TOSHIBA

8 Bit Microcontroller
TLCS-870/C Series

TMP86FH46ANG

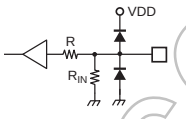
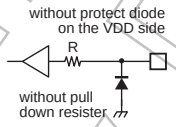
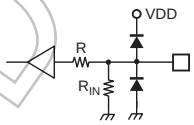
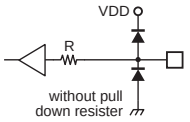
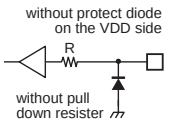
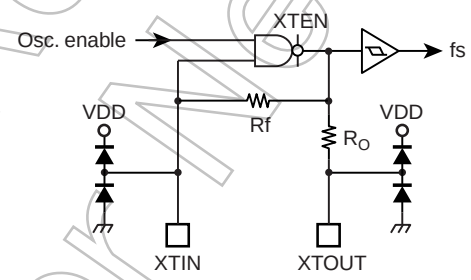
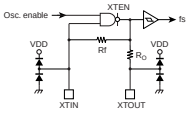
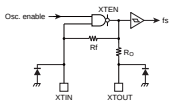
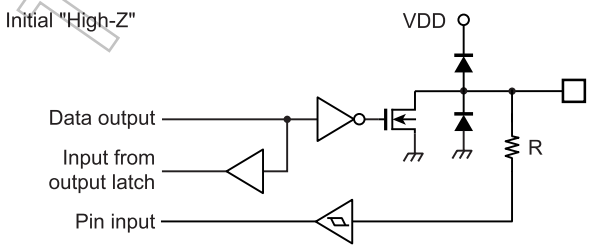
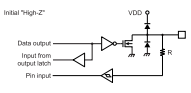
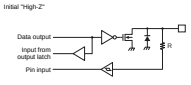
Not Recommended for New Design

TOSHIBA CORPORATION

Not Recommended for New Design

© 2010 TOSHIBA CORPORATION
All Rights Reserved

Difference among product (TMP86xx46 Series)

	86C846	86CH46 86CH46A	86CM46 86CM46A	86PH46	86PM46 86PM46A	86FH46	86FH46A 86FH46B
ROM	8192bytes (MASK)	16384bytes (MASK)	32768bytes (MASK)	16384bytes (OTP)	32768bytes (OTP)	16384bytes (FLASH)	
RAM	512bytes	512bytes	1024bytes	512bytes	1024bytes	512bytes	512bytes
DBR(note1)	-						128bytes (Flash control register contained)
I/O	33pins						
Large current output	19pins (LED direct drive)						
Interrupt	18interrupt sources (External : 6 Internal : 12)						
Timer counter	16-bit timer counter : 1ch 8-bit timer counter : 2ch						
UART	8-bit UART : 1ch						
SIO	High-Speed SIO : 1ch						
Key-on wakeup	4ch						
10-bit AD converter	Analog-input : 8ch						
Structure of TEST pin							86FH46A  86FH46B 
Structure of XTIN,XTOUT							86FH46A  86FH46B 
Structure of P2 port							86FH46A  86FH46B 

Not Recommended for New Design

	86C846	86CH46 86CH46A	86CM46 86CM46A	86PH46	86PM46 86PM46A	86FH46	86FH46A 86FH46B
Number of guaranteed writes to flash memory	-			-		100 Times	(a)86FH46B 1000 Times (b)86FH46A 100 Times
Terminal for SERIAL PROM MODE (note2)	-					BOOT1/RXD(P10) BOOT2/TXD(P11)	BOOT/RXD(P02) TXD(P03)
Emulation Chip	TMP86C947XB						
Package	SDIP42-P-600-1.78						

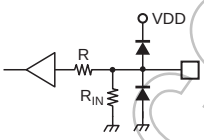
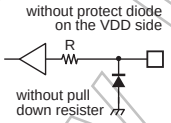
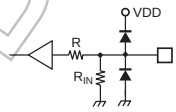
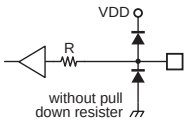
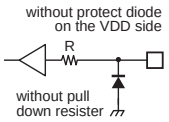
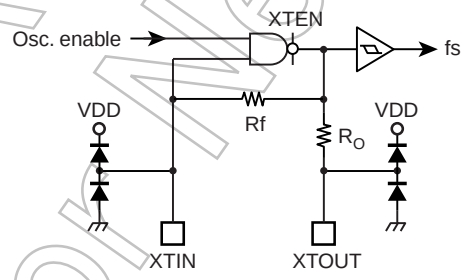
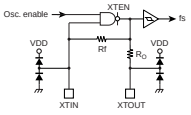
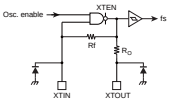
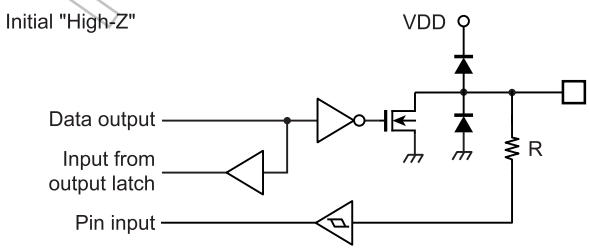
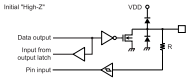
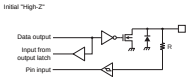
Note 1: The products with Flash memory (86FH46,86FH46A,86FH46B) contain the Flash control register (FLSCR) at 0FFFH in the DBR area. The products with mask ROM or OTP and the emulation chip do not have the FLSCR register. In these devices,therefore, a program that accesses the FLSCR register cannot function properly (executes differently as in the case of a Flash product).

Note 2: The TXD and RXD pins to be used in Serial PROM mode differ between the 86FH46 and the 86FH46A,86FH46B. Take this into consideration in your board design when you replace the product. Details of the function refer to the chapter of the 86FH46,86FH46A,86FH46B data sheet.

Note 3: P21,P22 combine XTIN,XTOUT and port.

Not Recommended for New Design

Difference among product (TMP86xx47 Series)

	86C847	86CH47 86CH47A	86CM47 86CM47A	86PH47	86PM47 86PM47A	86FH47	86FH47A 86FH47B
ROM	8192bytes (MASK)	16384bytes (MASK)	32768bytes (MASK)	16384bytes (OTP)	32768bytes (OTP)	16384bytes (FLASH)	
RAM	512bytes	512bytes	1024bytes	512bytes	1024bytes	512bytes	512bytes
DBR(note1)	-						128bytes (Flash control register contained)
I/O	35pins						
Large current output	19pins (LED direct drive)						
Interrupt	18interrupt sources (External : 6 Internal : 12)						
Timer counter	16-bit timer counter : 1ch 8-bit timer counter : 2ch						
UART	8-bit UART : 1ch						
SIO	High-Speed SIO : 1ch						
Key-on wakeup	4ch						
10-bit AD converter	Analog-input : 8ch						
Structure of TEST pin							86FH47A  86FH47B 
Structure of XTIN,XTOUT							86FH47A  86FH47B 
Structure of P2 port							86FH47A  86FH47B 

Not Recommended for New Design

	86C847	86CH47 86CH47A	86CM47 86CM47A	86PH47	86PM47 86PM47A	86FH47	86FH47A 86FH47B
Number of guaranteed writes to flash memory	-			-		100 Times	(a)86FH47B 1000 Times (b)86FH47A 100 Times
Terminal for SERIAL PROM MODE (note2)	-					BOOT1/RXD(P10) BOOT2/TXD(P11)	BOOT/RXD(P02) TXD(P03)
Emulation Chip	TMP86C947XB						
Package (LQFP44-P-1010-0.80A)	Available	Available (86CH47)	Available	N.A.	Available	Available	N.A.
Package (LQFP44-P-1010-0.80B)	N.A.	Available (86CH47A)	N.A.	Available	N.A.	N.A.	Available

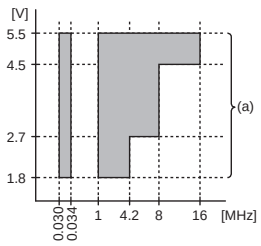
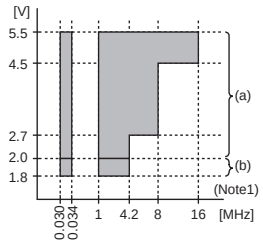
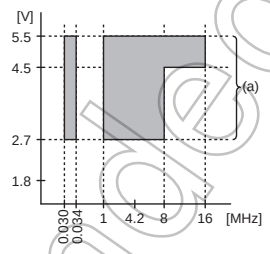
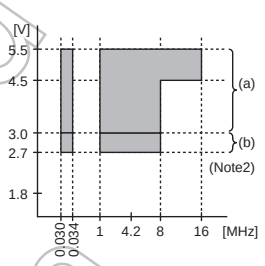
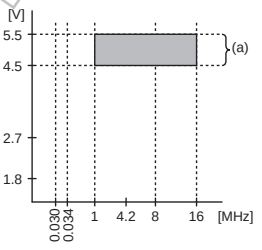
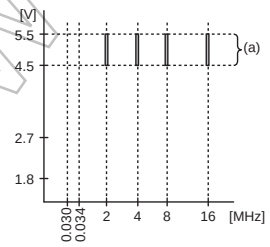
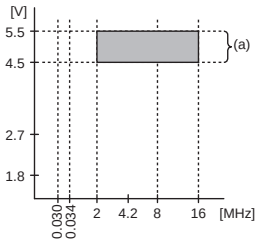
Note 1: The products with Flash memory (86FH47,86FH47A,86FH47B) contain the Flash control register (FLSCR) at 0FFFH in the DBR area. The products with mask ROM or OTP and the emulation chip do not have the FLSCR register. In these devices,therefore, a program that accesses the FLSCR register cannot function properly (executes differently as in the case of a Flash product).

Note 2: The TXD and RXD pins to be used in Serial PROM mode differ between the 86FH47 and the 86FH47A,86FH47B. Take this into consideration in your board design when you replace the product. Details of the function refer to the chapter of the 86FH47,86FH47A,86FH47B data sheet.

Note 3: P21,P22 combine XTIN,XTOUT and port.

Not Recommended for New Design

Differences in Electrical Characteristics (TMP86xx46 Series)

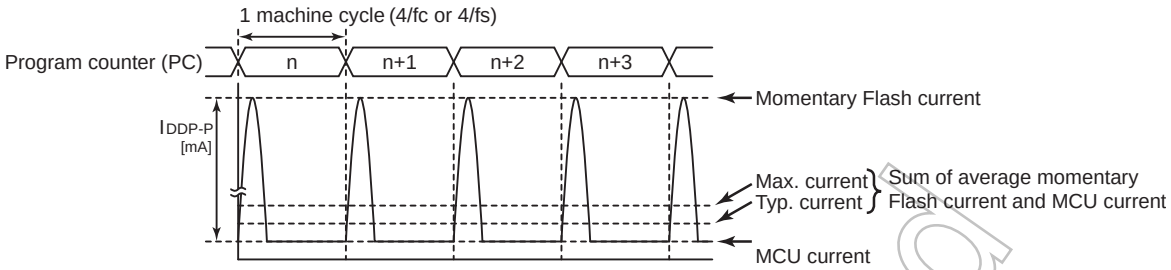
		86C846 / 86CH46 / 86CM46 86CM46A 86PM46	86PH46 86CH46A	86FH46	86FH46A 86FH46B
Operating condition (MCU mode)	Read/ Fetch				
		(a) 1.8V to 5.5V (-40 to 85 °C)	(a) 2.0V to 5.5V (-40 to 85 °C) (b) 1.8V to 2.0V (-20 to 85 °C)	(a) 2.7V to 5.5V (-40 to 85 °C)	TMP86FH46A (a) 3.0V to 5.5V (-40 to 85 °C) (b) 2.7V to 3.0V (-20 to 85 °C) TMP86FH46B (a) 2.7V to 5.5V (-40 to 85 °C)
	Erase/ Program	-	-	-	
					(a) 4.5V to 5.5V (-10 to 40 °C)
Operating condition (Serial PROM mode)		-	-		
				(a) 4.5V to 5.5V (20 to 30 °C)	(a) 4.5V to 5.5V (-10 to 40 °C)
Supply voltage (Absolute Maximum Ratings)		−0.3 ~ 6.5			86FH46A (a)−0.3 ~ 6.5 86FH46B (a)−0.3 ~ 6.0
Operating current		Operating current varies with each product. For details, refer to the datacheet (electrical chracteristics) of each product.			(Note3)

Note 1: With The 86CH46A,PH46 the operating temperature (Topr) is -20 °C to 85 °C when the supply voltage VDD is less than 2.0V.

Note 2: With The 86FH46A, the operating temperature (Topr) is -20 °C to 85 °C when the supply voltage VDD is less than 3.0V.

Note 3: With The 86FH46A,86FH46B 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 Flash memory momentarily, as shown in Figure. in this case, the supply current IDD(in NORMAL1,NORMAL2 and SLOW1 mode) is defined as the sum of the average peak current and MCU current.

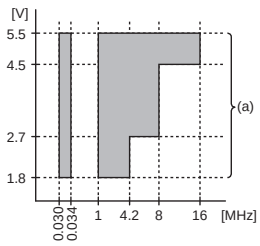
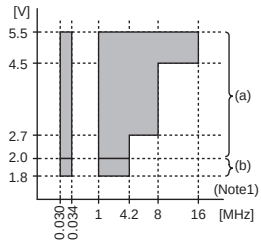
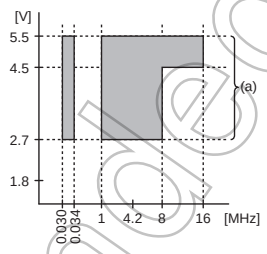
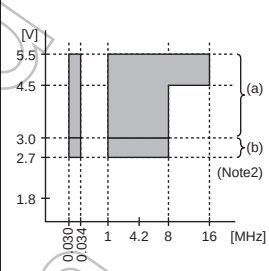
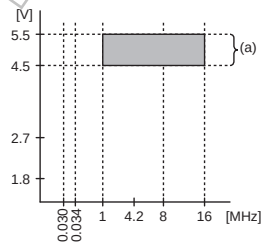
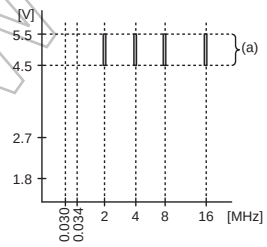
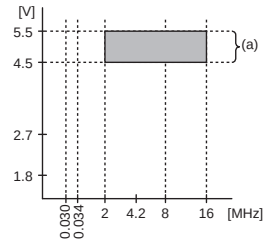
Not Recommended for New Design



Intermittent Operation of Flash Memory

Not Recommended for New Design

Differences in Electrical Characteristics (TMP86xx47 Series)

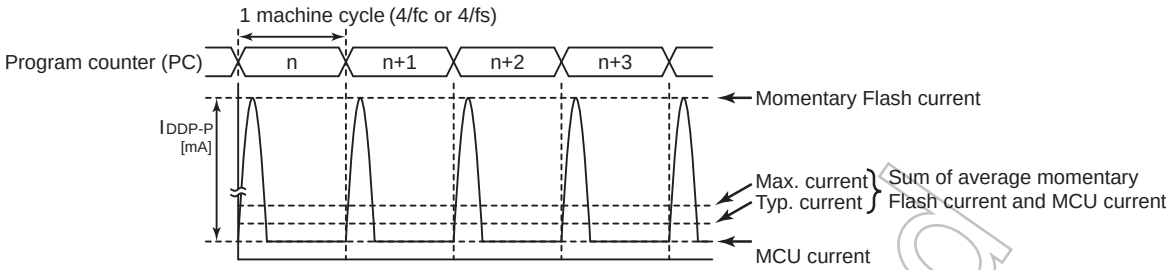
		86C847 / 86CH47 / 86CM47 86CM47A 86PM47	86PH47 86CH47A	86FH47	86FH47A 86FH47B
Operating condition (MCU mode)	Read/ Fetch				
		(a) 1.8V to 5.5V (-40 to 85 °C)	(a) 2.0V to 5.5V (-40 to 85 °C) (b) 1.8V to 2.0V (-20 to 85 °C)	(a) 2.7V to 5.5V (-40 to 85 °C)	86FH47A (a) 3.0V to 5.5V (-40 to 85 °C) (b) 2.7V to 3.0V (-20 to 85 °C) 86FH47B (a) 2.7V to 5.5V (-40 to 85 °C)
	Erase/ Program	-	-	-	
					(a) 4.5V to 5.5V (-10 to 40 °C)
Operating condition (Serial PROM mode)		-	-		
				(a) 4.5V to 5.5V (20 to 30 °C)	(a) 4.5V to 5.5V (-10 to 40 °C)
Supply voltage (Absolute Maximum Ratings)		−0.3 ~ 6.5			86FH47A (a)−0.3 ~ 6.5 86FH47B (a)−0.3 ~ 6.0
Operating current		Operating current varies with each product. For details, refer to the datacheet (electrical chracteristics) of each product.			(Note3)

Note 1: With The 86CH47A, PH47 the operating temperature (Topr) is -20 °C to 85 °C when the supply voltage VDD is less than 2.0V.

Note 2: With The 86FH47A, the operating temperature (Topr) is -20 °C to 85 °C when the supply voltage VDD is less than 3.0V.

Note 3: With The 86FH47A,86FH47B 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 Flash memory momentarily, as shown in Figure. in this case, the supply current IDD(in NORMAL1,NORMAL2 and SLOW1 mode) is defined as the sum of the average peak current and MCU current.

Not Recommended for New Design



Intermittent Operation of Flash Memory

Not Recommended for New Design

Revision History

Date	Revision	
2006/2/23	1	First Release
2006/3/13	2	Contents Revised
2006/6/29	3	Periodical updating.No change in contents.
2006/10/8	4	Contents Revised
2010/10/7	5	Contents Revised

Not Recommended for New Design

Not Recommended for New Design

Table of Contents

Difference among product (TMP86xx46 Series)

TMP86FH46ANG

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

2. Operational Description

2.1 CPU Core Functions 7
2.1.1 Memory Address Map 7
2.1.2 Program Memory (Flash) 7
2.1.3 Data Memory (RAM) 7
2.2 System Clock Controller 8
2.2.1 Clock Generator 8
2.2.2 Timing Generator 10
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
2.2.4 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
2.3 Reset Circuit 29
2.3.1 External Reset Input 29
2.3.2 Address trap reset 30
2.3.3 Watchdog timer reset 30
2.3.4 System clock reset 30

3. Interrupt Control Circuit

3.1 Interrupt latches (IL15 to IL2) 31
3.2 Interrupt enable register (EIR) 32
3.2.1 Interrupt master enable flag (IMF) 32
3.2.2 Individual interrupt enable flags (EF15 to EF4) 32
3.3 Interrupt Source Selector (INTSEL) 35
3.4 Interrupt Sequence 35
3.4.1 Interrupt acceptance processing is packaged as follows. 35
3.4.2 Saving/restoring general-purpose registers 36
3.4.2.1 Using PUSH and POP instructions
3.4.2.2 Using data transfer instructions
3.4.3 Interrupt return 38

Not Recommended for New Design

3.5 Software Interrupt (INTSW) 39
3.5.1 Address error detection 39
3.5.2 Debugging 39
3.6 Undefined Instruction Interrupt (INTUNDEF) 39
3.7 Address Trap Interrupt (INTATRAP) 39
3.8 External Interrupts 39

4. Special Function Register (SFR)

4.1 SFR 43
4.2 DBR 45

5. Time Base Timer (TBT)

5.1 Time Base Timer 47
5.1.1 Configuration 47
5.1.2 Control 47
5.1.3 Function 48
5.2 Divider Output (DVO) 49
5.2.1 Configuration 49
5.2.2 Control 49

6. Watchdog Timer (WDT)

6.1 Watchdog Timer Configuration 51
6.2 Watchdog Timer Control 52
6.2.1 Malfunction Detection Methods Using the Watchdog Timer 52
6.2.2 Watchdog Timer Enable 53
6.2.3 Watchdog Timer Disable 54
6.2.4 Watchdog Timer Interrupt (INTWDT) 54
6.2.5 Watchdog Timer Reset 55
6.3 Address Trap 56
6.3.1 Selection of Address Trap in Internal RAM (ATAS) 56
6.3.2 Selection of Operation at Address Trap (ATOUT) 56
6.3.3 Address Trap Interrupt (INTATRAP) 56
6.3.4 Address Trap Reset 57

7. I/O Ports

7.1 Port P0 (P07 to P00) 60
7.2 Port P1 (P15 to P10) 61
7.3 Port P2 (P22 to P20) 62
7.4 Port P3 (P37 to P30) 63
7.5 Port P4 (P47 to P40) 64

8. 16-Bit Timer/Counter 1 (TC1)

8.1 Configuration 65
8.2 Timer/Counter Control 66

Not Recommended for New Design

8.3 Function....68
8.3.1 Timer mode....68
8.3.2 External Trigger Timer Mode....70
8.3.3 Event Counter Mode....72
8.3.4 Window Mode....73
8.3.5 Pulse Width Measurement Mode....74
8.3.6 Programmable Pulse Generate (PPG) Output Mode....77

9. 8-Bit TimerCounter (TC3, TC4)

9.1 Configuration 81
9.2 TimerCounter Control....82
9.3 Function....87
9.3.1 8-Bit Timer Mode (TC3 and 4)....87
9.3.2 8-Bit Event Counter Mode (TC3, 4)....88
9.3.3 8-Bit Programmable Divider Output (PDO) Mode (TC3, 4)....88
9.3.4 8-Bit Pulse Width Modulation (PWM) Output Mode (TC3, 4)....91
9.3.5 16-Bit Timer Mode (TC3 and 4)....93
9.3.6 16-Bit Event Counter Mode (TC3 and 4)....94
9.3.7 16-Bit Pulse Width Modulation (PWM) Output Mode (TC3 and 4)....94
9.3.8 16-Bit Programmable Pulse Generate (PPG) Output Mode (TC3 and 4)....97
9.3.9 Warm-Up Counter Mode....99
9.3.9.1 Low-Frequency Warm-up Counter Mode
(NORMAL1 → NORMAL2 → SLOW2 → SLOW1)
9.3.9.2 High-Frequency Warm-Up Counter Mode
(SLOW1 → SLOW2 → NORMAL2 → NORMAL1)

10. Synchronous Serial Interface (SIO)

10.1 Configuration 101
10.2 Control....102
10.3 Function....104
10.3.1 Serial clock....104
10.3.1.1 Clock source
10.3.1.2 Shift edge
10.3.2 Transfer bit direction....106
10.3.2.1 Transmit mode
10.3.2.2 Receive mode
10.3.2.3 Transmit/receive mode
10.3.3 Transfer modes....107
10.3.3.1 Transmit mode
10.3.3.2 Receive mode
10.3.3.3 Transmit/receive mode

11. Asynchronous Serial interface (UART)

11.1 Configuration 119
11.2 Control 120
11.3 Transfer Data Format....123
11.4 Transfer Rate....124
11.5 Data Sampling Method....124
11.6 STOP Bit Length....125
11.7 Parity....125
11.8 Transmit/Receive Operation....125
11.8.1 Data Transmit Operation....125
11.8.2 Data Receive Operation....125

Not Recommended for New Design

11.9 Status Flag 126
11.9.1 Parity Error 126
11.9.2 Framing Error 126
11.9.3 Overrun Error 126
11.9.4 Receive Data Buffer Full 127
11.9.5 Transmit Data Buffer Empty 127
11.9.6 Transmit End Flag 128

12. 10-bit AD Converter (ADC)

12.1 Configuration 129
12.2 Register configuration 130
12.3 Function 133
12.3.1 Software Start Mode 133
12.3.2 Repeat Mode 133
12.3.3 Register Setting 134
12.4 STOP/SLOW Modes during AD Conversion 135
12.5 Analog Input Voltage and AD Conversion Result 136
12.6 Precautions about AD Converter 137
12.6.1 Analog input pin voltage range 137
12.6.2 Analog input shared pins 137
12.6.3 Noise Countermeasure 137

13. Key-on Wakeup (KWU)

13.1 Configuration 139
13.2 Control 139
13.3 Function 139

14. Flash Memory

14.1 Flash Memory Control 142
14.1.1 Flash Memory Command Sequence Execution Control (FLSCR<FLSMD>) 142
14.1.2 Flash Memory Standby Control (FLSSTB<FSTB>) 142
14.2 Command Sequence 144
14.2.1 Byte Program 144
14.2.2 Sector Erase (4-kbyte Erase) 144
14.2.3 Chip Erase (All Erase) 145
14.2.4 Product ID Entry 145
14.2.5 Product ID Exit 145
14.2.6 Security Program 145
14.3 Toggle Bit (D6) 146
14.4 Access to the Flash Memory Area 147
14.4.1 Flash Memory Control in the Serial PROM Mode 147
14.4.1.1 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)
14.4.2 Flash Memory Control in the MCU mode 149
14.4.2.1 How to write to the flash memory by executing a user write control program in the RAM area (in the MCU mode)

15. Serial PROM Mode

15.1 Outline 151
15.2 Memory Mapping 151

Not Recommended for New Design

15.3 Serial PROM Mode Setting 152
15.3.1 Serial PROM Mode Control Pins 152
15.3.2 Pin Function 152
15.3.3 Example Connection for On-Board Writing 153
15.3.4 Activating the Serial PROM Mode 154
15.4 Interface Specifications for UART 155
15.5 Operation Command 156
15.6 Operation Mode 156
15.6.1 Flash Memory Erasing Mode (Operating command: F0H) 158
15.6.2 Flash Memory Writing Mode (Operation command: 30H) 160
15.6.3 RAM Loader Mode (Operation Command: 60H) 163
15.6.4 Flash Memory SUM Output Mode (Operation Command: 90H) 165
15.6.5 Product ID Code Output Mode (Operation Command: C0H) 166
15.6.6 Flash Memory Status Output Mode (Operation Command: C3H) 167
15.6.7 Flash Memory security program Setting Mode (Operation Command: FAH) 168
15.7 Error Code 170
15.8 Checksum (SUM) 170
15.8.1 Calculation Method 170
15.8.2 Calculation data 171
15.9 Intel Hex Format (Binary) 172
15.10 Passwords 172
15.10.1 Password String 173
15.10.2 Handling of Password Error 173
15.10.3 Password Management during Program Development 173
15.11 Product ID Code 174
15.12 Flash Memory Status Code 174
15.13 Specifying the Erasure Area 176
15.14 Port Input Control Register 176
15.15 Flowchart 178
15.16 UART Timing 179

16. Input/Output Circuitry

16.1 Control Pins 181
16.2 Input/Output Ports 182

17. Electrical Characteristics

17.1 Absolute Maximum Ratings 183
17.2 Operating Conditions 184
17.2.1 Serial PROM mode 184
17.2.2 MCU mode (Except Flash Programming or erasing) 184
17.2.3 MCU mode (Flash Programming or erasing) 185
17.3 DC Characteristics 186
17.4 AD Characteristics 188
17.5 AC Characteristics 190
17.6 Flash Characteristics 191
17.6.1 Write Characteristics 191
17.7 Oscillating Conditions 192
17.8 Handling Precaution 192

Not Recommended for New Design

18. Package Dimensions

Not Recommended for New Design

CMOS 8-Bit Microcontroller

TMP86FH46ANG

The TMP86FH46ANG is a single-chip 8-bit high-speed and high-functionality microcomputer incorporating 16384 bytes of Flash Memory. It is pin-compatible with the TMP86CH46ANG/TMP86C846NG (Mask ROM version). The TMP86FH46ANG can realize operations equivalent to those of the TMP86CH46ANG/TMP86C846NG by programming the on-chip Flash Memory.

Product No.	ROM (FLASH)	RAM	Package	MaskROM MCU	Emulation Chip
TMP86FH46ANG	16384 bytes	512 bytes	P-SDIP42-600-1.78	TMP86CH46ANG/ TMP86C846NG	TMP86C947XB

Not Recommended for New Design

1.1 Features

1. 8-bit single chip microcomputer TLCS-870/C series
   - Instruction execution time :
     0.25 μs (at 16 MHz)
     122 μs (at 32.768 kHz)
   - 132 types & 731 basic instructions
2. 18interrupt sources (External : 6 Internal : 12)
3. Input / Output ports (33 pins)
   Large current output: 19pins (Typ. 20mA), LED direct drive
4. Prescaler
   - Time base timer
   - Divider output function
5. Watchdog Timer
6. 16-bit timer counter: 1 ch
   - Timer, External trigger, Window, Pulse width measurement,
     Event counter, Programmable pulse generate (PPG) modes
7. 8-bit timer counter : 2 ch
   - Timer, Event counter, Programmable divider output (PDO),
     Pulse width modulation (PWM) output,
     Programmable pulse generation (PPG),
     16bit mode (8bit timer 2ch combination) modes
8. Serial Interface
   - High-Speed 8-bit SIO: 1ch
9. 8-bit UART : 1 ch
10. 10-bit successive approximation type AD converter
    - Analog input: 8 ch
11. Key-on wakeup : 4 ch
12. Clock operation
    Single clock mode
    Dual clock mode

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

13. Low power consumption operation

    STOP mode: Oscillation stops. (Battery/Capacitor back-up.)

    SLOW1 mode: Low power consumption operation using low-frequency clock.(High-frequency clock stop.)

    SLOW2 mode: Low power consumption operation using low-frequency clock.(High-frequency clock oscillate.)

    IDLE0 mode: CPU stops, and only the Time-Based-Timer(TBT) on peripherals operate using high frequency clock. Release by falling edge of the source clock which is set by TBTCR<TBTCK>.

    IDLE1 mode: CPU stops and peripherals operate using high frequency clock. Release by interruputs(CPU restarts).

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

    SLEEP0 mode: CPU stops, and only the Time-Based-Timer(TBT) on peripherals operate using low frequency clock.Release by falling edge of the source clock which is set by TBTCR<TBTCK>.

    SLEEP1 mode: CPU stops, and peripherals operate using low frequency clock. Release by interruput.(CPU restarts).

    SLEEP2 mode: CPU stops and peripherals operate using high and low frequency clock. Release by interruput.

14. Wide operation voltage:

    4.5 V to 5.5 V at 16MHz /32.768 kHz
    2.7 V to 5.5 V at 8 MHz /32.768 kHz

Not Recommended for New Design

1.2 Pin Assignment

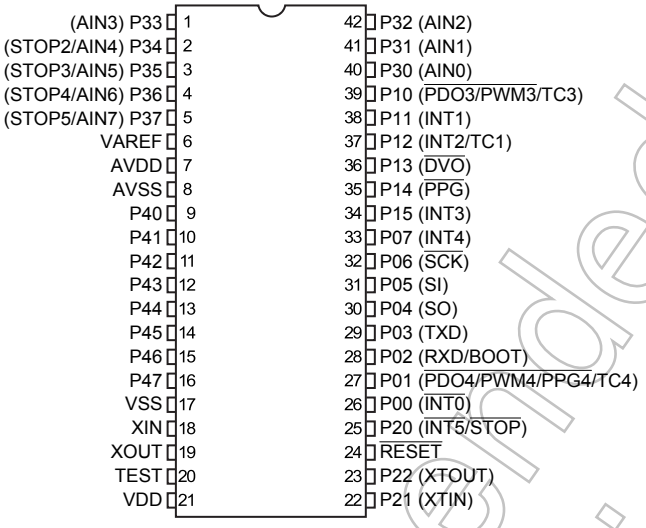
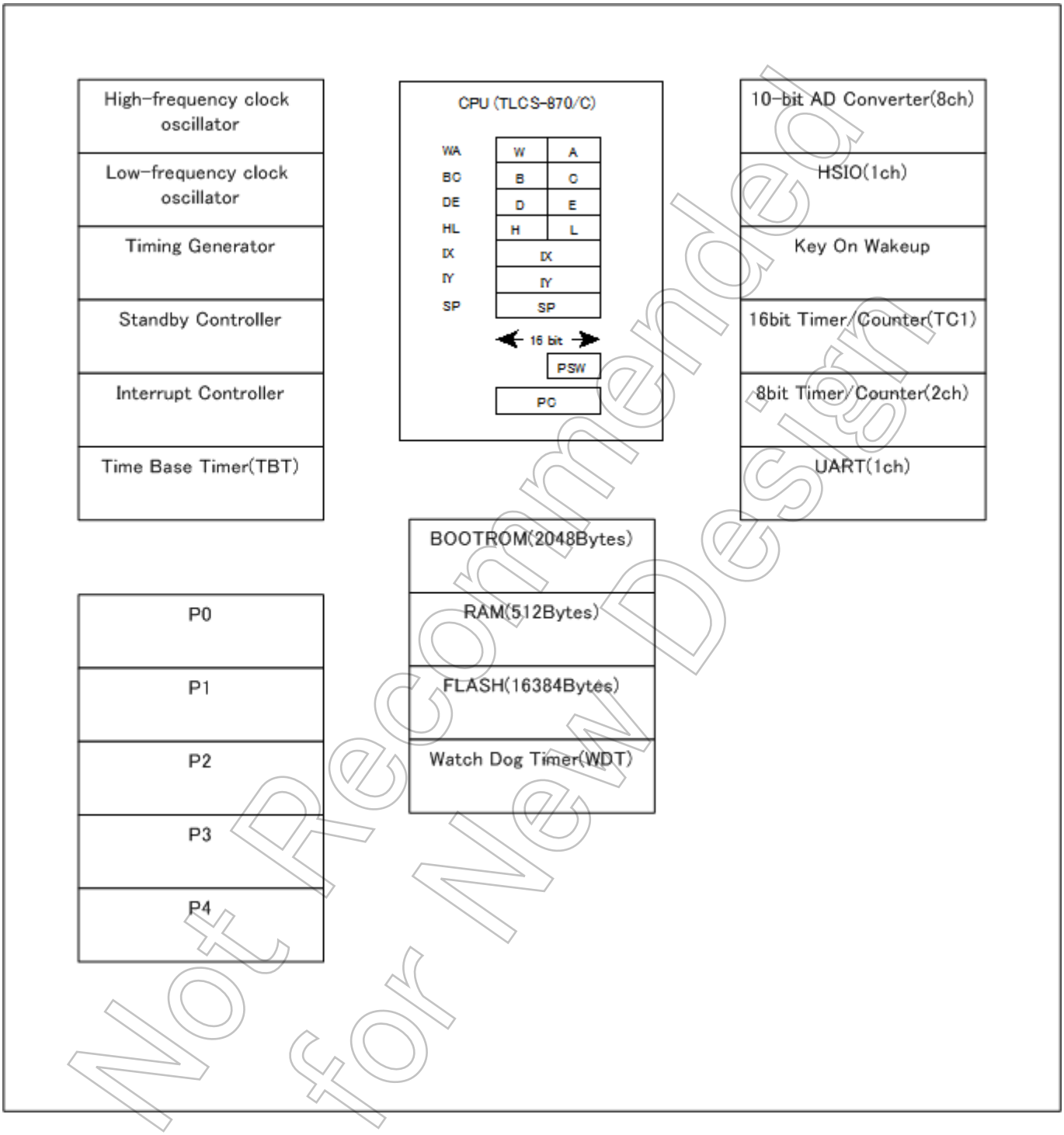


Figure 1-1 Pin Assignment

Not Recommended for New Design

1.3 Block Diagram



Not Recommended for New Design

Figure 1-2 Block Diagram

1.4 Pin Names and Functions

The TMP86FH46ANG has MCU mode, parallel PROM mode, and serial PROM mode. Table 1-1 shows the pin functions in MCU mode. The serial PROM mode is explained later in a separate chapter.

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

Pin Name	Pin Number	Input/Output	Functions
P07 INT4	33	IO I	PORT07 External interrupt 4 input
P06 $\overline{SCK}$	32	IO IO	PORT06 Serial clock input/output
P05 SI	31	IO I	PORT05 Serial data input
P04 SO	30	IO O	PORT04 Serial data output
P03 TXD	29	IO O	PORT03 UART data output
P02 RXD BOOT	28	IO I I	PORT02 UART data input Serial PROM mode control input
P01 $\overline{PDO4/PWM4/PPG4}$ TC4	27	IO O I	PORT01 PDO4/PWM4/PPG4 output TC4 input
P00 $\overline{INT0}$	26	IO I	PORT00 External interrupt 0 input
P15 INT3	34	IO I	PORT15 External interrupt 3 input
P14 $\overline{PPG}$	35	IO O	PORT14 PPG output
P13 $\overline{DVO}$	36	IO O	PORT13 Divider Output
P12 INT2 TC1	37	IO I I	PORT12 External interrupt 2 input TC1 input
P11 INT1	38	IO I	PORT11 External interrupt 1 input
P10 $\overline{PDO3/PWM3}$ TC3	39	IO O I	PORT10 PDO3/PWM3 output TC3 input
P22 XTOUT	23	IO O	PORT22 Resonator connecting pins(32.768kHz) for inputting external clock
P21 XTIN	22	IO I	PORT21 Resonator connecting pins(32.768kHz) for inputting external clock
P20 $\overline{INT5}$ $\overline{STOP}$	25	IO I I	PORT20 External interrupt 5 input STOP mode release signal input

Not Recommended for New Design

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

Pin Name	Pin Number	Input/Output	Functions
P37 AIN7 STOP5	5	IO I I	PORT37 Analog Input7 STOP5 input
P36 AIN6 STOP4	4	IO I I	PORT36 Analog Input6 STOP4 input
P35 AIN5 STOP3	3	IO I I	PORT35 Analog Input5 STOP3 input
P34 AIN4 STOP2	2	IO I I	PORT34 Analog Input4 STOP2 input
P33 AIN3	1	IO I	PORT33 Analog Input3
P32 AIN2	42	IO I	PORT32 Analog Input2
P31 AIN1	41	IO I	PORT31 Analog Input1
P30 AIN0	40	IO I	PORT30 Analog Input0
P47	16	IO	PORT47
P46	15	IO	PORT46
P45	14	IO	PORT45
P44	13	IO	PORT44
P43	12	IO	PORT43
P42	11	IO	PORT42
P41	10	IO	PORT41
P40	9	IO	PORT40
XIN	18	I	Resonator connecting pins for high-frequency clock
XOUT	19	O	Resonator connecting pins for high-frequency clock
$\overline{\text{RESET}}$	24	IO	Reset signal
TEST	20	I	Test pin for out-going test. Normally, be fixed to low.
VAREF	6	I	Analog Base Voltage Input Pin for A/D Conversion

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

Pin Name	Pin Number	Input/Output	Functions
AVDD	7	I	Analog Power Supply
AVSS	8	I	Analog Power Supply
VDD	21	I	+5V
VSS	17	I	0(GND)

Not Recommended for New Design

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 TMP86FH46ANG 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 TMP86FH46ANG memory address map.

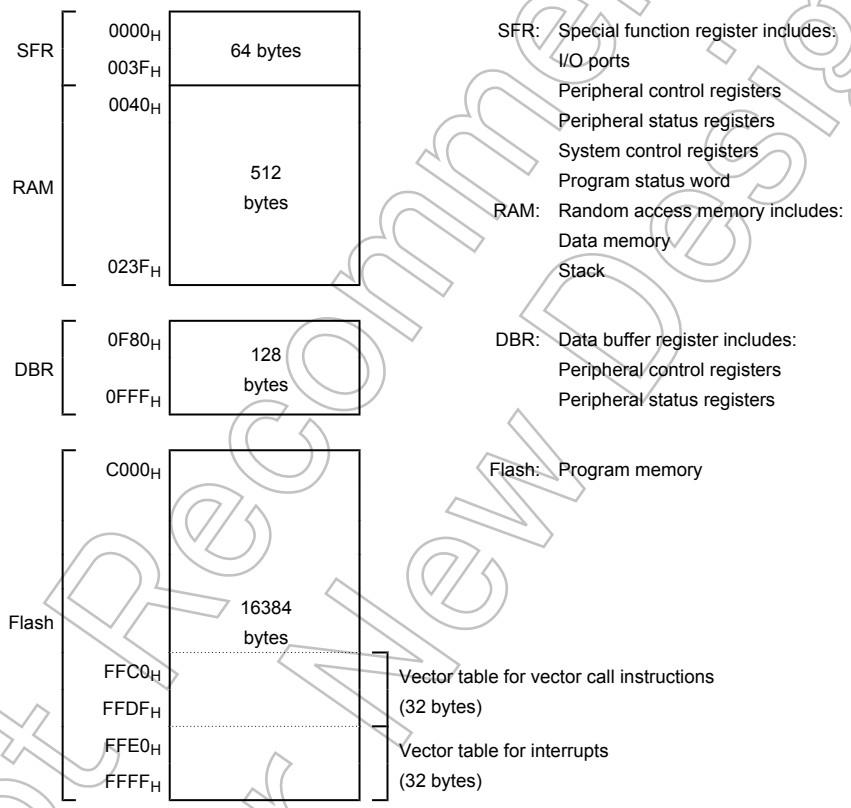


Figure 2-1 Memory Address Map

2.1.2 Program Memory (Flash)

The TMP86FH46ANG has a 16384 bytes (Address C000H to FFFFH) of program memory (Flash).

2.1.3 Data Memory (RAM)

The TMP86FH46ANG 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.

Not Recommended for New Design

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”. (TMP86FH46ANG)

```
        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.

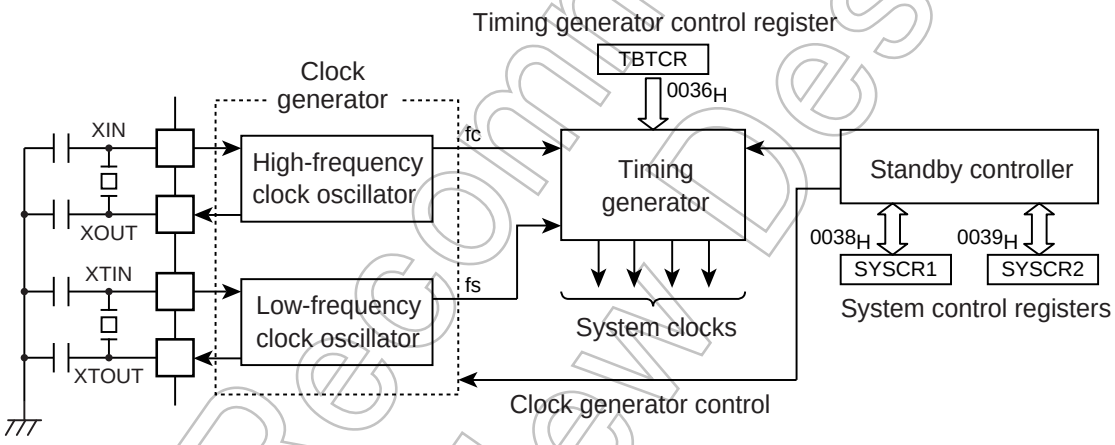


Figure 2-2 System Clock Control

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.

Not Recommended for New Design

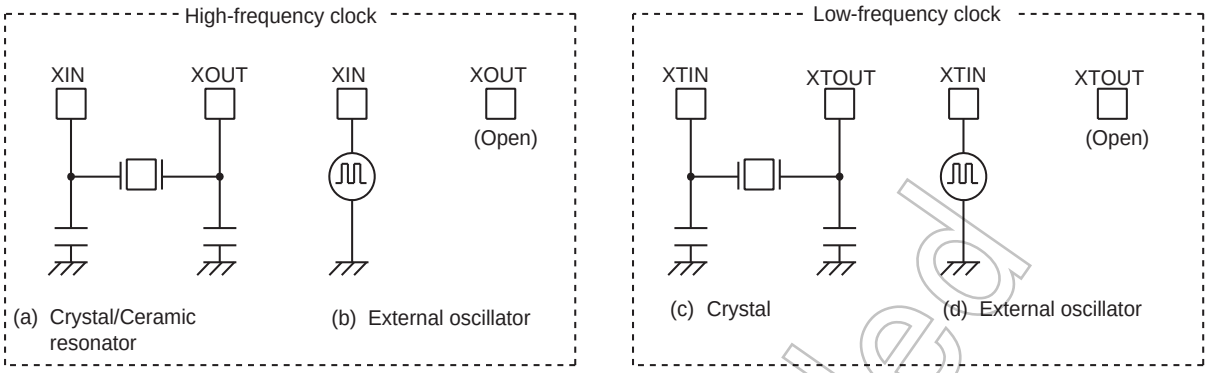


Figure 2-3 Examples of Resonator Connection

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.

Not Recommended for New Design

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 (fc or fs). 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, SYSCR2<SYSCK> and TBTCR<DV7CK>, that is shown in Figure 2-4. As reset and STOP mode started/canceled, the prescaler and the divider are cleared to "0".

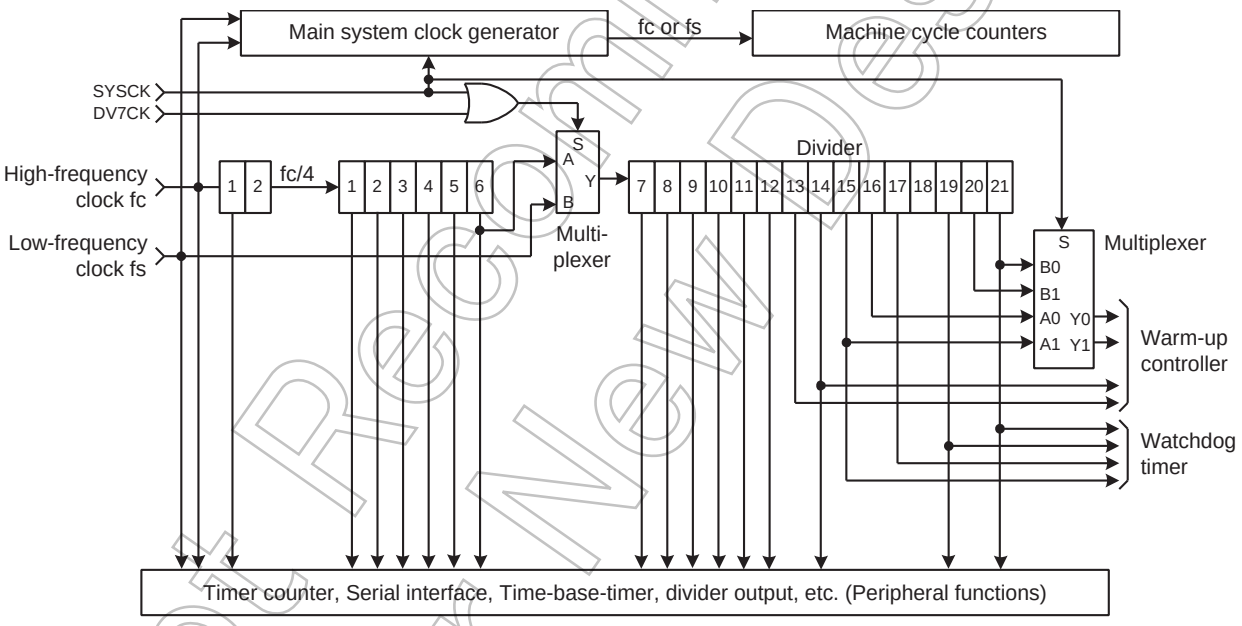


Figure 2-4 Configuration of Timing Generator

Not Recommended for New Design

Timing Generator Control Register

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

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

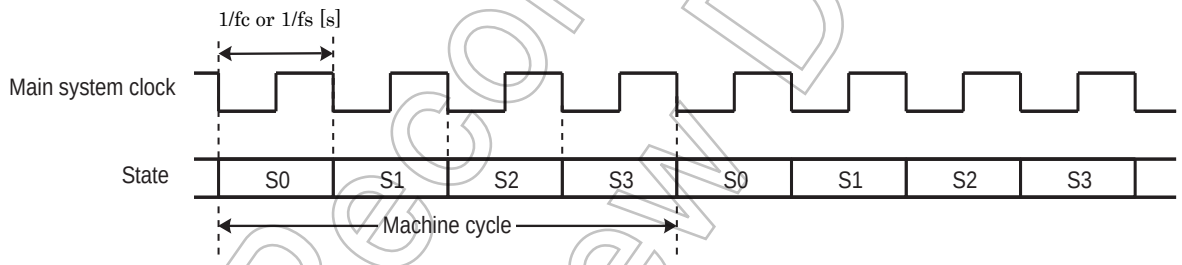


Figure 2-5 Machine Cycle

2.2.3 Operation Mode Control Circuit

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

2.2.3.1 Single-clock mode

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

(1) NORMAL1 mode

In this mode, both the CPU core and on-chip peripherals operate using the high-frequency clock. The TMP86FH46ANG is placed in this mode after reset.

Not Recommended for New Design

(2) IDLE1 mode

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

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

(3) IDLE0 mode

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

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

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

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

2.2.3.2 Dual-clock mode

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

Not Recommended for New Design

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

(4) IDLE2 mode

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

(5) SLEEP1 mode

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

(6) SLEEP2 mode

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

(7) SLEEP0 mode

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

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

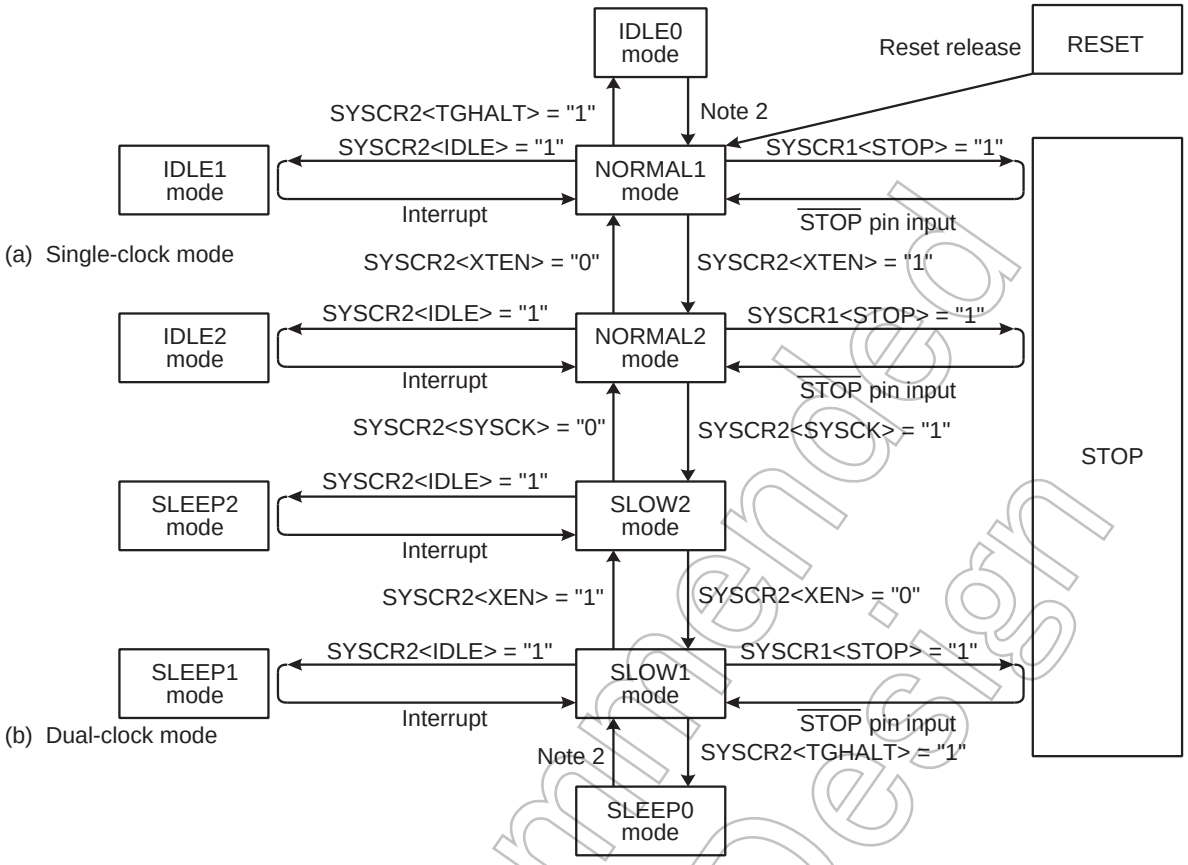
When returned from SLEEP0 mode, the CPU restarts operating, entering SLOW1 mode back again. SLEEP0 mode is entered and returned regardless of how TBTCR<TBTEN> is set. When IMF = “1”, EF6 (TBT interrupt individual enable flag) = “1”, and TBTCR<TBTEN> = “1”, interrupt processing is performed. When SLEEP0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

2.2.3.3 STOP mode

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

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

Not Recommended for New Design



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

Not Recommended for New Design

System Control Register 1

SYSCR1 (0038H)

7	6	5	4	3	2	1	0
STOP	RELM	RETM	OUTEN	WUT			

(Initial value: 0000 000*)

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
		000	3×2^{16}/fc	3×2^{13}/fs	
		010	2^{16}/fc	2^{13}/fs	
		100	3×2^{14}/fc	3×2^{6}/fs	
		110	2^{14}/fc	2^{6}/fs	
		*01	3×2^{10}/fc	3×2^{6}/fs	
		*11	2^{10}/fc	2^{6}/fs	

Note 1: Always set RETM to “0” when transiting from NORMAL mode to STOP mode. Always set RETM to “1” when transiting from SLOW mode to STOP mode.

Note 2: When STOP mode is released with $\overline{\text{RESET}}$ pin input, a return is made to NORMAL1 regardless of the RETM contents.

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

Note 4: Bits 1 in SYSCR1 are read as undefined data when a read instruction is executed.

Note 5: As the hardware becomes STOP mode under OUTEN = “0”, input value is fixed to “0”; therefore it may cause external interrupt request on account of falling edge.

Note 6: When the key-on wakeup is used, RELM should be set to "1".

Note 7: Port P20 is used as $\overline{\text{STOP}}$ pin. Therefore, when stop mode is started, OUTEN does not affect to P20, and P20 becomes High-Z mode.

Note 8: The warming-up time should be set correctly for using oscillator.

System Control Register 2

SYSCR2 (0039H)

7	6	5	4	3	2	1	0
XEN	XTEN	SYSCK	IDLE		TGHALT		

(Initial value: 1000 *0**)

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

Note 1: A reset is applied if both XEN and XTEN are cleared to “0”, XEN is cleared to “0” when SYSCK = “0”, or XTEN is cleared to “0” when SYSCK = “1”.

Note 2: *: Don’t care, TG: Timing generator, *; Don’t care

Note 3: Bits 3, 1 and 0 in SYSCR2 are always read as undefined value.

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

Note 5: Because returning from IDLE0/SLEEP0 to NORMAL1/SLOW1 is executed by the asynchronous internal clock, the period of IDLE0/SLEEP0 mode might be shorter than the period setting by TBTCR<TBTCK>.

Not Recommended for New Design

Note 6: When IDLE1/2 or SLEEP1/2 mode is released, IDLE is automatically cleared to “0”.

Note 7: When IDLE0 or SLEEP0 mode is released, TGHALT is automatically cleared to “0”.

Note 8: Before setting TGHALT to “1”, be sure to stop peripherals. If peripherals are not stopped, the interrupt latch of peripherals may be set after IDLE0 or SLEEP0 mode is released.

2.2.4 Operating Mode Control

2.2.4.1 STOP mode

STOP mode is controlled by the system control register 1, the $\overline{\text{STOP}}$ pin input and key-on wakeup input (STOP5 to STOP2) which is controlled by the STOP mode release control register (STOPCR).

The $\overline{\text{STOP}}$ pin is also used both as a port P20 and an $\overline{\text{INT5}}$ (external interrupt input 5) pin. STOP mode is started by setting SYSCR1<STOP> to “1”. During STOP mode, the following status is maintained.

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

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

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

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

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

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

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

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

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

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

Not Recommended for New Design

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

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

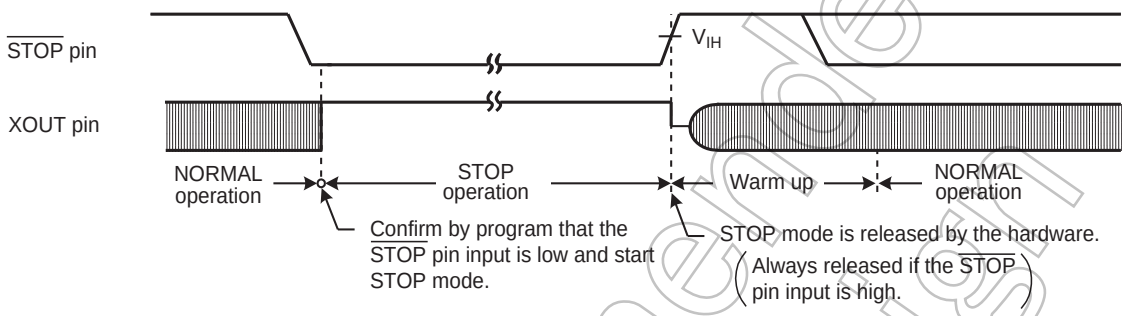


Figure 2-7 Level-sensitive Release Mode

Note 1: Even if the $\overline{\text{STOP}}$ pin input is low after warm-up start, the STOP mode is not restarted.

Note 2: In this case of changing to the level-sensitive mode from the edge-sensitive mode, the release mode is not switched until a rising edge of the $\overline{\text{STOP}}$ pin input is detected.

(2) Edge-sensitive release mode (RELM = “0”)

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

Example :Starting STOP mode from NORMAL mode

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

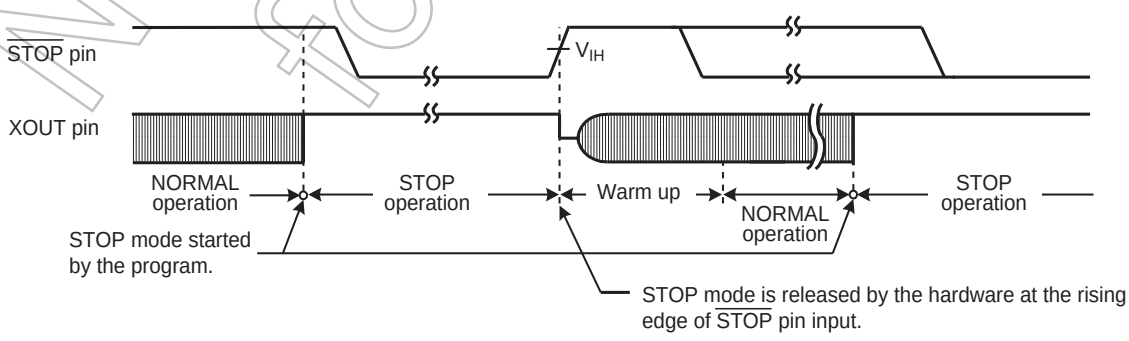


Figure 2-8 Edge-sensitive Release Mode

Not Recommended for New Design

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. Six 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 fc = 16.0 MHz, fs = 32.768 kHz)

WUT	Warm-up Time [ms]	
	Return to NORMAL Mode	Return to SLOW Mode
000	12.288	750
010	4.096	250
100	3.072	5.85
110	1.024	1.95
*01	0.192	5.9
*11	0.064	2.0

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.

Not Recommended for New Design

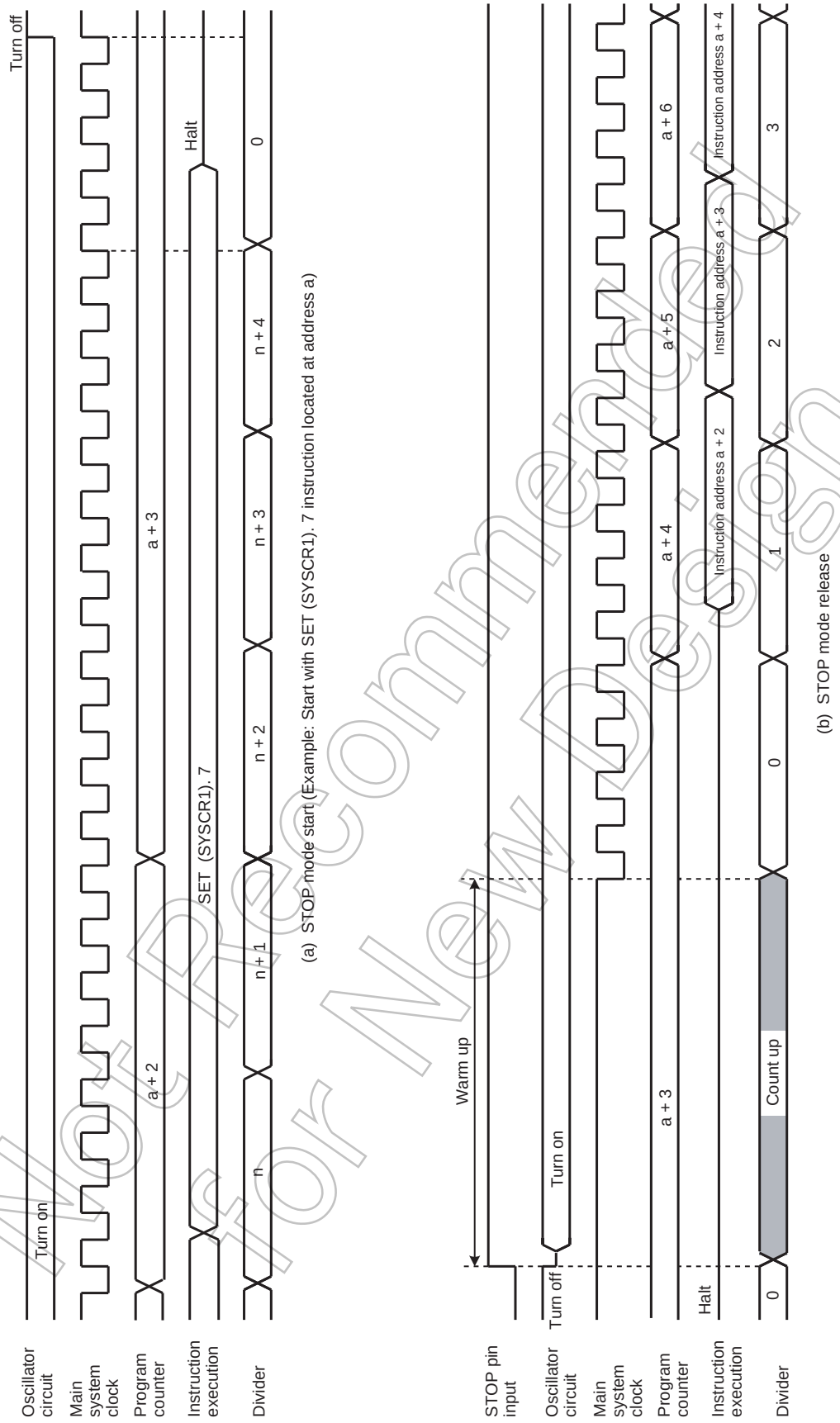


Figure 2-9 STOP Mode Start/Release

Not Recommended for New Design

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.

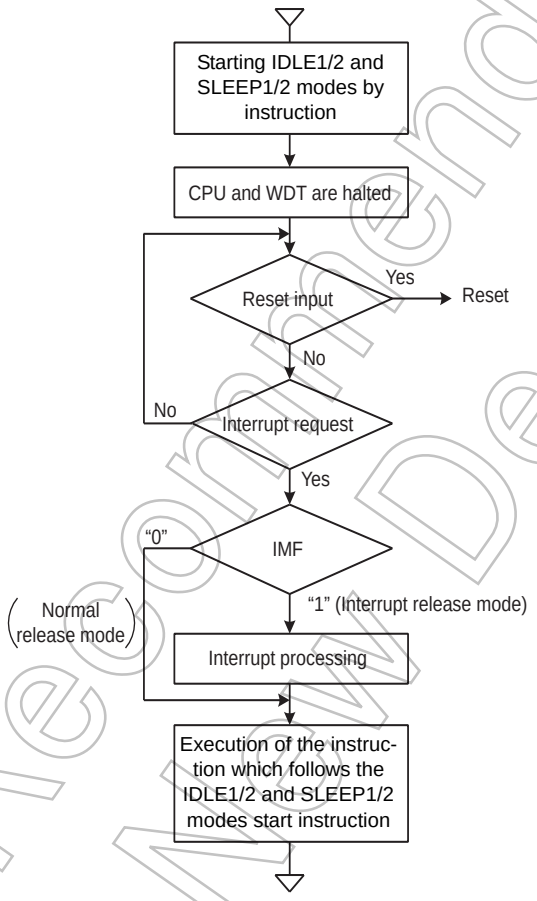


Figure 2-10 IDLE1/2 and SLEEP1/2 Modes

Not Recommended for New Design

- Start the IDLE1/2 and SLEEP1/2 modes

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

- Release the IDLE1/2 and SLEEP1/2 modes

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

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

(1) Normal release mode (IMF = “0”)

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

(2) Interrupt release mode (IMF = “1”)

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

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

Not Recommended for New Design

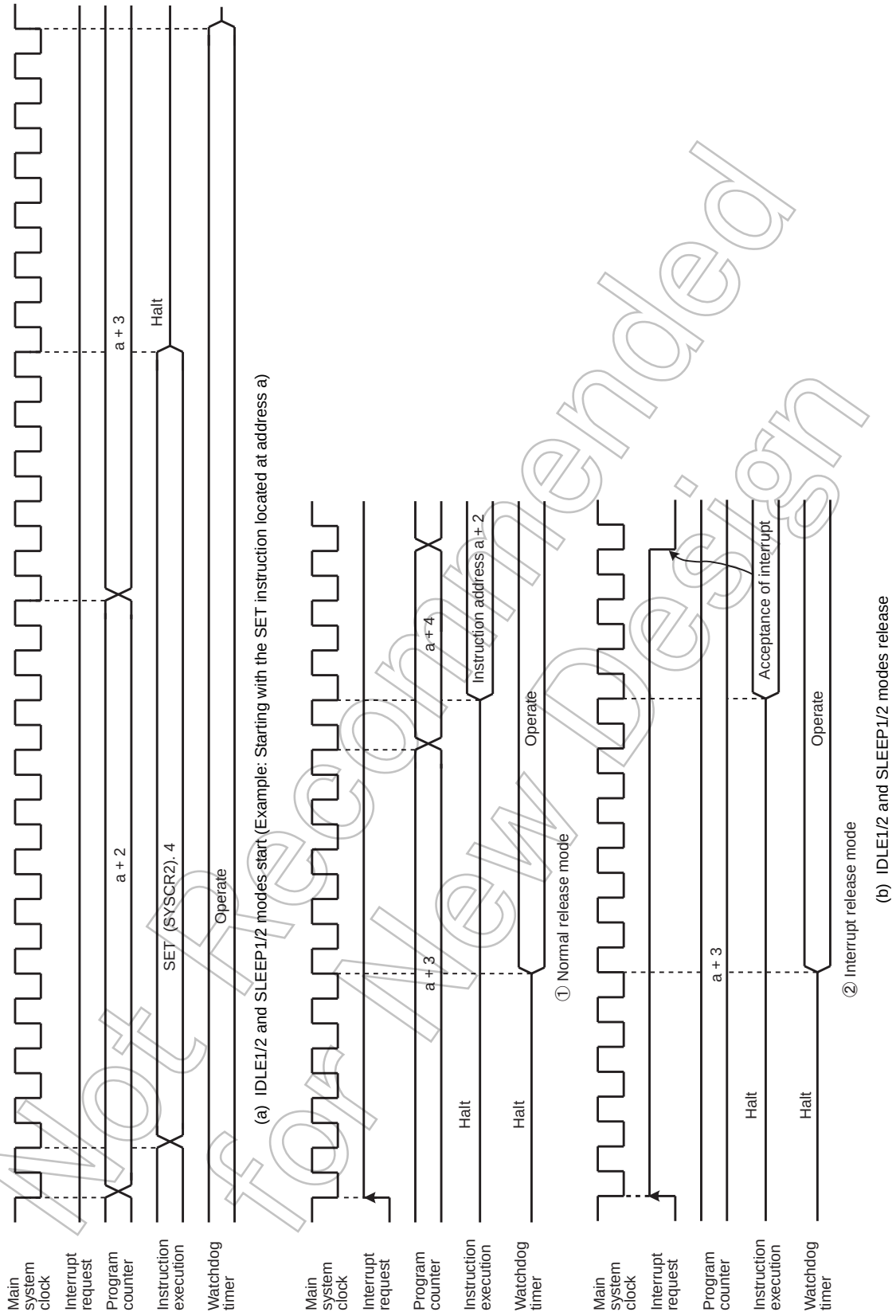


Figure 2-11 IDLE1/2 and SLEEP1/2 Modes Start/Release

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.

Not Recommended for New Design

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.

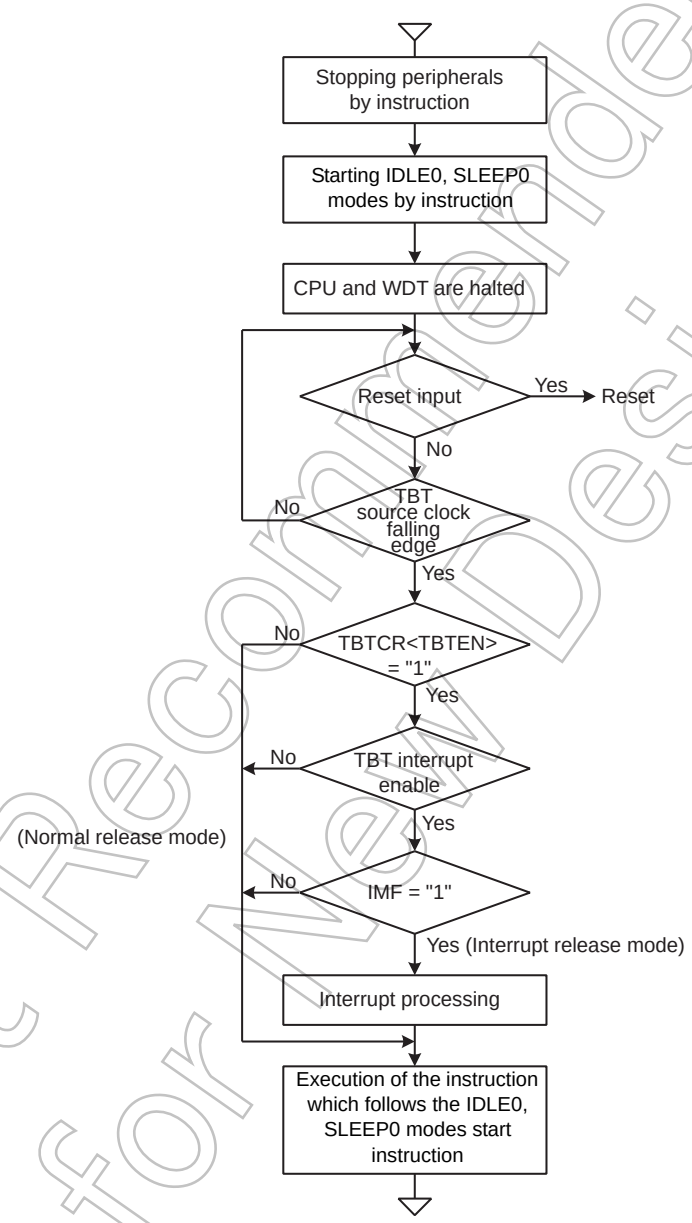


Figure 2-12 IDLE0 and SLEEP0 Modes

Not Recommended for New Design

- 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・EF6・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・EF6・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.

Not Recommended for New Design

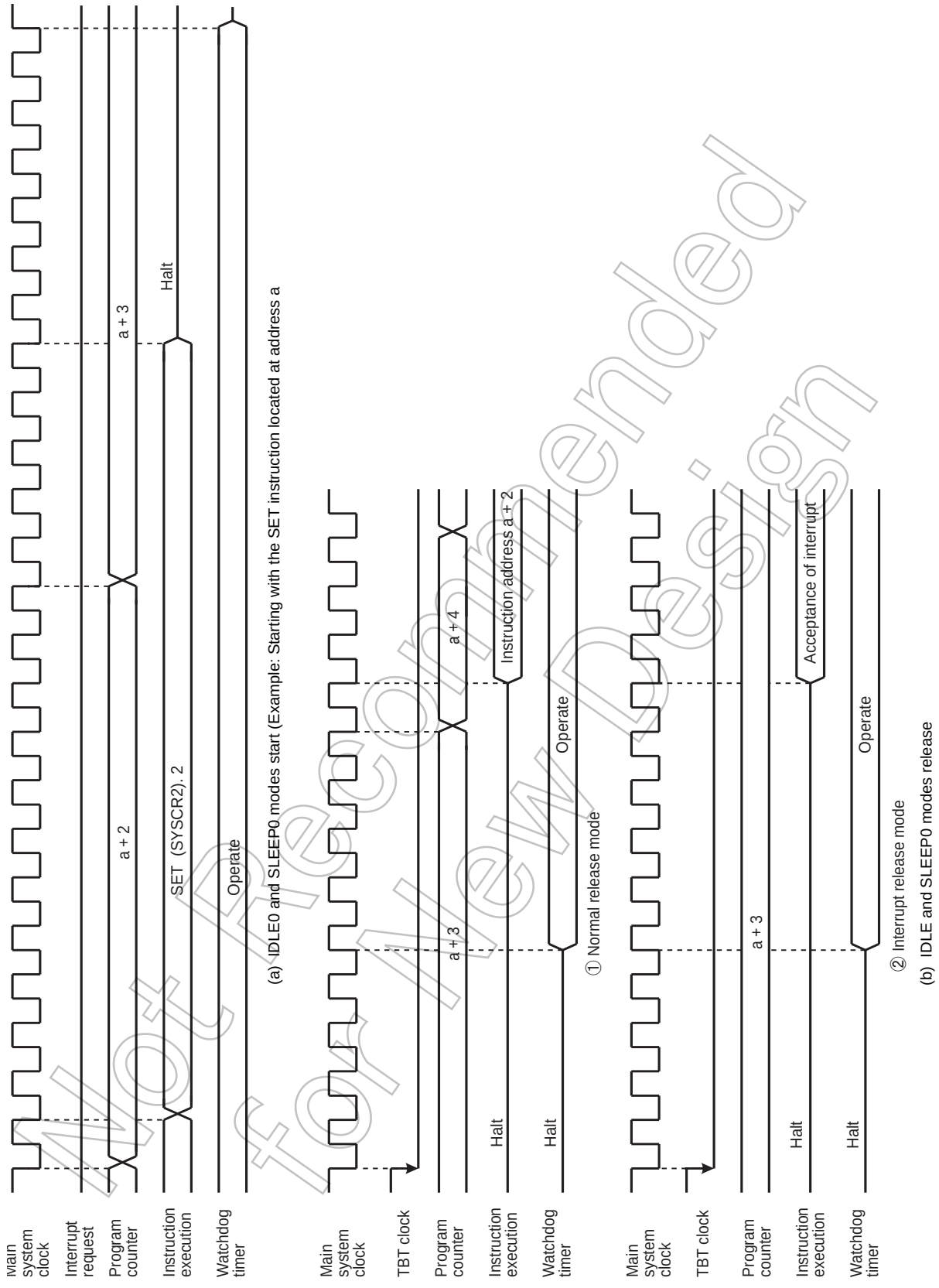


Figure 2-13 IDLE0 and SLEEP0 Modes Start/Release

2.2.4.4 SLOW mode

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

Not Recommended for New Design

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

(1) Switching from NORMAL2 mode to SLOW1 mode

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

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

Example 1 :Switching from NORMAL2 mode to SLOW1 mode.

```
        SET     (SYSCR2). 5     ; SYSCR2<SYSCK> ← 1 (Switches the main system clock to the low-frequency clock for SLOW2)
        CLR     (SYSCR2). 7     ; SYSCR2<XEN> ← 0 (Turns off high-frequency oscillation)
```

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

```
        SET     (SYSCR2). 6     ; SYSCR2<XTEN> ← 1
        LD      (TC3CR), 43H    ; Sets mode for TC4, 3 (16-bit mode, fs for source)
        LD      (TC4CR), 05H    ; Sets warming-up counter mode
        LDW     (TTREG3), 8000H ; Sets warm-up time (Depend on oscillator accompanied)
        DI                      ; IMF ← 0
        SET     (EIRH). 1       ; 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
```

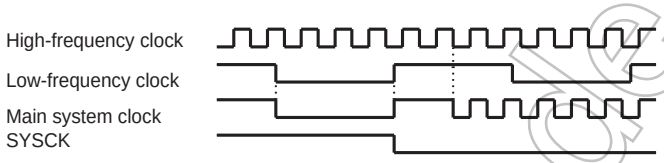
Not Recommended for New Design

(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). 1           ; 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
```

Not Recommended for New Design

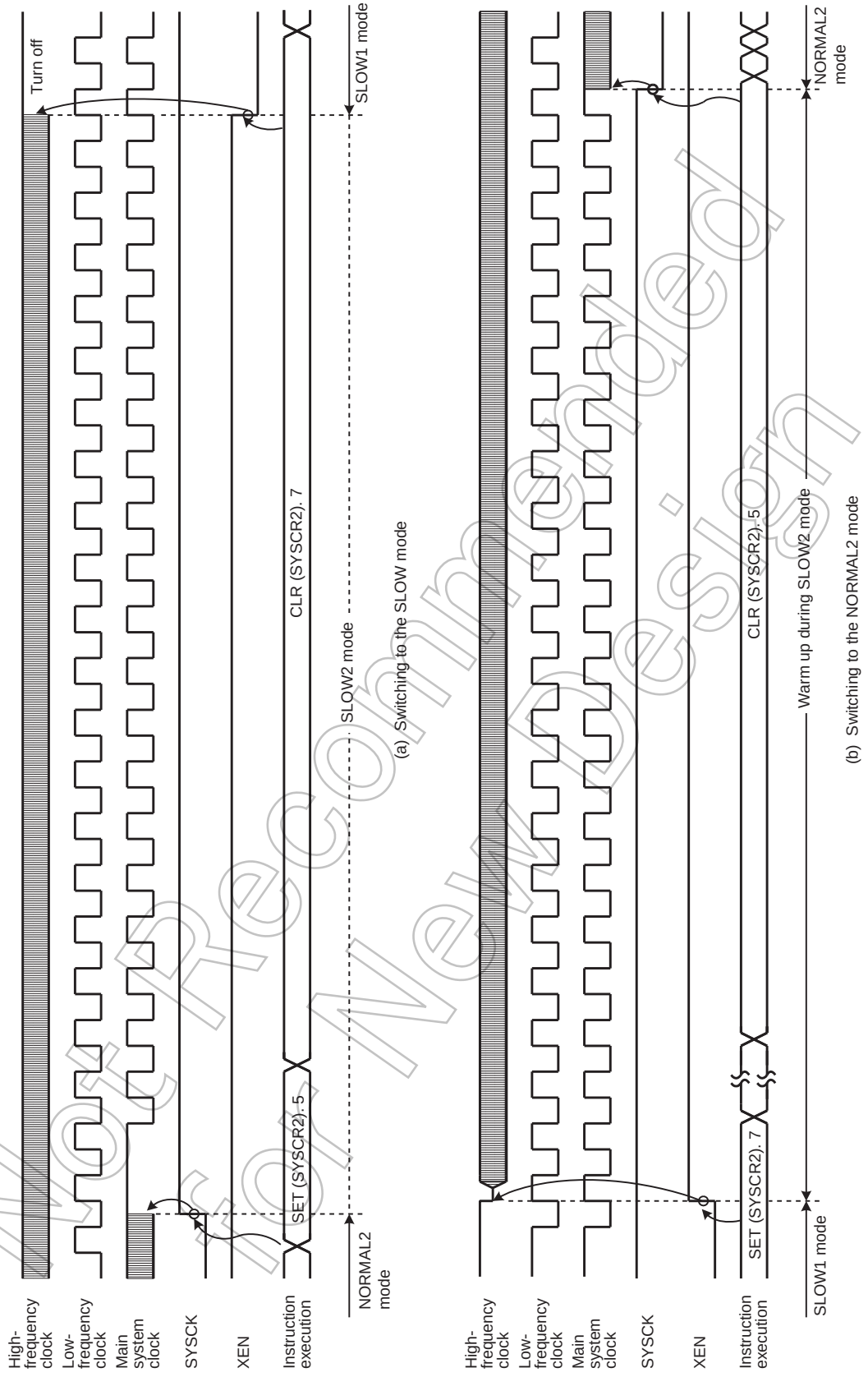


Figure 2-14 Switching between the NORMAL2 and SLOW Modes

Not Recommended for New Design

2.3 Reset Circuit

The TMP86FH46ANG has four types of reset generation procedures: An external reset input, an address trap reset, a watchdog timer reset and a system clock reset. Of these reset, the address trap reset, the watchdog timer and the system clock reset are a malfunction reset. When the malfunction reset request is detected, reset occurs during the maximum 24/fc[s] (The $\overline{\text{RESET}}$ pin outputs "L" level).

The malfunction reset circuit such as watchdog timer reset, address trap reset and system clock reset is not initialized when power is turned on. Therefore, reset may occur during maximum 24/fc[s] (1.5μs at 16.0 MHz) when power is turned on. $\overline{\text{RESET}}$ pin outputs "L" level during maximum 24/fc[s] (1.5μs at 16.0MHz).

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

Table 2-3 Initializing Internal Status by Reset Action

On-chip Hardware		Initial Value	On-chip Hardware	Initial Value
Program counter	(PC)	(FFFEH)	Prescaler and divider of timing generator	0
Stack pointer	(SP)	Not initialized		
General-purpose registers (W, A, B, C, D, E, H, L, IX, IY)		Not initialized		
Jump status flag	(JF)	Not initialized	Watchdog timer	Enable
Zero flag	(ZF)	Not initialized	Output latches of I/O ports	Refer to I/O port circuitry
Carry flag	(CF)	Not initialized		
Half carry flag	(HF)	Not initialized		
Sign flag	(SF)	Not initialized		
Overflow flag	(VF)	Not initialized		
Interrupt master enable flag	(IMF)	0		
Interrupt individual enable flags	(EF)	0	Control registers	Refer to each of control register
Interrupt latches	(IL)	0		
			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/fc [s]) with the power supply voltage within the operating voltage range and oscillation stable, a reset is applied and the internal state is initialized.

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

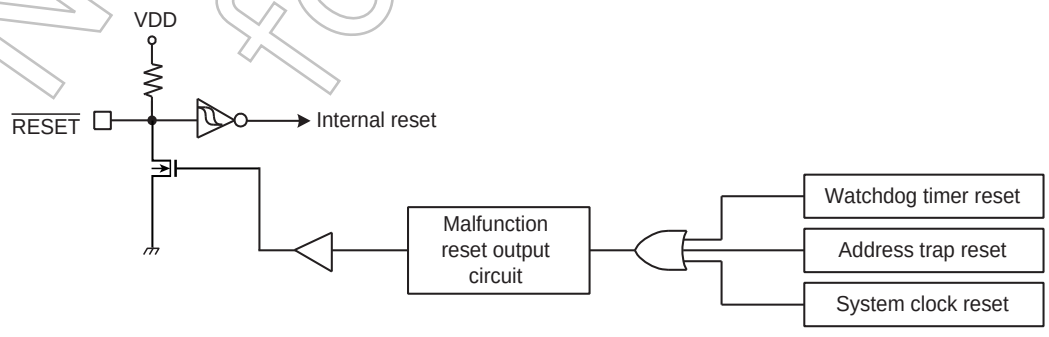


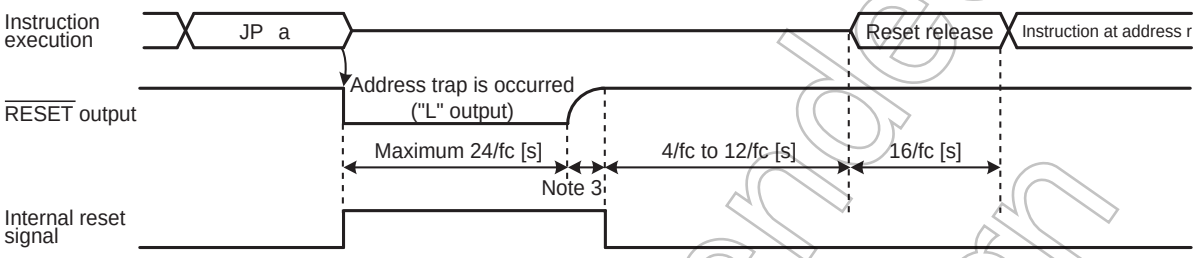
Figure 2-15 Reset Circuit

Not Recommended for New Design

2.3.2 Address trap reset

If the CPU should start looping for some cause such as noise and an attempt be made to fetch an instruction from the on-chip RAM (when WDTCR1<ATAS> is set to “1”), DBR or the SFR area, address trap reset will be generated. The reset time is maximum 24/fc[s] (1.5μs at 16.0 MHz). Then, the $\overline{\text{RESET}}$ pin outputs "L" level during maximum 24/fc[s].

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



Note 1: Address “a” is in the SFR, DBR or on-chip RAM (WDTCR1<ATAS> = “1”) space.
Note 2: During reset release, reset vector “r” is read out, and an instruction at address “r” is fetched and decoded.
Note 3: Varies on account of external condition: voltage or capacitance

Figure 2-16 Address Trap Reset

2.3.3 Watchdog timer reset

Refer to Section “Watchdog Timer”.

2.3.4 System clock reset

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

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

The reset time is maximum 24/fc (1.5 μs at 16.0 MHz). Then, the $\overline{\text{RESET}}$ pin outputs "L" level during maximum 24/fc[s] (1.5μs at 16.0MHz).

Not Recommended for New Design

3. Interrupt Control Circuit

The TMP86FH46ANG has a total of 18 interrupt sources excluding reset, of which 2 source levels are multiplexed. Interrupts can be nested with priorities. Four of the internal interrupt sources are non-maskable while the rest are maskable.

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

Interrupt Factors		Enable Condition	Interrupt Latch	Vector Address	Priority
Internal/External	(Reset)	Non-maskable	-	FFFE	1
Internal	INTSWI (Software interrupt)	Non-maskable	-	FFFC	2
Internal	INTUNDEF (Executed the undefined instruction interrupt)	Non-maskable	-	FFFC	2
Internal	INTATRAP (Address trap interrupt)	Non-maskable	IL2	FFFA	2
Internal	INTWDT (Watchdog timer interrupt)	Non-maskable	IL3	FFF8	2
External	$\overline{\text{INT0}}$	IMF・EF4 = 1, INT0EN = 1	IL4	FFF6	5
External	INT1	IMF・EF5 = 1	IL5	FFF4	6
Internal	INTTBT	IMF・EF6 = 1	IL6	FFF2	7
Internal	INTTC1	IMF・EF7 = 1	IL7	FFF0	8
External	INT2	IMF・EF8 = 1	IL8	FFEE	9
Internal	INTTC4	IMF・EF9 = 1	IL9	FFEC	10
Internal	INTTC3	IMF・EF10 = 1	IL10	FFEA	11
External	INT3	IMF・EF11 = 1	IL11	FFE8	12
Internal	INTSIO	IMF・EF12 = 1	IL12	FFE6	13
Internal	INTRXD	IMF・EF13 = 1	IL13	FFE4	14
External	INT4	IMF・EF14 = 1, IL14ER = 0	IL14	FFE2	15
Internal	INTTXD	IMF・EF14 = 1, IL14ER = 1			
External	$\overline{\text{INT5}}$	IMF・EF15 = 1, IL15ER = 0	IL15	FFE0	16
Internal	INTADC	IMF・EF15 = 1, IL15ER = 1			

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

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

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

3.1 Interrupt latches (IL15 to IL2)

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

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

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

Not Recommended for New Design

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

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

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

Example 1 :Clears interrupt latches

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

Example 2 :Reads interrupt latches

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

Example 3 :Tests interrupt latches

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

3.2 Interrupt enable register (EIR)

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

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

3.2.1 Interrupt master enable flag (IMF)

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

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

3.2.2 Individual interrupt enable flags (EF15 to EF4)

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

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

Not Recommended for New Design

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

(Initial value: 00000000 000000**)

ILH,ILL (003DH, 003CH)	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IL15	IL14	IL13	IL12	IL11	IL10	IL9	IL8	IL7	IL6	IL5	IL4	IL3	IL2		
	ILH (003DH)								ILL (003CH)							

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

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

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

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

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

Interrupt Enable Registers

(Initial value: 00000000 0000***0)

EIRH,EIRL (003BH, 003AH)	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EF15	EF14	EF13	EF12	EF11	EF10	EF9	EF8	EF7	EF6	EF5	EF4				IMF
	EIRH (003BH)								EIRL (003AH)							

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

Note 1: *: Don't care

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

Not Recommended for New Design

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".

Not Recommended for New Design

3.3 Interrupt Source Selector (INTSEL)

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

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

1. INT4 and INTTXD share the interrupt source level whose priority is 15.
2. $\overline{\text{INT5}}$ and INTADC share the interrupt source level whose priority is 16.

Interrupt source selector

INTSEL (003EH)	7	6	5	4	3	2	1	0	(Initial value: **** **00)
	-	-	-	-	-	-	IL14ER	IL15ER	

IL14ER	Selects INT4 or INTTXD	0: INT4 1: INTTXD	R/W
IL15ER	Selects $\overline{\text{INT5}}$ or INTADC	0: $\overline{\text{INT5}}$ 1: INTADC	R/W

3.4 Interrupt Sequence

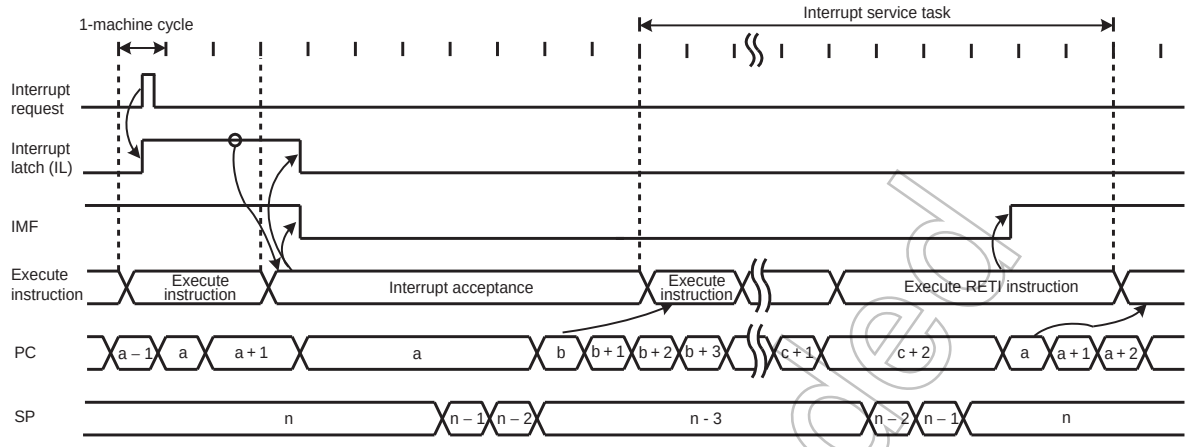
An interrupt request, which raised interrupt latch, is held, until interrupt is accepted or interrupt latch is cleared to "0" by resetting or an instruction. Interrupt acceptance sequence requires 8 machine cycles (2 μs @16 MHz) after the completion of the current instruction. The interrupt service task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

3.4.1 Interrupt acceptance processing is packaged as follows.

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

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

Not Recommended for New Design



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/fc [s] or 38/fs [s] at maximum (If the interrupt latch is set at the first machine cycle on 10 cycle instruction) to start interrupt acceptance processing since its interrupt latch is set.

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

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

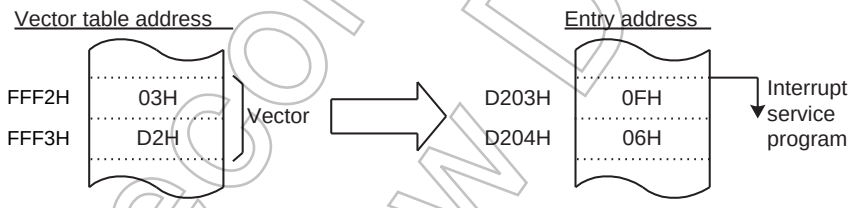


Figure 3-2 Vector table address,Entry address

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

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

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

3.4.2 Saving/restoring general-purpose registers

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

Not Recommended for New Design

3.4.2.1 Using PUSH and POP instructions

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

Example :Save/store register using PUSH and POP instructions

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

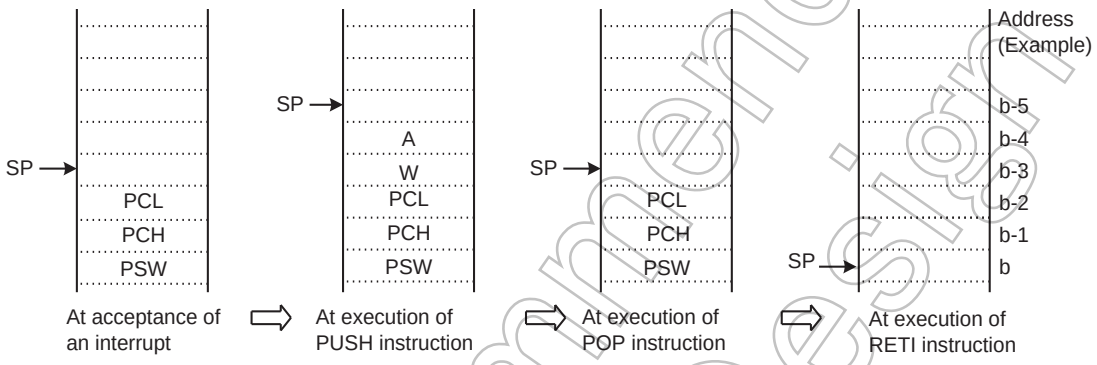


Figure 3-3 Save/store register using PUSH and POP instructions

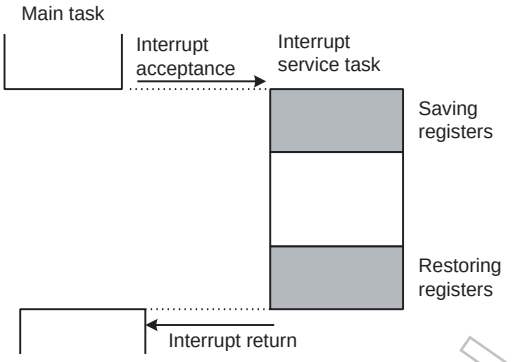
3.4.2.2 Using data transfer instructions

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

Example :Save/store register using data transfer instructions

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

Not Recommended for New Design



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

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

3.4.3 Interrupt return

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

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

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

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

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

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

Example 2 :Restarting without returning interrupt
(In this case, PSW (Includes IMF) before interrupt acceptance is discarded.)

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

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

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

Not Recommended for New Design

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

3.5 Software Interrupt (INTSW)

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

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

3.5.1 Address error detection

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

3.5.2 Debugging

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

3.6 Undefined Instruction Interrupt (INTUNDEF)

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

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

3.7 Address Trap Interrupt (INTATRAP)

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

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

3.8 External Interrupts

The TMP86FH46ANG has 6 external interrupt inputs. These inputs are equipped with digital noise reject circuits (Pulse inputs of less than a certain time are eliminated as noise).

Edge selection is also possible with INT1 to INT4. The $\overline{\text{INT0}}$/P00 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 $\overline{\text{INT0}}$/P00 pin function selection are performed by the external interrupt control register (EINTCR).

Not Recommended for New Design

Source	Pin	Enable Conditions	Release Edge (level)	Digital Noise Reject
INT0	$\overline{INT0}$	IMF × EF4 × INT0EN=1	Falling edge	Pulses of less than 2/fc [s] are eliminated as noise. Pulses of 7/fc [s] or more are considered to be signals. In the SLOW or the SLEEP mode, pulses of less than 1/fs [s] are eliminated as noise. Pulses of 3.5/fs [s] or more are considered to be signals.
INT1	INT1	IMF × EF5 = 1	Falling edge or Rising edge	Pulses of less than 15/fc or 63/fc [s] are eliminated as noise. Pulses of 49/fc or 193/fc [s] or more are considered to be signals. In the SLOW or the SLEEP mode, pulses of less than 1/fs [s] are eliminated as noise. Pulses of 3.5/fs [s] or more are considered to be signals.
INT2	INT2	IMF × EF8 = 1	Falling edge or Rising edge	Pulses of less than 7/fc [s] are eliminated as noise. Pulses of 25/fc [s] or more are considered to be signals. In the SLOW or the SLEEP mode, pulses of less than 1/fs [s] are eliminated as noise. Pulses of 3.5/fs [s] or more are considered to be signals.
INT3	INT3	IMF × EF11 = 1	Falling edge or Rising edge	Pulses of less than 7/fc [s] are eliminated as noise. Pulses of 25/fc [s] or more are considered to be signals. In the SLOW or the SLEEP mode, pulses of less than 1/fs [s] are eliminated as noise. Pulses of 3.5/fs [s] or more are considered to be signals.
INT4	INT4	IMF + EF14 = 1 and IL14ER=0	Falling edge, Rising edge, Falling and Rising edge or H level	Pulses of less than 7/fc [s] are eliminated as noise. Pulses of 25/fc [s] or more are considered to be signals. In the SLOW or the SLEEP mode, pulses of less than 1/fs [s] are eliminated as noise. Pulses of 3.5/fs [s] or more are considered to be signals.
INT5	$\overline{INT5}$	IMF × EF15 = 1 and IL15ER=0	Falling edge	Pulses of less than 2/fc [s] are eliminated as noise. Pulses of 7/fc [s] or more are considered to be signals. In the SLOW or the SLEEP mode, pulses of less than 1/fs [s] are eliminated as noise. Pulses of 3.5/fs [s] or more are considered to be signals.

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

Note 2: When INT0EN = "0", IL4 is not set even if a falling edge is detected on the $\overline{INT0}$ pin input.

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

External Interrupt Control Register

EINTCR	7	6	5	4	3	2	1	0
(0037H)	INT1NC	INT0EN	INT4ES		INT3ES	INT2ES	INT1ES	

(Initial value: 0000 000*)

INT1NC	Noise reject time select	0: Pulses of less than 63/fc [s] are eliminated as noise 1: Pulses of less than 15/fc [s] are eliminated as noise	R/W
INT0EN	P00/$\overline{INT0}$ pin configuration	0: P00 input/output port 1: $\overline{INT0}$ pin (Port P00 should be set to an input mode)	R/W
INT4 ES	INT4 edge select	00: Rising edge 01: Falling edge 10: Rising edge and Falling edge 11: H level	R/W
INT3 ES	INT3 edge select	0: Rising edge 1: Falling edge	R/W
INT2 ES	INT2 edge select	0: Rising edge 1: Falling edge	R/W
INT1 ES	INT1 edge select	0: Rising edge 1: Falling edge	R/W

Not Recommended for New Design

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

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

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

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.

Not Recommended for New Design

Not Recommended for New Design

4. Special Function Register (SFR)

The TMP86FH46ANG 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 TMP86FH46ANG.

4.1 SFR

Address	Read	Write
0000H	P0DR	
0001H	P1DR	
0002H	P2DR	
0003H	P3DR	
0004H	P4DR	
0005H	Reserved	
0006H	Reserved	
0007H	Reserved	
0008H	P0PRD	-
0009H	Reserved	
000AH	P2PRD	-
000BH	Reserved	
000CH	Reserved	
000DH	P1CR	
000EH	P3CR	
000FH	P4CR	
0010H	TC1DRAL	
0011H	TC1DRAH	
0012H	TC1DRBL	
0013H	TC1DRBH	
0014H	TC1CR	
0015H	Reserved	
0016H	TC3CR	
0017H	TC4CR	
0018H	TTREG3	
0019H	TTREG4	
001AH	PWREG3	
001BH	PWREG4	
001CH	ADCCR1	
001DH	ADCCR2	
001EH	ADCDR2	-
001FH	ADCDR1	-
0020H	UARTSR	UARTCR1
0021H	-	UARTCR2
0022H	RDBUF	TDBUF
0023H	Reserved	
0024H	Reserved	
0025H	Reserved	

Not Recommended for New Design

Address	Read	Write
0026H	SIOCR1	
0027H	SIOSR	
0028H	SIORDB	SIOTDB
0029H	Reserved	
002AH	Reserved	
002BH	Reserved	
002CH	Reserved	
002DH	Reserved	
002EH	Reserved	
002FH	Reserved	
0030H	Reserved	
0031H	-	STOPCR
0032H	Reserved	
0033H	Reserved	
0034H	-	WDTCR1
0035H	-	WDTCR2
0036H	TBTCR	
0037H	EINTCR	
0038H	SYSCR1	
0039H	SYSCR2	
003AH	EIRL	
003BH	EIRH	
003CH	ILL	
003DH	ILH	
003EH	INTSEL	
003FH	PSW	

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

Note 2: − ; Cannot be accessed.

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

Not Recommended for New Design

4.2 DBR

Address	Read	Write
0F80H	Reserved	
: :	: :	
0F9FH	Reserved	

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

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

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

Not Recommended for New Design

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.).

Not Recommended for New Design

5. 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).

5.1 Time Base Timer

5.1.1 Configuration

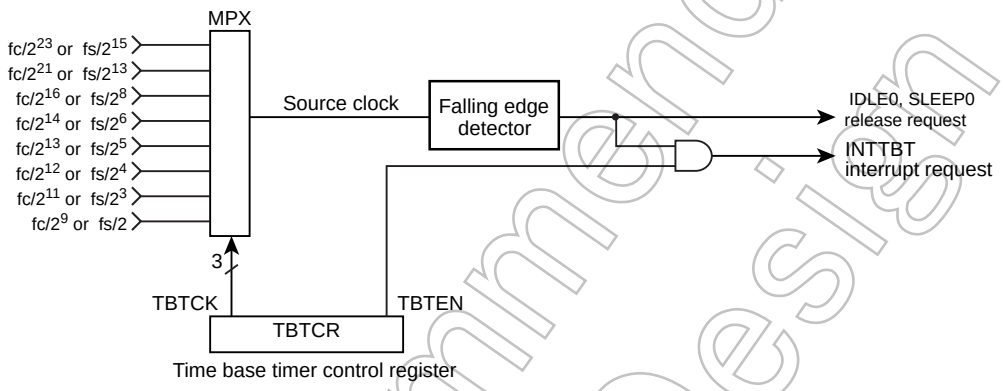


Figure 5-1 Time Base Timer configuration

5.1.2 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	TBTCK			(Initial Value: 0000 0000)

TBTEN	Time Base Timer enable / disable	0: Disable 1: Enable				
TBTCK	Time Base Timer interrupt Frequency select : [Hz]		NORMAL1/2, IDLE1/2 Mode		SLOW1/2 SLEEP1/2 Mode	R/W
			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

Not Recommended for New Design

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

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

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

Table 5-1 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	-

5.1.3 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 5-2).

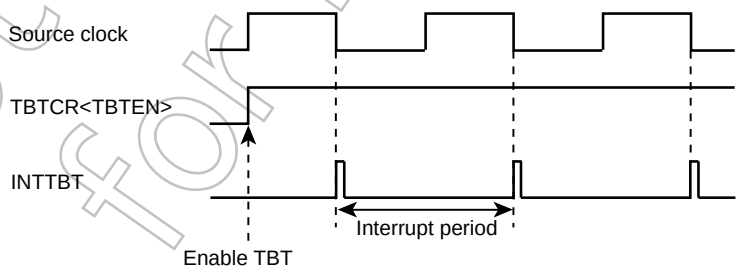


Figure 5-2 Time Base Timer Interrupt

Not Recommended for New Design

5.2 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.

5.2.1 Configuration

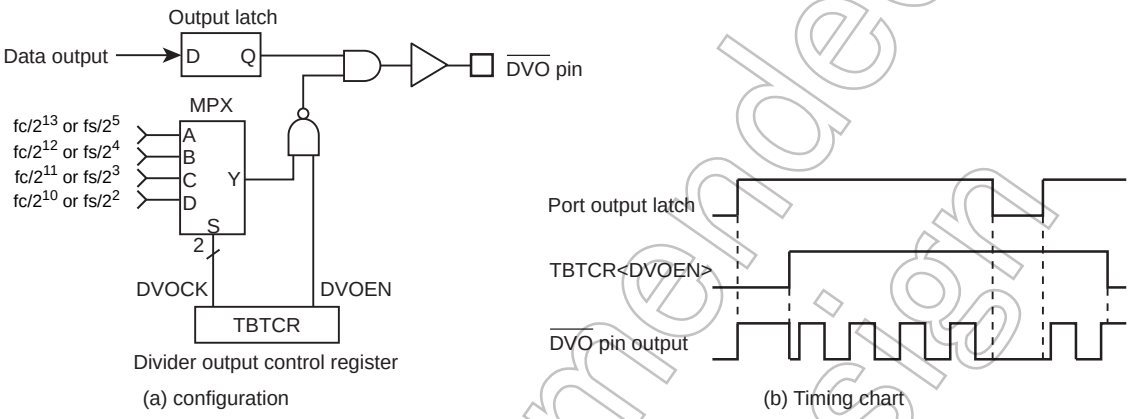


Figure 5-3 Divider Output

5.2.2 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
			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 disable(DVOEN="0"), do not change the setting of the divider output frequency.

Not Recommended for New Design

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

```
                Setting port

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

Table 5-2 Divider Output Frequency (Example : fc = 16.0 MHz, fs = 32.768 kHz)

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

Not Recommended for New Design

6. Watchdog Timer (WDT)

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

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

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

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

6.1 Watchdog Timer Configuration

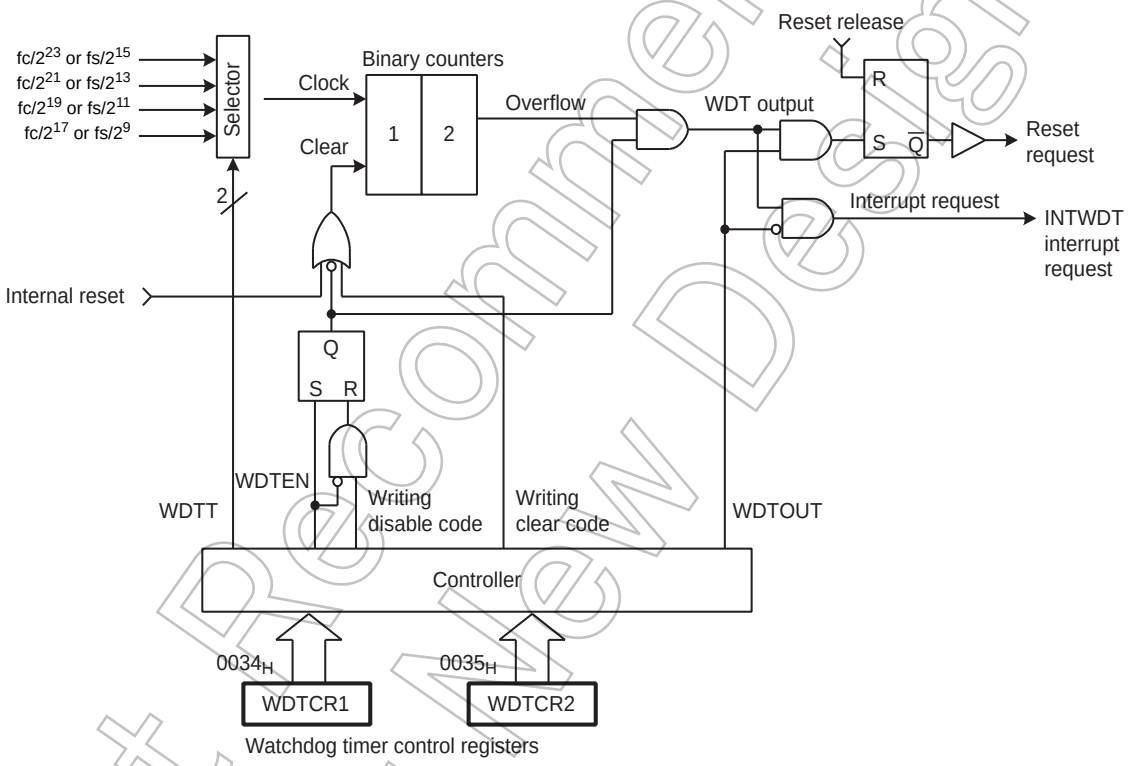


Figure 6-1 Watchdog Timer Configuration

Not Recommended for New Design

6.2 Watchdog Timer Control

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

6.2.1 Malfunction Detection Methods Using the Watchdog Timer

The CPU malfunction is detected, as shown below.

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

If the CPU malfunctions such as endless loops or the deadlock conditions occur for some reason, the watchdog timer output is activated by the binary-counter overflow unless the binary counters are cleared. When WDTCR1<WDTOUT> is set to "1" at this time, the reset request is generated and the $\overline{\text{RESET}}$ pin outputs a low-level signal, 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}/fc [s], and resetting the CPU malfunction detection

```
        LD      (WDTCR2), 4EH           : Clears the binary counters.
        LD      (WDTCR1), 00001101B     : WDTT ← 10, WDTOUT ← 1
        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.
Within 3/4 of WDT
detection time  :
                :
        LD      (WDTCR2), 4EH           : Clears the binary counters.
```

Not Recommended for New Design

Watchdog Timer Control Register 1

WDTCR1 (0034H)	7	6	5	4	3	2	1	0
			(ATAS)	(ATOUT)	WDTEN	WDTT		WDTOUT

(Initial value: **11 1001)

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 “6.2.3 Watchdog Timer Disable”.

Watchdog Timer Control Register 2

WDTCR2 (0035H)	7	6	5	4	3	2	1	0

(Initial value: **** ****)

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

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

Note 2: *: Don’t care

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

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

6.2.2 Watchdog Timer Enable

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

Not Recommended for New Design

6.2.3 Watchdog Timer Disable

To disable the watchdog timer, set the register in accordance with the following procedures. Setting the register in other procedures causes a malfunction of the 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 6-1 Watchdog Timer Detection Time (Example: fc = 16.0 MHz, fs = 32.768 kHz)

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

6.2.4 Watchdog Timer Interrupt (INTWDT)

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

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

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

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

Example :Setting watchdog timer interrupt

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

Not Recommended for New Design

6.2.5 Watchdog Timer Reset

When a binary-counter overflow occurs while WDTCR1<WDTOUT> is set to "1", a watchdog timer reset request is generated. When a watchdog timer reset request is generated, the $\overline{\text{RESET}}$ pin outputs a low-level signal and the internal hardware is reset. The reset time is maximum 24/fc [s] (1.5 μs @ fc = 16.0 MHz).

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

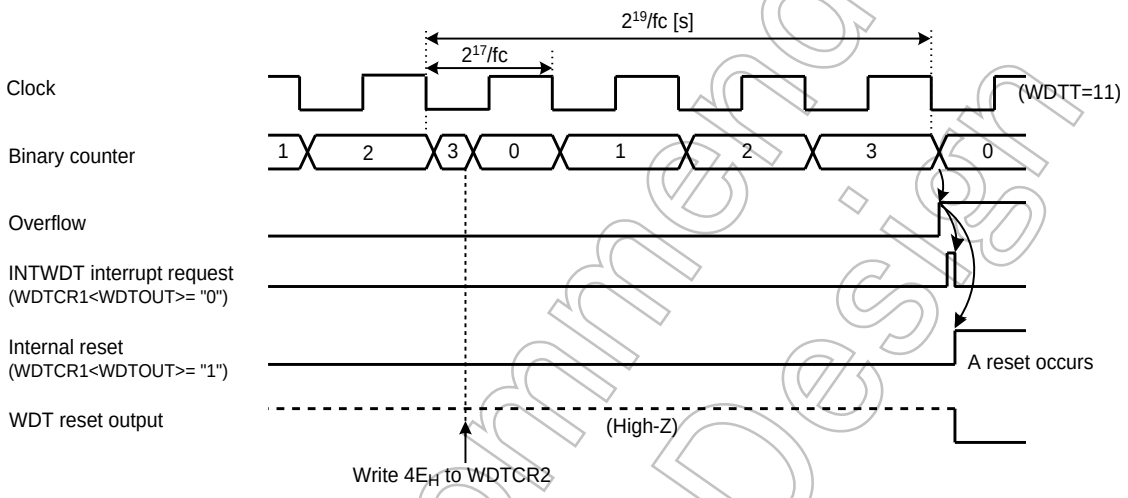


Figure 6-2 Watchdog Timer Interrupt/Reset

Not Recommended for New Design

6.3 Address Trap

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

Watchdog Timer Control Register 1

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

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

Watchdog Timer Control Register 2

WDTCR2 (0035H)	7	6	5	4	3	2	1	0	
									(Initial value: **** ****)

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

6.3.1 Selection of Address Trap in Internal RAM (ATAS)

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

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

6.3.2 Selection of Operation at Address Trap (ATOUT)

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

6.3.3 Address Trap Interrupt (INTATRAP)

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

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

When an address trap interrupt is generated while the other interrupt including 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.

Not Recommended for New Design

6.3.4 Address Trap Reset

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

When an address trap reset request is generated, the $\overline{\text{RESET}}$ pin outputs a low-level signal and the internal hardware is reset. The reset time is maximum 24/fc [s] (1.5 μs @ fc = 16.0 MHz).

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

Not Recommended for New Design

Not Recommended for New Design

7. I/O Ports

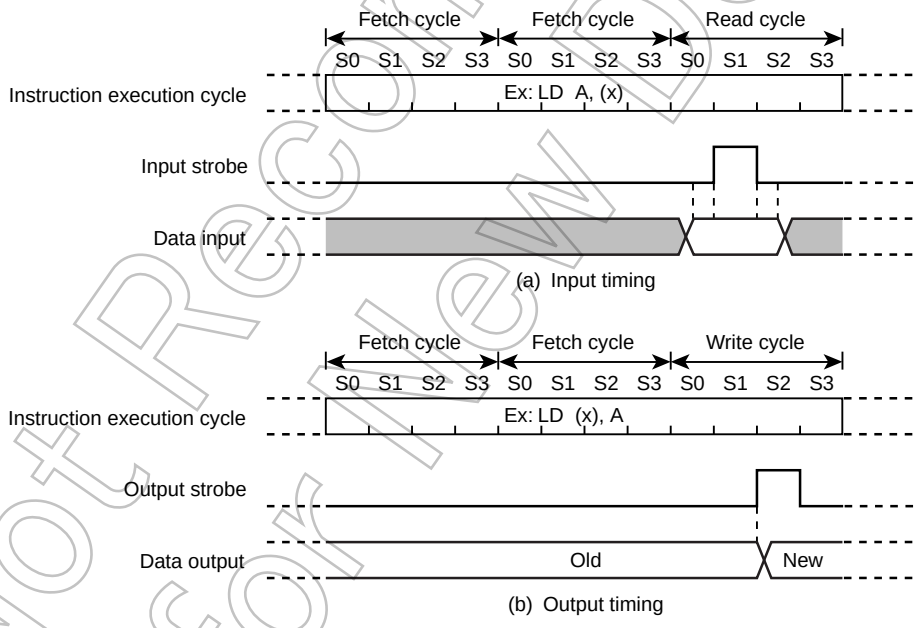
The TMP86FH46ANG have 5 parallel input/output ports (33 pins) as follows.

	Primary Function	Secondary Functions
Port P0	8-bit I/O port	External interrupt input, Serial PROM mode control input, serial and timer/counter input/output.
Port P1	6-bit I/O port	External interrupt input, and timer/counter input/output.
Port P2	3-bit I/O port	Low-frequency resonator connections, external interrupt input, and STOP mode release signal input.
Port P3	8-bit I/O port	Analog input, and STOP mode release signal input.
Port P4	8-bit I/O port	

Each output port contains a latch, which holds the output data. All input ports do not have latches, so the external input data should be externally held until the input data is read from outside or reading should be performed several timer before processing. Figure 7-1 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 7-1 Input/Output Timing (Example)

Not Recommended for New Design

7.1 Port P0 (P07 to P00)

Port P0 is an 8-bit input/output port which is also used as an external interrupt input, Serial PROM mode control input, serial interface input/output and timer/counter input/output.

When used as an input port or a secondary function pins, the respective output latch (P0DR) should be set to “1”. When used as an output port, the respective P0DR bit should be set to “0”. During reset, the output latch is initialized to “1”.

P0 port output latch (P0DR) and P0 port terminal input (P0PRD) are located on their respective address.

When read the output latch data, the P0DR should be read and when read the terminal input data, the P0PRD register should be read.

P00 port ($\overline{\text{INT0}}$) can be configured as either an I/O port or as external interrupt input with INT0EN (bit in EINTCR). During reset, P00 port ($\overline{\text{INT0}}$) is configured as an input port.

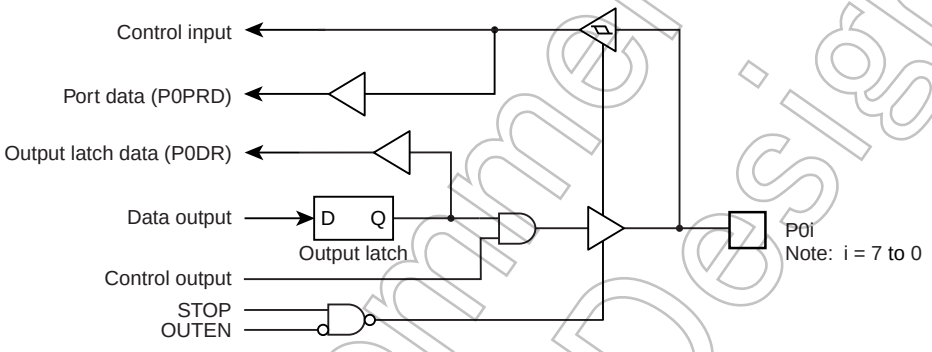


Figure 7-2 Port 0

P0DR (0000H) R/W

7	6	5	4	3	2	1	0
P07 INT4	P06 $\overline{\text{SCK}}$	P05 SI	P04 SO	P03 TXD	P02 RXD BOOT	P01 $\overline{\text{PWM4}}$ TC4 $\overline{\text{PDO4}}$ $\overline{\text{PPG4}}$	P00 $\overline{\text{INT0}}$

(Initial value: 1111 1111)

P0PRD (0008H) Read only

7	6	5	4	3	2	1	0
P07	P06	P05	P04	P03	P02	P01	P00

Not Recommended for New Design

7.2 Port P1 (P15 to P10)

Port P1 is an 6-bit input/output port which can be configured as an input or an output in one-bit unit under software control. Input/output mode is specified by the corresponding bit in the port P1 input/output control register (P1CR). Port P1 is configured as an input if its corresponding P1CR bit is cleared to "0", and as an output if its corresponding P1CR bit is set to "1".

During reset, the P1CR is initialized to "0" and port P1 is input mode. The P1 output latches are also initialized to "0".

Port P1 is also used as an external interrupt input, a timer/counter input/output, and a divider output. When used as an input port, an external interrupt input or a timer/counter input, the corresponding bit of P1CR is cleared to "0".

When used as a timer/counter output or divider output, the corresponding bit of P1CR is set to "1" and beforehand the corresponding output latch should be set to "1". Data can be written into the output latch regardless of P1CR contents, therefore initial output data should be written into the output latch before setting P1CR.

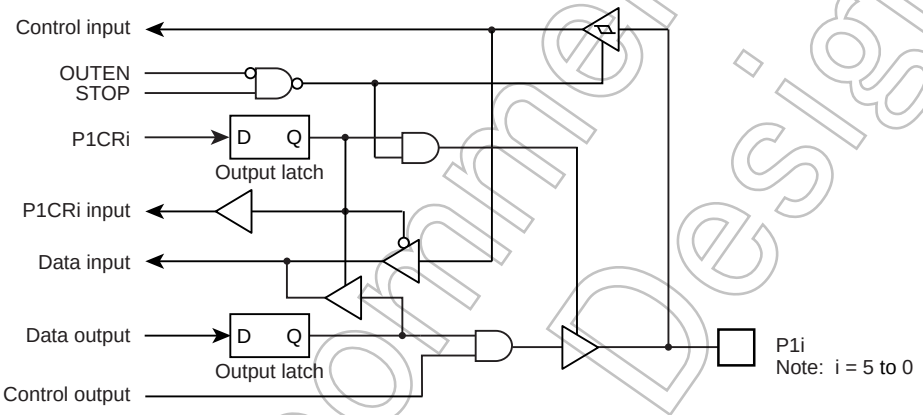


Figure 7-3 Port P1

	7	6	5	4	3	2	1	0	
P1DR (0001H) R/W			P15 INT3	P14 $\overline{PPG}$	P13 $\overline{DVO}$	P12 INT2 TC1	P11 INT1	P10 $\overline{PWM3}$ TC3 $\overline{PDO3}$	(Initial value: **00 0000)
P1CR (000DH)									(Initial value: **00 0000)

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

Note 1: Bit 7 and 6 in P1DR and P1CR are always read as undefined value.

Note 2: Please set the bit 7 and 6 of P1CR to "1" in an emulator.

Note 3: Ports set to the input mode read the pin states. Ports set to the output mode read the output latch. When input pin and output pin exist in port P1 together, the contents of the output latch which is specified as an input mode may be rewritten by executing the bit manipulation instructions.

Not Recommended for New Design

7.3 Port P2 (P22 to P20)

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

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

During reset, the P2DR is initialized to "1".

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

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

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

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

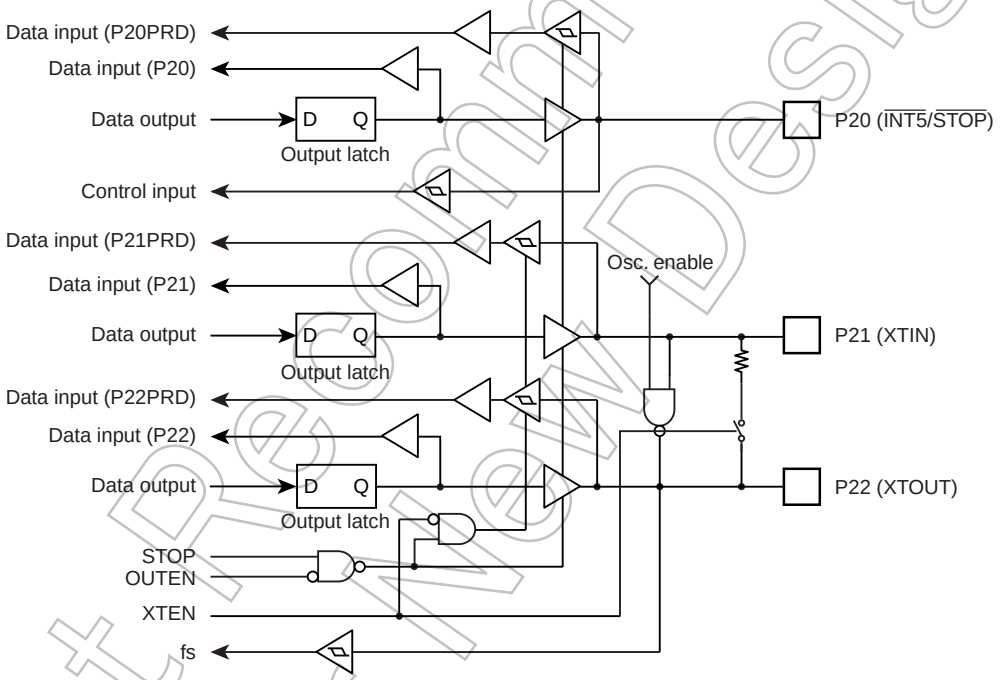


Figure 7-4 Port 2

	7	6	5	4	3	2	1	0	
P2DR (0002H) R/W						P22 XTOUT	P21 XTIN	P20 $\overline{INT5}$ $\overline{STOP}$	(Initial value: **** *111)

	7	6	5	4	3	2	1	0
P2PRD (000AH) Read only						P22	P21	P20

Not Recommended for New Design

7.4 Port P3 (P37 to P30)

Port P3 is an 8-bit input/output port which can be configured as an input or an output in one-bit unit under software control. Port P3 is also used as an analog input, key on wake up input. Input/output mode is specified by the corresponding bit in the port P3 input/output control register (P3CR), and ADCCR1<AINDS>. During reset, P3CR are initialized to "0" and ADCCR1<AINDS> is set to "1", therefore port P3 is configured as an input.

When used as an analog input, set an analog input channel to ADCCR1<SAIN> and clear ADCCR1<AINDS> to "0". When ADCCR1<AINDS> is "0", the pin which is specified as an analog input is used as analog input independent on the value of P3CR and P3DR.

When used as an input port or key on wake up input, the corresponding bit of P3CR is cleared to "0" without specifying as an analog input.

When the AD converter is enabled (ADCCR1<AINDS> is "0"), the data of port which is selected as an analog input is read "0". and the data of port which is not selected as an analog input is read "0" or "1", depend on the voltage level.

When used as an output port, the corresponding bit of P3CR is set to "1" without specifying as an analog input. Data can be written into the output latch regardless of P3CR contents, therefore initial output data should be written into the output latch before setting P3CR.

The pins not used as analog input can be used as an input/output port. But output instructions should not be executed to keep a precision. In addition, a variable signal should not be input to an adjacent port to the analog input during AD conversion.

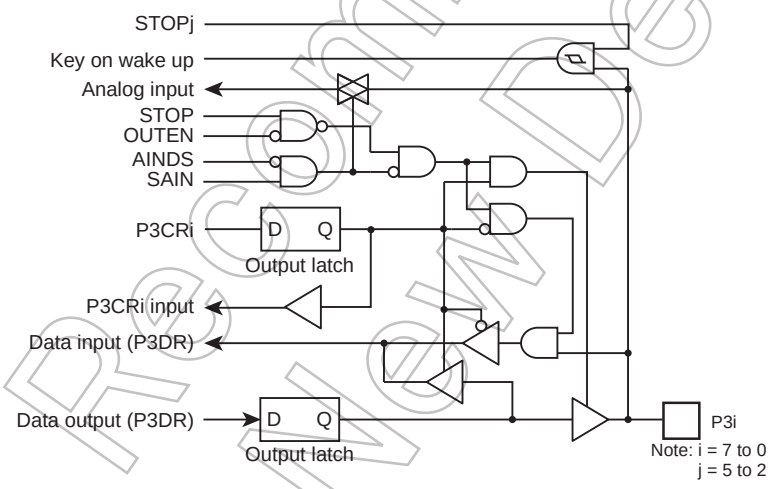


Figure 7-5 Port P3

	7	6	5	4	3	2	1	0	
P3DR (0003H) R/W	P37 AIN7 STOP5	P36 AIN6 STOP4	P35 AIN5 STOP3	P34 AIN4 STOP2	P33 AIN3	P32 AIN2	P31 AIN1	P30 AIN0	(Initial value: 0000 0000)

	7	6	5	4	3	2	1	0	
P3CR (000EH)									(Initial value: 0000 0000)

P3CR	I/O control (Specified for each bit)	0: Input mode 1: Output mode	R/W

Not Recommended for New Design

Note: Ports set to the input mode read the pin states. Ports set to the output mode read the output latch. When input pin and output pin exist in port P3 together, the contents of the output latch which is specified as an input mode may be rewritten by executing the bit manipulation instructions.

7.5 Port P4 (P47 to P40)

Port P4 is an 8-bit input/output port which can be configured as an input or an output in one-bit unit under software control. Input/output mode is specified by the corresponding bit in the port P4 input/output control register (P4CR). Port P4 is configured as an input if its corresponding P4CR bit is cleared to "0", and as an output if its corresponding P4CR bit is set to "1".

During reset, the P4CR is initialized to "0" and port P4 is input mode. The P4 output latches are also initialized to "0".

When used as an input port, the corresponding bit of P4CR is cleared to "0".

When used as an output port, the corresponding bit of P4CR is set to "1". Data can be written into the output latch regardless of P4CR contents, therefore initial output data should be written into the output latch before setting P4CR.

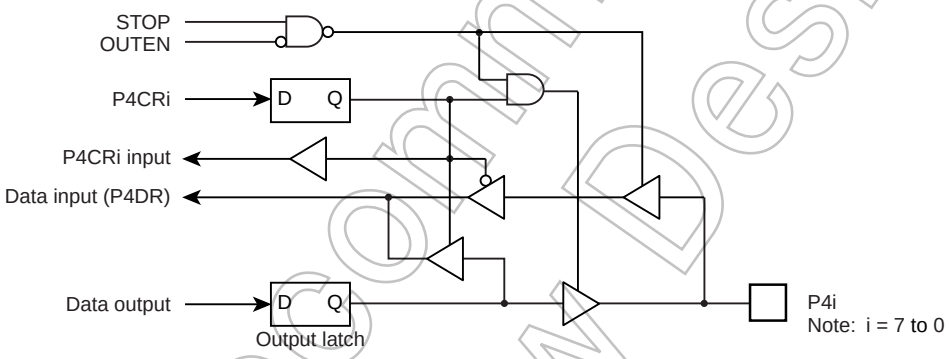
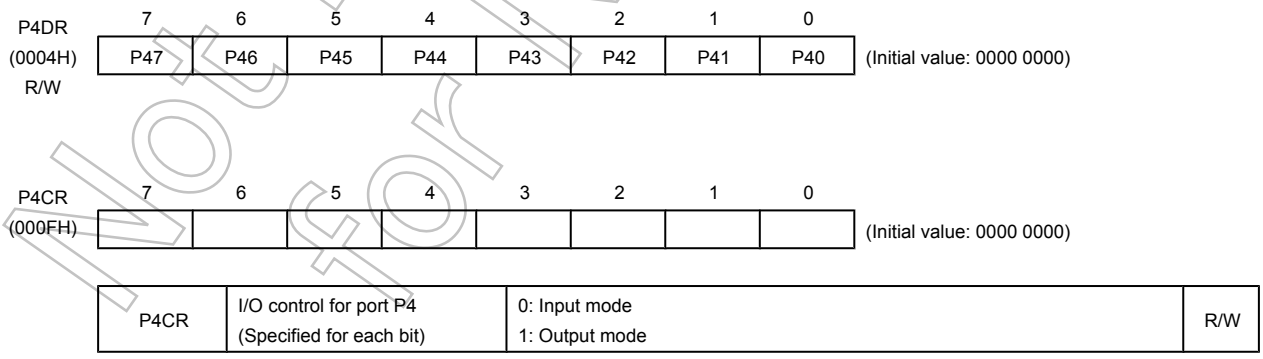


Figure 7-6 Port P4



Note: Ports set to the input mode read the pin states. Ports set to the output mode read the output latch. When input pin and output pin exist in port P4 together, the contents of the output latch which is specified as an input mode may be rewritten by executing the bit manipulation instructions.

Not Recommended for New Design

8. 16-Bit Timer/Counter 1 (TC1)

8.1 Configuration

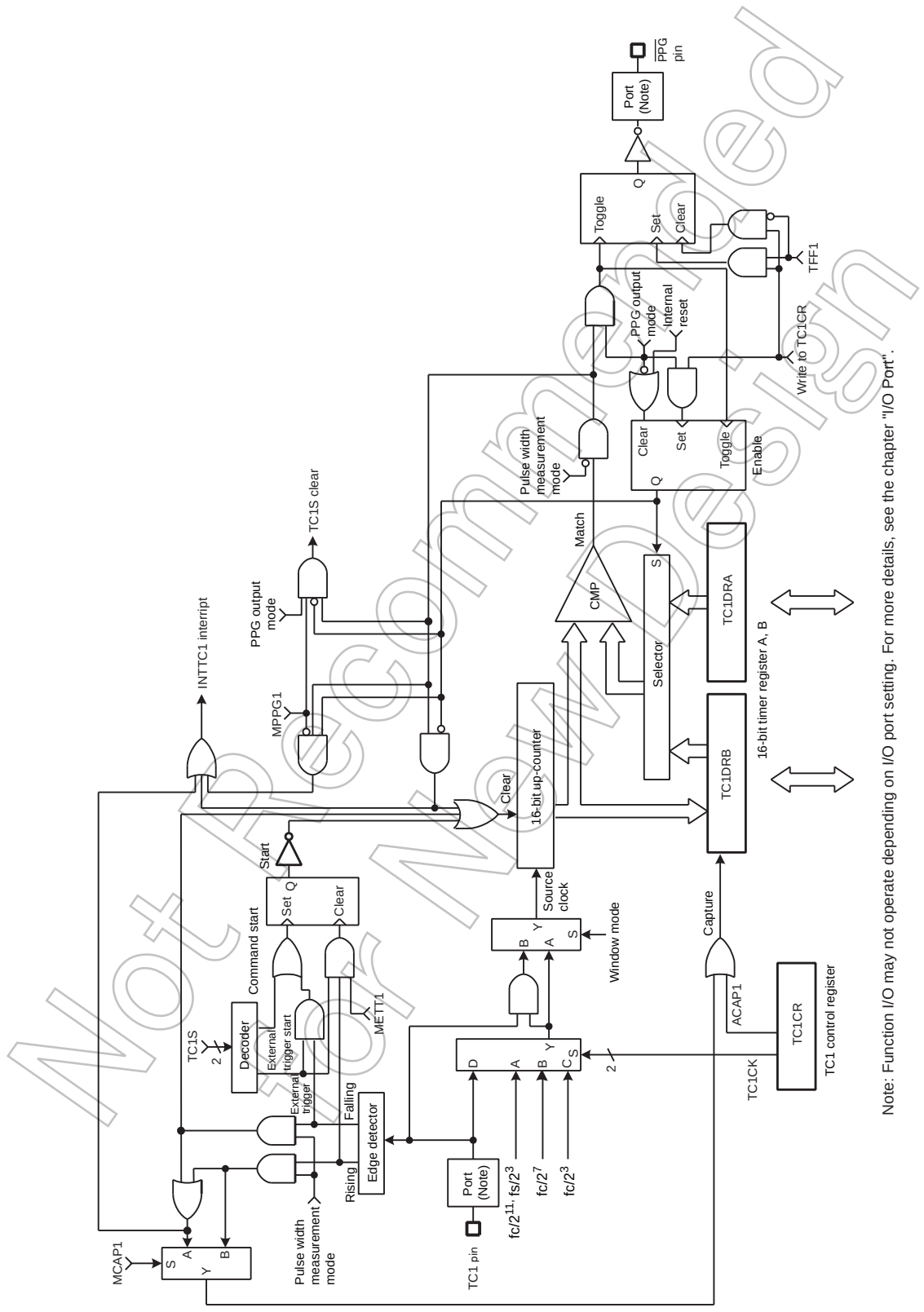


Figure 8-1 TimerCounter 1 (TC1)

Not Recommended for New Design

8.2 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)	TC1DRAL (0010H)
	(Initial value: 1111 1111 1111 1111)	Read/Write
TC1DRB (0013H, 0012H)	TC1DRBH (0013H)	TC1DRBL (0012H)
	(Initial value: 1111 1111 1111 1111)	Read/Write (Write enabled only in the PPG output mode)

TimerCounter 1 Control Register

	7	6	5 4	3 2	1 0	
TC1CR (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	O	O	O	O	O	O	
		01: Command start	O	-	-	-	-	O	
		10: Rising edge start (Ex-trigger/Pulse/PPG) Rising edge count (Event) Positive logic count (Window)	-	O	O	O	O	O	
		11: Falling edge start (Ex-trigger/Pulse/PPG) Falling edge count (Event) Negative logic count (Window)	-	O	O	O	O	O	
TC1CK	TC1 source clock select [Hz]		NORMAL1/2, IDLE1/2 mode				Divider	SLOW, SLEEP mode	R/W
			DV7CK = 0		DV7CK = 1				
		00	$fc/2^{11}$		$fs/2^3$		DV9	$fs/2^3$	
		01	$fc/2^7$		$fc/2^7$		DV5	-	
		10	$fc/2^3$		$fc/2^3$		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.

Not Recommended for New Design

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.

Not Recommended for New Design

8.3 Function

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

8.3.1 Timer mode

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

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

TC1CK	NORMAL1/2, IDLE1/2 mode				SLOW, SLEEP mode	
	DV7CK = 0		DV7CK = 1			
	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 $fc/2^{11}$ [Hz] and generating an interrupt 1 second later (fc = 16 MHz, TBTCR<DV7CK> = “0”)

```
LDW     (TC1DRA), 1E84H     ; Sets the timer register (1 s ÷ 2^11/fc = 1E84H)
DI                          ; IMF= “0”
SET     (EIRL). 7           ; 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.

Not Recommended for New Design

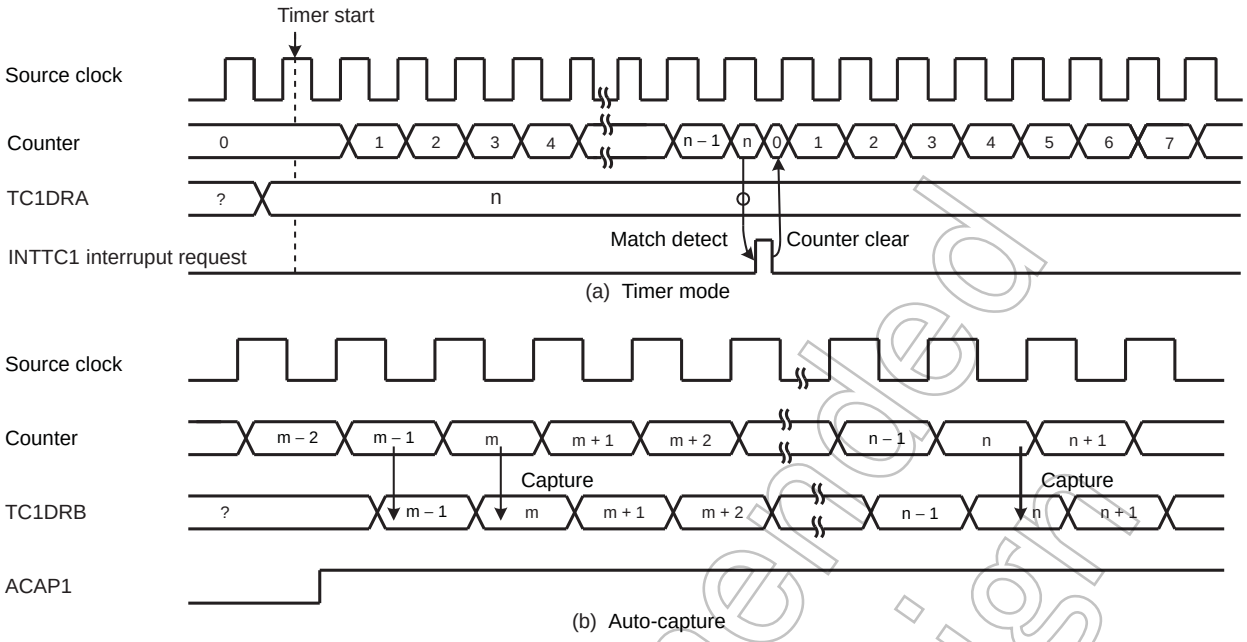


Figure 8-2 Timer Mode Timing Chart

Not Recommended for New Design

8.3.2 External Trigger Timer Mode

In the external trigger timer mode, the up-counter starts counting by the input pulse triggering of the TC1 pin, and counts up at the edge of the internal clock. For the trigger edge used to start counting, either the rising or falling edge is defined in TC1CR<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/fc [s] or less are rejected as noise. A pulse width of 12/fc [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.

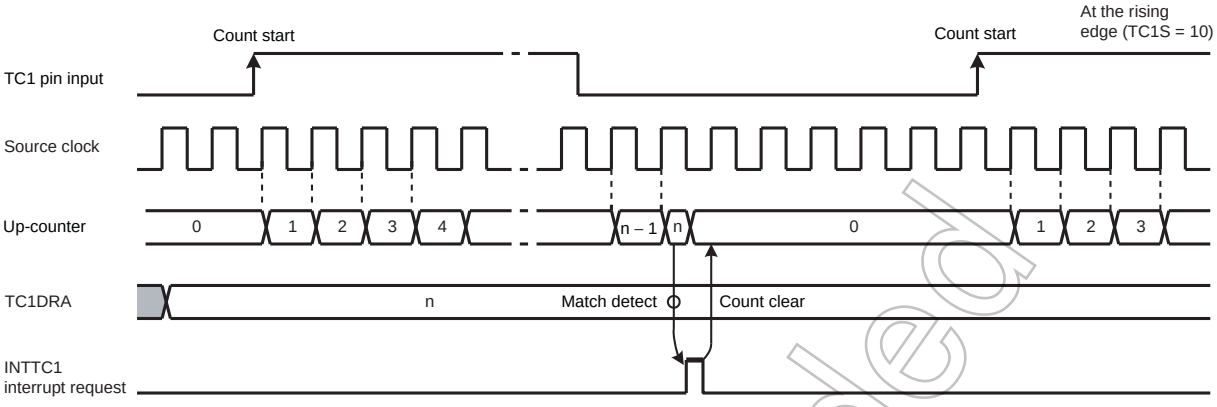
Example 1 :Generating an interrupt 1 ms after the rising edge of the input pulse to the TC1 pin
(fc = 16 MHz)

```
LDW     (TC1DRA), 007DH      ; 1ms ÷ 2^7/fc = 7DH
DI                           ; IMF= "0"
SET     (EIRL). 7            ; Enables INTTC1 interrupt
EI                           ; IMF= "1"
LD      (TC1CR), 00000100B   ; Selects the source clock and mode
LD      (TC1CR), 00100100B   ; Starts TC1 external trigger, METT1= 0
```

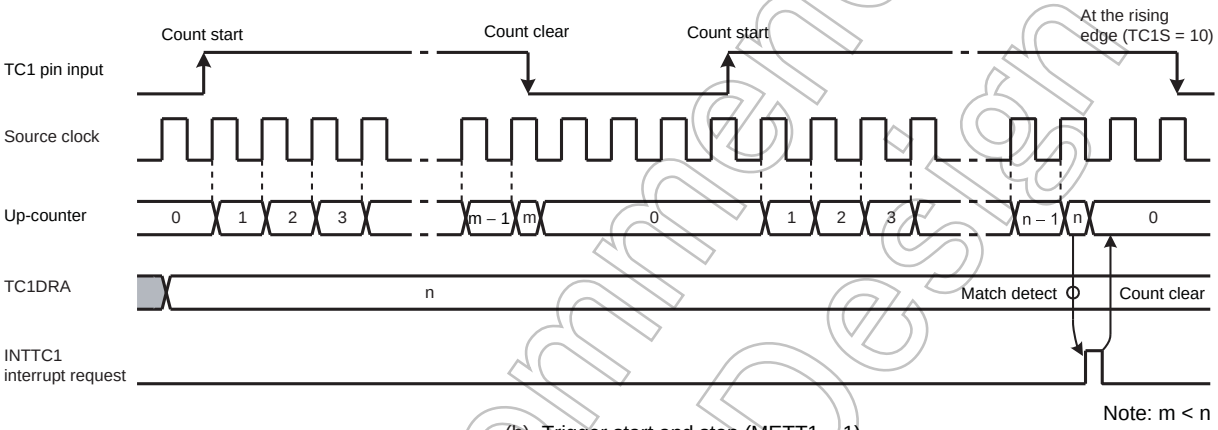
Example 2 :Generating an interrupt when the low-level pulse with 4 ms or more width is input to the TC1 pin
(fc = 16 MHz)

```
LDW     (TC1DRA), 01F4H      ; 4 ms ÷ 2^7/fc = 1F4H
DI                           ; IMF= "0"
SET     (EIRL). 7            ; Enables INTTC1 interrupt
EI                           ; IMF= "1"
LD      (TC1CR), 00000100B   ; Selects the source clock and mode
LD      (TC1CR), 01110100B   ; Starts TC1 external trigger, METT1= 1
```

Not Recommended for New Design



(a) Trigger start (METT1 = 0)



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

Figure 8-3 External Trigger Timer Mode Timing Chart

Not Recommended for New Design

8.3.3 Event Counter Mode

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

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

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

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

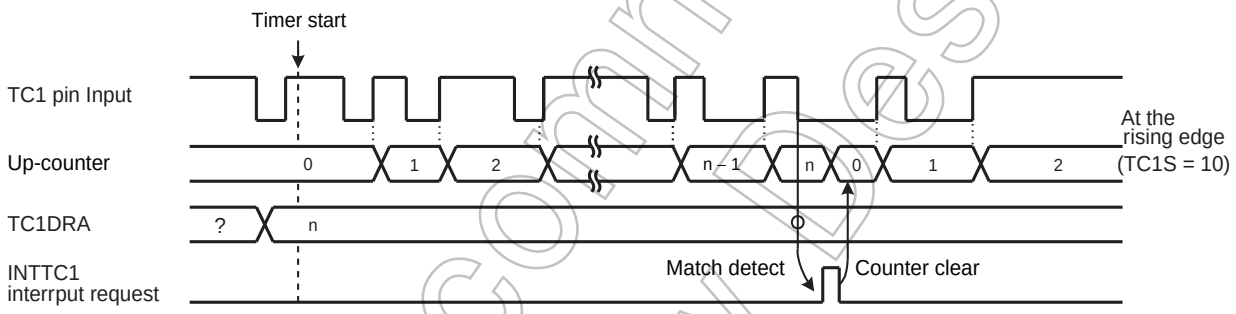


Figure 8-4 Event Counter Mode Timing Chart

Table 8-2 Input Pulse Width to TC1 Pin

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

Not Recommended for New Design

8.3.4 Window Mode

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

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

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

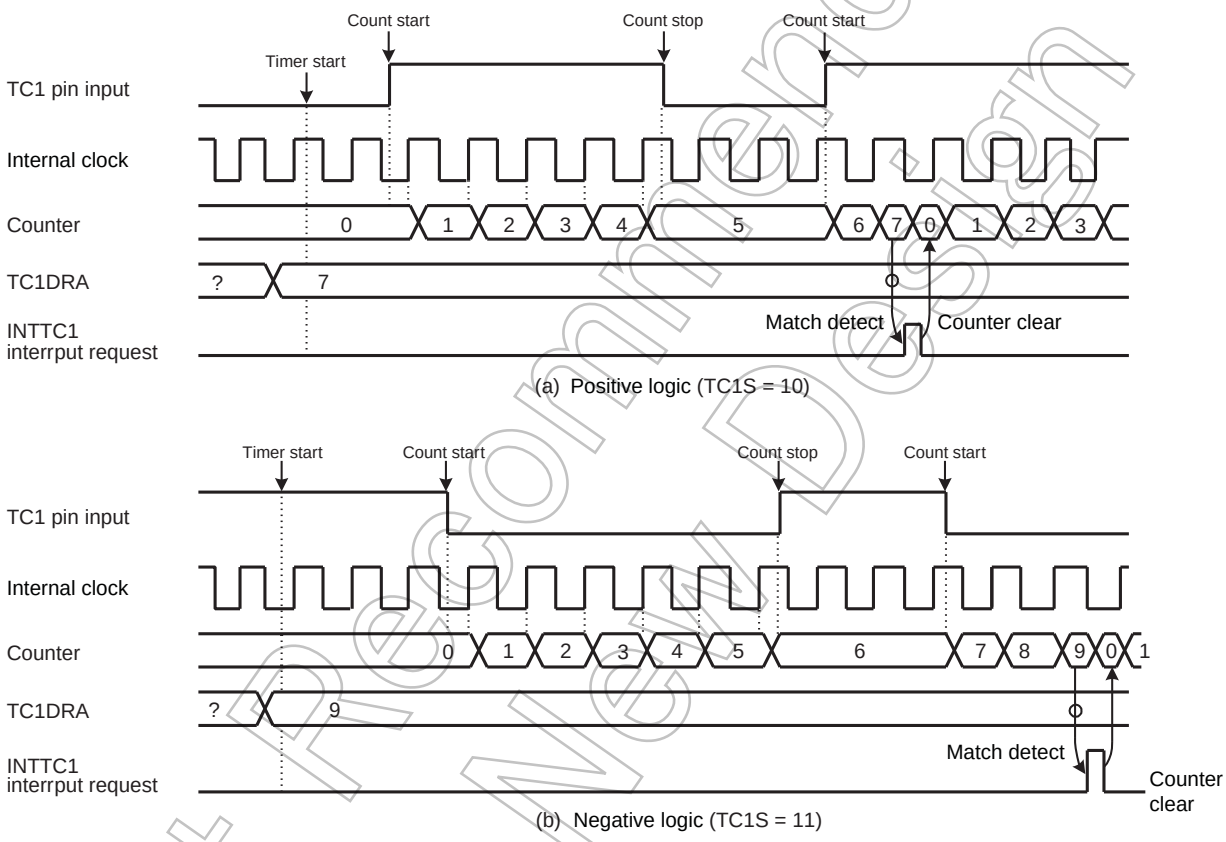


Figure 8-5 Window Mode Timing Chart

Not Recommended for New Design

8.3.5 Pulse Width Measurement Mode

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

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

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

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

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

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

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

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

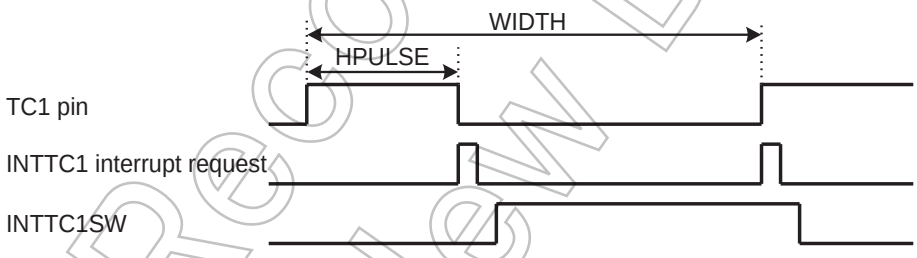
Note 2: For the single-edge capture, the counter after capturing the value stops at "1" until detecting the next edge. Therefore, the second captured value is "1" larger than the captured value immediately after counting starts.

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

Not Recommended for New Design

Example :Duty measurement (resolution fc/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     (EIRL). 7           ; Enables INTTC1
                EI                          ; IMF= "1"
                LD      (TC1CR), 00100110B  ; Starts TC1 with an external trigger at MCAP1 = 0
                :
PINTTC1:        CPL     (INTTC1SW). 0       ; INTTC1 interrupt, inverts and tests INTTC1 service switch
                JRS     F, SINTTC1
                LD      A, (TC1DRBL)        ; Reads TC1DRB (High-level pulse width)
                LD      W,(TC1DRBH)
                LD      (HPULSE), WA        ; Stores high-level pulse width in RAM
                RETI
SINTTC1:        LD      A, (TC1DRBL)        ; Reads TC1DRB (Cycle)
                LD      W,(TC1DRBH)
                LD      (WIDTH), WA         ; Stores cycle in RAM
                :
                RETI                        ; Duty calculation
                :
VINTTC1:        DW      PINTTC1             ; INTTC1 Interrupt vector
```



Not Recommended for New Design

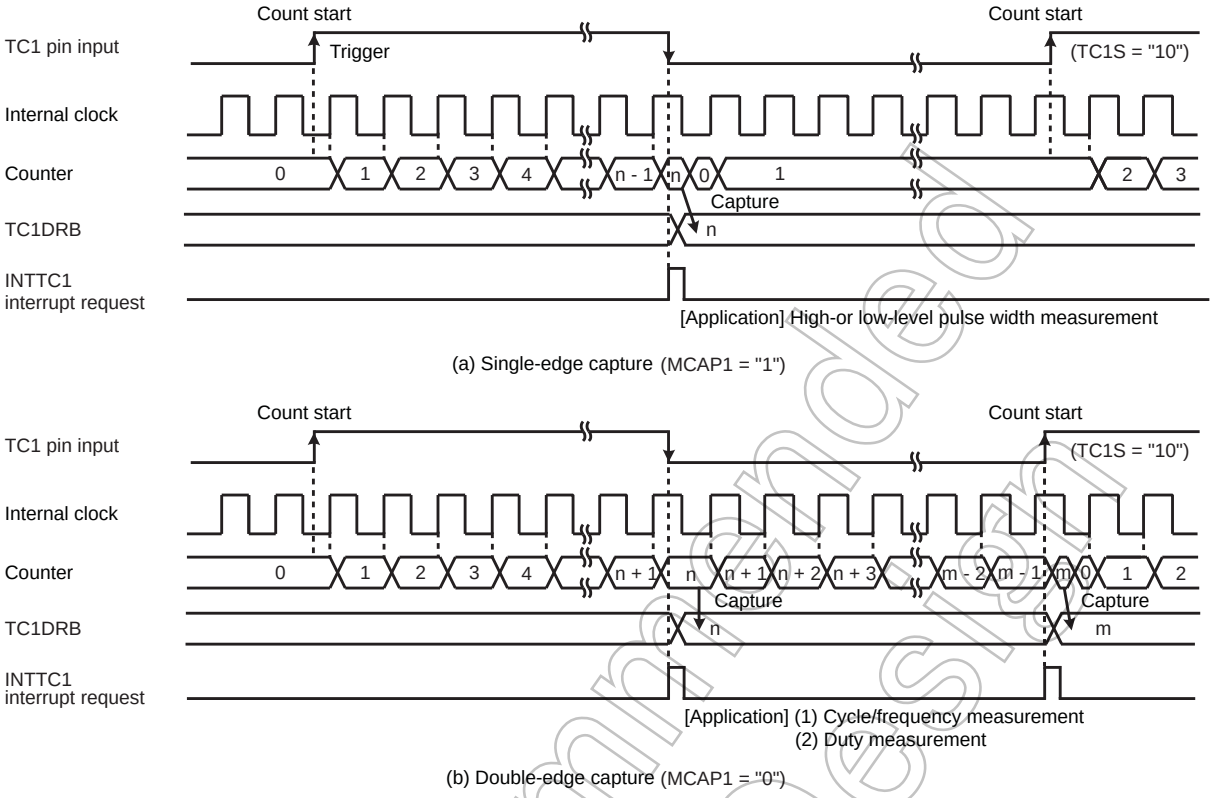


Figure 8-6 Pulse Width Measurement Mode

Not Recommended for New Design

8.3.6 Programmable Pulse Generate (PPG) Output Mode

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

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

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

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

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

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

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

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

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

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

TC1DRA > TC1DRB

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

Example :Generating a pulse which is high-going for 800 μs and low-going for 200 μs
(fc = 16 MHz)

```
                Setting port
LD      (TC1CR), 10000111B      ; Sets the PPG mode, selects the source clock
LDW     (TC1DRA), 007DH         ; Sets the cycle (1 ms ÷ 2^7/fc μs = 007DH)
LDW     (TC1DRB), 0019H         ; Sets the low-level pulse width (200 μs ÷ 2^7/fc = 0019H)
LD      (TC1CR), 10010111B      ; Starts the timer
```

Not Recommended for New Design

Example :After stopping PPG, setting the PPG pin to a high-level to restart PPG
(fc = 16 MHz)

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

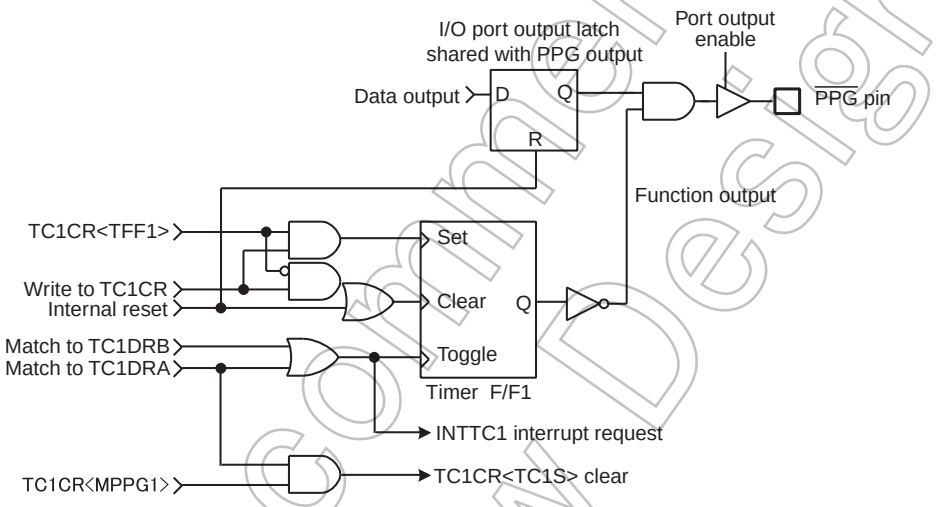


Figure 8-7 $\overline{\text{PPG}}$ Output

Not Recommended for New Design

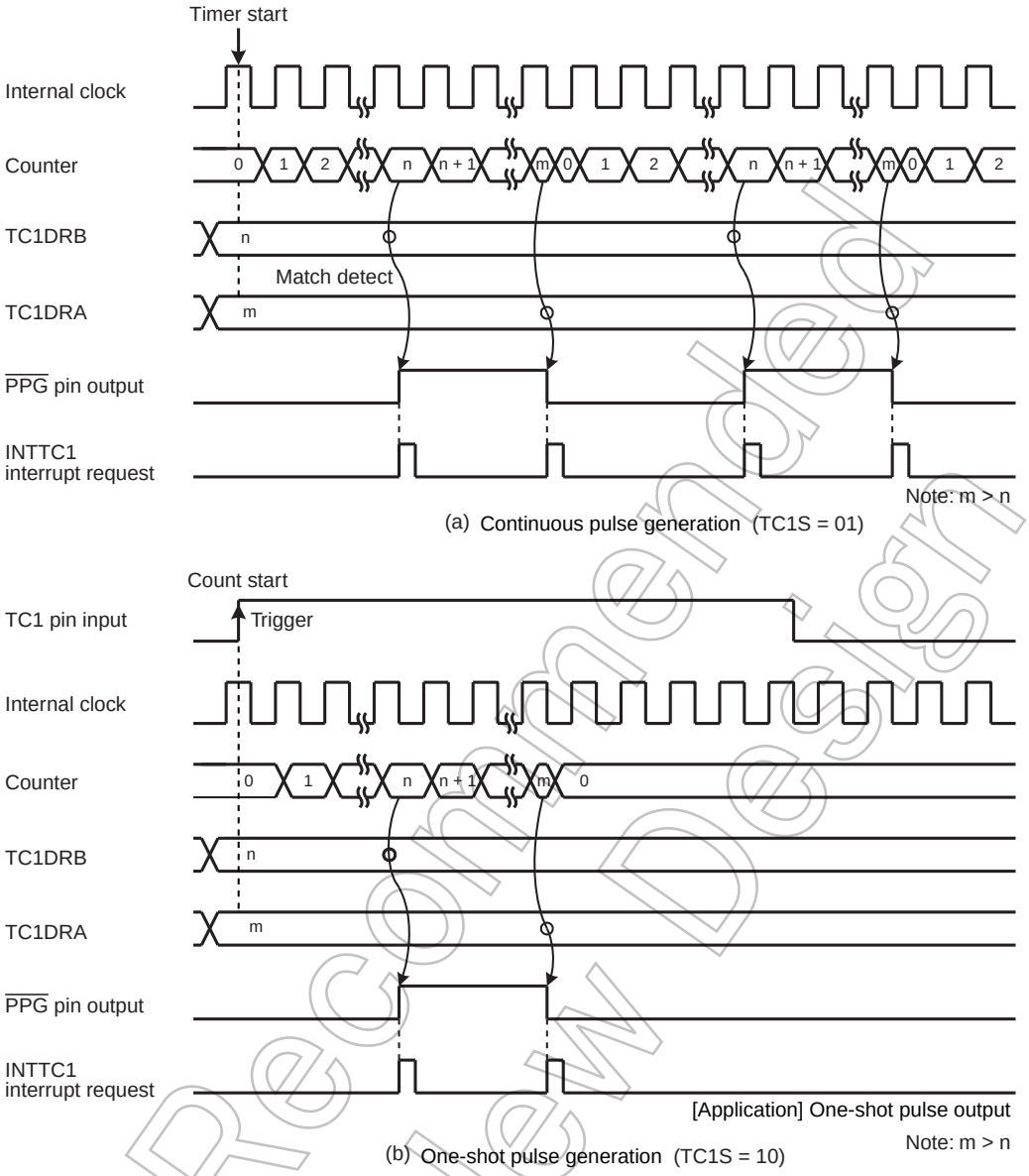


Figure 8-8 PPG Mode Timing Chart

Not Recommended for New Design

Not Recommended for New Design

9. 8-Bit TimerCounter (TC3, TC4)

9.1 Configuration

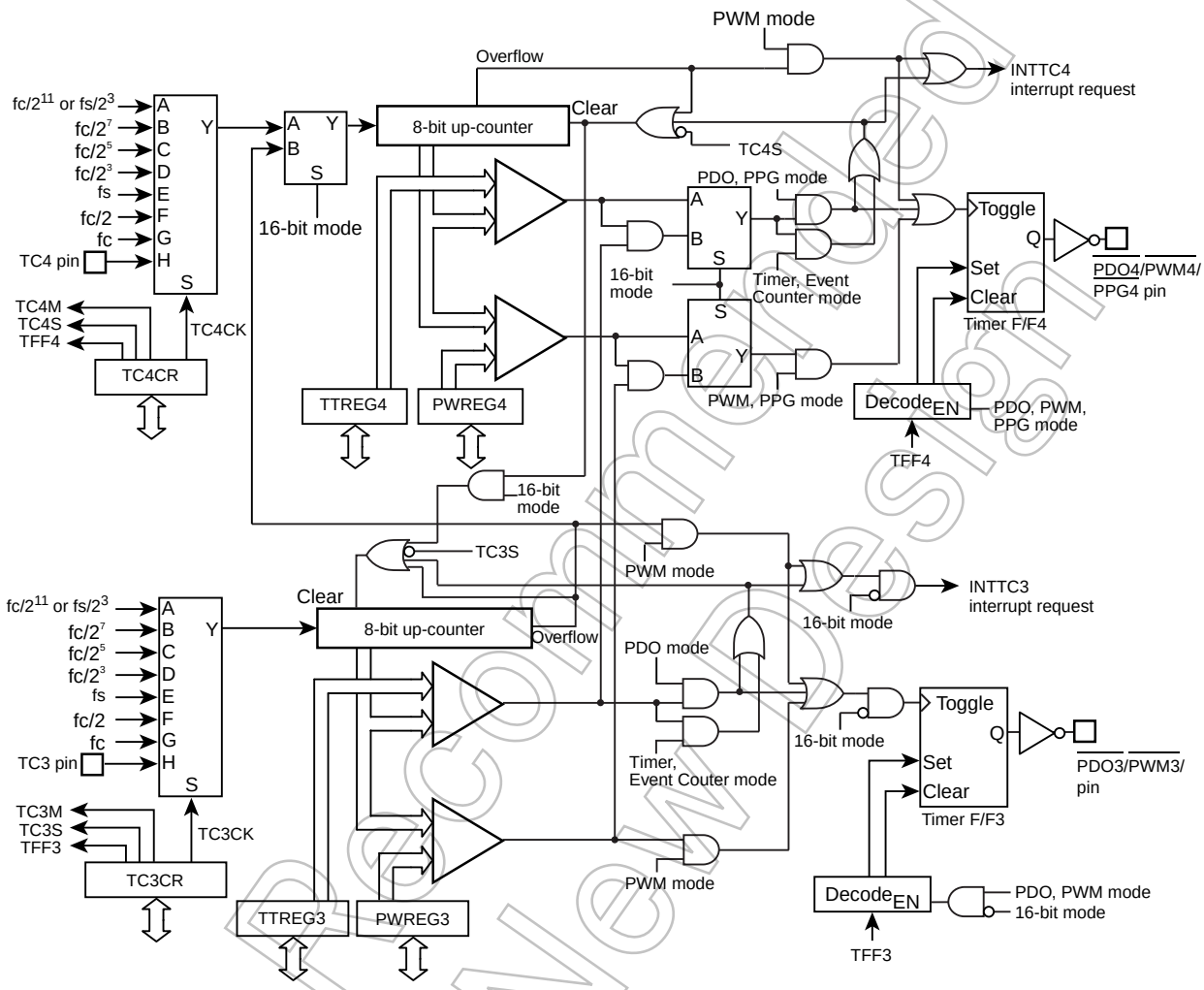


Figure 9-1 8-Bit TimerCounter 3, 4

Not Recommended for New Design

9.2 TimerCounter Control

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

TimerCounter 3 Timer Register

	7	6	5	4	3	2	1	0	
TTREG3 (0018H) R/W									(Initial value: 1111 1111)
PWREG3 (001AH) R/W									(Initial value: 1111 1111)

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

TC3CR (0016H)	7	6	5	4	3	2	1	0	
	TFF3	TC3CK			TC3S	TC3M			(Initial value: 0000 0000)

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		SLOW1/2 SLEEP1/2 mode	R/W
			DV7CK = 0	DV7CK = 1	
		000	fc/2^{11}	fs/2^3	fs/2^3	
		001	fc/2^7	fc/2^7	-	
		010	fc/2^5	fc/2^5	-	
		011	fc/2^3	fc/2^3	-	
		100	fs	fs	fs	
		101	fc/2	fc/2	-	
		110	fc (Note 8)	fc (Note 8)	fc (Note 8)	
		111	TC3 pin input			
TC3S	TC3 start control (Note 3)	0: Operation stop and counter clear 1: Operation start				R/W
TC3M	TC3M operating mode select (Note 2,3,7)	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				R/W

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

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

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

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

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

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

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

Not Recommended for New Design

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.

Not Recommended for New Design

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

	7	6	5	4	3	2	1	0	
TTREG4 (0019H) R/W									(Initial value: 1111 1111)
PWREG4 (001BH) R/W									(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

	7	6	5	4	3	2	1	0	
TC4CR (0017H)	TFF4	TC4CK			TC4S	TC4M			(Initial value: 0000 0000)

TFF4	Timer F/F4 control (Note 2,3)	0: Clear 1: Set			R/W
TC4CK	Operating clock selection [Hz] (Note 2,3,7)		NORMAL1/2, IDLE1/2 mode	SLOW1/2 SLEEP1/2 mode	R/W
			DV7CK = 0	DV7CK = 1	
		000	$fc/2^{11}$	$fs/2^3$	$fs/2^3$
		001	$fc/2^7$	$fc/2^7$	-
		010	$fc/2^5$	$fc/2^5$	-
		011	$fc/2^3$	$fc/2^3$	-
		100	fs	fs	fs
		101	fc/2	fc/2	-
		110	fc (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.

Not Recommended for New Design

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

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

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

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

Operating mode	$fc/2^{11}$ or $fs/2^3$	$fc/2^7$	$fc/2^5$	$fc/2^3$	fs	fc/2	fc	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 9-2 Operating Mode and Selectable Source Clock (SLOW1/2 and SLEEP1/2 Modes)

Operating mode	$fc/2^{11}$ or $fs/2^3$	$fc/2^7$	$fc/2^5$	$fc/2^3$	fs	fc/2	fc	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

Not Recommended for New Design

Table 9-3 Constraints on Register Values Being Compared

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

Note: n = 3 to 4

Not Recommended for New Design

9.3 Function

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

9.3.1 8-Bit Timer Mode (TC3 and 4)

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

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

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

Note 3: j = 3, 4

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

Source Clock (Note)			Resolution		Maximum Setting Time	
NORMAL1/2, IDLE1/2 mode		SLOW1/2, SLEEP1/2 mode	fc = 16 MHz	fs = 32.768 kHz	fc = 16 MHz	fs = 32.768 kHz
DV7CK = 0	DV7CK = 1					
$fc/2^{11}$ [Hz]	$fs/2^{3}$ [Hz]	$fs/2^{3}$ [Hz]	128 µs	244.14 µs	32.6 ms	62.3 ms
$fc/2^{7}$	$fc/2^{7}$	-	8 µs	-	2.0 ms	-
$fc/2^{5}$	$fc/2^{5}$	-	2 µs	-	510 µs	-
$fc/2^{3}$	$fc/2^{3}$	-	500 ns	-	127.5 µs	-

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

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

```
LD      (TTREG4), 0AH           ; Sets the timer register (80 µs ÷ 2^7/fc = 0AH).
DI
SET     (EIRH). 1               ; Enables INTTC4 interrupt.
EI
LD      (TC4CR), 00010000B      ; Sets the operating clock to fc/2^7, and 8-bit timer mode.
LD      (TC4CR), 00011000B      ; Starts TC4.
```

Not Recommended for New Design

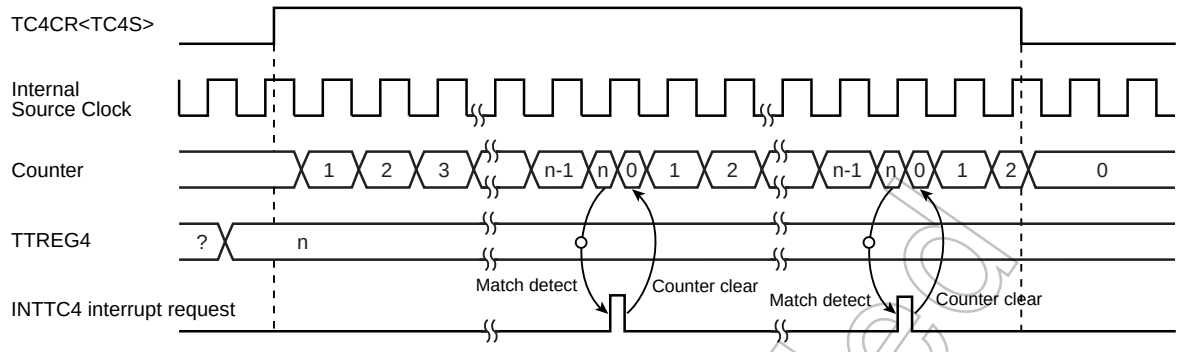


Figure 9-2 8-Bit Timer Mode Timing Chart (TC4)

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

In the 8-bit event counter mode, the up-counter counts up at the falling edge of the input pulse to the TCj pin. When a match between the up-counter and the TTREGj value is detected, an INTTCj interrupt is generated and the up-counter is cleared. After being cleared, the up-counter restarts counting at the falling edge of the input pulse to the TCj pin. Two machine cycles are required for the low- or high-level pulse input to the TCj pin. Therefore, a maximum frequency to be supplied is $fc/2^4$ Hz in the NORMAL1/2 or IDLE1/2 mode, and $fs/2^4$ 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 $\overline{\text{PDOj}}$, $\overline{\text{PWMj}}$ and $\overline{\text{PPGj}}$ pins may output pulses.

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

Note 3: j = 3, 4

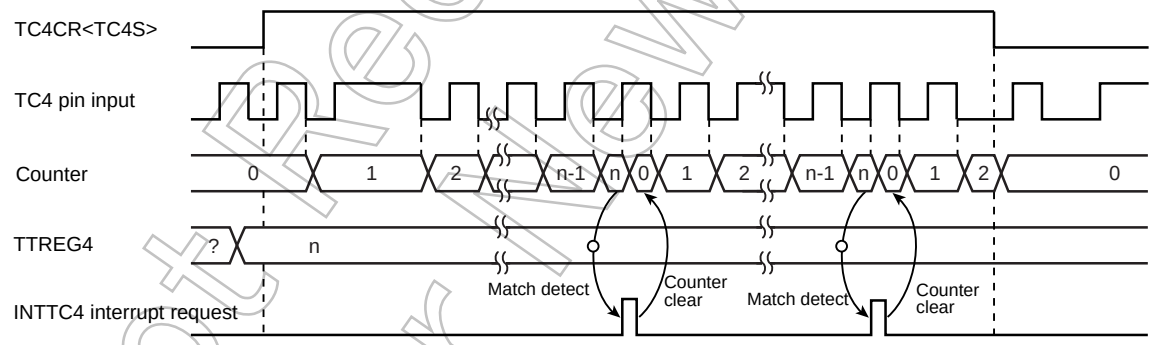


Figure 9-3 8-Bit Event Counter Mode Timing Chart (TC4)

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

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

Not Recommended for New Design

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

```
        Setting port
LD      (TTREG4), 3DH           ; 1/1024 ÷ 2^7/fc ÷ 2 = 3DH
LD      (TC4CR), 00010001B      ; Sets the operating clock to fc/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 $\overline{\text{PDOj}}$ 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 $\overline{\text{PDOj}}$ pin to the high level when the TimerCounter is stopped

```
CLR (TCjCR).3 ; Stops the timer.
CLR (TCjCR).7 ; Sets the PDOj pin to the high level.
```

Note 3: j = 3, 4

Not Recommended for New Design

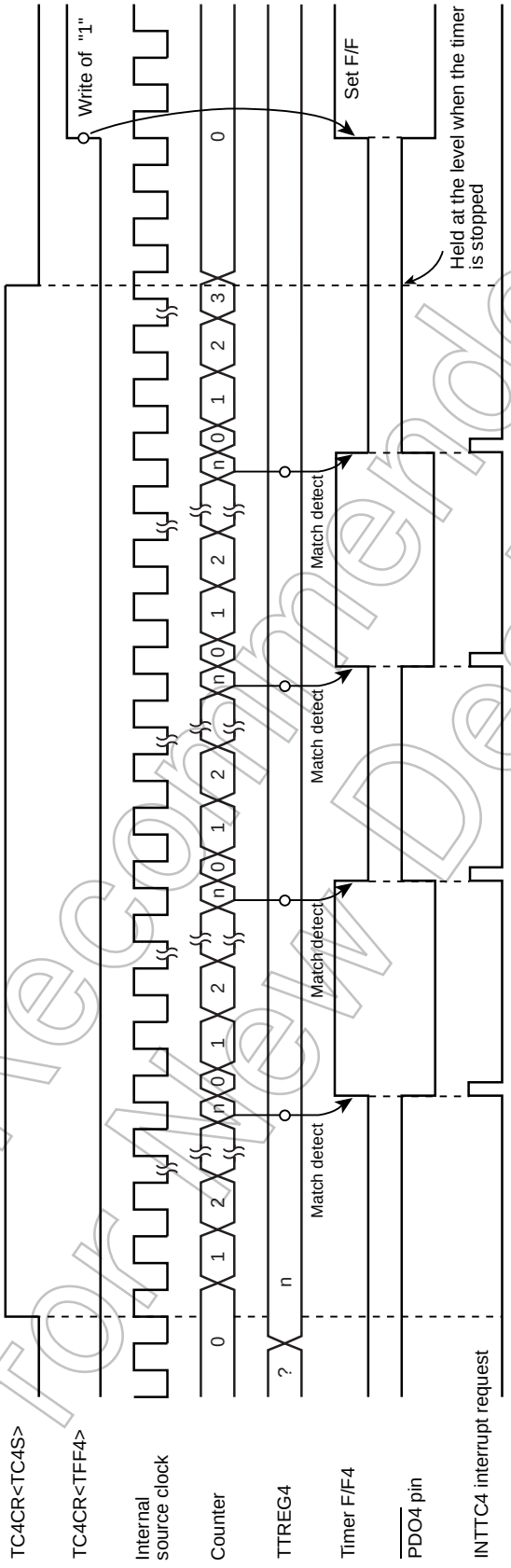


Figure 9-4 8-Bit PDO Mode Timing Chart (TC4)

Not Recommended for New Design

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

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

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

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

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

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

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

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

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

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

CLR (TCjCR).3 ; Stops the timer.

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

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

Note 4: j = 3, 4

Table 9-5 PWM Output Mode

Source Clock			Resolution		Repeated Cycle	
NORMAL1/2, IDLE1/2 mode		SLOW1/2, SLEEP1/2 mode	fc = 16 MHz	fs = 32.768 kHz	fc = 16 MHz	fs = 32.768 kHz
DV7CK = 0	DV7CK = 1					
$fc/2^{11}$ [Hz]	$fs/2^3$ [Hz]	$fs/2^3$ [Hz]	128 μs	244.14 μs	32.8 ms	62.5 ms
$fc/2^7$	$fc/2^7$	-	8 μs	-	2.05 ms	-
$fc/2^5$	$fc/2^5$	-	2 μs	-	512 μs	-
$fc/2^3$	$fc/2^3$	-	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	-

Not Recommended for New Design

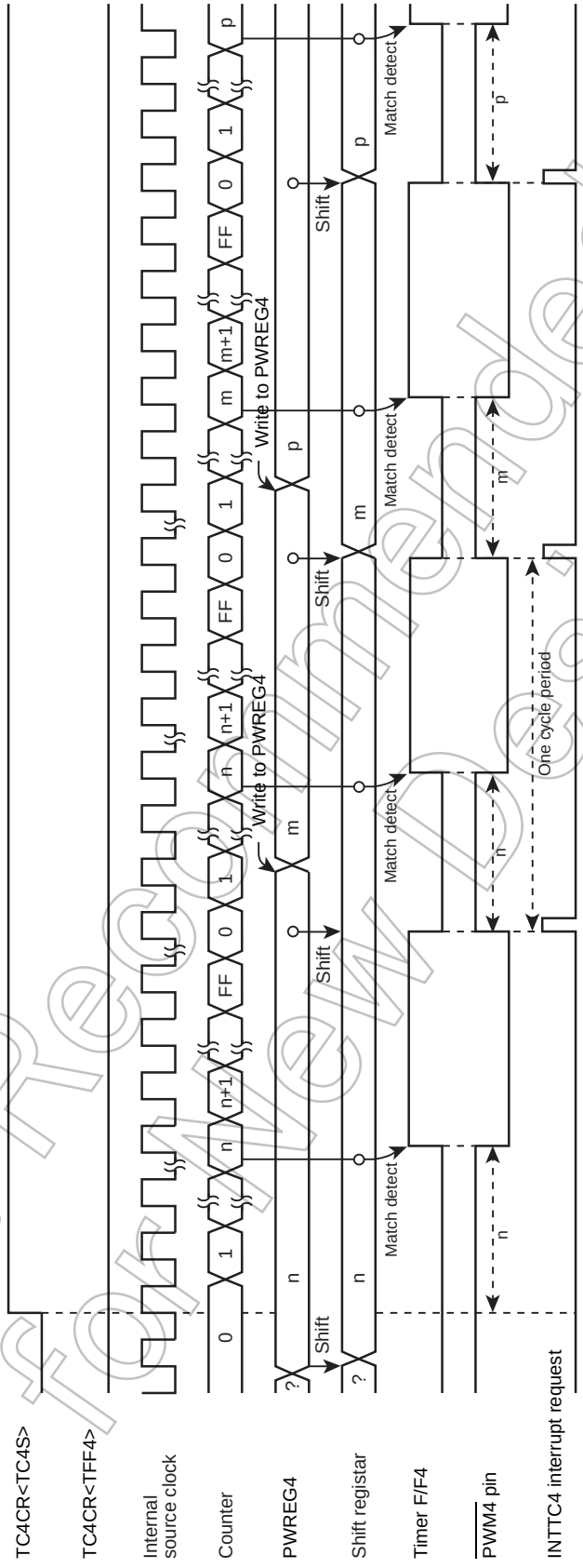


Figure 9-5 8-Bit PWM Mode Timing Chart (TC4)

Not Recommended for New Design

9.3.5 16-Bit Timer Mode (TC3 and 4)

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

When a match between the up-counter and the timer register (TTREG3, TTREG4) value is detected after the timer is started by setting TC4CR<TC4S> to 1, an INTTC4 interrupt is generated and the up-counter is cleared. After being cleared, the up-counter continues counting. Program the 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 9-6 Source Clock for 16-Bit Timer Mode

Source Clock			Resolution		Maximum Setting Time	
NORMAL1/2, IDLE1/2 mode		SLOW1/2, SLEEP1/2 mode	fc = 16 MHz	fs = 32.768 kHz	fc = 16 MHz	fs = 32.768 kHz
DV7CK = 0	DV7CK = 1					
$fc/2^{11}$	$fs/2^3$	$fs/2^3$	128 µs	244.14 µs	8.39 s	16 s
$fc/2^7$	$fc/2^7$	-	8 µs	-	524.3 ms	-
$fc/2^5$	$fc/2^5$	-	2 µs	-	131.1 ms	-
$fc/2^3$	$fc/2^3$	-	500 ns	-	32.8 ms	-

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

```
LDW     (TTREG3), 927CH     ; Sets the timer register (300 ms ÷ 2^7/fc = 927CH).
DI
SET     (EIRH). 1           ; Enables INTTC4 interrupt.
EI
LD      (TC3CR), 13H        ; Sets the operating clock to fc/2^7, and 16-bit timer mode
                            ; (lower byte).
LD      (TC4CR), 04H        ; Sets the 16-bit timer mode (upper byte).
LD      (TC4CR), 0CH        ; Starts the timer.
```

Not Recommended for New Design

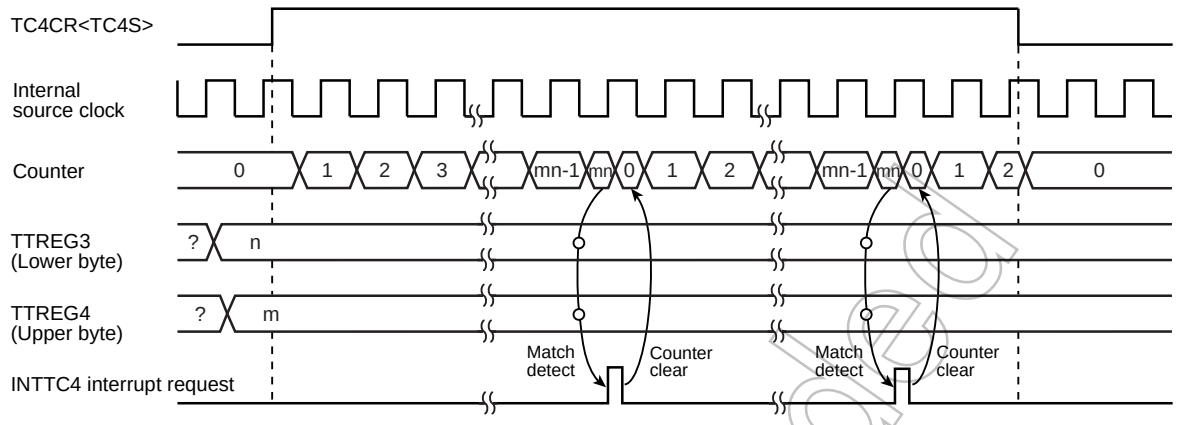


Figure 9-6 16-Bit Timer Mode Timing Chart (TC3 and TC4)

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

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

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

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

Therefore, a maximum frequency to be supplied is $fc/2^4$ Hz in the NORMAL1/2 or IDLE1/2 mode, and $fs/2^4$ 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 $\overline{\text{PDOj}}$, $\overline{\text{PWMj}}$ and $\overline{\text{PPGj}}$ pins may output pulses.

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

Note 3: j = 3, 4

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

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

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

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

Two machine cycles are required for the high- or low-level pulse input to the TC3 pin. Therefore, a maximum frequency to be supplied is $fc/2^4$ Hz in the NORMAL1/2 or IDLE1/2 mode, and $fs/2^4$ 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 $\overline{\text{PWM4}}$ pin is the opposite to the timer F/F4 logic level.)

Not Recommended for New Design

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

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

CLR (TC4CR).3 ; Stops the timer.

CLR (TC4CR).7 ; Sets the $\overline{\text{PWM4}}$ 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 $\overline{\text{PWM4}}$ pin during the warm-up period time after exiting the STOP mode.

Table 9-7 16-Bit PWM Output Mode

Source Clock			Resolution		Repeated Cycle	
NORMAL1/2, IDLE1/2 mode		SLOW1/2, SLEEP1/2 mode	fc = 16 MHz	fs = 32.768 kHz	fc = 16 MHz	fs = 32.768 kHz
DV7CK = 0	DV7CK = 1					
$fc/2^{11}$	$fs/2^3$ [Hz]	$fs/2^3$ [Hz]	128 μs	244.14 μs	8.39 s	16 s
$fc/2^7$	$fc/2^7$	-	8 μs	-	524.3 ms	-
$fc/2^5$	$fc/2^5$	-	2 μs	-	131.1 ms	-
$fc/2^3$	$fc/2^3$	-	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/2^3, and 16-bit PWM output mode
                            ; (lower byte).
LD      (TC4CR), 056H       ; Sets TFF4 to the initial value 0, and 16-bit PWM signal
                            ; generation mode (upper byte).
LD      (TC4CR), 05EH       ; Starts the timer.
```

Not Recommended for New Design

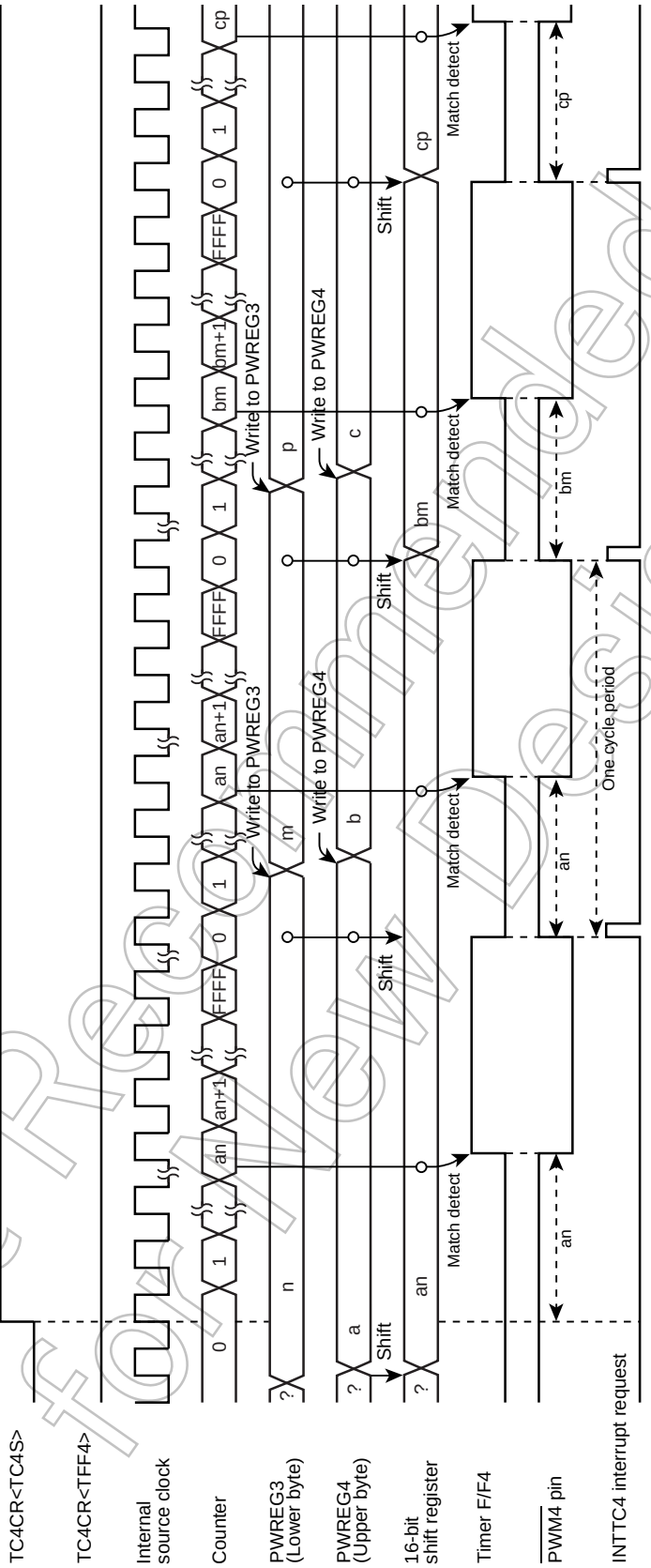


Figure 9-7 16-Bit PWM Mode Timing Chart (TC3 and TC4)

Not Recommended for New Design

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

This mode is used to generate pulses with up to 16-bits of resolution. The timer counter 3 and 4 are 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 $fc/2^4$ Hz in the NORMAL1/2 or IDLE1/2 mode, and $fs/2^4$ 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 (fc = 16.0 MHz)

```
        Setting ports
LDW     (PWREG3), 07D0H         ; Sets the pulse width.
LDW     (TTREG3), 8002H         ; Sets the cycle period.
LD      (TC3CR), 33H            ; Sets the operating clock to fc/2^3, and16-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 PPG4 pin to the high level
```

Note 3: i = 3, 4

Not Recommended for New Design

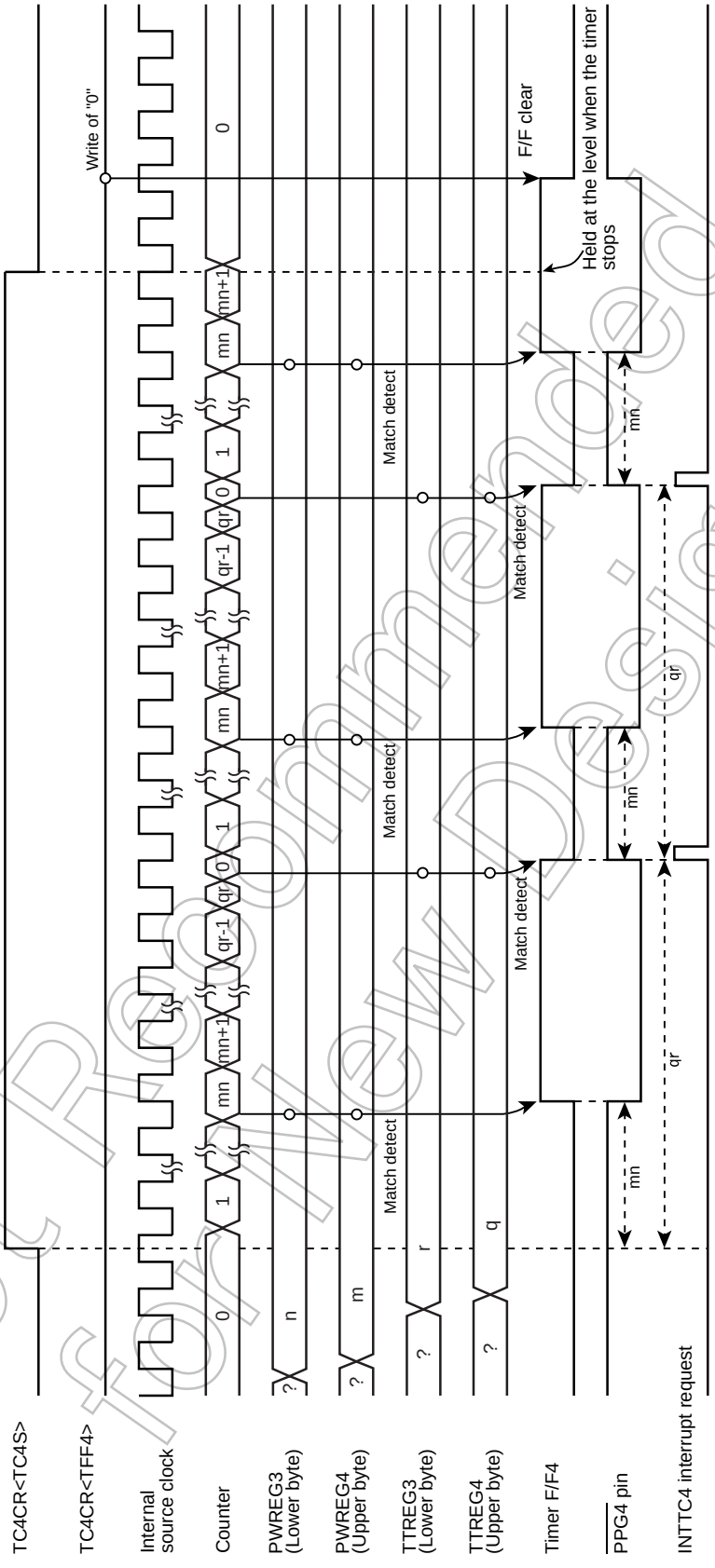


Figure 9-8 16-Bit PPG Mode Timing Chart (TC3 and TC4)

Not Recommended for New Design

9.3.9 Warm-Up Counter Mode

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

Note 1: In the warm-up counter mode, fix TCiCR<TFFi> to 0. If not fixed, the $\overline{PDOi}$, $\overline{PWMi}$ 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

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

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

Table 9-8 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). 1           ; 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
```

Not Recommended for New Design

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

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

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

Minimum time Setting (TTREG4, 3 = 0100H)	Maximum time Setting (TTREG4, 3 = FF00H)
16 μs	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 fc, 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). 1           ; 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
```

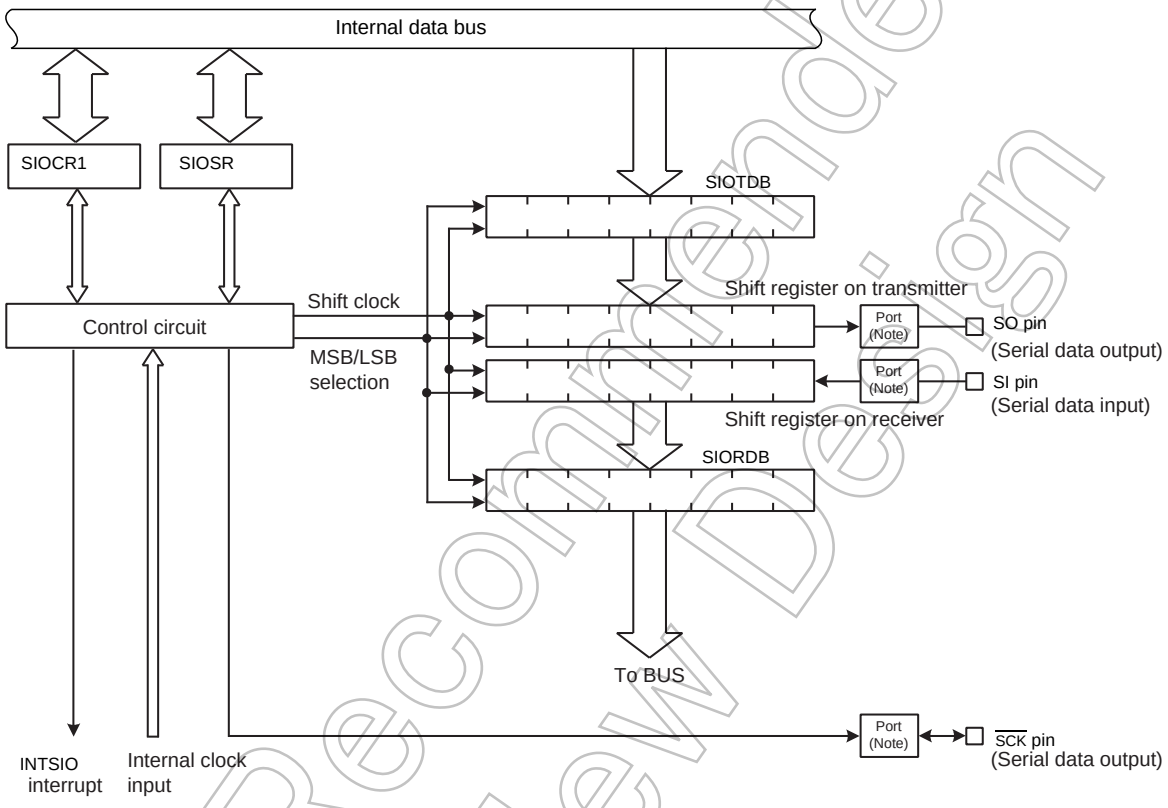
Not Recommended for New Design

10. Synchronous Serial Interface (SIO)

The serial interfaces connect to an external device via SI, SO, and $\overline{SCK}$ pins.

When these pins are used as serial interface, the output latches for each port should be set to "1".

10.1 Configuration



Note: Set the register of port correctly for the port assigned as serial interface pins.
For details, see the description of the input/output port control register.

Figure 10-1 Synchronous Serial Interface (SIO)

Not Recommended for New Design

10.2 Control

The SIO is controlled using the serial interface control register (SIOCR1). The operating status of the serial interface can be inspected by reading the status register (SIOCR1).

Serial Interface Control Register

SIOCR1 (0026H)

7	6	5	4	3	2	1	0
SIOS	SIOINH	SIOM		SIODIR	SCK		

(Initial value: 0000 0000)

SIOS	Specify start/stop of transfer	0: Stop 1: Start				R/W
SIOINH	Forcibly stops transfer (Note 1)	0: - 1: Forcibly stop (Automatically cleared to "0" after stopping)				
SIOM	Selects transfer mode	00: Transmit mode 01: Receive mode 10: Transmit/receive mode 11: Reserved				
SIODIR	Selects direction of transfer	0: MSB (Transfer beginning with bit7) 1: LSB (Transfer beginning with bit0)				
SCK	Selects serial clock		NORMAL1/2 or IDLE1/2 modes		SLOW/SLEEP mode	
			TBTCR <DV7CK> = "0"	TBTCR <DV7CK> = "1"		
		000	$fc/2^{12}$	$fs/2^4$	$fs/2^4$	
		001	$fc/2^8$	$fc/2^8$	Reserved	
		010	$fc/2^7$	$fc/2^7$	Reserved	
		011	$fc/2^6$	$fc/2^6$	Reserved	
		100	$fc/2^5$	$fc/2^5$	Reserved	
		101	$fc/2^4$	$fc/2^4$	Reserved	
		110	$fc/2^3$	$fc/2^3$	Reserved	
		111	External clock (Input from $\overline{SCK}$ pin)			

Note 1: When SIOCR1<SIOINH> is set to “1”, SIOCR1<SIOS>, SIOSR register, SIORDB register and SIOTDB register are initialized.

Note 2: Transfer mode, direction of transfer and serial clock must be select during the transfer is stopping (when SIOSR<SIOF> "0").

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

Not Recommended for New Design

Serial Interface Status Register

SIOSR (0027H)

7	6	5	4	3	2	1	0
SIOF	SEF	TXF	RXF	TXERR	RXERR		

(Initial value: 0010 00**)

SIOF	Serial transfer operation status monitor	0: Transfer finished 1: Transfer in progress	Read only
SEF	Number of clocks monitor	0: 8 clocks 1: 1 to 7 clocks	
TXF	Transmit buffer empty flag	0: Data exists in transmit buffer 1: No data exists in transmit buffer	
RXF	Receive buffer full flag	0: No data exists in receive buffer 1: Data exists in receive buffer	
TXERR	Transfer operation error flag	Read 0: - (No error exist) 1: Transmit buffer under run occurs in an external clock mode Write 0: Clear the flag 1: - (A write of "1" to this bit is ignored)	R/W
RXERR	Receive operation error flag	Read 0: - (No error exist) 1: Receive buffer over run occurs in an external clock mode Write 0: Clear the flag 1: - (A write of "1" to this bit is ignored)	

Note 1: The operation error flag (TXERR and RXERR) are not automatically cleared by stopping transfer with SIOCR1<SIOS> "0". Therefore, set these bits to "0" for clearing these error flag. Or set SIOCR1<SIOINH> to "1".

Note 2: *: Don't care

Receive buffer register

SIORDB (0028H)

7	6	5	4	3	2	1	0

Read only

(Initial value: 0000 0000)

Transmit buffer register

SIOTDB (0028H)

7	6	5	4	3	2	1	0

Write only

(Initial value: **** ****)

Note 1: SIOTDB is write only register. A bit manipulation should not be performed on the transmit buffer register using a read-modify-write instruction.

Note 2: The SIOTDB should be written after checking SIOSR<TXF> "1". When SIOSR<TXF> is "0", the writing data can't be transferred to SIOTDB even if write instruction is executed to SIOTDB

Note 3: *: Don't care

Not Recommended for New Design

10.3 Function

10.3.1 Serial clock

10.3.1.1 Clock source

The serial clock can be selected by using SIOCR1<SCK>. When the serial clock is changed, the writing instruction to SIOCR1<SCK> should be executed while the transfer is stopped (when SIOSR<SIOF> “0”)

(1) Internal clock

Setting the SIOCR1<SCK> to other than “111B” outputs the clock (shown in "Table 10-1 Serial Clock Rate (fc = 16 MHz, fs = 32.768kHz)") as serial clock outputs from $\overline{SCK}$ pin. At the before beginning or finishing of a transfer, $\overline{SCK}$ pin is kept in high level.

When writing (in the transmit mode) or reading (in the receive mode) data can not follow the serial clock rate, an automatic-wait function is executed to stop the serial clock automatically and hold the next shift operation until reading or writing is completed (shown in "Figure 10-2 Automatic-wait Function (Example of transmit mode)"). The maximum time from releasing the automatic-wait function by reading or writing a data is 1 cycle of the selected serial clock until the serial clock comes out from $\overline{SCK}$ pin.

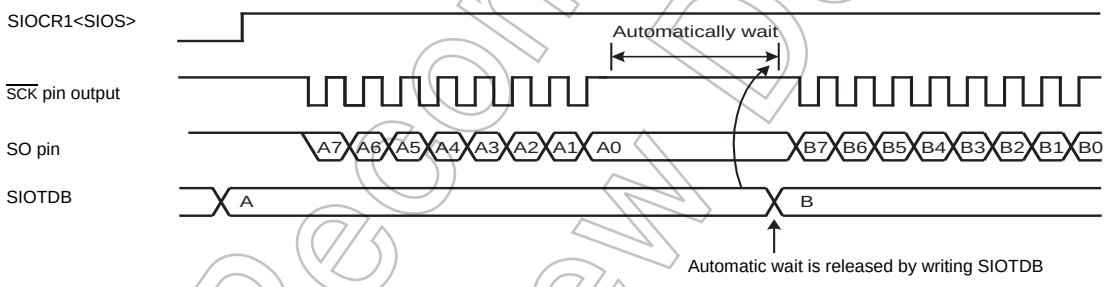


Figure 10-2 Automatic-wait Function (Example of transmit mode)

Table 10-1 Serial Clock Rate (fc = 16 MHz, fs = 32.768kHz)

	NORMAL1/2, IDLE1/2 Mode				SLOW1/2, SLEEP1/2 Mode	
	TBTCR<DV7CK> = "0"		TBTCR<DV7CK> = "1"		Serial Clock	Baud Rate
SCK	Serial Clock	Baud Rate	Serial Clock	Baud Rate		
000	$fc/2^{12}$	3.906 kbps	$fs/2^{4}$	2048 bps	$fs/2^{4}$	2048 bps
001	$fc/2^{8}$	62.5 kbps	$fc/2^{8}$	62.5 kbps	Reserved	-
010	$fc/2^{7}$	125 kbps	$fc/2^{7}$	125 kbps	Reserved	-
011	$fc/2^{6}$	250 kbps	$fc/2^{6}$	250 kbps	Reserved	-
100	$fc/2^{5}$	500 kbps	$fc/2^{5}$	500 kbps	Reserved	-
101	$fc/2^{4}$	1.00 Mbps	$fc/2^{4}$	1.00 Mbps	Reserved	-
110	$fc/2^{3}$	2.00 Mbps	$fc/2^{3}$	2.00 Mbps	Reserved	-

Not Recommended for New Design

(2) External clock

When an external clock is selected by setting SIOCR1<SCK> to "111B", the clock via the $\overline{SCK}$ pin from an external source is used as the serial clock.

To ensure shift operation, the serial clock pulse width must be 4/fc or more for both "H" and "L" levels.

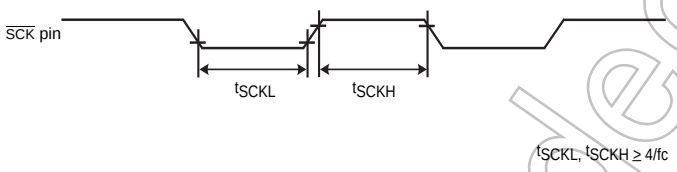


Figure 10-3 External Clock

10.3.1.2 Shift edge

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

(1) Leading edge shift

Data is shifted on the leading edge of the serial clock (falling edge of the $\overline{SCK}$ pin input/output).

(2) Trailing edge shift

Data is shifted on the trailing edge of the serial clock (rising edge of the $\overline{SCK}$ pin input/output).

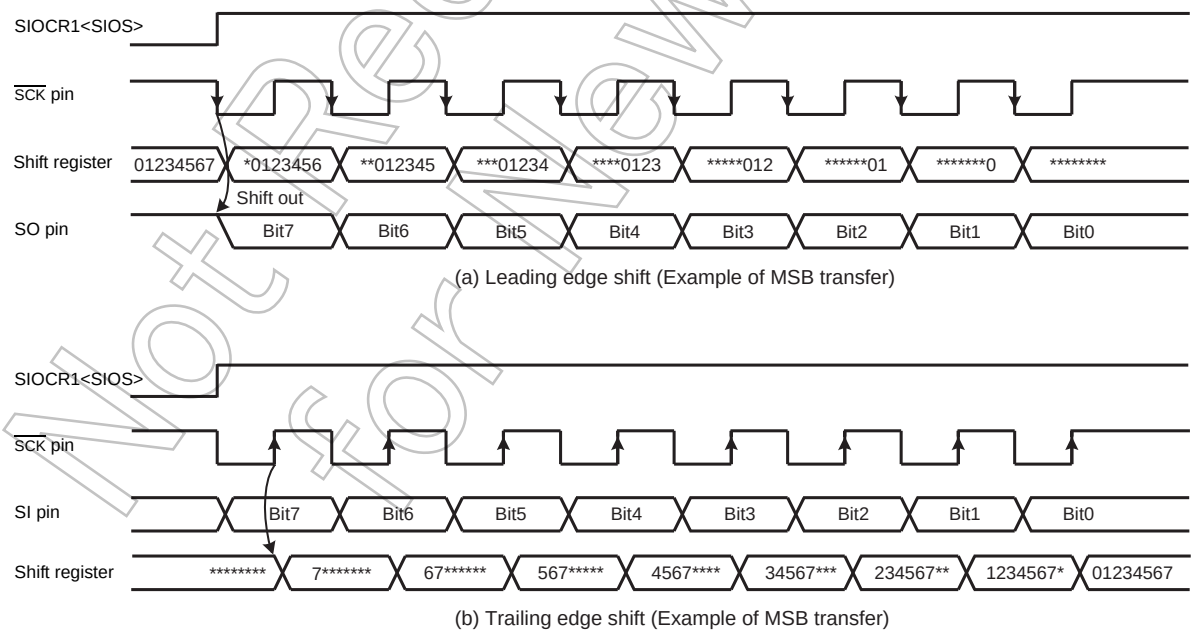


Figure 10-4 Shift Edge

Not Recommended for New Design

10.3.2 Transfer bit direction

Transfer data direction can be selected by using SIOCR1<SIODIR>. The transfer data direction can't be set individually for transmit and receive operations.

When the data direction is changed, the writing instruction to SIOCR1<SIODIR> should be executed while the transfer is stopped (when SIOCR1<SIOF>= “0”)

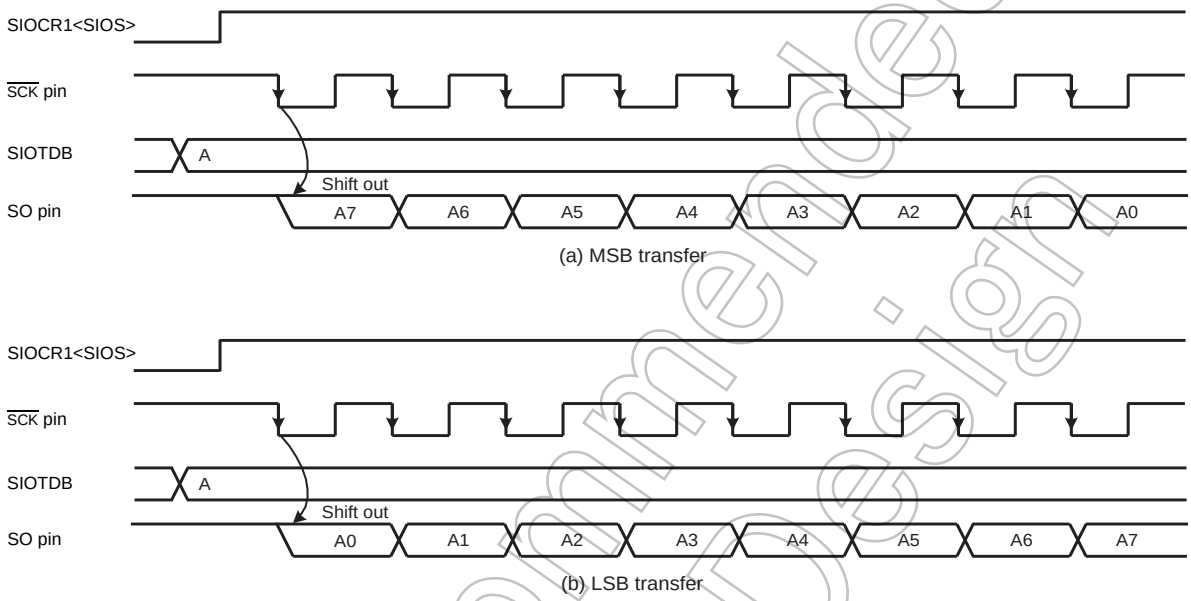


Figure 10-5 Transfer Bit Direction (Example of transmit mode)

10.3.2.1 Transmit mode

(1) MSB transmit mode

MSB transmit mode is selected by setting SIOCR1<SIODIR> to “0”, in which case the data is transferred sequentially beginning with the most significant bit (Bit7).

(2) LSB transmit mode

LSB transmit mode is selected by setting SIOCR1<SIODIR> to “1”, in which case the data is transferred sequentially beginning with the least significant bit (Bit0).

10.3.2.2 Receive mode

(1) MSB receive mode

MSB receive mode is selected by setting SIOCR1<SIODIR> to “0”, in which case the data is received sequentially beginning with the most significant bit (Bit7).

Not Recommended for New Design

(2) LSB receive mode

LSB receive mode is selected by setting SIOCR1<SIODIR> to “1”, in which case the data is received sequentially beginning with the least significant bit (Bit0).

10.3.2.3 Transmit/receive mode

(1) MSB transmit/receive mode

MSB transmit/receive mode are selected by setting SIOCR1<SIODIR> to “0” in which case the data is transferred sequentially beginning with the most significant bit (Bit7) and the data is received sequentially beginning with the most significant (Bit7).

(2) LSB transmit/receive mode

LSB transmit/receive mode are selected by setting SIOCR1<SIODIR> to “1”, in which case the data is transferred sequentially beginning with the least significant bit (Bit0) and the data is received sequentially beginning with the least significant (Bit0).

10.3.3 Transfer modes

Transmit, receive and transmit/receive mode are selected by using SIOCR1<SIOM>.

10.3.3.1 Transmit mode

Transmit mode is selected by writing “00B” to SIOCR1<SIOM>.

(1) Starting the transmit operation

Transmit mode is selected by setting “00B” to SIOCR1<SIOM>. Serial clock is selected by using SIOCR1<SCK>. Transfer direction is selected by using SIOCR1<SIODIR>.

When a transmit data is written to the transmit buffer register (SIOTDB), SIOSR<TXF> is cleared to “0”.

After SIOCR1<SIOS> is set to “1”, SIOSR<SIOF> is set synchronously to “1” the falling edge of $\overline{\text{SCK}}$ pin.

The data is transferred sequentially starting from SO pin with the direction of the bit specified by SIOCR1<SIODIR>, synchronizing with the $\overline{\text{SCK}}$ pin's falling edge.

SIOSR<SEF> is kept in high level, between the first clock falling edge of $\overline{\text{SCK}}$ pin and eighth clock falling edge.

SIOSR<TXF> is set to “1” at the rising edge of pin after the data written to the SIOTDB is transferred to shift register, then the INTSIO interrupt request is generated, synchronizing with the next falling edge on $\overline{\text{SCK}}$ pin.

Note 1: In internal clock operation, when SIOCR1<SIOS> is set to "1", transfer mode does not start without writing a transmit data to the transmit buffer register (SIOTDB).

Note 2: In internal clock operation, when the SIOCR1<SIOS> is set to "1", SIOTDB is transferred to shift register after maximum 1-cycle of serial clock frequency, then a serial clock is output from $\overline{\text{SCK}}$ pin.

Note 3: In external clock operation, when the falling edge is input from $\overline{\text{SCK}}$ pin after SIOCR1<SIOS> is set to "1", SIOTDB is transferred to shift register immediately.

Not Recommended for New Design

(2) During the transmit operation

When data is written to SIOTDB, SIOSR<TXF> is cleared to “0”.

In internal clock operation, in case a next transmit data is not written to SIOTDB, the serial clock stops to “H” level by an automatic-wait function when all of the bit set in the SIOTDB has been transmitted. Automatic-wait function is released by writing a transmit data to SIOTDB. Then, transmit operation is restarted after maximum 1-cycle of serial clock.

When the next data is written to the SIOTDB before termination of previous 8-bit data with SIOSR<TXF> “1”, the next data is continuously transferred after transmission of previous data.

In external clock operation, after SIOSR<TXF> is set to “1”, the transmit data must be written to SIOTDB before the shift operation of the next data begins.

If the transmit data is not written to SIOTDB, transmit error occurs immediately after shift operation is started. Then, INTSIO interrupt request is generated after SIOSR<TXERR> is set to “1”.

(3) Stopping the transmit operation

There are two ways for stopping transmits operation.

- The way of clearing SIOCR1<SIOS>.

  When SIOCR1<SIOS> is cleared to “0”, transmit operation is stopped after all transfer of the data is finished. When transmit operation is finished, SIOSR<SIOF> is cleared to “0” and SO pin is kept in high level. In external clock operation, SIOCR1<SIOS> must be cleared to “0” before SIOSR<SEF> is set to “1” by beginning next transfer.
- The way of setting SIOCR1<SIOINH>.

  Transmit operation is stopped immediately after SIOCR1<SIOINH> is set to “1”. In this case, SIOCR1<SIOS>, SIOSR register, SIORDB register and SIOTDB register are initialized.

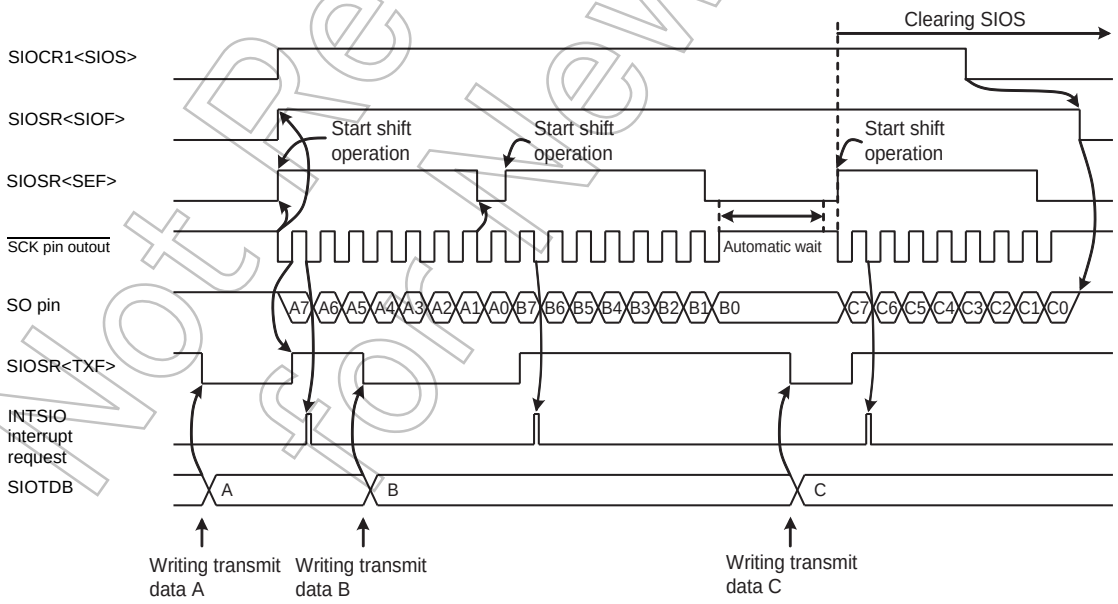


Figure 10-6 Example of Internal Clock and MSB Transmit Mode

Not Recommended for New Design

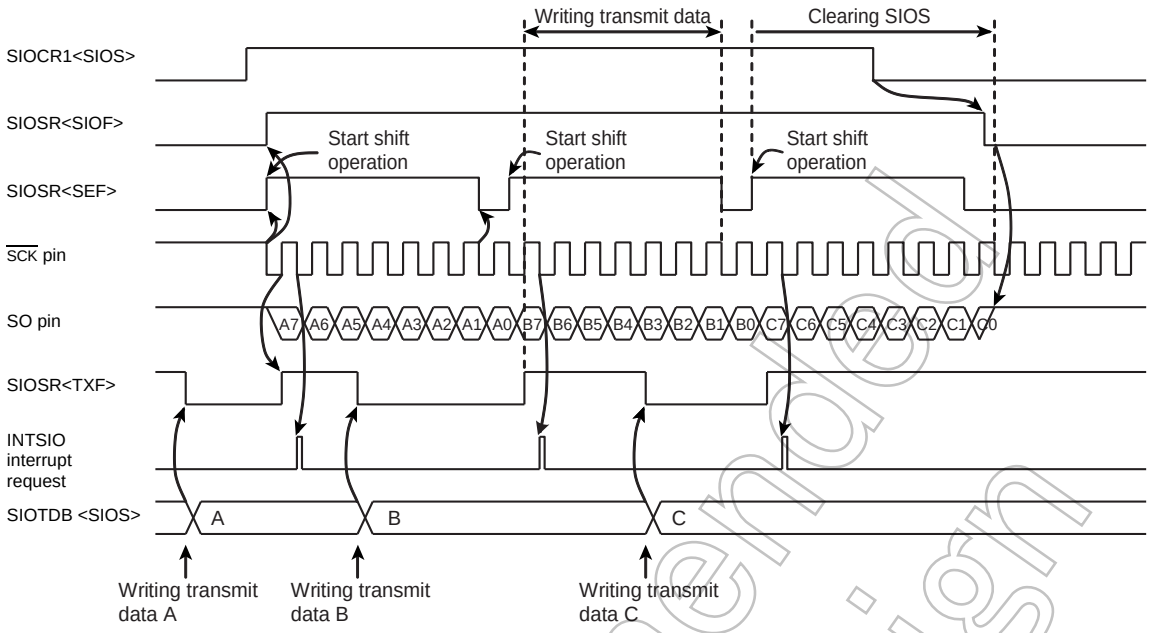


Figure 10-7 Example of External Clock and MSB Transmit Mode

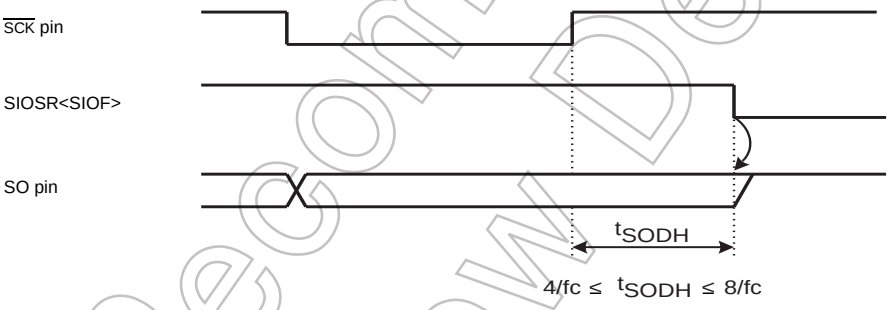


Figure 10-8 Hold Time of the End of Transmit Mode

(4) Transmit error processing

Transmit errors occur on the following situation.

- Shift operation starts before writing next transmit data to SIOTDB in external clock operation.

  If transmit errors occur during transmit operation, SIOSR<TXERR> is set to "1" immediately after starting shift operation. Synchronizing with the next serial clock falling edge, INTSIO interrupt request is generated. If shift operation starts before writing data to SIOTDB after SIOCR1<SIOS> is set to "1", SIOSR<TXERR> is set to "1" immediately after shift operation is started and then INTSIO interrupt request is generated.

  SIO pin is kept in high level when SIOSR<TXERR> is set to "1". When transmit error occurs, transmit operation must be forcibly stop by writing SIOCR1<SIOINH> to "1". In this case, SIOCR1<SIOS>, SIOSR register, SIORDB register and SIOTDB register are initialized.

Not Recommended for New Design

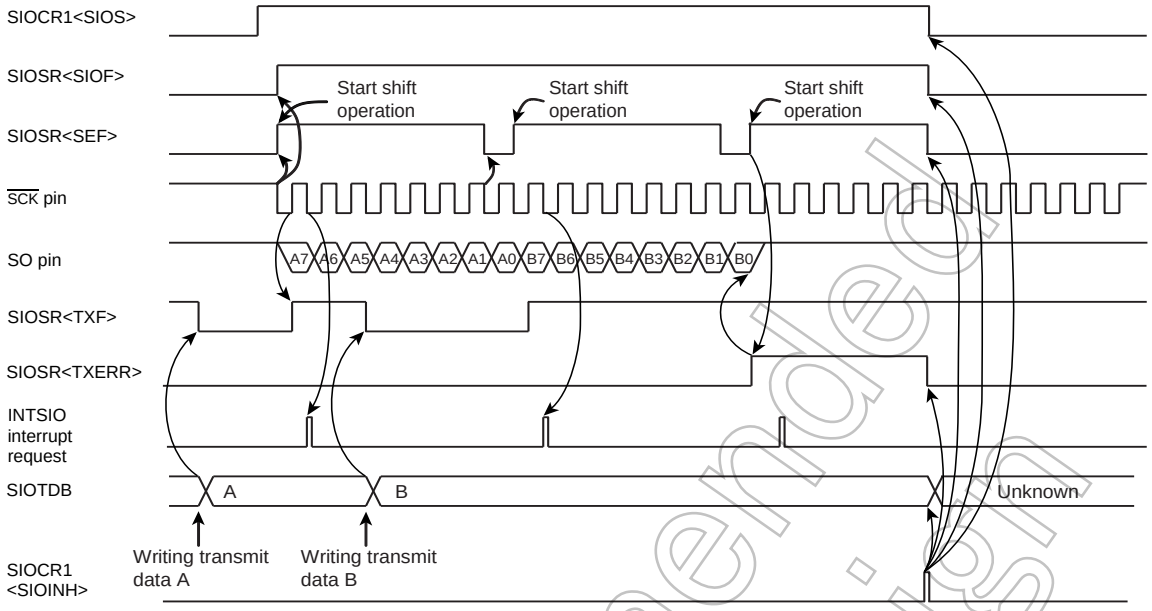


Figure 10-9 Example of Transmit Error Processing

10.3.3.2 Receive mode

The receive mode is selected by writing “01B” to SIOCR<SIOM>.

(1) Starting the receive operation

Receive mode is selected by setting “01” to SIOCR1<SIOM>. Serial clock is selected by using SIOCR1<SCK>. Transfer direction is selected by using SIOCR1<SIODIR>.

After SIOCR1<SIOS> is set to “1”, SIOSR<SIOF> is set synchronously to “1” the falling edge of $\overline{\text{SCK}}$ pin.

Synchronizing with the $\overline{\text{SCK}}$ pin's rising edge, the data is received sequentially from SI pin with the direction of the bit specified by SBIDIR<SIODIR>.

SIOSR<SEF> is kept in high level, between the first clock falling edge of $\overline{\text{SCK}}$ pin and eighth clock falling edge.

When 8-bit data is received, the data is transferred to SIORDB from shift register. INTSIO interrupt request is generated and SIOSR<RXF> is set to “1”

Note: In internal clock operation, when the SIOCR1<SIOS> is set to "1", the serial clock is generated from $\overline{\text{SCK}}$ pin after maximum 1-cycle of serial clock frequency.

(2) During the receive operation

The SIOSR<RXF> is cleared to “0” by reading a data from SIORDB.

In the internal clock operation, the serial clock stops to “H” level by an automatic-wait function when the all of the 8-bit data has been received. Automatic-wait function is released by reading a received data from SIORDB. Then, receive operation is restarted after maximum 1-cycle of serial clock.

In external clock operation, after SIOSR<RXF> is set to “1”, the received data must be read from SIORDB, before the next data shift-in operation is finished.

Not Recommended for New Design

If received data is not read out from SIORDB receive error occurs immediately after shift operation is finished. Then INTSIO interrupt request is generated after SIOSR<RXERR> is set to "1".

(3) Stopping the receive operation

There are two ways for stopping the receive operation.

- The way of clearing SIOCR1<SIOS>.

  When SIOCR1<SIOS> is cleared to "0", receive operation is stopped after all of the data is finished to receive. When receive operation is finished, SIOSR<SIOF> is cleared to "0". In external clock operation, SIOCR1<SIOS> must be cleared to "0" before SIOSR<SEF> is set to "1" by starting the next shift operation.

- The way of setting SIOCR1<SIOINH>.

  Receive operation is stopped immediately after SIOCR1<SIOINH> is set to "1". In this case, SIOCR1<SIOS>, SIOSR register, SIORDB register and SIOTDB register are initialized.

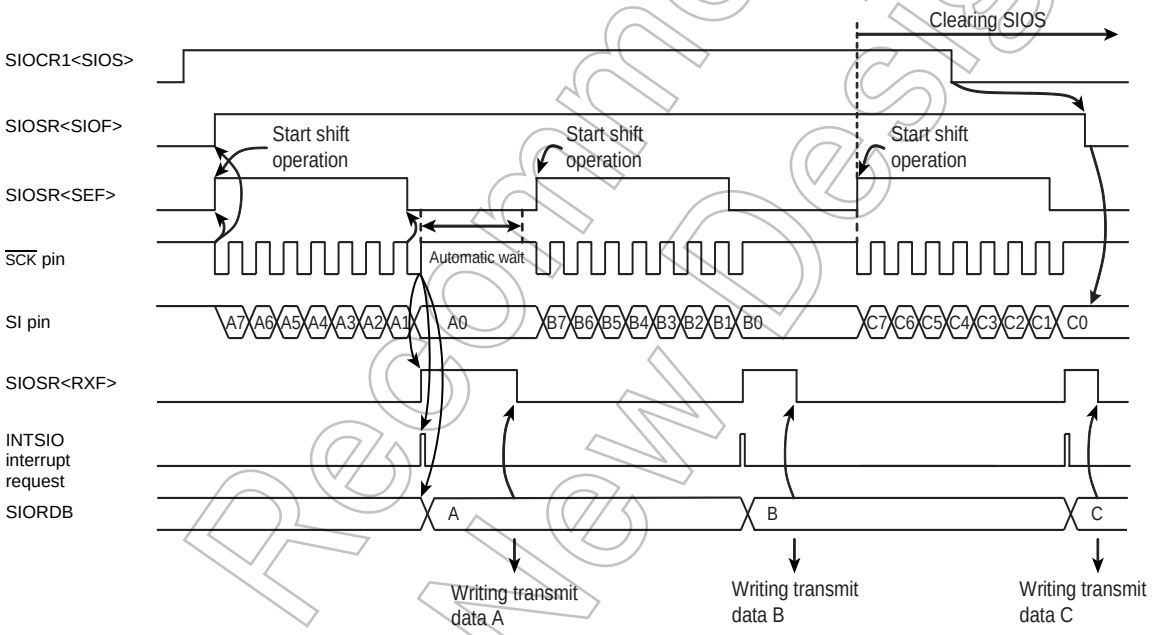


Figure 10-10 Example of Internal Clock and MSB Receive Mode

Not Recommended for New Design

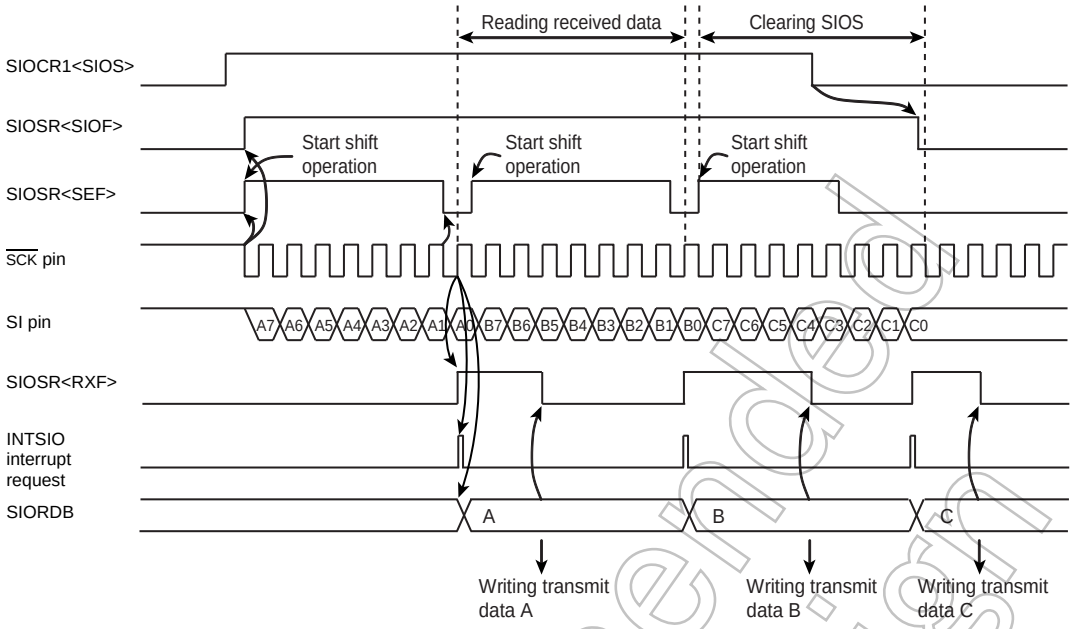


Figure 10-11 Example of External Clock and MSB Receive Mode

(4) Receive error processing

Receive errors occur on the following situation. To protect SIORDB and the shift register contents, the received data is ignored while the SIOSR<RXERR> is "1".

- Shift operation is finished before reading out received data from SIORDB at SIOSR<RXF> is "1" in an external clock operation. If receive error occurs, set the SIOCR1<SIOS> to "0" for reading the data that received immediately before error occurrence. And read the data from SIORDB. Data in shift register (at errors occur) can be read by reading the SIORDB again.

  When SIOSR<RXERR> is cleared to "0" after reading the received data, SIOSR<RXF> is cleared to "0". After clearing SIOCR1<SIOS> to "0", when 8-bit serial clock is input to $\overline{\text{SCK}}$ pin, receive operation is stopped. To restart the receive operation, confirm that SIOSR<SIOF> is cleared to "0".

  If the receive error occurs, set the SIOCR1<SIOINH> to "1" for stopping the receive operation immediately. In this case, SIOCR1<SIOS>, SIOSR register, SIORDB register and SIOTDB register are initialized.

Not Recommended for New Design

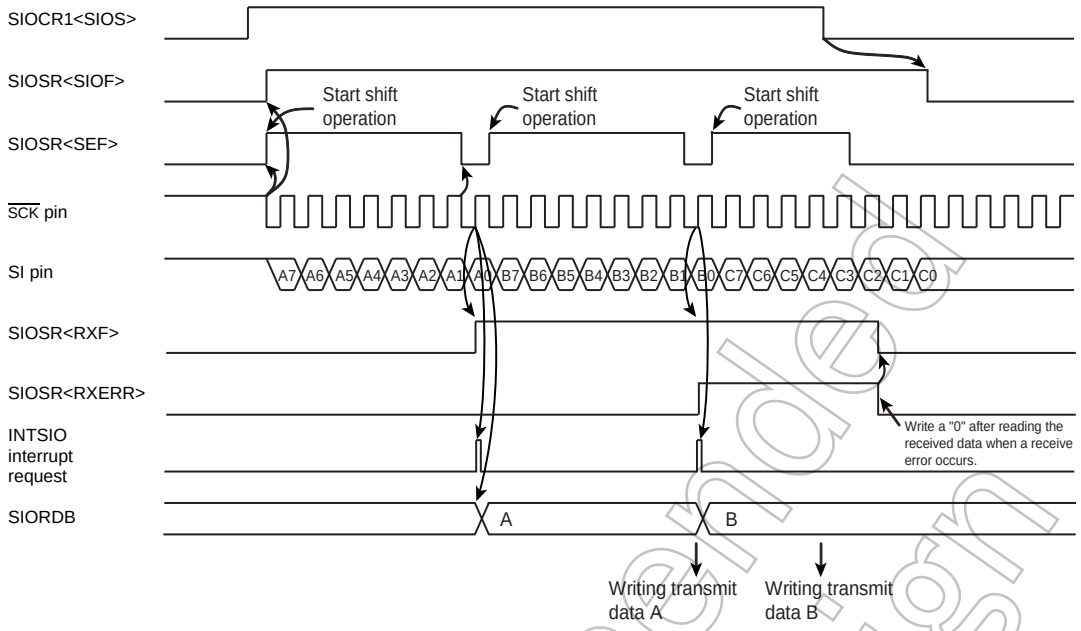


Figure 10-12 Example of Receive Error Processing

Note: If receive error is not corrected, an interrupt request does not generate after the error occurs.

10.3.3.3 Transmit/receive mode

The transmit/receive mode are selected by writing “10” to SIOCR1<SIOM>.

(1) Starting the transmit/receive operation

Transmit/receive mode is selected by writing “10B” to SIOCR1<SIOM>. Serial clock is selected by using SIOCR1<SCK>. Transfer direction is selected by using SIOCR1<SIODIR>.

When a transmit data is written to the transmit buffer register (SIOTDB), SIOSR<TXF> is cleared to “0”.

After SIOCR1<SIOS> is set to “1”, SIOSR<SIOF> is set synchronously to the falling edge of $\overline{\text{SCK}}$ pin.

The data is transferred sequentially starting from SO pin with the direction of the bit specified by SIOCR1<SIODIR>, synchronizing with the $\overline{\text{SCK}}$ pin's falling edge. And receiving operation also starts with the direction of the bit specified by SIOCR1<SIODIR>, synchronizing with the $\overline{\text{SCK}}$ pin's rising edge.

SIOSR<SEF> is kept in high level between the first clock falling edge of $\overline{\text{SCK}}$ pin and eighth clock falling edge.

SIOSR<TXF> is set to “1” at the rising edge of $\overline{\text{SCK}}$ pin after the data written to the SIOTDB is transferred to shift register. When 8-bit data has been received, the received data is transferred to SIORDB from shift register, then the INTSIO interrupt request occurs, synchronizing with setting SIOSR<RXF> to “1”.

Note 1: In internal clock operation, when the SIOCR1<SIOS> is set to "1", SIOTDB is transferred to shift register after maximum 1-cycle of serial clock frequency, then a serial clock is output from $\overline{\text{SCK}}$ pin.

Note 2: In external clock operation, when the falling edge is input from $\overline{\text{SCK}}$ pin after SIOCR1<SIOS> is set to "1", SIOTDB is transferred to shift register immediately. When the rising edge is input from $\overline{\text{SCK}}$ pin, receive operation also starts.

Not Recommended for New Design

(2) During the transmit/receive operation

When data is written to SIOTDB, SIOSR<TXF> is cleared to "0" and when a data is read from SIORDB, SIOSR<RXF> is cleared to "0".

In internal clock operation, in case of the condition described below, the serial clock stops to "H" level by an automatic-wait function when all of the bit set in the data has been transmitted.

- Next transmit data is not written to SIOTDB after reading a received data from SIORDB.
- Received data is not read from SIORDB after writing a next transmit data to SIOTDB.
- Neither SIOTDB nor SIORDB is accessed after transmission.

The automatic wait function is released by writing the next transmit data to SIOTDB after reading the received data from SIORDB, or reading the received data from SIORDB after writing the next data to SIOTDB. Then, transmit/receive operation is restarted after maximum 1 cycle of serial clock. In external clock operation, reading the received data from SIORDB and writing the next data to SIOTDB must be finished before the shift operation of the next data begins.

If the transmit data is not written to SIOTDB after SIOSR<TXF> is set to "1", transmit error occurs immediately after shift operation is started. When the transmit error occurred, SIOSR<TXERR> is set to "1". If received data is not read out from SIORDB before next shift operation starts after setting SIOSR<RXF> to "1", receive error occurs immediately after shift operation is finished. When the receive error has occurred, SIOSR<RXERR> is set to "1".

(3) Stopping the transmit/receive operation

There are two ways for stopping the transmit/receive operation.

- The way of clearing SIOCR1<SIOS>.

  When SIOCR1<SIOS> is cleared to "0", transmit/receive operation is stopped after all transfer of the data is finished. When transmit/receive operation is finished, SIOSR<SIOF> is cleared to "0" and SO pin is kept in high level. In external clock operation, SIOCR1<SIOS> must be cleared to "0" before SIOSR<SEF> is set to "1" by beginning next transfer.

- The way of setting SIOCR1<SIOINH>.

  Transmit/receive operation is stopped immediately after SIOCR1<SIOINH> is set to "1". In this case, SIOCR1<SIOS>, SIOSR register, SIORDB register and SIOTDB register are initialized.

Not Recommended for New Design

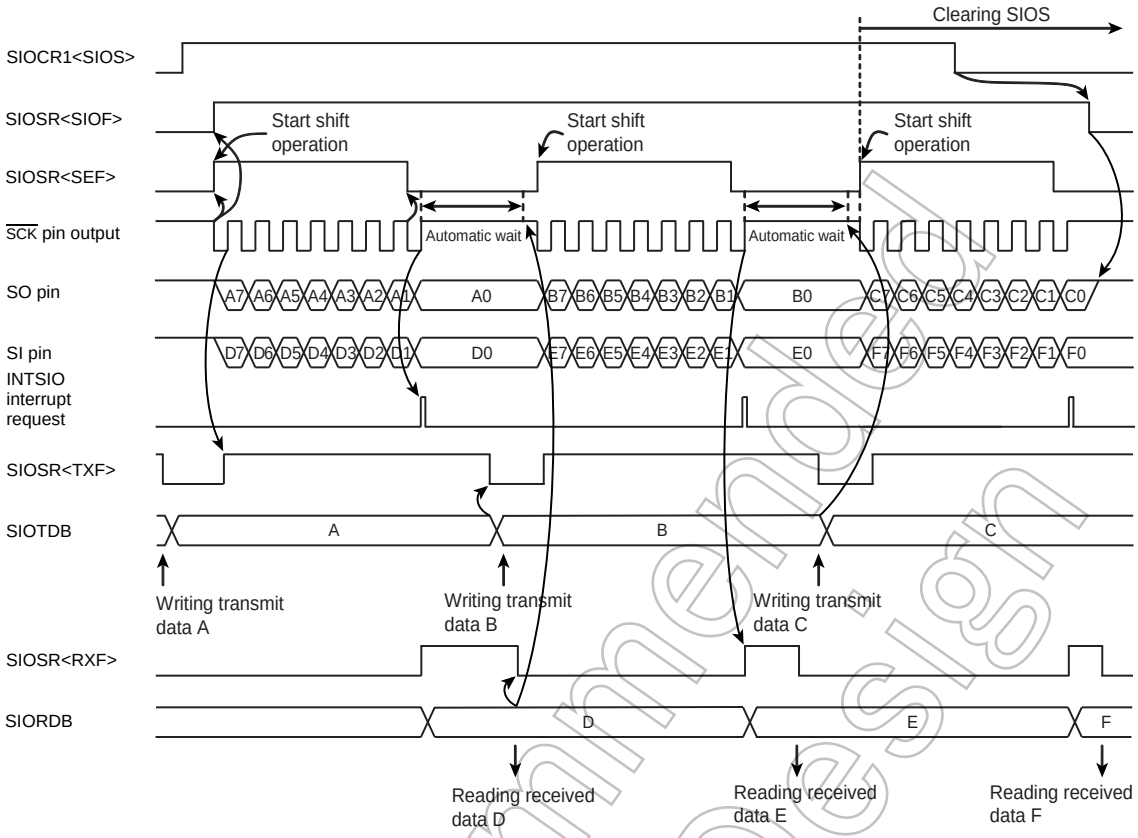


Figure 10-13 Example of Internal Clock and MSB Transmit/Receive Mode

Not Recommended for New Design

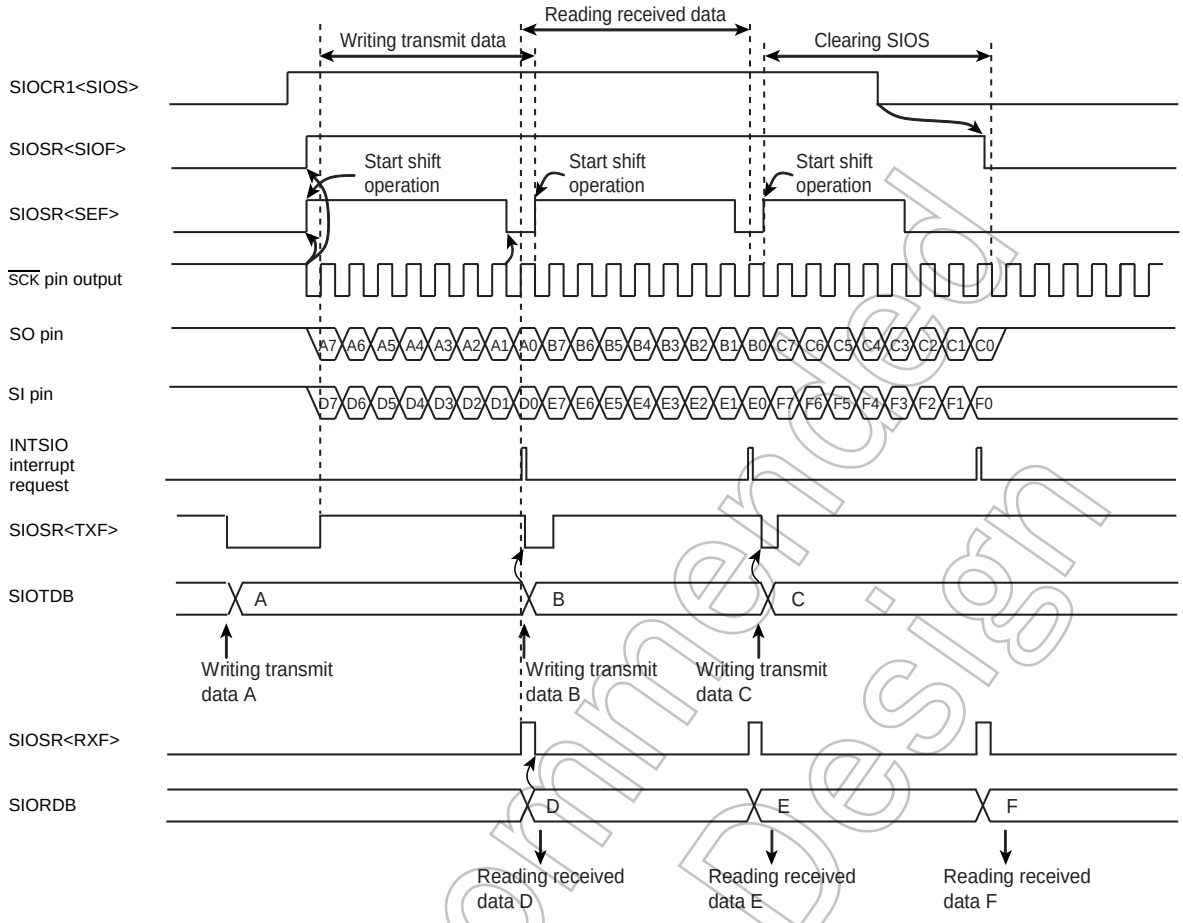


Figure 10-14 Example of External Clock and MSB Transmit/Receive Mode

(4) Transmit/receive error processing

Transmit/receive errors occur on the following situation. Corrective action is different, which errors occur transmits or receives.

(a) Transmit errors

Transmit errors occur on the following situation.

- Shift operation starts before writing next transmit data to SIOTDB in external clock operation.

  If transmit errors occur during transmit operation, SIOSR<TXERR> is set to "1" immediately after starting shift operation. And INTSIO interrupt request is generated after all of the 8-bit data has been received. If shift operation starts before writing data to SIOTDB after SIOCR1<SIOS> is set to "1", SIOSR<TXERR> is set immediately after starting shift operation. And INTSIO interrupt request is generated after all of the 8-bit data has been received.

  SO pin is kept in high level when SIOSR<TXERR> is set to "1". When transmit error occurs, transmit operation must be forcibly stop by writing SIOCR1<SIOINH> to "1" after the received data is read from SIORDB. In this case, SIOCR1<SIOS>, SIOSR register, SIORDB register and SIOTDB register are initialized.

Not Recommended for New Design

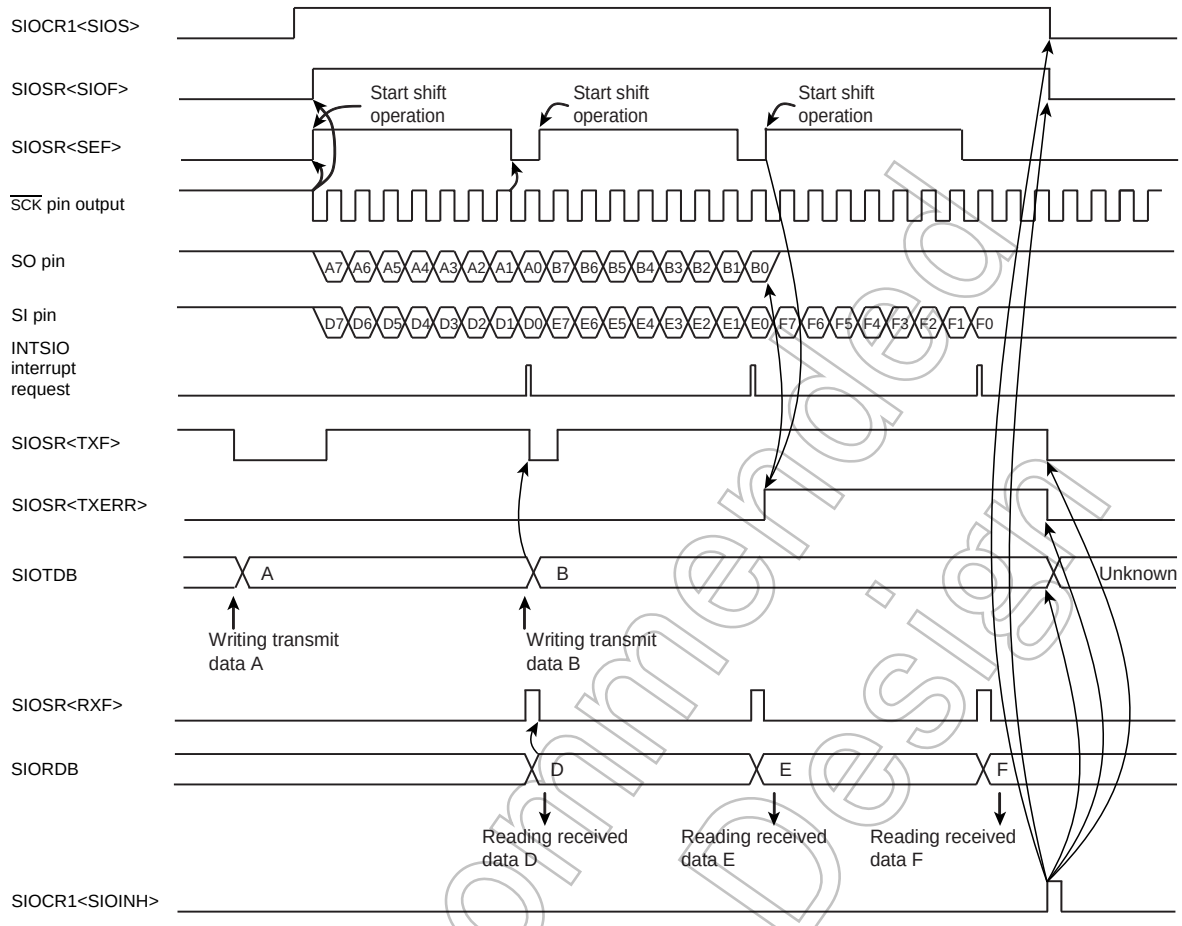


Figure 10-15 Example of Transmit/Receive (Transmit) Error Processing

(b) Receive errors

Receive errors occur on the following situation. To protect SIORDB and the shift register contents, the received data is ignored while the SIOSR<RXERR> is "1".

- Shift operation is finished before reading out received data from SIORDB at SIOSR<RXF> is "1" in an external clock operation. If receive error occurs, set the SIOCR1<SIOS> to "0" for reading the data that received immediately before error occurrence. And read the data from SIORDB. Data in shift register (at errors occur) can be read by reading the SIORDB again.

  When SIOSR<RXERR> is cleared to "0" after reading the received data, SIOSR<RXF> is cleared to "0". After clearing SIOCR1<SIOS> to "0", when 8-bit serial clock is input to SCK pin, receive operation is stopped. To restart the receive operation, confirm that SIOSR<SIOF> is cleared to "0".

  If the received error occurs, set the SIOCR1<SIOINH> to "1" for stopping the receive operation immediately. In this case, SIOCR1<SIOS>, SIOSR register, SIORDB register and SIOTDB register are initialized.

Not Recommended for New Design

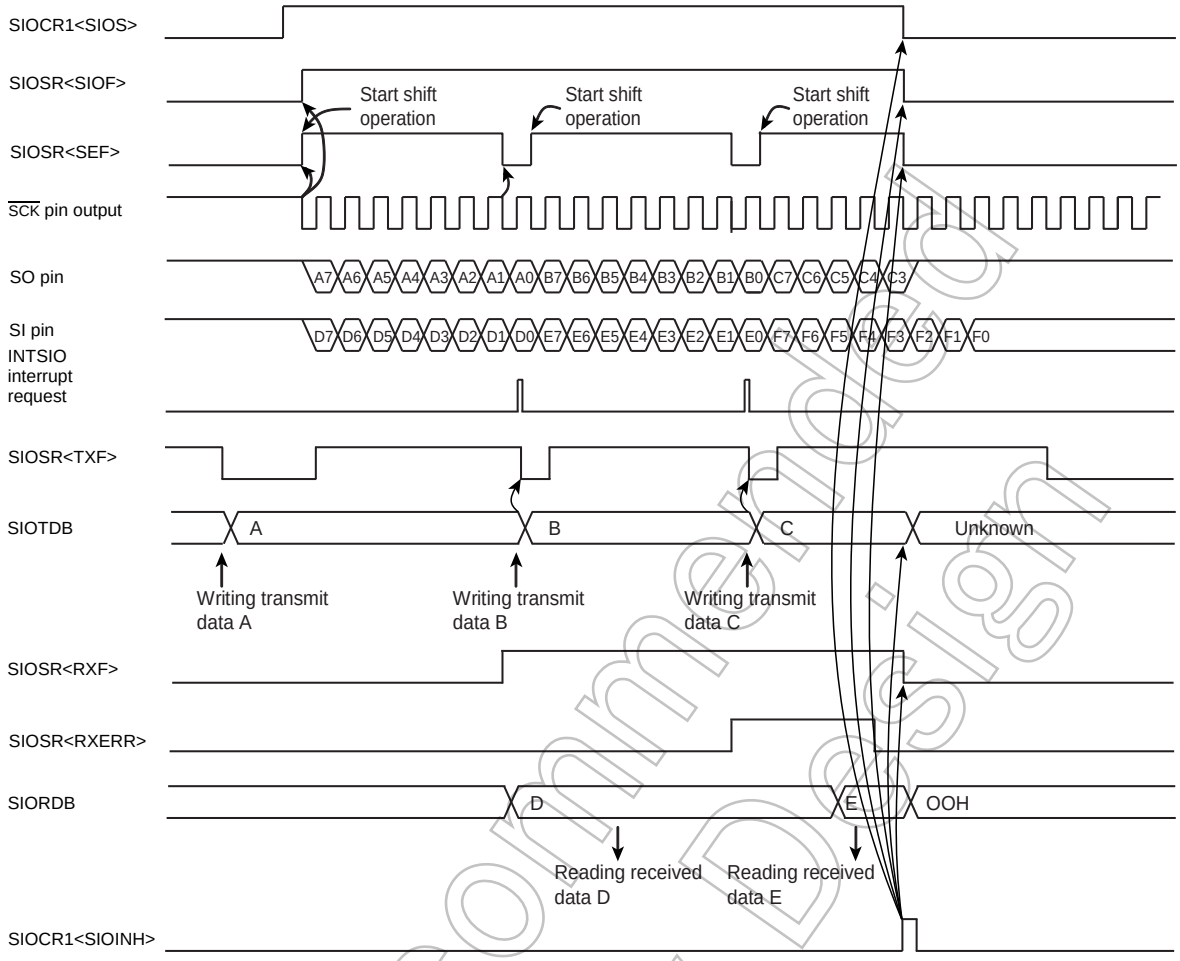


Figure 10-16 Example of Transmit/Receive (Receive) Error Processing

Note: If receive error is not corrected, an interrupt request does not generate after the error occurs.

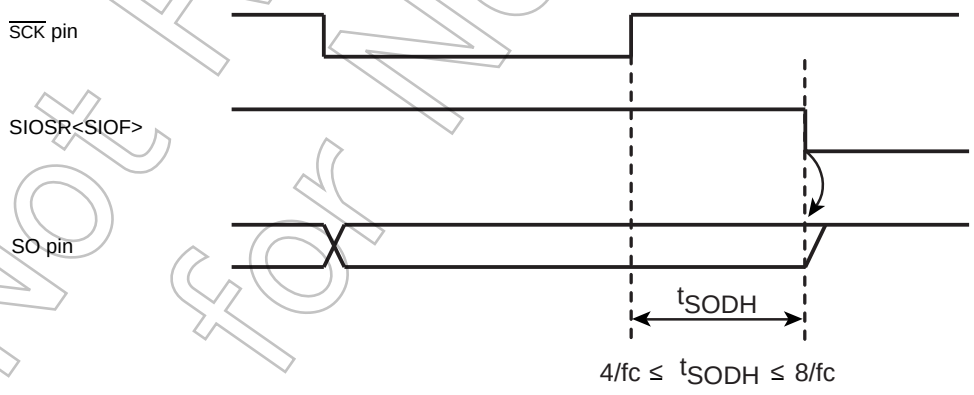


Figure 10-17 Hold Time of the End of Transmit/Receive Mode

Not Recommended for New Design

11. Asynchronous Serial interface (UART)

11.1 Configuration

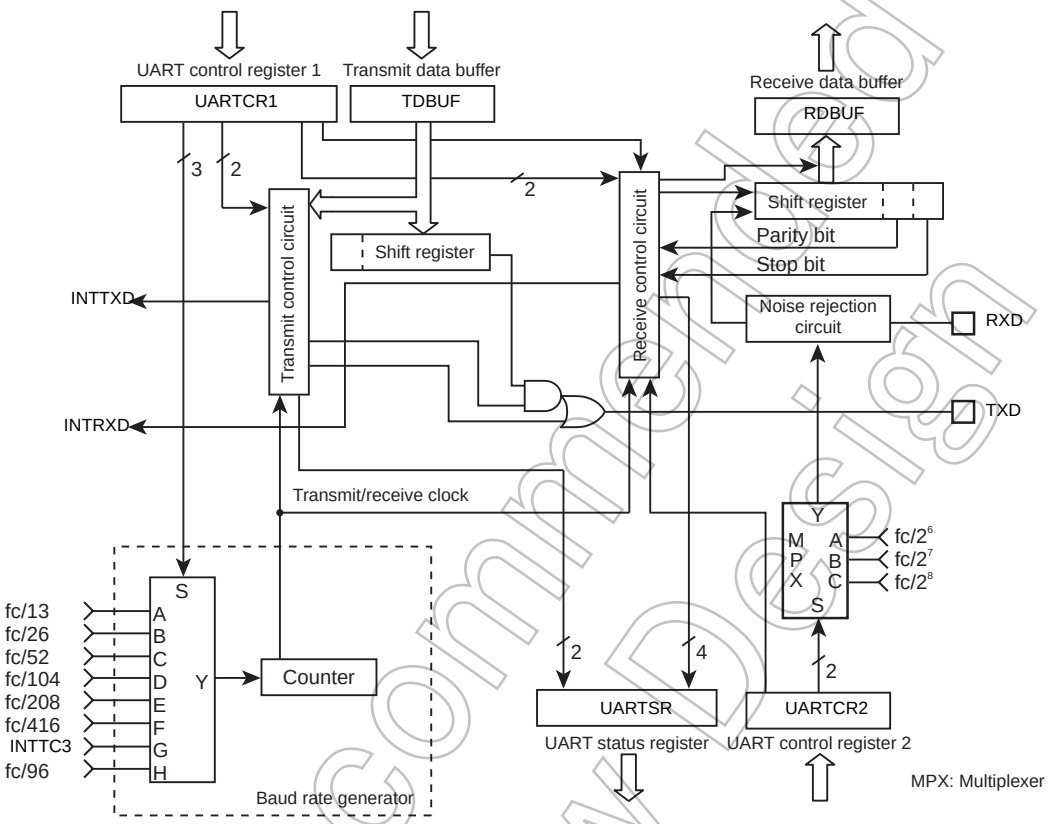


Figure 11-1 UART (Asynchronous Serial Interface)

Not Recommended for New Design

11.2 Control

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

UART Control Register1

UARTCR1 (0020H)

7	6	5	4	3	2	1	0
TXE	RXE	STBT	EVEN	PE	BRG		

(Initial value: 0000 0000)

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

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

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

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

Not Recommended for New Design

UART Control Register2

UARTCR2 (0021H)

7	6	5	4	3	2	1	0
					RXDNC	STOPBR	

(Initial value: **** *000)

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 "O" 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 (When the transfer clock generated by INTTC3 is the same as the right side column)	fc/8	O	-	-	-
	fc/16	O	O	-	-
	fc/32	O	O	O	-
The setting except the above		O	O	O	O

UART Status Register

UARTSR (0020H)

7	6	5	4	3	2	1	0
PERR	FERR	OERR	RBFL	TEND	TBEP		

(Initial value: 0000 11**)

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

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

UART Receive Data Buffer

RDBUF (0022H)

7	6	5	4	3	2	1	0

Read only

(Initial value: 0000 0000)

Not Recommended for New Design

UART Transmit Data Buffer

	7	6	5	4	3	2	1	0	
TDBUF									Write only
(0022H)									(Initial value: 0000 0000)

Not Recommended for New Design

11.3 Transfer Data Format

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

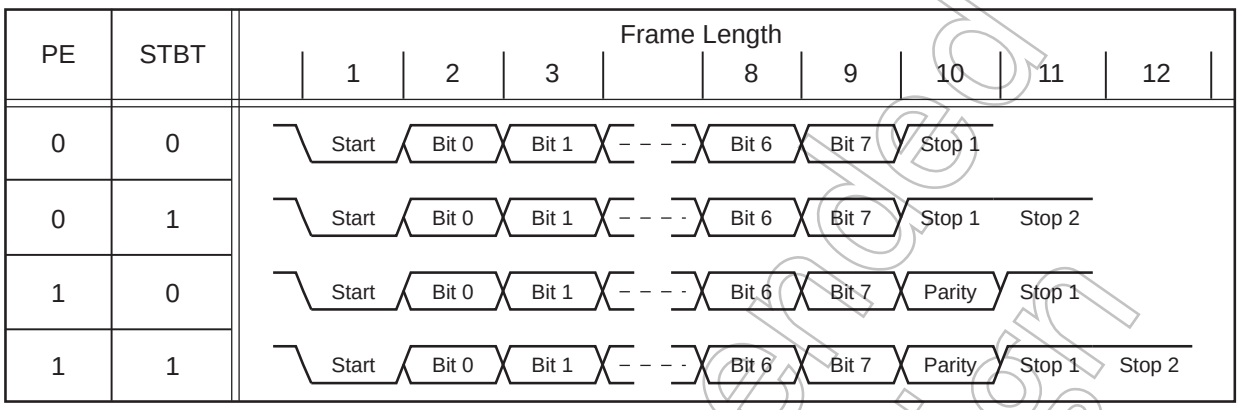


Figure 11-2 Transfer Data Format

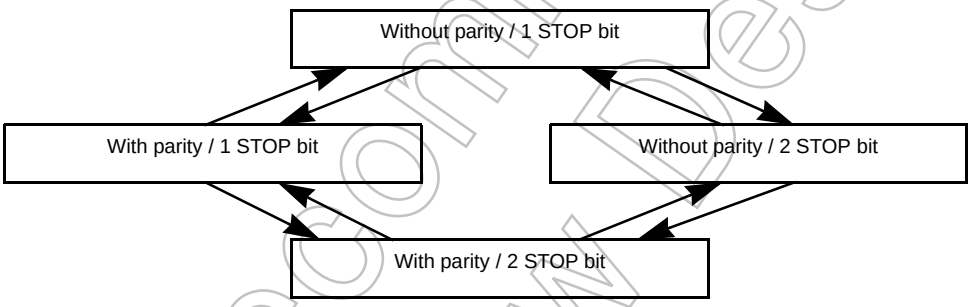


Figure 11-3 Caution on Changing Transfer Data Format

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

Not Recommended for New Design

11.4 Transfer Rate

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

Table 11-1 Transfer Rate (Example)

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

When TC3 is used as the UART transfer rate (when UARTCR1<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

11.5 Data Sampling Method

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

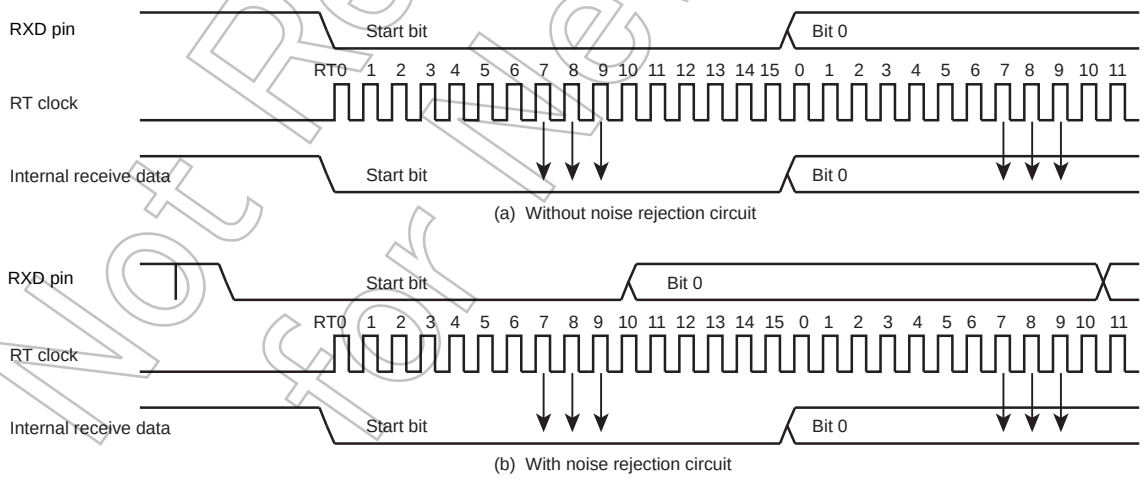


Figure 11-4 Data Sampling Method

Not Recommended for New Design

11.6 STOP Bit Length

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

11.7 Parity

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

11.8 Transmit/Receive Operation

11.8.1 Data Transmit Operation

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

While UARTCR1<TXE> = "0" and from when "1" is written to UARTCR1<TXE> to when send data are written to TDBUF, the TXD pin is fixed at high level. When transmitting data, first read UARTSR, then write data in TDBUF. Otherwise, UARTSR<TBEP> is not zero-cleared and transmit does not start.

11.8.2 Data Receive Operation

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

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

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

Not Recommended for New Design

11.9 Status Flag

11.9.1 Parity Error

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

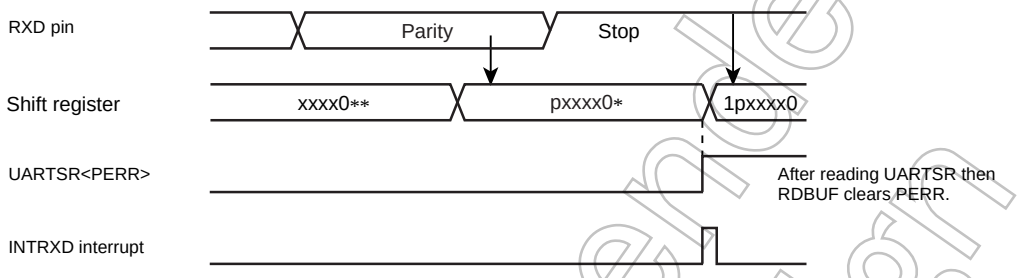


Figure 11-5 Generation of Parity Error

11.9.2 Framing Error

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

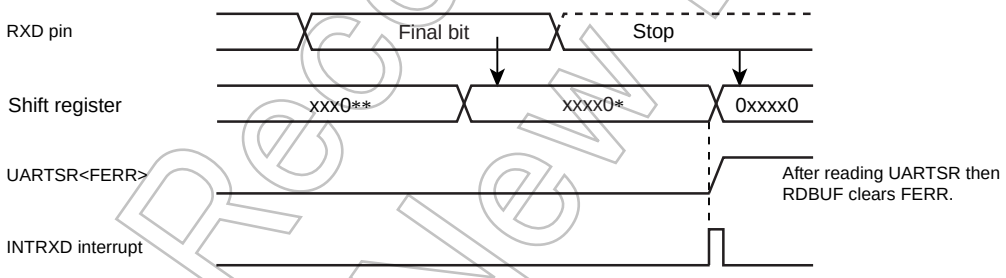


Figure 11-6 Generation of Framing Error

11.9.3 Overrun Error

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

Not Recommended for New Design

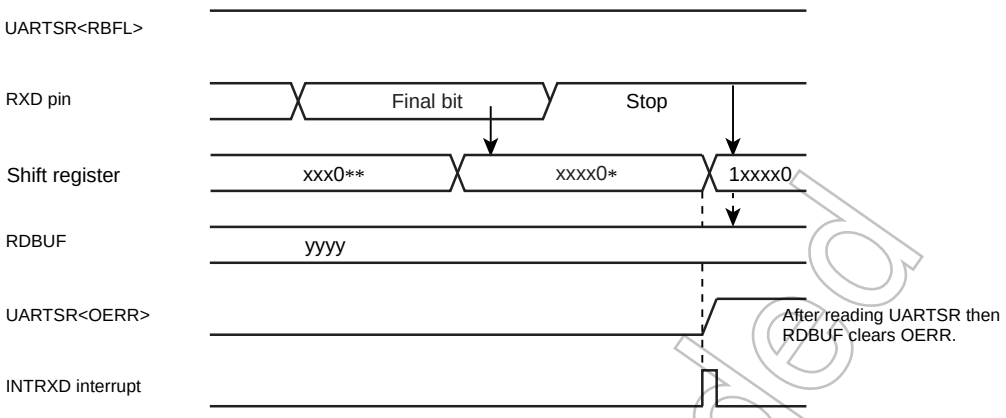


Figure 11-7 Generation of Overrun Error

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

11.9.4 Receive Data Buffer Full

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

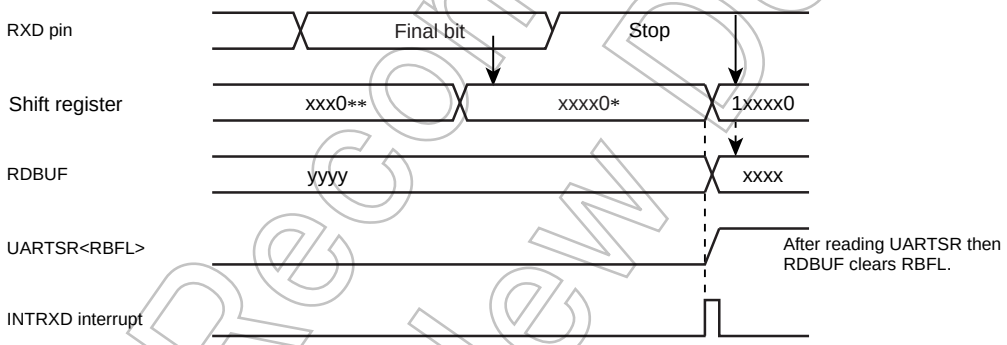


Figure 11-8 Generation of Receive Data Buffer Full

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

11.9.5 Transmit Data Buffer Empty

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

Not Recommended for New Design

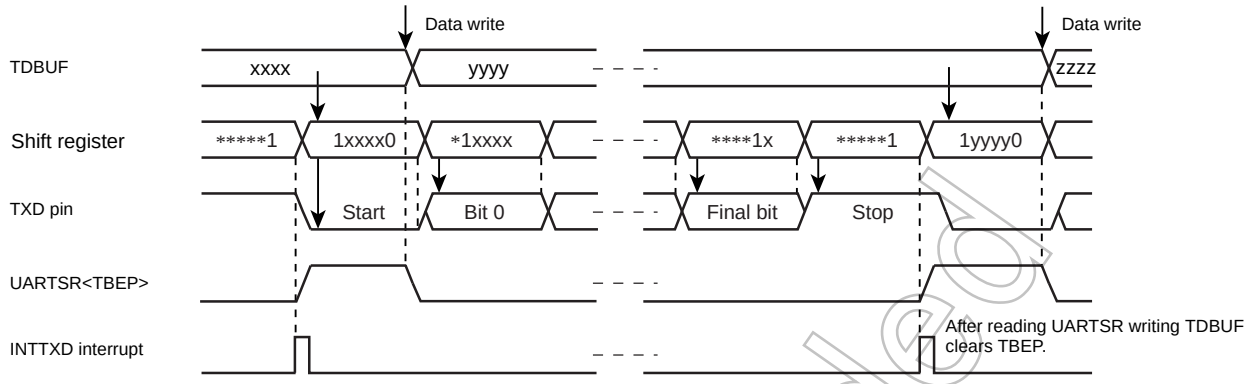


Figure 11-9 Generation of Transmit Data Buffer Empty

11.9.6 Transmit End Flag

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

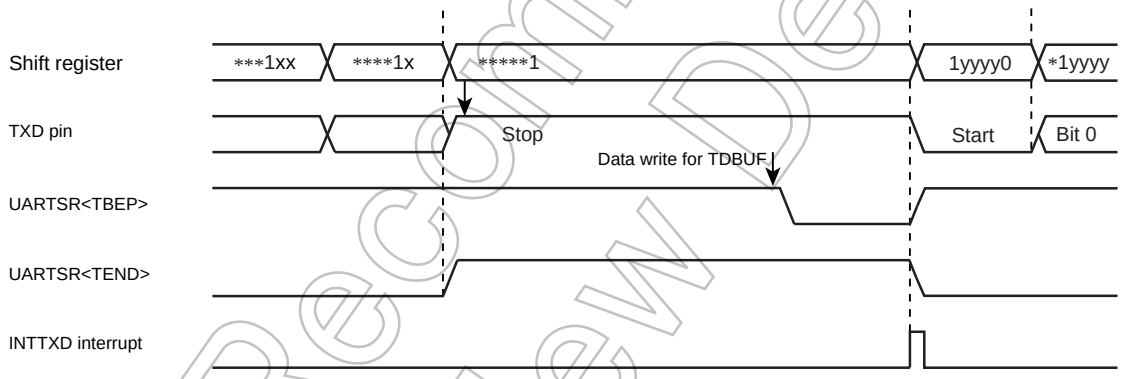


Figure 11-10 Generation of Transmit End Flag and Transmit Data Buffer Empty

Not Recommended for New Design

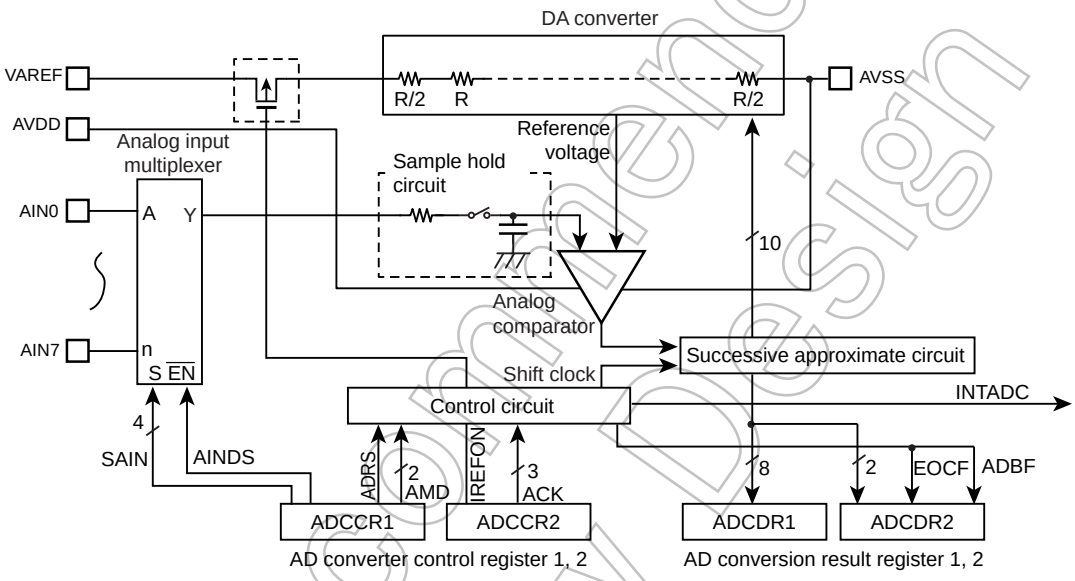
12. 10-bit AD Converter (ADC)

The TMP86FH46ANG have a 10-bit successive approximation type AD converter.

12.1 Configuration

The circuit configuration of the 10-bit AD converter is shown in Figure 12-1.

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 12-1 10-bit AD Converter

Not Recommended for New Design

12.2 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 (001CH)

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: AIN6 0111: AIN7 1000: Reserved 1001: Reserved 1010: Reserved 1011: Reserved 1100: Reserved 1101: Reserved 1110: Reserved 1111: Reserved	

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

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

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

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

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

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

Not Recommended for New Design

AD Converter Control Register 2

ADCCR2 (001DH)	7	6	5	4	3	2	1	0	
			IREFON	"1"	ACK			"0"	(Initial value: **0* 000*)

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

Note 1: Always set bit0 in ADCCR2 to "0" and set bit4 in ADCCR2 to "1".

Note 2: When a read instruction for ADCCR2, bit6 to 7 in ADCCR2 read in as undefined data.

Note 3: After STOP or SLOW/SLEEP mode are started, AD converter control register2 (ADCCR2) is all initialized and no data can be written in this register. Therefore, to use AD converter again, set the ADCCR2 newly after returning to NORMAL1 or NORMAL2 mode.

Table 12-1 ACK setting and Conversion time

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

Note 1: Setting for "−" in the above table are inhibited. fc: High Frequency oscillation clock [Hz]

Note 2: Set conversion time setting should be kept more than the following time by Analog reference voltage (VAREF) .

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

AD Converted value Register 1

ADCDR1 (001FH)	7	6	5	4	3	2	1	0	
	AD09	AD08	AD07	AD06	AD05	AD04	AD03	AD02	(Initial value: 0000 0000)

Not Recommended for New Design

AD Converted value Register 2

ADCDR2 (001EH)	7	6	5	4	3	2	1	0	
	AD01	AD00	EOCF	ADBF					(Initial value: 0000 ****)

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.

Not Recommended for New Design

12.3 Function

12.3.1 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).

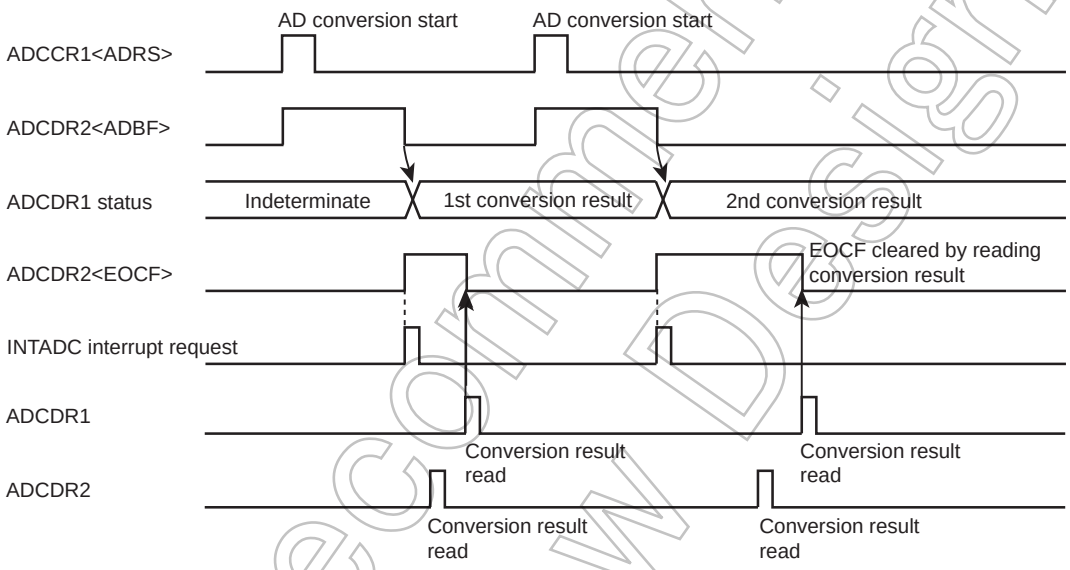


Figure 12-2 Software Start Mode

12.3.2 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.

Not Recommended for New Design

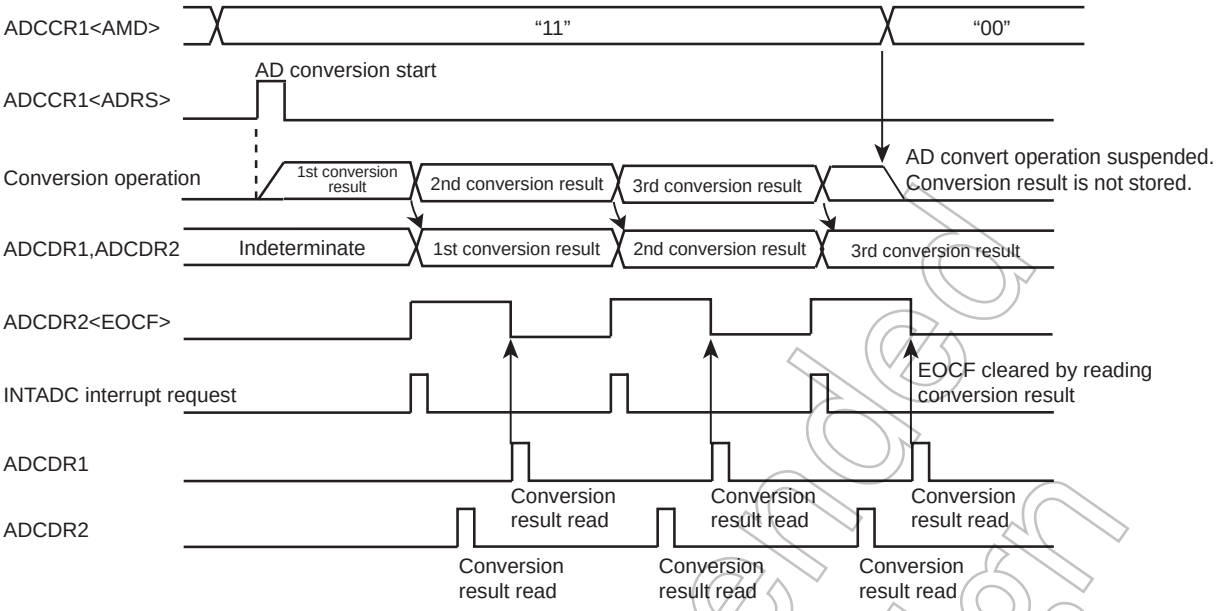


Figure 12-3 Repeat Mode

12.3.3 Register Setting

1. Set up the AD converter control register 1 (ADCCR1) as follows:
   - Choose the channel to AD convert using AD input channel select (SAIN).
   - Specify analog input enable for analog input control (AINDS).
   - Specify AMD for the AD converter control operation mode (software or repeat mode).
2. Set up the AD converter control register 2 (ADCCR2) as follows:
   - Set the AD conversion time using AD conversion time (ACK). For details on how to set the conversion time, refer to Figure 12-1 and AD converter control register 2.
   - Choose IREFON for DA converter control.
3. After setting up (1) and (2) above, set AD conversion start (ADRS) of AD converter control register 1 (ADCCR1) to "1". If software start mode has been selected, AD conversion starts immediately.
4. After an elapse of the specified AD conversion time, the AD converted value is stored in AD converted value register 1 (ADCDR1) and the AD conversion finished flag (EOCF) of AD converted value register 2 (ADCDR2) is set to "1", upon which time AD conversion interrupt INTADC is generated.
5. EOCF is cleared to "0" by a read of the conversion result. However, if reconverted before a register read, although EOCF is cleared the previous conversion result is retained until the next conversion is completed.

Not Recommended for New Design

Example :After selecting the conversion time 19.5 μs 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

                SET                 (ADCCR1) . 7            ; ADRS = 1(AD conversion start)
SLOOP :         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
```

12.4 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.

Not Recommended for New Design

12.5 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 12-4.

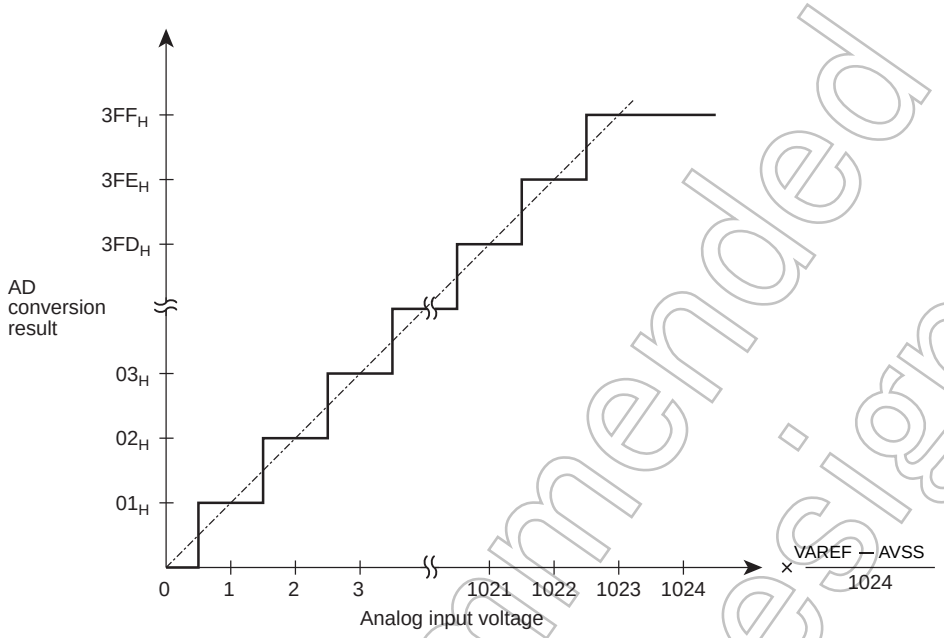


Figure 12-4 Analog Input Voltage and AD Conversion Result (Typ.)

Not Recommended for New Design

12.6 Precautions about AD Converter

12.6.1 Analog input pin voltage range

Make sure the analog input pins (AIN0 to AIN7) are used at voltages within VAREF to AVSS. If any voltage outside this range is applied to one of the analog input pins, the converted value on that pin becomes uncertain. The other analog input pins also are affected by that.

12.6.2 Analog input shared pins

The analog input pins (AIN0 to AIN7) are shared with input/output ports. When using any of the analog inputs to execute AD conversion, do not execute input/output instructions for all other ports. This is necessary to prevent the accuracy of AD conversion from degrading. Not only these analog input shared pins, some other pins may also be affected by noise arising from input/output to and from adjacent pins.

12.6.3 Noise Countermeasure

The internal equivalent circuit of the analog input pins is shown in Figure 12-5. The higher the output impedance of the analog input source, more easily they are susceptible to noise. Therefore, make sure the output impedance of the signal source in your design is 5 kΩ or less. Toshiba also recommends attaching a capacitor external to the chip.

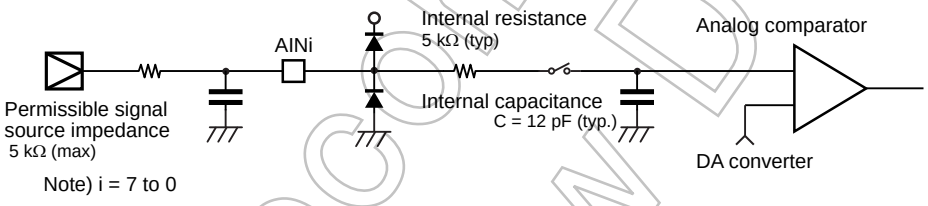


Figure 12-5 Analog Input Equivalent Circuit and Example of Input Pin Processing

Not Recommended for New Design

Not Recommended for New Design

13. Key-on Wakeup (KWU)

In the TMP86FH46ANG, the STOP mode is released by not only P20(INT5/STOP) pin but also four (STOP2 to STOP5) pins.

When the STOP mode is released by STOP2 to STOP5 pins, the STOP pin needs to be used. In details, refer to the following section "13.2 Control".

13.1 Configuration

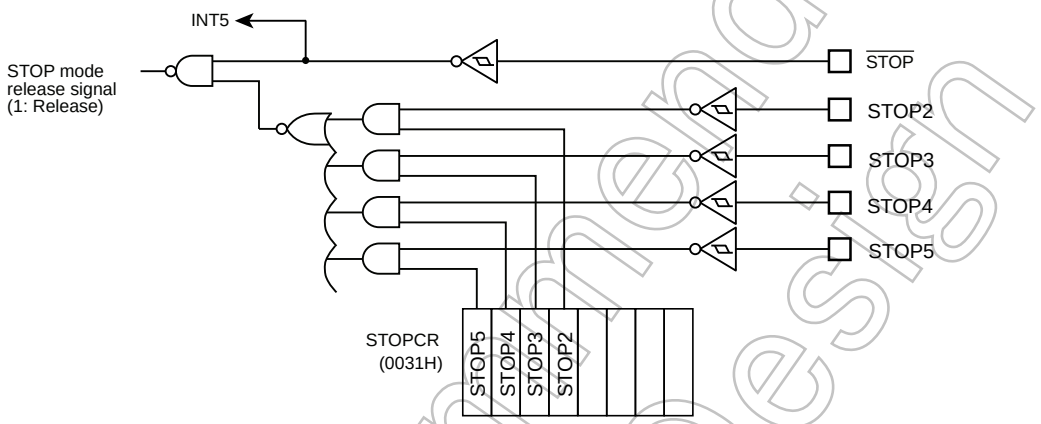


Figure 13-1 Key-on Wakeup Circuit

13.2 Control

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

Key-on Wakeup Control Register

STOPCR	7	6	5	4	3	2	1	0	
(0031H)	STOP5	STOP4	STOP3	STOP2					(Initial value: 0000 ****)

STOP5	STOP mode released by STOP5	0: Disable 1: Enable	Write only
STOP4 STOP4	STOP mode released by STOP4	0: Disable 1: Enable	Write only
STOP3	STOP mode released by STOP3	0: Disable 1: Enable	Write only
STOP2	STOP mode released by STOP2	0: Disable 1: Enable	Write only

13.3 Function

Stop mode can be entered by setting up the System Control Register (SYSCR1), and can be exited by detecting the "L" level on STOP2 to STOP5 pins, which are enabled by STOPCR, for releasing STOP mode (Note1).

Not Recommended for New Design

Also, each level of the STOP2 to STOP5 pins can be confirmed by reading corresponding I/O port data register, check all STOP2 to STOP5 pins "H" that is enabled by STOPCR before the STOP mode is started (Note2,3).

Note 1: When the STOP mode released by the edge release mode (SYSCR1<RELM> = “0”), inhibit input from STOP2 to STOP5 pins by Key-on Wakeup Control Register (STOPCR) or must be set "H" level into STOP2 to STOP5 pins that are available input during STOP mode.

Note 2: When the $\overline{STOP}$ pin input is high or STOP2 to STOP5 pins input which is enabled by STOPCR is low, executing an instruction which starts STOP mode will not place in STOP mode but instead will immediately start the release sequence (Warm up).

Note 3: The input circuit of Key-on Wakeup input and Port input is separated, so each input voltage threshold value is different. Therefore, a value comes from port input before STOP mode start may be different from a value which is detected by Key-on Wakeup input (Figure 13-2).

Note 4: $\overline{STOP}$ pin doesn’t have the control register such as STOPCR, so when STOP mode is released by STOP2 to STOP5 pins, $\overline{STOP}$ pin also should be used as STOP mode release function.

Note 5: In STOP mode, Key-on Wakeup pin which is enabled as input mode (for releasing STOP mode) by Key-on Wakeup Control Register (STOPCR) may generate the penetration current, so the said pin must be disabled AD conversion input (analog voltage input).

Note 6: When the STOP mode is released by STOP2 to STOP5 pins, the level of $\overline{STOP}$ pin should hold "L" level (Figure 13-3).

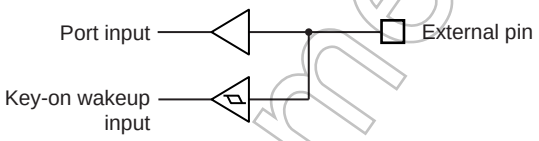


Figure 13-2 Key-on Wakeup Input and Port Input

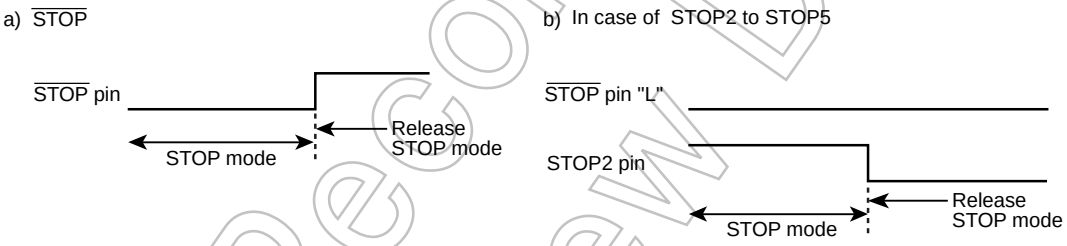


Figure 13-3 Priority of $\overline{STOP}$ pin and STOP2 to STOP5 pins

Table 13-1 Release level (edge) of STOP mode

Pin name	Release level (edge)	
	SYSCR1<RELM>="1" (Note2)	SYSCR1<RELM>="0"
$\overline{STOP}$	"H" level	Rising edge
STOP2	"L" level	Don’t use (Note1)
STOP3	"L" level	Don’t use (Note1)
STOP4	"L" level	Don’t use (Note1)
STOP5	"L" level	Don’t use (Note1)

Not Recommended for New Design

14. Flash Memory

TMP86FH46ANG 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. TMP86FH46ANG 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'.

Not Recommended for New Design

14.1 Flash Memory Control

The flash memory is controlled via the flash memory control register (FLSCR) and flash memory standby control resister (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: **** ***0)

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.

14.1.1 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.

14.1.2 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.

(Steps1 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.)

Not Recommended for New Design

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.

Not Recommended for New Design

14.2 Command Sequence

The command sequence in the MCU and the serial PROM modes consists of six commands (JEDEC compatible), as shown in Table 14-1. 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 14-1 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.

14.2.1 Byte Program

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

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

14.2.2 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).

Not Recommended for New Design

14.2.3 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.

14.2.4 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 14-2 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.

14.2.5 Product ID Exit

This command is used to exit the Product ID mode.

14.2.6 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 14-2.

It takes a maximum of 40 μs 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).

Not Recommended for New Design

14.3 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".

Not Recommended for New Design

14.4 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 BOOT-ROM 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.

14.4.1 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 BOOT-ROM 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 "14.4.1.1 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)".

14.4.1.1 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.

Not Recommended for New Design

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
```

Not Recommended for New Design

14.4.2 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.

14.4.2.1 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 micro-controller.

Not Recommended for New Design

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.
```

Not Recommended for New Design

15. Serial PROM Mode

15.1 Outline

The TMP86FH46ANG 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 15-1 shows memory address mapping in the serial PROM mode.

Table 15-1 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 15-5”.

15.2 Memory Mapping

The Figure 15-1 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.

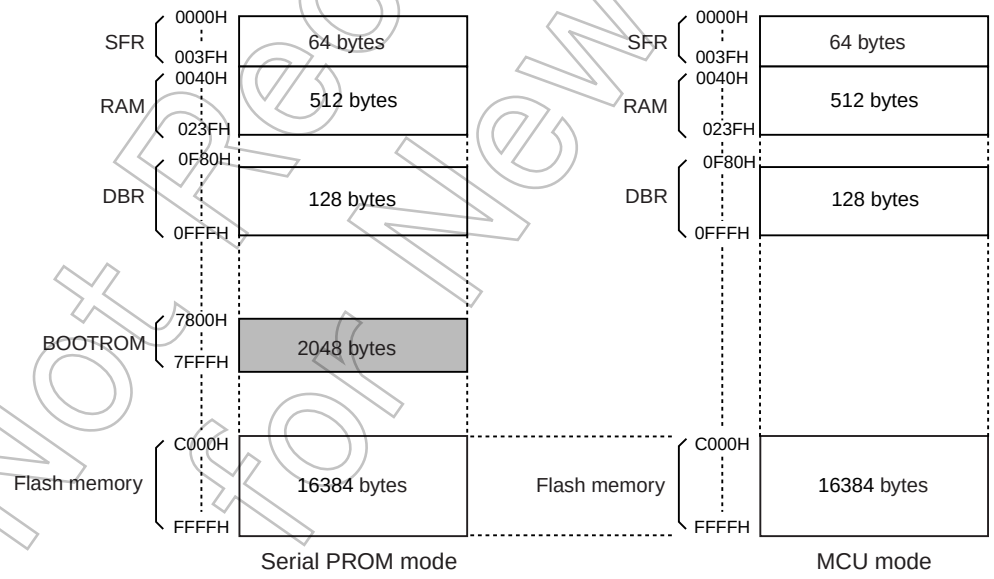


Figure 15-1 Memory Address Maps

Not Recommended for New Design

15.3 Serial PROM Mode Setting

15.3.1 Serial PROM Mode Control Pins

To execute on-board programming, activate the serial PROM mode. Table 15-2 shows pin setting to activate the serial PROM mode.

Table 15-2 Serial PROM Mode Setting

Pin	Setting
TEST pin	High
BOOT/RXD pin	High
$\overline{\text{RESET}}$ pin	

Note: The BOOT pin is shared with the UART communication pin (RXD pin) in the serial PROM mode. This pin is used as UART communication pin after activating serial PROM mode

15.3.2 Pin Function

In the serial PROM mode, TXD (P03) and RXD (P02) are used as a serial interface pin.

Table 15-3 Pin Function in the Serial PROM Mode

Pin Name (Serial PROM Mode)	Input/ Output	Function		Pin Name (MCU Mode)
TXD	Output	Serial data output	(Note 1)	P03
BOOT/RXD	Input/Input	Serial PROM mode control/Serial data input		P02
$\overline{\text{RESET}}$	Input	Serial PROM mode control		$\overline{\text{RESET}}$
TEST	Input	Fixed to high		TEST
VDD, AVDD	Power supply	4.5 to 5.5 V		
VSS, AVSS	Power supply	0 V		
VAREF	Power supply	Leave open or apply input reference voltage.		
I/O ports except P03, P02	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.

Not Recommended for New Design

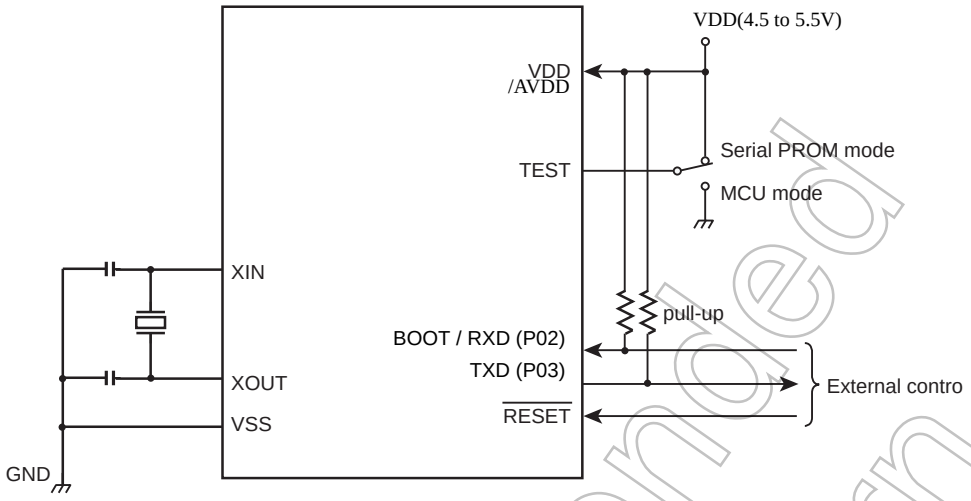


Figure 15-2 Serial PROM Mode Pin Setting

Note 1: For connection of other pins, refer to "Table 15-3 Pin Function in the Serial PROM Mode".

15.3.3 Example Connection for On-Board Writing

Figure 15-3 shows an example connection to perform on-board wring.

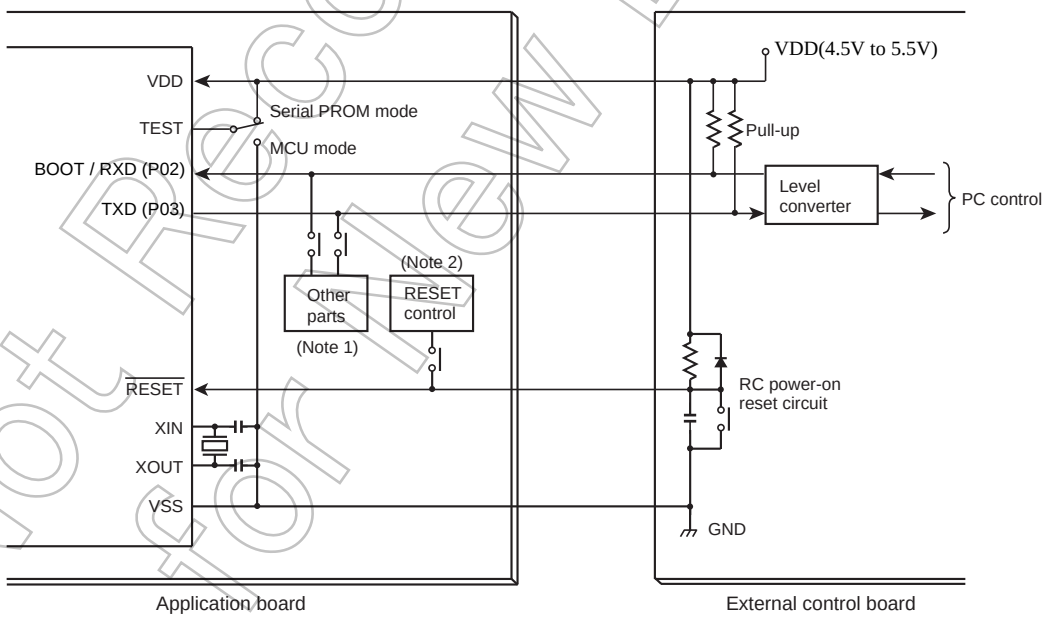


Figure 15-3 Example Connection for On-Board Writing

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 15-3 Pin Function in the Serial PROM Mode".

Not Recommended for New Design

15.3.4 Activating the Serial PROM Mode

The following is a procedure to activate the serial PROM mode. "Figure 15-4 Serial PROM Mode Timing" shows a serial PROM mode timing.

1. Supply power to the VDD pin.
2. Set the $\overline{\text{RESET}}$ pin to low.
3. Set the TEST pin and BOOT/RXD pins to high.
4. Wait until the power supply and clock oscillation stabilize.
5. Set the $\overline{\text{RESET}}$ pin to high.
6. Input the matching data (5AH) to the BOOT/RXD pin after setup sequence. For details of the setup timing, refer to "15.16 UART Timing".

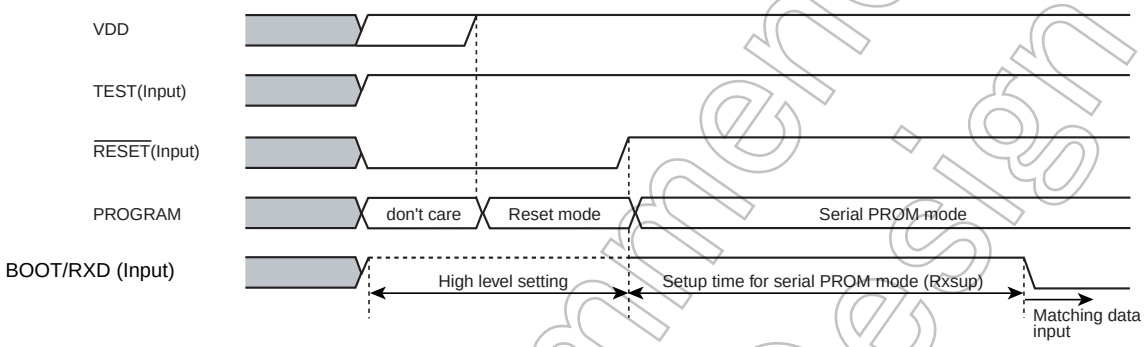


Figure 15-4 Serial PROM Mode Timing

Not Recommended for New Design

15.4 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 15-4 to TMP86FH46ANG. The Table 15-5 shows an operating frequency and baud rate. The frequencies which are not described in Table 15-5 can not be used.

- Baud rate (Default): 9600 bps
- Data length: 8 bits
- Parity addition: None
- Stop bit: 1 bit

Table 15-4 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 15-5 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 ±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.

Not Recommended for New Design

15.5 Operation Command

The eight commands shown in Table 15-6 are used in the serial PROM mode. After reset release, the TMP86FH46ANG waits for the matching data (5AH).

Table 15-6 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.

15.6 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, TMP86FH46ANG 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, TMP86FH46ANG 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, TMP86FH46ANG 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, TMP86FH46ANG 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

Not Recommended for New Design

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 TMP86FH46ANG, 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.

Not Recommended for New Design

15.6.1 Flash Memory Erasing Mode (Operating command: F0H)

Table 15-7 shows the flash memory erasing mode.

Table 15-7 Flash Memory Erasing Mode

	Transfer Byte	Transfer Data from the External Controller to TMP86FH46ANG	Baud Rate	Transfer Data from TMP86FH46ANG 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 15-4) -	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 "15.13 Specifying the Erasure Area".

Note 3: Refer to "15.8 Checksum (SUM)".

Note 4: Refer to "15.10 Passwords".

Note 5: Do not transmit the password string for a blank product.

Note 6: When a password error occurs, TMP86FH46ANG stops UART communication and enters the halt mode. Therefore, when a password error occurs, initialize TMP86FH46ANG by the $\overline{\text{RESET}}$ pin and reactivate the serial PROM mode.

Note 7: If an error occurs during transfer of a password address or a password string, TMP86FH46ANG stops UART communication and enters the halt condition. Therefore, when a password error occurs, initialize TMP86FH46ANG by the $\overline{\text{RESET}}$ pin and reactivate the serial PROM mode.

Not Recommended for New Design

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 15-6, 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 thorough 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 "15.13 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 "15.8 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.

Not Recommended for New Design

15.6.2 Flash Memory Writing Mode (Operation command: 30H)

Table 15-8 shows flash memory writing mode process.

Table 15-8 Flash Memory Writing Mode Process

	Transfer Byte	Transfer Data from External Controller to TMP86FH46ANG	Baud Rate	Transfer Data from TMP86FH46ANG 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 15-4) -	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 "15.7 Error Code".

Note 2: Refer to "15.9 Intel Hex Format (Binary)".

Note 3: Refer to "15.8 Checksum (SUM)".

Note 4: Refer to "15.10 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, TMP86FH46ANG stops UART communication and enters the halt condition. Therefore, when a password error occurs, initialize TMP86FH46ANG by the $\overline{\text{RESET}}$ pin and reactivate the serial ROM mode.

Note 6: If the security program is enabled or a password error occurs, TMP86FH46ANG stops UART communication and enters the halt condition. In this case, initialize TMP86FH46ANG by the $\overline{\text{RESET}}$ pin and reactivate the serial ROM mode.

Not Recommended for New Design

Note 7: If an error occurs during the reception of a password address or a password string, TMP86FH46ANG stops UART communication and enters the halt condition. In this case, initialize TMP86FH46ANG by the $\overline{\text{RESET}}$ 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, TMP86FH46ANG (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 15-5.
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 15-4 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 15-6, 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

Not Recommended for New Design

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 "15.8 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.

Not Recommended for New Design

15.6.3 RAM Loader Mode (Operation Command: 60H)

Table 15-9 shows RAM loader mode process.

Table 15-9 RAM Loader Mode Process

	Transfer Bytes	Transfer Data from External Controller to TMP86FH46ANG	Baud Rate	Transfer Data from TMP86FH46ANG 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 15-4) -	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 "15.7 Error Code".

Note 2: Refer to "15.9 Intel Hex Format (Binary)".

Note 3: Refer to "15.8 Checksum (SUM)".

Note 4: Refer to "15.10 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, TMP86FH46ANG stops UART communication and enters the halt condition. Therefore, when a password error occurs, initialize TMP86FH46ANG by the $\overline{\text{RESET}}$ pin and reactivate the serial ROM mode.

Note 6: After transmitting a password string, the external controller must not transmit only an end record. If receiving an end record after a password string, the device may not operate correctly.

Note 7: If the security program is enabled or a password error occurs, TMP86FH46ANG stops UART communication and enters the halt condition. In this case, initialize TMP86FH46ANG by the $\overline{\text{RESET}}$ pin and reactivate the serial PROM mode.

Not Recommended for New Design

Note 8: If an error occurs during the reception of a password address or a password string, TMP86FH46ANG stops UART communication and enters the halt condition. In this case, initialize TMP86FH46ANG by the $\overline{\text{RESET}}$ 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 15-6, 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 "15.8 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.

Not Recommended for New Design

15.6.4 Flash Memory SUM Output Mode (Operation Command: 90H)

Table 15-10 shows flash memory SUM output mode process.

Table 15-10 Flash Memory SUM Output Process

	Transfer Bytes	Transfer Data from External Controller to TMP86FH46ANG	Baud Rate	Transfer Data from TMP86FH46ANG 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 15-4)	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 "15.7 Error Code".

Note 2: Refer to "15.8 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 15-6, 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 "15.8 Checksum (SUM)".
5. After sending the checksum, the device waits for the next operation command data.

Not Recommended for New Design

15.6.5 Product ID Code Output Mode (Operation Command: C0H)

Table 15-11 shows product ID code output mode process.

Table 15-11 Product ID Code Output Process

	Transfer Bytes	Transfer Data from External Controller to TMP86FH46ANG	Baud Rate	Transfer Data from TMP86FH46ANG 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 15-4)	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 "15.7 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 15-6, 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 "15.11 Product ID Code".
5. After sending the checksum, the device waits for the next operation command data.

Not Recommended for New Design

15.6.6 Flash Memory Status Output Mode (Operation Command: C3H)

Table 15-12 shows Flash memory status output mode process.

Table 15-12 Flash Memory Status Output Mode Process

	Transfer Bytes	Transfer Data from External Controller to TMP86FH46ANG	Baud Rate	Transfer Data from TMP86FH46ANG 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 15-4) -	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 "15.7 Error Code".

Note 2: For the details on status code 1, refer to "15.12 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 15-6, 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 "15.12 Flash Memory Status Code".
5. After sending the status code, the device waits for the next operation command data.

Not Recommended for New Design

15.6.7 Flash Memory security program Setting Mode (Operation Command: FAH)

Table 15-13 shows Flash memory security program setting mode process.

Table 15-13 Flash Memory security program Setting Mode Process

	Transfer Bytes	Transfer Data from External Controller to TMP86FH46ANG	Baud Rate	Transfer Data from TMP86FH46ANG 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 15-4) -	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 "15.7 Error Code".

Note 2: Refer to "15.10 Passwords".

Note 3: If the security program is enabled for a blank product or a password error occurs for a non-blank product, TMP86FH46ANG stops UART communication and enters the halt mode. In this case, initialize TMP86FH46ANG by the $\overline{\text{RESET}}$ pin and reactivate the serial PROM mode.

Note 4: If an error occurs during reception of a password address or a password string, TMP86FH46ANG stops UART communication and enters the halt mode. In this case, initialize TMP86FH46ANG by the $\overline{\text{RESET}}$ pin and reactivate the serial PROM mode.

Description of the Flash memory security program setting mode

1. The 1st through 4th bytes of the transmitted and received data contain the same data as in the Flash memory writing mode.
2. The 5th byte of the received data contains the command data in the flash memory status output mode (FAH).

Not Recommended for New Design

3. When the 5th byte of the received data contains the operation command data shown in Table 15-6, 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.

Not Recommended for New Design

15.7 Error Code

When detecting an error, the device transmits the error code to the external controller, as shown in Table 15-14.

Table 15-14 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, TMP86FH46ANG does not transmit an error code.

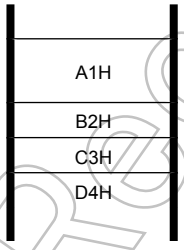
15.8 Checksum (SUM)

15.8.1 Calculation Method

The checksum (SUM) is calculated with the sum of all bytes, and the obtained result is returned as a word.

The data is read for each byte unit and the calculated result is returned as a word.

Example:



If the data to be calculated consists of the four bytes,
the checksum of the data is as shown below.

A1H + B2H + C3H + D4H = 02EAH
SUM (HIGH)= 02H
SUM (LOW)= EAH

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.

Not Recommended for New Design

15.8.2 Calculation data

The data used to calculate the checksum is listed in Table 15-15.

Table 15-15 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 "15.11 Product ID Code".
Flash Memory Status Output mode	9th through 12th bytes of the transferred data	For details, refer to "15.12 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.

Not Recommended for New Design

15.9 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

15.10 Passwords

The consecutive eight or more-byte data in the flash memory area can be specified to the password. TMP86FH46ANG 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 15-16 shows the password setting in the blank product and non-blank product.

Table 15-16 Password Setting in the Blank Product and Non-Blank Product

Password	Blank Product (Note 1)	Non-Blank Product
PNSA (Password count storage address)	C000H ≤ PNSA ≤ FF9FH	C000H ≤ PNSA ≤ FF9FH
PCSA (Password comparison start address)	C000H ≤ PCSA ≤ FF9FH	C000H ≤ PCSA ≤ FFA0 − N
N (Password count)	*	8 ≤ 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. TMP86FH46ANG 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.

Not Recommended for New Design

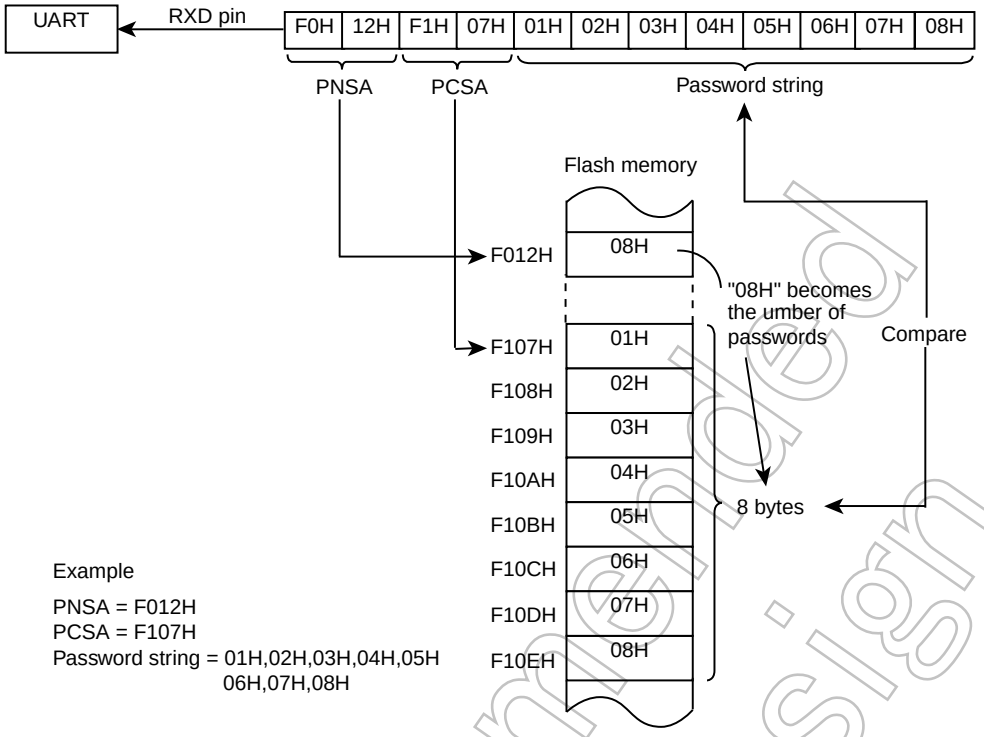


Figure 15-5 Password Comparison

15.10.1 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.

15.10.2 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.

15.10.3 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
```

Not Recommended for New Design

15.11 Product ID Code

The product ID code is the 13-byte data containing the start address and the end address of ROM. Table 15-17 shows the product ID code format.

Table 15-17 Product ID Code Format

Data	Description	In the Case of TMP86FH46ANG
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

15.12 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 15-18 shows the flash memory status code.

Table 15-18 Flash Memory Status Code

Data	Description	In the Case of TMP86FH46ANG	
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 00H 01H 02H 03H	checksum 00H FFH FEH FDH

Not Recommended for New Design

Status Code 1

7	6	5	4	3	2	1	0	
						RPENA	BLANK	(Initial Value: 0000 00**)

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, TMP86FH46ANG stops UART communication and enters the halt condition.)

Not Recommended for New Design

15.13 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)

7	6	5	4	3	2	1	0
ERASTA				ERAEND			

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, TMP86FH46ANG stops the UART communication and enters the halt condition.

15.14 Port Input Control Register

In the serial PROM mode, the input level is fixed to the all ports except P03 and P02 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.

Not Recommended for New Design

Port Input Control Register

SPCR (0FEAH)	7	6	5	4	3	2	1	0	(Initial value: **** ***0)
								PIN	

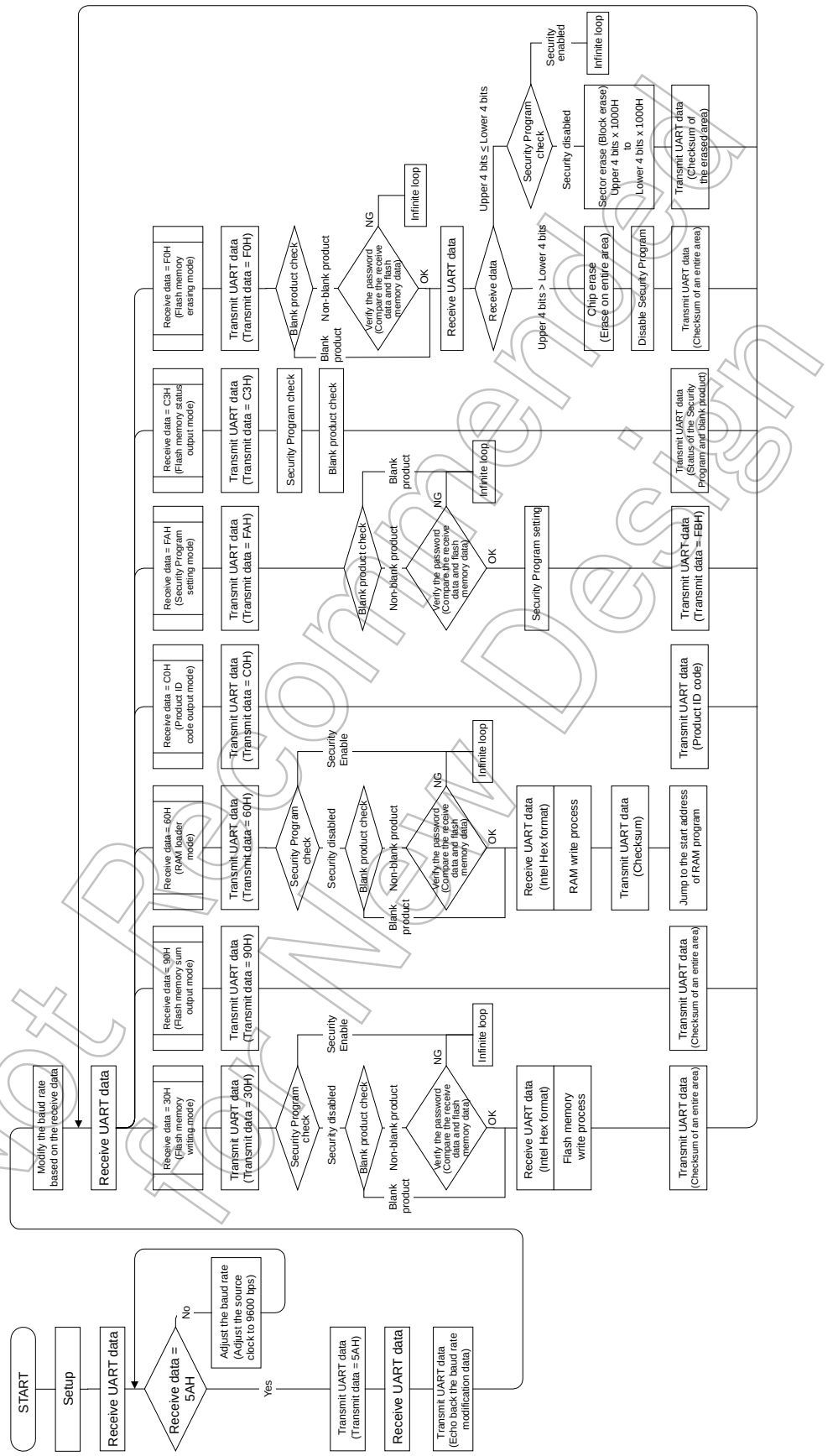
PIN	Port input control in the serial PROM mode	0: Invalid port inputs (The input level is fixed with a hardware feature.) 1: Valid port inputs	R/W

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 P03 and P02 ports are controlled by the SPCR register.

Not Recommended for New Design

15.15 Flowchart



Not Recommended for New Design

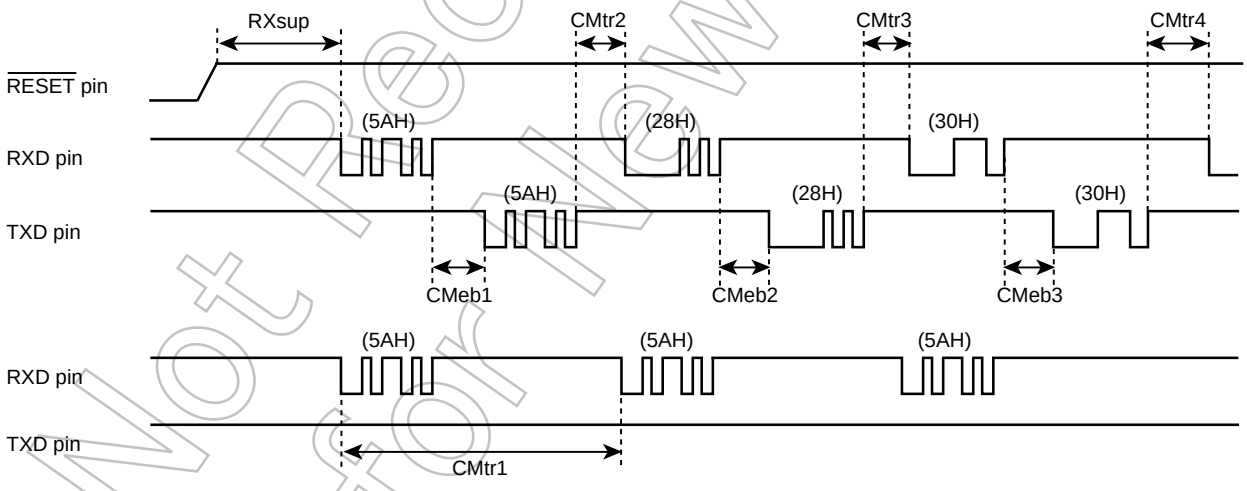
15.16 UART Timing

Table 15-19 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
Erasure time of an entire flash memory	CEall	-	30 ms	30 ms
Erasure time for a sector of a flash memory (in 4-kbyte units)	CEsec	-	15 ms	15 ms

Table 15-20 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



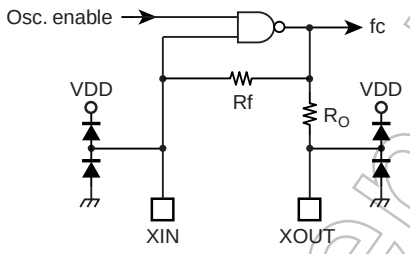
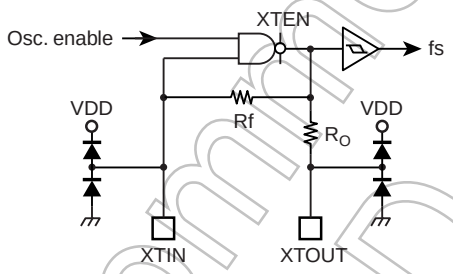
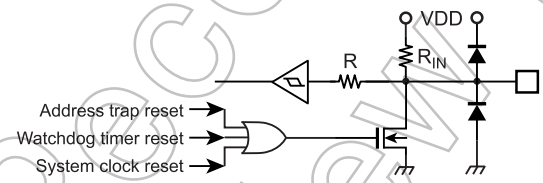
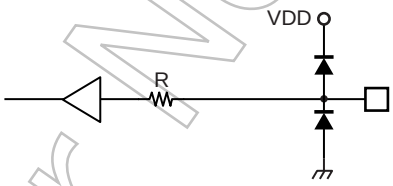
Not Recommended for New Design

Not Recommended for New Design

16. Input/Output Circuitry

16.1 Control Pins

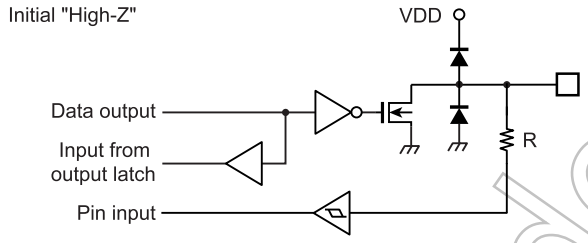
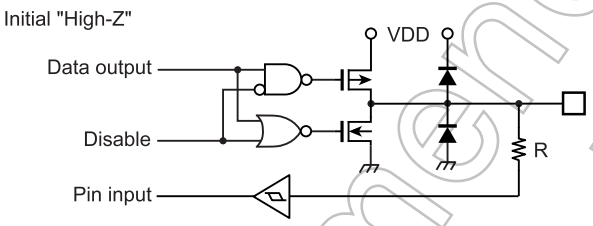
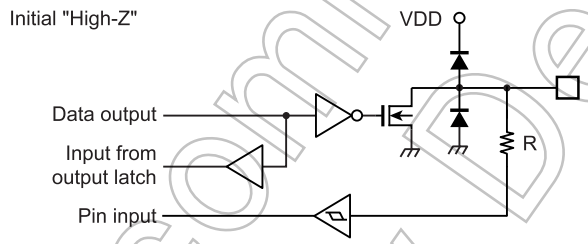
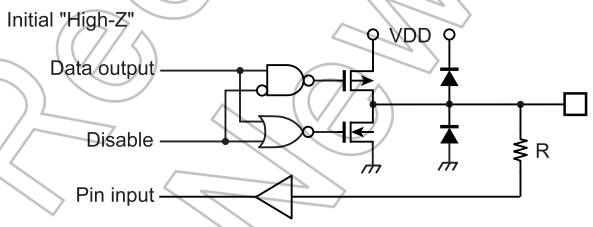
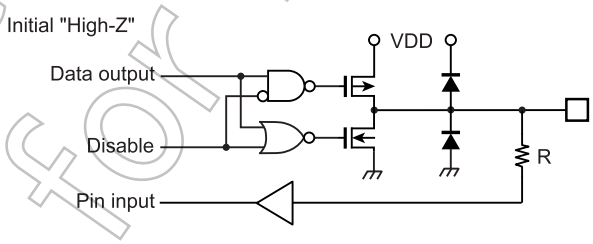
The input/output circuitries of the TMP86FH46ANG control pins are shown below.

Control Pin	I/O	Input/Output Circuitry	Remarks
XIN XOUT	Input Output		Resonator connecting pins (high-frequency) R_f = 1.2 MΩ (typ.) R_O = 1.5 kΩ (typ.)
XTIN XTOUT	Input Output		Resonator connecting pins (low-frequency) R_f = 6 MΩ (typ.) R_O = 220 kΩ (typ.)
$\overline{\text{RESET}}$	I/O		Hysteresis input Pull-up resistor R_{IN} = 220 kΩ (typ.) R = 1 kΩ (typ.)
TEST	Input		Without Pull-down resistor R = 1 kΩ (typ.) Fix the TEST pin at low-level

Note: The TEST pin of the TMP86FH46ANG does not have a pull-down resistor. Fix the TEST pin at low-level.

Not Recommended for New Design

16.2 Input/Output Ports

Port	I/O	Input/Output Circuitry	Remarks
P0	I/O		Sink open drain output High current output Hysteresis input R = 100 Ω (typ.)
P1	I/O		Tri-state I/O Hysteresis input R = 100 Ω (typ.)
P2	I/O		Sink open drain output High current output Hysteresis input R = 100 Ω (typ.)
P3	I/O		Tri-state I/O R = 100 Ω (typ.)
P4	I/O		Tri-state I/O High current output (Nch) R = 100 Ω (typ.)

Not Recommended for New Design

17. Electrical Characteristics

17.1 Absolute Maximum Ratings

The absolute maximum ratings are rated values which must not be exceeded during operation, even for an instant. Any one of the ratings must not be exceeded. If any absolute maximum rating is exceeded, a device may break down or its performance may be degraded, causing it to catch fire or explode resulting in injury to the user. Thus, when designing products which include this device, ensure that no absolute maximum rating value will ever be exceeded.

(VSS = 0 V)

Parameter	Symbol	Pins	Ratings	Unit
Supply voltage	V_{DD}		-0.3 to 6.5	V
Input voltage	V_{IN}		-0.3 to V_{DD} + 0.3	V
Output voltage	V_{OUT1}		-0.3 to V_{DD} + 0.3	V
Output current (Per 1 pin)	I_{OUT1}	P1, P3, P4 ports	-1.8	mA
	I_{OUT2}	P1, P3 ports	3.2	
	I_{OUT3}	P0, P2, P4 ports	30	
Output current (Total)	ΣI_{OUT1}	P1, P3 ports	60	
	ΣI_{OUT2}	P0, P2, P4 ports	80	
Power dissipation [Topr = 85 °C]	P_D		250	mW
Soldering temperature (time)	Tsld		260 (10 s)	°C
Storage temperature	Tstg		-55 to 125	
Operating temperature	Topr		-40 to 85	

Not Recommended for New Design

17.2 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.

17.2.1 Serial PROM mode

(V_{SS} = 0 V, Topr = −10 to 40 °C)

Parameter	Symbol	Pins	Condition	Min	Max	Unit
Supply voltage	V_{DD}		NORMAL1, 2 modes	4.5	5.5	V
Input high voltage	V_{IH1}	Except hysteresis input	$V_{DD} \geq 4.5$ V	$V_{DD} \times 0.70$	V_{DD}	
	V_{IH2}	Hysteresis input		$V_{DD} \times 0.75$		
Input low voltage	V_{IL1}	Except hysteresis input	$V_{DD} \geq 4.5$ V	0	$V_{DD} \times 0.30$	
	V_{IL2}	Hysteresis input			$V_{DD} \times 0.25$	
Clock frequency	fc	XIN, XOUT		2.0	16.0	MHz

17.2.2 MCU mode (Except Flash Programming or erasing)

(V_{SS} = 0 V, Topr = -40 to 85°C)

Parameter	Symbol	Pins	Ratings		Min	Max	Unit
Supply voltage (Condition 1)	V_{DD}		fc = 16 MHz	NORMAL1, 2 modes IDLE0, 1, 2 modes	4.5	5.5	V
			fc = 8 MHz	NORMAL1, 2 modes IDLE0, 1, 2 modes	3.0		
			fs = 32.768 KHz	SLOW1, 2 modes SLEEP0, 1, 2 modes			
			STOP mode				
Supply voltage (Condition 2) (Note1)			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_{IH1}	Except hysteresis input	$V_{DD} \geq 4.5$ V		$V_{DD} \times 0.70$	V_{DD}	V
	V_{IH2}	Hysteresis input			$V_{DD} \times 0.75$		
	V_{IH3}		$V_{DD} < 4.5$ V		$V_{DD} \times 0.90$		
Input low level	V_{IL1}	Except hysteresis input	$V_{DD} \geq 4.5$ V		0	$V_{DD} \times 0.30$	
	V_{IL2}	Hysteresis input				$V_{DD} \times 0.25$	
	V_{IL3}		$V_{DD} < 4.5$ V			$V_{DD} \times 0.10$	
Clock frequency	fc	XIN, XOUT	V_{DD} = 2.7 to 3.0V (Note1)		1.0	8.0	MHz
			V_{DD} = 3.0 to 5.5V				
			V_{DD} = 4.5 to 5.5V			16.0	
	fs	XTIN, XTOUT	V_{DD} = 2.7 to 3.0V (Note1)		30.0	34.0	kHz
			V_{DD} = 3.0 to 5.5V				

Note 1: When the supply voltage V_{DD} is less than 3.0 V, the operating temperature Topr must be in a range of -20°C to 85°C.

Not Recommended for New Design

17.2.3 MCU mode (Flash Programming or erasing)

(V_{SS} = 0 V, Topr = -10 to 40°C)

Parameter	Symbol	Pins	Ratings	Min	Max	Unit
Supply voltage	V_{DD}		NORMAL1, 2 modes	4.5	5.5	V
Input high level	V_{IH1}	Except hysteresis input	$V_{DD} \geq 4.5$ V	$V_{DD} \times 0.70$	V_{DD}	
	V_{IH2}	Hysteresis input		$V_{DD} \times 0.75$		
Input low level	V_{IL1}	Except hysteresis input	$V_{DD} \geq 4.5$ V	0	$V_{DD} \times 0.30$	
	V_{IL2}	Hysteresis input			$V_{DD} \times 0.25$	
Clock frequency	fc	XIN, XOUT		1.0	16.0	MHz

Not Recommended for New Design

17.3 DC Characteristics

(V_{SS} = 0 V, Topr = -40 to 85 °C)

Parameter	Symbol	Pins	Condition	Min	Typ.	Max	Unit
Hysteresis voltage	V_{HS}	Hysteresis input	V_{DD} = 5.5 V, V_{IN} = 5.5 V/0 V	-	0.9	-	V
Input current	I_{IN1}	TEST		-	-	±2	µA
	I_{IN2}	Sink open drain, tri - state port					
	I_{IN3}	$\overline{RESET}$, $\overline{STOP}$					
Input resistance	R_{IN2}	$\overline{RESET}$ pull - up	V_{DD} = 5.5 V, V_{IN} = 0 V	100	220	450	kΩ
Output leakage current	I_{LO1}	Sink open drain port	V_{DD} = 5.5 V, V_{OUT} = 5.5 V	-	-	2	µA
	I_{LO2}	Tri - state port	V_{DD} = 5.5 V, V_{OUT} = 5.5 V/0 V	-	-	±2	
Output high voltage	V_{OH}	Tri - state port	V_{DD} = 4.5 V, I_{OH} = -0.7 mA	4.1	-	-	V
Output low voltage	V_{OL}	Except XOUT, P0, P2, P4	V_{DD} = 4.5 V, I_{OL} = 1.6 mA	-	-	0.4	
Output low current	I_{OL}	High current port (P0, P2, P4 Port)	V_{DD} = 4.5 V, V_{OL} = 1.0 V	-	20	-	mA
Supply current in NORMAL1, 2 modes	I_{DD}		V_{DD} = 5.5 V, V_{IN} = 5.3 V/0.2 V, fc = 16 MHz, fs = 32.768 kHz; When a program operates on flash memory (Note5,6)	-	12.6	18	mA
			V_{DD} = 5.5 V, V_{IN} = 5.3 V/0.2 V, fc = 16 MHz, fs = 32.768 kHz; When a program operates on RAM (FLSSTB<FSTB>=0)	-	6.5	9	
Supply current in IDLE 0, 1, 2 modes			V_{DD} = 5.5 V, V_{IN} = 5.3 V/0.2 V, fc = 16 MHz, fs = 32.768 kHz	-	4.5	6.5	
Supply current in SLOW1 mode			V_{DD} = 3.0 V, V_{IN} = 2.8 V/0.2 V, fs = 32.768 kHz; When a program operates on flash memory (Note5,6)	-	20.0	50.0	µA
			V_{DD} = 3.0 V, V_{IN} = 2.8 V/0.2 V, fs = 32.768 kHz; When a program operates on RAM (FLSSTB<FSTB>=0)	-	14.0	28.0	
			V_{DD} = 3.0 V, V_{IN} = 2.8 V/0.2 V, fs = 32.768 kHz; When a program operates on RAM (FLSSTB<FSTB>=1)	-	9.0	-	
Supply current in SLEEP1 mode				-	5.0	15.0	
Supply current in SLEEP0 mode				-	4.0	13.0	
Supply current in STOP mode			V_{DD} = 5.5 V, V_{IN} = 5.3 V/0.2 V	-	0.5	10	
Peak current for SLOW1 mode (Note5,6)	I_{DDP-P}		V_{DD} = 5.5 V	-	10	-	mA
			V_{DD} = 3.0 V	-	2	-	

Note 1: Typical values show those at Topr = 25 °C and V_{DD} = 5 V.

Note 2: Input current I_{IN1}: The current through pull-down resistor is not included.

Note 3: I_{DD} does not include I_{REF}.

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

Note 5: When a program is executing in the flash memory or when data is being read from the flash memory, the flash memory operates in an intermittent manner, causing peak currents in the operation current, as shown in Figure 17-1.
In this case, the supply current I_{DD} (in NORMAL1, NORMAL2 and SLOW1 modes) is defined as the sum of the average peak current and MCU current.

Note 6: When designing the power supply, make sure that peak currents can be supplied. In SLOW1 mode, the difference between the peak current and the average current becomes large.

Not Recommended for New Design

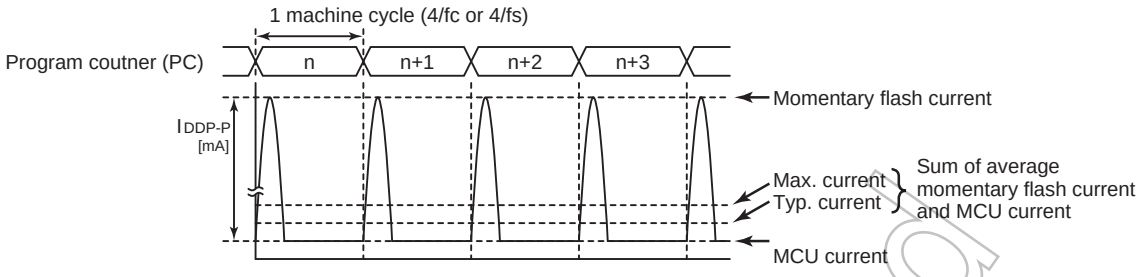


Figure 17-1 Intermittent Operation of Flash Memory

Not Recommended for New Design

17.4 AD Characteristics

(V_{SS} = 0.0 V, 4.5 V ≤ V_{DD} ≤ 5.5 V, Topr = −40 to 85 °C)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Analog reference voltage	V_{AREF}		A_{VDD} − 1.0	-	A_{VDD}	V
Power supply voltage of analog control circuit	A_{VDD}			V_{DD}		
	A_{VSS}			V_{SS}		
Analog reference voltage range (Note 4)	Δ V_{AREF}		3.5	-	-	
Analog input voltage	V_{AIN}		V_{SS}	-	V_{AREF}	
Power supply current of analog reference voltage	I_{REF}	V_{DD} = A_{VDD} = V_{AREF} = 5.5 V V_{SS} = A_{VSS} = 0.0 V	-	0.6	1.0	mA
Non linearity error		V_{DD} = A_{VDD} = V_{AREF} = 5.0 V V_{SS} = A_{VSS} = 0.0 V	-	-	±2	LSB
Zero point error			-	-	±2	
Full scale error			-	-	±2	
Total error			-	-	±2	

(V_{SS} = 0 V, 3.0 V ≤ V_{DD} < 4.5 V, Topr = −40 to 85°C)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Analog reference voltage	V_{AREF}		A_{VDD} − 1.0	-	A_{VDD}	V
Power supply voltage of analog control circuit	A_{VDD}			V_{DD}		
	A_{VSS}			V_{SS}		
Analog reference voltage range (Note 4)	Δ V_{AREF}		2.5	-	-	
Analog input voltage	V_{AIN}		V_{SS}	-	V_{AREF}	
Power supply current of analog reference voltage	I_{REF}	V_{DD} = A_{VDD} = V_{AREF} = 4.5 V V_{SS} = A_{VSS} = 0.0 V	-	0.5	0.8	mA
Non linearity error		V_{DD} = A_{VDD}= V_{AREF} = 3.0 V V_{SS} = A_{VSS} = 0.0 V	-	-	±2	LSB
Zero point error			-	-	±2	
Full scale error			-	-	±2	
Total error			-	-	±2	

(V_{SS} = 0 V, 2.7 V ≤ V_{DD} < 3.0 V, Topr = −20 to 85°C) (Note6)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Analog reference voltage	V_{AREF}		A_{VDD} − 0.5	-	A_{VDD}	V
Power supply voltage of analog control circuit	A_{VDD}			V_{DD}		
	A_{VSS}			V_{SS}		
Analog reference voltage range (Note 4)	Δ V_{AREF}		2.5	-	-	
Analog input voltage	V_{AIN}		V_{SS}	-	V_{AREF}	
Power supply current of analog reference voltage	I_{REF}	V_{DD} = A_{VDD} = V_{AREF} =2.7 V V_{SS} = A_{VSS} = 0.0 V	-	0.3	0.5	mA
Non linearity error		V_{DD} = A_{VDD} = V_{AREF} = 2.7 V V_{SS} = A_{VSS} = 0.0 V	-	-	±2	LSB
Zero point error			-	-	±2	
Full scale error			-	-	±2	
Total error			-	-	±2	

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

Note 2: Conversion time is different in recommended value by power supply voltage.

Note 3: The voltage to be input on the AIN input pin must not exceed the range between V_{AREF} and V_{SS}. If a voltage outside this range is input, conversion values will become unstable and conversion values of other channels will also be affected.

Note 4: Analog reference voltage range: $\Delta V_{AREF} = V_{AREF} - V_{SS}$

Note 5: When AD converter is not used, fix the A_{VDD} and V_{AREF} pin on the V_{DD} level.

Not Recommended for New Design

Note 6: When the supply voltage V_{DD} is less than 3.0 V, the operating temperature Topr must be in a range of −20°C to 85°C.

Not Recommended for New Design

17.5 AC Characteristics

(V_{SS} = 0 V, 4.5 V ≤ V_{DD} ≤ 5.5 V, Topr = −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_{WCH}	For external clock operation (XIN input) fc = 16 MHz	-	31.25	-	ns
Low-level clock pulse width	t_{WCL}					
High-level clock pulse width	t_{WSH}	For external clock operation (XTIN input) fs = 32.768 kHz	-	15.26	-	μs
Low-level clock pulse width	t_{WSL}					

(V_{SS} = 0 V, 3.0 V ≤ V_{DD} < 4.5 V, Topr = −40 to 85°C) (Note1)
(V_{SS} = 0 V, 2.7 V ≤ V_{DD} < 3.0 V, Topr = −20 to 85°C) (Note1)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Machine cycle time	t_{cy}	NORMAL1, 2 modes	0.5	-	4	μs
		IDLE0, 1, 2 modes				
		SLOW1, 2 modes	117.6	-	133.3	
		SLEEP0, 1, 2 modes				
High-level clock pulse width	t_{WCH}	For external clock operation (XIN input) fc = 8 MHz	-	62.5	-	ns
Low-level clock pulse width	t_{WCL}					
High-level clock pulse width	t_{WSH}	For external clock operation (XTIN input) fs = 32.768 kHz	-	15.26	-	μs
Low-level clock pulse width	t_{WSL}					

Note 1: When the supply voltage V_{DD} is less than 3.0 V, the operating temperature Topr must be in a range of −20°C to 85°C.

Not Recommended for New Design

17.6 Flash Characteristics

17.6.1 Write Characteristics

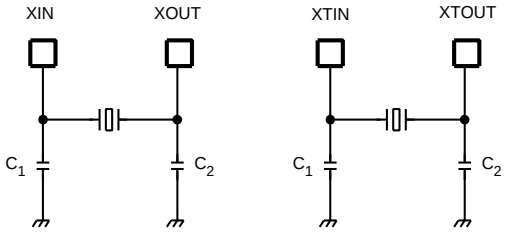
(V_{SS} = 0 V)

Parameter	Condition	Min	Typ.	Max.	Unit
Number of guaranteed writes to flash memory	V_{SS} = 0 V, Topr = -10 to 40°C	-	-	100	Times

Note: To rewrite data to Flashi memory addresses at whichi data is already written, make sure to erase the existing data before rewriting data.

Not Recommended for New Design

17.7 Oscillating Conditions



(1) High-frequency Oscillation (2) Low-frequency Oscillation

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 supplied by Murata Manufacturing Co., Ltd. are subject to change.

For up to date information, please refer to the following.

http://www.murata.com

17.8 Handling Precaution

- The solderability test conditions 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

  The pass criterion of the above test is as follows: Solderability rate until forming ≥ 95 %

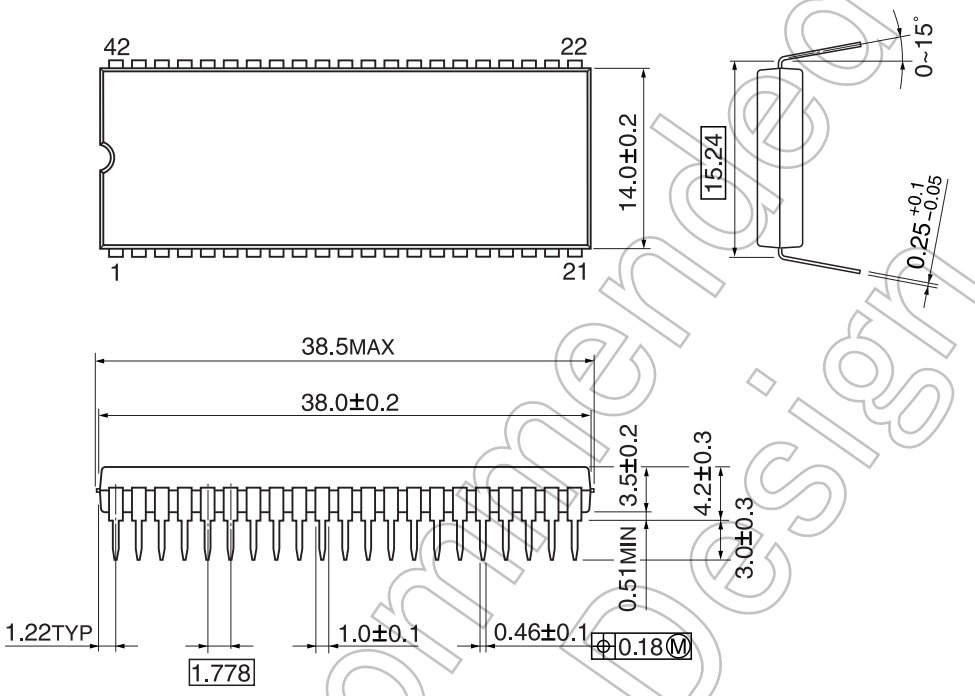
- When using the device (oscillator) in places exposed to high electric fields such as cathode-ray tubes, we recommend electrically shielding the package in order to maintain normal operating condition.

Not Recommended for New Design

18. Package Dimensions

SDIP42-P-600-1.78 Rev 01

Unit: mm



Not Recommended for New Design

Not Recommended for New Design

RESTRICTIONS ON PRODUCT USE

- Toshiba Corporation, and its subsidiaries and affiliates (collectively "TOSHIBA"), reserve the right to make changes to the information in this document, and related hardware, software and systems (collectively "Product") without notice.
- This document and any information herein may not be reproduced without prior written permission from TOSHIBA. Even with TOSHIBA's written permission, reproduction is permissible only if reproduction is without alteration/omission.
- Though TOSHIBA works continually to improve Product's quality and reliability, Product can malfunction or fail. Customers are responsible for complying with safety standards and for providing adequate designs and safeguards for their hardware, software and systems which minimize risk and avoid situations in which a malfunction or failure of Product could cause loss of human life, bodily injury or damage to property, including data loss or corruption. Before customers use the Product, create designs including the Product, or incorporate the Product into their own applications, customers must also refer to and comply with (a) the latest versions of all relevant TOSHIBA information, including without limitation, this document, the specifications, the data sheets and application notes for Product and the precautions and conditions set forth in the "TOSHIBA Semiconductor Reliability Handbook" and (b) the instructions for the application with which the Product will be used with or for. Customers are solely responsible for all aspects of their own product design or applications, including but not limited to (a) determining the appropriateness of the use of this Product in such design or applications; (b) evaluating and determining the applicability of any information contained in this document, or in charts, diagrams, programs, algorithms, sample application circuits, or any other referenced documents; and (c) validating all operating parameters for such designs and applications. **TOSHIBA ASSUMES NO LIABILITY FOR CUSTOMERS' PRODUCT DESIGN OR APPLICATIONS.**
- Product is intended for use in general electronics applications (e.g., computers, personal equipment, office equipment, measuring equipment, industrial robots and home electronics appliances) or for specific applications as expressly stated in this document. Product is neither intended nor warranted for use in equipment or systems that require extraordinarily high levels of quality and/or reliability and/or a malfunction or failure of which may cause loss of human life, bodily injury, serious property damage or serious public impact ("Unintended Use"). Unintended Use includes, without limitation, equipment used in nuclear facilities, equipment used in the aerospace industry, medical equipment, equipment used for automobiles, trains, ships and other transportation, traffic signaling equipment, equipment used to control combustions or explosions, safety devices, elevators and escalators, devices related to electric power, and equipment used in finance-related fields. Do not use Product for Unintended Use unless specifically permitted in this document.
- Do not disassemble, analyze, reverse-engineer, alter, modify, translate or copy Product, whether in whole or in part.
- Product shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable laws or regulations.
- The information contained herein is presented only as guidance for Product use. No responsibility is assumed by TOSHIBA for any infringement of patents or any other intellectual property rights of third parties that may result from the use of Product. No license to any intellectual property right is granted by this document, whether express or implied, by estoppel or otherwise.
- **ABSENT A WRITTEN SIGNED AGREEMENT, EXCEPT AS PROVIDED IN THE RELEVANT TERMS AND CONDITIONS OF SALE FOR PRODUCT, AND TO THE MAXIMUM EXTENT ALLOWABLE BY LAW, TOSHIBA (1) ASSUMES NO LIABILITY WHATSOEVER, INCLUDING WITHOUT LIMITATION, INDIRECT, CONSEQUENTIAL, SPECIAL, OR INCIDENTAL DAMAGES OR LOSS, INCLUDING WITHOUT LIMITATION, LOSS OF PROFITS, LOSS OF OPPORTUNITIES, BUSINESS INTERRUPTION AND LOSS OF DATA, AND (2) DISCLAIMS ANY AND ALL EXPRESS OR IMPLIED WARRANTIES AND CONDITIONS RELATED TO SALE, USE OF PRODUCT, OR INFORMATION, INCLUDING WARRANTIES OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, ACCURACY OF INFORMATION, OR NONINFRINGEMENT.**
- Do not use or otherwise make available Product or related software or technology for any military purposes, including without limitation, for the design, development, use, stockpiling or manufacturing of nuclear, chemical, or biological weapons or missile technology products (mass destruction weapons). Product and related software and technology may be controlled under the Japanese Foreign Exchange and Foreign Trade Law and the U.S. Export Administration Regulations. Export and re-export of Product or related software or technology are strictly prohibited except in compliance with all applicable export laws and regulations.
- Please contact your TOSHIBA sales representative for details as to environmental matters such as the RoHS compatibility of Product. Please use Product in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. TOSHIBA assumes no liability for damages or losses occurring as a result of noncompliance with applicable laws and regulations.

Not Recommended for New Design

Not Recommended for New Design