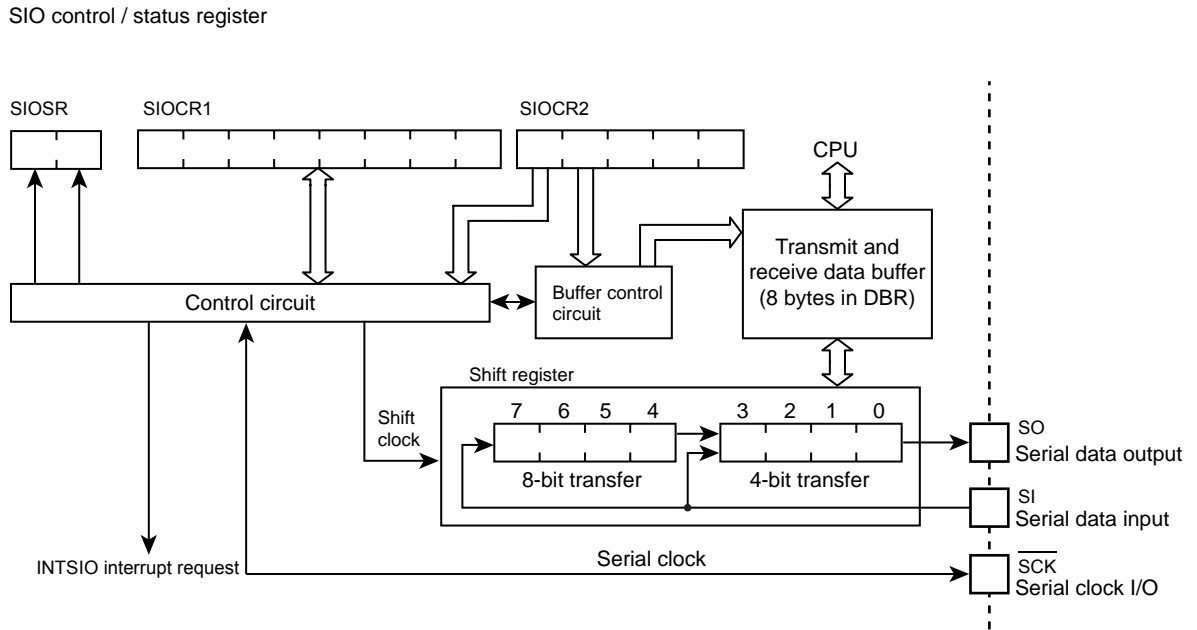


Figure 12-1 Serial Interface

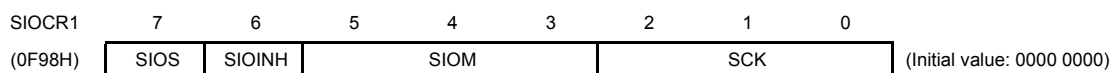
## 12.2 Control

The serial interface is controlled by SIO control registers (SIOCR1/SIOCR2). The serial interface status can be determined by reading SIO status register (SIOSR).

The transmit and receive data buffer is controlled by the SIOCR2<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 (INTSIO) 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 SIOCR2<WAIT>.

### SIO Control Register 1



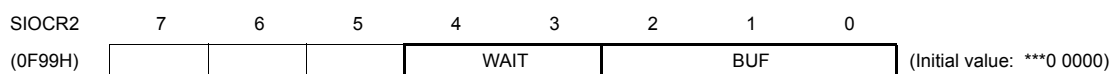
SIOS	Indicate transfer start / stop	0: Stop 1: Start			Write only	
SIOINH	Continue / abort transfer	0: Continuously transfer 1: Abort transfer (Automatically cleared after abort)				
SIOM	Transfer mode select	000: 8-bit transmit mode 010: 4-bit transmit mode 100: 8-bit transmit / receive mode 101: 8-bit receive mode 110: 4-bit receive mode Except the above: Reserved			Write only	
SCK	Serial clock select		NORMAL1/2, IDLE1/2 mode		Write only	
			DV7CK = 0	DV7CK = 1		SLOW1/2 SLEEP1/2 mode
		000	$fc/2^{13}$	$fs/2^5$		$fs/2^5$
		001	$fc/2^8$	$fc/2^8$		-
		010	$fc/2^7$	$fc/2^7$		-
		011	$fc/2^6$	$fc/2^6$		-
		100	$fc/2^5$	$fc/2^5$		-
		101	$fc/2^4$	$fc/2^4$		-
		Reserved				
		External clock ( Input from SCK 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: SIOCR1 is write-only register, which cannot access any of in read-modify-write instruction such as bit operate, etc.

### SIO Control Register 2

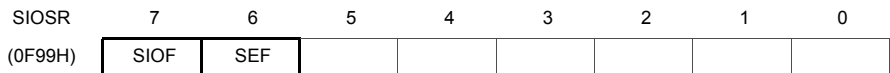




WAIT	Wait control	Always sets "00" except 8-bit transmit / receive mode. 00: $T_f = T_D$ (Non wait) 01: $T_f = 2T_D$ (Wait) 10: $T_f = 4T_D$ (Wait) 11: $T_f = 8T_D$ (Wait)	Write only
BUF	Number of transfer words (Buffer address in use)	000: 1 word transfer 0F90H 001: 2 words transfer 0F90H ~ 0F91H 010: 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: SIOCR2 must be set when the serial interface is stopped (SIOF = 0).
- Note 5: \*: Don't care
- Note 6: SIOCR2 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".

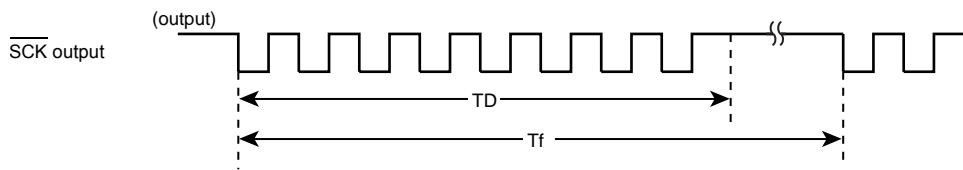


Figure 12-2 Frame time ( $T_f$ ) and Data transfer time ( $T_D$ )

12.3 Serial clock

12.3.1 Clock source

Internal clock or external clock for the source clock is selected by SIOCR1<SCK>.

### 12.3.1.1 Internal clock

Any of six frequencies can be selected. The serial clock is output to the outside on the SCK pin. The SCK 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 12-1 Serial Clock Rate

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

Note: 1 Kbit = 1024 bit ( $f_c = 16$  MHz,  $f_s = 32.768$  kHz)

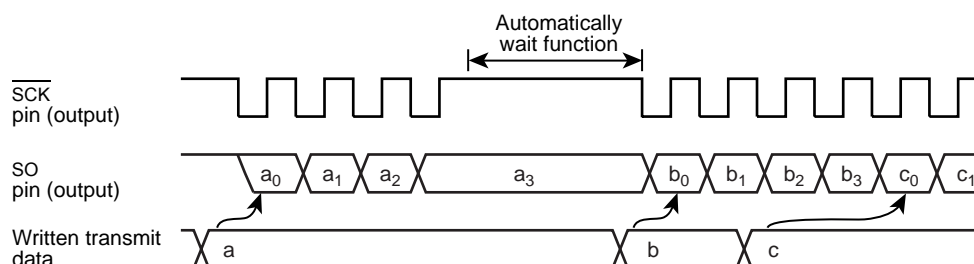


Figure 12-3 Automatic Wait Function (at 4-bit transmit mode)

### 12.3.1.2 External clock

An external clock connected to the SCK 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  $f_c=16$  MHz).

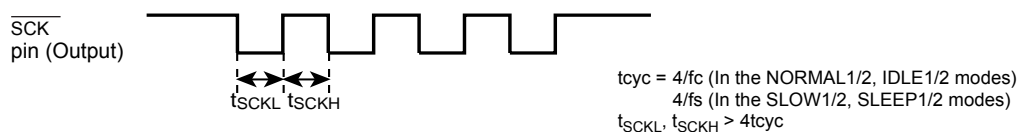


Figure 12-4 External clock pulse width

### 12.3.2 Shift edge

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

#### 12.3.2.1 Leading edge

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

#### 12.3.2.2 Trailing edge

Received data are shifted on the trailing edge of the serial clock (rising edge of the  $\overline{\text{SCK}}$  pin input/output).

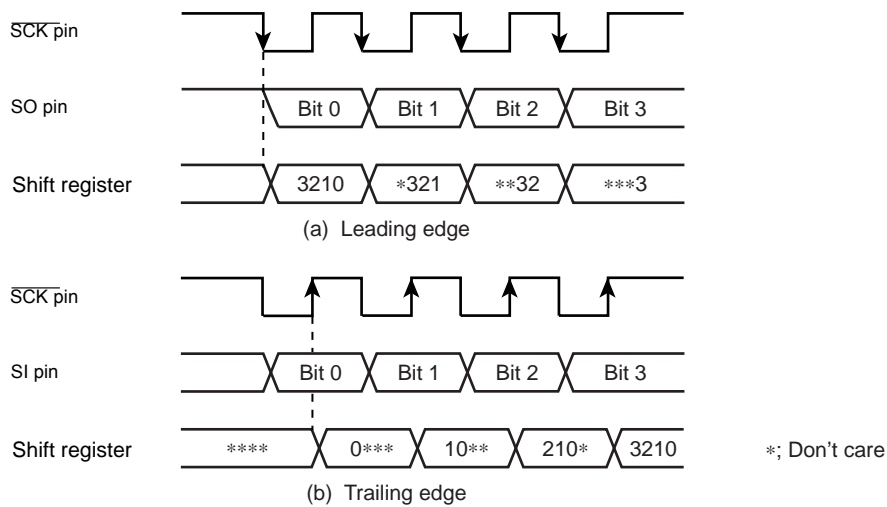


Figure 12-5 Shift edge

## 12.4 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).

## 12.5 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  $\text{SIOCR2}\langle\text{BUF}\rangle$ .

An INTSIO 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.

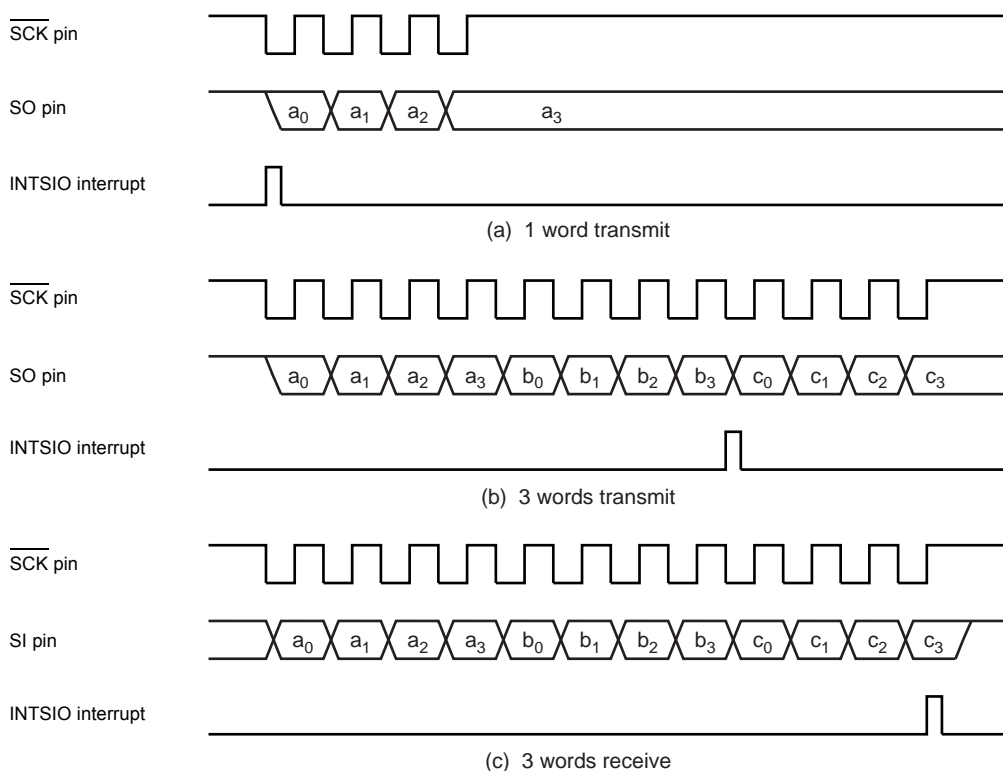


Figure 12-6 Number of words to transfer (Example: 1word = 4bit)

## 12.6 Transfer Mode

SIOCR1<SIOM> is used to select the transmit, receive, or transmit/receive mode.

### 12.6.1 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 SIOCR1<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 INTSIO (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 SIOCR2<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 SIOCR1<SIOS> to “0” or setting SIOCR1<SIOINH> to “1” in buffer empty interrupt service program.

SIOCR1<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 SIOSR<SIOF> because SIOSR<SIOF> is cleared to “0” when a transfer is completed.

When SIOCR1<SIOINH> is set, the transmission is immediately ended and SIOSR<SIOF> is cleared to “0”.

When an external clock is used, it is also necessary to clear SIOCR1<SIOS> to “0” before shifting the next data; If SIOCR1<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, SIOCR1<SIOS> should be cleared to “0”, then SIOCR2<BUF> must be rewritten after confirming that SIOSR<SIOF> has been cleared to “0”.

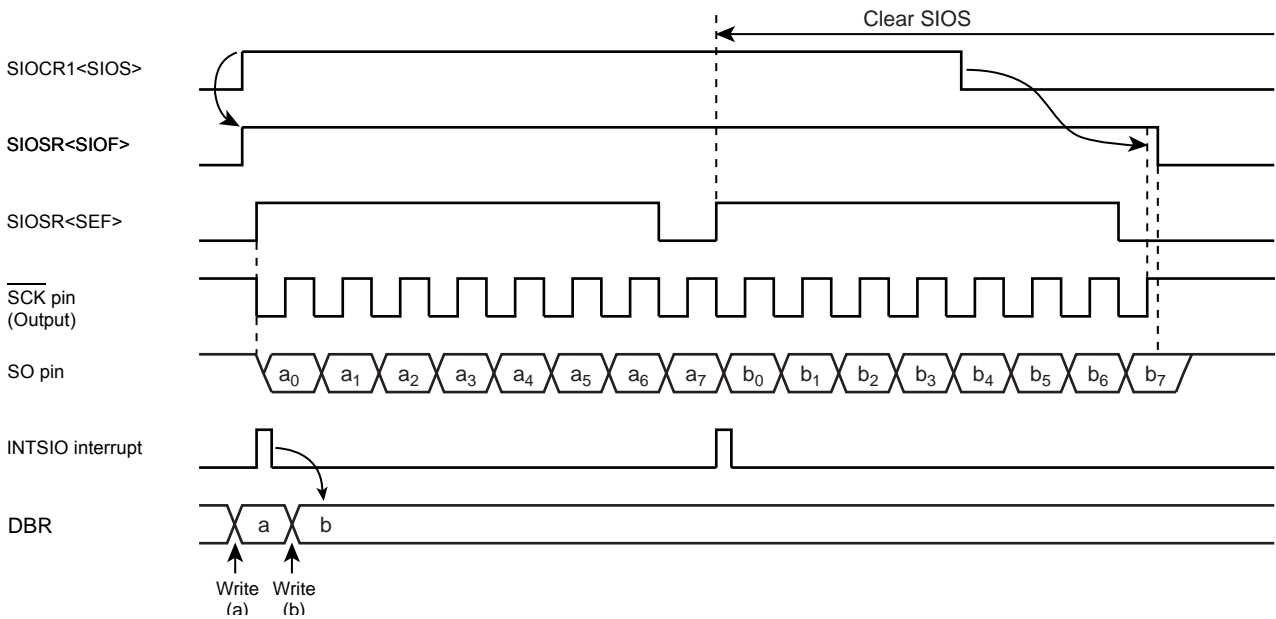


Figure 12-7 Transfer Mode (Example: 8bit, 1word transfer, Internal clock)

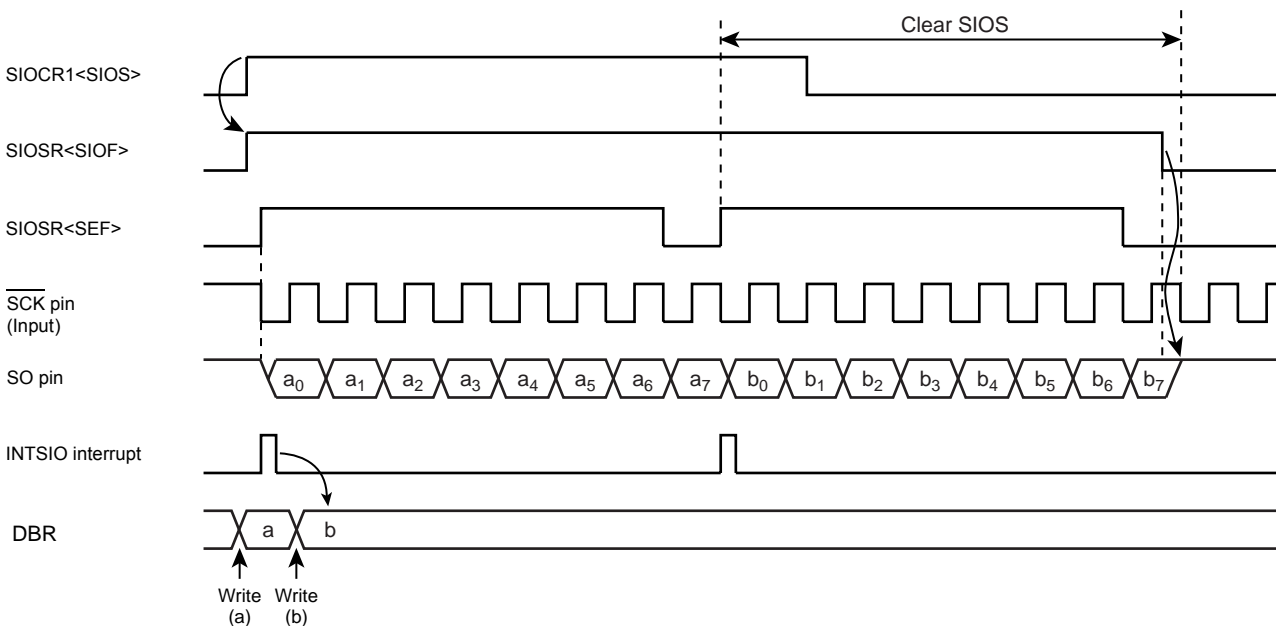


Figure 12-8 Transfer Mode (Example: 8bit, 1word transfer, External clock)

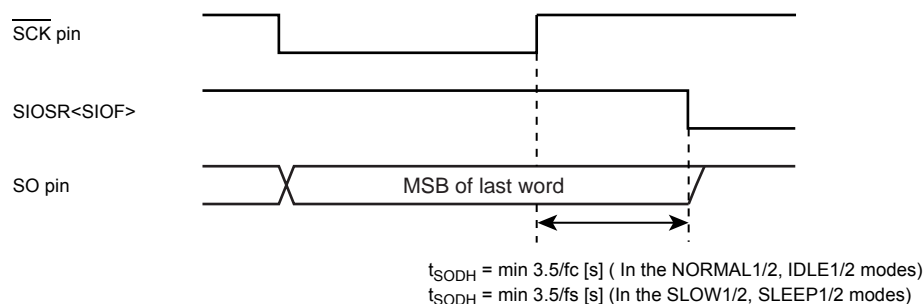


Figure 12-9 Transmitted Data Hold Time at End of Transfer

### 12.6.2 4-bit and 8-bit receive modes

After setting the control registers to the receive mode, set SIOCR1<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 SIOCR2<BUF> has been received, an INTSIO (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 SIO 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 SIOCR1<SIOS> to “0” or setting SIOCR1<SIOINH> to “1” in buffer full interrupt service program.

When SIOCR1<SIOS> is cleared, the current data are transferred to the buffer. After SIOCR1<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 SIOSR<SIOF>. SIOSR<SIOF> is cleared to “0” when the receiving is ended. After confirmed the receiving termination, the final receiving data is read. When SIOCR1<SIOINH> is set, the receiving is immediately ended and SIOSR<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, SIOCR1<SIOS> should be cleared to “0” then SIOCR2<BUF> must be rewritten after confirming that SIOSR<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, SIOCR2<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 SIOCR1<SIOS> to “0”, read the last data and then switch the transfer mode.

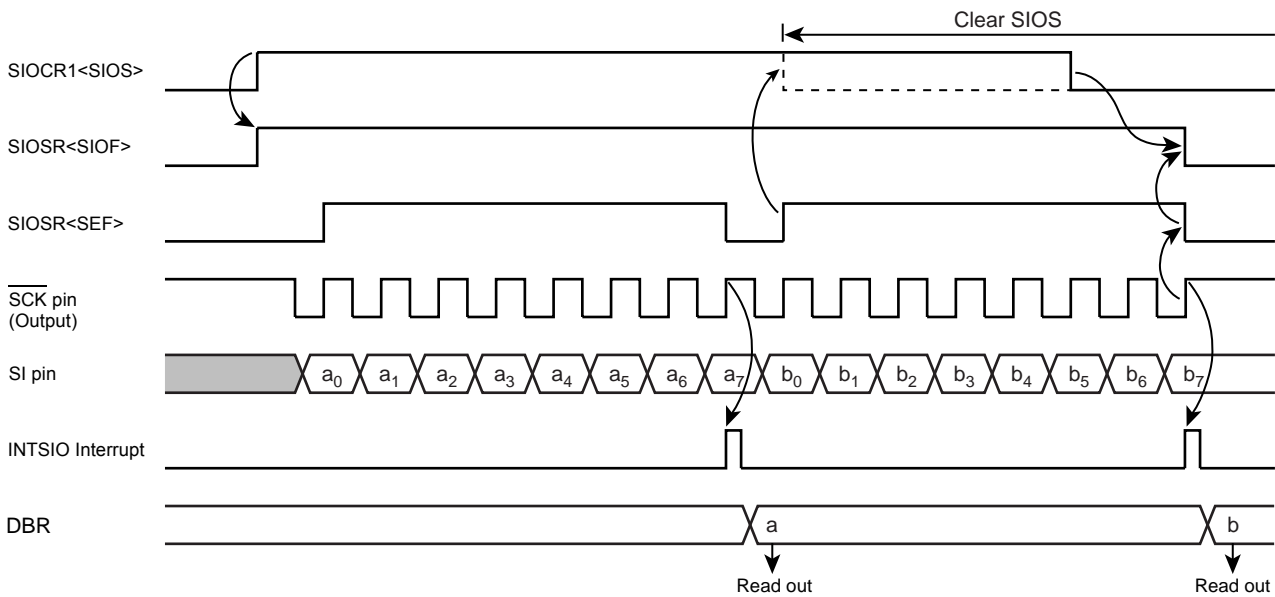


Figure 12-10 Receive Mode (Example: 8bit, 1word transfer, Internal clock)

### 12.6.3 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 SIOCR1<SIOS> to “1”. When transmitting, the data are output from the SO pin at leading edges of the serial clock. When receiving, the data are input to the SI 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 INTSIO interrupt is generated when the number of data words specified with the SIOCR2<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 SIOCR1<SIOS> to “0” or setting SIOCR1<SIOINH> to “1” in INTSIO interrupt service program.

When SIOCR1<SIOS> is cleared, the current data are transferred to the buffer. After SIOCR1<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 SIOSR<SIOF>. SIOSR<SIOF> is cleared to “0” when the transmitting/receiving is ended.

When SIOCR1<SIOINH> is set, the transmit/receive operation is immediately ended and SIOSR<SIOF> is cleared to “0”.

If it is necessary to change the number of words in external clock operation, SIOCR1<SIOS> should be cleared to “0”, then SIOCR2<BUF> must be rewritten after confirming that SIOSR<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, SIOCR2<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 SIOCR1<SIOS> to "0", read the last data and then switch the transfer mode.

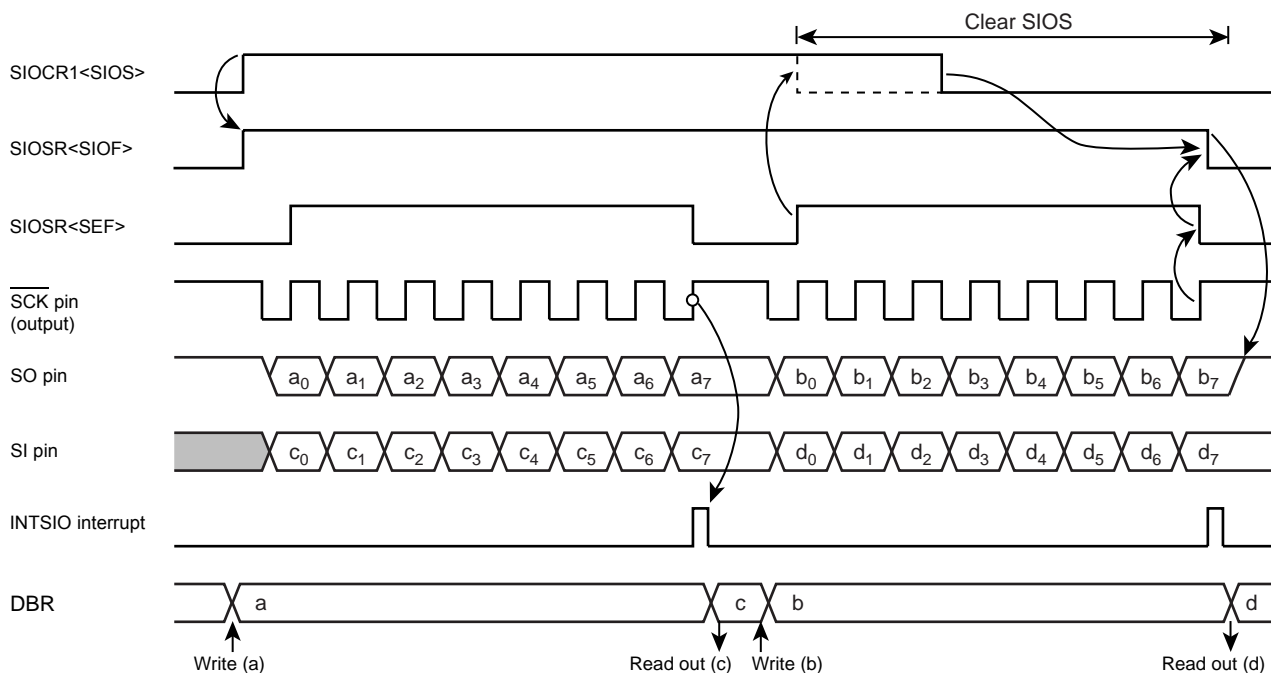


Figure 12-11 Transfer / Receive Mode (Example: 8bit, 1word transfer, Internal clock)

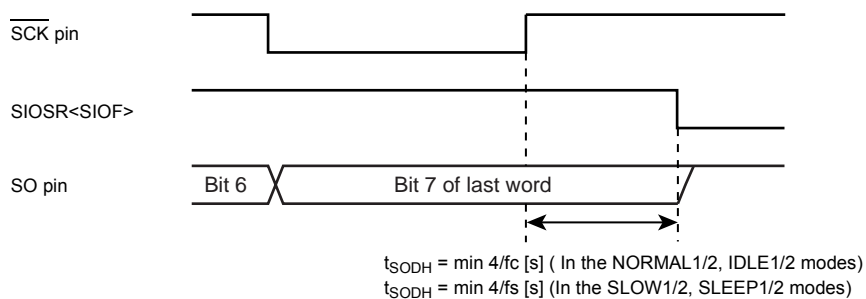


Figure 12-12 Transmitted Data Hold Time at End of Transfer / Receive



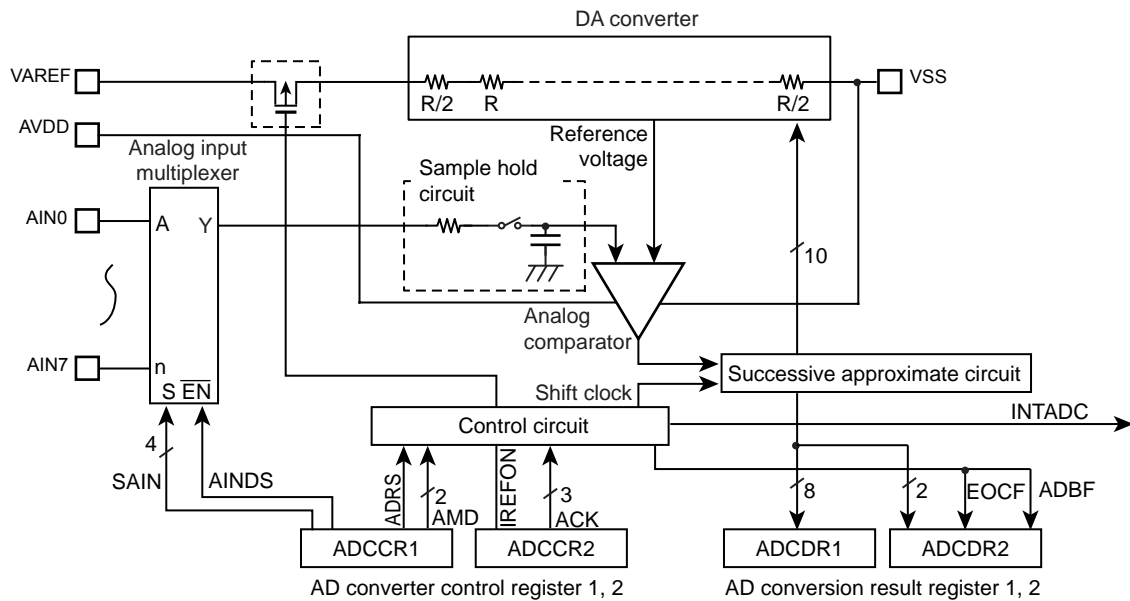
# 13. 10-bit AD Converter (ADC)

The TMP86FS23UG have a 10-bit successive approximation type AD converter.

## 13.1 Configuration

The circuit configuration of the 10-bit AD converter is shown in Figure 13-1.

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 13-1 10-bit AD Converter

## 13.2 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).

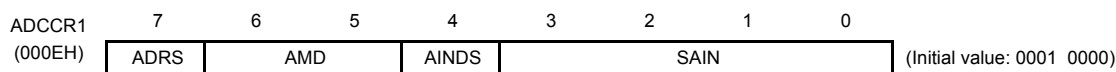
3. AD converted value register 1 (ADCDR1)

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: Reserved 1001: Reserved 1010: Reserved 1011: Reserved 1100: Reserved 1101: Reserved 1110: Reserved 1111: Reserved	

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".

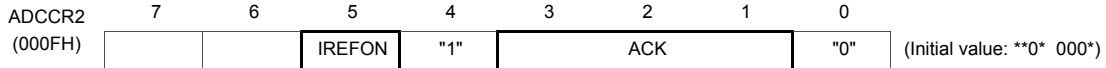
Note 3: During conversion, Do not perform port output instruction to maintain a precision for all of the pins because analog input port use as general input port. And for port near to analog input, Do not input intense signaling of change.

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	R/W
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	

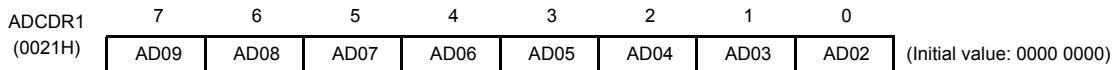
- Note 1: Always set bit0 in ADCCR2 to "0" and set bit4 in ADCCR2 to "1".  
 Note 2: When a read instruction for ADCCR2, bit6 to 7 in ADCCR2 read in as undefined data.  
 Note 3: After STOP or SLOW/SLEEP mode are started, AD converter control register2 (ADCCR2) is all initialized and no data can be written in this register. Therefore, to use AD converter again, set the ADCCR2 newly after returning to NORMAL1 or NORMAL2 mode.

**Table 13-1 ACK setting and Conversion time**

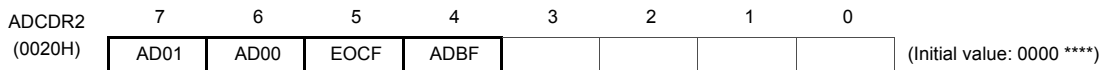
Condition ACK	Conversion time	16 MHz	8 MHz	4 MHz	2 MHz	10 MHz	5 MHz	2.5 MHz
000	39/fc	-	-	-	19.5 μs	-	-	15.6 μs
001	Reserved							
010	78/fc	-	-	19.5 μs	39.0 μs	-	15.6 μs	31.2 μs
011	156/fc	-	19.5 μs	39.0 μs	78.0 μs	15.6 μs	31.2 μs	62.4 μs
100	312/fc	19.5 μs	39.0 μs	78.0 μs	156.0 μs	31.2 μs	62.4 μs	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							

- 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.

### 13.3 Function

#### 13.3.1 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).

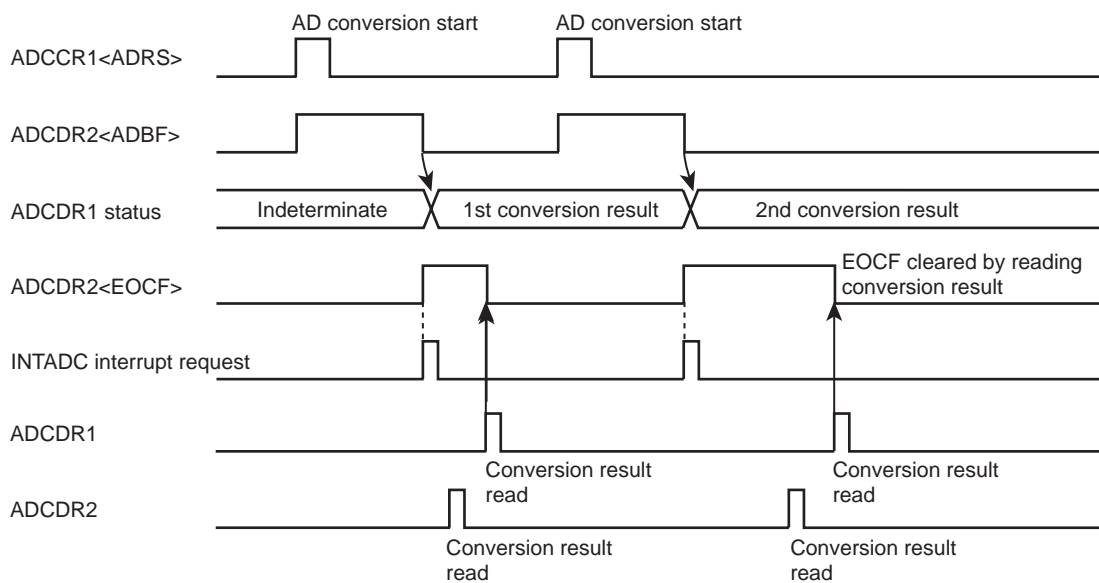


Figure 13-2 Software Start Mode

#### 13.3.2 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.

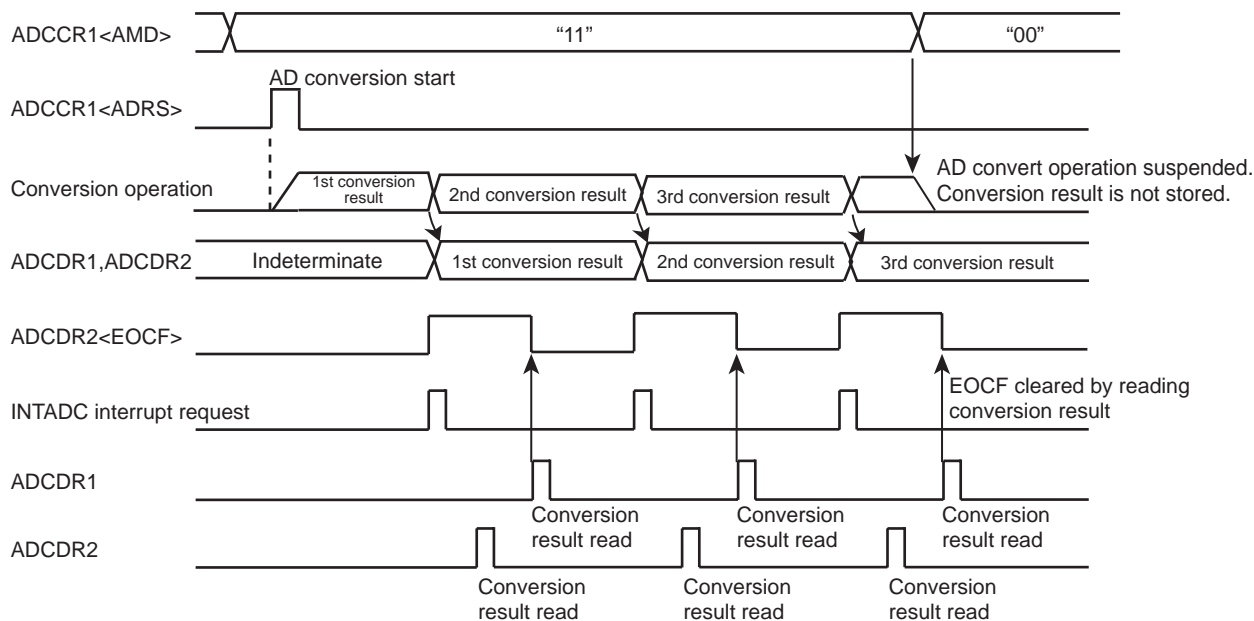


Figure 13-3 Repeat Mode

### 13.3.3 Register Setting

- 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).
- 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 13-1 and AD converter control register 2.
  - Choose IREFON for DA converter control.
- 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.
- 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.
- 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  $\mu$ 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
    
```

### 13.4 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.

## 13.5 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 13-4.

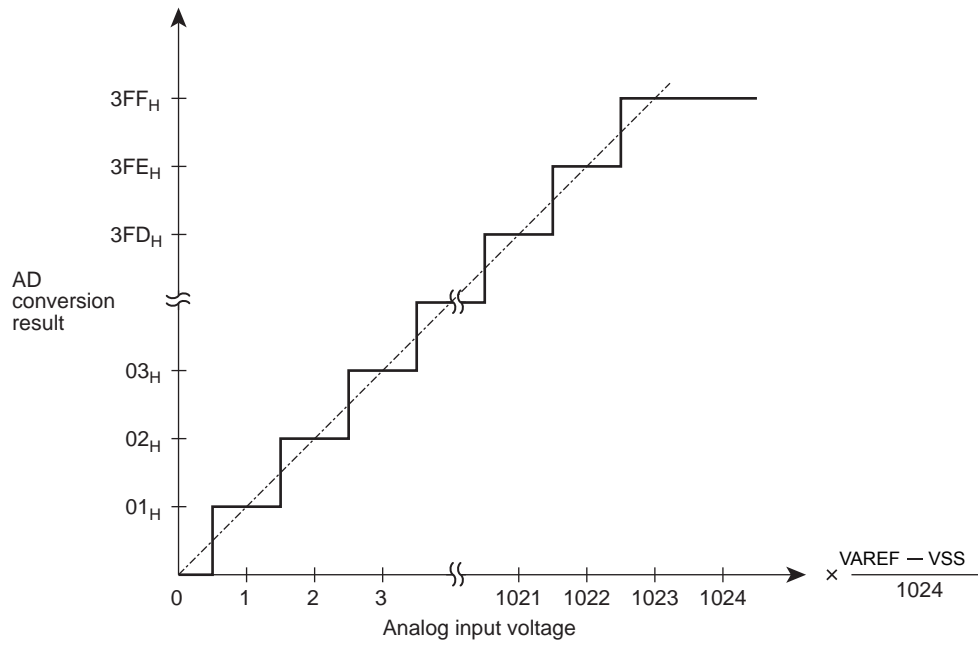


Figure 13-4 Analog Input Voltage and AD Conversion Result (Typ.)



## 13.6 Precautions about AD Converter

### 13.6.1 Restrictions for AD Conversion interrupt (INTADC) usage

When an AD interrupt is used, it may not be processed depending on program composition. For example, if an INTADC interrupt request is generated while an interrupt with priority lower than the interrupt latch IL15 (INTADC) is being accepted, the INTADC interrupt latch may be cleared without the INTADC interrupt being processed.

The completion of AD conversion can be detected by the following methods:

(1) Method not using the AD conversion end interrupt

Whether or not AD conversion is completed can be detected by monitoring the AD conversion end flag (EOCF) by software. This can be done by polling EOCF or monitoring EOCF at regular intervals after start of AD conversion.

(2) Method for detecting AD conversion end while a lower-priority interrupt is being processed

While an interrupt with priority lower than INTADC is being processed, check the AD conversion end flag (EOCF) and interrupt latch IL15. If  $IL15 = 0$  and  $EOCF = 1$ , call the AD conversion end interrupt processing routine with consideration given to PUSH/POP operations. At this time, if an interrupt request with priority higher than INTADC has been set, the AD conversion end interrupt processing routine will be executed first against the specified priority. If necessary, we recommend that the AD conversion end interrupt processing routine be called after checking whether or not an interrupt request with priority higher than INTADC has been set.

### 13.6.2 Analog input pin voltage range

Make sure the analog input pins (AIN0 to AIN7) are used at voltages within VAREF to VSS. 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.

### 13.6.3 Analog input shared pins

The analog input pins (AIN0 to AIN7) 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.

### 13.6.4 Noise Countermeasure

The internal equivalent circuit of the analog input pins is shown in Figure 13-5. 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 k $\Omega$  or less. Toshiba also recommends attaching a capacitor external to the chip.

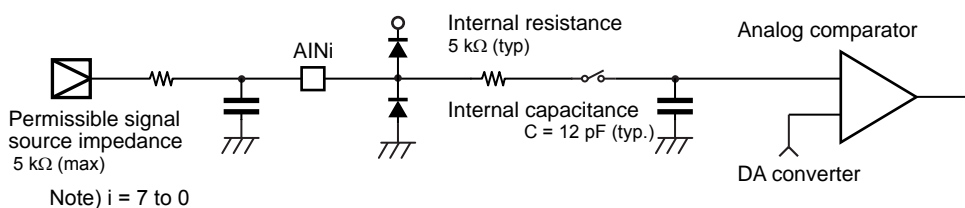


Figure 13-5 Analog Input Equivalent Circuit and Example of Input Pin Processing



# 14. Key-on Wakeup (KWU)

In the TMP86FS23UG, 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 " 14.2 Control ".

## 14.1 Configuration

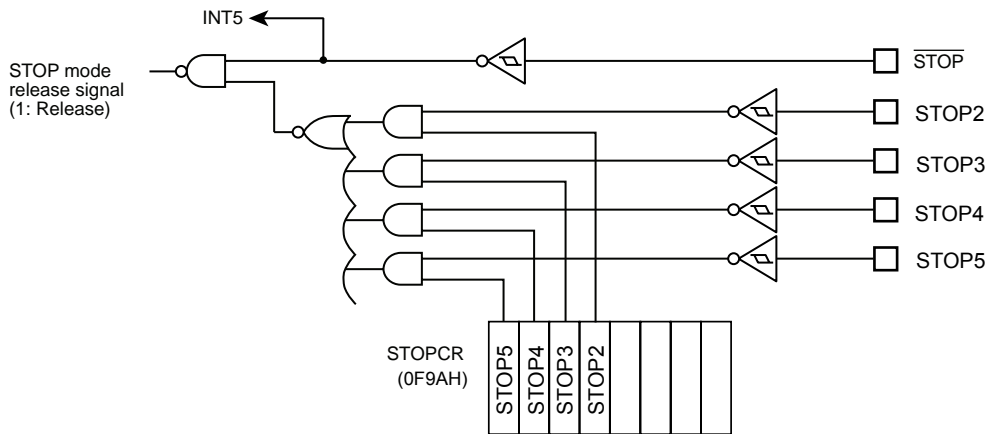


Figure 14-1 Key-on Wakeup Circuit

## 14.2 Control

STOP2 to STOP5 pins can 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	
(0F9AH)	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

## 14.3 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 14-2).

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 14-3).

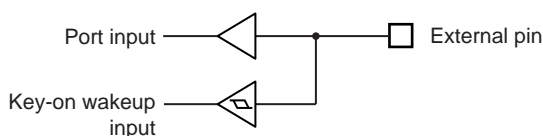


Figure 14-2 Key-on Wakeup Input and Port Input

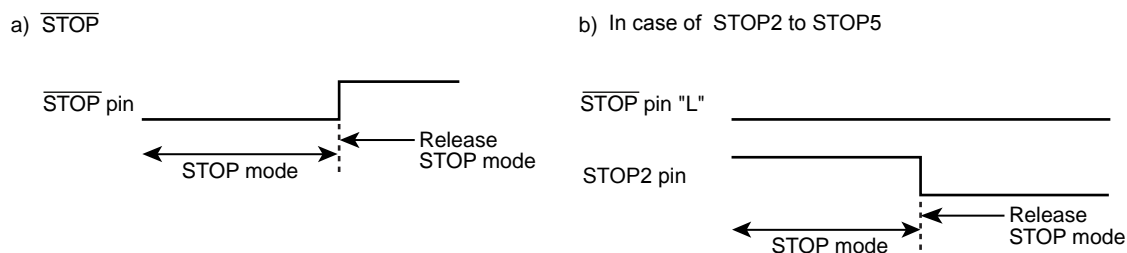


Figure 14-3 Priority of STOP pin and STOP2 to STOP5 pins

Table 14-1 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)

# 15. LCD Driver

The TMP86FS23UG has a driver and control circuit to directly drive the liquid crystal device (LCD). The pins to be connected to LCD are as follows:

1. Segment output port 32 pins (SEG31 to SEG0)
2. Common output port 4 pins (COM3 to COM0)

In addition, VLC pin is provided for the LCD power supply.

The devices that can be directly driven is selectable from LCD of the following drive methods:

1. 1/4 Duty (1/3 Bias) LCD Max 128 Segments (8 segments × 16 digits)
2. 1/3 Duty (1/3 Bias) LCD Max 96 Segments (8 segments × 12 digits)
3. 1/3 Duty (1/2 Bias) LCD Max 96 Segments (8 segments × 12 digits)
4. 1/2 Duty (1/2 Bias) LCD Max 64 Segments (8 segments × 8 digits)
5. Static LCD Max 32 Segments (8 segments × 4 digits)

## 15.1 Configuration

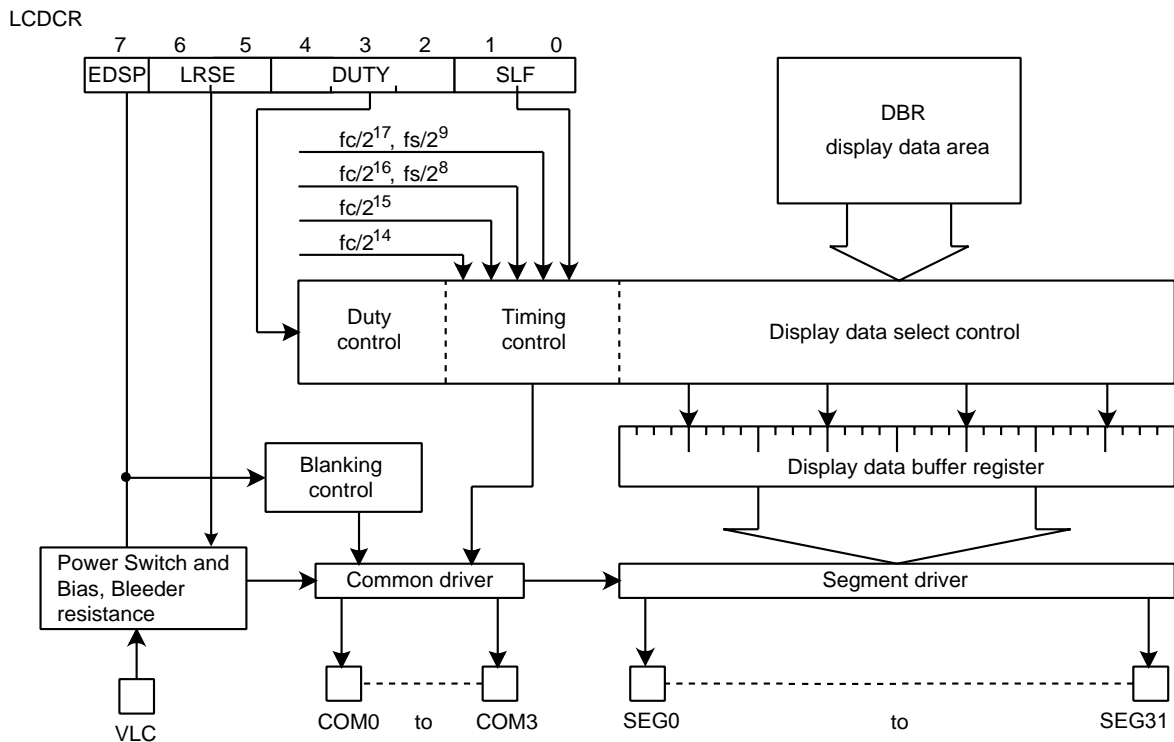


Figure 15-1 LCD Driver

## 15.2 Control

The LCD control register (LCDCR) controls the LCD driver. EDSP specifies whether to enable the LCD display. If EDSP is cleared to “0” for blanking, the power switch for the VLC pin is turned off. So, the COM pin and pin output selected with SEG enter GND level.

### LCD Driver Control Register

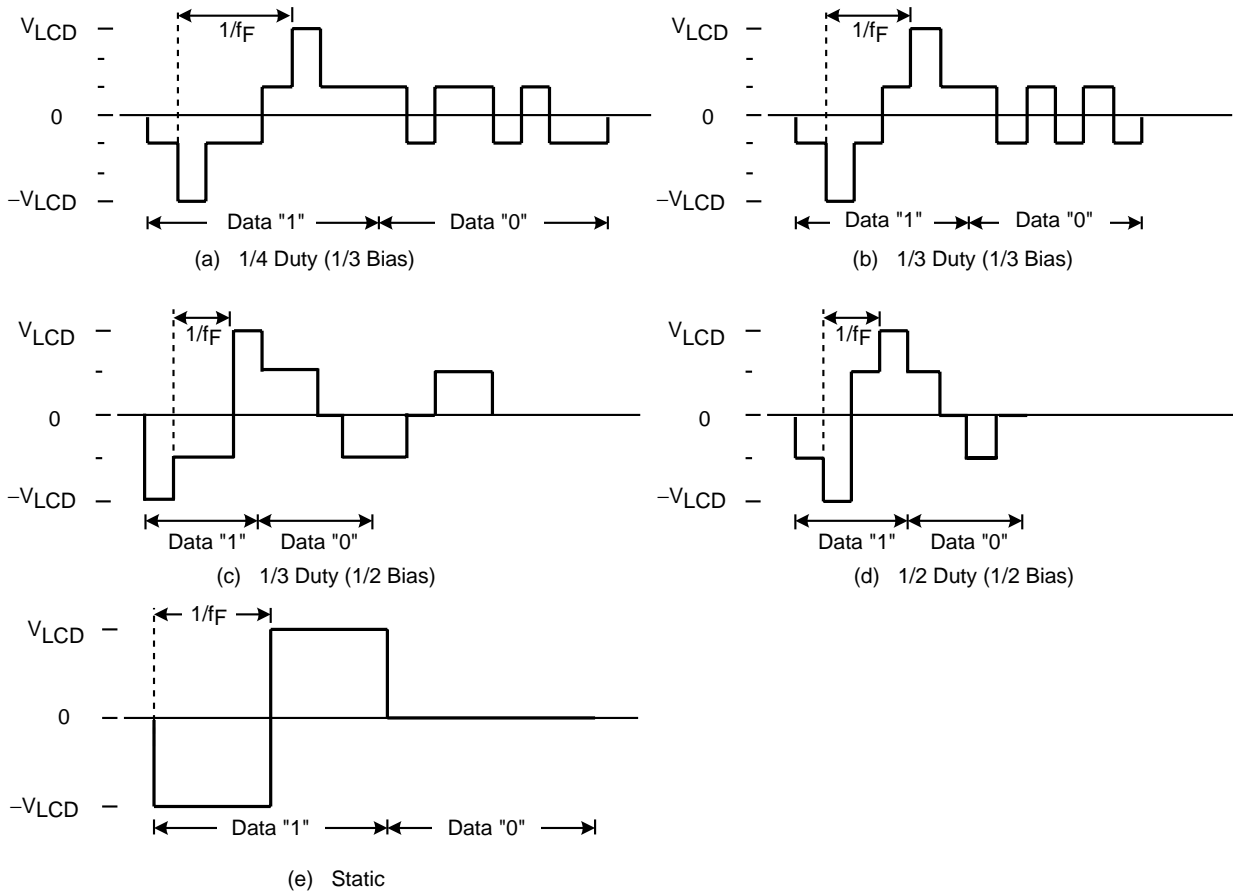
LCDCR (0027H)	7	6	5	4	3	2	1	0	
	EDSP	LRSEL		DUTY			SLF		(Initial value: 0000 0000)

Field	Description	Value	Mode	Value	Mode	Access			
EDSP	LCD display control	0: Blanking 1: Enables LCD display (Blanking is released)				R/W			
LRSE	Period selection of enabling (turn on) of the low bleeder resistor (for implementing appropriate LCD panel drive capability)	NORMAL1/2, IDLE1/2 mode		SLOW1/2, SLEEP1/2 mode					
		SLF Setting		SLF Setting					
			11	10	01		00	01	00
		00:	$2^9/fc$	$2^7/fc$	$2^8/fc$		$2^9/fc$	1/fs	2/fs
		01:	$2^9/fc$	$2^{10}/fc$	$2^{11}/fc$		$2^{12}/fc$	$2^3/fs$	$2^4/fs$
		10:	Always enabling						
11:	Reserved								
DUTY	Selection of driving methods	000: 1/4 Duty (1/3 Bias) 001: 1/3 Duty (1/3 Bias) 010: 1/3 Duty (1/2 Bias) 011: 1/2 Duty (1/2 Bias) 100: Static 101: Reserved 110: Reserved 111: Reserved							
SLF	Selection of LCD frame frequency		NORMAL1/2, IDLE0/1/2 mode		SLOW1/2, SLEEP1/2 mode				
		00:	$fc/2^{17}$ [Hz]		$fs/2^9$ [Hz]				
		01:	$fc/2^{16}$		$fs/2^8$				
		10:	$fc/2^{15}$		Reserved				
	11:	$fc/2^{13}$		Reserved					

- Note 1: The base-frequency (SLF) source clock is switched between high and low frequencies by the SYSCR2<SYSCK> programming. The base frequency does not depend on the TBTCR<DV7CK> programming.
- Note 2: If the setting of SYSCR2<SYSCK> is changed, be sure to turn off the LCD (clear EDSP to “0”) to avoid the output of incorrect waveform.
- Note 3: Programming LRSE properly according to the LCD panel used. As the LRSE programming increases (lengthen the period of enabling of the low resistor), the drive capability becomes higher while the power dissipation increases. Reversely, as the LRSE programming decreases shorten the period of enabling of the low resistor, the drive capability becomes lower while the power consumption decreases.
- Note 4: If the IDLE0, SLEEP0, or STOP mode is activated when the display is enabled, LCDCR<EDSP> is automatically changed to “0” to blank the display.

### 15.2.1 LCD driving methods

As for LCD driving method, 5 types can be selected by LCDCCR<DUTY>. The driving method is initialized in the initial program according to the LCD used.



Note 1:  $f_F$ : Frame frequency

Note 2:  $V_{LCD}$ : LCD drive voltage (=  $V_{DD} - V_{LC}$ )

Figure 15-2 LCD Drive Waveform (COM - SEG pins)

## 15.2.2 Frame frequency

Frame frequency ( $f_F$ ) is set according to driving method and base frequency as shown in the following Table 15-1. The base frequency is selected by LCDCR<SLF> according to the frequency  $f_c$  and  $f_s$  of the basic clock to be used.

Table 15-1 Setting of LCD Frame Frequency for high frequency clock

(a) At the SYSCR2<SYSCK> = "0".

SLF	Base Frequency [Hz]	Frame Frequency [Hz]			
		1/4 Duty	1/3 Duty	1/2 Duty	Static
00	$\frac{f_c}{2^{17}}$	$\frac{f_c}{2^{17}}$	$\frac{4}{3} \cdot \frac{f_c}{2^{17}}$	$\frac{4}{2} \cdot \frac{f_c}{2^{17}}$	$\frac{f_c}{2^{17}}$
	( $f_c = 16$ MHz)	122	163	244	122
	( $f_c = 8$ MHz)	61	81	122	61
01	$\frac{f_c}{2^{16}}$	$\frac{f_c}{2^{16}}$	$\frac{4}{3} \cdot \frac{f_c}{2^{16}}$	$\frac{4}{2} \cdot \frac{f_c}{2^{16}}$	$\frac{f_c}{2^{16}}$
	( $f_c = 8$ MHz)	122	163	244	122
	( $f_c = 4$ MHz)	61	81	122	61
10	$\frac{f_c}{2^{15}}$	$\frac{f_c}{2^{15}}$	$\frac{4}{3} \cdot \frac{f_c}{2^{15}}$	$\frac{4}{2} \cdot \frac{f_c}{2^{15}}$	$\frac{f_c}{2^{15}}$
	( $f_c = 4$ MHz)	122	163	244	122
	( $f_c = 2$ MHz)	61	81	122	61
11	$\frac{f_c}{2^{14}}$	$\frac{f_c}{2^{14}}$	$\frac{4}{3} \cdot \frac{f_c}{2^{14}}$	$\frac{4}{2} \cdot \frac{f_c}{2^{14}}$	$\frac{f_c}{2^{14}}$
	( $f_c = 2$ MHz)	122	162	244	122
	( $f_c = 1$ MHz)	61	81	122	61

Note:  $f_c$ : High-frequency clock [Hz]

Table 15-2 Setting of LCD Frame Frequency for low frequency clock

(b) At the SYSCR2<SYSCK> = "1".

SLF	Base Frequency [Hz]	Frame Frequency [Hz]			
		1/4 Duty	1/3 Duty	1/2 Duty	Static
00	$\frac{f_s}{2^9}$	$\frac{f_s}{2^9}$	$\frac{4}{3} \cdot \frac{f_s}{2^9}$	$\frac{4}{2} \cdot \frac{f_s}{2^9}$	$\frac{f_s}{2^9}$
	( $f_s = 32.768$ kHz)	64	85	128	64
01	$\frac{f_s}{2^8}$	$\frac{f_s}{2^8}$	$\frac{4}{3} \cdot \frac{f_s}{2^8}$	$\frac{4}{2} \cdot \frac{f_s}{2^8}$	$\frac{f_s}{2^8}$
	( $f_s = 32.768$ kHz)	128	171	256	128
1*	Reserved				

Note:  $f_s$ : Low-frequency clock [Hz]



### 15.2.3 LCD drive voltage

LCD driving voltage VLCD is given as potential difference  $VDD - VLC$  between pins VDD and VLC. Therefore, when the CPU voltage and LCD drive voltage are the same, VLC pin will be connected to VSS pin. The LCD lights when the potential difference between segment output and common output is  $\pm VLCD$ . Otherwise it turns off.

During reset, the power switch of LCD driver is automatically turned off, shutting off the VLC voltage.

After reset, if the P\*LCR register (\*; Port No.) for each port is set to “1” with  $LCDCR<EDSP> = “0”$ , a GND level is output from the pin which can be used as segment.

The power switch is turned on to supply VLC voltage to LCD driver by setting with  $LCDCR<EDSP>$  to “1”.

If the IDLE0, SLEEP0, or STOP mode is activated,  $LCDCR<EDSP>$  is automatically changed to “0” to blank the display. To turn the display back on after releasing from the previous mode, set  $LCDCR<EDSP>$  to “1” again.

Note: During reset, the LCD common outputs (COM3 to COM0) are fixed “0” level. However, the multiplex port (input/output port or SEG output is selectable) becomes high impedance. Therefore, when the reset input is long remarkably, ghost problem may appear in LCD display.

### 15.2.4 Adjusting the LCD panel drive capability

The LCD panel drive capability can be adjusted by programming  $LCDCR<LRSE>$ . When the period of enabling of the low bleeder resistor is lengthened, the drive capability becomes higher while the power consumption increases. Reversely, when the period of enabling of the low bleeder resistor is shortened, the drive capability becomes lower while the power consumption decreases. If the drive capability is not enough, the LCD display might present a ghost problem. So, implement the optimum drive capability for the LCD panel used. The figure below shows the bleeder resistance timing and equivalent circuit for 1/4 duty and 1/3 bias.

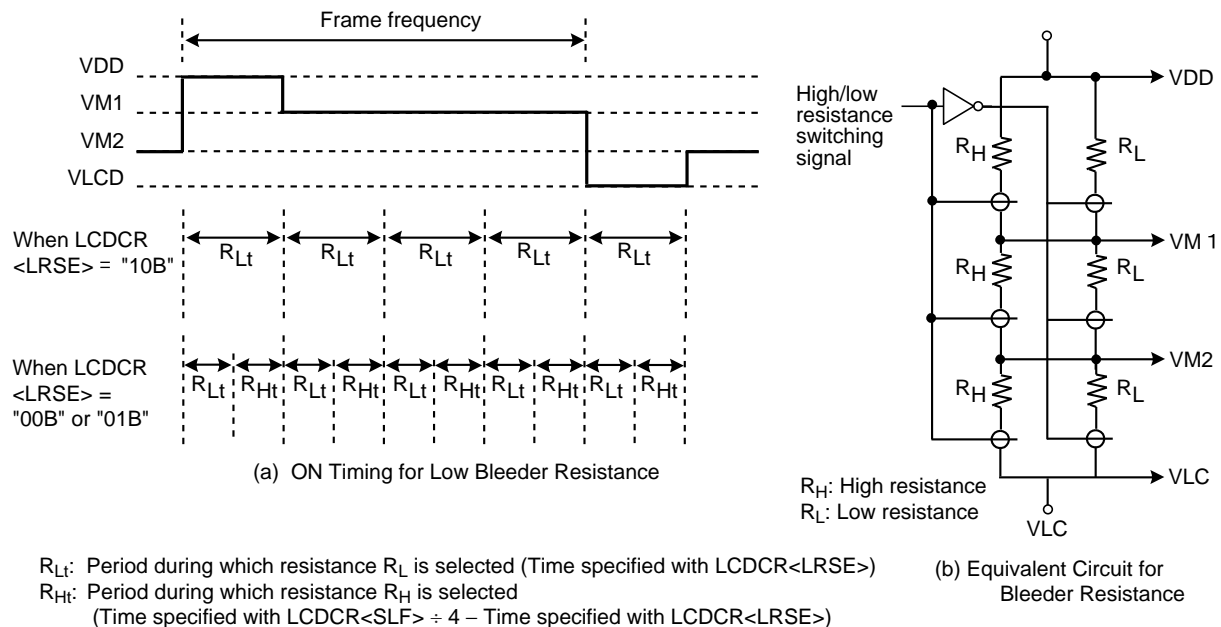


Figure 15-3 Bleeder Resistance Selection with  $LCDCR<LRSE>$  (for 1/4 duty and 1/3 bias)

## 15.3 LCD Display Operation

### 15.3.1 Display data setting

Display data is stored to the display data area (address 0F80H to 0F8FH, 16 bytes) in the DBR. The display data stored in the display data area is automatically read out and sent to the LCD driver by the hardware. The LCD driver generates the segment signal and common signal according to the display data and driving method. Therefore, display patterns can be changed by only over writing the contents of display data area by the program. Table 15-4 shows the correspondence between the display data area and SEG/COM pins.

LCD light when display data is “1” and turn off when “0”. According to the driving method of LCD, the number of pixels which can be driven becomes different, and the number of bits in the display data area which is used to store display data also becomes different.

Therefore, the bits which are not used to store display data as well as the data buffer which corresponds to the addresses not connected to LCD can be used to store general user process data (see Table 15-3).

Table 15-3 Driving Method and Bit for Display Data

Driving methods	Bit 7/3	Bit 6/2	Bit 5/1	Bit 4/0
1/4 Duty	COM3	COM2	COM1	COM0
1/3 Duty	–	COM2	COM1	COM0
1/2 Duty	–	–	COM1	COM0
Static	–	–	–	COM0

Note: –: This bit is not used for display data

Table 15-4 LCD Display Data Area (DBR)

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0F80H		SEG1				SEG0		
0F81H		SEG3				SEG2		
0F82H		SEG5				SEG4		
0F83H		SEG7				SEG6		
0F84H		SEG9				SEG8		
0F85H		SEG11				SEG10		
0F86H		SEG13				SEG12		
0F87H		SEG15				SEG14		
0F88H		SEG17				SEG16		
0F89H		SEG19				SEG18		
0F8AH		SEG21				SEG20		
0F8BH		SEG23				SEG22		
0F8CH		SEG25				SEG24		
0F8DH		SEG27				SEG26		
0F8EH		SEG29				SEG28		
0F8FH		SEG31				SEG30		
	COM3	COM2	COM1	COM0	COM3	COM2	COM1	COM0

### 15.3.2 Blanking

Blanking is enabled when LCDCR<EDSP> is cleared to “0”.

To blank the LCD display and turn it off, a GND-level signal is output to the COM pin and the port which can be used as the segment by setting of P\*LCR register (\*; Port No.). At this time, the power switch of VLC pin is turned off.

## 15.4 Control Method of LCD Driver

### 15.4.1 Initial setting

Figure 15-4 shows the flowchart of initialization.

Example :To operate a 1/4 duty LCD of 32 segments × 4 com-mons at frame frequency  $fc/2^{16}$  [Hz],  
 The period of enabling of the low bleeder resistor:  $2^8/fc$

LD	(LCDCR), 00000001B	; Sets LCD driving method, The period of enabling of low bleeder resistor and frame frequency.
LD	(P*LCR), 0FFH	; Sets segment output control register. (*; Port No.)
:	:	
:	:	; Sets the initial value of display data.
LD	(LCDCR), 10000001B	; Display enable

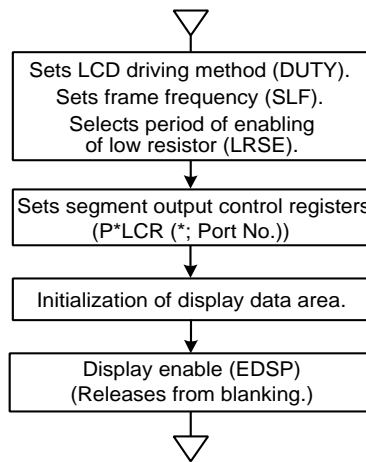


Figure 15-4 Initial Setting of LCD Driver

### 15.4.2 Store of display data

Generally, display data are prepared as fixed data in program memory (ROM) and stored in display data area by load command.

Example :(1) To display using 1/4 duty LCD a numerical value which corresponds to the LCD data stored in data memory at address 80H (when pins COM and SEG are connected to LCD as in Figure 15-5), display data become as shown in Table 15-5.

	LD	A, (80H)
	ADD	A, TABLE-\$-7
	LD	HL, 0F85H
	LD	W, (PC + A)
	LD	(HL), W
	RET	
TABLE:	DB	11011111B, 00000110B, 11100011B, 10100111B, 00110110B, 10110101B, 11110101B, 00010111B, 11110111B, 10110111B

Note: DB is a byte data definition instruction.

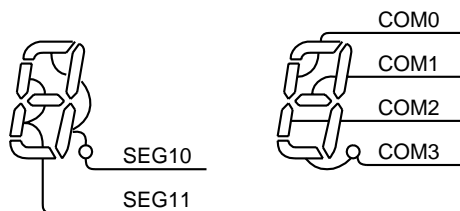
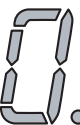
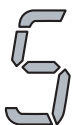




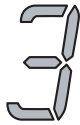

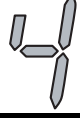



Figure 15-5 Example of COM, SEG Pin Connection (1/4 duty)

Table 15-5 Example of Display Data (1/4 duty)

No.	Display	Display data	No.	Display	Display data
0		11011111	5		10110101
1		00000110	6		11110101
2		11100011	7		00000111
3		10100111	8		11110111
4		00110110	9		10110111

Example: (2) Table 15-6 shows an example of display data which are displayed using 1/2 duty LCD in the same way as Table 15-5. The connection between pins COM and SEG are the same as shown in Figure 15-6.

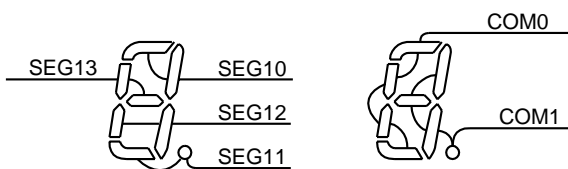


Figure 15-6 Example of COM, SEG Pin Connection

Table 15-6 Example of Display Data (1/2 duty)

Number	Display data		Number	Display data	
	High order address	Low order address		High order address	Low order address
0	**01**11	**01**11	5	**11**10	**01**01
1	**00**10	**00**10	6	**11**11	**01**01
2	**10**01	**01**11	7	**01**10	**00**11
3	**10**10	**01**11	8	**11**11	**01**11
4	**11**10	**00**10	9	**11**10	**01**11

Note: \*: Don't care

15.4.3 Example of LCD driver output

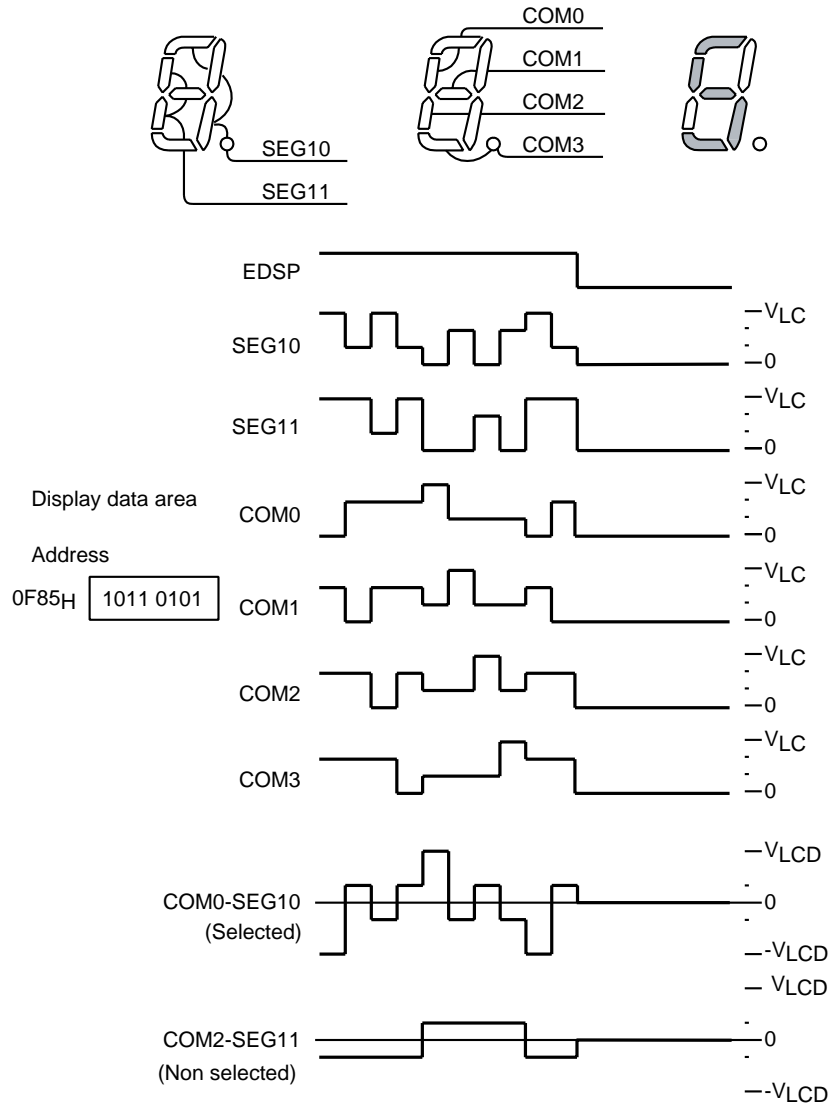


Figure 15-7 1/4 Duty (1/3 Bias) Drive

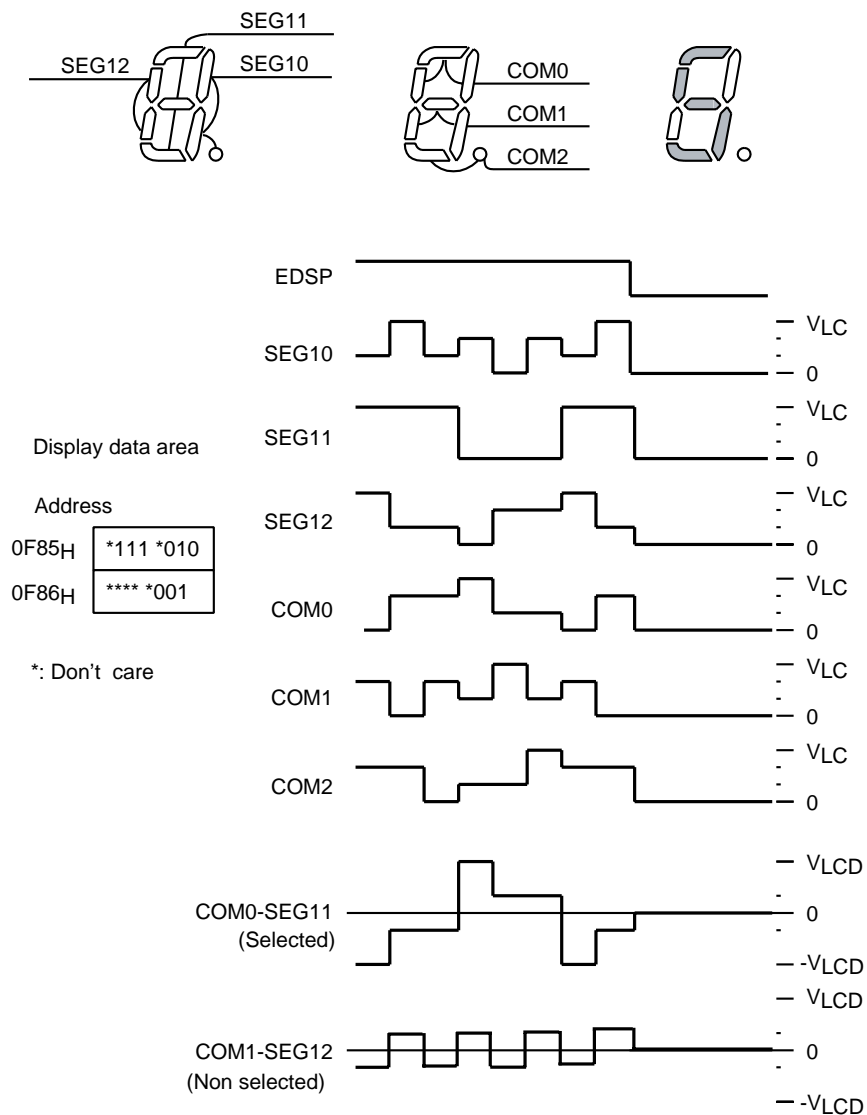


Figure 15-8 1/3 Duty (1/3 Bias) Drive

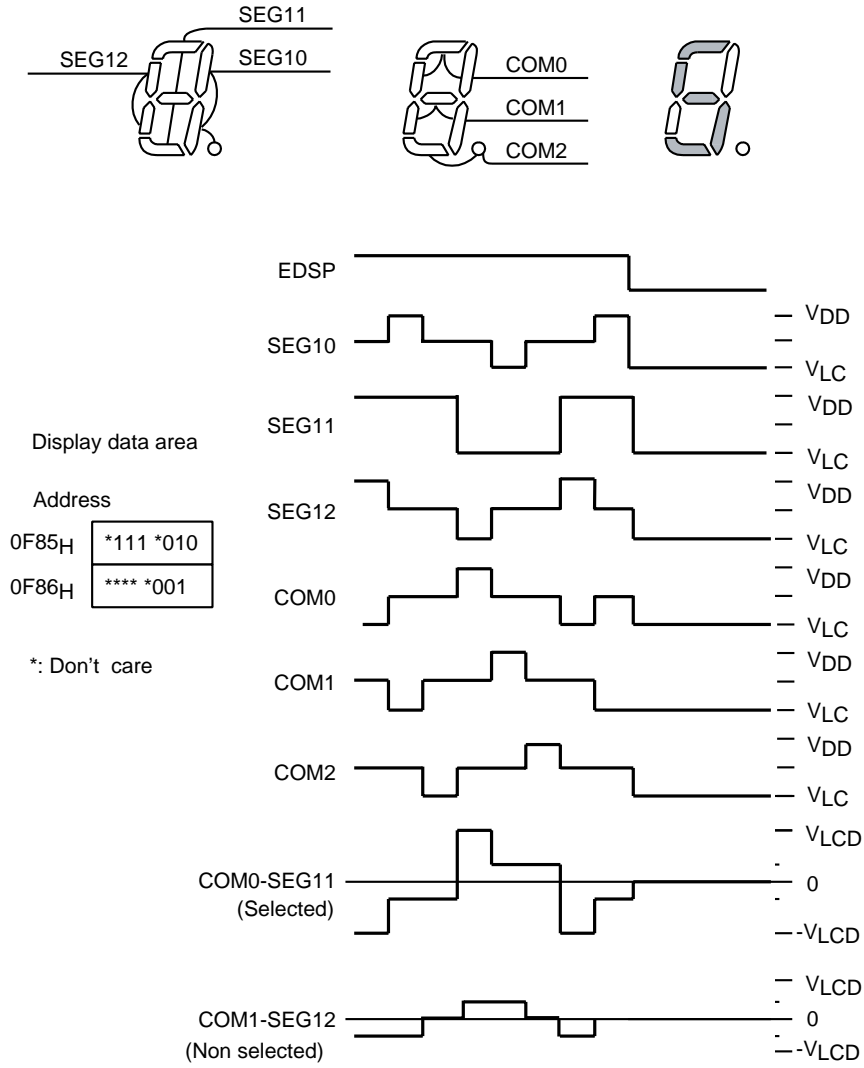


Figure 15-9 1/3 Duty (1/2 Bias) Drive

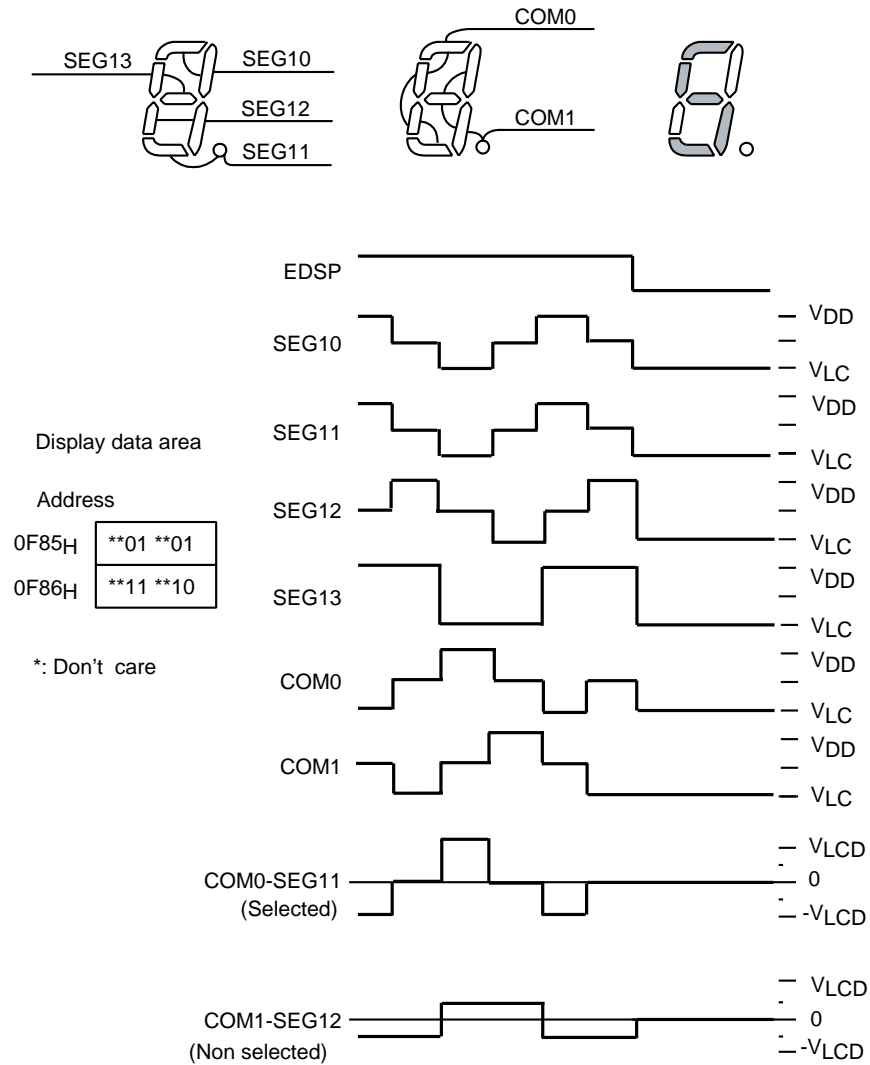


Figure 15-10 1/2 Duty (1/2 Bias) Drive



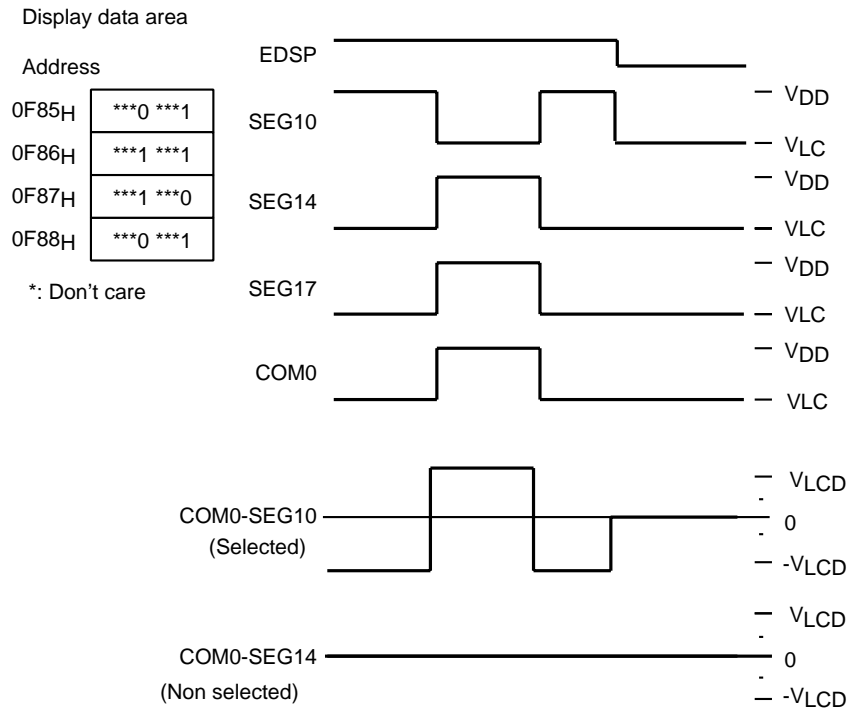
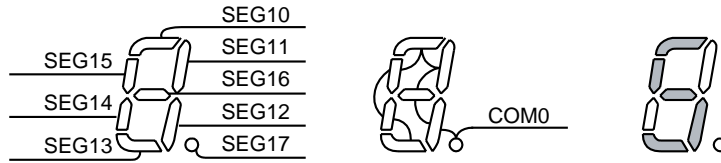


Figure 15-11 Static Drive



# 16. Real-Time Clock

The TMP86FS23UG include a real time counter (RTC). A low-frequency clock can be used to provide a periodic interrupt (0.0625[s],0.125[s],0.25[s],0.50[s]) at a programmed interval, implement the clock function. The RTC can be used in the mode in which the low-frequency oscillator is active (except for the SLEEP0 mode).

## 16.1 Configuration

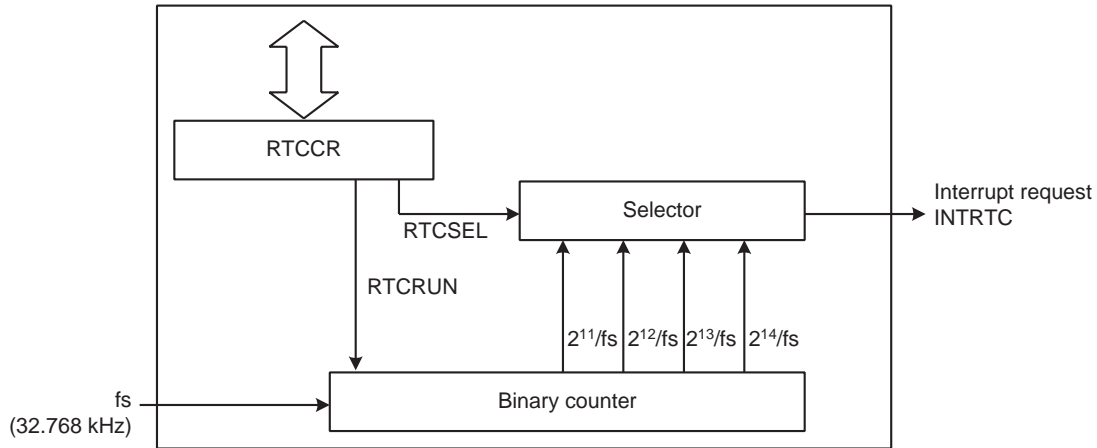


Figure 16-1 Configuration of the RTC

## 16.2 Control of the RTC

The RTC is controlled by the RTC control register (RTCCR).

### RTC Control Register

RTCCR (0017H)	7	6	5	4	3	2	1	0	
						RTCSEL		RTCRUN	(Initial value: **** *000)

RTCSEL	Interrupt generation period (fs = 32.768 kHz)	00: 0.50 [s] 01: 0.25 [s] 10: 0.125 [s] 11: 0.0625 [s]	R/W
RTCRUN	RTC control	0: Stops and clears the binary counter. 1: Starts counting	

Note 1: Program the RTCCR during low-frequency oscillation (when SYSCR2<XTEN> = "1"). For selecting an interrupt generation period, program the RTCSEL when the timer is inactive (RTCRUN = "0"). During the timer operation, do not change the RTCSEL programming at the same moment the timer stops.

Note 2: The timer automatically stops, and this register is initialized (the timer's binary counter is also initialized) if one of the following operations is performed while the timer is active:

1. Stopping the low-frequency oscillation (with SYSCR2<XTEN> = "0")
2. When the TMP86FS23UG are put in STOP or SLEEP0 mode

Therefore, before activating the timer after releasing from STOP or SLEEP0 mode, reprogram the registers again.

Note 3: If a read instruction for RTCCR is executed, undefined value is set to bits 7 to 3.

Note 4: If break processing is performed on the debugger for the development tool during the timer operation, the timer stops counting (contents of the RTCCR isn't altered). When the break is cancelled, processing is restarted from the point at which it was suspended.

## 16.3 Function

The RTC counts up on the internal low-frequency clock. When RTCCR<RTCRUN> is set to “1”, the binary counter starts counting up. Each time the end of the period specified with RTCCR<RTCSEL> is detected, an INTRTC interrupt is generated, and the binary counter is cleared. The timer continues counting up even after the binary counter is cleared.

# 17. Multiply-Accumulate (MAC) Unit

The TMP86FS23UG includes a multiply-accumulate (MAC) unit.

The MAC unit is capable of executing 16-bit × 16-bit multiplications and 16-bit × 16-bit + 32-bit multiply-accumulate operations.

The MAC unit supports only integer arithmetic, not fixed-point or floating-point arithmetic. Both signed and unsigned operations can be performed.

The MAC unit can only be used in NORMAL1 or NORMAL2 mode. All the registers of the MAC unit are initialized upon entering a mode other than NORMAL mode.

With development tools, if break mode is entered while the MAC unit is calculating, the calculation is continued but its result is unpredictable. In this case, the calculation must be re-executed after break mode is exited. Do not write to the multiplicand register in break mode. When the calculation is completed, it is possible to enter break mode and read the calculation result in break mode.

## 17.1 Configuration

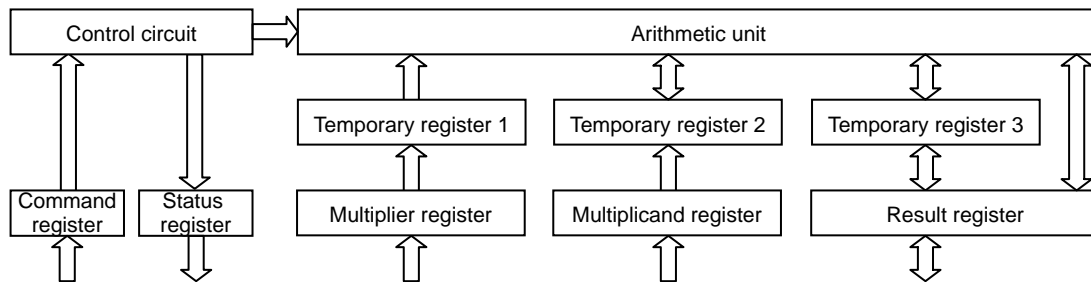


Figure 17-1 MAC Unit

## 17.2 Registers

The MAC unit consists of the following registers:

Table 17-1 Registers in the MAC Unit

Register	Address	Number of Bits
Command register (MACCR)	0FA4H	8 bits
Status register (MACSR)	0FA5H	8 bits
Multiplier data register (MPLDRH, MPLDRL)	0FA7H, 0FA6H	16 bits
Multiplicand data register (MPCDRH, MPCDRL)	0FA9H, 0FA8H	16 bits
Result register (RCALDR4 to RCALDR1)	0FAAH to 0FADH	32 bits
Addend register (MADDR4 to MADDR1)	0FAAH to 0FADH	32 bits

### 17.2.1 Command Register

The command register is used to enable and disable the MAC unit, specify the arithmetic mode, and clear the result register.

### 17.2.2 Status Register

The status register contains flags to indicate the operation status of the MAC unit and the calculation result.

### 17.2.3 Multiplier data Register

The data written to this register is calculated as a multiplier.

### 17.2.4 Multiplicand data Register

The data written to this register is calculated as a multiplicand.

### 17.2.5 Result Register

The calculation result is stored in this register.

### 17.2.6 Addend Register

The data written to this register is calculated as an addend in a multiply-accumulate operation. An addend must be written to this register while calculation is not being performed (CALC = "0").

## 17.3 Control

### Command Register

MACCR (0FA4H)	7	6	5	4	3	2	1	0	
	RCLR	"1"	"1"	"1"	CMOD		EMAC		(Initial value: 0111 0000)

RCLR	Result register clear	0: -(Keeps the value of the result register.) 1: Clears the result register. (This bit is automatically cleared to "0" one machine cycle after it is set to "1".)	R/W
CMOD	Arithmetic mode	000: Unsigned multiply (16 bits × 16 bits) 001: Unsigned multiply-accumulate (16 bits × 16 bits + 32 bits) 010: Signed multiply (16 bits × 16 bits) 011: Signed multiply-accumulate (16 bits × 16 bits + 32 bits) 1**: Reserved	
EMAC	MAC unit control	0: Disables the MAC unit. 1: Enables the MAC unit.	

Note 1: Setting RCLR to "1" causes the result, addend, and status registers to be initialized. The multiplier, multiplicand, and command registers remain the same as before. (RCLR is automatically cleared to "0" one machine cycle after it is set to "1".)

Note 2: Writing to CMOD (including an overwrite) makes no changes to the status, multiplier, multiplicand, result, and addend registers.

Note 3: Before changing the arithmetic mode, be sure to check that calculation is not being performed (CALC = "0").

Note 4: Clearing the result register with RCLR is possible only when calculation is not being performed (CALC = "0"). (RCLR cannot be set to "1" during calculation.)

Note 5: Bits 6 to 4 are always read as "1". ("0" cannot be written.)

### Status Register

MACSR (0FA5H)	7	6	5	4	3	2	1	0	
	"1"	"1"	"1"	CARF	ZERF	SIGN	OVRF	CALC	(Initial value: 1110 0000)

CARF	Carry flag	0: No carry occurred in multiply-accumulate operation. 1: Carry occurred in multiply-accumulate operation.	Read only
ZERF	Zero flag	0: Calculation result is other than "00000000H". 1: Calculation result is "00000000H".	
SIGN	Sign flag	0: Result register contents are positive or "00000000H". 1: Result register contents are negative.	
OVRF	Overflow flag	0: Overflow occurred. 1: No overflow occurred.	
CALC	Operation status flag	0: Calculation not in progress 1: Calculation in progress	

Note 1: The status register is initialized when the result register is cleared (RCLR = "1").

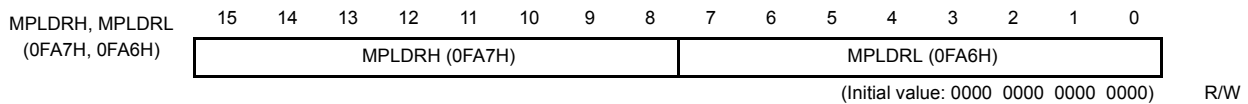
Note 2: CARF, ZERF, SIGN, and OVRF are programmed at the end of calculation. They are not affected by a read from the status register.

Note 3: ZERF and SIGN are not affected by a write to the addend register.

Note 4: In multiply mode, OVRF and CARF are always read as "0".

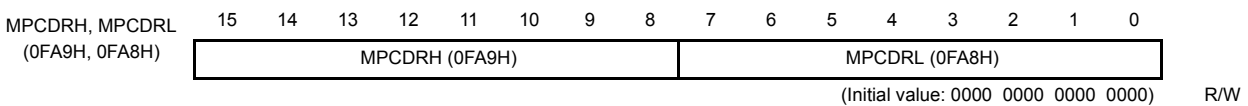
Note 5: Bit 7 to 5 are always read as "1".

## Multiplier data Register



Note: In signed arithmetic mode, bit 15 is treated as the sign bit.

## Multiplicand data Register

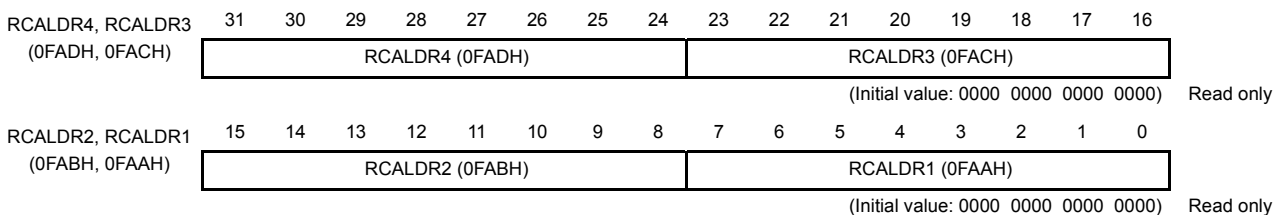


Note 1: In signed arithmetic mode, bit 15 is treated as the sign bit.

Note 2: Calculation can only be started by writing to both the lower byte (MPCDRL) and upper byte (MPCDRH) of the multiplicand register in this order.

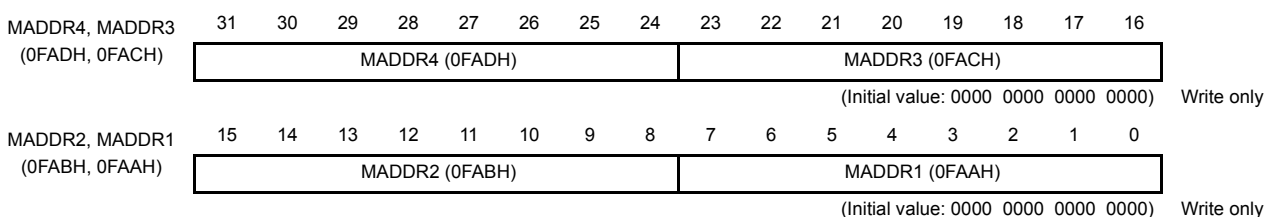
Note 3: The multiplicand register can only be programmed when data is written in the order of lower byte and upper byte. If data is only written to the upper byte, the written data cannot be read out. (If data is only written to the lower byte, the written data can be read out.)

## Result Register



Note: In signed arithmetic mode, bit 31 contains the sign of the calculation result.

## Addend Register



Note 1: In signed arithmetic mode, bit 31 is treated as the sign bit.

Note 2: Writing to the addend register changes the contents of the result register. Thus, read from the result register before writing to the addend register.

## 17.4 Register Description

### 17.4.1 EMAC

Setting MACCR<EMAC> to “1” enables the MAC unit. Once enabled, the MAC unit remains enabled until it is disabled.

### 17.4.2 CMOD

The MACCR<CMOD> is used to specify the arithmetic mode.

Calculation is started automatically when data is written to both the lower byte (MPCDRL) and upper byte (MPCDRH) of the multiplicand register in this order. Thus, the multiplier register (MPLDRH, MPLDRL) must be set before the multiplicand register. When calculation is completed, the result is stored in the result register (RCALDR4 to RCALDR1).

The arithmetic mode is valid until the CMOD field is changed. Note that if the operation mode is changed to IDLE0/1/2, SLOW1/2, or STOP mode, CMOD is initialized.

During calculation, the next data can be written to the multiplier and multiplicand registers only once. Do not write to these registers more than once. Whether or not calculation is in progress can be checked by reading the MACSR<CALC> flag.

Note 1: Before changing the arithmetic mode, ensure that calculation is not being performed (CALC = “0”).

Note 2: Writing to the CMOD field (including an overwrite) makes no changes to the status, multiplier, multiplicand, result, and addend registers. Thus, to clear the status, result, and addend registers after a change of the arithmetic mode, set the RCLR bit to “1”.

### 17.4.3 RCLR

When calculation is not being performed (CALC = “0”), setting MACCR<RCLR> to “1” causes the result, addend, and status registers to be initialized. (The multiplier and multiplicand registers remain the same as before.) RCLR is automatically cleared to “0” one machine cycle after it is set to “1”

Note: When calculation is in progress (CALC = “1”), RCLR cannot be set to “1”. (The instruction to set it to “1” is invalid.)

As shown in Table 17-2, the state of each register changes when: the MAC unit is disabled (EMAC = “0”); the result register is cleared (RCLR = “1”); or the operation mode is changed.

Table 17-2 Effects of the EMAC and RCLR Bits on the MAC Registers

Register	EMAC = “0” (Disable)	RCLR = “1” (register clear)	IDLE0/1/2, SLOW1/2, or STOP Mode
Command register (MACCR)	Bits other than EMAC remain the same as before	Bits other than RCLR remain the same as before. RCLR is cleared to “0” after one machine cycle.	Initialized
Status register (MACSR)	Initialized	Initialized	Initialized
Multiplier data register (MPLDRH, MPLDRL)	Initialized	Remains the same as before	Initialized
Multiplicand data register (MPCDRH, MPCDRL)	Initialized	Remains the same as before	Initialized
Result register (RCALDR4 to RCALDR1)	Initialized	Initialized	Initialized
Addend register (MADDR4 to MADDR1)	Initialized	Initialized	Initialized

Note 1: The multiplier, multiplicand, and addend registers can be written to only when the MAC unit is enabled (EMAC = “1”).

Note 2: When writing to the multiplicand register, be sure to write to the lower byte (MPCDRL) first and then to the upper byte (MPCDRH).

Note 3: RCLR can be written to only when calculation is not being performed (CALC = “0”).



Note 4: When the MAC unit is enabled (EMAC = “1”), if the operation mode is changed to IDLE0/1/2, SLOW1/2, or STOP mode, the command register (MACCR) is initialized and its contents are discarded. Thus, program the MACCR again after each of these operation modes is exited.

## 17.5 Arithmetic Modes

The following four arithmetic modes are available:

1. Unsigned multiply (16 bits × 16 bits)
2. Signed multiply (16 bits × 16 bits)
3. Unsigned multiply-accumulate (16 bits × 16 bits + 32 bits)
4. Signed multiply-accumulate (16 bits × 16 bits + 32 bits)

### 17.5.1 Unsigned Multiply Mode

Setting the MACCR<CMOD> field in the command register to “000B” places the MAC unit in unsigned multiply mode. In this mode, the values of the multiplier and multiplicand registers are each treated as 16-bit data for calculation.

Calculation is started automatically by writing a multiplier to the multiplier register (MPLDRH, MPLDRL) and then writing a multiplicand to the lower byte (MPCDRL) and upper byte (MPCDRH) of the multiplicand register in this order. The calculation result is stored as 32-bit data in the result register (RCALDR4 to RCALDR1). (The previous calculation result is cleared.)

### 17.5.2 Signed Multiply Mode

Setting the MACCR<CMOD> field in the command register to “010B” places the MAC unit in signed multiply mode. In this mode, bit 15 in the multiplier and multiplicand registers is each treated as the sign bit.

Calculation is started automatically by writing a multiplier to the multiplier register (MPLDRH, MPLDRL) and then writing a multiplicand to the lower byte (MPCDRL) and upper byte (MPCDRH) of the multiplicand register in this order. The calculation result is stored as 32-bit data in the result register (RCALDR4 to RCALDR1). (Bit 31 contains the sign, and the previous calculation result is cleared.) The sign of the calculation result varies depending on the signs of the multiplier and multiplicand, as shown in Table 17-3.

Table 17-3 Signs Used in Singed Multiply Mode

Sign of Multiplier	Sign of Multiplicand	Sign of Calculation Result
0	0	0
0	1	1
1	0	1
1	1	0

### 17.5.3 Unsigned Multiply-Accumulate Mode

Setting the MACCR<CMOD> field in the command register to “001B” places the MAC unit in unsigned multiply-accumulate mode. In this mode, the values of the multiplier and multiplicand registers are each treated as 16-bit data for calculation.

Calculation is started automatically by writing a multiplier to the multiplier register (MPLDRH, MPLDRL) and then writing a multiplicand to the lower byte (MPCDRH) and upper byte (MPCDRH) of the multiplicand register in this order. First, the multiplier and multiplicand are multiplied. Then, the contents of the addend register are added to the product. The sum is stored as 32-bit data in the result register.

In unsigned multiply-accumulate mode, any addend can be written to the addend register when calculation is not being performed. If, for example,  $A \times B$  is executed after arbitrary data  $C$  is written to the addend register, the result of  $A \times B + C$  is stored in the result register (RCALDR4 to RCALDR1). Setting the RCLR bit to “1”

causes the result and addend registers to be cleared. After calculation is completed, the contents of the result register are automatically stored in the addend register. Thus, if the contents of the addend register are not changed, the result of the previous multiply-accumulate operation is used as an addend for the next calculation.

Note 1: Be sure to write to the addend register when calculation is not being performed (CALC = "0").

Note 2: Writing to the addend register changes the contents of the result register. Thus, read from the result register before writing to the addend register.

### 17.5.4 Signed Multiply-Accumulate Mode

Setting the MACCR<CMOD> field in the command register to "011B" places the MAC unit in signed multiply-accumulate mode. In this mode, bit 15 in the multiplier and multiplicand register is each treated as the sign bit.

Calculation is started automatically by writing a multiplier to the multiplier register (MPLDRH, MPLDRL) and then writing a multiplicand to the lower byte (MPCDRL) and upper byte (MPCDRH) of the multiplicand register in this order. First, the multiplier and multiplicand are multiplied. Then, the contents of the addend register are added to the product. The sum is stored as signed 32-bit data in the result register (RCALDR4 to RCALDR1). The sign of the result varies as shown in Table 17-4. As in the case of unsigned multiply-accumulate mode, any addend can be written to the addend register when calculation is not being performed.

Note: In signed multiply-accumulate mode, bit 31 in the addend register is treated as the sign bit.

Table 17-4 Signs Used in Signed Multiply-Accumulate Mode

Sign of Product	Sign of Addend	Sign (bit 31) of Calculation Result	
		When OVRF = "0"	When OVRF = "1"
0	0	0	1
0	1	"1" when sum < 0 "0" when sum ≥ 0	–
1	0	"1" when sum < 0 "0" when sum ≥ 0	–
1	1	1	0

### 17.5.5 Valid Numerical Ranges

Table 17-5 shows the numerical range that can be handled in each arithmetic mode.

Table 17-5 Valid Numerical Ranges in Decimal (with hexadecimal shown in brackets)

Mode	Multiplier/Multiplicand	Addend	Sum
Unsigned multiply	0 to 65535 (0000H to FFFFH)	–	0 to 4294836225 (00000000H to FFFE0001H)
Signed multiply	-32768 to 32767 (8000H to 7FFFH)	–	-1073709056 to 1073741824 (C0008000H to 40000000H)
Unsigned multiply-accumulate	0 to 65535 (0000H to FFFFH)	0 to 4294967295 (00000000H to FFFFFFFFH)	0 to 4294967295 (00000000H to FFFFFFFFH)
Signed multiply-accumulate	-32768 to 32767 (8000H to 7FFFH)	-2147483648 to 2147483647 (80000000H to 7FFFFFFFH)	-2147483648 to 2147483647 (80000000H-7FFFFFFFH)

## 17.6 Status Flags

The status register MACSR contains the following five flags. OVRF, CARF, SIGN, and ZERF are programmed when calculation is completed, and these flags are not affected by a read from the status register.

1. Operation status flag (CALC)
2. Overflow flag (OVRF)
3. Carry flag (CARF)
4. Sign flag (SIGN)
5. Zero flag (ZERF)

### 17.6.1 Operation Status Flag (CALC)

CALC indicates the status of the MAC unit. It is set to "1" when calculation is in progress and "0" when calculation is not in progress.

### 17.6.2 Overflow Flag (OVRF)

OVRF is set to "1" if the sum of positive values is negative or the sum of negative values is positive in signed multiply-accumulate mode. In other cases, it is cleared to "0".

Note: In multiply mode, OVRF is always read as "0".

### 17.6.3 Carry Flag (CARF)

CARF is set to "1" if a carry occurs in the highest-order bit (bit 31) in a multiply-accumulate operation. In other cases, it is cleared to "0".

Note: In multiply mode, CARF is always read as "0".

### 17.6.4 Sign Flag (SIGN)

SIGN contains the same data as the highest-order bit (bit 31) of the calculation result (regardless of whether calculation is performed in signed or unsigned mode).

Note: SIGN is programmed by the calculation result. This flag is not affected by a write to the addend register.

### 17.6.5 Zero Flag (ZERF)

ZERF is set to "1" if the result register contains "00000000H". In other cases, it is cleared to "0". It is also set to "1" if the result register contains "00000000H" after an overflow or carry has occurred.

Note: ZERF is programmed by the calculation result. This flag is not affected by a write to the addend register.

## 17.7 Example of Software Processing

The following shows an example of calculating  $X = \alpha x + \beta y + \gamma z$ . The calculation time is 3μs when  $f_c = 8$  MHz. The multiplier and multiplicand are separately stored in data RAM. The W and A registers are used as general-purpose registers. The general-purpose registers are not saved on the stack. The processing for enabling/disabling the MAC unit is not included.

Note 1: If the operation mode is changed by processing an interrupt during calculation, the correct calculation result may not be obtained. Thus, before starting calculation, be sure to execute the DI instruction to disable interrupts.

Note 2: Before reading the result register after calculation is started, check that the CALC flag in the MACSR register is "0" or wait for at least three machine cycles (e.g. NOP x 3).

Instruction	Processing Time
DI	(Disables interrupts.)
LD                   WA, (RAM_Multiplier α)	6 cycles/3 μs
LD                   (MPLDRL), WA                   ; Multiplier register	6 cycles/3 μs
LD                   WA, (RAM_Multiplier x)	6 cycles/3 μs
LD                   (MPCDRL), WA                   ; Multiplicand register	6 cycles/3 μs

; The next data can be written in succession.

LD	WA, (RAM_Multiplier $\beta$ )		6 cycles/3 $\mu$ s
LD	(MPLDRL), WA	; Multiplier register	6 cycles/3 $\mu$ s
LD	WA, (RAM_Multiplicand Z)		6 cycles/3 $\mu$ s
LD	(MPCDRL), WA	; Multiplicand register	6 cycles/3 $\mu$ s
; The first calculation is already completed. Thus, the next data can be written.			
LD	WA, (RAM_Multiplier $\gamma$ )		6 cycles/3 $\mu$ s
LD	(MPLDRL), WA	; Multiplier register	6 cycles/3 $\mu$ s
LD	WA, (RAM_Multiplicand Z)		6 cycles/3 $\mu$ s
LD	(MPCDRL), WA	; Multiplicand register	6 cycles/3 $\mu$ s
NOP		; Wait three machine cycles or longer.	
NOP		; (Note 2)	
NOP			
LD	WA, (RCALDR1)	; Low-order part of the result register	6 cycles/3 $\mu$ s
LD	(RAM_Low-order part of result X), WA		6 cycles/3 $\mu$ s
LD	WA, (RCALDR3)	; High-order part of the result register	6 cycles/3 $\mu$ s
LD	(RAM_High-order part of result X), WA		6 cycles/3 $\mu$ s
EI		(Enables interrupts.)	
Processing time			51 ms
<CALL mn instruction>			6 cycles/3 $\mu$ s
RET			6 cycles/3 $\mu$ s
Total processing time			57 $\mu$ s

## 18. Flash Memory

TMP86FS23UG 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. TMP86FS23UG 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

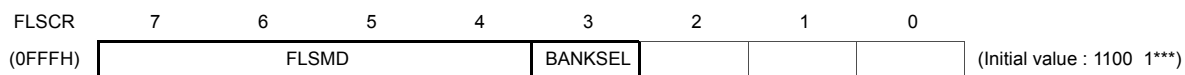
The parallel PROM mode allows the flash memory to be accessed as a stand-alone flash memory by the program writer provided by the third party. High-speed access to the flash memory is available by controlling address and data signals directly. For the support of the program writer, please ask Toshiba 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.

## 18.1 Flash Memory Control

The flash memory is controlled via the flash memory control register (FLSCR) and flash memory standby control register (FLSSTB).

### 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

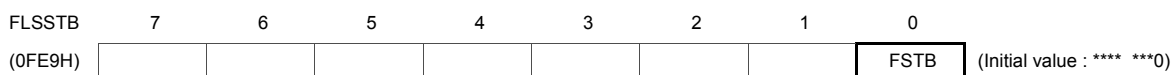
Note 1: The command sequence of the flash memory can be executed only when FLSMD="0011B". In other cases, any attempts to execute the command sequence are ineffective.

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.

### Flash Memory Standby Control Register



FSTB	Flash memory standby control	0: Disable the standby function. 1: Enable the standby function.	Write only
------	------------------------------	---	------------

Note 1: When FSTB is set to 1, do not execute the read/write instruction to the flash memory because there is a possibility that the expected data is not read or the program is not operated correctly. If executing the read/write instruction, FSTB is initialized to 0 automatically.

Note 2: If an interrupt is issued when FSTB is set to 1, FSTB is initialized to 0 automatically and then the vector area of the flash memory is read.

Note 3: If the IDLE0/1/2, SLEEP0/1/2 or STOP mode is activated when FSTB is set to 1, FSTB is initialized to 0 automatically. In the IDLE0/1/2, SLEEP0/1/2 or STOP mode, the standby function operates regardless of FSTB setting.

#### 18.1.1 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.

#### 18.1.2 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 18-1 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	

### 18.1.3 Flash Memory Standby Control (FLSSTB<FSTB>)

Low power consumption is enabled by cutting off the steady-state current of the flash memory. In the IDLE0/1/2, SLEEP0/1/2 or STOP mode, the steady-state current of the flash memory is cut off automatically.

When the program is executed in the RAM area (without accessing the flash memory) in the NORMAL 1/2 or SLOW1/2 mode, the current can be cut off by the control of the register. To cut off the steady-state current of the flash memory, set FLSSTB<FSTB> to "1" by the control program in the RAM area. The procedures for controlling the FLSSTB register are explained below.

(Steps 1 and 2 are controlled by the program in the flash memory, and steps 3 through 8 are controlled by the write control program executed in the RAM area.)

1. Transfer the control program of the FLSSTB register to the RAM area.
2. Jump to the RAM area.
3. Disable (DI) the interrupt master enable flag (IMF = "0").
4. Set FLSSTB<FSTB> to "1".
5. Execute the user program.
6. Repeat step 5 until the return request to the flash memory is detected.
7. Set FLSSTB<FSTB> to "0".
8. Jump to the flash memory area.

Note 1: The standby function is not operated by setting FLSSTB<FSTB> with the program in the flash memory. You must set FLSSTB<FSTB> by the program in the RAM area.

Note 2: To use the standby function by setting FLSSTB<FSTB> to "1" with the program in the RAM area, FLSSTB<FSTB> must be set to "0" by the program in the RAM area before returning the program control to the flash memory. If the program control is returned to the flash memory with FLSSTB<FSTB> set to "1", the program may misoperate and run out of control.

## 18.2 Command Sequence

The command sequence in the MCU and the serial PROM modes consists of six commands (JEDEC compatible), as shown in Table 18-2. 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 18-3.

Table 18-2 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 18-3 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

### 18.2.1 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  $\mu$ s. 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.

### 18.2.2 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).

### 18.2.3 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.

### 18.2.4 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 18-4 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.

### 18.2.5 Product ID Exit

This command is used to exit the Product ID mode.

### 18.2.6 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 18-4.

It takes a maximum of 40  $\mu$ 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).

### 18.3 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".

## 18.4 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.

### 18.4.1 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 "18.4.1.1 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)".

#### 18.4.1.1 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
```

## 18.4.2 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.

### 18.4.2.1 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.

```

# 19. Serial PROM Mode

## 19.1 Outline

The TMP86FS23UG 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 19-1 shows memory address mapping in the serial PROM mode.

Table 19-1 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 19-5".

## 19.2 Memory Mapping

The Figure 19-1 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 19-1 (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.

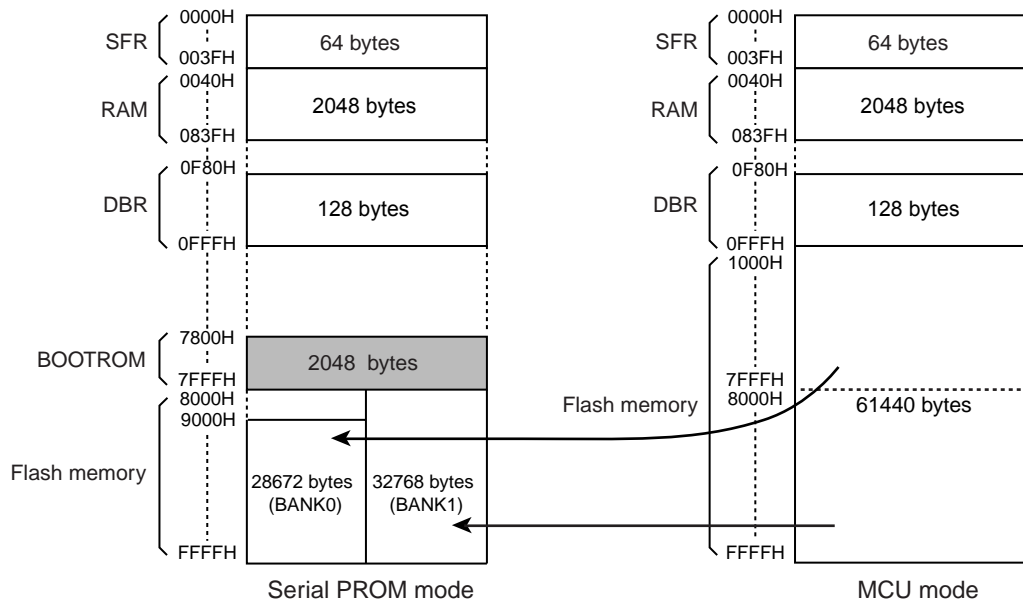



Figure 19-1 Memory Address Maps

## 19.3 Serial PROM Mode Setting

### 19.3.1 Serial PROM Mode Control Pins

To execute on-board programming, activate the serial PROM mode. Table 19-2 shows pin setting to activate the serial PROM mode.

Table 19-2 Serial PROM Mode Setting

Pin	Setting
TEST pin	High
BOOT/RXD pin	High
$\overline{\text{RESET}}$ pin	

Note: The BOOT pin is shared with the UART communication pin (RXD pin) in the serial PROM mode. This pin is used as UART communication pin after activating serial PROM mode

### 19.3.2 Pin Function

In the serial PROM mode, TXD (P11) and RXD (P10) are used as a serial interface pin.

Table 19-3 Pin Function in the Serial PROM Mode

Pin Name (Serial PROM Mode)	Input/ Output	Function	Pin Name (MCU Mode)
TXD	Output	Serial data output	P11
BOOT/RXD	Input/Input	Serial PROM mode control/Serial data input	P10
$\overline{\text{RESET}}$	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	Power supply	0 V	
VAREF	Power supply	Leave open or apply input reference voltage.	
I/O (Output) ports except P11, P10	I/O (Output)	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.	
COM3 to COM0	Output	Low output in the serial PROM mode	
VLC	Power supply	Connect to GND or apply LCD drive voltage.	
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.



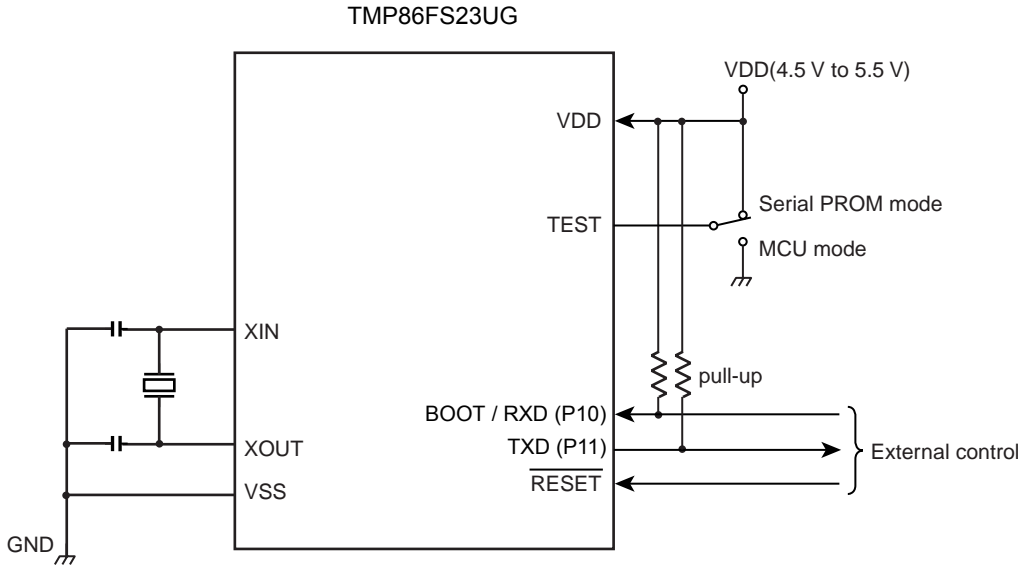


Figure 19-2 Serial PROM Mode Pin Setting

Note 1: For connection of other pins, refer to " Table 19-3 Pin Function in the Serial PROM Mode ".

### 19.3.3 Example Connection for On-Board Writing

Figure 19-3 shows an example connection to perform on-board writing.

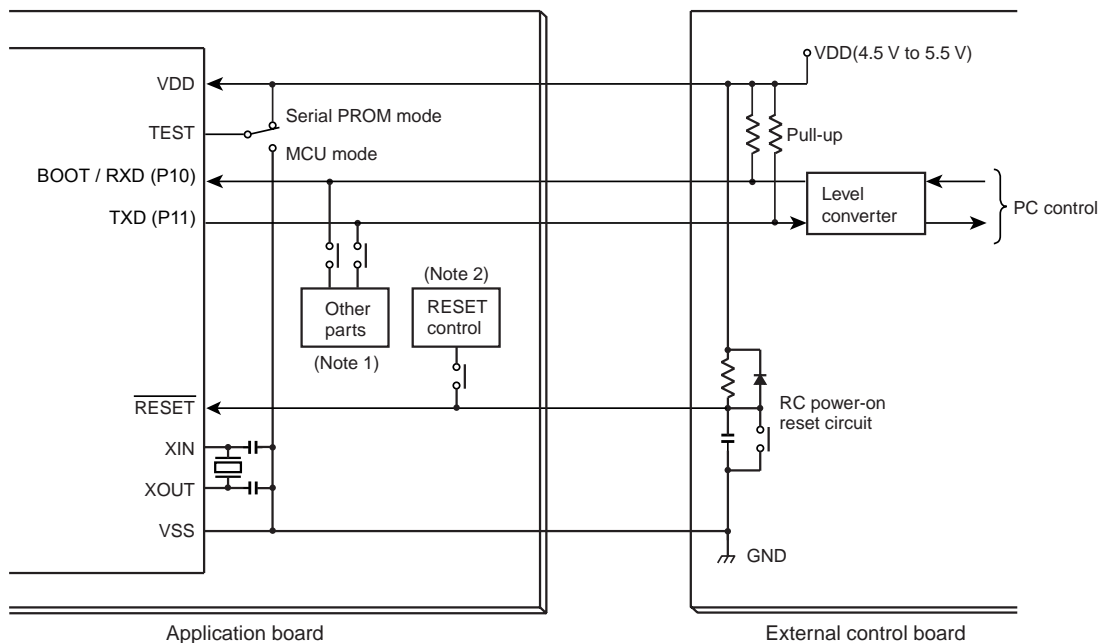


Figure 19-3 Example Connection for On-Board Writing

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 19-3 Pin Function in the Serial PROM Mode ".

### 19.3.4 Activating the Serial PROM Mode

The following is a procedure to activate the serial PROM mode. " Figure 19-4 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/RXD 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/RXD pin after setup sequence. For details of the setup timing, refer to " 19.16 UART Timing ".

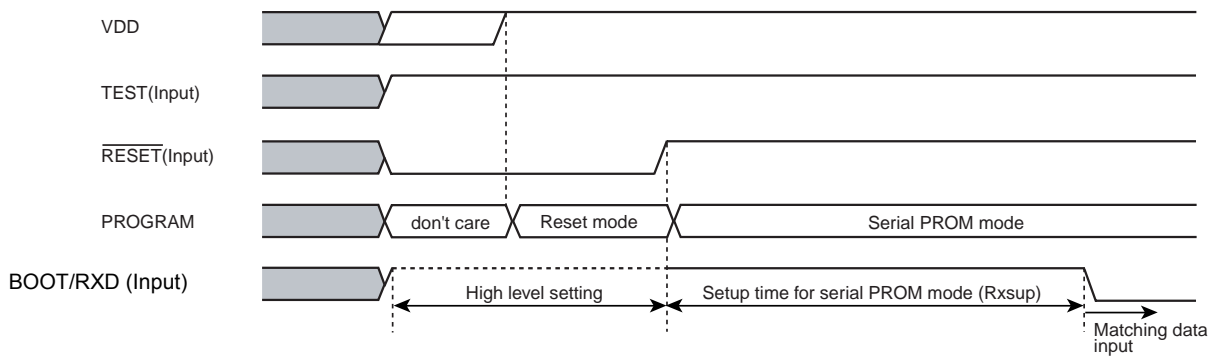


Figure 19-4 Serial PROM Mode Timing

## 19.4 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 TMP86FS23UG. The Table 19-5 shows an operating frequency and baud rate. The frequencies which are not described in Table 19-5 can not be used.

- Baud rate (Default): 9600 bps
- Data length: 8 bits
- Parity addition: None
- Stop bit: 1 bit

Table 19-4 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 19-5 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.

## 19.5 Operation Command

The eight commands shown in Table 19-6 are used in the serial PROM mode. After reset release, the TMP86FS23UG waits for the matching data (5AH).

Table 19-6 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.

## 19.6 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, TMP86FS23UG 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, TMP86FS23UG 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, TMP86FS23UG 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, TMP86FS23UG 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 TMP86FS23UG, 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.

19.6.1 Flash Memory Erasing Mode (Operating command: F0H)

Table 19-7 shows the flash memory erasing mode.

Table 19-7 Flash Memory Erasing Mode

	Transfer Byte	Transfer Data from the External Controller to TMP86FS23UG	Baud Rate	Transfer Data from TMP86FS23UG 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 19-4) -	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 " 19.13 Specifying the Erasure Area ".

Note 3: Refer to " 19.8 Checksum (SUM) ".

Note 4: Refer to " 19.10 Passwords ".

Note 5: Do not transmit the password string for a blank product.

Note 6: When a password error occurs, TMP86FS23UG stops UART communication and enters the halt mode. Therefore, when a password error occurs, initialize TMP86FS23UG by the RESET pin and reactivate the serial PROM mode.

Note 7: If an error occurs during transfer of a password address or a password string, TMP86FS23UG stops UART communication and enters the halt condition. Therefore, when a password error occurs, initialize TMP86FS23UG by the 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 19-6, 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.



19.6.2 Flash Memory Writing Mode (Operation command: 30H)

Table 19-8 shows flash memory writing mode process.

Table 19-8 Flash Memory Writing Mode Process

	Transfer Byte	Transfer Data from External Controller to TMP86FS23UG	Baud Rate	Transfer Data from TMP86FS23UG 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 19-4) -	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 " 19.7 Error Code ".

Note 2: Refer to " 19.9 Intel Hex Format (Binary) ".

Note 3: Refer to " 19.8 Checksum (SUM) ".

Note 4: Refer to " 19.10 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, TMP86FS23UG stops UART communication and enters the halt condition. Therefore, when a password error occurs, initialize TMP86FS23UG 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, TMP86FS23UG stops UART communication and enters the halt condition. In this case, initialize TMP86FS23UG 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, TMP86FS23UG stops UART communication and enters the halt condition. In this case, initialize TMP86FS23UG 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, TMP86FS23UG (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 19-5.
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 19-4 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 " 19.8 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.

### 19.6.3 RAM Loader Mode (Operation Command: 60H)

Table 19-9 shows RAM loader mode process.

Table 19-9 RAM Loader Mode Process

	Transfer Bytes	Transfer Data from External Controller to TMP86FS23UG	Baud Rate	Transfer Data from TMP86FS23UG 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 19-4) -	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 " 19.7 Error Code ".

Note 2: Refer to " 19.9 Intel Hex Format (Binary) ".

Note 3: Refer to " 19.8 Checksum (SUM) ".

Note 4: Refer to " 19.10 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, TMP86FS23UG stops UART communication and enters the halt condition. Therefore, when a password error occurs, initialize TMP86FS23UG 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, TMP86FS23UG stops UART communication and enters the halt condition. In this case, initialize TMP86FS23UG 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, TMP86FS23UG stops UART communication and enters the halt condition. In this case, initialize TMP86FS23UG 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 " 19.8 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.

### 19.6.4 Flash Memory SUM Output Mode (Operation Command: 90H)

Table 19-10 shows flash memory SUM output mode process.

Table 19-10 Flash Memory SUM Output Process

	Transfer Bytes	Transfer Data from External Controller to TMP86FS23UG	Baud Rate	Transfer Data from TMP86FS23UG 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 19-4) -	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 " 19.7 Error Code ".

Note 2: Refer to " 19.8 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 " 19.8 Checksum (SUM) ".
5. After sending the checksum, the device waits for the next operation command data.

19.6.5 Product ID Code Output Mode (Operation Command: C0H)

Table 19-11 shows product ID code output mode process.

Table 19-11 Product ID Code Output Process

	Transfer Bytes	Transfer Data from External Controller to TMP86FS23UG	Baud Rate	Transfer Data from TMP86FS23UG 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 19-4) -	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 " 19.7 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 19-6, 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 " 19.11 Product ID Code ".

5. After sending the checksum, the device waits for the next operation command data.



## 19.6.6 Flash Memory Status Output Mode (Operation Command: C3H)

Table 19-12 shows Flash memory status output mode process.

Table 19-12 Flash Memory Status Output Mode Process

	Transfer Bytes	Transfer Data from External Controller to TMP86FS23UG	Baud Rate	Transfer Data from TMP86FS23UG 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 19-4) -	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 " 19.7 Error Code ".

Note 2: For the details on status code 1, refer to " 19.12 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 19-6, 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 " 19.12 Flash Memory Status Code ".
5. After sending the status code, the device waits for the next operation command data.

### 19.6.7 Flash Memory Read Protection Setting Mode (Operation Command: FAH)

Table 19-13 shows Flash memory read protection setting mode process.

Table 19-13 Flash Memory Read Protection Setting Mode Process

	Transfer Bytes	Transfer Data from External Controller to TMP86FS23UG	Baud Rate	Transfer Data from TMP86FS23UG 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 19-4) -	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 " 19.7 Error Code ".

Note 2: Refer to " 19.10 Passwords ".

Note 3: If the read protection is enabled for a blank product or a password error occurs for a non-blank product, TMP86FS23UG stops UART communication and enters the halt mode. In this case, initialize TMP86FS23UG by the RESET pin and reactivate the serial PROM mode.

Note 4: If an error occurs during reception of a password address or a password string, TMP86FS23UG stops UART communication and enters the halt mode. In this case, initialize TMP86FS23UG by the 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.

## 19.7 Error Code

When detecting an error, the device transmits the error code to the external controller, as shown in Table 19-14.

Table 19-14 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, TMP86FS23UG does not transmit an error code.

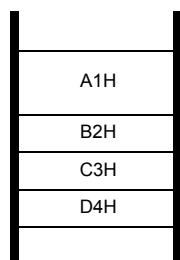
## 19.8 Checksum (SUM)

### 19.8.1 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.

## 19.8.2 Calculation data

The data used to calculate the checksum is listed in Table 19-15.

Table 19-15 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 " 19.11 Product ID Code ".
Flash Memory Status Output mode	9th through 12th bytes of the transferred data	For details, refer to " 19.12 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.

## 19.9 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

## 19.10 Passwords

The consecutive eight or more-byte data in the flash memory area can be specified to the password. TMP86FS23UG 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 19-16 shows the password setting in the blank product and non-blank product.

Table 19-16 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. TMP86FS23UG 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.

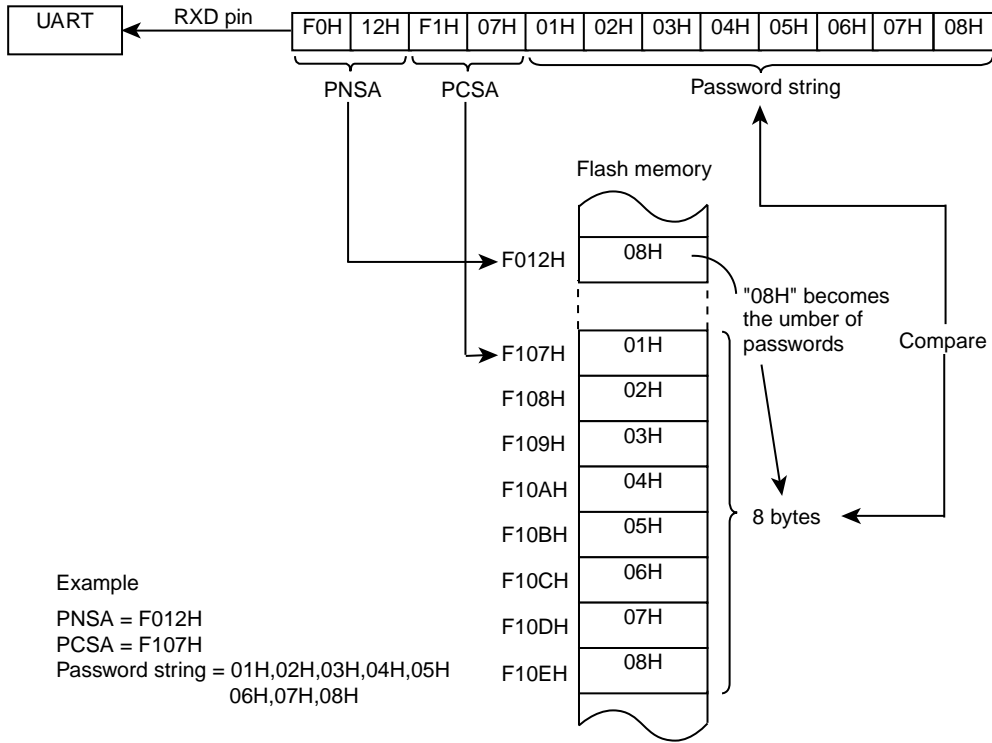


Figure 19-5 Password Comparison

19.10.1 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.

19.10.2 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.

19.10.3 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

## 19.11 Product ID Code

The product ID code is the 13-byte data containing the start address and the end address of ROM. Table 19-17 shows the product ID code format.

Table 19-17 Product ID Code Format

Data	Description	In the Case of TMP86FS23UG
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

## 19.12 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 19-18 shows the flash memory status code.

Table 19-18 Flash Memory Status Code

Data	Description	In the Case of TMP86FS23UG										
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, TMP86FS23UG stops UART communication and enters the halt condition.)

## 19.13 Specifying the Erasure Area

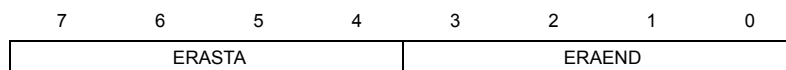
In the flash memory erasing mode, the erasure area of the flash memory is specified by n–2 byte data.

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

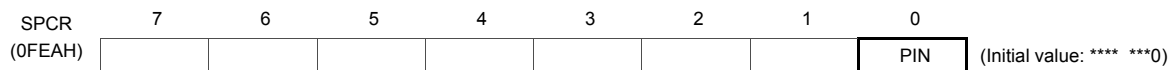
Note: When the sector erase is executed for the area containing no flash cell, TMP86FS23UG stops the UART communication and enters the halt condition.

## 19.14 Port Input Control Register

In the serial PROM mode, the input level is fixed to the all ports except P11 and P10 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

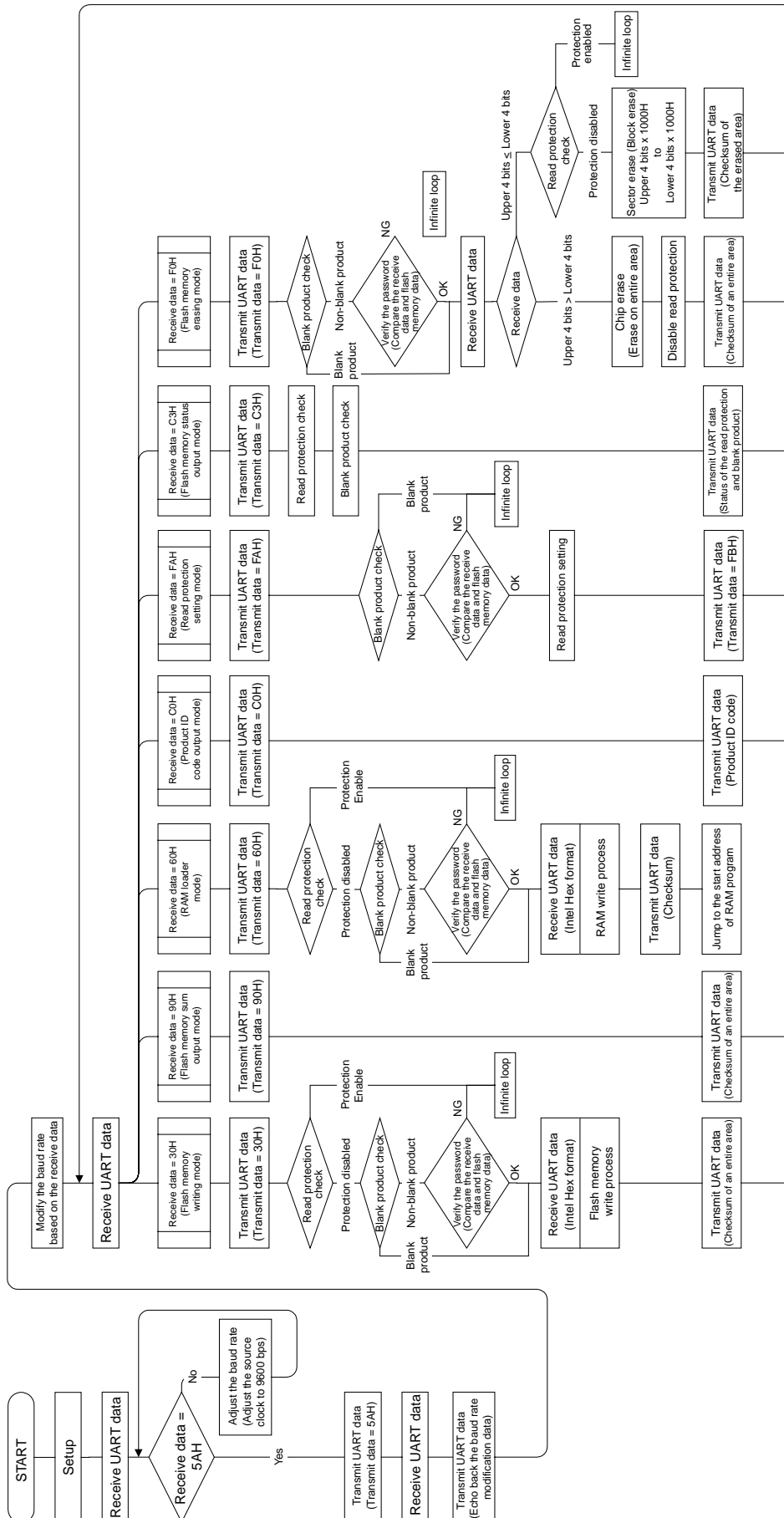


PIN	Port input control in the serial PROM mode	0 : Invalid port inputs (The input level is fixed with a hardware feature.) 1 : Valid port inputs	R/W
-----	--	--	-----

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 P11 and P10 ports are controlled by the SPCR register.

19.15Flowchart



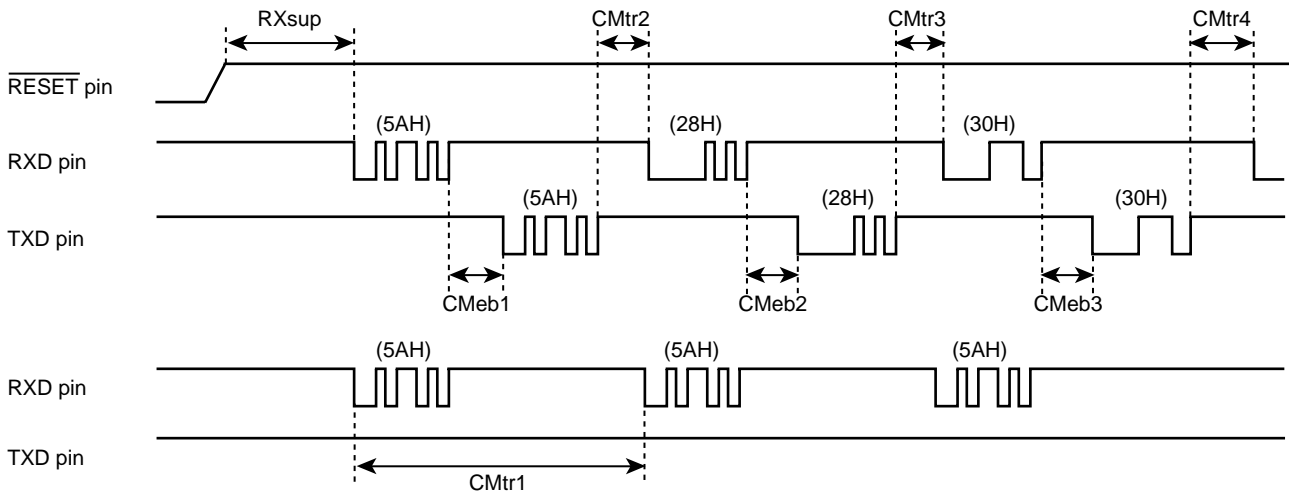
## 19.16UART Timing

Table 19-19 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 19-20 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





## 20. Input/Output Circuitry

### 20.1 Control Pins

The input/output circuitries of the TMP86FS23UG 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.) $R_O = 0.5 \text{ k}\Omega$ (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.)
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 TMP86FS23 does not have a pull-down resistor. Fix the TEST pin at low-level in MCU mode.

## 20.2 Input/Output Ports

Port	I/O	Input/Output Circuitry	Remarks
P1	I/O	<p>Initial "High-Z" SEG output</p>	<p>Tri-state I/O Hysteresis input R = 100 Ω (typ.) LCD segment output</p>
P5 P7 P8	I/O	<p>Initial "High-Z" SEG output</p>	<p>Tri-state I/O R = 100 Ω (typ.) LCD segment output</p>
P2	I/O	<p>Initial "High-Z"</p>	<p>Sink open drain output Hysteresis input R = 100 Ω (typ.)</p>
P34 to P30	I/O	<p>Initial "High-Z"</p>	<p>Sink open drain output or C-MOS output Hysteresis input High current output (Nch) (Only P33, P34 port) R = 100 Ω (typ.)</p>
P37 to P35	Output	<p>Initial "High-Z"</p>	<p>Sink open drain output High current output (Nch)</p>
P6	I/O	<p>Initial "High-Z"</p>	<p>Tri-state I/O Hysteresis input AIN input R = 100 Ω (typ.)</p>



## 21. Electrical Characteristics

### 21.1 Absolute Maximum Ratings

The absolute maximum ratings are rated values which must not be exceeded during operation, even for an instant. Any one of the ratings must not be exceeded. If any absolute maximum rating is exceeded, a device may break down or its performance may be degraded, causing it to catch fire or explode resulting in injury to the user. Thus, when designing products which include this device, ensure that no absolute maximum rating value will ever be exceeded.

( $V_{SS} = 0\text{ V}$ )

Parameter	Symbol	Pins	Ratings	Unit
Supply voltage	$V_{DD}$		-0.3 to 6.5	V
Input voltage	$V_{IN}$		-0.3 to $V_{DD} + 0.3$	
Output voltage	$V_{OUT}$		-0.3 to $V_{DD} + 0.3$	
Output current (Per 1 pin)	$I_{OUT1}$	P1, P30 to P34, P5, P6, P7, P8 port	-1.8	mA
	$I_{OUT2}$	P1, P2, P30 to P32, P5, P6, P7, P8 port	3.2	
	$I_{OUT3}$	P33 to P37 port	30	
Output current (Total)	$\Sigma I_{OUT1}$	P1, P30 to P34, P5, P6, P7, P8 port	-30	
	$\Sigma I_{OUT2}$	P1, P2, P30 to P32, P5, P6, P7, P8 port	60	
	$\Sigma I_{OUT3}$	P33 to P37 port	80	
Power dissipation [ $T_{opr} = 85^{\circ}\text{C}$ ]	$P_D$		350	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	

## 21.2 Recommended Operating Condition

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.

### 21.2.1 When Programming Flash memory in MCU mode

( $V_{SS} = 0\text{ V}$ ,  $T_{opr} = -10\text{ to }40^{\circ}\text{C}$ )

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	$V_{DD} \geq 4.5\text{ V}$	$V_{DD} \times 0.70$	$V_{DD}$	
	$V_{IH2}$	Hysteresis input		$V_{DD} \times 0.75$		
Input low level	$V_{IL1}$	Except hysteresis input	$V_{DD} \geq 4.5\text{ V}$	0	$V_{DD} \times 0.30$	
	$V_{IL2}$	Hysteresis input			$V_{DD} \times 0.25$	
Clock frequency	fc	XIN, XOUT		1.0	16.0	MHz

### 21.2.2 When Not Programming Flash Memory in MCU Mode

( $V_{SS} = 0\text{ V}$ ,  $T_{opr} = -40\text{ to }85^{\circ}\text{C}$ )

Parameter	Symbol	Pins	Ratings	Min	Max	Unit
Supply voltage	$V_{DD}$		fc = 16 MHz	NORMAL1, 2 modes IDLE0, 1, 2 modes	2.7 (Note1)	5.5
			fc = 8 MHz	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	$V_{DD} \geq 4.5\text{ V}$	$V_{DD} \times 0.70$	$V_{DD}$	V
	$V_{IH2}$	Hysteresis input		$V_{DD} \times 0.75$		
	$V_{IH3}$		$V_{DD} < 4.5\text{ V}$	$V_{DD} \times 0.90$		
Input low level	$V_{IL1}$	Except hysteresis input	$V_{DD} \geq 4.5\text{ V}$	0	$V_{DD} \times 0.30$	
	$V_{IL2}$	Hysteresis input			$V_{DD} \times 0.25$	
	$V_{IL3}$		$V_{DD} < 4.5\text{ V}$		$V_{DD} \times 0.10$	
Clock frequency	fc	XIN, XOUT	$V_{DD} = 2.7\text{ to }5.5\text{ V}$	1.0	8.0	MHz
			$V_{DD} = 3.5\text{ to }5.5\text{ V}$		16.0	
	fs	XTIN, XTOUT	$V_{DD} = 2.7\text{ to }5.5\text{ V}$	30.0	34.0	kHz

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 to 85 °C.

## 21.2.3 Serial PROM mode

(V<sub>SS</sub> = 0 V, T<sub>opr</sub> = -10 to 40 °C)

Parameter	Symbol	Pins	Condition	Min	Max	Unit
Supply voltage	V <sub>DD</sub>		NORMAL 1, 2 modes	4.5	5.5	V
Input high voltage	V <sub>IH1</sub>	Except hysteresis input	V <sub>DD</sub> ≥ 4.5 V	V <sub>DD</sub> × 0.70	V <sub>DD</sub>	
	V <sub>IH2</sub>	Hysteresis input		V <sub>DD</sub> × 0.75		
Input low voltage	V <sub>IL1</sub>	Except hysteresis input	V <sub>DD</sub> ≥ 4.5 V	0	V <sub>DD</sub> × 0.30	
	V <sub>IL2</sub>	Hysteresis input		V <sub>DD</sub> × 0.25		
Clock frequency	f <sub>c</sub>	XIN, XOUT		2.0	16.0	MHz

## 21.3 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	$V_{DD} = 5.5\text{ V}$ , $V_{IN} = 5.5\text{ V}/0\text{ V}$	–	–	$\pm 2$	$\mu\text{A}$	
	$I_{IN2}$	Sink open drain, Tri-state						
	$I_{IN3}$	$\overline{\text{RESET}}$ , $\overline{\text{STOP}}$						
Input resistance	$R_{IN2}$	$\overline{\text{RESET}}$ pull-up		100	220	450	$\text{k}\Omega$	
Output leakage current	$I_{LO}$	Sink open drain, Tri-state	$V_{DD} = 5.5\text{ V}$ , $V_{OUT} = 5.5\text{ V}/0\text{ V}$	–	–	$\pm 2$	$\mu\text{A}$	
Output high voltage	$V_{OH}$	C-MOS, 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 and P3 port	$V_{DD} = 4.5\text{ V}$ , $I_{OL} = 1.6\text{ mA}$	–	–	0.4		
Output low current	$I_{OL}$	High current port (P33 to P37 port)	$V_{DD} = 4.5\text{ V}$ , $V_{OL} = 1.0\text{ V}$	–	20	–	mA	
Supply current in NORMAL 1, 2 modes	$I_{DD}$		$V_{DD} = 5.5\text{ V}$ $V_{IN} = 5.3\text{ V}/0.2\text{ V}$ $f_c = 16\text{ MHz}$ $f_s = 32.768\text{ kHz}$	–	12.6	20	mA	
Supply current in IDLE 0, 1, 2 modes								–
Supply current in SLOW 1 mode			$V_{DD} = 3.0\text{ V}$ $V_{IN} = 2.8\text{ V}/0.2\text{ V}$ $f_s = 32.768\text{ kHz}$	When a program operates on flash memory (Note 10, 11)	–	40	260	$\mu\text{A}$
				When a program operates on RAM	–	18	25	
Supply current in SLEEP 1 mode			–	10	18			
Supply current in SLEEP 0 mode			–	8	16			
Supply current in STOP mode	$V_{DD} = 5.5\text{ V}$ $V_{IN} = 5.3\text{ V}/0.2\text{ V}$	–	0.5	10				
Segment output low resistance	$R_{OS1}$	SEG pin		–	20	–	$\text{k}\Omega$	
Common output low resistance	$R_{OC1}$	COM pin			20			
Segment output high resistance	$R_{OS2}$	SEG pin			200			
Common output high resistance	$R_{OC2}$	COM pin			200			
Segment/common output voltage	$V_{O2/3}$	SEG/COM pin	$V_{DD} = 5.0\text{ V}$ $V_{LC} = 2.0\text{ V}$	3.8	–	4.2	V	
	$V_{O1/2}$			3.3		3.7		
	$V_{O1/3}$			2.8		3.2		

Note 1: Typical values show those at  $T_{opr} = 25^{\circ}\text{C}$ ,  $V_{DD} = 5\text{ V}$

Note 2: Input current ( $I_{IN1}$ ,  $I_{IN2}$ ); The current through pull-up or pull-down resistor is not included.

Note 3:  $I_{DD}$  does not include  $I_{REF}$  current.

Note 4: The supply currents of SLOW 2 and SLEEP 2 modes are equivalent to IDLE 0, 1, 2.

Note 5: Output resistors ROS and ROC indicate "ON" when switching levels.

Note 6:  $V_{O2/3}$  indicates the output voltage at the 2/3 level when operating in the 1/4 or 1/3 duty mode.

Note 7:  $V_{O1/2}$  indicates the output voltage at the 1/2 level when operating in the 1/2 duty or static mode.

Note 8:  $V_{O1/3}$  indicates the output voltage at the 1/3 level when operating in the 1/4 or 1/3 duty mode.

Note 9: When using LCD, it is necessary to consider values of  $R_{OS1/2}$  and  $R_{OC1/2}$ .

Note 10: 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 21-1.

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 11: 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.

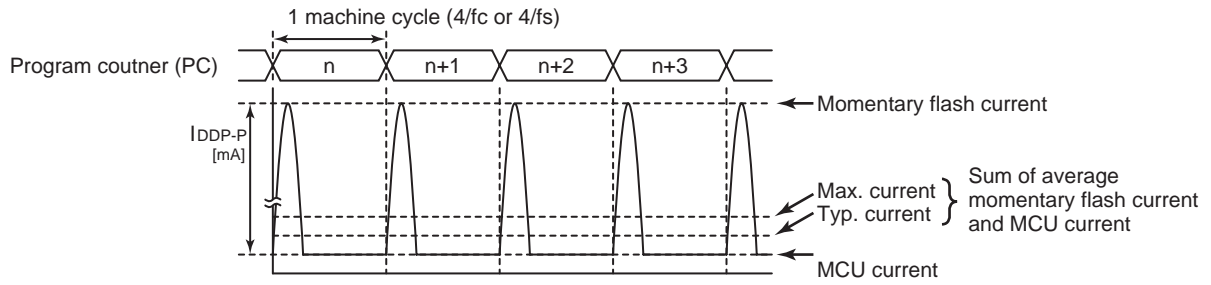


Figure 21-1 Intermittent Operation of Flash Memory

## 21.4 AD Conversion Characteristics

( $V_{SS} = 0.0\text{ V}$ ,  $4.5\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ ,  $T_{opr} = -40\text{ to }85^\circ\text{C}$ )

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 (Note 5)	$A_{VDD}$		$V_{DD}$			
Analog reference voltage range (Note 4)	$\Delta V_{AREF}$		3.5	–	–	
Analog input voltage	$V_{AIN}$		$V_{SS}$	–	$V_{AREF}$	
Power supply current of analog reference voltage	$I_{REF}$	$V_{DD} = A_{VDD} = V_{AREF} = 5.5\text{ V}$ $V_{SS} = 0.0\text{ V}$	–	0.6	1.0	mA
Non linearity error		$V_{DD} = A_{VDD} = 5.0\text{ V}$ $V_{SS} = 0.0\text{ V}$ $V_{AREF} = 5.0\text{ V}$	–	–	$\pm 2$	LSB
Zero point error			–	–	$\pm 2$	
Full scale error			–	–	$\pm 2$	
Total error			–	–	$\pm 2$	

( $V_{SS} = 0.0\text{ V}$ ,  $2.7\text{ V} \leq V_{DD} < 4.5\text{ V}$ ,  $T_{opr} = -40\text{ to }85^\circ\text{C}$ )

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Analog reference voltage	$V_{AREF}$		$A_{VDD} - 1.0$	–	$A_{VDD}$	V
Power supply voltage of analog control circuit (Note 5)	$A_{VDD}$		$V_{DD}$			
Analog reference voltage range (Note 4)	$\Delta V_{AREF}$		2.5	–	–	
Analog input voltage	$V_{AIN}$		$V_{SS}$	–	$V_{AREF}$	
Power supply current of analog reference voltage	$I_{REF}$	$V_{DD} = A_{VDD} = V_{AREF} = 4.5\text{ V}$ $V_{SS} = 0.0\text{ V}$	–	0.5	0.8	mA
Non linearity error		$V_{DD} = A_{VDD} = 2.7\text{ V}$ $V_{SS} = 0.0\text{ V}$ $V_{AREF} = 2.7\text{ V}$	–	–	$\pm 2$	LSB
Zero point error			–	–	$\pm 2$	
Full scale error			–	–	$\pm 2$	
Total error			–	–	$\pm 2$	

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.  
About conversion time, please refer to "Register Framing".

Note 3: Please use input voltage to AIN input Pin in limit of  $V_{AREF}$  to  $V_{SS}$ . When voltage of range outside is input, conversion value becomes unsettled and gives affect to other channel conversion value.

Note 4: Analog reference voltage range:  $\Delta V_{AREF} = V_{AREF} - V_{SS}$

Note 5: The  $A_{VDD}$  pin should be fixed on the  $V_{DD}$  level even though AD converter is not used.

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  $-20\text{ to }85^\circ\text{C}$ .

## 21.5 AC Characteristics

( $V_{SS} = 0\text{ V}$ ,  $V_{DD} = 3.5\text{ to }5.5\text{ V}$ ,  $T_{opr} = -40\text{ to }85^\circ\text{C}$ )

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Machine cycle time	t <sub>cy</sub>	NORMAL1, 2 mode	0.25	–	4	μs
		IDLE1, 2 mode				
		SLOW1, 2 mode	117.6	–	133.3	
		SLEEP1, 2 mode				
High level clock pulse width	t <sub>WCH</sub>	For external clock operation (XIN input)	–	31.25	–	ns
Low level clock pulse width	t <sub>WCL</sub>	fc = 16 MHz				
High level clock pulse width	t <sub>WCH</sub>	For external clock operation (XTIN input)	–	15.26	–	μs
Low level clock pulse width	t <sub>WCL</sub>	fs = 32.768 kHz				

( $V_{SS} = 0\text{ V}$ ,  $V_{DD} = 2.7\text{ to }5.5\text{ V}$ ,  $T_{opr} = -40\text{ to }85^\circ\text{C}$ )

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Machine cycle time	t <sub>cy</sub>	NORMAL1, 2 mode	0.5	–	4	μs
		IDLE1, 2 mode				
		SLOW1, 2 mode	117.6	–	133.3	
		SLEEP1, 2 mode				
High level clock pulse width	t <sub>WCH</sub>	For external clock operation (XIN input)	–	62.5	–	ns
Low level clock pulse width	t <sub>WCL</sub>	fc = 8 MHz				
High level clock pulse width	t <sub>WCH</sub>	For external clock operation (XTIN input)	–	15.26	–	μs
Low level clock pulse width	t <sub>WCL</sub>	fs = 32.768 kHz				

Note: When the supply voltage  $V_{DD}$  is less than 3.0 V, the operating temperature  $T_{opr}$  must be in a range of  $-20\text{ to }85^\circ\text{C}$ .

## 21.6 Timer Counter 1 input (ECIN) Characteristics

( $V_{SS} = 0\text{ V}$ ,  $T_{opr} = -40\text{ to }85^\circ\text{C}$ )

Parameter	Symbol	Condition	Min	Typ.	Max	Unit	
TC1 input (ECIN input)	t <sub>TC1</sub>	Frequency measurement mode $V_{DD} = 3.5\text{ to }5.5\text{ V}$	Single edge count	–	–	16	MHz
			Both edge count	–	–		
		Frequency measurement mode $V_{DD} = 2.7\text{ to }5.5\text{ V}$	Single edge count	–	–	8	
			Both edge count	–	–		

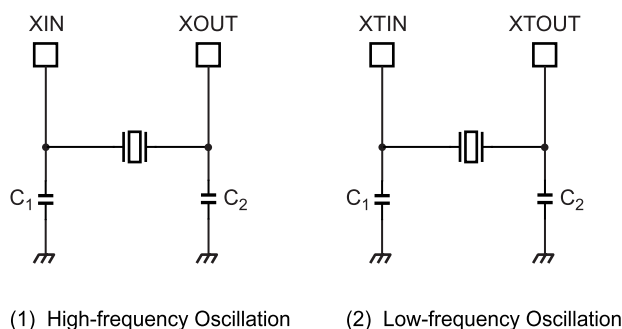
## 21.7 Flash Characteristics

### 21.7.1 Write/Retention Characteristics

( $V_{SS} = 0\text{ V}$ )

Parameter	Condition	Min	Max.	Typ.	Unit
Number of guaranteed writes to flash memory	$V_{SS} = 0\text{ V}$ , $T_{opr} = -10\text{ to }40^\circ\text{C}$	–	–	100	Times

## 21.8 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: For the resonators to be used with Toshiba microcontrollers, we recommend ceramic resonators manufactured by Murata Manufacturing Co., Ltd.

For details, please visit the website of Murata at the following URL:  
<http://www.murata.com>

## 21.9 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  $\geq 95\%$

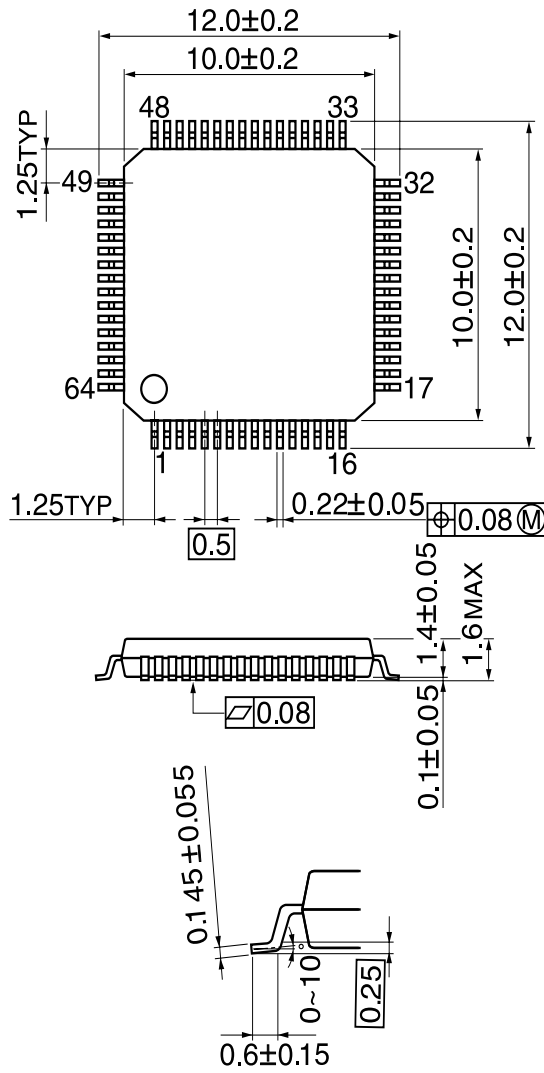
- When using the device (oscillator) in places exposed to high electric fields such as cathode-ray tubes, we recommend electrically shielding the package in order to maintain normal operating condition.



## 22. Package Dimension

P-LQFP64-1010-0.50D

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.

