

**TOSHIBA**

8 Bit Microcontroller  
TLCS-870/C Series

**TMP86CS25AFG**

**TOSHIBA CORPORATION**

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

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

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

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

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

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

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

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

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

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

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

## Revision History

Date	Revision	
2005/12/21	1	First Release
2006/2/13	2	Contents Revised
2006/7/10	3	Periodical updating.No change in contents.
2006/7/27	4	Periodical updating.No change in contents.
2006/9/4	5	Contents Revised
2006/11/13	6	Contents Revised



# Table of Contents

---

---

## TMP86CS25AFG

---

---

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 (MaskROM)	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 (IL15 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 (EF15 to EF4)	36
3.3	Interrupt Source Selector (INTSEL)	39
3.4	Interrupt Sequence	39
3.4.1	Interrupt acceptance processing is packaged as follows	39
3.4.2	Saving/restoring general-purpose registers	40
3.4.2.1	Using PUSH and POP instructions	
3.4.2.2	Using data transfer instructions	
3.4.3	Interrupt return	42
3.5	Software Interrupt (INTSW)	43
3.5.1	Address error detection	43
3.5.2	Debugging	43

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

---

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

---

4.1	SFR . . . . .	47
4.2	DBR . . . . .	49

---

#### 5. I/O Ports

---

5.1	Port P1 (P17 to P10) . . . . .	54
5.2	Port P2 (P22 to P20) . . . . .	55
5.3	Port P3 (P36 to P30) . . . . .	56
5.4	Port P5 (P57 to P50) . . . . .	57
5.5	Port P6 (P67 to P60) . . . . .	58
5.6	Port P7 (P77 to P70) . . . . .	59
5.7	Multi Function Register . . . . .	60

---

#### 6. Watchdog Timer (WDT)

---

6.1	Watchdog Timer Configuration . . . . .	61
6.2	Watchdog Timer Control . . . . .	62
6.2.1	Malfunction Detection Methods Using the Watchdog Timer . . . . .	62
6.2.2	Watchdog Timer Enable . . . . .	63
6.2.3	Watchdog Timer Disable . . . . .	64
6.2.4	Watchdog Timer Interrupt (INTWDT) . . . . .	64
6.2.5	Watchdog Timer Reset . . . . .	65
6.3	Address Trap . . . . .	66
6.3.1	Selection of Address Trap in Internal RAM (ATAS) . . . . .	66
6.3.2	Selection of Operation at Address Trap (ATOUT) . . . . .	66
6.3.3	Address Trap Interrupt (INTATRAP) . . . . .	66
6.3.4	Address Trap Reset . . . . .	67

---

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

---

7.1	Time Base Timer . . . . .	69
7.1.1	Configuration . . . . .	69
7.1.2	Control . . . . .	69
7.1.3	Function . . . . .	70
7.2	Divider Output (DVO) . . . . .	71
7.2.1	Configuration . . . . .	71
7.2.2	Control . . . . .	71

---

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

---

8.1	Configuration . . . . .	73
8.2	Control . . . . .	74
8.3	Function . . . . .	77

8.3.1	Timer mode.....	77
8.3.2	Event Counter mode.....	78
8.3.3	Pulse Width Measurement mode.....	79
8.3.4	Frequency Measurement mode.....	80

---

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

---

9.1	Configuration .....	83
9.2	TimerCounter Control .....	84
9.3	Function .....	89
9.3.1	8-Bit Timer Mode (TC3 and 4).....	89
9.3.2	8-Bit Event Counter Mode (TC3, 4).....	90
9.3.3	8-Bit Programmable Divider Output (PDO) Mode (TC3, 4).....	90
9.3.4	8-Bit Pulse Width Modulation (PWM) Output Mode (TC3, 4).....	93
9.3.5	16-Bit Timer Mode (TC3 and 4).....	95
9.3.6	16-Bit Event Counter Mode (TC3 and 4).....	96
9.3.7	16-Bit Pulse Width Modulation (PWM) Output Mode (TC3 and 4).....	96
9.3.8	16-Bit Programmable Pulse Generate (PPG) Output Mode (TC3 and 4).....	99
9.3.9	Warm-Up Counter Mode.....	101
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 .....	103
10.2	TimerCounter Control .....	104
10.3	Function .....	108
10.3.1	8-Bit Timer Mode (TC5 and 6).....	108
10.3.2	8-Bit Event Counter Mode (TC6).....	109
10.3.3	8-Bit Programmable Divider Output (PDO) Mode (TC6).....	109
10.3.4	8-Bit Pulse Width Modulation (PWM) Output Mode (TC6).....	112
10.3.5	16-Bit Timer Mode (TC5 and 6).....	114
10.3.6	16-Bit Pulse Width Modulation (PWM) Output Mode (TC5 and 6).....	115
10.3.7	16-Bit Programmable Pulse Generate (PPG) Output Mode (TC5 and 6).....	118
10.3.8	Warm-Up Counter Mode.....	120
10.3.8.1	Low-Frequency Warm-up Counter Mode (NORMAL1 → NORMAL2 → SLOW2 → SLOW1)	
10.3.8.2	High-Frequency Warm-Up Counter Mode (SLOW1 → SLOW2 → NORMAL2 → NORMAL1)	

---

## 11. Asynchronous Serial interface (UART )

---

11.1	Configuration .....	123
11.2	Control .....	124
11.3	Transfer Data Format .....	126
11.4	Transfer Rate.....	127
11.5	Data Sampling Method .....	127
11.6	STOP Bit Length .....	128
11.7	Parity .....	128
11.8	Transmit/Receive Operation .....	128
11.8.1	Data Transmit Operation .....	128
11.8.2	Data Receive Operation .....	128
11.9	Status Flag .....	129
11.9.1	Parity Error.....	129

11.9.2	Framing Error.....	129
11.9.3	Overrun Error.....	129
11.9.4	Receive Data Buffer Full.....	130
11.9.5	Transmit Data Buffer Empty.....	130
11.9.6	Transmit End Flag.....	131

---

## 12. Synchronous Serial Interface (SIO0)

---

12.1	Configuration.....	133
12.2	Control.....	134
12.3	Serial clock.....	135
12.3.1	Clock source.....	135
12.3.1.1	Internal clock	
12.3.1.2	External clock	
12.3.2	Shift edge.....	137
12.3.2.1	Leading edge	
12.3.2.2	Trailing edge	
12.4	Number of bits to transfer.....	137
12.5	Number of words to transfer.....	137
12.6	Transfer Mode.....	138
12.6.1	4-bit and 8-bit transfer modes.....	138
12.6.2	4-bit and 8-bit receive modes.....	140
12.6.3	8-bit transfer / receive mode.....	141

---

## 13. Synchronous Serial Interface (SIO1)

---

13.1	Configuration.....	143
13.2	Control.....	144
13.3	Serial clock.....	145
13.3.1	Clock source.....	145
13.3.1.1	Internal clock	
13.3.1.2	External clock	
13.3.2	Shift edge.....	147
13.3.2.1	Leading edge	
13.3.2.2	Trailing edge	
13.4	Number of bits to transfer.....	147
13.5	Number of words to transfer.....	147
13.6	Transfer Mode.....	148
13.6.1	4-bit and 8-bit transfer modes.....	148
13.6.2	4-bit and 8-bit receive modes.....	150
13.6.3	8-bit transfer / receive mode.....	151

---

## 14. 8-Bit AD Converter (ADC)

---

14.1	Configuration.....	153
14.2	Control.....	154
14.3	Function.....	156
14.3.1	AD Converter Operation.....	156
14.3.2	AD Converter Operation.....	156
14.3.3	STOP and SLOW Mode during AD Conversion.....	157
14.3.4	Analog Input Voltage and AD Conversion Result.....	158
14.4	Precautions about AD Converter.....	159
14.4.1	Analog input pin voltage range.....	159
14.4.2	Analog input shared pins.....	159
14.4.3	Noise countermeasure.....	159



---

---

## 15. Key-on Wakeup (KWU)

---

15.1	Configuration .....	161
15.2	Control .....	161
15.3	Function .....	161

---

---

## 16. LCD Driver

---

16.1	Configuration of LCD Driver .....	163
16.2	Controlling LCD Driver .....	164
16.2.1	Frame frequency .....	165
16.3	LCD Booster Circuit .....	165
16.4	Methods of Connecting LCD Booster Circuit .....	166
16.4.1	Method of connecting booster circuit by using a regulator .....	166
16.4.2	Method of connecting booster circuit without using a regulator .....	166
16.5	LCD Display Operation .....	167
16.5.1	Setting display data .....	167
16.5.2	Blanking .....	168
16.6	Method of Controlling LCD Driver .....	168
16.6.1	Initial setting .....	168
16.6.2	Storing display data .....	168

---

---

## 17. Input/Output Circuitry

---

17.1	Control Pins .....	173
17.2	Input/Output Ports .....	174

---

---

## 18. Electrical Characteristics

---

18.1	Absolute Maximum Ratings .....	175
18.2	Recommended Operating Condition .....	176
18.3	DC Characteristics .....	177
18.4	AD Conversion Characteristics .....	178
18.5	AC Characteristics .....	179
18.6	Timer Counter 1 input (ECIN) Characteristics .....	179
18.7	Recommended Oscillating Conditions .....	180
18.8	Handling Precaution .....	180

---

---

## 19. 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  
**TMP86CS25AFG**

Product No.	ROM (MaskROM)	RAM	Package	OTP MCU	Emulation Chip
TMP86CS25AFG	61440 bytes	2048 bytes	P-QFP100-1420-0.65A	TMP86PS25FG	TMP86C925XB

**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 (42 pins)
  - Large current output: 4pins (Typ. 20mA), LED direct drive
4. Watchdog Timer
5. Prescaler
  - Time base timer
  - Divider output function
6. 18-bit Timer/Counter : 1ch
  - Timer Mode
  - Event Counter Mode
  - Pulse Width Measurement Mode
  - Frequency Measurement Mode
7. 8-bit timer counter : 4 ch
  - Timer, Event counter, Programmable divider output (PDO),  
Pulse width modulation (PWM) output,

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

## Programmable pulse generation (PPG) modes

8. 8-bit UART/SIO : 1 ch
9. 8-bit SIO: 1 ch
10. 8-bit successive approximation type AD converter (with sample hold)

Analog inputs: 8ch

11. Key-on wakeup : 4 ch
12. LCD driver/controller

Built-in voltage booster for LCD driver With display memory  
LCD direct drive capability (60 seg × 16 com, 60 seg × 8 com, 60 seg × 4 com)  
1/16, 1/8, 1/4 duties or static drive are programmably selectable

13. Clock operation

Single clock mode

Dual clock mode

14. 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 interrupt. (CPU restarts).

IDLE2 mode: CPU stops and peripherals operate using high and low frequency clock. Release by interrupt. (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.

15. Wide operation voltage:

4.5 V to 5.5 V at 16.0MHz /32.768 kHz

2.7 V to 5.5 V at 8.0 MHz /32.768 kHz

1.8 V to 5.5 V at 4.2MHz /32.768 kHz

1.2 Pin Assignment

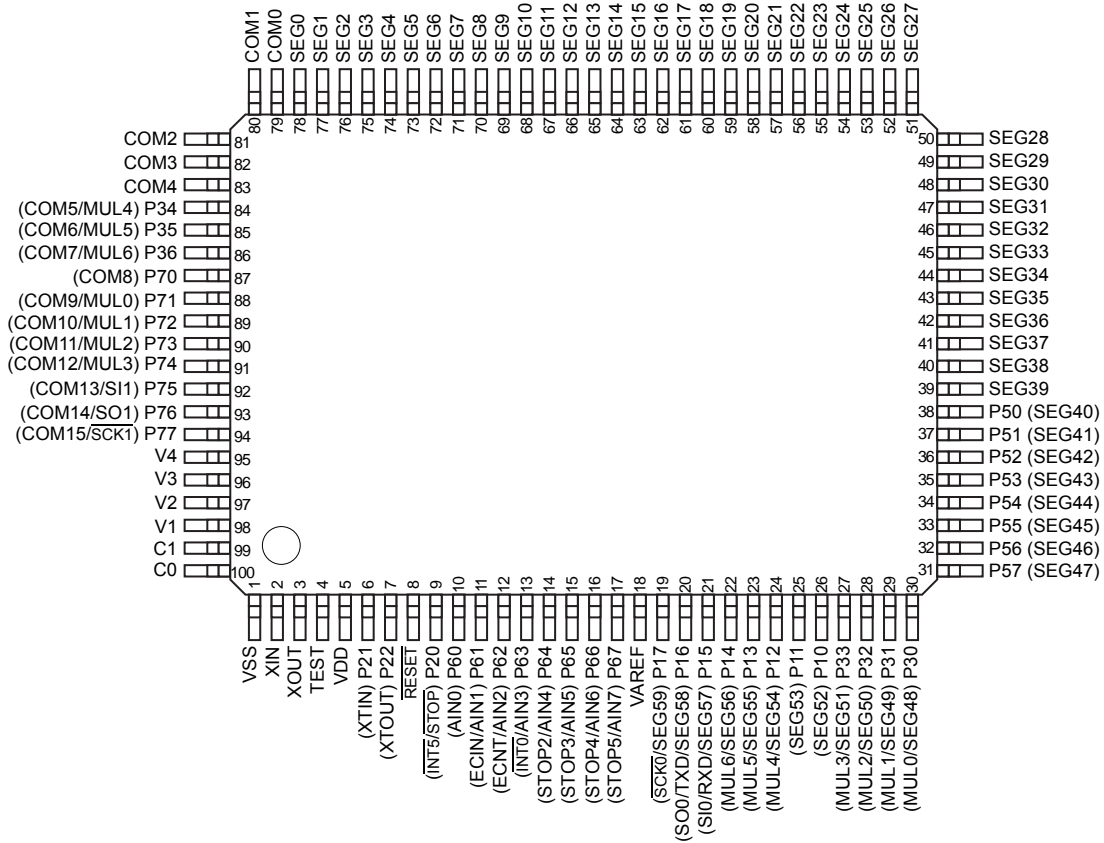


Figure 1-1 Pin Assignment

### 1.3 Block Diagram

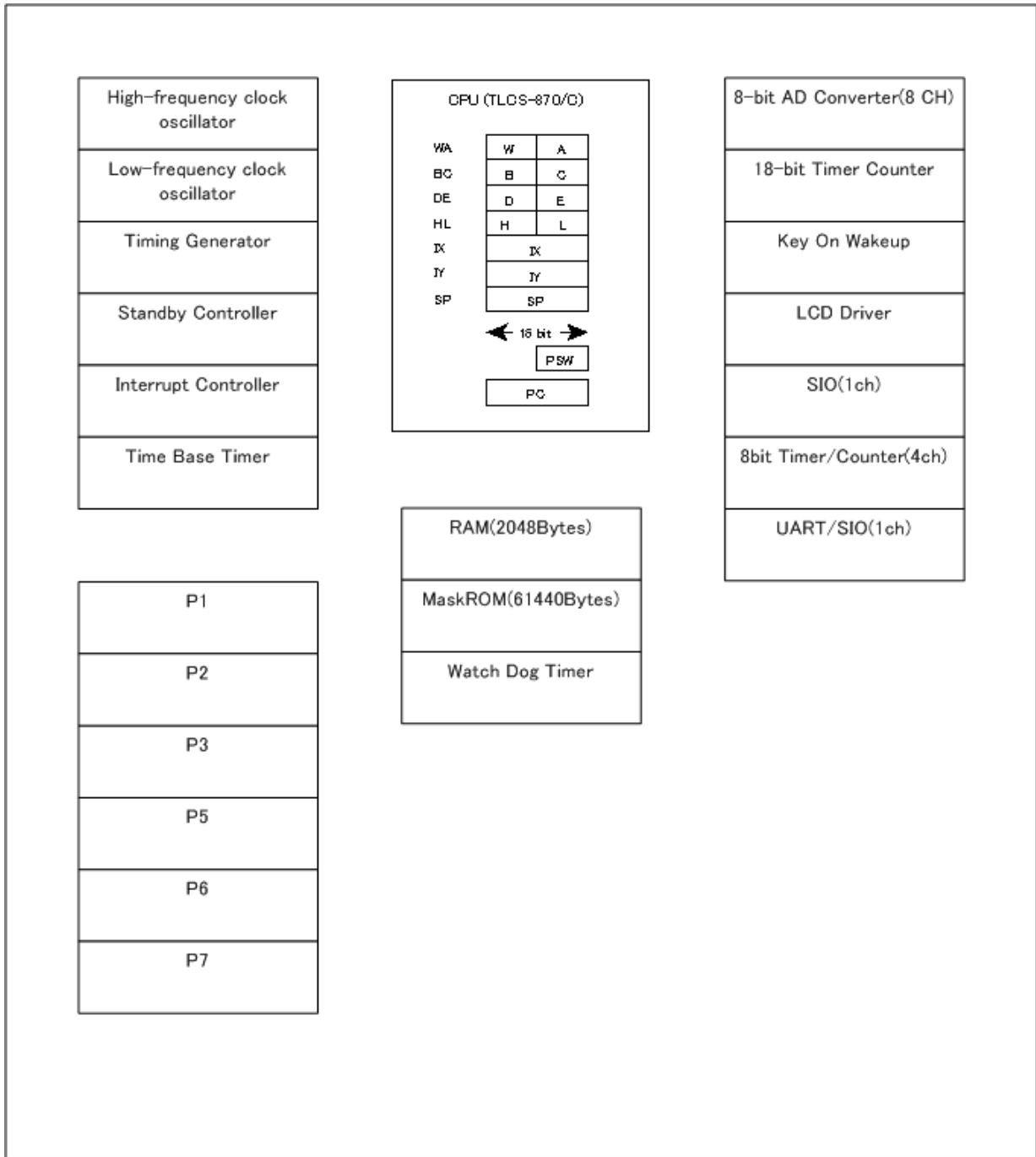


Figure 1-2 Block Diagram

## 1.4 Pin Names and Functions

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

Pin Name	Pin Number	Input/Output	Functions
P17 SEG59 SCK0	19	IO O IO	PORT17 LCD segment output 59 Serial Clock I/O 0
P16 SEG58 TXD SO0	20	IO O O O	PORT16 LCD segment output 58 UART data output Serial Data Output 0
P15 SEG57 RXD SI0	21	IO O I I	PORT15 LCD segment output 57 UART data input Serial Data Input 0
P14 SEG56 MUL6	22	IO O I	PORT14 LCD segment output 56 Multi Function 6 pin
P13 SEG55 MUL5	23	IO O I	PORT13 LCD segment output 55 Multi Function 5 pin
P12 SEG54 MUL4	24	IO O I	PORT12 LCD segment output 54 Multi Function 4 pin
P11 SEG53	25	IO O	PORT11 LCD segment output 53
P10 SEG52	26	IO O	PORT10 LCD segment output 52
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 STOP INT5	9	IO I I	PORT20 STOP mode release signal input External interrupt 5 input
P36 MUL6 COM7	86	IO I O	PORT36 Multi Function 6 pin LCD common output 7
P35 MUL5 COM6	85	IO I O	PORT35 Multi Function 5 pin LCD common output 6
P34 MUL4 COM5	84	IO I O	PORT34 Multi Function 4 pin LCD common output 5
P33 SEG51 MUL3	27	IO O I	PORT33 LCD segment output 51 Multi Function 3 pin
P32 SEG50 MUL2	28	IO O I	PORT32 LCD segment output 50 Multi Function 2 pin

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

Pin Name	Pin Number	Input/Output	Functions
P31 SEG49 MUL1	29	IO O I	PORT31 LCD segment output 49 Multi Function 1 pin
P30 SEG48 MUL0	30	IO O I	PORT30 LCD segment output 48 Multi Function 0 pin
P57 SEG47	31	IO O	PORT57 LCD segment output 47
P56 SEG46	32	IO O	PORT56 LCD segment output 46
P55 SEG45	33	IO O	PORT55 LCD segment output 45
P54 SEG44	34	IO O	PORT54 LCD segment output 44
P53 SEG43	35	IO O	PORT53 LCD segment output 43
P52 SEG42	36	IO O	PORT52 LCD segment output 42
P51 SEG41	37	IO O	PORT51 LCD segment output 41
P50 SEG40	38	IO O	PORT50 LCD segment output 40
P67 AIN7 STOP5	17	IO I I	PORT67 AD converter analog input 7 STOP5 input
P66 AIN6 STOP4	16	IO I I	PORT66 AD converter analog input 6 STOP4 input
P65 AIN5 STOP3	15	IO I I	PORT65 AD converter analog input 5 STOP3 input
P64 AIN4 STOP2	14	IO I I	PORT64 AD converter analog input 4 STOP2 input
P63 AIN3 INT0	13	IO I I	PORT63 AD converter analog input 3 External interrupt 0 input
P62 AIN2 ECNT	12	IO I I	PORT62 AD converter analog input 2 ECNT input
P61 AIN1 ECIN	11	IO I I	PORT61 AD converter analog input 1 ECIN input
P60 AIN0	10	IO I	PORT60 AD converter analog input 0
P77 SCK1 COM15	94	IO IO O	PORT77 Serial Clock I/O 1 LCD common output 15



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

Pin Name	Pin Number	Input/Output	Functions
P76 SO1 COM14	93	IO O O	PORT76 Serial Data Output 1 LCD common output 14
P75 SI1 COM13	92	IO I O	PORT75 Serial Data Input 1 LCD common output 13
P74 MUL3 COM12	91	IO IO O	PORT74 Multi Function 3 pin LCD common output 12
P73 MUL2 COM11	90	IO IO O	PORT73 Multi Function 2 pin LCD common output 11
P72 MUL1 COM10	89	IO IO O	PORT72 Multi Function 1 pin LCD common output 10
P71 MUL0 COM9	88	IO O O	PORT71 Multi Function 0 pin LCD common output 9
P70 COM8	87	IO O	PORT70 LCD common output 8
SEG39	39	O	LCD segment output 39
SEG38	40	O	LCD segment output 38
SEG37	41	O	LCD segment output 37
SEG36	42	O	LCD segment output 36
SEG35	43	O	LCD segment output 35
SEG34	44	O	LCD segment output 34
SEG33	45	O	LCD segment output 33
SEG32	46	O	LCD segment output 32
SEG31	47	O	LCD segment output 31
SEG30	48	O	LCD segment output 30
SEG29	49	O	LCD segment output 29
SEG28	50	O	LCD segment output 28
SEG27	51	O	LCD segment output 27
SEG26	52	O	LCD segment output 26
SEG25	53	O	LCD segment output 25
SEG24	54	O	LCD segment output 24
SEG23	55	O	LCD segment output 23
SEG22	56	O	LCD segment output 22
SEG21	57	O	LCD segment output 21
SEG20	58	O	LCD segment output 20
SEG19	59	O	LCD segment output 19
SEG18	60	O	LCD segment output 18

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

Pin Name	Pin Number	Input/Output	Functions
SEG17	61	O	LCD segment output 17
SEG16	62	O	LCD segment output 16
SEG15	63	O	LCD segment output 15
SEG14	64	O	LCD segment output 14
SEG13	65	O	LCD segment output 13
SEG12	66	O	LCD segment output 12
SEG11	67	O	LCD segment output 11
SEG10	68	O	LCD segment output 10
SEG9	69	O	LCD segment output 9
SEG8	70	O	LCD segment output 8
SEG7	71	O	LCD segment output 7
SEG6	72	O	LCD segment output 6
SEG5	73	O	LCD segment output 5
SEG4	74	O	LCD segment output 4
SEG3	75	O	LCD segment output 3
SEG2	76	O	LCD segment output 2
SEG1	77	O	LCD segment output 1
SEG0	78	O	LCD segment output 0
COM4	83	O	LCD common output 4
COM3	82	O	LCD common output 3
COM2	81	O	LCD common output 2
COM1	80	O	LCD common output 1
COM0	79	O	LCD common output 0
V4	95	I	LCD voltage booster pin
V3	96	I	LCD voltage booster pin
V2	97	I	LCD voltage booster pin
V1	98	I	LCD voltage booster pin
C1	99	I	LCD voltage booster pin
C0	100	I	LCD voltage booster pin
XIN	2	I	Resonator connecting pins for high-frequency clock
XOUT	3	O	Resonator connecting pins for high-frequency clock
$\overline{\text{RESET}}$	8	I	Reset signal
TEST	4	I	Test pin for out-going test. Normally, be fixed to low.
VAREF	18	I	Analog reference voltage input (High)
VDD	5	I	Power Supply
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 TMP86CS25AFG memory is composed MaskROM, RAM, DBR(Data buffer register) and SFR(Special function register). They are all mapped in 64-Kbyte address space. Figure 2-1 shows the TMP86CS25AFG memory address map.

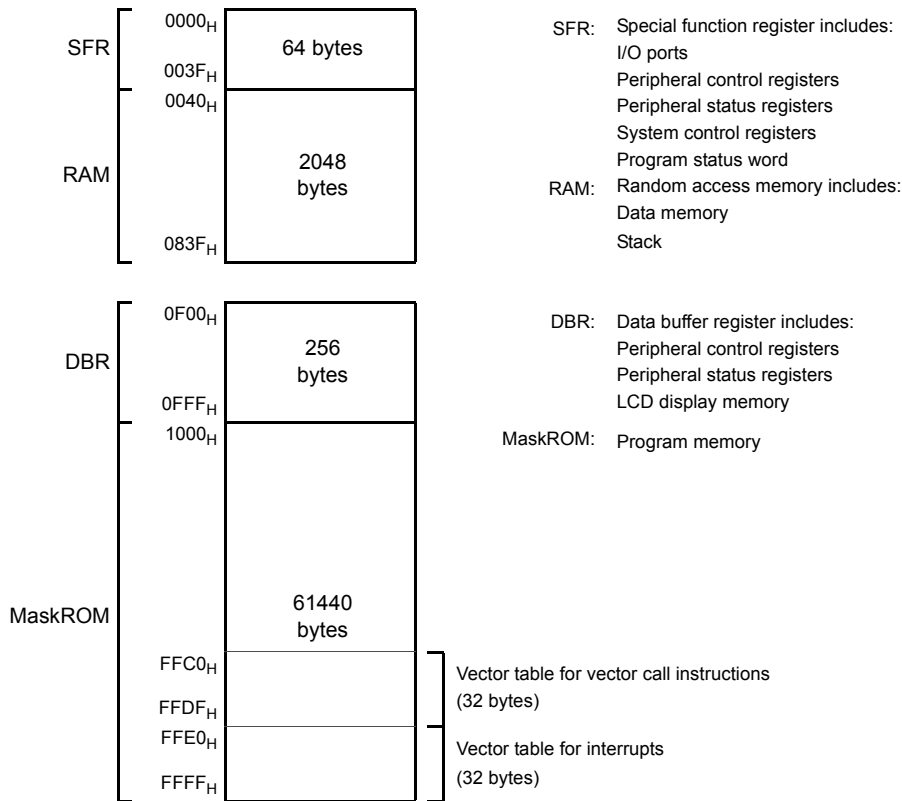


Figure 2-1 Memory Address Map

#### 2.1.2 Program Memory (MaskROM)

The TMP86CS25AFG has a 61440 bytes (Address 1000H to FFFFH) of program memory (MaskROM).

#### 2.1.3 Data Memory (RAM)

The TMP86CS25AFG 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”. (TMP86CS25AFG)

```

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.

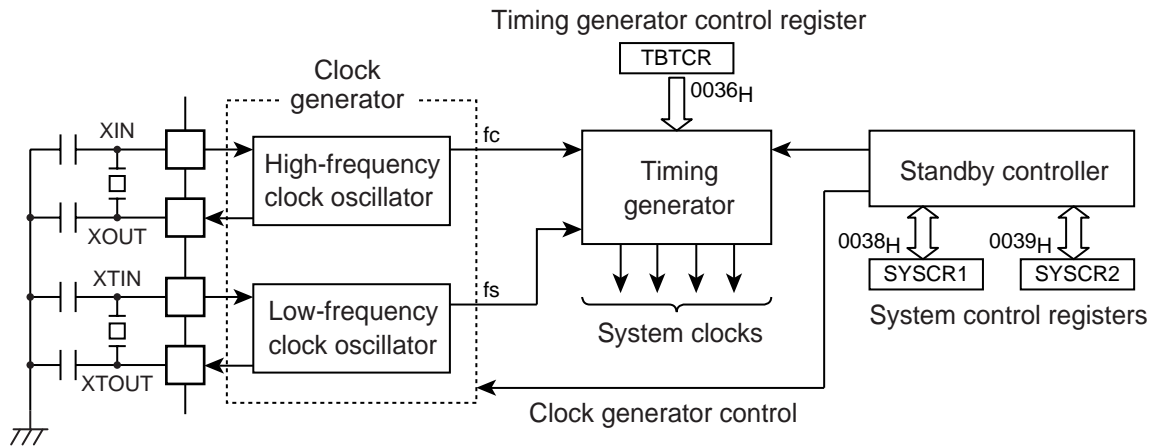


Figure 2-2 System Colck Control

### 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.

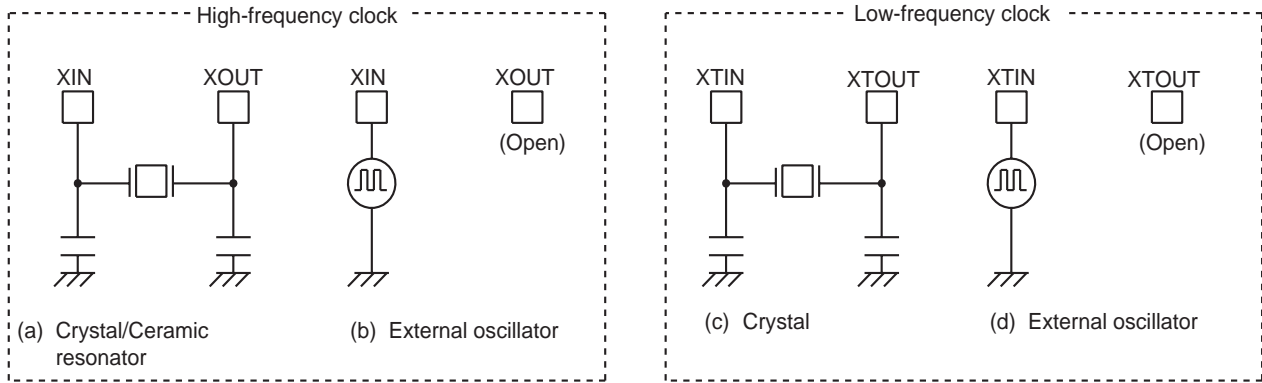


Figure 2-3 Examples of Resonator Connection

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”.

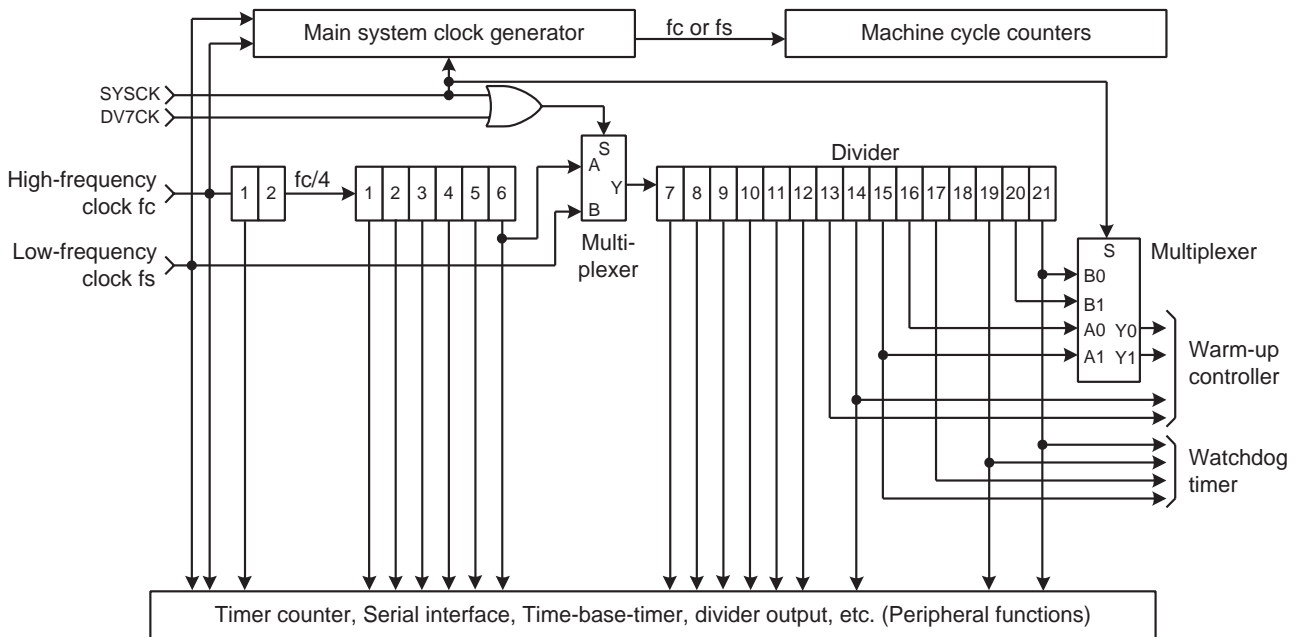
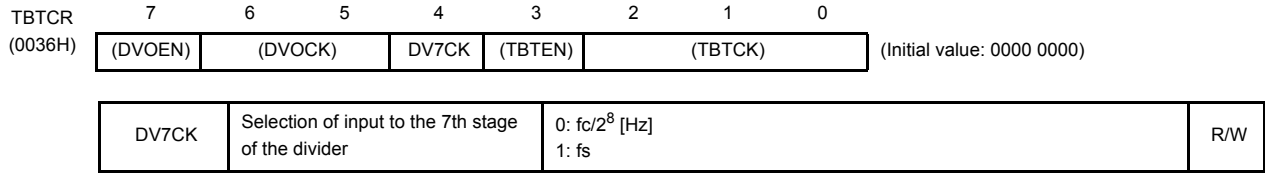


Figure 2-4 Configuration of Timing Generator

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 a "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.

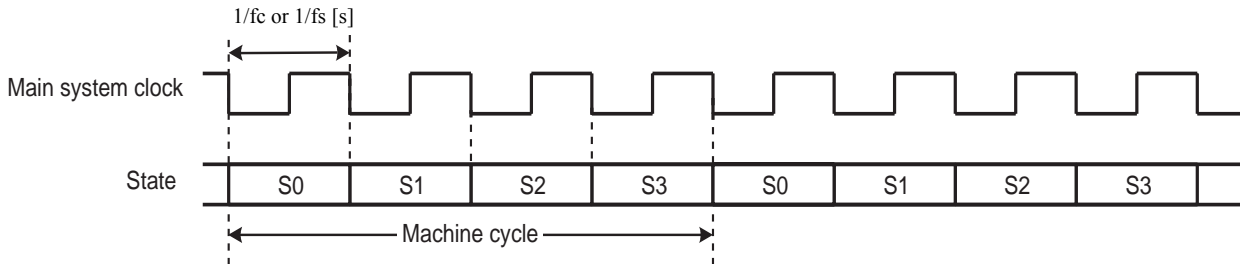


Figure 2-5 Machine Cycle

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 TMP86CS25AFG 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>.

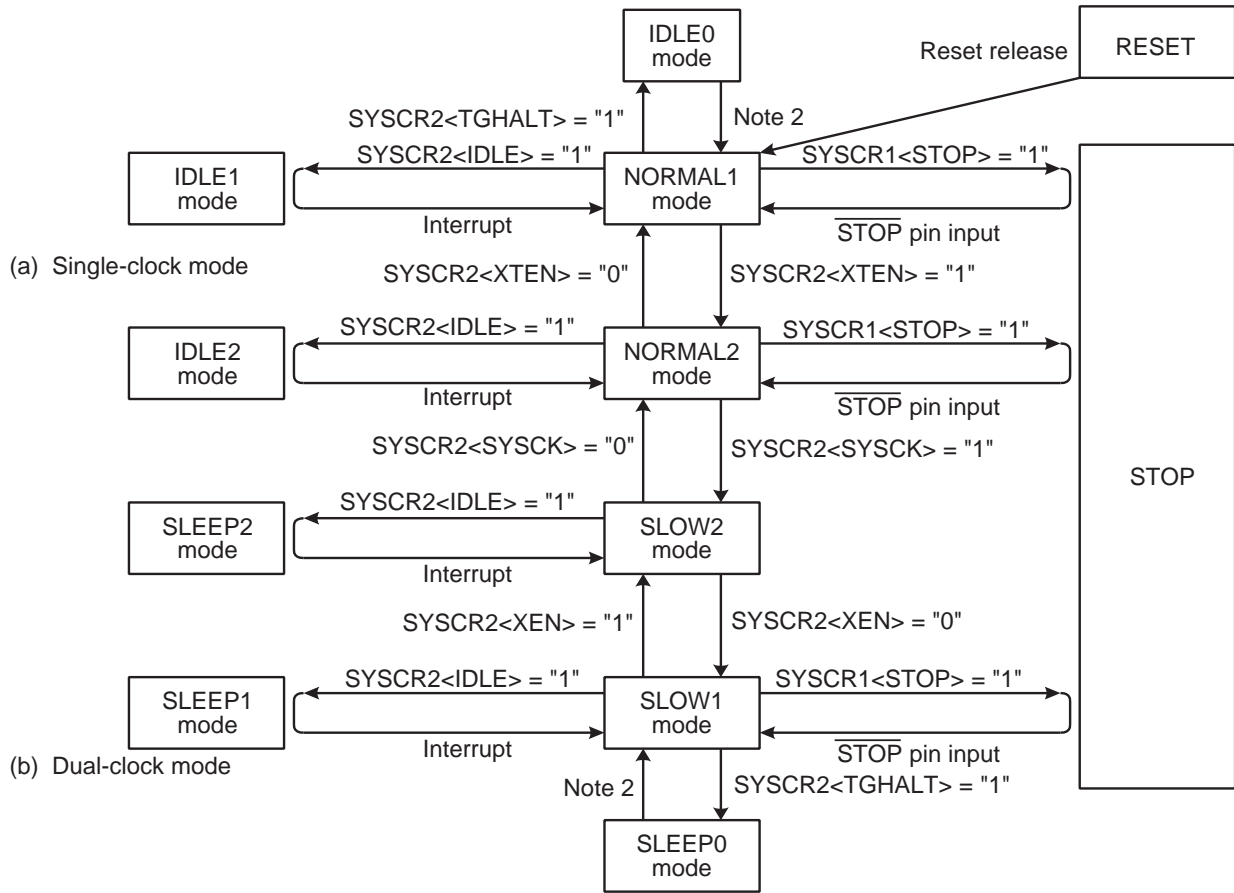
When SLEEP0 mode starts, the CPU stops and the timing generator stops feeding the clock to the peripheral circuits other than TBT. Then, upon detecting the falling edge of the source clock selected with TBTCR<TBTCK>, the timing generator starts feeding the clock to all peripheral circuits.

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			Operate	Operate	Operate			
	IDLE0			Operate	Operate	Halt			
	STOP	Stop	Halt	Halt	–				
Dual clock	NORMAL2	Oscillation	Oscillation	Operate with high frequency	Operate	Operate	4/fc [s]		
	IDLE2			Halt					
	SLOW2			Operate with low frequency					
	SLEEP2			Halt					
	SLOW1	Stop	Stop	Operate with low frequency			Operate	Operate	4/fs [s]
	SLEEP1			Halt					
	SLEEP0			Halt					
	STOP			Stop					

**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  $\overline{RESET}$  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  $\overline{STOP}$  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  $TBTCR < TBTCCK >$ .
- 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
    
```

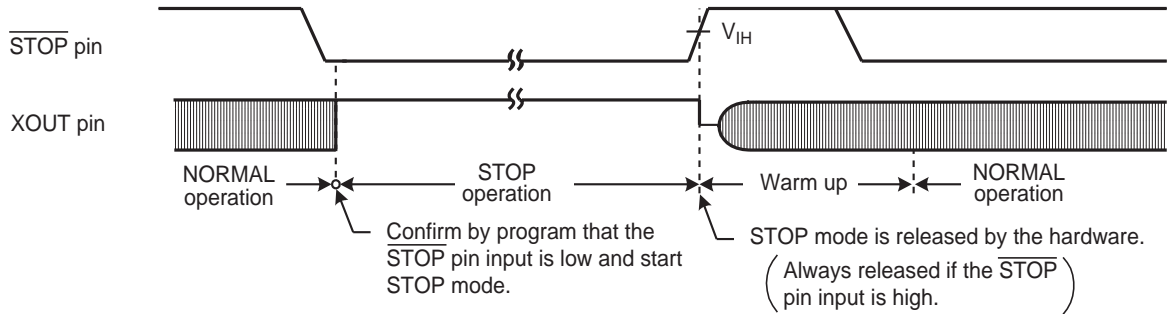


Figure 2-7 Level-sensitive Release Mode

- 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
    
```

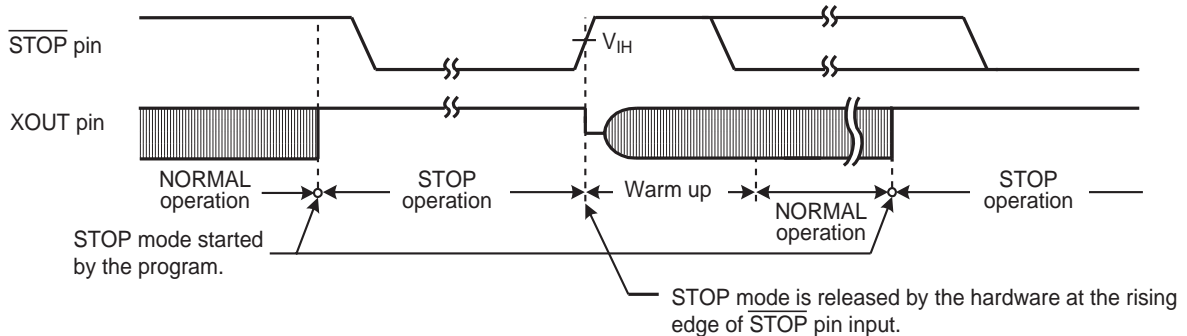


Figure 2-8 Edge-sensitive Release Mode

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.

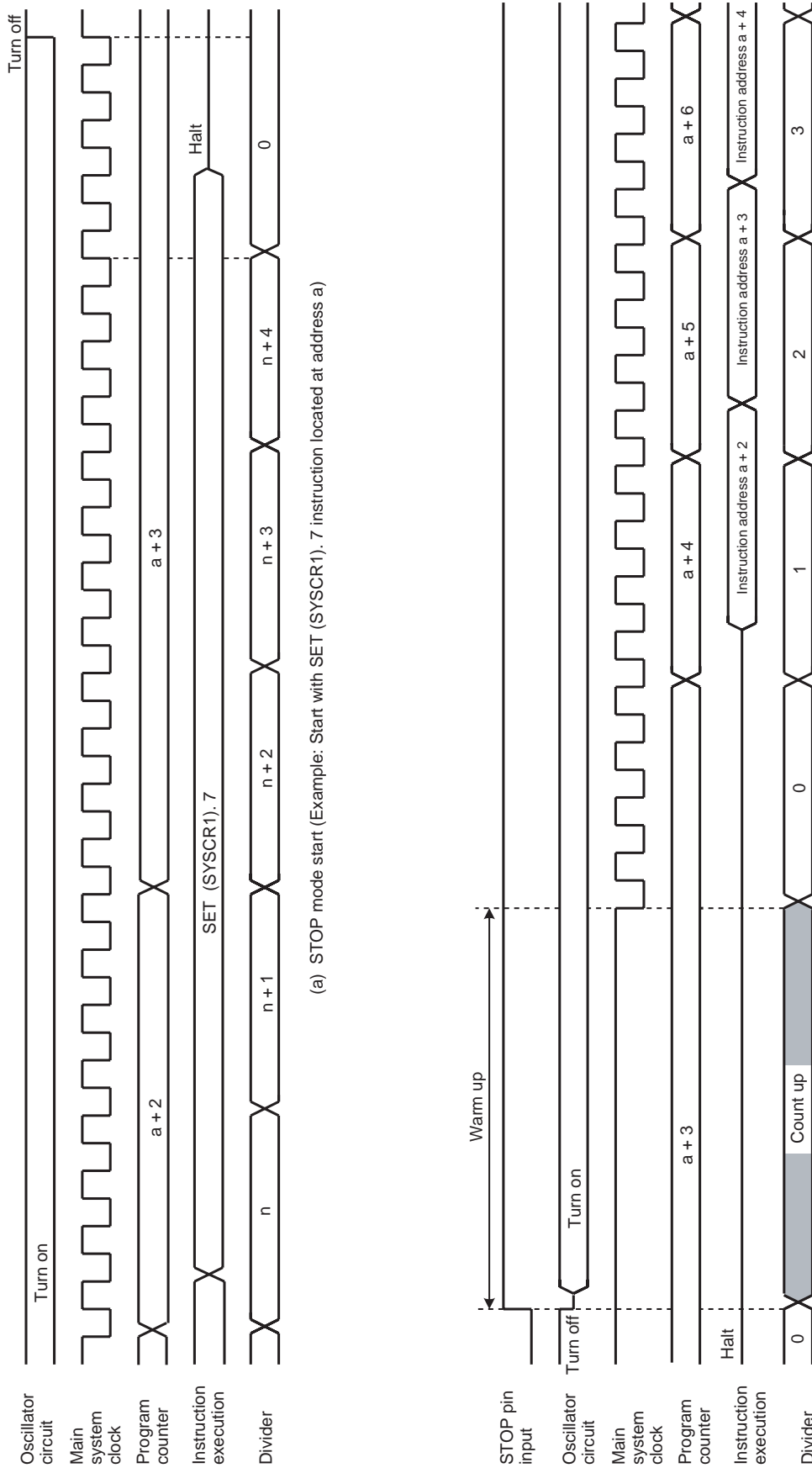


Figure 2-9 STOP Mode Start/Release

### 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.

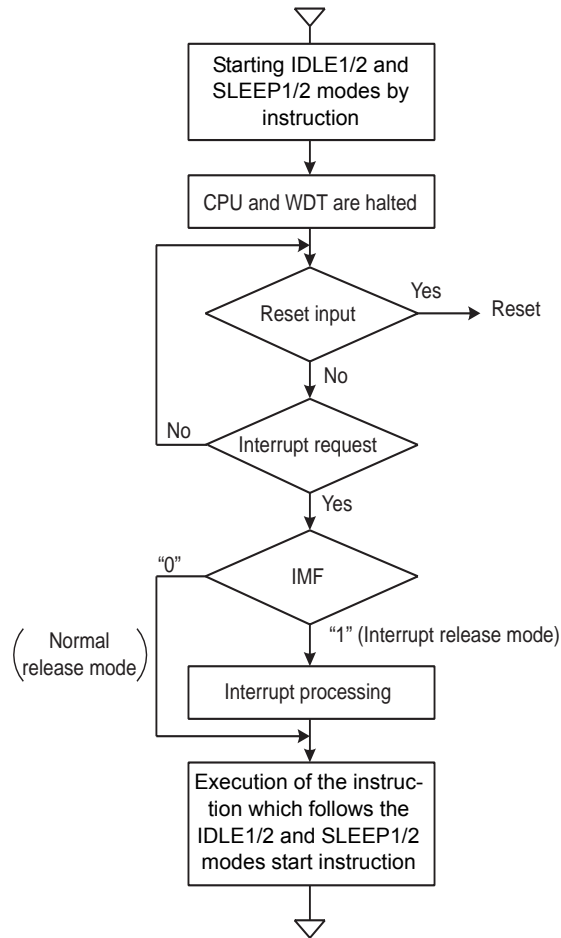


Figure 2-10 IDLE1/2 and SLEEP1/2 Modes



- 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.

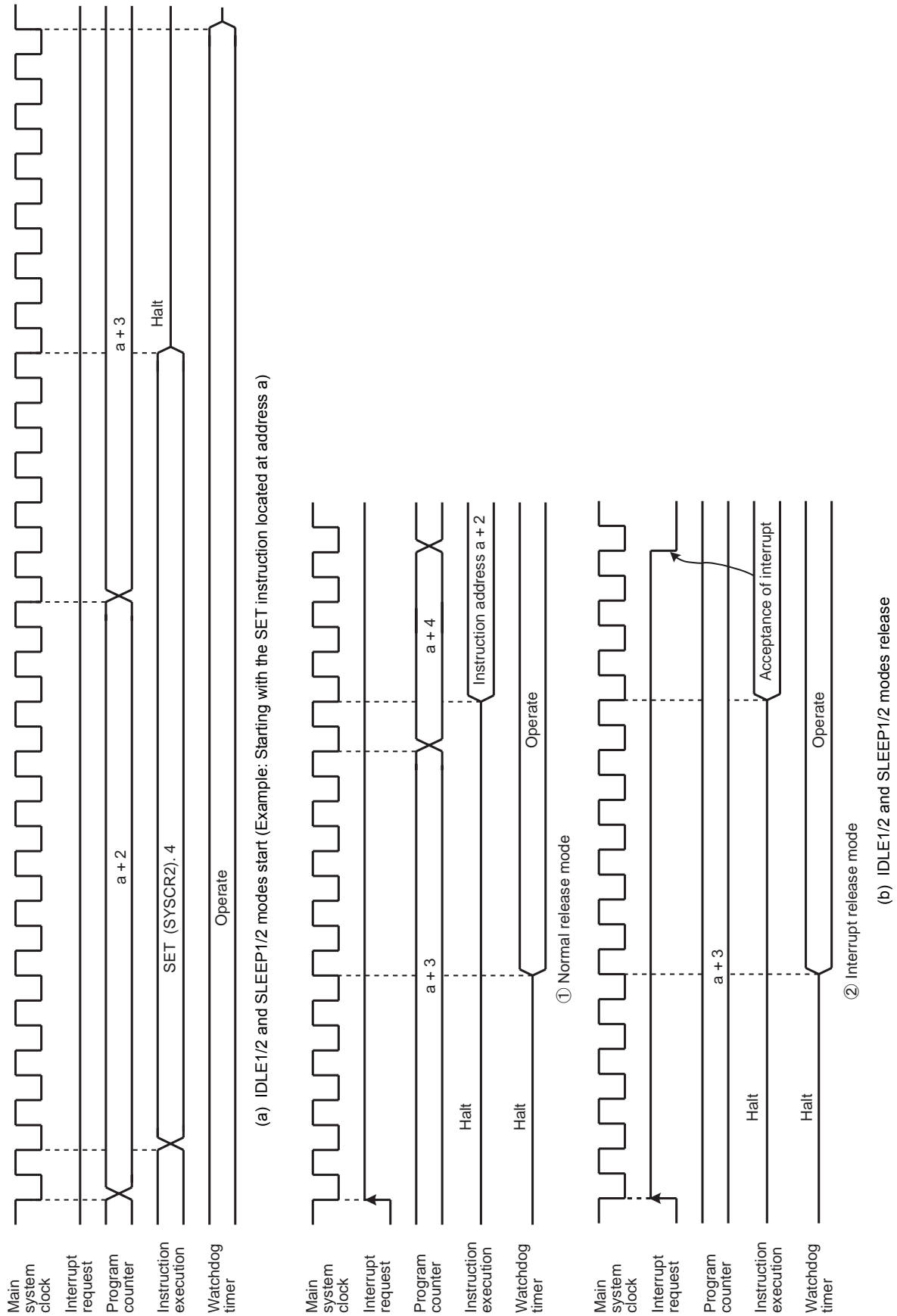


Figure 2-11 IDLE1/2 and SLEEP1/2 Modes Start/Release

2.2.4.3 IDLE0 and SLEEP0 modes (IDLE0, SLEEP0)

IDLE0 and SLEEP0 modes are controlled by the system control register 2 (SYSCR2) and the time base timer control register (TBTCCR). The following status is maintained during IDLE0 and SLEEP0 modes.

1. Timing generator stops feeding clock to peripherals except TBT.
2. The data memory, CPU registers, program status word and port output latches are all held in the status in effect before IDLE0 and SLEEP0 modes were entered.
3. The program counter holds the address 2 ahead of the instruction which starts IDLE0 and SLEEP0 modes.

Note: Before starting IDLE0 or SLEEP0 mode, be sure to stop (Disable) peripherals.

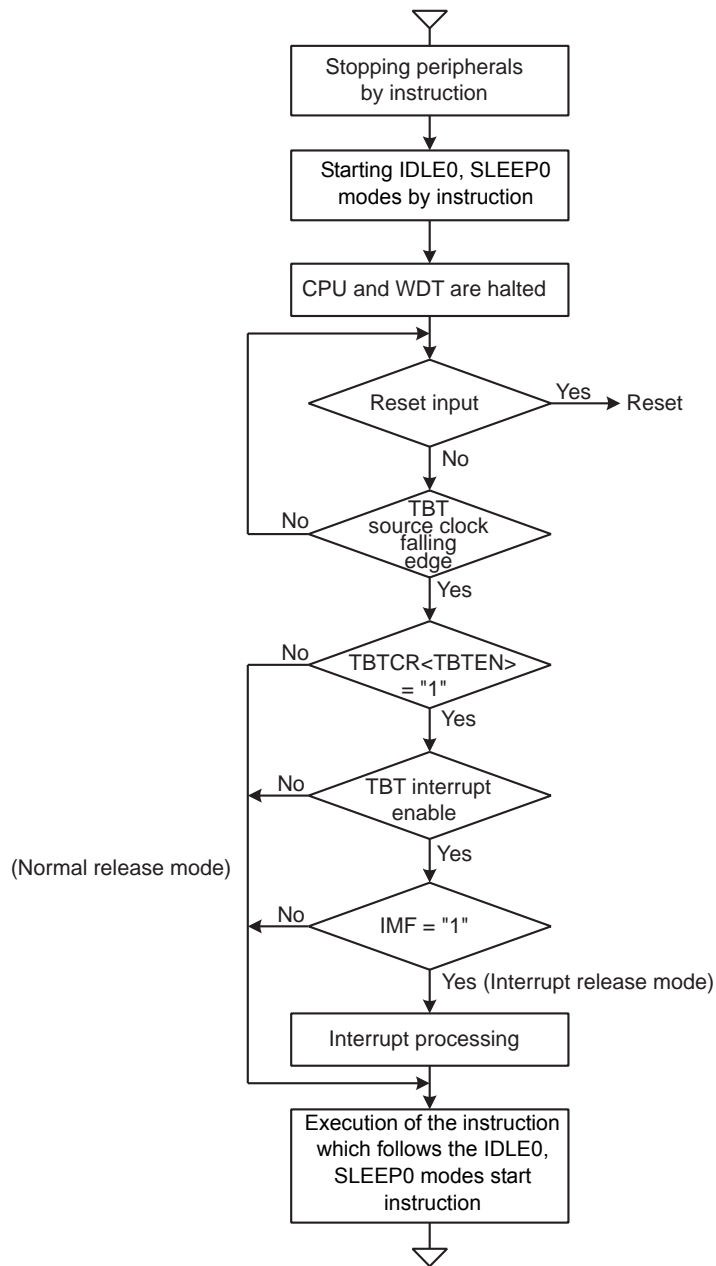


Figure 2-12 IDLE0 and SLEEP0 Modes

- 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”)

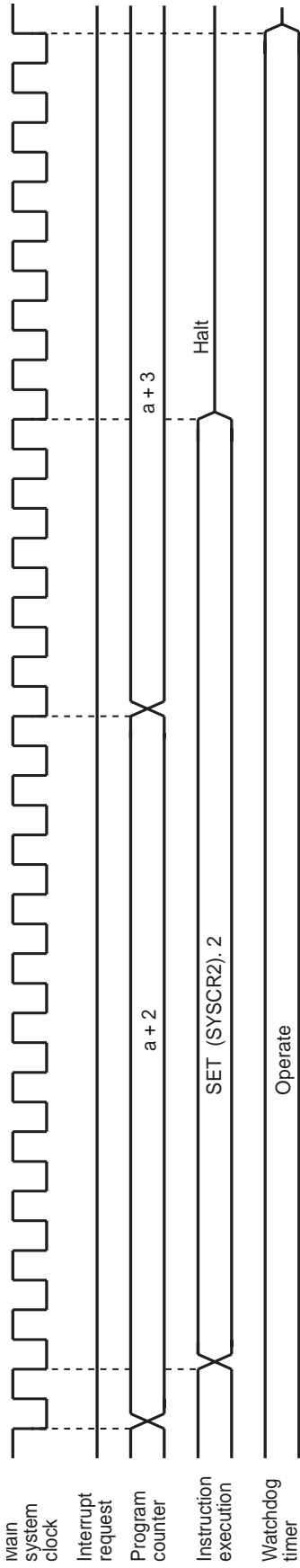
IDLE0 and SLEEP0 modes are released by the source clock falling edge, which is setting by the TBTCR<TBTCK>. After the falling edge is detected, the program operation is resumed from the instruction following the IDLE0 and SLEEP0 modes start instruction. Before starting the IDLE0 or SLEEP0 mode, when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

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

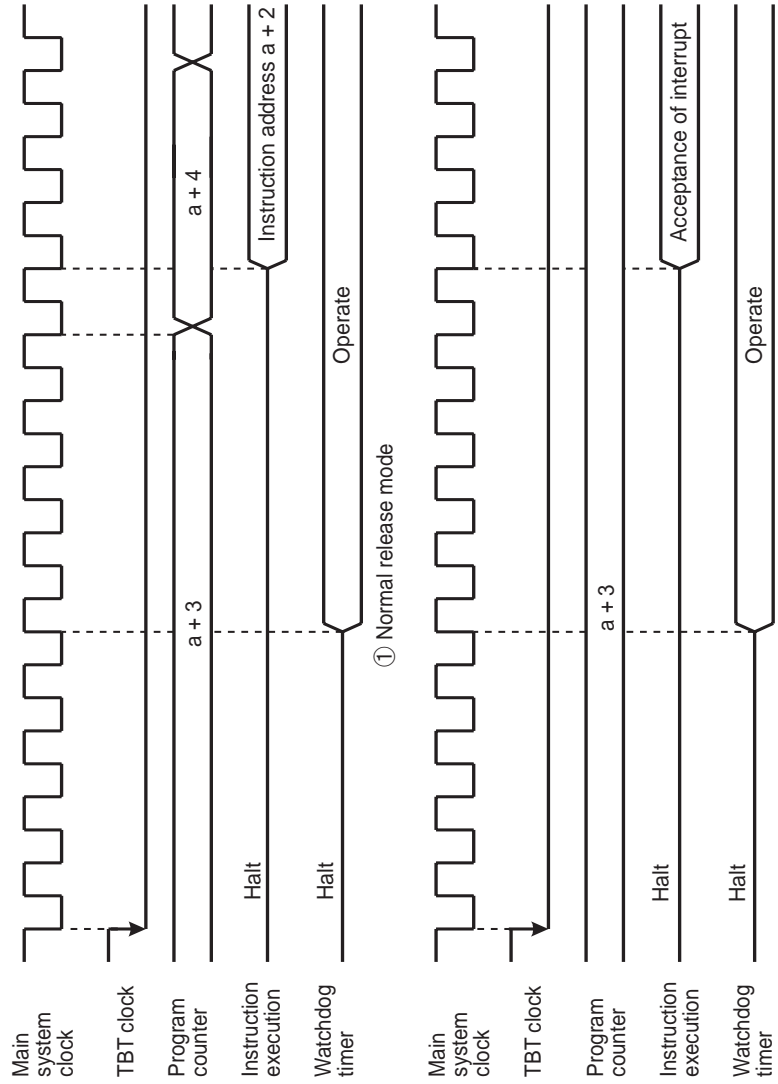
IDLE0 and SLEEP0 modes are released by the source clock falling edge, which is setting by the TBTCR<TBTCK> and INTTBT interrupt processing is started.

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). 3       ; 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

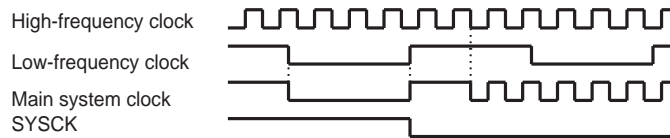
```

## (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). 3      ; 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
    
```

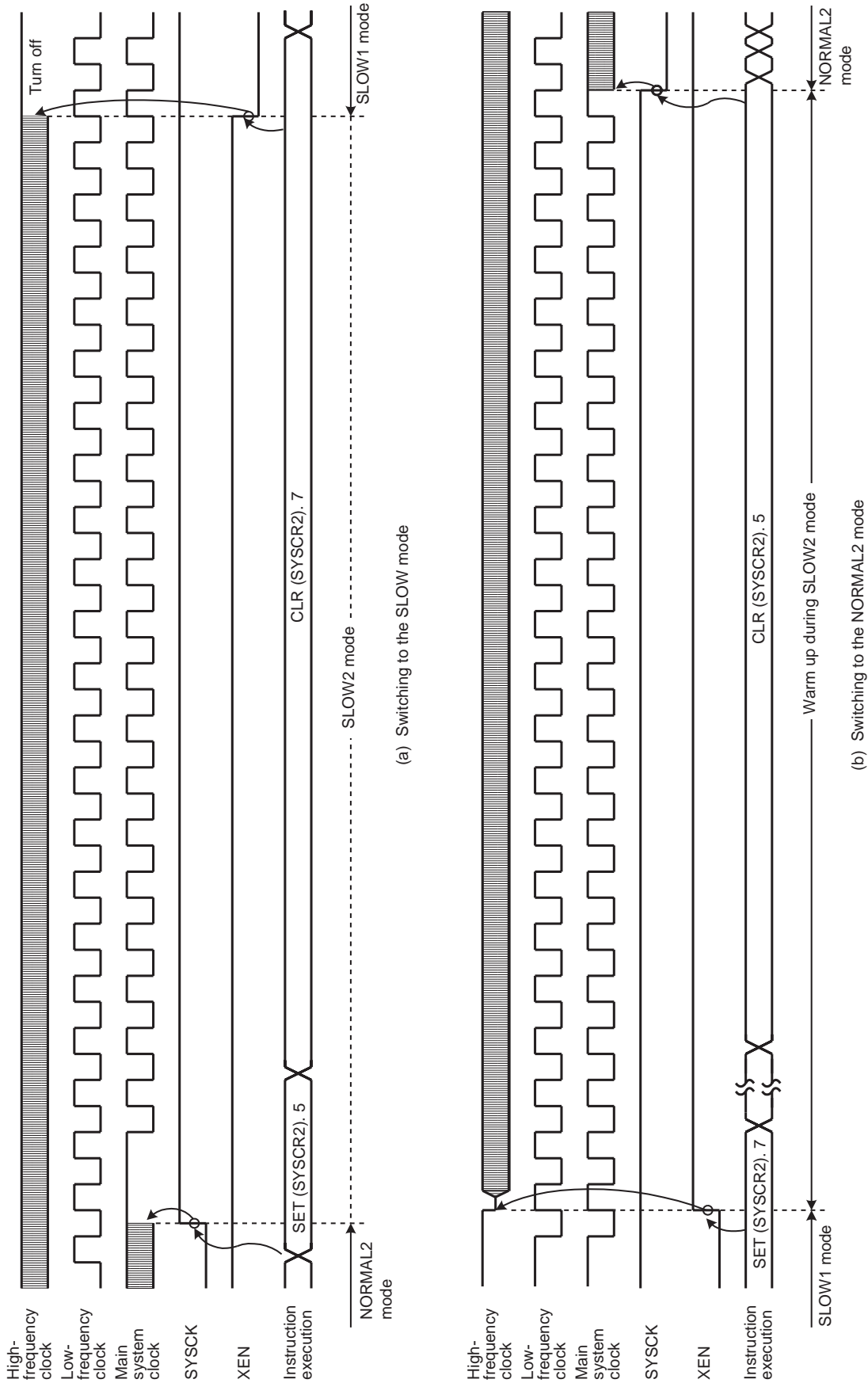


Figure 2-14 Switching between the NORMAL2 and SLOW Modes



## 2.3 Reset Circuit

The TMP86CS25AFG 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.

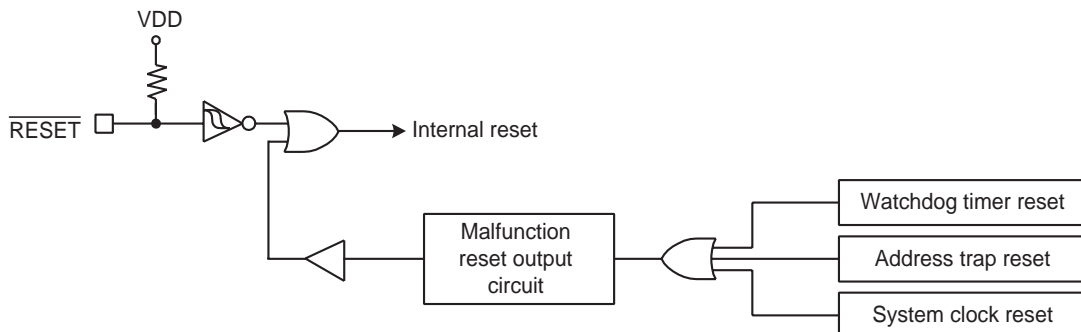
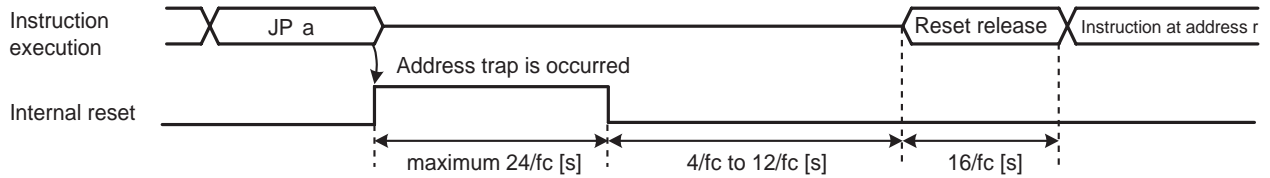


Figure 2-15 Reset Circuit

### 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 WDTCR1<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 $\mu$ 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 (WDTCR1<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  $\mu$ s at 16.0 MHz).





### 3. Interrupt Control Circuit

The TMP86CS25AFG has a total of 20 interrupt sources excluding reset, of which 4 source levels are multiplexed. Interrupts can be nested with priorities. Four of the internal interrupt sources are non-maskable while the rest are maskable.

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

Interrupt Factors		Enable Condition	Interrupt Latch	Vector Address	Priority
Internal/External	(Reset)	Non-maskable	–	FFFE	1
Internal	INTSWI (Software interrupt)	Non-maskable	–	FFFC	2
Internal	INTUNDEF (Executed the undefined instruction interrupt)	Non-maskable	–	FFFC	2
Internal	INTATRAP (Address trap interrupt)	Non-maskable	IL2	FFFA	2
Internal	INTWDT (Watchdog timer interrupt)	Non-maskable	IL3	FFF8	2
External	$\overline{INT0}$	IMF• EF4 = 1, INTOEN = 1	IL4	FFF6	5
External	INT1	IMF• EF5 = 1	IL5	FFF4	6
Internal	INTTBT	IMF• EF6 = 1	IL6	FFF2	7
External	INT2	IMF• EF7 = 1	IL7	FFF0	8
Internal	INTTC1	IMF• EF8 = 1	IL8	FFEE	9
Internal	INTRXD	IMF• EF9 = 1, IL9ER = 0	IL9	FFEC	10
Internal	INTSIO0	IMF• EF9 = 1, IL9ER = 1			
Internal	INTTXD	IMF• EF10 = 1, IL10ER = 0	IL10	FFEA	11
Internal	INTSIO1	IMF• EF10 = 1, IL10ER = 1			
Internal	INTTC4	IMF• EF11 = 1	IL11	FFE8	12
Internal	INTTC6	IMF• EF12 = 1	IL12	FFE6	13
Internal	INTADC	IMF• EF13 = 1	IL13	FFE4	14
External	INT3	IMF• EF14 = 1, IL14ER = 0	IL14	FFE2	15
Internal	INTTC3	IMF• EF14 = 1, IL14ER = 1			
External	$\overline{INT5}$	IMF• EF15 = 1, IL15ER = 0	IL15	FFE0	16
Internal	INTTC5	IMF• EF15 = 1, IL15ER = 1			

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

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

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

#### 3.1 Interrupt latches (IL15 to IL2)

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

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

Interrupt latches are not set to “1” by an instruction.

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

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

Example 1 :Clears interrupt latches

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

Example 2 :Reads interrupt latches

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

Example 3 :Tests interrupt latches

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

## 3.2 Interrupt enable register (EIR)

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

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

### 3.2.1 Interrupt master enable flag (IMF)

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

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

### 3.2.2 Individual interrupt enable flags (EF15 to EF4)

Each of these flags enables and disables the acceptance of its maskable interrupt. Setting the corresponding bit of an individual interrupt enable flag to “1” enables acceptance of its interrupt, and setting the bit to “0” disables acceptance. During reset, all the individual interrupt enable flags (EF15 to EF4) are initialized to “0” and all maskable interrupts are not accepted until they are set to “1”.

Note: In main program, before manipulating the interrupt enable flag (EF) or the interrupt latch (IL), be sure to clear IMF to “0” (Disable interrupt by DI instruction). Then set IMF newly again as required after operating on the EF or IL (Enable interrupt by EI instruction)  
In interrupt service routine, because the IMF becomes “0” automatically, clearing IMF need not execute nor-

mally 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.
:

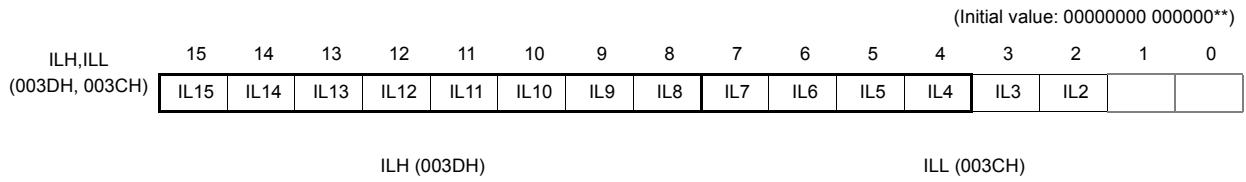
EI                                     ; IMF ← 1
    
```

Example 2 :C compiler description example

```

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

Interrupt Latches



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

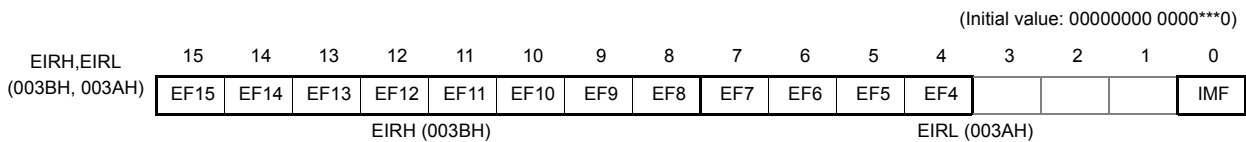
Note 1: To clear any one of bits IL7 to IL4, be sure to write "1" into IL2 and IL3.

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

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

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

Interrupt Enable Registers



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

Note 1: \*: Don't care

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

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

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



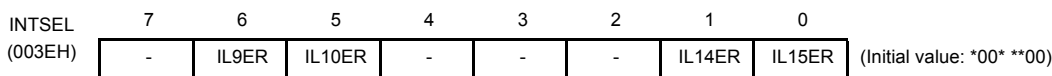
### 3.3 Interrupt Source Selector (INTSEL)

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

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

1. INTRXD and INTSIO0 share the interrupt source level whose priority is 10.
2. INTTXD and INTSIO1 share the interrupt source level whose priority is 11.
3. INT3 and INTTC3 share the interrupt source level whose priority is 15.
4.  $\overline{\text{INT5}}$  and INTTC5 share the interrupt source level whose priority is 16.

#### Interrupt source selector



IL9ER	Selects INTRXD or INTSIO0	0: INTRXD 1: INTSIO0	R/W
IL10ER	Selects INTTXD or INTSIO1	0: INTTXD 1: INTSIO1	R/W
IL14ER	Selects INT3 or INTTC3	0: INT3 1: INTTC3	R/W
IL15ER	Selects $\overline{\text{INT5}}$ or INTTC5	0: $\overline{\text{INT5}}$ 1: INTTC5	R/W

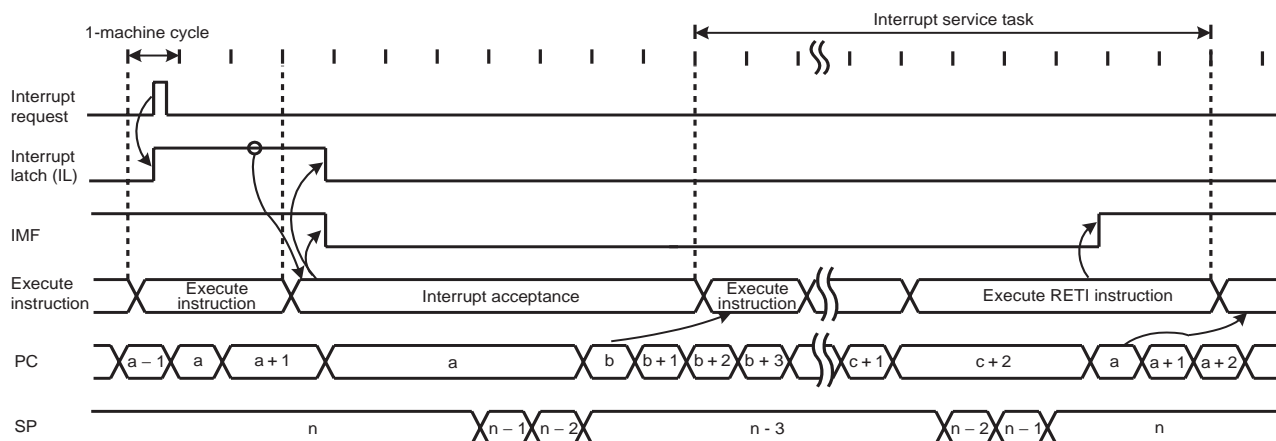
### 3.4 Interrupt Sequence

An interrupt request, which raised interrupt latch, is held, until interrupt is accepted or interrupt latch is cleared to “0” by resetting or an instruction. Interrupt acceptance sequence requires 8 machine cycles (2 μs @16 MHz) after the completion of the current instruction. The interrupt service task terminates upon execution of an interrupt return instruction [RETI] (for maskable interrupts) or [RETN] (for non-maskable interrupts). Figure 3-1 shows the timing chart of interrupt acceptance processing.

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

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

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



Note 1: a: Return address entry address, b: Entry address, c: Address which RETI instruction is stored

Note 2: On condition that interrupt is enabled, it takes  $38/f_c$  [s] or  $38/f_s$  [s] at maximum (If the interrupt latch is set at the first machine cycle on 10 cycle instruction) to start interrupt acceptance processing since its interrupt latch is set.

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

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

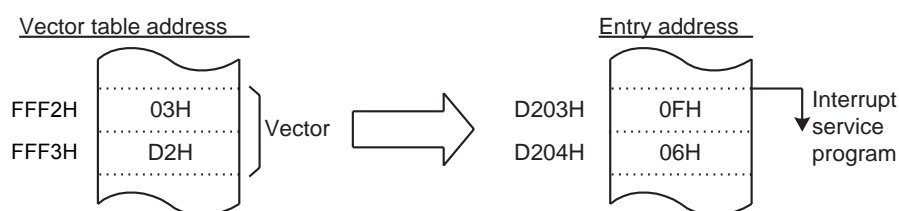


Figure 3-2 Vector table address,Entry address

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

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

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

### 3.4.2 Saving/restoring general-purpose registers

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

3.4.2.1 Using PUSH and POP instructions

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

Example :Save/store register using PUSH and POP instructions

```

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

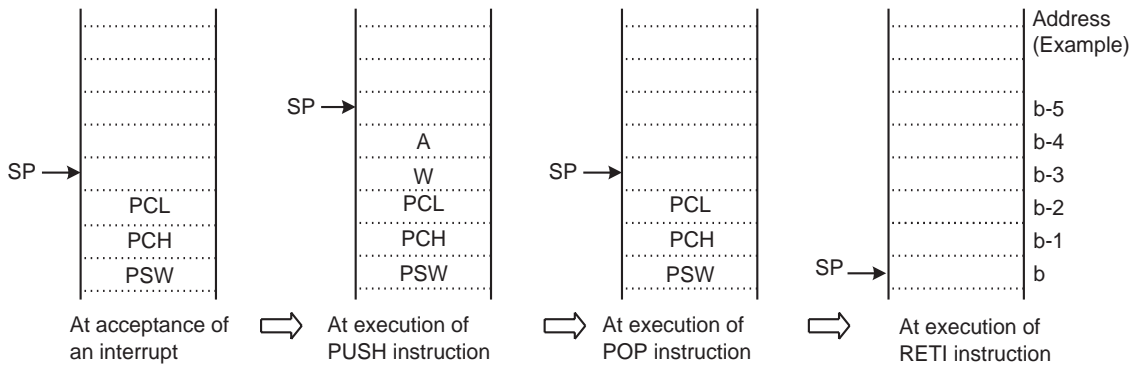


Figure 3-3 Save/store register using PUSH and POP instructions

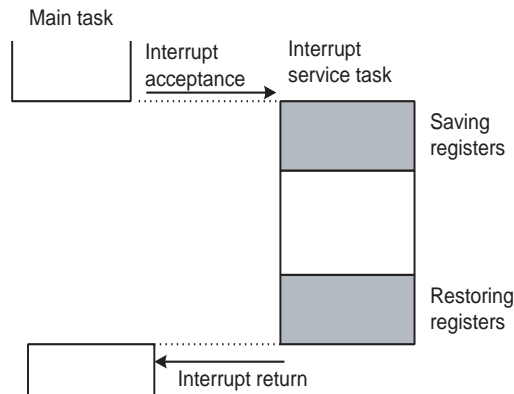
3.4.2.2 Using data transfer instructions

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

Example :Save/store register using data transfer instructions

```

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



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

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

### 3.4.3 Interrupt return

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

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

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

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

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

```

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

Example 2 :Restarting without returning interrupt

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

```

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

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

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

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

## 3.5 Software Interrupt (INTSW)

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

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

### 3.5.1 Address error detection

FFH is read if for some cause such as noise the CPU attempts to fetch an instruction from a non-existent memory address during single chip mode. Code FFH is the SWI instruction, so a software interrupt is generated and an address error is detected. The address error detection range can be further expanded by writing FFH to unused areas of the program memory. Address trap reset is generated in case that an instruction is fetched from RAM, DBR or SFR areas.

### 3.5.2 Debugging

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

## 3.6 Undefined Instruction Interrupt (INTUNDEF)

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

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

## 3.7 Address Trap Interrupt (INTATRAP)

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

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

## 3.8 External Interrupts

The TMP86CS25AFG 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 • INT0EN=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 • EF7 = 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 • EF14 = 1 and IL14ER=0	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 • EF15 = 1 and IL15ER=0	Falling edge	Pulses of less than 2/fc [s] are eliminated as noise. Pulses of 7/fc [s] or more are considered to be signals. In the SLOW or the SLEEP mode, pulses of less than 1/fs [s] are eliminated as noise. Pulses of 3.5/fs [s] or more are considered to be signals.

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

Note 2: When INT0EN = "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 TMP86CS25AFG 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 0F00H to 0FFFH.

This chapter shows the arrangement of the special function register (SFR) and data buffer register (DBR) for TMP86CS25AFG.

### 4.1 SFR

Address	Read	Write
0000H	Reserved	
0001H	P1DR	
0002H	P2DR	
0003H	P3DR	
0004H	P3LCR	
0005H	P5DR	
0006H	P6DR	
0007H	P7DR	
0008H	P1PRD	-
0009H	P2PRD	-
000AH	P3PRD	-
000BH	P5PRD	-
000CH	P6CR	
000DH	P7PRD	-
000EH	ADCCR1	
000FH	ADCCR2	
0010H	TREG1AL	
0011H	TREG1AM	
0012H	TREG1AH	
0013H	TREG1B	
0014H	TC1CR1	
0015H	TC1CR2	
0016H	TC1SR	-
0017H	Reserved	
0018H	TC3CR	
0019H	TC4CR	
001AH	TC5CR	
001BH	TC6CR	
001CH	TTREG3	
001DH	TTREG4	
001EH	TTREG5	
001FH	TTREG6	
0020H	ADCDR1	-
0021H	ADCDR2	-
0022H	Reserved	
0023H	Reserved	
0024H	Reserved	
0025H	UARTSR	UARTCR1

Address	Read	Write
0026H	-	UARTCR2
0027H		LCDCTL1
0028H		LCDCTL2
0029H		P1LCR
002AH		P5LCR
002BH		P7LCR
002CH		PWREG3
002DH		PWREG4
002EH		PWREG5
002FH		PWREG6
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		INTSEL
003FH		PSW

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

Note 2: - ; Cannot be accessed.

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

## 4.2 DBR

Address	Read	Write
0F80H		Reserved
0F81H		Reserved
0F82H		Reserved
0F83H		Reserved
0F84H		Reserved
0F85H		Reserved
0F86H		Reserved
0F87H		Reserved
0F88H		Reserved
0F89H		Reserved
0F8AH		Reserved
0F8BH		Reserved
0F8CH		Reserved
0F8DH		Reserved
0F8EH		Reserved
0F8FH		Reserved
0F90H		SIO0BR0
0F91H		SIO0BR1
0F92H		SIO0BR2
0F93H		SIO0BR3
0F94H		SIO0BR4
0F95H		SIO0BR5
0F96H		SIO0BR6
0F97H		SIO0BR7
0F98H	-	SIO0CR1
0F99H	SIO0SR	SIO0CR2
0F9AH	-	STOPCR
0F9BH	RDBUF	TDBUF
0F9CH		Reserved
0F9DH		Reserved
0F9EH		Reserved
0F9FH		Reserved

Address	Read	Write
0FA0H		SIO1BR0
0FA1H		SIO1BR1
0FA2H		SIO1BR2
0FA3H		SIO1BR3
0FA4H		SIO1BR4
0FA5H		SIO1BR5
0FA6H		SIO1BR6
0FA7H		SIO1BR7
0FA8H	-	SIO1CR1
0FA9H	SIO1SR	SIO1CR2
0FAAH		Reserved
0FABH		Reserved
0FACH		Reserved
0FADH		Reserved
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		MULSEL
0FC1H		Reserved
0FC2H		Reserved
0FC3H		Reserved
0FC4H		Reserved
0FC5H		Reserved
0FC6H		Reserved
0FC7H		Reserved
0FC8H		Reserved
0FC9H		Reserved
0FCAH		Reserved
0FCBH		Reserved
0FCCH		Reserved
0FCDH		Reserved
0FCEH		Reserved
0FCFH		Reserved
0FD0H		Reserved
0FD1H		Reserved
0FD2H		Reserved
0FD3H		Reserved
0FD4H		Reserved
0FD5H		Reserved
0FD6H		Reserved
0FD7H		Reserved
0FD8H		Reserved
0FD9H		Reserved
0FDAH		Reserved
0FDBH		Reserved
0FDC		Reserved
0FDDH		Reserved
0FDEH		Reserved
0DFH		Reserved

Address	Read	Write
0FE0H		Reserved
::		::
0FFFH		Reserved

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.).

Note 4: This product has a LCD display data buffer (assigned to address 0F00H to 0F7FH). For detail, refer to the chapter of LCD driver.



## 5. I/O Ports

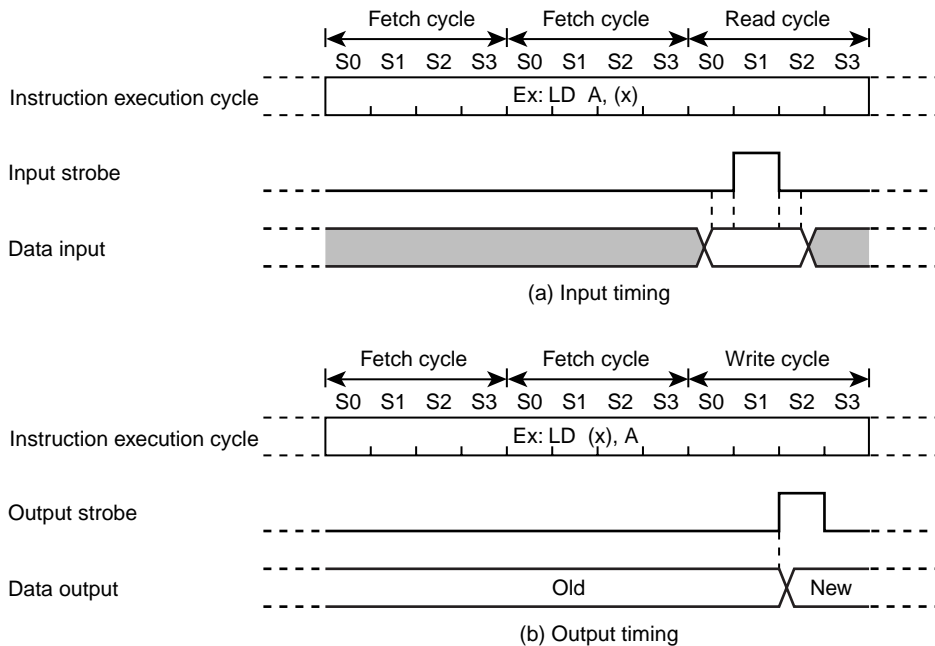
The TMP86CS25AFG have 6 parallel input/output ports (42 pins) as follows.

	Primary Function	Secondary Functions
Port P1	8-bit I/O port	External interrupt input, serial interface input/output, UART input/output and segment output.
Port P2	3-bit I/O port	Low-frequency resonator connections, external interrupt input, STOP mode release signal input.
Port P3	7-bit I/O port	Timer/Counter input/output and divider output and segment/common output.
Port P5	8-bit I/O port	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	Common output. Timer/Counter input/output and divider 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 is also used as an external interrupt input, serial interface input/output, UART input/output and segment output of LCD. When used as a segment pins of LCD, the respective bit of P1LCR should be set to “1”.

When used as an input port or a secondary function (except for segment) pins, the respective output latch (P1DR) should be set to “1” and its corresponding P1LCR bit should be set to “0”. When used as an output port, the respective P1LCR bit should be set to “0”. During reset, the output latch is initialized to “1”.

P1 port output latch (P1DR) and P1 port terminal input (P1PRD) are located on their respective address.

When read the output latch data, the P1DR should be read and when read the terminal input data, the P1PRD register should be read.

If the terminal input data which is configured as LCD segment output is read, unstable data is read.

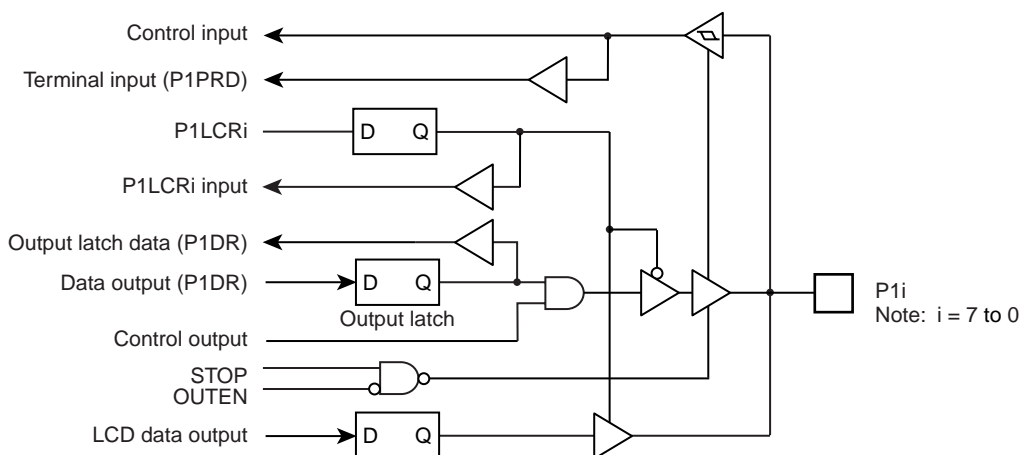


Figure 5-2 Port 1

	7	6	5	4	3	2	1	0	
P1DR (0001H) R/W	P17 SEG59 SCK0	P16 SEG58 TxD SO0	P15 SEG57 RxD SI0	P14 SEG56 MUL6	P13 SEG55 MUL5	P12 SEG54 MUL4	P11 SEG53	P10 SEG52	(Initial value: 1111 1111)
P1LCR (0029H)									(Initial value: 0000 0000)
P1LCR	Port P1/segment output select (set for each bit individually)						0: P1 input/output port or secondary function (except for segment) 1: segment output		R/W
P1PRD (0008H) Read only	7	6	5	4	3	2	1	0	
	P17	P16	P15	P14	P13	P12	P11	P10	

Note: With ports assigned as MUL6 to MUL0, assigned pins can be switched by the multi function register (MULSEL). The assigned functions are shown in “5.7 Multi Function Register”.



## 5.2 Port P2 (P22 to P20)

Port P2 is a 3-bit input/output port.

It is also used as an external interrupt, a STOP mode release signal input, and low-frequency crystal oscillator connection pins. When used as an input port or a secondary function pins, respective output latch (P2DR) should be set to “1”.

During reset, the P2DR is initialized to “1”.

A low-frequency crystal oscillator (32.768 kHz) is connected to pins P21 (XTIN) and P22 (XTOUT) in the dual-clock mode. In the single-clock mode, pins P21 and P22 can be used as normal input/output ports.

It is recommended that pin P20 should be used as an external interrupt input, a STOP mode release signal input, or an input port. If it is used as an output port, the interrupt latch is set on the falling edge of the output pulse.

P2 port output latch (P2DR) and P2 port terminal input (P2PRD) are located on their respective address.

When read the output latch data, the P2DR should be read and when read the terminal input data, the P2PRD register should be read. If a read instruction is executed for port P2, read data of bits 7 to 3 are unstable.

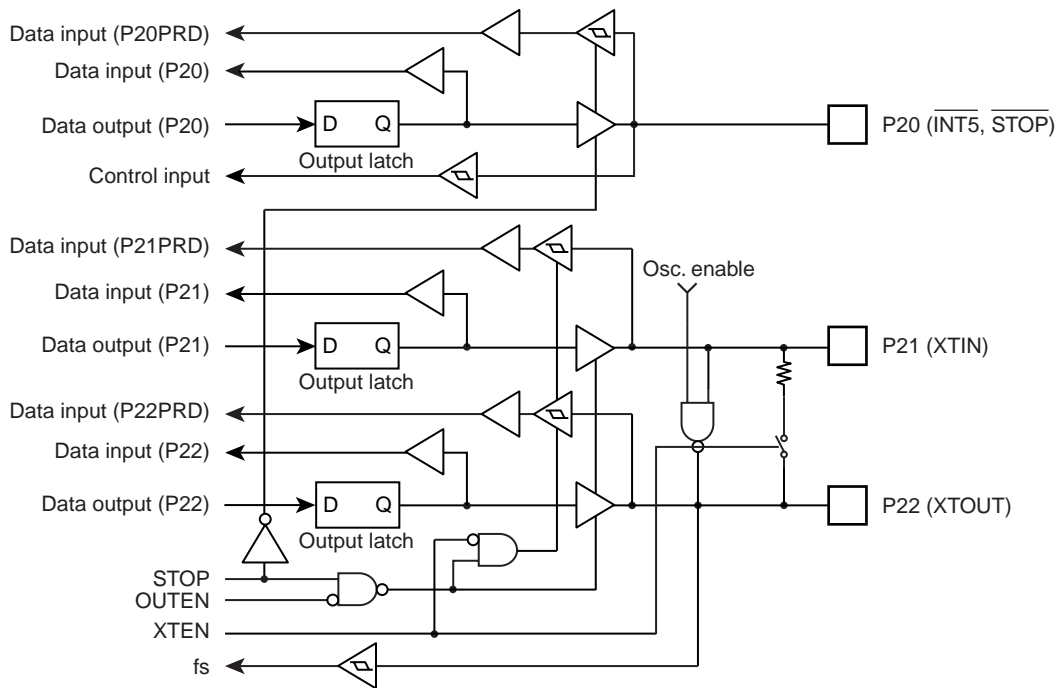


Figure 5-3 Port 2

	7	6	5	4	3	2	1	0	
P2DR (0002H) R/W						P22 XTOUT	P21 XTIN	P20 INT5 STOP	(Initial value: **** *111)
P2PRD (0009H) Read only						P22	P21	P20	

Note: 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.

### 5.3 Port P3 (P36 to P30)

Port P3 is a 7-bit input/output port.

It is also used as a External interrupt input, timer/counter input/output, divider output and LCD common/segment output.

When used as an input port or a secondary function pins, after setting segment/common output control (P3LCR) to “0” respective output latch (P3DR) should be set to “1”. During reset, the P3DR is initialized to “1”, and segment output control (P3LCR) is initialized by “0”. In using it as LCD segment/ common output, it sets the bit to which P3LCR corresponds to “1”.

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 port P3, read data of bit 7 is unstable.

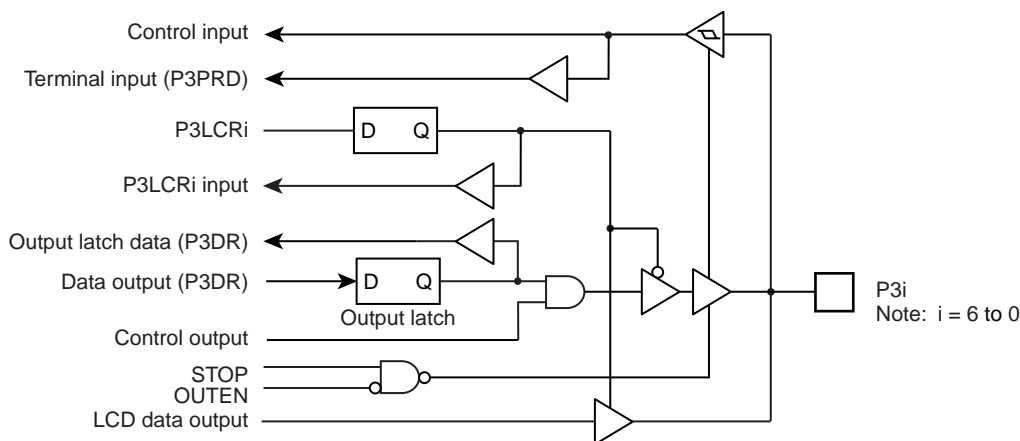


Figure 5-4 Port 3

	7	6	5	4	3	2	1	0	
P3DR (0003H) R/W		P36 COM7 MUL6	P35 COM6 MUL5	P34 COM5 MUL4	P33 SEG51 MUL3	P32 SEG50 MJUL2	P31 SEG49 MJUL1	P30 SEG48 MJUL0	(Initial value: *111 1111)
P3LCR (0004H)	7	6	5	4	3	2	1	0	(Initial value: *000 0000)
P3LCR	Port P3 control (set for each bit individually)			0: P3 input/output port or function except LCD segment or common output 1: LCD segment output/common output				R/W	
P3PRD (000AH) Read only	7	6	5	4	3	2	1	0	
		P36	P35	P34	P33	P32	P31	P30	

Note: With ports assigned as MUL6 to MUL0, assigned pins can be switched by the multi function register (MULSEL). The assigned functions are shown in “5.7 Multi Function Register”.

### 5.4 Port P5 (P57 to P50)

Port P5 is an 8-bit input/output port which is also used as a segment pins of LCD.

When used as input port, the respective output latch (P5DR) should be set to “1”.

During reset, the P5DR is initialized to “1”.

When used as a segment pins of LCD, the respective bit of P5LCR should be set to “1”. When used as an output port, the respective P5LCR bit should be set to “0”.

P5 port output latch (P5DR) and P5 port terminal input (P5PRD) are located on their respective address.

When read the output latch data, the P5DR should be read and when read the terminal input data, the P5PRD register should be read. If the terminal input data which is configured as LCD segment output is read, unstable data is read.

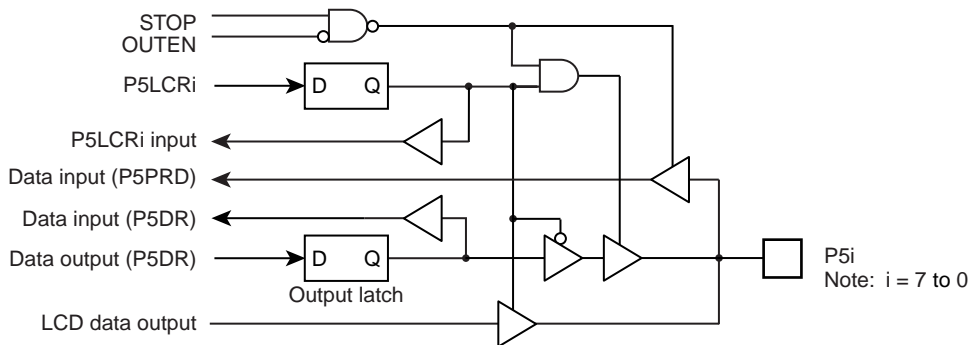


Figure 5-5 Port 5

	7	6	5	4	3	2	1	0	
P5DR (0005H) R/W	P57 SEG47	P56 SEG46	P55 SEG45	P54 SEG44	P53 SEG43	P52 SEG42	P51 SEG41	P50 SEG40	(Initial value: 1111 1111)
P5LCR (002AH)									(Initial value: 0000 0000)
P5LCR	Port P5/segment output select (set for each bit individually)							0: P5 input/output port 1: LCD segment output	R/W
	7	6	5	4	3	2	1	0	
P5PRD (000BH) Read only	P57	P56	P55	P54	P53	P52	P51	P50	

## 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 (P6CR), the P6 output latch (P6DR), and ADCCR1<AINDS>. During reset, P6CR and P6DR are initialized to “0” and ADCCR1<AINDS> is set to “1”. At the same time, the input data of pins P67 to P60 are fixed to “0”. To use port P6 as an input port, external interrupt input, timer/counter input or key on wake up input, set data of P6DR to “1” and P6CR to “0”. To use it as an output port, set data of P6CR to “1”. To use it as an analog input, set data of P6DR to “0” and P6CR to “0”, and start the AD. It is the penetration electric current measures by the analog voltage.

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.

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

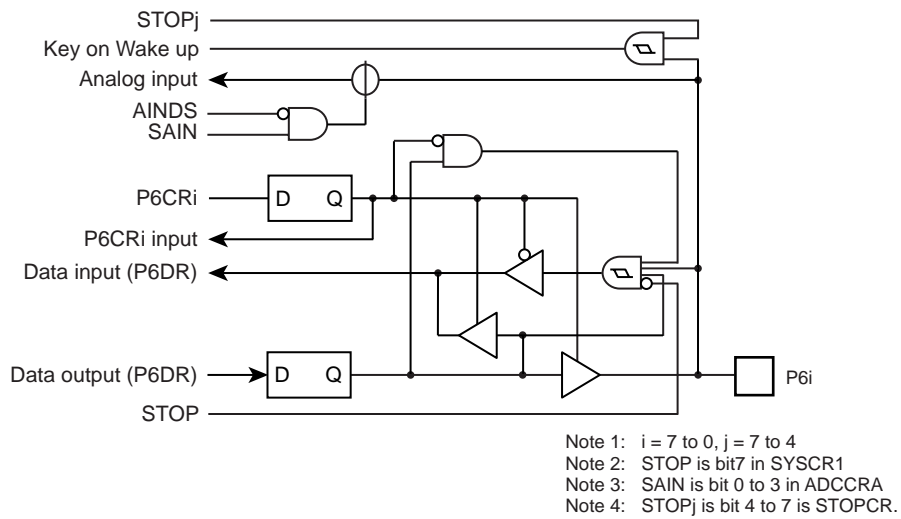


Figure 5-6 Port 6

	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)
P6CR (000CH)									(Initial value: 0000 0000)

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

- Note 1: Do not set output mode to pin which is used for an analog input.
- Note 2: When used as an INT0, ECNT and ECIN pins of a secondary function, the respective bit of P6CR should be set to “0” and the P6 should set to “1”.
- Note 3: When used as an STOP2 to STOP5 pins of Key on Wake up, the respective bit of P6CR should be set to “0”.
- Note 4: When a read instruction for port P6 is executed, the bit of Analog input mode becomes read data “0”.
- Note 5: Although P6DR is a read/writer register, because it is also used as an input mode control function, read-modify-write instructions such as bit manipulate instructions cannot be used.  
Read-modify-write instruction writes the all data of 8-bit after data is read and modified. Because a bit setting Input mode read data of terminal, the output latch is changed by these instruction. So P6 port can not input data.

### 5.6 Port P7 (P77 to P70)

Port P7 is an 8-bit input/output port which is also used as an external interrupt input, a divider output a segment pins of LCD.

When used as input port or a secondary function pins, the respective output latch (P7DR) should be set to “1”.

During reset, the P7DR is initialized to “1”.

When used as a segment pins of LCD, the respective bit of P7LCR should be set to “1”. When used as an output port, the respective P7LCR bit should be set to “0”.

P7 port output latch (P7DR) and P7 port terminal input (P7PRD) are located on their respective address.

When read the output latch data, the P7DR should be read and when read the terminal input data, the P7PRD register should be read. If the terminal input data which is configured as LCD segment output is read, unstable data is read.

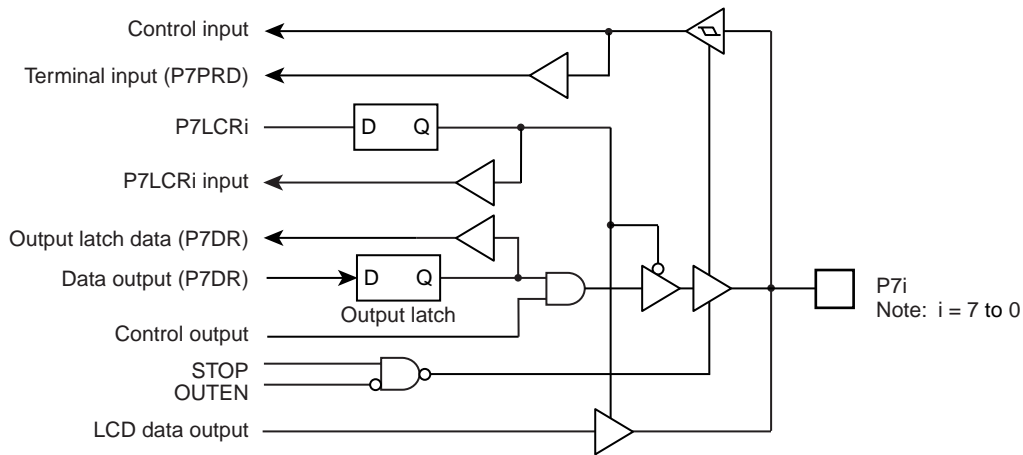


Figure 5-7 Port 7

	7	6	5	4	3	2	1	0	
P7DR (0007H) R/W	P77 COM15 SCK1	P76 COM14 SO0	P75 COM13 SI1	P74 COM12 MUL3	P73 COM11 MUL2	P72 COM10 MUL1	P71 COM9 MUL0	P70 COM8	(Initial value: 1111 1111)
P7LCR (002BH)									(Initial value: 0000 0000)
P7LCR	Port P7/segment output select (set for each bit individually)						0: P7 input/Output port 1: segment output		R/W
P7PRD (000DH) Read only	P77	P76	P75	P74	P73	P72	P71	P70	

Note: With ports assigned as MUL6 to MUL0, assigned pins can be switched by the multi function register (MULSEL). The assigned functions are shown in “5.7 Multi Function Register”.

## 5.7 Multi Function Register

With ports assigned as MUL6 to MUL0, assigned pins can be switched by the multi function register (MULSEL).

### Multi Function Register

MULSEL (OFC0H)	7	6	5	4	3	2	1	0
		MUL6	MUL5	MUL4	MUL3	MUL2	MUL1	MUL0

MUL6	INT3 function pin select	0: P14 1: P36	R/W
MUL5	INT2 function pin select	0: P13 1: P35	
MUL4	INT1 function pin select	0: P12 1: P34	
MUL3	$\overline{\text{PPG6}}/\overline{\text{PWM6}}/\overline{\text{PDO6}}$ and TC6 functions pin select	0: P33 1: P74	
MUL2	$\overline{\text{PPG4}}/\overline{\text{PWM4}}/\overline{\text{PDO4}}$ and TC4 functions pin select	0: P32 1: P73	
MUL1	$\overline{\text{PWM3}}/\overline{\text{PDO3}}$ and TC3 functions pin select	0: P31 1: P72	
MUL0	$\overline{\text{DVO}}$ function pin select	0: P30 1: P71	

## 6. Watchdog Timer (WDT)

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

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

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

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

### 6.1 Watchdog Timer Configuration

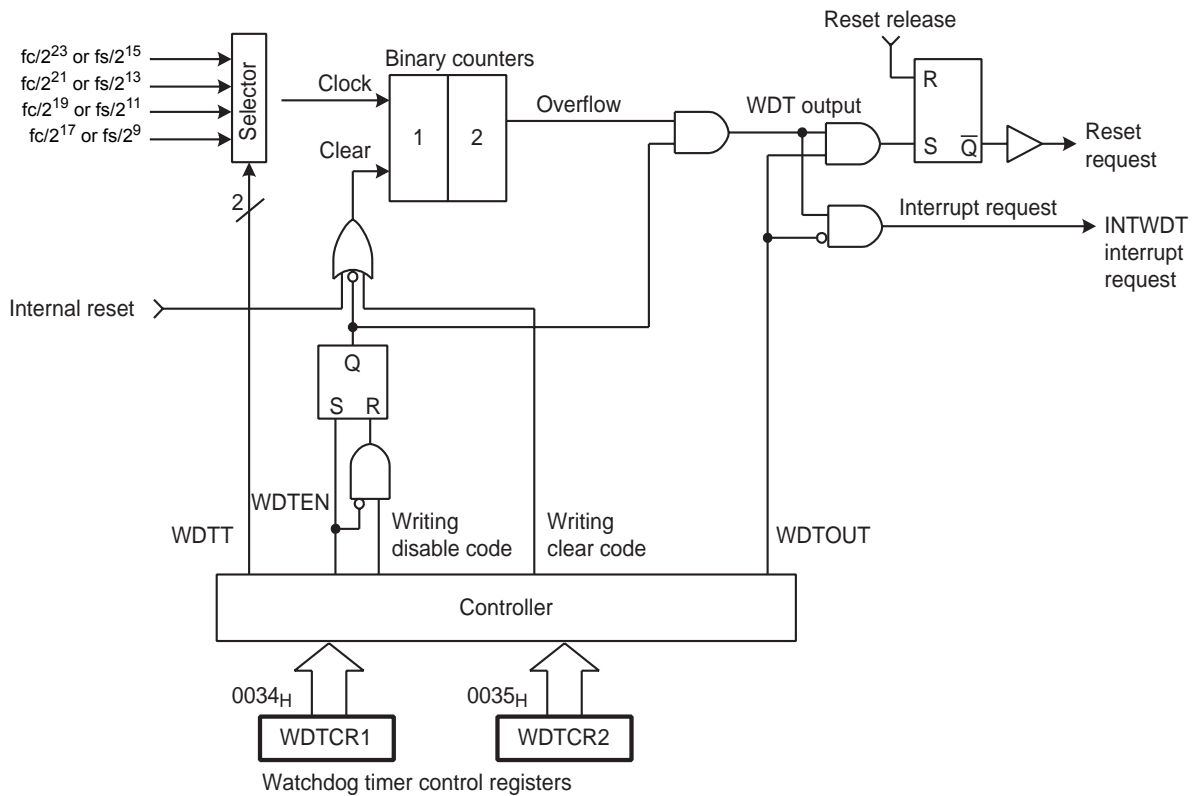


Figure 6-1 Watchdog Timer Configuration

## 6.2 Watchdog Timer Control

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

### 6.2.1 Malfunction Detection Methods Using the Watchdog Timer

The CPU malfunction is detected, as shown below.

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

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

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

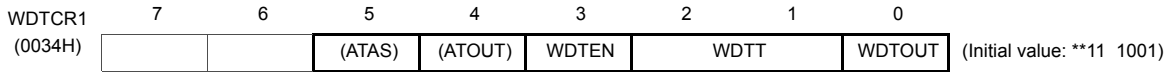
Note: The watchdog timer consists of an internal divider and a two-stage binary counter. When the clear code 4EH is written, only the binary counter is cleared, but not the internal divider. The minimum binary-counter overflow time, that depends on the timing at which the clear code (4EH) is written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

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

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



Watchdog Timer Control Register 1



WDTEN	Watchdog timer enable/disable	0: Disable (Writing the disable code to WDTCR2 is required.) 1: Enable	Write only					
WDTT	Watchdog timer detection time [s]	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td colspan="2" style="text-align: center;">NORMAL1/2 mode</td> <td rowspan="2" style="text-align: center;">SLOW1/2 mode</td> </tr> <tr> <td style="text-align: center;">DV7CK = 0</td> <td style="text-align: center;">DV7CK = 1</td> </tr> </table>	NORMAL1/2 mode		SLOW1/2 mode	DV7CK = 0	DV7CK = 1	Write only
		NORMAL1/2 mode		SLOW1/2 mode				
		DV7CK = 0	DV7CK = 1					
		00	$2^{25}/fc$	$2^{17}/fs$	$2^{17}/fs$			
		01	$2^{23}/fc$	$2^{15}/fs$	$2^{15}/fs$			
10	$2^{21}/fc$	$2^{13}/fs$	$2^{13}/fs$					
11	$2^{19}/fc$	$2^{11}/fs$	$2^{11}/fs$					
WDTOUT	Watchdog timer output select	0: Interrupt request 1: Reset request	Write only					

Note 1: After clearing WDTOUT to “0”, the program cannot set it to “1”.

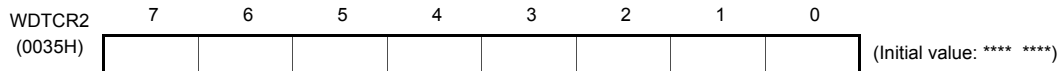
Note 2: fc: High-frequency clock [Hz], fs: Low-frequency clock [Hz], \*: Don't care

Note 3: WDTCR1 is a write-only register and must not be used with any of read-modify-write instructions. If WDTCR1 is read, a don't care is read.

Note 4: To activate the STOP mode, disable the watchdog timer or clear the counter immediately before entering the STOP mode. After clearing the counter, clear the counter again immediately after the STOP mode is inactivated.

Note 5: To clear WDTEEN, set the register in accordance with the procedures shown in “1.2.3 Watchdog Timer Disable”.

Watchdog Timer Control Register 2



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

Note 1: The disable code is valid only when WDTCR1<WDTEN> = 0.

Note 2: \*: Don't care

Note 3: The binary counter of the watchdog timer must not be cleared by the interrupt task.

Note 4: Write the clear code 4EH using a cycle shorter than 3/4 of the time set in WDTCR1<WDTT>.

6.2.2 Watchdog Timer Enable

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

### 6.2.3 Watchdog Timer Disable

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

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

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

Example :Disabling the watchdog timer

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

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

WDTT	Watchdog Timer Detection Time[s]		
	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

### 6.2.4 Watchdog Timer Interrupt (INTWDT)

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

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

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

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

Example :Setting watchdog timer interrupt

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

### 6.2.5 Watchdog Timer Reset

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

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

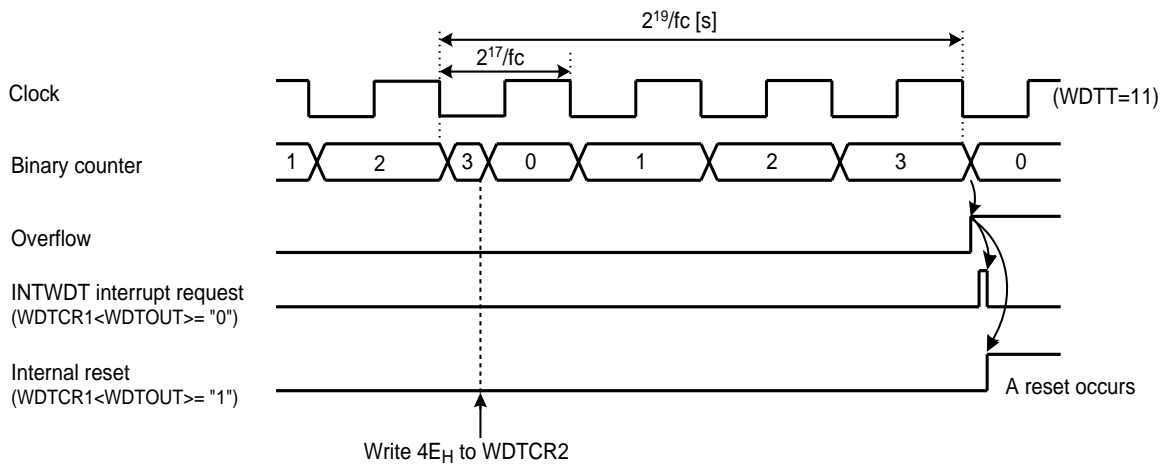
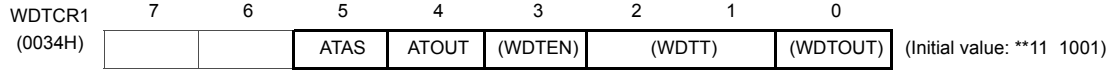


Figure 6-2 Watchdog Timer Interrupt

## 6.3 Address Trap

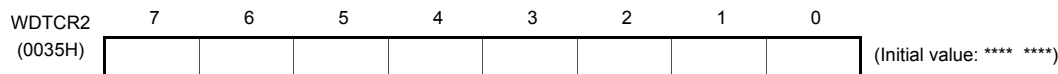
The Watchdog Timer Control Register 1 and 2 share the addresses with the control registers to generate address traps.

### Watchdog Timer Control Register 1



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

### Watchdog Timer Control Register 2



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

#### 6.3.1 Selection of Address Trap in Internal RAM (ATAS)

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

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

#### 6.3.2 Selection of Operation at Address Trap (ATOUT)

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

#### 6.3.3 Address Trap Interrupt (INTATRAP)

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

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

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

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

### 6.3.4 Address Trap Reset

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

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



## 7. Time Base Timer (TBT)

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

### 7.1 Time Base Timer

#### 7.1.1 Configuration

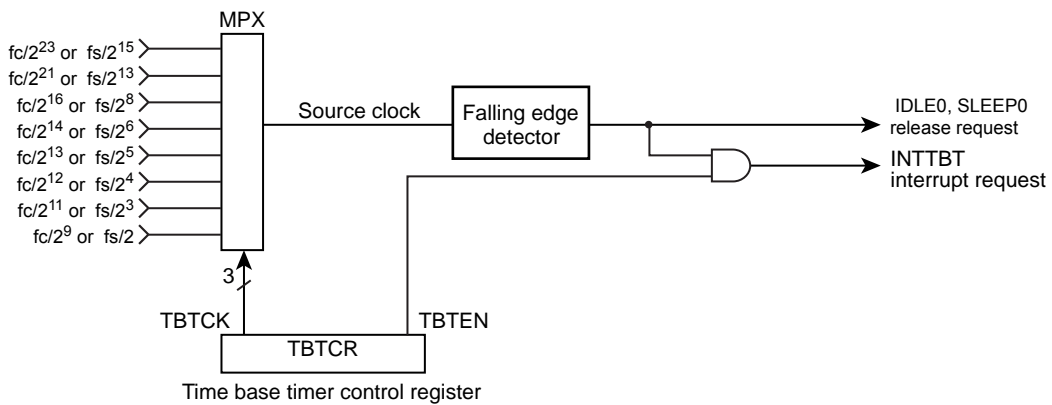


Figure 7-1 Time Base Timer configuration

#### 7.1.2 Control

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

#### Time Base Timer Control Register

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

TBTCR	Time Base Timer enable / disable	0: Disable 1: Enable			R/W	
		NORMAL 1/2, IDLE 1/2 Mode		SLOW 1/2 SLEEP 1/2 Mode		
TBTC	Time Base Timer interrupt Frequency select : [Hz]	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 7-1 Time Base Timer Interrupt Frequency ( Example :  $fc = 16.0$  MHz,  $fs = 32.768$  kHz )

TBTCK	Time Base Timer Interrupt Frequency [Hz]		
	NORMAL1/2, IDLE1/2 Mode	NORMAL1/2, IDLE1/2 Mode	SLOW1/2, SLEEP1/2 Mode
	DV7CK = 0	DV7CK = 1	
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	-

### 7.1.3 Function

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

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

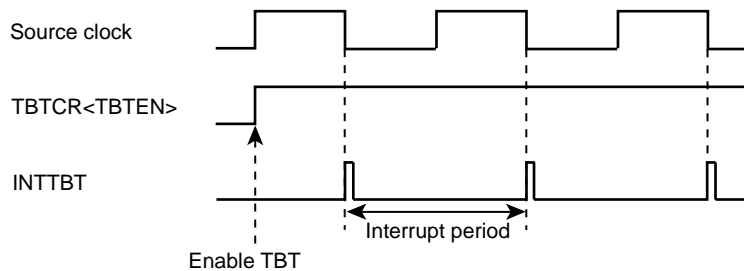


Figure 7-2 Time Base Timer Interrupt



## 7.2 Divider Output ( $\overline{DVO}$ )

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

### 7.2.1 Configuration

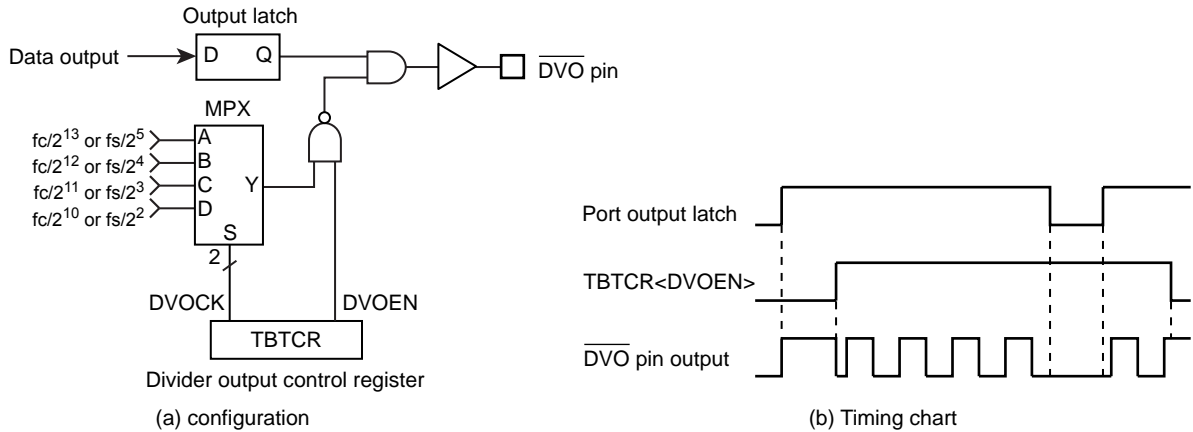
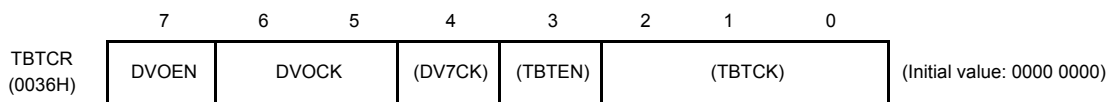


Figure 7-3 Divider Output

### 7.2.2 Control

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

#### Time Base Timer Control Register



DVOEN	Divider output enable / disable	0: Disable 1: Enable			R/W	
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)

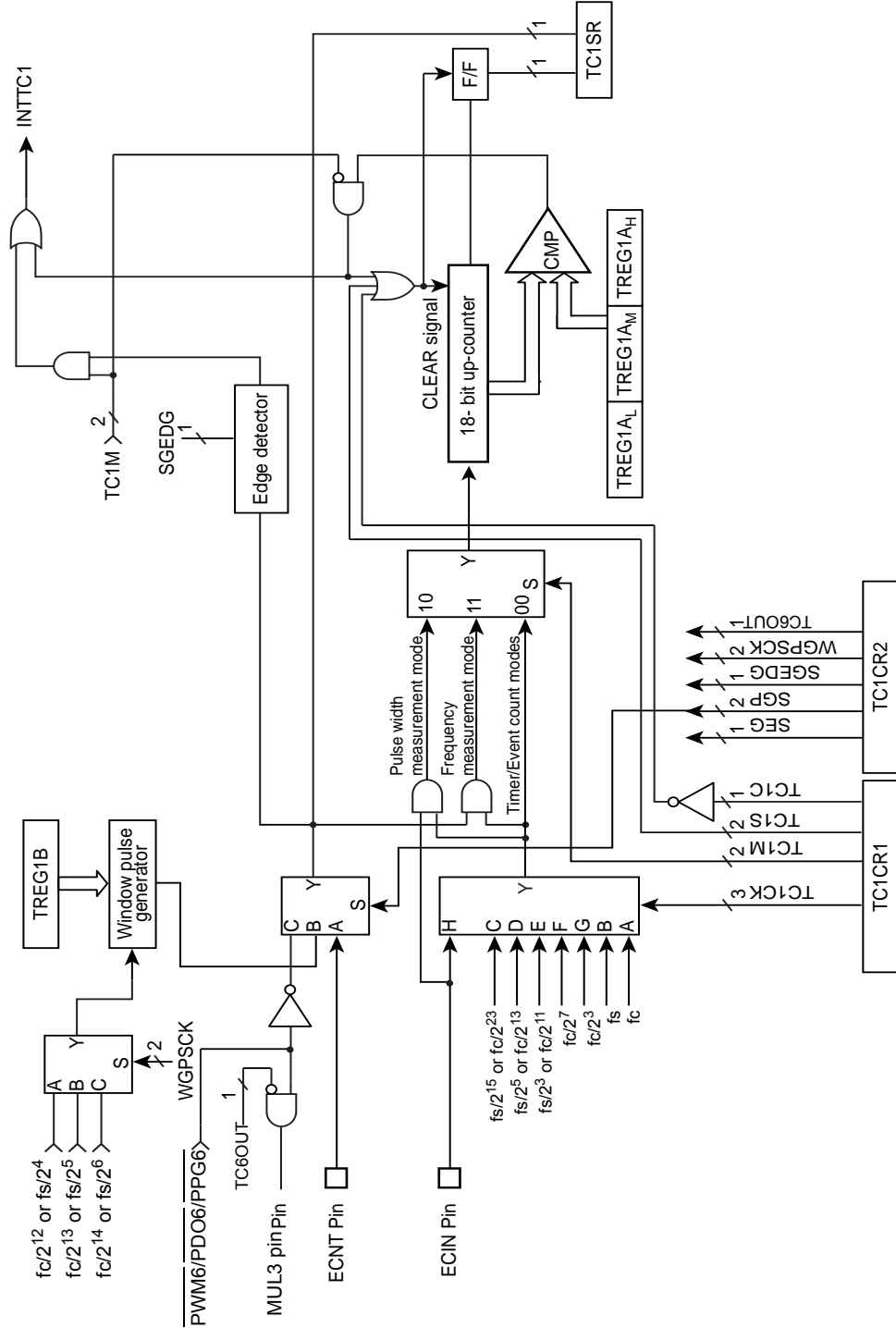
```
LD      (TBTCR), 00000000B      ; DVOCK ← "00"
LD      (TBTCR), 10000000B      ; DVOEN ← "1"
```

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

DVOCK	Divider Output Frequency [Hz]		
	NORMAL1/2, IDLE1/2 Mode		SLOW1/2, SLEEP1/2 Mode
	DV7CK = 0	DV7CK = 1	
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

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

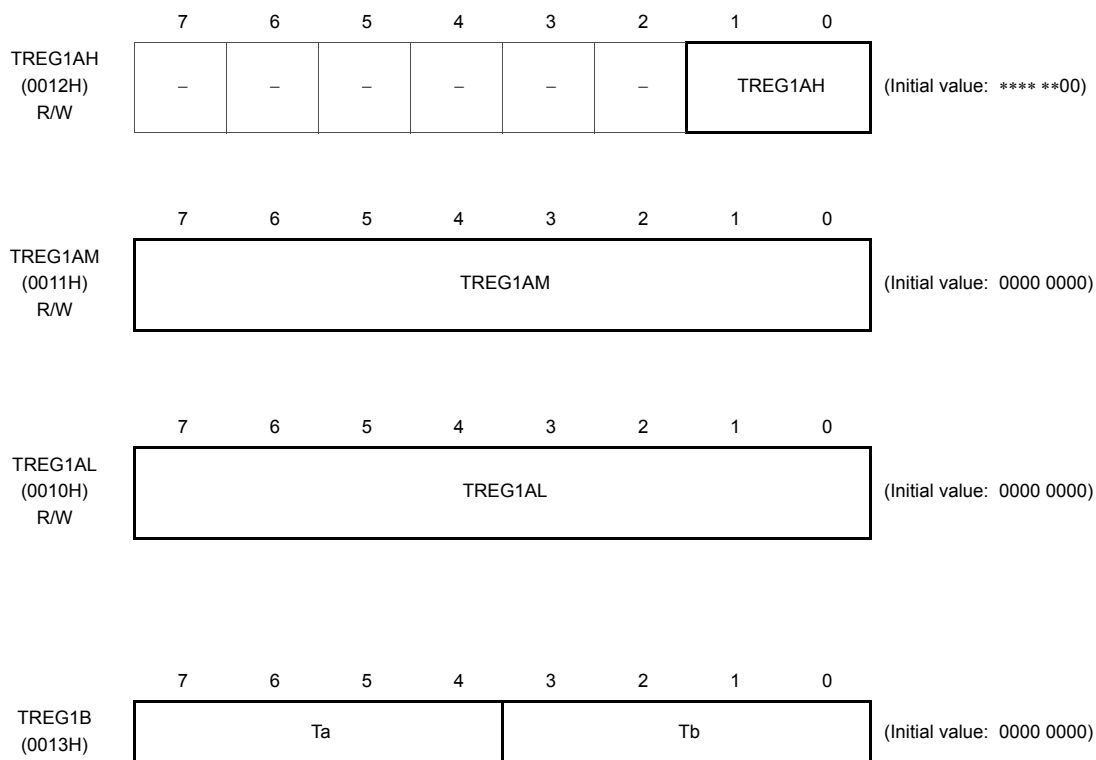
## 8.1 Configuration



## 8.2 Control

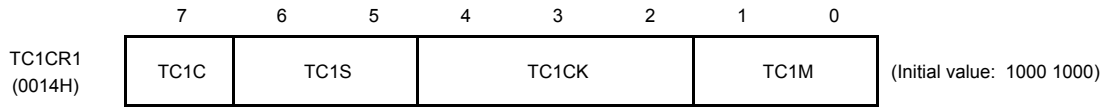
The Timer/counter 1 is controlled by timer/counter 1 control registers (TC1CR1/TC1CR2), an 18-bit timer register (TREG1A), and an 8-bit internal window gate pulse setting register (TREG1B).

### 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	$(16 - Ta) \times 2^{12}/f_c$	$(16 - Ta) \times 2^4/f_s$	$(16 - Ta) \times 2^4/f_s$	R/W
		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	$(16 - Tb) \times 2^{12}/f_c$	$(16 - Tb) \times 2^4/f_s$	$(16 - Tb) \times 2^4/f_s$	R/W
		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 <sup>23</sup>	fs/2 <sup>15</sup>	fs/2 <sup>15</sup>	fs/2 <sup>15</sup>	
		011:	fc/2 <sup>13</sup>	fs/2 <sup>5</sup>	fs/2 <sup>5</sup>	fs/2 <sup>5</sup>	
		100:	fc/2 <sup>11</sup>	fs/2 <sup>3</sup>	fs/2 <sup>3</sup>	fs/2 <sup>3</sup>	
		101:	fc/2 <sup>7</sup>	fc/2 <sup>7</sup>	-	-	
110:	fc/2 <sup>3</sup>	fc/2 <sup>3</sup>	-	-			
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 ≥1.

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

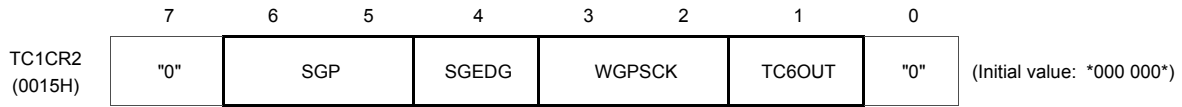
Note 8: When using the event counter mode, set TC1CK (TC1 source clock select) to external clock.

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<sup>7</sup>, fc/2<sup>3</sup> 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



SGP	Window gate pulse select	00: ECNT input 01: Internal window gate pulse (TREG1B) 10: $\overline{PWM6/PDO6/PPG6}$ (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
			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 MUL3 pin 1: No output to MUL3 pin				R/W

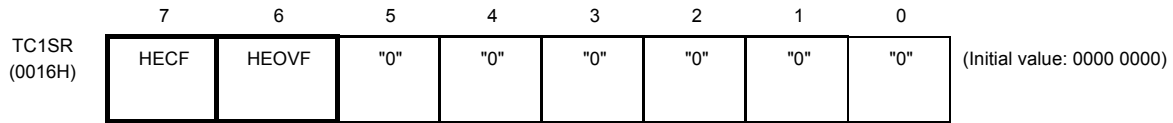
Note 1: fc; High-frequency clock [Hz] fs; Low-frequency clock [Hz] \*; Don't care

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 "0,7" to bit 0 in TC1CR2.

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 <sup>23</sup> [Hz]	fs/2 <sup>15</sup> [Hz]	fs/2 <sup>15</sup> [Hz]	fs/2 <sup>15</sup> [Hz]	0.52 s	1 s	38.2 h	72.8 h
fc/2 <sup>13</sup>	fs/2 <sup>5</sup>	fs/2 <sup>5</sup>	fs/2 <sup>5</sup>	512 ms	0.98 ms	2.2 min	4.3 min
fc/2 <sup>11</sup>	fs/2 <sup>3</sup>	fs/2 <sup>3</sup>	fs/2 <sup>3</sup>	128 ms	244 ms	0.6 min	1.07 min
fc/2 <sup>7</sup>	fc/2 <sup>7</sup>	-	-	8 ms	-	2.1 s	-
fc/2 <sup>3</sup>	fc/2 <sup>3</sup>	-	-	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.

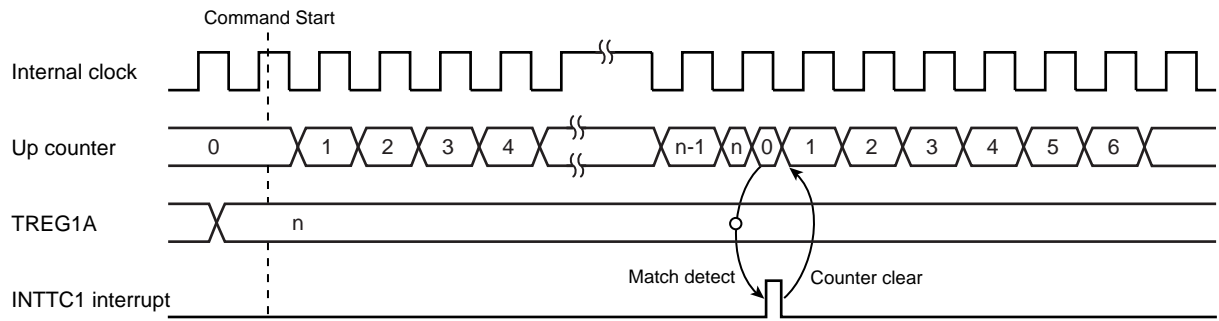


Figure 8-1 Timing chart for timer mode

### 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.

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.

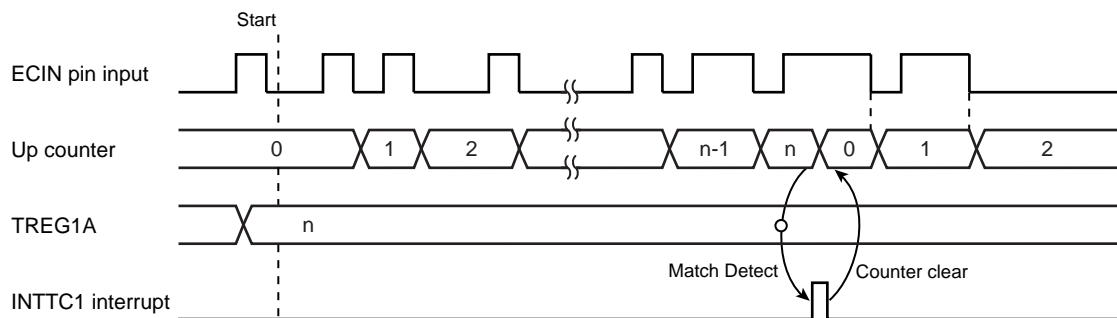


Figure 8-2 Event counter mode timing chart



### 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

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          (EIRH). 0    ; Clear bit0 of EIRH
    LD           (TC1CR1), 00011010B ; Stop timer couer 1
    LD           (ILH), 11111110B   ; Clear bit0 of ILH
    SET          (EIRH). 0    ; Set bit0 of EIRH
    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".

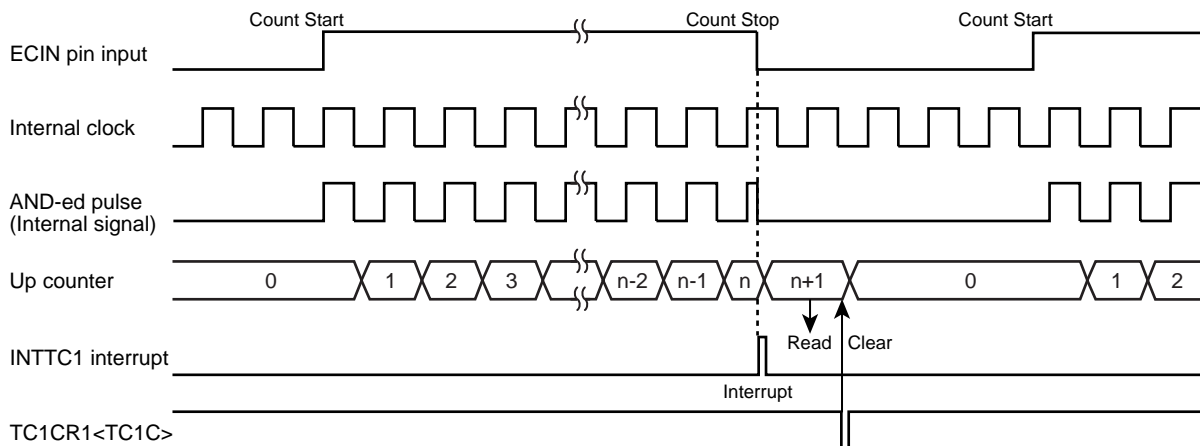


Figure 8-3 Pulse width measurement mode timing chart

### 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 MUL3 pin can be controlled using TC1CR2<TC6OUT>. Zero-clearing TC1CR2<TC6OUT> outputs  $\overline{\text{PWM6/PDO6/PPG6}}$  to MUL3 pin; setting 1 in TC1CR2<TC6OUT> does not output  $\overline{\text{PWM6/PDO6/PPG6}}$  to MUL3 pin. (TC1CR2<TC6OUT> is used to control output to MUL3 pin 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: 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

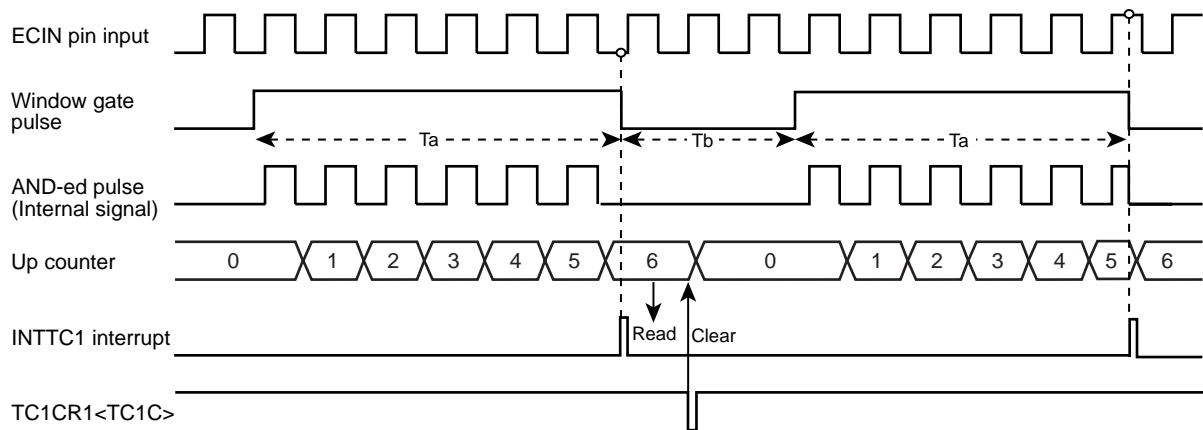


Figure 8-4 Timing chart for the frequency measurement mode (Window gate pulse falling interrupt)



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

## 9.1 Configuration

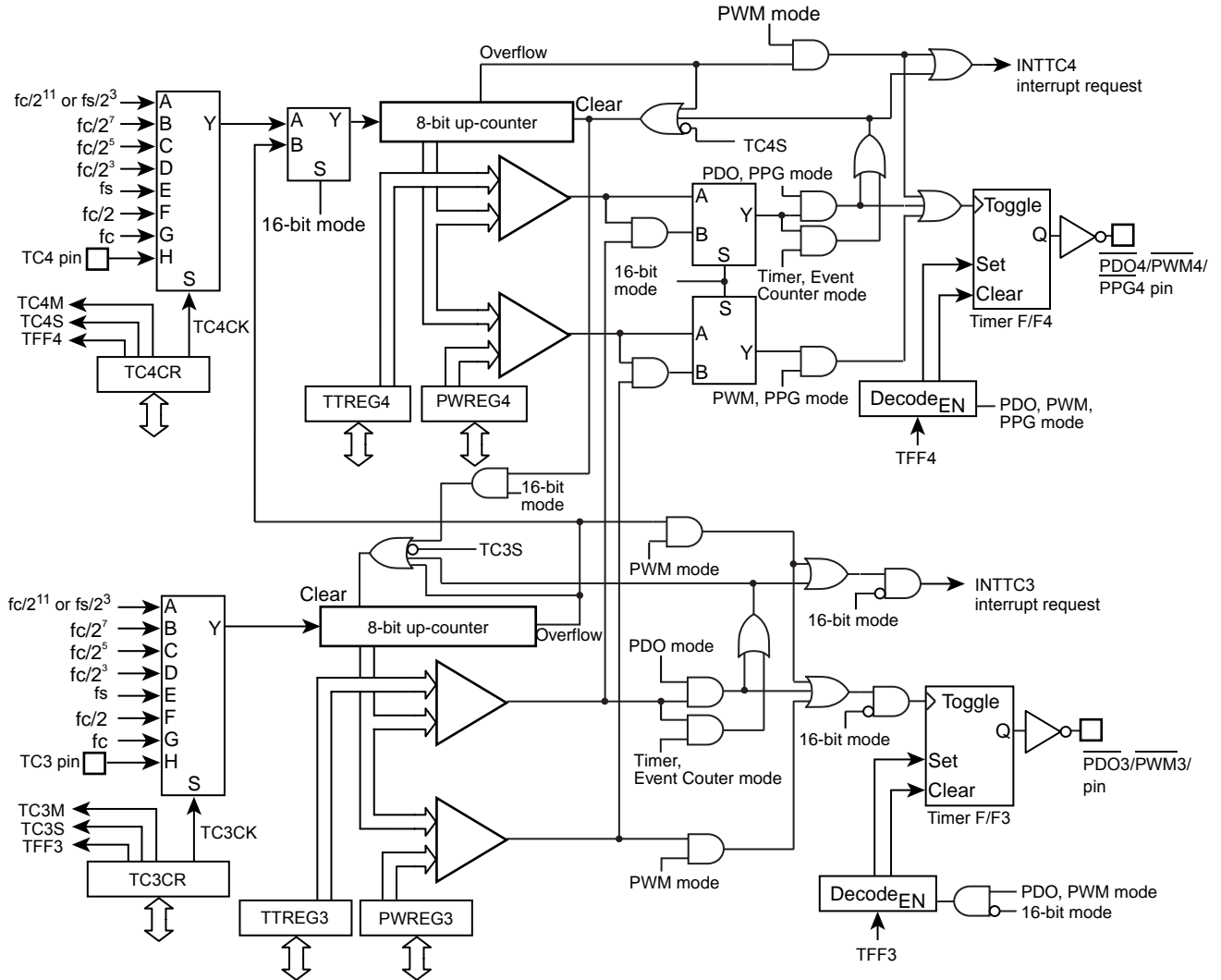
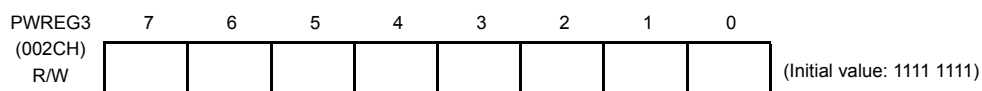
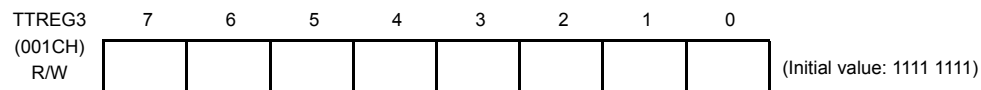


Figure 9-1 8-Bit TimerCounter 3, 4

## 9.2 TimerCounter Control

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

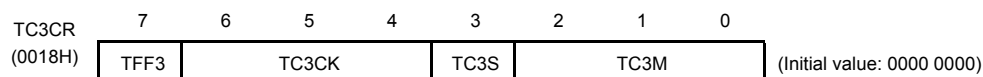
### TimerCounter 3 Timer Register



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.

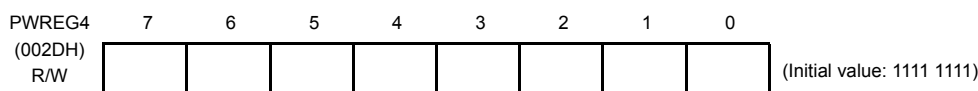
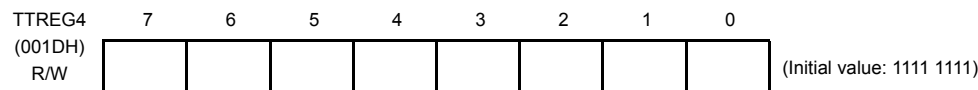
Note 7: The timer register settings are limited depending on the timer operating mode. For the detailed descriptions, see Table 9-3.

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).

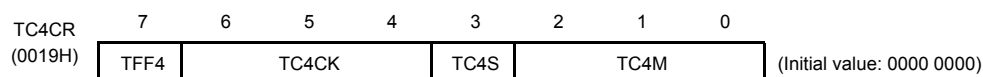
TimerCounter 4 Timer Register



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 TC $j$ CR<TFF $j$ > to 0. If not fixed, the  $\overline{PDO_j}$ ,  $\overline{PWM_j}$  and  $\overline{PPG_j}$  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 ( $80 \mu\text{s} \div 2^7 / f_c = 0AH$ ).
DI
SET     (EIRH). 3         : 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.
```

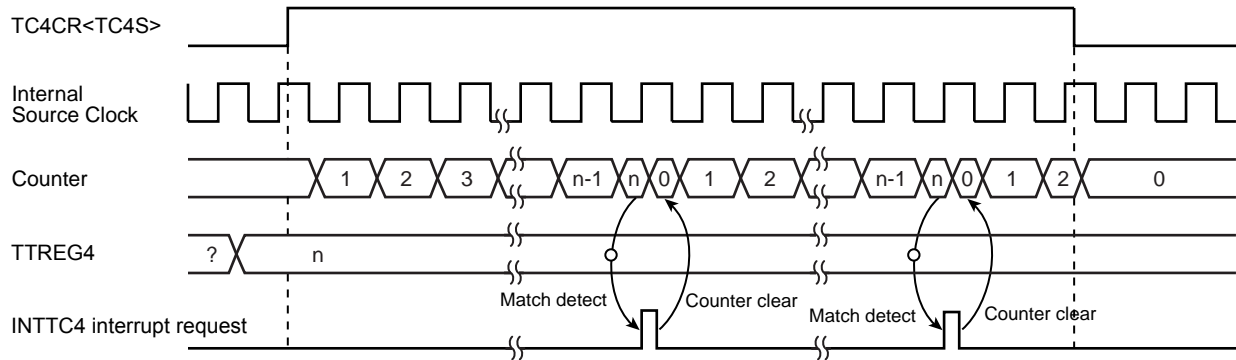


Figure 9-2 8-Bit Timer Mode Timing Chart (TC4)

### 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

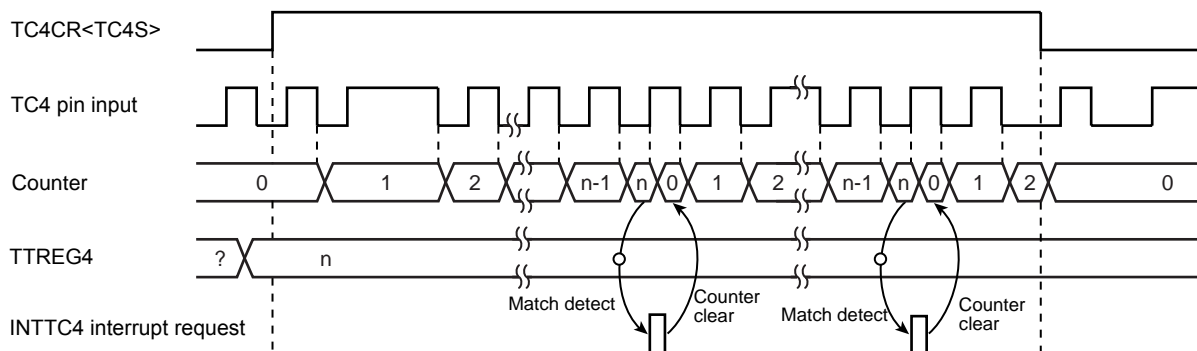


Figure 9-3 8-Bit Event Counter Mode Timing Chart (TC4)

### 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

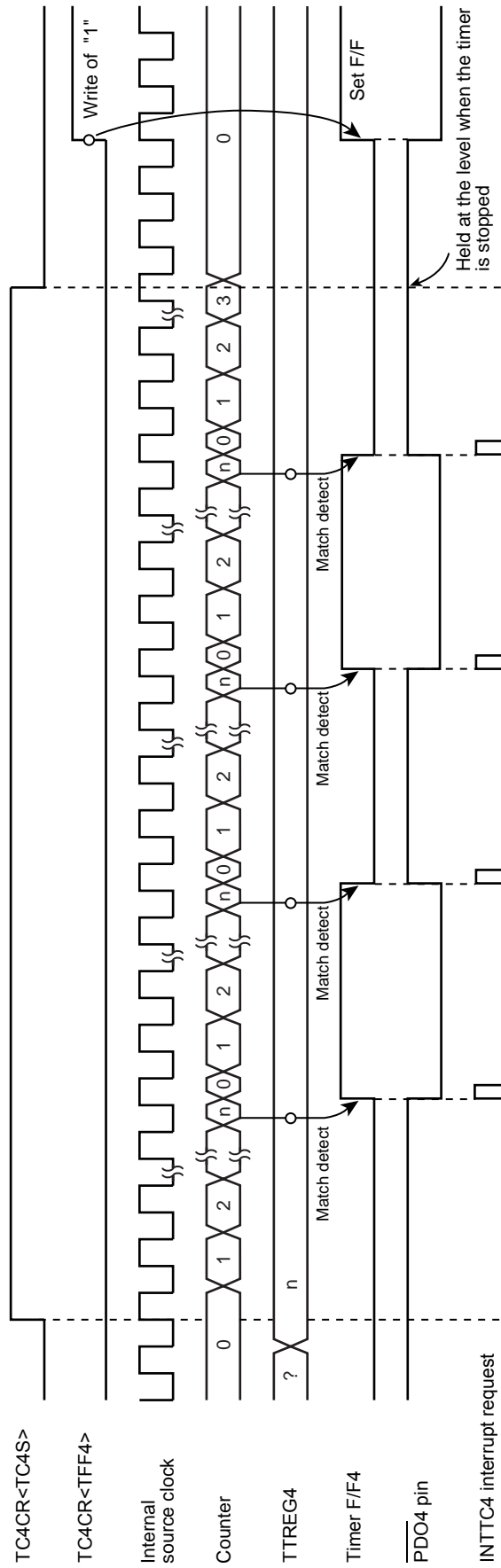


Figure 9-4 8-Bit PDO Mode Timing Chart (TC4)

### 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  $\overline{\text{PWMj}}$  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 = 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	–

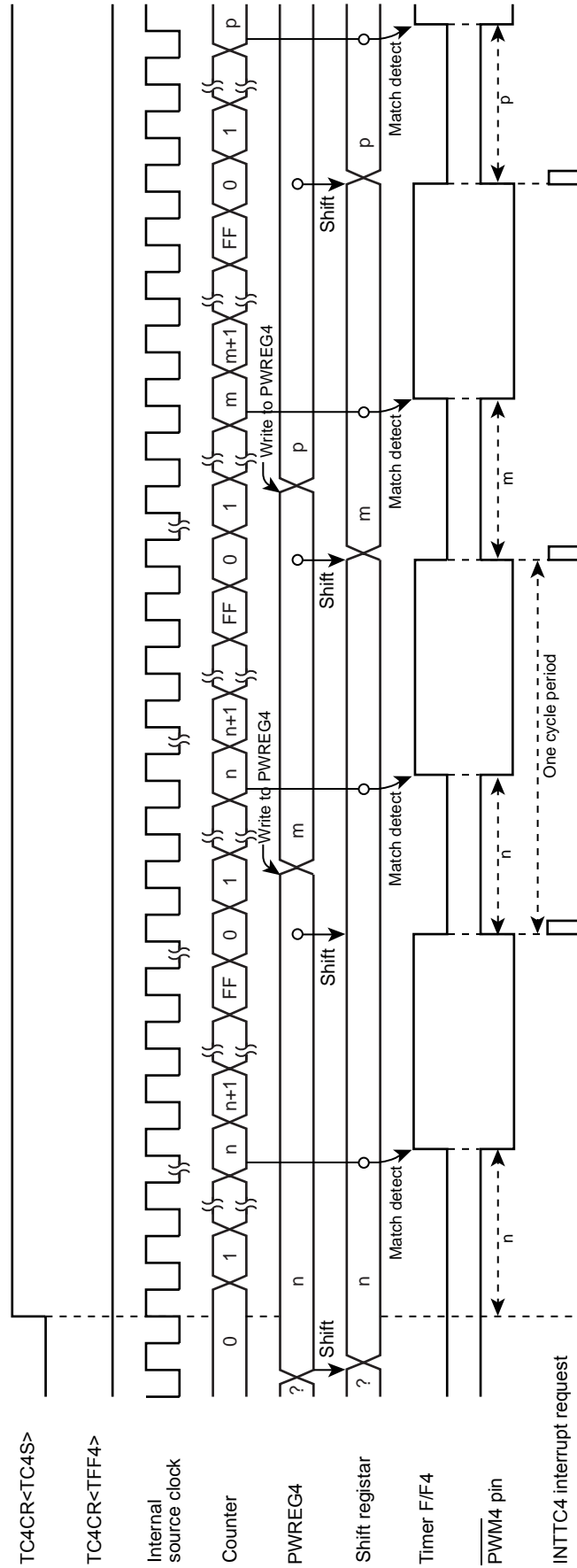


Figure 9-5 8-Bit PWM Mode Timing Chart (TC4)



### 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  $\overline{P\bar{D}Oj}$ ,  $\overline{P\bar{W}Mj}$ , and  $\overline{P\bar{P}Gj}$  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), 3 : 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.

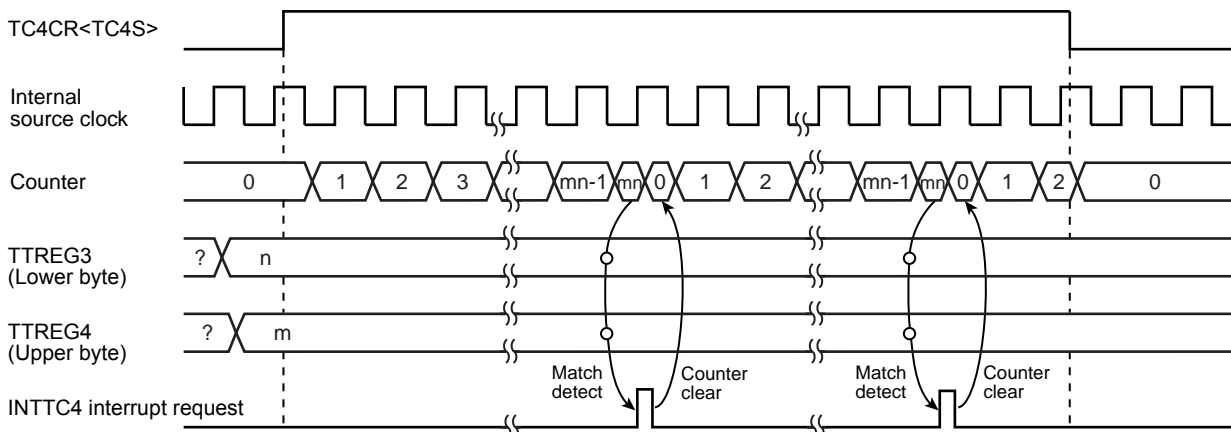


Figure 9-6 16-Bit Timer Mode Timing Chart (TC3 and TC4)

### 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	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 (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.

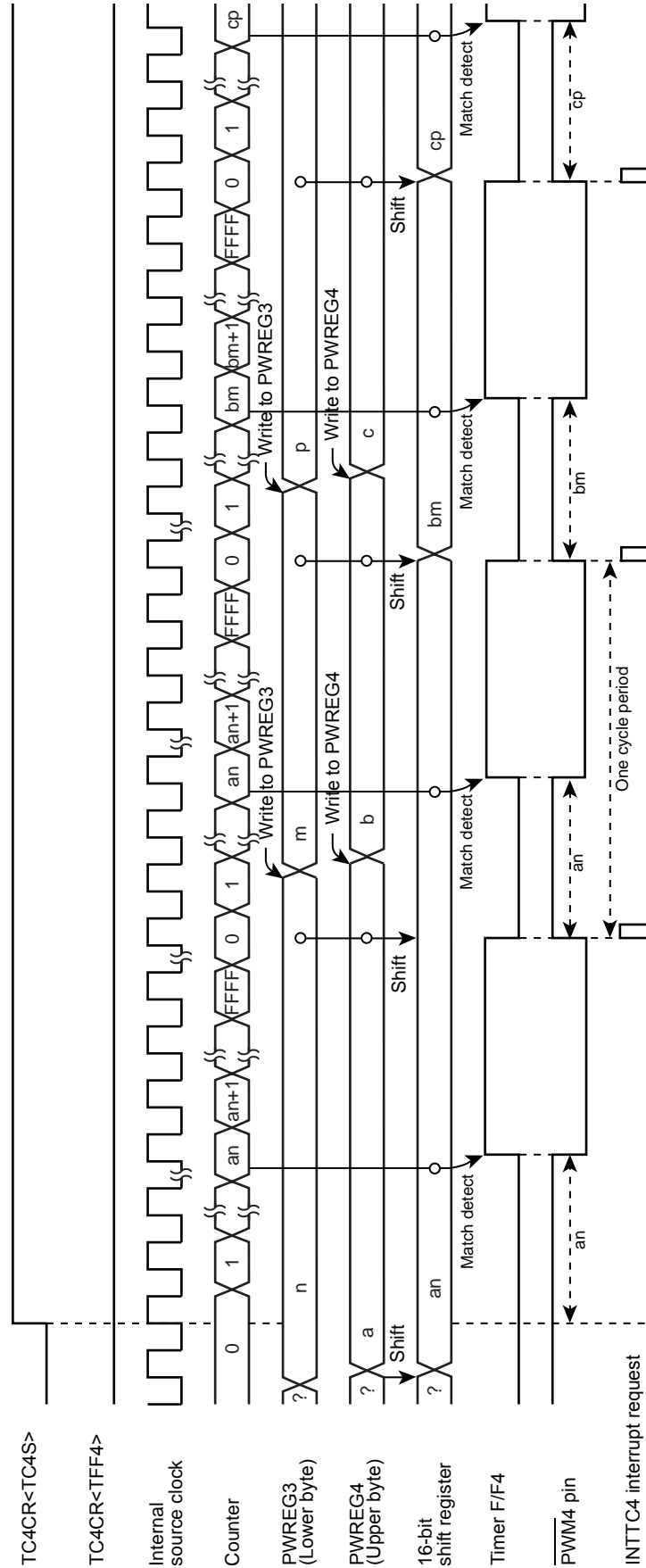


Figure 9-7 16-Bit PWM Mode Timing Chart (TC3 and TC4)

### 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

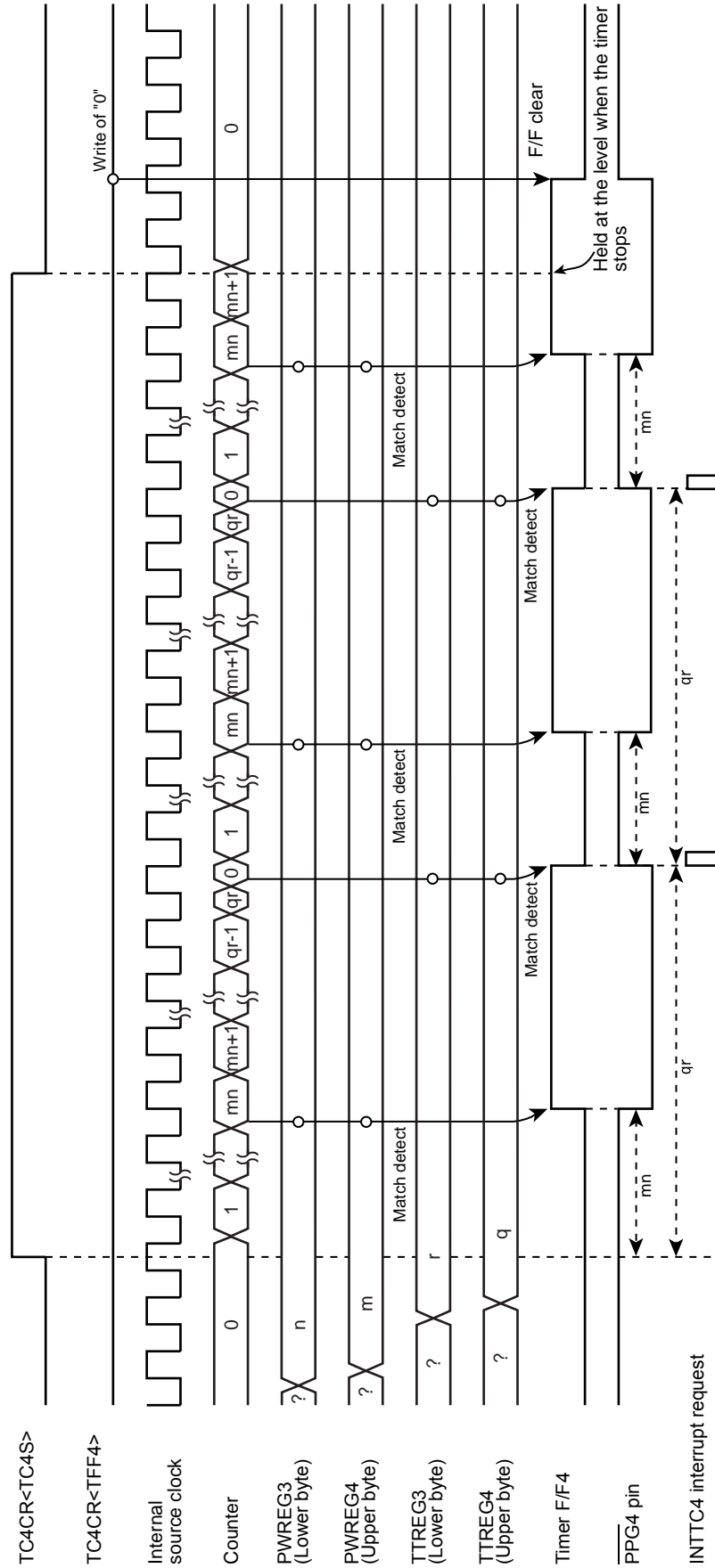


Figure 9-8 16-Bit PPG Mode Timing Chart (TC3 and TC40)

### 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). 3      : 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). 3      : 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

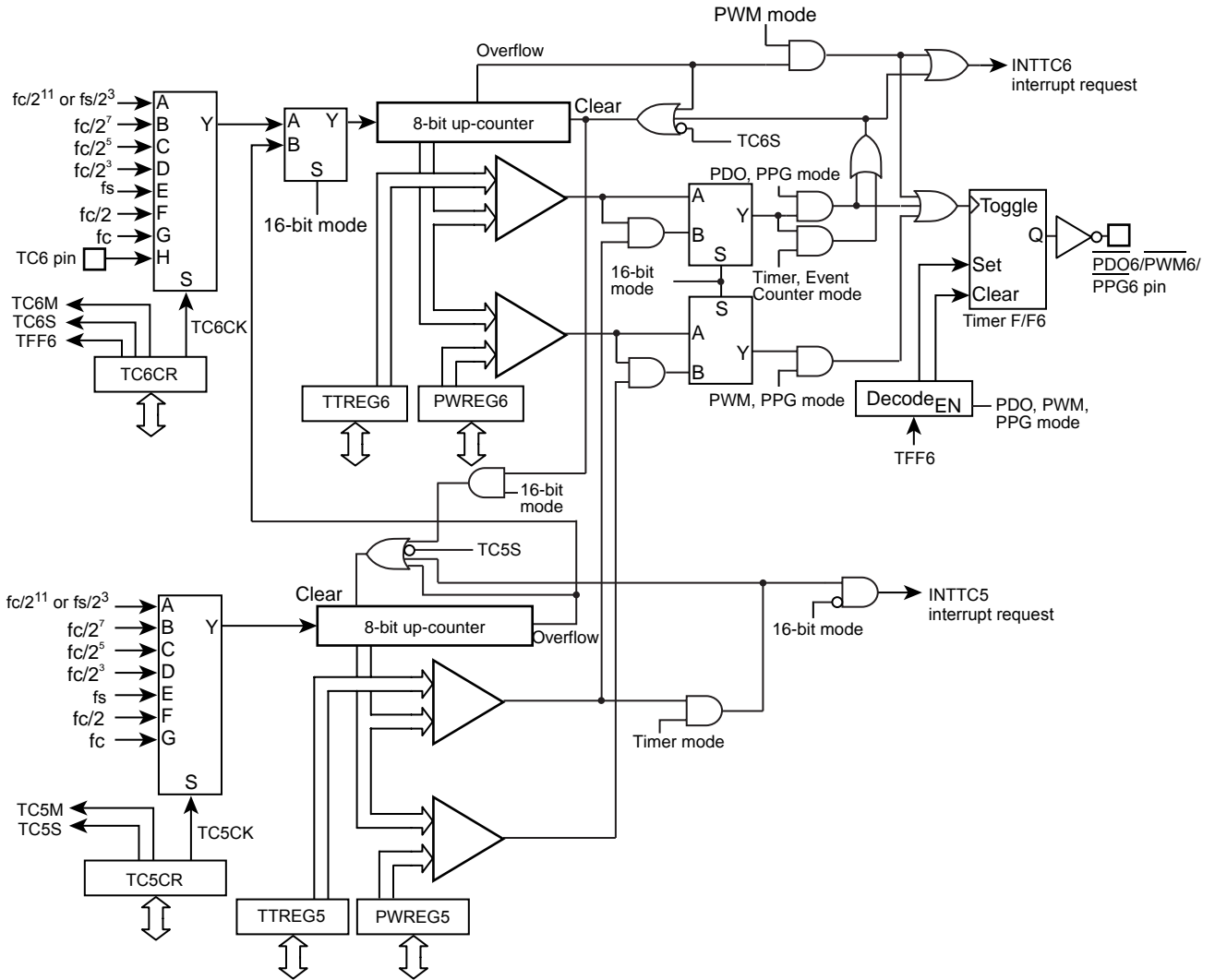
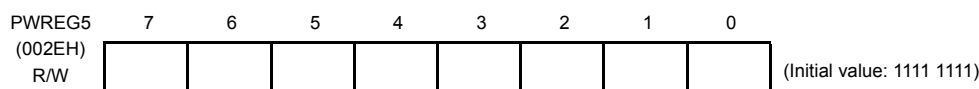
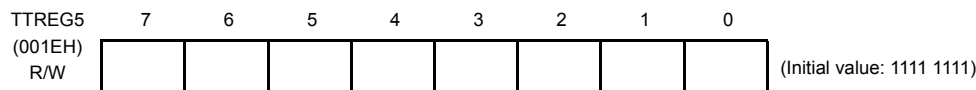


Figure 10-1 8-Bit TimerCounter 5, 6

## 10.2 TimerCounter Control

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

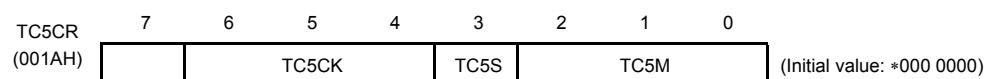
### TimerCounter 5 Timer Register



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



			NORMAL1/2, IDLE1/2 mode		SLOW1/2 SLEEP1/2 mode	
			DV7CK = 0	DV7CK = 1		
TC5CK	Operating clock selection [Hz]	000	$fc/2^{11}$	$fs/2^3$	$fs/2^3$	R/W
		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	Reserved			
TC5S	TC5 start control	0: Operation stop and counter clear 1: Operation start				R/W
TC5M	TC5M operating mode select	000: 8-bit timer 001: Reserved 010: Reserved 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 and TC5CK settings. To start the timer operation (TC5S= 0 → 1), TC5M and TC5CK 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.

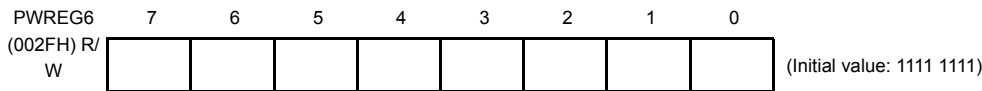
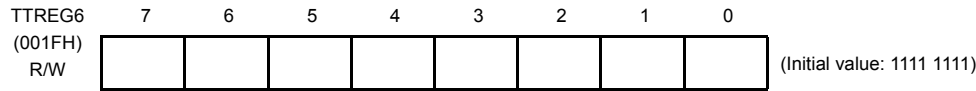
Note 6: The operating clock settings are limited depending on the timer operating mode. For the detailed descriptions, see Table 10-1 and Table 10-2.

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 fc 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).

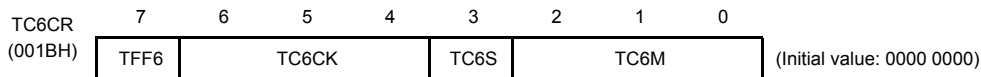
### TimerCounter 6 Timer Register



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	$fc/2^{11}$ or $fs/2^3$	$fc/2^7$	$fc/2^5$	$fc/2^3$	fs	$fc/2$	fc	TC5 pin input	TC6 pin input
8-bit timer	0	0	0	0	–	–	–	–	–
8-bit event counter	–	–	–	–	–	–	–	–	0
8-bit PDO	0	0	0	0	–	–	–	–	–
8-bit PWM	0	0	0	0	0	0	0	–	–
16-bit timer	0	0	0	0	–	–	–	–	–
Warm-up counter	–	–	–	–	0	–	–	–	–
16-bit PWM	0	0	0	0	0	0	0	–	–
16-bit PPG	0	0	0	0	–	–	–	–	–

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

Note 2: 0 : Available source clock

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

Operating mode	$fc/2^{11}$ or $fs/2^3$	$fc/2^7$	$fc/2^5$	$fc/2^3$	fs	$fc/2$	fc	TC5 pin input	TC6 pin input
8-bit timer	0	–	–	–	–	–	–	–	–
8-bit event counter	–	–	–	–	–	–	–	–	0
8-bit PDO	0	–	–	–	–	–	–	–	–
8-bit PWM	0	–	–	–	0	–	–	–	–
16-bit timer	0	–	–	–	–	–	–	–	–
Warm-up counter	–	–	–	–	–	–	0	–	–
16-bit PWM	0	–	–	–	0	–	–	–	–
16-bit PPG	0	–	–	–	–	–	–	–	–

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

Note2: 0 : Available source clock

Table 10-3 Constraints on Register Values Being Compared

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

Note: n = 5 to 6

## 10.3 Function

The TimerCounter 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  $j$  (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:  $j = 5, 6$

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			$f_c = 16 \text{ MHz}$	$f_s = 32.768 \text{ kHz}$	$f_c = 16 \text{ MHz}$	$f_s = 32.768 \text{ kHz}$
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 (TimerCounter6,  $f_c = 16.0 \text{ MHz}$ )

```
LD      (TTREG6), 0AH      : Sets the timer register ( $80 \mu\text{s} \div 2^7 / f_c = 0AH$ ).
DI
SET     (EIRH). 4         : Enables INTTC6 interrupt.
EI
LD      (TC6CR), 0001000B  : Sets the operating clock to  $f_c/2^7$ , and 8-bit timer mode.
LD      (TC6CR), 00011000B : Starts TC6.
```

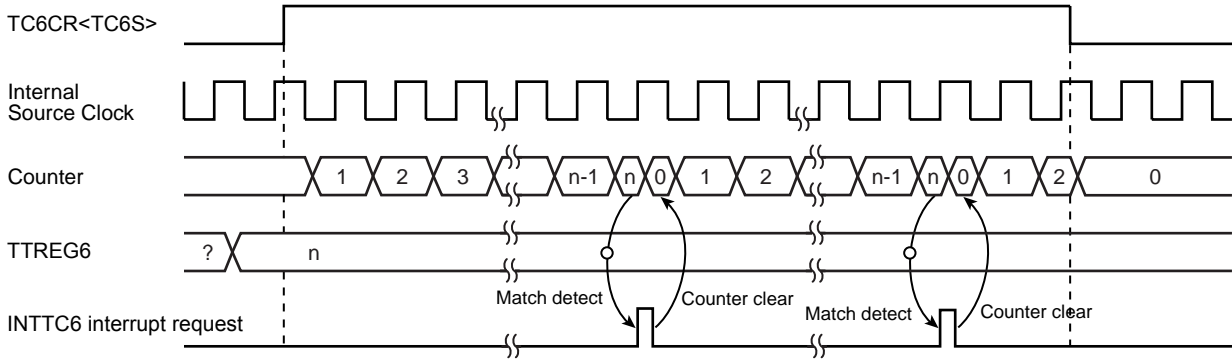


Figure 10-2 8-Bit Timer Mode Timing Chart (TC6)

### 10.3.2 8-Bit Event Counter Mode (TC6)

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 TTR6Gj 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 TTR6Gj setting while the timer is running. Since TTR6Gj is not in the shift register configuration in the event counter mode, the new value programmed in TTR6Gj is in effect immediately after the programming. Therefore, if TTR6Gi is changed while the timer is running, an expected operation may not be obtained.

Note 3: j = 6

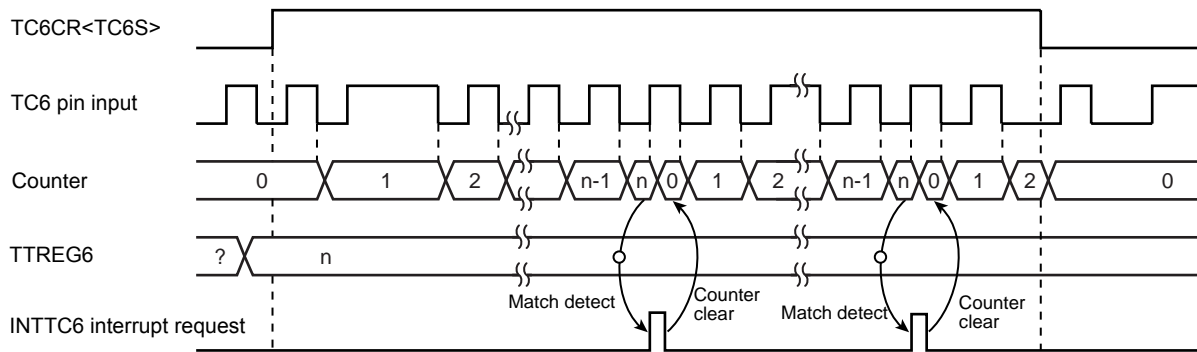


Figure 10-3 8-Bit Event Counter Mode Timing Chart (TC6)

### 10.3.3 8-Bit Programmable Divider Output (PDO) Mode (TC6)

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 TTR6Gj 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 ( $f_c = 16.0$  MHz)

Setting port		
LD	(TTREG6), 3DH	: $1/1024 \div 2^7 / f_c \div 2 = 3DH$
LD	(TC6CR), 00010001B	: Sets the operating clock to $f_c/2^7$ , 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 = 6



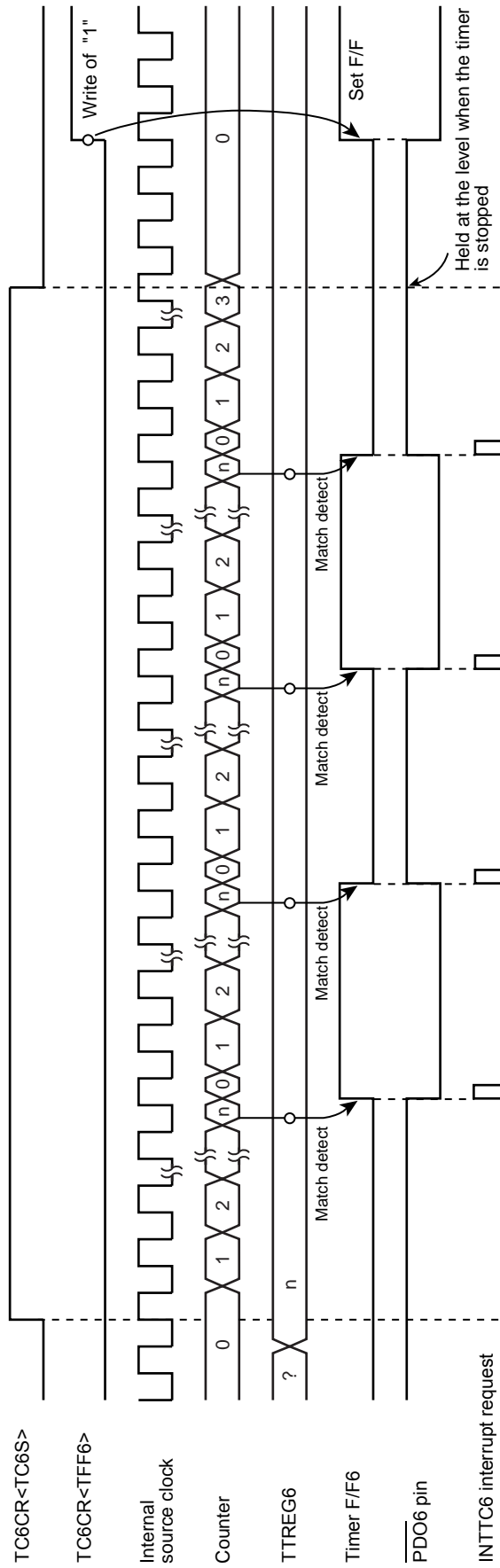


Figure 10-4 8-Bit PDO Mode Timing Chart (TC6)

### 10.3.4 8-Bit Pulse Width Modulation (PWM) Output Mode (TC6)

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 = 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	–

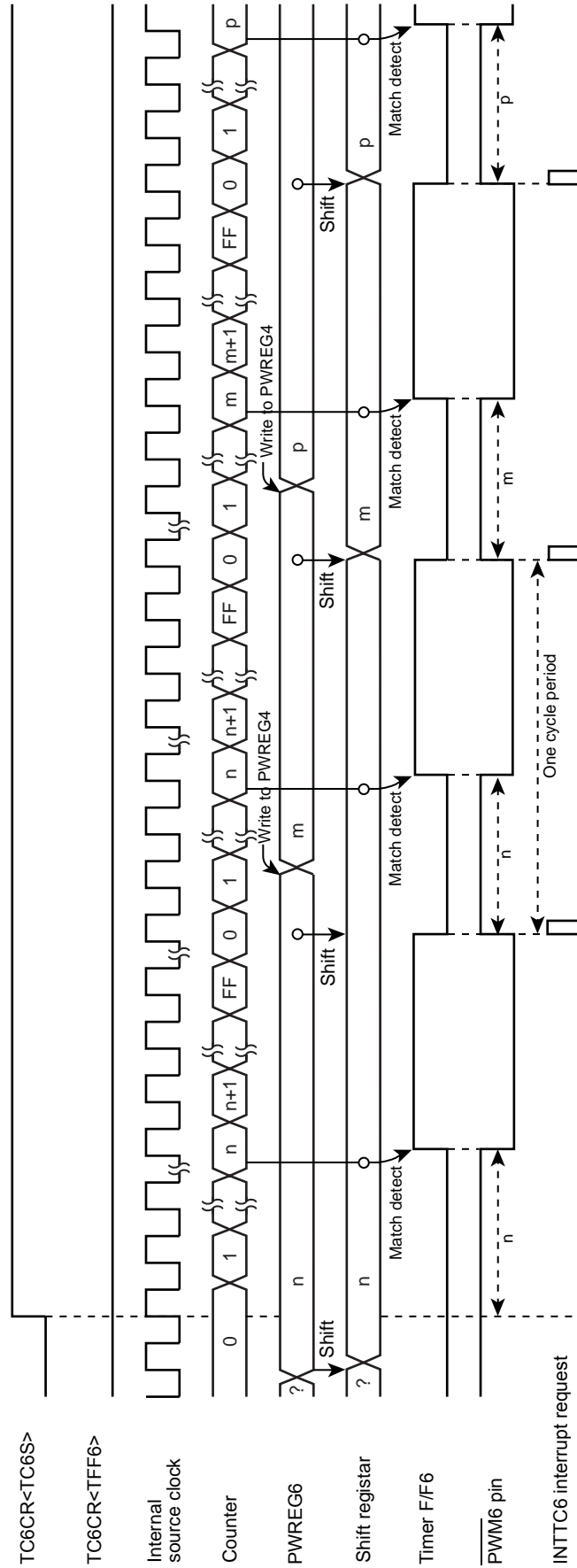


Figure 10-5 8-Bit PWM Mode Timing Chart (TC6)

### 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  $\overline{P\overline{D}O_j}$ ,  $\overline{P\overline{W}M_j}$ , and  $\overline{P\overline{P}G_j}$  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), 4 : 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.

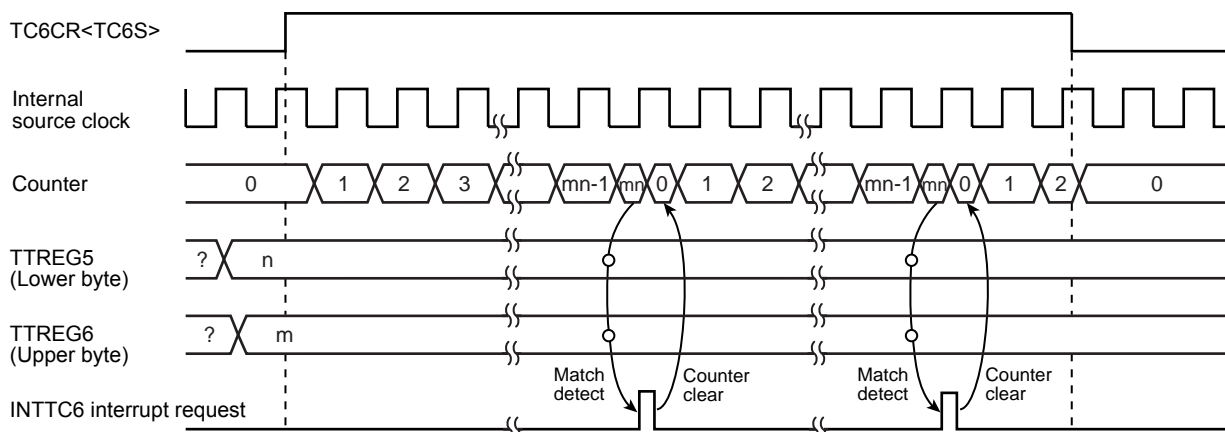


Figure 10-6 16-Bit Timer Mode Timing Chart (TC5 and TC6)

### 10.3.6 16-Bit 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.

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  $\overline{\text{PWM6}}$  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  $\overline{\text{PWM6}}$  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  $\overline{\text{PWM6}}$  pin to the high level when the TimerCounter is stopped

CLR (TC6CR).3: Stops the timer.

CLR (TC6CR).7 : Sets the  $\overline{\text{PWM6}}$  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  $\overline{\text{PWM6}}$  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	
DV7CK = 0	DV7CK = 1		$f_c = 16 \text{ MHz}$	$f_s = 32.768 \text{ kHz}$	$f_c = 16 \text{ MHz}$	$f_s = 32.768 \text{ kHz}$
$f_c/2^{11}$	$f_s/2^3$ [Hz]	$f_s/2^3$ [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 ( $f_c = 16.0$  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.

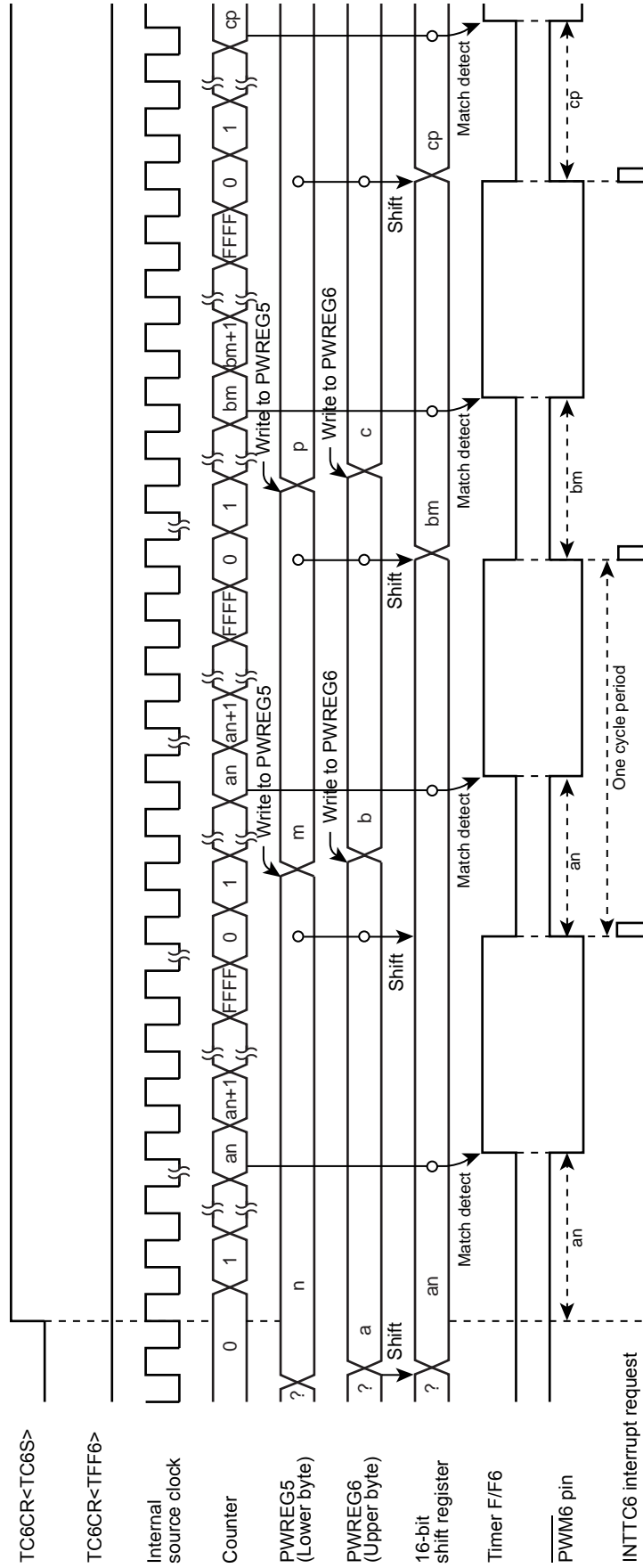


Figure 10-7 16-Bit PWM Mode Timing Chart (TC5 and TC6)

### 10.3.7 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. 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.

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  $\overline{\text{PPG6}}$  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 ( $f_c = 16.0$  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 $f_c/2^3$ , 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 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  $\overline{\text{PPG6}}$  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  $\overline{\text{PPG6}}$  pin to the high level when the TimerCounter is stopped

CLR (TC6CR).3: Stops the timer

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

Note 3: i = 5, 6



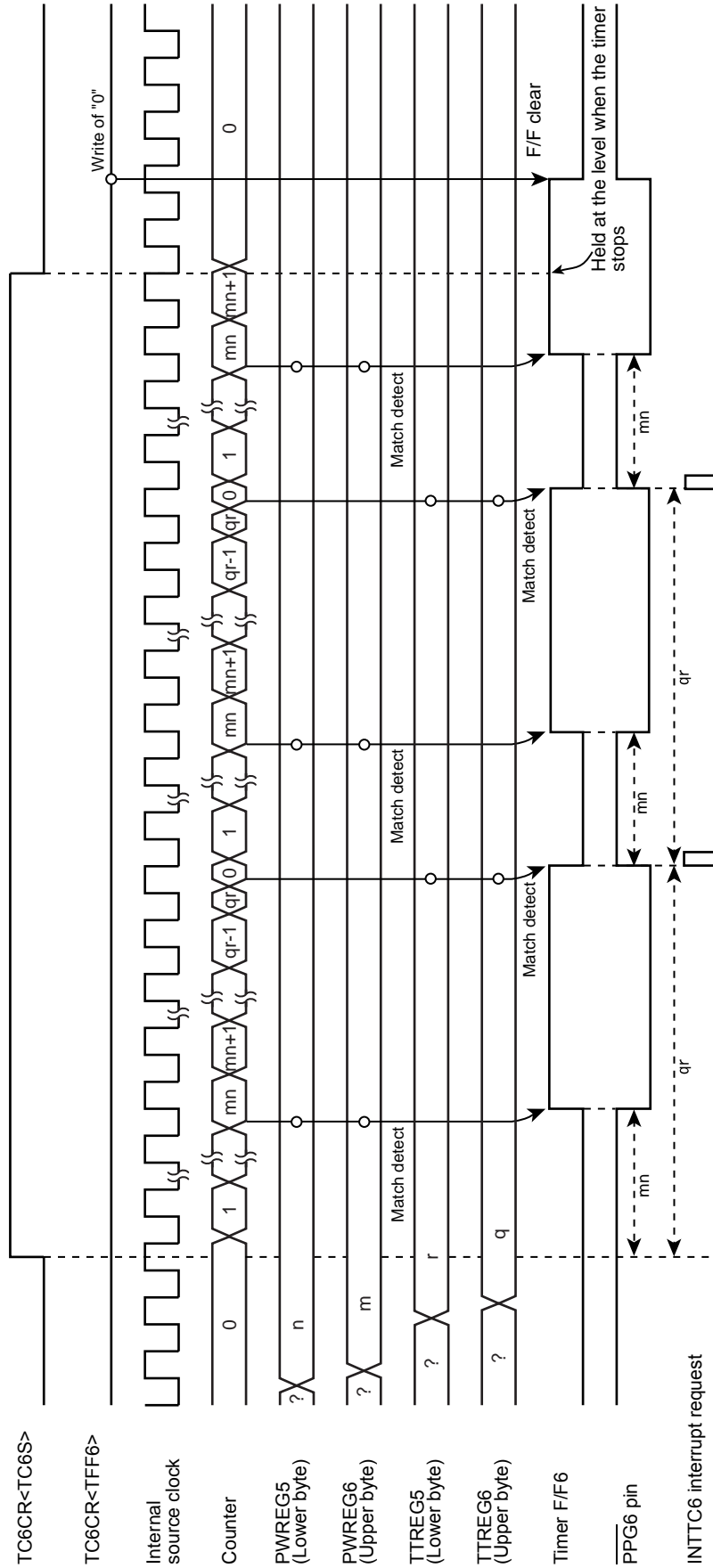


Figure 10-8 16-Bit PPG Mode Timing Chart (TC5 and TC60)

### 10.3.8 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

#### 10.3.8.1 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), 4       : 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
    
```

10.3.8.2 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 fs, 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). 4      : 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

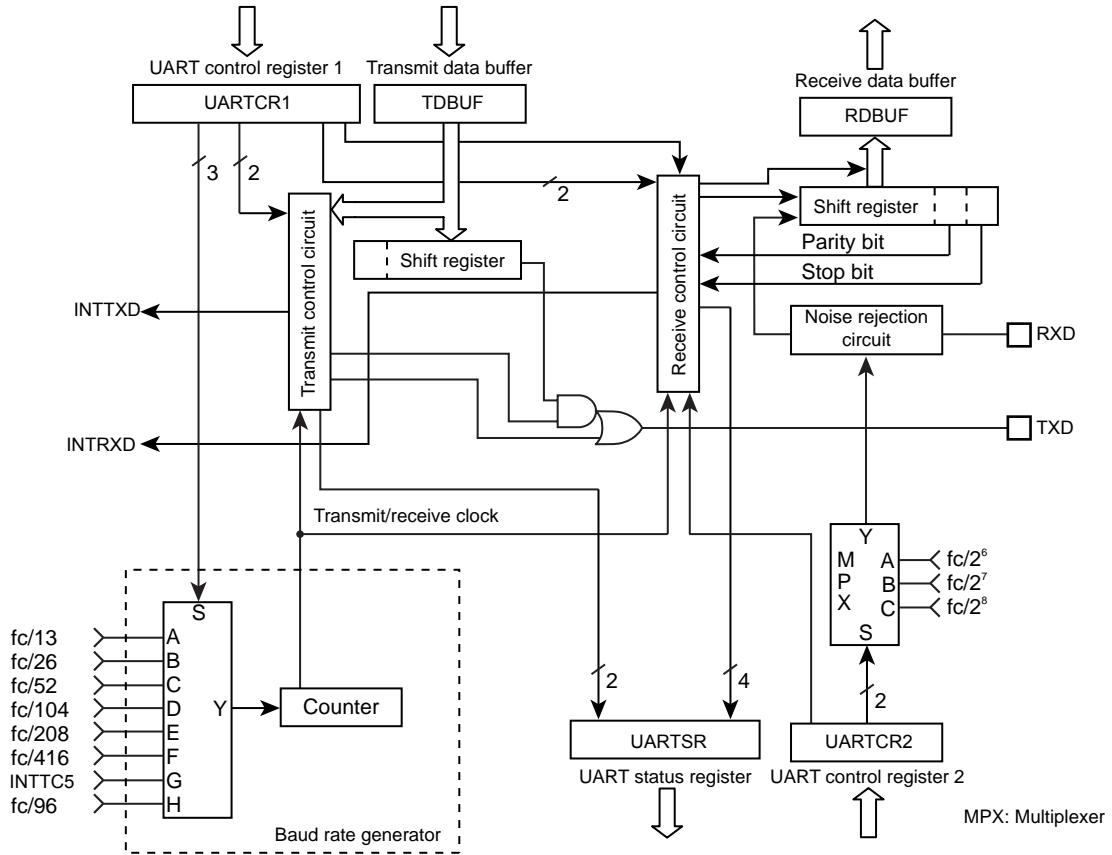


Figure 11-1 UART (Asynchronous Serial Interface)

## 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

UARTSR (0025H)	7	6	5	4	3	2	1	0	
	PERR	FERR	OERR	RBFL	TEND	TBEP			(Initial value: 0000 11**)

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

RDBUF (0F9BH)	7	6	5	4	3	2	1	0	Read only
									(Initial value: 0000 0000)

## UART Transmit Data Buffer

TDBUF (0F9BH)	7	6	5	4	3	2	1	0	Write only
									(Initial value: 0000 0000)

### 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

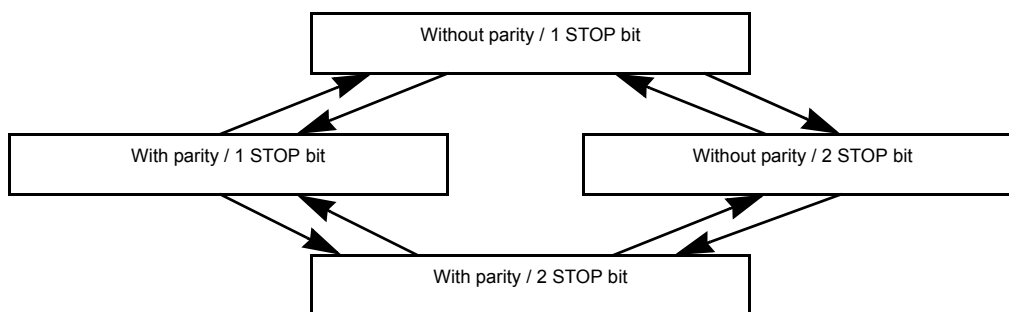


Figure 11-3 Caution on Changing Transfer Data Format

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).

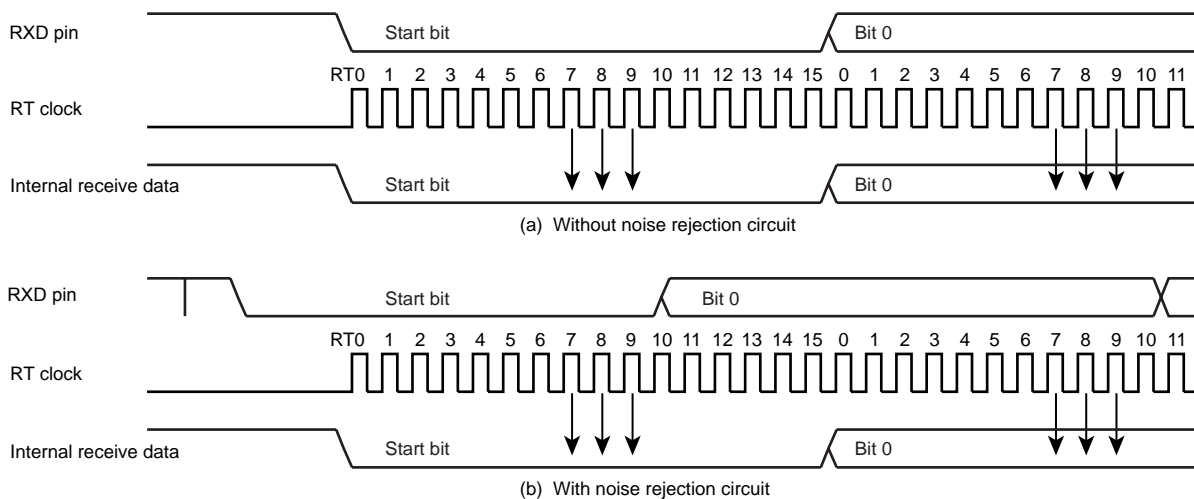


Figure 11-4 Data Sampling Method

---

## 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.

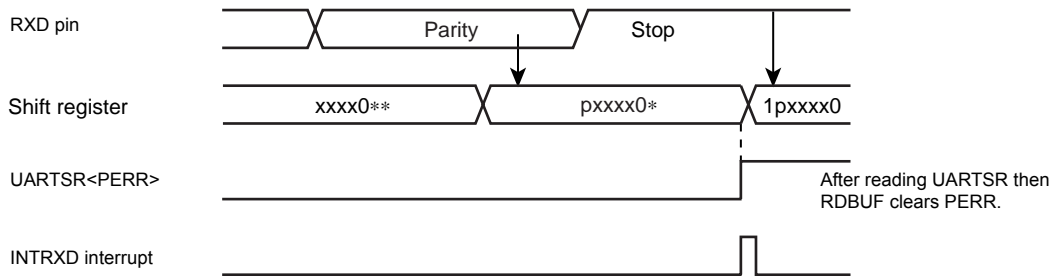


Figure 11-5 Generation of Parity Error

### 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.

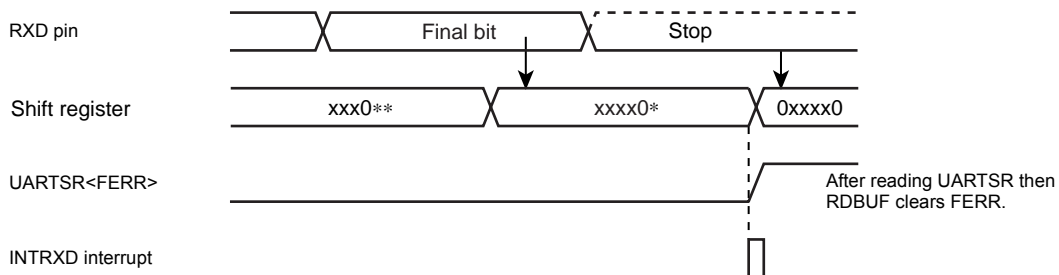


Figure 11-6 Generation of Framing Error

### 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.

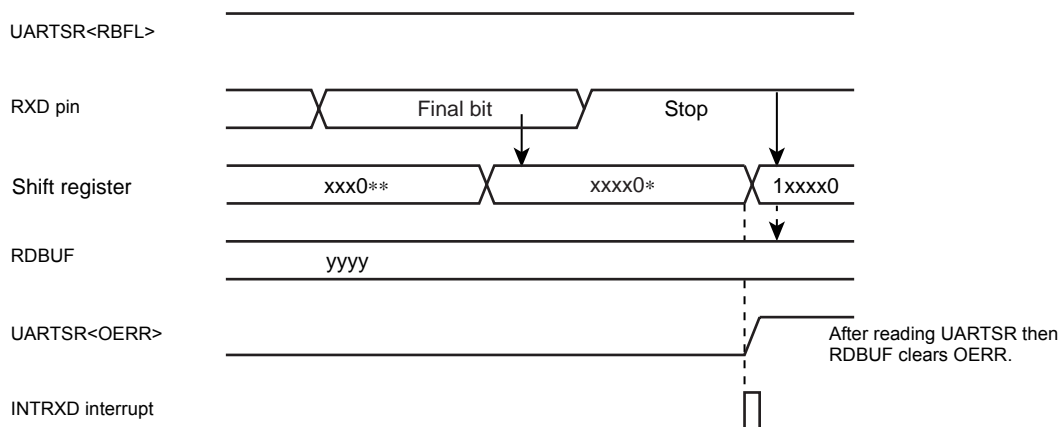


Figure 11-7 Generation of Overrun Error

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.

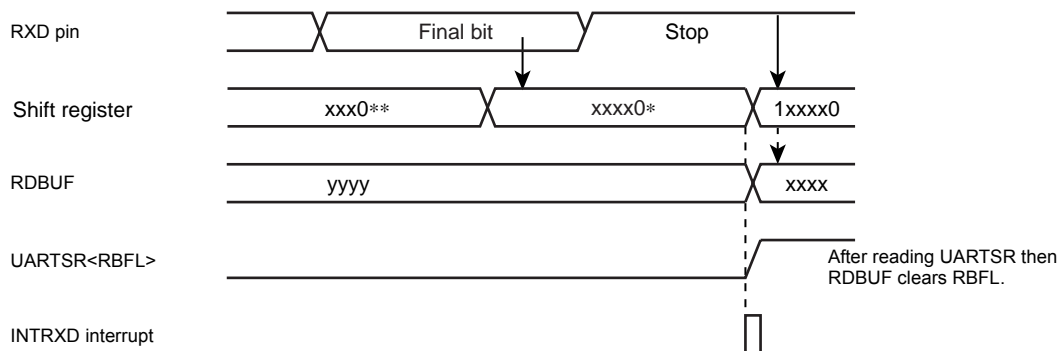


Figure 11-8 Generation of Receive Data Buffer Full

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.

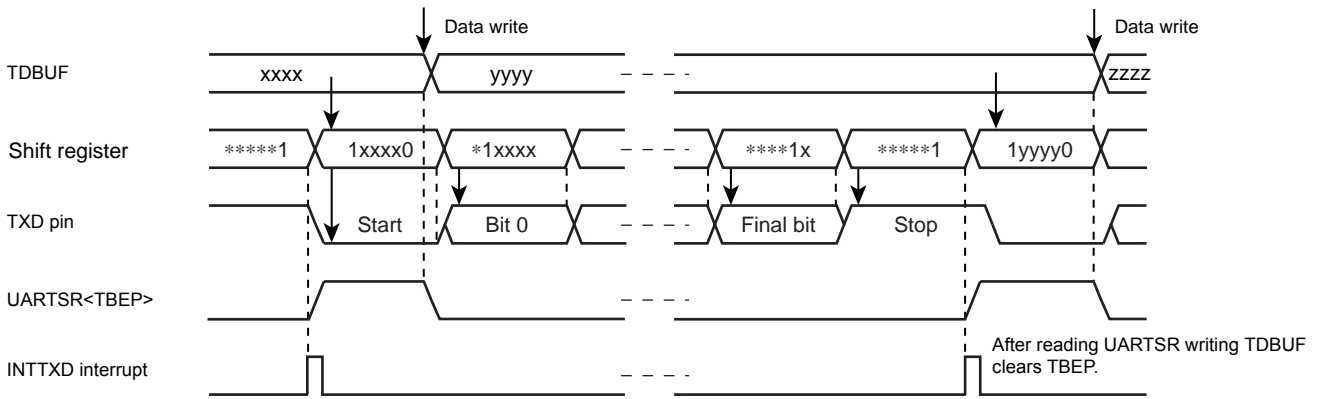


Figure 11-9 Generation of Transmit Data Buffer Empty

### 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.

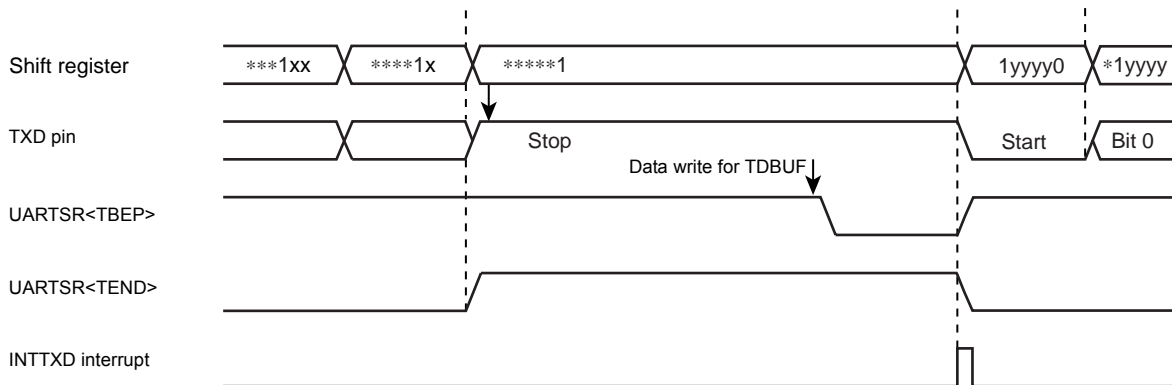


Figure 11-10 Generation of Transmit End Flag and Transmit Data Buffer Empty



## 12. Synchronous Serial Interface (SIO0)

The TMP86CS25AFG 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 SO0, SI0, SCK0 port.

### 12.1 Configuration

SIO control / status register

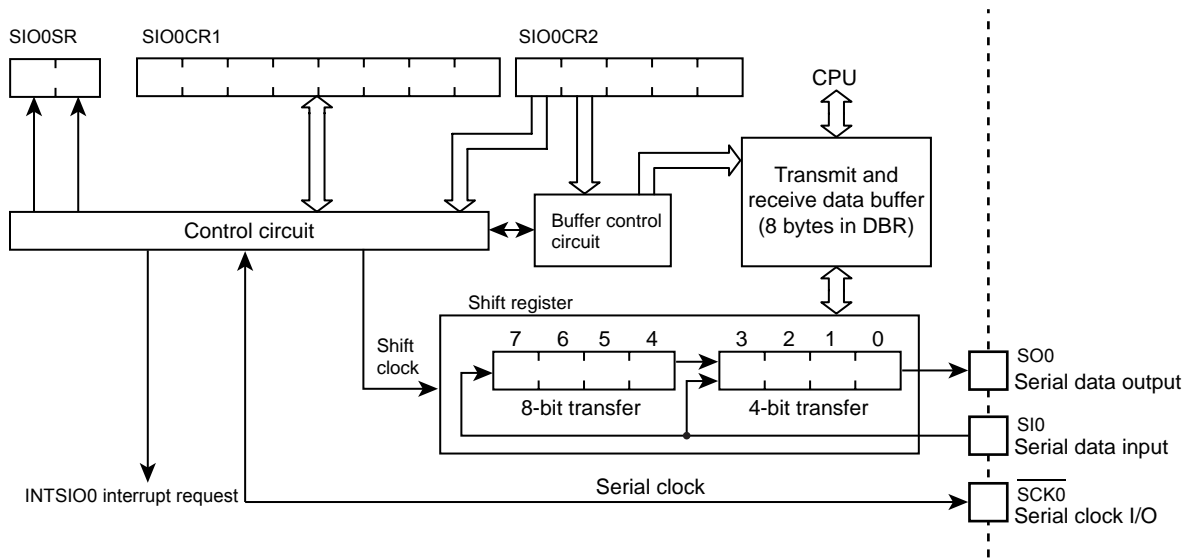


Figure 12-1 Serial Interface

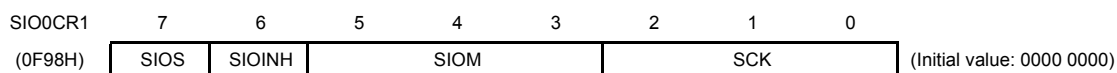
## 12.2 Control

The serial interface is controlled by SIO control registers (SIO0CR1/SIO0CR2). The serial interface status can be determined by reading SIO status register (SIO0SR).

The transmit and receive data buffer is controlled by the SIO0CR2<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 (INTSIO0) 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 SIO0CR2<WAIT>.

### SIO Control Register 1



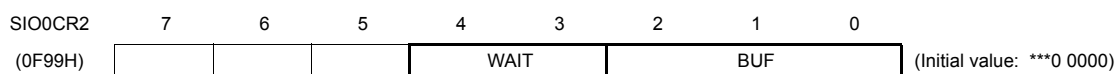
SIOS	Indicate transfer start / stop	0: Stop 1: Start	Write only		
SIOINH	Continue / abort transfer	0: Continuously transfer 1: Abort transfer (Automatically cleared after abort)			
SIOM	Transfer mode select	000: 8-bit transmit mode 010: 4-bit transmit mode 100: 8-bit transmit / receive mode 101: 8-bit receive mode 110: 4-bit receive mode Except the above: Reserved	Write only		
SCK	Serial clock select				
			NORMAL1/2, IDLE1/2 mode	SLOW1/2 SLEEP1/2 mode	
			DV7CK = 0	DV7CK = 1	
		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	$fc/2^4$	$fc/2^4$	-		
110	Reserved				
111	External clock ( Input from SCK0 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: SIO0CR1 is write-only register, which cannot access any of in read-modify-write instruction such as bit operate, etc.

### SIO Control Register 2

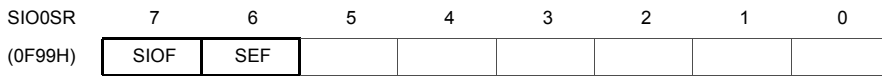




WAIT	Wait control	Always sets "00" except 8-bit transmit / receive mode. 00: $T_f = T_D$ (Non wait) 01: $T_f = 2T_D$ (Wait) 10: $T_f = 4T_D$ (Wait) 11: $T_f = 8T_D$ (Wait)	Write only
BUF	Number of transfer words (Buffer address in use)	000: 1 word transfer 0F90H 001: 2 words transfer 0F90H ~ 0F91H 010: 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: SIO0CR2 must be set when the serial interface is stopped (SIOF = 0).
- Note 5: \*: Don't care
- Note 6: SIO0CR2 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".

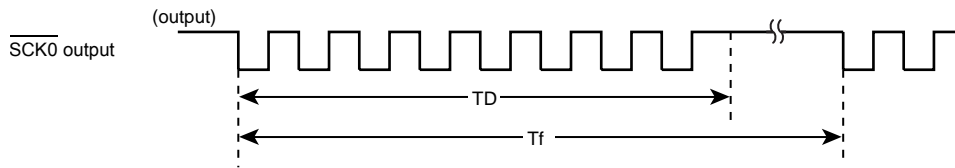


Figure 12-2 Frame time ( $T_f$ ) and Data transfer time ( $T_D$ )

12.3 Serial clock

12.3.1 Clock source

Internal clock or external clock for the source clock is selected by SIO0CR1<SCK>.

### 12.3.1.1 Internal clock

Any of six frequencies can be selected. The serial clock is output to the outside on the SCK0 pin. The SCK0 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)

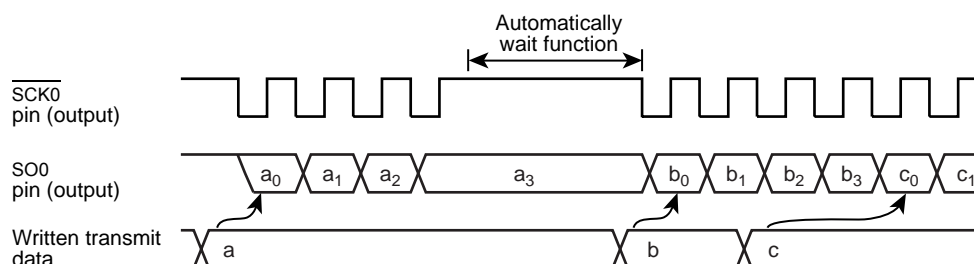


Figure 12-3 Automatic Wait Function (at 4-bit transmit mode)

### 12.3.1.2 External clock

An external clock connected to the SCK0 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).

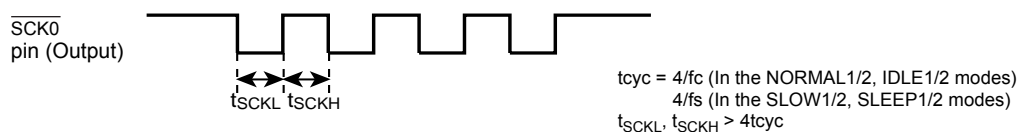


Figure 12-4 External clock pulse width

### 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  $\overline{SCK0}$  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  $\overline{SCK0}$  pin input/output).

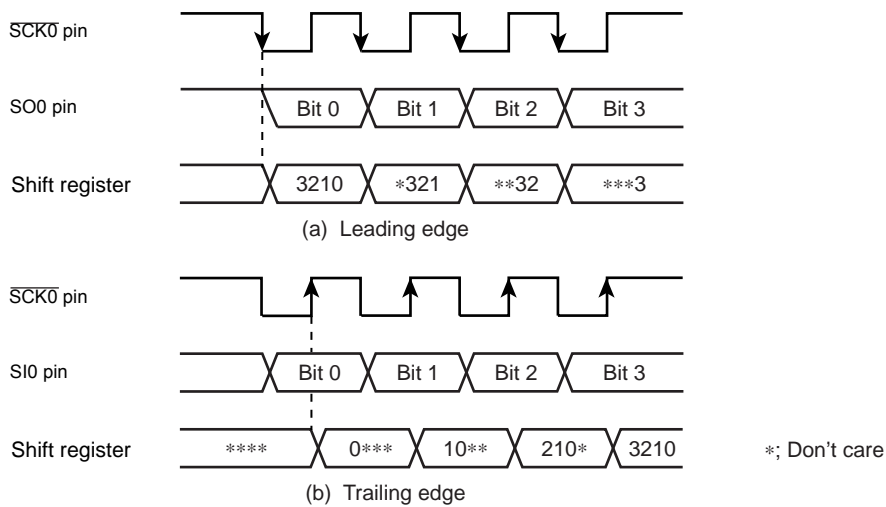


Figure 12-5 Shift edge

## 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 SIO0CR2<BUF>.

An INTSIO0 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.

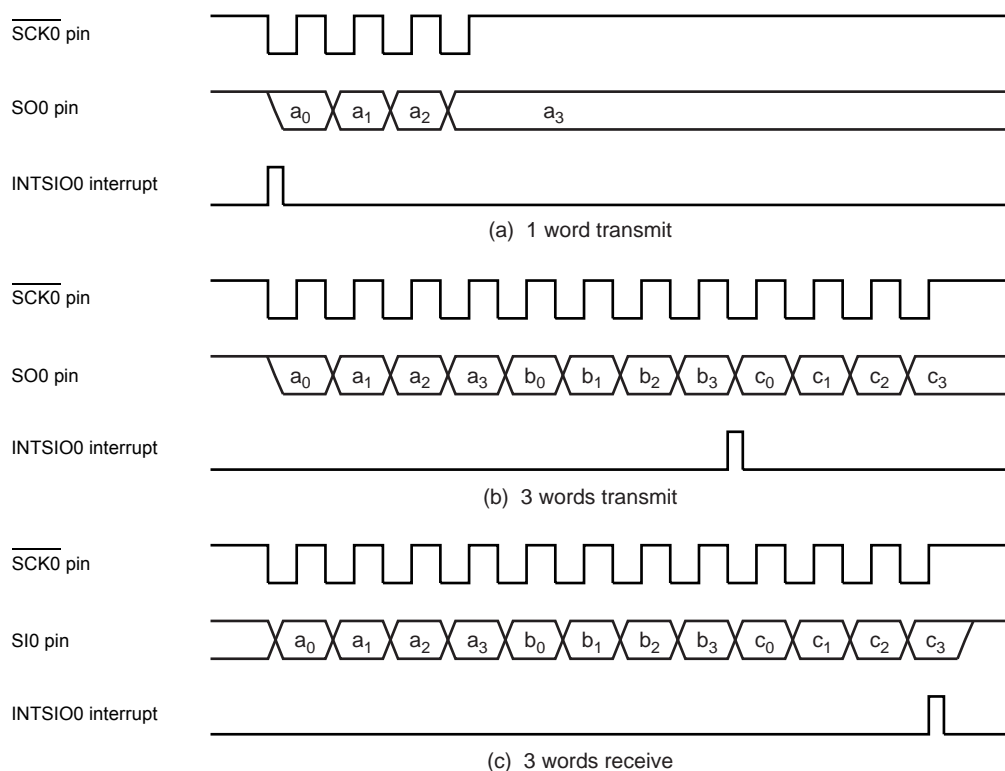


Figure 12-6 Number of words to transfer (Example: 1word = 4bit)

## 12.6 Transfer Mode

SIO0CR1<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 SIO0CR1<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 INTSIO0 (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 SIO0CR2<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 SIO0CR1<SIOS> to “0” or setting SIO0CR1<SIOINH> to “1” in buffer empty interrupt service program.

SIO0CR1<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 SIO0SR<SIOF> because SIO0SR<SIOF> is cleared to “0” when a transfer is completed.

When SIO0CR1<SIOINH> is set, the transmission is immediately ended and SIO0SR<SIOF> is cleared to “0”.

When an external clock is used, it is also necessary to clear SIO0CR1<SIOS> to “0” before shifting the next data; If SIO0CR1<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, SIO0CR1<SIOS> should be cleared to “0”, then SIO0CR2<BUF> must be rewritten after confirming that SIO0SR<SIOF> has been cleared to “0”.

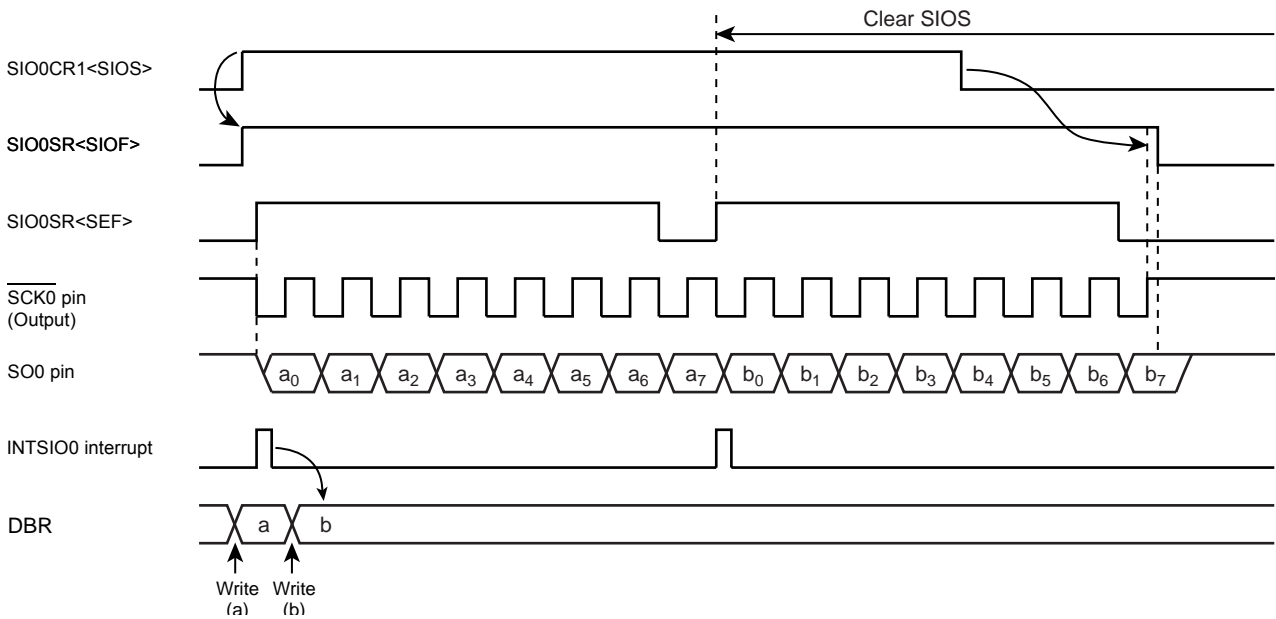


Figure 12-7 Transfer Mode (Example: 8bit, 1word transfer, Internal clock)

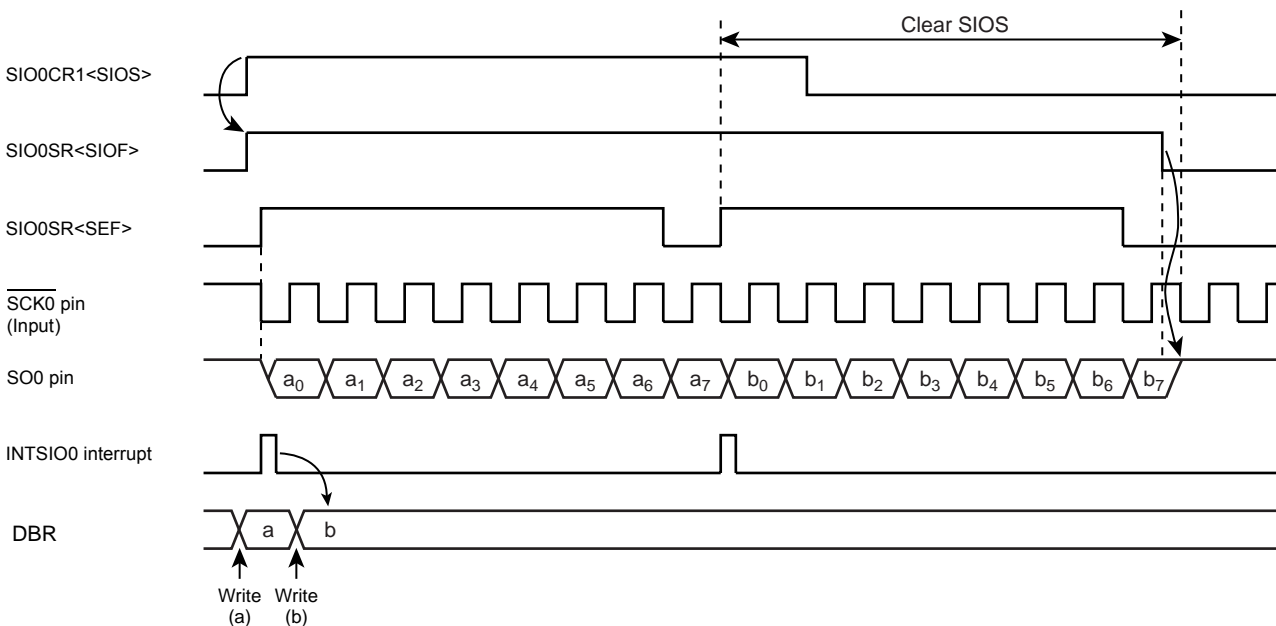


Figure 12-8 Transfer Mode (Example: 8bit, 1word transfer, External clock)

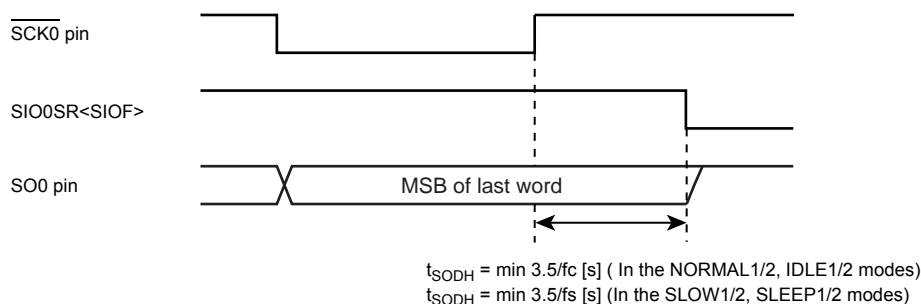


Figure 12-9 Transmitted Data Hold Time at End of Transfer

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

After setting the control registers to the receive mode, set SIO0CR1<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 SIO0CR2<BUF> has been received, an INTSIO0 (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 SIO0 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 SIO0CR1<SIOS> to “0” or setting SIO0CR1<SIOINH> to “1” in buffer full interrupt service program.

When SIO0CR1<SIOS> is cleared, the current data are transferred to the buffer. After SIO0CR1<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 SIO0SR<SIOF>. SIO0SR<SIOF> is cleared to “0” when the receiving is ended. After confirmed the receiving termination, the final receiving data is read. When SIO0CR1<SIOINH> is set, the receiving is immediately ended and SIO0SR<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, SIO0CR1<SIOS> should be cleared to “0” then SIO0CR2<BUF> must be rewritten after confirming that SIO0SR<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, SIO0CR2<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 SIO0CR1<SIOS> to “0”, read the last data and then switch the transfer mode.

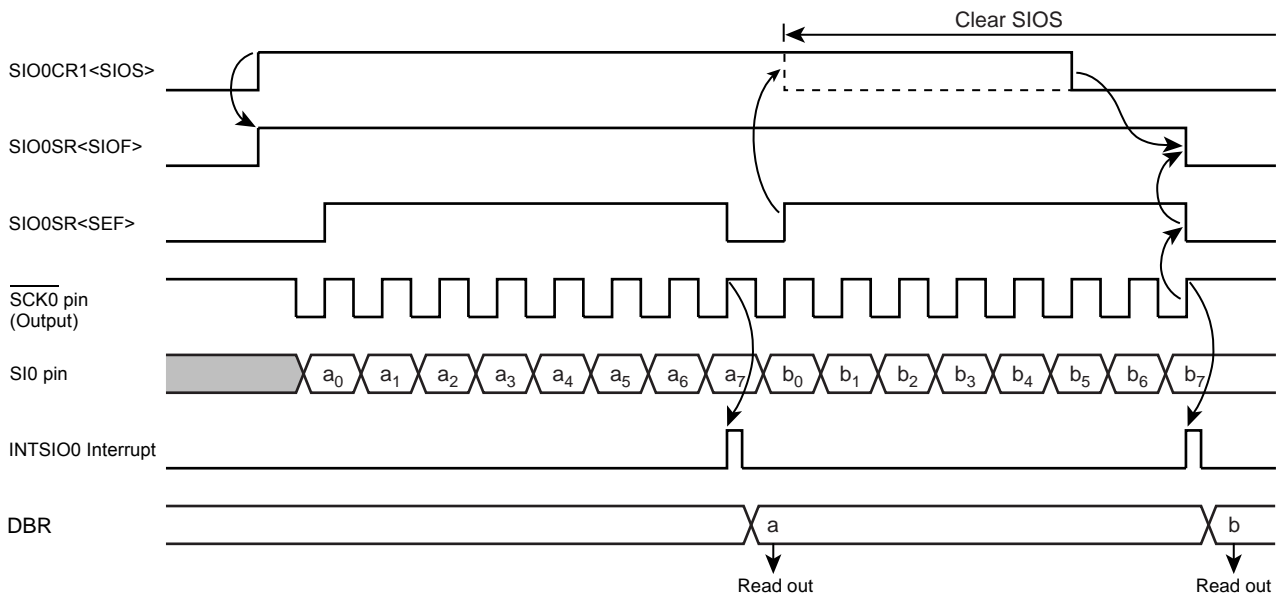


Figure 12-10 Receive Mode (Example: 8bit, 1word transfer, Internal clock)

### 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 SIO0CR1<SIOS> to “1”. When transmitting, the data are output from the SO0 pin at leading edges of the serial clock. When receiving, the data are input to the SIO 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 INTSIO0 interrupt is generated when the number of data words specified with the SIO0CR2<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 SIO0CR1<SIOS> to “0” or setting SIO0CR1<SIOINH> to “1” in INTSIO0 interrupt service program.

When SIO0CR1<SIOS> is cleared, the current data are transferred to the buffer. After SIO0CR1<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 SIO0SR<SIOF>. SIO0SR<SIOF> is cleared to “0” when the transmitting/receiving is ended.

When SIO0CR1<SIOINH> is set, the transmit/receive operation is immediately ended and SIO0SR<SIOF> is cleared to “0”.

If it is necessary to change the number of words in external clock operation, SIO0CR1<SIOS> should be cleared to “0”, then SIO0CR2<BUF> must be rewritten after confirming that SIO0SR<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, SIO0CR2<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 SIO0CR1<SIOS> to “0”, read the last data and then switch the transfer mode.

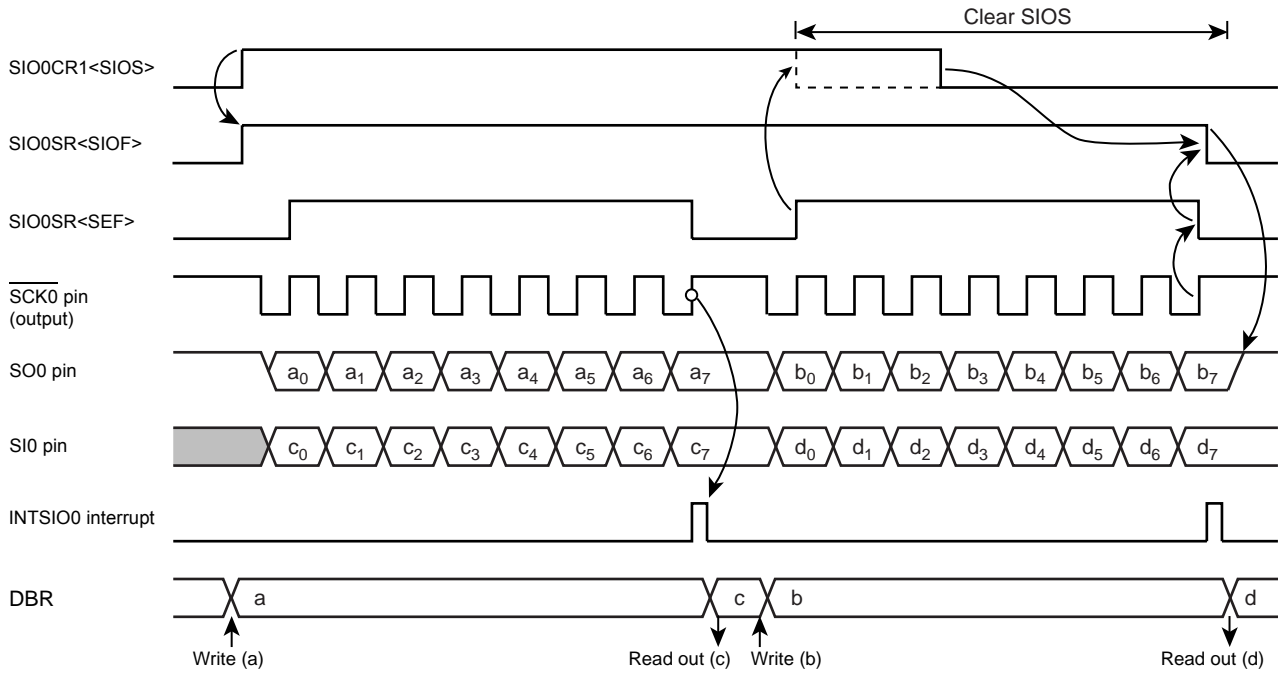


Figure 12-11 Transfer / Receive Mode (Example: 8bit, 1word transfer, Internal clock)

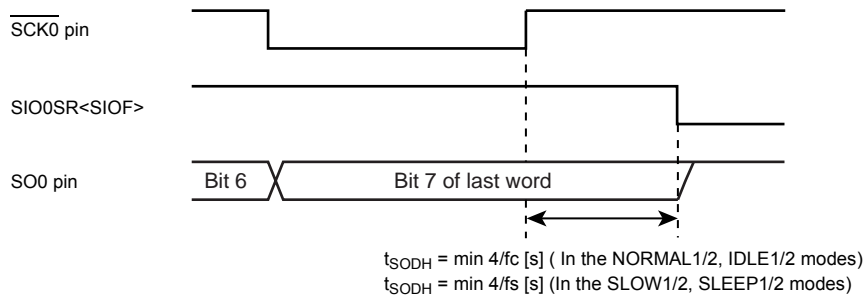


Figure 12-12 Transmitted Data Hold Time at End of Transfer / Receive



## 13. Synchronous Serial Interface (SIO1)

The TMP86CS25AFG has a clocked-synchronous 8-bit serial interface. Serial interface has an 8-byte transmit and receive data buffer that can automatically and continuously transfer up to 64 bits of data.

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

### 13.1 Configuration

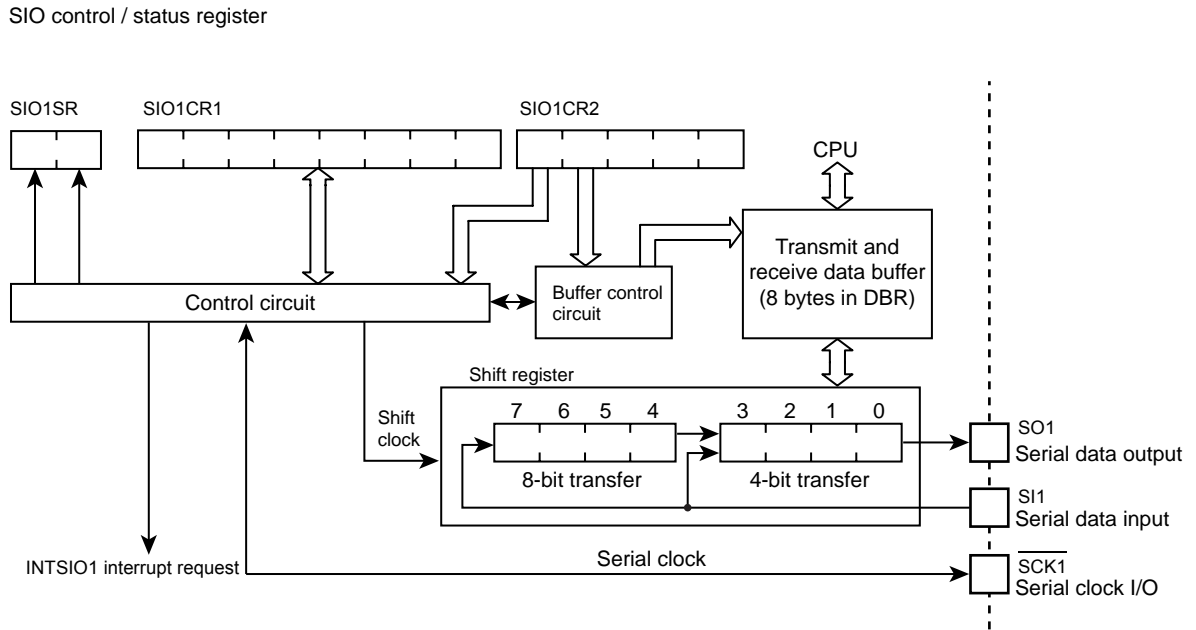


Figure 13-1 Serial Interface

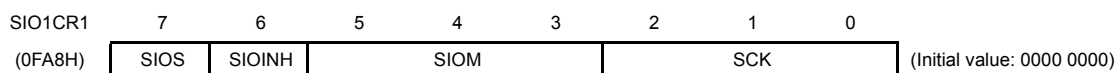
## 13.2 Control

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

The transmit and receive data buffer is controlled by the SIO1CR2<BUF>. The data buffer is assigned to address 0FA0H to 0FA7H for SIO in the DBR area, and can continuously transfer up to 8 words (bytes or nibbles) at one time. When the specified number of words has been transferred, a buffer empty (in the transmit mode) or a buffer full (in the receive mode or transmit/receive mode) interrupt (INTSIO1) is generated.

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

### SIO Control Register 1



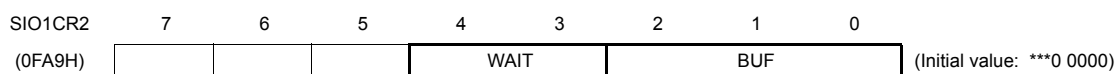
SIOS	Indicate transfer start / stop	0: Stop 1: Start	Write only		
SIOINH	Continue / abort transfer	0: Continuously transfer 1: Abort transfer (Automatically cleared after abort)			
SIOM	Transfer mode select	000: 8-bit transmit mode 010: 4-bit transmit mode 100: 8-bit transmit / receive mode 101: 8-bit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

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

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

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

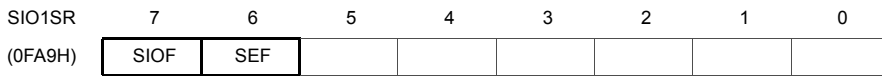
### SIO Control Register 2



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 0FA0H 001: 2 words transfer 0FA0H ~ 0FA1H 010: 3 words transfer 0FA0H ~ 0FA2H 011: 4 words transfer 0FA0H ~ 0FA3H 100: 5 words transfer 0FA0H ~ 0FA4H 101: 6 words transfer 0FA0H ~ 0FA5H 110: 7 words transfer 0FA0H ~ 0FA6H 111: 8 words transfer 0FA0H ~ 0FA7H	

- 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 0FA0H ).
- Note 3: The value to be loaded to BUF is held after transfer is completed.
- Note 4: SIO1CR2 must be set when the serial interface is stopped (SIOF = 0).
- Note 5: \*: Don't care
- Note 6: SIO1CR2 is write-only register, which cannot access any of in read-modify-write instruction such as bit operate, etc.

SIO Status Register



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

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

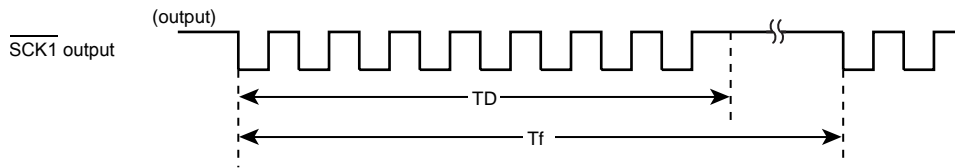


Figure 13-2 Frame time ( $T_f$ ) and Data transfer time ( $T_D$ )

13.3 Serial clock

13.3.1 Clock source

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

### 13.3.1.1 Internal clock

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

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

Table 13-1 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)

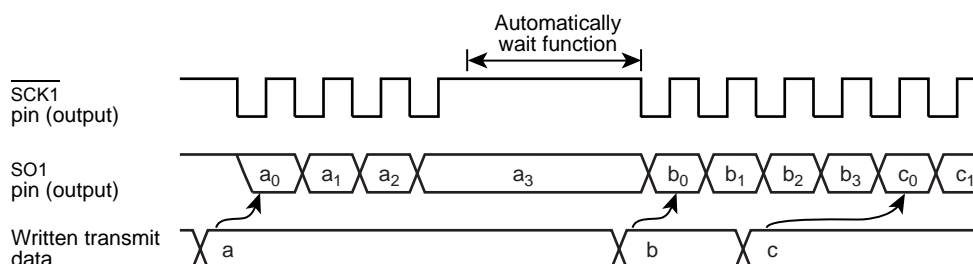


Figure 13-3 Automatic Wait Function (at 4-bit transmit mode)

### 13.3.1.2 External clock

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

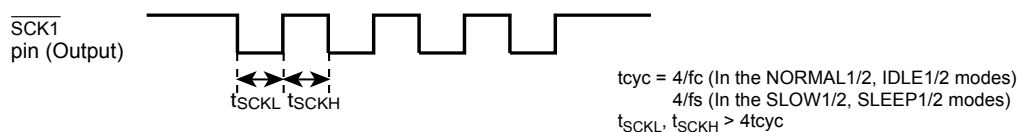


Figure 13-4 External clock pulse width

### 13.3.2 Shift edge

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

#### 13.3.2.1 Leading edge

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

#### 13.3.2.2 Trailing edge

Received data are shifted on the trailing edge of the serial clock (rising edge of the  $\overline{SCK1}$  pin input/output).

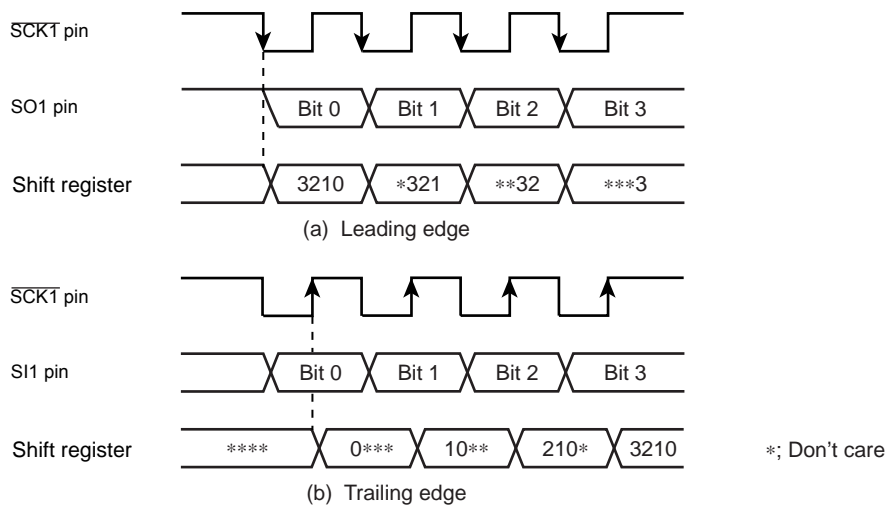


Figure 13-5 Shift edge

### 13.4 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).

### 13.5 Number of words to transfer

Up to 8 words consisting of 4 bits of data (4-bit serial transfer) or 8 bits (8-bit serial transfer) of data can be transferred continuously. The number of words to be transferred can be selected by SIO1CR2<BUF>.

An INTSIO1 interrupt is generated when the specified number of words has been transferred. If the number of words is to be changed during transfer, the serial interface must be stopped before making the change. The number of words can be changed during automatic-wait operation of an internal clock. In this case, the serial interface is not required to be stopped.

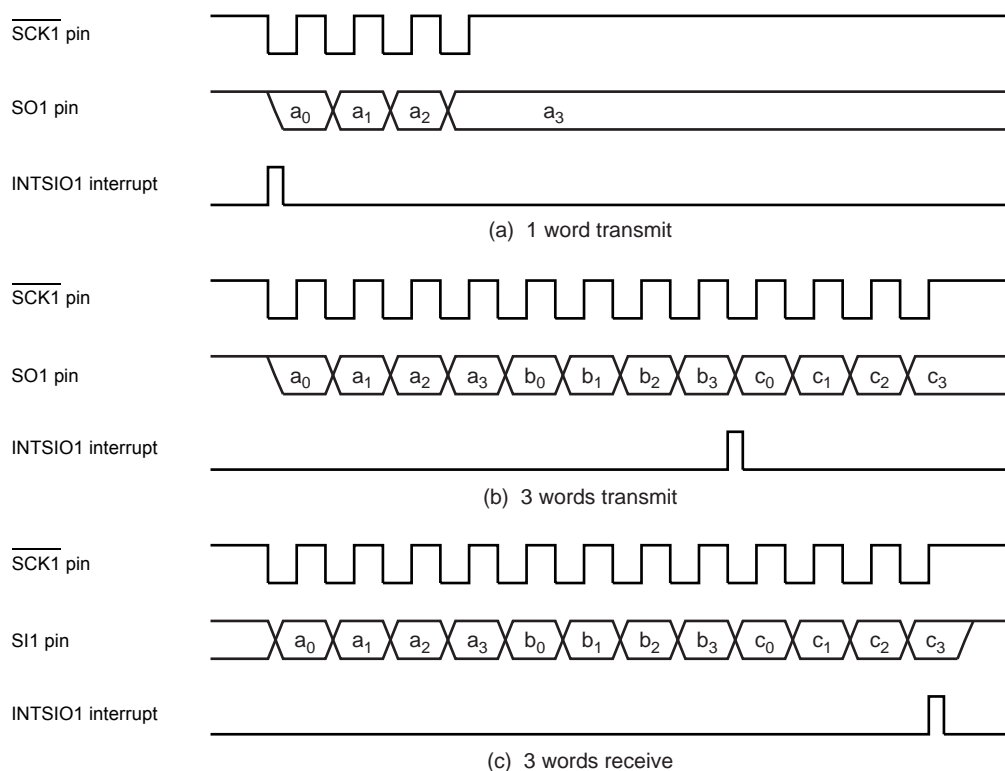


Figure 13-6 Number of words to transfer (Example: 1word = 4bit)

## 13.6 Transfer Mode

SIO1CR1<SIOM> is used to select the transmit, receive, or transmit/receive mode.

### 13.6.1 4-bit and 8-bit transfer modes

In these modes, firstly set the SIO control register to the transmit mode, and then write first transmit data (number of transfer words to be transferred) to the data buffer registers (DBR).

After the data are written, the transmission is started by setting SIO1CR1<SIOS> to “1”. The data are then output sequentially to the SO pin in synchronous with the serial clock, starting with the least significant bit (LSB). As soon as the LSB has been output, the data are transferred from the data buffer register to the shift register. When the final data bit has been transferred and the data buffer register is empty, an INTSIO1 (Buffer empty) interrupt is generated to request the next transmitted data.

When the internal clock is used, the serial clock will stop and an automatic-wait will be initiated if the next transmitted data are not loaded to the data buffer register by the time the number of data words specified with the SIO1CR2<BUF> has been transmitted. Writing even one word of data cancels the automatic-wait; therefore, when transmitting two or more words, always write the next word before transmission of the previous word is completed.

Note: Automatic waits are also canceled by writing to a DBR not being used as a transmit data buffer register; therefore, during SIO do not use such DBR for other applications. For example, when 3 words are transmitted, do not use the DBR of the remained 5 words.

When an external clock is used, the data must be written to the data buffer register before shifting next data. Thus, the transfer speed is determined by the maximum delay time from the generation of the interrupt request to writing of the data to the data buffer register by the interrupt service program.

The transmission is ended by clearing SIO1CR1<SIOS> to “0” or setting SIO1CR1<SIOINH> to “1” in buffer empty interrupt service program.

SIO1CR1<SIOS> is cleared, the operation will end after all bits of words are transmitted.

That the transmission has ended can be determined from the status of SIO1SR<SIOF> because SIO1SR<SIOF> is cleared to “0” when a transfer is completed.

When SIO1CR1<SIOINH> is set, the transmission is immediately ended and SIO1SR<SIOF> is cleared to “0”.

When an external clock is used, it is also necessary to clear SIO1CR1<SIOS> to “0” before shifting the next data; If SIO1CR1<SIOS> is not cleared before shift out, dummy data will be transmitted and the operation will end.

If it is necessary to change the number of words, SIO1CR1<SIOS> should be cleared to “0”, then SIO1CR2<BUF> must be rewritten after confirming that SIO1SR<SIOF> has been cleared to “0”.

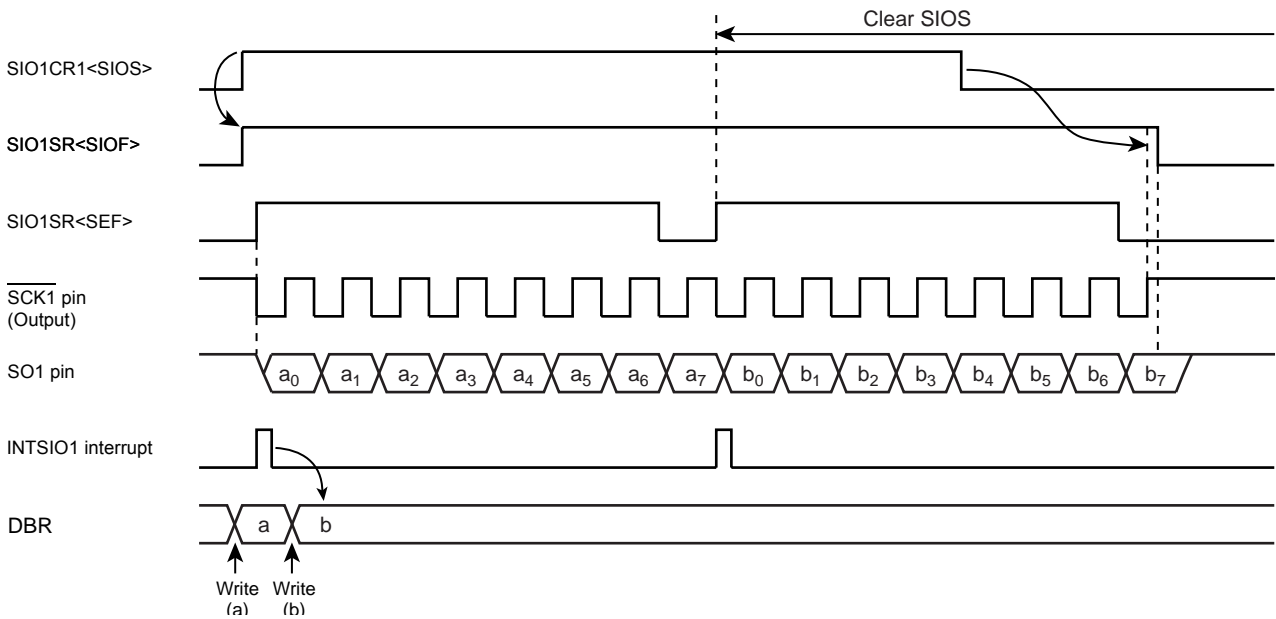


Figure 13-7 Transfer Mode (Example: 8bit, 1word transfer, Internal clock)

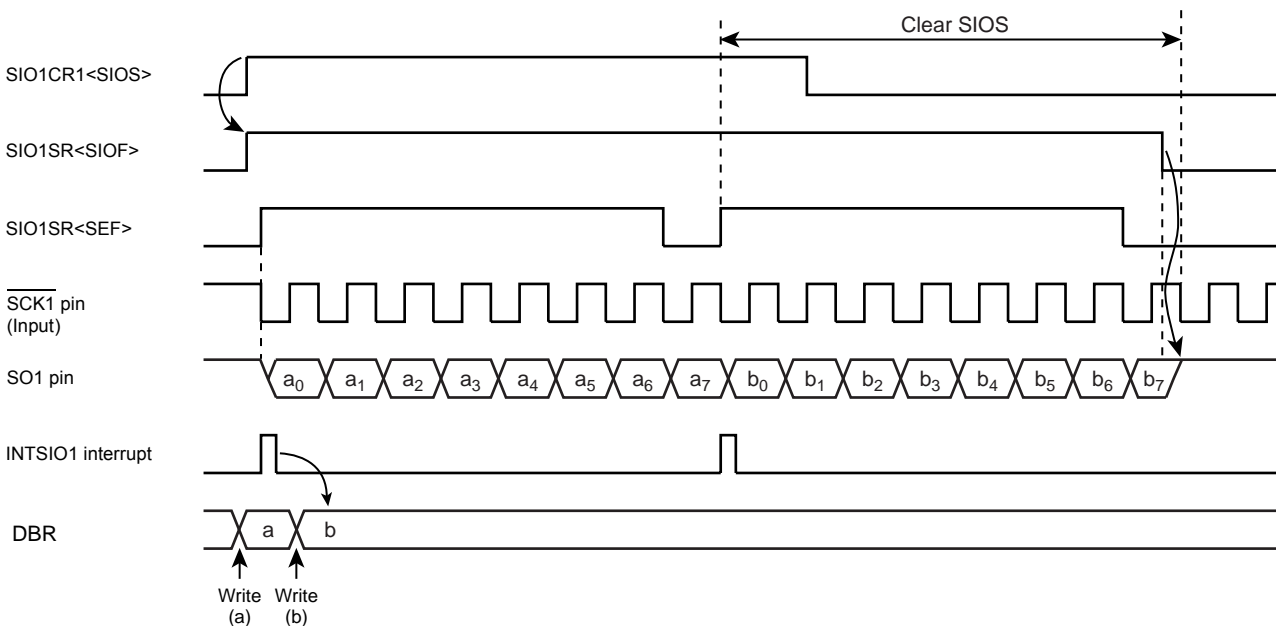


Figure 13-8 Transfer Mode (Example: 8bit, 1word transfer, External clock)

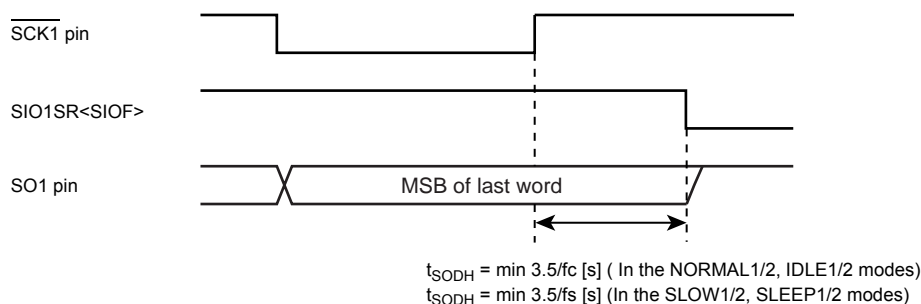


Figure 13-9 Transmitted Data Hold Time at End of Transfer

### 13.6.2 4-bit and 8-bit receive modes

After setting the control registers to the receive mode, set SIO1CR1<SIOS> to “1” to enable receiving. The data are then transferred to the shift register via the SI pin in synchronous with the serial clock. When one word of data has been received, it is transferred from the shift register to the data buffer register (DBR). When the number of words specified with the SIO1CR2<BUF> has been received, an INTSIO1 (Buffer full) interrupt is generated to request that these data be read out. The data are then read from the data buffer registers by the interrupt service program.

When the internal clock is used, and the previous data are not read from the data buffer register before the next data are received, the serial clock will stop and an automatic-wait will be initiated until the data are read. A wait will not be initiated if even one data word has been read.

Note: Waits are also canceled by reading a DBR not being used as a received data buffer register is read; therefore, during SIO1 do not use such DBR for other applications.

When an external clock is used, the shift operation is synchronized with the external clock; therefore, the previous data are read before the next data are transferred to the data buffer register. If the previous data have not been read, the next data will not be transferred to the data buffer register and the receiving of any more data will be canceled. When an external clock is used, the maximum transfer speed is determined by the delay between the time when the interrupt request is generated and when the data received have been read.

The receiving is ended by clearing SIO1CR1<SIOS> to “0” or setting SIO1CR1<SIOINH> to “1” in buffer full interrupt service program.

When SIO1CR1<SIOS> is cleared, the current data are transferred to the buffer. After SIO1CR1<SIOS> cleared, the receiving is ended at the time that the final bit of the data has been received. That the receiving has ended can be determined from the status of SIO1SR<SIOF>. SIO1SR<SIOF> is cleared to “0” when the receiving is ended. After confirmed the receiving termination, the final receiving data is read. When SIO1CR1<SIOINH> is set, the receiving is immediately ended and SIO1SR<SIOF> is cleared to “0”. (The received data is ignored, and it is not required to be read out.)

If it is necessary to change the number of words in external clock operation, SIO1CR1<SIOS> should be cleared to “0” then SIO1CR2<BUF> must be rewritten after confirming that SIO1SR<SIOF> has been cleared to “0”. If it is necessary to change the number of words in internal clock, during automatic-wait operation which occurs after completion of data receiving, SIO1CR2<BUF> must be rewritten before the received data is read out.

Note: The buffer contents are lost when the transfer mode is switched. If it should become necessary to switch the transfer mode, end receiving by clearing SIO1CR1<SIOS> to “0”, read the last data and then switch the transfer mode.



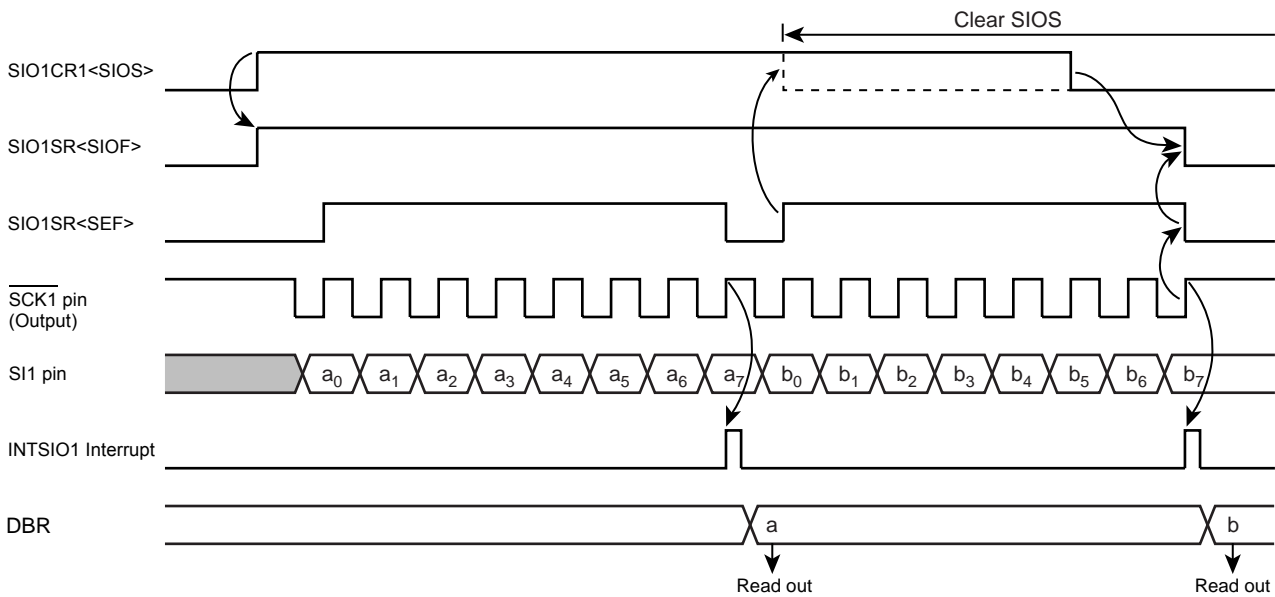


Figure 13-10 Receive Mode (Example: 8bit, 1word transfer, Internal clock)

### 13.6.3 8-bit transfer / receive mode

After setting the SIO control register to the 8-bit transmit/receive mode, write the data to be transmitted first to the data buffer registers (DBR). After that, enable the transmit/receive by setting SIO1CR1<SIOS> to "1". When transmitting, the data are output from the SO1 pin at leading edges of the serial clock. When receiving, the data are input to the SII pin at the trailing edges of the serial clock. When the all receive is enabled, 8-bit data are transferred from the shift register to the data buffer register. An INTSIO1 interrupt is generated when the number of data words specified with the SIO1CR2<BUF> has been transferred. Usually, read the receive data from the buffer register in the interrupt service. The data buffer register is used for both transmitting and receiving; therefore, always write the data to be transmitted after reading the all received data.

When the internal clock is used, a wait is initiated until the received data are read and the next transfer data are written. A wait will not be initiated if even one transfer data word has been written.

When an external clock is used, the shift operation is synchronized with the external clock; therefore, it is necessary to read the received data and write the data to be transmitted next before starting the next shift operation. When an external clock is used, the transfer speed is determined by the maximum delay between generation of an interrupt request and the received data are read and the data to be transmitted next are written.

The transmit/receive operation is ended by clearing SIO1CR1<SIOS> to "0" or setting SIO1CR1<SIOINH> to "1" in INTSIO1 interrupt service program.

When SIO1CR1<SIOS> is cleared, the current data are transferred to the buffer. After SIO1CR1<SIOS> cleared, the transmitting/receiving is ended at the time that the final bit of the data has been transmitted.

That the transmitting/receiving has ended can be determined from the status of SIO1SR<SIOF>. SIO1SR<SIOF> is cleared to "0" when the transmitting/receiving is ended.

When SIO1CR1<SIOINH> is set, the transmit/receive operation is immediately ended and SIO1SR<SIOF> is cleared to "0".

If it is necessary to change the number of words in external clock operation, SIO1CR1<SIOS> should be cleared to "0", then SIO1CR2<BUF> must be rewritten after confirming that SIO1SR<SIOF> has been cleared to "0".

If it is necessary to change the number of words in internal clock, during automatic-wait operation which occurs after completion of transmit/receive operation, SIO1CR2<BUF> must be rewritten before reading and writing of the receive/transmit data.

Note: The buffer contents are lost when the transfer mode is switched. If it should become necessary to switch the transfer mode, end receiving by clearing SIO1CR1<SIOS> to "0", read the last data and then switch the transfer mode.

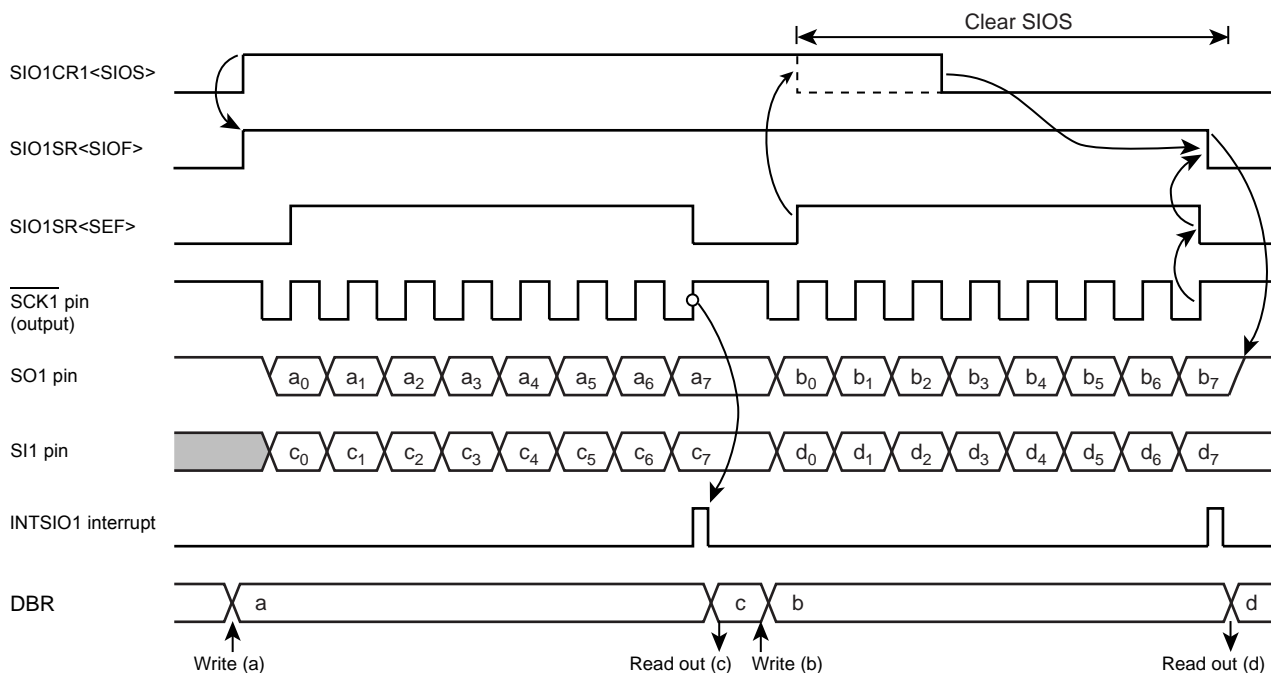


Figure 13-11 Transfer / Receive Mode (Example: 8bit, 1word transfer, Internal clock)

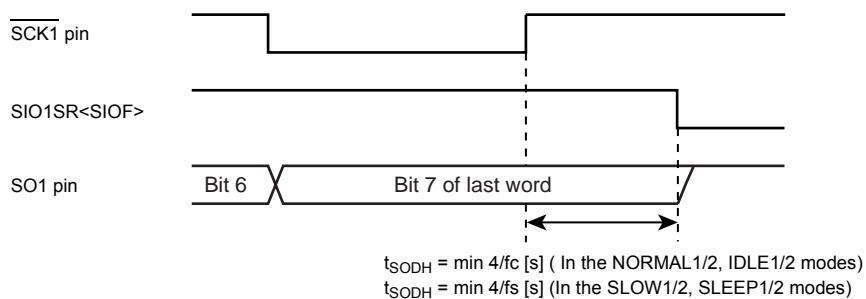


Figure 13-12 Transmitted Data Hold Time at End of Transfer / Receive

# 14. 8-Bit AD Converter (ADC)

The TMP86CS25AFG have a 8-bit successive approximation type AD converter.

## 14.1 Configuration

The circuit configuration of the 8-bit AD converter is shown in Figure 14-1.

It consists of control registers ADCCR1 and ADCCR2, converted value registers ADCDR1 and ADCDR2, a DA converter, a sample-and-hold circuit, a comparator, and a successive comparison circuit.

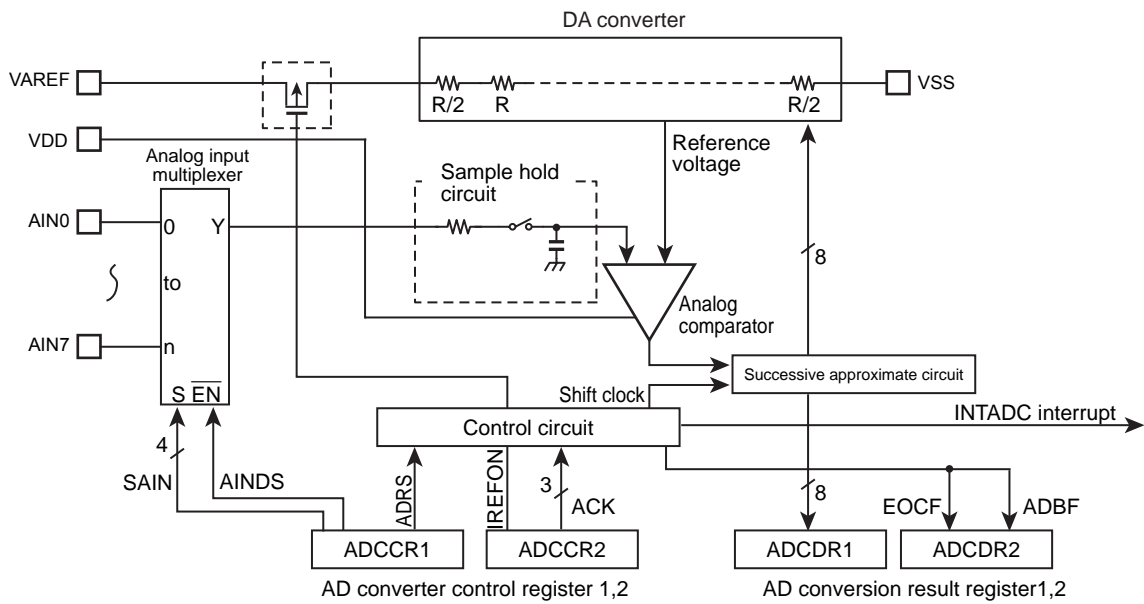


Figure 14-1 8-bit AD Converter (ADC)

## 14.2 Control

The AD converter consists of the following four registers:

1. AD converter control register 1 (ADCCR1)

This register selects the analog channels 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).

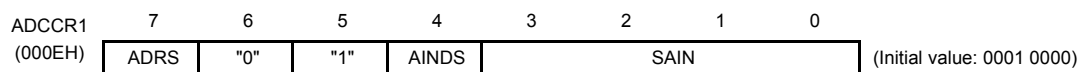
3. AD converted value register (ADCDR1)

This register is used to store the digital value after being converted by the AD converter.

4. AD converted value register (ADCDR2)

This register monitors the operating status of the AD converter.

### AD Converter Control Register 1



ADRS	AD conversion start	0: – 1: Start	R/W
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 when AD converter stops (ADCDR2<ADBF> = "0").

Note 2: When the analog input is all use disabling, the ADCCR1<AINDS> should be set to "1".

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

Note 4: The 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 register 1 (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.

Note 7: Always set bit 5 in ADCCR1 to "1" and set bit 6 in ADCCR1 to "0".

## AD Converter Control Register 2

ADCCR2 (000FH)	7	6	5	4	3	2	1	0	
			IREFON	"1"		ACK		"0"	(Initial value: **0* 000*)

IREFON	DA converter (ladder resistor) connection control	0: Connected only during AD conversion 1: Always connected	R/W
ACK	AD conversion time select	000: 39/fc 001: Reserved 010: 78/fc 011: 156/fc 100: 312/fc 101: 624/fc 110: 1248/fc 111: Reserved	R/W

Note 1: Always set bit 0 in ADCCR2 to "0" and set bit 4 in ADCCR2 to "1".

Note 2: When a read instruction for ADCCR2, bit 6 to 7 in ADCCR2 read in as undefined data.

Note 3: After STOP or SLOW/SLEEP mode are started, AD converter control register 2 (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 14-1 Conversion Time according to ACK Setting and Frequency

Condition ACK	Conversion time'	16MHz	8MHz	4 MHz	2 MHz	10MHz	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: Settings for "-" in the above table are inhibited.

Note 2: Set conversion time by Analog Reference Voltage ( $V_{AREF}$ ) as follows.

- $V_{AREF} = 4.5$  to  $5.5$  V (15.6 μs or more)
- $V_{AREF} = 2.7$  to  $5.5$  V (31.2 μs or more)
- $V_{AREF} = 1.8$  to  $5.5$  V (124.8 μs or more)

## AD Conversion Result Register

ADCDR1 (0020H)	7	6	5	4	3	2	1	0	
	AD07	AD06	AD05	AD04	AD03	AD02	AD01	AD00	(Initial value: 0000 0000)

## AD Conversion Result Register

ADCDR2 (0021H)	7	6	5	4	3	2	1	0	
			EOCF	ADBF					(Initial value: **00 ****)

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: ADCDR2<ADBF> is set to "1" when AD conversion starts and cleared to "0" when the AD conversion is finished. It also is cleared upon entering STOP or SLOW mode.

Note 3: If a read instruction is executed for ADCDR2, read data of bits 7, 6 and 3 to 0 are unstable.

## 14.3 Function

### 14.3.1 AD Converter Operation

When ADCCR1<ADRS> is set 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) and at the same time ADCDR2<EOCF> is set to "1", the AD conversion finished interrupt (INTADC) is generated.

ADCCR1<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 ADCDR<EOCF> to see that the conversion is completed or wait until the interrupt signal (INTADC) is generated (e.g., interrupt handling routine).

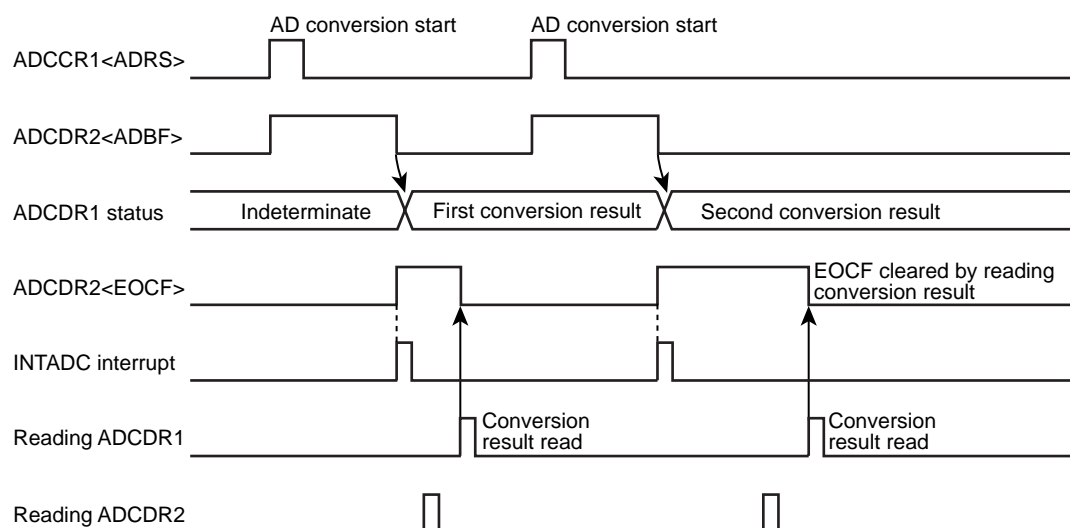


Figure 14-2 AD Converter Operation

### 14.3.2 AD Converter Operation

- 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).
- 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 Table 14-1.
  - 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".
- 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 of 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 and store the 8-bit data in address 009FH on RAM.

```

; AIN SELECT
:
: ; Before setting the AD converter register, set each port reg-
: ; ister suitably (For detail, see chapter of I/O port.)
LD (ADCCR1), 00100011B ; Select AIN3
LD (ADCCR2), 11011000B ; Select conversion time (312/fc) and operation mode
; AD CONVERT START
SET (ADCCR1), 7 ; ADRS = 1
SLOOP: TEST (ADCDR2), 5 ; EOCF = 1 ?
JRS T, SLOOP
; RESULT DATA READ
LD A, (ADCDR1)
LD (9FH), A

```

### 14.3.3 STOP and SLOW Mode during AD Conversion

When the 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 STOP or SLOW mode.) When restored from 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.

### 14.3.4 Analog Input Voltage and AD Conversion Result

The analog input voltage is corresponded to the 8-bit digital value converted by the AD as shown in Figure 14-3.

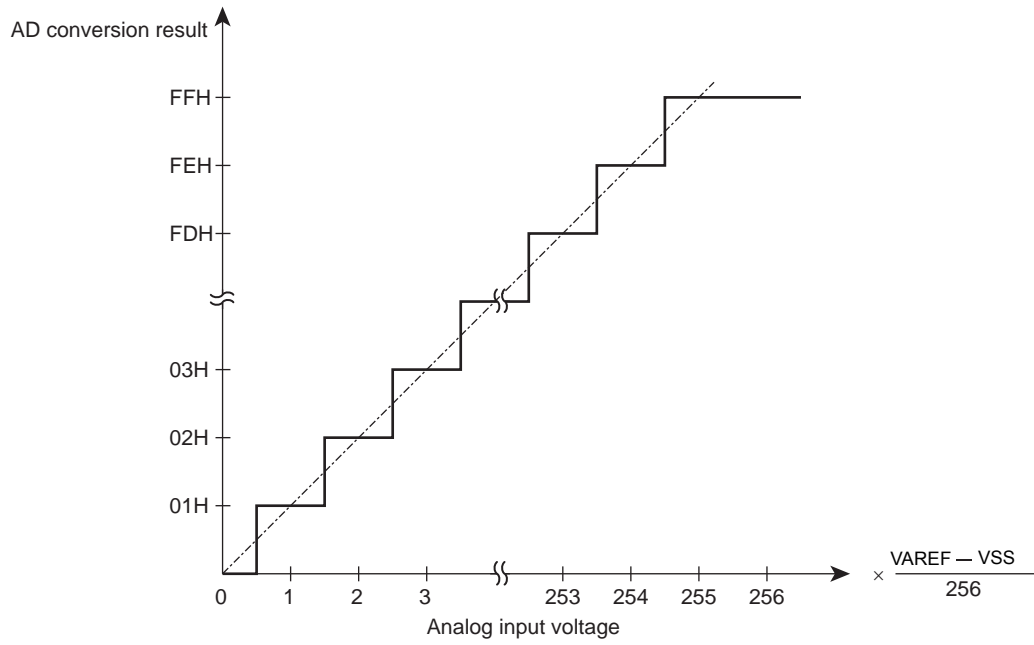


Figure 14-3 Analog Input Voltage and AD Conversion Result (typ.)



## 14.4 Precautions about AD Converter

### 14.4.1 Analog input pin voltage range

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

### 14.4.2 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.

### 14.4.3 Noise countermeasure

The internal equivalent circuit of the analog input pins is shown in Figure 14-4. The higher the output impedance of the analog input source, more easily they are susceptible to noise. Therefore, make sure the output impedance of the signal source in your design is 5 kΩ or less. Toshiba also recommends attaching a capacitor external to the chip.

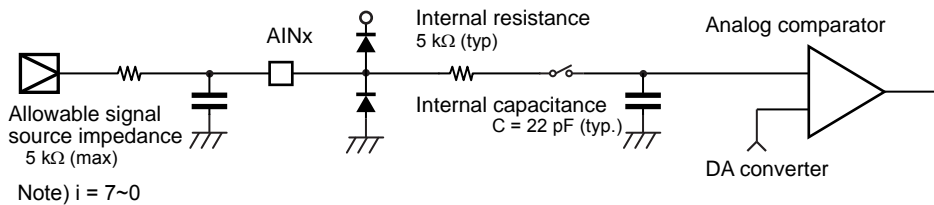


Figure 14-4 Analog Input Equivalent Circuit and Example of Input Pin Processing



# 15. Key-on Wakeup (KWU)

In the TMP86CS25AFG, 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 " 15.2 Control ".

## 15.1 Configuration

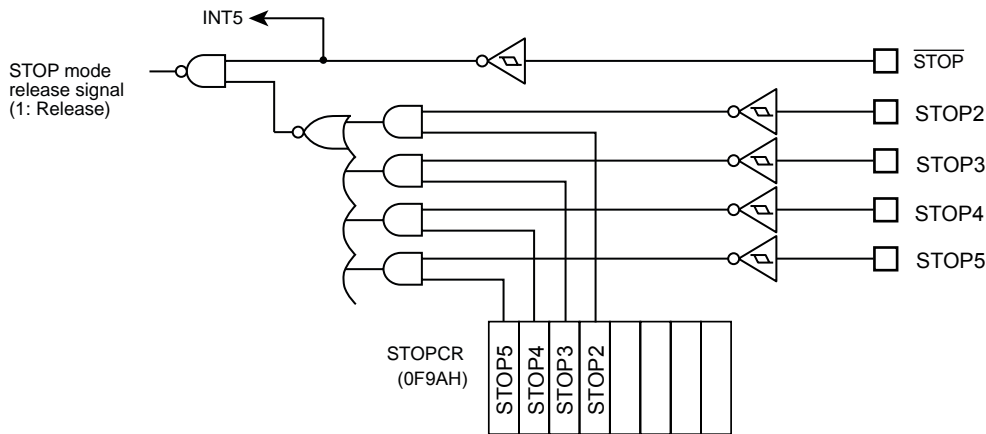


Figure 15-1 Key-on Wakeup Circuit

## 15.2 Control

STOP2 to STOP5 pins can be controlled by Key-on Wakeup Control Register (STOPCR). It can be configured as enable/disable in 1-bit unit. When those pins are used for STOP mode release, configure corresponding I/O pins to input mode by I/O port register beforehand.

### Key-on Wakeup Control Register

STOPCR	7	6	5	4	3	2	1	0	
(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

## 15.3 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).

Note 1: When the STOP mode 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:  $\overline{\text{STOP}}$  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 4: 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 AD conversion input (analog voltage input).

Note 5: When the STOP mode is released by STOP2 to STOP5 pins, the level of  $\overline{\text{STOP}}$  pin should hold "L" level (Figure 15-2).

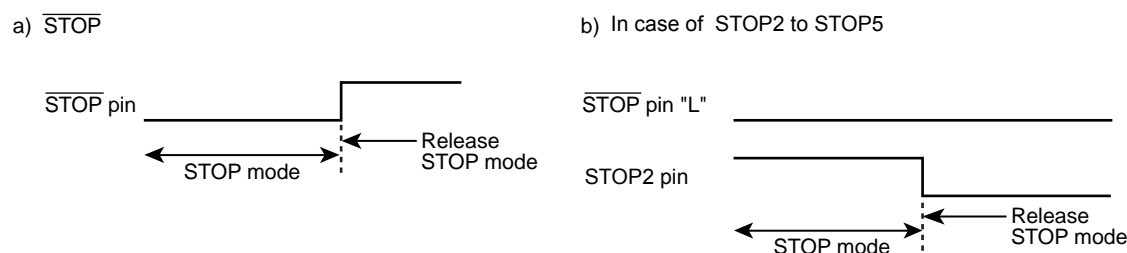


Figure 15-2 Priority of STOP pin and STOP2 to STOP5 pins

Table 15-1 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)

## 16. LCD Driver

The TMP86CS25AFG incorporates a driver to directly drive the liquid crystal display (LCD) and its control circuit. The connecting pins with the LCD are as shown below:

1. Segment output pin: 40 pins (SEG39 to SEG0)
2. Segment output/ I/O port pin (shared): 20 pins (SEG59 to SEG40)
3. Common output pin: 5 pins (COM4 to COM0)
4. Common output I/O port pin (shared): 11 pins (COM15 to COM5)

In addition, C0, C1, V1, V2, V3 and V4 are provided as the LCD drive booster circuit pins.

The following three types of LCD can be driven directly:

1. 1/4 duty LCD: Maximum 240 pixels (60 segments × 4 digits)
2. 1/8 duty LCD: Maximum 480 pixels (60 segments × 8 digits)
3. 1/16 duty LCD: Maximum 960 pixels (60 segments × 16 digits)

### 16.1 Configuration of LCD Driver

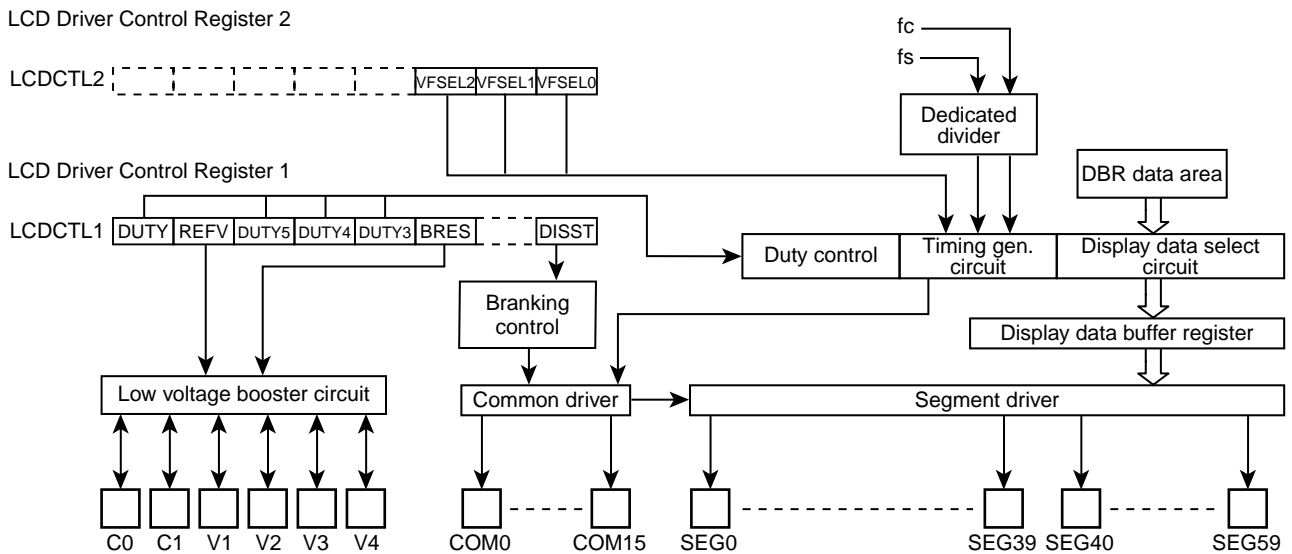


Figure 16-1 LCD Driver Block Diagram

Note: The LCD driver circuit has a built-in dedicated divider circuit. Thus, during use of the tool, LCD outputting is not stopped by debugger break processing.

## 16.2 Controlling LCD Driver

The LCD driver is controlled by the LCD control register 1 (LCDCTL1) and the LCD control register 2 (LCDCTL2). The display of the LCD driver is enabled by DISST.

### LCD Control Register 1

LCDCTL1 (0027H)	7	6	5	4	3	2	1	0	
	DUTY7	REFV	DUTY5	DUTY4	DUTY3	BRES		DISST	(Initial value: 0000 00*0)

DUTY7 DUTY5 DUTY4 DUTY3	Select duty.	0***: Reserved 1000: 1/4 duty 1001: Reserved 1010: 1/8 duty 1011: Reserved 1100: Reserved 1101: Reserved 1110: 1/16 duty 1111: Reserved	R/W
REFV	Sets LCD reference voltage.	0: $V4 \leq VDD$ (Note 4) 1: $VDD < V4 \leq 5.5 V$	
BRES	Sets booster circuit.	0: Booster circuit disable 1: Booster circuit enable (Note 5)	
DISST	Controls LCD display.	0: LCD display blanking 1: LCD display enable	

Note 1: After reset, LCDCTL1<DUTYs> are set to "0000" (Initial value: reserved). Set the duty as appropriate for LCD panel.

Note 2: Switch LCDCTL1<REFV> according to VDD. If it is not set appropriately, an overcurrent may flow causing damage to the device. Caution is especially required when VDD is battery-driven.

Note 3: If LCDCTL1<DISST> is set to "0" (LCD display blanking), all SEG/COM pins become VSS level.

Note 4: When LCDCTL1<REFV> for the LCD reference voltage is set to "0", always make sure the reference power supply is entered from the V4 pin. In this case, input voltage from V4 pin should be kept within  $2.7 V \leq V4 \leq VDD$ .

Note 5: When LCD is used, always set LCDCTL1<BRES> to "1".

Note 6: Reserved: Not to be set.

### LCD Control Register 2

LCDCTL2 (0028H)	7	6	5	4	3	2	1	0	
						VFSEL2	VFSEL1	VFSEL0	(Initial value: **** *011)

VFSEL	Selects base frequency for frame frequency.	000: $fc/2^9$ (at 16 MHz) 001: $fc/2^8$ (at 8 MHz) 010: $fc/2^7$ (at 6 MHz) 011: $fc/2^6$ (at 2 MHz) 1** : fs (at 32.768 kHz)	R/W
-------	---	---	-----

Note: Set the LCD control register 2 according to operating frequency. For details of the actual frame frequency, see Table 16-1

### 16.2.1 Frame frequency

The frame frequency is set depending on the driving method and the base frequency as shown in Table 16-1.

The base frequency is selected with LCDCTL2 <VFSEL> depending on the basic clock frequencies  $f_c$  and  $f_s$  to be used.

Table 16-1 Frame Frequency Settings

VFSEL	Base frequency [Hz]	Frame frequency [Hz]		
		1/4 duty	1/8 duty	1/16 duty
000	$\frac{f_c}{2^9}$	$\frac{f_c}{2^9 \cdot 84 \cdot 4}$	$\frac{f_c}{2^9 \cdot 42 \cdot 8}$	$\frac{f_c}{2^9 \cdot 21 \cdot 16}$
	( $f_c = 16$ MHz)	93	93	93
001	$\frac{f_c}{2^8}$	$\frac{f_c}{2^8 \cdot 84 \cdot 4}$	$\frac{f_c}{2^8 \cdot 42 \cdot 8}$	$\frac{f_c}{2^8 \cdot 21 \cdot 16}$
	( $f_c = 8$ MHz)	93	93	93
010	$\frac{f_c}{2^7}$	$\frac{f_c}{2^7 \cdot 84 \cdot 4}$	$\frac{f_c}{2^7 \cdot 42 \cdot 8}$	$\frac{f_c}{2^7 \cdot 21 \cdot 16}$
	( $f_c = 4$ MHz)	93	93	93
011	$\frac{f_c}{2^6}$	$\frac{f_c}{2^6 \cdot 84 \cdot 4}$	$\frac{f_c}{2^6 \cdot 42 \cdot 8}$	$\frac{f_c}{2^6 \cdot 21 \cdot 16}$
	( $f_c = 2$ MHz)	93	93	93
1**	$f_s$	$\frac{f_s}{84 \cdot 4}$	$\frac{f_s}{42 \cdot 8}$	$\frac{f_s}{21 \cdot 16}$
	( $f_s = 32.768$ kHz)	97.5	97.5	97.5

Note 1:  $f_c$  ; High-frequency clock frequency [Hz],  $f_s$  ; Low-frequency clock frequency [Hz]

Note 2: Although this product is guaranteed to operate at  $f_c = 1.32$  [MHz] or less is not recommended for LCD display as the frame frequency becomes 61 [Hz] or less.

### 16.3 LCD Booster Circuit

The TMP86CS25AFG can boost (divide) the externally-supplied reference voltage using the built-in booster circuit as a power supply for driving the LCD. When V1 pin is the reference voltage, the inputted reference voltage is boosted by two times (V2), 3 times (V3) and 4 times (V4) to generate a voltage for a segment/common signal. When V2 pin is the reference voltage, the inputted reference voltage is divided/boosted by 1/2 time (V1), 3/2 times (V3) and two times (V4). Likewise, when V3 pin or V4 pin is the reference, the inputted reference voltage is boosted/divided and the voltage ratio is  $V1 \times 4 = V2 \times 2 = V3 \times (4/3) = V4$ . As this circuit uses a 4-times boosting method, the bias ratio is 1/4 only.

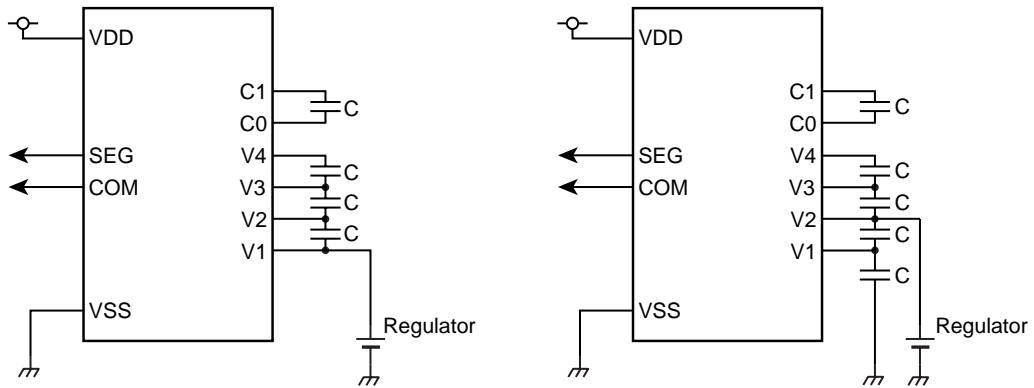
Note 1: When the reference pin is other than V1 pin, a condenser is required between V1 pin and GND.

Note 2: When LCDCTL1<REFV> is set to "0", input voltage from V4 pin should be kept within  $2.7 \text{ V} \leq V4 \leq \text{VDD}$ .

## 16.4 Methods of Connecting LCD Booster Circuit

### 16.4.1 Method of connecting booster circuit by using a regulator

If VDD is not stable because it is battery-driven, etc., we recommend a connection method using a regulator as shown below in order to preserve the quality of display.



(a) Example of voltage step-up operation by a regulator (relative V1 pin)

(b) Example of voltage step-up operation by a regulator (relative V2 pin)

Note:  $C = 0.1$  to  $0.47 \mu\text{F}$

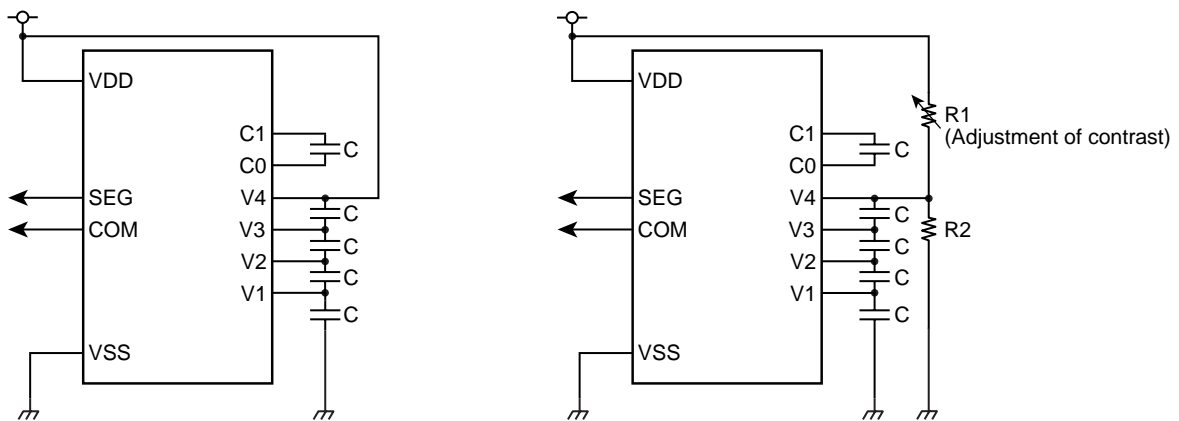
Note: For use with  $VDD \geq V4$  (LCDCTL1 <REFV> = 0), always make sure the reference power supply is entered from V4.

Figure 16-2 Method of Connecting Booster Circuit by Using a Regulator

### 16.4.2 Method of connecting booster circuit without using a regulator

If stable VDD supply is achieved ( $VDD \geq V4$ ), the booster circuit can be connected without using a regulator as shown below. In this case, set LCDCTL1 <REFV> to "0" and make sure the reference power supply is entered from the V4 pin.

Note: When LCDCTL1<REFV> is set to "0", input voltage from V4 pin should be kept within  $2.7 \text{ V} \leq V4 \leq VDD$ .



(c) Example of voltage division from VDD (relative to V4 pin)

(d) Example of voltage division from VDD (relative to V4 pin)

Note:  $C = 0.1$  to  $0.47 \mu\text{F}$

Figure 16-3 Method of Connecting Booster Circuit without Using a Regulator



## 16.5 LCD Display Operation

### 16.5.1 Setting display data

Display data is stored in display data area (128 bytes in addresses 0F00H to 0F7FH) provided in DBR.

Display data stored in the display data area is automatically read by hardware and sent to the LCD driver. The LCD driver generates segment and common signals according to display data and the driving method. Thus, display patterns can be changed simply by rewriting the contents of display data area in the program.

Table 16-2 shows the correspondence between display data areas and SEG/COM pins. The light comes on when display data is “1” and it goes out when “0”. Because the number of pixels that can be driven varies with the method of driving the LCD, the number of bytes in the display data area used to store display data also varies. Thus, bytes not used to store display data and data memory corresponding to addresses not connected to the LCD can be used for storing generally processed data. (See Table 16-3)

Note: Because the contents of display data area become unstable at powering on, execute the initialize routine for the initial setting.

Table 16-2 LCD Display Data Area (DBR)

0F00H	0F10H	0F20H	0F30H	0F40H	0F50H	0F60H	0F70H	COM0
0F01H	0F11H	0F21H	0F31H	0F41H	0F51H	0F61H	0F71H	COM1
0F02H	0F12H	0F22H	0F32H	0F42H	0F52H	0F62H	0F72H	COM2
0F03H	0F13H	0F23H	0F33H	0F43H	0F53H	0F63H	0F73H	COM3
0F04H	0F14H	0F24H	0F34H	0F44H	0F54H	0F64H	0F74H	COM4
0F05H	0F15H	0F25H	0F35H	0F45H	0F55H	0F65H	0F75H	COM5
0F06H	0F16H	0F26H	0F36H	0F46H	0F56H	0F66H	0F76H	COM6
0F07H	0F17H	0F27H	0F37H	0F47H	0F57H	0F67H	0F77H	COM7
0F08H	0F18H	0F28H	0F38H	0F48H	0F58H	0F68H	0F78H	COM8
0F09H	0F19H	0F29H	0F39H	0F49H	0F59H	0F69H	0F79H	COM9
0F0AH	0F1AH	0F2AH	0F3AH	0F4AH	0F5AH	0F6AH	0F7AH	COM10
0F0BH	0F1BH	0F2BH	0F3BH	0F4BH	0F5BH	0F6BH	0F7BH	COM11
0F0CH	0F1CH	0F2CH	0F3CH	0F4CH	0F5CH	0F6CH	0F7CH	COM12
0F0DH	0F1DH	0F2DH	0F3DH	0F4DH	0F5DH	0F6DH	0F7DH	COM13
0F0EH	0F1EH	0F2EH	0F3EH	0F4EH	0F5EH	0F6EH	0F7EH	COM14
0F0FH	0F1FH	0F2FH	0F3FH	0F4FH	0F5FH	0F6FH	0F7FH	COM15
SEG7 to SEG0	SEG15 to SEG8	SEG23 to SEG16	SEG31 to SEG24	SEG39 to SEG32	SEG47 to SEG40	SEG55 to SEG48	SEG59 to SEG56	

Table 16-3 Areas Used to Store Display Data

Driving Method	COM number to be used
1/16 Duty	COM15 to COM0
1/8 Duty	COM7 to COM0
1/4 Duty	COM3 to COM0

## 16.5.2 Blanking

The LCD display can be blanked by clearing DISST to “0”. Blanking extinguishes the LCD by outputting GND level to COM/SEG pins.

If the STOP mode is entered while the LCD display is on, DISST is cleared to “0” and blanking is performed automatically. If the STOP mode is then reverted, DISST is set to “1” and display is resumed automatically.

Note: At reset, the segment dedicated pins (SEG39 to SEG0) and common output becomes GND level, whereas the I/O port/segment shared pins (P1,P3,P5 ports) output, the I/O port/common shared pins (P3,P7 ports) output become the high-impedance state. Thus, if an external reset input lasts for a significant length of time, it may affect the LCD display such as blurring.

## 16.6 Method of Controlling LCD Driver

### 16.6.1 Initial setting

The procedure of initial setting is shown below.

Example : When 60 seg × 8 com, 1/8 duty, 5 V-system LCD operates with  $f_c = 8$  MHz (at  $V_{DD} = 5$  V)

```
LD      (LCDCTL1), 10010100B    ; 1/8 duty, LCD reference voltage (VDD = V4), booster circuit enable set

Port setting          ; Set port condition for LCD related pins

LD      (LCDCTL1), 10010101B    ; LCD display enable set
```

### 16.6.2 Storing display data

Display data is normally prepared as fixed data in the program memory (ROM) and stored in the display data area by a load instruction.

Example 1: Corresponding to the connection and display using a 1/8 duty LCD shown in Figure 16-4, the Table 16-4 shows display data and Figure 16-5 shows display timing.

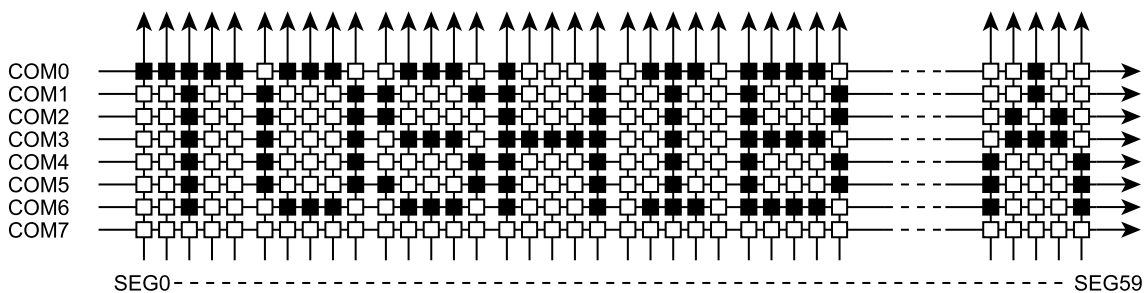


Figure 16-4 Example of Display Data (1/8 duty)

Table 16-4 Example of Display Data (1/8 duty)

	DBR	SE	SE	SE	SE	SE	SE	SE	SE	HEX	DBR	SE	SE	SE	SE	SE	SE	SE	SE	HEX	
		G	G	G	G	G	G	G	G			G	G	G	G	G	G	G	G		
		0	1	2	3	4	5	6	7			0	1	2	3	4	5	6	7		
		Bit	Bit	Bit	Bit	Bit	Bit	Bit			Bit	Bit	Bit	Bit	Bit	Bit	Bit	Bit			
COM0	0F00H	1	1	1	1	1	0	1	1	DF	0F10H	1	0	0	1	1	1	0	1	B9	...
COM1	0F01H	0	0	1	0	0	1	0	0	24	0F11H	0	1	1	0	0	0	1	1	C6	...
COM2	0F02H	0	0	1	0	0	1	0	0	24	0F12H	0	1	1	0	0	0	0	1	86	...
COM3	0F03H	0	0	1	0	0	1	0	0	24	0F13H	0	1	0	1	1	1	0	1	BA	...
COM4	0F04H	0	0	1	0	0	1	0	0	24	0F14H	0	1	0	0	0	0	1	1	C2	...
COM5	0F05H	0	0	1	0	0	1	0	0	24	0F15H	0	1	1	0	0	0	1	1	C6	...
COM6	0F06H	0	0	1	0	0	0	1	1	C4	0F16H	1	0	0	1	1	1	0	1	B9	...
COM7	0F07H	0	0	0	0	0	0	0	0	00	0F17H	0	0	0	0	0	0	0	0	00	...

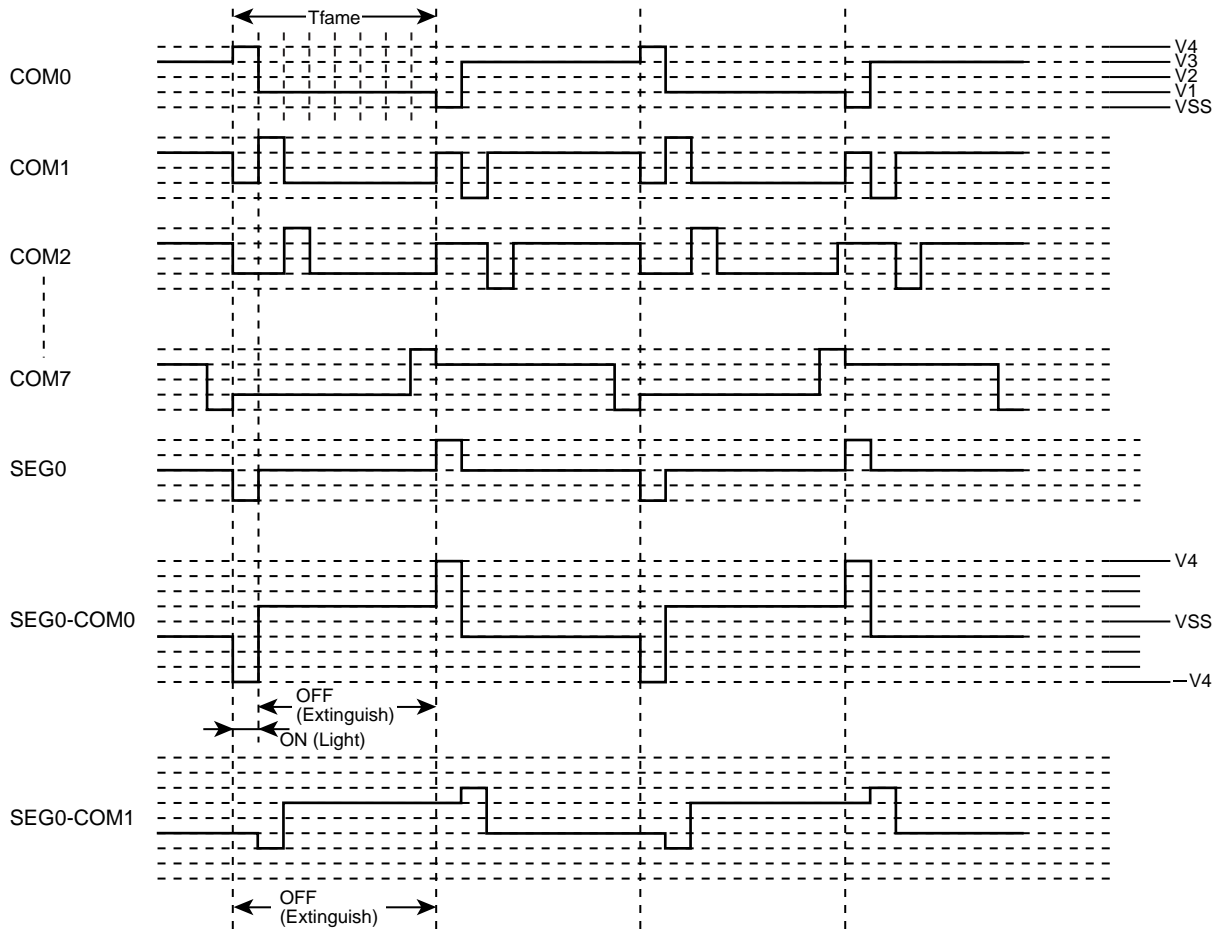


Figure 16-5 Example of Display Timing (1/8 duty)

Example 2: Corresponding to the connection and display using a 1/16 duty LCD shown in Figure 16-6, Table 16-5 shows display data and Figure 16-7 shows display timing.

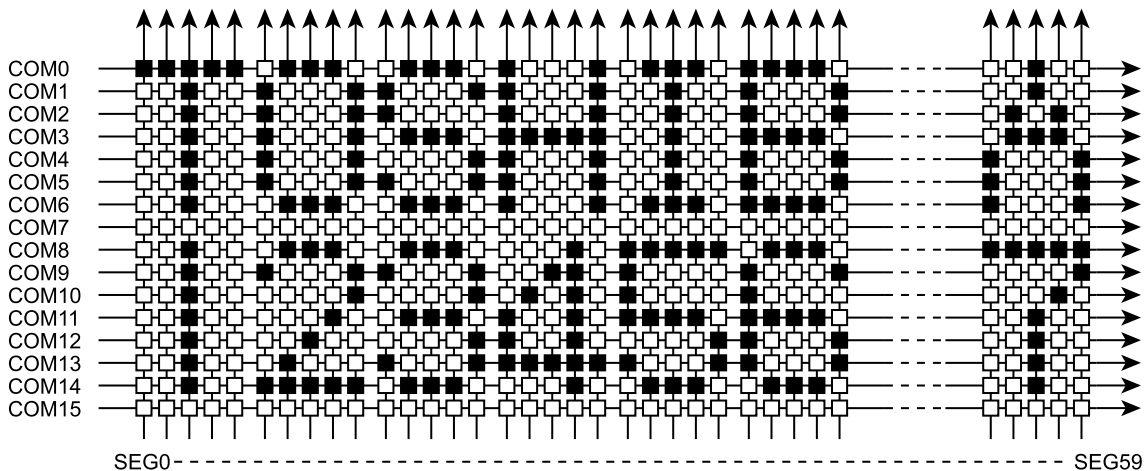


Figure 16-6 Example of Display Data (1/16 duty)

Table 16-5 Example of Display Data (1/16 duty)

	DBR	SE	SE	SE	SE	SE	SE	SE	SE	HEX	DBR	SE	SE	SE	SE	SE	SE	SE	SE	HEX	
		G0	G1	G2	G3	G4	G5	G6	G7			G8	G9	G10	G11	G12	G13	G14	G15		
		Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7			Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7		
COM0	0F00H	1	1	1	1	1	0	1	1	DF	0F10H	1	0	0	1	1	1	0	1	B9	...
COM1	0F01H	0	0	1	0	0	1	0	0	24	0F11H	0	1	1	0	0	0	1	1	C6	...
COM2	0F02H	0	0	1	0	0	1	0	0	24	0F12H	0	1	1	0	0	0	0	1	86	...
COM3	0F03H	0	0	1	0	0	1	0	0	24	0F13H	0	1	0	1	1	1	0	1	BA	...
COM4	0F04H	0	0	1	0	0	1	0	0	24	0F14H	0	1	0	0	0	0	1	1	C2	...
COM5	0F05H	0	0	1	0	0	1	0	0	24	0F15H	0	1	1	0	0	0	1	1	C6	...
COM6	0F06H	0	0	1	0	0	0	1	1	C4	0F16H	1	0	0	1	1	1	0	1	B9	...
COM7	0F07H	0	0	0	0	0	0	0	0	00	0F17H	0	0	0	0	0	0	0	0	00	...
COM8	0F08H	0	0	1	0	0	0	1	1	C4	0F18H	1	0	0	1	1	1	0	0	39	...
COM9	0F09H	0	0	1	0	0	1	0	0	24	0F19H	0	1	1	0	0	0	1	0	46	...
COM10	0F0AH	0	0	1	0	0	0	0	0	04	0F1AH	0	1	0	0	0	0	1	0	42	...
COM11	0F0BH	0	0	1	0	0	0	0	0	04	0F1BH	1	0	0	1	1	1	0	1	B9	...
COM12	0F0CH	0	0	1	0	0	0	0	1	84	0F1CH	0	0	0	0	0	0	1	1	C0	...
COM13	0F0DH	0	0	1	0	0	0	1	0	44	0F1DH	0	0	1	0	0	0	1	1	C4	...
COM14	0F0EH	0	0	1	0	0	1	1	1	E4	0F1EH	1	1	0	1	1	1	0	0	3B	...
COM15	0F0FH	0	0	0	0	0	0	0	0	00	0F1FH	0	0	0	0	0	0	0	0	00	...

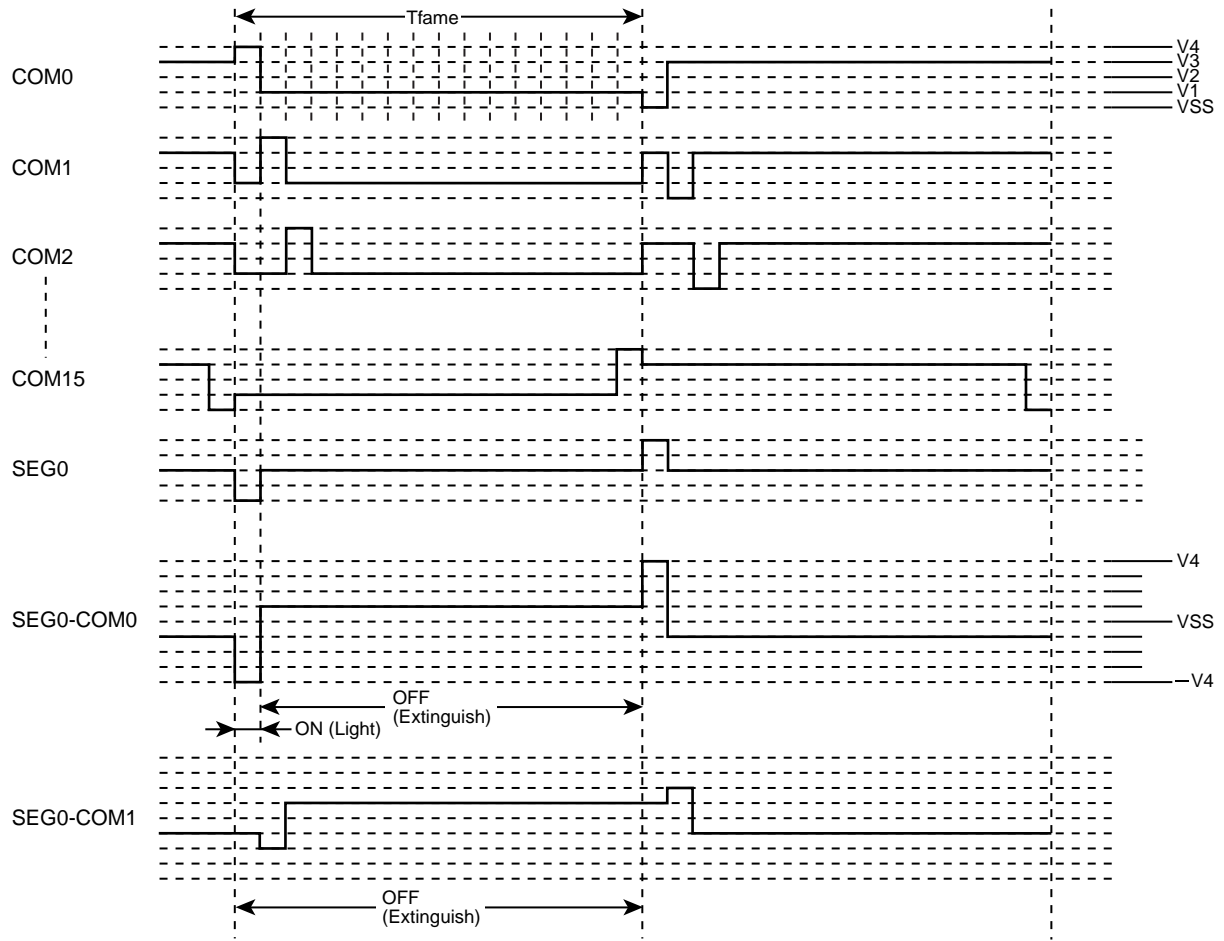


Figure 16-7 Example of Display Timing (1/16 duty)



# 17. Input/Output Circuitry

## 17.1 Control Pins

The input/output circuitries of the TMP86CS25AFG control pins are shown below.

Control Pin	I/O	Input/Output Circuitry	Remarks
XIN XOUT	Input Output		Resonator connecting pins (high-frequency) $R_f = 1.2\text{ M}\Omega$ (typ.) $R_o = 1.0\text{ k}\Omega$ (typ.)
XTIN XTOUT	Input Output		Resonator connecting pins (Low-frequency) $R_f = 6\text{ M}\Omega$ (typ.) $R_o = 220\text{ k}\Omega$ (typ.)
$\overline{\text{RESET}}$	I/O		Sink open drain output Hysteresis input Pull-up resistor $R_{IN} = 220\text{ k}\Omega$ (typ.) $R = 1\text{ k}\Omega$ (typ.)
TEST	Input		Pull-down resistor $R_{IN} = 70\text{ k}\Omega$ (typ.) $R = 1\text{ k}\Omega$ (typ.)

Note: The TEST pin of the TMP86PS25 does not have a pull-down resistor and protect diode ( $D_1$ ). Fix the TEST pin at low-level.

## 17.2 Input/Output Ports

Port	I/O	Input/Output Circuitry	Remarks
P1 P7	I/O	<p>Initial "High-Z"</p> <p>P1LCR/P7LCR SEG output Data output Input from output latch Pin input</p>	Sink open drain output Hysteresis input R = 100 Ω (typ.)
P5	I/O	<p>Initial "High-Z"</p> <p>P5LCR SEG output Data output Input from output latch Pin input</p>	Sink open drain output R = 100 Ω (typ.)
P2	I/O	<p>Initial "High-Z"</p> <p>Data output Input from output latch Pin input</p>	Sink open drain output Hysteresis input R = 100 Ω (typ.)
P30 P31 P32 P33	I/O	<p>Initial "High-Z"</p> <p>P3LCR SEG output Data output Input from output latch Pin input</p>	Sink open drain output Hysteresis input High current output (Nch) R = 100 Ω (typ.)
P34 P35 P36	I/O	<p>Initial "High-Z"</p> <p>P3LCR COM output Data output Input from output latch Pin input</p>	Sink open drain output Hysteresis input R = 100 Ω (typ.)
P6	I/O	<p>Initial "High-Z"</p> <p>Data output Disable Pin input</p>	Tri-state I/O Hysteresis input R = 100 Ω (typ.)

Note: Port P1, P3, P5 and P7 are sink open drain output. But they are also used as a segment output of LCD. Therefore, absolute maximum ratings of port input voltage should be used in  $-0.3$  to  $V_{DD} + 0.3$  volts.



## 18. Electrical Characteristics

### 18.1 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<sub>SS</sub> = 0 V)

Parameter	Symbol	Pins	Rating	Unit
Supply Voltage	V <sub>DD</sub>		-0.3 to 6.5	V
Input Voltage	V <sub>IN</sub>		-0.3 to V <sub>DD</sub> + 0.3	V
Output Voltage	V <sub>OUT</sub>		-0.3 to V <sub>DD</sub> + 0.3	V
Output Current (Per 1 pin)	I <sub>OUT1</sub>	P6 Port	-1.8	mA
	I <sub>OUT2</sub>	P1, P2, P34 to P36, P5, P6, P7 Port	3.2	
	I <sub>OUT3</sub>	P30 to P33 Port	30	
Output Current (Total)	Σ I <sub>OUT2</sub>	P1, P2, P34 to P36, P5, P6, P7 Port	60	
	Σ I <sub>OUT3</sub>	P30 to P33 Port	80	
Power Dissipation [T <sub>opr</sub> = 85°C]	PD		350	mW
Soldering Temperature (Time)	T <sub>sld</sub>		260 (10 s)	°C
Storage Temperature	T <sub>stg</sub>		-55 to 125	
Operating Temperature	T <sub>opr</sub>		-40 to 85	

## 18.2 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.

(V<sub>SS</sub> = 0 V, Topr = -40 to 85°C)

Parameter	Symbol	Pins	Condition	Min	Max	Unit
Supply Voltage	V <sub>DD</sub>		f <sub>c</sub> = 16 MHz	NORMAL1, 2 mode	4.5	5.5
				IDLE0, 1, 2 mode		
			f <sub>c</sub> = 8 MHz	NORMAL1, 2 mode	2.7	
				IDLE0, 1, 2 mode		
			f <sub>c</sub> = 4.2 MHz	NORMAL1, 2 mode	1.8 (Note1)	
				IDLE0, 1, 2 mode		
f <sub>s</sub> = 32.768 kHz	SLOW1, 2 mode					
	SLEEP0, 1, 2 mode					
	STOP mode					
Input High Level	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
	V <sub>IH2</sub>	Hysteresis input		V <sub>DD</sub> × 0.75		
	V <sub>IH3</sub>			V <sub>DD</sub> < 4.5 V		
Input Low Level	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	
	V <sub>IL3</sub>		V <sub>DD</sub> < 4.5 V		V <sub>DD</sub> × 0.10	
LCD Reference Voltage Range	V <sub>1IN</sub>	V1	LCDCTL1<REFV> = "1" V <sub>DD</sub> < V4 <sup>#1</sup>	1.0	1.375	
	V <sub>2IN</sub>	V2		2.0	2.750	
	V <sub>3IN</sub>	V3		3.0	4.125	
	V <sub>4IN</sub>	V4		4.0	5.500	
	V <sub>4IN</sub>	V4 <sup>#2</sup>	LCDCTL1<REFV> = "0"	2.7	V <sub>DD</sub>	
Clock Frequency	f <sub>c</sub>	XIN, XOUT	V <sub>DD</sub> = 1.8 V to 5.5 V	1.0	4.2	MHz
			V <sub>DD</sub> = 2.7 V to 5.5 V		8.0	
			V <sub>DD</sub> = 4.5 V to 5.5 V		16.0	
	f <sub>s</sub>	XTIN, XTOUT	30.0		34.0	

#1 When LCDCTL1<REFV> is set to "1", always keep the condition of V<sub>DD</sub> < V4.

#2 When LCDCTL1<REFV> is set to "0", always supply the reference voltage from V4 pin.

Note 1: When the supply voltage is V<sub>DD</sub>=1.8 to 2.0V, the operating temperature is Topr= -20 to 85 °C.

### 18.3 DC Characteristics

(V<sub>SS</sub> = 0 V, Topr = -40 to 85°C)

Parameter	Symbol	Pins	Condition	Min	Typ.	Max	Unit
Hysteresis Voltage	V <sub>HS</sub>	Hysteresis input		-	0.9	-	V
Input Current	I <sub>IN1</sub>	TEST	V <sub>DD</sub> = 5.5 V V <sub>IN</sub> = 5.5 V/0 V	-	-	±2	μA
	I <sub>IN2</sub>	Sink Open Drain, Tri-state Port					
	I <sub>IN3</sub>	RESET, STOP					
Input Resistance	R <sub>IN1</sub>	TEST Pull-Down	V <sub>DD</sub> = 5.5 V, V <sub>IN</sub> = 5.5 V	-	70	-	kΩ
	R <sub>IN2</sub>	RESET Pull-Up	V <sub>DD</sub> = 5.5 V, V <sub>IN</sub> = 0 V	100	220	450	
Output Leakage Current	I <sub>LO</sub>	Sink Open Drain, Tri-state Port	V <sub>DD</sub> = 5.5 V, V <sub>OUT</sub> = 5.5 V/0 V	-	-	±2	μA
Output High Voltage	V <sub>OH2</sub>	Tri-state Port	V <sub>DD</sub> = 4.5 V, I <sub>OH</sub> = -0.7 mA	4.1	-	-	V
Output Low Voltage	V <sub>OL</sub>	Except XOUT and P30 to P33 Port	V <sub>DD</sub> = 4.5 V, I <sub>OL</sub> = 1.6 mA	-	-	0.4	
Output Low Current	I <sub>OL</sub>	High Current Port (P30 to P33 Port)	V <sub>DD</sub> = 4.5 V, V <sub>OL</sub> = 1.0 V	-	20	-	mA
Supply Current in NORMAL 1, 2 mode	I <sub>DD</sub>		V <sub>DD</sub> = 5.5 V V <sub>IN</sub> = 5.3 V/0.2 V fc = 16 MHz fs = 32.768 kHz	-	6.0	7.0	mA
Supply Current in IDLE0, 1, 2 mode				-	4.2	5.0	
Supply Current in SLOW1 mode			V <sub>DD</sub> = 3.0 V V <sub>IN</sub> = 2.8 V/0.2 V fs = 32.768 kHz	-	8.5	25	μA
Supply Current in SLEEP1 mode				-	5.0	15	
Supply Current in SLEEP0 mode				-	3.0	13	
Supply Current in STOP mode				V <sub>DD</sub> = 5.5 V V <sub>IN</sub> = 5.3 V/0.2 V	-	0.5	

Note 1: Typical values show those at Topr = 25°C, V<sub>DD</sub> = 5 V

Note 2: Input current (I<sub>IN1</sub>, I<sub>IN3</sub>): The current through pull-up or pull-down resistor is not included.

Note 3: I<sub>DD</sub> does not include I<sub>REF</sub> current.

Note 4: The supply currents in SLOW2 and SLEEP2 modes are equivalent to those in IDLE0, IDLE1, and IDLE2 modes.

## 18.4 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	$V_{AREF}$		$V_{DD} - 1.5$	–	$V_{DD}$	V
Analog Reference Voltage Range (Note 4)	$\Delta V_{AREF}$		3.0	–	–	
Analog Input Voltage	$V_{AIN}$		$V_{SS}$	–	$V_{AREF}$	
Power Supply Current of Analog Reference Voltage	$I_{REF}$	$V_{DD} = V_{AREF} = 5.5\text{ V}$ $V_{SS} = 0.0\text{ V}$	–	0.6	1.0	mA
Non linearity Error		$V_{DD} = 5.0\text{ V}$ , $V_{SS} = 0.0\text{ V}$ $V_{AREF} = 5.0\text{ V}$	–	–	$\pm 1$	LSB
Zero Point Error			–	–	$\pm 1$	
Full Scale Error			–	–	$\pm 1$	
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	$V_{AREF}$		$V_{DD} - 1.5$	–	$V_{DD}$	V
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	$I_{REF}$	$V_{DD} = V_{AREF} = 4.5\text{ V}$ $V_{SS} = 0.0\text{ V}$	–	0.5	0.8	mA
Non linearity Error		$V_{DD} = 2.7\text{ V}$ , $V_{SS} = 0.0\text{ V}$ $V_{AREF} = 2.7\text{ V}$	–	–	$\pm 1$	LSB
Zero Point Error			–	–	$\pm 1$	
Full Scale Error			–	–	$\pm 1$	
Total Error			–	–	$\pm 2$	

( $V_{SS} = 0.0\text{ V}$ ,  $2.0\text{ V} \leq V_{DD} < 2.7\text{ V}$ ,  $T_{opr} = -40\text{ to }85^{\circ}\text{C}$ ) (Note 5)

( $V_{SS} = 0.0\text{ V}$ ,  $1.8\text{ V} \leq V_{DD} < 2.0\text{ V}$ ,  $T_{opr} = -10\text{ to }85^{\circ}\text{C}$ ) (Note 5)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Analog Reference Voltage	$V_{AREF}$		$V_{DD} - 0.9$	–	$V_{DD}$	V
Analog Reference Voltage Range (Note 4)	$\Delta V_{AREF}$	$1.8\text{ V} \leq V_{DD} < 2.0\text{ V}$	1.8	–	–	
		$2.0\text{ V} \leq V_{DD} < 2.7\text{ V}$	2.0	–	–	
Analog Input Voltage	$V_{AIN}$		$V_{SS}$	–	$V_{AREF}$	
Power Supply Current of Analog Reference Voltage	$I_{REF}$	$V_{DD} = V_{AREF} = 2.7\text{ V}$ $V_{SS} = 0.0\text{ V}$	–	0.3	0.5	mA
Non linearity Error		$V_{DD} = 1.8\text{ V}$ , $V_{SS} = 0.0\text{ V}$ $V_{AREF} = 1.8\text{ V}$	–	–	$\pm 2$	LSB
Zero Point Error			–	–	$\pm 2$	
Full Scale Error			–	–	$\pm 2$	
Total Error			–	–	$\pm 4$	

Note 1: The total error includes all errors except a quantization error, and is defined as maximum deviation from the ideal conversion line.

Note 2: Conversion time is different in recommended value by power supply voltage. About conversion time, refer to “8-bit AD converter(ADC)”.

Note 3: Please use input voltage to AIN input Pin in limit of  $V_{AREF} - V_{SS}$ .

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: When AD is used with  $V_{DD} < 2.7\text{ V}$ , the guaranteed temperature range varies with the operating voltage.

### 18.5 AC Characteristics

(V<sub>SS</sub> = 0 V, V<sub>DD</sub> = 4.5 to 5.5 V, Topr = -40 to 85°C)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Machine Cycle Time	t <sub>cy</sub>	NORMAL1, 2 mode	0.25	-	4	μs
		IDLE0, 1, 2 mode				
		SLOW1, 2 mode	117.6	-	133.3	
		SLEEP0, 1, 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<sub>SS</sub> = 0 V, V<sub>DD</sub> = 2.7 to 4.5 V, Topr = -40 to 85°C)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Machine Cycle Time	t <sub>cy</sub>	NORMAL1, 2 mode	0.5	-	4	μs
		IDLE0, 1, 2 mode				
		SLOW1, 2 mode	117.6	-	133.3	
		SLEEP0, 1, 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				

(V<sub>SS</sub> = 0 V, V<sub>DD</sub> = 1.8 to 2.7 V, Topr = -40 to 85°C)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Machine Cycle Time	t <sub>cy</sub>	NORMAL1, 2 mode	0.95	-	4	μs
		IDLE0, 1, 2 mode				
		SLOW1, 2 mode	117.6	-	133.3	
		SLEEP0, 1, 2 mode				
High Level Clock Pulse Width	t <sub>WCH</sub>	For external clock operation (XIN input)	-	119.05	-	ns
Low Level Clock Pulse Width	t <sub>WCL</sub>	fc = 4.2 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				

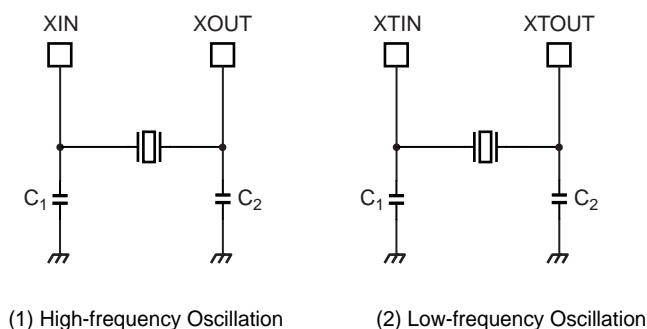
Note 1: When the supply voltage is V<sub>DD</sub>=1.8 to 2.0V, the operating temperature is Topr= -20 to 85 °C.

### 18.6 Timer Counter 1 input (ECIN) Characteristics

(V<sub>SS</sub> = 0 V, Topr = -40 to 85°C)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit	
TC1 input (ECIN input)	t <sub>TC1</sub>	Frequency measurement mode V <sub>DD</sub> = 4.5 to 5.5 V	Single edge count	-	-	1.0	MHz
		Frequency measurement mode V <sub>DD</sub> = 2.7 to 4.5 V	Single edge count	-	-	0.5	
		Frequency measurement mode V <sub>DD</sub> = 1.8 to 2.7 V	Single edge count	-	-	0.262	

## 18.7 Recommended Oscillating Conditions



Note 1: A quartz resonator can be used for high-frequency oscillation only when  $V_{DD}$  is 2.7 V or above. If  $V_{DD}$  is below 2.7 V, use a ceramic resonator.

Note 2: 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 3: 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>

## 18.8 Handling Precaution

- The solderability test conditions for lead-free products (indicated by the suffix G in product name) are shown below.
  1. When using the Sn-37Pb solder bath
    - Solder bath temperature = 230 °C
    - Dipping time = 5 seconds
    - Number of times = once
    - R-type flux used
  2. When using the Sn-3.0Ag-0.5Cu solder bath
    - Solder bath temperature = 245 °C
    - Dipping time = 5 seconds
    - Number of times = once
    - R-type flux used

Note: The pass criterion of the above test is as follows:

Solderability rate until forming  $\geq 95\%$

- When using the device (oscillator) in places exposed to high electric fields such as cathode-ray tubes, we recommend electrically shielding the package in order to maintain normal operating condition.







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.

