

**TOSHIBA**

8 Bit Microcontroller  
TLCS-870/C Series

**TMP86FH92DMG**

**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 patents or other rights of TOSHIBA or the third parties. 070122\_C

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

## Precaution for Using the Emulation Chip / Difference among Products

### (1) Precaution for Using the Emulation Chip (Development Tool)

- Precaution for debugging the voltage detection circuit

The functions of the voltage detection circuit vary between the TMP86FH92DMG and the emulation chip TMP86C993XB. Therefore, please ensure that the final verification of software operation related to the voltage detection circuit is conducted with the TMP86FH92DMG.

For details, refer to the chapter on the voltage detection circuit.

- Precaution for debugging the power-on reset circuit

The power-on reset circuit cannot be emulated with the TMP86C993XB. Therefore, when using the development tool for debugging, ensure that operation is performed within the operating voltage range of the TMP86FH92DMG. For the operating voltage range, refer to the chapter on electrical characteristics.

- Precaution for debugging the Flash control register

Although the TMP86FH92DMG contains the Flash control register (FLSCR) at 0FFFH in the DBR area, the TMP86C993XB do not contain the FLSCR register. Therefore, when using the development tool for debugging, a program that accesses the FLSCR register cannot function properly (executes differently as in the case of TMP86FH92DMG).

---

(2) Difference among Products

• Differences in Functions

Products	TMP86FH92DMG	TMP86FH93NG
CPU CORE	TLCS-870/C	
ROM	16K bytes (FLASH)	
RAM	512 bytes	
Interrupt	22 interrupts (External:5 Internal:17)	
I/O	24 pins	26 pins
Port 0	8pins(Large current output/sink open-drain or C-MOS output/with Programmable Pull-up resistance)	
Port 1	5 pins	7pins
	(sink-opendrain or C-MOS output/with Programmable Pull-up resistance)	
Port 2	3 pins (P20 is addition programmable Pull-up resistance)	
Port 3	8 pins	
Watchdog Timer	1 channel	
Timer/counter	16 bit: 1 channel	
	8 bit: 2 channels	
UART	2 channels (1 channel is shared with I <sup>2</sup> C bus)	2 channels
Serial bus interface (I <sup>2</sup> C bus)	1 channel (shared with UART) SDA(P13) and SCL(P14) are fixed	1 channel SDA(P13) and SCL(P14) or SDA(P15) and SCL(P16) are selectable
SEI	1 channel	
10 bit ADconverter	6 channels	
Key-on wake-up	4 channels	
Clock Oscillation circuit	2 circuits (single / dual clock modes are selectable)	
Low power consumption operating	9 modes (STOP/SLOW1/SLOW2/IDLE0/IDLE1/IDLE2/SLEEP0/SLEEP1/SLEEP2)	
otehr functions	Power on reset circuit Low voltage detector circuit	
Operating voltage(VDD)	4.0V to 5.5V (at 16MHz / 32.768KHz) 2.7V to 5.5V (at 8MHz / 32.768KHz)	
Package	30pin (SSOP30-P-56-0.65)	32pin (SDIP32-P-400-1.78)

• Difference in Electrical Characteristics

		TMP86FH92DMG	TMP86FH93NG
Operating condition (MCU mode)	Read/Fetch	3.0V to 5.5V (-40 to 85°C) 2.7V to 3.0V (-20 to 85°C)	
	Erase/Program	4.5V to 5.5V (-10 to 40°C)	
Operating condition (Serial PROM mode)		4.5V to 5.5V (-10 to 40°C)	

## Revision History

Date	Revision	
2007/4/19	1	First Release
2007/5/17	2	Contents Revised
2007/6/26	3	Contents Revised
2008/1/31	4	Contents Revised
2008/2/27	5	Contents Revised
2008/9/26	6	Contents Revised



# Table of Contents

---

---

## Precaution for Using the Emulation Chip / Difference among Products

---

---

### TMP86FH92DMG

---

---

<b>1.1 Features</b> .....	1
<b>1.2 Pin Assignment</b> .....	3
<b>1.3 Block Diagram</b> .....	4
<b>1.4 Pin Names and Functions</b> .....	5

---

---

## 2. Operational Description

---

---

<b>2.1 CPU Core Functions</b> .....	7
2.1.1 Memory Address Map .....	7
2.1.2 Program Memory (Flash) .....	7
2.1.3 Data Memory (RAM) .....	7
<b>2.2 System Clock Controller</b> .....	8
2.2.1 Clock Generator .....	8
2.2.2 Timing Generator .....	9
2.2.2.1 Configuration of timing generator	
2.2.2.2 Machine cycle	
2.2.3 Operation Mode Control Circuit .....	11
2.2.3.1 Single-clock mode	
2.2.3.2 Dual-clock mode	
2.2.3.3 STOP mode	
2.2.3.4 Operating Mode Transition	
2.2.4 Operating Mode Control .....	16
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	
<b>2.3 Reset Circuit</b> .....	30
2.3.1 External Reset Input .....	30
2.3.2 Address trap reset .....	31
2.3.3 Watchdog timer reset .....	31
2.3.4 System clock reset .....	31
2.3.5 Power-on Reset.....	32
2.3.6 Voltage detection reset.....	32
2.3.7 Trimming data reset.....	32
<b>2.4 Internal Reset Detection Flags</b> .....	33

---

---

## 3. Interrupt Control Circuit

---

---

<b>3.1 Interrupt latches (IL21 to IL2)</b> .....	36
<b>3.2 Interrupt enable register (EIR)</b> .....	36
3.2.1 Interrupt master enable flag (IMF) .....	36
3.2.2 Individual interrupt enable flags (EF21 to EF4) .....	37
<b>3.3 Interrupt Sequence</b> .....	38
3.3.1 Interrupt acceptance processing is packaged as follows. ....	38

3.3.2	Saving/restoring general-purpose registers .....	39
3.3.2.1	Using PUSH and POP instructions .....	
3.3.2.2	Using data transfer instructions .....	
3.3.3	Interrupt return .....	42
<b>3.4</b>	<b>Software Interrupt (INTSW) .....</b>	<b>42</b>
3.4.1	Address error detection .....	42
3.4.2	Debugging .....	43
<b>3.5</b>	<b>Undefined Instruction Interrupt (INTUNDEF) .....</b>	<b>43</b>
<b>3.6</b>	<b>Address Trap Interrupt (INTATRAP) .....</b>	<b>43</b>
<b>3.7</b>	<b>External Interrupts .....</b>	<b>44</b>

---

## 4. Special Function Register (SFR)

---

<b>4.1</b>	<b>SFR .....</b>	<b>47</b>
<b>4.2</b>	<b>DBR .....</b>	<b>49</b>

---

## 5. I/O Ports

---

<b>5.1</b>	<b>P0 (P07 to P00) Port (High Current) .....</b>	<b>52</b>
<b>5.2</b>	<b>P1 (P14 to P10) Port .....</b>	<b>54</b>
<b>5.3</b>	<b>P2 (P22 to P20) Port .....</b>	<b>56</b>
<b>5.4</b>	<b>P3 (P37 to P30) Port .....</b>	<b>58</b>

---

## 6. Power-on reset circuit

---

<b>6.1</b>	<b>Power-on reset circuit .....</b>	<b>61</b>
6.1.1	Configuration .....	61
6.1.2	function .....	61

---

## 7. Voltage detection circuit (VLTD)

---

<b>7.1</b>	<b>Configuration .....</b>	<b>63</b>
<b>7.2</b>	<b>Control .....</b>	<b>64</b>
<b>7.3</b>	<b>Function .....</b>	<b>66</b>
7.3.1	Enabling/Disabling Voltage Detection Operation .....	66
7.3.2	Selecting the Voltage Detect Operating Mode .....	66
7.3.3	Detection voltage level section .....	66
7.3.4	Voltage detection flag and voltage detection status flag .....	66
<b>7.4</b>	<b>Setting of register .....</b>	<b>67</b>
7.4.1	Setting Procedure for Generate an interrupt .....	67
7.4.2	Setting procedure to generate a reset .....	68

---

## 8. Watchdog Timer (WDT)

---

<b>8.1</b>	<b>Watchdog Timer Configuration .....</b>	<b>69</b>
<b>8.2</b>	<b>Watchdog Timer Control .....</b>	<b>70</b>
8.2.1	Malfunction Detection Methods Using the Watchdog Timer .....	70
8.2.2	Watchdog Timer Enable .....	71
8.2.3	Watchdog Timer Disable .....	72
8.2.4	Watchdog Timer Interrupt (INTWDT) .....	72



8.2.5	Watchdog Timer Reset .....	73
<b>8.3</b>	<b>Address Trap .....</b>	<b>74</b>
8.3.1	Selection of Address Trap in Internal RAM (ATAS) .....	74
8.3.2	Selection of Operation at Address Trap (ATOUT) .....	74
8.3.3	Address Trap Interrupt (INTATRAP).....	74
8.3.4	Address Trap Reset.....	75

---

## 9. Time Base Timer (TBT)

---

<b>9.1</b>	<b>Time Base Timer.....</b>	<b>77</b>
9.1.1	Configuration.....	77
9.1.2	Control.....	77
9.1.3	Function.....	78
<b>9.2</b>	<b>Divider Output (DVO).....</b>	<b>79</b>
9.2.1	Configuration.....	79
9.2.2	Control.....	79

---

## 10. 16-Bit Timer/Counter 1 (TC1)

---

<b>10.1</b>	<b>Configuration.....</b>	<b>81</b>
<b>10.2</b>	<b>Timer/Counter Control.....</b>	<b>82</b>
<b>10.3</b>	<b>Function.....</b>	<b>84</b>
10.3.1	Timer mode.....	84
10.3.2	External Trigger Timer Mode.....	86
10.3.3	Event Counter Mode.....	88
10.3.4	Window Mode.....	89
10.3.5	Pulse Width Measurement Mode.....	90
10.3.6	Programmable Pulse Generate (PPG) Output Mode.....	93

---

## 11. 8-Bit TimerCounter (TC3, TC4)

---

<b>11.1</b>	<b>Configuration .....</b>	<b>97</b>
<b>11.2</b>	<b>TimerCounter Control.....</b>	<b>98</b>
<b>11.3</b>	<b>Function.....</b>	<b>103</b>
11.3.1	8-Bit Timer Mode (TC3 and 4).....	103
11.3.2	8-Bit Event Counter Mode (TC3, 4).....	104
11.3.3	8-Bit Programmable Divider Output (PDO) Mode (TC3, 4).....	104
11.3.4	8-Bit Pulse Width Modulation (PWM) Output Mode (TC3, 4).....	107
11.3.5	16-Bit Timer Mode (TC3 and 4).....	109
11.3.6	16-Bit Event Counter Mode (TC3 and 4).....	110
11.3.7	16-Bit Pulse Width Modulation (PWM) Output Mode (TC3 and 4).....	110
11.3.8	16-Bit Programmable Pulse Generate (PPG) Output Mode (TC3 and 4).....	113
11.3.9	Warm-Up Counter Mode.....	115
11.3.9.1	Low-Frequency Warm-up Counter Mode (NORMAL1 → NORMAL2 → SLOW2 → SLOW1)	
11.3.9.2	High-Frequency Warm-Up Counter Mode (SLOW1 → SLOW2 → NORMAL2 → NORMAL1)	

---

## 12. Asynchronous Serial interface (UART1)

---

<b>12.1</b>	<b>Configuration .....</b>	<b>117</b>
<b>12.2</b>	<b>Control .....</b>	<b>118</b>
<b>12.3</b>	<b>Transfer Data Format.....</b>	<b>121</b>
<b>12.4</b>	<b>Transfer Rate.....</b>	<b>122</b>

<b>12.5</b>	<b>Data Sampling Method</b> .....	122
<b>12.6</b>	<b>STOP Bit Length</b> .....	123
<b>12.7</b>	<b>Parity</b> .....	123
<b>12.8</b>	<b>Transmit/Receive Operation</b> .....	123
12.8.1	Data Transmit Operation.....	123
12.8.2	Data Receive Operation.....	123
<b>12.9</b>	<b>Status Flag</b> .....	124
12.9.1	Parity Error.....	124
12.9.2	Framing Error.....	124
12.9.3	Overrun Error.....	124
12.9.4	Receive Data Buffer Full.....	125
12.9.5	Transmit Data Buffer Empty.....	125
12.9.6	Transmit End Flag.....	126

---

## 13. Asynchronous Serial interface (UART2)

---

<b>13.1</b>	<b>Configuration</b> .....	127
<b>13.2</b>	<b>Control</b> .....	128
<b>13.3</b>	<b>Transfer Data Format</b> .....	131
<b>13.4</b>	<b>Transfer Rate</b> .....	132
<b>13.5</b>	<b>Data Sampling Method</b> .....	132
<b>13.6</b>	<b>STOP Bit Length</b> .....	133
<b>13.7</b>	<b>Parity</b> .....	133
<b>13.8</b>	<b>Transmit/Receive Operation</b> .....	133
13.8.1	Data Transmit Operation.....	133
13.8.2	Data Receive Operation.....	133
<b>13.9</b>	<b>Status Flag</b> .....	134
13.9.1	Parity Error.....	134
13.9.2	Framing Error.....	134
13.9.3	Overrun Error.....	134
13.9.4	Receive Data Buffer Full.....	135
13.9.5	Transmit Data Buffer Empty.....	135
13.9.6	Transmit End Flag.....	136

---

## 14. Serial Expansion Interface (SEI)

---

<b>14.1</b>	<b>Features</b> .....	137
<b>14.2</b>	<b>SEI Registers</b> .....	138
14.2.1	SEI Control Register (SECR).....	138
14.2.1.1	Transfer rate .....	
14.2.2	SEI Status Register (SESR).....	139
14.2.3	SEI Data Register (SEDR).....	139
<b>14.3</b>	<b>SEI Operation</b> .....	140
14.3.1	Controlling SEI clock polarity and phase .....	140
14.3.2	SEI data and clock timing .....	140
<b>14.4</b>	<b>SEI Pin Functions</b> .....	141
14.4.1	SCLK pin .....	141
14.4.2	MISO/MOSI pins .....	141
14.4.3	SS pin .....	141
<b>14.5</b>	<b>SEI Transfer Formats</b> .....	142
14.5.1	CPHA (SECR register bit 2) = 0 format .....	142
14.5.2	CPHA = 1 format .....	143
<b>14.6</b>	<b>Functional Description</b> .....	144
<b>14.7</b>	<b>Interrupt Generation</b> .....	145
<b>14.8</b>	<b>SEI System Errors</b> .....	145

14.8.1	Write collision error.....	145
14.8.2	Overflow error .....	145
14.8.3	Mode fault error .....	146
<b>14.9</b>	<b>Bus Driver Protection .....</b>	<b>146</b>

---

## 15. Serial Bus Interface(I<sup>2</sup>C Bus) Ver.-D (SBI)

---

<b>15.1</b>	<b>Configuration .....</b>	<b>147</b>
<b>15.2</b>	<b>Control .....</b>	<b>147</b>
<b>15.3</b>	<b>Software Reset.....</b>	<b>147</b>
<b>15.4</b>	<b>The Data Format in the I2C Bus Mode .....</b>	<b>148</b>
<b>15.5</b>	<b>I2C Bus Control.....</b>	<b>149</b>
15.5.1	Acknowledgement mode specification .....	151
15.5.1.1	Acknowledgment mode (ACK = "1") .....	
15.5.1.2	Non-acknowledgment mode (ACK = "0") .....	
15.5.2	Number of transfer bits.....	152
15.5.3	Serial clock .....	152
15.5.3.1	Clock source .....	
15.5.3.2	Clock synchronization .....	
15.5.4	Slave address and address recognition mode specification.....	154
15.5.5	Master/slave selection.....	154
15.5.6	Transmitter/receiver selection.....	154
15.5.7	Start/stop condition generation.....	154
15.5.8	Interrupt service request and cancel.....	155
15.5.9	Setting of I2C bus mode.....	156
15.5.10	Arbitration lost detection monitor.....	156
15.5.11	Slave address match detection monitor.....	157
15.5.12	GENERAL CALL detection monitor.....	157
15.5.13	Last received bit monitor.....	157
<b>15.6</b>	<b>Data Transfer of I2C Bus.....</b>	<b>157</b>
15.6.1	Device initialization.....	157
15.6.2	Start condition and slave address generation.....	158
15.6.3	1-word data transfer.....	158
15.6.3.1	When the MST is "1" (Master mode) .....	
15.6.3.2	When the MST is "0" (Slave mode) .....	
15.6.4	Stop condition generation.....	161
15.6.5	Restart.....	162

---

## 16. 10-bit AD Converter (ADC)

---

<b>16.1</b>	<b>Configuration .....</b>	<b>163</b>
<b>16.2</b>	<b>Register configuration.....</b>	<b>164</b>
<b>16.3</b>	<b>Function.....</b>	<b>167</b>
16.3.1	Software Start Mode.....	167
16.3.2	Repeat Mode.....	167
16.3.3	Register Setting.....	168
<b>16.4</b>	<b>STOP/SLOW Modes during AD Conversion.....</b>	<b>169</b>
<b>16.5</b>	<b>Analog Input Voltage and AD Conversion Result.....</b>	<b>170</b>
<b>16.6</b>	<b>Precautions about AD Converter.....</b>	<b>171</b>
16.6.1	Analog input pin voltage range.....	171
16.6.2	Analog input shared pins.....	171
16.6.3	Noise Countermeasure.....	171

---

## 17. Key-on Wakeup (KWU)

---

<b>17.1</b>	<b>Configuration.....</b>	<b>173</b>
-------------	---------------------------	------------

---

<b>17.2 Control</b> .....	174
---------------------------	-----

---

---

## **18. Flash Memory**

---

<b>18.1 Flash Memory Control</b> .....	176
18.1.1 Flash Memory Command Sequence Execution Control (FLSCR<FLSMD>).....	176
18.1.2 Flash Memory Standby Control (FLSSTB<FSTB>).....	176
<b>18.2 Command Sequence</b> .....	178
18.2.1 Byte Program.....	178
18.2.2 Sector Erase (4-kbyte Erase).....	178
18.2.3 Chip Erase (All Erase).....	179
18.2.4 Product ID Entry.....	179
18.2.5 Product ID Exit.....	179
18.2.6 Security Program.....	179
<b>18.3 Toggle Bit (D6)</b> .....	180
<b>18.4 Access to the Flash Memory Area</b> .....	181
18.4.1 Flash Memory Control in the Serial PROM Mode.....	181
18.4.1.1 How to write to the flash memory by executing the control program in the RAM area (in the RAM loader mode within the serial PROM mode)	
18.4.2 Flash Memory Control in the MCU mode.....	183
18.4.2.1 How to write to the flash memory by executing a user write control program in the RAM area (in the MCU mode)	

---

---

## **19. Serial PROM Mode**

---

<b>19.1 Outline</b> .....	185
<b>19.2 Memory Mapping</b> .....	185
<b>19.3 Serial PROM Mode Setting</b> .....	186
19.3.1 Serial PROM Mode Control Pins.....	186
19.3.2 Pin Function.....	186
19.3.3 Example Connection for On-Board Writing.....	187
19.3.4 Activating the Serial PROM Mode.....	188
<b>19.4 Interface Specifications for UART</b> .....	189
<b>19.5 Operation Command</b> .....	190
<b>19.6 Operation Mode</b> .....	190
19.6.1 Flash Memory Erasing Mode (Operating command: F0H).....	192
19.6.2 Flash Memory Writing Mode (Operation command: 30H).....	194
19.6.3 RAM Loader Mode (Operation Command: 60H).....	197
19.6.4 Flash Memory SUM Output Mode (Operation Command: 90H).....	199
19.6.5 Product ID Code Output Mode (Operation Command: C0H).....	200
19.6.6 Flash Memory Status Output Mode (Operation Command: C3H).....	202
19.6.7 Flash Memory security program Setting Mode (Operation Command: FAH).....	203
<b>19.7 Error Code</b> .....	205
<b>19.8 Checksum (SUM)</b> .....	205
19.8.1 Calculation Method.....	205
19.8.2 Calculation data.....	206
<b>19.9 Intel Hex Format (Binary)</b> .....	207
<b>19.10 Passwords</b> .....	207
19.10.1 Password String.....	208
19.10.2 Handling of Password Error.....	208
19.10.3 Password Management during Program Development.....	208
<b>19.11 Product ID Code</b> .....	209
<b>19.12 Flash Memory Status Code</b> .....	209
<b>19.13 Specifying the Erasure Area</b> .....	211
<b>19.14 Port Input Control Register</b> .....	211
<b>19.15 Flowchart</b> .....	213
<b>19.16 UART Timing</b> .....	214

---

---

## 20. Input/Output Circuitry

---

20.1	Control Pins.....	215
20.2	Input/Output Ports.....	216

---

---

## 21. Electrical Characteristics

---

21.1	Absolute Maximum Ratings.....	219
21.2	Operating Conditions.....	220
21.2.1	MCU mode (Flash Programming or erasing) .....	220
21.2.2	MCU mode (Except Flash Programming or erasing) .....	220
21.2.3	Serial PROM mode.....	221
21.3	DC Characteristics .....	222
21.4	AD Conversion Characteristics.....	224
21.5	Power-on reset circuit Characteristics.....	224
21.6	Voltage detection circuit characteristics.....	225
21.7	AC Characteristics.....	226
21.8	Flash Characteristics.....	226
21.8.1	Write/Erase Characteristics.....	226
21.9	Oscillating Conditions.....	227
21.10	Handling Precaution.....	227

---

---

## 22. Package Dimensions

---



## CMOS 8-Bit Microcontroller

# TMP86FH92DMG

Product No.	ROM (FLASH)	RAM	Package	Emulation Chip
TMP86FH92DMG	16384 bytes	512 bytes	SSOP30-P-56-0.65	TMP86C993XB

## 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. 22interrupt sources (External : 5 Internal : 17)
3. Input / Output ports (24 pins)
  - Large current output: 8pins (Typ. 20mA), LED direct drive
4. Power-on reset circuit
5. Voltage detection circuit
6. Watchdog Timer
7. Prescaler
  - Time base timer
  - Divider output function
8. 16-bit timer counter: 1 ch
  - Timer, External trigger, Window, Pulse width measurement,  
Event counter, Programmable pulse generate (PPG) modes
9. 8-bit timer counter : 2 ch
  - Timer, Event counter, Programmable divider output (PDO),

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

- 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 patents or other rights of TOSHIBA or the third parties. 070122\_C
- 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

- 
- Pulse width modulation (PWM) output,
  - Programmable pulse generation (PPG),
  - 16bit mode (8bit timer 2ch combination) modes
10. 8-bit UART : 2 ch
  11. 8bit Serial Expansion Interface (SEI): 1 channel
    - (MSB/LSB selectable and max. 4Mbps at 16MHz)
  12. Serial Bus Interface(I<sup>2</sup>C Bus): 1ch
  13. 10-bit successive approximation type AD converter
    - Analog input: 6 ch
  14. Key-on wakeup : 4 channels
  15. Clock operation
    - Single clock mode
    - Dual clock mode
  16. 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<TBTCCK>.
    - IDLE1 mode: CPU stops and peripherals operate using high frequency clock. Release by interrupts(CPU restarts).
    - IDLE2 mode: CPU stops and peripherals operate using high and low frequency clock. Release by interrupts. (CPU restarts).
    - SLEEP0 mode: CPU stops, and only the Time-Based-Timer(TBT) on peripherals operate using low frequency clock. Release by falling edge of the source clock which is set by TBTCR<TBTCCK>.
    - 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.
  17. Wide operation voltage:
    - 4.0 V to 5.5 V at 16MHz /32.768 kHz
    - 2.7 V to 5.5 V at 8 MHz /32.768 kHz
-



## 1.2 Pin Assignment

VSS	1	30	P37 (AIN5/STOP5)
XIN	2	29	P36 (AIN4/STOP4)
XOUT	3	28	P35 (AIN3/STOP3)
TEST	4	27	P34 (AIN2/STOP2)
VDD	5	26	P33 (AIN1)
(XTIN) P21	6	25	P32 (AIN0)
(XTOUT) P22	7	24	P31 (TC4/PDO4/PWM4/PPG4)
RESET	8	23	P30 (TC3/PDO3/PWM3)
(INT5/STOP) P20	9	22	P12 (DVO)
(TXD1) P00	10	21	P11 (INT1)
(BOOT/RXD1) P01	11	20	P10 (INT0)
(SCLK) P02	12	19	P07 (TC1/INT4)
(MOSI) P03	13	18	P06 (INT3/PPG)
(MISO) P04	14	17	P14 (SCL/TXD2)
(SS) P05	15	16	P13 (SDA/RXD2)

Figure 1-1 Pin Assignment

### 1.3 Block Diagram

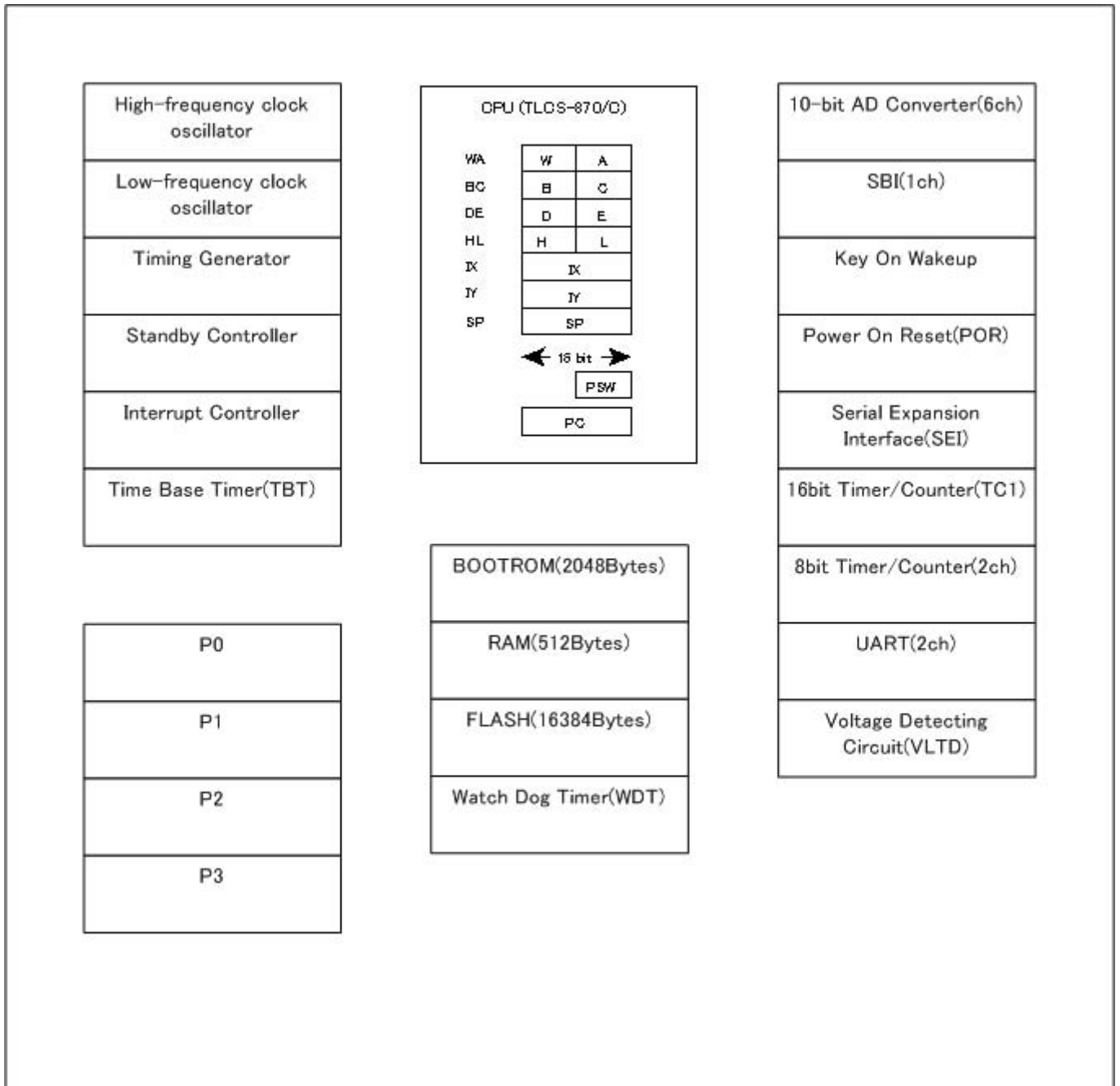


Figure 1-2 Block Diagram

## 1.4 Pin Names and Functions

The TMP86FH92DMG has MCU mode, parallel PROM mode, and serial PROM mode. Table 1-1 shows the pin functions in MCU mode. The serial PROM mode is explained later in a separate chapter.

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

Pin Name	Pin Number	Input/Output	Functions
P07 TC1 INT4	19	IO I I	PORT07 TC1 input External interrupt 4 input
P06 INT3 PPG	18	IO I O	PORT06 External interrupt 3 input PPG output
P05 SS	15	IO I	PORT05 SEI master/slave select input
P04 MISO	14	IO IO	PORT04 SEI master input, slave output
P03 MOSI	13	IO IO	PORT03 SEI master input, slave output
P02 SCLK	12	IO IO	PORT02 SEI serial clock input/output pin
P01 RXD1 BOOT	11	IO I I	PORT01 UART data input 1 Serial PROM mode control input
P00 TXD1	10	IO O	PORT00 UART data output 1
P14 SCL TXD2	17	IO IO O	PORT14 I2C bus clock UART data output 2
P13 SDA RXD2	16	IO IO I	PORT13 I2C bus data UART data input 2
P12 DVO	22	IO O	PORT12 Divider Output
P11 INT1	21	IO I	PORT11 External interrupt 1 input
P10 INT0	20	IO I	PORT10 External interrupt 0 input
P22 XTOUT	7	IO O	PORT22 Resonator connecting pins(32.768kHz) for inputting external clock
P21 XTIN	6	IO I	PORT21 Resonator connecting pins(32.768kHz) for inputting external clock
P20 STOP INT5	9	IO I I	PORT20 STOP mode release signal input External interrupt 5 input

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

Pin Name	Pin Number	Input/Output	Functions
P37 AIN5 STOP5	30	IO I I	PORT37 Analog Input5 STOP5
P36 AIN4 STOP4	29	IO I I	PORT36 Analog Input4 STOP4
P35 AIN3 STOP3	28	IO I I	PORT35 Analog Input3 STOP3
P34 AIN2 STOP2	27	IO I I	PORT34 Analog Input2 STOP2
P33 AIN1	26	IO I	PORT33 Analog Input1
P32 AIN0	25	IO I	PORT32 Analog Input0
P31 TC4 PDO4/PWM4/PPG4	24	IO I O	PORT31 TC4 input PDO4/PWM4/PPG4 output
P30 TC3 PDO3/PWM3	23	IO I O	PORT30 TC3 input PDO3/PWM3 output
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.
VDD	5	I	+5V
VSS	1	I	0(GND)

## 2. Operational Description

### 2.1 CPU Core Functions

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

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

#### 2.1.1 Memory Address Map

The TMP86FH92DMG memory is composed Flash, RAM, DBR(Data buffer register) and SFR(Special function register). They are all mapped in 64-Kbyte address space. Figure 2-1 shows the TMP86FH92DMG memory address map.

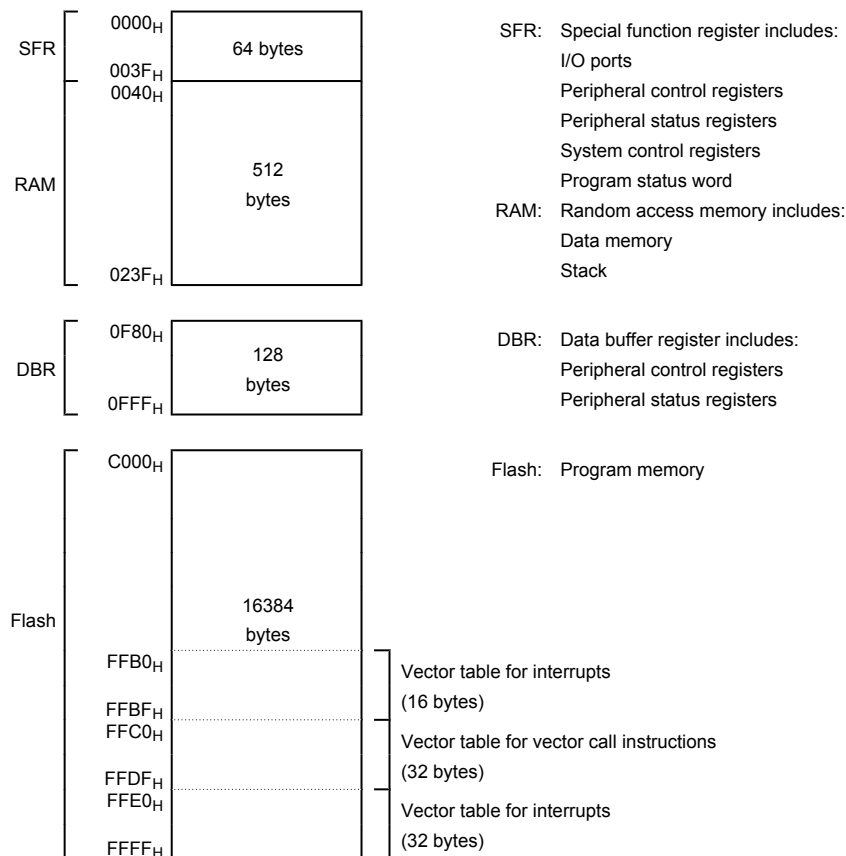


Figure 2-1 Memory Address Map

#### 2.1.2 Program Memory (Flash)

The TMP86FH92DMG has a 16384 bytes (Address C000H to FFFFH) of program memory (Flash).

#### 2.1.3 Data Memory (RAM)

The TMP86FH92DMG has 512bytes (Address 0040H to 023FH) 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”. (TMP86FH92DMG)

```

LD      HL, 0040H      ; Start address setup
LD      A, H          ; Initial value (00H) setup
LD      BC, 01FFH
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.

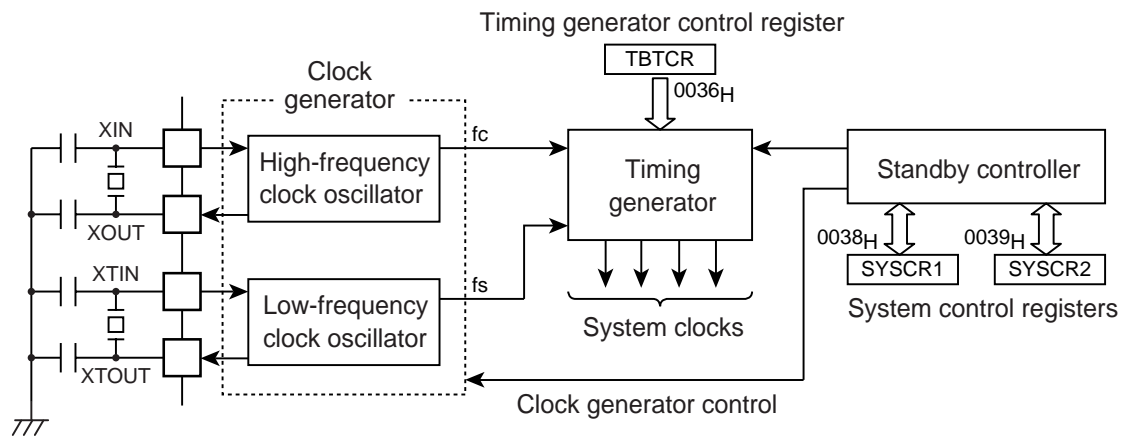


Figure 2-2 System Clock Control

### 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.

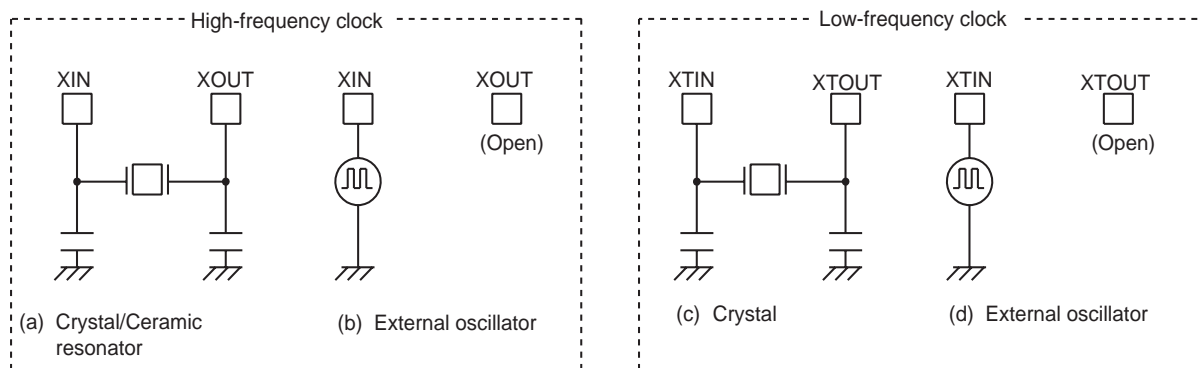


Figure 2-3 Examples of Resonator Connection

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 ( $\overline{DVO}$ ) 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

### 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,  $SYSR2<SYSCK>$  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".

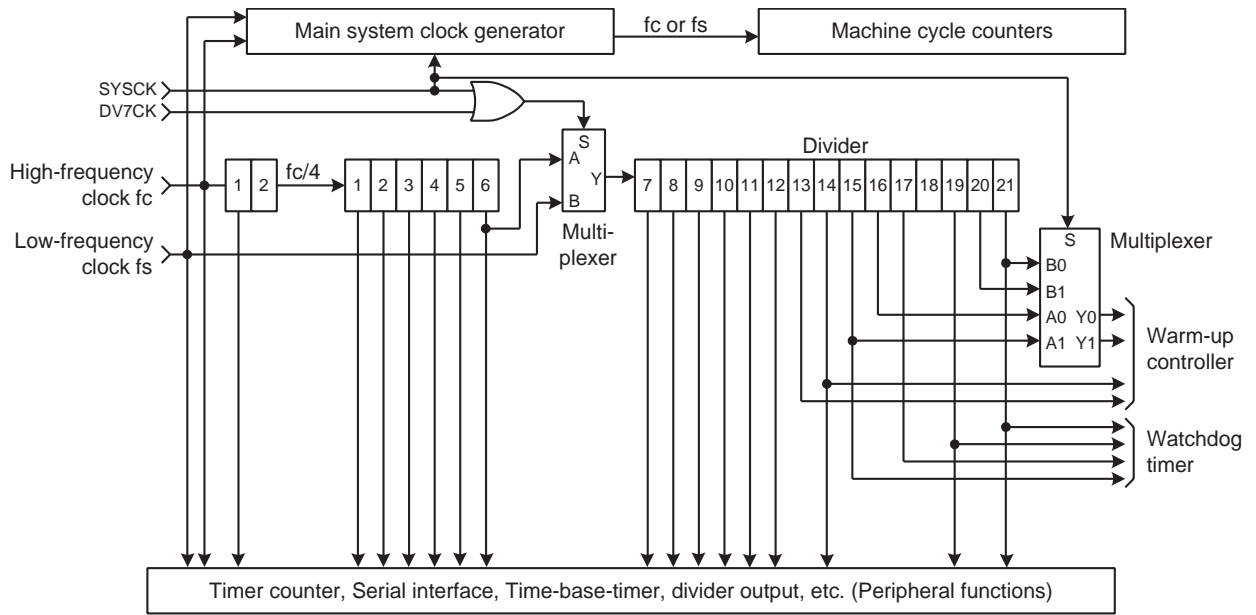


Figure 2-4 Configuration of Timing Generator



Timing Generator Control Register

TBTCR (0036H)	7	6	5	4	3	2	1	0	(Initial value: 0000 0000)
	(DVOEN)	(DVOCK)	DV7CK	(TBTEN)	(TBTCK)				
DV7CK	Selection of input to the 7th stage of the divider		0: $fc/2^8$ [Hz] 1: fs						R/W

- 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.

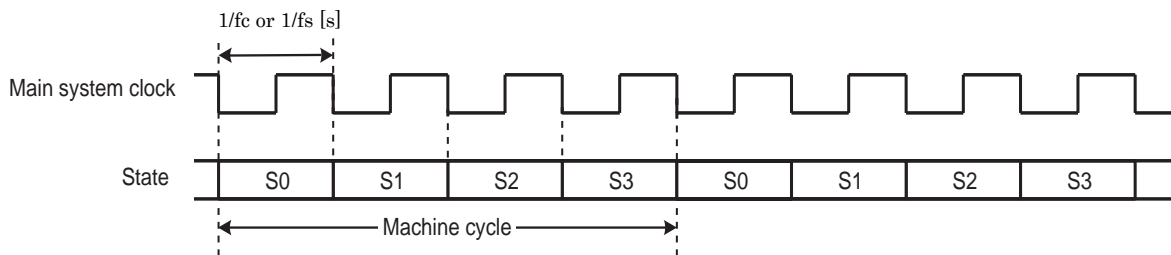


Figure 2-5 Machine Cycle

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 TMP86FH92DMG 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"`, `EF7` (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] ( $122 \mu\text{s}$  at  $f_s = 32.768 \text{ kHz}$ ) in the SLOW and SLEEP modes.

The TLCS-870/C is placed in the single-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.

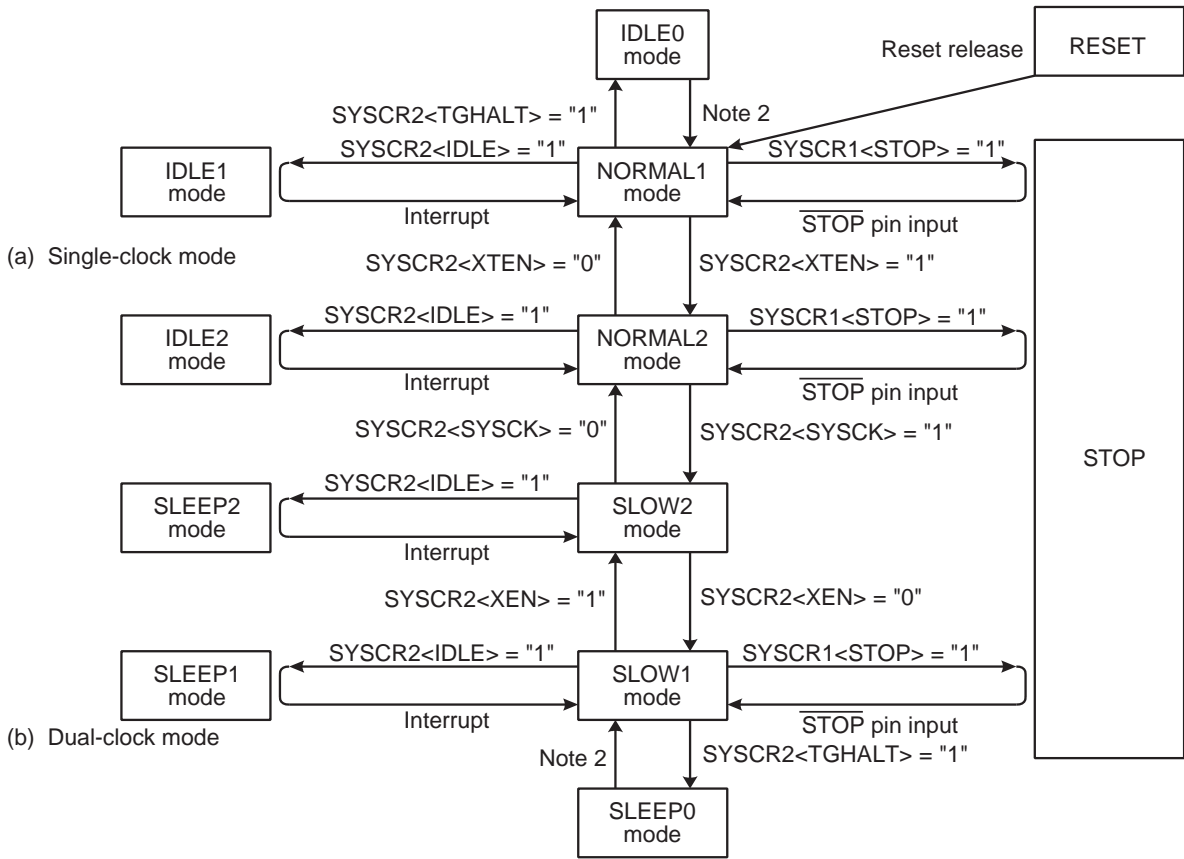
When returned from SLEEP0 mode, the CPU restarts operating, entering SLOW1 mode back again. SLEEP0 mode is entered and returned regardless of how TBTCR<TBTEN> is set. When IMF = "1", EF7 (TBT interrupt individual enable flag) = "1", and TBTCR<TBTEN> = "1", interrupt processing is performed. When SLEEP0 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 programmable selected) to the STOP pin. After the warm-up period is completed, the execution resumes with the instruction which follows the STOP mode start instruction.

2.2.3.4 Operating Mode Transition



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	WDT	TBT	AD Converter	Power-on Rest Voltage Detect Reset	Other Peripherals	Machine Cycle Time		
		High Frequency	Low Frequency									
Single clock	RESET	Oscillation	Stop	Reset	Reset	Reset	Reset	Operate	Reset	4/fc [s]		
	NORMAL1			Operate	Operate	Operate	Operate		Operate			
	IDLE1			Halt	Halt		Halt		Halt		Halt	
	IDLE0											
	STOP	Stop	Halt	Halt	Halt	Halt	-					
Dual clock	NORMAL2	Oscillation	Oscillation	Operate with High-freq.	Operate with High or Low-freq.	Operate	Operate	Operate	Operate	4/fs [s]		
	IDLE2			Halt	Halt						Operate	Halt
	SLOW2			Operate with Low-freq.	Operate with Low-freq.							
	SLEEP2			Halt	Halt							
	SLOW1	Stop	Stop	Operate with Low-freq.	Operate with Low-freq.	Halt	Halt		Halt	Halt	4/fs [s]	
	SLEEP1			Halt	Halt							
	SLEEP0											
	STOP			Stop	Halt							Halt

## 2.2.4 Operating Mode Control

### System Control Register 1

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

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	$3 \times 2^{14}/fc$	$3 \times 2^9/fs$	
		11	$2^{14}/fc$	$2^9/fs$	

- 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 0 and 1 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: In case of setting as STOP mode is released by a rising edge of  $\overline{STOP}$  pin input, the release setting by STOP5 to STOP2 on STOPCR register is prohibited.
- Note 8: 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 9: The warming-up time should be set correctly for using oscillator.

### System Control Register 2

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

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)		R/W
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)		
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.

- 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<TBTKC>.
- 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.

2.2.4.1 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 are 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 pins (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 detecting high or low edge input for the STOP5 to STOP2 pins which are 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, 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 and the STOP5 to STOP2 inputs are high. The following two methods can be used for confirmation.

1. Testing a port.
2. Using an external interrupt input 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
    
```

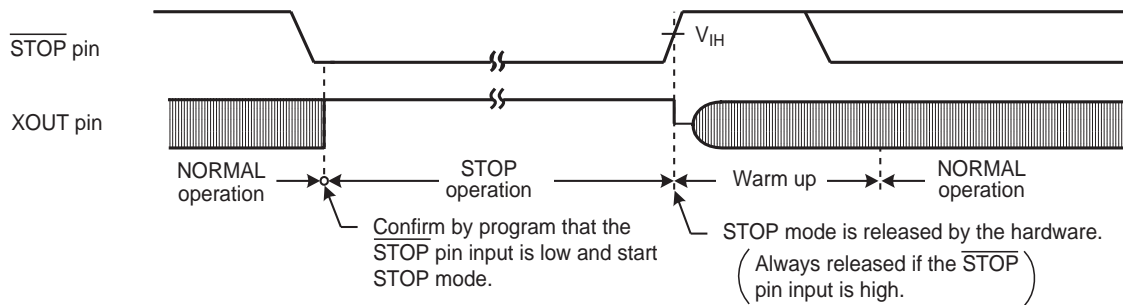


Figure 2-7 Level-sensitive Release Mode

- Note 1: Even if the  $\overline{\text{STOP}}$  pin input is low or the STOP5 to STOP2 pin inputs are high 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 inputs 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
    
```

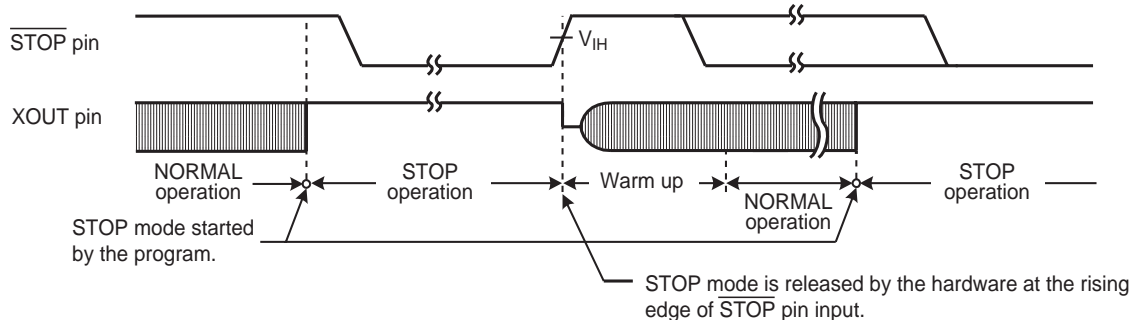


Figure 2-8 Edge-sensitive Release Mode



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.

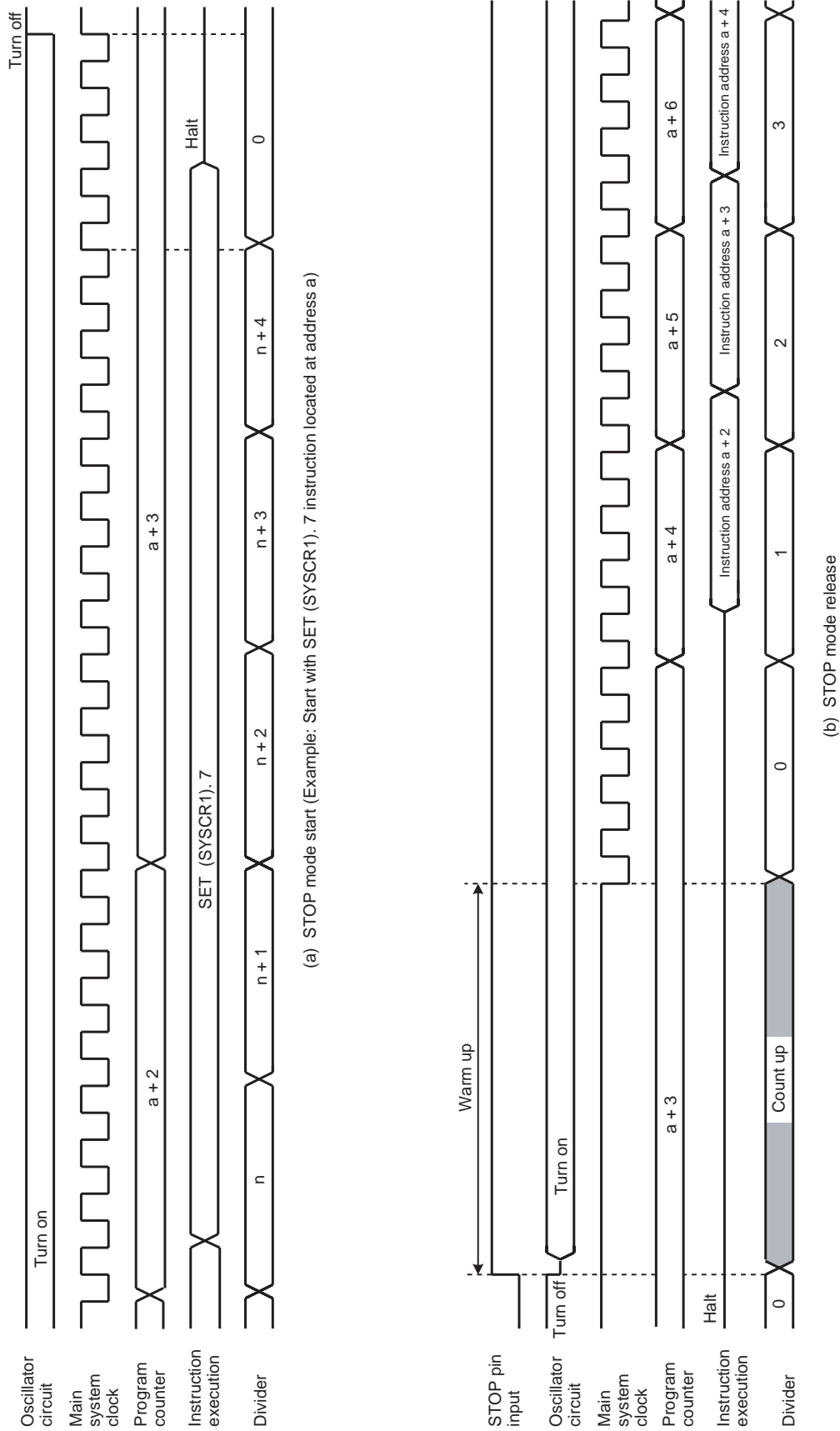


Figure 2-9 STOP Mode Start/Release

### 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.

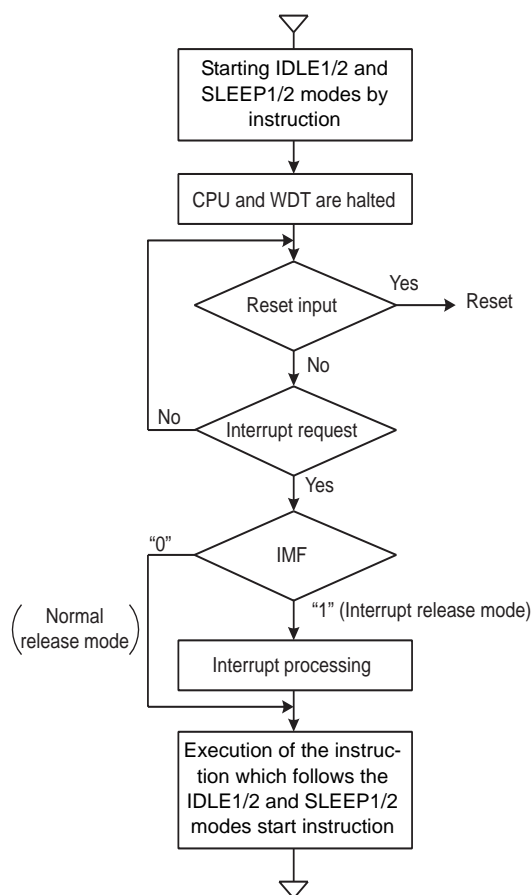


Figure 2-10 IDLE1/2 and SLEEP1/2 Modes

- 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 interrupt 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.

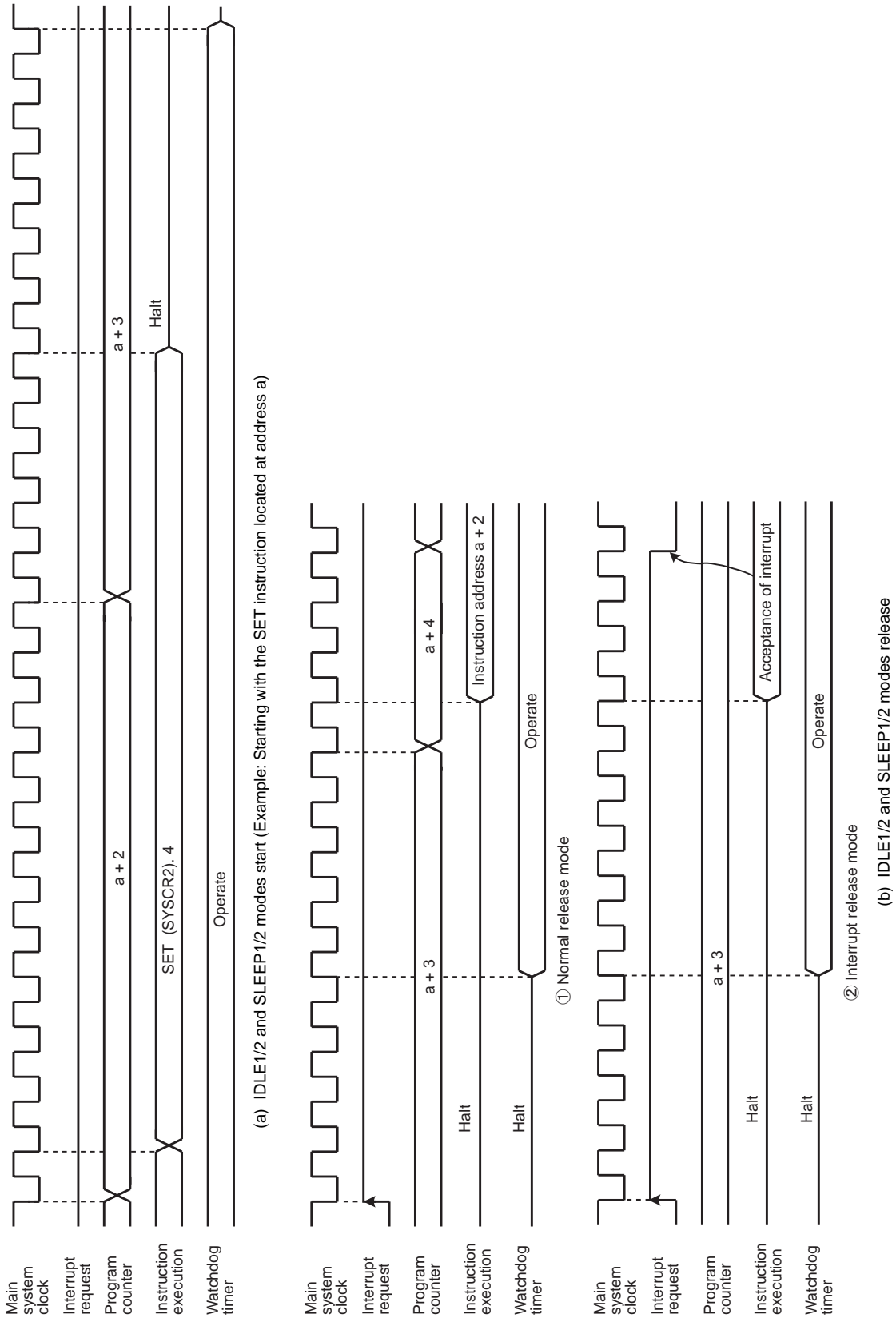


Figure 2-11 IDLE1/2 and SLEEP1/2 Modes Start/Release

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 (TBTCR). 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.

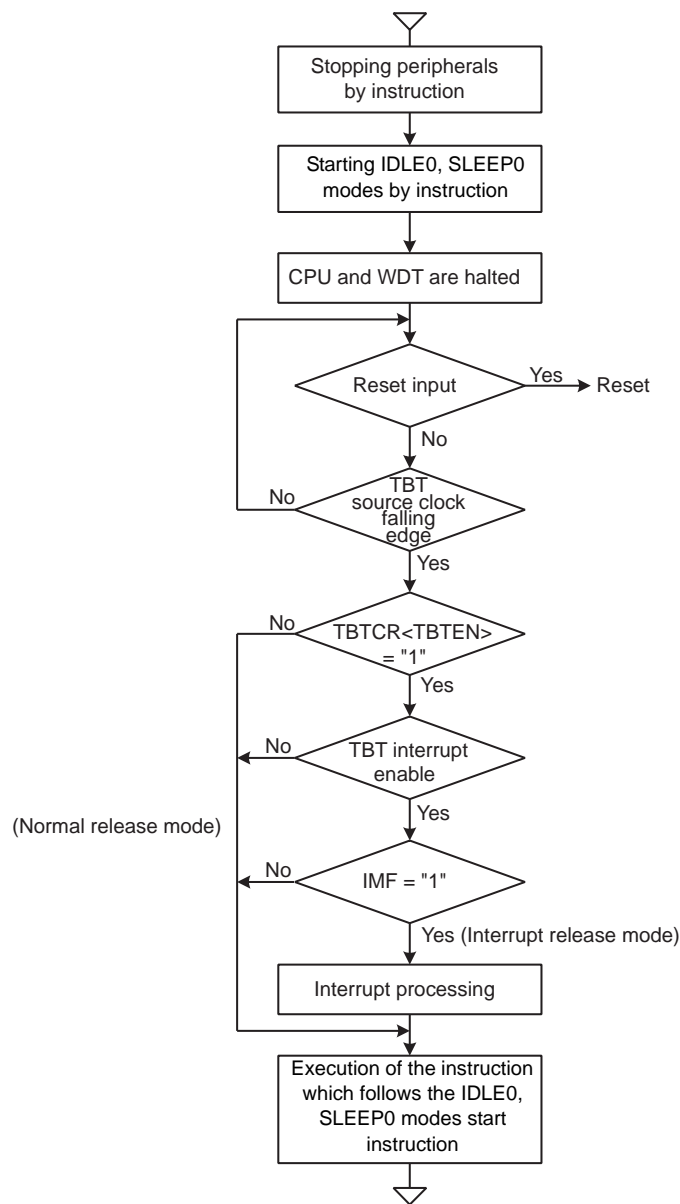


Figure 2-12 IDLE0 and SLEEP0 Modes

- Start the IDLE0 and SLEEP0 mode s
  - Stop (Disable) peripherals such as a timer counter.
  - To start IDLE0 and SLEEP0 mode s, set SYSCR2<TGHALT> to “1”.
- Release the IDLE0 and SLEEP0 mode s
  - IDLE0 and SLEEP0 mode s 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 mode s, the SYSCR2<TGHALT> is automatically cleared to “0” and the operation mode is returned to the mode preceding IDLE0 and SLEEP0 mode s.
  - 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 mode s 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 mode s start/release without reference to TBTCR<TBTEN> setting.

(1) Normal release mode (IMF · EF7 · TBTCR<TBTEN> = “0”)

IDLE0 and SLEEP0 mode s 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 mode s 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 · EF7 · TBTCR<TBTEN> = “1”)

IDLE0 and SLEEP0 mode s 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.

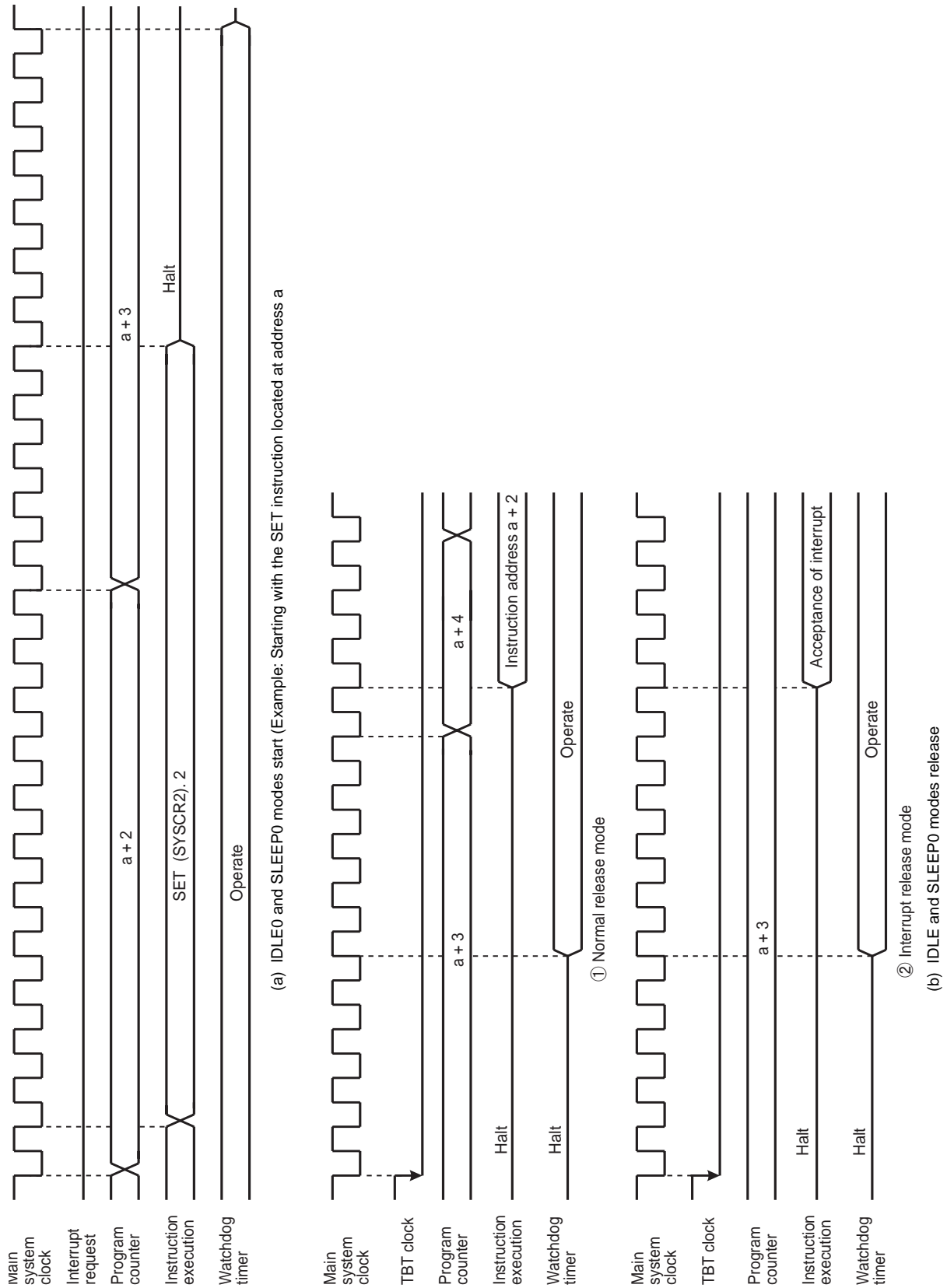


Figure 2-13 IDLE0 and SLEEP0 Modes Start/Release



## 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.

When the low-frequency clock oscillation is unstable, wait until oscillation stabilizes before performing the above operations. The timer/counter (TC4, TC3) can conveniently be used to confirm that low-frequency clock oscillation has stabilized.

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). 7       ; 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

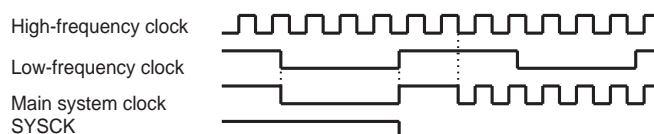
```

(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 SYSCR2<SYSCK> is cleared to 0, instructions are executed continuously by the low-frequency clock during synchronization period for high-frequency and low-frequency clocks.



Example :Switching from 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), 7      ; 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
  
```

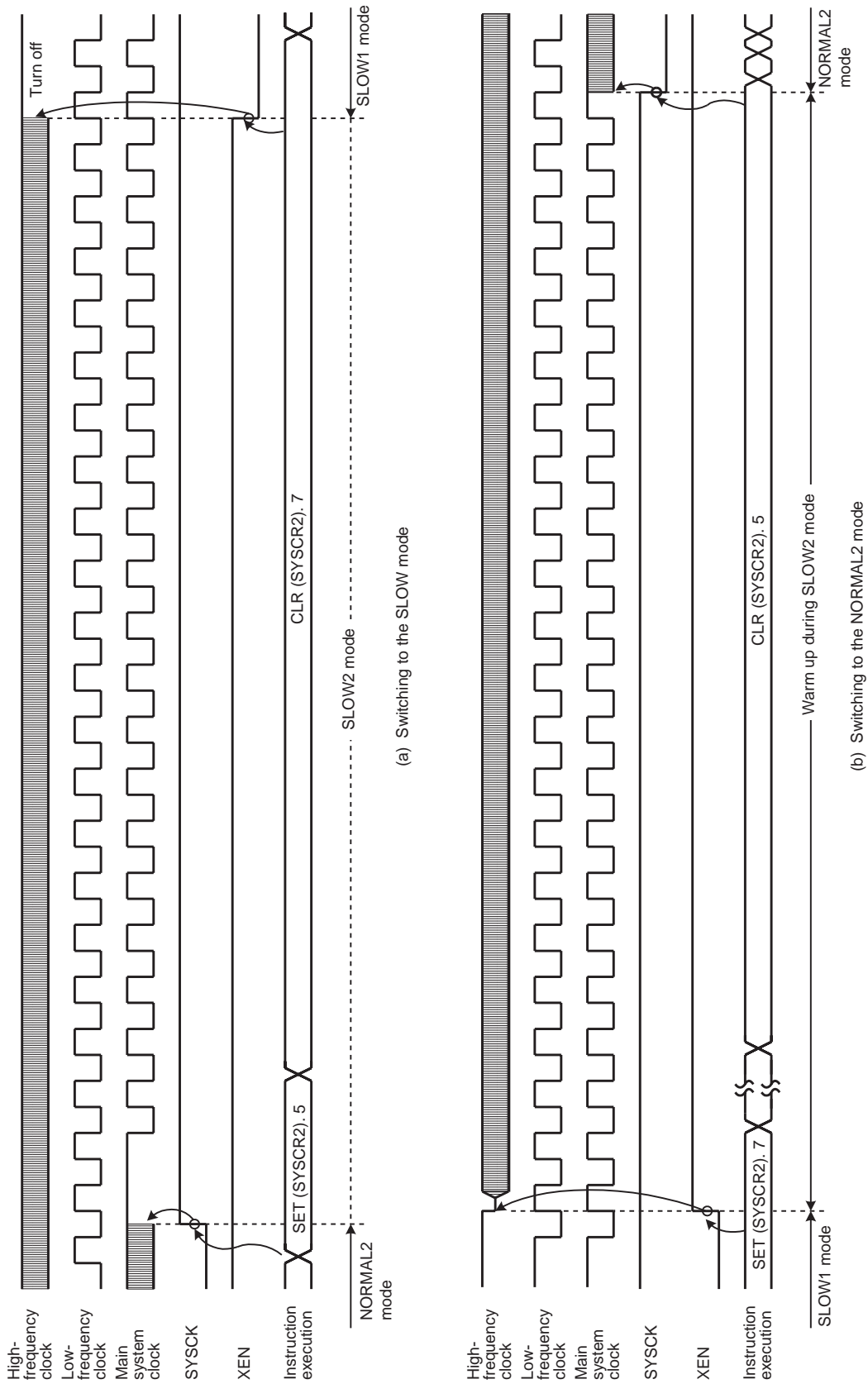


Figure 2-14 Switching between the NORMAL2 and SLOW Modes

## 2.3 Reset Circuit

The TMP86FH92DMG has types of reset generation procedures: An external reset input, an address trap reset, a watchdog timer reset and a system clock reset, voltage detect reset 1, voltage detection 2, power on reset, trimming data reset. Of these reset, the address trap reset, the watchdog timer and the system clock reset, voltage detect reset 1, voltage detection 2 are a malfunction reset. When the malfunction reset request is detected, reset occurs during the maximum  $24/f_c$  [s].

The power-on reset signal and trimming data reset signal are input to the power-on warming-up reset circuit, which causes the device to enter a reset state. After the power-on warming-up time (tPOWUP) has elapsed, the reset is released. For details, refer to the section on the power-on reset circuit.

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

Table 2-3 On-chip hardware Initialization 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	Watchdog timer	Enable
Jump status flag (JF)	Not initialized	voltage detection circuit	Disable
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	Control registers	Refer to each of control register
Interrupt individual enable flags (EF)	0		
Interrupt latches (IL)	0	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.

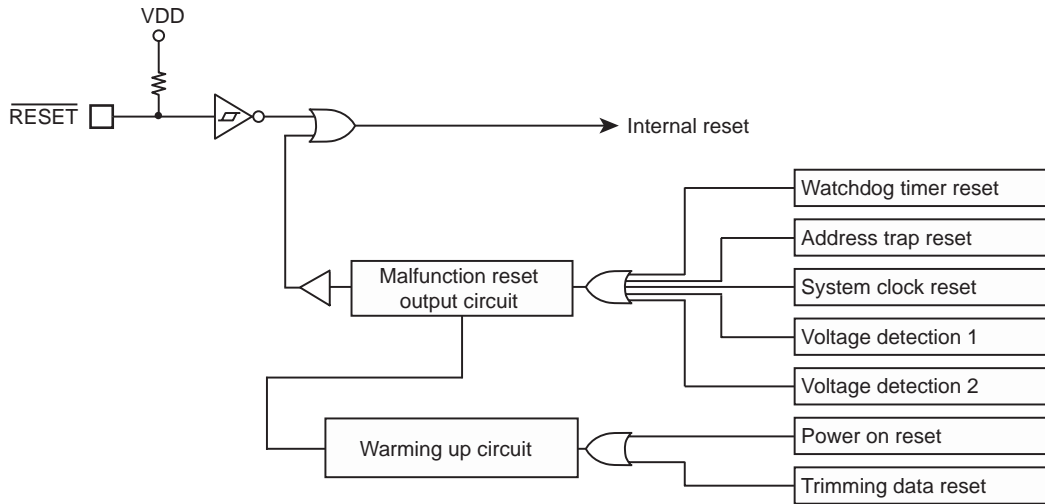
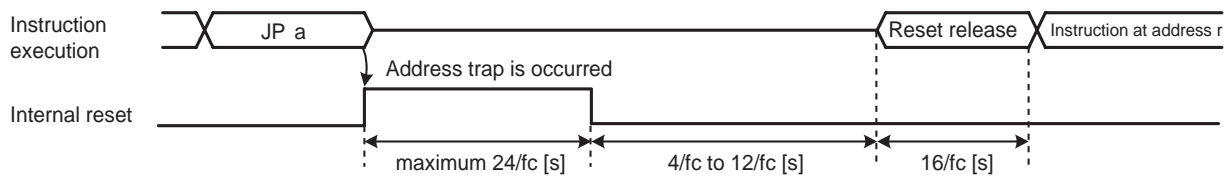


Figure 2-15 Reset Circuit

### 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 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 on-chip RAM (WDTCR1<ATAS> = “1”) space, DBR or SFR area.

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<XEN> to “1” when the SYSCR2<XTEN> is “0”.

The reset time is maximum  $24/f_c[s]$  (1.5  $\mu$ s at 16.0 MHz).

### 2.3.5 Power-on Reset

A power-on reset is generated internally when the supply voltage (VDD) is turned on.

Refer to Section “Power on reset”.

### 2.3.6 Voltage detection reset

A voltage detection reset is generated internally when the supply voltage (VDD) falls below the predefined threshold voltage.

Refer to Section “Voltage detection circuit”.

### 2.3.7 Trimming data reset

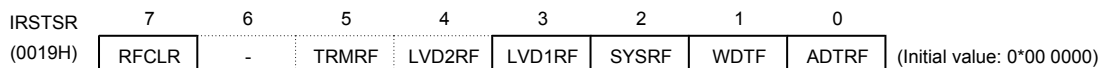
Trimming data bits are provided for adjusting the ladder resistor used to generate the reference voltages for the power-on reset signal and voltage detecting signal. These bits are read from the flash memory and latched internally during the power-on warming up period (tPOWUP). The trimming data reset is generated if the trimming data is corrupted due to noise or other causes. A supply voltage reset is generated internally when the supply voltage (VDD) falls below the predefined threshold voltage.

## 2.4 Internal Reset Detection Flags

After an internal reset is released, the cause of this internal reset can be identified by reading the internal reset detection flag register (IRSCR). IRSCR<SYSRSF> corresponds to system clock reset, IRSCR<ADTRF> to address trap reset, IRSCR<WDTF> to watchdog timer reset, and IRSCR<CLKSF> to clock stop detection reset. Each of these bits is set to “1” when the corresponding reset is generated.

To clear IRSCR<SYSRSF, ADTRF, WDTF, CLKSF> to “0”, write “1” in IRSCR<FCLR> or set the  $\overline{\text{RESET}}$  pin (external reset) to “L” level

### Internal Reset Detection Flag Register



Bit	Flag Name	Description	Initial State	State	Access
7	RFCLR	Reset flag of initialized	0: Initial state	1: Internal request reset flag to “0”	write only
5	TRMRF	Trimming data reset detection flag	0: Initial state	1: Trimming data reset detection flag	Read only
4	LVD2RF	Voltage detection2 reset flag	0: Initial state	1: Voltage detection2 reset detected	
3	LVD1RF	Voltage detection1 reset flag	0: Initial state	1: Voltage detection1 reset detected	
2	SYSRF	System clock reset detection flag	0: Initial state	1: System clock reset detected	
1	WDTF	Watchdog timer reset flag	0: Initial state	1: Watch dog reset detected	
0	ADTRF	Address latch reset detection flag	0: Initial state	1: Address trap reset detected	





### 3. Interrupt Control Circuit

The TMP86FH92DMG has a total of 22 interrupt sources excluding reset. Interrupts can be nested with priorities. Four of the internal interrupt sources are non-maskable while the rest are maskable.

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

Interrupt Factors		Enable Condition	Interrupt Latch	Vector Address	Priority
Internal/External	(Reset)	Non-maskable	-	FFFE	1
Internal	INTSWI (Software interrupt)	Non-maskable	-	FFFC	2
Internal	INTUNDEF (Executed the undefined instruction interrupt)	Non-maskable	-	FFFC	2
Internal	INTATRAP (Address trap interrupt)	Non-maskable	IL2	FFFA	3
Internal	INTWDT (Watchdog timer interrupt)	Non-maskable	IL3	FFF8	4
Internal	INTVLTD	IMF · EF4 = 1	IL4	FFF6	5
External	$\overline{\text{INT0}}$	IMF · EF5 = 1, INT0EN = 1	IL5	FFF4	6
External	INT1	IMF · EF6 = 1	IL6	FFF2	7
Internal	INTTBT	IMF · EF7 = 1	IL7	FFF0	8
Internal	INTSBI	IMF · EF8 = 1	IL8	FFEE	9
Internal	INTRXD1	IMF · EF9 = 1	IL9	FFEC	10
Internal	INTTXD1	IMF · EF10 = 1	IL10	FFEA	11
Internal	INTTC1	IMF · EF11 = 1	IL11	FFE8	12
Internal	INTRXD2	IMF · EF12 = 1	IL12	FFE6	13
Internal	INTTXD2	IMF · EF13 = 1	IL13	FFE4	14
Internal	INTTC3	IMF · EF14 = 1	IL14	FFE2	15
Internal	INTTC4	IMF · EF15 = 1	IL15	FFE0	16
External	INT3	IMF · EF16 = 1	IL16	FFBE	17
Internal	INTADC	IMF · EF17 = 1	IL17	FFBC	18
Internal	INTSEI0	IMF · EF18 = 1	IL18	FFBA	19
Internal	INTSEI1	IMF · EF19 = 1	IL19	FFB8	20
External	INT4	IMF · EF20 = 1	IL20	FFB6	21
External	$\overline{\text{INT5}}$	IMF · EF21 = 1	IL21	FFB4	22
-	Reserved	IMF · EF22 = 1	IL22	FFB2	23
-	Reserved	IMF · EF23 = 1	IL23	FFB0	24

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

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

## 3.1 Interrupt latches (IL21 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, 003DH, and 003EH 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, 003BH, and 0032H 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 (EF21 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 (EF21 to EF4) are initialized to "0" and all maskable interrupts are not accepted until they are set to "1".

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

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

Example 1 :Enables interrupts individually and sets IMF

```

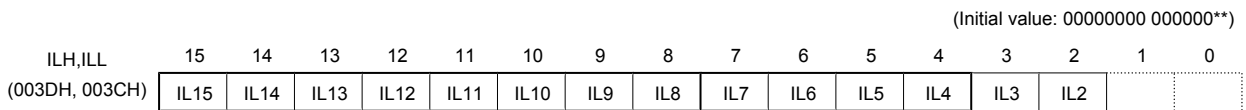
DI                                     ; IMF ← 0
LDW      (EIRL), 1110100010100000B    ; EF15 to EF13, EF11, EF7, EF5 ← 1
:                                       Note: IMF should not be set.
:
EI                                     ; IMF ← 1
    
```

Example 2 :C compiler description example

```

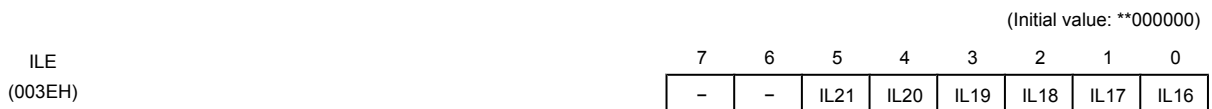
unsigned int _io (3AH) EIRL;           /* 3AH shows EIRL address */
_DI();
EIRL = 10100000B;
:
_EI();
    
```

### Interrupt Latches



ILH (003DH)

ILL (003CH)



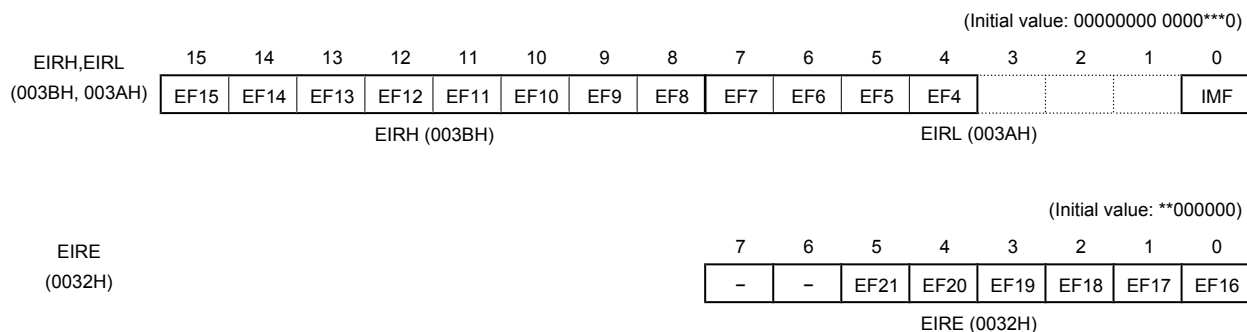
ILE (003EH)

IL21 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



EF21 to EF4	Individual-interrupt enable flag (Specified for each bit)	0: Disables the acceptance of each maskable interrupt. 1: Enables the acceptance of each maskable interrupt.	R/W
IMF	Interrupt master enable flag	0: Disables the acceptance of all maskable interrupts 1: Enables the acceptance of all maskable interrupts	

- Note 1: \*: Don't care
- Note 2: Do not set IMF and the interrupt enable flag (EF15 to EF4) to "1" at the same time.
- Note 3: In main program, before manipulating the interrupt enable flag (EF) or the interrupt latch (IL), be sure to clear IMF to "0" (Disable interrupt by DI instruction). Then set IMF newly again as required after operating on the EF or IL (Enable interrupt by EI instruction)  
 In interrupt service routine, because the IMF becomes "0" automatically, clearing IMF need not execute normally on interrupt service routine. However, if using multiple interrupt on interrupt service routine, manipulating EF or IL should be executed before setting IMF="1".

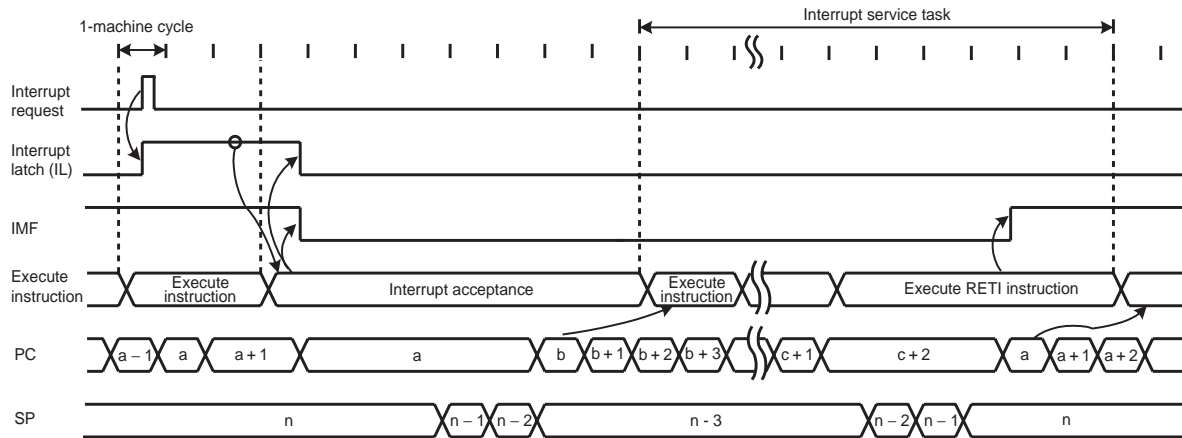
## 3.3 Interrupt Sequence

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

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

- 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, 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

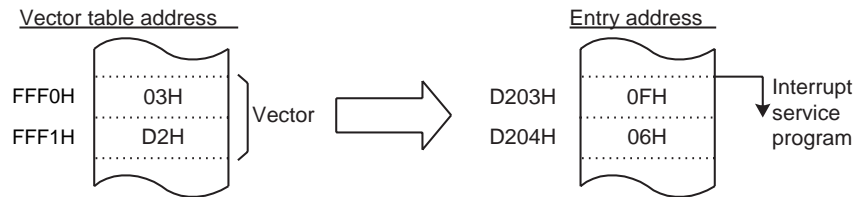


Figure 3-2 Vector Table Address and Entry Address

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

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

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

### 3.3.2 Saving/restoring general-purpose registers

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

### 3.3.2.1 Using PUSH and POP instructions

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

Example :Save/store register using PUSH and POP instructions

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

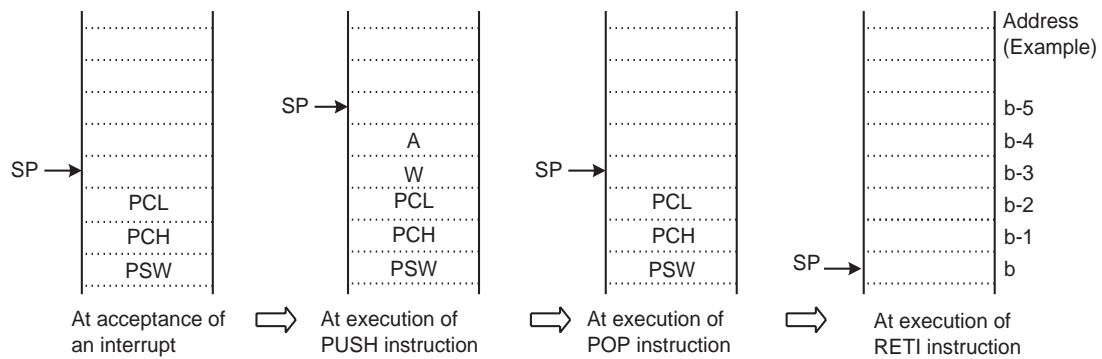


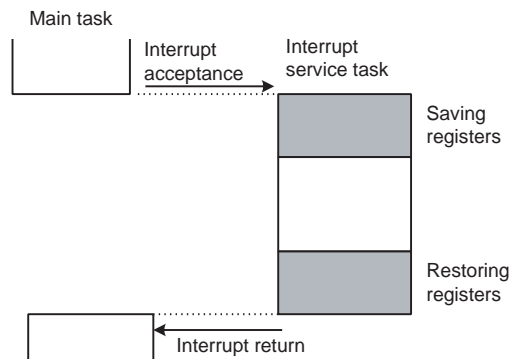
Figure 3-3 Saving/Restoring General-purpose Registers under PUSH and POP instructions

### 3.3.2.2 Using data transfer instructions

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

Example :Save/store register using data transfer instructions

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



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

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

### 3.3.3 Interrupt return

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

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

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

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

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

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

Example 2 :Restarting without returning interrupt

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

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

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

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

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

## 3.4 Software Interrupt (INTSW)

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

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

### 3.4.1 Address error detection

FFH is read if for some cause such as noise the CPU attempts to fetch an instruction from a non-existent memory address during single chip mode. Code FFH is the SWI instruction, so a software interrupt is generated



and an address error is detected. The address error detection range can be further expanded by writing FFH to unused areas of the program memory. Address trap reset is generated in case that an instruction is fetched from RAM, DBR or SFR areas.

### 3.4.2 Debugging

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

## 3.5 Undefined Instruction Interrupt (INTUNDEF)

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

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

## 3.6 Address Trap Interrupt (INTATRAP)

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

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

## 3.7 External Interrupts

The TMP86FH92DMG 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,INT3,INT4. The  $\overline{\text{INT0}}$ /P10 pin can be configured as either an external interrupt input pin or an input/output port, and is configured as an input port during reset.

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

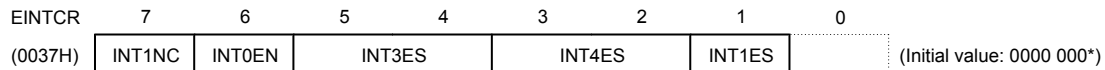
Source	Pin	Enable Conditions	Release Edge (level)	Digital Noise Reject
INT0	$\overline{\text{INT0}}$	$\text{IMF} \times \text{EF5} \times \text{INT0EN}=1$	Falling edge	Pulses of less than $2/f_c$ [s] are eliminated as noise. Pulses of $7/f_c$ [s] or more are considered to be signals. In the SLOW or the SLEEP mode, pulses of less than $1/f_s$ [s] are eliminated as noise. Pulses of $3.5/f_s$ [s] or more are considered to be signals.
INT1	INT1	$\text{IMF} \times \text{EF6} = 1$	Falling edge or Rising edge	Pulses of less than $15/f_c$ or $63/f_c$ [s] are eliminated as noise. Pulses of $49/f_c$ or $193/f_c$ [s] or more are considered to be signals. In the SLOW or the SLEEP mode, pulses of less than $1/f_s$ [s] are eliminated as noise. Pulses of $3.5/f_s$ [s] or more are considered to be signals.
INT3	INT3	$\text{IMF} \times \text{EF16} = 1$	Falling edge Rising edge Falling and Rising edge or H level	Pulses of less than $7/f_c$ [s] are eliminated as noise. Pulses of $25/f_c$ [s] or more are considered to be signals. In the SLOW or the SLEEP mode, pulses of less than $1/f_s$ [s] are eliminated as noise. Pulses of $3.5/f_s$ [s] or more are considered to be signals.
INT4	INT4	$\text{IMF} \times \text{EF20} = 1$	Falling edge Rising edge Falling and Rising edge or H level	Pulses of less than $7/f_c$ [s] are eliminated as noise. Pulses of $25/f_c$ [s] or more are considered to be signals. In the SLOW or the SLEEP mode, pulses of less than $1/f_s$ [s] are eliminated as noise. Pulses of $3.5/f_s$ [s] or more are considered to be signals.
INT5	$\overline{\text{INT5}}$	$\text{IMF} \times \text{EF21} = 1$	Falling edge	Pulses of less than $2/f_c$ [s] are eliminated as noise. Pulses of $7/f_c$ [s] or more are considered to be signals. In the SLOW or the SLEEP mode, pulses of less than $1/f_s$ [s] are eliminated as noise. Pulses of $3.5/f_s$ [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 +  $6/f_s$ [s]" from the input signal's edge to set the interrupt latch.

Note 2: When INT0EN = "0", IL5 is not set even if a falling edge is detected on the  $\overline{\text{INT0}}$  pin input.

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

External Interrupt Control Register



INT1NC	Noise reject time select	0: Pulses of less than $63/f_c$ [s] are eliminated as noise 1: Pulses of less than $15/f_c$ [s] are eliminated as noise	R/W
INT0EN	P10/ $\overline{\text{INT0}}$ pin configuration	0: P10 input/output port 1: $\overline{\text{INT0}}$ pin (Port P10 should be set to an input mode)	R/W
INT4 ES	INT4 edge select	00: Rising edge 01: Falling edge 10: Rising edge and Falling edge 11: "H" level	R/W
INT3 ES	INT3 edge select	00: Rising edge 01: Falling edge 10: Rising edge and Falling edge 11: "H" level	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  $2^9/f_c$ .

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



## 4. Special Function Register (SFR)

The TMP86FH92DMG adopts the memory mapped I/O system, and all peripheral control and data transfers are performed through the special function register (SFR) or the data buffer register (DBR). The SFR is mapped on address 0000H to 003FH, DBR is mapped on address 0F80H to 0FFFH.

This chapter shows the arrangement of the special function register (SFR) and data buffer register (DBR) for TMP86FH92DMG.

### 4.1 SFR

Address	Read	Write
0000H		P0DR
0001H		P1DR
0002H		P2DR
0003H		P3DR
0004H		P0PUCR
0005H		P1PUCR
0006H		P2PUCR
0007H		Reserved
0008H		P3CR1
0009H		P1OUTCR
000AH		P3CR2
000BH		P0OUTCR
000CH	P0PRD	-
000DH	P2PRD	-
000EH		ADCCR1
000FH		ADCCR2
0010H		TC1DRAL
0011H		TC1DRAH
0012H		TC1DRBL
0013H		TC1DRBH
0014H		TC1CR
0015H	SBISRA	SBICRA
0016H		SBIDBR
0017H	-	I2CAR
0018H	SBISRB	SBICRB
0019H		IRSTSR
001AH		TC3CR
001BH		TC4CR
001CH		TTREG3
001DH		TTREG4
001EH		PWREG3
001FH		PWREG4
0020H	ADCDR2	-
0021H	ADCDR1	-
0022H	UART2SR	UART2CR1
0023H	-	UART2CR2
0024H	RD2BUF	TD2BUF
0025H	UART1SR	UART1CR1

Address	Read	Write
0026H	-	UART1CR2
0027H	RD1BUF	TD1BUF
0028H	SESR	-
0029H	SEDR	
002AH	SECR	
002BH	VDCR1	
002CH	VDCR2	
002DH	P1PRD	-
002EH	Reserved	
002FH	Reserved	
0030H	Reserved	
0031H	-	STOPCR
0032H	EIRE	
0033H	Reserved	
0034H	-	WDTCR1
0035H	-	WDTCR2
0036H	TBTCR	
0037H	EINTCR	
0038H	SYSCR1	
0039H	SYSCR2	
003AH	EIRL	
003BH	EIRH	
003CH	ILL	
003DH	ILH	
003EH	ILE	
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	
::	::	
0F9FH	Reserved	

Address	Read	Write
0FA0H	Reserved	
::	::	
0FBFH	Reserved	

Address	Read	Write
0FC0H	Reserved	
::	::	
0FDH	Reserved	

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

Address	Read	Write
0FFFH	FLSCR	

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

Note 2: - ; Cannot be accessed.

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



## 5. I/O Ports

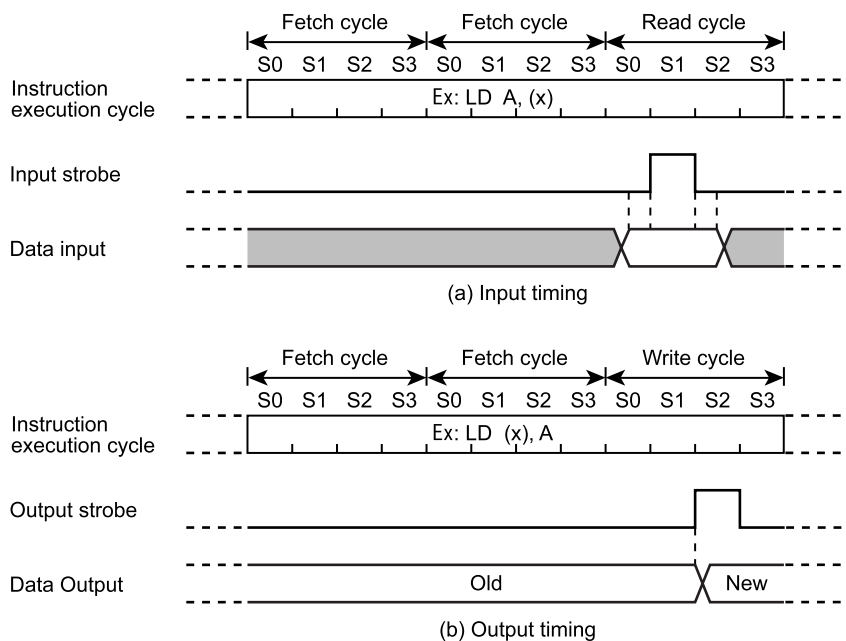
The TMP86FH92DMG have 4 parallel input/output ports as follows.

	Primary Function	Secondary Functions
Port P0	8-bit I/O port	External interrupt input, Timer/Counter input/output, UART input/output, Serial expansion interface input/output and serial PROM mode control input.
Port P1	5-bit I/O port	External interrupt input, divider output, UART input/output and serial bus interface input/output
Port P2	3-bit I/O port	External interrupt input, STOP mode release signal input and low frequency resonator connection
Port P3	8-bit I/O port	Analog input, STOP mode release signal input and Timer/Counter input/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 P0 (P07 to P00) Port (High Current)

The P0 port is an 8-bit input/output port shared with external interrupt input, serial expansion interface input/output, UART1 input/output, timer counter input/output and serial PROM mode control input. When using this port as serial expansion interface output or UART1 output, set the output latch to 1. When using this port as a port output, the output latch data (P0DR) is output to the P0 port.

When reset, the output latch (P0DR) and the push-pull control register (P0OUTCR) are initialized to 1 and 0, respectively.

The P0 port allows its output circuit to be selected between N-channel open-drain output or push-pull output by the P0OUTCR register.

The P0 port has programmable internal Pull-up resistance to be controlled by P0PUCR registers.

When using this port as a port input, external interrupt input, serial expansion interface input, UART1 input and timer counter input, set the P0OUTCR register's corresponding bit to 0 after setting the P0DR to 1. The P0 port has independent data input registers. To inspect the output latch status, read the P0DR register. To inspect the pin status, read the P0PRD register.

In the serial PROM mode, P01 pin used as a BOOT/RXD1 pin, P00 pin used as a TXD1 pin. For details, see "Serial PROM Mode Setting".

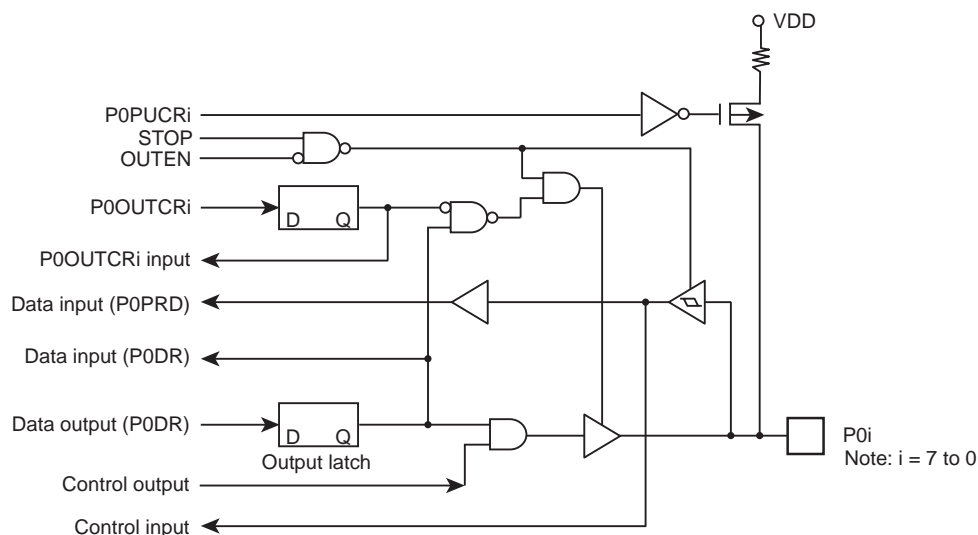


Figure 5-2 P0 Port

	7	6	5	4	3	2	1	0	
P0DR (0000H) R/W	P07 TC1 INT4	P06 INT3 PPG	P05 $\overline{SS}$	P04 MISO	P03 MOSI	P02 SCLK	P01 RxD1 BOOT	P00 TxD1	(Initial value: 1111 1111)

	7	6	5	4	3	2	1	0	
P0OUTCR (000BH) R/W									(Initial value: 0000 0000)

P0OUTCR	P0 port input/output control (specified bitwise)	0: Nch open-drain output 1: Push-pull output	R/W
---------	--	---	-----

	7	6	5	4	3	2	1	0	
P0PUCR (0004H) R/W									(Initial value: 0000 0000)

P0PUCR	P0 port pull-up resistance control (specified bitwise)	0: No Pull-up resistance 1: Pull-up resistance	R/W
--------	--	---	-----

	7	6	5	4	3	2	1	0	
P0PRD (000CH) Read only	P07	P06	P05	P04	P03	P02	P01	P00	

## 5.2 P1 (P14 to P10) Port

The P1 port is a 5-bit input/output port shared with external interrupt input, divider output, UART2 input/output and serial bus interface input/output. When using this port as divider output, UART2 output and serial bus interface output, set the output latch to 1. When using this port as a port output, the output latch data(P1DR) is output to the P1 port.

When reset, the output latch (P1DR) and the push-pull control register(P1OUTCR) are initialized to 1 and 0, respectively.

The P1 allows its output circuit to be selected between N-channel open-drain or push-pull output by the P1OUTCR register.

The P1 port has programmable internal pull-up resistance to be controlled by P1PUCR.

When using this port as a port input, external interrupt input, UART2 input and serial bus interface input, set the P1OUTCR register's corresponding bit to 0 after setting the P1DR to 1.

The P1 port has independent data input registers. To inspect the output latch status, read the P1DR register. To inspect the pin status, read the P1PRD register.

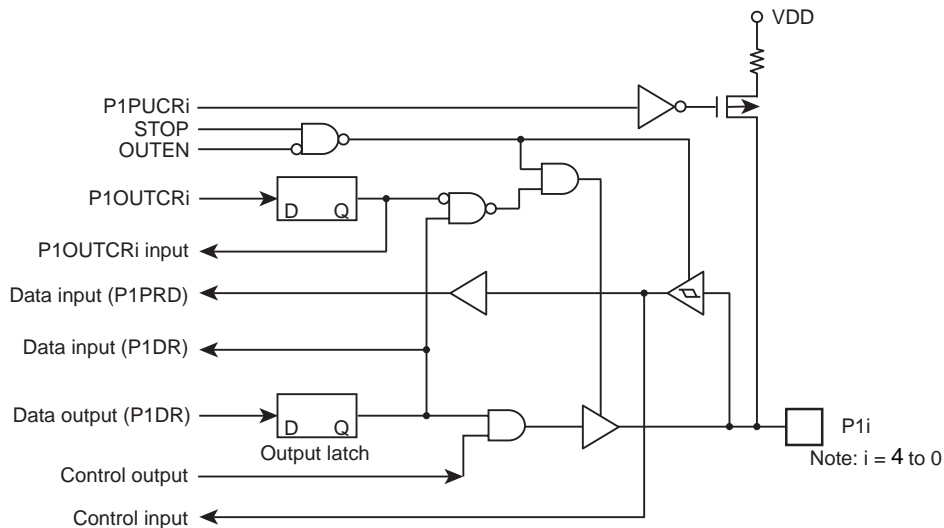


Figure 5-3 P1 Port

	7	6	5	4	3	2	1	0	
P1DR (0001H) R/W				P14 TXD2 SCL	P13 RXD2 SDA	P12 DVO	P11 INT1	P10 INT0	(Initial value: ***1 1111)

	7	6	5	4	3	2	1	0	
P1OUTCR (0009H) R/W									(Initial value: ***0 0000)

P1OUTCR	P1port input/output control (specified bitwise)	0: Nch open-drain output 1: Push-pull output	R/W
---------	---	---	-----

	7	6	5	4	3	2	1	0	
P1PUCR (0005H) R/W									(Initial value: ***0 0000)

P1PUCR	P1 port pull-up resistance control (specified bitwise)	0: No Pull-up resistance 1: Pull-up resistance	R/W
--------	--	---	-----

	7	6	5	4	3	2	1	0	
P1PRD (002DH) Read only				P14	P13	P12	P11	P10	

Note: P13 and P14 can be used as the input/output for either UART2 or I<sup>2</sup>C bus control signals. Therefore, UART2 and serial bus interface cannot be used at the same time. These functions can be enabled and disabled in their respective function registers. UART2 and I<sup>2</sup>C bus cannot be enabled at the same time.

### 5.3 P2 (P22 to P20) Port

The P2 port is a 3-bit input/output port shared with external interrupt input, STOP mode release signal input, and low-frequency resonator connecting pin. When using this port as a port input or function pin, set the output latch to 1. The output latch is initialized to 1 when reset. When operating in dual-clock mode, connect a low-frequency resonator (32.768 kHz) to the P21 (XTIN) and P22 (XTOUT) pins. When operating in single-clock mode, the P21 and P22 pins can be used as ordinary input/output ports. We recommend using the P20 pin for external interrupt input or STOP mode release signal input or as a port input. (When used as a port output, the interrupt latch is set by a falling edge.)

The P2 port has independent data input registers. To inspect the output latch status, read the P2DR register. To inspect the pin status, read the P2PRD register. When the P2DR or P2PRD read instruction is executed for the P2 port, the values read from bits 7 to 3 are indeterminate.

The P20 port has programmable internal Pull-up resistance to be controlled by P2PUCR.

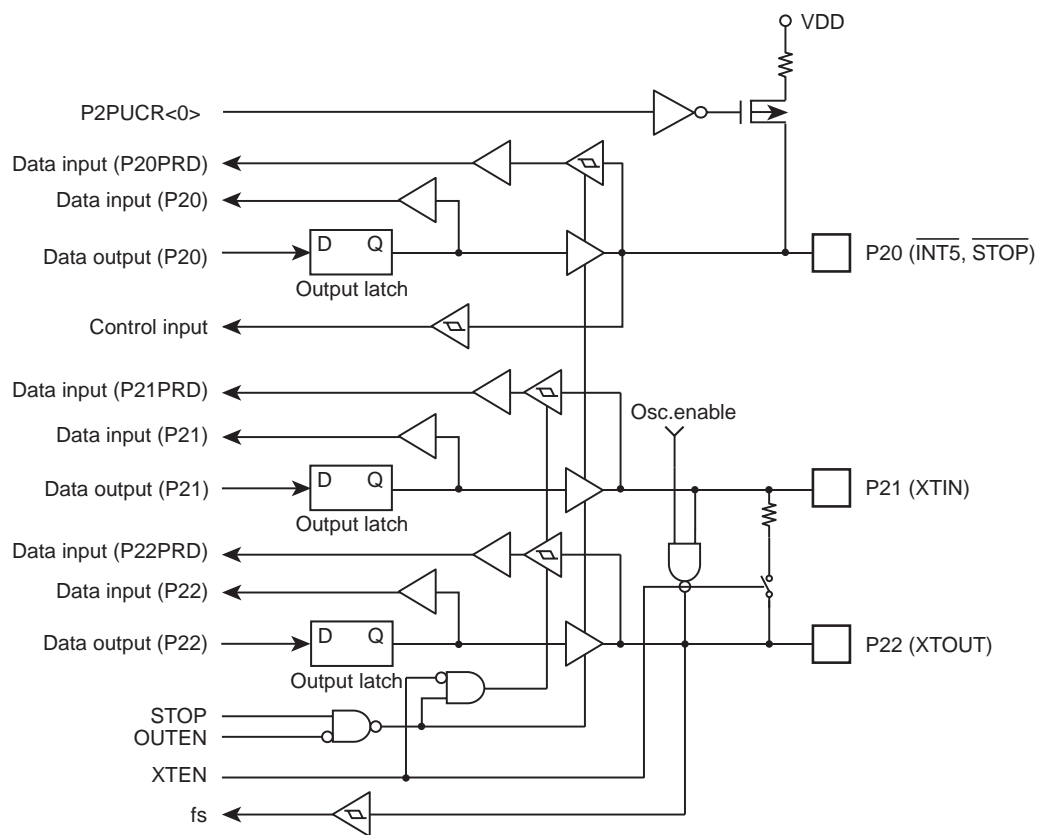
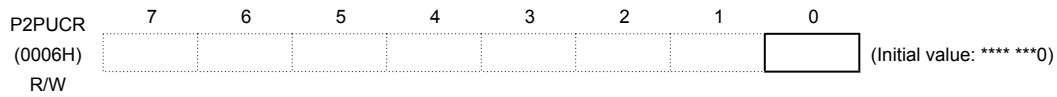
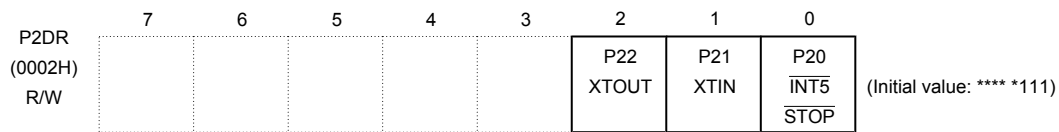


Figure 5-4 P2 Port



P2PUCR	P2 port pull-up resistance control (specified bitwise)	0: No Pull-up resistance 1: Pull-up resistance	R/W
--------	--	---	-----



Note: The P20 pin is shared with the  $\overline{\text{STOP}}$  pin, so that when in STOP mode, its output goes to a High-Z state regardless of the OUTEN status.

## 5.4 P3 (P37 to P30) Port

The P3 port is an 8-bit input/output port that can be specified for input or output bitwise, and is shared with analog input and key-on wakeup input (KWI).

The P3 port input/output control registers P3CR1 and P3CR2 are used to specify the function of each pin. After reset, the P3CR1 and P3CR2 are initialized to 0 and 1, respectively, so the P3 port is configured for input mode. The P3 port output latches are initialized to 0.

To use each pin as a port output, set the corresponding bit in the P3CR1 to 1.

To use each pin as a port input, set the corresponding bit in the P3CR1 to 0 and then set the corresponding bit in the P3CR2 to 1.

To use each pin as a key-on wakeup input, set the corresponding bit in the P3CR1 to 0 and then set the corresponding bit in the STOPCR to 1.

To use each pin as an analog input, set the corresponding bit in the P3CR1 to 0 and then set the corresponding bit in the P3CR2 to 0.

When P3CR1=1, reading the P3DR returns the values of the respective output latches.

Any pins of the P3 port which are not used for analog input can be used as input/output ports. During AD conversion, however, avoid executing output instructions on these pins to ensure the accuracy of conversion. Also note that, during AD conversion, rapidly changing signals should not be input on any pins near analog input pins.

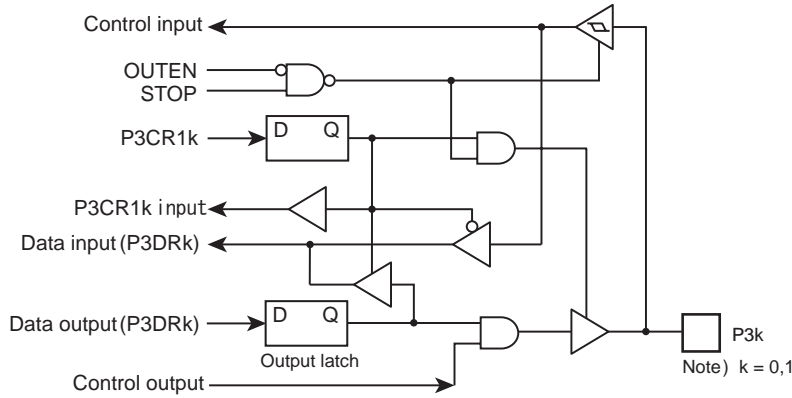
Table 5-1 Setting of register according to each function value

Function	Set value			
	P3DR	P3CR1	P3CR2	STOPCR
Port input	-	0	1	-
Key-on wakeup input	-	0	-	1
Analog input	-	0	0	-
Port 0 output	0	1	-	-
Port 1 output	1	1	-	-

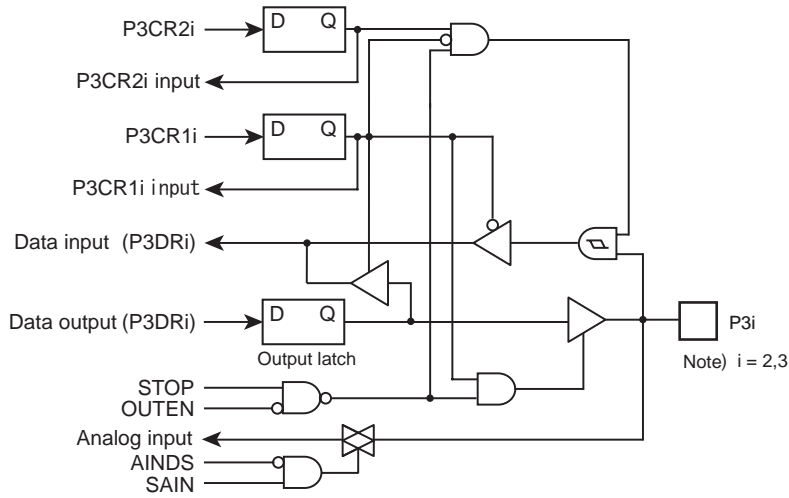
Table 5-2 Setting of register according to each function value

Condition		Reading value of P3DR
P3CR1	P3CR2	
0	0	0
0	1	State of terminal
1	0	Contents of output latch
	1	

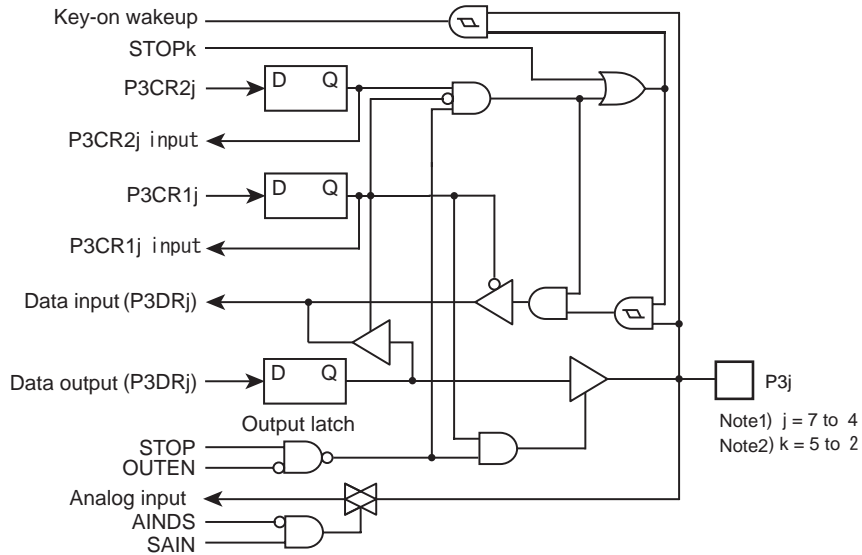




a) Equivalent circuit of P30, P31



b) Equivalent circuit of P32, P33



c) Equivalent circuit of P34 to P37

- Note 1) STOP = bit 7 of SYSCR1
- Note 2) SAIN = AD input select signal
- Note 3) STOPk = bit 7 to bit 4 of STOPCR

Figure 5-5 P3 Port

	7	6	5	4	3	2	1	0	
P3DR (0003H) R/W	P37 AIN5 STOP5	P36 AIN4 STOP4	P35 AIN3 STOP3	P34 AIN2 STOP2	P33 AIN1	P32 AIN0	P31 TC4 PDO4 PWM4 PPG4	P30 TC3 PDO3 PWM3	(Initial value: 0000 0000)

P3CR1 (0008H)	7	6	5	4	3	2	1	0	(Initial value: 0000 0000)

P3CR1	Controls P3 port input/output (specified bitwise)	0: Input mode (port input, or analog input or key on wake up input) 1: Output mode	R/W
-------	---	---	-----

P3CR2 (000AH)	7	6	5	4	3	2	1	0	(Initial value: 1111 11**)

P3CR2	Controls P3 port input (specified bitwise)	0: Analog input 1: Port Input	R/W
-------	--	----------------------------------	-----

- Note 1: The port placed in input mode reads the pin input state. Therefore, when the input and output modes are used together, the output latch contents for the port in input mode might be changed by executing a bit manipulation instruction.
- Note 2: As for the analog input pin because of penetration electric current measure, please be sure to clear the bit to which P3CR2 corresponds in "0".
- Note 3: Do not set the output mode (P3CR1="1") for the pin used as an analog input, due to avoid external short-circuits.
- Note 4: Pins not used for analog input can be used as I/O ports. During AD conversion, output instructions should not be executed to keep a precision. In addition, a variable signal should not be input to a port adjacent to the analog input during AD conversion.

## 6. Power-on reset circuit

### 6.1 Power-on reset circuit

The power-on reset circuit generates a reset when the TMP86FH92DMG is powered on. It also generates a power-on reset signal if the supply voltage drops below the threshold voltage of the power-on reset circuit.

Note: The power-on reset circuit cannot be emulated with the TMP86C993XB (Emulation Chip). Therefore, when using the development tool for debugging, ensure that operation is performed within the operating voltage range of the TMP86FH92DMG. For the operating voltage range, refer to the chapter on electrical characteristics.

#### 6.1.1 Configuration

The power-on reset circuit is comprised of a reference voltage generator and a comparator.

The comparator compares the supply voltage divided by a resistor ladder with the reference voltage generated by the reference voltage generator.

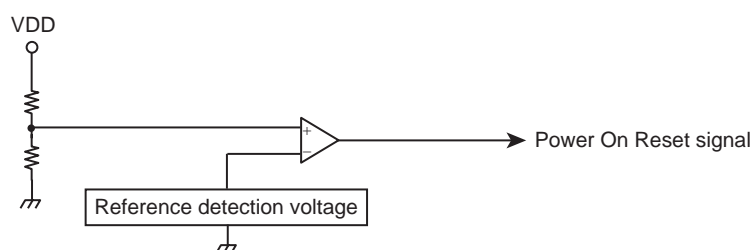


Figure 6-1 Power-on reset circuit

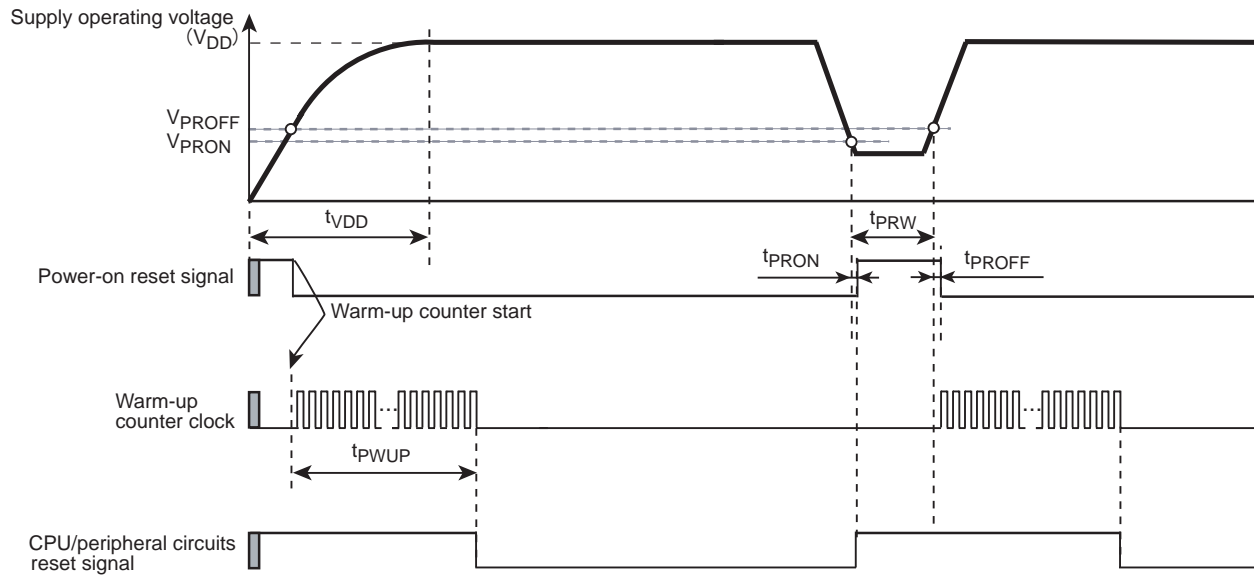
#### 6.1.2 function

When the TMP86FH92DMG is powered on, the power-on reset circuit generates a power-on reset signal while the supply voltage is below the power-on reset release voltage. The power-on reset signal is released when the supply voltage rises above the power-on reset release voltage.

When the TMP86FH92DMG is shut off, the power-on reset circuit generates a power-on reset signal when the supply voltage drops below the power-on reset threshold voltage. While the power-on reset signal is generated, the warm-up counter circuit, CPU and peripheral circuits are reset.

Upon release of the power-on reset signal, the warm-up counter circuit starts operating. After the warm-up time has elapsed, the CPU and peripheral circuits are released from the reset state.

After the supply voltage reaches the power-on reset release voltage level, it must be raised to the operating range before the power-on warm-up time expires. If the supply voltage is not in the operating range at the completion of the power-on warm-up time, the TMP86FH92DMG cannot operate properly.



Note 1: The power-on reset circuit may not operate properly depending on transitions in supply voltage ( $V_{DD}$ ). When designing your application system, careful consideration must be given to ensure proper operation of the power-on reset circuit by referring to the device's electrical characteristics.

Note 2: The input clock to the warm-up counter is derived from the oscillation circuit. Because the oscillation frequency is unstable until the oscillation circuit stabilizes, the warm-up time includes error.

Note 3: The supply voltage must be raised to satisfy the condition  $t_{VDD} < t_{PWUP}$ .

Figure 6-2 Operation of the power on reset circuit

## 7. Voltage detection circuit (VLTD)

The voltage detecting circuit monitors the supply voltage level and generates an interrupt or reset upon detection of a low-voltage condition.

Note: The voltage detecting circuit may not operate properly depending on transitions in supply voltage (VDD). When designing your application system, careful consideration must be given to ensure proper operation of the voltage detecting circuit by referring to the device's electrical characteristics.

### 7.1 Configuration

The voltage detecting circuit is comprised of a reference voltage generator, two threshold voltage select circuits, two comparators and control registers.

The supply voltage (VDD) is divided by a resistor ladder and is input into each threshold voltage select circuit. The threshold voltage (VDxLVL) (x = 1, 2) is selected through the VDxLVL bits of the Voltage Detect Control Register 1 (VDCR1). The selected threshold voltage is used to generate the input voltage into the comparator, which is then compared with the reference voltage. When the supply voltage (VDD) falls below the threshold voltage (VDxLVL), the voltage detecting circuit generates an INTVLTD interrupt or a voltage detect x reset signal.

Whether to generate an INTVLTD interrupt or a voltage detect x reset signal can be programmed by software.

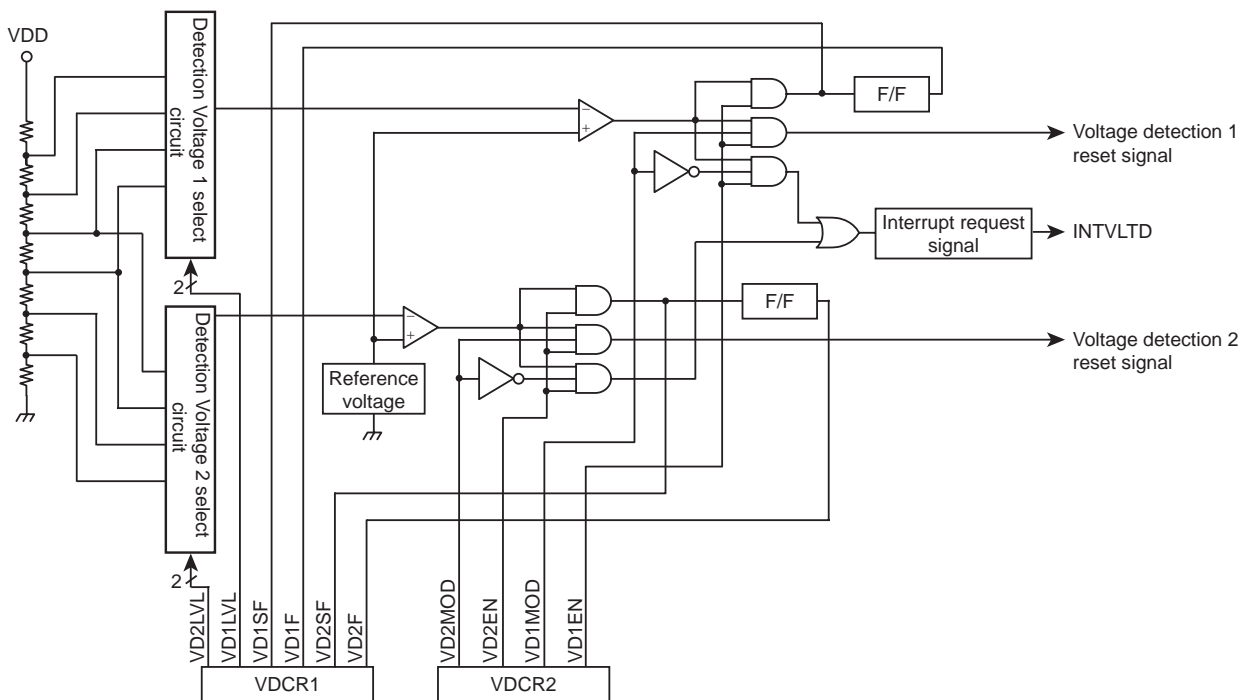


Figure 7-1 Voltage detection circuit diagram

## 7.2 Control

The voltage detection circuit is controlled by the Voltage Detection Control Register 1 (VDCR1) and Voltage Detection Control Register 2 (VDCR2).

The functions of the VDCR1 and VDCR2 vary between the TMP86FH92DMG and the emulation chip TMP86C993XB. For details, refer to the register descriptions below.

The TMP86C993XB does not allow an interrupt or a reset to be generated by voltage detection. Instead, the VD1S and VD2S bits are provided in the VDCR2 to support the emulation of voltage detection operation. Setting VDCR2<VDxS> to 0 while VDCR2<VDxEN> is set to 1 generates an interrupt or a reset depending on the VDCR2<VDxMOD> setting (x = 1, 2). In debugging the voltage detection circuit with the TMP86C993XB (development tool), this function can be used to emulate interrupt/reset generation by inserting a write instruction to the VDCR2 in a software program.

Please ensure that the final verification of software operation is conducted with the TMP86FH92DMG.

### Voltage detection control register1

VDCR1	7	6	5	4	3	2	1	0	
(002BH)	VD2F	VD2SF	VD2LVL	VD1F	VD1SF	VD1LVL	(Initial value: 0010 0000)		

		TMP86FH92DMG		TMP86C993XB		
VD2F	Voltage detection 2 flag (Latches the state when VDD < VD2LVL) (Note 2)	Read		Read		R/W
		0:	VDD ≥ VD2LVL	0:	-	
		1:	VDD < VD2LVL	1:	Writing a 0 to VDCR2 <VD2S> sets this bit to 1.	
		Write		Write		
		0:	Clearing the flag	0:	Clear the flag	
		1:	- (Note 4)	1:	- (Note 4)	
VD2SF	Voltage detection 2 status flag (Indicates the relation between VDD and VD2LVL when read.)	0:	VDD ≥ VD2LVL	This bit is only read as 1 immediately (for 3 machine cycles) after a 0 is written to VDCR2<VD2S>. At other times, it is always read as 0.		Read Only
		1:	VDD < VD2LVL			
VD2LVL	Voltage detection 2 level select (Note 3)	00:	Reserved	This bit has no meaning as voltage detection is performed based on the VDCR2<VD2S> setting.		R/W
		01:	2.9 to 3.3V			
		10:	Reserved			
		11:	Reserved			
VD1F	Voltage detection 1 flag (Latches the state when VDD < VD1LVL.) (Note 2)	Read		Read		R/W
		0:	VDD ≥ VD1LVL	0:	-	
		1:	VDD < VD1LVL	1:	Writing a 0 to VDCR2 <VD1S> sets this bit to 1.	
		Write		Write		
		0:	Clearing the flag	0:	Clear the flag	
		1:	- (Note 4)	1:	- (Note 4)	
VD1SF	Voltage detection 1 status flag (Indicates the relation between VDD and VD1LVL when read.)	0:	VDD ≥ VD1LVL	This bit is only read as 1 immediately (for 3 machine cycles) after a 0 is written to VDCR2<VD1S>. At other times, it is always read as 0.		Read only
		1:	VDD < VD1LVL			
VD1LVL	Voltage detection 1 level select	00:	4.0 to 4.7V	This bit has no meaning as voltage detection is performed based on the VDCR2<VD1S> setting.		R/W
		01:	Reserved			
		10:	Reserved			
		11:	2.9 to 3.3V			

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

Note 2: If VDCR1<VD2F> or VDCR1<VD1F> is cleared by software simultaneously as it is set by detection of a low-voltage condition, the setting operation overrides the clearing operation so that the bit is set to 1.



## 7.3 Function

The voltage detecting circuit allows two threshold voltage levels (VDxLVL, x = 1, 2) to be specified. For each threshold voltage, whether to enable or disable voltage detect operation and the action to be taken when the supply voltage (VDD) falls below the threshold voltage (VDxLVL) can be programmed by software.

### 7.3.1 Enabling/Disabling Voltage Detection Operation

Setting the VDCR2<VDxEN> bit to 1 enables voltage detect operation and clearing this bit to 0 disables it. Immediately after release of a power-on reset, VDCR2<VDxEN>, is cleared to 0.

Note: Setting VDCR2<VDxEN> to 1 while the supply voltage is below the threshold voltage (VDD<VDxLVL) immediately causes an INTVLTD interrupt or a voltage detect x reset signal to be generated.

### 7.3.2 Selecting the Voltage Detect Operating Mode

When the voltage detect operation mode is set to generating an interrupt request (VDCR2<VDxMOD>=0) and voltage detect operation is enabled (VDCR2<VDxEN>=1), an INTVLTD interrupt is generated if the supply voltage (VDD) falls below the threshold voltage (VDxLVL).

When the voltage detect operation mode is set to generating a reset signal (VDCR2<VDxMOD>=1) and voltage detect operation is enabled (VDCR2<VDxEN>=1), a voltage detect x reset signal is generated if the supply voltage (VDD) falls below the threshold voltage (VDxLVL).

The VDCR1 and VDCR2 are only initialized by a power-on reset or an external reset input. Therefore, the reset input signal remains asserted while the supply voltage (VDD) is below the threshold voltage (VDxLVL).

Note: The "reset input signal" mentioned above is the reset input shown in "Reset Circuit" section.

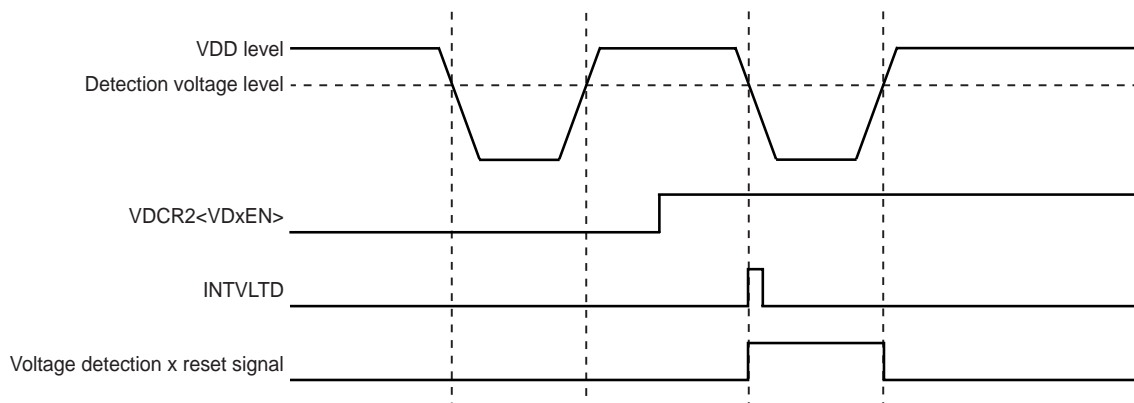


Figure 7-2 interrupt request signal and reset signal

### 7.3.3 Detection voltage level section

The threshold voltage level is selected by programming the VDCR1<VDxLVL> bits.

### 7.3.4 Voltage detection flag and voltage detection status flag

The voltage detect flag (VDCR1<VDxF>) and voltage detect status flag (VDCR1<VDxSF>) indicate the relation between the supply voltage (VDD) and the threshold voltage (VDxLVL), i.e, whether VDD is above or equal to VDxLVL or VDD is below VDxLVL.



When  $VDCR2\langle VD_xEN \rangle = 1$ , a drop of the supply voltage (VDD) below the threshold voltage ( $VD_xLVL$ ) causes the  $VDCR1\langle VD_xF \rangle$  flag to be set. This flag remains set until it is cleared by software.  $VDCR1\langle VD_xF \rangle$  is not cleared to 0 when the supply voltage (VDD) becomes equal to or higher than the detection voltage ( $VD_xLVL$ ).

Once  $VDCR1\langle VD_xF \rangle$  has been set to 1, this state is retained even if  $VDCR2\langle VD_xEN \rangle$  is cleared to 0.  $VDCR1\langle VD_xF \rangle$  can only be cleared by writing a 0.

When  $VDCR2\langle VD_xEN \rangle = 1$ , a drop of the supply voltage (VDD) below the threshold voltage ( $VD_xLVL$ ) also causes the  $VDCR1\langle VD_xSF \rangle$  flag to be set to 1. When VDD rises above  $VD_xLVL$ , the  $VDCR1\langle VD_xSF \rangle$  flag is cleared to 0.

Unlike the  $VDCR1\langle VD_xF \rangle$  flag, the  $VDCR1\langle VD_xSF \rangle$  flag does not remain set and changes its state according to the relation between VDD and  $VD_xLVL$ .

Note: Depending on the voltage detect timing, the voltage detect status flag ( $VDCR1\langle VD_xSF \rangle$ ) may be set one machine cycle before the voltage detect flag ( $VDCR1\langle VD_xF \rangle$ ) is set.

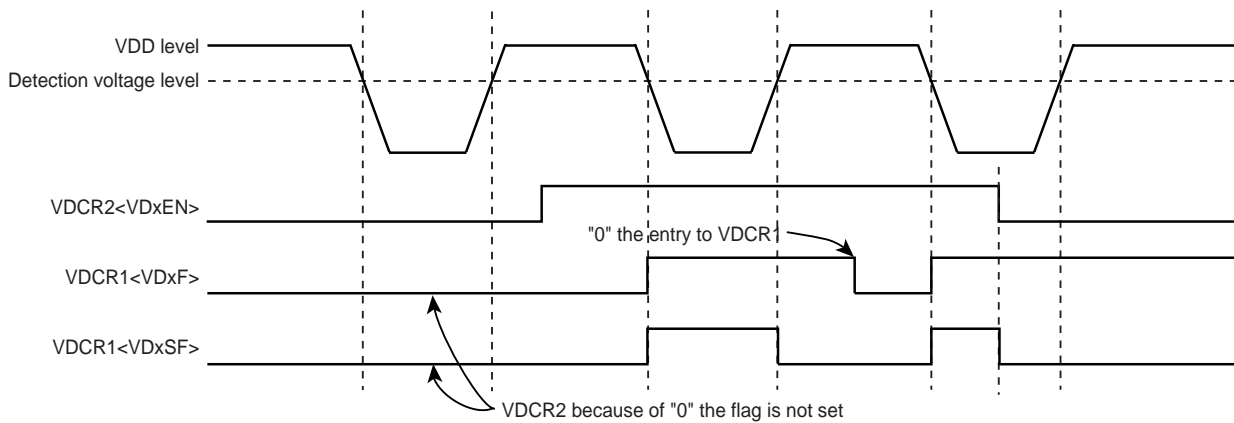


Figure 7-3 Change of voltage detection flag and voltage detection status flag

## 7.4 Setting of register

### 7.4.1 Setting Procedure for Generate an interrupt

The following shows the setting procedure for generating an INTVLTD interrupt upon detection of a low-voltage condition.

1. Clear the INTVLTD interrupt enable flag <EF4> to 0.
2. Select the threshold voltage by programming the  $VDCR1\langle VD_xLVL \rangle$  ( $x=1, 2$ ) bits.
3. Clear the  $VDCR2\langle VD_xMOD \rangle$  bit to 0 to generate an INTVLTD interrupt upon detection of a low-voltage condition.
4. Set the  $VDCR2\langle VD_xEN \rangle$  bit to 1 to enable voltage detect operation.
5. Wait for 5  $\mu$ s until the voltage detecting circuit stabilizes.
6. Ensure that the  $VDCR1\langle VD_xSF \rangle$  is 0.
7. Clear  $VDCR1\langle VD_xF \rangle$  to 0.
8. Clear the INTVLTD interrupt latch <IL4> to 0 and set the interrupt enable flag <EF4> to 1 to enable interrupts.

Note: If the selected threshold voltage ( $VD_xLVL$ ) is close to the supply voltage (VDD), INTVLTD interrupts may be generated frequently. After returning from the low-voltage detect interrupt routine, insert wait processing as appropriate with consideration given to transitions in the system supply voltage before clearing the interrupt latch.

To disable the voltage detection circuit while it is enabled with the voltage detection interrupt request, make the following setting:

1. Clear the INTVLTD interrupt enable flag <EF4> to 0.
2. Clear VDCR2<VDxEN> to 0 to disable the voltage detection operation.

Note: If the voltage detection circuit is disabled without clearing interrupt enable flag, unexpected interrupt request may occur.

### 7.4.2 Setting procedure to generate a reset

The following shows the setting procedure for generating a voltage detect x reset signal upon detection of a low-voltage condition.

1. Clear the INTVLTD interrupt enable flag <EF4> to 0.
2. Select the threshold voltage by programming the VDCR1<VDxLVL> (x=1, 2) bits.
3. Clear the VDCR2<VDxMOD> bit to 0 to generate an INTVLTD interrupt upon detection of a low-voltage condition.
4. Set the VDCR2<VDxEN> bit to 1 to enable voltage detect operation.
5. Wait for 5 us until the voltage detecting circuit stabilizes.
6. Ensure that the VDCR1<VDxSF> bit is 0.
7. Clear VDCR1<VDxF> to 0.
8. Set the VDCR2<VDxMOD> bit to 1 to generate a voltage detect x reset signal upon detection of a low-voltage condition.

Note 1: The VDCR1 and VDCR2 are only initialized by a power-on reset or an external reset input. Therefore, at the time when a low-voltage detect reset is released, if the supply voltage (VDD) is found to be below the threshold voltage (VDxLVL) before VDCR2<VDxEN> is cleared to 0, a voltage detect reset will immediately be generated.

Note 2: The voltage detect reset signal remains asserted while the supply voltage (VDD) is below the threshold voltage (LDxLVL).

To disable the voltage detection circuit while it is enabled with the voltage detection reset, make the following setting:

1. Clear the INTVLTD interrupt enable flag <EF4> to 0.
2. Clear VDCR2<VDxMOD> to 0 to set the operation mode to generate voltage detection interrupt request signals.
3. Clear VDCR2<VDxEN> to 0 to disable the voltage detection operation.

Note: If the voltage detection circuit is disabled without clearing interrupt enable flag, unexpected interrupt request may occur.

## 8. 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.

### 8.1 Watchdog Timer Configuration

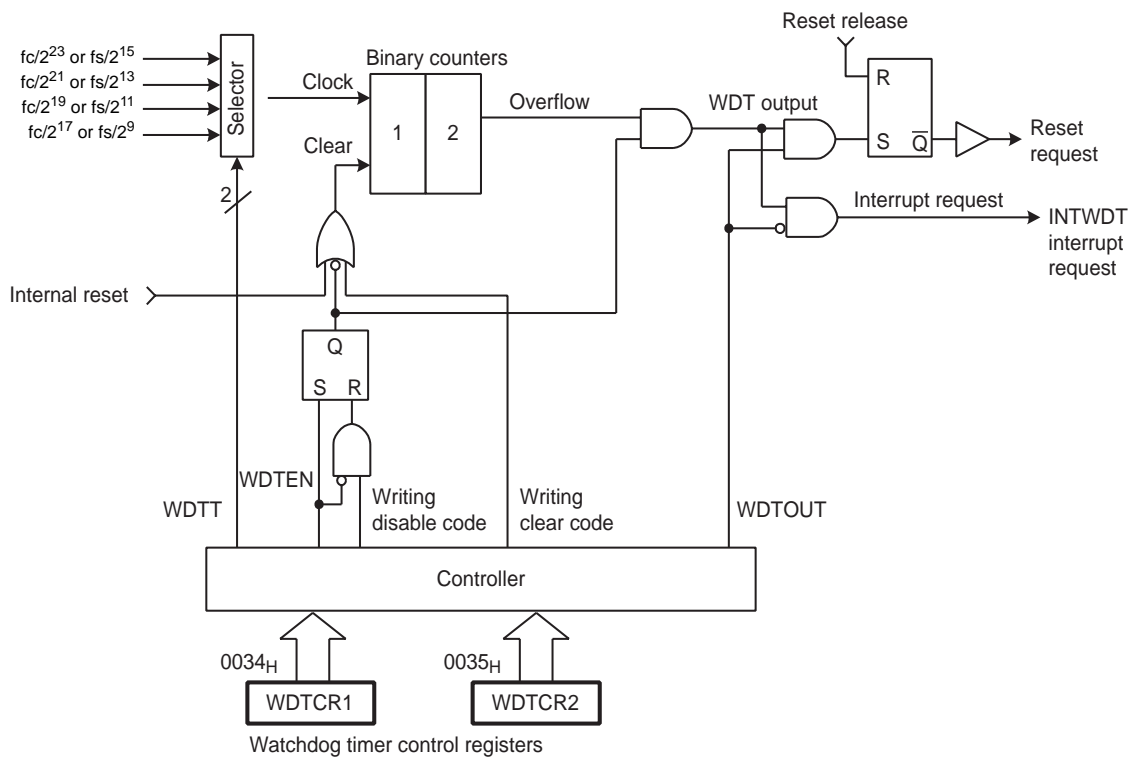


Figure 8-1 Watchdog Timer Configuration

## 8.2 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.

### 8.2.1 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.

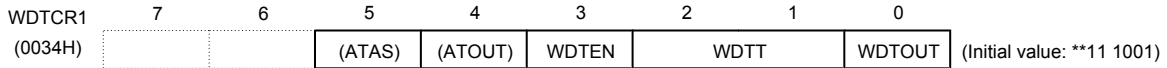
The watchdog timer temporarily stops counting in the STOP mode including the warm-up or IDLE/SLEEP mode, and automatically restarts (continues counting) when the STOP/IDLE/SLEEP mode is inactivated.

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.

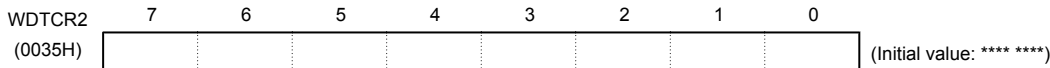
Watchdog Timer Control Register 1



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]		NORMAL1/2 mode		SLOW1/2 mode	Write only
			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 WDTEN, set the register in accordance with the procedures shown in “8.2.3 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>.

8.2.2 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.

### 8.2.3 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 micro controller.

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 counter
LDW              (WDTCR1), 0B101H   : WDTEN ← 0, WDTCR2 ← Disable code
```

Table 8-1 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

### 8.2.4 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, 023FH        : Sets the stack pointer
LD                (WDTCR1), 00001000B : WDTOUT ← 0
```

### 8.2.5 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  $24/f_c$  [s] ( $1.5 \mu\text{s}$  @  $f_c = 16.0 \text{ MHz}$ ).

Note: When a watchdog timer 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.

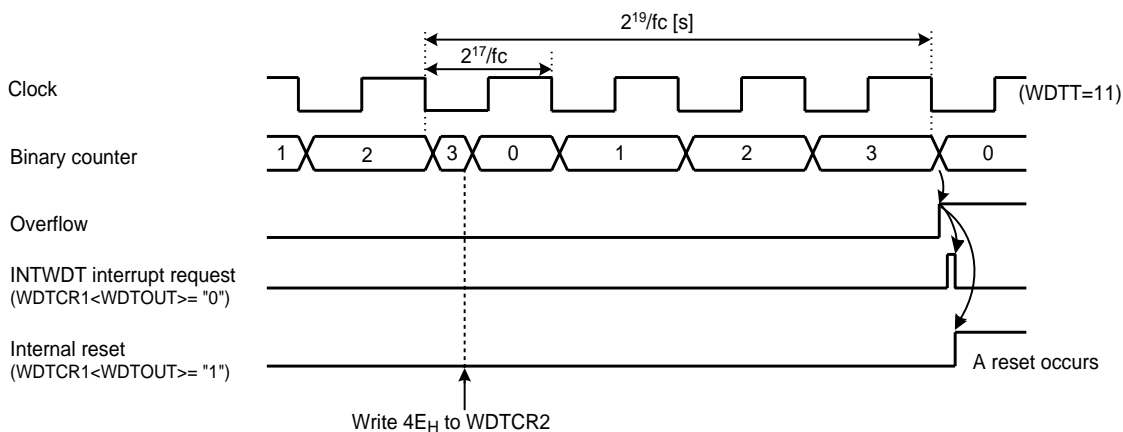
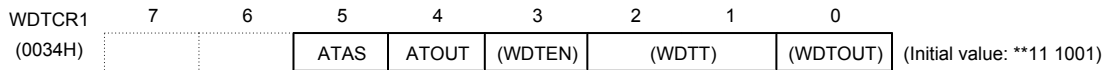


Figure 8-2 Watchdog Timer Interrupt

## 8.3 Address Trap

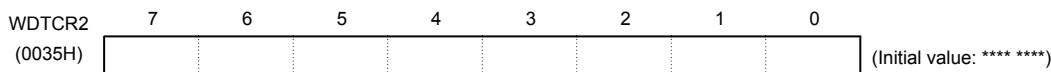
The Watchdog Timer Control Register 1 and 2 share the addresses with the control registers to generate address traps.

### 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
--------	--	--	------------

#### 8.3.1 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>.

#### 8.3.2 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>.

#### 8.3.3 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 an address trap 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.



### 8.3.4 Address Trap Reset

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

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

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



## 9. 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).

### 9.1 Time Base Timer

#### 9.1.1 Configuration

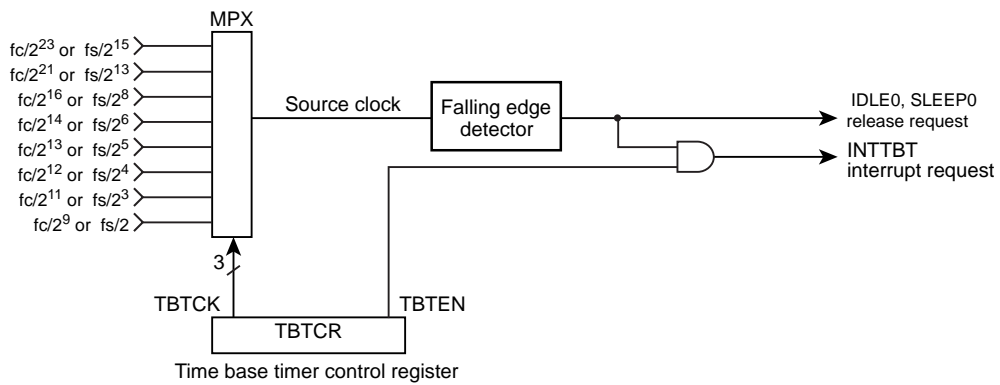


Figure 9-1 Time Base Timer configuration

#### 9.1.2 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	TBCK				(Initial Value: 0000 0000)

TBTEN	Time Base Timer enable / disable	0: Disable 1: Enable			R/W	
		NORMAL1/2, IDLE1/2 Mode		SLOW1/2 SLEEP1/2 Mode		
TBCK	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      (TBTCCR) , 00000010B      ; TBTCK ← 010
LD      (TBTCCR) , 00001010B      ; TBTEN ← 1
DI                                     ; IMF ← 0
SET     (EIRL) . 7
```

Table 9-1 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	-

### 9.1.3 Function

An INTTBT ( Time Base Timer Interrupt ) is generated on the first falling edge of source clock ( The divider output of the timing generator 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 9-2 ).

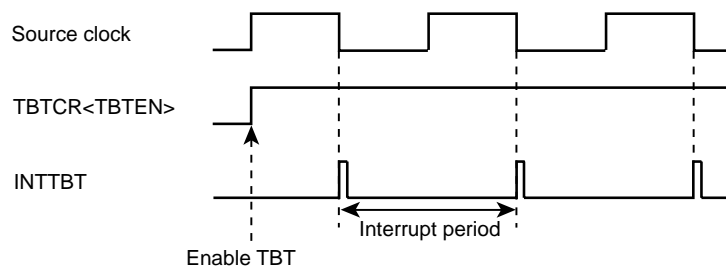


Figure 9-2 Time Base Timer Interrupt

## 9.2 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.

### 9.2.1 Configuration

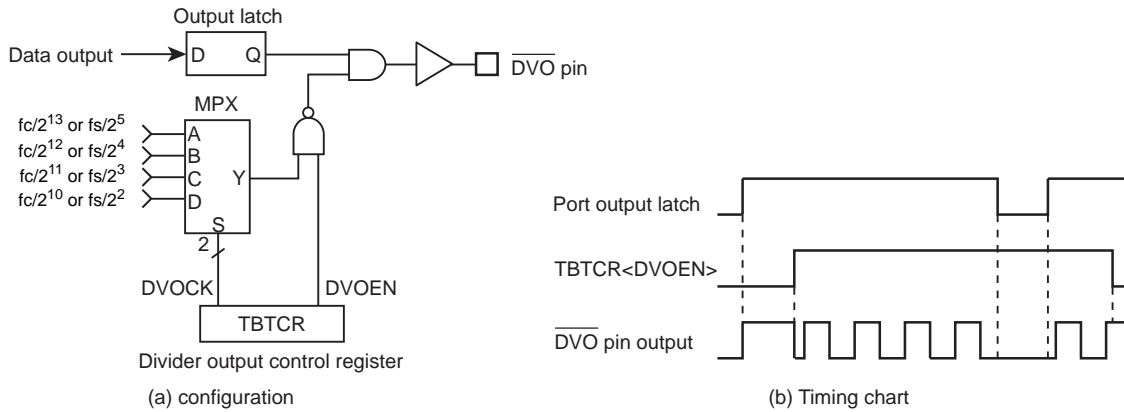


Figure 9-3 Divider Output

### 9.2.2 Control

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

#### Time Base Timer Control Register

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

DVOEN	Divider output enable / disable	0: Disable 1: Enable			R/W	
DVOCK	Divider Output ( $\overline{DVO}$ ) frequency selection: [Hz]	NORMAL1/2, IDLE1/2 Mode		SLOW1/2 SLEEP1/2 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 ( $f_c = 16.0$  MHz)

Setting port

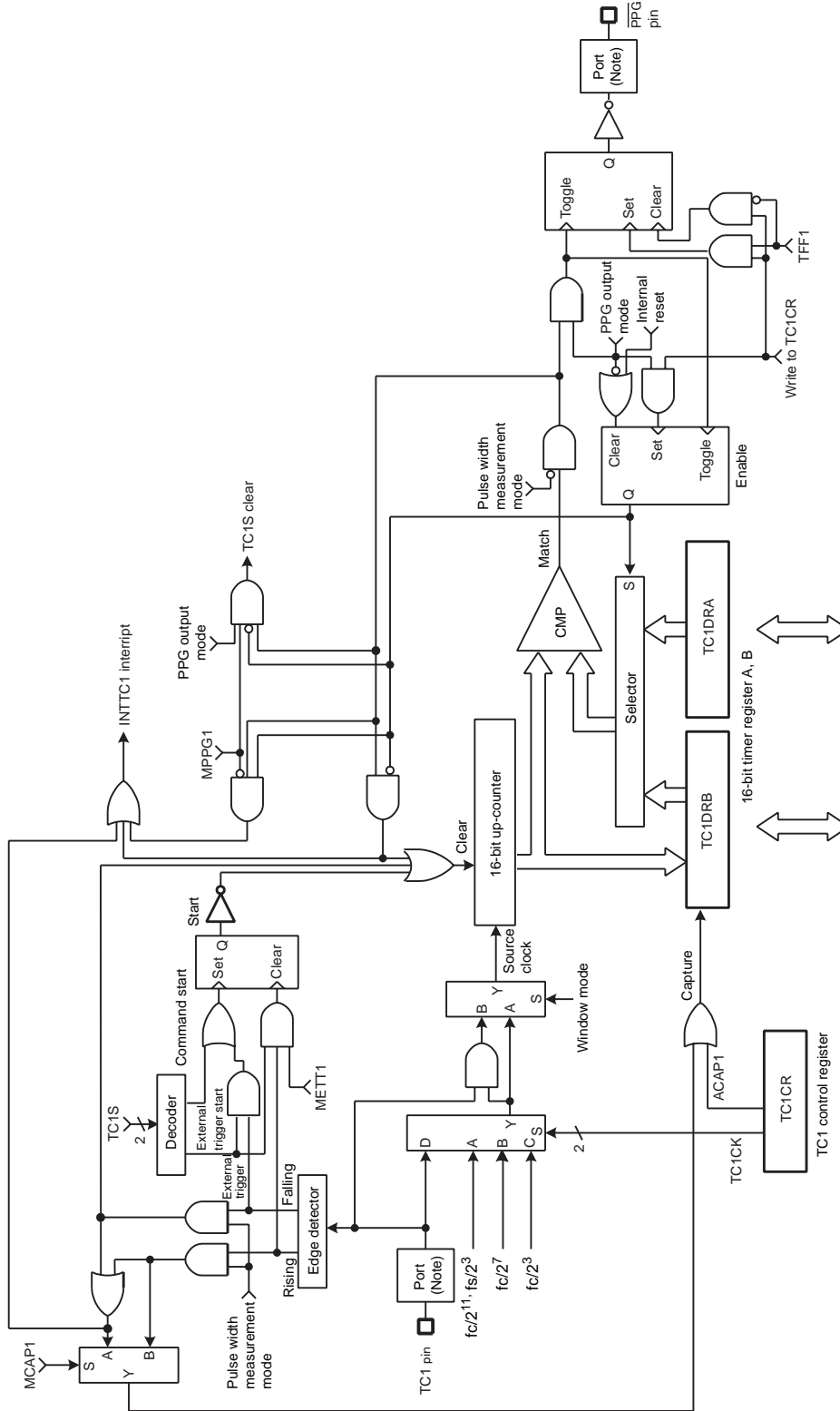
```
LD      (TBTCR) , 00000000B      ; DVOCK ← "00"
LD      (TBTCR) , 10000000B      ; DVOEN ← "1"
```

Table 9-2 Divider Output Frequency ( Example :  $f_c = 16.0$  MHz,  $f_s = 32.768$  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

# 10. 16-Bit Timer/Counter 1 (TC1)

## 10.1 Configuration



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

Figure 10-1 TimerCounter 1 (TC1)

## 10.2 Timer/Counter Control

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

### Timer Register

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TC1DRA (0011H, 0010H)	TC1DRAH (0011H) (Initial value: 1111 1111 1111 1111)								TC1DRAL (0010H) Read/Write							
TC1DRB (0013H, 0012H)	TC1DRBH (0013H) (Initial value: 1111 1111 1111 1111)								TC1DRBL (0012H) Read/Write (Write enabled only in the PPG output mode)							

### TimerCounter 1 Control Register

	7	6	5	4	3	2	1	0
TC1CR (0014H)	TFF1	ACAP1 MCAP1 METT1 MPPG1	TC1S	TC1CK	TC1M	Read/Write (Initial value: 0000 0000)		

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

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

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



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

## 10.3 Function

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

### 10.3.1 Timer mode

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

Table 10-1 Internal Source Clock for TimerCounter 1 (Example:  $f_c = 16$  MHz,  $f_s = 32.768$  kHz)

TC1CK	NORMAL1/2, IDLE1/2 mode				SLOW, SLEEP mode	
	DV7CK = 0		DV7CK = 1		Resolution [μs]	Maximum Time Setting [s]
	Resolution [μs]	Maximum Time Setting [s]	Resolution [μs]	Maximum Time Setting [s]		
00	128	8.39	244.14	16.0	244.14	16.0
01	8.0	0.524	8.0	0.524	-	-
10	0.5	32.77 m	0.5	32.77 m	-	-

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

```
LDW      (TC1DRA), 1E84H      ; Sets the timer register ( $1\text{ s} \div 2^{11}/f_c = 1\text{E84H}$ )
DI                          ; IMF= "0"
SET      (EIRH), 3           ; Enables INTTC1
EI                          ; IMF= "1"
LD       (TC1CR), 00000000B   ; Selects the source clock and mode
LD       (TC1CR), 00010000B   ; Starts TC1
```

Example 2 :Auto-capture

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

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

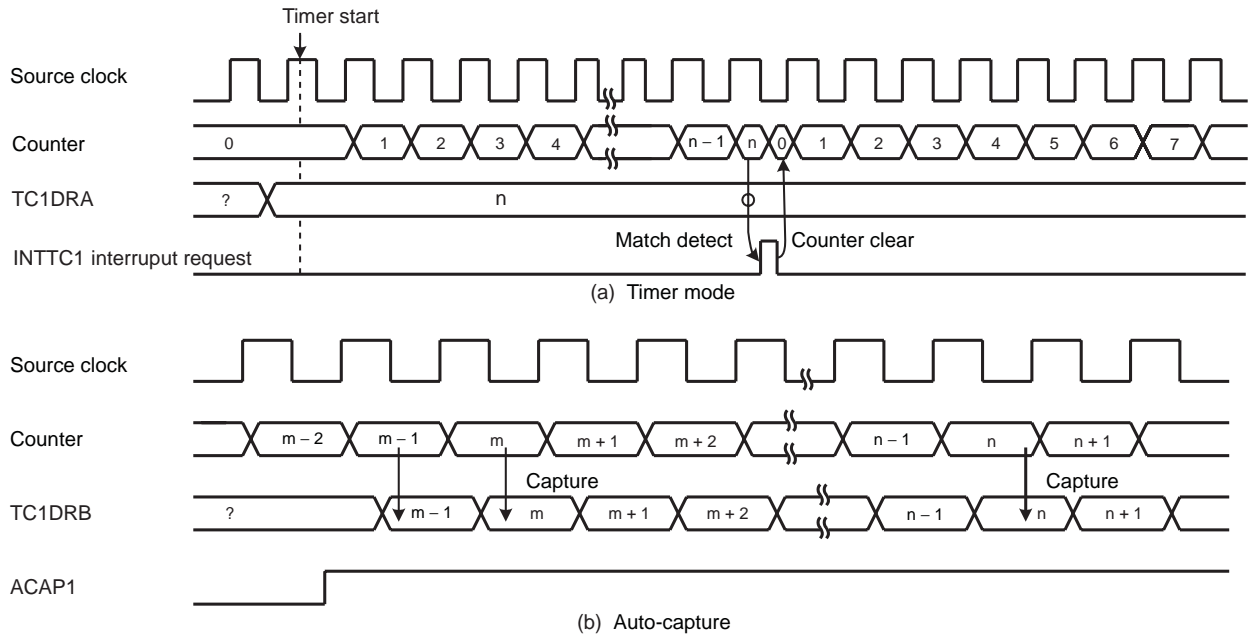


Figure 10-2 Timer Mode Timing Chart

### 10.3.2 External Trigger Timer Mode

In the external trigger timer mode, the up-counter starts counting by the input pulse triggering of the TC1 pin, and counts up at the edge of the internal clock. For the trigger edge used to start counting, either the rising or falling edge is defined in TC1CR<TC1S>.

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

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

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

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

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

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

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

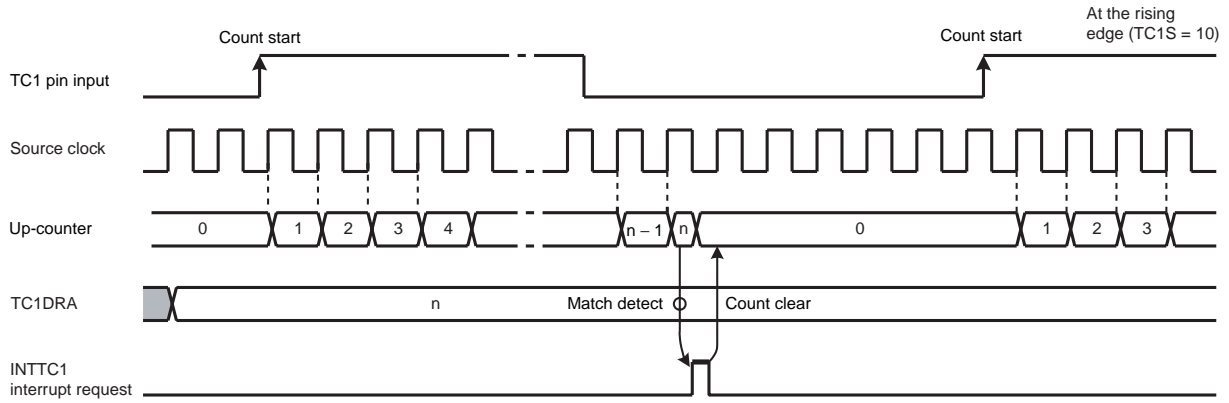
Since the TC1 pin input has the noise rejection, pulses of  $4/f_c$  [s] or less are rejected as noise. A pulse width of  $12/f_c$  [s] or more is required to ensure edge detection. The rejection circuit is turned off in the SLOW1/2 or SLEEP1/2 mode, but a pulse width of one machine cycle or more is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

Example 1 :Generating an interrupt 1 ms after the rising edge of the input pulse to the TC1 pin  
( $f_c = 16$  MHz)

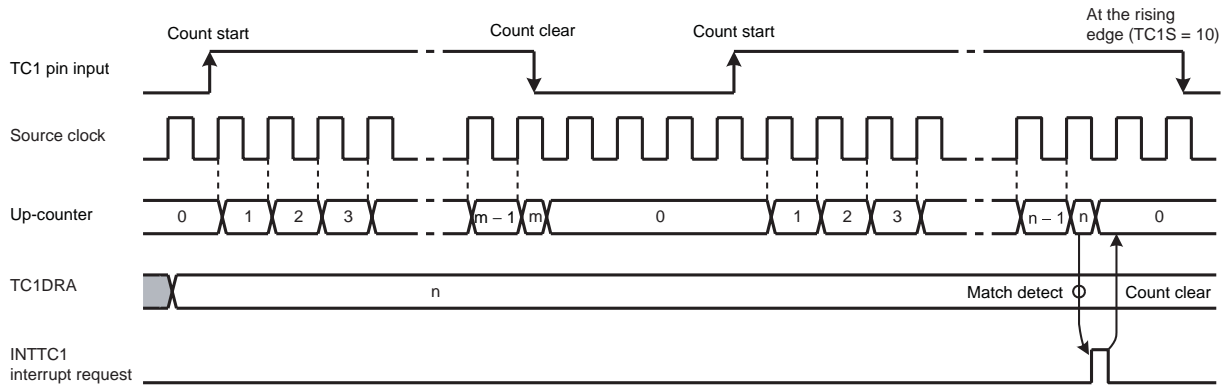
```
LDW      (TC1DRA), 007DH      ; 1ms ÷ 27/fc = 7DH
DI                          ; IMF= "0"
SET      (EIRH), 3           ; Enables INTTC1 interrupt
EI                          ; IMF= "1"
LD       (TC1CR), 00000100B   ; Selects the source clock and mode
LD       (TC1CR), 00100100B   ; Starts TC1 external trigger, METT1= 0
```

Example 2 :Generating an interrupt when the low-level pulse with 4 ms or more width is input to the TC1 pin  
( $f_c = 16$  MHz)

```
LDW      (TC1DRA), 01F4H      ; 4 ms ÷ 27/fc = 1F4H
DI                          ; IMF= "0"
SET      (EIRH), 3           ; Enables INTTC1 interrupt
EI                          ; IMF= "1"
LD       (TC1CR), 00000100B   ; Selects the source clock and mode
LD       (TC1CR), 01110100B   ; Starts TC1 external trigger, METT1= 1
```



(a) Trigger start (METT1 = 0)



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

Note:  $m < n$

Figure 10-3 External Trigger Timer Mode Timing Chart

### 10.3.3 Event Counter Mode

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

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

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

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

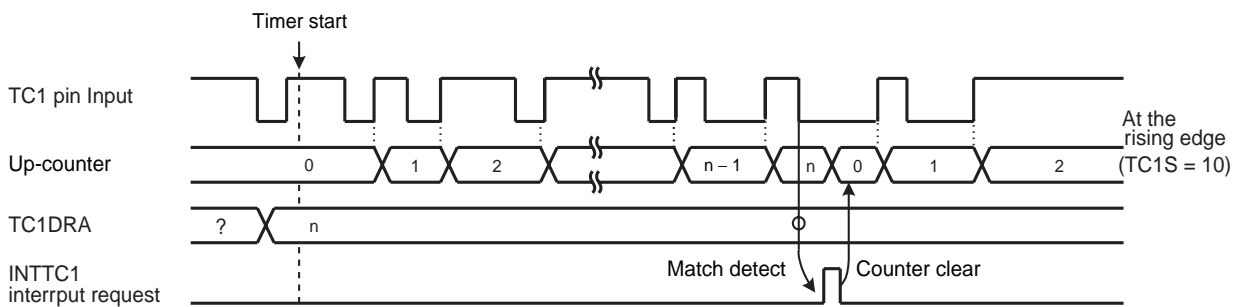


Figure 10-4 Event Counter Mode Timing Chart

Table 10-2 Input Pulse Width to TC1 Pin

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

### 10.3.4 Window Mode

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

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

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

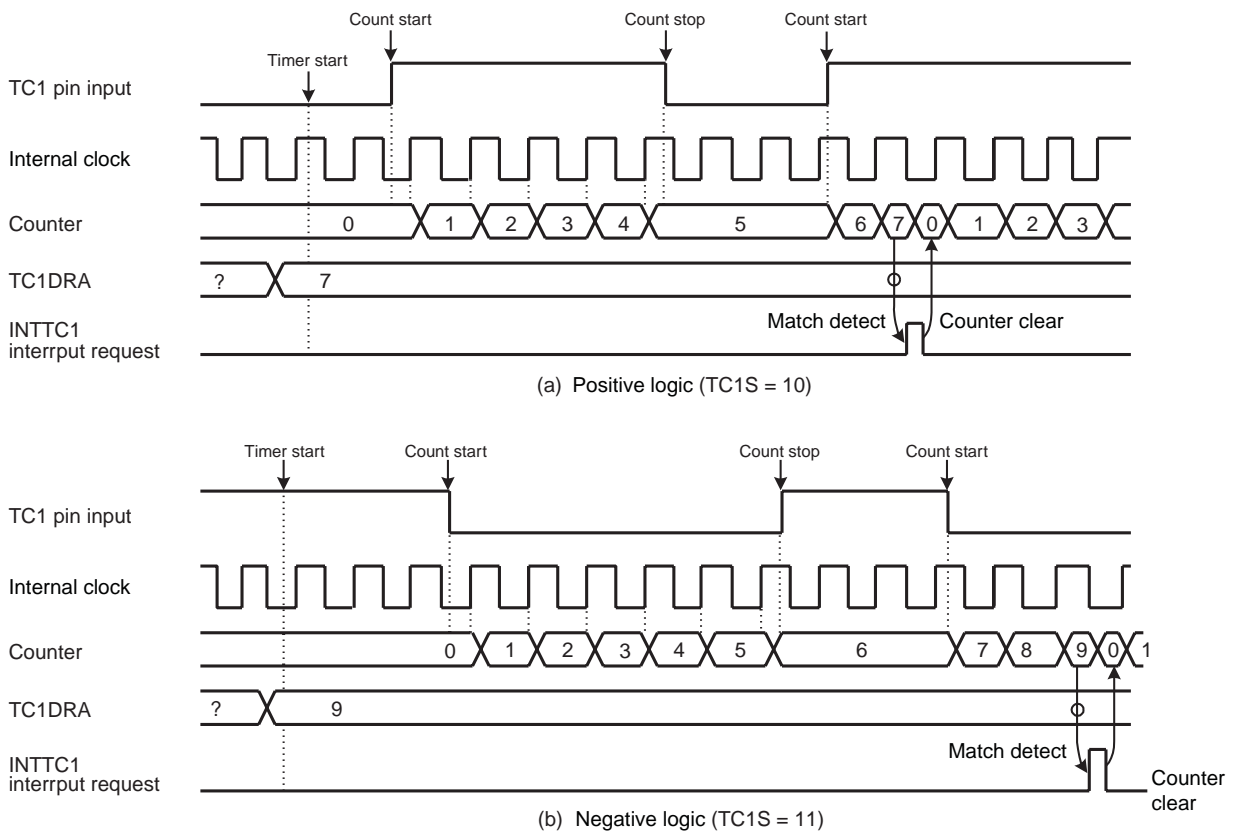


Figure 10-5 Window Mode Timing Chart

### 10.3.5 Pulse Width Measurement Mode

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

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

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

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

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

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

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

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

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

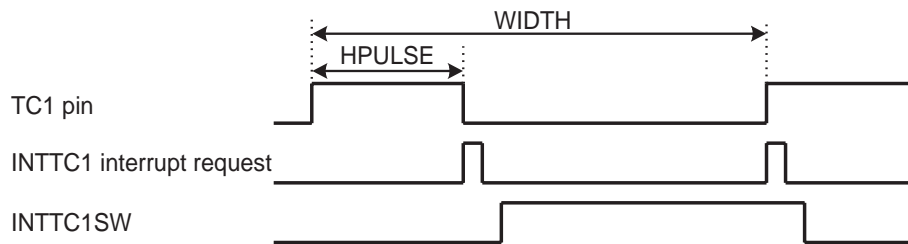
Note 3: The first captured value after the timer starts may be read incorrectly, therefore, ignore the first period captured values.

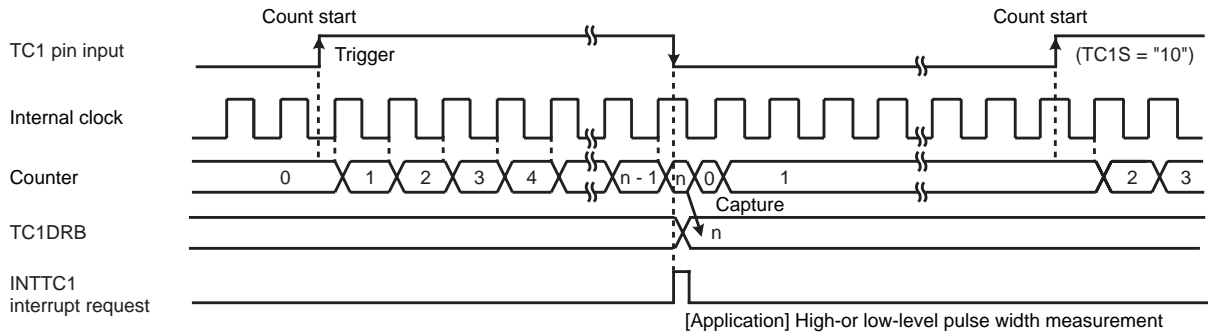


Example :Duty measurement (resolution  $f_c/2^7$  [Hz])

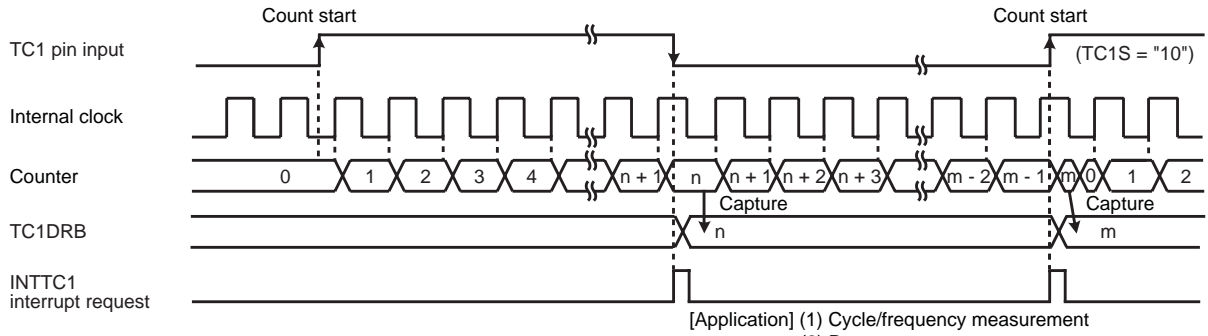
```

                CLR        (INTTC1SW). 0          ; INTTC1 service switch initial setting
                ; Address set to convert INTTC1SW at each INTTC1
                LD         (TC1CR), 00000110B     ; Sets the TC1 mode and source clock
                DI         ; IMF= "0"
                SET        (EIRH), 3             ; Enables INTTC1
                EI         ; IMF= "1"
                LD         (TC1CR), 00100110B     ; Starts TC1 with an external trigger at MCAPI = 0
                :
PINTTC1:       CPL        (INTTC1SW). 0          ; INTTC1 interrupt, inverts and tests INTTC1 service switch
                JRS        F, SINTTC1
                LD         A, (TC1DRBL)          ; Reads TC1DRB (High-level pulse width)
                LD         W,(TC1DRBH)
                LD         (HPULSE), WA         ; Stores high-level pulse width in RAM
                RETI
SINTTC1:       LD         A, (TC1DRBL)          ; Reads TC1DRB (Cycle)
                LD         W,(TC1DRBH)
                LD         (WIDTH), WA         ; Stores cycle in RAM
                :
                RETI         ; Duty calculation
                :
VINTTC1:       DW         PINTTC1              ; INTTC1 Interrupt vector
    
```





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



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

Figure 10-6 Pulse Width Measurement Mode

### 10.3.6 Programmable Pulse Generate (PPG) Output Mode

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

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

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

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

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

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

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

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

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

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

$$\text{TC1DRA} > \text{TC1DRB}$$

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

Example : Generating a pulse which is high-going for 800  $\mu\text{s}$  and low-going for 200  $\mu\text{s}$   
( $f_c = 16 \text{ MHz}$ )

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

Example :After stopping PPG, setting the PPG pin to a high-level to restart PPG  
( $f_c = 16 \text{ MHz}$ )

```

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

```

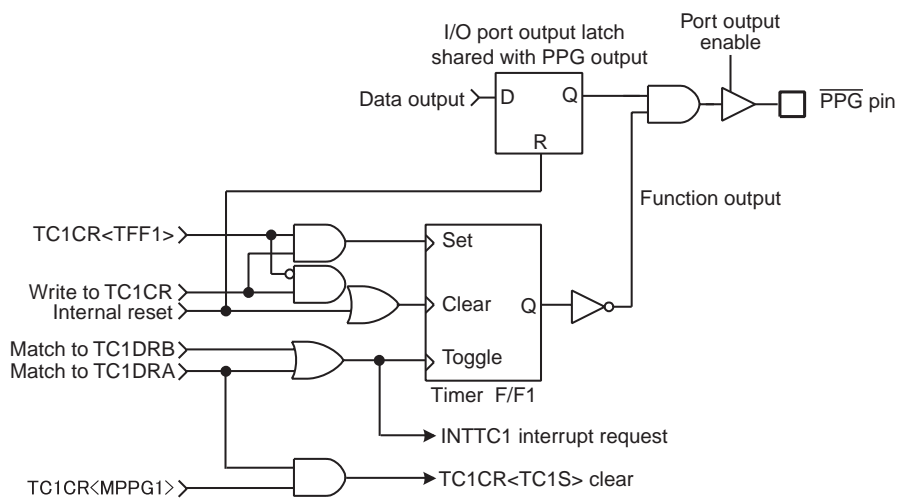
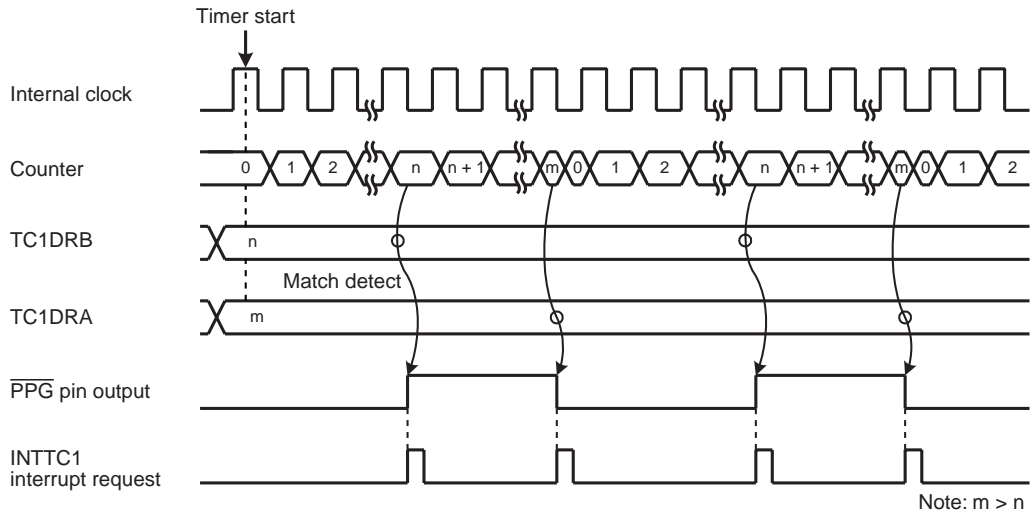
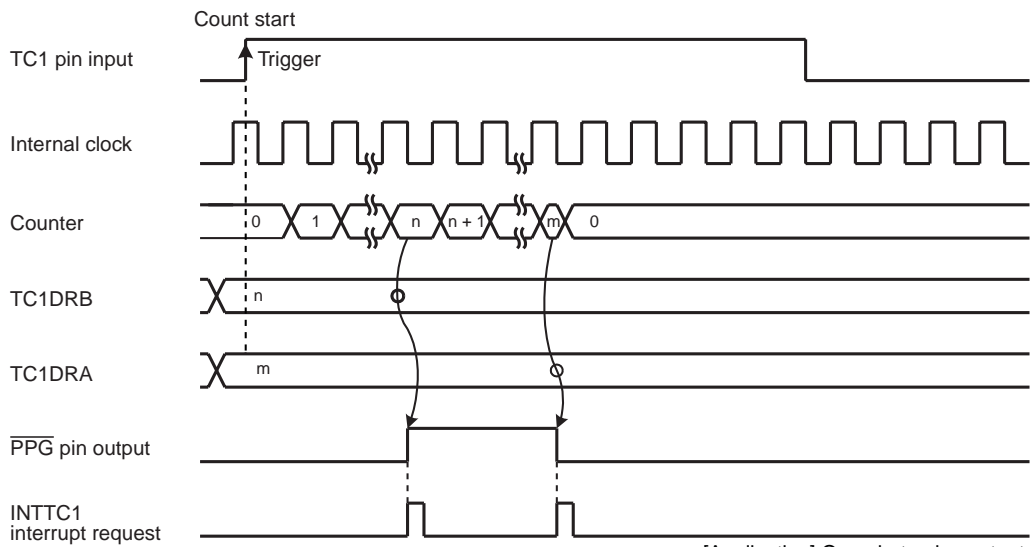


Figure 10-7  $\overline{\text{PPG}}$  Output



(a) Continuous pulse generation (TC1S = 01)



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

Figure 10-8 PPG Mode Timing Chart



# 11. 8-Bit TimerCounter (TC3, TC4)

## 11.1 Configuration

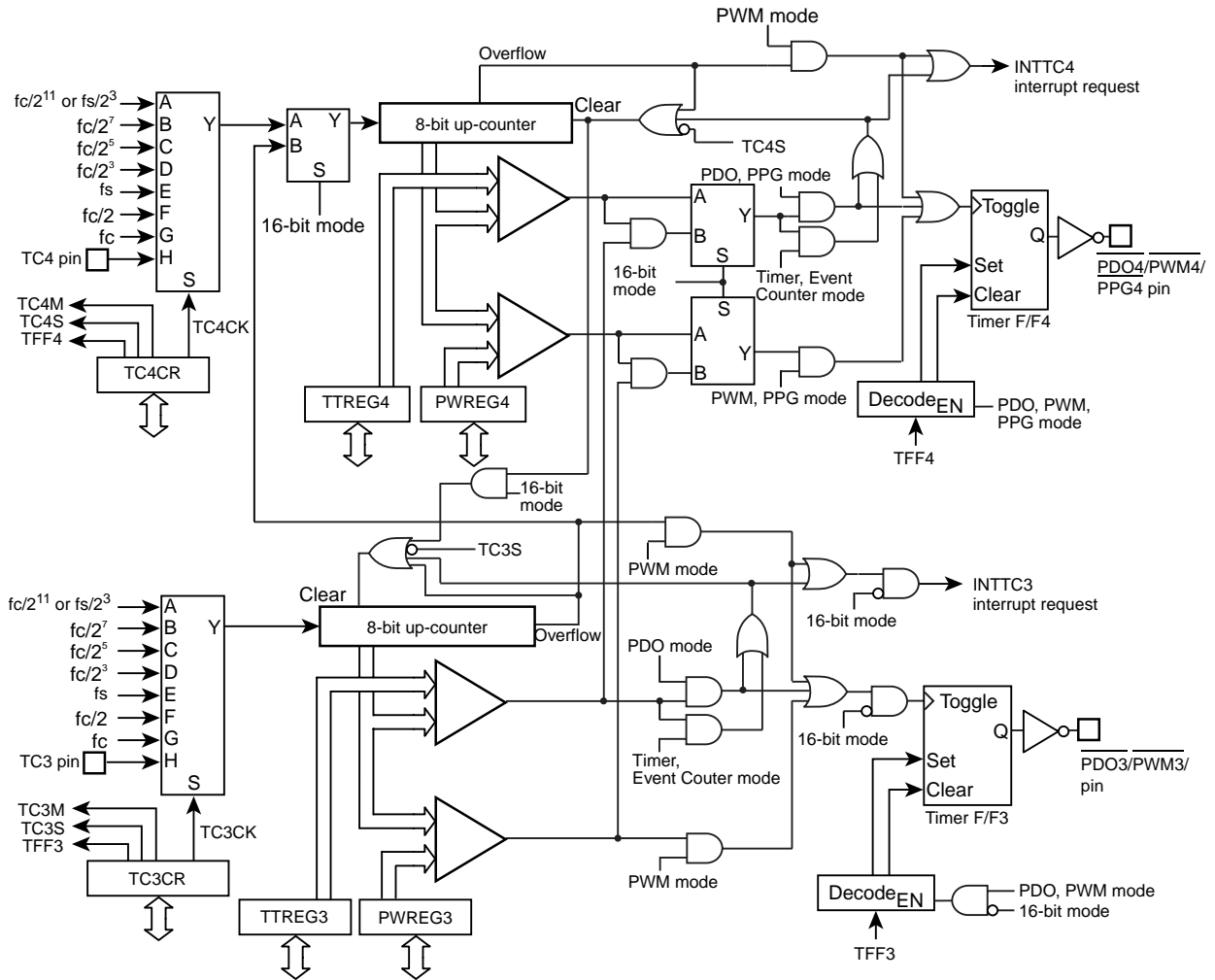
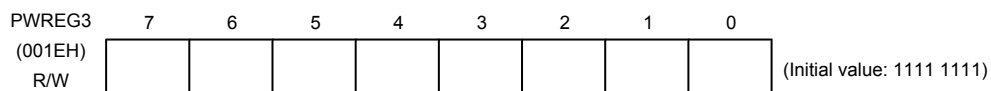
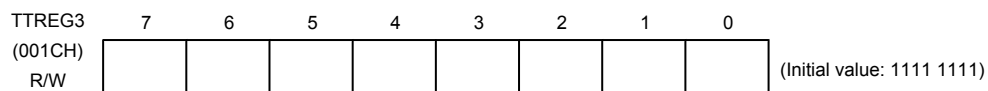


Figure 11-1 8-Bit TimerCounter 3, 4

## 11.2 TimerCounter Control

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

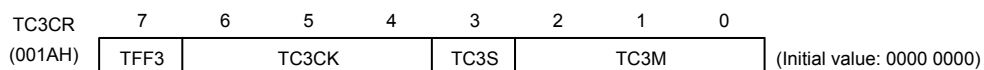
### TimerCounter 3 Timer Register



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 (Note 2,3)	0: Clear 1: Set	R/W			
TC3CK	Operating clock selection [Hz] (Note 2,3,6)	NORMAL1/2, IDLE1/2 mode		R/W		
		SLOW1/2 SLEEP1/2 mode				
		000	fc/2 <sup>11</sup>		fs/2 <sup>3</sup>	fs/2 <sup>3</sup>
		001	fc/2 <sup>7</sup>		fc/2 <sup>7</sup>	-
		010	fc/2 <sup>5</sup>		fc/2 <sup>5</sup>	-
		011	fc/2 <sup>3</sup>		fc/2 <sup>3</sup>	-
		100	fs		fs	fs
		101	fc/2		fc/2	-
TC3S	TC3 start control (Note 3)	0: Operation stop and counter clear 1: Operation start		R/W		
		110	fc (Note 8)		fc (Note 8)	fc (Note 8)
TC3M	TC3M operating mode select (Note 2,3,7)	TC3 pin input		R/W		
		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 (Note 4,5) (Each mode is selectable with TC4M.) 1**: Reserved				

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 11-1 and Table 11-2.



- Note 7: The timer register settings are limited depending on the timer operating mode. For the detailed descriptions, see Table 11-3.
- Note 8: The clock "fc" can be selected as the source clock only in 8/16 bit PWM mode and in warming-up counter mode in SLOW or SLEEP 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

TTREG4 (001DH) R/W	7	6	5	4	3	2	1	0	(Initial value: 1111 1111)

PWREG4 (001FH) R/W	7	6	5	4	3	2	1	0	(Initial value: 1111 1111)

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

TC4CR (001BH)	7	6	5	4	3	2	1	0	(Initial value: 0000 0000)
	TFF4	TC4CK		TC4S	TC4M				

TFF4	Timer F/F4 control (Note 2,3)	0: Clear 1: Set			R/W	
TC4CK	Operating clock selection [Hz] (Note 2,3,7)	NORMAL 1/2, IDLE 1/2 mode		SLOW 1/2 SLEEP 1/2 mode	R/W	
		DV7CK = 0	DV7CK = 1			
		000	fc/2 <sup>11</sup>	fs/2 <sup>3</sup>		fs/2 <sup>3</sup>
		001	fc/2 <sup>7</sup>	fc/2 <sup>7</sup>		-
		010	fc/2 <sup>5</sup>	fc/2 <sup>5</sup>		-
		011	fc/2 <sup>3</sup>	fc/2 <sup>3</sup>		-
		100	fs	fs		fs
		101	fc/2	fc/2		-
110	fc (Note 9)	fc (Note 9)	-			
111	TC4 pin input					
TC4S	TC4 start control (Note 3)	0: Operation stop and counter clear 1: Operation start			R/W	
TC4M	TC4M operating mode select (Note 2,3,8)	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 TC3 overflow signal regardless of the TC4CK setting.

Note 5: To use the TimerCounter in the 16-bit mode, select the operating mode by programming TC4M, where TC3CR<TC3M> 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 11-1 and Table 11-2.

Note 8: The timer register settings are limited depending on the timer operating mode. For the detailed descriptions, see Table 11-3.

Note 9: The clock "fc" can be selected as the source clock only in 8 bit PWM mode.

Table 11-1 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	O	O	O	O	-	-	-	-	-
8-bit event counter	-	-	-	-	-	-	-	O	O
8-bit PDO	O	O	O	O	-	-	-	-	-
8-bit PWM	O	O	O	O	O	O	O	-	-
16-bit timer	O	O	O	O	-	-	-	-	-
16-bit event counter	-	-	-	-	-	-	-	O	-
Warm-up counter	-	-	-	-	O	-	-	-	-
16-bit PWM	O	O	O	O	O	O	O	O	-
16-bit PPG	O	O	O	O	-	-	-	O	-

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: O : Available source clock

Table 11-2 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	O	-	-	-	-	-	-	-	-
8-bit event counter	-	-	-	-	-	-	-	O	O
8-bit PDO	O	-	-	-	-	-	-	-	-
8-bit PWM	O	-	-	-	O	-	-	-	-
16-bit timer	O	-	-	-	-	-	-	-	-
16-bit event counter	-	-	-	-	-	-	-	O	-
Warm-up counter	-	-	-	-	-	-	O	-	-
16-bit PWM	O	-	-	-	O	-	-	O	-
16-bit PPG	O	-	-	-	-	-	-	O	-

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: O : Available source clock

Table 11-3 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/event counter	$1 \leq (\text{TTREG4, 3}) \leq 65535$
Warm-up counter	$256 \leq (\text{TTREG4, 3}) \leq 65535$
16-bit PWM	$2 \leq (\text{PWREG4, 3}) \leq 65534$
16-bit PPG	$1 \leq (\text{PWREG4, 3}) < (\text{TTREG4, 3}) \leq 65535$ and $(\text{PWREG4, 3}) + 1 < (\text{TTREG4, 3})$

Note: n = 3 to 4

## 11.3 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.

### 11.3.1 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  $\overline{PDOj}$ ,  $\overline{PWMj}$  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 $j$  is changed while the timer is running, an expected operation may not be obtained.

Note 3:  $j = 3, 4$

Table 11-4 Source Clock for Timer Counter 3, 4 (Internal Clock)

Source Clock (Note)		SLOW1/2, SLEEP1/2 mode	Resolution		Maximum Setting Time	
NORMAL1/2, IDLE1/2 mode DV7CK = 0	DV7CK = 1		$f_c = 16 \text{ MHz}$	$f_s = 32.768 \text{ kHz}$	$f_c = 16 \text{ MHz}$	$f_s = 32.768 \text{ kHz}$
$f_c/2^{11}$ [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 $\mu\text{s}$	-

Note: In the timer mode, do not select a source clock other than those shown above.

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), 7              ; 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.
```

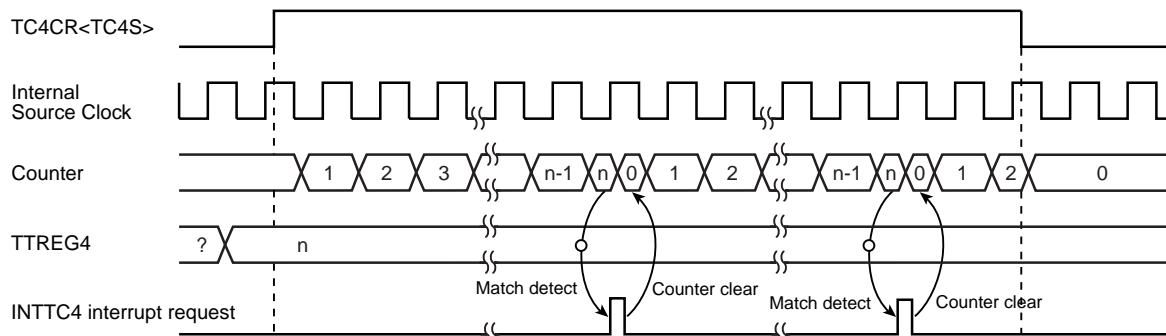


Figure 11-2 8-Bit Timer Mode Timing Chart (TC4)

### 11.3.2 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

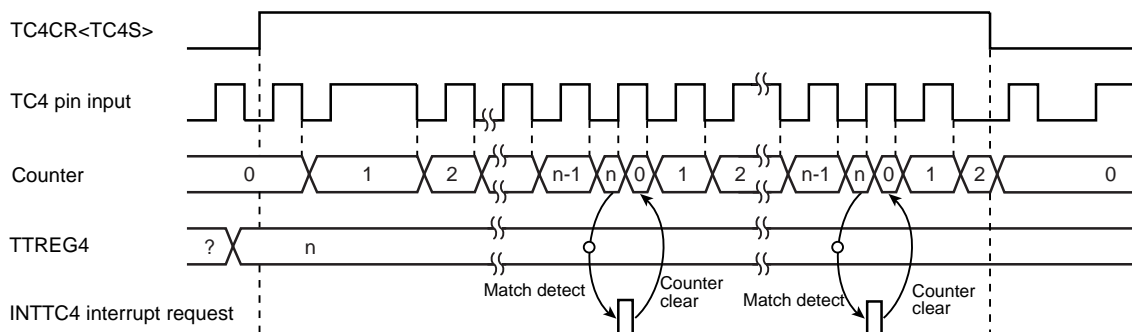


Figure 11-3 8-Bit Event Counter Mode Timing Chart (TC4)

### 11.3.3 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	; $1/1024 \div 2^7/f_c + 2 = 3DH$
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 TTREGj 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

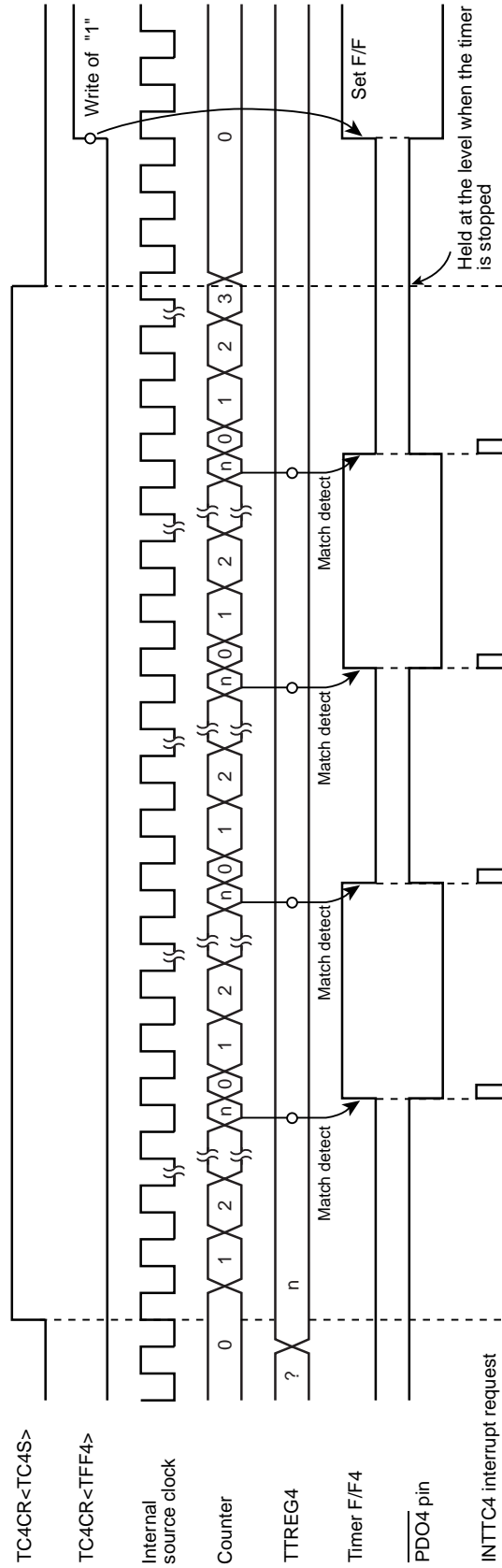


Figure 11-4 8-Bit PDO Mode Timing Chart (TC4)



### 11.3.4 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 11-5 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 <sup>11</sup> [Hz]	fs/2 <sup>3</sup> [Hz]	fs/2 <sup>3</sup> [Hz]	128 μs	244.14 μs	32.8 ms	62.5 ms
fc/2 <sup>7</sup>	fc/2 <sup>7</sup>	-	8 μs	-	2.05 ms	-
fc/2 <sup>5</sup>	fc/2 <sup>5</sup>	-	2 μs	-	512 μs	-
fc/2 <sup>3</sup>	fc/2 <sup>3</sup>	-	500 ns	-	128 μs	-
fs	fs	fs	30.5 μs	30.5 μs	7.81 ms	7.81 ms
fc/2	fc/2	-	125 ns	-	32 μs	-
fc	fc	-	62.5 ns	-	16 μs	-

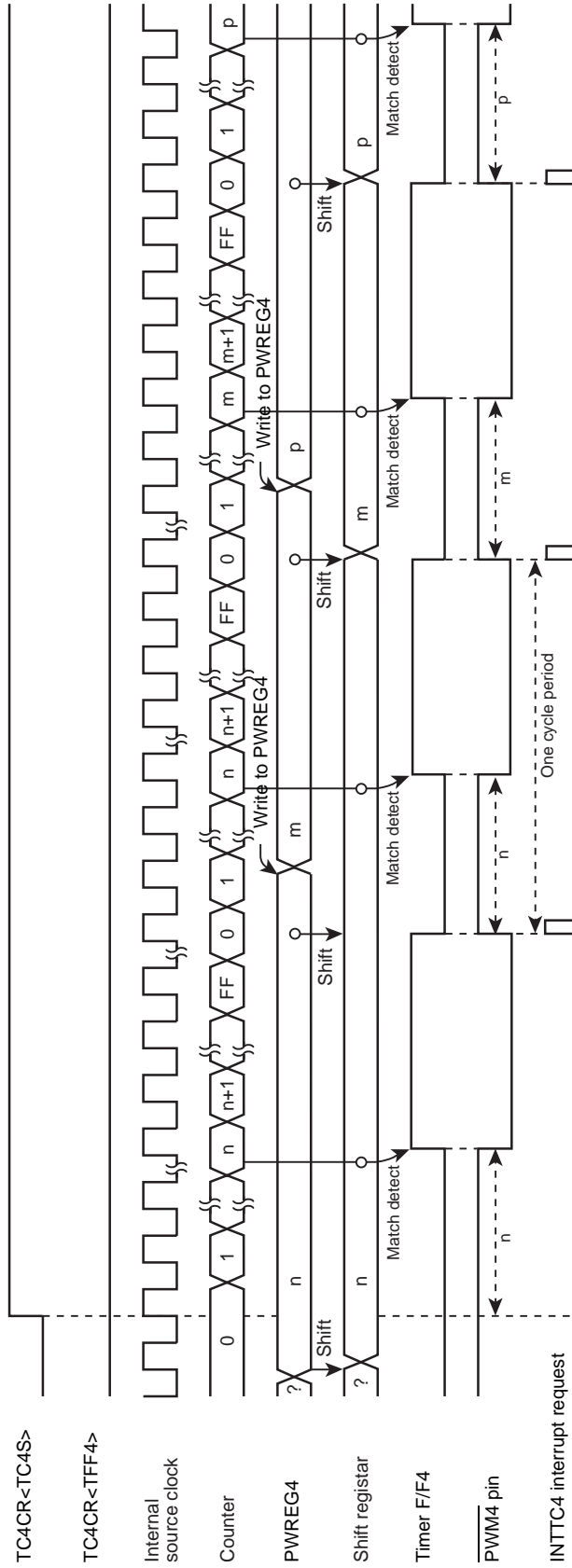


Figure 11-5 8-Bit PWM Mode Timing Chart (TC4)

### 11.3.5 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 lower byte and upper byte in this order in the timer register. (Programming only the upper or lower byte should not be attempted.)

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

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

Note 3: j = 3, 4

Table 11-6 Source Clock for 16-Bit Timer Mode

Source Clock		SLOW1/2, SLEEP1/2 mode	Resolution		Maximum Setting Time	
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 <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 ÷ 27/fc = 927CH).
DI
SET      (EIRH), 7           ; Enables INTTC4 interrupt.
EI
LD       (TC3CR), 13H        ; Sets the operating clock to fc/27, and 16-bit timer mode
                                ; (lower byte).
LD       (TC4CR), 04H        ; Sets the 16-bit timer mode (upper byte).
LD       (TC4CR), 0CH        ; Starts the timer.
```

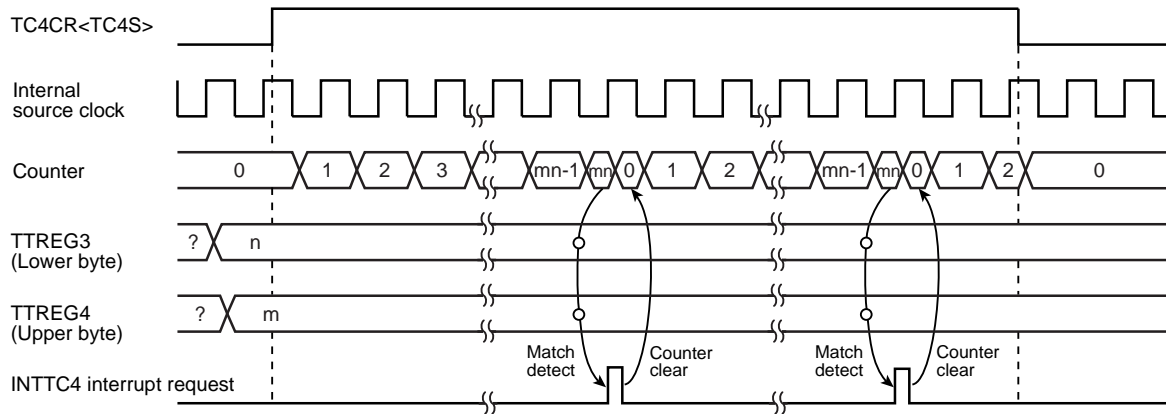


Figure 11-6 16-Bit Timer Mode Timing Chart (TC3 and TC4)

### 11.3.6 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 NORMAL1/2 or IDLE1/2 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

### 11.3.7 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 NORMAL1/2 or IDLE1/2 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 (PWREG4) 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 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 PWM4 pin to the high level when the TimerCounter is stopped

CLR (TC4CR).3 ; Stops the timer.

CLR (TC4CR).7 ; Sets the 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 fc, fc/2 or fs is selected as the source clock, a pulse is output from the PWM4 pin during the warm-up period time after exiting the STOP mode.

Table 11-7 16-Bit 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 <sup>11</sup>	fs/2 <sup>3</sup> [Hz]	fs/2 <sup>3</sup> [Hz]	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	-
fs	fs	fs	30.5 μs	30.5 μs	2 s	2 s
fc/2	fc/2	-	125 ns	-	8.2 ms	-
fc	fc	-	62.5 ns	-	4.1 ms	-

Example :Generating a pulse with 1-ms high-level width and a period of 32.768 ms (fc = 16.0 MHz)

Setting ports

- LDW (PWREG3), 07D0H ; Sets the pulse width.
- LD (TC3CR), 33H ; Sets the operating clock to fc/2<sup>3</sup>, 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.

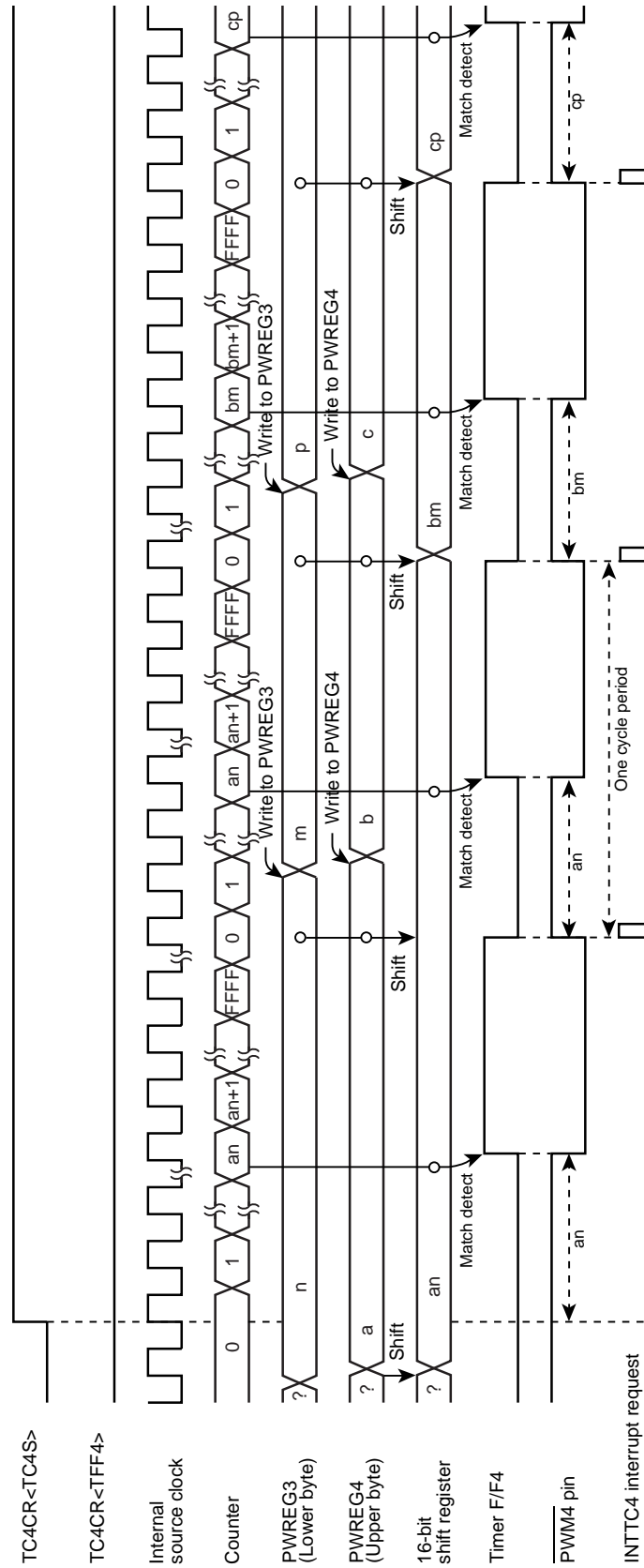


Figure 11-7 16-Bit PWM Mode Timing Chart (TC3 and TC4)

### 11.3.8 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 cascadable 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  $f_c/2^4$  Hz in the NORMAL1/2 or IDLE1/2 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  $\overline{\text{PPG4}}$  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 ( $f_c = 16.0$  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 $f_c/2^3$ , 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 PWREGi and TTREGi settings while the timer is running. Since PWREGi and TTREGi are not in the shift register configuration in the PPG mode, the new values programmed in PWREGi and TTREGi are in effect immediately after programming PWREGi and TTREGi. Therefore, if PWREGi and TTREGi are changed while the timer is running, an expected operation may not be obtained.

Note 2: When the timer is stopped during PPG output, the  $\overline{\text{PPG4}}$  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  $\overline{\text{PPG4}}$  pin to the high level when the TimerCounter is stopped

CLR (TC4CR).3 ; Stops the timer

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

Note 3: i = 3, 4

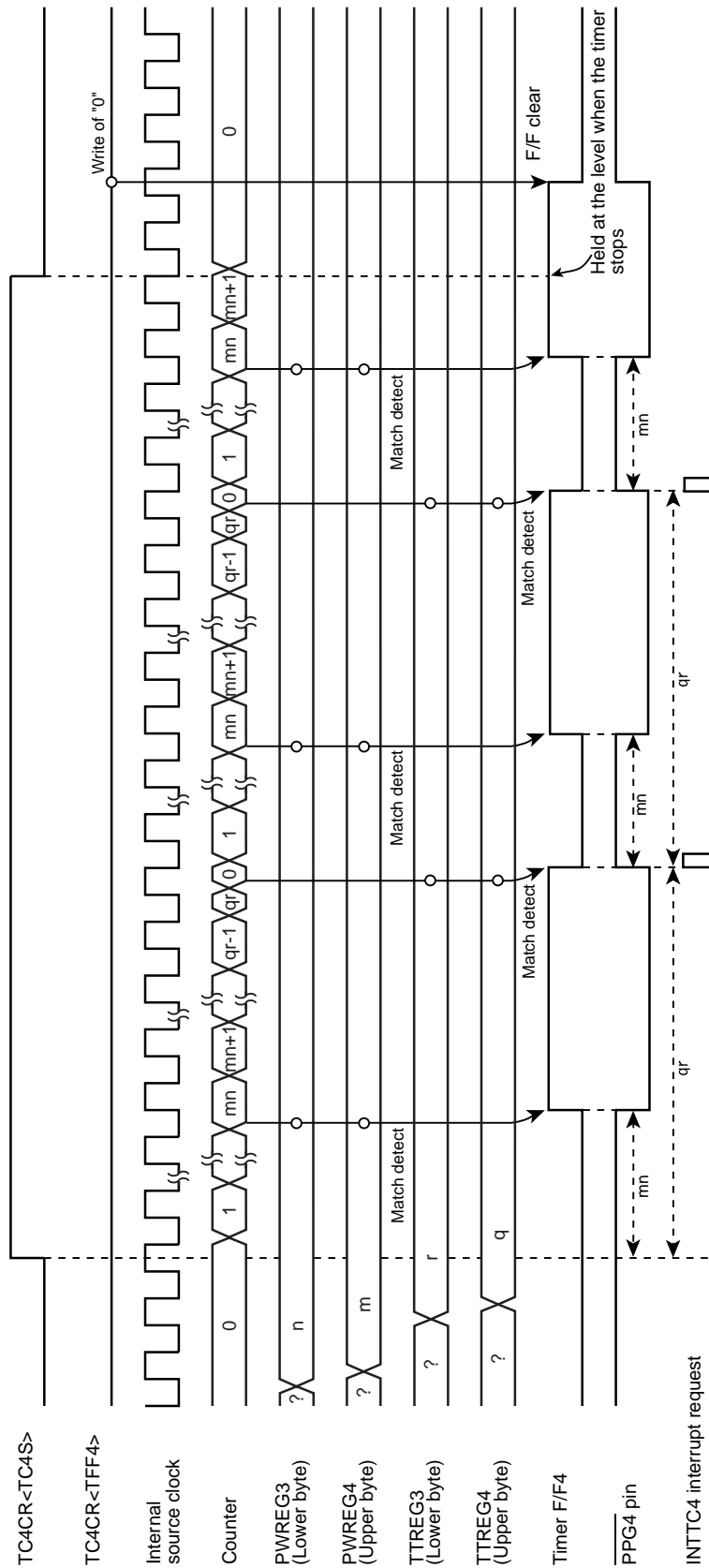


Figure 11-8 16-Bit PPG Mode Timing Chart (TC3 and TC4)



### 11.3.9 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  $\overline{PDOi}$ ,  $\overline{PWMi}$  and  $\overline{PPGi}$  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

#### 11.3.9.1 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<XEN> to 0 to stop the high-frequency clock.

Table 11-8 Setting Time of Low-Frequency Warm-Up Counter Mode (fs = 32.768 kHz)

Minimum 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.
LDW     (TTREG3), 8000H ; Sets the warm-up time.
                          ; (The warm-up time depends on the oscillator characteristic.)
DI       ; IMF ← 0
SET      (EIRH). 7      ; 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
    
```

### 11.3.9.2 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 11-9 Setting Time in High-Frequency Warm-Up Counter Mode

Minimum time Setting (TTREG4, 3 = 0100H)	Maximum time Setting (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  $f_c$ , and 16-bit mode.
LD       (TC4CR), 05H    ; Sets TFF4=0, and warm-up counter mode.
LDW     (TTREG3), 0F800H ; Sets the warm-up time.
                               ; (The warm-up time depends on the oscillator characteristic.)
DI                               ; IMF ← 0
SET     (EIRH). 7        ; 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

```

## 12. Asynchronous Serial interface (UART1)

### 12.1 Configuration

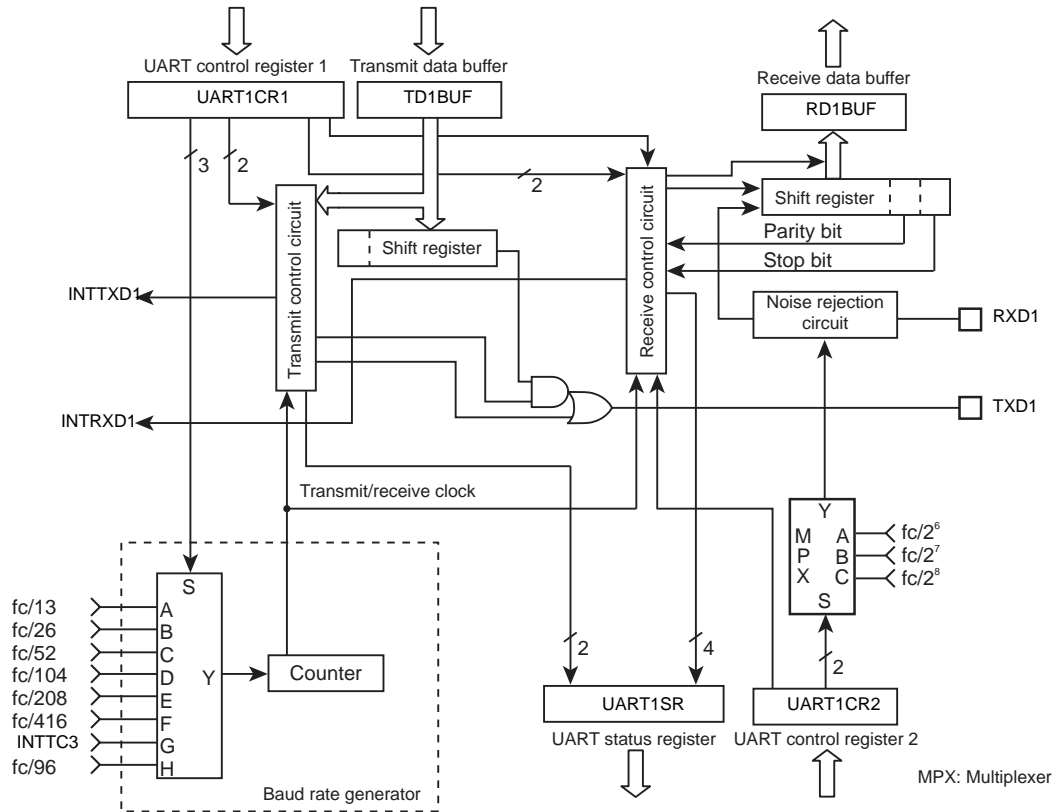


Figure 12-1 UART1 (Asynchronous Serial Interface)

## 12.2 Control

UART1 is controlled by the UART1 Control Registers (UART1CR1, UART1CR2). The operating status can be monitored using the UART status register (UART1SR).

### UART1 Control Register1

UART1CR1 (0025H)	7	6	5	4	3	2	1	0	
	TXE	RXE	STBT	EVEN	PE	BRG			(Initial value: 0000 0000)

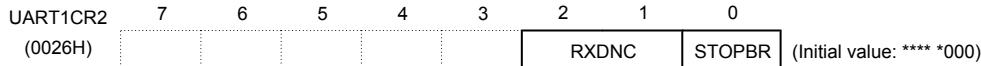
TXE	Transfer operation	0: Disable 1: Enable	Write only
RXE	Receive operation	0: Disable 1: Enable	
STBT	Transmit stop bit length	0: 1 bit 1: 2 bits	
EVEN	Even-numbered parity	0: Odd-numbered parity 1: Even-numbered parity	
PE	Parity addition	0: No parity 1: Parity	
BRG	Transmit clock select	000: $f_c/13$ [Hz] 001: $f_c/26$ 010: $f_c/52$ 011: $f_c/104$ 100: $f_c/208$ 101: $f_c/416$ 110: TC3 (Input INTTC3) 111: $f_c/96$	

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: UART1CR1<RXE> and UART1CR1<TXE> should be set to "0" before UART1CR1<BRG> is changed.

**UART1 Control Register2**

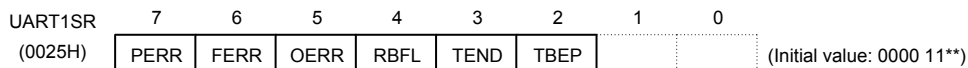


RXDNC	Selection of RXD input noise rejection time	00: No noise rejection (Hysteresis input) 01: Rejects pulses shorter than 31/fc [s] as noise 10: Rejects pulses shorter than 63/fc [s] as noise 11: Rejects pulses shorter than 127/fc [s] as noise	Write only
STOPBR	Receive stop bit length	0: 1 bit 1: 2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

Note: Settings of RXDNC are limited depending on the transfer clock specified by BRG. The combination "0" is available but please do not select the combination "-". The transfer clock is calculated by the following equation :  
 Transfer clock [Hz] = Timer/counter source clock [Hz] ÷ TTREG3 set value

BRG setting	Transfer clock [Hz]	RXDNC setting			
		00 (No noise rejection)	01 (Reject pulses shorter than 31/fc[s] as noise)	10 (Reject pulses shorter than 63/fc[s] as noise)	11 (Reject pulses shorter than 127/fc[s] as noise)
000	fc/13	O	O	O	-
110	fc/8	O	-	-	-
(When the transfer clock generated by INTTC3 is the same as the right side column)	fc/16	O	O	-	-
	fc/32	O	O	O	-
The setting except the above		O	O	O	O

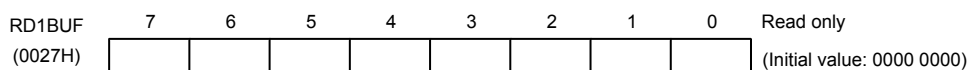
**UART1 Status Register**



PERR	Parity error flag	0: No parity error 1: Parity error	
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	Read only

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

**UART1 Receive Data Buffer**



## UART1 Transmit Data Buffer

TD1BUF	7	6	5	4	3	2	1	0	Write only
(0027H)									(Initial value: 0000 0000)

### 12.3 Transfer Data Format

In UART1, an one-bit start bit (Low level), stop bit (Bit length selectable at high level, by UART1CR1<STBT>), and parity (Select parity in UART1CR1<PE>; even- or odd-numbered parity by UART1CR1<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	11
0	0											
0	1											
1	0											
1	1											

Figure 12-2 Transfer Data Format

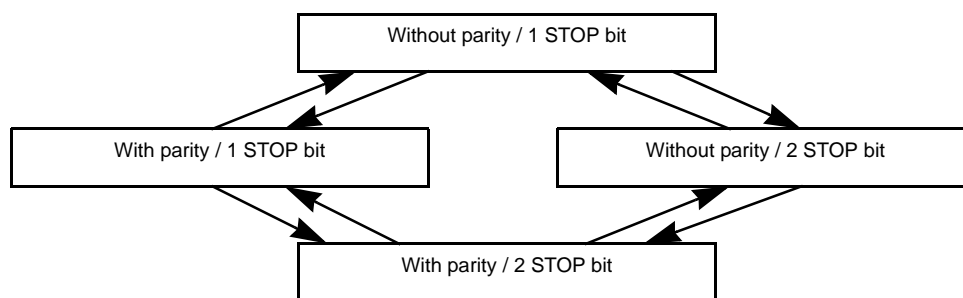


Figure 12-3 Caution on Changing Transfer Data Format

Note: In order to switch the transfer data format, perform transmit operations in the above Figure 12-3 sequence except for the initial setting.

## 12.4 Transfer Rate

The baud rate of UART1 is set of UART1CR1<BRG>. The example of the baud rate are shown as follows.

Table 12-1 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 TC3 is used as the UART1 transfer rate (when UART1CR1<BRG> = “110”), the transfer clock and transfer rate are determined as follows:

$$\text{Transfer clock [Hz]} = \text{TC3 source clock [Hz]} / \text{TTREG3 setting value}$$

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

## 12.5 Data Sampling Method

The UART1 receiver keeps sampling input using the clock selected by UART1CR1<BRG> until a start bit is detected in RXD1 pin input. RT clock starts detecting “L” level of the RXD1 pin. Once a start bit is detected, the start bit, data bits, stop bit(s), and parity bit are sampled at three times of RT7, RT8, and RT9 during one receiver clock interval (RT clock). (RT0 is the position where the bit supposedly starts.) Bit is determined according to majority rule (The data are the same twice or more out of three samplings).

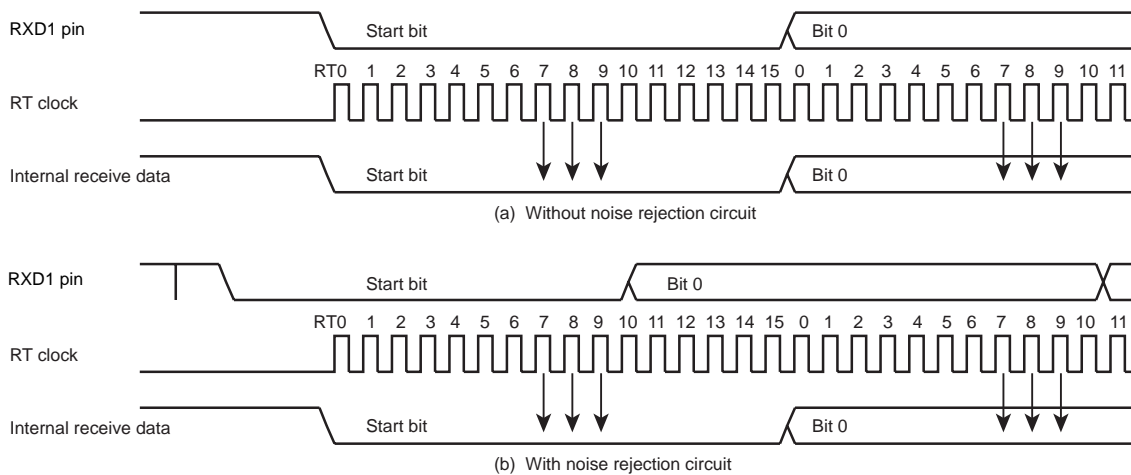


Figure 12-4 Data Sampling Method



## 12.6 STOP Bit Length

Select a transmit stop bit length (1 bit or 2 bits) by UART1CR1<STBT>.

## 12.7 Parity

Set parity / no parity by UART1CR1<PE> and set parity type (Odd- or Even-numbered) by UART1CR1<EVEN>.

## 12.8 Transmit/Receive Operation

### 12.8.1 Data Transmit Operation

Set UART1CR1<TXE> to “1”. Read UART1SR to check UART1SR<TBEP> = “1”, then write data in TD1BUF (Transmit data buffer). Writing data in TD1BUF zero-clears UART1SR<TBEP>, transfers the data to the transmit shift register and the data are sequentially output from the TXD1 pin. The data output include a one-bit start bit, stop bits whose number is specified in UART1CR1<STBT> and a parity bit if parity addition is specified. Select the data transfer baud rate using UART1CR1<BRG>. When data transmit starts, transmit buffer empty flag UART1SR<TBEP> is set to “1” and an INTTXD1 interrupt is generated.

While UART1CR1<TXE> = “0” and from when “1” is written to UART1CR1<TXE> to when send data are written to TD1BUF, the TXD1 pin is fixed at high level. When transmitting data, first read UART1SR, then write data in TD1BUF. Otherwise, UART1SR<TBEP> is not zero-cleared and transmit does not start.

### 12.8.2 Data Receive Operation

Set UART1CR1<RXE> to “1”. When data are received via the RXD1 pin, the receive data are transferred to RD1BUF (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 RD1BUF (Receive data buffer). Then the receive buffer full flag UART1SR<RBFL> is set and an INTRXD1 interrupt is generated. Select the data transfer baud rate using UART1CR1<BRG>.

If an overrun error (OERR) occurs when data are received, the data are not transferred to RD1BUF (Receive data buffer) but discarded; data in the RD1BUF are not affected.

Note: When a receive operation is disabled by setting UART1CR1<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.

## 12.9 Status Flag

### 12.9.1 Parity Error

When parity determined using the receive data bits differs from the received parity bit, the parity error flag UART1SR<PERR> is set to “1”. The UART1SR<PERR> is cleared to “0” when the RD1BUF is read after reading the UART1SR.

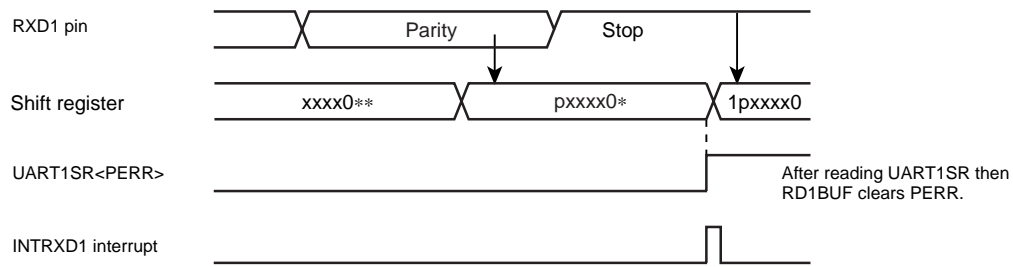


Figure 12-5 Generation of Parity Error

### 12.9.2 Framing Error

When “0” is sampled as the stop bit in the receive data, framing error flag UART1SR<FERR> is set to “1”. The UART1SR<FERR> is cleared to “0” when the RD1BUF is read after reading the UART1SR.

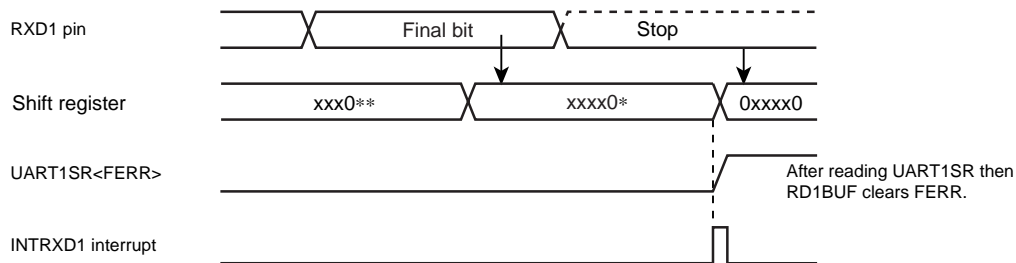


Figure 12-6 Generation of Framing Error

### 12.9.3 Overrun Error

When all bits in the next data are received while unread data are still in RD1BUF, overrun error flag UART1SR<OERR> is set to “1”. In this case, the receive data is discarded; data in RD1BUF are not affected. The UART1SR<OERR> is cleared to “0” when the RD1BUF is read after reading the UART1SR.

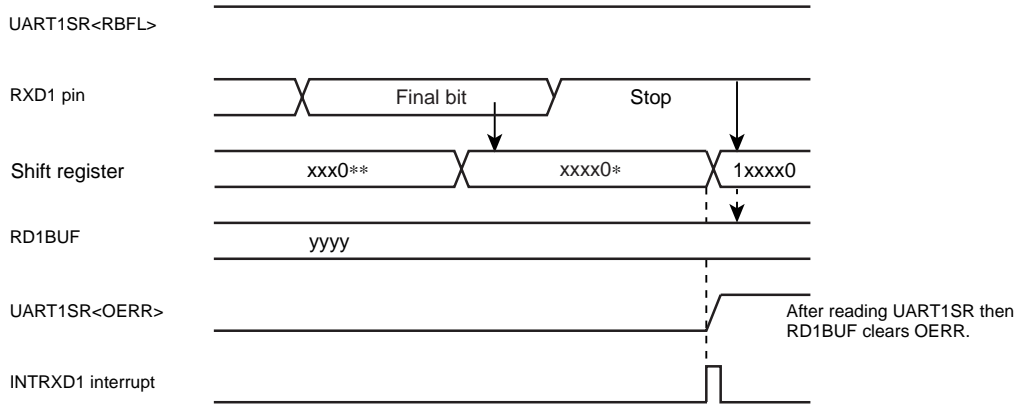


Figure 12-7 Generation of Overrun Error

Note: Receive operations are disabled until the overrun error flag UART1SR<OERR> is cleared.

### 12.9.4 Receive Data Buffer Full

Loading the received data in RD1BUF sets receive data buffer full flag UART1SR<RBFL> to "1". The UART1SR<RBFL> is cleared to "0" when the RD1BUF is read after reading the UART1SR.

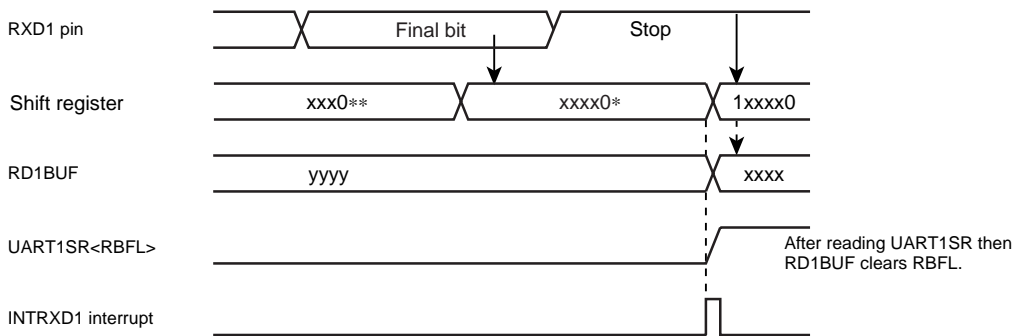


Figure 12-8 Generation of Receive Data Buffer Full

Note: If the overrun error flag UART1SR<OERR> is set during the period between reading the UART1SR and reading the RD1BUF, it cannot be cleared by only reading the RD1BUF. Therefore, after reading the RD1BUF, read the UART1SR again to check whether or not the overrun error flag which should have been cleared still remains set.

### 12.9.5 Transmit Data Buffer Empty

When no data is in the transmit buffer TD1BUF, that is, when data in TD1BUF are transferred to the transmit shift register and data transmit starts, transmit data buffer empty flag UART1SR<TBEP> is set to "1". The UART1SR<TBEP> is cleared to "0" when the TD1BUF is written after reading the UART1SR.

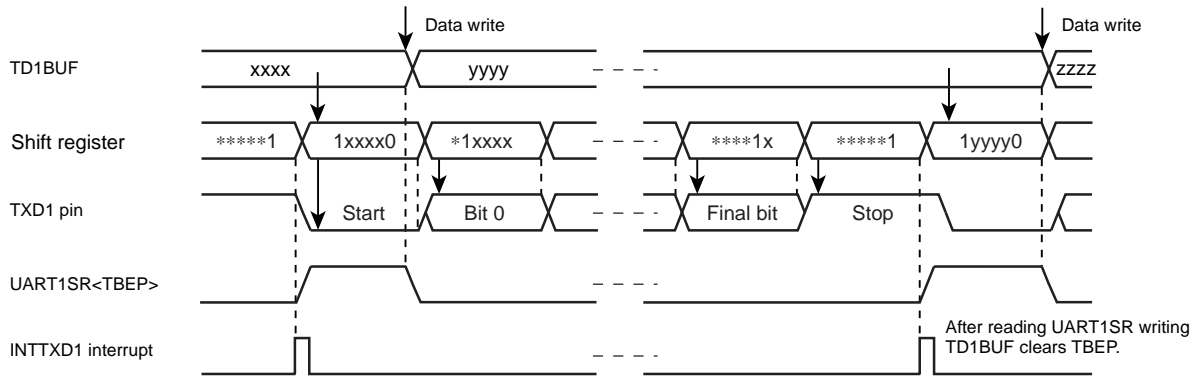


Figure 12-9 Generation of Transmit Data Buffer Empty

### 12.9.6 Transmit End Flag

When data are transmitted and no data is in TD1BUF (UART1SR<TBEP> = "1"), transmit end flag UART1SR<TEND> is set to "1". The UART1SR<TEND> is cleared to "0" when the data transmit is started after writing the TD1BUF.

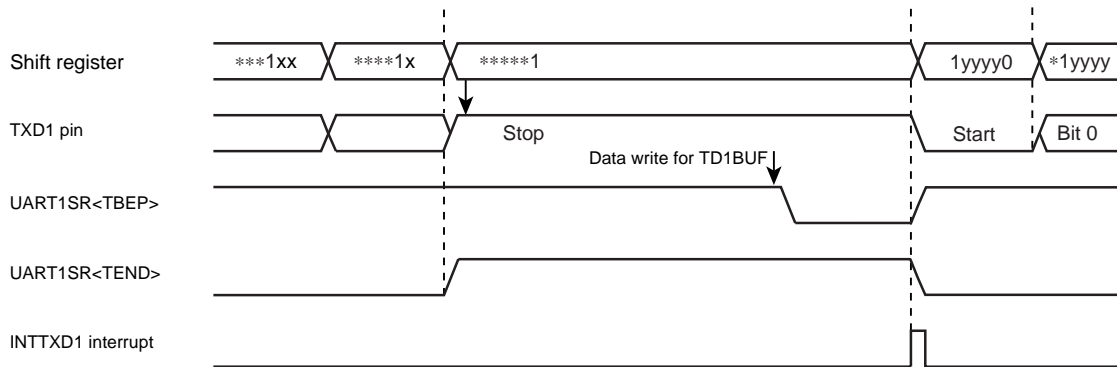


Figure 12-10 Generation of Transmit End Flag and Transmit Data Buffer Empty

### 13. Asynchronous Serial interface (UART2)

#### 13.1 Configuration

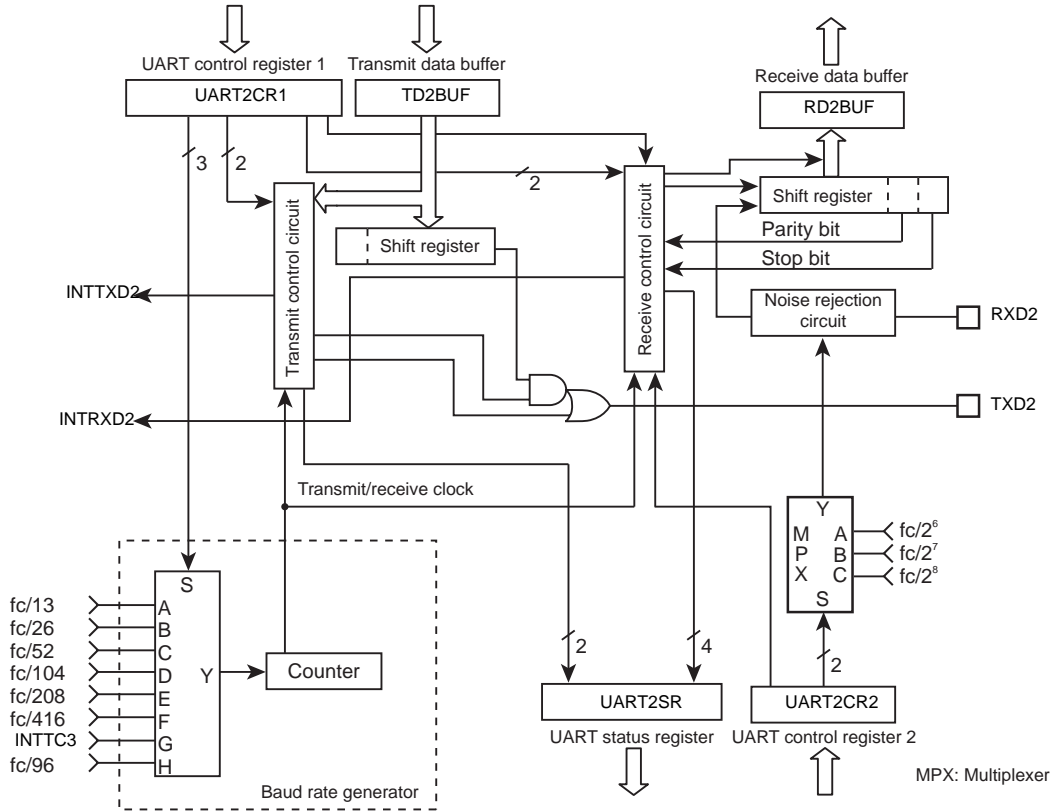


Figure 13-1 UART2 (Asynchronous Serial Interface)

## 13.2 Control

UART2 is controlled by the UART2 Control Registers (UART2CR1, UART2CR2). The operating status can be monitored using the UART status register (UART2SR).

### UART2 Control Register1

UART2CR1 (0022H)	7	6	5	4	3	2	1	0	
	TXE	RXE	STBT	EVEN	PE	BRG			(Initial value: 0000 0000)

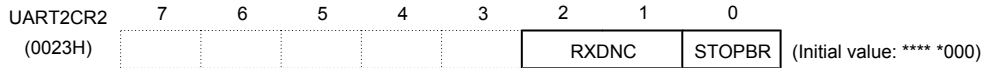
TXE	Transfer operation	0: Disable 1: Enable	Write only
RXE	Receive operation	0: Disable 1: Enable	
STBT	Transmit stop bit length	0: 1 bit 1: 2 bits	
EVEN	Even-numbered parity	0: Odd-numbered parity 1: Even-numbered parity	
PE	Parity addition	0: No parity 1: Parity	
BRG	Transmit clock select	000: $f_c/13$ [Hz] 001: $f_c/26$ 010: $f_c/52$ 011: $f_c/104$ 100: $f_c/208$ 101: $f_c/416$ 110: TC3 (Input INTTC3) 111: $f_c/96$	

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: UART2CR1<RXE> and UART2CR1<TXE> should be set to "0" before UART2CR1<BRG> is changed.

**UART2 Control Register2**

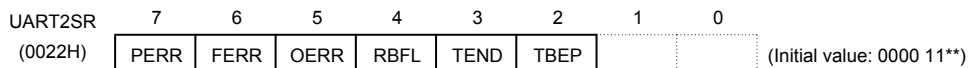


RXDNC	Selection of RXD input noise rejection time	00: No noise rejection (Hysteresis input) 01: Rejects pulses shorter than 31/fc [s] as noise 10: Rejects pulses shorter than 63/fc [s] as noise 11: Rejects pulses shorter than 127/fc [s] as noise	Write only
STOPBR	Receive stop bit length	0: 1 bit 1: 2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

Note: Settings of RXDNC are limited depending on the transfer clock specified by BRG. The combination "0" is available but please do not select the combination "-". The transfer clock is calculated by the following equation :  
 Transfer clock [Hz] = Timer/counter source clock [Hz] ÷ TTREG3 set value

BRG setting	Transfer clock [Hz]	RXDNC setting			
		00 (No noise rejection)	01 (Reject pulses shorter than 31/fc[s] as noise)	10 (Reject pulses shorter than 63/fc[s] as noise)	11 (Reject pulses shorter than 127/fc[s] as noise)
000	fc/13	O	O	O	-
110 <small>(When the transfer clock generated by INTTC3 is the same as the right side column)</small>	fc/8	O	-	-	-
	fc/16	O	O	-	-
	fc/32	O	O	O	-
The setting except the above		O	O	O	O

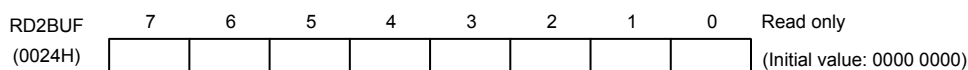
**UART2 Status Register**



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

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

**UART2 Receive Data Buffer**



## UART2 Transmit Data Buffer

TD2BUF	7	6	5	4	3	2	1	0	Write only
(0024H)									(Initial value: 0000 0000)



### 13.3 Transfer Data Format

In UART2, an one-bit start bit (Low level), stop bit (Bit length selectable at high level, by UART2CR1<STBT>), and parity (Select parity in UART2CR1<PE>; even- or odd-numbered parity by UART2CR1<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	11
0	0											
0	1											
1	0											
1	1											

Figure 13-2 Transfer Data Format

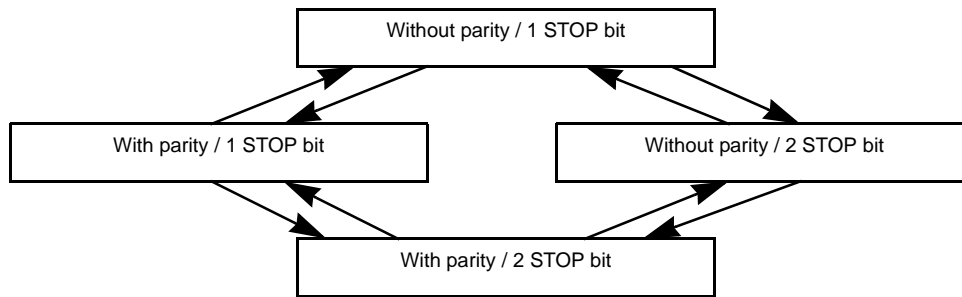


Figure 13-3 Caution on Changing Transfer Data Format

Note: In order to switch the transfer data format, perform transmit operations in the above Figure 13-3 sequence except for the initial setting.

## 13.4 Transfer Rate

The baud rate of UART2 is set of UART2CR1<BRG>. The example of the baud rate are shown as follows.

Table 13-1 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 TC3 is used as the UART2 transfer rate (when UART2CR1<BRG> = “110”), the transfer clock and transfer rate are determined as follows:

Transfer clock [Hz] = TC3 source clock [Hz] / TTREG3 setting value

Transfer Rate [baud] = Transfer clock [Hz] / 16

## 13.5 Data Sampling Method

The UART2 receiver keeps sampling input using the clock selected by UART2CR1<BRG> until a start bit is detected in RXD2 pin input. RT clock starts detecting “L” level of the RXD2 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).

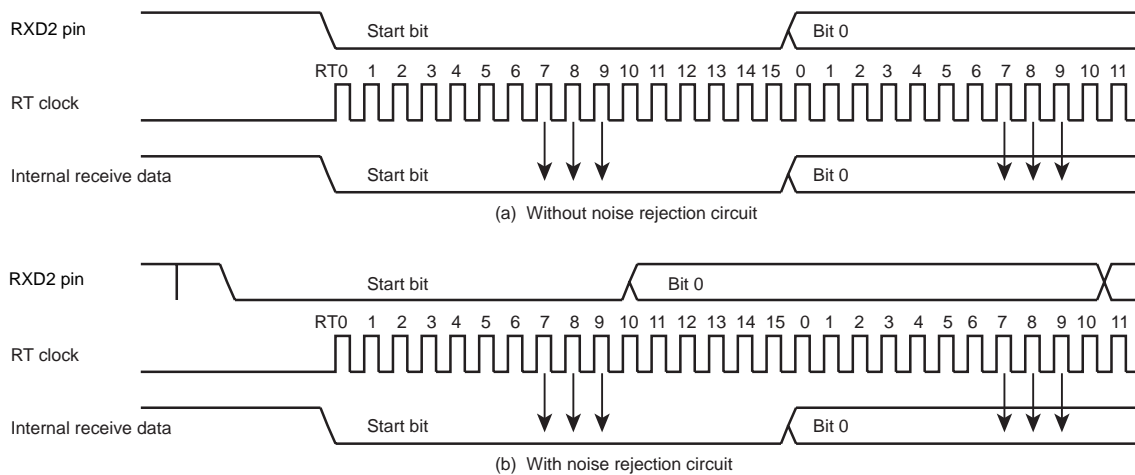


Figure 13-4 Data Sampling Method

## 13.6 STOP Bit Length

Select a transmit stop bit length (1 bit or 2 bits) by UART2CR1<STBT>.

## 13.7 Parity

Set parity / no parity by UART2CR1<PE> and set parity type (Odd- or Even-numbered) by UART2CR1<EVEN>.

## 13.8 Transmit/Receive Operation

### 13.8.1 Data Transmit Operation

Set UART2CR1<TXE> to “1”. Read UART2SR to check UART2SR<TBEP> = “1”, then write data in TD2BUF (Transmit data buffer). Writing data in TD2BUF zero-clears UART2SR<TBEP>, transfers the data to the transmit shift register and the data are sequentially output from the TXD2 pin. The data output include a one-bit start bit, stop bits whose number is specified in UART2CR1<STBT> and a parity bit if parity addition is specified. Select the data transfer baud rate using UART2CR1<BRG>. When data transmit starts, transmit buffer empty flag UART2SR<TBEP> is set to “1” and an INTTXD2 interrupt is generated.

While UART2CR1<TXE> = “0” and from when “1” is written to UART2CR1<TXE> to when send data are written to TD2BUF, the TXD2 pin is fixed at high level. When transmitting data, first read UART2SR, then write data in TD2BUF. Otherwise, UART2SR<TBEP> is not zero-cleared and transmit does not start.

### 13.8.2 Data Receive Operation

Set UART2CR1<RXE> to “1”. When data are received via the RXD2 pin, the receive data are transferred to RD2BUF (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 RD2BUF (Receive data buffer). Then the receive buffer full flag UART2SR<RBFL> is set and an INTRXD2 interrupt is generated. Select the data transfer baud rate using UART2CR1<BRG>.

If an overrun error (OERR) occurs when data are received, the data are not transferred to RD2BUF (Receive data buffer) but discarded; data in the RD2BUF are not affected.

Note: When a receive operation is disabled by setting UART2CR1<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.

## 13.9 Status Flag

### 13.9.1 Parity Error

When parity determined using the receive data bits differs from the received parity bit, the parity error flag UART2SR<PERR> is set to “1”. The UART2SR<PERR> is cleared to “0” when the RD2BUF is read after reading the UART2SR.

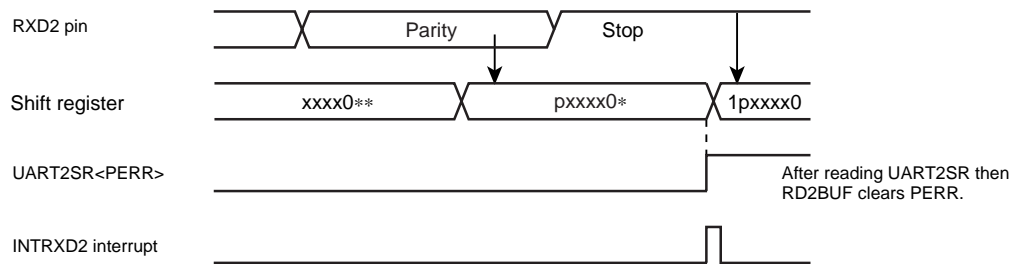


Figure 13-5 Generation of Parity Error

### 13.9.2 Framing Error

When “0” is sampled as the stop bit in the receive data, framing error flag UART2SR<FERR> is set to “1”. The UART2SR<FERR> is cleared to “0” when the RD2BUF is read after reading the UART2SR.

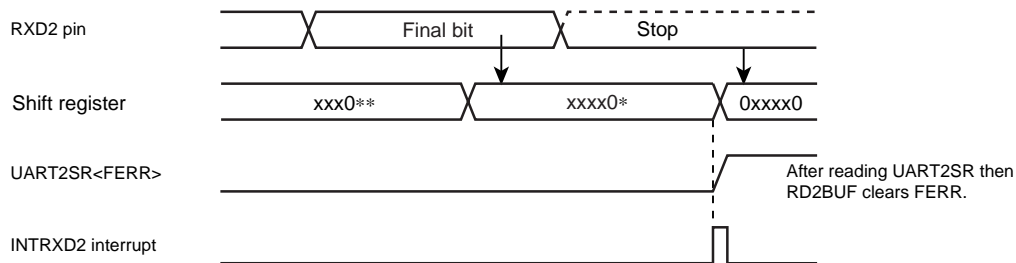


Figure 13-6 Generation of Framing Error

### 13.9.3 Overrun Error

When all bits in the next data are received while unread data are still in RD2BUF, overrun error flag UART2SR<OERR> is set to “1”. In this case, the receive data is discarded; data in RD2BUF are not affected. The UART2SR<OERR> is cleared to “0” when the RD2BUF is read after reading the UART2SR.

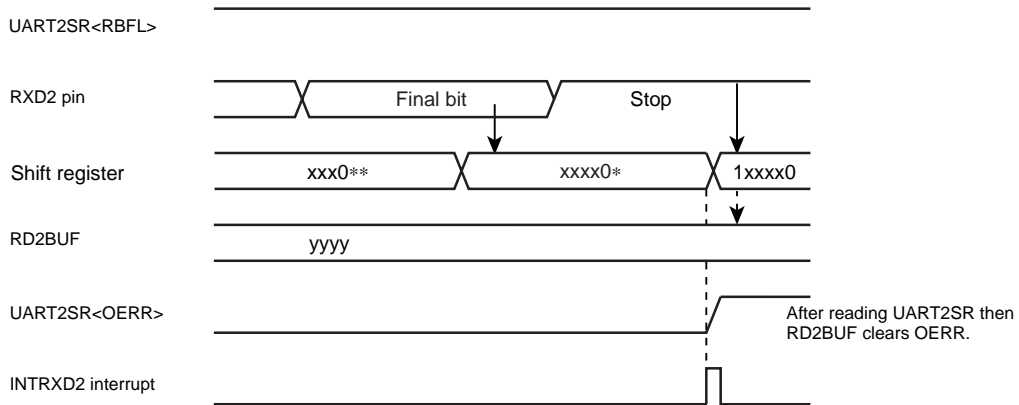


Figure 13-7 Generation of Overrun Error

Note: Receive operations are disabled until the overrun error flag UART2SR<OERR> is cleared.

### 13.9.4 Receive Data Buffer Full

Loading the received data in RD2BUF sets receive data buffer full flag UART2SR<RBFL> to "1". The UART2SR<RBFL> is cleared to "0" when the RD2BUF is read after reading the UART2SR.

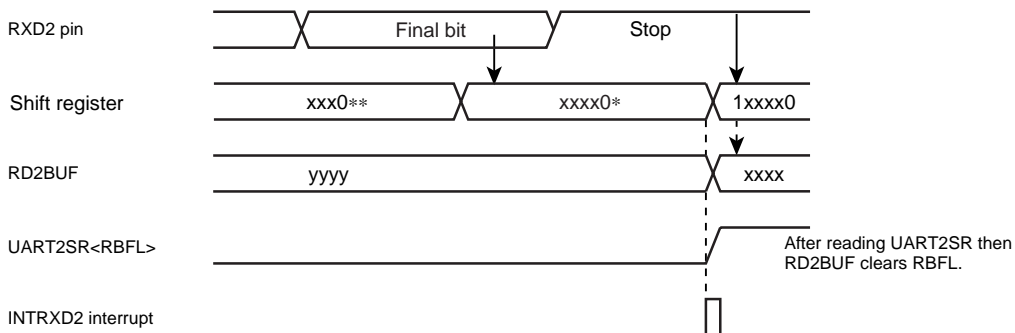


Figure 13-8 Generation of Receive Data Buffer Full

Note: If the overrun error flag UART2SR<OERR> is set during the period between reading the UART2SR and reading the RD2BUF, it cannot be cleared by only reading the RD2BUF. Therefore, after reading the RD2BUF, read the UART2SR again to check whether or not the overrun error flag which should have been cleared still remains set.

### 13.9.5 Transmit Data Buffer Empty

When no data is in the transmit buffer TD2BUF, that is, when data in TD2BUF are transferred to the transmit shift register and data transmit starts, transmit data buffer empty flag UART2SR<TBEP> is set to "1". The UART2SR<TBEP> is cleared to "0" when the TD2BUF is written after reading the UART2SR.

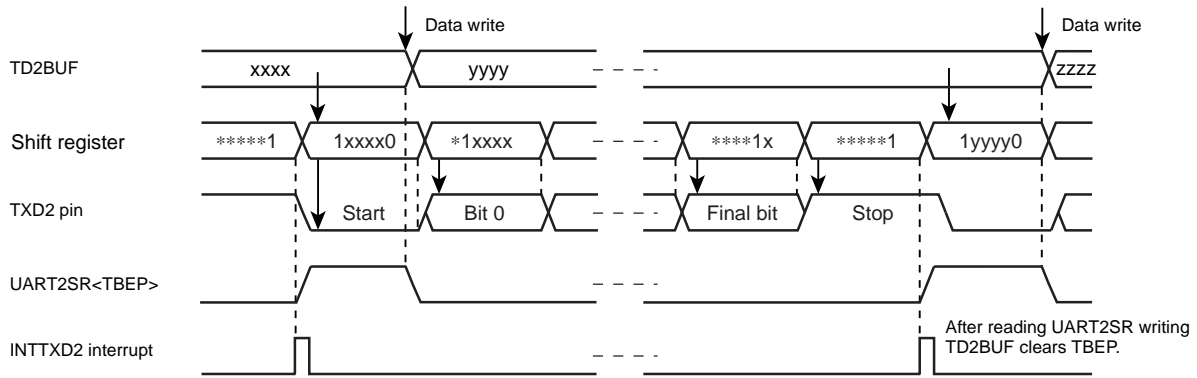


Figure 13-9 Generation of Transmit Data Buffer Empty

### 13.9.6 Transmit End Flag

When data are transmitted and no data is in TD2BUF (UART2SR<TBEP> = “1”), transmit end flag UART2SR<TEND> is set to “1”. The UART2SR<TEND> is cleared to “0” when the data transmit is started after writing the TD2BUF.

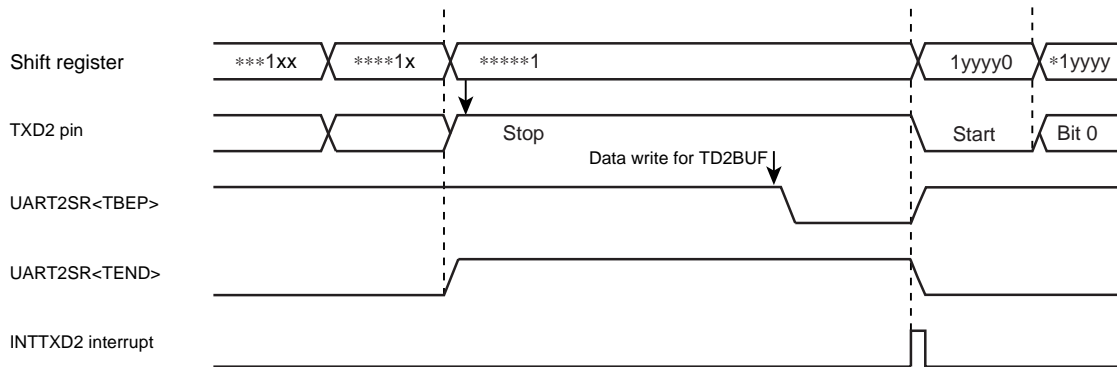


Figure 13-10 Generation of Transmit End Flag and Transmit Data Buffer Empty

## 14. Serial Expansion Interface (SEI)

SEI is one of the serial interfaces incorporated in the TMP86FH92DMG. It allows connection to peripheral devices via full-duplex synchronous communication protocols. The TMP86FH92DMG contain one channel of SEI.

SEI is connected with an external device through SCLK, MOSI, MISO and the terminal  $\overline{SS}$ . SCLK, MOSI, MISO, and  $\overline{SS}$  pins respectively are shared with P02, P03, P04 and P05. When using these ports as SCLK, MOSI, MISO, or  $\overline{SS}$  pins, set the each Port Output Latch to "1".

### 14.1 Features

- The master outputs the shift clock for only a data transfer period.
- The clock polarity and phase are programmable.
- The data is 8 bits long.
- MSB or LSB-first can be selected.
- The programmable data and clock timing of SEI can be connected to almost all synchronous serial peripheral devices. Refer to "14.5 SEI Transfer Formats".
- The transfer rate can be selected from the following four (master only):
  - 4 Mbps, 2 Mbps, 1 Mbps, or 250 kbps (when operating at 16 MHz)
- The error detection circuit supports the following functions:
  1. Write collision detection: When the shift register is accessed for write during transfer
  2. Overflow detection: When new data is received while the transfer-finished flag is set (slave only)
  3. Mode fault error input: When the SEI device is set as the master, driver output is disabled immediately upon input of a low level on the  $\overline{SS}$  pin (in open-drain output mode only). Overflow detection: When new data is received while the transfer-finished flag is set (slave only)

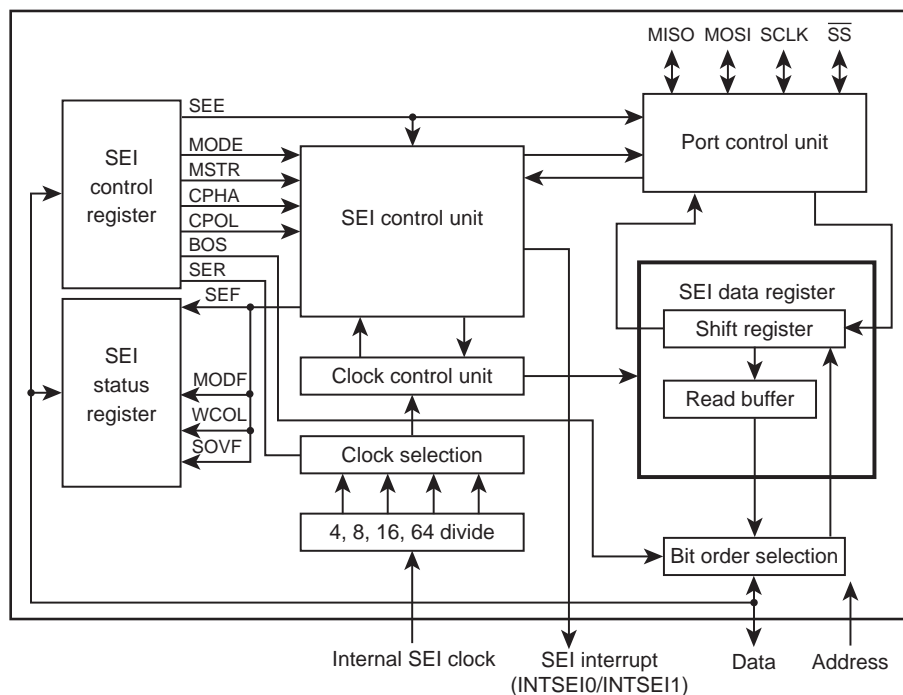
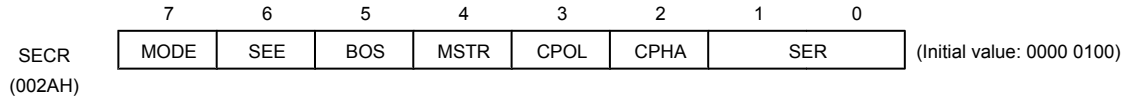


Figure 14-1 SEI (Serial Extended Interface)

## 14.2 SEI Registers

The SEI interface has the SEI Control Register (SECR), SEI Status Register (SESR), and SEI Data Register (SEDR) which are used to set up the SEI system and enable/disable SEI operation.

### 14.2.1 SEI Control Register (SECR)



Read-modify-write instruction are prohibited.

Bit	Field Name	Description	
MODE	Mode fault detection (Note1)	0: Enables mode fault detection 1: Disables mode fault detection It is available in Master mode only. (Note: Make sure to set <MODE> bit to "1" for disabling Mode fault de- tection	R/W
SEE	SEI operation (Note2)	0: Disables SEI operation 1: Enables SEI operation	
BOS	Bit order selection	0: Transmitted beginning with the MSB (bit 7) of SEDR register 1: Transmitted beginning with the LSB (bit 0) of SEDR register	
MSTR	Mode selection (Note3)	0: Sets SEI for slave 1: Sets SEI for master	
CPOL	Clock polarity	0: Selects active-"H" clock. SCLK remains "L" when IDLE. 1: Selects active-"L" clock. SCLK remains "H" when IDLE.	
CPHA	Clock phase	Selects clock phase. For details, refer to Section "SEI Transfer Formats".	
SER	Selects SEI transfer rate	00: Divide-by-4 01: Divide-by-8 10: Divide-by-16 11: Divide-by-64	

Note 1: If mode fault detection is enabled, an interrupt is generated when the MODF flag (SESR<MODF>) is set.

Note 2: SEI operation can only be disabled after transfer is completed. Before the SEI can be used, the each Port Control Register and Output Latch Control must be set for the SEI function (In case P0 port, P0OUTCR and P0DR).

When using the SEI as the master, set the SECR<SEE> bit to "1" (to enable SEI operation) and then place transmit data in the SEDR register. This initiates transmission/reception.

Note 3: Master/slave settings must be made before enabling SEI operation (This means that the SECR<MSTR> bit must first be set before setting the SECR<SEE> bit to "1").

#### 14.2.1.1 Transfer rate

##### (1) Master mode (Transfer rate = $f_c / \text{Internal clock divide ratio}$ (unit: bps))

The table below shows the relationship between settings of the SER bit and transfer bit rates when the SEI is operating as the master.

Table 14-1 SEI Transfer Rate

SER	Internal Clock Divide Ratio of SEI	Transfer Rate when $f_c = 16 \text{ MHz}$
00	4	4 Mbps
01	8	2 Mbps
10	16	1 Mbps
11	64	250 kbps



(2) Slave mode

When the SEI is operating as a slave, the serial clock is input from the master and the setting of the SER bit has no effect. The maximum transfer rate is  $f_c/4$ .

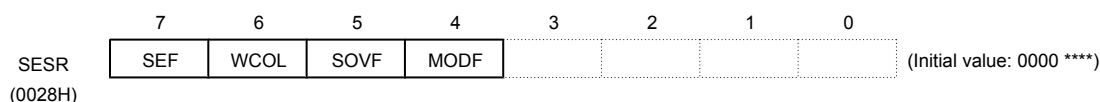
Note: Take note of the following relationship between the serial clock speed and  $f_c$  on the master side:

15.625 kbps < Transfer rate <  $f_c/4$  bps

Example) 15.625 kbps < Transfer rate < 4 Mbps ( $f_c = 16$  MHz at  $V_{DD} = 4.5$  to 5.5 V)

15.625 kbps < Transfer rate < 2 Mbps ( $f_c = 8$  MHz at  $V_{DD} = 2.7$  to 5.5 V)

14.2.2 SEI Status Register (SESR)



SEF	Transfer-finished flag (Note1)	0: Transfer in progress 1: Transfer completed	Read only
WCOL	Write collision error flag (Note2)	0: No write collision error occurred 1: Write collision error occurred	
SOVF	Overflow error flag (slave) (Note3)	0: No overflow occurred 1: Overflow occurred	
MODF	Mode fault flow error flag (master) (Note4)	0: No mode fault occurred 1: Mode fault occurred	

Note 1: The SEF flag is automatically set at completion of transfer. The SEF flag thus set is automatically cleared by reading the SESR register and accessing the SEDR register.

Note 2: The WCOL flag is automatically set by a write to the SEDR register while transfer is in progress. Writing to the SEDR register during transfer has no effect. The WCOL flag thus set is automatically cleared by reading the SESR register and accessing the SEDR register. No interrupts are generated for reasons that the WCOL flag is set.

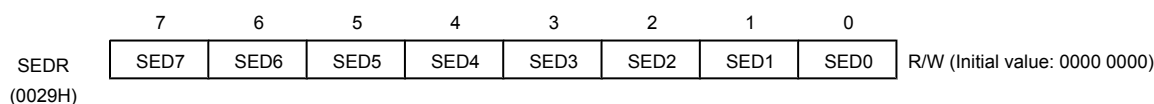
Note 3: During master mode:  
This bit does not function; its data when read is "0".  
During slave mode:

The SOVF flag is automatically set when the device finishes reading the next data while the SEF flag is set. The SOVF flag thus set is automatically cleared by reading the SESR register and accessing the SEDR register. The SOVF flag also is cleared by a switchover to master mode. No interrupts are generated for reasons that the SOVF flag is set.

Note 4: Master mode:  
The MODF flag is set when the  $\overline{SS}$  pin is driven low. At this time, the SEI performs the following operations:  
1. Disables the SEI pin driver and sets the SCLK and MOSI pins as inputs in the high-impedance state.  
2. Clears the SECR<MSTR> bit.  
3. Forcefully clears the SECR<SEE> bit to disable the SEI system.  
4. The MODF flag thus set is automatically cleared by a read of the SESR register and a write to the SECR register.  
Slave mode:  
This bit does not function; its data when read is 0.

14.2.3 SEI Data Register (SEDR)

The SEI Data Register (SEDR) is used to send and receive data. When the SEI is set for master, data transfer is initiated by writing to this SEDR register. If the master device needs to write to the SEDR register after transfer began, always check to see by means of an interrupt or by polling that the SEF flag (SESR<SEF>) is set, before writing to the SEDR register.



## 14.3 SEI Operation

During a SEI transfer, data transmission (serial shift-out) and reception (serial shift-in) are performed simultaneously. The serial clock synchronizes the timing at which information on the two serial data lines are shifted or sampled. Slave device can be selected individually using the slave select pin ( $\overline{SS}$  pin). For unselected slave devices, data on the SEI bus cannot be taken in.

When operating as the master devices, the  $\overline{SS}$  pin can be used to indicate multiple-master bus connection.

### 14.3.1 Controlling SEI clock polarity and phase

The SEI clock allows its phase and polarity to be selected in software from four combinations available by using two bits, CPHA and CPOL (SECR<CPHL,CPOL>).

The clock polarity is set by CPOL to select between active-high or active-low (The transfer format is unaffected).

The clock phase is set by CPHA. The master device and the slave devices to communicate with must have the same clock phase and polarity.

If multiple slave devices with different transfer formats exist on the same bus, the format can be changed to that of the slave device to which to transfer.

Table 14-2 Clock Phase and Polarity

CPHA	SEI control register (SECR 002AH) bit 2
CPOL	SEI control register (SECR 002AH) bit 3

### 14.3.2 SEI data and clock timing

The programmable data and clock timing of SEI allows connection to almost all synchronous serial peripheral devices. Refer to Section “14.5 SEI Transfer Formats”.

## 14.4 SEI Pin Functions

The TMP86FH92DMG have four input/output pins associated with SEI transfer. The functionality of each pin depends on the SEI device's mode (master or slave). The SCLK pin, MOSI pin and MISO pin of all SEI devices are connected with the same name pin to each other.

### 14.4.1 SCLK pin

The SCLK pin functions as an output pin when SEI is set for master, or as an input pin when SEI is set for slave.

When SEI is set for master, serial clock is output from the SCLK pin to external devices. After the master starts transfer, eight serial clock pulses are output from the SCLK pin only during transfer.

When SEI is set for slave, the SCLK pin functions as an input pin.

During data transfer between master and slave, device operation is synchronized by the serial clock output from the master.

When the  $\overline{SS}$  pin of the slave device is "H", data is not taken in regardless of whether the serial clock is available.

For both master and slave devices, data is shifted in and out at a rising or falling edge of the serial clock, and is sampled at the opposite edge where the data is stable. The active edge is determined by SEI transfer protocols.

Note: Noise in a slave device's SCLK input may cause the device to operate erratically.

### 14.4.2 MISO/MOSI pins

The MISO and MOSI pins are used for serial data transmission/reception. The status of each pin during master and slave are shown in the table below.

Table 14-3 MISO/MOSI Pin Status

	MISO	MOSI
Master	Input	Output
Slave	Output	Input

Also, the SCLK, MOSI, and MISO pins can be set for open-drain by the each pin's input/output control register (In case P0 Port, Input/output Control Register is P0OUTCR).

The MISO pin of a slave device becomes an output when the SECR<SEE> bit is set to 1 (SEI operation enabled). To set the MISO pin of an inactive slave device to a high-impedance state, clear the SECR<SEE> bit to 0.

### 14.4.3 $\overline{SS}$ pin

The  $\overline{SS}$  pin function differently when the SEI is the master and when it is a slave.

- When the SEI is a slave, this pin is used to enable the SEI transmission/reception. When the slave's  $\overline{SS}$  pin is high, the slave device ignores the serial clock from the master. Nor does it receive data from the MISO pin. When the slave's  $\overline{SS}$  pin is L, the SEI operates as slave.

- When the SEI is the master, the  $\overline{SS}$  pin is used for SEI error input. When the  $\overline{SS}$  pin of the master device goes low, output is immediately disabled on the SCLK and MOSI pins if these pins are configured as open-drain outputs. This causes the SESR<MODF> flag of the master device to be set. This state is called a mode fault. The mode fault function is provided to prevent damage caused by a collision between drivers that may occur, for example, when another device on the same bus becomes the master.

## 14.5 SEI Transfer Formats

The transfer formats are set using CPHA and CPOL (SECR<CPHA,CPOL>). CPHA allows transfer protocols to be selected between two.

### 14.5.1 CPHA (SECR register bit 2) = 0 format

Figure 14-2 shows a transfer format when CPHA = 0.

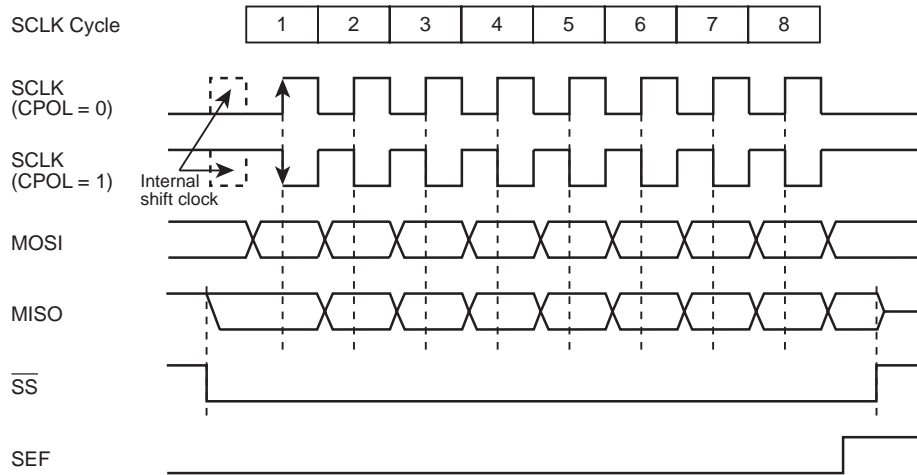


Figure 14-2 Transfer Format When CPHA = 0

Table 14-4 Transfer Format Details when CPHA = 0

	SCLK Level when not Communicating (IDLE)	Data Shift	Data Sampling
CPOL = 0	"L" level	Falling edge of transfer clock	Rising edge of transfer clock
CPOL = 1	"H" level	Rising edge of transfer clock	Falling edge of transfer clock

- In master mode, transfer is initiated by writing new data to the SEDR register. At this time, the new data changes state on the MOSI pin a half clock period before the shift clock starts pulsing. Use BOS (SECR<BOS>) to select whether the data should be shifted out beginning with the MSB or LSB. The SEF flag (SESR<SEF>) is set after the last shift cycle.
- In slave mode, writing data to the SEDR register is inhibited when the  $\overline{SS}$  pin is "L". A write during this period causes collision of writes, so that the WCOL flag (SESR<WCOL>) is set. Therefore, when writing data to the SEDR (SEI Data Register) after the SEF flag is set upon completion of transfer, make sure the  $\overline{SS}$  pin goes "H" again before writing the next data to the SEDR register.

Note: In slave mode, be careful not to write data while the SEF flag is set and the  $\overline{SS}$  pin remains "L".

14.5.2 CPHA = 1 format

Figure 14-3 shows a transfer format when CPHA = 1.

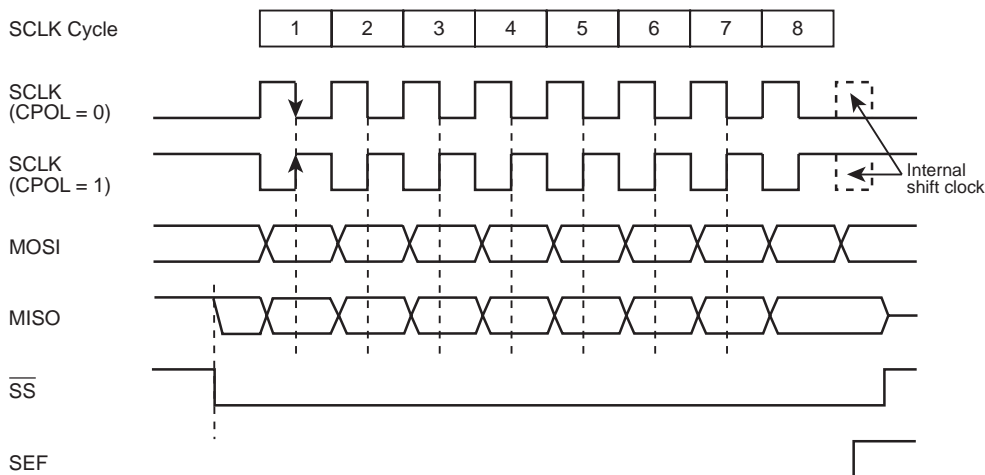


Figure 14-3 Transfer Format When CPHA = 1

Table 14-5 Transfer Format Details when CPHA = 1

	SCLK Level when Not Communicating (IDLE)	Data Shift	Data Sampling
CPOL=0	“L” level	Rising edge of transfer clock	Falling edge of transfer clock
CPOL=1	“H” level	Falling edge of transfer clock	Rising edge of transfer clock

- In master mode, transfer is initiated by writing new data to the SEDR register. The new data changes state on the MOSI pin at the first edge of the shift clock. Use BOS (SECR<BOS>) to select whether the data should be shifted out beginning with the MSB or LSB.
- In slave mode, unlike in the case of CPHA = 0 format, data can be written to the SEDR (SEI Data Register) regardless of whether the SS pin is “L” or “H”. In both master and slave modes, the SEF flag (SESR<SEF>) is set after the last shift cycle. Writing data to the SEDR register while data transfer is in progress causes collision of writes. Therefore, wait until the SEF flag is set before writing new data to the SEDR register.

## 14.6 Functional Description

Figure 14-4 shows how the SEI master and slave are connected.

When the master device sends data from its MOSI pin to a slave device's MOSI pin, the slave device returns data from its MISO pin to the master device's MISO pin. This means that data are exchanged between master and slave via full-duplex communication, with data output and input operations synchronized by the same clock signal. After end of transfer, the transmit byte in 8 bit shift register is replaced with the receive byte.

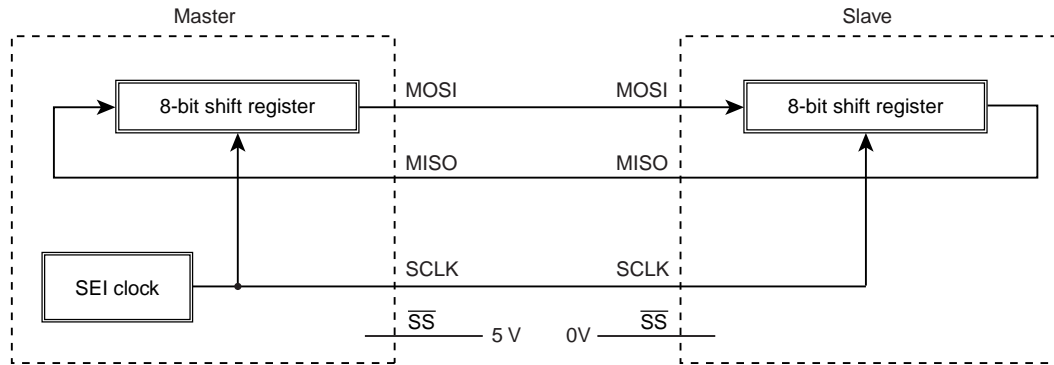


Figure 14-4 Master and Slave Connection in SEI

Figure 14-5 shows an example of how an SEI system can be configured. The general-purpose pins used for SEI transfer can be programmed to be open-drain outputs. This feature enables these pins to be connected with multiple devices. (We recommend using these pins in open-drain output mode.)

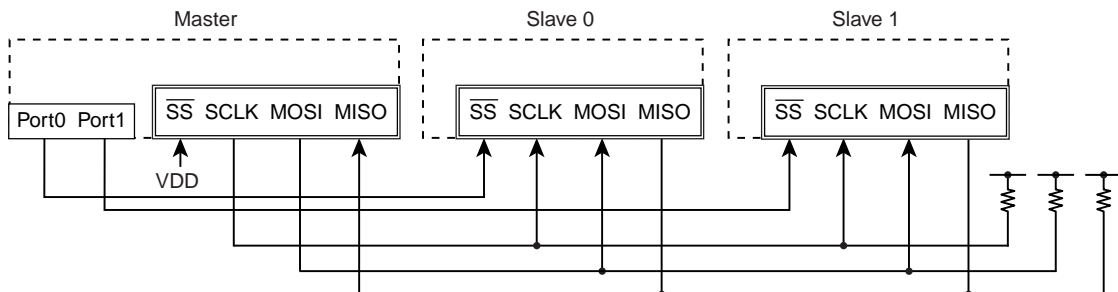


Figure 14-5 Example of SEI System Configuration (1 Master, 2 Slaves)

In this example, all the SCLK pins are interconnected, and all the MOSI and MISO pins are interconnected. One SEI device is set as the master and all the other SEI devices on the SEI bus are set as slaves. The master device sends data from its SCLK and MOSI pins to the SCLK and MOSI pins of a slave device.

The slave device selected by the master sends data from its MISO pin to the MISO pin of the master device.

## 14.7 Interrupt Generation

The TMP86FH92DMG is provided with the SEI interrupt channels 0 and 1 (INTSEI0 and INTSEI1) for processing SEI interrupts.

INTSEI0 generates an interrupt pulse when the SESR<MODF> flag is set. INTSEI1 generates an interrupt pulse when the SESR<SEF> flag is set.

Table 14-6 SEI Interrupt

SEI interrupt channel 0 (INTSEI0)	Generates an interrupt pulse when the MODF flag is set
SEI interrupt channel 1 (INTSEI1)	Generates an interrupt pulse when the SEF flag is set

## 14.8 SEI System Errors

The SEI can detect the following three types of system errors:

- Mode fault error: A mode fault error occurs if the  $\overline{SS}$  pin of the master device is driven low.
- Write collision error: A write collision error occurs if data is written to the SEDR register while a transfer is in progress.
- Overflow error: An overflow error occurs if new data is received in a slave device before the previous data is read. "

### 14.8.1 Write collision error

A write collision error occurs if an attempt is made to write to the SEDR register while data is being transferred. Because the SEDR register is not configured as dual-buffers for sending data, a write to the SEDR register directly results in writing to the SEI shift register. Therefore, writing to the SEDR register while a transfer is in progress causes a write collision error.

In no case is data transfer stopped in the middle, so that the write data which caused a write collision error will not be written to the shift register. Because slaves cannot control the timing at which the master starts a transfer, write collision errors normally occur on slave devices. The master has the right to perform a transfer at any time, and thus write collision errors do not normally occur on the master side. However, both master and slave SEI devices are capable of detecting write collision errors.

A write collision error tends to occur on a slave device when the master shifts out data at a speed faster than that at which the slave processes the transferred data. More specifically, a write collision error occurs if the slave transfers a new value to the SEDR register when the master has already started a shift cycle for the next byte.

### 14.8.2 Overflow error

The transfer bit rate on the SEI bus is determined by the master. At higher bit rates, a slave device may not be able to keep up with transfer from the master. This occurs when the master shifts out data faster than can be processed by the slave. The SEI module uses the SESR<SOVF> flag to detect an overflow error.

The SOVF flag is set if the following conditions are both met:

- When the SEI module is set as a slave
- When the previous data byte remains to be read after a new data byte has been received

When the SOVF flag is set, the SEDR register is overwritten with a new data byte.

Note: Please carefully examine the communication processing routine and communication rate when designing your application system.

### 14.8.3 Mode fault error

When the SEI device is set as the master, a mode fault error occurs if the  $\overline{SS}$  pin is driven low. When a mode fault error occurs, the SEI immediately performs the following operations: The SOVF flag is set if the following conditions are both met:

- Clears the SECR<MSTR> bit to 0 to set the SEI device as a slave.
- Clears the SECR<SEE> bit to 0 to disable SEI operation.
- Sets the SESR<MODF> flag to 1 to generate an INTSEI0 interrupt pulse.
- Sets the SCLK and MOSI pins to output 1. (These pins become high-impedance in open-drain output mode, or high level in CMOS output mode.)

The SESR<MODF> flag thus set is automatically cleared by a read of the SESR register and a write to the SECR register. Once the SESR<MODF> flag is cleared, the mode setting can be made again.

When open-drain output mode is selected, this mode fault error function can be used to prevent the collision between the SCLK pin and MOSI pin drivers if two or more devices on the same bus are set as the master at the same time. (It is not possible to prevent the collision of the MISO pins if the  $\overline{SS}$  pins of two or more slave devices on the same bus are simultaneously driven low.)

## 14.9 Bus Driver Protection

- One method to protect the device against latch-up due to collision of the bus drivers is the use of an open-drain option. This means changing the SEI pins' CMOS outputs to the open-drain type, which is accomplished by setting the SCLK, MOSI, and MISO pins for open-drain individually by using the each Port Input/output Control Register. In this case, these pins must be provided with pull-up resistors external to the chip.
- When using the SEI pins as CMOS outputs, we recommend connecting them to the bus via resistors in order to protect the device against collision of drivers. However, be sure to select the appropriate resistance value which will not affect actual device operation (Example: 1  $\Omega$  to several k $\Omega$ ).



## 15. Serial Bus Interface(I<sup>2</sup>C Bus) Ver.-D (SBI)

The TMP86FH92DMG has a serial bus interface which employs an I<sup>2</sup>C bus.

The serial interface is connected to an external devices through SDA and SCL.

The serial bus interface pins are also used as the port. When used as serial bus interface pins, set the output latches of these pins to "1". When not used as serial bus interface pins, the port is used as a normal I/O port.

Note 1: The serial bus interface can be used only in NORMAL 1/2 and IDLE1/2 mode. It can not be used in IDLE0, SLOW1/2 and SLEEP0/1/2 mode.

Note 2: The serial bus interface can be used only in the Standard mode of I<sup>2</sup>C. The fast mode and the high-speed mode can not be used.

Note 3: Please refer to the I/O port section about the detail of setting port.

### 15.1 Configuration

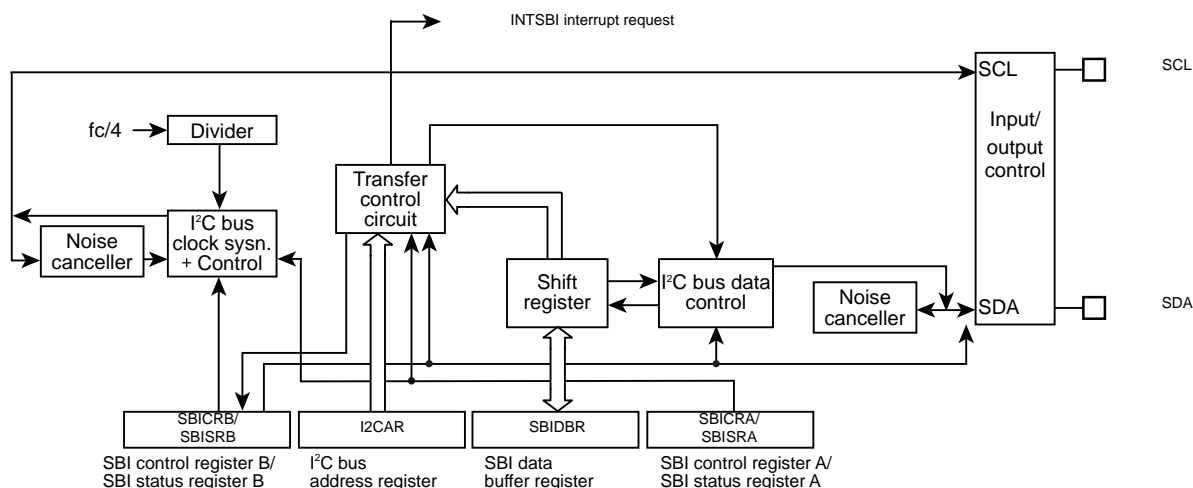


Figure 15-1 Serial Bus Interface (SBI)

### 15.2 Control

The following registers are used for control the serial bus interface and monitor the operation status.

- Serial bus interface control register A (SBICRA)
- Serial bus interface control register B (SBICRB)
- Serial bus interface data buffer register (SBIDBR)
- I<sup>2</sup>C bus address register (I2CAR)
- Serial bus interface status register A (SBISRA)
- Serial bus interface status register B (SBISRB)

### 15.3 Software Reset

A serial bus interface circuit has a software reset function, when a serial bus interface circuit is locked by an external noise, etc.

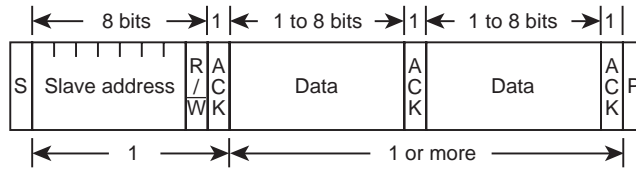
To reset the serial bus interface circuit, write "10", "01" into the SWRST (Bit1, 0 in SBICRB).

And a status of software reset can be read from SWRMON (Bit0 in SBISRA).

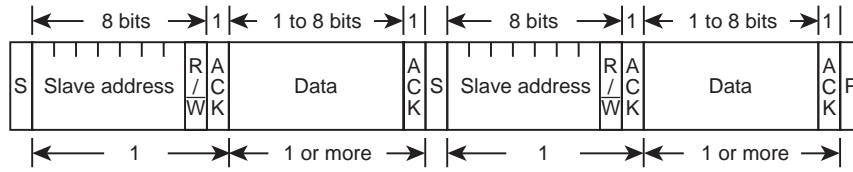
## 15.4 The Data Format in the I<sup>2</sup>C Bus Mode

The data format of the I<sup>2</sup>C bus is shown below.

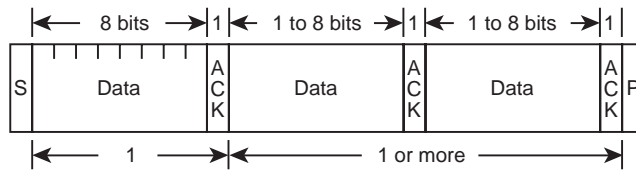
(a) Addressing format



(b) Addressing format (with restart)



(c) Free data format



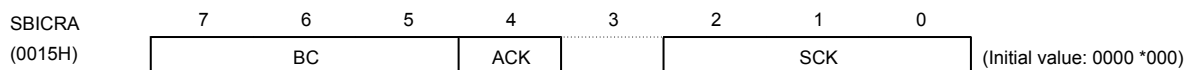
S : Start condition  
 R/W : Direction bit  
 ACK : Acknowledge bit  
 P : Stop condition

Figure 15-2 Data Format in of I<sup>2</sup>C Bus

## 15.5 I<sup>2</sup>C Bus Control

The following registers are used to control the serial bus interface and monitor the operation status of the I<sup>2</sup>C bus.

### Serial Bus Interface Control Register A



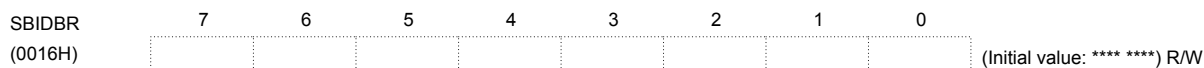
		BC	ACK = 0		ACK = 1		
			Number of Clock	Bits	Number of Clock	Bits	
BC	Number of transferred bits	000:	8	8	9	8	Write only
		001:	1	1	2	1	
		010:	2	2	3	2	
		011:	3	3	4	3	
		100:	4	4	5	4	
		101:	5	5	6	5	
		110:	6	6	7	6	
		111:	7	7	8	7	
ACK	Acknowledgement mode specification	ACK	Master mode		Slave mode		R/W
		0:	Not generate a clock pulse for an acknowledgement.		Not count a clock pulse for an acknowledgement.		
		1:	Generate a clock pulse for an acknowledgement.		Count a clock pulse for an acknowledgement.		
SCK	Serial clock (fsc) selection (Output on SCL pin)  [fsc = 1/(2 <sup>n+1</sup> /fc + 8/fc)]	SCK	n	At fc = 16 MHz	At fc = 8 MHz	At fc = 4 MHz	Write only
		000:	4	Reserved	Reserved	100.0 kHz	
		001:	5	Reserved	Reserved	55.6 kHz	
		010:	6	Reserved	58.8 kHz	29.4 kHz	
		011:	7	60.6 kHz	30.3 kHz	15.2 kHz	
		100:	8	30.8 kHz	15.4 kHz	7.7 kHz	
		101:	9	15.5 kHz	7.8 kHz	3.9 kHz	
		110:	10	7.8 kHz	3.9 kHz	1.9 kHz	
111:	Reserved						

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

Note 2: SBICRA cannot be used with any of read-modify-write instructions such as bit manipulation, etc.

Note 3: Do not set SCK as the frequency that is over 100 kHz.

### Serial Bus Interface Data Buffer Register



Note 1: For writing transmitted data, start from the MSB (Bit7).

Note 2: The data which was written into SBIDBR can not be read, since a write data buffer and a read buffer are independent in SBIDBR. Therefore, SBIDBR cannot be used with any of read-modify-write instructions such as bit manipulation, etc.

Note 3: \*: Don't care

I<sup>2</sup>C bus Address Register

I2CAR (0017H)	7	6	5	4	3	2	1	0	
	Slave address							ALS	(Initial value: 0000 0000)
	SA6	SA5	SA4	SA3	SA2	SA1	SA0		

SA	Slave address selection		
ALS	Address recognition mode specification	0: Slave address recognition 1: Non slave address recognition	Write only

Note 1: I2CAR is write-only register, which cannot be used with any of read-modify-write instruction such as bit manipulation, etc.

Note 2: Do not set I2CAR to "00H" to avoid the incorrect response of acknowledgment in slave mode. ( If "00H" is set to I2CAR as the Slave Address and a START Byte "01H" in I<sup>2</sup>C bus standard is received, the device detects slave address match.)

## Serial Bus Interface Control Register B

SBICRB (0018H)	7	6	5	4	3	2	1	0	
	MST	TRX	BB	PIN	SBIM		SWRST1	SWRST0	(Initial value: 0001 0000)

MST	Master/slave selection	0: Slave 1: Master	Write only
TRX	Transmitter/receiver selection	0: Receiver 1: Transmitter	
BB	Start/stop generation	0: Generate a stop condition when MST, TRX and PIN are "1" 1: Generate a start condition when MST, TRX and PIN are "1"	
PIN	Cancel interrupt service request	0: - (Can not clear this bit by a software) 1: Cancel interrupt service request	
SBIM	Serial bus interface operating mode selection	00: Port mode (Serial bus interface output disable) 01: Reserved 10: I <sup>2</sup> C bus mode 11: Reserved	
SWRST1 SWRST0	Software reset start bit	Software reset starts by first writing "10" and next writing "01"	

Note 1: Switch a mode to port after confirming that the bus is free.

Note 2: Switch a mode to I<sup>2</sup>C bus mode after confirming that the port is high level.

Note 3: SBICRB has write-only register and must not be used with any of read-modify-write instructions such as bit manipulation, etc.

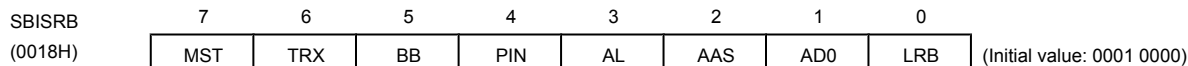
Note 4: When the SWRST (Bit1, 0 in SBICRB) is written to "10", "01" in I<sup>2</sup>C bus mode, software reset is occurred. In this case, the SBICRA, I2CAR, SBISRA and SBISRB registers are initialized and the bits of SBICRB except the SBIM (Bit3, 2 in SBICRB) are also initialized.

## Serial Bus Interface Status Register A

SBISRA (0015H)	7	6	5	4	3	2	1	0	
								SWRMON	(Initial value: **** *1)

SWRMON	Software reset monitor	0: During software reset 1: - (Initial value)	Read only
--------	------------------------	--	-----------

Serial Bus Interface Status Register B



MST	Master/slave selection status monitor	0: Slave 1: Master	Read only
TRX	Transmitter/receiver selection status monitor	0: Receiver 1: Transmitter	
BB	Bus status monitor	0: Bus free 1: Bus busy	
PIN	Interrupt service requests status monitor	0: Requesting interrupt service 1: Releasing interrupt service request	
AL	Arbitration lost detection monitor	0: - 1: Arbitration lost detected	
AAS	Slave address match detection monitor	0: - 1: Detect slave address match or "GENERAL CALL"	
AD0	"GENERAL CALL" detection monitor	0: - 1: Detect "GENERAL CALL"	
LRB	Last received bit monitor	0: Last receive bit is "0" 1: Last receive bit is "1"	

### 15.5.1 Acknowledgement mode specification

#### 15.5.1.1 Acknowledgment mode (ACK = "1")

To set the device as an acknowledgment mode, the ACK (Bit4 in SBICRA) should be set to "1". When a serial bus interface circuit is a master mode, an additional clock pulse is generated for an acknowledge signal. In a slave mode, a clock is counted for the acknowledge signal.

In the master transmitter mode, the SDA pin is released in order to receive an acknowledge signal from the receiver during additional clock pulse cycle. In the master receiver mode, the SDA pin is set to low level generation an acknowledge signal during additional clock pulse cycle.

In a slave mode, when a received slave address matches to a slave address which is set to the I2CAR or when a "GENERAL CALL" is received, the SDA pin is set to low level generating an acknowledge signal. After the matching of slave address or the detection of "GENERAL CALL", in the transmitter, the SDA pin is released in order to receive an acknowledge signal from the receiver during additional clock pulse cycle. In a receiver, the SDA pin is set to low level generation an acknowledge signal during additional clock pulse cycle after the matching of slave address or the detection of "GENERAL CALL"

The Table 15-1 shows the SCL and SDA pins status in acknowledgment mode.

Table 15-1 SCL and SDA Pins Status in Acknowledgement Mode

Mode	Pin		Transmitter	Receiver
Master	SCL		An additional clock pulse is generated.	
	SDA		Released in order to receive an acknowledge signal.	Set to low level generating an acknowledge signal
Slave	SCL		A clock is counted for the acknowledge signal.	
	SDA	When slave address matches or a general call is detected	-	Set to low level generating an acknowledge signal.
		After matching of slave address or general call	Released in order to receive an acknowledge signal.	Set to low level generating an acknowledge signal.

### 15.5.1.2 Non-acknowledgment mode (ACK = "0")

To set the device as a non-acknowledgement mode, the ACK (Bit4 in SBICRA) should be cleared to "0".

In the master mode, a clock pulse for an acknowledge signal is not generated.

In the slave mode, a clock for a acknowledge signal is not counted.

### 15.5.2 Number of transfer bits

The BC (Bits7 to 5 in SBICRA) is used to select a number of bits for next transmitting and receiving data.

Since the BC is cleared to "000" by a start condition, a slave address and direction bit transmissions are always executed in 8 bits. Other than these, the BC retains a specified value.

### 15.5.3 Serial clock

#### 15.5.3.1 Clock source

The SCK (Bits2 to 0 in SBICRA) is used to select a maximum transfer frequency output from the SCL pin in the master mode.

Four or more machine cycles are required for both high and low levels of pulse width in the external clock which is input from SCL pin.

Note: Since the serial bus interface can not be used as the fast mode and the high-speed mode, do not set SCK as the frequency that is over 100 kHz.

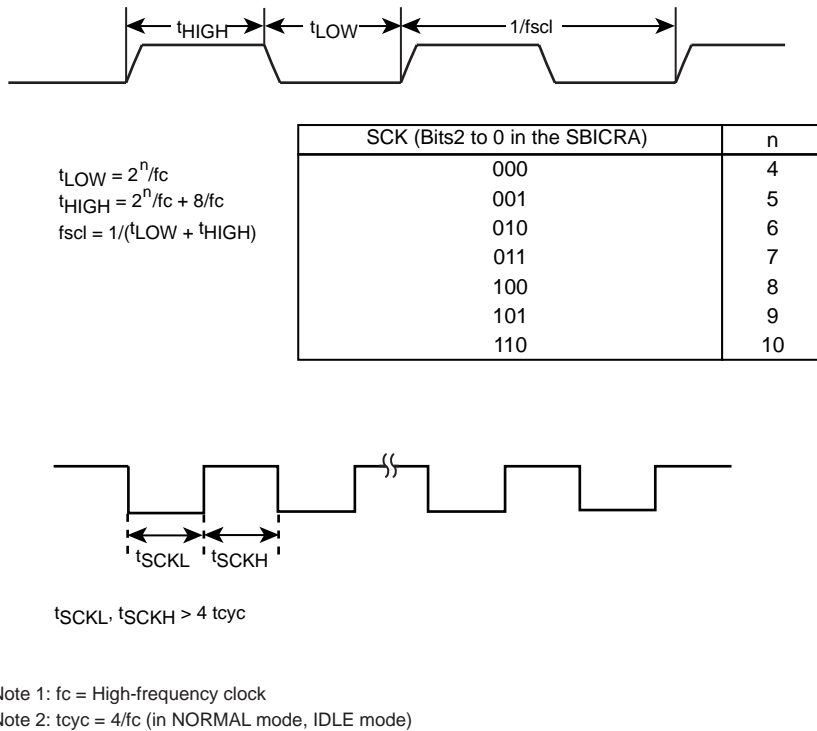


Figure 15-3 Clock Source

15.5.3.2 Clock synchronization

In the I<sup>2</sup>C bus, in order to drive a bus with a wired AND, a master device which pulls down a clock pulse to low will, in the first place, invalidate a clock pulse of another master device which generates a high-level clock pulse.

The serial bus interface circuit has a clock synchronization function. This function ensures normal transfer even if there are two or more masters on the same bus.

The example explains clock synchronization procedures when two masters simultaneously exist on a bus.

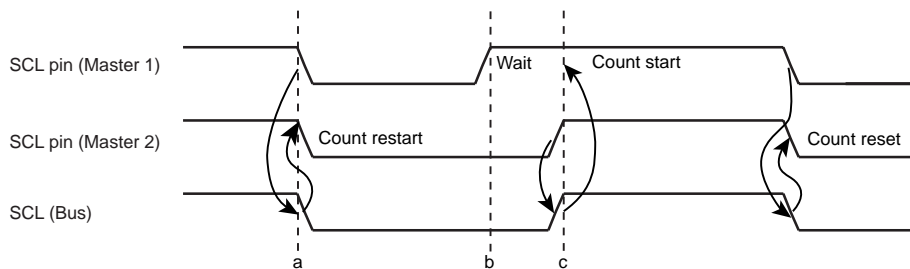


Figure 15-4 Clock Synchronization

As Master 1 pulls down the SCL pin to the low level at point “a”, the SCL line of the bus becomes the low level. After detecting this situation, Master 2 resets counting a clock pulse in the high level and sets the SCL pin to the low level.

Master 1 finishes counting a clock pulse in the low level at point “b” and sets the SCL pin to the high level. Since Master 2 holds the SCL line of the bus at the low level, Master 1 waits for counting a clock pulse in the high level. After Master 2 sets a clock pulse to the high level at point “c” and detects the SCL line of the bus at the high level, Master 1 starts counting a clock pulse in the high level. Then, the master, which has finished the counting a clock pulse in the high level, pulls down the SCL pin to the low level.

The clock pulse on the bus is determined by the master device with the shortest high-level period and the master device with the longest low-level period from among those master devices connected to the bus.

#### 15.5.4 Slave address and address recognition mode specification

When the serial bus interface circuit is used with an addressing format to recognize the slave address, clear the ALS (Bit0 in I2CAR) to "0", and set the SA (Bits7 to 1 in I2CAR) to the slave address.

When the serial bus interface circuit is used with a free data format not to recognize the slave address, set the ALS to "1". With a free data format, the slave address and the direction bit are not recognized, and they are processed as data from immediately after start condition.

#### 15.5.5 Master/slave selection

To set a master device, the MST (Bit7 in SBICRB) should be set to "1". To set a slave device, the MST should be cleared to "0".

When a stop condition on the bus or an arbitration lost is detected, the MST is cleared to "0" by the hardware.

#### 15.5.6 Transmitter/receiver selection

To set the device as a transmitter, the TRX (Bit6 in SBICRB) should be set to "1". To set the device as a receiver, the TRX should be cleared to "0". When data with an addressing format is transferred in the slave mode, the TRX is set to "1" by a hardware if the direction bit ( $R/\bar{W}$ ) sent from the master device is "1", and is cleared to "0" by a hardware if the bit is "0". In the master mode, after an acknowledge signal is returned from the slave device, the TRX is cleared to "0" by a hardware if a transmitted direction bit is "1", and is set to "1" by a hardware if it is "0". When an acknowledge signal is not returned, the current condition is maintained.

When a stop condition on the bus or an arbitration lost is detected, the TRX is cleared to "0" by the hardware. "Table 15-2 TRX changing conditions in each mode" shows TRX changing conditions in each mode and TRX value after changing

Table 15-2 TRX changing conditions in each mode

Mode	Direction Bit	Conditions	TRX after Changing
Slave Mode	"0"	A received slave address is the same value set to I2CAR	"0"
	"1"		"1"
Master Mode	"0"	ACK signal is returned	"1"
	"1"		"0"

When a serial bus interface circuit operates in the free data format, a slave address and a direction bit are not recognized. They are handled as data just after generating a start condition. The TRX is not changed by a hardware.

#### 15.5.7 Start/stop condition generation

When the BB (Bit5 in SBISRB) is "0", a slave address and a direction bit which are set to the SBIDBR are output on a bus after generating a start condition by writing "1" to the MST, TRX, BB and PIN. It is necessary to set ACK to "1" beforehand.



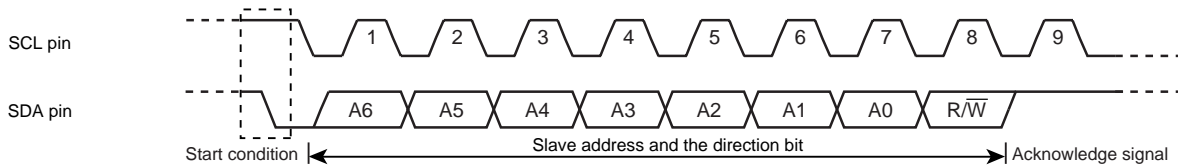


Figure 15-5 Start Condition Generation and Slave Address Generation

When the BB is “1”, sequence of generating a stop condition is started by writing “1” to the MST, TRX and PIN, and “0” to the BB. Do not modify the contents of MST, TRX, BB and PIN until a stop condition is generated on a bus.

When a stop condition is generated and the SCL line on a bus is pulled-down to low level by another device, a stop condition is generated after releasing the SCL line.

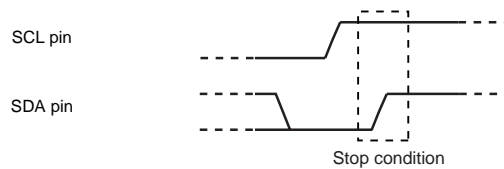


Figure 15-6 Stop Condition Generation

The bus condition can be indicated by reading the contents of the BB (Bit5 in SBISRB). The BB is set to “1” when a start condition on a bus is detected (Bus Busy State) and is cleared to “0” when a stop condition is detected (Bus Free State).

### 15.5.8 Interrupt service request and cancel

When a serial bus interface circuit is in the master mode and transferring a number of clocks set by the BC and the ACK is complete, a serial bus interface interrupt request (INTSBI) is generated.

In the slave mode, the conditions of generating INTSBI interrupt request are follows:

- At the end of acknowledge signal when the received slave address matches to the value set by the I2CAR
- At the end of acknowledge signal when a “GENERAL CALL” is received
- At the end of transferring or receiving after matching of slave address or receiving of “GENERAL CALL”

When a serial bus interface interrupt request occurs, the PIN (Bit4 in SBISRB) is cleared to “0”. During the time that the PIN is “0”, the SCL pin is pulled-down to low level.

Either writing data to SBIDBR or reading data from the SBIDBR sets the PIN to “1”.

The time from the PIN being set to “1” until the SCL pin is released takes  $t_{LOW}$ .

Although the PIN (Bit4 in SBICRB) can be set to “1” by the software, the PIN can not be cleared to “0” by the software.

Note: When the arbitration lost occurs, if the slave address sent from the other master devices is not match, the INTSBI interrupt request is generated. But the PIN is not cleared.

### 15.5.9 Setting of I<sup>2</sup>C bus mode

The SBIM (Bit3 and 2 in SBICRB) is used to set I<sup>2</sup>C bus mode.

Set the SBIM to “10” in order to set I<sup>2</sup>C bus mode. Before setting of I<sup>2</sup>C bus mode, confirm serial bus interface pins in a high level, and then, write “10” to SBIM. And switch a port mode after confirming that a bus is free.

### 15.5.10 Arbitration lost detection monitor

Since more than one master device can exist simultaneously on a bus, a bus arbitration procedure is implemented in order to guarantee the contents of transferred data.

Data on the SDA line is used for bus arbitration of the I<sup>2</sup>C bus.

The following shows an example of a bus arbitration procedure when two master devices exist simultaneously on a bus. Master 1 and Master 2 output the same data until point “a”. After that, when Master 1 outputs “1” and Master 2 outputs “0”, since the SDA line of a bus is wired AND, the SDA line is pulled-down to the low level by Master 2. When the SCL line of a bus is pulled-up at point “b”, the slave device reads data on the SDA line, that is data in Master 2. Data transmitted from Master 1 becomes invalid. The state in Master 1 is called “arbitration lost”. A master device which loses arbitration releases the SDA pin and the SCL pin in order not to effect data transmitted from other masters with arbitration. When more than one master sends the same data at the first word, arbitration occurs continuously after the second word.

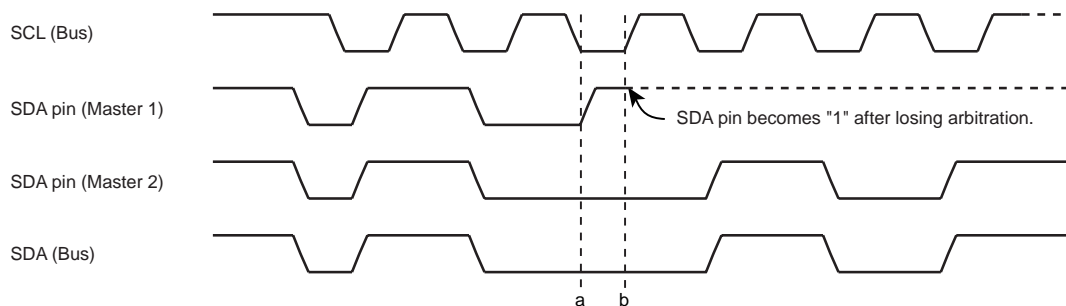


Figure 15-7 Arbitration Lost

The serial bus interface circuit compares levels of a SDA line of a bus with its SDA pin at the rising edge of the SCL line. If the levels are unmatched, arbitration is lost and the AL (Bit3 in SBISRB) is set to “1”.

When the AL is set to “1”, the MST and TRX are cleared to “0” and the mode is switched to a slave receiver mode. Thus, the serial bus interface circuit stops output of clock pulses during data transfer after the AL is set to “1”.

The AL is cleared to “0” by writing data to the SBIDBR, reading data from the SBIDBR or writing data to the SBICRB.

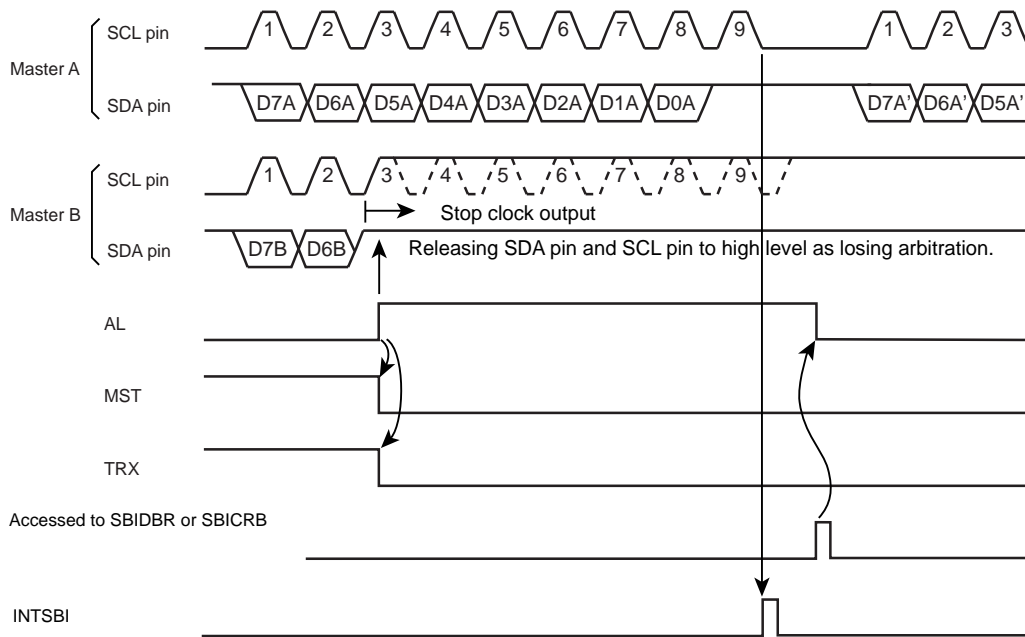


Figure 15-8 Example of when a Serial Bus Interface Circuit is a Master B

### 15.5.11 Slave address match detection monitor

In the slave mode, the AAS (Bit2 in SBISRB) is set to “1” when the received data is “GENERAL CALL” or the received data matches the slave address setting by I2CAR with an address recognition mode (ALS = 0).

When a serial bus interface circuit operates in the free data format (ALS = 1), the AAS is set to “1” after receiving the first 1-word of data.

The AAS is cleared to “0” by writing data to the SBIDBR or reading data from the SBIDBR.

### 15.5.12 GENERAL CALL detection monitor

The AD0 (Bit1 in SBISRB) is set to “1” when all 8-bit received data is “0” immediately after a start condition in a slave mode. The AD0 is cleared to “0” when a start or stop condition is detected on a bus.

### 15.5.13 Last received bit monitor

The SDA line value stored at the rising edge of the SCL line is set to the LRB (Bit0 in SBISRB). In the acknowledge mode, immediately after an INTSBI interrupt request is generated, an acknowledge signal is read by reading the contents of the LRB.

## 15.6 Data Transfer of I<sup>2</sup>C Bus

### 15.6.1 Device initialization

For initialization of device, set the ACK in SBICRA to “1” and the BC to “000”. Specify the data length to 8 bits to count clocks for an acknowledge signal. Set a transfer frequency to the SCK in SBICRA.

Next, set the slave address to the SA in I2CAR and clear the ALS to “0” to set an addressing format.

After confirming that the serial bus interface pin is high level, for specifying the default setting to a slave receiver mode, clear “0” to the MST, TRX and BB in SBICRB, set “1” to the PIN, “10” to the SBIM, and “00” to bits SWRST1 and SWRST0.

Note: The initialization of a serial bus interface circuit must be complete within the time from all devices which are connected to a bus have initialized to and device does not generate a start condition. If not, the data can not be received correctly because the other device starts transferring before an end of the initialization of a serial bus interface circuit.

### 15.6.2 Start condition and slave address generation

Confirm a bus free status (BB = 0).

Set the ACK to “1” and specify a slave address and a direction bit to be transmitted to the SBIDBR.

By writing “1” to the MST, TRX, BB and PIN, the start condition is generated on a bus and then, the slave address and the direction bit which are set to the SBIDBR are output. The time from generating the START condition until the falling SCL pin takes  $t_{LOW}$ .

An INTSBI interrupt request occurs at the 9th falling edge of a SCL clock cycle, and the PIN is cleared to “0”. The SCL pin is pulled-down to the low level while the PIN is “0”. When an interrupt request occurs, the TRX changes by the hardware according to the direction bit only when an acknowledge signal is returned from the slave device.

Note 1: Do not write a slave address to be output to the SBIDBR while data is transferred. If data is written to the SBIDBR, data to be outputting may be destroyed.

Note 2: The bus free must be confirmed by software within 98.0  $\mu$ s (The shortest transmitting time according to the I<sup>2</sup>C bus standard) after setting of the slave address to be output. Only when the bus free is confirmed, set “1” to the MST, TRX, BB, and PIN to generate the start conditions. If the writing of slave address and setting of MST, TRX, BB and PIN doesn't finish within 98.0  $\mu$ s, the other masters may start the transferring and the slave address data written in SBIDBR may be broken.

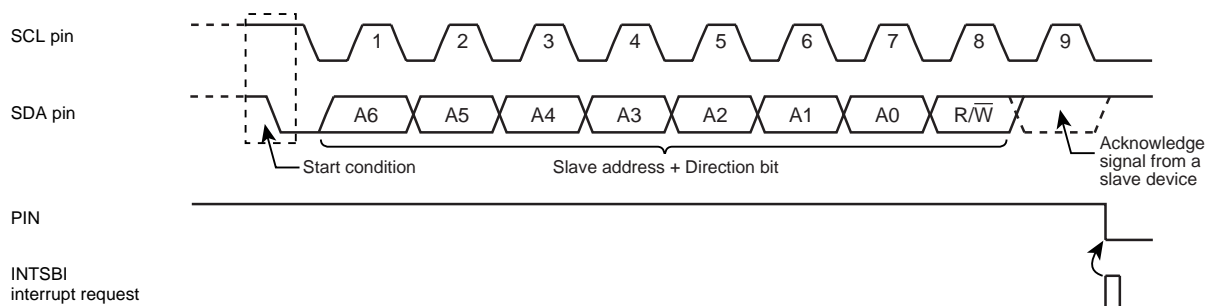


Figure 15-9 Start Condition Generation and Slave Address Transfer

### 15.6.3 1-word data transfer

Check the MST by the INTSBI interrupt process after an 1-word data transfer is completed, and determine whether the mode is a master or slave.

#### 15.6.3.1 When the MST is “1” (Master mode)

Check the TRX and determine whether the mode is a transmitter or receiver.

##### (1) When the TRX is “1” (Transmitter mode)

Test the LRB. When the LRB is “1”, a receiver does not request data. Implement the process to generate a stop condition (Described later) and terminate data transfer.

When the LRB is “0”, the receiver requests next data. When the next transmitted data is other than 8 bits, set the BC, set the ACK to “1”, and write the transmitted data to the SBIDBR. After writing the data, the PIN becomes “1”, a serial clock pulse is generated for transferring a next 1 word of data from the SCL pin, and then the 1 word of data is transmitted. After the data is transmitted, and an INTSBI interrupt request occurs. The PIN become “0” and the SCL pin is set to low level. If the data to be transferred is more than one word in length, repeat the procedure from the LRB test above.

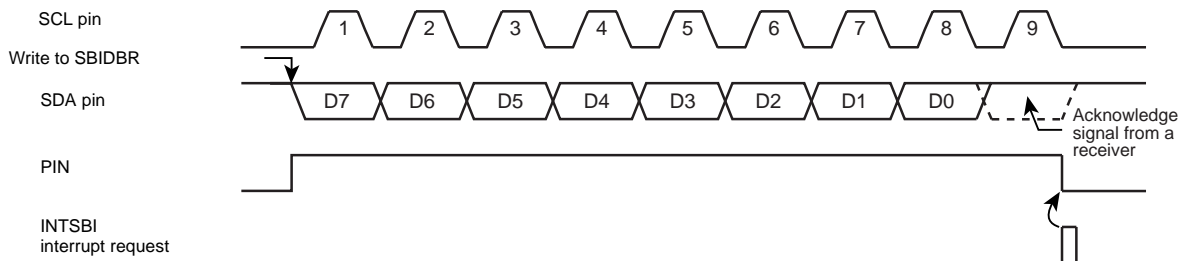


Figure 15-10 Example of when BC = “000”, ACK = “1”

(2) When the TRX is “0” (Receiver mode)

When the next transmitted data is other than of 8 bits, set the BC again. Set the ACK to “1” and read the received data from the SBIDBR (Reading data is undefined immediately after a slave address is sent). After the data is read, the PIN becomes “1”. A serial bus interface circuit outputs a serial clock pulse to the SCL pin to transfer next 1-word of data and sets the SDA pin to “0” at the acknowledge signal timing.

An INTSBI interrupt request occurs and the PIN becomes “0”. Then a serial bus interface circuit outputs a clock pulse for 1-word of data transfer and the acknowledge signal each time that received data is read from the SBIDBR.

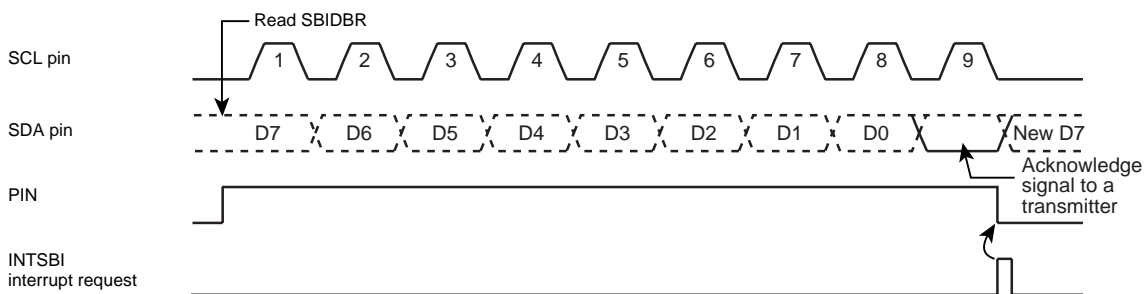


Figure 15-11 Example of when BC = “000”, ACK = “1”

To make the transmitter terminate transmit, clear the ACK to “0” before reading data which is 1-word before the last data to be received. A serial bus interface circuit does not generate a clock pulse for the acknowledge signal by clearing ACK. In the interrupt routine of end of transmission, when the BC is set to “001” and read the data, PIN is set to “1” and generates a clock pulse for a 1-bit data transfer. In this case, since the master device is a receiver, the SDA line on a bus keeps the high-level. The transmitter receives the high-level signal as an ACK signal. The receiver indicates to the transmitter that data transfer is complete.

After 1-bit data is received and an interrupt request has occurred, generate the stop condition to terminate data transfer.

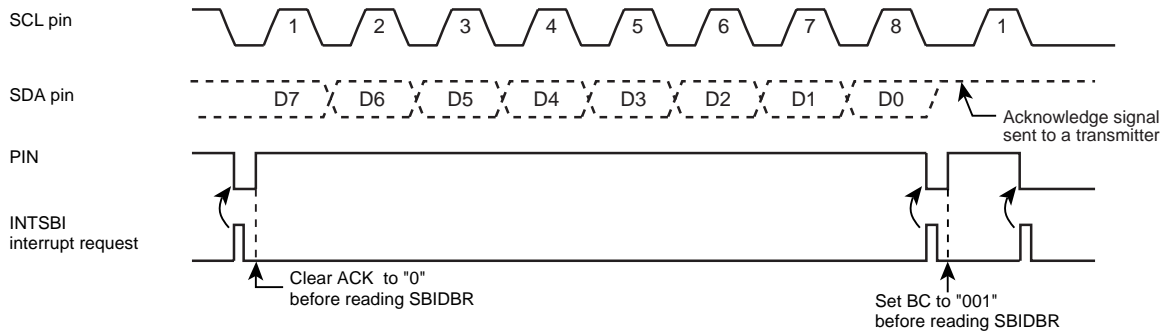


Figure 15-12 Termination of Data Transfer in Master Receiver Mode

15.6.3.2 When the MST is "0" (Slave mode)

In the slave mode, a serial bus interface circuit operates either in normal slave mode or in slave mode after losing arbitration.

In the slave mode, the conditions of generating INTSBI interrupt request are follows:

- At the end of acknowledge signal when the received slave address matches to the value set by the I2CAR
- At the end of acknowledge signal when a "GENERAL CALL" is received
- At the end of transferring or receiving after matching of slave address or receiving of "GENERAL CALL"

A serial bus interface circuit changes to a slave mode if arbitration is lost in the master mode. And an INTSBI interrupt request occurs when word data transfer terminates after losing arbitration. The behavior of INTSBI interrupt request and PIN after losing arbitration are shown in Table 15-3.

Table 15-3 The Behavior of INTSBI interrupt request and PIN after Losing Arbitration

	When the Arbitration Lost Occurs during Transmission of Slave Address as a Master	When the Arbitration Lost Occurs during Transmission of Data as a Master Transmit Mode
INTSBI interrupt request	INTSBI interrupt request is generated at the termination of word data.	
PIN	When the slave address matches the value set by I2CAR, the PIN is cleared to "0" by generating of INTSBI interrupt request. When the slave address doesn't match the value set by I2CAR, the PIN keeps "1".	PIN keeps "1" (PIN is not cleared to "0").

When an INTSBI interrupt request occurs, the PIN (bit 4 in the SBICRB) is reset, and the SCL pin is set to low level. Either reading or writing from or to the SBIDBR or setting the PIN to "1" releases the SCL pin after taking  $t_{LOW}$ .

Check the AL (Bit3 in the SBISRB), the TRX (Bit6 in the SBISRB), the AAS (Bit2 in the SBISRB), and the AD0 (Bit1 in the SBISRB) and implements processes according to conditions listed in "Table 15-4 Operation in the Slave Mode".

Table 15-4 Operation in the Slave Mode

TRX	AL	AAS	AD0	Conditions	Process
1	1	1	0	A serial bus interface circuit loses arbitration when transmitting a slave address. And receives a slave address of which the value of the direction bit sent from another master is "1".	Set the number of bits in 1 word to the BC and write transmitted data to the SBIDBR.
				In the slave receiver mode, a serial bus interface circuit receives a slave address of which the value of the direction bit sent from the master is "1".	
	0	0	0	In the slave transmitter mode, 1-word data is transmitted.	Test the LRB. If the LRB is set to "1", set the PIN to "1" since the receiver does not request next data. Then, clear the TRX to "0" to release the bus. If the LRB is set to "0", set the number of bits in 1 word to the BC and write transmitted data to the SBIDBR since the receiver requests next data.
0				1	1/0
	0	0	A serial bus interface circuit loses arbitration when transmitting a slave address or data. And terminates transferring word data.		
			0	1	1/0
	0	1/0			

Note: In the slave mode, if the slave address set in I2CAR is "00H", a START Byte "01H" in I<sup>2</sup>C bus standard is received, the device detects slave address match and the TRX is set to "1".

#### 15.6.4 Stop condition generation

When the BB is "1", a sequence of generating a stop condition is started by setting "1" to the MST, TRX and PIN, and clear "0" to the BB. Do not modify the contents of the MST, TRX, BB, PIN until a stop condition is generated on a bus.

When a SCL line on a bus is pulled-down by other devices, a serial bus interface circuit generates a stop condition after they release a SCL line.

The time from the releasing SCL line until the generating the STOP condition takes  $t_{LOW}$ .

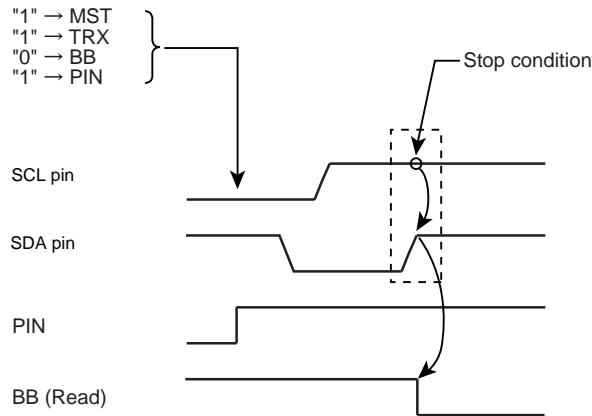


Figure 15-13 Stop Condition Generation

### 15.6.5 Restart

Restart is used to change the direction of data transfer between a master device and a slave device during transferring data. The following explains how to restart a serial bus interface circuit.

Clear “0” to the MST, TRX and BB and set “1” to the PIN. The SDA pin retains the high-level and the SCL pin is released. Since a stop condition is not generated on a bus, a bus is assumed to be in a busy state from other devices. Test the BB until it becomes “0” to check that the SCL pin of a serial bus interface circuit is released. Test the LRB until it becomes “1” to check that the SCL line on a bus is not pulled-down to the low level by other devices. After confirming that a bus stays in a free state, generate a start condition with procedure "15.6.2 Start condition and slave address generation".

In order to meet setup time when restarting, take at least 4.7 μs of waiting time by software from the time of restarting to confirm that a bus is free until the time to generate a start condition.

Note:When the master is in the receiver mode, it is necessary to stop the data transmission from the slave device before the STOP condition is generated. To stop the transmission, the master device make the slave device receiving a negative acknowledge. Therefore, the LRB is "1" before generating the Restart and it can not be confirmed that SCL line is not pulled-down by other devices. Please confirm the SCL line state by reading the port.

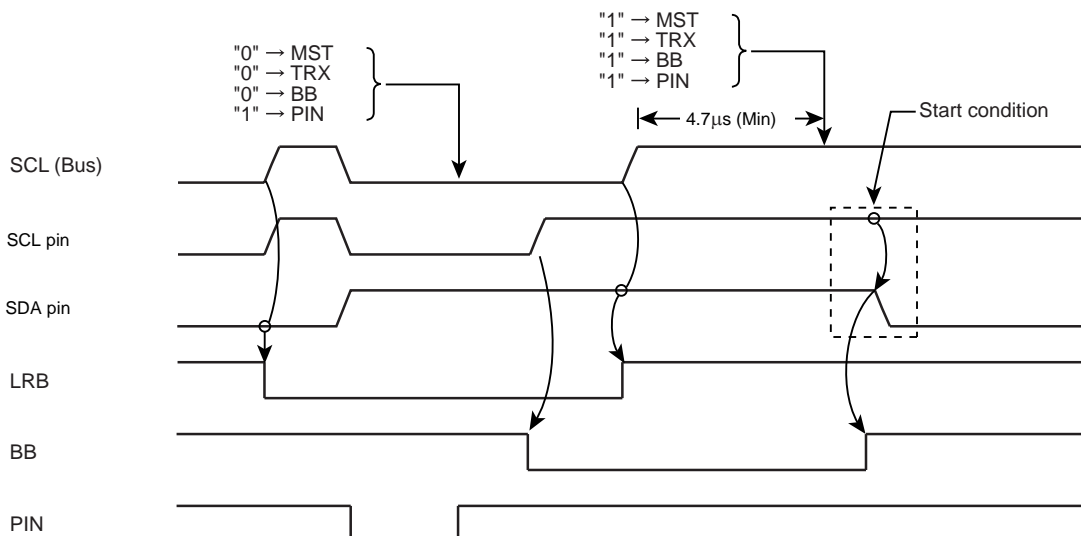


Figure 15-14 Timing Diagram when Restarting



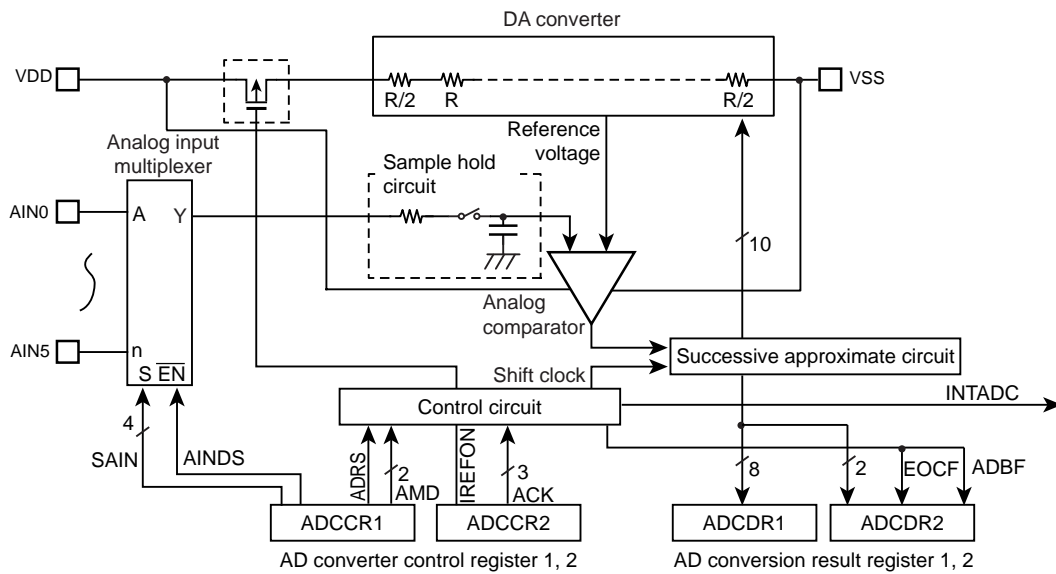
## 16. 10-bit AD Converter (ADC)

The TMP86FH92DMG have a 10-bit successive approximation type AD converter.

### 16.1 Configuration

The circuit configuration of the 10-bit AD converter is shown in Figure 16-1.

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



Note: Before using AD converter, set appropriate value to I/O port register combining a analog input port. For details, see the section on "I/O ports".

Figure 16-1 10-bit AD Converter

## 16.2 Register configuration

The AD converter consists of the following four registers:

1. AD converter control register 1 (ADCCR1)

This register selects the analog channels and operation mode (Software start or repeat) in which to perform AD conversion and controls the AD converter as it starts operating.

2. AD converter control register 2 (ADCCR2)

This register selects the AD conversion time and controls the connection of the DA converter (Ladder resistor network).

3. AD converted value register 1 (ADCDR1)

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

4. AD converted value register 2 (ADCDR2)

This register monitors the operating status of the AD converter.

### AD Converter Control Register 1

ADCCR1 (000EH)	7	6	5	4	3	2	1	0	
	ADRS	AMD		AINDS	SAIN				(Initial value: 0001 0000)

ADRS	AD conversion start	0: - 1: AD conversion start	R/W
AMD	AD operating mode	00: AD operation disable 01: Software start mode 10: Reserved 11: Repeat mode	
AINDS	Analog input control	0: Analog input enable 1: Analog input disable	
SAIN	Analog input channel select	0000: AIN0 0001: AIN1 0010: AIN2 0011: AIN3 0100: AIN4 0101: AIN5 0110: Reserved 0111: Reserved 1000: Reserved 1001: Reserved 1010: Reserved 1011: Reserved 1100: Reserved 1101: Reserved 1110: Reserved 1111: Reserved	

Note 1: Select analog input channel during AD converter stops (ADCDR2<ADBF> = "0").

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

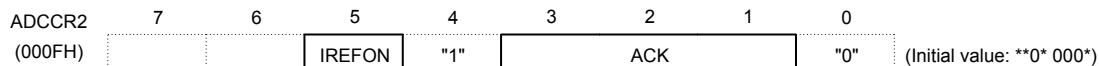
Note 3: During conversion, Do not perform port output instruction to maintain a precision for all of the pins because analog input port use as general input port. And for port near to analog input, Do not input intense signaling of change.

Note 4: The ADCCR1<ADRS> is automatically cleared to "0" after starting conversion.

Note 5: Do not set ADCCR1<ADRS> newly again during AD conversion. Before setting ADCCR1<ADRS> newly again, check ADCDR2<EOCF> to see that the conversion is completed or wait until the interrupt signal (INTADC) is generated (e.g., interrupt handling routine).

Note 6: After STOP or SLOW/SLEEP mode are started, AD converter control register1 (ADCCR1) is all initialized and no data can be written in this register. Therefore, to use AD converter again, set the ADCCR1 newly after returning to NORMAL1 or NORMAL2 mode.

AD Converter Control Register 2



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

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

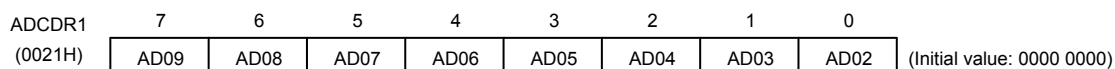
Table 16-1 ACK setting and Conversion time

Condition ACK	Conversion time	16 MHz	8 MHz	4 MHz	2 MHz	10 MHz	5 MHz	2.5 MHz
000	39/fc	-	-	-	19.5 μs	-	-	15.6 μs
001	Reserved							
010	78/fc	-	-	19.5 μs	39.0 μs	-	15.6 μs	31.2 μs
011	156/fc	-	19.5 μs	39.0 μs	78.0 μs	15.6 μs	31.2 μs	62.4 μs
100	312/fc	19.5 μs	39.0 μs	78.0 μs	156.0 μs	31.2 μs	62.4 μs	124.8 μs
101	624/fc	39.0 μs	78.0 μs	156.0 μs	-	62.4 μs	124.8 μs	-
110	1248/fc	78.0 μs	156.0 μs	-	-	124.8 μs	-	-
111	Reserved							

- Note 1: Setting for "-" in the above table are inhibited. fc: High Frequency oscillation clock [Hz]
- Note 2: Set conversion time setting should be kept more than the following time by Power supply voltage(VDD) .

- VDD = 4.5 to 5.5 V      15.6 μs and more
- VDD = 2.7 to 5.5 V      31.2 μs and more

AD Converted value Register 1



## AD Converted value Register 2



EOCF	AD conversion end flag	0: Before or during conversion 1: Conversion completed	Read only
ADBF	AD conversion BUSY flag	0: During stop of AD conversion 1: During AD conversion	

Note 1: The ADCDR2<EOCF> is cleared to "0" when reading the ADCDR1. Therefore, the AD conversion result should be read to ADCDR2 more first than ADCDR1.

Note 2: The ADCDR2<ADBF> is set to "1" when AD conversion starts, and cleared to "0" when AD conversion finished. It also is cleared upon entering STOP mode or SLOW mode.

Note 3: If a read instruction is executed for ADCDR2, read data of bit3 to bit0 are unstable.

## 16.3 Function

### 16.3.1 Software Start Mode

After setting ADCCR1<AMD> to “01” (software start mode), set ADCCR1<ADRS> to “1”. AD conversion of the voltage at the analog input pin specified by ADCCR1<SAIN> is thereby started.

After completion of the AD conversion, the conversion result is stored in AD converted value registers (ADCDR1, ADCDR2) and at the same time ADCDR2<EOCF> is set to 1, the AD conversion finished interrupt (INTADC) is generated.

ADRS is automatically cleared after AD conversion has started. Do not set ADCCR1<ADRS> newly again (Restart) during AD conversion. Before setting ADRS newly again, check ADCDR2<EOCF> to see that the conversion is completed or wait until the interrupt signal (INTADC) is generated (e.g., interrupt handling routine).

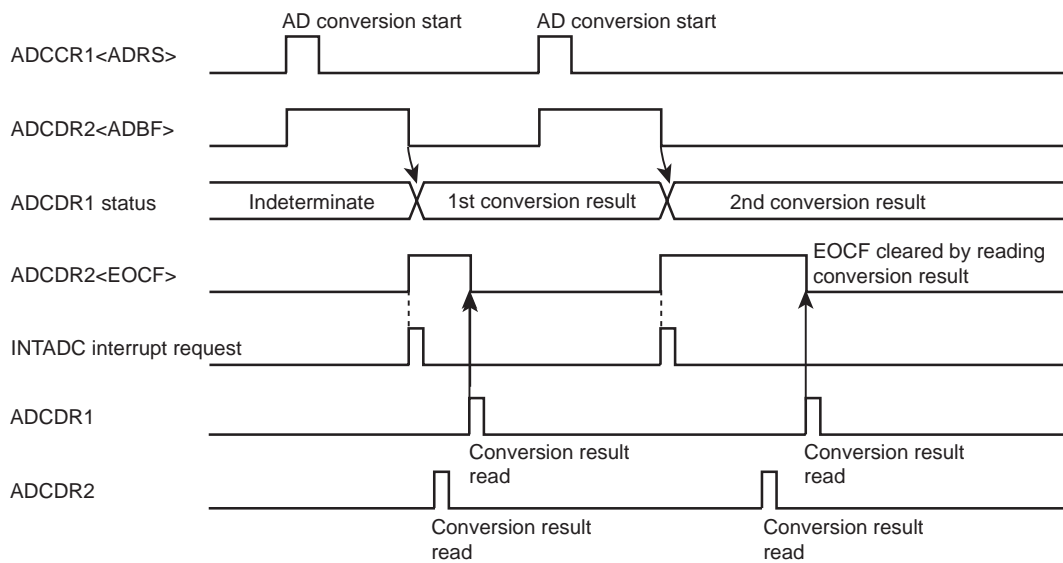


Figure 16-2 Software Start Mode

### 16.3.2 Repeat Mode

AD conversion of the voltage at the analog input pin specified by ADCCR1<SAIN> is performed repeatedly. In this mode, AD conversion is started by setting ADCCR1<ADRS> to “1” after setting ADCCR1<AMD> to “11” (Repeat mode).

After completion of the AD conversion, the conversion result is stored in AD converted value registers (ADCDR1, ADCDR2) and at the same time ADCDR2<EOCF> is set to 1, the AD conversion finished interrupt (INTADC) is generated.

In repeat mode, each time one AD conversion is completed, the next AD conversion is started. To stop AD conversion, set ADCCR1<AMD> to “00” (Disable mode) by writing 0s. The AD convert operation is stopped immediately. The converted value at this time is not stored in the AD converted value register.

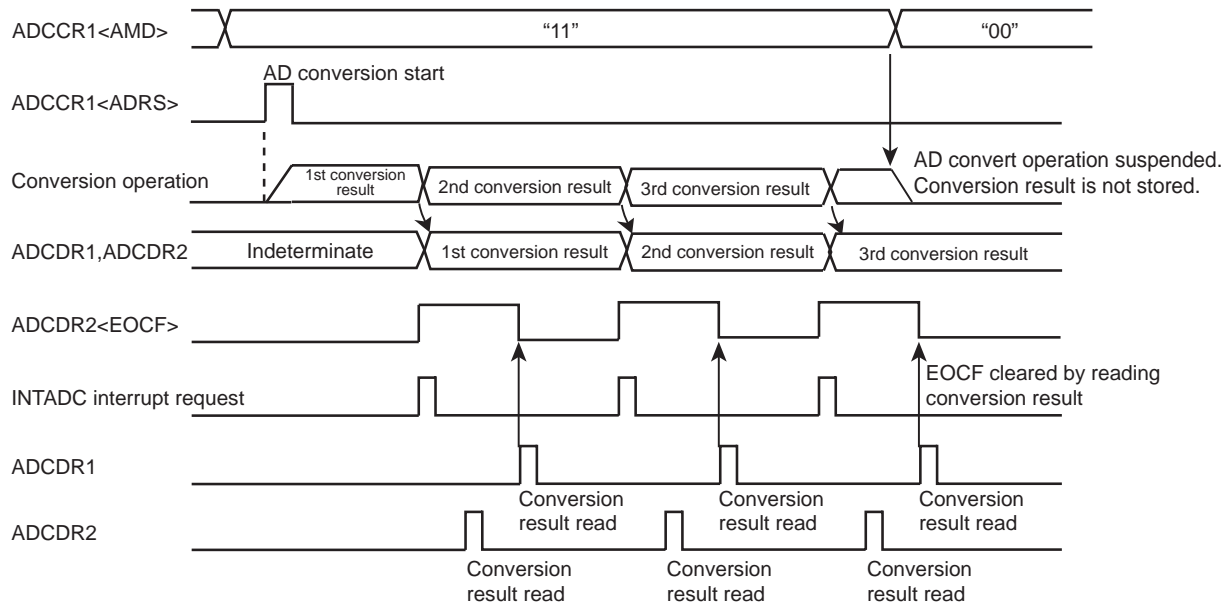


Figure 16-3 Repeat Mode

### 16.3.3 Register Setting

- Set up the AD converter control register 1 (ADCCR1) as follows:
  - Choose the channel to AD convert using AD input channel select (SAIN).
  - Specify analog input enable for analog input control (AINDS).
  - Specify AMD for the AD converter control operation mode (software or repeat mode).
- Set up the AD converter control register 2 (ADCCR2) as follows:
  - Set the AD conversion time using AD conversion time (ACK). For details on how to set the conversion time, refer to Figure 16-1 and AD converter control register 2.
  - Choose IREFON for DA converter control.
- After setting up (1) and (2) above, set AD conversion start (ADRS) of AD converter control register 1 (ADCCR1) to "1". If software start mode has been selected, AD conversion starts immediately.
- After an elapse of the specified AD conversion time, the AD converted value is stored in AD converted value register 1 (ADCDR1) and the AD conversion finished flag (EOCF) of AD converted value register 2 (ADCDR2) is set to "1", upon which time AD conversion interrupt INTADC is generated.
- EOCF is cleared to "0" by a read of the conversion result. However, if reconverted before a register read, although EOCF is cleared the previous conversion result is retained until the next conversion is completed.

Example :After selecting the conversion time 19.5  $\mu$ s at 16 MHz and the analog input channel AIN3 pin, perform AD conversion once. After checking EOCF, read the converted value, store the lower 2 bits in address 0009EH and store the upper 8 bits in address 0009FH in RAM. The operation mode is software start mode.

```

: (port setting)      :                               ;Set port register appropriately before setting AD
:                               ; converter registers.
:                               ;(Refer to section I/O port in details)
LD      (ADCCR1) , 00100011B   ; Select AIN3
LD      (ADCCR2) , 11011000B   ;Select conversion time(312/fc) and operation mode

SLOOP : SET      (ADCCR1) . 7     ; ADRS = 1(AD conversion start)
        TEST     (ADCDR2) . 5     ; EOCF= 1 ?
        JRS      T, SLOOP

        LD      A , (ADCDR2)      ; Read result data
        LD      (9EH) , A
        LD      A , (ADCDR1)      ; Read result data
        LD      (9FH) , A

```

## 16.4 STOP/SLOW Modes during AD Conversion

When standby mode (STOP or SLOW mode) is entered forcibly during AD conversion, the AD convert operation is suspended and the AD converter is initialized (ADCCR1 and ADCCR2 are initialized to initial value). Also, the conversion result is indeterminate. (Conversion results up to the previous operation are cleared, so be sure to read the conversion results before entering standby mode (STOP or SLOW mode).) When restored from standby mode (STOP or SLOW mode), AD conversion is not automatically restarted, so it is necessary to restart AD conversion. Note that since the analog reference voltage is automatically disconnected, there is no possibility of current flowing into the analog reference voltage.

## 16.5 Analog Input Voltage and AD Conversion Result

The analog input voltage is corresponded to the 10-bit digital value converted by the AD as shown in Figure 16-4.

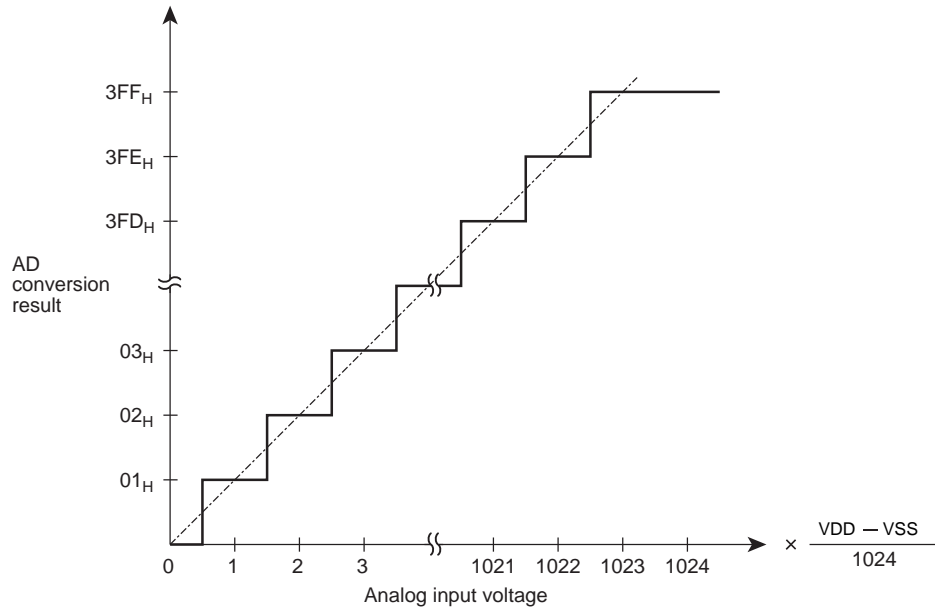


Figure 16-4 Analog Input Voltage and AD Conversion Result (Typ.)



## 16.6 Precautions about AD Converter

### 16.6.1 Analog input pin voltage range

Make sure the analog input pins (AIN0 to AIN5) are used at voltages within VDD to VSS. If any voltage outside this range is applied to one of the analog input pins, the converted value on that pin becomes uncertain. The other analog input pins also are affected by that.

### 16.6.2 Analog input shared pins

The analog input pins (AIN0 to AIN5) 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.

### 16.6.3 Noise Countermeasure

The internal equivalent circuit of the analog input pins is shown in Figure 16-5. The higher the output impedance of the analog input source, more easily they are susceptible to noise. Therefore, make sure the output impedance of the signal source in your design is 5 k $\Omega$  or less. Toshiba also recommends attaching a capacitor external to the chip.

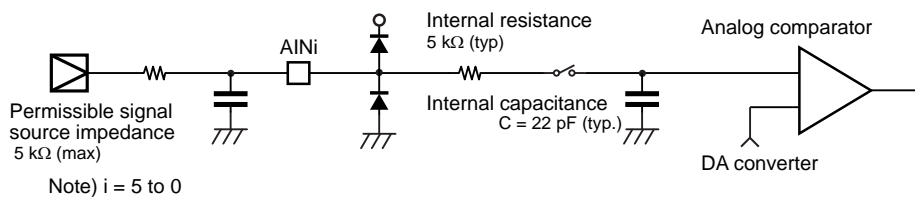


Figure 16-5 Analog Input Equivalent Circuit and Example of Input Pin Processing



## 17. Key-on Wakeup (KWU)

TMP86FH92DMG have four pins P34 to P37, in addition to the P20 ( $\overline{\text{INT5}}/\overline{\text{STOP}}$ ) pin, that can be used to exit STOP mode.

When using these P34 to P37 pin's input to exit STOP mode, pay attention to the logic of P20 pin. In details, refer to the following section "17.2 Control".

### 17.1 Configuration

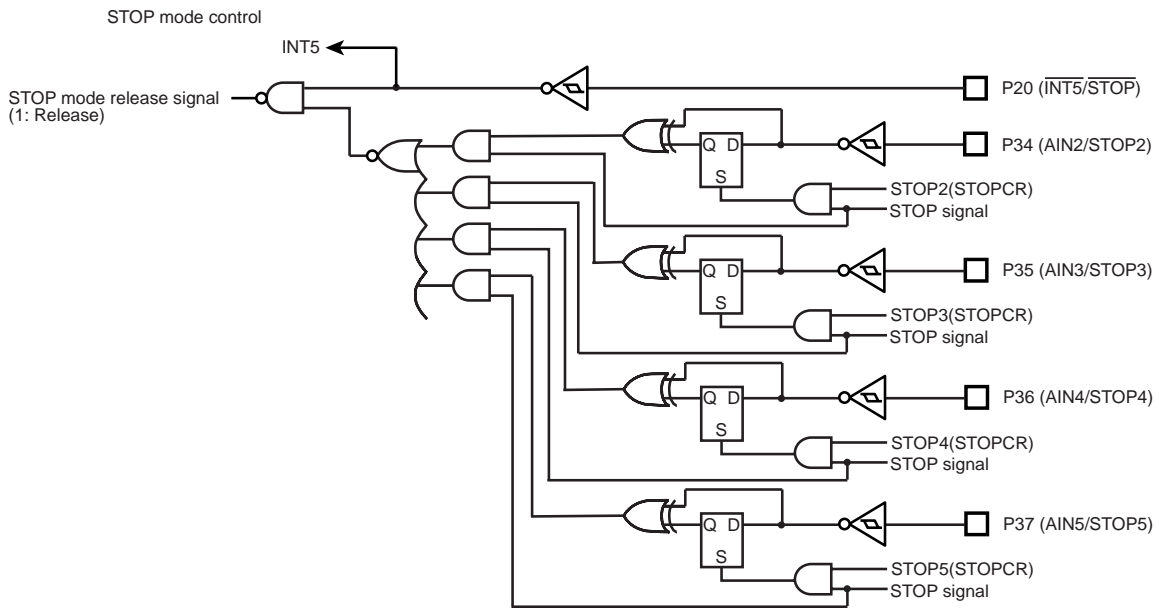
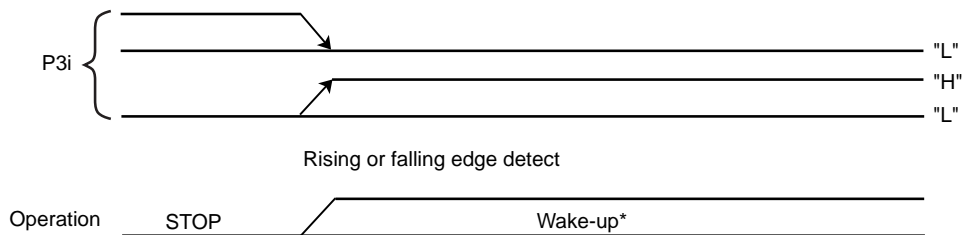


Figure 17-1 Key-on Wakeup Circuit

Example of STOP mode release operation

STOP mode release operation(P34 to 37)



\* : The time required for wakeup from releasing STOP mode includes the warming-up time. For details, refer to section "Control of Operation Modes".

Figure 17-2 Example of STOP Mode Release Operation

## 17.2 Control

The P34 to P37 (STOP2 to STOP5) pins can individually be disabled/enabled using Key-on Wakeup Control Register (STOPCR). When these pins are used as a release input of STOP mode, beforehand they set to key-on wakeup by each register of P3 Port. For details, refer to section "I/O Ports".

STOP mode can be entered by setting up the System Control Register (SYSCR1), and can be released by detecting the active edge (rising or falling edge) on any STOP2 to STOP5 pins which are available for STOP mode release.

Note: When using Key-on Wakeup function, select level mode ( set SYSCR1<RELM> to "1" ) for selection of STOP mode release method.

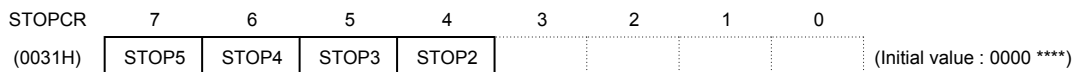
Although P20 pin is shared with  $\overline{\text{INT5}}$  and  $\overline{\text{STOP}}$  pin input, use mainly  $\overline{\text{STOP}}$  pin to release STOP mode. This is because Key-on Wakeup function is comprised of  $\overline{\text{STOP}}$  pin and STOP2 to STOP5 pins as shown in the configuration diagram.

Note 1: When STOP mode release by an edge on STOP pin, follow one of the two methods described below.

- (1) Disable all of STOP2 to STOP5 pin inputs.
- (2) Fix STOP2 to 5 pin inputs high or low level.

Note 2: When using key-on wakeup (STOP2 to 5 pins) to exit STOP mode, make sure  $\overline{\text{STOP}}$  pin is held low and STOP2 to 5 pin inputs are held high or low level, because STOP mode release signal is created by OR circuit the  $\overline{\text{STOP}}$  pin input and the STOP2 to 5 pin input together.

### Key-on Wakeup STOP Mode Control Register



STOP2	STOP mode release by P34 (STOP2)	0: Disable	Write only
		1: Enable	
STOP3	STOP mode release by P35 (STOP3)	0: Disable	
		1: Enable	
STOP4	STOP mode release by P36 (STOP4)	0: Disable	
		1: Enable	
STOP5	STOP mode release by P37 (STOP5)	0: Disable	
		1: Enable	

<Example of STOP mode release> The device is released from STOP mode in the following condition.		
	P20( $\overline{\text{STOP}}$ )	P3x
STOP mode release using P3x (STOP2 to 5)	Level detection mode: Low Edge detection mode: Disable	Edge detection Rising or falling edge
STOP mode release using P20 ( $\overline{\text{STOP}}$ )	Level detection mode: High Edge detection mode: Rising edge	STOPCR: inhibited

Note: Assertion of the STOP mode release signal is not recognized within three instruction cycles after executing the STOP instruction.

## 18. Flash Memory

TMP86FH92DMG has 16384byte flash memory (address: C000H to FFFFH). The write and erase operations to the flash memory are controlled in the following three types of mode.

- MCU mode

The flash memory is accessed by the CPU control in the MCU mode. This mode is used for software bug correction and firmware change after shipment of the device since the write operation to the flash memory is available by retaining the application behavior.

- Serial PROM mode

The flash memory is accessed by the CPU control in the serial PROM mode. Use of the serial interface (UART) enables the flash memory to be controlled by the small number of pins. TMP86FH92DMG in the serial PROM mode supports on-board programming which enables users to program flash memory after the microcontroller is mounted on a user board.

- Parallel PROM mode

The parallel PROM mode allows the flash memory to be accessed as a stand-alone flash memory by the program writer provided by the third party. High-speed access to the flash memory is available by controlling address and data signals directly. For the support of the program writer, please ask Toshiba sales representative.

In the MCU and serial PROM modes, the flash memory control register (FLSCR) is used for flash memory control. This chapter describes how to access the flash memory using the flash memory control register (FLSCR) in the MCU and serial PROM modes.

Note 1: The 'Read Protect' described by data sheet of old edition was changed into 'Security Program'.

## 18.1 Flash Memory Control

The flash memory is controlled via the flash memory control register (FLSCR) and flash memory standby control register (FLSSTB).

### Flash Memory Control Register

FLSCR      7      6      5      4      3      2      1      0  
(0FFFH)      FLSMD      "1"                           (Initial value: 1100 \*\*\*\*)

FLSMD	Flash memory command sequence execution control	1100: Disable command sequence execution 0011: Enable command sequence execution Others: Reserved	R/W
-------	---	---	-----

Note 1: The command sequence of the flash memory can be executed only when FLSMD="0011B". In other cases, any attempts to execute the command sequence are ineffective.

Note 2: FLSMD must be set to either "1100B" or "0011B".

Note 3: Bits 3 through 0 in FLSCR are always read as don't care.

Note 4: Always set bit3 in FLSCR to "1".

### Flash Memory Standby Control Register

FLSSTB      7      6      5      4      3      2      1      0  
(0FE9H)                                                FSTB      (Initial value: \*\*\*\* \*\*\*)

FSTB	Flash memory standby control	0: Disable the standby function. 1: Enable the standby function.	Write only
------	------------------------------	---	------------

Note 1: When FSTB is set to 1, do not execute the read/write instruction to the flash memory because there is a possibility that the expected data is not read or the program is not operated correctly. If executing the read/write instruction, FSTB is initialized to 0 automatically.

Note 2: If an interrupt is issued when FSTB is set to 1, FSTB is initialized to 0 automatically and then the vector area of the flash memory is read.

Note 3: If the IDLE0/1/2, SLEEP0/1/2 or STOP mode is activated when FSTB is set to 1, FSTB is initialized to 0 automatically. In the IDLE0/1/2, SLEEP0/1/2 or STOP mode, the standby function operates regardless of FSTB setting.

#### 18.1.1 Flash Memory Command Sequence Execution Control (FLSCR<FLSMD>)

The flash memory can be protected from inadvertent write due to program error or microcontroller misoperation. This write protection feature is realized by disabling flash memory command sequence execution via the flash memory control register (write protect). To enable command sequence execution, set FLSCR<FLSMD> to "0011B". To disable command sequence execution, set FLSCR<FLSMD> to "1100B". After reset, FLSCR<FLSMD> is initialized to "1100B" to disable command sequence execution. Normally, FLSCR<FLSMD> should be set to "1100B" except when the flash memory needs to be written or erased.

#### 18.1.2 Flash Memory Standby Control (FLSSTB<FSTB>)

Low power consumption is enabled by cutting off the steady-state current of the flash memory. In the IDLE0/1/2, SLEEP0/1/2 or STOP mode, the steady-state current of the flash memory is cut off automatically.

When the program is executed in the RAM area (without accessing the flash memory) in the NORMAL 1/2 or SLOW1/2 mode, the current can be cut off by the control of the register. To cut off the steady-state current of the flash memory, set FLSSTB<FSTB> to "1" by the control program in the RAM area. The procedures for controlling the FLSSTB register are explained below.

(Steps 1 and 2 are controlled by the program in the flash memory, and steps 3 through 8 are controlled by the write control program executed in the RAM area.)

1. Transfer the control program of the FLSSTB register to the RAM area.
2. Jump to the RAM area.
3. Disable (DI) the interrupt master enable flag (IMF = "0").
4. Set FLSSTB<FSTB> to "1".
5. Execute the user program.
6. Repeat step 5 until the return request to the flash memory is detected.
7. Set FLSSTB<FSTB> to "0".
8. Jump to the flash memory area.

Note 1: The standby function is not operated by setting FLSSTB<FSTB> with the program in the flash memory. You must set FLSSTB<FSTB> by the program in the RAM area.

Note 2: To use the standby function by setting FLSSTB<FSTB> to "1" with the program in the RAM area, FLSSTB<FSTB> must be set to "0" by the program in the RAM area before returning the program control to the flash memory. If the program control is returned to the flash memory with FLSSTB<FSTB> set to "1", the program may malfunction and run out of control.

## 18.2 Command Sequence

The command sequence in the MCU and the serial PROM modes consists of six commands (JEDEC compatible), as shown in Table 18-1. Addresses specified in the command sequence are recognized with the lower 12 bits (excluding BA, SA, and FF7FH used for security program). The upper 4 bits are used to specify the flash memory area,

Table 18-1 Command Sequence

	Command Sequence	1st Bus Write Cycle		2nd Bus Write Cycle		3rd Bus Write Cycle		4th Bus Write Cycle		5th Bus Write Cycle		6th Bus Write Cycle	
		Address	Data	Address	Data	Address	Data	Address	Data	Address	Data	Address	Data
1	Byte program	555H	AAH	AAAH	55H	555H	A0H	BA (Note 1)	Data (Note 1)	-	-	-	-
2	Sector Erase (4-kbyte Erase)	555H	AAH	AAAH	55H	555H	80H	555H	AAH	AAAH	55H	SA (Note 2)	30H
3	Chip Erase (All Erase)	555H	AAH	AAAH	55H	555H	80H	555H	AAH	AAAH	55H	555H	10H
4	Product ID Entry	555H	AAH	AAAH	55H	555H	90H	-	-	-	-	-	-
5	Product ID Exit	XXH	F0H	-	-	-	-	-	-	-	-	-	-
	Product ID Exit	555H	AAH	AAAH	55H	555H	F0H	-	-	-	-	-	-
6	Security Program	555H	AAH	AAAH	55H	555H	A5H	FF7FH	00H	-	-	-	-

Note 1: Set the address and data to be written.

Note 2: The area to be erased is specified with the upper 4 bits of the address.

### 18.2.1 Byte Program

This command writes the flash memory for each byte unit. The addresses and data to be written are specified in the 4th bus write cycle. Each byte can be programmed in a maximum of 40  $\mu$ s. The next command sequence cannot be executed until the write operation is completed. To check the completion of the write operation, perform read operations repeatedly until the same data is read twice from the same address in the flash memory. During the write operation, any consecutive attempts to read from the same address is reversed bit 6 of the data (toggling between 0 and 1).

Note: To rewrite data to Flash memory addresses at which data (including FFH) is already written, make sure to erase the existing data by "sector erase" or "chip erase" before rewriting data.

### 18.2.2 Sector Erase (4-kbyte Erase)

This command erases the flash memory in units of 4 kbytes. The flash memory area to be erased is specified by the upper 4 bits of the 6th bus write cycle address. For example, to erase 4 kbytes from F000H to FFFFH, specify one of the addresses in F000H-FFFFH as the 6th bus write cycle. The sector erase command is effective only in the MCU and serial PROM modes, and it cannot be used in the parallel PROM mode.

A maximum of 30 ms is required to erase 4 kbytes. The next command sequence cannot be executed until the erase operation is completed. To check the completion of the erase operation, perform read operations repeatedly for data polling until the same data is read twice from the same address in the flash memory. During the erase operation, any consecutive attempts to read from the same address is reversed bit 6 of the data (toggling between 0 and 1).



### 18.2.3 Chip Erase (All Erase)

This command erases the entire flash memory in approximately 30 ms. The next command sequence cannot be executed until the erase operation is completed. To check the completion of the erase operation, perform read operations repeatedly for data polling until the same data is read twice from the same address in the flash memory. During the erase operation, any consecutive attempts to read from the same address is reversed bit 6 of the data (toggling between 0 and 1). After the chip is erased, all bytes contain FFH.

### 18.2.4 Product ID Entry

This command activates the Product ID mode. In the Product ID mode, the vendor ID, the flash ID, and the security program status can be read from the flash memory.

Table 18-2 Values To Be Read in the Product ID Mode

Address	Meaning	Read Value
F000H	Vendor ID	98H
F001H	Flash macro ID	41H
F002H	Flash size	0EH: 60 kbytes 0BH: 48 kbytes 07H: 32 kbytes 05H: 24 kbytes 03H: 16 kbytes 01H: 8 kbytes 00H: 4 kbytes
FF7FH	Security program status	FFH: Security program disabled Other than FFH: Security program enabled

Note: The value at address F002H (flash size) depends on the size of flash memory incorporated in each product. For example, if the product has 60-kbyte flash memory, "0EH" is read from address F002H.

### 18.2.5 Product ID Exit

This command is used to exit the Product ID mode.

### 18.2.6 Security Program

This command enables the read protection setting in the flash memory. When the security program is enabled, the flash memory cannot be read in the parallel PROM mode. In the serial PROM mode, the flash write and RAM loader commands cannot be executed.

To disable the security program setting, it is necessary to execute the chip erase command sequence. Whether or not the security program is enabled can be checked by reading FF7FH in the Product ID mode. For details, see Table 18-2.

It takes a maximum of 40  $\mu$ s to set security program in the flash memory. The next command sequence cannot be executed until this operation is completed. To check the completion of the security program operation, perform read operations repeatedly for data polling until the same data is read twice from the same address in the flash memory. During the security program operation, any attempts to read from the same address is reversed bit 6 of the data (toggling between 0 and 1).

### 18.3 Toggle Bit (D6)

After the byte program, chip erase, and security program command sequence is executed, any consecutive attempts to read from the same address is reversed bit 6 (D6) of the data (toggling between 0 and 1) until the operation is completed. Therefore, this toggle bit provides a software mechanism to check the completion of each operation. Usually perform read operations repeatedly for data polling until the same data is read twice from the same address in the flash memory. After the byte program, chip erase, or security program command sequence is executed, the initial read of the toggle bit always produces a "1".

## 18.4 Access to the Flash Memory Area

When the write, erase and security program are set in the flash memory, read and fetch operations cannot be performed in the entire flash memory area. Therefore, to perform these operations in the entire flash memory area, access to the flash memory area by the control program in the BOOTROM or RAM area. (The flash memory program cannot write to the flash memory.) The serial PROM or MCU mode is used to run the control program in the BOOTROM or RAM area.

Note 1: The flash memory can be written or read for each byte unit. Erase operations can be performed either in the entire area or in units of 4 kbytes, whereas read operations can be performed by an one transfer instruction. However, the command sequence method is adopted for write and erase operations, requiring several-byte transfer instructions for each operation.

Note 2: To rewrite data to Flash memory addresses at which data (including FFH) is already written, make sure to erase the existing data by "sector erase" or "chip erase" before rewriting data.

### 18.4.1 Flash Memory Control in the Serial PROM Mode

The serial PROM mode is used to access to the flash memory by the control program provided in the BOOTROM area. Since almost of all operations relating to access to the flash memory can be controlled simply by the communication data of the serial interface (UART), these functions are transparent to the user. For the details of the serial PROM mode, see "Serial PROM Mode."

To access to the flash memory by using peripheral functions in the serial PROM mode, run the RAM loader command to execute the control program in the RAM area. The procedures to execute the control program in the RAM area is shown in "18.4.1.1 How to write to the flash memory by executing the control program in the RAM area (in the RAM loader mode within the serial PROM mode)".

#### 18.4.1.1 How to write to the flash memory by executing the control program in the RAM area (in the RAM loader mode within the serial PROM mode)

(Steps 1 and 2 are controlled by the BOOTROM, and steps 3 through 9 are controlled by the control program executed in the RAM area.)

1. Transfer the write control program to the RAM area in the RAM loader mode.
2. Jump to the RAM area.
3. Disable (DI) the interrupt master enable flag (IMF←"0").
4. Set FLSCR<FLSMD> to "0011B" (to enable command sequence execution).
5. Execute the erase command sequence.
6. Read the same flash memory address twice.  
(Repeat step 6 until the same data is read by two consecutive reads operations.)
7. Execute the write command sequence.
8. Read the same flash memory address twice.  
(Repeat step 8 until the same data is read by two consecutive reads operations.)
9. Set FLSCR<FLSMD> to "1100B" (to disable command sequence execution).

Note 1: Before writing to the flash memory in the RAM area, disable interrupts by setting the interrupt master enable flag (IMF) to "0". Usually disable interrupts by executing the DI instruction at the head of the write control program in the RAM area.

Note 2: Since the watchdog timer is disabled by the BOOTROM in the RAM loader mode, it is not required to disable the watchdog timer by the RAM loader program.

Example :After chip erasure, the program in the RAM area writes data 3FH to address F000H.

```

DI                                     : Disable interrupts (IMF←"0")
LD      (FLSCR),00111000B             : Enable command sequence execution.
LD      IX,0F555H
LD      IY,0FAAAH
LD      HL,0F000H
;#### Flash Memory Chip erase Process ####
LD      (IX),0AAH                     : 1st bus write cycle
LD      (IY),55H                      : 2nd bus write cycle
LD      (IX),80H                      : 3rd bus write cycle
LD      (IX),0AAH                     : 4th bus write cycle
LD      (IY),55H                      : 5th bus write cycle
LD      (IX),10H                      : 6th bus write cycle
sLOOP1: LD      W,(HL)
      CMP     W,(HL)
      JR      NZ,sLOOP1                : Loop until the same value is read.
;#### Flash Memory Write Process ####
LD      (IX),0AAH                     : 1st bus write cycle
LD      (IY),55H                      : 2nd bus write cycle
LD      (IX),0A0H                     : 3rd bus write cycle
LD      (HL),3FH                      : 4th bus write cycle, (F000H)=3FH
sLOOP2: LD      W,(HL)
      CMP     W,(HL)
      JR      NZ,sLOOP2                : Loop until the same value is read.
LD      (FLSCR),11001000B             : Disable command sequence execution.
sLOOP3: JP      sLOOP3

```

## 18.4.2 Flash Memory Control in the MCU mode

In the MCU mode, write operations are performed by executing the control program in the RAM area. Before execution of the control program, copy the control program into the RAM area or obtain it from the external using the communication pin. The procedures to execute the control program in the RAM area in the MCU mode are described below.

### 18.4.2.1 How to write to the flash memory by executing a user write control program in the RAM area (in the MCU mode)

(Steps 1 and 2 are controlled by the program in the flash memory, and steps 3 through 11 are controlled by the control program in the RAM area.)

1. Transfer the write control program to the RAM area.
2. Jump to the RAM area.
3. Disable (DI) the interrupt master enable flag (IMF←"0").
4. Disable the watchdog timer, if it is used.
5. Set FLSCR<FLSMD> to "0011B" (to enable command sequence execution).
6. Execute the erase command sequence.
7. Read the same flash memory address twice.  
(Repeat step 7 until the same data is read by two consecutive read operations.)
8. Execute the write command sequence.
9. Read the same flash memory address twice.  
(Repeat step 9 until the same data is read by two consecutive read operations.)
10. Set FLSCR<FLSMD> to "1100B" (to disable command sequence execution).
11. Jump to the flash memory area.

Note 1: Before writing to the flash memory in the RAM area, disable interrupts by setting the interrupt master enable flag (IMF) to "0". Usually disable interrupts by executing the DI instruction at the head of the write control program in the RAM area.

Note 2: When writing to the flash memory, do not intentionally use non-maskable interrupts (the watchdog timer must be disabled if it is used). If a non-maskable interrupt occurs while the flash memory is being written, unexpected data is read from the flash memory (interrupt vector), resulting in malfunction of the microcontroller.

Example :After sector erasure (E000H-EFFFH), the program in the RAM area writes data 3FH to address E000H.

```

DI                : Disable interrupts (IMF←"0")
LD                (WDTCR2),4EH      : Clear the WDT binary counter.
LDW              (WDTCR1),0B101H    : Disable the WDT.
LD               (FLSCR),00111000B  : Enable command sequence execution.
LD               IX,0F555H
LD               IY,0FAAAH
LD               HL,0E000H
; #### Flash Memory Sector Erase Process ####
LD               (IX),0AAH          : 1st bus write cycle
LD               (IY),55H           : 2nd bus write cycle
LD               (IX),80H           : 3rd bus write cycle
LD               (IX),0AAH          : 4th bus write cycle
LD               (IY),55H           : 5th bus write cycle
LD               (HL),30H           : 6th bus write cycle
sLOOP1:          LD               W,(HL)
                  CMP            W,(HL)
                  JR              NZ,sLOOP1      : Loop until the same value is read.
; #### Flash Memory Write Process ####
LD               (IX),0AAH          : 1st bus write cycle
LD               (IY),55H           : 2nd bus write cycle
LD               (IX),0A0H          : 3rd bus write cycle
LD               (HL),3FH           : 4th bus write cycle, (E000H)=3FH
sLOOP2:          LD               W,(HL)
                  CMP            W,(HL)
                  JR              NZ,sLOOP2      : Loop until the same value is read.
LD               (FLSCR),11001000B  : Disable command sequence execution.
JP               XXXXH              : Jump to the flash memory area.

```

Example :This write control program reads data from address F000H and stores it to 98H in the RAM area.

```

LD               A,(0F000H)          : Read data from address F000H.
LD               (98H),A            : Store data to address 98H.

```

## 19. Serial PROM Mode

### 19.1 Outline

The TMP86FH92DMG has a 2048 byte BOOTROM (Mask ROM) for programming to flash memory. The BOOTROM is available in the serial PROM mode, and controlled by TEST, BOOT and RESET pins. Communication is performed via UART. The serial PROM mode has seven types of operating mode: Flash memory writing, RAM loader, Flash memory SUM output, Product ID code output, Flash memory status output, Flash memory erasing and Flash memory security program setting. Memory address mapping in the serial PROM mode differs from that in the MCU mode. Figure 19-1 shows memory address mapping in the serial PROM mode.

Table 19-1 Operating Range in the Serial PROM Mode

Parameter	Min	Max	Unit
Power supply	4.5	5.5	V
High frequency (Note)	2	16	MHz

Note: Though included in above operating range, some of high frequencies are not supported in the serial PROM mode. For details, refer to "Table 19-5".

### 19.2 Memory Mapping

The Figure 19-1 shows memory mapping in the Serial PROM mode and MCU mode.

In the serial PROM mode, the BOOTROM (Mask ROM) is mapped in addresses from 7800H to 7FFFH.

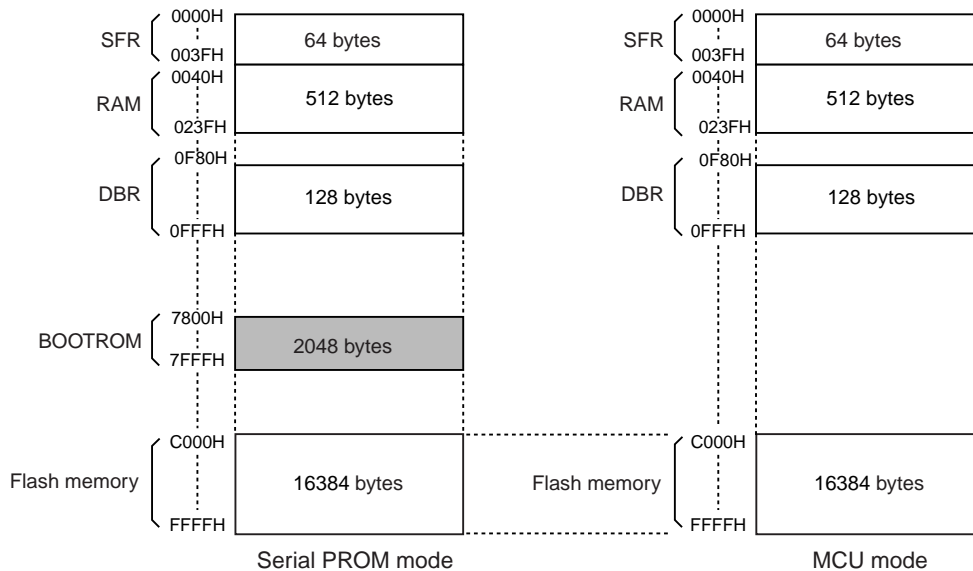



Figure 19-1 Memory Address Maps

## 19.3 Serial PROM Mode Setting

### 19.3.1 Serial PROM Mode Control Pins

To execute on-board programming, activate the serial PROM mode. Table 19-2 shows pin setting to activate the serial PROM mode.

Table 19-2 Serial PROM Mode Setting

Pin	Setting
TEST pin	High
BOOT/RXD1 pin	High
$\overline{\text{RESET}}$ pin	

Note: The BOOT pin is shared with the UART communication pin (RXD1 pin) in the serial PROM mode. This pin is used as UART communication pin after activating serial PROM mode

### 19.3.2 Pin Function

In the serial PROM mode, TXD1 (P00) and RXD1 (P01) are used as a serial interface pin.

Table 19-3 Pin Function in the Serial PROM Mode

Pin Name (Serial PROM Mode)	Input/ Output	Function	Pin Name (MCU Mode)
TXD1	Output	Serial data output	P00
BOOT/RXD1	Input/In- put	Serial PROM mode control/Serial data input	P01
$\overline{\text{RESET}}$	Input	Serial PROM mode control	$\overline{\text{RESET}}$
TEST	Input	Fixed to high	TEST
VDD	Power supply	4.5 to 5.5 V	
VSS	Power supply	0 V	
I/O ports except P00, P01	I/O	These ports are in the high-impedance state in the serial PROM mode. The input level is fixed to the port inputs with a hardware feature to prevent overlap current. (The port inputs are invalid.) To make the port inputs valid, set the pin of the SPCR register to "1" by the RAM loader control program.	
XIN	Input	Self-oscillate with an oscillator.	(Note 2)
XOUT	Output		

Note 1: During on-board programming with other parts mounted on a user board, be careful no to affect these communication control pins.

Note 2: Operating range of high frequency in serial PROM mode is 2 MHz to 16 MHz.



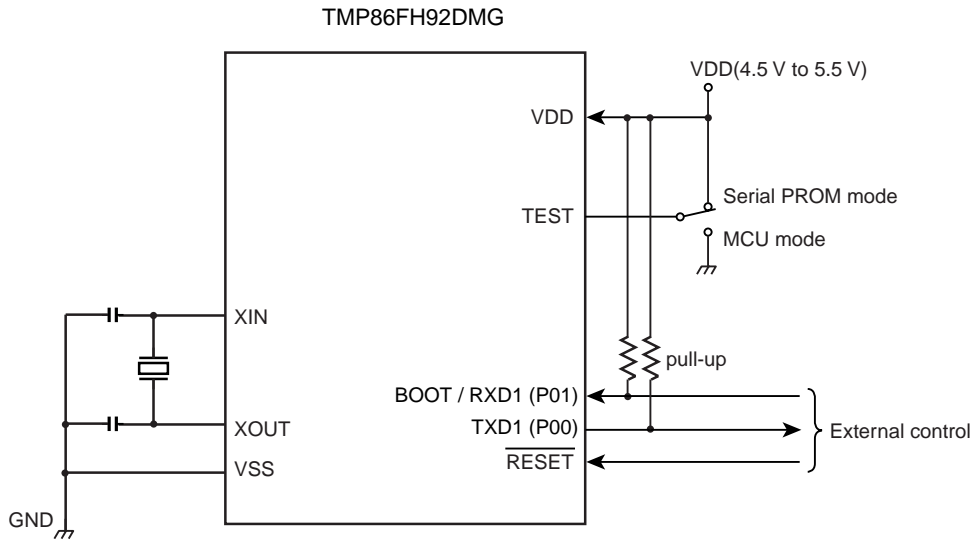


Figure 19-2 Serial PROM Mode Pin Setting

Note 1: For connection of other pins, refer to "Table 19-3 Pin Function in the Serial PROM Mode".

### 19.3.3 Example Connection for On-Board Writing

Figure 19-3 shows an example connection to perform on-board writing.

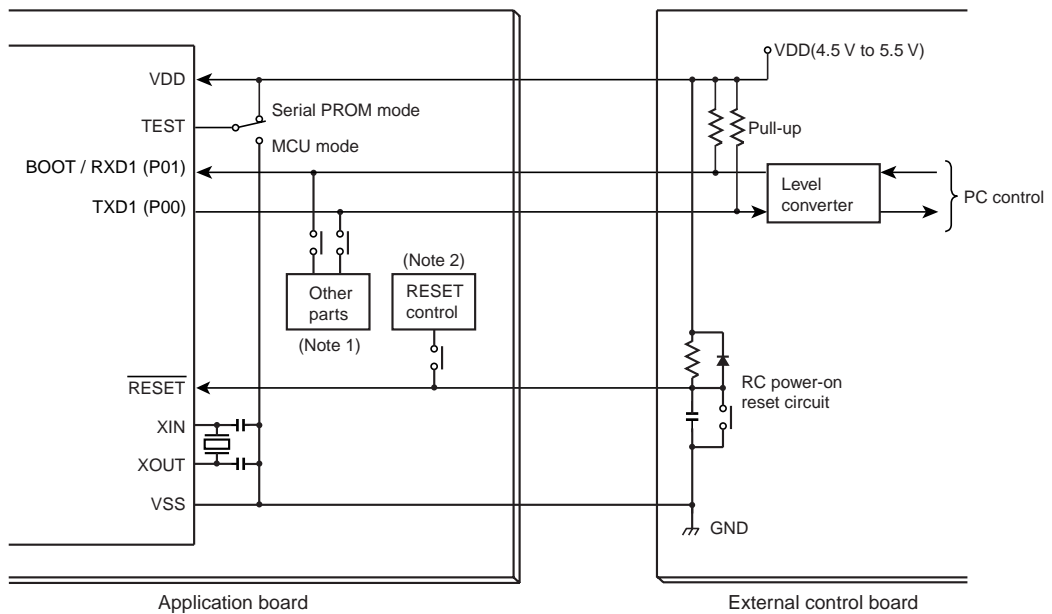


Figure 19-3 Example Connection for On-Board Writing

Note 1: When other parts on the application board effect the UART communication in the serial PROM mode, isolate these pins by a jumper or switch.

Note 2: When the reset control circuit on the application board effects activation of the serial PROM mode, isolate the pin by a jumper or switch.

Note 3: For connection of other pins, refer to "Table 19-3 Pin Function in the Serial PROM Mode".

### 19.3.4 Activating the Serial PROM Mode

The following is a procedure to activate the serial PROM mode. "Figure 19-4 Serial PROM Mode Timing" shows a serial PROM mode timing.

1. Supply power to the VDD pin.
2. Set the  $\overline{\text{RESET}}$  pin to low.
3. Set the TEST pin and BOOT/RXD1 pins to high.
4. Wait until the power supply and clock oscillation stabilize.
5. Set the  $\overline{\text{RESET}}$  pin to high.
6. Input the matching data (5AH) to the BOOT/RXD1 pin after setup sequence. For details of the setup timing, refer to "19.16 UART Timing".

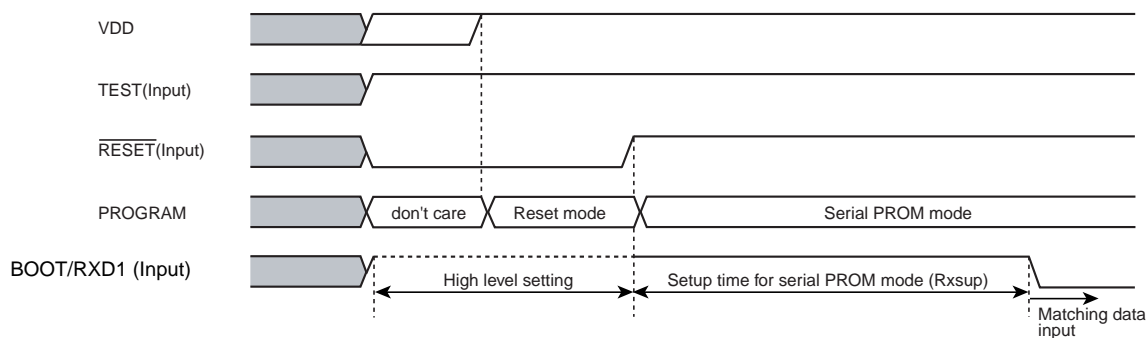


Figure 19-4 Serial PROM Mode Timing

## 19.4 Interface Specifications for UART

The following shows the UART communication format used in the serial PROM mode.

To perform on-board programming, the communication format of the write controller must also be set in the same manner.

The default baud rate is 9600 bps regardless of operating frequency of the microcontroller. The baud rate can be modified by transmitting the baud rate modification data shown in Table 19-4 to TMP86FH92DMG. The Table 19-5 shows an operating frequency and baud rate. The frequencies which are not described in Table 19-5 can not be used.

- Baud rate (Default): 9600 bps
- Data length: 8 bits
- Parity addition: None
- Stop bit: 1 bit

Table 19-4 Baud Rate Modification Data

Baud rate modification data	04H	05H	06H	07H	0AH	18H	28H
Baud rate (bps)	76800	62500	57600	38400	31250	19200	9600

Table 19-5 Operating Frequency and Baud Rate in the Serial PROM Mode

(Note 3)	Reference Baud Rate (bps)		76800		62500		57600		38400		31250		19200		9600	
	Baud Rate Modification Data		04H		05H		06H		07H		0AH		18H		28H	
	Ref. Frequency (MHz)	Rating (MHz)	Baud rate (bps)	(%)	(bps)	(%)	(bps)	(%)	(bps)	(%)	(bps)	(%)	(bps)	(%)	(bps)	(%)
1	2	1.91 to 2.10	-	-	-	-	-	-	-	-	-	-	-	-	9615	+0.16
2	4	3.82 to 4.19	-	-	-	-	-	-	-	-	31250	0.00	19231	+0.16	9615	+0.16
	4.19	3.82 to 4.19	-	-	-	-	-	-	-	-	32734	+4.75	20144	+4.92	10072	+4.92
3	4.9152	4.70 to 5.16	-	-	-	-	-	-	38400	0.00	-	-	19200	0.00	9600	0.00
	5	4.70 to 5.16	-	-	-	-	-	-	39063	+1.73	-	-	19531	+1.73	9766	+1.73
4	6	5.87 to 6.45	-	-	-	-	-	-	-	-	-	-	-	-	9375	-2.34
	6.144	5.87 to 6.45	-	-	-	-	-	-	-	-	-	-	-	-	9600	0.00
5	7.3728	7.05 to 7.74	-	-	-	-	57600	0.00	-	-	-	-	19200	0.00	9600	0.00
6	8	7.64 to 8.39	-	-	62500	0.00	-	-	38462	+0.16	31250	0.00	19231	+0.16	9615	+0.16
7	9.8304	9.40 to 10.32	76800	0.00	-	-	-	-	38400	0.00	-	-	19200	0.00	9600	0.00
	10	9.40 to 10.32	78125	+1.73	-	-	-	-	39063	+1.73	-	-	19531	+1.73	9766	+1.73
8	12	11.75 to 12.90	-	-	-	-	57692	+0.16	-	-	31250	0.00	18750	-2.34	9375	-2.34
	12.288	11.75 to 12.90	-	-	-	-	59077	+2.56	-	-	32000	+2.40	19200	0.00	9600	0.00
	12.5	11.75 to 12.90	-	-	60096	-3.85	60096	+4.33	-	-	30048	-3.85	19531	+1.73	9766	+1.73
9	14.7456	14.10 to 15.48	-	-	-	-	57600	0.00	38400	0.00	-	-	19200	0.00	9600	0.00
10	16	15.27 to 16.77	76923	+0.16	62500	0.00	-	-	38462	+0.16	31250	0.00	19231	+0.16	9615	+0.16

Note 1: "Ref. Frequency" and "Rating" show frequencies available in the serial PROM mode. Though the frequency is supported in the serial PROM mode, the serial PROM mode may not be activated correctly due to the frequency difference in the external controller (such as personal computer) and oscillator, and load capacitance of communication pins.

Note 2: It is recommended that the total frequency difference is within  $\pm 3\%$  so that auto detection is performed correctly by the reference frequency.

Note 3: The external controller must transmit the matching data (5AH) repeatedly till the auto detection of baud rate is performed. This number indicates the number of times the matching data is transmitted for each frequency.

## 19.5 Operation Command

The eight commands shown in Table 19-6 are used in the serial PROM mode. After reset release, the TMP86FH92DMG waits for the matching data (5AH).

Table 19-6 Operation Command in the Serial PROM Mode

Command Data	Operating Mode	Description
5AH	Setup	Matching data. Execute this command after releasing the reset.
F0H	Flash memory erasing	Erases the flash memory area (address C000H to FFFFH).
30H	Flash memory writing	Writes to the flash memory area (address C000H to FFFFH).
60H	RAM loader	Writes to the specified RAM area (address 0050H to 023FH).
90H	Flash memory SUM output	Outputs the 2-byte checksum upper byte and lower byte in this order for the entire area of the flash memory (address C000H to FFFFH).
C0H	Product ID code output	Outputs the product ID code (13-byte data).
C3H	Flash memory status output	Outputs the status code (7-byte data) such as the security program condition.
FAH	Flash memory security program setting	Enables the security program.

## 19.6 Operation Mode

The serial PROM mode has seven types of modes, that are (1) Flash memory erasing, (2) Flash memory writing, (3) RAM loader, (4) Flash memory SUM output, (5) Product ID code output, (6) Flash memory status output and (7) Flash memory security program setting modes. Description of each mode is shown below.

### 1. Flash memory erasing mode

The flash memory is erased by the chip erase (erasing an entire flash area) or sector erase (erasing sectors in 4-kbyte units). The erased area is filled with FFH. When the security program is enabled, the sector erase in the flash erasing mode can not be performed. To disable the security program, perform the chip erase. Before erasing the flash memory, TMP86FH92DMG checks the passwords except a blank product. If the password is not matched, the flash memory erasing mode is not activated.

### 2. Flash memory writing mode

Data is written to the specified flash memory address for each byte unit. The external controller must transmit the write data in the Intel Hex format (Binary). If no error is encountered till the end record, TMP86FH92DMG calculates the checksum for the entire flash memory area (C000H to FFFFH), and returns the obtained result to the external controller. When the security program is enabled, the flash memory writing mode is not activated. In this case, perform the chip erase command beforehand in the flash memory erasing mode. Before activating the flash memory writing mode, TMP86FH92DMG checks the password except a blank product. If the password is not matched, flash memory writing mode is not activated.

### 3. RAM loader mode

The RAM loader transfers the data in Intel Hex format sent from the external controller to the internal RAM. When the transfer is completed normally, the RAM loader calculates the checksum. After transmitting the results, the RAM loader jumps to the RAM address specified with the first data record in order to execute the user program. When the security program is enabled, the RAM loader mode is not activated. In this case, perform the chip erase beforehand in the flash memory erasing mode. Before activating the RAM loader mode, TMP86FH92DMG checks the password except a blank product. If the password is not matched, flash RAM loader mode is not activated.

### 4. Flash memory SUM output mode

The checksum is calculated for the entire flash memory area (C000H to FFFFH), and the result is returned to the external controller. Since the BOOTROM does not support the operation command to read the flash memory, use this checksum to identify programs when managing revisions of application programs.

### 5. Product ID code output

The code used to identify the product is output. The code to be output consists of 13-byte data, which includes the information indicating the area of the ROM incorporated in the product. The external controller reads this code, and recognizes the product to write.

(In the case of TMP86FH92DMG, the addresses from C000H to FFFFH become the ROM area.)

6. Flash memory status output mode

The status of the area from FFE0H to FFFFH, and the security program condition are output as 7-byte code. The external controller reads this code to recognize the flash memory status.

7. Flash memory security program setting mode

This mode disables reading the flash memory data in parallel PROM mode. In the serial PROM mode, the flash memory writing and RAM loader modes are disabled. To disable the flash memory security program, perform the chip erase in the flash memory erasing mode.

### 19.6.1 Flash Memory Erasing Mode (Operating command: F0H)

Table 19-7 shows the flash memory erasing mode.

Table 19-7 Flash Memory Erasing Mode

	Transfer Byte	Transfer Data from the External Controller to TMP86FH92DMG	Baud Rate	Transfer Data from TMP86FH92DMG to the External Controller
BOOT ROM	1st byte 2nd byte	Matching data (5AH) -	9600 bps 9600 bps	- (Automatic baud rate adjustment) OK: Echo back data (5AH) Error: No data transmitted
	3rd byte 4th byte	Baud rate change data (Table 19-4) -	9600 bps 9600 bps	- OK: Echo back data Error: A1H × 3, A3H × 3, 62H × 3 (Note 1)
	5th byte 6th byte	Operation command data (F0H) -	Modified baud rate Modified baud rate	- OK: Echo back data (F0H) Error: A1H × 3, A3H × 3, 63H × 3 (Note 1)
	7th byte 8th byte	Password count storage address bit 15 to 08 (Note 4, 5)	Modified baud rate Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	9th byte 10th byte	Password count storage address bit 07 to 00 (Note 4, 5)	Modified baud rate Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	11th byte 12th byte	Password comparison start address bit 15 to 08 (Note 4, 5)	Modified baud rate Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	13th byte 14th byte	Password comparison start address bit 07 to 00 (Note 4, 5)	Modified baud rate Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	15th byte : m'th byte	Password string (Note 4, 5) -	Modified baud rate Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	n'th - 2 byte	Erase area specification (Note 2)	Modified baud rate	-
	n'th - 1 byte	-	Modified baud rate	OK: Checksum (Upper byte) (Note 3) Error: Nothing transmitted
	n'th byte	-	Modified baud rate	OK: Checksum (Lower byte) (Note 3) Error: Nothing transmitted
	n'th + 1 byte	(Wait for the next operation command data)	Modified baud rate	-

Note 1: "xxH × 3" indicates that the device enters the halt condition after transmitting 3 bytes of xxh.

Note 2: Refer to "19.13 Specifying the Erasure Area".

Note 3: Refer to "19.8 Checksum (SUM)".

Note 4: Refer to "19.10 Passwords".

Note 5: Do not transmit the password string for a blank product.

Note 6: When a password error occurs, TMP86FH92DMG stops UART communication and enters the halt mode. Therefore, when a password error occurs, initialize TMP86FH92DMG by the RESET pin and reactivate the serial PROM mode.

Note 7: If an error occurs during transfer of a password address or a password string, TMP86FH92DMG stops UART communication and enters the halt condition. Therefore, when a password error occurs, initialize TMP86FH92DMG by the RESET pin and reactivate the serial PROM mode.

#### Description of the flash memory erasing mode

1. The 1st through 4th bytes of the transmitted and received data contain the same data as in the flash memory writing mode.
2. The 5th byte of the received data contains the command data in the flash memory erasing mode (F0H).
3. When the 5th byte of the received data contains the operation command data shown in Table 19-6, the device echoes back the value which is the same data in the 6th byte position of the received data (in this case, F0H). If the 5th byte of the received data does not contain the operation command data, the device enters the halt condition after sending 3 bytes of the operation command error code (63H).
4. The 7th through m'th bytes of the transmitted and received data contain the same data as in the flash memory writing mode. In the case of a blank product, do not transmit a password string. (Do not transmit a dummy password string.)
5. The n'th - 2 byte contains the erasure area specification data. The upper 4 bits and lower 4 bits specify the start address and end address of the erasure area, respectively. For the detailed description, see "19.13 Specifying the Erasure Area".
6. The n'th - 1 byte and n'th byte contain the upper and lower bytes of the checksum, respectively. For how to calculate the checksum, refer to "19.8 Checksum (SUM)". Checksum is calculated unless a receiving error or Intel Hex format error occurs. After sending the end record, the external controller judges whether the transmission is completed correctly by receiving the checksum sent by the device.
7. After sending the checksum, the device waits for the next operation command data.

## 19.6.2 Flash Memory Writing Mode (Operation command: 30H)

Table 19-8 shows flash memory writing mode process.

Table 19-8 Flash Memory Writing Mode Process

	Transfer Byte	Transfer Data from External Controller to TMP86FH92DMG	Baud Rate	Transfer Data from TMP86FH92DMG to External Controller
BOOT ROM	1st byte 2nd byte	Matching data (5Ah) -	9600 bps 9600 bps	- (Automatic baud rate adjustment) OK: Echo back data (5AH) Error: Nothing transmitted
	3rd byte 4th byte	Baud rate modification data (See Table 19-4) -	9600 bps 9600 bps	- OK: Echo back data Error: A1H × 3, A3H × 3, 62H × 3 (Note 1)
	5th byte 6th byte	Operation command data (30H) -	Modified baud rate Modified baud rate	- OK: Echo back data (30H) Error: A1H × 3, A3H × 3, 63H × 3 (Note 1)
	7th byte 8th byte	Password count storage address bit 15 to 08 (Note 4)	Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	9th byte 10th byte	Password count storage address bit 07 to 00 (Note 4)	Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	11th byte 12th byte	Password comparison start address bit 15 to 08 (Note 4)	Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	13th byte 14th byte	Password comparison start address bit 07 to 00 (Note 4)	Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	15th byte : m'th byte	Password string (Note 5) -	Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	m'th + 1 byte : n'th - 2 byte	Intel Hex format (binary) (Note 2)	Modified baud rate	- -
	n'th - 1 byte	-	Modified baud rate	OK: SUM (Upper byte) (Note 3) Error: Nothing transmitted
	n'th byte	-	Modified baud rate	OK: SUM (Lower byte) (Note 3) Error: Nothing transmitted
	n'th + 1 byte	(Wait state for the next operation command data)	Modified baud rate	-

Note 1: "xxH × 3" indicates that the device enters the halt condition after sending 3 bytes of xxH. For details, refer to "19.7 Error Code".

Note 2: Refer to "19.9 Intel Hex Format (Binary)".

Note 3: Refer to "19.8 Checksum (SUM)".

Note 4: Refer to "19.10 Passwords".

Note 5: If addresses from FFE0H to FFFFH are filled with "FFH", the passwords are not compared because the device is considered as a blank product. Transmitting a password string is not required. Even in the case of a blank product, it is required to specify the password count storage address and the password comparison start address. Transmit these data from the external controller. If a password error occurs due to incorrect password count storage address or password comparison start address, TMP86FH92DMG stops UART communication and enters the halt condition. Therefore, when a password error occurs, initialize TMP86FH92DMG by the RESET pin and reactivate the serial ROM mode.

Note 6: If the security program is enabled or a password error occurs, TMP86FH92DMG stops UART communication and enters the halt condition. In this case, initialize TMP86FH92DMG by the RESET pin and reactivate the serial ROM mode.



- Note 7: If an error occurs during the reception of a password address or a password string, TMP86FH92DMG stops UART communication and enters the halt condition. In this case, initialize TMP86FH92DMG by the RESET pin and reactivate the serial PROM mode.
- Note 8: Do not write only the address from FFE0H to FFFFH when all flash memory data is the same. If only these area are written, the subsequent operation can not be executed due to password error.
- Note 9: To rewrite data to Flash memory addresses at which data (including FFH) is already written, make sure to erase the existing data by "sector erase" or "chip erase" before rewriting data.

#### Description of the flash memory writing mode

1. The 1st byte of the received data contains the matching data. When the serial PROM mode is activated, TMP86FH92DMG (hereafter called device), waits to receive the matching data (5AH). Upon reception of the matching data, the device automatically adjusts the UART's initial baud rate to 9600 bps.
2. When receiving the matching data (5AH), the device transmits an echo back data (5AH) as the second byte data to the external controller. If the device can not recognize the matching data, it does not transmit the echo back data and waits for the matching data again with automatic baud rate adjustment. Therefore, the external controller should transmit the matching data repeatedly till the device transmits an echo back data. The transmission repetition count varies depending on the frequency of device. For details, refer to Table 19-5.
3. The 3rd byte of the received data contains the baud rate modification data. The five types of baud rate modification data shown in Table 19-4 are available. Even if baud rate is not modified, the external controller should transmit the initial baud rate data (28H: 9600 bps).
4. Only when the 3rd byte of the received data contains the baud rate modification data corresponding to the device's operating frequency, the device echoes back data the value which is the same data in the 4th byte position of the received data. After the echo back data is transmitted, baud rate modification becomes effective. If the 3rd byte of the received data does not contain the baud rate modification data, the device enters the halts condition after sending 3 bytes of baud rate modification error code (62H).
5. The 5th byte of the received data contains the command data (30H) to write the flash memory.
6. When the 5th byte of the received data contains the operation command data shown in Table 19-6, the device echoes back the value which is the same data in the 6th byte position of the received data (in this case, 30H). If the 5th byte of the received data does not contain the operation command data, the device enters the halt condition after sending 3 bytes of the operation command error code (63H).
7. The 7th byte contains the data for 15 to 8 bits of the password count storage address. When the data received with the 7th byte has no receiving error, the device does not send any data. If a receiving error or password error occurs, the device does not send any data and enters the halt condition.
8. The 9th byte contains the data for 7 to 0 bits of the password count storage address. When the data received with the 9th byte has no receiving error, the device does not send any data. If a receiving error or password error occurs, the device does not send any data and enters the halt condition.
9. The 11th byte contains the data for 15 to 8 bits of the password comparison start address. When the data received with the 11th byte has no receiving error, the device does not send any data. If a receiving error or password error occurs, the device does not send any data and enters the halt condition.
10. The 13th byte contains the data for 7 to 0 bits of the password comparison start address. When the data received with the 13th byte has no receiving error, the device does not send any data. If a receiving error or password error occurs, the device does not send any data and enters the halt condition.
11. The 15th through m'th bytes contain the password data. The number of passwords becomes the data (N) stored in the password count storage address. The external password data is compared with N-byte data from the address specified by the password comparison start address. The external controller should send N-byte password data to the device. If the passwords do not match, the device enters the halt condition without returning an error code to the external controller. If the addresses from FFE0H to FFFFH are filled with "FFH", the passwords are not compared because the device is considered as a blank product.
12. The m'th + 1 through n'th - 2 bytes of the received data contain the binary data in the Intel Hex format. No received data is echoed back to the external controller. After receiving the start mark (3AH for ":"), in the Intel Hex format, the device starts data record reception. Therefore, the received data except 3AH is ignored until the start mark is received. After receiving the start mark, the device receives the data record, that consists of data length, address, record type, write data and checksum. Since the device

starts checksum calculation after receiving an end record, the external controller should wait for the checksum after sending the end record. If a receiving error or Intel Hex format error occurs, the device enters the halts condition without returning an error code to the external controller.

13. The  $n$ 'th - 1 and  $n$ 'th bytes contain the checksum upper and lower bytes. For details on how to calculate the SUM, refer to "19.8 Checksum (SUM)". The checksum is calculated only when the end record is detected and no receiving error or Intel Hex format error occurs. After sending the end record, the external controller judges whether the transmission is completed correctly by receiving the checksum sent by the device.
14. After transmitting the checksum, the device waits for the next operation command data.

19.6.3 RAM Loader Mode (Operation Command: 60H)

Table 19-9 shows RAM loader mode process.

Table 19-9 RAM Loader Mode Process

	Transfer Bytes	Transfer Data from External Controller to TMP86FH92DMG	Baud Rate	Transfer Data from TMP86FH92DMG to External Controller	
BOOT ROM	1st byte 2nd byte	Matching data (5AH) -	9600 bps 9600 bps	- (Automatic baud rate adjustment) OK: Echo back data (5AH) Error: Nothing transmitted	
	3rd byte 4th byte	Baud rate modification data (See Table 19-4) -	9600 bps 9600 bps	- OK: Echo back data Error: A1H × 3, A3H × 3, 62H × 3 (Note 1)	
	5th byte 6th byte	Operation command data (60H) -	Modified baud rate Modified baud rate	- OK: Echo back data (60H) Error: A1H × 3, A3H × 3, 63H × 3 (Note 1)	
	7th byte 8th byte	Password count storage address bit 15 to 08 (Note 4)	Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted	
	9th byte 10th byte	Password count storage address bit 07 to 00 (Note 4)	Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted	
	11th byte 12th byte	Password comparison start address bit 15 to 08 (Note 4)	Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted	
	13th byte 14th byte	Password comparison start address bit 07 to 00 (Note 4)	Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted	
	15th byte : m'th byte	Password string (Note 5) -	Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted	
	m'th + 1 byte : n'th - 2 byte	Intel Hex format (Binary) (Note 2)	Modified baud rate Modified baud rate	- -	
	n'th - 1 byte	-	Modified baud rate	OK: SUM (Upper byte) (Note 3) Error: Nothing transmitted	
	n'th byte	-	Modified baud rate	OK: SUM (Lower byte) (Note 3) Error: Nothing transmitted	
	RAM	-	The program jumps to the start address of RAM in which the first transferred data is written.		

Note 1: "xxH × 3" indicates that the device enters the halt condition after sending 3 bytes of xxH. For details, refer to "19.7 Error Code".

Note 2: Refer to "19.9 Intel Hex Format (Binary)".

Note 3: Refer to "19.8 Checksum (SUM)".

Note 4: Refer to "19.10 Passwords".

Note 5: If addresses from FFE0H to FFFFH are filled with "FFH", the passwords are not compared because the device is considered as a blank product. Transmitting a password string is not required. Even in the case of a blank product, it is required to specify the password count storage address and the password comparison start address. Transmit these data from the external controller. If a password error occurs due to incorrect password count storage address or password comparison start address, TMP86FH92DMG stops UART communication and enters the halt condition. Therefore, when a password error occurs, initialize TMP86FH92DMG by the RESET pin and reactivate the serial ROM mode.

Note 6: After transmitting a password string, the external controller must not transmit only an end record. If receiving an end record after a password string, the device may not operate correctly.

- Note 7: If the security program is enabled or a password error occurs, TMP86FH92DMG stops UART communication and enters the halt condition. In this case, initialize TMP86FH92DMG by the RESET pin and reactivate the serial PROM mode.
- Note 8: If an error occurs during the reception of a password address or a password string, TMP86FH92DMG stops UART communication and enters the halt condition. In this case, initialize TMP86FH92DMG by the RESET pin and reactivate the serial PROM mode.
- Note 9: To rewrite data to Flash memory addresses at which data (including FFH) is already written, make sure to erase the existing data by "sector erase" or "chip erase" before rewriting data.

#### Description of RAM loader mode

1. The 1st through 4th bytes of the transmitted and received data contains the same data as in the flash memory writing mode.
2. In the 5th byte of the received data contains the RAM loader command data (60H).
3. When the 5th byte of the received data contains the operation command data shown in Table 19-6, the device echoes back the value which is the same data in the 6th byte position (in this case, 60H). If the 5th byte does not contain the operation command data, the device enters the halt condition after sending 3 bytes of operation command error code (63H).
4. The 7th through m'th bytes of the transmitted and received data contain the same data as in the flash memory writing mode.
5. The m'th + 1 through n'th - 2 bytes of the received data contain the binary data in the Intel Hex format. No received data is echoed back to the external controller. After receiving the start mark (3AH for ":"), in the Intel Hex format, the device starts data record reception. Therefore, the received data except 3AH is ignored until the start mark is received. After receiving the start mark, the device receives the data record, that consists of data length, address, record type, write data and checksum. The writing data of the data record is written into RAM specified by address. Since the device starts checksum calculation after receiving an end record, the external controller should wait for the checksum after sending the end record. If a receiving error or Intel Hex format error occurs, the device enters the halts condition without returning an error code to the external controller.
6. The n'th - 1 and n'th bytes contain the checksum upper and lower bytes. For details on how to calculate the SUM, refer to "19.8 Checksum (SUM)". The checksum is calculated only when the end record is detected and no receiving error or Intel Hex format error occurs. After sending the end record, the external controller judges whether the transmission is completed correctly by receiving the checksum sent by the device.
7. After transmitting the checksum to the external controller, the boot program jumps to the RAM address that is specified by the first received data record.

### 19.6.4 Flash Memory SUM Output Mode (Operation Command: 90H)

Table 19-10 shows flash memory SUM output mode process.

Table 19-10 Flash Memory SUM Output Process

	Transfer Bytes	Transfer Data from External Controller to TMP86FH92DMG	Baud Rate	Transfer Data from TMP86FH92DMG to External Controller
BOOT ROM	1st byte	Matching data (5AH)	9600 bps	- (Automatic baud rate adjustment)
	2nd byte	-	9600 bps	OK: Echo back data (5AH) Error: Nothing transmitted
	3rd byte	Baud rate modification data (See Table 19-4)	9600 bps	-
	4th byte	-	9600 bps	OK: Echo back data Error: A1H × 3, A3H × 3, 62H × 3 (Note 1)
	5th byte	Operation command data (90H)	Modified baud rate	-
	6th byte	-	Modified baud rate	OK: Echo back data (90H) Error: A1H × 3, A3H × 3, 63H × 3 (Note 1)
	7th byte	-	Modified baud rate	OK: SUM (Upper byte) (Note 2) Error: Nothing transmitted
	8th byte	-	Modified baud rate	OK: SUM (Lower byte) (Note 2) Error: Nothing transmitted
	9th byte	(Wait for the next operation command data)	Modified baud rate	-

Note 1: "xxH × 3" indicates that the device enters the halt condition after sending 3 bytes of xxH. For details, refer to "19.7 Error Code".

Note 2: Refer to "19.8 Checksum (SUM)".

#### Description of the flash memory SUM output mode

1. The 1st through 4th bytes of the transmitted and received data contains the same data as in the flash memory writing mode.
2. The 5th byte of the received data contains the command data in the flash memory SUM output mode (90H).
3. When the 5th byte of the received data contains the operation command data shown in Table 19-6, the device echoes back the value which is the same data in the 6th byte position of the received data (in this case, 90H). If the 5th byte of the received data does not contain the operation command data, the device enters the halt condition after transmitting 3 bytes of operation command error code (63H).
4. The 7th and the 8th bytes contain the upper and lower bits of the checksum, respectively. For how to calculate the checksum, refer to "19.8 Checksum (SUM)".
5. After sending the checksum, the device waits for the next operation command data.

### 19.6.5 Product ID Code Output Mode (Operation Command: C0H)

Table 19-11 shows product ID code output mode process.

Table 19-11 Product ID Code Output Process

	Transfer Bytes	Transfer Data from External Controller to TMP86FH92DMG	Baud Rate	Transfer Data from TMP86FH92DMG to External Controller	
BOOT ROM	1st byte	Matching data (5AH)	9600 bps	- (Automatic baud rate adjustment)	
	2nd byte	-	9600 bps	OK: Echo back data (5AH) Error: Nothing transmitted	
	3rd byte	Baud rate modification data (See Table 19-4)	9600 bps	-	
	4th byte	-	9600 bps	OK: Echo back data Error: A1H × 3, A3H × 3, 62H × 3 (Note 1)	
	5th byte	Operation command data (C0H)	Modified baud rate	-	
	6th byte	-	Modified baud rate	OK: Echo back data (C0H) Error: A1H × 3, A3H × 3, 63H × 3 (Note 1)	
	7th byte		Modified baud rate	3AH	Start mark
	8th byte		Modified baud rate	0AH	The number of transfer data (from 9th to 18th bytes)
	9th byte		Modified baud rate	02H	Length of address (2 bytes)
	10th byte		Modified baud rate	1DH	Reserved data
	11th byte		Modified baud rate	00H	Reserved data
	12th byte		Modified baud rate	00H	Reserved data
	13th byte		Modified baud rate	00H	Reserved data
	14th byte		Modified baud rate	01H	ROM block count (1 block)
	15th byte		Modified baud rate	C0H	First address of ROM (Upper byte)
	16th byte		Modified baud rate	00H	First address of ROM (Lower byte)
	17th byte		Modified baud rate	FFH	End address of ROM (Upper byte)
	18th byte		Modified baud rate	FFH	End address of ROM (Lower byte)
	19th byte		Modified baud rate	22H	Checksum of transferred data (9th through 18th byte)
	20th byte	(Wait for the next operation command data)	Modified baud rate	-	

Note: "xxH × 3" indicates that the device enters the halt condition after sending 3 bytes of xxH. For details, refer to "19.7 Error Code".

#### Description of Product ID code output mode

1. The 1st through 4th bytes of the transmitted and received data contain the same data as in the flash memory writing mode.
2. The 5th byte of the received data contains the product ID code output mode command data (C0H).
3. When the 5th byte contains the operation command data shown in Table 19-6, the device echoes back the value which is the same data in the 6th byte position of the received data (in this case, C0H). If the 5th byte data does not contain the operation command data, the device enters the halt condition after sending 3 bytes of operation command error code (63H).
4. The 9th through 19th bytes contain the product ID code. For details, refer to "19.11 Product ID Code".

5. After sending the checksum, the device waits for the next operation command data.

## 19.6.6 Flash Memory Status Output Mode (Operation Command: C3H)

Table 19-12 shows Flash memory status output mode process.

Table 19-12 Flash Memory Status Output Mode Process

	Transfer Bytes	Transfer Data from External Controller to TMP86FH92DMG	Baud Rate	Transfer Data from TMP86FH92DMG to External Controller
BOOT ROM	1st byte 2nd byte	Matching data (5AH) -	9600 bps 9600 bps	- (Automatic baud rate adjustment) OK: Echo back data (5AH) Error: Nothing transmitted
	3rd byte 4th byte	Baud rate modification data (See Table 19-4) -	9600 bps 9600 bps	- OK: Echo back data Error: A1H × 3, A3H × 3, 62H × 3 (Note 1)
	5th byte 6th byte	Operation command data (C3H) -	Modified baud rate Modified baud rate	- OK: Echo back data (C3H) Error: A1H × 3, A3H × 3, 63H × 3 (Note 1)
	7th byte		Modified baud rate	3AH Start mark
	8th byte		Modified baud rate	04H Byte count (from 9th to 12th byte)
	9th byte		Modified baud rate	00H to 03H Status code 1
	10th byte		Modified baud rate	00H Reserved data
	11th byte		Modified baud rate	00H Reserved data
	12th byte		Modified baud rate	00H Reserved data
	13th byte		Modified baud rate	Checksum 2's complement for the sum of 9th through 12th bytes 9th byte Checksum 00H: 00H 01H: FFH 02H: FEH 03H: FDH
	14th byte	(Wait for the next operation command data)	Modified baud rate	-

Note 1: "xxH × 3" indicates that the device enters the halt condition after sending 3 bytes of xxH. For details, refer to "19.7 Error Code".

Note 2: For the details on status code 1, refer to "19.12 Flash Memory Status Code".

### Description of Flash memory status output mode

1. The 1st through 4th bytes of the transmitted and received data contain the same data as in the Flash memory writing mode.
2. The 5th byte of the received data contains the flash memory status output mode command data (C3H).
3. When the 5th byte contains the operation command data shown in Table 19-6, the device echoes back the value which is the same data in the 6th byte position of the received data (in this case, C3H). If the 5th byte does not contain the operation command data, the device enters the halt condition after sending 3 bytes of operation command error code (63H).
4. The 9th through 13th bytes contain the status code. For details on the status code, refer to "19.12 Flash Memory Status Code".
5. After sending the status code, the device waits for the next operation command data.



## 19.6.7 Flash Memory security program Setting Mode (Operation Command: FAH)

Table 19-13 shows Flash memory security program setting mode process.

Table 19-13 Flash Memory security program Setting Mode Process

	Transfer Bytes	Transfer Data from External Controller to TMP86FH92DMG	Baud Rate	Transfer Data from TMP86FH92DMG to External Controller
BOOT ROM	1st byte 2nd byte	Matching data (5AH) -	9600 bps 9600 bps	- (Automatic baud rate adjustment) OK: Echo back data (5AH) Error: Nothing transmitted
	3rd byte 4th byte	Baud rate modification data (See Table 19-4) -	9600 bps 9600 bps	- OK: Echo back data Error: A1H × 3, A3H × 3, 62H × 3 (Note 1)
	5th byte 6th byte	Operation command data (FAH) -	Modified baud rate Modified baud rate	- OK: Echo back data (FAH) Error: A1H × 3, A3H × 3, 63H × 3 (Note 1)
	7th byte 8th byte	Password count storage address 15 to 08 (Note 2)	Modified baud rate Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	9th byte 10th byte	Password count storage address 07 to 00 (Note 2)	Modified baud rate Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	11th byte 12th byte	Password comparison start address 15 to 08 (Note 2)	Modified baud rate Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	13th byte 14th byte	Password comparison start address 07 to 00 (Note 2)	Modified baud rate Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	15th byte : m'th byte	Password string (Note 2) -	Modified baud rate Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	n'th byte	-	Modified baud rate	OK: FBH (Note 3) Error: Nothing transmitted
	n'+1th byte	(Wait for the next operation command data)	Modified baud rate	-

Note 1: "xxH × 3" indicates that the device enters the halt condition after sending 3 bytes of xxH. For details, refer to "19.7 Error Code".

Note 2: Refer to "19.10 Passwords".

Note 3: If the security program is enabled for a blank product or a password error occurs for a non-blank product, TMP86FH92DMG stops UART communication and enters the halt mode. In this case, initialize TMP86FH92DMG by the RESET pin and reactivate the serial PROM mode.

Note 4: If an error occurs during reception of a password address or a password string, TMP86FH92DMG stops UART communication and enters the halt mode. In this case, initialize TMP86FH92DMG by the RESET pin and reactivate the serial PROM mode.

### Description of the Flash memory security program setting mode

- The 1st through 4th bytes of the transmitted and received data contain the same data as in the Flash memory writing mode.
- The 5th byte of the received data contains the command data in the flash memory status output mode (FAH).

3. When the 5th byte of the received data contains the operation command data shown in Table 19-6, the device echoes back the value which is the same data in the 6th byte position of the received data (in this case, FAH). If the 5th byte does not contain the operation command data, the device enters the halt condition after transmitting 3 bytes of operation command error code (63H).
4. The 7th through m'th bytes of the transmitted and received data contain the same data as in the flash memory writing mode.
5. The n'th byte contains the status to be transmitted to the external controller in the case of the successful security program.

## 19.7 Error Code

When detecting an error, the device transmits the error code to the external controller, as shown in Table 19-14.

Table 19-14 Error Code

Transmit Data	Meaning of Error Data
62H, 62H, 62H	Baud rate modification error.
63H, 63H, 63H	Operation command error.
A1H, A1H, A1H	Framing error in the received data.
A3H, A3H, A3H	Overrun error in the received data.

Note: If a password error occurs, TMP86FH92DMG does not transmit an error code.

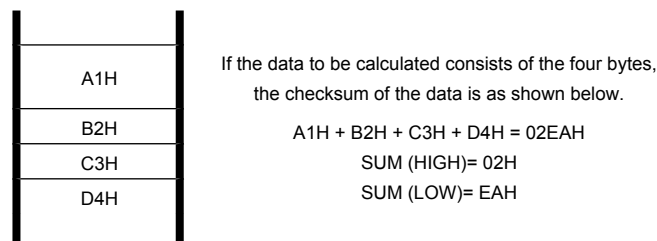
## 19.8 Checksum (SUM)

### 19.8.1 Calculation Method

The checksum (SUM) is calculated with the sum of all bytes, and the obtained result is returned as a word.

The data is read for each byte unit and the calculated result is returned as a word.

Example:



The checksum which is transmitted by executing the flash memory write command, RAM loader command, or flash memory SUM output command is calculated in the manner, as shown above.

## 19.8.2 Calculation data

The data used to calculate the checksum is listed in Table 19-15.

Table 19-15 Checksum Calculation Data

Operating Mode	Calculation Data	Description
Flash memory writing mode	Data in the entire area of the flash memory	Even when a part of the flash memory is written, the checksum of the entire flash memory area (C000H to FFFH) is calculated. The data length, address, record type and checksum in Intel Hex format are not included in the checksum.
Flash memory SUM output mode		
RAM loader mode	RAM data written in the first received RAM address through the last received RAM address	The length of data, address, record type and checksum in Intel Hex format are not included in the checksum.
Product ID Code output mode	9th through 18th bytes of the transferred data	For details, refer to "19.11 Product ID Code".
Flash Memory Status Output mode	9th through 12th bytes of the transferred data	For details, refer to "19.12 Flash Memory Status Code"
Flash Memory Erasing mode	All data in the erased area of the flash memory (the whole or part of the flash memory)	When the sector erase is executed, only the erased area is used to calculate the checksum. In the case of the chip erase, an entire area of the flash memory is used.

## 19.9 Intel Hex Format (Binary)

1. After receiving the checksum of a data record, the device waits for the start mark (3AH “:”) of the next data record. After receiving the checksum of a data record, the device ignores the data except 3AH transmitted by the external controller.
2. After transmitting the checksum of end record, the external controller must transmit nothing, and wait for the 2-byte receive data (upper and lower bytes of the checksum).
3. If a receiving error or Intel Hex format error occurs, the device enters the halt condition without returning an error code to the external controller. The Intel Hex format error occurs in the following case:

When the record type is not 00H, 01H, or 02H

When a checksum error occurs

When the data length of an extended record (record type = 02H) is not 02H

When the device receives the data record after receiving an extended record (record type = 02H) with extended address of 1000H or larger.

When the data length of the end record (record type = 01H) is not 00H

## 19.10 Passwords

The consecutive eight or more-byte data in the flash memory area can be specified to the password. TMP86FH92DMG compares the data string specified to the password with the password string transmitted from the external controller. The area in which passwords can be specified is located at addresses C000H to FF9FH. The area from FFA0H to FFFFH can not be specified as the passwords area.

If addresses from FFE0H through FFFFH are filled with “FFH”, the passwords are not compared because the product is considered as a blank product. Even in this case, the password count storage addresses and password comparison start address must be specified. Table 19-16 shows the password setting in the blank product and non-blank product.

Table 19-16 Password Setting in the Blank Product and Non-Blank Product

Password	Blank Product (Note 1)	Non-Blank Product
PNSA (Password count storage address)	$C000H \leq PNSA \leq FF9FH$	$C000H \leq PNSA \leq FF9FH$
PCSA (Password comparison start address)	$C000H \leq PCSA \leq FF9FH$	$C000H \leq PCSA \leq FFA0 - N$
N (Password count)	*	$8 \leq N$
Password string setting	Not required (Note 5)	Required (Note 2)

Note 1: When addresses from FFE0H through FFFFH are filled with “FFH”, the product is recognized as a blank product.

Note 2: The data including the same consecutive data (three or more bytes) can not be used as a password. (This causes a password error data. TMP86FH92DMG transmits no data and enters the halt condition.)

Note 3: \*: Don't care.

Note 4: When the above condition is not met, a password error occurs. If a password error occurs, the device enters the halt condition without returning the error code.

Note 5: In the flash memory writing mode or RAM loader mode, the blank product receives the Intel Hex format data immediately after receiving PCSA without receiving password strings. In this case, the subsequent processing is performed correctly because the blank product ignores the data except the start mark (3AH “:”) as the Intel Hex format data, even if the external controller transmits the dummy password string. However, if the dummy password string contains “3AH”, it is detected as the start mark erroneously. The microcontroller enters the halt mode. If this causes the problem, do not transmit the dummy password strings.

Note 6: In the flash memory erasing mode, the external controller must not transmit the password string for the blank product.

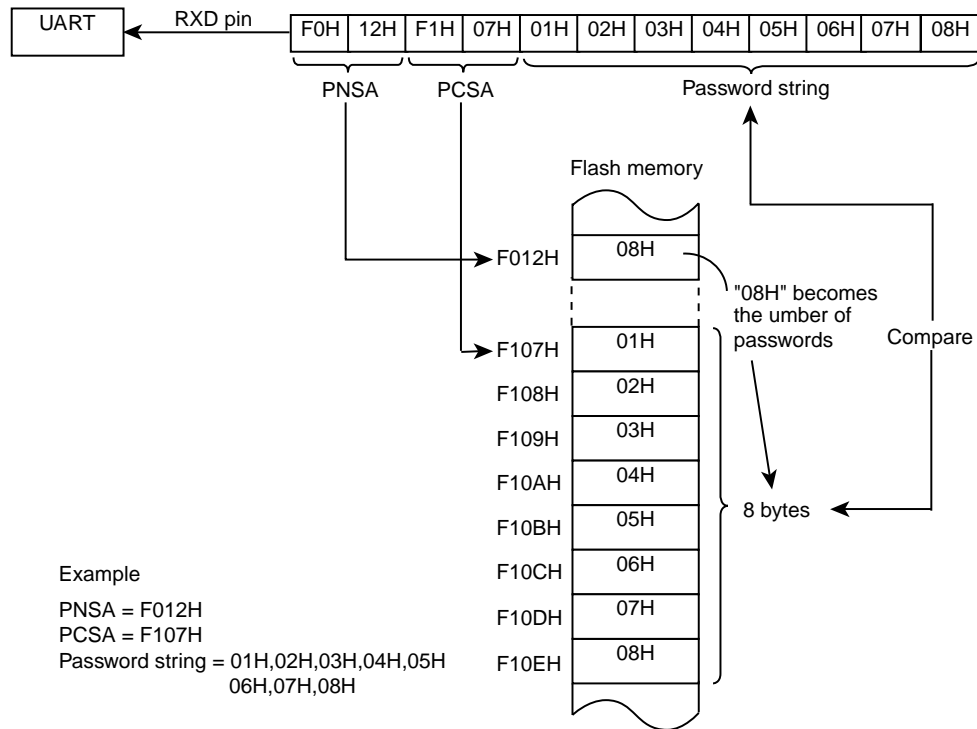


Figure 19-5 Password Comparison

### 19.10.1 Password String

The password string transmitted from the external controller is compared with the specified data in the flash memory. When the password string is not matched to the data in the flash memory, the device enters the halt condition due to the password error.

### 19.10.2 Handling of Password Error

If a password error occurs, the device enters the halt condition. In this case, reset the device to reactivate the serial PROM mode.

### 19.10.3 Password Management during Program Development

If a program is modified many times in the development stage, confusion may arise as to the password. Therefore, it is recommended to use a fixed password in the program development stage.

Example :Specify PNSA to F000H, and the password string to 8 bytes from address F001H (PCSA becomes F001H.)

```

Password Section code abs = 0F000H
    DB  08H          : PNSA definition
    DB  "CODE1234"  : Password string definition
    
```

## 19.11 Product ID Code

The product ID code is the 13-byte data containing the start address and the end address of ROM. Table 19-17 shows the product ID code format.

Table 19-17 Product ID Code Format

Data	Description	In the Case of TMP86FH92DMG
1st	Start Mark (3AH)	3AH
2nd	The number of transfer data (10 bytes from 3rd to 12th byte)	0AH
3rd	Address length (2 bytes)	02H
4th	Reserved data	1DH
5th	Reserved data	00H
6th	Reserved data	00H
7th	Reserved data	00H
8th	ROM block count	01H
9th	The first address of ROM (Upper byte)	C0H
10th	The first address of ROM (Lower byte)	00H
11th	The end address of ROM (Upper byte)	FFH
12th	The end address of ROM (Lower byte)	FFH
13th	Checksum of the transferred data (2's compliment for the sum of 3rd through 12th bytes)	22H

## 19.12 Flash Memory Status Code

The flash memory status code is the 7-byte data including the security program status and the status of the data from FFE0H to FFFFH. Table 19-18 shows the flash memory status code.

Table 19-18 Flash Memory Status Code

Data	Description	In the Case of TMP86FH92DMG	
1st	Start mark	3AH	
2nd	Transferred data count (3rd through 6th byte)	04H	
3rd	Status code	00H to 03H (See figure below)	
4th	Reserved data	00H	
5th	Reserved data	00H	
6th	Reserved data	00H	
7th	Checksum of the transferred data (2's compliment for the sum of 3rd through 6th data)	3rd byte	checksum
		00H	00H
		01H	FFH
		02H	FEH
		03H	FDH

## Status Code 1



RPENA	Flash memory security program status	0:	Security program is disabled.
		1:	Security program is enabled.
BLANK	The status from FFE0H to FFFFH.	0:	All data is FFH in the area from FFE0H to FFFFH.
		1:	The value except FFH is included in the area from FFE0H to FFFFH.

Some operation commands are limited by the flash memory status code 1. If the security program is enabled, flash memory writing mode command and RAM loader mode command can not be executed. Erase all flash memory before executing these command.

RPENA	BLANK	Flash Memory Writing Mode	RAM Loader Mode	Flash memory SUM Output Mode	Product ID Code Output Mode	Flash Memory Status Output Mode	Flash Memory Erasing Mode		Security program Setting Mode
							Chip Erase	Sector Erase	
0	0	O	O	O	O	O	O		×
0	1	Pass	Pass	O	O	O	Pass		Pass
1	0	×	×	O	O	O	O	×	×
1	1	×	×	O	O	O	Pass	×	Pass

Note: O: The command can be executed.

Pass: The command can be executed with a password.

×: The command can not be executed.

(After echoing the command back to the external controller, TMP86FH92DMG stops UART communication and enters the halt condition.)



### 19.13 Specifying the Erasure Area

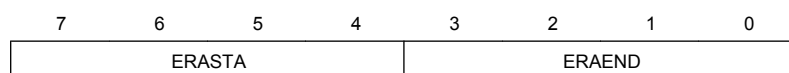
In the flash memory erasing mode, the erasure area of the flash memory is specified by n-2 byte data.

The start address of an erasure area is specified by ERASTA, and the end address is specified by ERAEND.

If ERASTA is equal to or smaller than ERAEND, the sector erase (erasure in 4 kbyte units) is executed. Executing the sector erase while the security program is enabled results in an infinite loop.

If ERASTA is larger than ERAEND, the chip erase (erasure of an entire flash memory area) is executed and the security program is disabled. Therefore, execute the chip erase (not sector erase) to disable the security program.

#### Erasure Area Specification Data (n-2 byte data)



ERASTA	The start address of the erasure area	0000: from 0000H 0001: from 1000H 0010: from 2000H 0011: from 3000H 0100: from 4000H 0101: from 5000H 0110: from 6000H 0111: from 7000H 1000: from 8000H 1001: from 9000H 1010: from A000H 1011: from B000H 1100: from C000H 1101: from D000H 1110: from E000H 1111: from F000H
ERAEND	The end address of the erasure area	0000: to 0FFFH 0001: to 1FFFH 0010: to 2FFFH 0011: to 3FFFH 0100: to 4FFFH 0101: to 5FFFH 0110: to 6FFFH 0111: to 7FFFH 1000: to 8FFFH 1001: to 9FFFH 1010: to AFFFH 1011: to BFFFH 1100: to CFFFH 1101: to DFFFH 1110: to EFFFH 1111: to FFFFH

Note: When the sector erase is executed for the area containing no flash cell, TMP86FH92DMG stops the UART communication and enters the halt condition.

### 19.14 Port Input Control Register

In the serial PROM mode, the input level is fixed to the all ports except P00 and P01 ports with a hardware feature to prevent overlap current to unused ports. (All port inputs and peripheral function inputs shared with the ports become invalid.) Therefore, to access to the flash memory in the RAM loader mode without UART communication, port inputs must be valid. To make port inputs valid, set the pin of the port input control register (SPCR) to “1”.

The SPCR register is not operated in the MCU mode.

## Port Input Control Register

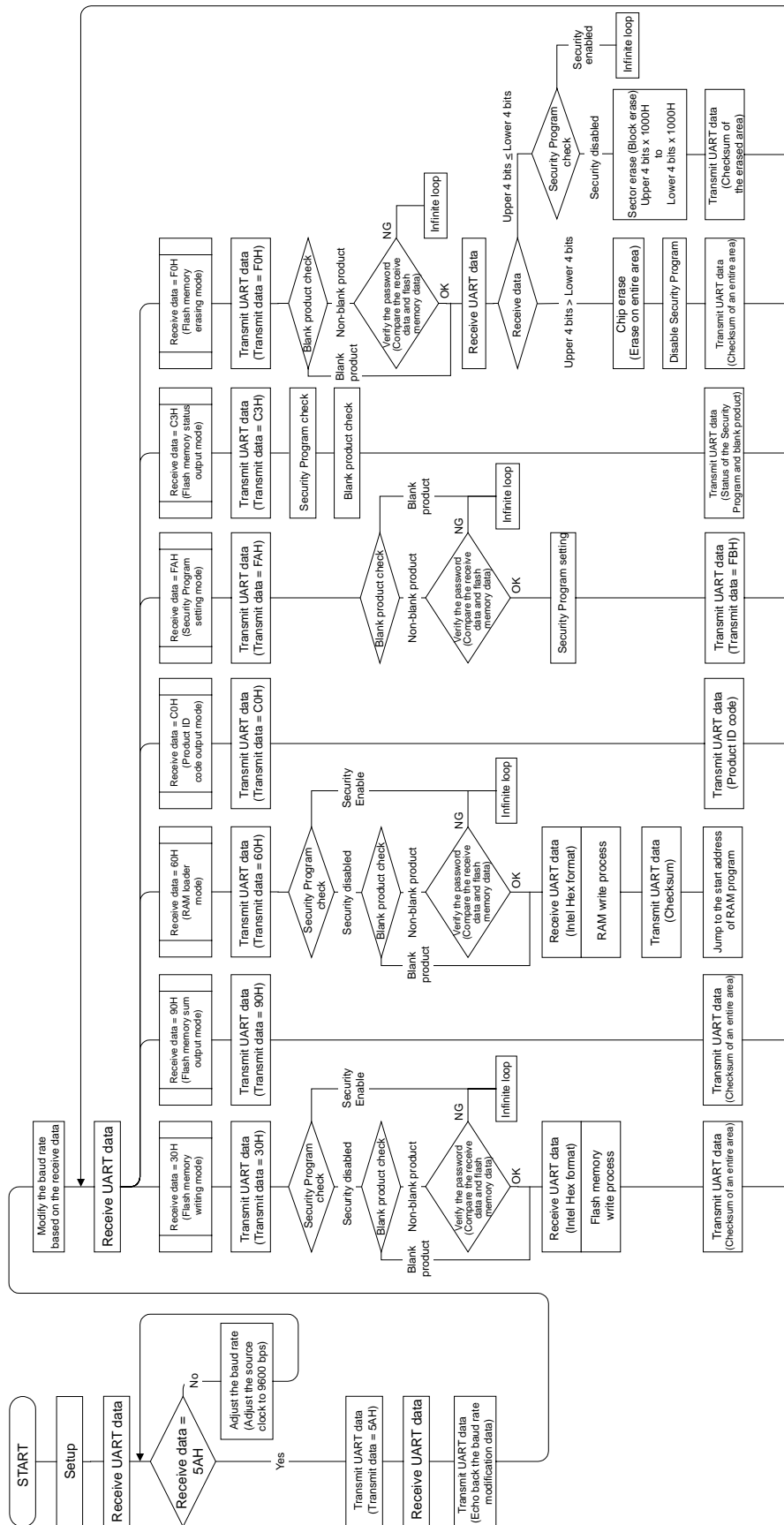


PIN	Port input control in the serial PROM mode	0: Invalid port inputs (The input level is fixed with a hardware feature.) 1: Valid port inputs	R/W
-----	--	--	-----

Note 1: The SPCR register can be read or written only in the serial PROM mode. When the write instruction is executed to the SPCR register in the MCU mode, the port input control can not be performed. When the read instruction is executed for the SPCR register in the MCU mode, read data of bit7 to 1 are unstable.

Note 2: All I/O ports except P00 and P01 ports are controlled by the SPCR register.

19.15 Flowchart



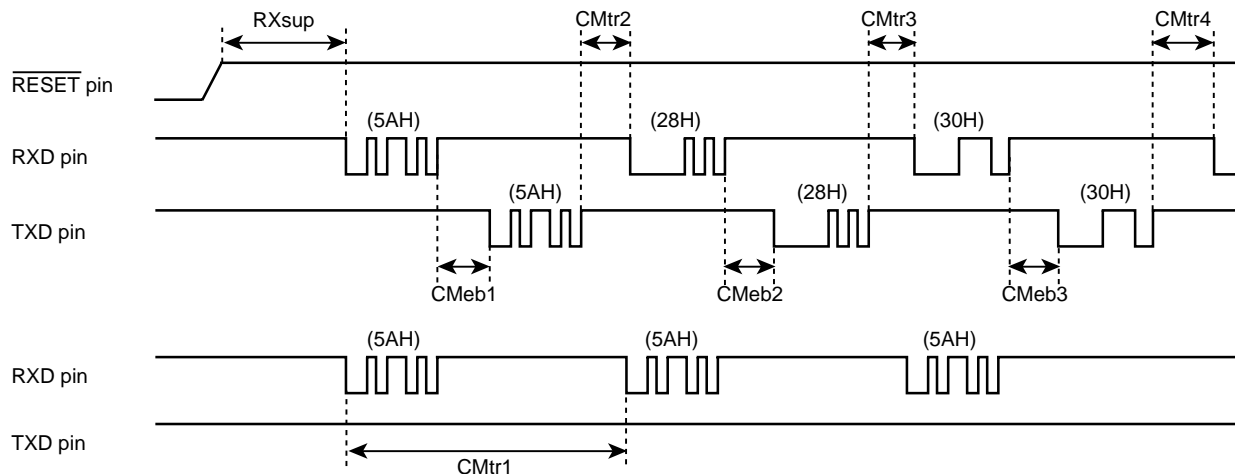
## 19.16 UART Timing

Table 19-19 UART Timing-1 (VDD = 4.5 to 5.5 V, fc = 2 to 16 MHz, Topr = -10 to 40°C)

Parameter	Symbol	Clock Frequency (fc)	Minimum Required Time	
			At fc = 2 MHz	At fc = 16 MHz
Time from matching data reception to the echo back	CMeb1	Approx. 930	465 μs	58.1 μs
Time from baud rate modification data reception to the echo back	CMeb2	Approx. 980	490 μs	61.3 μs
Time from operation command reception to the echo back	CMeb3	Approx. 800	400 μs	50 μs
Checksum calculation time	CKsm	Approx. 7864500	3.93 s	491.5 μs
Erase time of an entire flash memory	CEall	-	30 ms	30 ms
Erase time for a sector of a flash memory (in 4-kbyte units)	CEsec	-	15 ms	15 ms

Table 19-20 UART Timing-2 (VDD = 4.5 to 5.5 V, fc = 2 to 16 MHz, Topr = -10 to 40°C)

Parameter	Symbol	Clock Frequency (fc)	Minimum Required Time	
			At fc = 2 MHz	At fc = 16 MHz
Time from the reset release to the acceptance of start bit of RXD pin	RXsup	2100	1.05 ms	131.3 ms
Matching data transmission interval	CMtr1	28500	14.2 ms	1.78 ms
Time from the echo back of matching data to the acceptance of baud rate modification data	CMtr2	380	190 μs	23.8 μs
Time from the echo back of baud rate modification data to the acceptance of an operation command	CMtr3	650	325 μs	40.6 μs
Time from the echo back of operation command to the acceptance of password count storage addresses (Upper byte)	CMtr4	800	400 μs	50 μs



## 20. Input/Output Circuitry

### 20.1 Control Pins

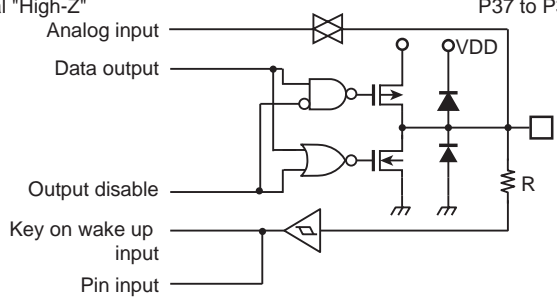
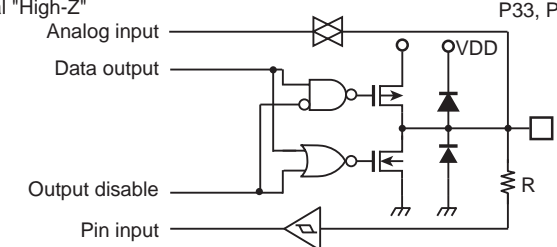
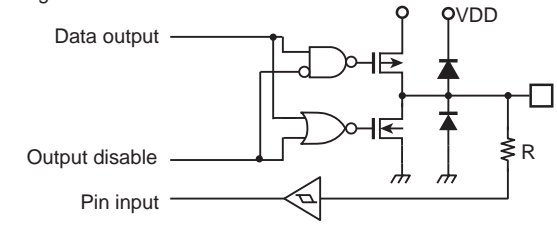
The input/output circuitries of the TMP86FH92DMG control pins are shown below.

Control Pin	I/O	Input/Output Circuitry	Remarks
XIN XOUT	Input Output		Resonator connecting pins $R_f = 1.5\text{ M}\Omega$ (typ.) $R_o = 0.5\text{ k}\Omega$ (typ.)
XTIN XTOUT	Input Output		Resonator connecting $R_f = 8\text{ M}\Omega$ (typ.) $R_o = 200\text{ k}\Omega$ (typ.)
$\overline{\text{RESET}}$	Input		Hysteresis input Pull-up resistor $R_{IN} = 220\text{ k}\Omega$ (typ.) $R = 100\text{ }\Omega$ (typ.)
TEST	Input		$R = 100\text{ }\Omega$ (typ.)

Note: The Test pins of TMP86FH92DMG does not have a pull-down resistor and diode(D1). Fix the TEST pin at Low level in MCU mode.

## 20.2 Input/Output Ports

Control Pin	I/O	Input/Output Circuitry	Remarks
P0	I/O	<p>Initial "High-Z"</p> <p>Programmable pull-up resistor <math>R_{IN}</math></p> <p>Pull-up control</p> <p>Pch control</p> <p>Data output</p> <p>Input from output latch</p> <p>High-Z control</p> <p>Pin input</p> <p>VDD</p> <p>R</p>	<p>Sink open drain output or Push-Pull output Hysteresis input High current output (Nch) (Programmable port option) <math>R = 100 \Omega</math> (typ.) <math>R_{IN} = 100 \text{ K}\Omega</math> (typ.)</p>
P1	I/O	<p>Initial "High-Z"</p> <p>Programmable pull-up resistor <math>R_{IN}</math></p> <p>Pull-up control</p> <p>Data output</p> <p>Disable</p> <p>Pin input</p> <p>VDD</p> <p>R</p>	<p>Sink open drain output or Push-Pull output Hysteresis input Programmable port option <math>R = 100 \Omega</math> (typ.) <math>R_{IN} = 100 \text{ K}\Omega</math> (typ.)</p>
P2	I/O	<p>Initial "High-Z"</p> <p>Initial "High-Z"</p> <p>Programmable pull-up resistor <math>R_{IN}</math></p> <p>Pull-up control</p> <p>Data output</p> <p>Input from output latch</p> <p>Pin input</p> <p>VDD</p> <p>R</p> <p>P20</p> <p>P22 to P21</p>	<p>Sink open drain output or Hysteresis input Programmable port option <math>R = 100 \Omega</math> (typ.) <math>R_{IN} = 100 \text{ K}\Omega</math> (typ.)</p>

Port	I/O	Input/Output Circuitry	Remarks
P3	I/O	<p>Initial "High-Z" P37 to P34</p> 	<p>Tri-state Input/Output Hysteresis input R = 100 Ω (typ.)</p>
		<p>Initial "High-Z" P33, P32</p> 	
		<p>Initial "High-Z" P31, P30</p> 	





## 21. Electrical Characteristics

### 21.1 Absolute Maximum Ratings

The absolute maximum ratings are rated values which must not be exceeded during operation, even for an instant. Any one of the ratings must not be exceeded. If any absolute maximum rating is exceeded, a device may break down or its performance may be degraded, causing it to catch fire or explode resulting in injury to the user. Thus, when designing products which include this device, ensure that no absolute maximum rating value will ever be exceeded.

(VSS = 0 V)

Parameter	Symbol	Pins	Ratings	Unit
Supply voltage	$V_{DD}$		-0.3 to 6.0	V
Input voltage	$V_{IN}$		-0.3 to $V_{DD} + 0.3$	V
Output voltage	$V_{OUT1}$		-0.3 to $V_{DD} + 0.3$	V
Output current (Per 1 pin)	$I_{OUT1}$	P0, P1, P3 ports	-1.8	mA
	$I_{OUT2}$	P1, P2, P3 ports	3.2	
	$I_{OUT3}$	P0 ports	30	
Output current (Total)	$\Sigma I_{OUT1}$	P0, P1, P3 ports	-30	
	$\Sigma I_{OUT2}$	P1, P2, P3 ports	60	
	$\Sigma I_{OUT3}$	P0 ports	80	
Power dissipation [ $T_{opr} = 85\text{ }^{\circ}\text{C}$ ]	$P_D$		200	mW
Soldering temperature (time)	$T_{sld}$		260 (10 s)	$^{\circ}\text{C}$
Storage temperature	$T_{stg}$		-55 to 125	
Operating temperature	$T_{opr}$		-40 to 85	

## 21.2 Operating Conditions

The Operating Conditions show the conditions under which the device be used in order for it to operate normally while maintaining its quality. If the device is used outside the range of Operating Conditions (power supply voltage, operating temperature range, or AC/DC rated values), it may operate erratically. Therefore, when designing your application equipment, always make sure its intended working conditions will not exceed the range of Operating Conditions.

### 21.2.1 MCU mode (Flash Programming or erasing)

(V<sub>SS</sub> = 0 V, Topr = -10 to 40°C)

Parameter	Symbol	Pins	Ratings	Min.	Max	Unit
Supply voltage	V <sub>DD</sub>		NORMAL1, 2 modes	4.5	5.5	V
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 <sub>IH2</sub>	Hysteresis input		V <sub>DD</sub> × 0.75		
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	
Clock frequency	fc	XIN, XOUT		1.0	16.0	

### 21.2.2 MCU mode (Except Flash Programming or erasing)

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

Parameter	Symbol	Pins	Ratings	Min	Max	Unit
Supply voltage (Condition 1)	V <sub>DD</sub>		fc = 16 MHz	NORMAL1, 2 modes IDLE0, 1, 2 modes	4.0	V
			fc = 8 MHz	NORMAL1, 2 modes IDLE0, 1, 2 modes		
			fs = 32.768 KHz	SLOW1, 2 modes SLEEP0, 1, 2 modes	3.0	
			STOP mode			
Supply voltage (Condition 2) (Note)	V <sub>DD</sub>		fc = 8 MHz	NORMAL1, 2 modes IDLE0, 1, 2 modes	2.7	3.0
			fs = 32.768 KHz	SLOW1, 2 modes SLEEP0, 1, 2 modes		
			STOP mode			
Input high level	V <sub>IH1</sub>	Except hysteresis input	V <sub>DD</sub> ≥ 4.0V	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> < 4.0V	V <sub>DD</sub> × 0.90		
Input low level	V <sub>IL1</sub>	Except hysteresis input	V <sub>DD</sub> ≥ 4.0V	0	V <sub>DD</sub> × 0.30	V
	V <sub>IL2</sub>	Hysteresis input			V <sub>DD</sub> × 0.25	
	V <sub>IL3</sub>		V <sub>DD</sub> < 4.0V		V <sub>DD</sub> × 0.10	
Clock frequency	fc	XIN, XOUT	V <sub>DD</sub> = 2.7 to 5.5V	1.0	8.0	MHz
			V <sub>DD</sub> = 4.0 to 5.5V		16.0	
	fs	XTIN, XTOUT	V <sub>DD</sub> = 2.7 to 5.5V	30.0	34.0	kHz

Note: When the supply voltage V<sub>DD</sub> is less than 3.0V, the operating temperature Topr must be in a range of -20°C to 85°C.

21.2.3 Serial PROM mode

(V<sub>SS</sub> = 0 V, T<sub>opr</sub> = -10 to 40 °C)

Parameter	Symbol	Pins	Condition	Min	Max	Unit
Supply voltage	V <sub>DD</sub>		NORMAL1, 2 modes	4.5	5.5	V
Input high voltage	V <sub>IH1</sub>	Except hysteresis input	V <sub>DD</sub> ≥ 4.5 V	V <sub>DD</sub> × 0.70	V <sub>DD</sub>	
	V <sub>IH2</sub>	Hysteresis input		V <sub>DD</sub> × 0.75		
Input low voltage	V <sub>IL1</sub>	Except hysteresis input	V <sub>DD</sub> ≥ 4.5 V	0	V <sub>DD</sub> × 0.30	
	V <sub>IL2</sub>	Hysteresis input			V <sub>DD</sub> × 0.25	
Clock frequency	f <sub>c</sub>	XIN, XOUT		2.0	16.0	MHz

## 21.3 DC Characteristics

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

Parameter	Symbol	Pins	Condition	Min	Typ.	Max	Unit		
Hysteresis voltage	V <sub>HS</sub>	Hysteresis input	V <sub>DD</sub> = 5.0V	-	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	V <sub>DD</sub> = 5.5 V, V <sub>IN</sub> = 5.5 V						
Input resistance	R <sub>IN1</sub>	RESET pull - up	V <sub>DD</sub> = 5.5 V, V <sub>IN</sub> = 0 V	100	200	450	kΩ		
	R <sub>IN2</sub>	PORT pull - up	V <sub>DD</sub> = 5.5 V, V <sub>IN</sub> = 0 V	50	100	200	kΩ		
Output leakage current	I <sub>LOO</sub>	P0,P1,P2,P3	V <sub>DD</sub> = 5.5 V, V <sub>OUT</sub> = 5.3 V/0.2 V	-	-	±2	μA		
Output high voltage	V <sub>OH</sub>	P0,P1,P2,P3	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 P0	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 (P0 Port)	V <sub>DD</sub> = 4.5 V, V <sub>OL</sub> = 1.0 V	-	20	-	mA		
Supply current in NORMAL1, 2 modes	I <sub>DD</sub>		V <sub>DD</sub> = 5.5 V V <sub>IN</sub> = 5.3 V/0.2 V f <sub>c</sub> = 16 MHz f <sub>s</sub> = 32.768 kHz	When a program operates on flash memory (Note4,5)	-	12.5	20	mA	
Supply current in IDLE 0, 1, 2 modes					When a program operates on RAM	-	7.5		14
						-	5.5		9
Supply current in SLOW1 mode			When a program operates on flash memory (Note4,5)	-	22	65	μA		
				When a program operates on RAM (FLSSTB<FSTB>= 0)	-	21		30	
					When a program operates on RAM (FLSSTB<FSTB>= 1)	-		16	25
Supply current in SLEEP1 mode						-	14	22	
Supply current in SLEEP0 mode				-	12	20			
Supply current in STOP mode			V <sub>DD</sub> = 5.5 V V <sub>IN</sub> = 5.3 V/0.2 V	-	10	20			
Peak current of intermittent operation (Note4,5)	I <sub>DDP-P</sub>		V <sub>DD</sub> = 5.5 V	-	10	-	mA		
			V <sub>DD</sub> = 3.0 V	-	2	-			

Note 1: Typical values show those at T<sub>opr</sub> = 25 °C and V<sub>DD</sub> = 5 V.

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

Note 3: The supply currents of SLOW2 and SLEEP2 modes are equivalent to those of IDLE0, IDLE1 and IDLE2 modes.

Note 4: When a program is executing in the flash memory or when data is being read from the flash memory, the flash memory operates in an intermittent manner, causing peak currents in the operation current, as shown in Figure 21-1.

In this case, the supply current I<sub>DD</sub> (in NORMAL1, NORMAL2 and SLOW1 modes) is defined as the sum of the average peak current and MCU current.

Note 5: When designing the power supply, make sure that peak currents can be supplied. In SLOW1 mode, the difference between the peak current and the average current becomes large.

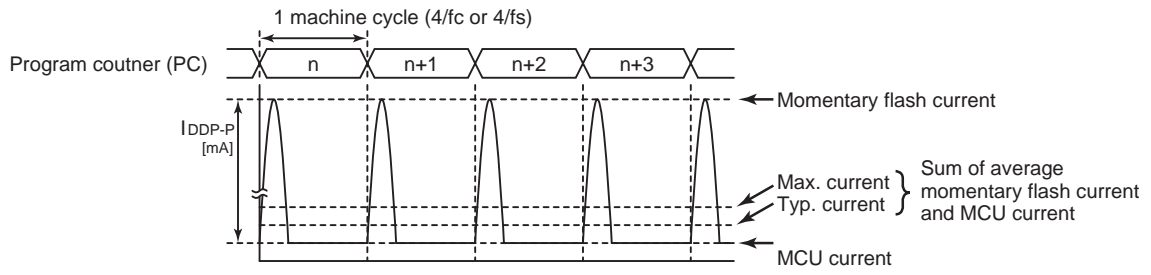


Figure 21-1 Intermittent Operation of Flash Memory

## 21.4 AD Conversion Characteristics

(Topr = -40 to 85°C)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Analog input voltage	$V_{AIN}$		$V_{SS}$	-	$V_{DD}$	V
Non linearity error		$V_{DD} = 3.0V/5.0V$ $V_{SS} = 0.0V$	-	-	$\pm 6$	LSB
Zero point error			-	-	$\pm 6$	
Full scale error			-	-	$\pm 6$	
Total error			-	-	$\pm 6$	

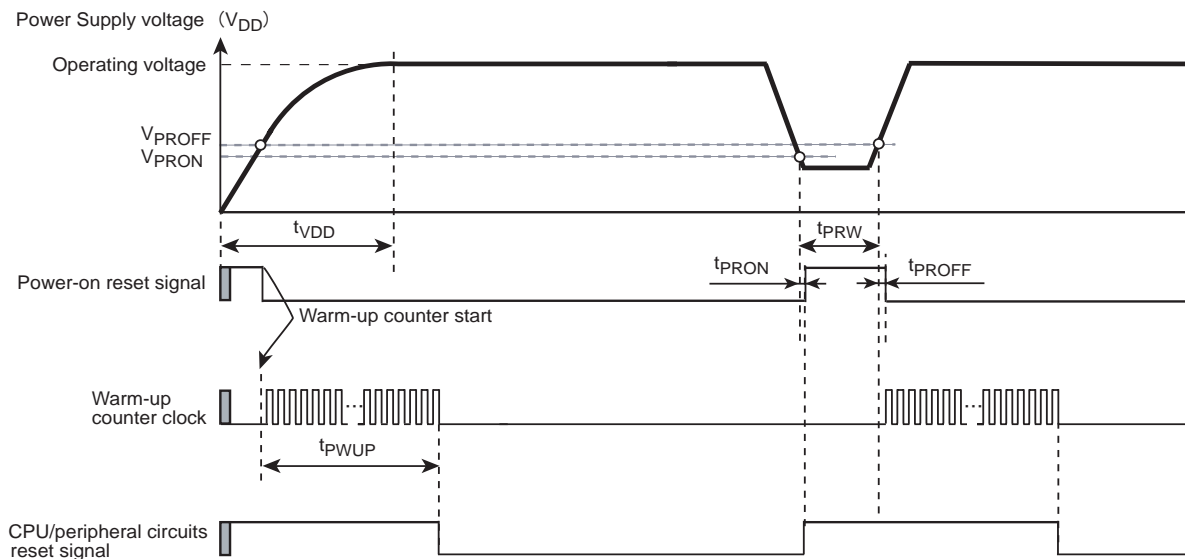
Note 1: The total error includes all errors except a quantization error, and is defined as a maximum deviation from the ideal conversion line.

Note 2: Conversion time is different in recommended value by power supply voltage.

Note 3: The voltage to be input on the AIN input pin must not exceed the range between  $V_{DD}$  and  $V_{SS}$ . If a voltage outside this range is input, conversion values will become unstable and conversion values of other channels will also be affected.

Note 4: When the supply voltage  $V_{DD}$  is less than 3.0V, the operating temperature Topr must be in a range of -20°C to 85°C.

## 21.5 Power-on reset circuit Characteristics



Note: The power-on reset circuit may not operate properly depending on transitions in supply voltage ( $V_{DD}$ ). When designing your application system, careful consideration must be given to ensure proper operation of the power-on reset circuit by referring to the device's electrical characteristics.

Figure 21-2 Operation of the Power-on reset circuit

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

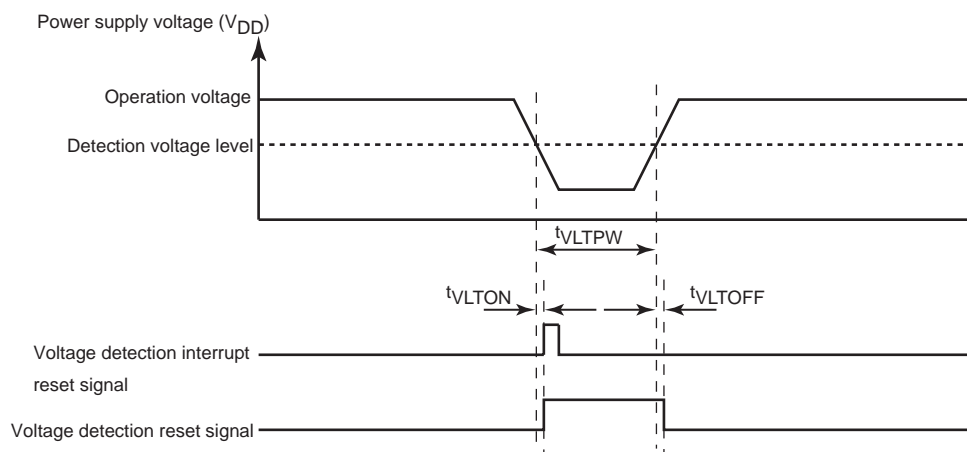
Symbol	Parameter	Min	Typ.	Max	Unit
V <sub>PROFF</sub>	Power-on reset release voltage (Note1)	2.2	2.4	2.6	V
V <sub>PRON</sub>	Power-on reset threshold voltage (Note1)	2.0	2.2	2.3	
t <sub>PROFF</sub>	Power-on reset release response time	-	0.01	0.1	ms
t <sub>PRON</sub>	Power-on reset generation response time	-	0.01	0.1	
t <sub>PRW</sub>	Power-on reset minimum pulse width	1.0	-	-	
t <sub>PWUP</sub>	Power-on warm-up time	9.0	15.0	48.0	
t <sub>VDD</sub>	Power-on reset	-	-	5	

Note 1: Because the power-on reset releasing voltage and the power-on reset detecting voltage change relative to one another, the detected voltage will never become inverted.

Note 2: The input clock to the warm-up counter is derived from the oscillation circuit. Because the oscillation frequency is unstable until the oscillation circuit stabilizes, the warm-up time includes error.

Note 3: The supply voltage must be raised to satisfy the condition t<sub>VDD</sub> < t<sub>PWUP</sub>.

## 21.6 Voltage detection circuit characteristics



Note: The voltage detection circuit may not operate properly depending on transitions in supply voltage (VDD). When designing your application system, careful consideration must be given to ensure proper operation of the voltage detection circuit by referring to the device's electrical characteristics.

Figure 21-3 Operation timing of Voltage detection circuit

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

Symbol	Parameter	Min	Typ.	Max	Unit
t <sub>VLOFF</sub>	Voltage detection release response time	-	0.01	0.1	ms
t <sub>VLTON</sub>	Voltage detection response time	-	0.01	0.1	
t <sub>VLTPW</sub>	Voltage detection minimum pulse width	1.0	-	-	

## 21.7 AC Characteristics

(V<sub>SS</sub> = 0 V, 4.0 V ≤ V<sub>DD</sub> ≤ 5.5 V, T<sub>opr</sub> = -40 to 85°C)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Machine cycle time	tcy	NORMAL1, 2 modes	0.25	-	4	μs
		IDLE0, 1, 2 modes				
		SLOW1, 2 modes	117.6	-	133.3	
		SLEEP0, 1, 2 modes				
High-level clock pulse width	t <sub>WCH</sub>	For external clock operation (XIN input) fc = 16 MHz	-	31.25	-	ns
Low-level clock pulse width	t <sub>WCL</sub>					
High-level clock pulse width	t <sub>WSH</sub>	For external clock operation (XTIN input) fs = 32.768 kHz	-	15.26	-	μs
Low-level clock pulse width	t <sub>WSL</sub>					

(V<sub>SS</sub> = 0 V, 3.0 V ≤ V<sub>DD</sub> ≤ 5.5 V, T<sub>opr</sub> = -40 to 85°C)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Machine cycle time	tcy	NORMAL1, 2 modes	0.5	-	4	μs
		IDLE0, 1, 2 modes				
		SLOW1, 2 modes	117.6	-	133.3	
		SLEEP0, 1, 2 modes				
High-level clock pulse width	t <sub>WCH</sub>	For external clock operation (XIN input) fc = 8 MHz	-	62.5	-	ns
Low-level clock pulse width	t <sub>WCL</sub>					
High-level clock pulse width	t <sub>WSH</sub>	For external clock operation (XTIN input) fs = 32.768 kHz	-	15.26	-	μs
Low-level clock pulse width	t <sub>WSL</sub>					

(V<sub>SS</sub> = 0 V, 2.7 V ≤ V<sub>DD</sub> < 3.0 V, T<sub>opr</sub> = -20 to 85°C)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Machine cycle time	tcy	NORMAL1, 2 modes	0.5	-	4	μs
		IDLE0, 1, 2 modes				
		SLOW1, 2 modes	117.6	-	133.3	
		SLEEP0, 1, 2 modes				
High-level clock pulse width	t <sub>WCH</sub>	For external clock operation (XIN input) fc = 8 MHz	-	62.5	-	ns
Low-level clock pulse width	t <sub>WCL</sub>					
High-level clock pulse width	t <sub>WSH</sub>	For external clock operation (XTIN input) fs = 32.768 kHz	-	15.26	-	μs
Low-level clock pulse width	t <sub>WSL</sub>					

## 21.8 Flash Characteristics

### 21.8.1 Write/Erase Characteristics

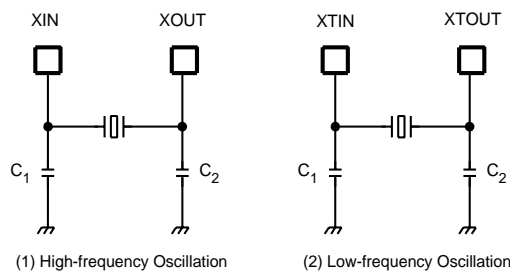
(V<sub>SS</sub> = 0 V)

Parameter	Condition	Min	Typ.	Max.	Unit
Number of guaranteed writes to flash memory	V <sub>SS</sub> = 0 V, T <sub>opr</sub> = -10 to 40°C	-	-	100	Times

Note: To rewrite data to Flash memory addresses at which data is already written, make sure to erase the existing data before rewriting data.



## 21.9 Oscillating Conditions



Note 1: To ensure stable oscillation, the resonator position, load capacitance, etc. must be appropriate. Because these factors are greatly affected by board patterns, please be sure to evaluate operation on the board on which the device will actually be mounted.

Note 2: The product numbers and specifications of the resonators by Murata Manufacturing Co., Ltd. are subject to change. For up-to-date information, please refer to the following URL:  
<http://www.murata.com>

## 21.10 Handling Precaution

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

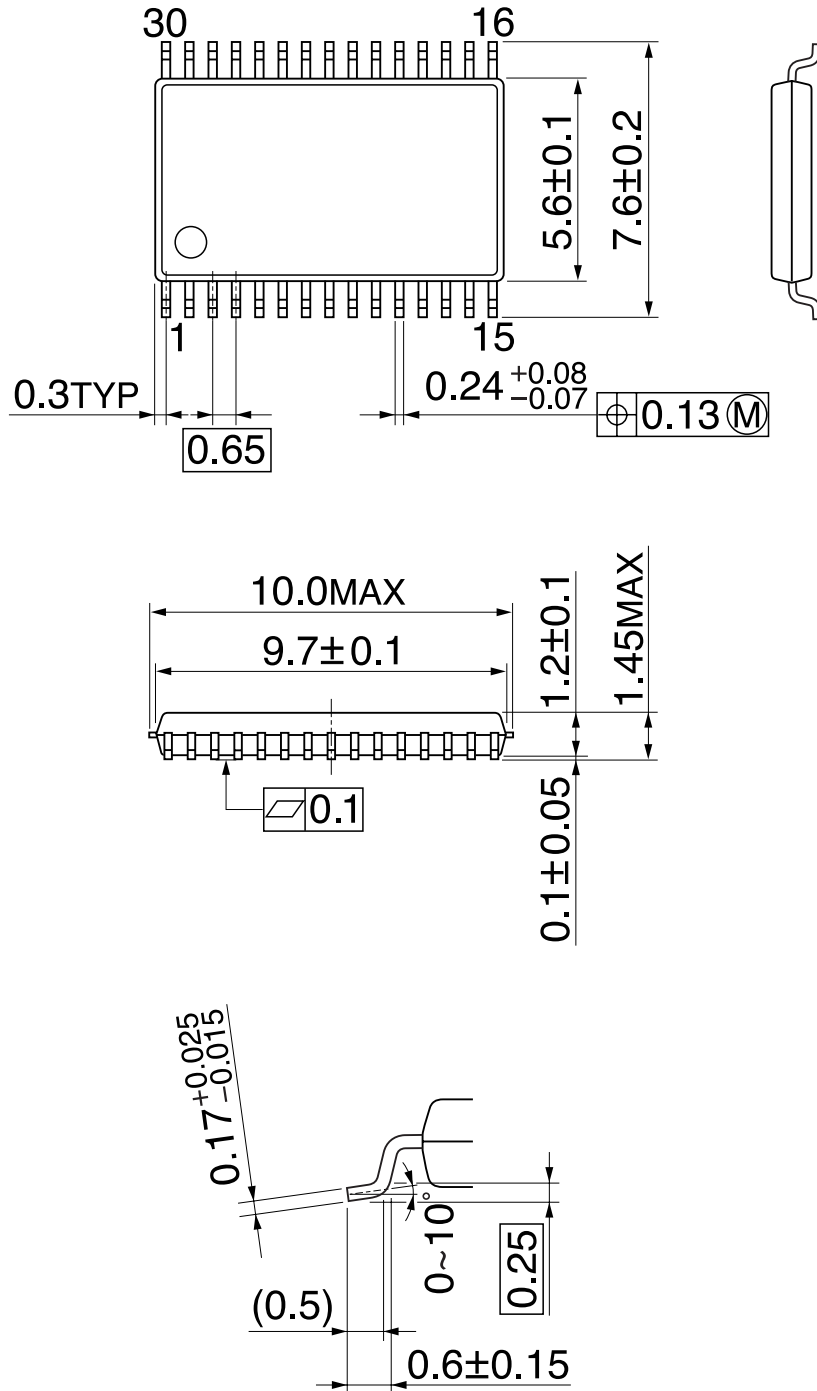
Note: The pass criterion of the above test is as follows: Solderability rate until forming  $\geq 95\%$
- When using the device (oscillator) in places exposed to high electric fields such as cathode-ray tubes, we recommend electrically shielding the package in order to maintain normal operating condition.



## 22. Package Dimensions

SSOP30-P-56-0.65 Rev 02

Unit: mm





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

Toshiba provides a variety of development tools and basic software to enable efficient software development.

These development tools have specifications that support advances in microcomputer hardware (LSI) and can be used extensively. Both the hardware and software are supported continuously with version updates.

The recent advances in CMOS LSI production technology have been phenomenal and microcomputer systems for LSI design are constantly being improved. The products described in this document may also be revised in the future. Be sure to check the latest specifications before using.

Toshiba is developing highly integrated, high-performance microcomputers using advanced MOS production technology and especially well proven CMOS technology.

We are prepared to meet the requests for custom packaging for a variety of application areas.

We are confident that our products can satisfy your application needs now and in the future.

