TOSHIBA

8 Bit Microcontroller

TLCS-870/X Series

TMP88CH40NG

TOSHIBA CORPORATION

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

TOSHIBA is continually working to improve the quality and reliability of its products. Nevertheless, semiconductor devices in general can malfunction or fail due to their inherent electrical sensitivity and vulnerability to physical stress.
It is the responsibility of the buyer, when utilizing TOSHIBA products, to comply with the standards of safety in making a safe design for the entire system, and to avoid situations in which a malfunction or failure of such TOSHIBA products could cause loss of human life, bodily injury or damage to property.
In developing your designs, please ensure that TOSHIBA products are used within specified operating ranges as set forth in the most recent TOSHIBA products specifications. Also, please keep in mind the precautions and conditions set forth in the "Handling Guide for Semiconductor Devices," or "TOSHIBA Semiconductor Reliability Handbook" etc. 021023_A

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

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

The information contained herein is presented only as a guide for the applications of our products. No responsibility is assumed by TOSHIBA for any infringements of patents or other rights of the third parties which may result from its use. No license is granted by implication or otherwise under any patents or other rights of TOSHIBA or the third parties. 070122_C

Please contact your sales representative for product-by-product details in this document regarding RoHS compatibility.
Please use these products in this document in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances.
Toshiba assumes no liability for damage or losses occurring as a result of noncompliance with applicable laws and regulations. 060819_Z

The products described in this document may include products subject to the foreign exchange and foreign trade control laws. 060925_F

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

© 2007 TOSHIBA CORPORATION
All Rights Reserved

Revision History

Date	Revision	
2007/7/10	1	First Release

Table of Contents

TMP88CH40NG

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

2. Functional Description

2.1 Functions of the CPU Core 7
2.1.1 Memory Address Map 7
2.1.2 Program Memory (ROM) 8
2.1.3 Data Memory (RAM) 8
2.1.4 System Clock Control Circuit 9
2.1.4.1 Clock Generator
2.1.4.2 Timing Generator
2.1.4.3 Standby Control Circuit
2.1.4.4 Controlling Operation Modes
2.1.5 Reset Circuit 17
2.1.5.1 External Reset Input
2.1.5.2 Adress Trap Reset
2.1.5.3 Watchdog Timer Reset
2.1.5.4 System Clock Reset

3. Interrupt Control Circuit

3.1 Interrupt latches (IL38 to IL2) 20
3.2 Interrupt enable register (EIR) 21
3.2.1 Interrupt master enable flag (IMF) 21
3.2.2 Individual interrupt enable flags (EF38 to EF3) 21
3.3 Interrupt Sequence 24
3.3.1 Interrupt acceptance processing is packaged as follows. 24
3.3.2 Saving/restoring general-purpose registers 25
3.3.2.1 Using Automatic register bank switcing
3.3.2.2 Using register bank switching
3.3.2.3 Using PUSH and POP instructions
3.3.2.4 Using data transfer instructions
3.3.3 Interrupt return 27
3.4 Software Interrupt (INTSW) 28
3.4.1 Address error detection 28
3.4.2 Debugging 28
3.5 External Interrupts 29

4. Special Function Register

4.1 SFR 31
4.2 DBR 33

5. Input/Output Ports

5.1 Port P1 (Only P10) 36
5.2 Port P3 (P37 to P30) 37
5.3 Port P4 (P45 to P40) 38
5.4 Port P6 (P63 to P60) 39

6. Watchdog Timer (WDT)

6.1 Watchdog Timer Configuration 41
6.2 Watchdog Timer Control 42
6.2.1 Malfunction Detection Methods Using the Watchdog Timer 42
6.2.2 Watchdog Timer Enable 43
6.2.3 Watchdog Timer Disable 44
6.2.4 Watchdog Timer Interrupt (INTWDT) 44
6.2.5 Watchdog Timer Reset 45

7. Time Base Timer (TBT)

7.1 Time Base Timer 47
Table 7-1 48

8. 16-Bit TimerCounter 1 (TC1)

8.1 Configuration 49
8.2 TimerCounter Control 49
8.3 Function 51
8.3.1 Timer mode 51
Figure 8-2 52
Figure 8-2 52
Figure 8-2 52
Figure 8-2 52
Figure 8-2 52

9. 8-Bit TimerCounter 3 (TC3)

9.1 Configuration 53
9.2 TimerCounter Control 54
9.3 Function 55
9.3.1 Timer mode 55
Figure 9-3 56
Figure 9-3 56

10. 8-Bit TimerCounter 4 (TC4)

10.1 Configuration 57
10.2 TimerCounter Control 58
10.3 Function 59

10.3.1 Timer Mode 59
Table 10-1 59
Table 10-1 59

11. Motor Control Circuit (PMD: Programmable motor driver)

11.1 Outline of Motor Control 62
11.2 Configuration of the Motor Control Circuit 64
11.3 Position Detection Unit 65
11.3.1 Configuration of the position detection unit 66
11.3.2 Position Detection Circuit Register Functions 67
11.3.3 Outline Processing in the Position Detection Unit 70
11.4 Timer Unit 71
11.4.1 Configuration of the Timer Unit 72
11.4.1.1 Timer Circuit Register Functions
11.4.1.2 Outline Processing in the Timer Unit
11.5 Three-phase PWM Output Unit 76
11.5.1 Configuration of the three-phase PWM output unit 76
11.5.1.1 Pulse width modulation circuit (PWM waveform generating unit)
11.5.1.2 Commutation control circuit
11.5.2 Register Functions of the Waveform Synthesis Circuit 80
11.5.3 Port output as set with UOC/VOC/WOC bits and UPWM/VPWM/WPWM bits 82
11.5.4 Protective Circuit 84
11.5.5 Functions of Protective Circuit Registers 86
11.6 Electrical Angle Timer and Waveform Arithmetic Circuit 88
11.6.1 Electrical Angle Timer and Waveform Arithmetic Circuit 89
11.6.1.1 Functions of the Electrical Angle Timer and Waveform Arithmetic Circuit Registers
11.6.1.2 List of PMD Related Control Registers

12. Asynchronous Serial interface (UART)

12.1 Configuration 101
12.2 Control 102
12.3 Transfer Data Format 104
12.4 Transfer Rate 105
12.5 Data Sampling Method 105
12.6 STOP Bit Length 106
12.7 Parity 106
12.8 Transmit/Receive Operation 106
12.8.1 Data Transmit Operation 106
12.8.2 Data Receive Operation 106
12.9 Status Flag 107
12.9.1 Parity Error 107
12.9.2 Framing Error 107
12.9.3 Overrun Error 107
12.9.4 Receive Data Buffer Full 108
12.9.5 Transmit Data Buffer Empty 108
12.9.6 Transmit End Flag 109

13. Synchronous Serial Interface (SIO)

13.1 Configuration 111
13.2 Control 112
13.3 Serial clock 113
13.3.1 Clock source 113

13.3.1.1 Internal clock
13.3.1.2 External clock
13.3.2 Shift edge 115
13.3.2.1 Leading edge
13.3.2.2 Trailing edge
13.4 Number of bits to transfer 115
13.5 Number of words to transfer 115
13.6 Transfer Mode 116
13.6.1 4-bit and 8-bit transfer modes 116
13.6.2 4-bit and 8-bit receive modes 118
13.6.3 8-bit transfer / receive mode 119

14. 10-bit AD Converter (ADC)

14.1 Configuration 121
14.2 Register configuration 122
14.3 Function 125
14.3.1 Software Start Mode 125
14.3.2 Repeat Mode 125
14.3.3 Register Setting 126
14.4 Analog Input Voltage and AD Conversion Result 128
14.5 Precautions about AD Converter 129
14.5.1 Analog input pin voltage range 129
14.5.2 Analog input shared pins 129
14.5.3 Noise Countermeasure 129

15. Input/Output Circuitry

15.1 Control pins 131
15.2 Input/output ports 132

16. Electrical Characteristics

16.1 Absolute Maximum Ratings 133
16.2 Operating Conditions 134
16.3 DC Characteristics 134
16.4 AD Conversion Characteristics 135
16.5 AC Characteristics 135
16.6 Recommended Oscillation Conditions 136
16.7 Handling Precaution 136

17. Package Dimensions

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

CMOS 8-Bit Microcontroller

TMP88CH40NG

Product No.	ROM (MaskROM)	RAM	Package	OTP MCU
TMP88CH40NG	16384 bytes	512+128 bytes	SDIP28-P-400-1.78	TMP88PH40

1.1 Features

1. 8-bit single chip microcomputer TLCS-870/X series
   - Instruction execution time :

     0.20 μs (at 20 MHz)
   - 181 types & 842 basic instructions
2. 19 interrupt sources (External : 1 Internal : 18)
3. Input / Output ports (19 pins)

   Large current output: 14pins (Typ. 20mA), LED direct drive

4. Watchdog Timer

   Select of "internal reset request" or "interrupt request".
5. Prescaler
   - Time base timer
6. 16-bit timer counter: 1 ch
   - Timer mode

7. 8-bit timer counter : 1 ch
   - Timer mode
8. 8-bit timer counter : 1 ch
   - Timer mode

9. Programmable motor driver (PMD) : 1 ch

- Sine wave drive circuit (built-in sine wave data-table RAM)

  Rotor position detect function

  Motor contro timer and capture function

  Overload protective function

  Auto commutation and auto position detection start function

10. 8-bit UART/SIO : 1 ch
11. 10-bit successive approximation type AD converter
    - Analog input: 4 ch
12. Clock oscillation circuit : 1 set
13. Low power consumption operation
    - IDLE mode: CPU stops.

      Only peripherals operate using high frequency clock. Release by interruputs (CPU restarts).
14. Operation voltage:

    4.5 V to 5.5 V at 20MHz

070122EBP

- The information contained herein is subject to change without notice. 021023_D
- TOSHIBA is continually working to improve the quality and reliability of its products. Nevertheless, semiconductor devices in general can malfunction or fail due to their inherent electrical sensitivity and vulnerability to physical stress. It is the responsibility of the buyer, when utilizing TOSHIBA products, to comply with the standards of safety in making a safe design for the entire system, and to avoid situations in which a malfunction or failure of such TOSHIBA products could cause loss of human life, bodily injury or damage to property.
  In developing your designs, please ensure that TOSHIBA products are used within specified operating ranges as set forth in the most recent TOSHIBA products specifications. Also, please keep in mind the precautions and conditions set forth in the "Handling Guide for Semiconductor Devices," or "TOSHIBA Semiconductor Reliability Handbook" etc. 021023_A
- The TOSHIBA products listed in this document are intended for usage in general electronics applications (computer, personal equipment, office equipment, measuring equipment, industrial robotics, domestic appliances, etc.). These TOSHIBA products are neither intended nor warranted for usage in equipment that requires extraordinarily high quality and/or reliability or a malfunction or failure of which may cause loss of human life or bodily injury ("Unintended Usage"). Unintended Usage include atomic energy control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, combustion control instruments, medical instruments, all types of safety devices, etc. Unintended Usage of TOSHIBA products listed in this document shall be made at the customer's own risk. 021023_B
- The products described in this document shall not be used or embedded to any downstream products of which manufacture, use and/or sale are prohibited under any applicable laws and regulations. 060106_Q
- The information contained herein is presented only as a guide for the applications of our products. No responsibility is assumed by TOSHIBA for any infringements of patents or other rights of the third parties which may result from its use. No license is granted by implication or otherwise under any patents or other rights of TOSHIBA or the third parties. 070122_C
- The products described in this document are subject to foreign exchange and foreign trade control laws. 060925_E
- For a discussion of how the reliability of microcontrollers can be predicted, please refer to Section 1.3 of the chapter entitled Quality and Reliability Assurance/Handling Precautions. 030619_S

1.2 Pin Assignment

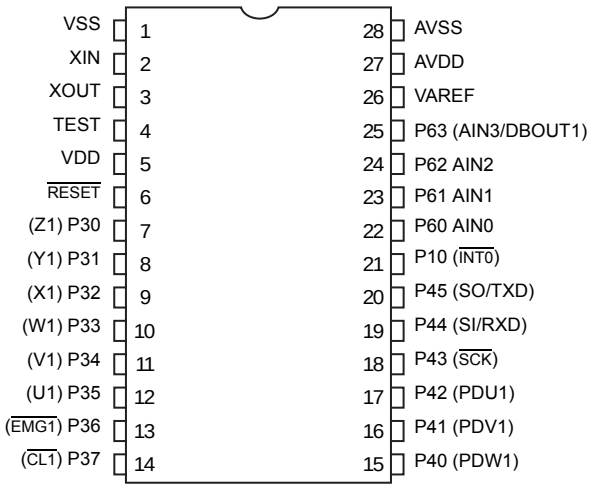


Figure 1-1 Pin Assignment

1.3 Block Diagram

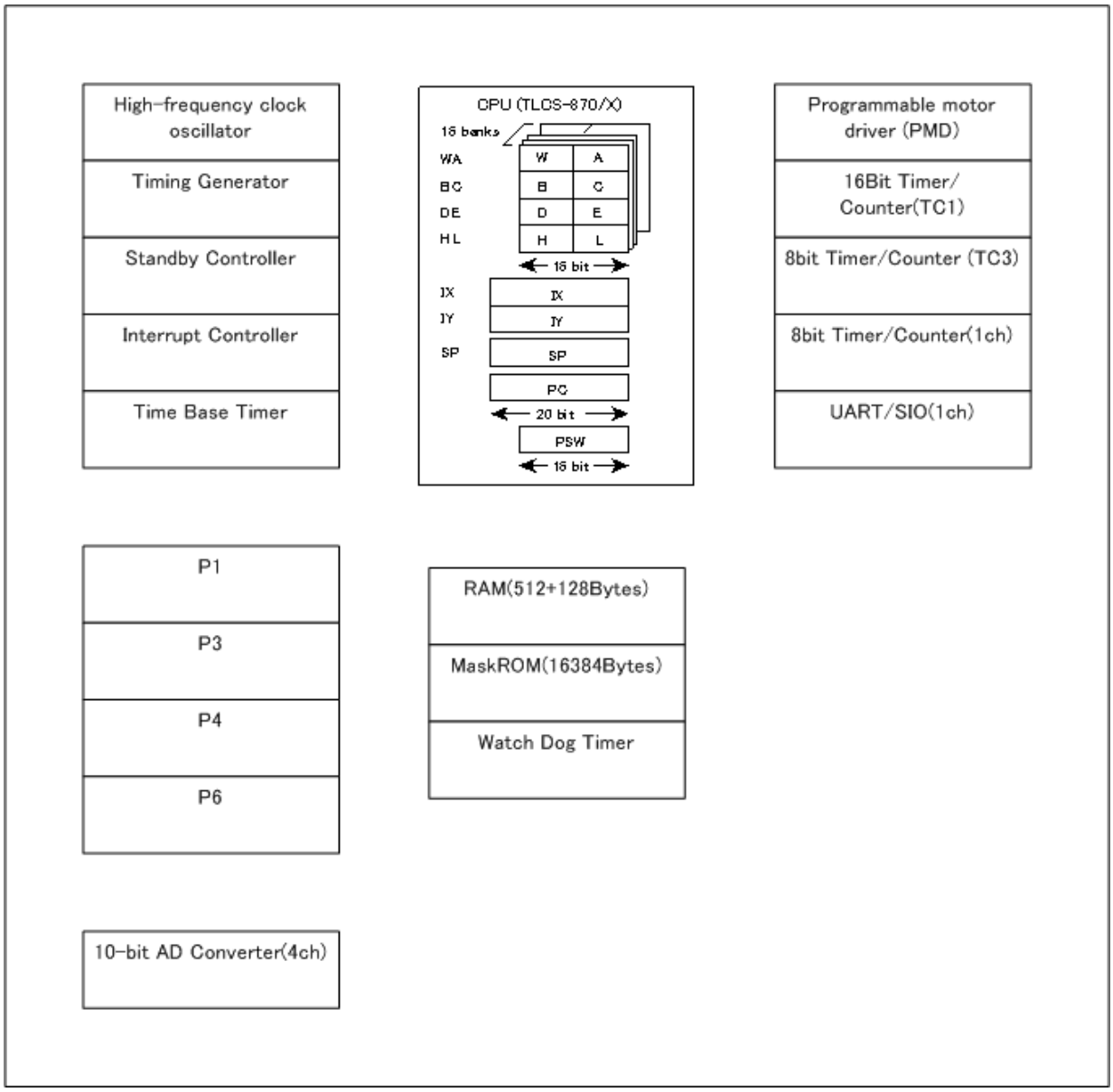


Figure 1-2 Block Diagram

1.4 Pin Names and Functions

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

Pin Name	Pin Number	Input/Output	Functions
P10 $\overline{\text{INT0}}$	21	IO I	PORT10 External interrupt 0 input
P37 $\overline{\text{CL1}}$	14	IO I	PORT37 PMD over load protection input1
P36 $\overline{\text{EMG1}}$	13	IO I	PORT36 PMD emergency stop input1
P35 U1	12	IO O	PORT35 PMD control output U1
P34 V1	11	IO O	PORT34 PMD control output V1
P33 W1	10	IO O	PORT33 PMD control output W1
P32 X1	9	IO O	PORT32 PMD control output X1
P31 Y1	8	IO O	PORT31 PMD control output Y1
P30 Z1	7	IO O	PORT30 PMD control output Z1
P45 SO TXD	20	IO O O	PORT45 Serial Data Output UART data output
P44 SI RXD	19	IO I I	PORT44 Serial Data Input UART data input
P43 $\overline{\text{SCK}}$	18	IO IO	PORT43 Serial Clock I/O
P42 PDU1	17	IO I	PORT42 PMD control input U1
P41 PDV1	16	IO I	PORT41 PMD control input V1
P40 PDW1	15	IO I	PORT40 PMD control input W1
P63 AIN3 DBOUT1	25	IO I O	PORT63 Analog Input3 PMD debug output1
P62 AIN2	24	IO I	PORT62 Analog Input2
P61 AIN1	23	IO I	PORT61 Analog Input1
P60 AIN0	22	IO I	PORT60 Analog Input0
XIN	2	I	Resonator connecting pins for high-frequency clock
XOUT	3	O	Resonator connecting pins for high-frequency clock
$\overline{\text{RESET}}$	6	I	Reset signal

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

Pin Name	Pin Number	Input/Output	Functions
TEST	4	I	Test pin for out-going test and the Serial PROM mode control pin. Usually fix to low level. Fix to high level when the Serial PROM mode starts.
VAREF	26	I	Analog Base Voltage Input Pin for A/D Conversion
AVDD	27	I	Analog Power Supply
AVSS	28	I	Analog Power Supply
VDD	5	I	+5V
VSS	1	I	0(GND)

2. Functional Description

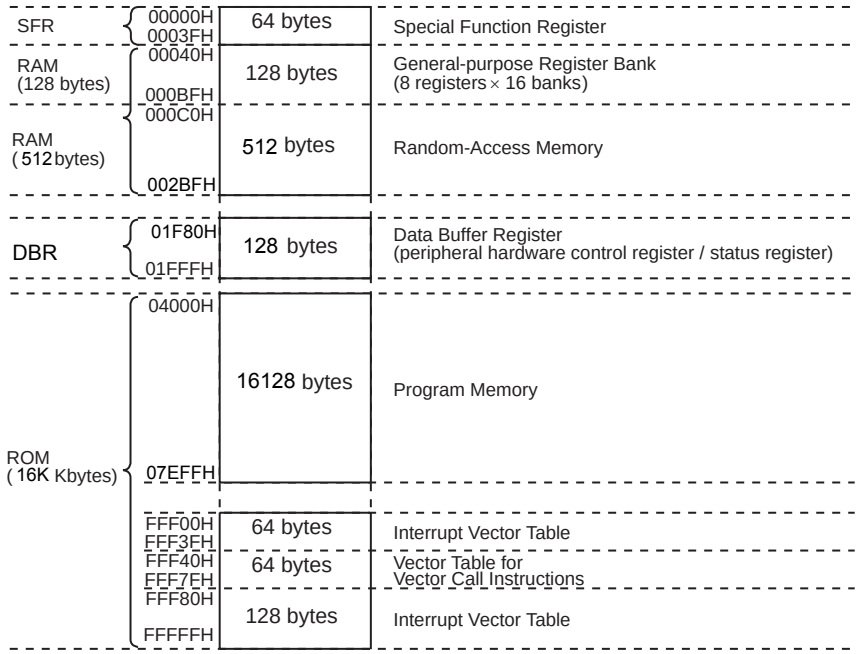
2.1 Functions of the CPU Core

The CPU core consists mainly of the CPU, system clock control circuit, and interrupt control circuit.

This chapter describes the CPU core, program memory, data memory, and reset circuit of the TMP88CH40NG.

2.1.1 Memory Address Map

The memory of the TMP88CH40NG consists of four blocks: ROM, RAM, SFR (Special Function Registers), and DBR (Data Buffer Registers), which are mapped into one 1-Mbyte address space. The general-purpose registers consist of 16 banks, which are mapped into the RAM address space. Figure 2-1 shows a memory address map of the TMP88CH40NG.



ROM: Read-Only Memory
- Program memory
- Vector Table

RAM: Random Access Memory
- Data memory
- Stack
- General-purpose register bank

SFR: Special Function Registers
- Input/output port
- Peripheral hardware control register
- Peripheral hardware status register
- System control register
- Interrupt control register
- Program status word

DBR: Data Buffer Registers
- Input/output port
- Peripheral hardware control register
- Peripheral hardware status register

Figure 2-1 Memory address map

2.1.2 Program Memory (ROM)

The TMP88CH40NG contains 16Kbytes program memory (MaskROM) located at addresses 04000H to 07EFFH and addresses FFF00H to FFFFFH.

2.1.3 Data Memory (RAM)

The TMP88CH40NG contains 512bytes +128bytes RAM. The first 128bytes location (00040H to 000BFH) of the internal RAM is shared with a general-purpose register bank.

The content of the data memory is indeterminate at power-on, so be sure to initialize it in the initialize routine.

Example :Clearing the internal RAM of the TMP88CH40NG (clear all RAM addresses to 0, except bank 0)

```
            LD      HL, 0048H       ; Set the start address
            LD      A, 00H          ; Set the initialization data (00H)
            LD      BC, 277H        ; Set byte counts (-1)
SRAMCLR:    LD      (HL+), A
            DEC     BC
            JRS     F, SRAMCLR
```

Note: Because general-purpose registers exist in the RAM, never clear the current bank address of RAM. In the above example, the RAM is cleared except bank 0.

2.1.4 System Clock Control Circuit

The System Clock Control Circuit consists of a clock generator, timing generator, and standby control circuit.

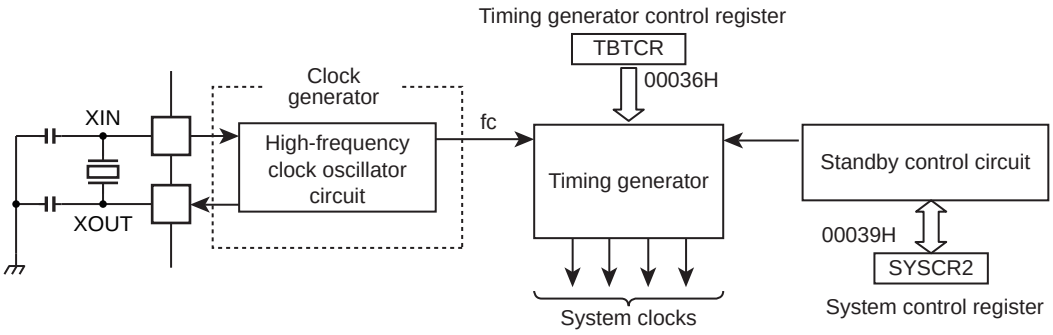


Figure 2-2 System Clock Control Circuit

2.1.4.1 Clock Generator

The Clock Generator generates the fundamental clock which serves as the reference for the system clocks supplied to the CPU core and peripheral hardware units.

The high-frequency clock (frequency fc) can be obtained easily by connecting a resonator to the XIN and XOUT pins. Or a clock generated by an external oscillator can also be used. In this case, enter the external clock from the XIN pin and leave the XOUT pin open. The TMP88CH40NG does not support the CR network that produces a time constant.

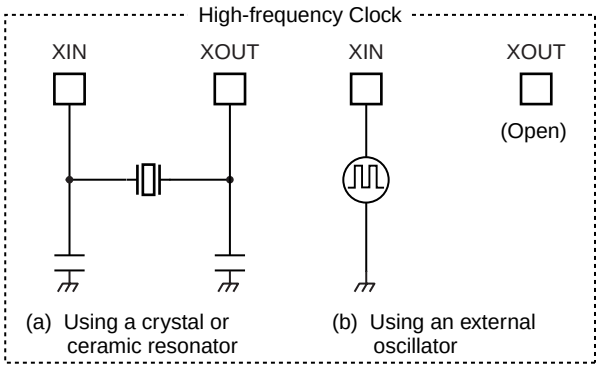


Figure 2-3 Example for Connecting a Resonator

Adjusting the oscillation frequency

Note: Although no hardware functions are provided that allow the fundamental clock to be monitored directly from the outside, the oscillation frequency can be adjusted by forwarding the pulse of a fixed frequency (e.g., clock output) to a port and monitoring it in a program while interrupts and the watchdog timer are disabled. For systems that require adjusting the oscillation frequency, an adjustment program must be created beforehand.

2.1.4.2 Timing Generator

The Timing Generator generates various system clocks from the fundamental clock that are supplied to the CPU core and peripheral hardware units. The Timing Generator has the following functions:

1. Generate the source clock for the time base timer
2. Generate the source clock for the watchdog timer
3. Generate the internal source clock for the timer counter

(1) Configuration of the Timing Generator

The Timing Generator a 3-stage prescaler, 21-stage dividers, and a machine cycle counter.

When reset, the prescaler and dividers are cleared to 0.

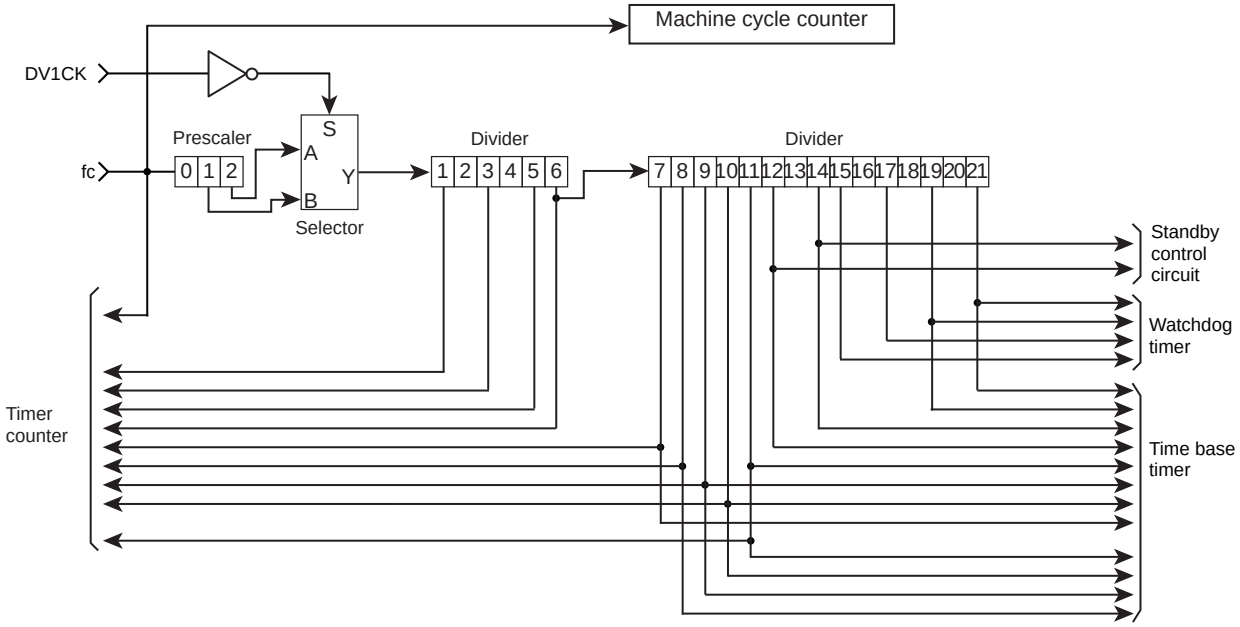


Figure 2-4 Configuration of the Timing Generator

Divider Control Register

CGCR (0030H)	7	6	5	4	3	2	1	0	
	0	0	DV1CK			0	0	0	(Initial value: 000* *000)

DV1CK	Selects input clock to the first divider stage	0: fc/4 1: fc/8	R/W

Note 1: fc: the high-frequency clock [Hz], *: Don't care
Note 2: The CGCR Register bits 4 and 3 show an indeterminate value when read.
Note 3: Be sure to write "0" to CGCR Register bits 7, 6, 2, 1 and 0.

(2) Machine cycle

Instruction execution and the internal hardware operations are synchronized to the system clocks.

The minimum unit of instruction execution is referred to as the "mgmachine cycle". The TLCS-870/X series has 15 types of instructions, from 1-cycle instructions which are executed in one machine cycle up to 15-cycle instructions that require a maximum of 15 machine cycles.

A machine cycle consists of four states (S0 to S3), with each state comprised of one main system clock cycle.

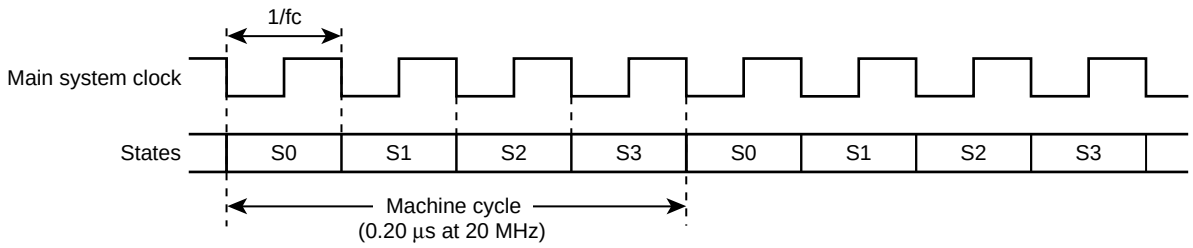


Figure 2-5 Machine Cycles

2.1.4.3 Standby Control Circuit

The Standby Control Circuit starts/stops the high-frequency clock oscillator circuit and selects the main system clock. The System Control Registers (SYSCR2) are used to control operation modes of this circuit. Figure 2-6 shows an operation mode transition diagram, followed by description of the System Control Registers.

(1) Single clock mode

Only the high-frequency clock oscillator circuit is used. Because the main system clock is generated from the high-frequency clock, the machine cycle time in single clock mode is 4/fc [s].

1. NORMAL mode

   In this mode, the CPU core and peripheral hardware units are operated with the high-frequency clock. The TMP88CH40NG enters this NORMAL mode after reset.

2. IDLE mode

   In this mode, the CPU and watchdog timer are turned off while the peripheral hardware units are operated with the high-frequency clock. IDLE mode is entered into by using System Control Register 2. The device is placed out of this mode and back into NORMAL mode by an interrupt from the peripheral hardware or an external interrupt. When IMF (interrupt master enable flag) = 1 (interrupt enabled), the device returns to normal operation after the interrupt has been serviced. When IMF = 0 (interrupt disabled), the device restarts execution beginning with the instruction next to one that placed it in IDLE mode.

Table 2-1 Single Clock Mode

Operation Mode		Oscillator Circuit		CPU Core	Peripheral Circuit	Machine Cycle Time
		High Frequency	Low Frequency			
Single Clock	RESET	Oscillate	-	Reset	Reset	4/fc [s]
	NORMAL			Operate	Operate	
	IDLE			Stop		

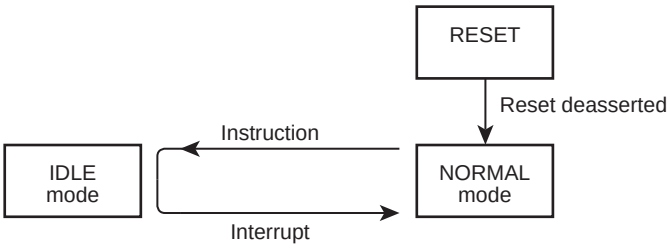


Figure 2-6 Operation Mode Transition Diagram

System Control Register 2

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

IDLE	Place the device in IDLE mode	0: Keep the CPU and WDT operating 1: Stop the CPU and WDT (IDLE mode entered)	R/W

Note 1: Be sure to set "1" to SYSCR2 Register bit7. When it is cleared to 0, the device is reset.
Note 2: WDT: Watchdog Timer, *: Don't care
Note 3: Be sure to write "0" to SYSCR2 Register bit6 and bit5.
Note 4: The values of the SYSCR2 Register bits 3 to 0 are indeterminate when read.

2.1.4.4 Controlling Operation Modes

(1) IDLE mode

IDLE mode is controlled by System Control Register 2 (SYSCR2) and a maskable interrupt. During IDLE mode, the device retains the following state.

1. The CPU and watchdog timer stop operating.

   The peripheral hardware continues operating.
2. The data memory, register, program status word, and port output latch hold the state in which they were immediately before entering IDLE mode.
3. The program counter holds the instruction address two instructions ahead the one that placed the device in IDLE mode.

Example :Placing the device in IDLE mode

```
SET        (SYSCR2) . 4
```

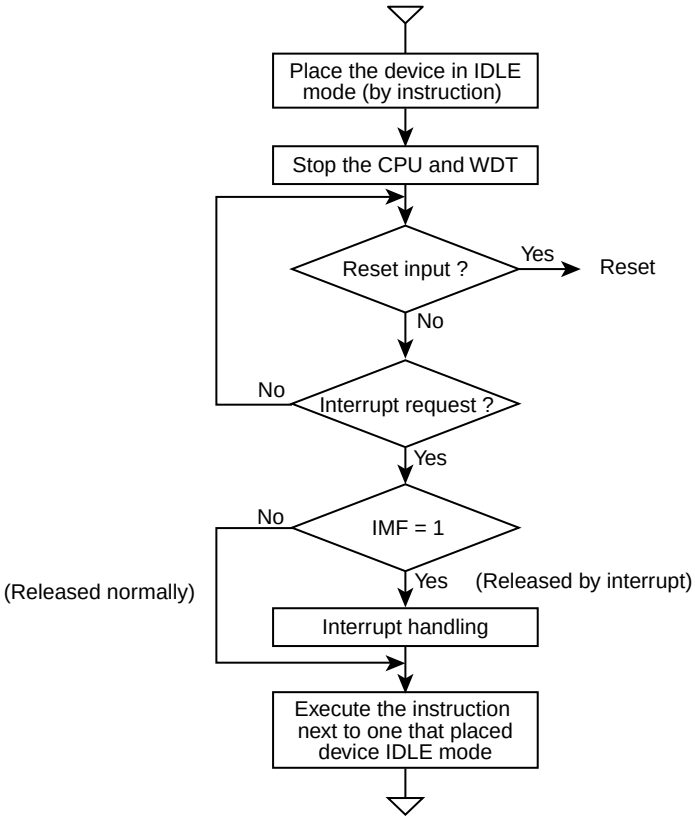


Figure 2-7 IDLE Mode

The device can be released from IDLE mode normally or by an interrupt as selected with the interrupt master enable flag (IMF).

a. Released normally (when IMF = 0)

The device can be released from IDLE mode by the interrupt source enabled by the interrupt individual enable flag (EF), and restarts execution beginning with the instruction next to one that placed it in IDLE mode. The interrupt latch (IL) for the interrupt source used to exit IDLE mode normally needs to be cleared to 0 using a load instruction.

b. Released by interrupt (when IMF = 1)

The device can be released from IDLE mode by the interrupt source enabled by the interrupt individual enable flag (EF), and enters interrupt handling. After interrupt handling, the device returns to the instruction next to one that placed it in IDLE mode.

The device can also be released from IDLE mode by pulling the $\overline{\text{RESET}}$ pin input low, in which case the device is immediately reset as is normally reset by $\overline{\text{RESET}}$. After reset, the device starts operating from NORMAL mode.

Note: If a watchdog timer interrupt occurs immediately before entering IDLE mode, the device processes the watchdog timer interrupt without entering IDLE mode.

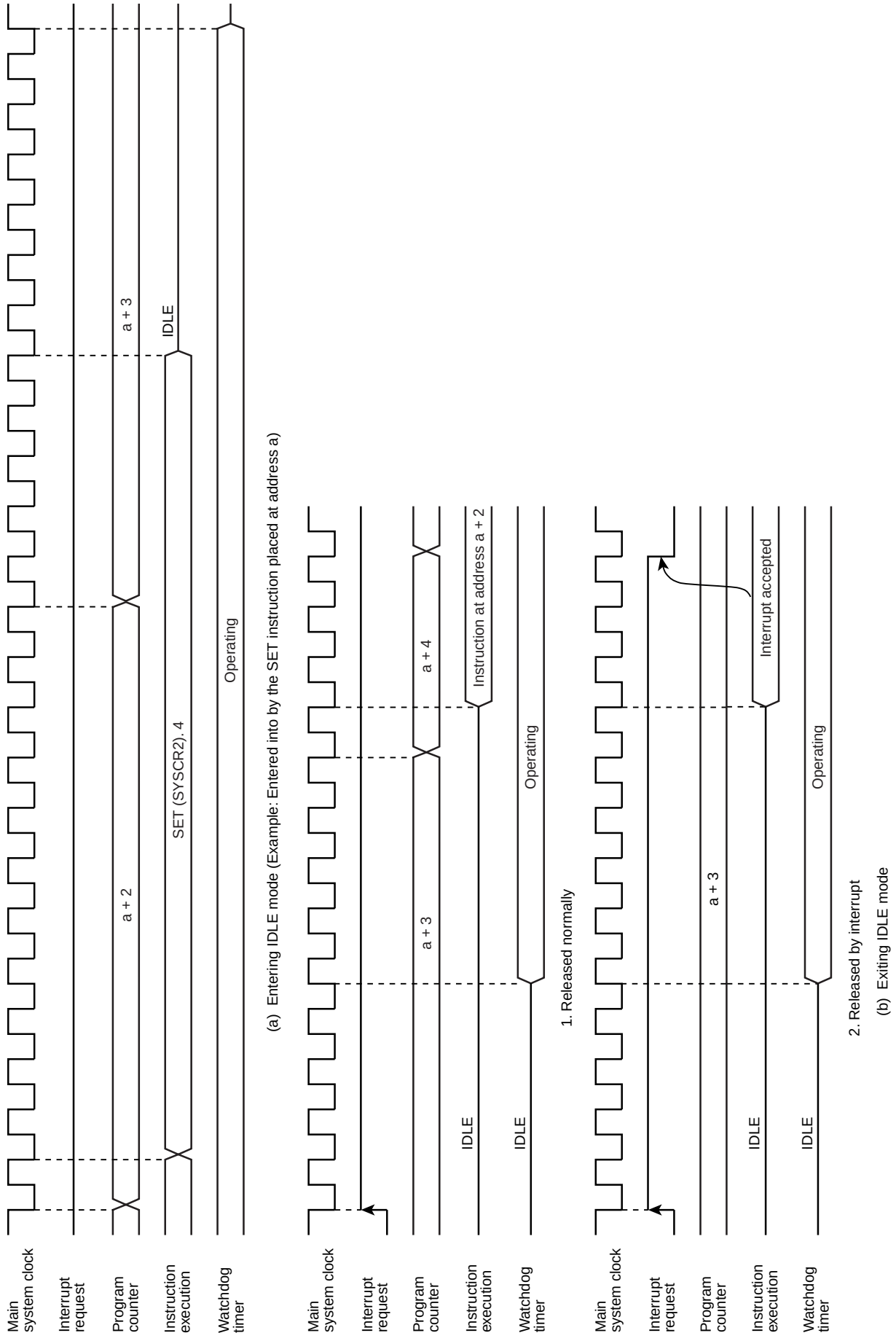


Figure 2-8 Entering and Exiting IDLE Mode

2.1.5 Reset Circuit

The TMP88CH40NG has four ways to generate a reset: external reset input, address trap reset, watchdog timer reset, or system clock reset.

Table 2-2 shows how the internal hardware is initialized by reset operation.

At power-on time, the internal cause reset circuits (watchdog timer reset, address trap reset, and system clock reset) are not initialized.

Table 2-2 Internal Hardware Initialization by Reset Operation

Internal Hardware	Initial Value	Internal Hardware	Initial Value
Program Counter (PC)	(FFFFEH to FFFFCH)	Prescaler and divider for the timing generator	0
Stack Pointer (SP)	Not initialized		
General-purpose Registers (W, A, B, C, D, E, H, L)	Not initialized		
Register Bank Selector (RBS)	0	Watchdog timer	Enable
Jump Status Flag (JF)	1		
Zero Flag (ZF)	Not initialized	Output latch of input/output port	See description of each input/output port.
Carry Flag (CF)	Not initialized		
Half Carry Flag (HF)	Not initialized		
Sign Flag (SF)	Not initialized		
Overflow Flag (VF)	Not initialized		
Interrupt Master Enable Flag (IMF)	0		
Interrupt Individual Enable Flag (EF)	0	Control register	See description of each control register.
Interrupt Latch (IL)	0		
Interrupt Nesting Flag (INF)	0	RAM	Not initialized

2.1.5.1 External Reset Input

The $\overline{\text{RESET}}$ pin is a hysteresis input with a pull-up resistor included. By holding the $\overline{\text{RESET}}$ pin low for at least three machine cycles (12/fc [s]) or more while the power supply voltage is within the rated operating voltage range and the oscillator is oscillating stably, the device is reset and its internal state is initialized.

When the $\overline{\text{RESET}}$ pin input is released back high, the device is freed from reset and starts executing the program beginning with the vector address stored at addresses FFFFCH to FFFFEH.

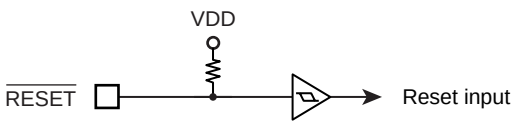


Figure 2-9 Reset Circuit

2.1.5.2 Adress Trap Reset

If the CPU should start looping for reasons of noise, etc. and attempts to fetch instructions from the internal RAM,SFR or DBR area, the device generats an internal reset.

The addess trap permission/prohibition is set by the address trap reset control register (ATAS,ATKEY). The address trap is permited initially and the internal reset is generated by fetching from internal RAM,SFR or DBR area. If the address trap is prohibited, instructions in the internal RAM area can be executed.

Address Trap Control Register

ATAS (1F94H)	7	6	5	4	3	2	1	0	
	-	-	-	-	-	-	-	ATAS	(initial value: **** ***0)

ATAS	Select the address trap permission / prohibition	0: Permit address trap 1: Prohibit address trap (It may be available after setting control code for ATKEY register)	Write only

Address Trap Control Code Register

ATKEY (1F95H)	7	6	5	4	3	2	1	0	
									(initial value: **** ****)

ATKEY	Write control code to prohibit address trap	D2H: Address trap prohibition code Others: Ineffective	Write only

Note: Read-modify-write instructions, such as a bit manipulation, cannot access ATAS or ATKEY register because these register are write only.

Note 1: In development tools, address trap cannot be prohibited in the internal RAM,SFR or DBR area with the address trap control registers. When using development tools, even if the address trap permission/prohibition setting is changed in the user's program, this change is ineffective. To execute instructions from the RAM area, development tools must be set accordingly.

Note 2: While the SWI instruction at an address immediately before the address trap area is executing, the program counter is incremented to point to the next address in the address trap area; an address trap is therefore taken immediately.

Development tool setting

- To prohibit the address trap:

  1. Modify the iram (mapping attribute) area to (00040H to 000BFH) in the memory map window.
  2. Set 000C0H to "address trap prohibition area" as a new eram (mapping attribute) area.
  3. Load the user program
  4. Execute the address trap prohibition code in the user's program

2.1.5.3 Watchdog Timer Reset

Refer to the Section "Watchdog Timer."

2.1.5.4 System Clock Reset

When SYSCR2 Register bit 7 is cleared to 0, the system clock is turned off, causing the CPU to become locked up. To prevent this problem, upon detecting "0" to SYSCR2 Register bit 7 or detecting "1" to SYSCR2 Register bit 5, the device automatically generates an internal reset signal to let the system clock continue oscillating.

3. Interrupt Control Circuit

The TMP88CH40NG has a total of 19 interrupt sources excluding reset. Interrupts can be nested with priorities. Two of the internal interrupt sources are pseudo nonmaskable 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)	Nonmaskable	–	FFFFC	High 0
Internal	INTSW (Software interrupt)	Pseudo nonmaskable	–	FFFF8	1
Internal	INTWDT (Watchdog timer interrupt)	Pseudo nonmaskable	IL2	FFFF4	2
External	INT0 (External interrupt 0)	IMF• EF3 = 1, INT0EN = 1	IL3	FFFF0	3
-	Reserved	IMF• EF4 = 1	IL4	FFFEC	4
-	Reserved	IMF• EF5 = 1	IL5	FFFE8	5
Internal	INTTBT (TBT interrupt)	IMF• EF6 = 1	IL6	FFFE4	6
-	Reserved	IMF• EF7 = 1	IL7	FFFE0	7
Internal	INTEMG1 (ch1 Error detect interrupt)	IMF• EF8 = 1	IL8	FFFDC	8
-	Reserved	IMF• EF9 = 1	IL9	FFFD8	9
Internal	INTCLM1 (ch1 Overload protection interrupt)	IMF• EF10 = 1	IL10	FFFD4	10
-	Reserved	IMF• EF11 = 1	IL11	FFFD0	11
Internal	INTTMR31 (ch1 Timer 3 interrupt)	IMF• EF12 = 1	IL12	FFFCC	12
-	Reserved	IMF• EF13 = 1	IL13	FFFC8	13
-	Reserved	IMF• EF14 = 1	IL14	FFFC4	14
-	Reserved	IMF• EF15 = 1	IL15	FFFC0	15
Internal	INTPDC1 (ch1 Posision detect interrupt)	IMF• EF16 = 1	IL16	FFFBC	16
-	Reserved	IMF• EF17 = 1	IL17	FFFB8	17
Internal	INTPWM1 (ch1 Waveform generater interrupt)	IMF• EF18 = 1	IL18	FFFB4	18
-	Reserved	IMF• EF19 = 1	IL19	FFFB0	19
Internal	INTEDT1 (ch1 Erectric angle Timer interrupt)	IMF• EF20 = 1	IL20	FFFAC	20
-	Reserved	IMF• EF21 = 1	IL21	FFFA8	21
Internal	INTTMR11 (ch1 Timer1 interrupt)	IMF• EF22 = 1	IL22	FFFA4	22
-	Reserved	IMF• EF23 = 1	IL23	FFFA0	23
Internal	INTTMR21 (ch1 Timer2 interrupt)	IMF• EF24 = 1	IL24	FFF9C	24
-	Reserved	IMF• EF25 = 1	IL25	FFF98	25
Internal	INTTC1 (TC1 interrupt)	IMF• EF26 = 1	IL26	FFF94	26
-	Reserved	IMF• EF27 = 1	IL27	FFF90	27
-	Reserved	IMF• EF28 = 1	IL28	FFF8C	28
-	Reserved	IMF• EF29 = 1	IL29	FFF88	29
-	Reserved	IMF• EF30 = 1	IL30	FFF84	30
-	Reserved	IMF• EF31 = 1	IL31	FFF80	31
Internal	INTRX (UART receive interrupt)	IMF• EF32 = 1	IL32	FFF3C	32
Internal	INTTX (UART transmit interrupt)	IMF• EF33 = 1	IL33	FFF38	33
Internal	INTSIO (SIO interrupt)	IMF• EF34 = 1	IL34	FFF34	34
Internal	INTTC3 (TC3 interrupt)	IMF• EF35= 1	IL35	FFF30	35
Internal	INTTC4 (TC4 interrupt)	IMF• EF36 = 1	IL36	FFF2C	36
-	Reserved	IMF• EF37 = 1	IL37	FFF28	37
Internal	INTADC (A/D converter interrupt)	IMF• EF38 = 1	IL38	FFF24	Low 38

Note 1: To use the watchdog timer interrupt (INTWDT), clear WDTCR1<WDTOUT> to "0" (It is set for the "Reset request" after reset is released). It is described in the section "Watchdog Timer" for details.

3.1 Interrupt latches (IL38 to IL2)

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

The interrupt latches are located on address 003CH, 003DH, 002EH, 002FH and 002BH 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 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.

Since interrupt latches can be read, the status for interrupt requests can be monitored by software. But interrupt latches are not set to “1” by an instruction.

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
LD      (ILL), 1110100000111111B    ; IL2 to IL7 ← 0
LD      (ILH), 1110100000111111B    ; IL8 to IL15 ← 0
LD      (ILE), 1110100000111111B    ; IL16 to IL23 ← 0
LD      (ILD), 1110100000111111B    ; IL24 to IL31 ← 0
LD      (ILC), 1110100000111111B    ; IL32 toIL38 ← 0
EI                                  ; IMF ← 1
```

Example 2 :Reads interrupt latches

```
LD      WA, (ILL)                   ; W ← (ILH), A ← (ILL)
LD      BC, (ILE)                   ; B ← (ILD), C ← (ILE)
LD      D, (ILC)                    ; D ← (ILC)
```

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 pseudo non-maskable interrupts (Software interrupt, undefined instruction interrupt, address trap interrupt and watchdog interrupt). Pseudo non-maskable interrupt is accepted regardless of the contents of the EIR.

The EIR consists of an interrupt master enable flag (IMF) and the individual interrupt enable flags (EF). These registers are located on address 003AH, 003BH, 002CH, 002DH and 002AH 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 temporarily. IMF flag is set to "1" by the maskable interrupt return instruction [RETI] after executing the interrupt service program routine, and MCU can accept the interrupt again. The latest interrupt request is generated already, it is available immediately after the [RETI] instruction is executed.

On the pseudo non-maskable interrupt, the non-maskable return instruction [RETN] is adopted. In this case, IMF flag is set to "1" only when it performs the pseudo non-maskable interrupt service routine on the interrupt acceptable status (IMF=1). However, IMF is set to "0" in the pseudo non-maskable interrupt service routine, it maintains its status (IMF="0").

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 (EF38 to EF3)

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 (EF38 to EF3) are initialized to “0” and all maskable interrupts are not accepted until they are set to “1”.

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

Example :Enables interrupts individually and sets IMF

```
DI                          ; IMF ← 0
SET     (EIRL), .5          ; EF5 ← 1
CLR     (EIRL), .6          ; EF6 ← 0
CLR     (EIRH), .4          ; EF12 ← 0
CLR     (EIRD), .0          ; EF24 ← 0
  :
EI                          ; IMF ← 1
```

Interrupt Latches

(Initial value: ***0*0*0 *0**0000)

ILH,ILL (003DH, 003CH)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	IL12	-	IL10	-	IL8	-	IL6	-	-	IL3	IL2	INF	
ILH (003DH)								ILL (003CH)							

(Initial value: *****0*0 *0*0*0*0)

ILD,ILE (002FH, 002EH)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	IL26	-	IL24	-	IL22	-	IL20	-	IL18	-	IL16
ILD (002FH)								ILE (002EH)							

(Initial value: *0*00000)

ILC (002BH)

7	6	5	4	3	2	1	0
-	IL38	-	IL36	IL35	IL34	IL33	IL32
ILE (002BH)							

		Read	Write	
IL38 to IL2	Interrupt latches	0: No interrupt request 1: Interrupt request	0: Clears the interrupt request (Note1) 1: (Unable to set interrupt latch)	R/W
INF	Interrupt Nesting Flag	00: Out of interrupt service 01: On interrupt service of level 1 01: On interrupt service of more than level 2 01: On interrupt service of more than level 3	00: Reserved 01: Clear the nesting counter 10: Count-down 1 step for the nesting counter (Note2) 11: Reserved	

Note 1: IL2 cannot alone be cleard.

Note 2: Unable to detect the under-flow of counter.

Note 3: The nesting counter is set "0" initially, it performs count-up by the interrupt acceptance and count-down by executing the interrupt return instruction.

Note 4: 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 5: Do not clear IL with read-modify-write instructions such as bit operations.

Interrupt Enable Registers

(Initial value: ***0*0*0 *0**0**0)

EIRH,EIRL (003BH, 003AH)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	EF12	-	EF10	-	EF8	-	EF6	-	-	EF3			IMF
EIRH (003BH)								EIRL (003AH)							

(Initial value: *****0*0 *0*0*0*0)

EIRD,EIRE (002DH, 002CH)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	EF26	-	EF24	-	EF22	-	EF20	-	EF18	-	EF16
EIRD (002DH)								EIRE (002CH)							

(Initial value: *0*00000)

EIRE (002AH)

7	6	5	4	3	2	1	0
-	EF38	-	EF36	EF35	EF34	EF33	EF32
EIRE (002AH)							

EF38 to EF3	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: Do not set IMF and the interrupt enable flag (EF38 to EF3) to “1” at the same time.

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".

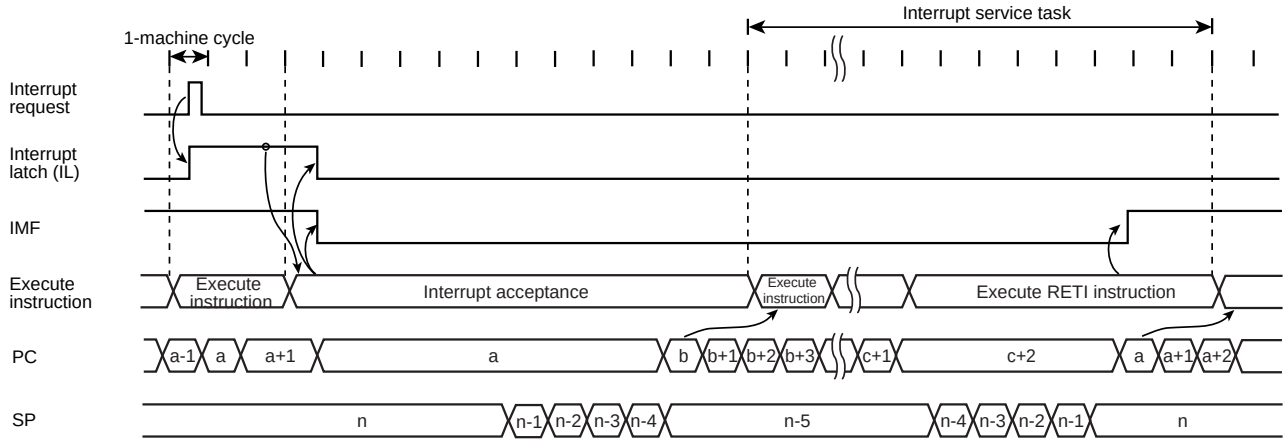
3.3 Interrupt Sequence

An interrupt request, which raised interrupt latch, is held, until interrupt is accepted or interrupt latch is cleared to "0" by resetting or an instruction. Interrupt acceptance sequence requires 12 machine cycles (2.4 μs @20 MHz) after the completion of the current instruction. The interrupt service task terminates upon execution of an interrupt return instruction [RETI] (for maskable interrupts) or [RETN] (for non-maskable interrupts). Figure 3-1 shows the timing chart of interrupt acceptance processing.

3.3.1 Interrupt acceptance processing is packaged as follows.

a. The interrupt master enable flag (IMF) is cleared to "0" in order to disable the acceptance of any following interrupt.
b. The interrupt latch (IL) for the interrupt source accepted is cleared to "0".
c. The contents of the program counter (PC) and the program status word, including the interrupt master enable flag (IMF), are saved (Pushed) on the stack in sequence of PSWH, PSWL, PCE, PCH, PCL. Meanwhile, the stack pointer (SP) is decremented by 5.
d. The entry address (Interrupt vector) of the corresponding interrupt service program, loaded on the vector table, is transferred to the program counter.
e. Read the RBS control code from the vector table, add its MSB(4bit) to the register bank selecter (RBS).
f. Count up the interrupt nesting counter.
g. The instruction stored at the entry address of the interrupt service program is executed.

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



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

Note 2: On condition that interrupt is enabled, it takes 62/fc [s] at maximum (If the interrupt latch is set at the first machine cycle on 15 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

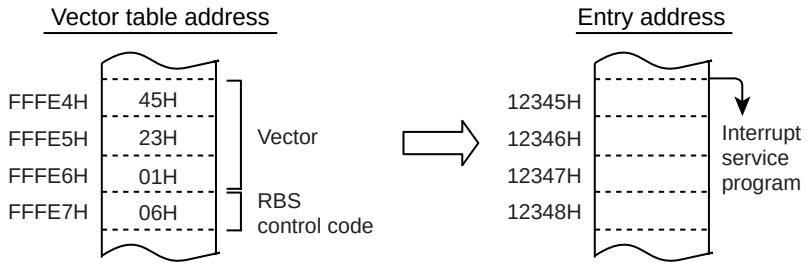


Figure 3-2 Vector table address,Entry address

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.
But don’t use the read-modify-write instruction for EIRL(0003AH) on the pseudo non-maskable interrupt service task.
To avoid overloaded nesting, clear the individual interrupt enable flag whose interrupt is currently serviced, before setting IMF to “1”. As for non-maskable interrupt, keep interrupt service shorten compared with length between interrupt requests; otherwise the status cannot be recovered as non-maskable interrupt would simply nested.

3.3.2 Saving/restoring general-purpose registers

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

3.3.2.1 Using Automatic register bank switcing

By switching to non-use register bank, it can restore the general-purpose register at hige speed.
Usually the bank register "0" is assigned for main task and the bank register "1 to 15" are for the each interrupt service task. To make up its data memory efficiency, the common bank is assigned for non-multiple intrrupt factor.

It can return back to main-flow by executing the interrupt return instructions ([RETI]/[RETN]) from the current interrupt register bank automatically. Thus, no need to restore the RBS by a program.

Example :Register bank switching

```
PINTxx:     (interrupt processing)          ; Begin of interrupt routine
            RETI                            ; End of interrupt
            :
VINTxx:     DP      PINTxx                  ; PINTxx vector address setting
            DB      1                       ; RBS <- RBS + 1      RBS setting on PINTxx
```

3.3.2.2 Using register bank switching

By switching to non-use register bank, it can restore the general-purpose register at hige speed.
Usually the bank register "0" is assigned for main task and the bank register "1 to 15" are for the each interrupt service task.

Example :Register bank switching

```
PINTxx:   LD      RBS, n        ; RBS <- n      Begin of interrupt routine
          (interrupt processing)
          RETI                  ; End of interrupt , restore RBS and interrupt return
          :
VINTxx:   DP      PINTxx        ; PINTxx vector address setting
          DB      0             ; RBS <- RBS + 0   RBS setting on PINTxx
```

3.3.2.3 Using PUSH and POP instructions

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

Example :Save/store register using PUSH and POP instructions

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

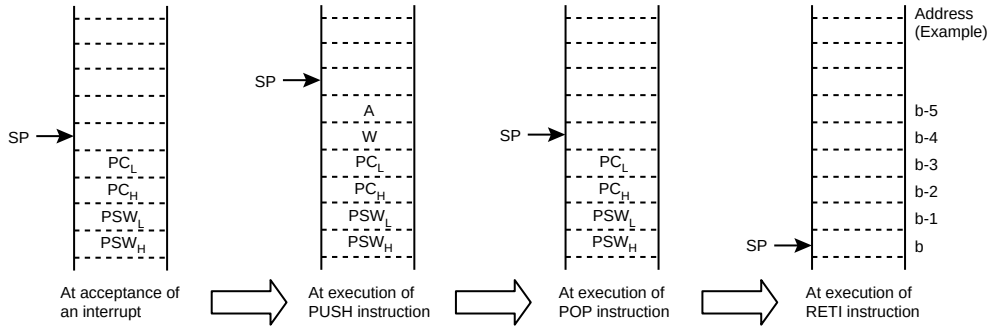


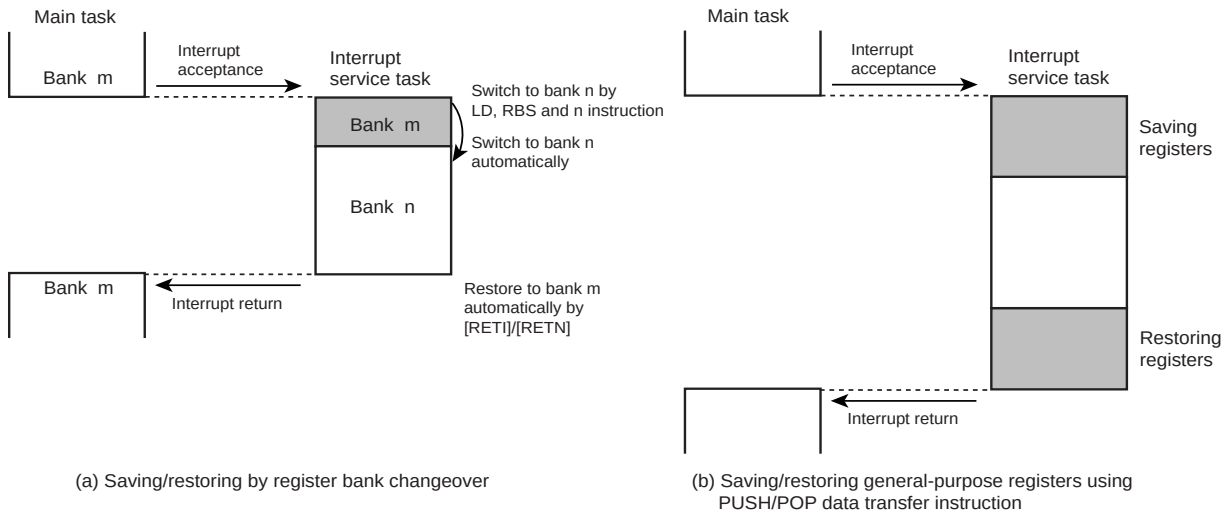
Figure 3-3 Save/store register using PUSH and POP instructions

3.3.2.4 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
```



(a) Saving/restoring by register bank changeover

(b) Saving/restoring general-purpose registers using PUSH/POP data transfer instruction

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

3.3.3 Interrupt return

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

[RETI] Maskable Interrupt Return	[RETN] Non-maskable Interrupt Return
1. The contents of the program counter and the program status word are restored from the stack. 2. The stack pointer is incremented 5 times. 3. The interrupt master enable flag is set to "1". 4. The interrupt nesting counter is decremented, and the interrupt nesting flag is changed.	1. The contents of the program counter and the program status word are restored from the stack. 2. The stack pointer is incremented 5 times. 3. The interrupt master enable flag is set to "1" only when a non-maskable interrupt is accepted in interrupt enable status. However, the interrupt master enable flag remains at "0" when so clear by an interrupt service program. 4. The interrupt nesting counter is decremented, and the interrupt nesting flag is changed.

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

3.4 Software Interrupt (INTSW)

Executing the SWI instruction generates a software interrupt and immediately starts interrupt processing (INTSW is highest prioritized interrupt). However, if processing of a non-maskable inerrupt is already underway, executing the SWI instruction will not generate a software interrupt but will result in the same operation as the NOP instruction.

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

3.4.1 Address error detection

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

3.4.2 Debugging

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

3.5 External Interrupts

The TMP88CH40NG has 1 external interrupt inputs. These inputs are equipped with digital noise reject circuits (Pulse inputs of less than a certain time are eliminated as noise).

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

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

Source	Pin	Sub-Pin	Enable Conditions	Release Edge (level)	Digital Noise Reject
INT0	$\overline{\text{INT0}}$	P10	IMF • EF3 • INT0EN=1	Falling edge	Pulses of less than 2/fc [s] are eliminated as noise. Pulses of 6/fc [s] or more are considered to be signals. (at CGCR<DV1CK>=0).

Note 1: When EINTCR<INT0EN> = "0", IL3 is not set even if a falling edge is detected on the $\overline{\text{INT0}}$ pin input.

Note 2: 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)		INT0EN							(Initial value: *0** ****)

INT0EN	P10/$\overline{\text{INT0}}$ pin configuration	0: P10 input/output port 1: $\overline{\text{INT0}}$ pin (Port P10 should be set to an input mode)	R/W

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

Note 2: 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).

4. Special Function Register

The TMP88CH40NG adopts the memory mapped I/O system, and all peripheral control and 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 1F80H to 1FFFH.

This chapter shows the arrangement of the special function register (SFR) and data buffer register (DBR) for TMP88CH40NG.

4.1 SFR

Address	Read	Write
0000H	Reserved	
0001H	P1DR	
0002H	Reserved	
0003H	P3DR	
0004H	P4DR	
0005H	Reserved	
0006H	P6DR	
0007H	Reserved	
0008H	Reserved	
0009H	Reserved	
000AH	Reserved	
000BH	P1CR	
000CH	Reserved	
000DH	Reserved	
000EH	Reserved	
000FH	TC1CR	
0010H	TC1DRAL	
0011H	TC1DRAH	
0012H	TC1DRBL	-
0013H	TC1DRBH	-
0014H	Reserved	
0015H	Reserved	
0016H	Reserved	
0017H	Reserved	
0018H	Reserved	
0019H	Reserved	
001AH	TC4CR	
001BH	TC4DR	
001CH	TC3DRA	
001DH	TC3DRB	-
001EH	TC3CR	
001FH	Reserved	
0020H	Reserved	
0021H	Reserved	
0022H	Reserved	
0023H	Reserved	
0024H	Reserved	
0025H	Reserved	

Address	Read	Write
0026H	ADCCRA	
0027H	ADCCRB	
0028H	ADCDRL	-
0029H	ADCDRH	-
002AH	EIRC	
002BH	ILC	
002CH	EIRE	
002DH	EIRD	
002EH	ILE	
002FH	ILD	
0030H	CGCR	
0031H	Reserved	
0032H	Reserved	
0033H	Reserved	
0034H	-	WDTCR1
0035H	-	WDTCR2
0036H	TBTCR	
0037H	EINTCR	
0038H	Reserved	
0039H	SYSCR2	
003AH	EIRL	
003BH	EIRH	
003CH	ILL	
003DH	ILH	
003EH	PSWL	
003FH	PSWH	

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

Note 2: – ; Cannot be accessed.

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

4.2 DBR

Address	PMD ch	Read	Write
1F80H		–	
1F81H		–	
1F82H		–	
1F83H		P3ODE	
1F84H		P4ODE	
1F85H		–	
1F86H		–	
1F87H		–	
1F88H		–	
1F89H		P3CR	
1F8AH		P4CR	
1F8BH		–	
1F8CH		P6CR	
1F8DH		–	
1F8EH		–	
1F8FH		–	
1F90H		–	
1F91H		UARTSR	UARTCRA
1F92H		–	UARTCRB
1F93H		RDBUF	TDBUF
1F94H		–	ATAS
1F95H		–	ATKEY
1F96H		–	SIOCR1
1F97H		SIOSR	SIOCR2
1F98H		SIOBR0	
1F99H		SIOBR1	
1F9AH		SIOBR2	
1F9BH		SIOBR3	
1F9CH		SIOBR4	
1F9DH		SIOBR5	
1F9EH		SIOBR6	
1F9FH		SIOBR7	
1FA0H	for PMD ch.1	PDCRA	
1FA1H	for PMD ch.1	PDCRB	
1FA2H	for PMD ch.1	PDCRC	–
1FA3H	for PMD ch.1	SDREG	
1FA4H	for PMD ch.1	MTCRA	
1FA5H	for PMD ch.1	MTCRB	
1FA6H	for PMD ch.1	MCAPL	–
1FA7H	for PMD ch.1	MCAPH	–
1FA8H	for PMD ch.1	CMP1L	
1FA9H	for PMD ch.1	CMP1H	
1FAAH	for PMD ch.1	CMP2L	
1FABH	for PMD ch.1	CMP2H	
1FACH	for PMD ch.1	CMP3L	
1FADH	for PMD ch.1	CMP3H	
1FAEH	for PMD ch.1	MDCRA	
1FAFH	for PMD ch.1	MDCRB	

Address	PMD ch	Read	Write
1FB0H	for PMD ch.1	EMGCRA	
1FB1H	for PMD ch.1	EMGCRB	
1FB2H	for PMD ch.1	MDOUTL	
1FB3H	for PMD ch.1	MDOUTH	
1FB4H	for PMD ch.1	MDCNTL	–
1FB5H	for PMD ch.1	MDCNTH	–
1FB6H	for PMD ch.1	MDPRDL	
1FB7H	for PMD ch.1	MDPRDH	
1FB8H	for PMD ch.1	CMPUL	
1FB9H	for PMD ch.1	CMPUH	
1FBAH	for PMD ch.1	CMPVL	
1FBBH	for PMD ch.1	CMPVH	
1FBCH	for PMD ch.1	CMPWL	
1FBDH	for PMD ch.1	CMPWH	
1FBEH	for PMD ch.1	DTR	
1FBFH	for PMD ch.1	–	EMGREL
1FC0H	for PMD ch.1	EDCRA	
1FC1H	for PMD ch.1	EDCRB	
1FC2H	for PMD ch.1	EDSETL	
1FC3H	for PMD ch.1	EDSETH	
1FC4H	for PMD ch.1	ELDEGL	
1FC5H	for PMD ch.1	ELDEGH	
1FC6H	for PMD ch.1	AMPL	
1FC7H	for PMD ch.1	AMPH	
1FC8H	for PMD ch.1	EDCAPL	–
1FC9H	for PMD ch.1	EDCAPH	–
1FCAH	for PMD ch.1	–	WFMDR
1FCBH		–	
1FCCH		Reserved	
to		:	
1FFFH		Reserved	

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

Note 2: – ; Cannot be accessed.

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

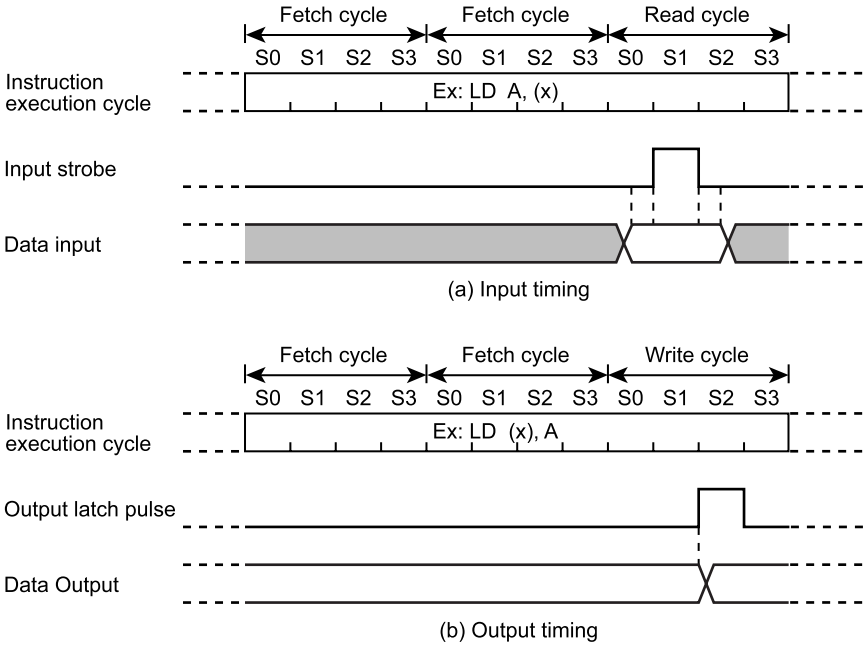
5. Input/Output Ports

The TMP88CH40NG contains 4 input/output ports comprised of 19 pins.

	Primary Function	Secondary Functions
Port P1	1-bit I/O port	External interrupt input
Port P3	8-bit I/O port	Motor control input/output
Port P4	6-bit I/O port	Serial interface input/output, motor control circuit input
Port P6	4-bit I/O port	Analog input and motor control circuit output

All output ports contain a latch, and the output data therefore are retained by the latch. But none of the input ports have a latch, so it is desirable that the input data be retained externally until it is read out, or read several times before being processed. Figure 5-1 shows input/output timing.

The timing at which external data is read in from input/output ports is S1 state in the read cycle of instruction execution. Because this timing cannot be recognized from the outside, transient input data such as chattering needs to be dealt with in a program. The timing at which data is forwarded to input/output ports is S2 state in the write cycle of instruction execution.



Note: The read/write cycle positions vary depending on instructions.

Figure 5-1 Example of Input/Output Timing

When an operation is performed for read from any input/output port except programmable input/output ports, whether the input value of the pin or the content of the output latch is read depends on the instruction executed, as shown below.

1. Instructions which read the content of the output latch
   - XCH r, (src)
   - SET/CLR/CPL (src).b
   - SET/CLR/CPL (pp).g
   - LD (src).b, CF
   - LD (pp).b, CF
   - XCH CF, (src). b
   - ADD/ADDC/SUB/SUBB/AND/OR/XOR (src), n
   - ADD/ADDC/SUB/SUBB/AND/OR/XOR (src), (HL) instructions, the (src) side thereof
   - MXOR (src), m
2. Instructions which read the input value of the pin

   Any instructions other than those listed above and ADD/ADDC/SUB/SUBB/AND/OR/XOR (src),(HL) instructions, the (HL) side thereof.

5.1 Port P1 (Only P10)

Port P1 is an 8-bit input/output port shared with external interrupt input. This port is switched between input and output modes using the P1 port input/output control register (P1CR). When reset, the P1CR register is initialized to 0, with the P1 port set for input mode. Also, the output latch (P1DR) is initialized to 0 when reset.

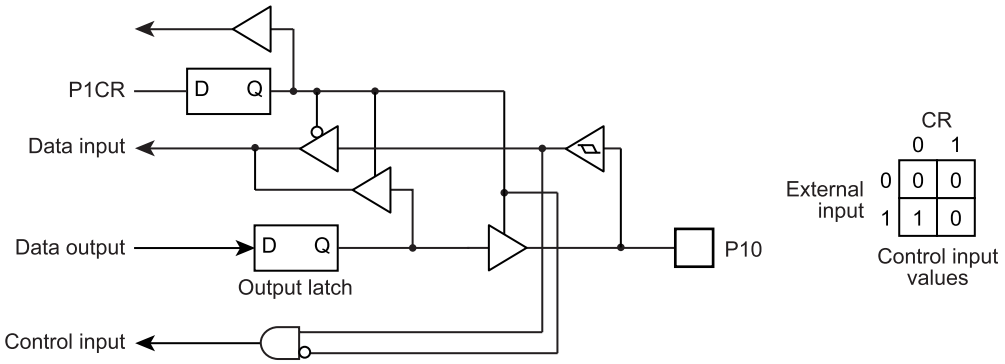


Figure 5-2 Port P1

P1 port input/output register

	7	6	5	4	3	2	1	0	
P1DR (00001H) R/W								P10 / $\overline{INT0}$	(Initial value: **** ***0)
P1CR (0000BH)	7	6	5	4	3	2	1	0	
									(Initial value: **** ***0)

P1CR	P1 port input/output control (Specify bitwise)	0: Input mode 1: Output mode	R/W

5.2 Port P3 (P37 to P30)

Port P3 is an 8-bit input/output port. This port is switched between input and output modes using the P3 port Input/output Control Register (P3CR). When reset, the P3CR Register is initialized to 0, with the P3 port set for input mode. Also, the Output Latch (P3DR) is initialized to 0 when reset.

The P3 port contains bitwise programmable open-drain control. The P3 Port Open-drain Control Register (P3ODE) is used to select open-drain or tri-state mode for the port. When reset, the P3ODE Register is initialized to 0, with tri-state mode selected for the port.

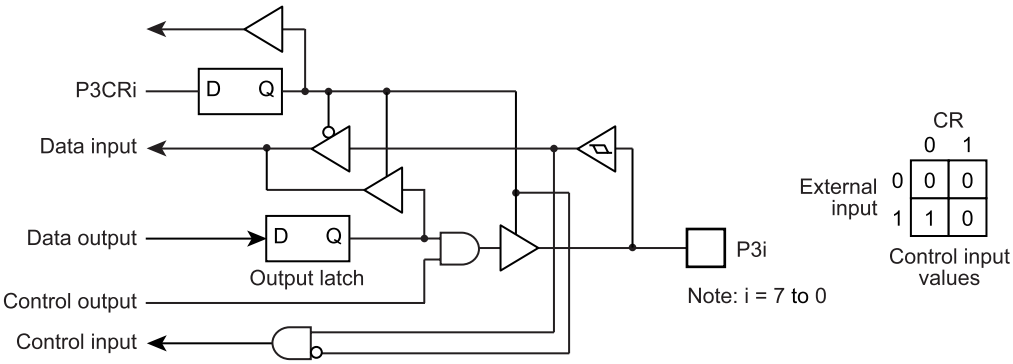


Figure 5-3 Port P3

P3 port input/output registers

	7	6	5	4	3	2	1	0	
P3DR (00003H) R/W	P37 $\overline{\text{CL1}}$	P36 $\overline{\text{EMG1}}$	P35 U1	P34 V1	P33 W1	P32 X1	P31 Y1	P30 Z1	(Initial value: 0000 0000)
P3CR (01F89H)									(Initial value: 0000 0000)

P3CR	P3 port input/output control (Specify bitwise)	0: Input mode 1: Output mode	R/W

	7	6	5	4	3	2	1	0	
P3ODE (01F83H)									(Initial value: 0000 0000)

P3ODE	P3 port open-drain control (Specify bitwise)	0: Tri-state 1: Open-drain	R/W

Note 1: Even when open-drain mode is selected, the protective diode remains connected. Therefore, do not apply voltages exceeding V_{DD}.

Note 2: If read-modify-write instruction is executed while the register is selecting open-drain mode, output latch data are read out. At the other instruction is executed, external pin states are read out.

Note 3: For PMD circuit output, set the P3DR output latch to 1.

Note 4: When using P3 port as an input/output port, disable the EMG1 circuit.

5.3 Port P4 (P45 to P40)

Port P4 is an 6-bit input/output port shared with serial interface input/output. This port is switched between input and output modes using the P4 port input/output control register (P4CR). When reset, the P4CR register is initialized to 0, with the P4 port set for input mode. Also, the output latch (P4DR) is initialized to 0 when reset.

The P4 port contains bitwise programmable open-drain control. The P4 port open-drain control register (P4ODE) is used to select open-drain or tri-state mode for the port. When reset, the P4ODE register is initialized to 0, with tri-state mode selected for the port.

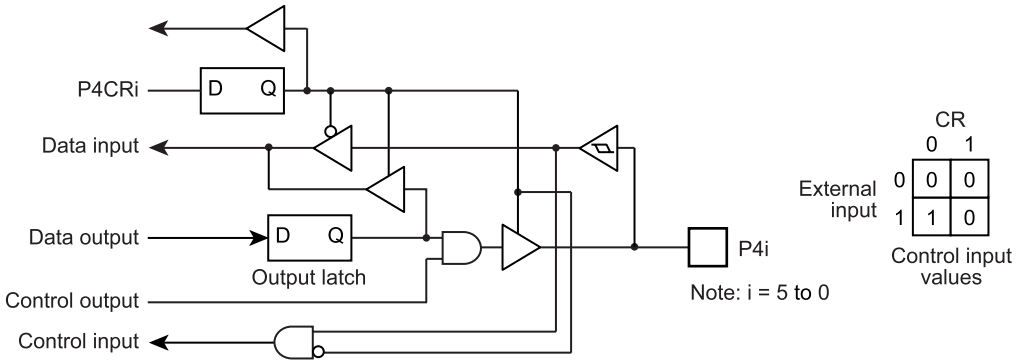


Figure 5-4 Port P4

P4 port input/output registers

	7	6	5	4	3	2	1	0	
P4DR (00004H) R/W			P45 SO TXD1	P44 SI RXD1	P43 $\overline{SCK}$	P42 PDU1	P41 PDV1	P40 PDW1	(Initial value: **00 0000)
P4CR (01F8AH)	7	6	5	4	3	2	1	0	
									(Initial value: **00 0000)

P4CR	P4 port input/output control (Specify bitwise)	0: Input mode 1: Output mode	R/W

	7	6	5	4	3	2	1	0	
P4ODE (01F84H)									(Initial value: **00 0000)

P4ODE	P4 port open-drain control (Specify bitwise)	0: Tri-state 1: Open-drain	R/W

Note 1: Even when open-drain mode is selected, the protective diode remains connected. Therefore, do not apply voltages exceeding V_{DD}.

Note 2: If read-modify-write instruction is executed while the register is selecting open-drain mode, output latch data are read out. At the other instruction is executed, external pin states are read out.

Note 3: *: Don't care

5.4 Port P6 (P63 to P60)

Port P6 is an 4-bit input/output port shared with AD converter analog input. This port is switched between input and output modes using the P6 port input/output control register (P6CR), P6 port output latch (P6DR), and ADCCRA<AINDS>. When reset, the P6CR Register and the P6DR output latch are initialized to 0 while ADCCRA<AINDS> is set to 1, so that P63 to P60 have their inputs fixed low (= 0). When using the P6 port as an input port, set the corresponding bits for input mode (P6CR = 0, P6DR = 1). The reason why the output latch = 1 is because it is necessary to prevent current from flowing into the shared data input circuit. When using the port as an output port, set the P6CR Register's corresponding bits to 1. When using the port for analog input, set the corresponding bits for analog input (P6CR = 0, P6DR = 0). Then set ADCCRA<AINDS> = 0, and AD conversion will start.

The ports used for analog input must have their output latches set to 0 beforehand. The actual input channels for AD conversion are selected using ADCCRA<SAIN>.

Although the bits of P6 port not used for analog input can be used as input/output ports, do not execute output instructions on these ports during AD conversion. This is necessary to maintain the accuracy of AD conversion. Also, do not apply rapidly changing signals to ports adjacent to analog input during AD conversion.

If an input instruction is executed while the P6DR output latch is cleared to 0, data “0” is read in from said bits.

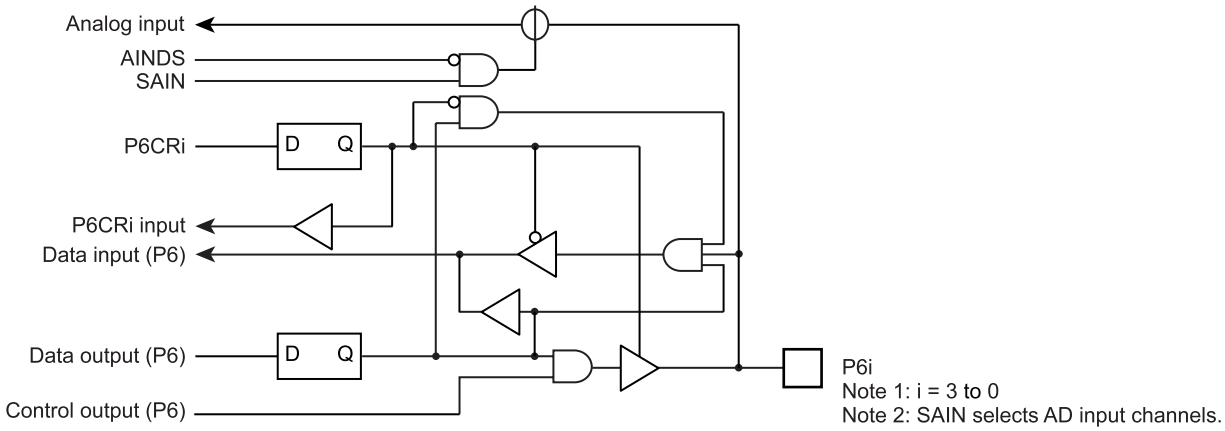


Figure 5-5 Port P6

P6 port input/output registers

	7	6	5	4	3	2	1	0	
P6DR (00006H) R/W					P63 AIN3 DBOUT	P62 AIN2	P61 AIN1	P60 AIN0	(Initial value: **** 0000)
P6CR (01F8CH)	7	6	5	4	3	2	1	0	
									(Initial value: **** 0000)

			AINDS = 1 (when not using AD)		AINDS = 0 (when using AD)		
			P6DR = “0”	P6DR = “1”	P6DR = “0”	P6DR = “1”	
P6CR	P6 port input/output control (Specify bitwise)	0	Inputs fixed to 0	Input mode	Analog input mode (Note2)	Input mode	R/W
		1	Output mode		Output mode		

Note 1: The pins used for analog input cannot be set for output mode (P6CR = 1) because they become shorted with external signals.

Note 2: When a read instruction is executed on bits of this port which are set for analog input mode, data "0" is read in.

Note 3: For DBOUT output, set the P6DR (P63) output latch to 1.

Note 4: *: Don’t care

Note 5: When using this port in input mode (including analog input), do not use bit manipulating or other read-modify-write instructions. When a read instruction is executed on the bits of this port that are set for input, the contents of the pins are read in, so that if a read-modify-write instruction is executed, their output latches may be rewritten, making the pins unable to

accept input. (A read-modify-write instruction first reads data from all of the eight bits and after modifying them (bit manipulation), writes data for all of the eight bits to the output latches.)

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 “pseudo nonmaskable interrupt request”. Upon the reset release, this signal is initialized to “reset request”.

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

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

6.1 Watchdog Timer Configuration

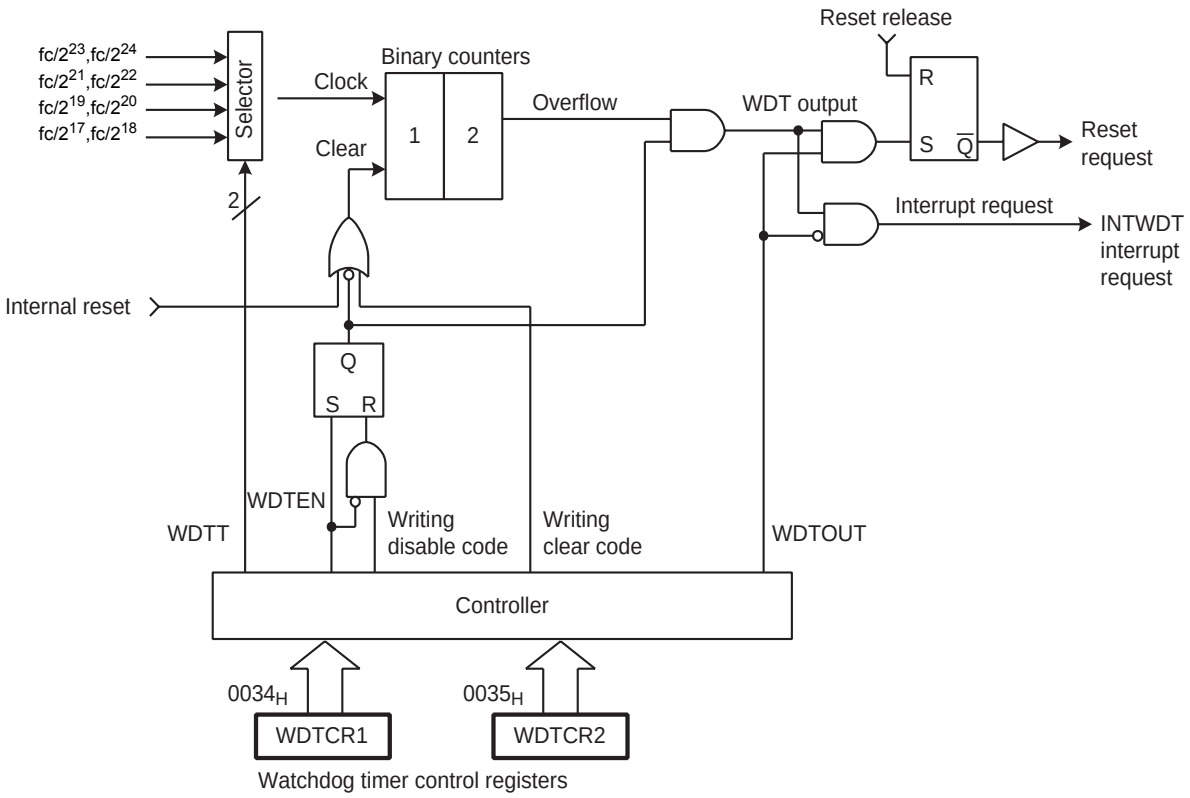


Figure 6-1 Watchdog Timer Configuration

6.2 Watchdog Timer Control

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

6.2.1 Malfunction Detection Methods Using the Watchdog Timer

The CPU malfunction is detected, as shown below.

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

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

The watchdog timer temporarily stops counting in the IDLE mode, and automatically restarts (continues counting) when the IDLE 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.
```

Watchdog Timer Control Register 1

WDTCR1 (0034H)	7	6	5	4	3	2	1	0	
					WDTEN	WDTT		WDTOUT	(Initial value: **** 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]		NORMAL mode		Write only
			DV1CK = 0	DV1CK = 1	
		00	2^{25}/fc	2^{26}/fc	
		01	2^{23}/fc	2^{24}/fc	
		10	2^{21}fc	2^{22}fc	
		11	2^{19}/fc	2^{20}/fc	
WDTOUT	Watchdog timer output select	0: Interrupt request 1: Reset request			Write only

Note 1: After clearing WDTCR1<WDTOUT> to “0”, the program cannot set it to “1”.

Note 2: fc: High-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 unknown data is read.

Note 4: To clear WDTCR1<WDTEN>, set the register in accordance with the procedures shown in “6.2.3 Watchdog Timer Disable”.

Note 5: If the watchdog timer is disabled during watchdog timer interrupt processing, the watchdog timer interrupt will never be cleared. Therefore, clear the watchdog timer (set the clear code (4EH) to WDTCR2) before disabling it, or disable the watchdog timer a sufficient time before it overflows.

Note 6: The watchdog timer consists of an internal divider and a two-stage binary counter. When clear code (4EH) is written, only the binary counter is cleared, not the internal divider.
Depending on the timing at which clear code (4EH) is written on the WDTCR2 register, the overflow time of the binary counter may be at minimum 3/4 of the time set in WDTCR1<WDTT>. Thus, write the clear code using a shorter cycle than 3/4 of the time set in WDTCR1<WDTT>.

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) 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>.

Note 5: WDTCR2 is a write-only register and must not be used with any of read-modify-write instructions. If WDTCR2 is read, a unknown data is read.

6.2.2 Watchdog Timer Enable

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

6.2.3 Watchdog Timer Disable

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

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

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

Example :Disabling the watchdog timer

```
DI                                  : IMF ← 0
LD      (WDTCR2), 04EH              : Clears the binary coutner
LDW     (WDTCR1), 0B101H            : WDTEN ← 0, WDTCR2 ← Disable code
EI                                  : IMF ← 1
```

Table 6-1 Watchdog Timer Detection Time (Example: fc = 20 MHz)

WDTT	Watchdog Timer Detection Time[s]	
	NORMAL Mode	
	DV1CK = 0	DV1CK = 1
00	1.678	3.355
01	419.430 m	838.861 m
10	104.858 m	209.715 m
11	26.214 m	52.429 m

Note: If the watchdog timer is disabled during watchdog timer interrupt processing, the watchdog timer interrupt will never be cleared. Therefore, clear the watchdog timer (set the clear code (4EH) to WDTCR2) before disabling it, or disable the watchdog timer a sufficient time before it overflows.

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, 02BFH              : Sets the stack pointer

LD      (WDTCR1), 00001000B    : WDTOUT ← 0
```

6.2.5 Watchdog Timer Reset

When a binary-counter overflow occurs while WDTCR1<WDTOUT> is set to “1”, a watchdog timer reset request is generated. When a watchdog timer reset request is generated, the internal hardware is reset. The reset time is maximum 24/fc [s] (max. 1.2 μs @ fc = 20 MHz).

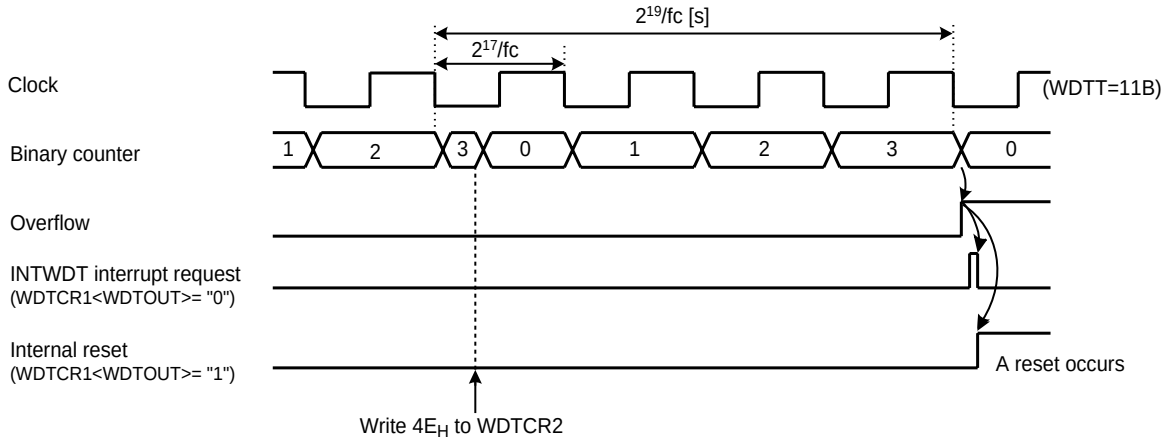


Figure 6-2 Watchdog timer Interrupt and Reset

7. Time Base Timer (TBT)

7.1 Time Base Timer

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

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 7-2).

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

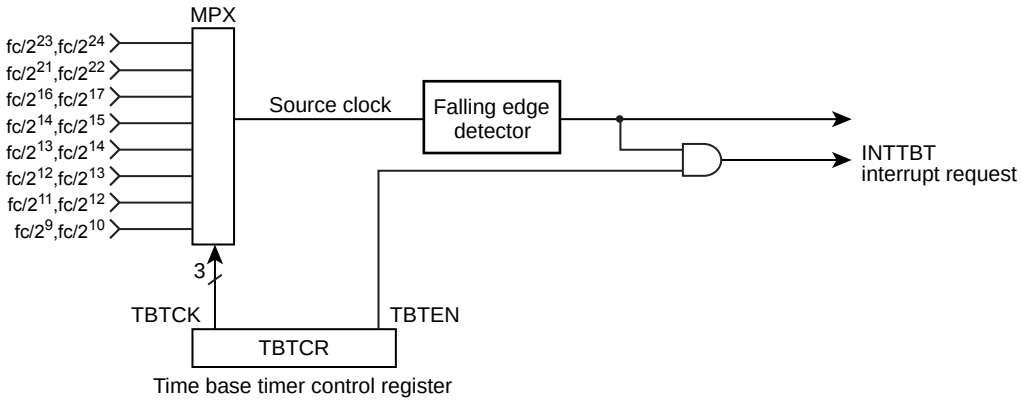


Figure 7-1 Time Base Timer configuration

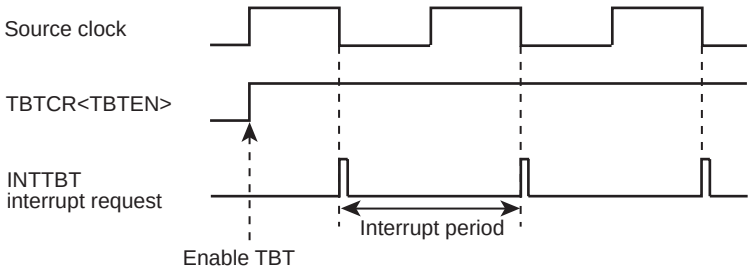


Figure 7-2 Time Base Timer Interrupt

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

```
LD      (TBTCR) ,  00000010B        ; TBTCK ← 010 (Freq. set)
LD      (TBTCR) ,  00001010B        ; TBTEN ← 1 (TBT enable)
DI
SET     (EIRL) .  6
EI
```

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

Time Base Timer Control Register

	7	6	5	4	3	2	1	0	
TBTCR (00036H)	0	0		0	TBTEN	TBTCK			(Initial Value: 0000 0000)

TBTEN	Time Base Timer Enable / Disable	0: Disable 1: Enable			
TBTCK	Time Base Timer interrupt Frequency select : [Hz]		NORMAL, IDLE Mode		R/W
			DV1CK=0	DV1CK=1	
		000	$fc/2^{23}$	$fc/2^{24}$	
		001	$fc/2^{21}$	$fc/2^{22}$	
		010	$fc/2^{16}$	$fc/2^{17}$	
		011	$fc/2^{14}$	$fc/2^{15}$	
		100	$fc/2^{13}$	$fc/2^{14}$	
		101	$fc/2^{12}$	$fc/2^{13}$	
		110	$fc/2^{11}$	$fc/2^{12}$	
		111	$fc/2^{9}$	$fc/2^{10}$	

Note 1: fc; High-frequency clock [Hz], *; Don't care
Note 2: Always set "0" in bit4 to bit7 on TBTCR register.

Table 7-1 Time Base Timer Interrupt Frequency (Example : fc = 20.0 MHz)

TBTCK	Time Base Timer Interrupt Frequency [Hz]	
	NORMAL, IDLE Mode	
	DV1CK = 0	DV1CK = 1
000	2.38	1.20
001	9.53	4.78
010	305.18	153.50
011	1220.70	610.35
100	2441.40	1220.70
101	4882.83	2441.40
110	9765.63	4882.83
111	39063.00	19531.25

8. 16-Bit TimerCounter 1 (TC1)

8.1 Configuration

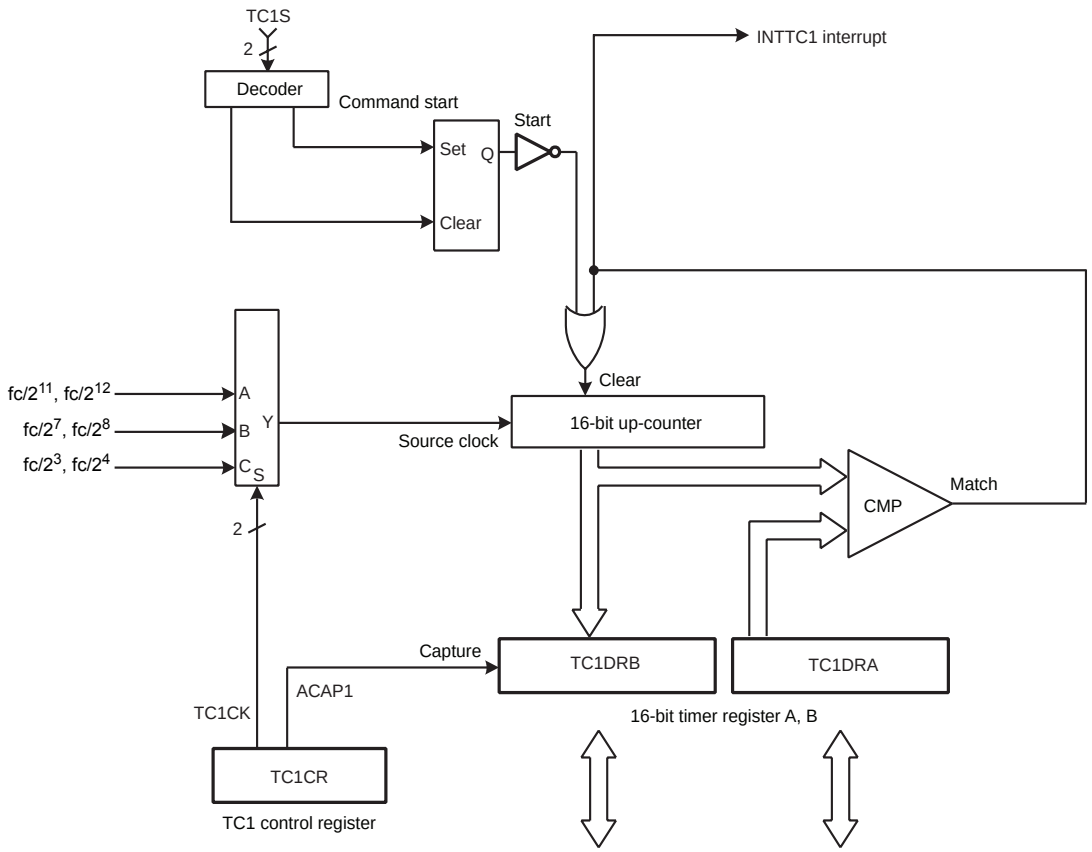


Figure 8-1 TimerCounter 1 (TC1)

8.2 TimerCounter Control

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

Timer Register

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TC1DRA (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 only							

TimerCounter 1 Control Register

	7	6	5	4	3	2	1	0	
TC1CR (000FH)	0	ACAP1	TC1S		TC1CK		TC1M		Read/Write (Initial value: 0000 0000)

ACAP1	Auto capture control	0:Auto-capture disable		1:Auto-capture enable	
TC1S	TC1 start control	00: Stop and counter clear 01: Command start 10: Reserved 11: Reserved			R/W
TC1CK	TC1 source clock select [Hz]		NORMAL, IDLE mode		R/W
			DV1CK = 0	DV1CK = 1	
		00	$fc/2^{11}$	$fc/2^{12}$	
		01	$fc/2^{7}$	$fc/2^{8}$	
		10	$fc/2^{3}$	$fc/2^{4}$	
		11	Reserved		
TC1M	TC1 operating mode select	00: Timer mode 01: Reserved 10: Reserved 11: Reserved			R/W

Note 1: fc: High-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) does not enable the setting of the timer register.

Note 3: To set the mode and source clock, write to TC1CR during TC1CR<TC1S>=00.

Note 4: To set the timer registers, the following relationship must be satisfied.
TC1DRA > 1

Note 5: Set TC1CR Register bit7 to “0”.

Note 6: 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 7: Since the up-counter value is captured into TC1DRB by the source clock of up-counter after setting TC1CR<ACAP1> to "1". Therefore, to read the captured value, wait at least one cycle of the internal source clock before reading TC1DRB for the first time.

8.3 Function

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 Source Clock for TimerCounter 1 (Example: fc = 20 MHz)

TC1CK	NORMAL, IDLE Mode			
	DV1CK = 0		DV1CK = 1	
	Resolution [μs]	Maximum Time Setting [s]	Resolution [μs]	Maximum Time Setting [s]
00	102.4	6.7108	204.8	13.4216
01	6.4	0.4194	12.8	0.8388
10	0.5	26.214 m	0.8	52.428 m

Example 1 :Setting the timer mode with source clock $fc/2^{11}$ [Hz] and generating an interrupt 1 second later (fc = 20 MHz, CGCR<DV1CK> = “0”)

```
LDW     (TC1DRA), 2625H      ; Sets the timer register (1 s ÷ 2^11/fc = 2625H)
DI                           ; IMF= “0”
SET     (EIRD). 2            ; 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
:       :                    ; Wait at least one cycle of the internal source clock
LD      WA, (TC1DRB)         ; Reads the capture value
```

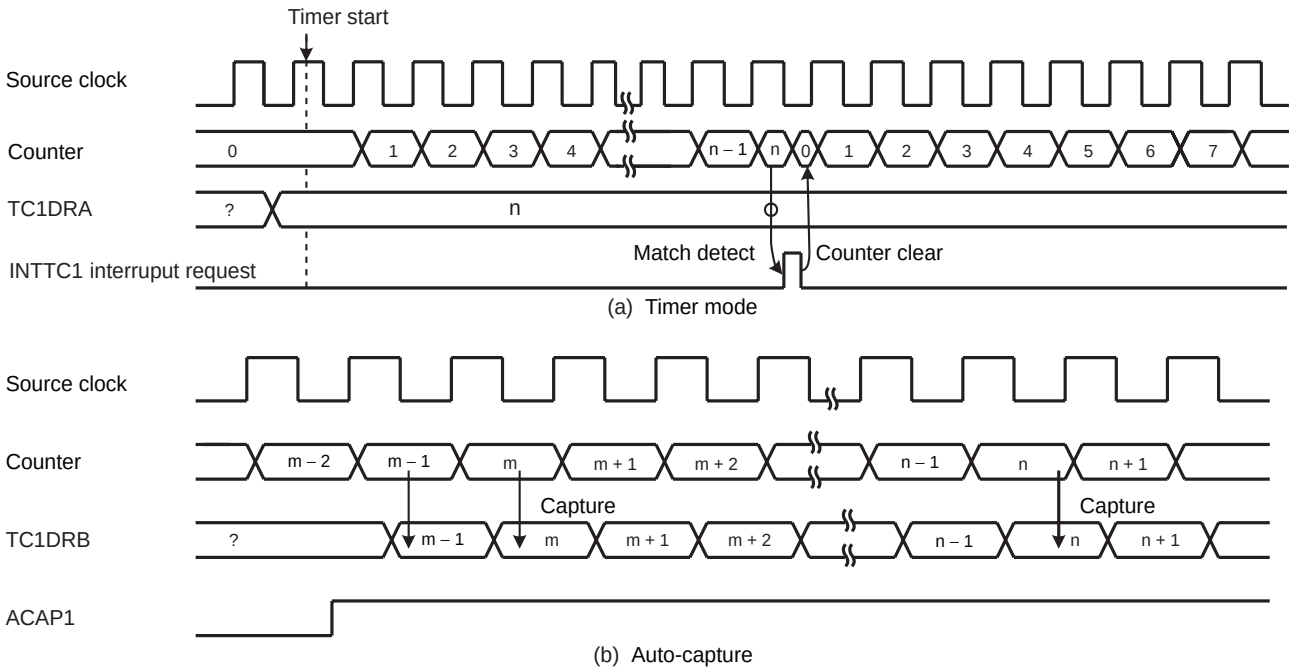
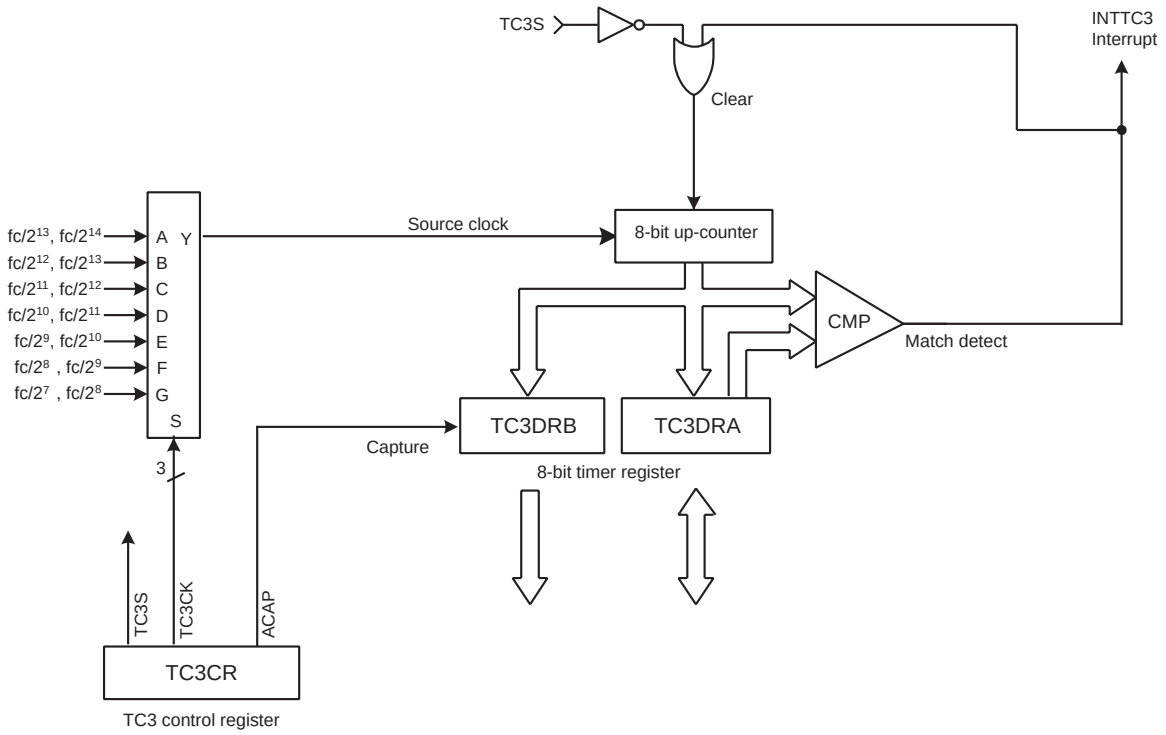


Figure 8-2 Timer Mode Timing Chart

9. 8-Bit TimerCounter 3 (TC3)

9.1 Configuration



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

Figure 9-1 TimerCounter 3 (TC3)

9.2 TimerCounter Control

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

Timer Register and Control Register

	7	6	5	4	3	2	1	0	
TC3DRA (001CH)									Read/Write (Initial value: 1111 1111)
TC3DRB (001DH)									Read only (Initial value: 1111 1111)

	7	6	5	4	3	2	1	0	
TC3CR (001EH)		ACAP		TC3S	TC3CK			TC3M	(Initial value: *0*0 0000)

ACAP	Auto capture control	0: – 1: Auto capture			R/W
TC3S	TC3 start control	0: Stop and counter clear 1: Start			R/W
TC3CK	TC3 source clock select [Hz]		NORMAL, IDLE mode		R/W
			DV1CK=0	DV1CK=1	
		000	$fc/2^{13}$	$fc/2^{14}$	
		001	$fc/2^{12}$	$fc/2^{13}$	
		010	$fc/2^{11}$	$fc/2^{12}$	
		011	$fc/2^{10}$	$fc/2^{11}$	
		100	$fc/2^{9}$	$fc/2^{10}$	
		101	$fc/2^{8}$	$fc/2^{9}$	
		110	$fc/2^{7}$	$fc/2^{8}$	
		111	Reserved		
TC3M	TC3 operating mode select	0: Timer mode 1: Reserved			R/W

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

Note 2: Set the source clock when TimerCounter stops (TC3CR<TC3S> = 0).

Note 3: To set the timer registers, the following relationship must be satisfied.
TC3DRA > 1

Note 4: When the read instruction is executed to TC3CR, the bit 5 and 7 are read as a don't care.

Note 5: Do not program TC3DRA when the timer is running (TC3CR<TC3S> = 1).

9.3 Function

9.3.1 Timer mode

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

Note: 00H which is stored in the up-counter immediately after detection of a match is not captured into TC3DRB. (Figure 9-2)

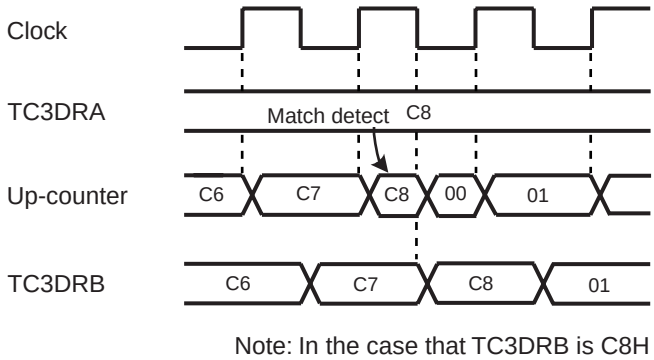
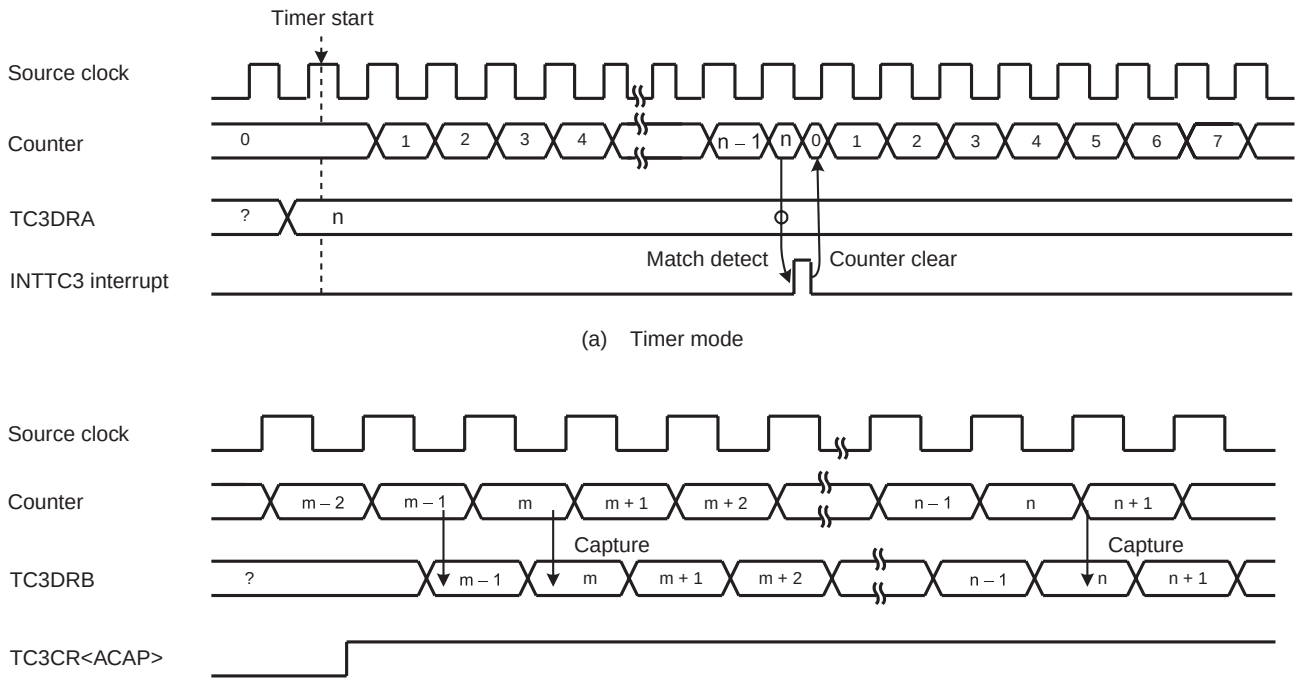


Figure 9-2 Auto-Capture Function

Table 9-1 Source Clock for TimerCounter 3 (Example: fc = 20 MHz)

TC3CK	NORMAL, IDLE mode			
	DV1CK = 0		DV1CK = 1	
	Resolution [μs]	Maximum Time Setting [ms]	Resolution [μs]	Maximum Time Setting [ms]
000	409.6	104.45	819.2	208.90
001	204.8	52.22	409.6	104.45
010	102.4	26.11	204.8	52.22
011	51.2	13.06	102.4	26.11
100	25.6	6.53	51.2	13.06
101	12.8	3.06	25.6	6.53
110	6.4	1.63	12.8	3.06



(a) Timer mode

(b) Auto capture

Figure 9-3 Timer Mode Timing Chart

10. 8-Bit TimerCounter 4 (TC4)

10.1 Configuration

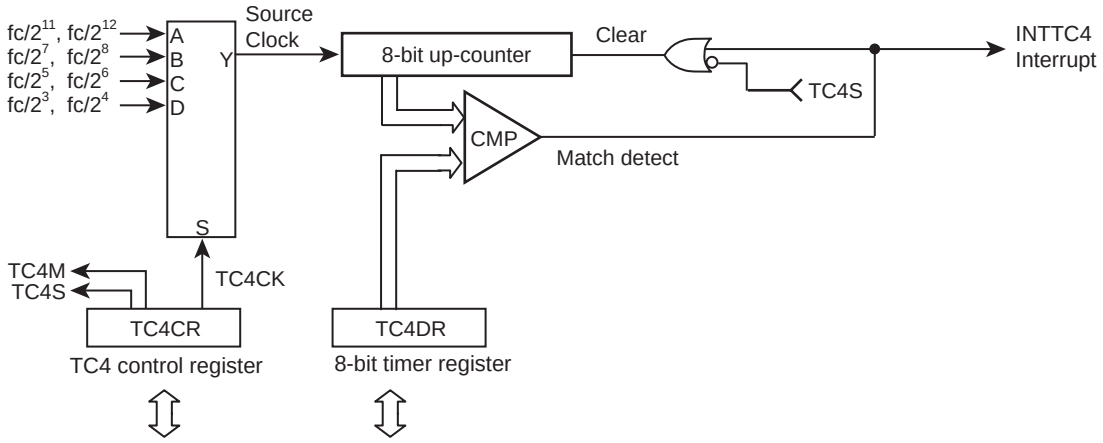


Figure 10-1 TimerCounter 4 (TC4)

10.2 TimerCounter Control

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

Timer Register and Control Register

TC4DR (001BH)	7	6	5	4	3	2	1	0	
									Read/Write (Initial value: 1111 1111)

TC4CR (001AH)	7	6	5	4	3	2	1	0	
			TC4S	TC4CK			TC4M		Read/Write (Initial value: **00 0000)

TC4S	TC4 start control	0: Stop and counter clear 1: Start			R/W
TC4CK	TC4 source clock select [Hz]		NORMAL, IDLE mode		R/W
			DV1CK = 0	DV1CK = 1	
		000	$fc/2^{11}$	$fc/2^{12}$	
		001	$fc/2^{7}$	$fc/2^{8}$	
		010	$fc/2^{5}$	$fc/2^{6}$	
		011	$fc/2^{3}$	$fc/2^{4}$	
		100	Reserved	Reserved	
		101	Reserved	Reserved	
		110	Reserved	Reserved	
		111	Reserved		
TC4M	TC4 operating mode select	00: Timer mode 01: Reserved 10: Reserved 11: Reserved			R/W

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

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

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

Note 4: The bit 6 and 7 of TC4CR are read as a don't care when these bits are read.

Note 5: Do not change the TC4DR setting when the timer is running.

10.3 Function

10.3.1 Timer Mode

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

Table 10-1 Internal Source Clock for TimerCounter 4 (Example: fc = 20 MHz)

TC4CK	NORMAL, IDLE Mode			
	DV1CK = 0		DV1CK = 1	
	Resolution [μs]	Maximum Time Setting [ms]	Resolution [μs]	Maximum Time Setting [ms]
000	102.4	26.11	204.8	52.22
001	6.4	1.63	12.8	3.28
010	1.6	0.41	3.2	0.82
011	0.4	0.10	0.8	0.20

11. Motor Control Circuit (PMD: Programmable motor driver)

The TMP88CH40NG contains one channel of motor control circuits used for sinusoidal waveform output. This motor control circuit can control brushless DC motors or AC motors with or without sensors. With its primary functions like those listed below incorporated in hardware, it helps to accomplish sine wave motor control easily, with the software load significantly reduced.

1. Rotor position detect function
   - Can detect the rotor position, with or without sensors
   - Can be set to determine the rotor position when detection matched a number of times, to prevent erroneous detection
   - Can set a position detection inhibit period immediately after PWM-on
2. Independent timer and timer capture functions for motor control
   - Contains one-channel magnitude comparison timer and two-channel coincidence comparison timers that operate synchronously for position detection
3. PWM waveform generating function
   - Generates 12-bit PWM with 100 ns resolution
   - Can set a frequency of PWM interrupt occurrence
   - Can set the dead time at PWM-on
4. Protective function
   - Provides overload protective function based on protection signal input
5. Emergency stop function in case of failure
   - Can be made to stop in an emergency by EMG input or timer overflow interrupt
   - Not easily cleared by software runaway
6. Auto commutation/Auto position detection start function
   - Comprised of dual-buffers, can activate auto commutation synchronously with position detection or timer
   - Can set a position detection period using the timer function and start auto position detection at the set time
7. Electrical angle timer function
   - Can count 360 degrees of electrical angle with a set period in the range of 0 to 383
   - Can output the counted electrical angle to the waveform arithmetic circuit
8. Waveform arithmetic circuit
   - Calculate the output duty cycle from the sine wave data and voltage data which are read from the RAM based on the electrical angle timer
   - Output the calculation result to the waveform synthesis circuit

11.1 Outline of Motor Control

The following explains the method for controlling a brushless DC motor with sine wave drive. In a brushless DC motor, the rotor windings to which to apply electric current are determined from the rotor's magnetic pole position, and the current-applied windings are changed as the rotor turns. The rotor's magnetic pole position is determined using a sensor such as a hall IC or by detecting polarity change (zero-cross) points of the induced voltage that develops in the motor windings (sensorless control). For the sensorless case, the induced voltage is detected by applying electric current to two phases and not applying electric current to the remaining other phase. In this two-phase current on case, there are six current application patterns as shown in Table 11-1, which are changed synchronously with the phases of the rotor. In this two-phase current on case, the current on time in each phase is 120 degrees relative to 180 degrees of the induced voltage.

Table 11-1 Current Application Patterns

Current Application Pattern	Upper Transistor			Lower Transistor			Current on Winding
	u	v	w	x	y	z	
Mode 0	ON	OFF	OFF	OFF	ON	OFF	U→V
Mode 1	ON	OFF	OFF	OFF	OFF	ON	U→W
Mode 2	OFF	ON	OFF	OFF	OFF	ON	V→W
Mode 3	OFF	ON	OFF	ON	OFF	OFF	V→U
Mode 4	OFF	OFF	ON	ON	OFF	OFF	W→U
Mode 5	OFF	OFF	ON	OFF	ON	OFF	W→V

Note: One of the upper or lower transistors is PWM controlled.

For brushless DC motors, the number of revolutions is controlled by an applied voltage, and the voltage application is controlled by PWM. At this time, the current on windings need to be changed in synchronism with the phases of the voltage induced by revolutions. Control timing in cases where the current on windings are changed by means of sensorless control is illustrated in Figure 11-4. For three-phase motors, zero-crossing occurs six times during one cycle of the induced voltage (electrical angle 360 degrees), so that the electrical angle from one zero-cross point to the next is 60 degrees. Assuming that this period comprises one mode, the rotor position can be divided into six modes by zero-cross points. The six current application patterns shown above correspond one for one to these six modes. The timing at which the current application patterns are changed (commutation) is out of phase by 30 degrees of electrical angle, with respect to the position detection by an induced voltage.

Mode time is obtained by detecting a zero-cross point at some timing and finding an elapsed time from the preceding zero-cross point. Because mode time corresponds to 60 degrees of electrical angle, the following applies for the case illustrated in Figure 11-4.

1. Current on windings changeover (commutation) timing
   30 degrees of electrical angle = mode time/2
2. Position detection start timing 45 degrees of electrical angle = mode time × 3/4
3. Failure determination timing 120 degrees of electrical angle = mode time × 2

Timings are calculated in this way. The position detection start timing in 2 is needed to prevent erroneous detection of the induced voltage for reasons that even after current application is turned off, the current continues flowing due to the motor reactance.

Control is exercised by calculating the above timings successively for each of the zero-cross points detected six times during 360 degrees of electrical angle and activating commutation, position detection start, and other operations according to that timing.

In this way, operations can be synchronized to the phases of the induced voltage of the motor.

The timing needed for motor control as in this example can be set freely as desired by using the internal timers of the microcontroller's PMD unit.

Also, sine wave control requires controlling the PWM duty cycle for each pulse. Control of PWM duty cycles is accomplished by counting degrees of electrical angle and calculating the sine wave data and voltage data at the counted degree of electrical angle.

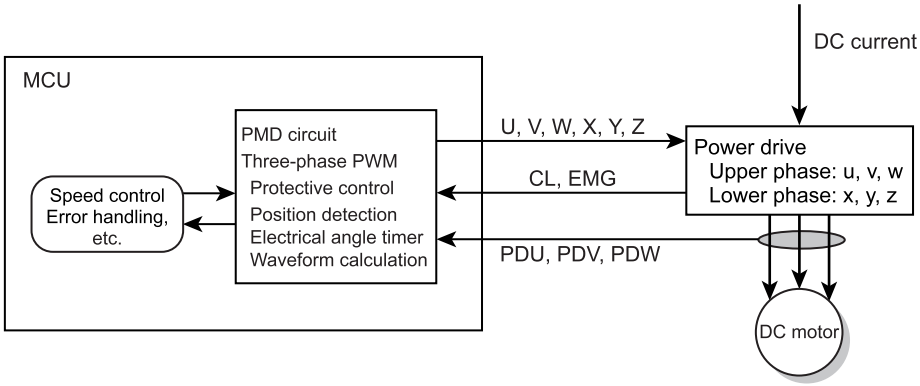


Figure 11-1 Conceptual Diagram of DC Motor Control

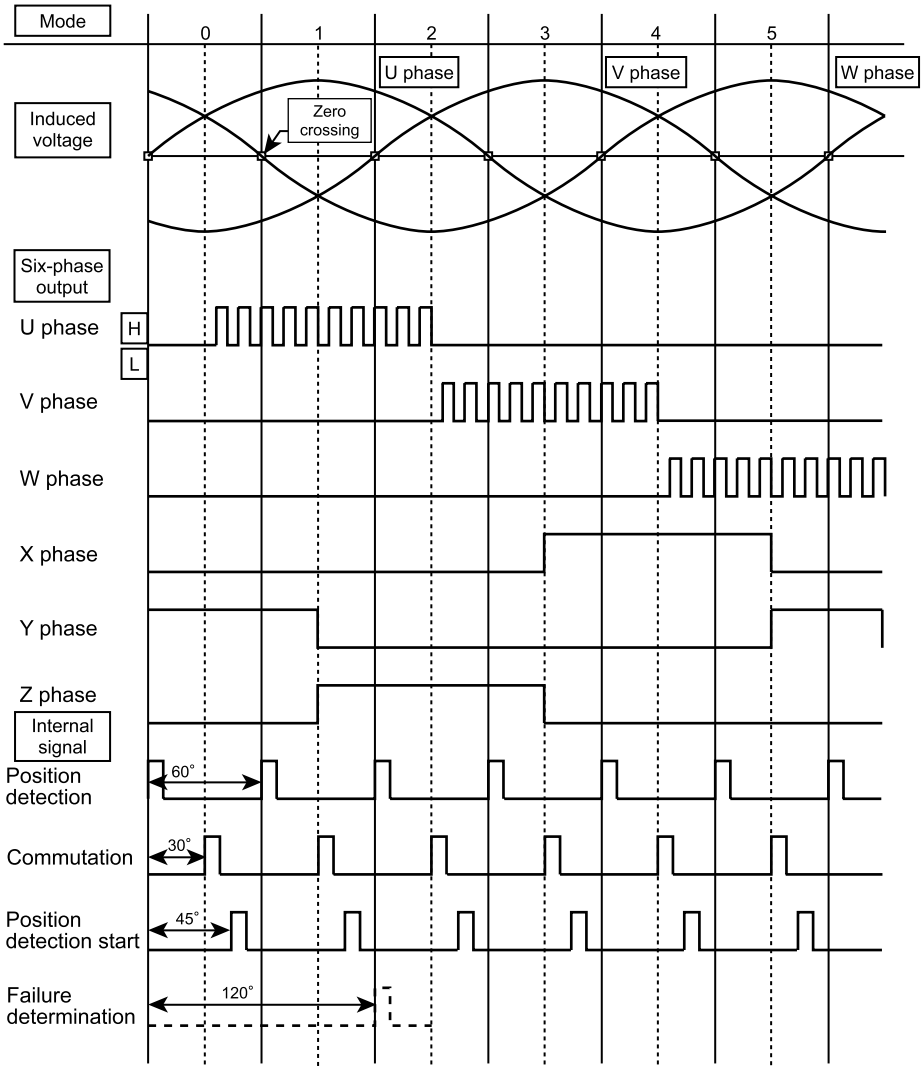


Figure 11-2 Example of Sensorless DC Motor Control Timing Chart

11.2 Configuration of the Motor Control Circuit

The motor control circuit consists of various units. These include a position detection unit to detect the zero-cross points of the induced voltage or position sensor signal, a timer unit to generate events at three instances of electrical angle timing, and a three-phase PWM output unit to produce three-phase output PWM waveforms. Also included are an electrical angle timer unit to count degrees of electrical angle and a waveform arithmetic unit to calculate sinusoidal waveform output duty cycles. The input/output units are configured as shown in the diagram below. When using ports for the PMD function, set the Port input/output control register (P3CRi) to 0 for the input ports, and for the output ports, set the data output latch (P3i) to 1 and then the port input/output control register to 1. Other input/output ports can be set in the same way for use of the PMD function.

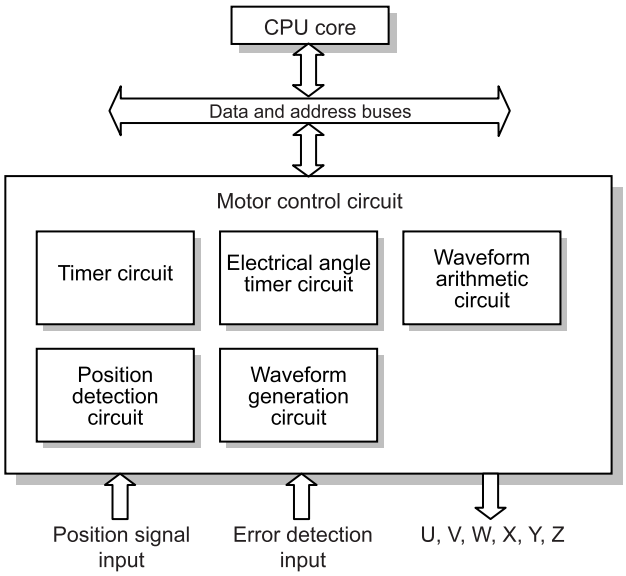


Figure 11-3 Block Diagram of the Motor Control Circuit

Note 1: Always use the LDW instruction to set data in the 9, 12 and 16-bit data registers.

Note 2: The EMG circuit initially is enabled. For PMD output, fix the EMG input port (P36) "H" high level or disable the EMG circuit before using for PMD output.

Note 3: The EMG circuit initially is enabled. When using Port P3 as input/output IO ports, disable EMG.

Note 4: When going to STOP mode, be sure to turn all of the PMD functions off before entering STOP mode.

11.3 Position Detection Unit

The Position Detection Unit identifies the motor's rotor position from input patterns on the position signal input port. Applied to this position signal input port is the voltage status of the motor windings for the case of sensorless DC motors or a Hall element signal for the case of DC motors with sensors included. The expected patterns corresponding to specific rotor positions are set in the PMD Output Register (MDOUT) beforehand, and when the input position signal and the expected value match as the rotation, a position detection interrupt (INTPDC) is generated. Also, unmatch detection mode is used to detect the direction of motor rotation, where when the status of the position detection input port changes from the status in which it was at start of sampling, a position detection interrupt is generated.

For three-phase brushless DC motors, there are six patterns of position signals, one for each mode, as summarized in Table 11-2 from the timing chart in Figure 11-2. Once a predicted position signal pattern is set in the MDOUT register, a position detection interrupt is generated the moment the position signal input port goes to mode indicated by this expected value. The position signals at each phase in the diagram are internal signals which cannot be observed from the outside.

Table 11-2 Position Signal Input Patterns

Position Detection Mode	U Phase (PDU)	V Phase (PDV)	W Phase (PDW)
Mode 0	H	L	H
Mode 1	H	L	L
Mode 2	H	H	L
Mode 3	L	H	L
Mode 4	L	H	H
Mode 5	L	L	H

11.3.1 Configuration of the position detection unit

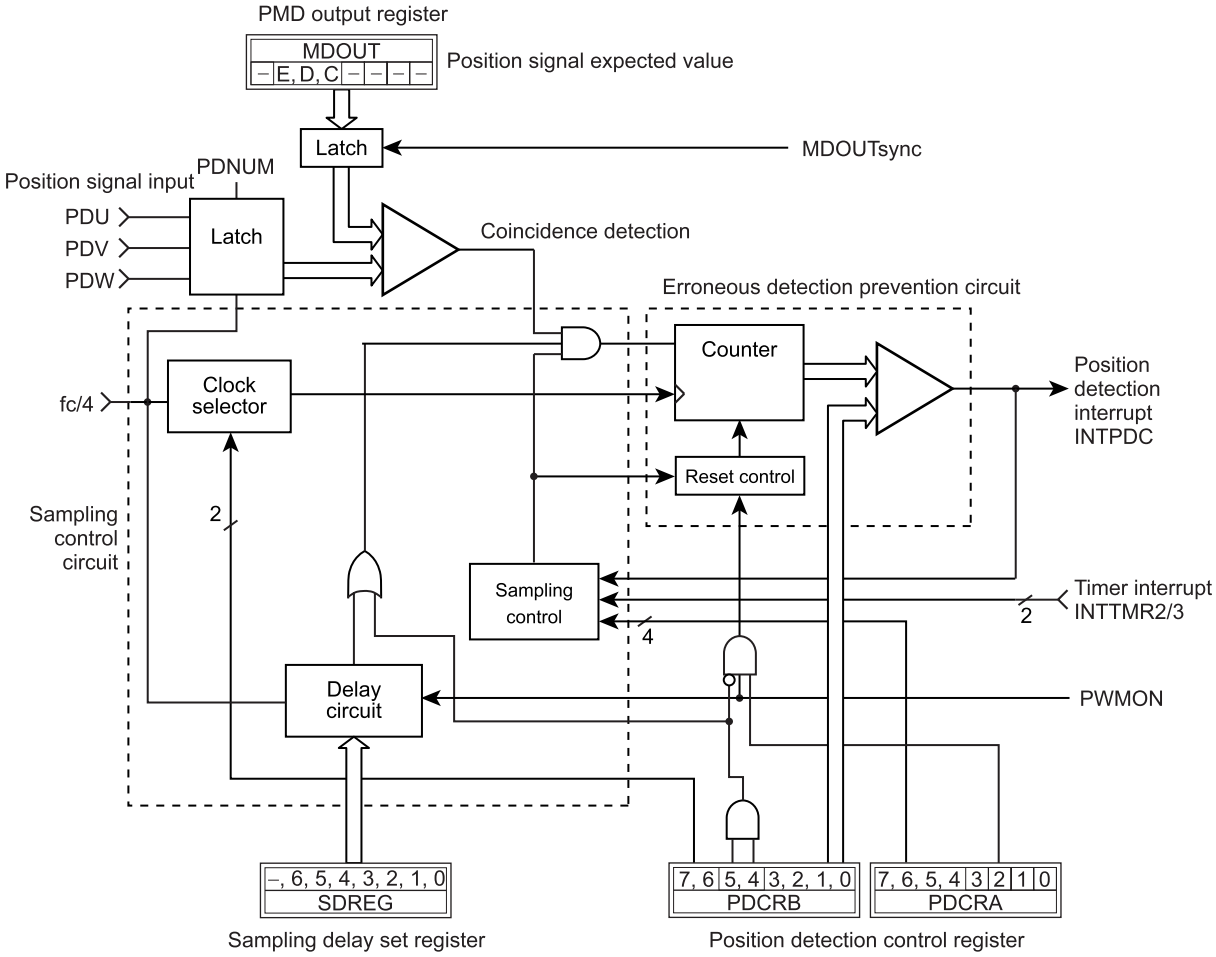


Figure 11-4 Configuration of the Position Detection Circuit

- The position detection unit is controlled by the Position Detection Control Register (PDCRA, PDCRB). After the position detection function is enabled, the unit starts sampling the position detection port with Timer 2 or in software. For the case of ordinary mode, when the status of the position detection input port matches the expected value of the PMD Output Register, the unit generates a position detection interrupt and finishes sampling, waiting for start of the next sampling.
- When unmatch detection mode is selected for position detection, the unit stores the sampled status of the position detection port in memory at the time it started sampling. When the port input status changes from the status in which it was at start of sampling, an interrupt is generated.
- In unmatch detection mode, the port status at start of sampling can be read (PDCRC<PDTCT>).
- When starting and stopping position detection synchronously with the timer, position detection is started by Timer 2 and position detection is stopped by Timer 3.
- Sampling mode can be selected from three modes available: mode where sampling is performed only while PWM is on, mode where sensors such as Hall elements are sampled regularly, and mode where sampling is performed while the lower side is conducting current (when performing sampling only while PWM is on, DUTY must be set for all three phases in common).
- When sampling mode is selected for detecting position while the lower phases are conducting current, sampling is performed for a period from when the set sampling delay time has elapsed after the lower side started conducting current till when the current application is turned off. Sampling is performed independently at each phase, and the sampling result is retained while sampling is idle. If while sampling at some phase is idle, the input and the expected value at other phase being sampled match, position is detected and an interrupt is generated.

- A sampling delay is provided for use in modes where sampling is made while PWM is on or the lower phases are conducting current. It helps to prevent erroneous detection due to noise that occurs immediately after the transistor turns on, by starting sampling a set time after the PWM signal turned on.
- When detecting position while PWM is on or the lower phases are conducting current, a method can be selected whether to recount occurrences of matched position detection after being compared for each PWM signal on (logical sum of three-phase PWM signals) (e.g., starting from 0 in each PWM cycle) or counting occurrences of matching continuously (PDCRB<SPLMD> is used to enable/disable recounting occurrences of matching while PWM is on).

11.3.2 Position Detection Circuit Register Functions

PDCRC

5, 4	EMEM	Hold result of position detection at PWM edge (Detect position detected position)	These bits hold the comparison result of position detection at falling or rising edge of PWM pulse. Bits 5 and 4 are set to 1 when position is detected at the falling or the rising edge, respectively. They show whether position is detected in the current PWM pulse, during PWM off, or in the immediately preceding PWM pulse.
3	SMON	Monitor sampling status	When read, this bit shows the sampling status.
2 to 0	PDTCT	Hold position signal input status	This bit holds the status of the position signal input at the time position detection started in unmatch mode.

PDCRB

7, 6	SPLCK	Sampling period	Select $fc/2^2$, $fc/2^3$, $fc/2^4$, or $fc/2^5$ for the position detection sampling period.
5, 4	SPLMD	Sampling mode	Select one of three modes: sampling only when PWM signal is active (when PWM is on), sampling regularly, or sampling when the lower side (X, Y, Z) phases are conducting current.
3 to 0	PDCMP	Sampling count	In ordinary mode, when the port status and the set expected value match and continuously match as many times as the sampling counts set, a position detection signal is output and an interrupt is generated. In unmatch detection mode, when the said status and value do not match and continuously unmatch as many times as the sampling counts set, a position detection signal is output and an interrupt is generated.

PDCRA

7	SWSTP	Stop sampling in software	Sampling can be stopped in software by setting this bit to 1 (e.g., by writing to this register). Sampling is performed before stopping and when position detection results match, a position detection interrupt is generated, with sampling thereby stopped.
6	SWSTT	Start sampling in software	Sampling can be started by setting this bit to 1 (e.g., by writing to this register).
5	SPTM3	Stop sampling using Timer 3	Sampling can be stopped by a trigger from Timer 3 by setting this bit to 1. Sampling is performed before stopping and when position detection results match, a position detection interrupt is generated, with sampling thereby stopped.
4	STTM2	Start sampling using Timer 2	Sampling can be started by a trigger from Timer 3 by setting this bit to 1.
3	PDNUM	Number of position signal input pins	Select whether to use three pins (PDU/PDV/PDW) or one pin (PDU only) for position signal input. When one pin is selected, the expected values of PDV and PDW are ignored. When performing position detection with two pins or a pin other than PDU, position signal input can be masked as 0 by setting unused pin(s) for output.
2	RCEN	Recount occurrences of matching when PWM is on	When performing sampling while PWM is on, occurrences of matching are recounted each time PWM signal turns on by setting this bit to 1 (when recounting occurrences of matching, the count is reset each time PWM turns off). When this bit is set to 0, occurrences of matching are counted continuously regardless PWM interval.
1	DTMD	Position detection mode	Setting this bit to 0 selects ordinary mode where position is detected when the expected value set in the register and the port input unmatch and then match. Setting this bit to 1 selects unmatch detection mode where position is detected at the time the port status changes to another one from the status in which it was when sampling started.
0	PDCEN	Position detection function	The position detection function is activated by setting this bit to 1.

SDREG

6 to 0	SDREG	Sampling delay	Set a time for which to stop sampling in order to prevent erroneous detection due to noise that occurs immediately after PWM output turns on (immediately after the transistor turns on). (Figure 11-5)

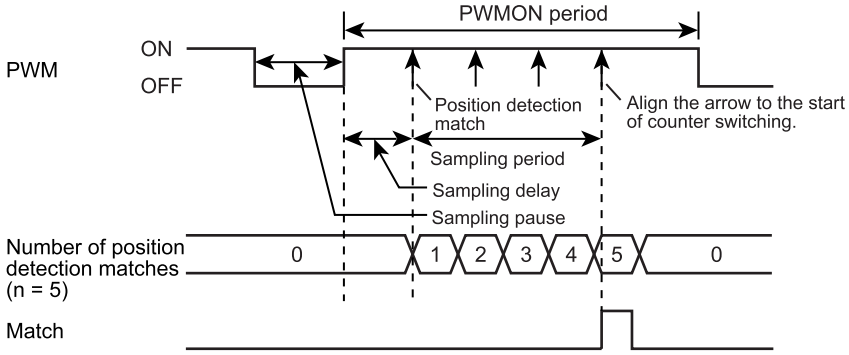


Figure 11-5 Position Detection Sampling Timing with the PWMON Period Selected

EMEM: Detects when a position detection match has occurred (the value is held aftr position detection).
(Check on whether sampling has started on the previous pulse)

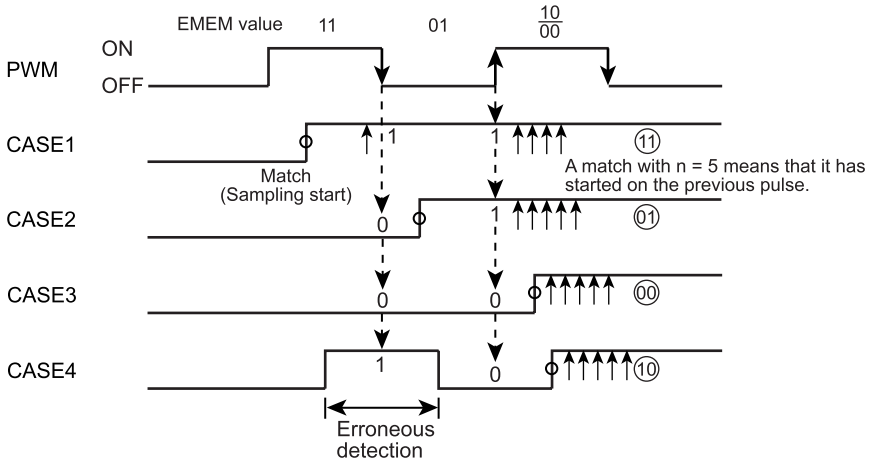


Figure 11-6 Detection Timing of the Position Detection Position

Position Detection Circuit Registers [Addresses (PMD1)]

PDCRC (01FA2H)

7	6	5	4	3	2	1	0
–	–	EMEM		SMON	PDTCT		

(Initial value: **00 0000)

5, 4	EMEM	Hold result of position detection at PWM edge (Detect position detected position)	00: Detected in the current pulse 01: Detected while PWM off 10: Detected in the current pulse 11: Detected in the preceding pulse	R
3	SMON	Monitor sampling status	0: Sampling idle 1: Sampling in progress	
2 to 0	PDTCT	Hold position signal input sta-tus	Holds the status of the position signal input during unmatch detection mode. Bits 2 to 0 correspond to W, V, and U phases.	

PDCRB (01FA1H)

7	6	5	4	3	2	1	0
SPLCK		SPLMD		PDCMP			

(Initial value: 0000 0000)

7, 6	SPLCK	Select sampling input clock	00: $fc/2^2$ [Hz] (200 ns at 20 MHz) 01: $fc/2^3$ (400 ns at 20 MHz) 10: $fc/2^4$ (800 ns at 20 MHz) 11: $fc/2^5$ (1.6 μs at 20 MHz)	R/W
5, 4	SPLMD	Sampling mode	00: Sample when PWM is on 01: Sample regularly 10: Sample when lower phases conducting current 11: Reserved	
3 to 0	PDCMP	Position detection matched counts	1 to 15 times (Counts 0 and 1 are assumed to be one time.)	

Note: When changing setting, keep the PDCEN bit reset to "0" (disable position detection function).

PDCRA (01FA0H)

7	6	5	4	3	2	1	0
SWSTP	SWSTT	SPTM3	STTM2	PDNUM	RCEN	DTMD	PDCEN

(Initial value: 0000 0000)

7	SWSTP	Stop sampling in software	0: No operation 1: Stop sampling	W
6	SWSTT	Start sampling in software	0: No operation 1: Start sampling	
5	SPTM3	Stop sampling using Timer 3	0: Disable 1: Enable	R/W
4	STTM2	Start sampling using Timer 2	0: Disable 1: Enable	
3	PDNUM	Number of position signal input pins	0: Compare three pins (PDU/PDV/PDW) 1: Compare one pin (PDU) only	
2	RCEN	Recount occurrences of match-ing when PWM is on	0: Continue counting from previously PWM on 1: Recount each time PWM turns on	
1	DTMD	Position detection mode	0: Ordinary mode 1: Unmatch detection mode	
0	PDCEN	Enable/Disable position detec-tion function	0: Disable 1: Enable (Sampling starts)	

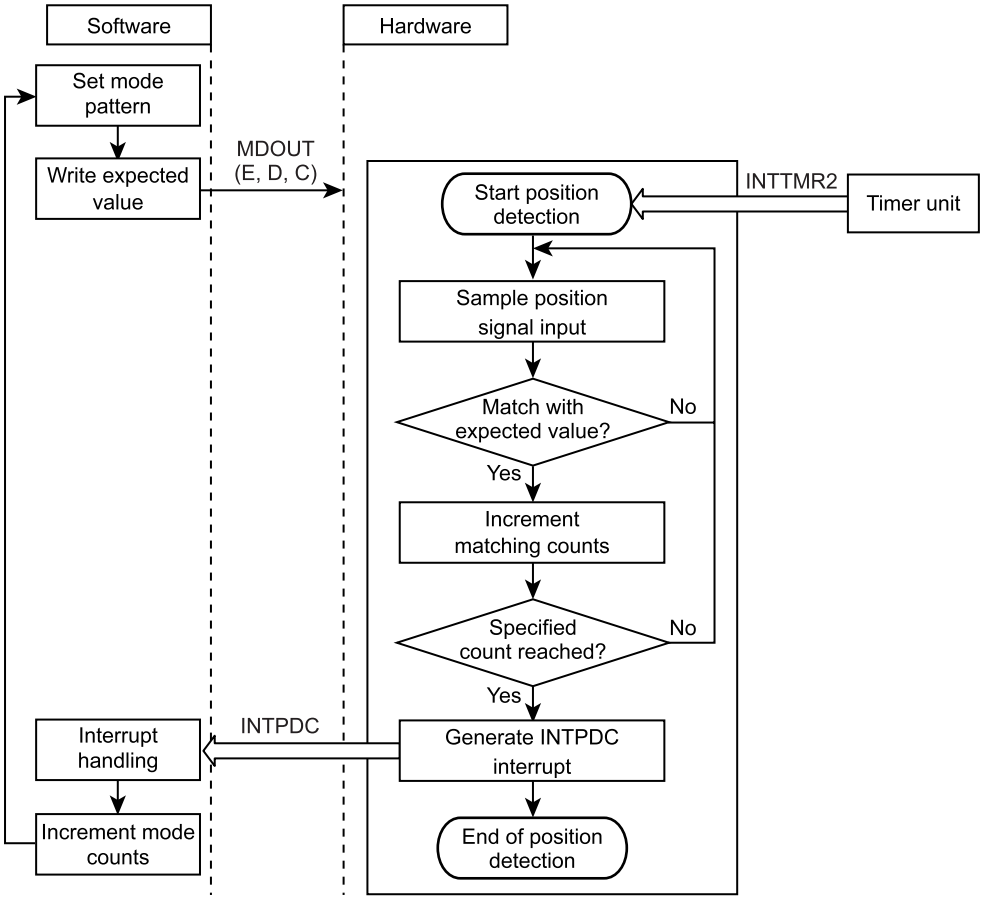
Note: Read-modify-write instructions, such as a bit manipulation instruction, cannot access the PDCRA because it contains a write only bit.

SDREG (01FA3H)	7	6	5	4	3	2	1	0	
	–	D6	D5	D4	D3	D2	D1	D0	(Initial value: *000 0000)

6 to 0	SDREG	Sampling delay	2^3/fc × n bits (n = 0 to 6, maximum 50.8 μs, resolution of 400 ns at 20 MHz)	R/W

Note: When changing setting, keep the PDCEN bit reset to "0" (disable position detection function).

11.3.3 Outline Processing in the Position Detection Unit



11.4 Timer Unit

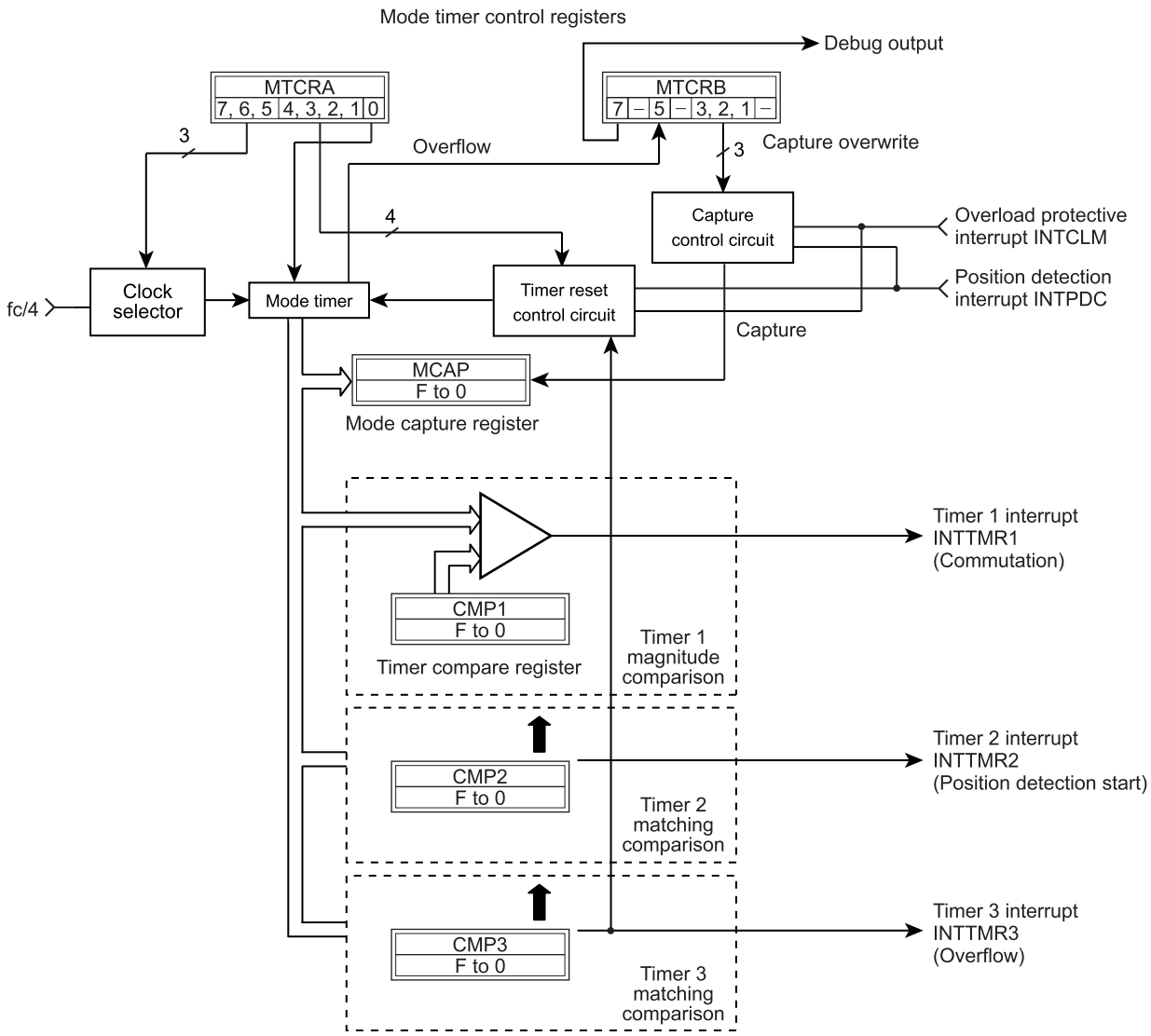


Figure 11-7 Timer Circuit Configuration

The timer unit has an up counter (mode timer) which is cleared by a position detection interrupt (INTPDC). Using this counter, it can generate three types of timer interrupts (INTTMR1 to 3). These timer interrupts may be used to produce a commutation trigger, position detection start trigger, etc. Also, the mode timer has a capture function which automatically captures register data in synchronism with position detection or overload protection. This capture function allows motor revolutions to be calculated by measuring position detection intervals.

11.4.1 Configuration of the Timer Unit

The timer unit consists mainly of a mode timer, three timer comparator, and mode capture register, and is controlled by timer control registers and timer compare registers.

- The mode timer can be reset by a signal from the position detection circuit, Timer 3, or overload protective circuit. If the mode timer overflows without being reset, it stops at FFFFH and sets an overflow flag in the control register.
- The value of the mode timer during counting can be read by capturing the count in software and reading the capture register.
- Timer 1 and Timers 2 and 3 generate an interrupt signal by magnitude comparison and matching comparison, respectively. Therefore, Timer 1 can generate an interrupt signal even when it could not write to the compare register in time and the counter value at the time of writing happens to exceed the register's set value.
- When any one of Timers 1 to 3 interrupts occurs, the next interrupts can be enabled by writing a new value to the respective compare registers (CMP1, CMP2, CMP3).
- When capturing by position detection is enabled, the capture register has the timer value captured in it each time position is detected. In this way, the capture register always holds the latest value.

11.4.1.1 Timer Circuit Register Functions

MTCRB

7	DBOUT	Debug output	Debug output can be produced by setting this bit to 1. Because interrupt signals to the interrupt control circuit are used for each interrupt, hardware debugging without software delays are possible. See the debug output diagram (Figure 11-8). Output ports: P67 for PMD1.
5	TMOF	Mode timer overflow	This bit shows that the timer has overflowed.
3	CLCP	Capture mode timer by overload protection	When this bit is set to 1, the timer value can be captured using the overload protection signal (CL) as a trigger.
2	SWCP	Capture mode timer in software	When this bit is set to 1, the timer value can be captured in software (e.g., by writing to this register).
1	PDCCP	Capture mode timer by position detection	When this bit is set to 1, the timer value can be captured using the position detection signal as a trigger.

MTCRA

7, 6, 5	TMCK	Select clock	Select the timer clock.
4	RBTM3	Reset mode timer from Timer 3	When this bit is set to 1, the mode timer is reset by a trigger from Timer 3.
3	RBCL	Reset mode timer by overload protection	When this bit is set to 1, the mode timer is reset by the overload protection signal (CL) as a trigger.
2	SWRES	Reset mode timer in software	When this bit is set to 1, the mode timer is reset in software (e.g., by writing to this register)
1	RBPDC	Reset mode timer by position detection	When this bit is set to 1, the mode timer is reset by the position detection signal as a trigger.
0	TMEN	Enable/disable mode timer	The mode timer is started by setting this bit to 1. Therefore, Timers 1 to 3 must be set with CMP before setting this bit. If this bit is set to 0 after setting CMP, CMP settings become ineffective.

MCAP	Mode capture	Position detection interval can be read out.

CMP1	Timer 1 (commutation)	Timers 1 to 3 are enabled while the mode timer is operating. An interrupt can be generated once by setting the corresponding bit in this register. The interrupt is disable when an interrupt is generated or the timer is reset. To use the timer again, set the register back again even if data is same.
CMP2	Timer 2 (position detection start)	
CMP3	Timer 3 (overflow)	

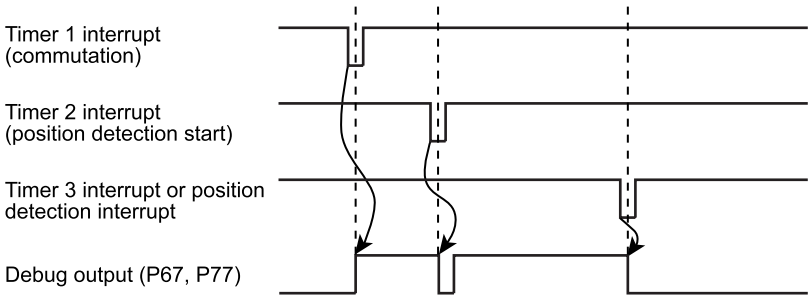


Figure 11-8 DBOUT Debug Output Diagram

Timer Circuit Registers [Addresses (PMD1)]

MTCRB (01FA5H)

7	6	5	4	3	2	1	0	
DBOUT	–	TMOF	–	CLCP	SWCP	PDCCP	–	(Initial value: 0*0*0 000*)

7	DBOUT	Debug output	0: Disable 1: Enable (P67 for PMD1, P77 for PMD2)	R/W
5	TMOF	Mode timer overflow	0: No overflow 1: Overflowed	R
3	CLCP	Capture mode timer by over-load protection	0: Disable 1: Enable	R/W
2	SWCP	Capture mode timer in software	0: No operation 1: Capture	W
1	PDCCP	Capture mode timer by position detection	0: Disable 1: Enable	R/W

Note: Read-modify-write instructions, such as a bit manipulation instruction, cannot access the MTCRB because it contains a write-only bit.

MTCRA (01FA4H)

7	6	5	4	3	2	1	0	
TMCK			RBTM3	RBCL	SWRES	RBPDC	TMEN	(Initial value: 0000 0000)

7, 6, 5	TMCK	Select clock	000: fc/2^3 (400 ns at 20 MHz) 010: fc/2^4 (800 ns at 20 MHz) 100: fc/2^5 (1.6 μs at 20 MHz) 110: fc/2^6 (3.2 μs at 20 MHz) 001: fc/2^7 (6.4 μs at 20 MHz) 011: Reserved 101: Reserved 111: Reserved	R/W
4	RBTM3	Reset mode timer from Timer 3	0: Disable 1: Enable	
3	RBCL	Reset mode timer by overload protection	0: Disable 1: Enable	
2	SWRES	Reset mode timer in software	0: No operation 1: Reset	W
1	RBPDC	Reset mode timer by position detection	0: Disable 1: Enable	R/W
0	TMEN	Enable/disable mode timer	0: Disable 1: Enable timer start	

Note 1: When changing MTCRA<TMCK> setting, keep the MTCRA<TMEN> bit reset to “0” (disable mode timer).

Note 2: Read-modify-write instructions, such as a bit manipulation instruction, cannot access the MTCRA because it contains a write-only bit.

MCAP (01FA7H, 01FA6H)

F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	
DF	DE	DD	DC	DB	DA	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	(Initial value: 0000 0000 0000 0000)

MCAP	Mode capture	Position detection interval	R

CMP1 (01FA9H, 01FA8H)

F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	
DF	DE	DD	DC	DB	DA	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	(Initial value: 0000 0000 0000 0000)

CMP2 (01FABH, 01FAAH)

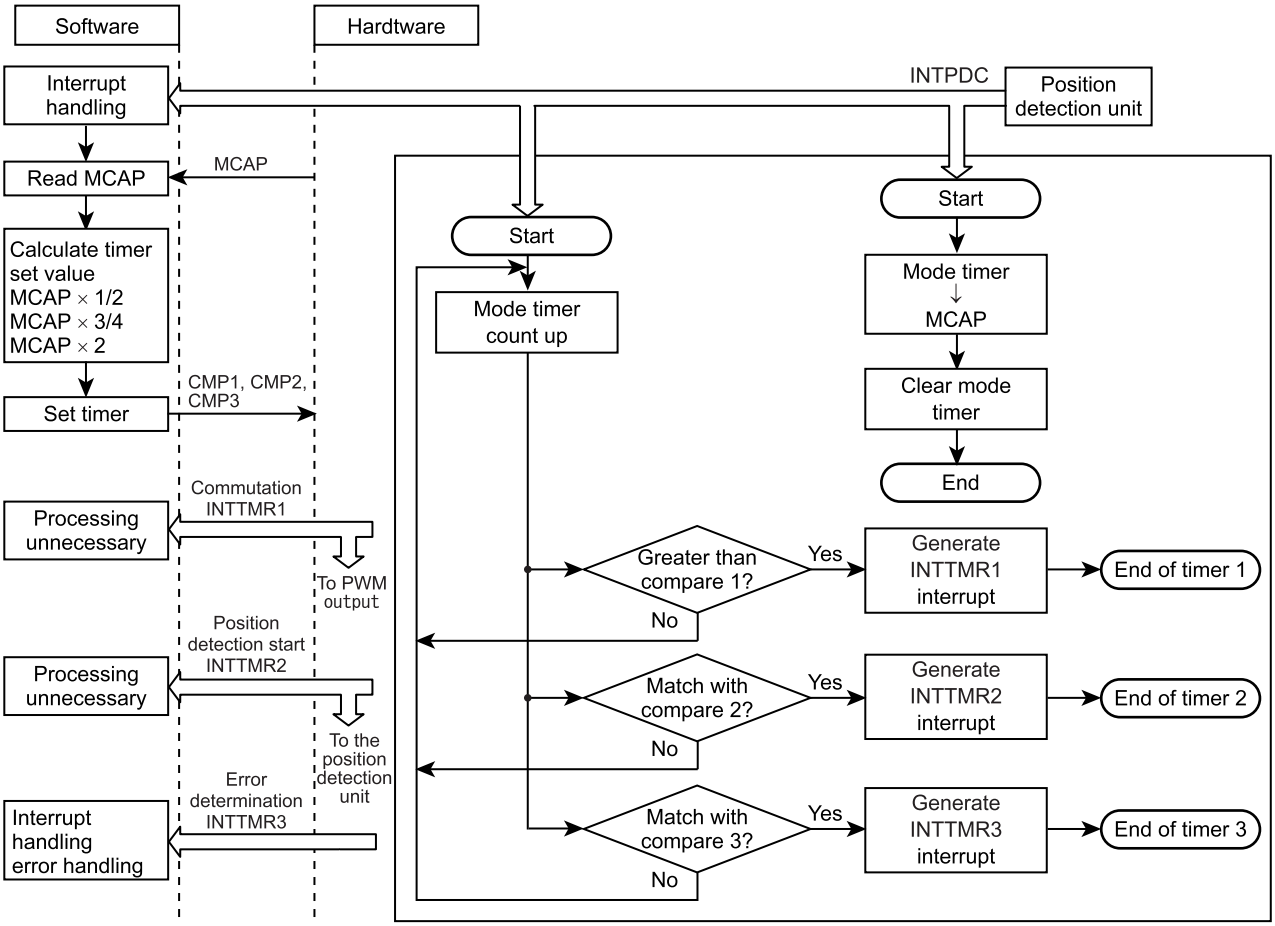
F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	
DF	DE	DD	DC	DB	DA	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	(Initial value: 0000 0000 0000 0000)

CMP3 (01FADH, 01FACH)	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	
	DF	DE	DD	DC	DB	DA	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	(Initial value: 0000 0000 0000 0000)

CMP1	Timer 1	Magnitude comparison compare register	R/W
CMP2	Timer 2	Matching comparison compare register	
CMP3	Timer 3	Matching comparison compare register	

Note: Read-modify-write instructions, such as a bit manipulation instruction, cannot access the MTCRB or MTCRA register because these registers contain write-only bits.

11.4.1.2 Outline Processing in the Timer Unit



11.5 Three-phase PWM Output Unit

The Three-phase PWM Output Unit has the function to generate three-phase PWM waves with any desired pulse width and the commutation function capable of brushless DC motor control. In addition, it has the protective functions such as overload protection and emergency stop functions necessary to protect the power drive unit, and the dead time adding function which helps to prevent the in-phase upper/lower transistors from getting shorted by simultaneous turn-on when switched over.

For the PWM output pin (U,V,W,X,Y,Z), set the port register PxDR and PxCR (x = 3) to 1. The PWM output initially is set to be active low, so that if the output needs to be used active high, set up the MDCRA Register accordingly.

11.5.1 Configuration of the three-phase PWM output unit

The three-phase PWM output unit consists of a pulse width modulation circuit, commutation control circuit, protective circuit (emergency stop and overload), and a dead time control circuit.

11.5.1.1 Pulse width modulation circuit (PWM waveform generating unit)

This circuit produces three-phase independent PWM waveforms with an equal PWM frequency. For PWM waveform mode, triangular wave modulation or sawtooth wave modulation can be selected by using the PMD Control Register (MDCRA) bit 1. The PWM frequency is set by using the PMD Period Register (MDPRD). The following shows the relationship between the value of this register and the PWM counter clock set by the MDCRB Register, PWMCK.

$$\text{Sawtooth wave PWM: MDPRD Register set value} = \frac{1}{\text{PWM frequency [Hz]} \times \text{PWMCK}}$$

$$\text{Triangular wave PWM: MDPRD Register set value} = \frac{1}{\text{PWM frequency [Hz]} \times 2 \times \text{PWMCK}}$$

The PMD Period Register (MDPRD) is comprised of dual-buffers, so that CMPU, V, W Register is updated with PWM period.

When the waveform arithmetic circuit is operating, the PWM waveform output unit receives calculation results from the waveform arithmetic circuit and by using the results as CMPU, V, W Register set value, it outputs independent three-phase PWM waveforms. When the waveform calculation function is enabled by the waveform arithmetic circuit and transfer of calculation results into the CMPU to W Registers is enabled (with EDCRA Register bit 2), the CMPU to W Registers are disabled against writing.

When the waveform calculation function is enabled (with EDCRA Register bit 1) and transfer of calculation results into the CMPU, V, W Registers is disabled (with EDCRA Register bit 4), the calculation results are transferred to the buffers of CMPU, V, W Registers, but not output to the port.

Read-accessing the CMPU, V, and W registers can read the calculation results of the waveform arithmetic circuit that have been input to a buffer. After changing the read calculation result data by software, writing the changed data to the CMPU, V, and W registers enables an arbitrary waveform other than a sinusoidal wave to be output. When the registers are read after writing, the values written to the registers are read out if accessed before the calculation results are transferred after calculation is finished.

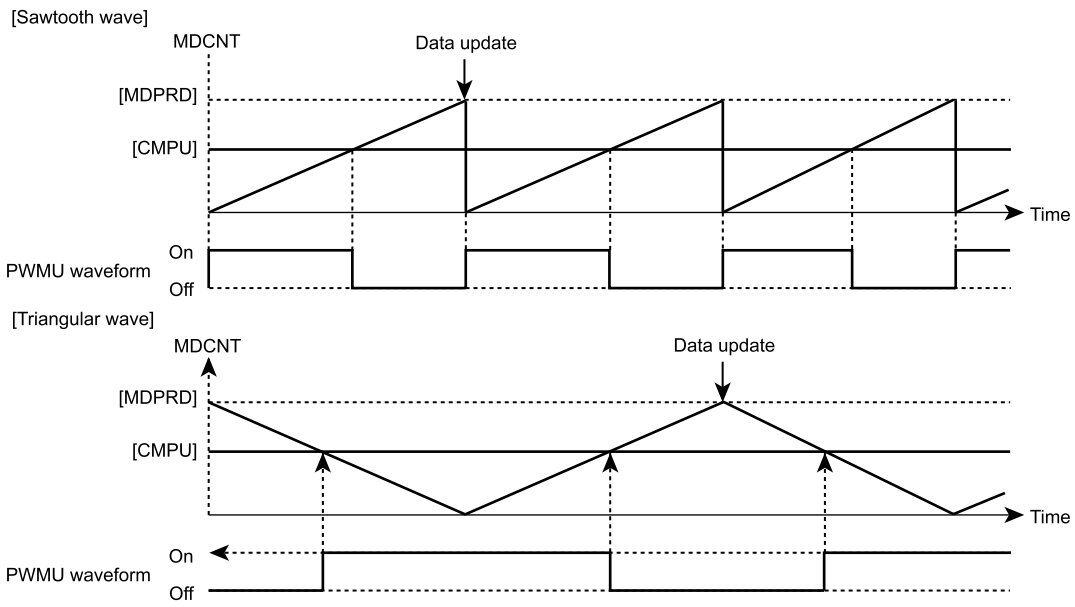


Figure 11-9 PWM Waveforms

The values of the PWM Compare Registers (CMPU/V/W) and the carrier wave generated by the PWM Counter (MDCNT) are compared for the relative magnitude by the comparator to produce PWM waveforms.

The PWM Counter is a 12-bit up/down counter with a 100 ns (at fc = 20 MHz) resolution.

For three-phase output control, two methods of generating three-phase PWM waveforms can be set.

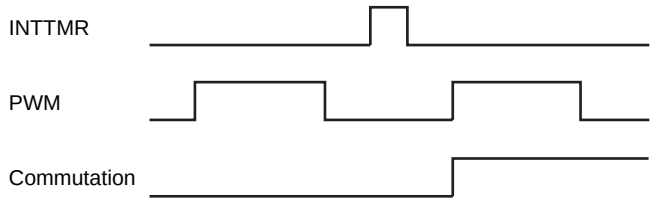
1. Three-phase independent mode: Values are set independently in the three-phase PMD Compare Registers to produce three-phase independent PWM waveforms. This method may be used to produce sinusoidal or any other desired drive waveforms.
2. Three-phase common mode: A value is set in only the U-phase PMD Compare Register to produce three in-phase PWM waveforms using the U phase set value. This method may be used for DC motor square wave drive.

The three-phase PMD Compare Registers each have a comparison register to comprise a dual-buffer structure. The values of the PMD Compare Registers are loaded into their respective comparison registers synchronously with PWM period.

11.5.1.2 Commutation control circuit

Output ports are controlled depending on the contents set in the PMD Output Register (MDOUT). The contents set in this register are divided into two, one for selecting the synchronizing signal for port output, and one for setting up port output. The synchronizing signal can be selected from Timers 1 or 2, position detection signal, or without sync. Port output can be synchronized to this synchronizing signal before being further synchronized to the PWM signal sync. The MDOUT Register's synchronizing signal select bit becomes effective immediately after writing. Other bits are dual-buffered, and are updated by the selected synchronizing signal.

Example: Commutation timing for one timer period with PWM synchronization specified



Output on six ports can be set to be active high or active low independently of each other by using the MDCRA Register bits 5 and 4. Furthermore, the U, V, and W phases can individually be selected between PWM output and H/L output by using the MDOUT Register bits A to 8 and 5 to 0. When PWM output is selected, PWM waveforms are output; when H/L output is selected, a waveform which is fixed high or low is output. The MDOUT Register bits E to C set the expected position signal value for the position detection circuit.

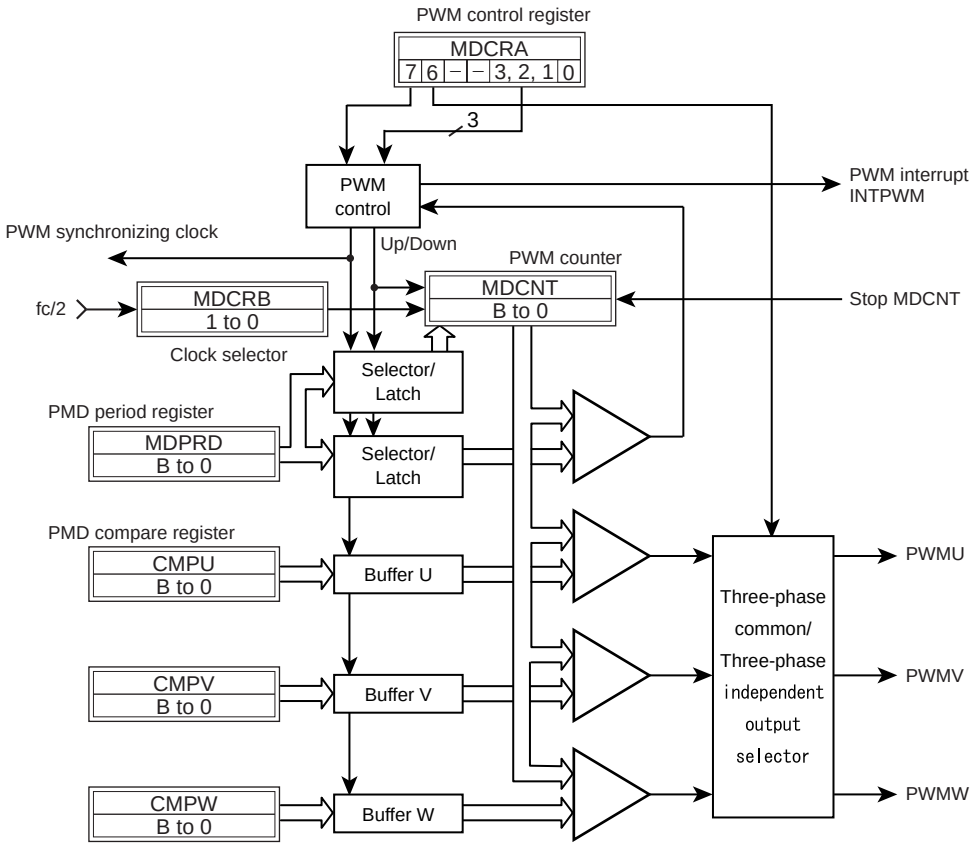


Figure 11-10 Pulse Width Modulation Circuit

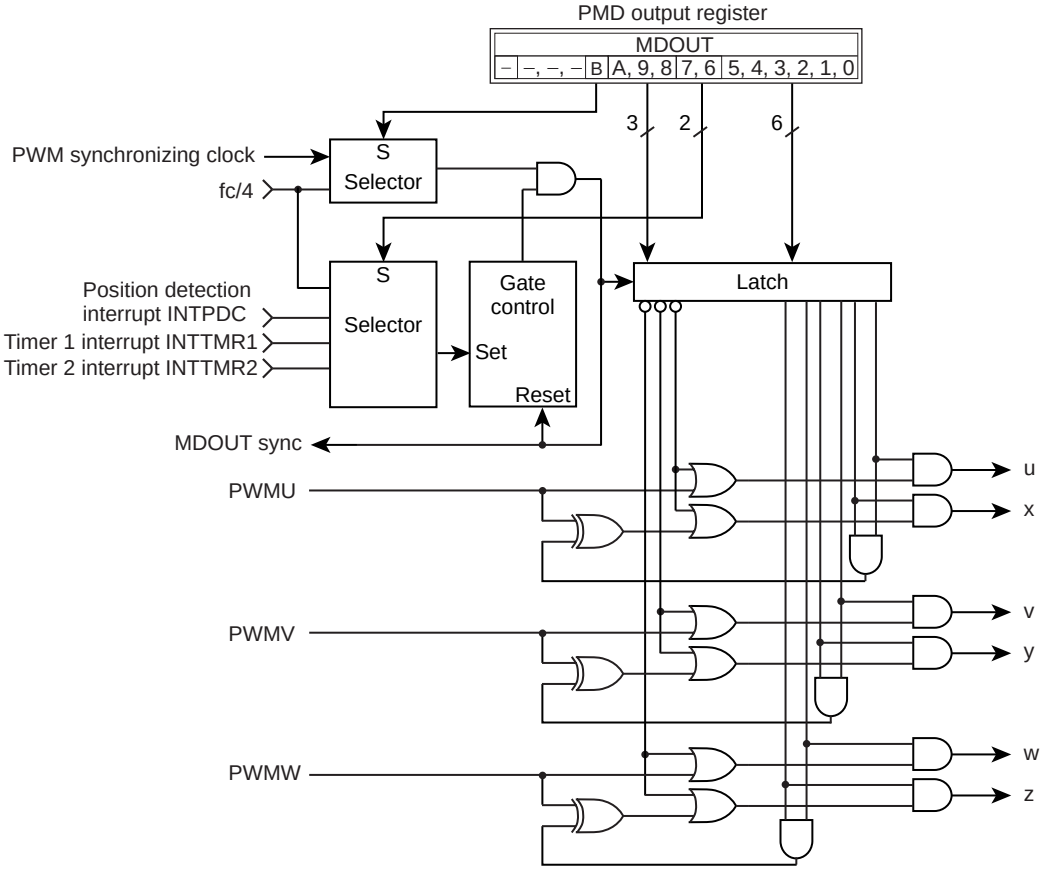


Figure 11-11 Commutation Control Circuit

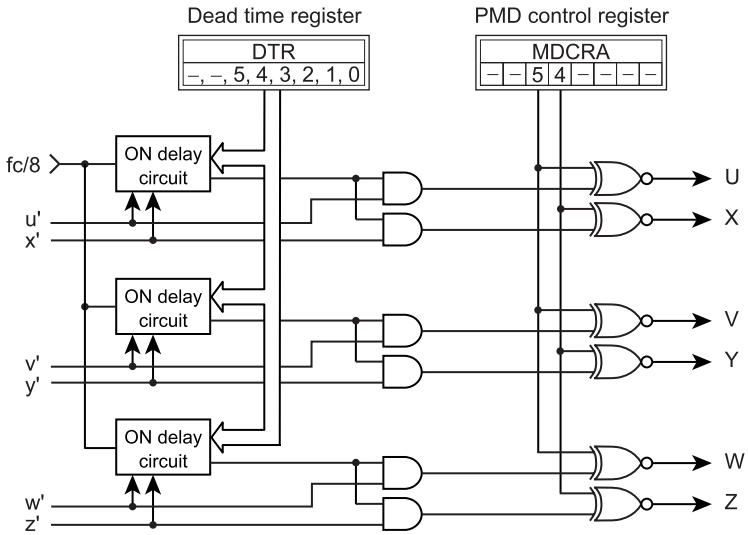


Figure 11-12 Dead Time Circuit

11.5.2 Register Functions of the Waveform Synthesis Circuit

MDCRB

PWMCK	Select PWM counter clock	Select PWM counter clock.

MDCRA

7	HLFINT	Select half-period interrupt	When this bit is set to 1, INTPWM is generated every half period (at triangular wave peak and valley) in the case of center PWM output and PINT = 00. In other cases, this setting has no meaning.
6	DTYMD	DUTY mode	Select whether to set the duty cycle independently for three phases using the CMPU to W Registers or in common for all three phases by setting the CMPU Register only.
5	POLH	Upper-phase port polarity	Select the upper-phase output port polarity. Make sure the waveform synthesis function (MDCRA Register bit 0) is idle before selecting this port polarity.
4	POLL	Lower-phase port polarity	Select the lower-phase output port polarity. Make sure the waveform synthesis function (MDCRA Register bit 0) is idle before selecting this port polarity.
3, 2	PINT	PWM interrupt frequency	Select the frequency at which to generate a PWM interrupt from four choices available: every PWM period or once every 2, 4, or 8 PWM periods. When setting of this bit is altered while operating, an interrupt may be generated at the time the bit is altered.
1	PWMMD	PWM mode	Select PWM mode. PWM mode 0 is an edge PWM (sawtooth wave), and PWM mode 1 is a center PWM (triangular wave).
0	PWMEN	Enable/Disable waveform generation circuit	When enabling this circuit (for waveform output), be sure to set the output port polarity and other bits of this register (other than MDCRA bit 0) beforehand.

DTR

DTR	Dead time	Set the dead time between the upper-phase and lower-phase outputs.

MDOUT

F	UPDWN	PWM counter flag	This bit indicates whether the PWM counter is counting up or down. When edge PWM (sawtooth wave) is selected, it is always set to 0.
E, D, C	PDEXP	Mode compare register	Set the data to be compared with the position detection input port. The comparison data is adopted as the expected value simultaneously when port output sync settings made with MDOUT are reflected in the ports. (This is the expected position detection input value for the output set with MDOUT next time.)
B	PSYNC	Select PWM synchronization	Select whether or not to synchronize port output to PWM period after being synchronized to the synchronizing signal selected with SYNCS. If selected to be synchronized to PWM, output is kept waiting for the next PWM after being synchronized with SYNCS. Waveform settings are overwritten if new settings are written to the register during this time, and output is generated with those settings.
A 9 8	WPWM VPWM UPWM	Control UVW-phase PWM outputs	Set U, V, and W-phase port outputs. (See the Table 11-3)
7, 6	SYNCS	Select port output sync signal	Select the synchronizing signal with which to output UVW-phase settings to ports. The synchronizing signal can be selected from Timers 1 or 2, position detection, or asynchronous. Select asynchronous when the initial setting, otherwise the above setting isn't reflected immediately.
5, 4 3, 2 1, 0	WOC VOC UOC	Control UVW-phase outputs	Set U, V, and W-phase port outputs. (See the Table 11-3)

MDCNT	PWM counter	This is a 12-bit read-only register used to count PWM periods.

MDPRD	Set PWM period	This register determines PWM period, and is dual-buffered, allowing PWM period to be altered even while the PWM counter is operating. The buffers are loaded every PWM period. When 100 ns is selected for the PWM counter clock, make sure the least significant bit is set to 0.

CMPU CMPV CMPW	Set PWM pulse width	This comparison register determines the pulse widths output in the respective UVW phases. This register is dual-buffered, and the pulse widths are determined by comparing the buffer and PWM counter.

Waveform Synthesis Circuit Registers [Addresses (PMD1)]

MDCRB (01FAFH)

7	6	5	4	3	2	1	0	
–	–	–	–	–	–	PWMCK		(Initial value: **** **00)

1, 0	PWMCK	PWM counterSelect clock	00: fc/2 [Hz] (100 ns at 20 MHz) 01: fc/2^2 (200 ns at 20 MHz) 10: fc/2^3 (400 ns at 20 MHz) 11: fc/2^4 (800 ns at 20 MHz)	R/W

Note: When changing setting, keep the PWMEN bit reset to “0” (disable wave form synthesis function).

MDCRA (01FAEH)

7	6	5	4	3	2	1	0	
HLFINT	DTYMD	POLH	POLL	PINT		PWMMD	PWMEN	(Initial value: 0000 0000)

7	HLFINT	Select half-period interrupt	0: Interrupt as specified in PINT 1: Interrupt every half period when PINT = 00	R/W
6	DTYMD	DUTY mode	0: U phase in common 1: Three phases independent	
5	POLH	Upper-phase port polarity	0: Active low 1: Active high	
4	POLL	Lower-phase port polarity	0: Active low 1: Active high	
3, 2	PINT	Select PWM interrupt (trigger)	00: Interrupt every period 01: Interrupt once every 2 periods 10: Interrupt once every 4 periods 11: Interrupt once every 8 periods	
1	PWMMD	PWM mode	0: PWM mode0 (Edge: Sawtooth wave) 1: PWM mode1 (Center: Triangular wave)	
0	PWMEN	Enable/disable waveform synthesis function	0: Disable 1: Enable (Waveform output)	

DTR (01FBEH)

7	6	5	4	3	2	1	0	
–	–	D5	D4	D3	D2	D1	D0	(Initial value: **00 0000)

5 to 0	DTR	Dead time	2^3/fc × 6 bit (maximum 25.2 μs at 20 MHz)	R/W

Note: When changing setting, keep the MDCRA<PWMEN> bit reset to "0" (disable wave form synthesis function).

MDOUT (01FB3H, 01FB2H)

F	E	D	C	B	A	9	8
UPDWN	PDEXP			PSYNC	WPWM	VPWM	UPWM
7	6	5	4	3	2	1	0
SYNCS		WOC		VOC		UOC	

(Initial value: 00000000 00000000)

F	UPDWN	PWM counter flag	0: Counting up 1: Counting down	R
E, D, C	PDEXP	Comparison register for position detection	bit E: W-phase expected value bit D: V-phase expected value bit C: U-phase expected value	R/W
B	PSYNC	Select PWM synchronization	0: Asynchronous 1: Synchronized	
A	WPWM	W-phase PWM output	0: H/L level output 1: PWM waveform output	
9	VPWM	V-phase PWM output	0: H/L level output 1: PWM waveform output	
8	UPWM	U-phase PWM output	0: H/L level output 1: PWM waveform output	
7, 6	SYNCS	Select port output synchronizing signal	00: Asynchronous 01: Synchronized to position detection 10: Synchronized to Timer 1 11: Synchronized to Timer 2	
5, 4	WOC	Control W-phase output	See the table 1-3	
3, 2	VOC	Control V-phase output		
1, 0	UOC	Control U-phase output		

11.5.3 Port output as set with UOC/VOC/WOC bits and UPWM/VPWM/WPWM bits

Table 11-3 Example of Pin Output Settings

U-phase output polarity: Active high (POLH,POLL = 1)

UOC	UPWM			
	1: PWM output		0: H/L level output	
	U phase	X phase	U phase	X phase
0 0	$\overline{PWM}$	PWM	L	L
0 1	L	PWM	L	H
1 0	PWM	L	H	L
1 1	PWM	$\overline{PWM}$	H	H

U-phase output polarity: Active low (POLH,POLL = 0)

UOC	UPWM			
	1: PWM output		0: H/L level output	
	U phase	X phase	U phase	X phase
0 0	PWM	$\overline{PWM}$	H	H
0 1	H	$\overline{PWM}$	H	L
1 0	$\overline{PWM}$	H	L	H
1 1	$\overline{PWM}$	PWM	L	L

MDCNT (01FB5H, 01FB4H)

F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
–	–	–	–	DB	DA	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

(Initial value: ****000000000000)

B to 0	PWM counter	PWM period counter value	R

MDPRD (01FB7H, 01FB6H)

F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
–	–	–	–	DB	DA	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

(Initial value: ****000000000000)

B to 0	PWM period	PWM period MDPRD ≥ 010H	R/W

CMPU (01FB9H, 01FB8H)

F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
–	–	–	–	DB	DA	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

(Initial value: ****000000000000)

CMPV (01FBBH, 01FBAH)

F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
–	–	–	–	DB	DA	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

(Initial value: ****000000000000)

CMPW (01FBDH, 01FBCH)

F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
–	–	–	–	DB	DA	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

(Initial value: ****000000000000)

B to 0	CMPU	PWM compare U register	Set U-phase duty cycle	R/W
	CMPV	PWM compare V register	Set V-phase duty cycle	
	CMPW	PWM compare W register	Set W-phase duty cycle	

11.5.4 Protective Circuit

This circuit consists of an EMG protective circuit and overload protective circuit. These circuits are activated by driving their respective port inputs active.

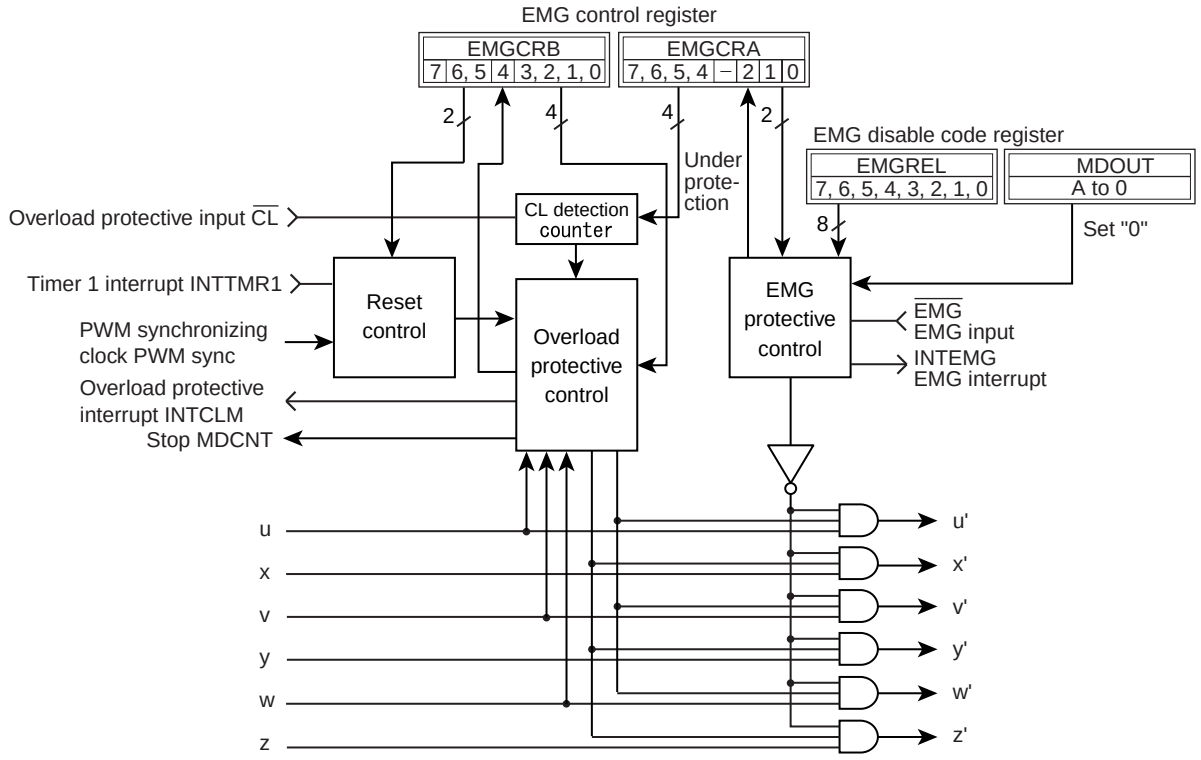


Figure 11-13 Configuration of the Protective Circuit

a. EMG protective circuit

This protective circuit is used for emergency stop, when the EMG protective circuit is enabled. When the signal on EMG input port goes active (negative edge triggered), the six ports are immediately disabled high-impedance against output and an EMG interrupt (INTEMG) is generated. The EMG Control Register (EMGCRA) is used to set EMG protection. If the EMGCRA<EMGST> shows the value “1” when read, it means that the EMG protective circuit is operating. To return from the EMG protective state, reset the MDOUT Register bits A to 0 and set the EMGCRA<RTE> to 1. Returning from the EMG protective state is effective when the EMG protective input has been released back high. To disable the EMG function, set data “5AH“ and “A5H“sequentially in the EMG disable Register (EMGREL) and reset the EMGCRA<EMGEN> to 0. When the EMG function is disabled, EMG interrupts (INTEMG) are not generated.

The EMG protective circuit is initially enabled. Before disabling it, fully study on adequacy.

b. Overload protective circuit

The overload protective circuit is set by using the EMG Control Registers (EMGCRA/B). To activate overload protection, set the EMGCRB<CLEN> to 1 to enable the overload protective circuit. The circuit starts operating when the overload protective input is pulled low.

To return from overload state, there are three methods to use: return by a timer (EMGCRB<RTTM1>), return by PWM sync (EMGCRB<RTPWM>), or return manually (EMGCRB<RTCL>). These methods are usable when the overload protective input has been released back high.

The number of times the overload protective input is sampled can be set by using the EMGCRA<CLCNT>. The sampling times can be set in the range of 1 to 15 times at 200 ns period (when fc = 20 MHz). If a low level is detected as many times as the specified number, overload protection is assumed.

The output disabled phases during overload protection are set by using the EMGCRB<CLMD>. This facility allows selecting to disable no phases, all phases, PWM phases, or all upper phases/all lower phases. When selected to disable all upper phases/all lower phases, port output is determined by their turn-on status immediately before being disabled. When two or more upper phases are active, all upper phases are turned on and all lower phases are turned off; when two or more lower phases are active, all upper phases are turned off and all lower phases are turned on.

When output phase are cut off, output is inactive (low in the case of high active). When the overload protective circuit is disabled, overload protective interrupts (INTCLM) are not generated.

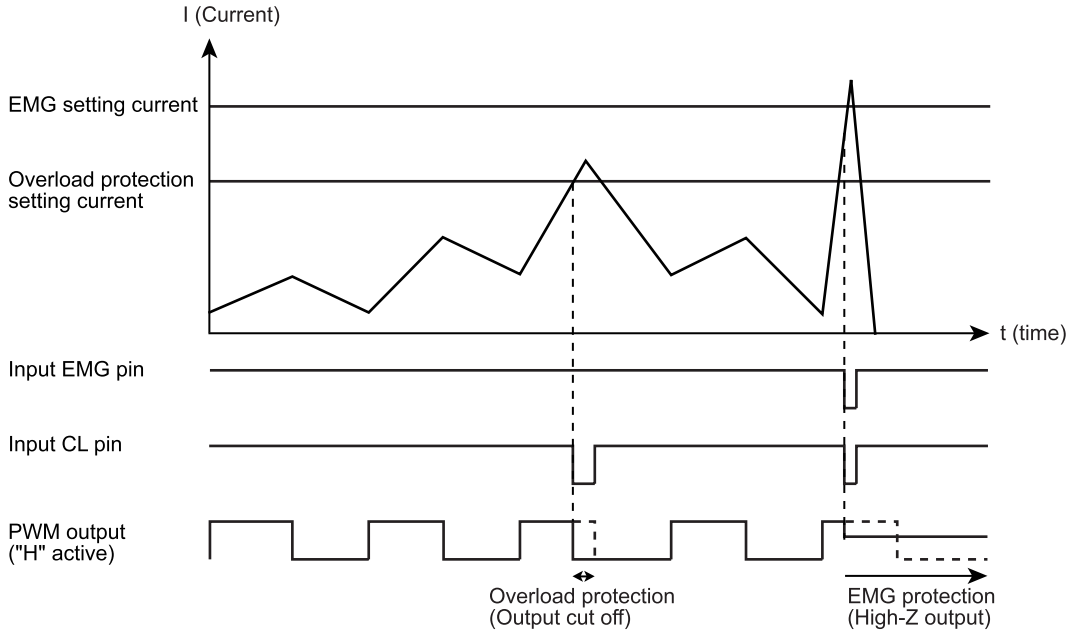


Figure 11-14 Example of Protection Circuit Operation

11.5.5 Functions of Protective Circuit Registers

EMGREL	EMG disable	The EMG protective circuit is disable from the disabled state by writing “5AH“ and “A5H“ to this register in that order. After that, the EMGCRA Register needs to be set.

EMGCRB

7	RTCL	Return from overload protective state	When this bit is set to 1, the motor control circuit is returned from overload protective state in software (e.g., by writing to this register). Also, the current state can be known by reading this bit. MDOUT outputs at return from the overload protective state remain as set before the overload protective input was driven active.
6	RTPWM	Return by PWM sync	When this bit is set to 1, the motor control circuit is returned from overload protective state by PWM sync. If RTCL is set to 1, RTCL has priority.
5	RTTM1	Return by timer sync	When this bit is set to 1, the motor control circuit is returned from overload protective state by Timer 1 sync. If RTCL is set to 1, RTCL has priority.
4	CLST	Overload protective state	The status of overload protection can be known by reading this bit.
3, 2	CLMD	Select output disabled phases during overload protection	Select the phases to be disabled against output during overload protection. This facility allows selecting to disable no phases, all phases, PWM phases, or all upper phases/all lower phases.
1	CNTST	Stop counter during overload protection	Can stop the PWM counter during overload protection.
0	CLEN	Enable/Disable overload protection	Enable or disable the overload protective function.

EMGCRA

7 to 4	CLCNT	Overload protection sampling time	Set the length of time the overload protective input port is sampled.
2	EMGST	EMG protective state	The status of EMG protection can be known by reading this bit.
1	RTE	Return from EMG protective state	The motor control circuit is returned from EMG protective state by setting this bit to “1” . When returning, set the MDOUT Register A to 0 bits to “0” . Then set the EMGCRA Register bit 1 to “1” and set MDOUT waveform output. Then set up the MDCRA Register.
0	EMGEN	Enable/Disable EMG protective circuit	The EMG protective circuit is activated by setting this bit to 1. This circuit initially is enabled. (To disable this circuit, make sure key code 5AH and A5H are written to the EMGREL1 Register beforehand.)

Protective Circuit Registers [Addresses (PMD1)]

EMGREL (01FBFH)

7	6	5	4	3	2	1	0	
D7	D6	D5	D4	D3	D2	D1	D0	(Initial value: 0000 0000)

7 to 0	EMGREL	EMG disable	Can disable by writing 5AH and then A5H.	W

Note: Read-modify-write instructions, such as a bit manipulation instruction, cannot access the EMGREL register because this register is write only.

EMGCRB (01FB1H)

7	6	5	4	3	2	1	0	
RTCL	RTPWM	RTTM1	CLST	CLMD		CNTST	CLEN	(Initial value: 0000 0000)

7	RTCL	Return from overload protective state	0: No operation 1: Return from protective state	W
6	RTPWM	Enable/Disable return from overload protective state by PWM sync	0: Disable 1: Enable	R/W
5	RTTM1	Enable/Disable return from overload protective state by timer 1	0: Disable 1: Enable	
4	CLST	Overload protective state	0: No operation 1: Under protection	R
3, 2	CLMD	Select output disabled phases during overload protection	00: No phases disabled against output 01: All phases disabled against output 10: PWM phases disabled against output 11: All upper/All lower phases disabled against output (Note)	R/W
1	CNTST	Stop PWM counter during overload protection	0: Do not stop 1: Stop the counter	
0	CLEN	Enable/Disable overload protective circuit	0: Disable 1: Enable	

Note: If during overload protection the port output state in two or more upper phases is on, all lower phases are disabled and all upper phases are enabled for output; when two or more lower phases are on, all upper phases are disabled and all lower phases are enabled for output.

EMGCRA (01FB0H)

7	6	5	4	3	2	1	0	
CLCNT					EMGST	RTE	EMGEN	(Initial value: 0000 *001)

7 to 4	CLCNT	Overload protection sampling number of times.	2^2/fc × n (n = 1 to 15, 0 and 1 are set as 1 at 20 MHz)	R/W
2	EMGST	EMG protective state	0: No operation 1: Under protection	R
1	RTE	Return from EMG state	0: No operation 1: Return from protective state (Note 1)	W
0	EMGEN	Enable/Disable EMG protective circuit	0: Disable 1: Enable	R/W

Note 1: An instruction specifying a return from the EMG state is invalid if the $\overline{\text{EMG}}$ input is “L”.

Note 2: Read-modify-write instructions, such as a bit manipulation instruction, cannot access the EMGCRB or EMGCRA register because these registers contain write-only bits.

11.6 Electrical Angle Timer and Waveform Arithmetic Circuit

Electrical Angle Timer

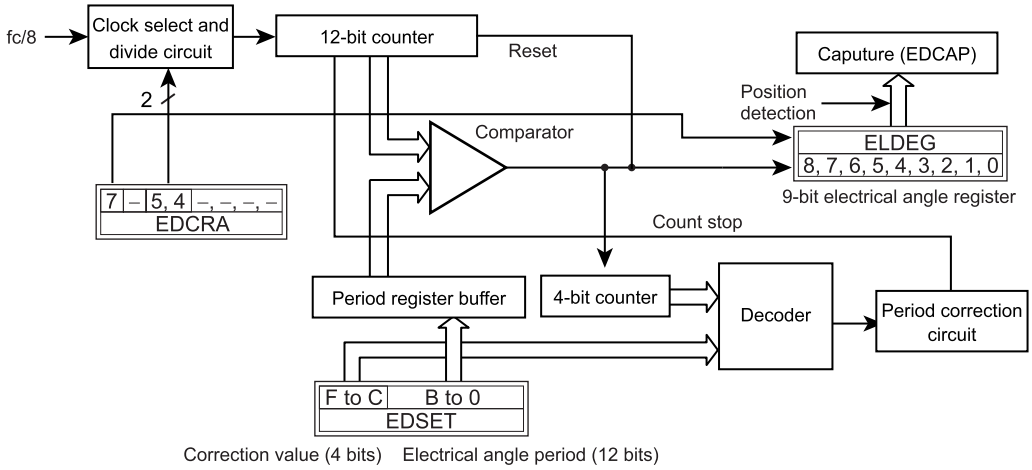


Figure 11-15 Electrical Angle Timer Circuit

Waveform Arithmetic Circuit

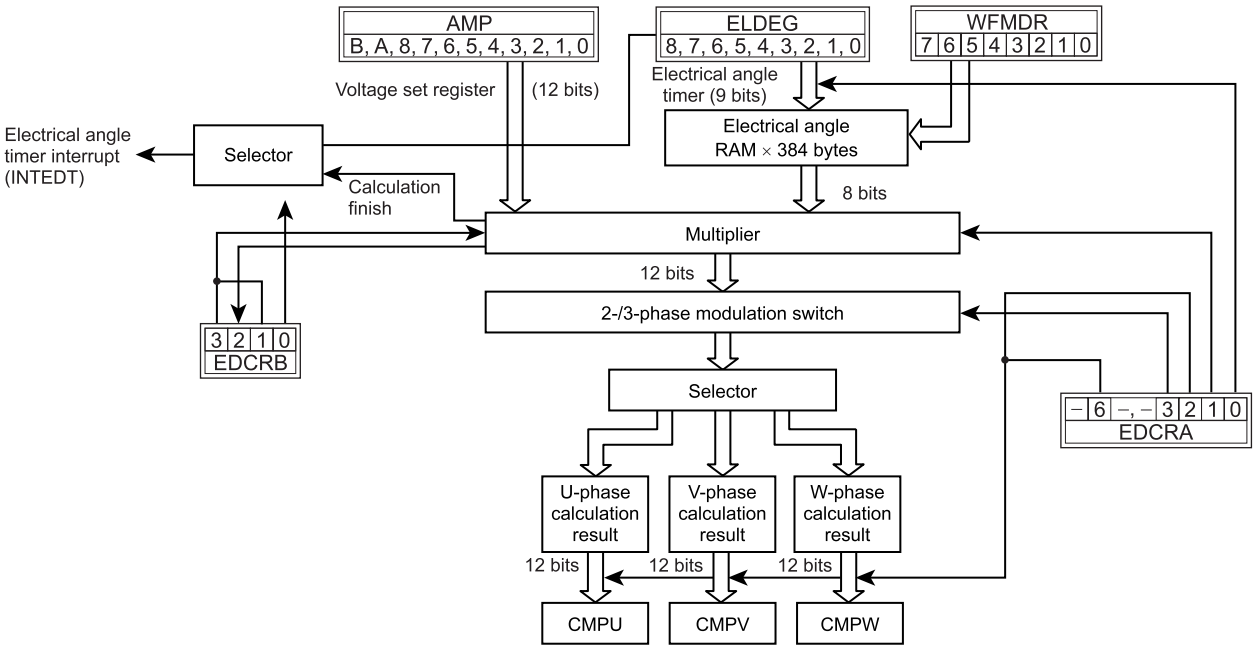


Figure 11-16 Waveform Arithmetic Circuit

11.6.1 Electrical Angle Timer and Waveform Arithmetic Circuit

The Electrical Angle Timer finishes counting upon reaching the value set by the Period Set Register (EDSET). The Electrical Angle Timer counts 360 degrees of electrical angle in the range of 0 to 383 (17FH) and is cleared to 0 upon reaching 383. In this way, it is possible to obtain the electrical angle of the frequency proportional to the value set by the Period Set Register. The period with which to count up can be corrected by using the Period Correction Register, allowing for fine adjustment of the frequency. The electrical angles counted by the Electrical Angle Timer are presented to the Waveform Arithmetic Circuit. An electrical angle timer interrupt signal is generated each time the Electrical Angle Timer finishes counting.

The Waveform Arithmetic Circuit has a sine wave data table, which is used to extract sine wave data based on the electrical angle data received from the Electrical Angle Timer. This sine wave data is multiplied by the value of the Voltage Amplitude Register. For 2-phase modulation, the product obtained by this multiplication is presented to the waveform synthesis circuit. For 3-phase modulation, waveform data is further calculated based on the product of multiplication and the electrical angle data and the value of the PWM Period Register. The calculation is performed each time the Electrical Angle Timer finishes counting or when a value is set in the Electrical Angle Register, and the calculation results consisting of the U phase, the V phase (+120 degrees), and the W phase (+240 degrees) are sequentially presented to the PWM waveform output circuit. The sine wave data table is stored in the RAM and requires initialization.

- To correct the period, set the number of times ‘n’ to be corrected in the Period Correction Register (EDSET Register F to C bits). The period is corrected by adding 1 to electrical angle counts 16 for ‘n’ times. For example, when a value 3 is set in the Period Correction Register, the period for 13 times out of electrical angle counts 16 is the value “mH” set in the Period Set Register, and that for 3 times is “m + 1H”. (Correction is made almost at equal intervals.)
- Because the electrical angle counter (ELDEG) can be accessed even while the Electrical Angle Timer is operating, the electrical angles can be corrected during operation.
- The Electrical Angle Capture EDCAP captures the electrical angle value from the Electrical Angle Counter at the time the position is detected.
- When the waveform calculation function is enabled, waveform calculation is performed each time the electrical angle counter (ELDEG) are accessed for write or the Electrical Angle Timer finishes counting.
- The calculation is performed in 35 machine cycle of execution time, or 7 μs (at 20 MHz).
- When transfer of calculation result to the CMP Registers is enabled (EDCRA<RWREN>), the calculation results are transferred to the CMPU to W Registers. (This applies only when the waveform calculation function is enabled with the EDCRA<CALCEN>.) The CMPU to W Registers are disabled against write while the transfer remains enabled. The calculation results can be read from the CMPU to W Registers while the waveform calculation function remains enabled.
- The calculated results can be modified and the modified data can be set in the CMPU to W Registers in software. This makes it possible to output any desired waveform other than sine waves.
  If a transfer (EDCRA register bit 2) of the calculated results to the CMP register is disabled, read-accessing the CMPU to W registers can read the calculated results. (Before read-accessing these registers, make sure that the calculation is completed.)
- To initialize the entire RAM data of the sine wave data table, set the addresses at which to set, sequentially from 000H to 17FH, in the ELDEG Register, and write waveform data to the WFMDR Register each time. Make sure the Waveform Arithmetic Circuit is disabled when writing this data.

Note 1: The value set in the Period Set Register (EDSET Register EDT bits) must be equal to or greater than 010H. Any value smaller than this is assumed to be 010H.

Note 2: The sine wave data that is read consists of the U phase, the V phase whose electrical angle is +120 degrees relative to the U phase, and the W phase whose electrical angle is +240 degrees relative to the U phase.

Note 3: If a period corresponding to an electrical angle of one degree is shorter than the required calculation time, the previously calculated results are used.

11.6.1.1 Functions of the Electrical Angle Timer and Waveform Arithmetic Circuit Registers

EDCRB

3	CALCST	Start calculation by software	Forcefully start calculation. When this bit is written while the waveform arithmetic circuit is calculating, the calculation is terminated and then newly started.
2	CALCBSY	Calculation flag	By reading this bit, the operation status of the waveform arithmetic circuit can be obtained.
1	EDCALEN	Enable/disable calculation start synchronized with electrical angle	Select whether to start calculation when the electrical angle timer finishes counting or when a value is set in the electrical angle register. When disabled, calculation is only started when CALCST is set to 1.
0	EDISEL	Electrical angle interrupt	Set the electrical angle interrupt signal request timing to either when the electrical angle timer finishes counting or upon end of calculation.

EDCRA

7	EDCNT	Electrical angle count up/down	Set whether the electrical angle timer counts up or down.
6	EDRV	Select V-, W-phase	Select phase direction of V-phase and W-phase in relation to U-phase.
5, 4	EDCK	Select clock	Select the clock for the electrical angle timer. This setting can be altered even while the electrical angle timer is operating.
3	C2PEN	Switch between 2-phase and 3-phase modulations	Select the modulation method with which to perform waveform calculation. Two-phase modulation DATA = ramdata (ELDEG) × AMP Three-phase modulation: $DATA = \frac{MOPRD}{2} \pm \frac{ramdata(ELDEG) \times AMP}{2}$ Note: The ± sign during 3-phase modulation changes depending on the electrical angle. + for electrical angles 0 to 179 degrees (191) – for electrical angles 180 (192) to 360 (383) degrees
2	RWREN	Auto transfer calculation results to CPM registers	Enable/disable transfer of calculation results by the waveform arithmetic circuit. When the waveform calculation function is enabled while at the same time transfer is enabled, calculation results are set as U, V, and W-phase duty cycles of the PWM generation circuit and are reflected in the ports.
1	CALCEN	Enable/disable waveform calculation function	Enable/disable the waveform calculation function. Calculations are performed by the waveform arithmetic circuit by enabling the waveform calculation function. When the waveform calculation function is enabled, the calculated results can be read from the U, V, and W-phase compare registers (CMPU, V, W) of the PWM generation circuit.
0	EDTEN	Electrical angle timer	Enable/disable the electrical angle timer. When enabled, the electrical angle timer starts counting; when disabled, the electrical angle timer stops counting and is cleared to 0.

EDSET

F to C	EDTH	Correct electrical angle period	Correct the period by adding 1 to electrical angle counts 16 for “n” times. The timer counts the electrical angle period set value “m”’for (16 – n) times and counts (m + 1) for “n” times
B to 0	EDT	Electrical angle period	Set the electrical angle period.

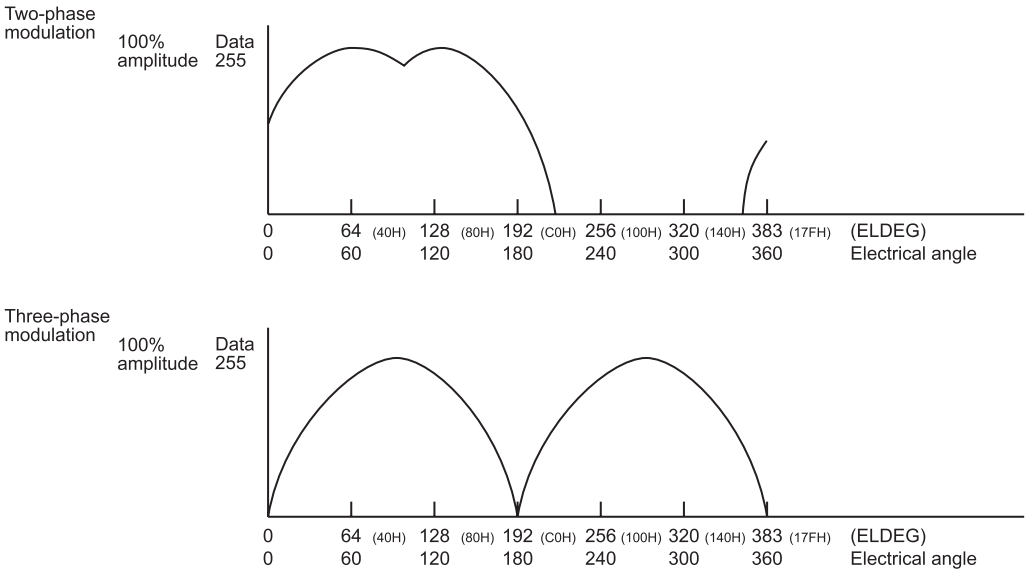
ELDEG	Electrical angle	Read the electrical angle. This register can also be set to initialize or correct the angle while counting. Any value greater than 17FH cannot be set.

AMP	Set voltage amplitude	Set the voltage amplitude. The waveform arithmetic circuit multiplies the data set here by the sine wave data read out from the sine wave RAM. The amplitude has its upper limit determined by the set value of the MDPRD register when performing this multiplication.

EDCAP	Capture electrical angle	Capture the value from the electrical angle timer when the position is detected.

WFMDR	Set sine wave data	To initialize the entire RAM data of the sine wave table, set the addresses at which to set, sequentially from 000H to 17FH, in the ELDEG register, and write waveform data to the WFMDR register each time. Make sure the waveform arithmetic circuit is disabled when writing this data.

Typical Settings of Sine Wave Data



Note: During 3-phase modulation, the sign changes at 180 degrees of electrical angle.

Figure 11-17 Typical Settings of Sine Wave Data

List of the Electrical Angle Timer and Waveform Arithmetic Circuit Registers [Addresses (PMD1)]

EDCRB (01FC1H)	7	6	5	4	3	2	1	0	
	–	–	–	–	CALCST	CALCBSY	EDCALEN	EDISEL	(Initial value: **** 0000)

3	CALCST	Start calculation by software	0: No operation 1: Start calculation	W
2	CALCBSY	Calculation flag	0: Waveform Arithmetic Circuit stopped 1: Waveform Arithmetic Circuit calculating	R
1	EDCALEN	Enable/disable calculation start synchronized with electrical angle	0: Start calculation insync with electrical angle 1: Do notcalculation insync with electrical angle	R/W
0	EDISEL	Electrical angle interrupt	0: Interrupt when the Electrical Angle Timer finishes counting 1: Interrupt upon end of calculation	

Note: Read-modify-write instructions, such as a bit manipulation instruction, cannot access the EDCRB register because this register is write only.

EDCRA (01FC0H)	7	6	5	4	3	2	1	0	
	EDCNT	EDRV	EDCK		C2PEN	RWREN	CALCEN	EDTEN	(Initial value: 0000 0000)

7	EDCNT	Electrical angle count up/down	0: Count up 1: Count down	R/W
6	EDRV	Select V-, W-phase	0: V = U + 120°, W = U + 240° 1: V = U − 120°, W = U − 240°	
5, 4	EDCK	Select clock	00: $fc/2^3$ (400 ns at 20 MHz) 01: $fc/2^4$ (800 ns at 20 MHz) 10: $fc/2^5$ (1.6 μs at 20 MHz) 11: $fc/2^6$ (3.2 μs at 20 MHz)	
3	C2PEN	Switch between 2-/3-phase modulations	0: 2-phase modulation 1: 3-phase modulation	
2	RWREN	Transfer calculation result to CMP registers	0: Disable 1: Enable	
1	CALC	Enable/disable waveform calculation function	0: Disable 1: Enable	
0	EDTEN	Electrical angleEnable/disable mode timer	0: Disable 1: Enable	

Note: When changing the EDCRA<EDCK> setting, keep the EDCRA<EDTEN> bit reset “0” (Disable electrical angle timer).

EDSET (01FC3H, 01FC2H)

F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
EDTH				EDT											

(Initial value: 00000000 00010000)

F to C	EDTH	Correct period (n)	0 to 15 times	R/W
B to 0	EDT	Set period (m)	≥ 010H	

One period of the Electrical Angle Timer, T, is expressed by the equation below.

$$T = \left(m + \frac{n}{16}\right) \times 384 \times \text{set clock}[s] \quad \text{where } m = \text{ set period, } n = \text{ period correction}$$

ELDEG (01FC5H, 01FC4H)

F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	D8	D7	D6	D5	D4	D3	D2	D1	D0

(Initial value: *******0 00000000)

8 to 0	ELDEG	Electrical angle	Set the Initially and the count values of electrical angle.	R/W

AMP (01FC7H, 01FC6H)

F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
–	–	–	–	DB	DA	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

(Initial value: ****0000 00000000)

B to 0	AMP	Set voltage	Set the voltage to be used during waveform calculation.	R/W

EDCAP (01FC9H, 01FC8H)

F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	D8	D7	D6	D5	D4	D3	D2	D1	D0

(Initial value: ******0 00000000)

8 to 0	EDCAP	Captured value of electrical angle	Electrical angle timer value when position is detected.	R

WFMDR (01FCAH)

7	6	5	4	3	2	1	0
D7	D6	D5	D4	D3	D2	D1	D0

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

7 to 0	WFMDR	Sine wave data	Write sine wave data to RAM of sine wave	W

Note: Read-modify-write instructions, such as a bit manipulation instruction, cannot access the WFMDR register because this register is write only.

11.6.1.2 List of PMD Related Control Registers

(1) Input/output Pins and Input/output Control Registers

PMD1 Input/Output Pins (P3, P4) and Port Input/Output Control Registers (P3CR, P4CR)

Name	Address	Bit	R or W	Description
P3DR	00003H	7	R/W	Overload protection ($\overline{CL1}$)
		6	R/W	EMG input ($\overline{EMG1}$)
		5 to 0	R/W	U1/V1/W1/X1/Y1/Z1 outputs.
P4DR	00004H	2 to 0	R/W	Position signal inputs (PDU1, PDV1, PDW1).
P3CR	01F89H	7 to 0	R/W	P3 port input/output control (can be set bitwise). 0: Input mode 1: Output mode
P4CR	01F8AH	2, 1, 0	R/W	P0 port input/output control (can be set bitwise). 0: Input mode 1: Output mode

Note: When using these pins as PMD function or input port, set the Output Latch (P*DR) to 1.

Example of the PMD Pin Port Setting

	Input/Output	P3DR	P3CR	P4DR	P4CR
CL1	Input	*	0	–	–
EMG1	Input	*	0	–	–
U1	Output	1	1	–	–
PDU1	Input	–	–	*	0

(2) Motor Control Circuit Control Registers
[Address : PMD1]

Position Detection Control Register (PDCR) and Sampling Delay Register (SDREG)

Name	Address	Bit	R or W	Description
PDCRC	01FA2H	5, 4	R	Detect the position-detected position. 00: Within the current pulse 01: When PWM is off 10: Within the current pulse 11: Within the preceding pulse
		3	R	Monitor the sampling status. 0: Sampling idle 1: Sampling in progress
		2 to 0	R	Holds the status of the position signal input during unmatch detection mode. Bits 2, 1, and 0: W, V, and U phases
PDCRB	01FA1H	7, 6	R/W	Select the sampling input clock [Hz]. 00: $fc/2^2$ 01: $fc/2^3$ 10: $fc/2^4$ 11: $fc/2^5$
		5, 4	R/W	Sampling mode. 00: When PWM is on 01: Regularly 10: When lower phases are turned on
		3 to 0	R/W	Detection position match counts 1 to 15.
PDCRA	01FA0H	7	W	0: No operation 1: Stop sampling in software
		6	W	0: No operation 1: Start sampling in software
		5	R/W	Stop sampling using Timer 3. 0: Disable 1: Enable
		4	R/W	Start sampling using Timer 2. 0: Disable 1: Enable
		3	R/W	Number of position signal input pins. 0: Compare three pins (PDU/PDV/PDW) 1: Compare one pin (PDU) only
		2	R/W	Count occurrences of matching when PWM is on. 0: Subsequent to matching counts when PWM previously was on 1: Eecount occurrences of matching each time PWM is on
		1	R/W	Position detection mode. 0: Ordinary mode 1: Unmatch detection mode
		0	R/W	Enable/Disable position detection function. 0: Disable 1: Enable (Sampling starts)
SDREG	01FA3H	6 to 0	R/W	Sampling delay. $2^3/fc \times n$ bits (n = 0 to 6, maximum 50.8 μs at 20 MHz).

Mode Timer Control Register (MTCR), Mode Capture Register (MCAP), and Compare Registers (CMP1, CMP2, CMP3)

Name	Address	Bit	R or W	Description
MTCRB	01FA5H	7	R/W	Debug output. 0: Disable 1: Enable (P67 for PMD1)
		5	R	Mode timer overflow. 0: No overflow 1: Overflowed occurred
		3	R/W	Capture mode timer by overload protection. 0: Disable 1: Enable
		2	W	Capture mode timer by software. 0: No operation 1: Capture
		1	R/W	Capture mode timer by position detection. 0: Disable 1: Enable
MTCRA	01FA4H	7, 6, 5	R/W	Select clock for mode timer [Hz]. 000: $fc/2^3$ (400 ns at 20 MHz) 010: $fc/2^4$ (800 ns at 20 MHz) 100: $fc/2^5$ (1.6 μs at 20 MHz) 110: $fc/2^6$ (3.2 μs at 20 MHz) 001: $fc/2^7$ (6.4 μs at 20 MHz) 011: Reserved 101: Reserved 111: Reserved
		4	R/W	Reset timer by Timer 3. 0: Disable 1: Enable
		3	R/W	Reset timer by overload protection. 0: Disable 1: Enable
		2	W	Reset timer by software. 0: No operation 1: Reset
		1	R/W	Reset timer by position detection. 0: Disable 1: Enable
		0	R/W	Enable/Disable mode timer. 0: Disable 1: Enable (timer starts)
MCAP	01FA7H, 01FA6H	F to 0	R	Mode capture register.
CMP1	01FA9H, 01FA8H	F to 0	R/W	Compare Register 1.
CMP2	01FABH, 01FAAH	F to 0	R/W	Compare Register 2.
CMP3	01FADH, 01FACH	F to 0	R/W	Compare Register 3.

PMD Control Register (MDCR), Dead Time Register (DTR), and PMD Output Register (MDOUT)

Name	Address	Bit	R or W	Description
MDCRB	01FAFH	1, 0	R/W	Select clock for PWM counter. 00: fc/2 (100 ns at 20 MHz) 01: fc/2^2 (200 ns at 20 MHz) 10: fc/2^3 (400 ns at 20 MHz) 11: fc/2^4 (800 ns at 20 MHz)
MDCRA	01FAEH	7	R/W	Select half-period interrupt 0: Interrupt every period as specified in PINT. 1: Interrupt every half-period only PINT=00.
		6	R/W	DUTY mode. 0: U phase in common 1: Three phases independent
		5	R/W	Upper-phase port polarity. 0: Active low 1: Active high
		4	R/W	Lower-phase port polarity. 0: Active low 1: Active high
		3, 2	R/W	Select PWM interrupt (trigger). 00: Interrupt once every period 01: Interrupt once 2 periods 10: Interrupt once 4 periods 11: Interrupt once 8 periods
		1	R/W	PWM mode. 0: PWM mode0 (edge: sawtooth wave) 1: PWM mode1 (center: triangular wave)
		0	R/W	Enable/disable waveform synthesis function. 0: Disable 1: Enable (waveform output)
DTR	01FBEH	5 to 0	R/W	Set dead time. 2^3/fc × 6bit (maximum 25.2 μs at 20 MHz).
MDOUT	01FB3H, 01FB2H	F	R	0: Count up 1: Count down
		E, D, C	R/W	Comparison register for position detection. 6: W 5: V 4: U
		B	R/W	Select PWM synchronization. 0: Asynchronous with PWM period 1: Synchronized
		A	R/W	W-phase PWM output. 0: H/L level output 1: PWM waveform output
		9	R/W	V-phase PWM output. 0: H/L level output 1: PWM waveform output
		8	R/W	U-phase PWM output. 0: H/L level output 1: PWM waveform output
		7, 6	R/W	Select port output synchronizing signal. 00: Asynchronous 01: Synchronized to position detection 10: Synchronized to Timer 1 11: Synchronized to Timer 2
		5, 4	R/W	Control W-phase output
		3, 2	R/W	Control V-phase output
		1, 0	R/W	Control U-phase output

PWM Counter (MDCNT), PMD Period Register (MDPRD), and PMD Compare Registers (CMPU, CMPV, CMPW)

Name	Address	Bit	R or W	Description
MDCNT	01FB5H, 01FB4H	B to 0	R	Read the PWM period counter value.
MDPRD	01FB7H, 01FB6H	B to 0	R/W	PWM period MDPRD ≥ 010H.
CMPU	01FB9H, 01FB8H	B to 0	R/W	Set U-phase PWM duty cycle.
CMPV	01FBBH, 01FBAH	B to 0	R/W	Set V-phase PWM duty cycle.
CMPW	01FBDH, 01FBCH	B to 0	R/W	Set W-phase PWM duty cycle.

EMG Disable Code Register (EMGREL) and EMG Control Register (EMGCR)

Name	Address	Bit	R or W	Description
EMGREL	01FBFH	7 to 0	W	Code input for disable EMG protection circuit. Can be disable by writing 5AH and then A5H.
EMGCRB	01FB1H	7	W	Return from overload protective state. 0: No operation 1: Return from protective state
		6	R/W	Condition for returning from overload protective state: Synchronized to PWM. 0: Disable 1: Enable
		5	R/W	Enable/Disable return from overload protective state by timer 1. 0: Disable 1: Enable
		4	R	Overload protective state. 0: No operation 1: Under protection
		3, 2	R/W	Select output disabled phases during overload protection. 00: No phases disabled against output 01: All phases disabled against output 10: PWM phases disabled against output 11: All upper/All lower phases disabled against output
		1	R/W	Stop PWM counter (MDCNT) during overload protection. 0: Do not stop 1: Stop
		0	R/W	Enable/Disable overload protective circuit. 0: Disable 1: Enable
EMGCRA	01FB0H	7 to 4	R/W	Overload protection sampling time. 2^2/fc × n (n = 1 to 15, at 20 MHz)
		2	R	EMG protective state. 0: No operation 1: Under protection
		1	W	Return from EMG protective state. 0: No operation 1: Return from protective state
		0	R/W	Enable/Disable fanction of the EMG protective circuit. 0: Disable 1: Enable (This circuit initially is enabled (= 1). To disable this circuit, make sure key code 5AH and A5H are written to the EMGREL1 Register before-hand.)

Electrical Angle Control Register (EDCR), Electrical Angle Period Register (EDSET), Electrical Angle Set Register (ELDEG), Voltage Set Register (AMP), and Electrical Angle Capture Register (EDCAP).

Name	Address	Bit	R or W	Description
EDCRB	01FC1H	3	W	0: No operation 1: Start calculation
		2	R	0: Waveform Arithmetic Circuit stopped 1: Waveform Arithmetic Circuit calculatin
		1	R/W	0: Start calculation insync with electrical angle 1: Do not calculation insync with electrical angle
		0	R/W	0: Interrupt when the Electrical Angle Timer finishes counting 1: Interrupt upon end of calculation
EDCRA	01FC0H	7	R/W	0: Count up 1: Count down
		6	R/W	0: V = U + 120°, W = U + 240° 1: V = U − 120°, W = U − 240°
		5, 4	R/W	Select clock. 00: $fc/2^3$ 01: $fc/2^4$ 10: $fc/2^5$ 11: $fc/2^6$
		3	R/W	Switch between 2/3-phase modulations. 0: Two-phase modulation 1: Three-phase modulation
		2	R/W	Transfer calculation result to CMP registers. 0: Disable 1: Enable
		1	R/W	Enable/disable waveform calculation function. 0: Disable 1: Enable
		0	R/W	Electrical angle timer. 0: Disable 1: Enable
EDSET	01FC3H, 01FC2H	F to C	R/W	Correct period (n) 0 to 15 times.
		B to 0	R/W	Set period (1/m counter) ≥ 010H
ELDEG	01FC5H, 01FC4H	8 to 0	R/W	Initially set and count values of electrical angle.
AMP	01FC7H, 01FC6H	B to 0	R/W	Set voltage used during waveform calculation.
EDCAP	01FC9H, 01FC8H	8 to 0	R	Electrical angle timer value when position is detected.
WFMDR	01FCAH	7 to 0	W	Set sine wave data.

12. Asynchronous Serial interface (UART)

The TMP88CH40NG has a asynchronous serial interface (UART) .
It can connect the peripheral circuits through TXD and RXD pin. TXD and RXD pin are also used as the general port. For TXD pin, the corresponding general port should be set output mode (Set its output control register to "1" after its output port latch to "1"). For RXD pin, should be set input mode.

This UART and SIO can not use simultaneously because their input/output ports are common.

12.1 Configuration

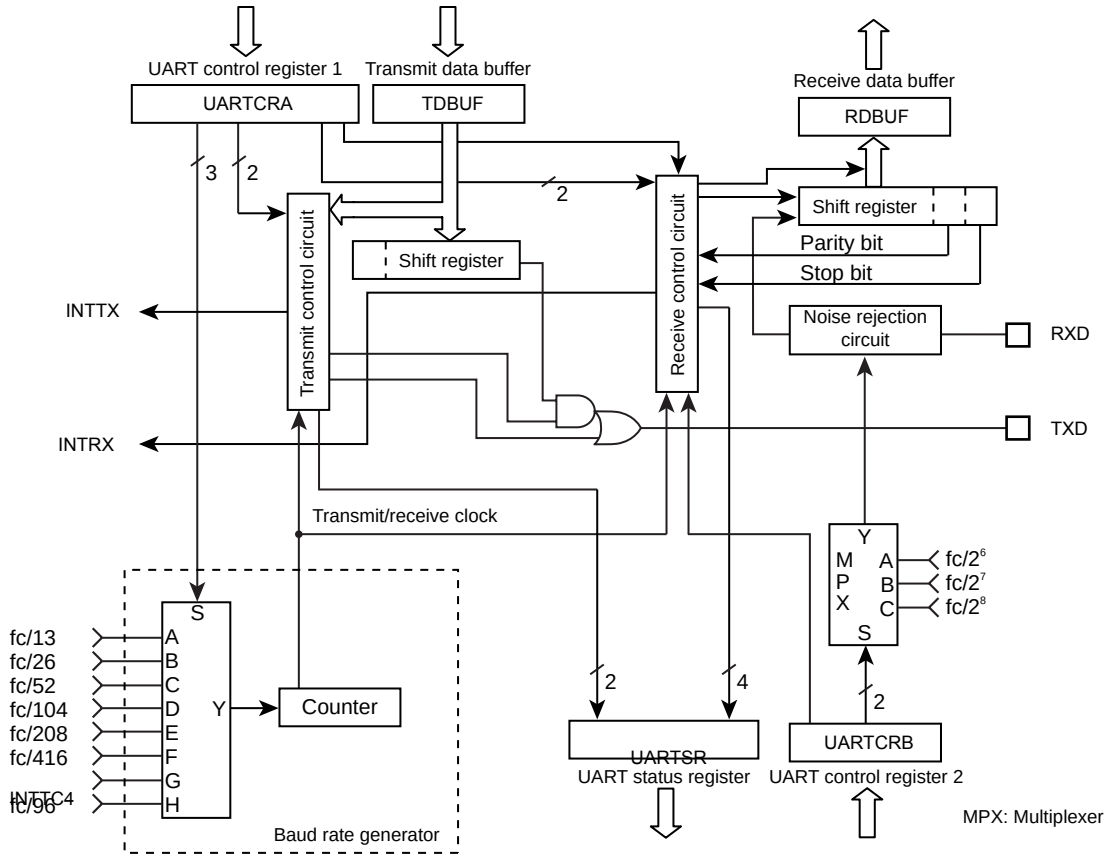


Figure 12-1 UART (Asynchronous Serial Interface)

12.2 Control

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

UART Control Register1

UARTCRA (01F91H)	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: Input INTTC4 111: fc/96	

Note 1: When operations are disabled by setting UARTCRA<TXE and RXE> bits 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: UARTCRA<RXE> and UARTCRA<TXE> should be set to “0” before UARTCRA<BRG> is changed.

Note 4: In case fc = 20MHz, the timer counter 4 (TC4) is available as a baud rate generator.

UART Control Register2

UARTCRB (01F92H)	7	6	5	4	3	2	1	0	
						RXDNC		STOPBR	(Initial value: **** *000)

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

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

UART Status Register

UARTSR (01F91H)

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 (01F93H)

7	6	5	4	3	2	1	0

Read only

(Initial value: 0000 0000)

UART Transmit Data Buffer

TDBUF (01F93H)

7	6	5	4	3	2	1	0

Write only

(Initial value: 0000 0000)

12.3 Transfer Data Format

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

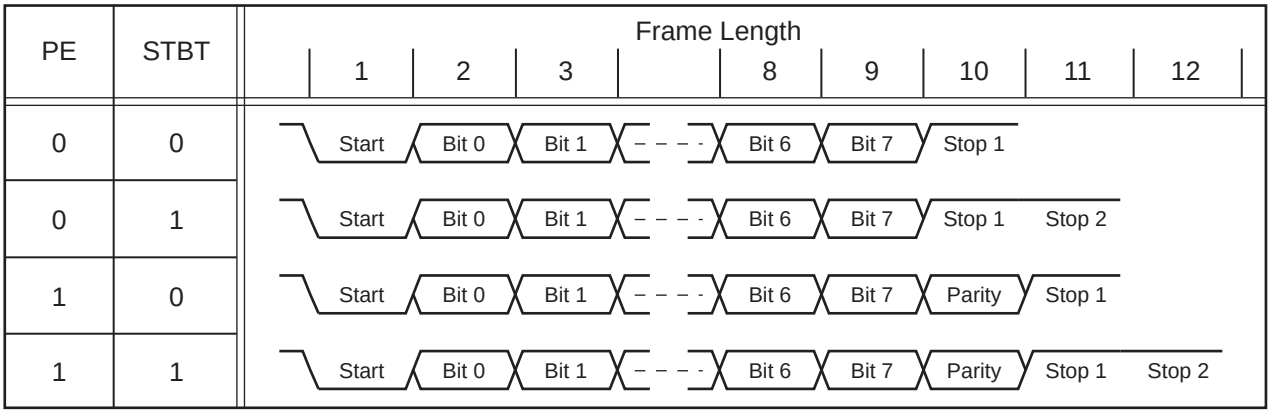


Figure 12-2 Transfer Data Format

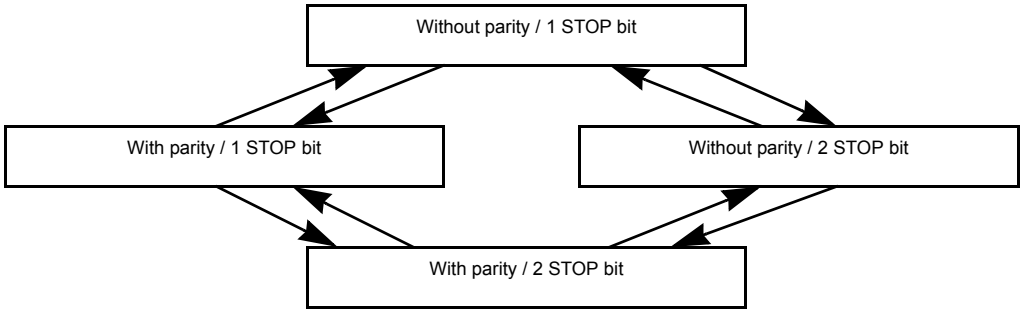


Figure 12-3 Caution on Changing Transfer Data Format

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

12.4 Transfer Rate

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

Table 12-1 Transfer Rate (Example)

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

When INTTC4 is used as the UART transfer rate (when UARTCRA<BRG> = “110”), the transfer clock and transfer rate are determined as follows:

Transfer clock [Hz] = TC4 source clock [Hz] / TC4DR setting value

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

12.5 Data Sampling Method

The UART receiver keeps sampling input using the clock selected by UARTCRA<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).

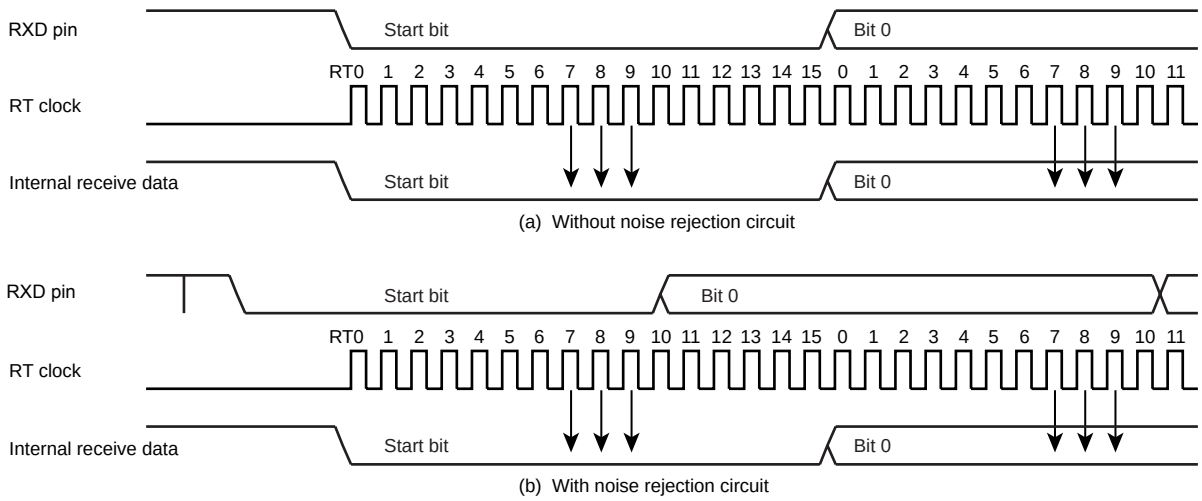


Figure 12-4 Data Sampling Method

12.6 STOP Bit Length

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

12.7 Parity

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

12.8 Transmit/Receive Operation

12.8.1 Data Transmit Operation

Set UARTCRA<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 UARTCRA<STBT> and a parity bit if parity addition is specified. Select the data transfer baud rate using UARTCRA<BRG>. When data transmit starts, transmit buffer empty flag UARTSR<TBEP> is set to “1” and an INTTXD interrupt is generated.

While UARTCRA<TXE> = “0” and from when “1” is written to UARTCRA<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.

12.8.2 Data Receive Operation

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

12.9 Status Flag

12.9.1 Parity Error

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

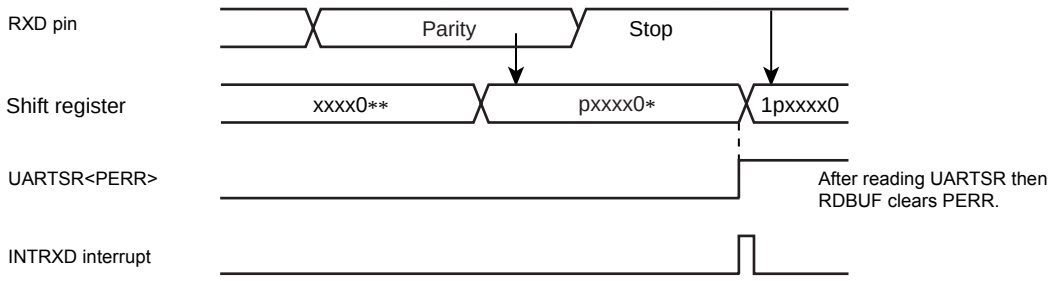


Figure 12-5 Generation of Parity Error

12.9.2 Framing Error

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

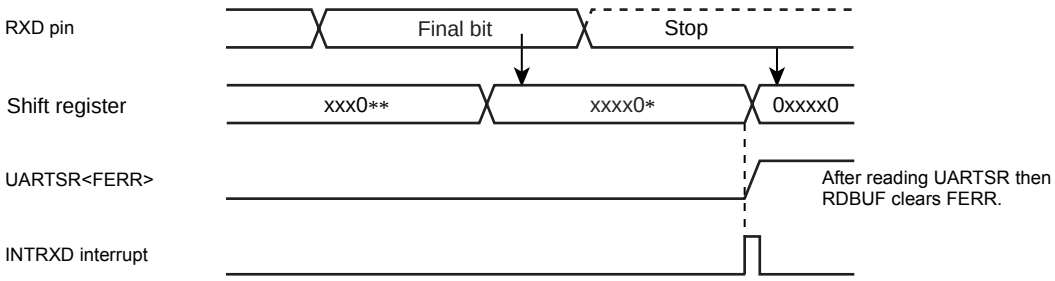


Figure 12-6 Generation of Framing Error

12.9.3 Overrun Error

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

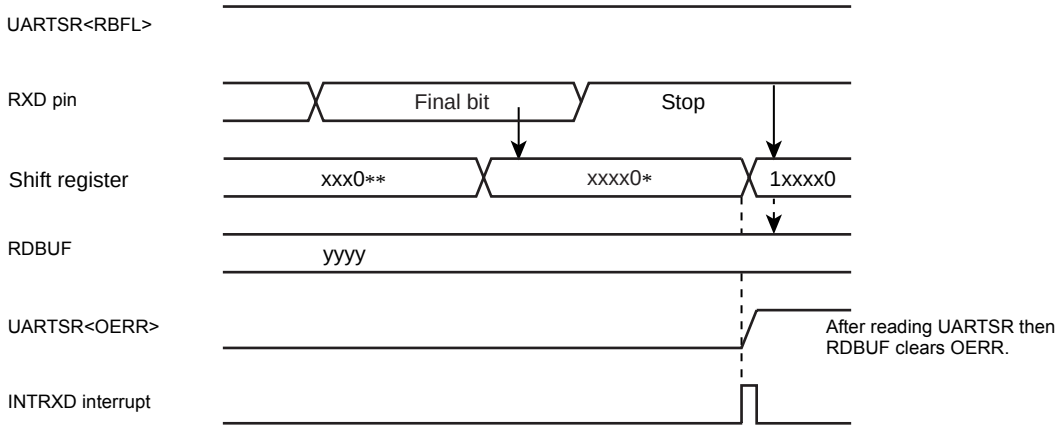


Figure 12-7 Generation of Overrun Error

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

12.9.4 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.

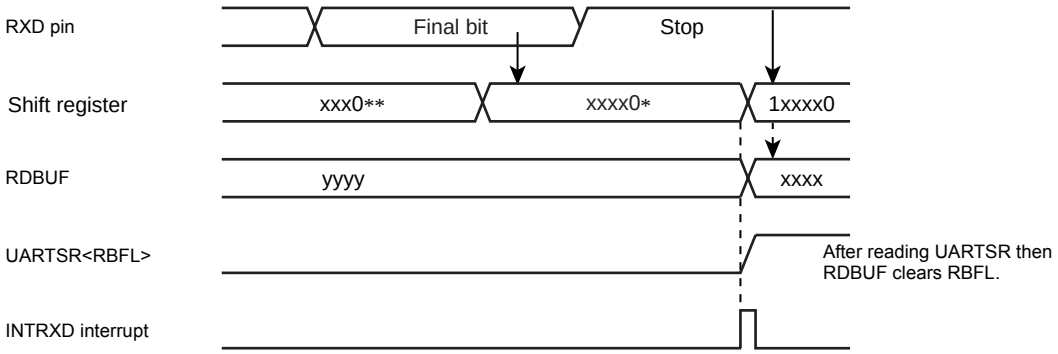


Figure 12-8 Generation of Receive Data Buffer Full

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.

12.9.5 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.

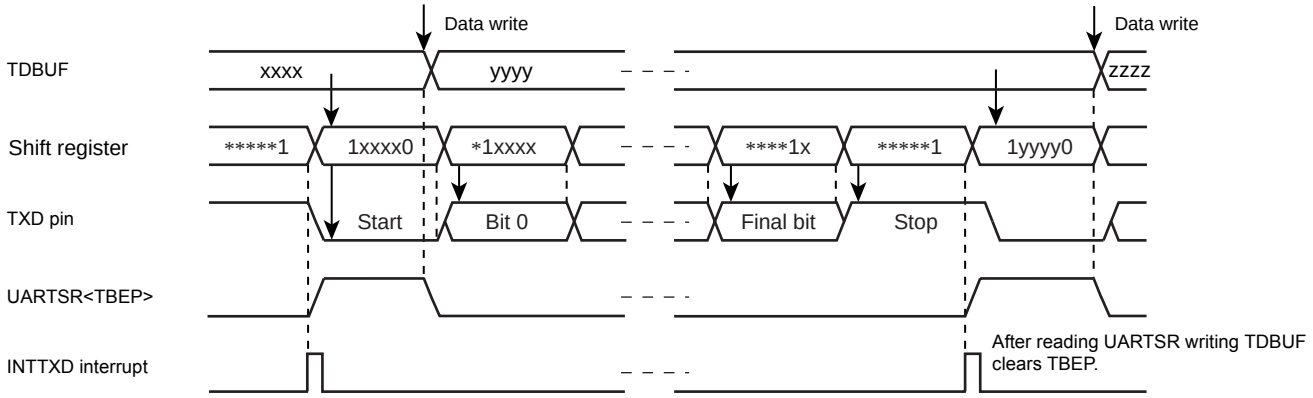


Figure 12-9 Generation of Transmit Data Buffer Empty

12.9.6 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 stated after writing the TDBUF.

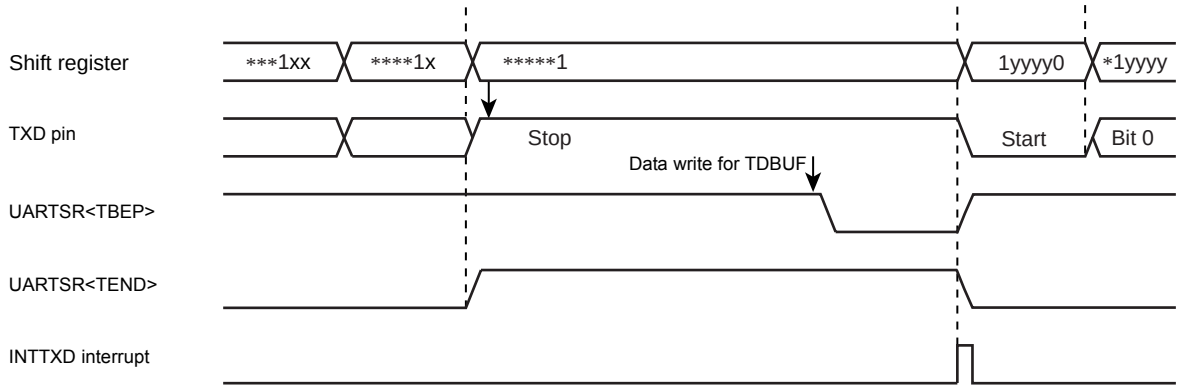


Figure 12-10 Generation of Transmit End Flag and Transmit Data Buffer Empty

13. Synchronous Serial Interface (SIO)

The TMP88CH40NG has a clocked-synchronous 8-bit serial interface. Serial interface has an 8-byte transmit and receive data buffer that can automatically and continuously transfer up to 64 bits of data.

Serial interface is connected to outside peripherl devices via SO, SI, SCK port.

This SIO and UART can not use simultaneously because their input/output ports are common.

13.1 Configuration

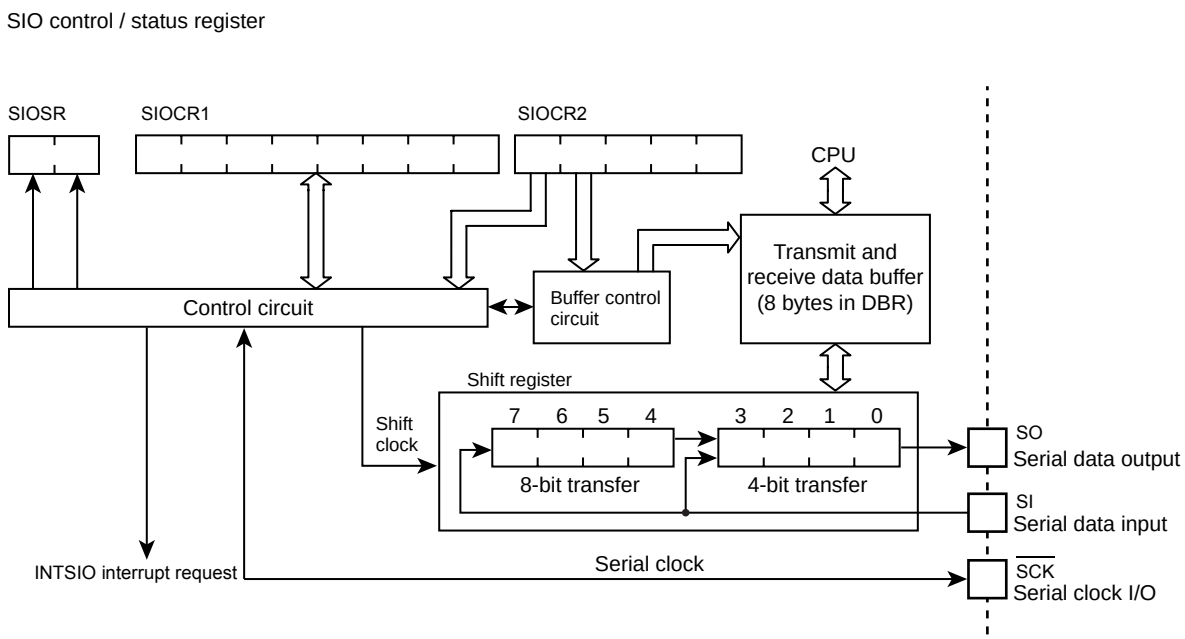


Figure 13-1 Serial Interface

13.2 Control

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

The transmit and receive data buffer is controlled by the SIOCR2<BUF>. The data buffer is assigned to address 01F98H to 01F9FH for SIO in the DBR area, and can continuously transfer up to 8 words (bytes or nibbles) at one time. When the specified number of words has been transferred, a buffer empty (in the transmit mode) or a buffer full (in the receive mode or transmit/receive mode) interrupt (INTSIO) is generated.

When the internal clock is used as the serial clock in the 8-bit receive mode and the 8-bit transmit/receive mode, a fixed interval wait can be applied to the serial clock for each word transferred. Four different wait times can be selected with SIOCR2<WAIT>.

SIO Control Register 1

SIOCR1	7	6	5	4	3	2	1	0	
(1F96H)	SIOS	SIOINH	SIOM			SCK			(Initial value: 0000 0000)

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

Note 1: fc; High-frequency clock [Hz]
Note 2: Set SIOCR1<SIOS> to "0" and SIOCR1<SIOINH> to "1" when setting the transfer mode or serial clock.
Note 3: SIOCR1 is write-only register, which cannot access any of in read-modify-write instruction such as bit operate, etc.

SIO Control Register 2

SIOCR2	7	6	5	4	3	2	1	0	
(1F97H)				WAIT		BUF			(Initial value: ***0 0000)

WAIT	Wait control	Always sets "00" except 8-bit transmit / receive mode. 00: $T_f = T_D$(Non wait) 01: $T_f = 2T_D$(Wait) 10: $T_f = 4T_D$(Wait) 11: $T_f = 8T_D$ (Wait)	Write only
BUF	Number of transfer words (Buffer address in use)	000: 1 word transfer 01F98H 001: 2 words transfer 01F98H ~ 01F99H 010: 3 words transfer 01F98H ~ 01F9AH 011: 4 words transfer 01F98H ~ 01F9BH 100: 5 words transfer 01F98H ~ 01F9CH 101: 6 words transfer 01F98H ~ 01F9DH 110: 7 words transfer 01F98H ~ 01F9EH 111: 8 words transfer 01F98H ~ 01F9FH	

Note 1: The lower 4 bits of each buffer are used during 4-bit transfers. Zeros (0) are stored to the upper 4bits when receiving.
Note 2: Transmitting starts at the lowest address. Received data are also stored starting from the lowest address to the highest address. (The first buffer address transmitted is 01F98H).
Note 3: The value to be loaded to BUF is held after transfer is completed.
Note 4: SIOCR2 must be set when the serial interface is stopped (SIOF = 0).
Note 5: *: Don't care
Note 6: SIOCR2 is write-only register, which cannot access any of in read-modify-write instruction such as bit operate, etc.
Note 7: T_f; Frame time, T_D; Data transfer time

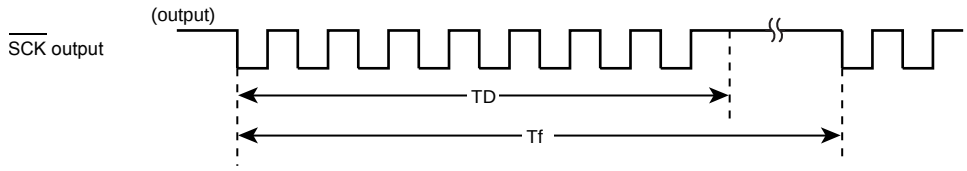


Figure 13-2 Frame time (T_f) and Data transfer time (T_D)

SIO Status Register

SIOSR	7	6	5	4	3	2	1	0	
(1F97H)	SIOF	SEF							(Initial value: 00** ****)

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

Note 1: After SIOCR1<SIOS> is cleared to "0", SIOSR<SIOF> is cleared to "0" at the termination of transfer or the setting of SIOCR1<SIOINH> to "1".

13.3 Serial clock

13.3.1 Clock source

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

13.3.1.1 Internal clock

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

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

Table 13-1 Serial Clock Rate

	NORMAL, IDLE mode	
SCK	Clock	Baud Rate
000	$fc/2^{13}$	2.44 Kbps
001	$fc/2^{8}$	78.13 Kbps
010	$fc/2^{7}$	156.25 Kbps
011	$fc/2^{6}$	312.50 Kbps
100	$fc/2^{5}$	625.00 Kbps
101	$fc/2^{4}$	125.00 Kbps
110	-	-
111	External	External

Note: 1 Kbit = 1024 bit (fc = 20 MHz)

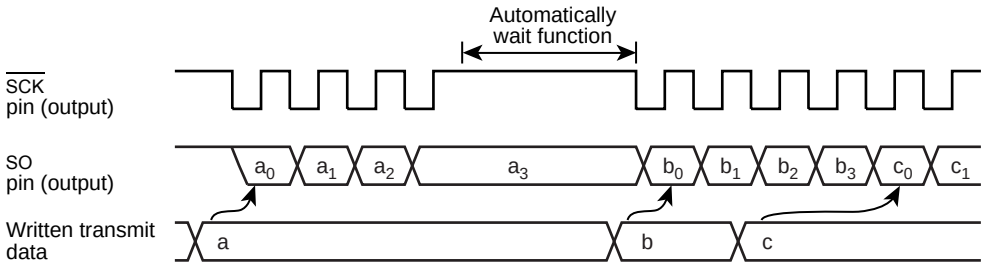


Figure 13-3 Automatic Wait Function (at 4-bit transmit mode)

13.3.1.2 External clock

An external clock connected to the $\overline{SCK}$ pin is used as the serial clock. In this case, the $\overline{SCK}$ (P43) port should be set to input mode. To ensure shifting, a pulse width of more than 2^4/fc is required. This pulse is needed for the shift operation to execute certainly. Actually, there is necessary processing time for interrupting, writing, and reading. The minimum pulse is determined by setting the mode and the program.

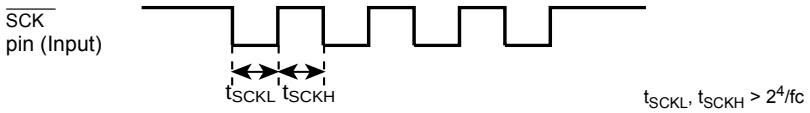


Figure 13-4 External clock pulse width

13.3.2 Shift edge

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

13.3.2.1 Leading edge

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

13.3.2.2 Trailing edge

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

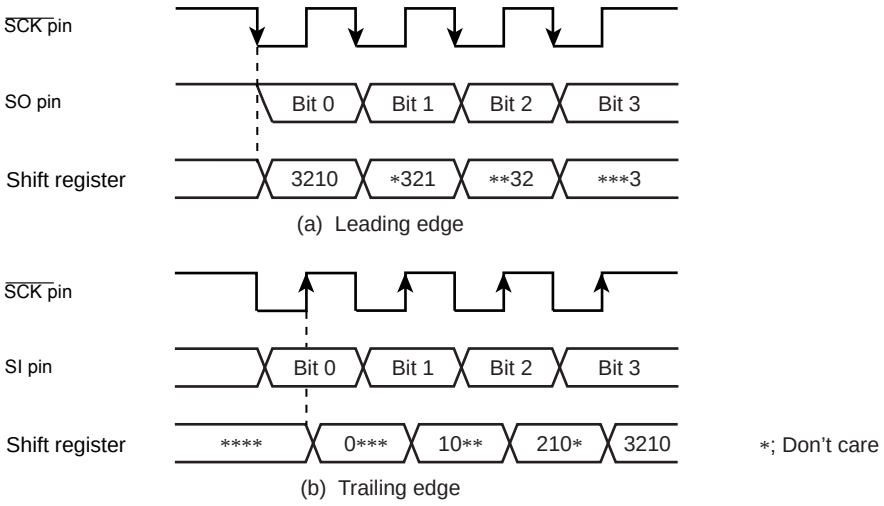


Figure 13-5 Shift edge

13.4 Number of bits to transfer

Either 4-bit or 8-bit serial transfer can be selected. When 4-bit serial transfer is selected, only the lower 4 bits of the transmit/receive data buffer register are used. The upper 4 bits are cleared to "0" when receiving.
The data is transferred in sequence starting at the least significant bit (LSB).

13.5 Number of words to transfer

Up to 8 words consisting of 4 bits of data (4-bit serial transfer) or 8 bits (8-bit serial transfer) of data can be transferred continuously. The number of words to be transferred can be selected by SIOCR2<BUF>.

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

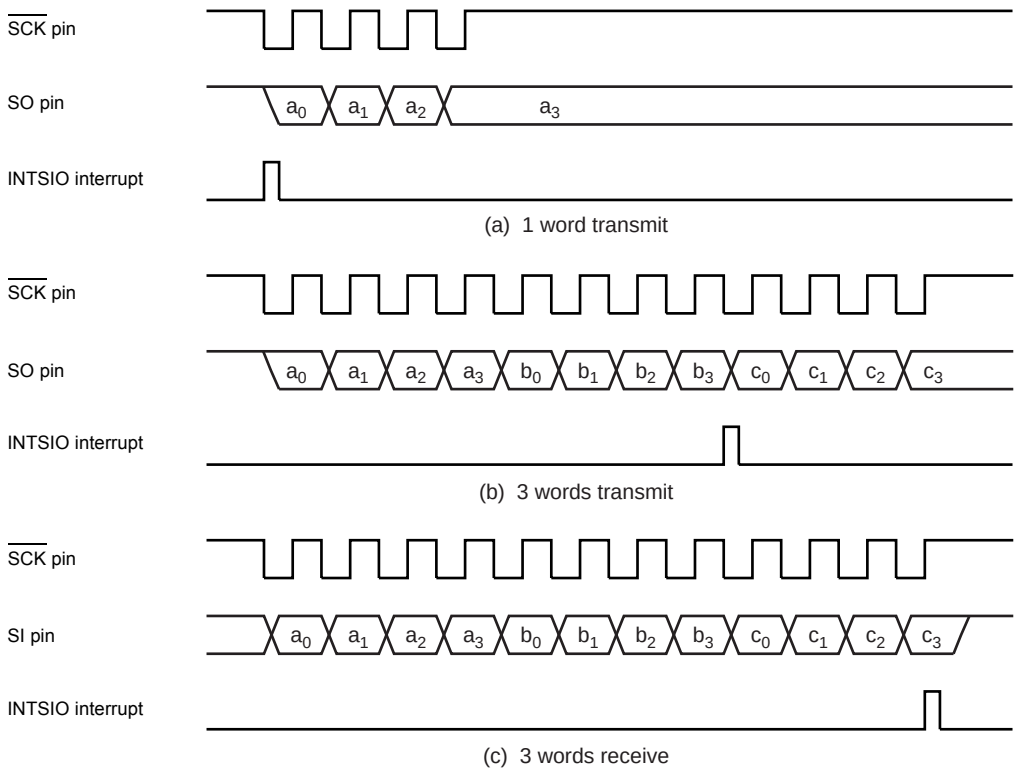


Figure 13-6 Number of words to transfer (Example: 1word = 4bit)

13.6 Transfer Mode

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

13.6.1 4-bit and 8-bit transfer modes

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

After the data are written, the transmission is started by setting SIOCR1<SIOS> to “1”. The data are then output sequentially to the SO pin in synchronous with the serial clock, starting with the least significant bit (LSB). As soon as the LSB has been output, the data are transferred from the data buffer register to the shift register. When the final data bit has been transferred and the data buffer register is empty, an INTSIO (Buffer empty) interrupt is generated to request the next transmitted data.

When the internal clock is used, the serial clock will stop and an automatic-wait will be initiated if the next transmitted data are not loaded to the data buffer register by the time the number of data words specified with the SIOCR2<BUF> has been transmitted. Writing even one word of data cancels the automatic-wait; therefore, when transmitting two or more words, always write the next word before transmission of the previous word is completed.

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

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

The transmission is ended by clearing SIOCR1<SIOS> to “0” or setting SIOCR1<SIOINH> to “1” in buffer empty interrupt service program.

SIOCR1<SIOS> is cleared, the operation will end after all bits of words are transmitted.

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

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

When an external clock is used, it is also necessary to clear SIOCR1<SIOS> to “0” before shifting the next data; If SIOCR1<SIOS> is not cleared before shift out, dummy data will be transmitted and the operation will end.

If it is necessary to change the number of words, SIOCR1<SIOS> should be cleared to “0”, then SIOCR2<BUF> must be rewritten after confirming that SIOSR<SIOF> has been cleared to “0”.

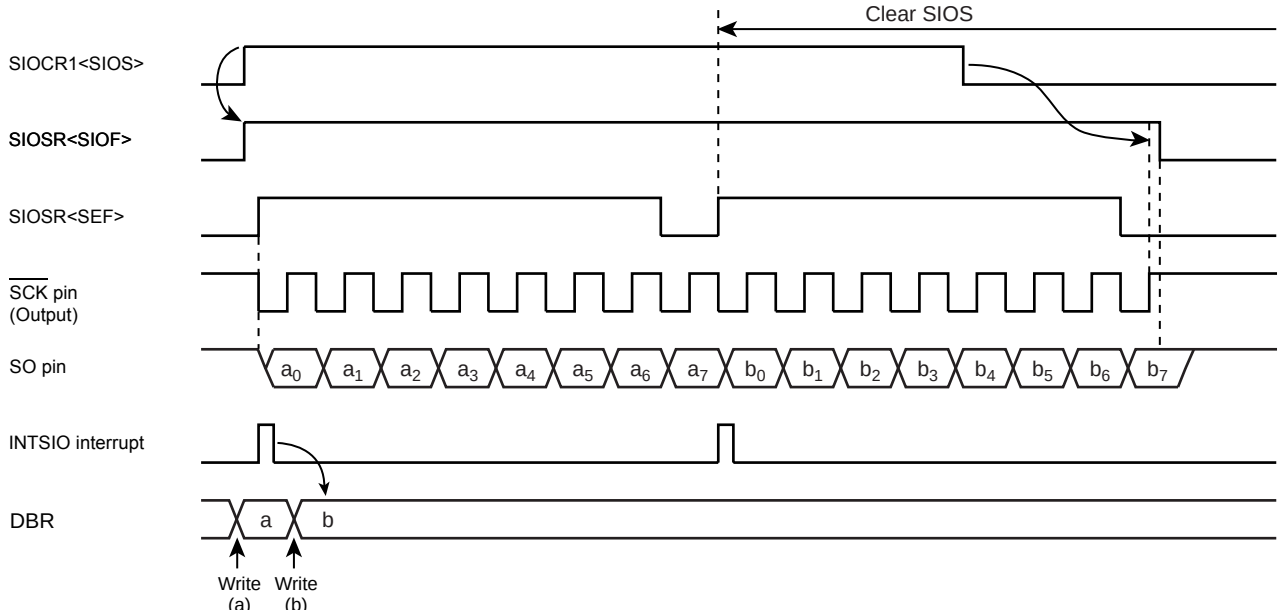


Figure 13-7 Transfer Mode (Example: 8bit, 1word transfer, Internal clock)

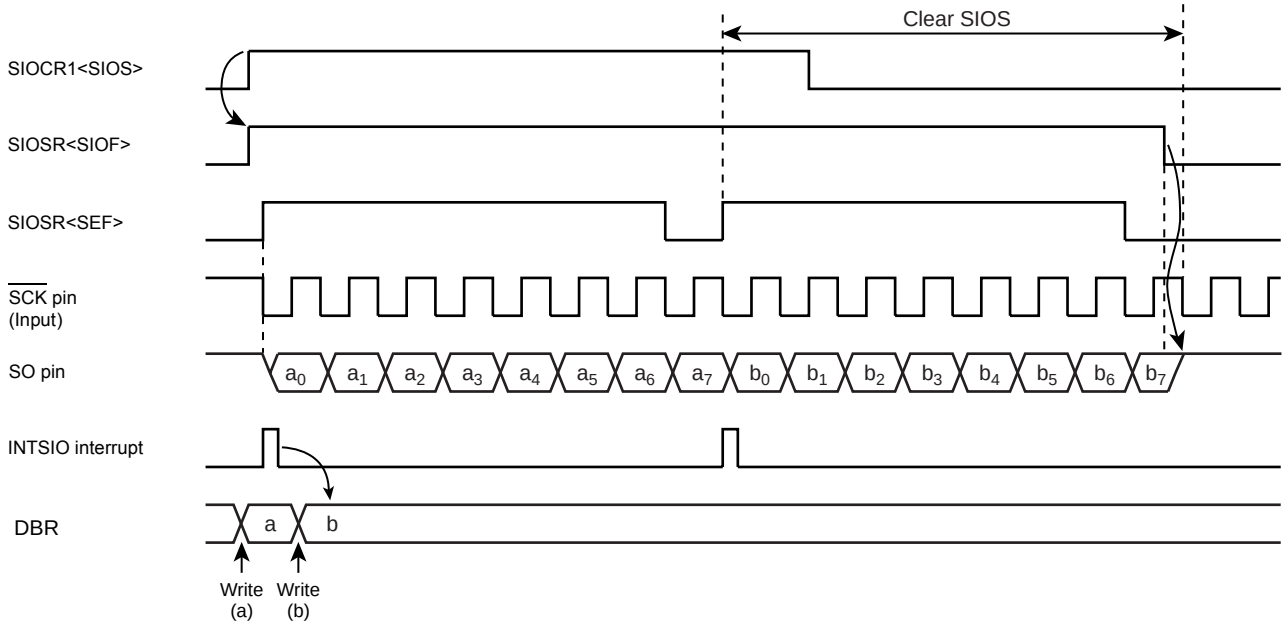


Figure 13-8 Transfer Mode (Example: 8bit, 1word transfer, External clock)

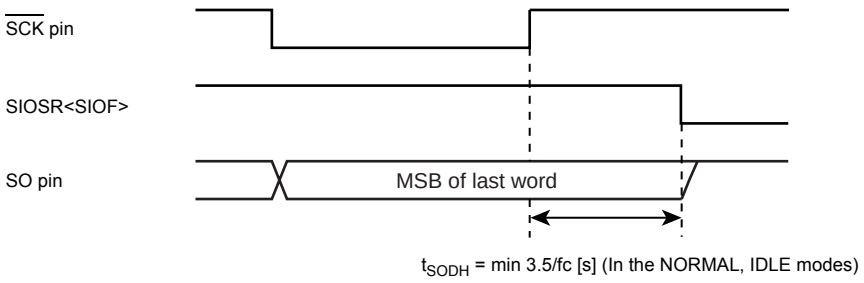


Figure 13-9 Transmiied Data Hold Time at End of Transfer

13.6.2 4-bit and 8-bit receive modes

After setting the control registers to the receive mode, set SIOCR1<SIOS> to "1" to enable receiving. The data are then transferred to the shift register via the SI pin in synchronous with the serial clock. When one word of data has been received, it is transferred from the shift register to the data buffer register (DBR). When the number of words specified with the SIOCR2<BUF> has been received, an INTSIO (Buffer full) interrupt is generated to request that these data be read out. The data are then read from the data buffer registers by the interrupt service program.

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

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

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

The receiving is ended by clearing SIOCR1<SIOS> to "0" or setting SIOCR1<SIOINH> to "1" in buffer full interrupt service program.
When SIOCR1<SIOS> is cleared, the current data are transferred to the buffer. After SIOCR1<SIOS> cleared, the receiving is ended at the time that the final bit of the data has been received. That the receiving has ended can be determined from the status of SIOSR<SIOF>. SIOSR<SIOF> is cleared to "0" when the receiving is ended. After confirmed the receiving termination, the final receiving data is read. When SIOCR1<SIOINH> is set, the receiving is immediately ended and SIOSR<SIOF> is cleared to "0". (The received data is ignored, and it is not required to be read out.)

If it is necessary to change the number of words in external clock operation, SIOCR1<SIOS> should be cleared to "0" then SIOCR2<BUF> must be rewritten after confirming that SIOSR<SIOF> has been cleared to "0". If it is necessary to change the number of words in internal clock, during automatic-wait operation which occurs after completion of data receiving, SIOCR2<BUF> must be rewritten before the received data is read out.

Note: The buffer contents are lost when the transfer mode is switched. If it should become necessary to switch the transfer mode, end receiving by clearing SIOCR1<SIOS> to "0", read the last data and then switch the transfer mode.

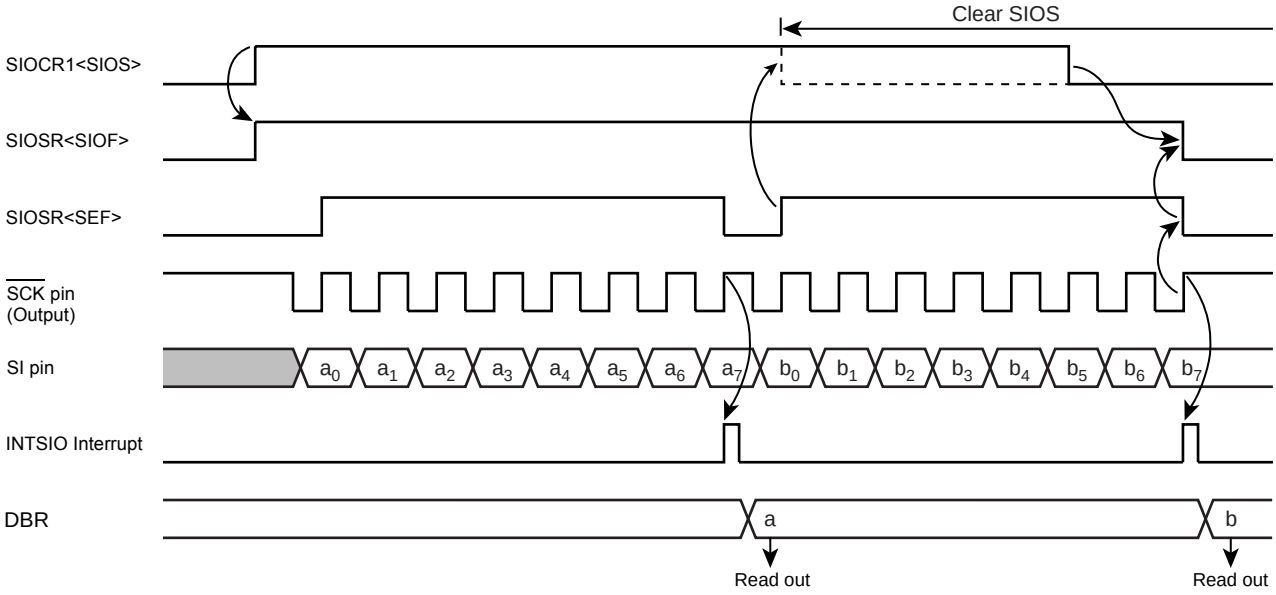


Figure 13-10 Receive Mode (Example: 8bit, 1word transfer, Internal clock)

13.6.3 8-bit transfer / receive mode

After setting the SIO control register to the 8-bit transmit/receive mode, write the data to be transmitted first to the data buffer registers (DBR). After that, enable the transmit/receive by setting SIOCR1<SIOS> to "1". When transmitting, the data are output from the SO pin at leading edges of the serial clock. When receiving, the data are input to the SI pin at the trailing edges of the serial clock. When the all receive is enabled, 8-bit data are transferred from the shift register to the data buffer register. An INTSIO interrupt is generated when the number of data words specified with the SIOCR2<BUF> has been transferred. Usually, read the receive data from the buffer register in the interrupt service. The data buffer register is used for both transmitting and receiving; therefore, always write the data to be transmitted after reading the all received data.

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

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

The transmit/receive operation is ended by clearing SIOCR1<SIOS> to "0" or setting SIOCR1<SIOINH> to "1" in INTSIO interrupt service program.

When SIOCR1<SIOS> is cleared, the current data are transferred to the buffer. After SIOCR1<SIOS> cleared, the transmitting/receiving is ended at the time that the final bit of the data has been transmitted.
That the transmitting/receiving has ended can be determined from the status of SIOSR<SIOF>. SIOSR<SIOF> is cleared to "0" when the transmitting/receiving is ended.

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

If it is necessary to change the number of words in external clock operation, SIOCR1<SIOS> should be cleared to "0", then SIOCR2<BUF> must be rewritten after confirming that SIOSR<SIOF> has been cleared to "0".

If it is necessary to change the number of words in internal clock, during automatic-wait operation which occurs after completion of transmit/receive operation, SIOCR2<BUF> must be rewritten before reading and writing of the receive/transmit data.

Note: The buffer contents are lost when the transfer mode is switched. If it should become necessary to switch the transfer mode, end receiving by clearing SIOCR1<SIOS> to “0”, read the last data and then switch the transfer mode.

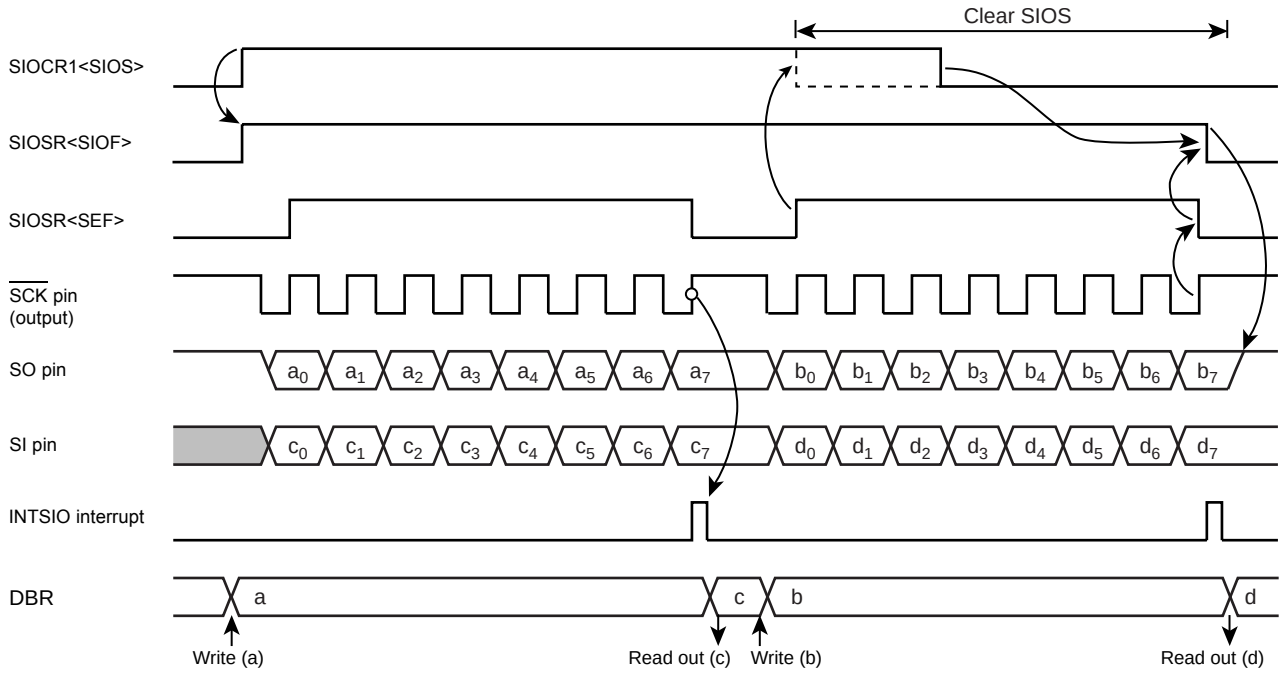


Figure 13-11 Transfer / Receive Mode (Example: 8bit, 1word transfer, Internal clock)

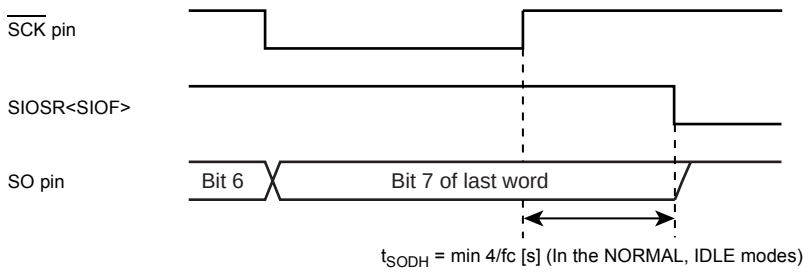


Figure 13-12 Transmitted Data Hold Time at End of Transfer / Receive

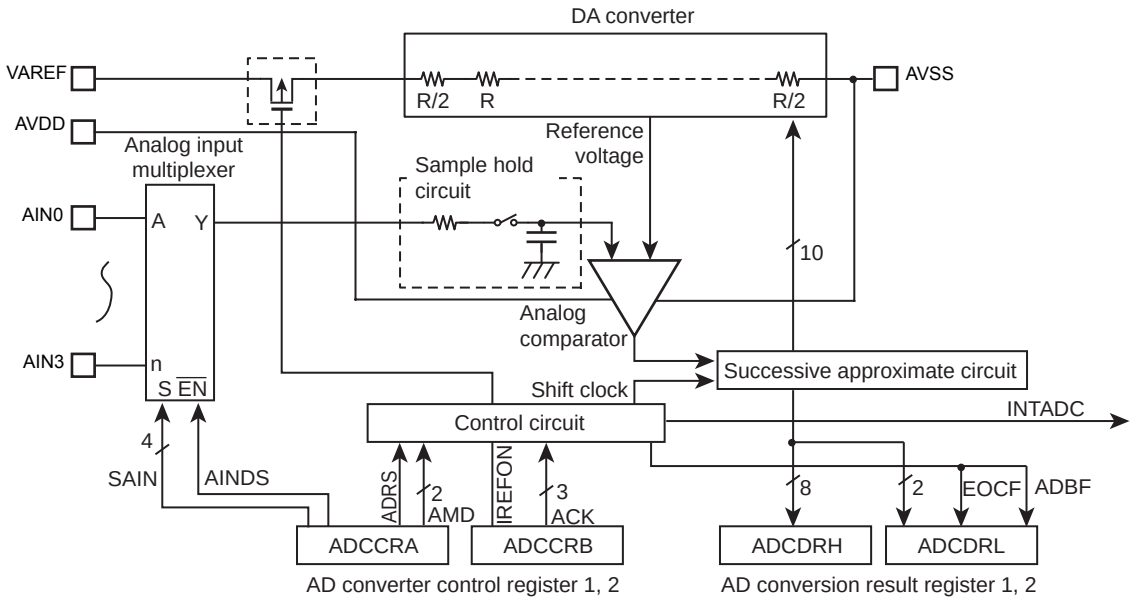
14. 10-bit AD Converter (ADC)

The TMP88CH40NG have a 10-bit successive approximation type AD converter.

14.1 Configuration

The circuit configuration of the 10-bit AD converter is shown in Figure 14-1.

It consists of control register ADCCRA and ADCCRB, converted value register ADCDRH and ADCDRL, 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 conbining a analog input port. For details, see the section on "I/O ports".

Figure 14-1 10-bit AD Converter

14.2 Register configuration

The AD converter consists of the following four registers:

1. AD converter control register 1 (ADCCRA)

   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 (ADCCRB)

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

3. AD converted value register 1 (ADCDRH)

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

4. AD converted value register 2 (ADCDRL)

   This register monitors the operating status of the AD converter.

AD Converter Control Register 1

ADCCRA (0026H)	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: Reserved 0101: Reserved 0110: Reserved 0111: Reserved 1000: Reserved 1001: Reserved 1010: Reserved 1011: Reserved 1100: Reserved 1101: Reserved 1110: Reserved 1111: Reserved	

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

Note 2: When the analog input channel is all use disabling, the ADCCRA<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 ADCCRA<ADRS> is automatically cleared to "0" after starting conversion.

Note 5: Do not set ADCCRA<ADRS> newly again during AD conversion. Before setting ADCCRA<ADRS> newly again, check ADCDRL<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 RESET, ADCCRA<SAIN> is initialized Reserved setting. Therfore, set the appropriate analog input channel to ADCCRA<SAIN> when use AD converter.

Note 7: After ADCCRA is set to 00H, AD conversion can not be started for four cycles. Thus, four NOPs must be inserted before setting the ADCCRA<ADRS>.

AD Converter Control Register 2

ADCCRB (0027H)	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 ADCCRB to "0" and set bit4 in ADCCRB to "1".
Note 2: When a read instruction for ADCCRB, bit6 to 7 in ADCCRB read in as undefined data.

Table 14-1 ACK setting and Conversion time (at CGCR<DV1CK>="0")

Condition / ACK	Conversion time	20 MHz	16 MHz	8 MHz
000	39/fc	-	-	-
001	Reserved			
010	78/fc	-	-	-
011	156/fc	-	-	19.5 μs
100	312/fc	15.6 μs	19.5 μs	39.0 μs
101	624/fc	31.2 μs	39.0 μs	78.0 μs
110	1248/fc	62.4 μs	78.0 μs	156.0 μs
111	Reserved			

Table 14-2 ACK setting and Conversion time (at CGCR<DV1CK>="1")

Condition / ACK	Conversion time	20 MHz	16 MHz	8 MHz
000	39/fc	-	-	-
001	Reserved			
010	78/fc	-	-	-
011	156/fc	-	-	19.5 μs
100	312/fc	15.6 μs	19.5 μs	39.0 μs
101	624/fc	31.2 μs	39.0 μs	78.0 μs
110	1248/fc	62.4 μs	78.0 μs	156.0 μs
111	Reserved			

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

- VAREF = 4.5 to 5.5 V 15.6 μs and more

AD Converted value Register 1

ADCDRH (0029H)	7	6	5	4	3	2	1	0	
	AD09	AD08	AD07	AD06	AD05	AD04	AD03	AD02	(Initial value: 0000 0000)

AD Converted value Register 2

ADCDRL (0028H)	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 ADCDRL<EOCF> is cleared to "0" when reading the ADCDRH. Therfore, the AD conversion result should be read to ADCDRL more first than ADCDRH.

Note 2: The ADCDRL<ADBF> is set to "1" when AD conversion starts, and cleared to "0" when AD conversion finished.

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

14.3 Function

14.3.1 Software Start Mode

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

After completion of the AD conversion, the conversion result is stored in AD converted value registers (ADCDRH, ADCDRL) and at the same time ADCDRL<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 ADCCRA<ADRS> newly again (Restart) during AD conversion. Before setting ADCCRA<ADRS> newly again, check ADCDRL<EOCF> to see that the conversion is completed or wait until the interrupt signal (INTADC) is generated (e.g., interrupt handling routine).

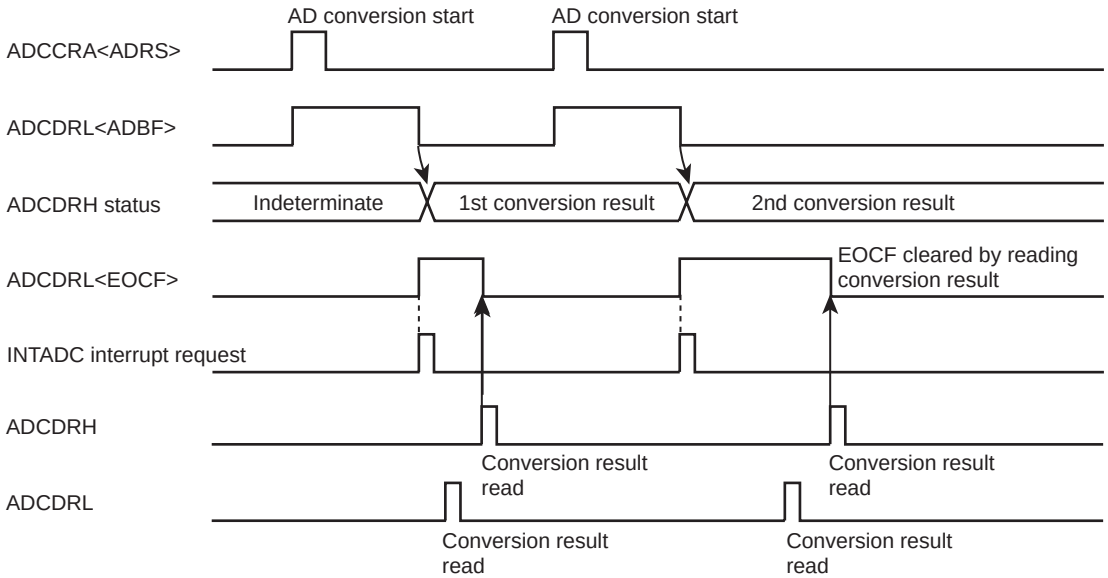


Figure 14-2 Software Start Mode

14.3.2 Repeat Mode

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

After completion of the AD conversion, the conversion result is stored in AD converted value registers (ADCDRH, ADCDRL) and at the same time ADCDRL<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 ADCCRA<AMD> to “00” (Disable mode) by writing 0s. The AD convert operation is stopped immediately. The converted value at this time is not stored in the AD converted value register.

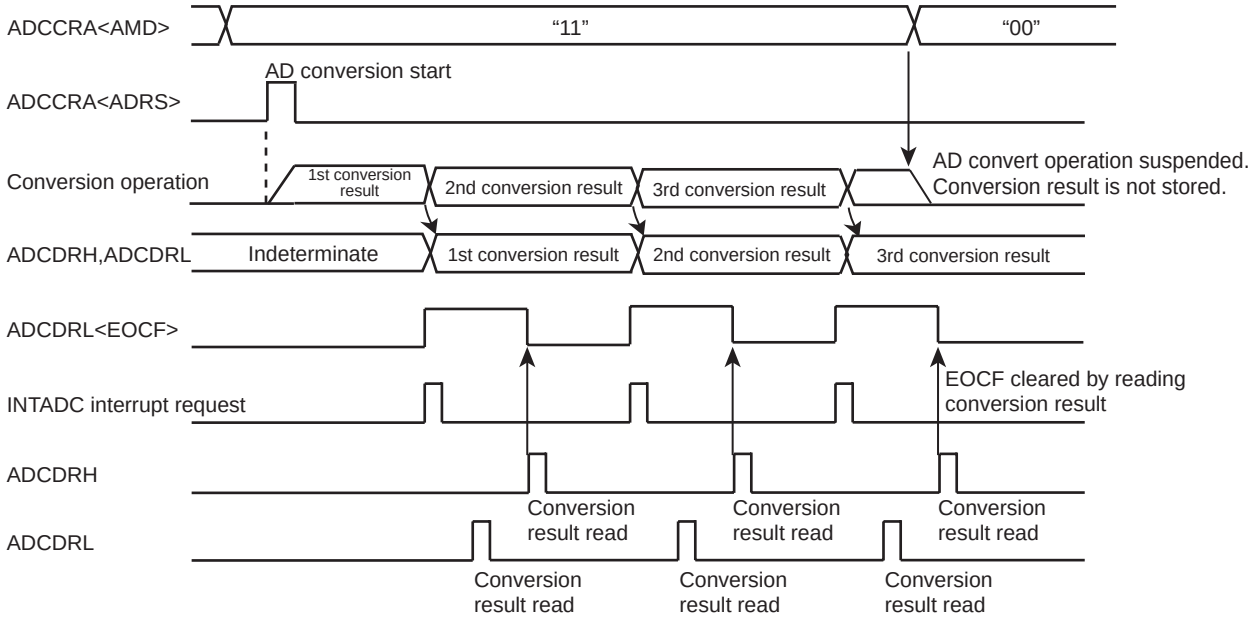


Figure 14-3 Repeat Mode

14.3.3 Register Setting

1. Set up the AD converter control register 1 (ADCCRA) 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 (ADCCRB) as follows:
   - Set the AD conversion time using AD conversion time (ACK). For details on how to set the conversion time, refer to Figure 14-1, Figure 14-2 and AD converter control register 2.
   - Choose IREFON for DA converter control.
3. After setting up (1) and (2) above, set AD conversion start (ADRS) of AD converter control register 1 (ADCCRA) 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 (ADCDRH) and the AD conversion finished flag (EOCF) of AD converted value register 2 (ADCDRL) is set to "1", upon which time AD conversion interrupt INTADC is generated.
5. EOCF is cleared to "0" by a read of the conversion result. However, if reconverted before a register read, although EOCF is cleared the previous conversion result is retained until the next conversion is completed.

Example :After selecting the conversion time 15.6 μs at 20 MHz and the analog input channel AIN4 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 approrriately before setting AD
                                                    converter registers.
        :                  :                        (Refer to section I/O port in details)
        LD                 (ADCCRA) ,  00100100B    ; Select Software start mode, Analog input enable,
                                                    and AIN4
        LD                 (ADCCRB) ,  00011000B    ;Select conversion time(312/fc) and operation
                                                    mode

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

        LD                 A , (ADCDRL)             ; Read result data
        LD                 (9EH) , A
        LD                 A , (ADCDRH)             ; Read  result data
        LD                 (9FH), A
```

14.4 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 14-4.

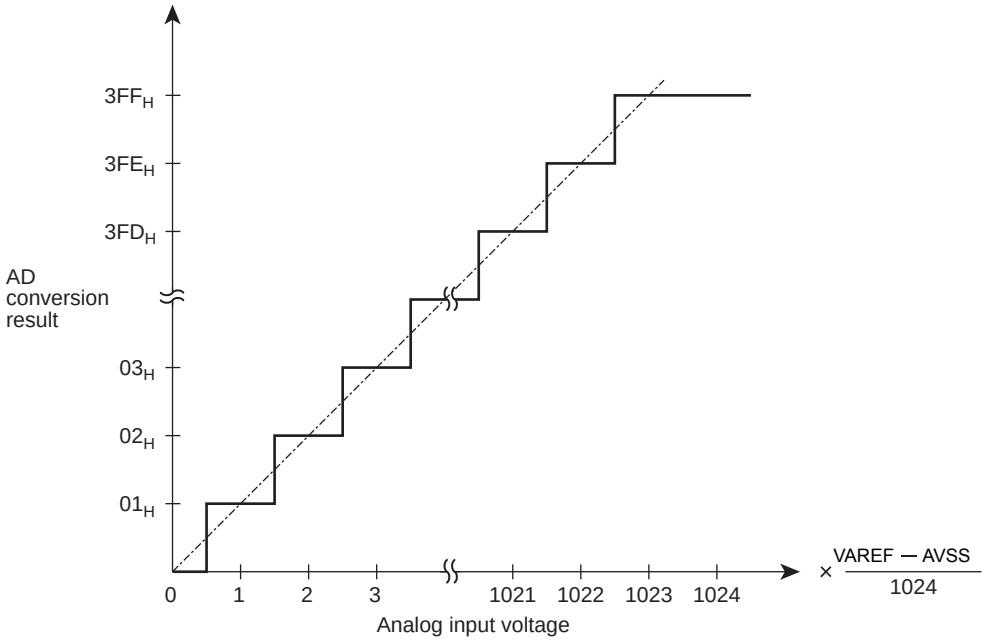


Figure 14-4 Analog Input Voltage and AD Conversion Result (Typ.)

14.5 Precautions about AD Converter

14.5.1 Analog input pin voltage range

Make sure the analog input pins (AIN0 to AIN3) 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.

14.5.2 Analog input shared pins

The analog input pins (AIN0 to AIN3) are shared with input/output ports. When using any of the analog inputs to execute AD conversion, do not execute input/output instructions for all other ports. This is necessary to prevent the accuracy of AD conversion from degrading. Not only these analog input shared pins, some other pins may also be affected by noise arising from input/output to and from adjacent pins.

14.5.3 Noise Countermeasure

The internal equivalent circuit of the analog input pins is shown in Figure 14-5. 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.

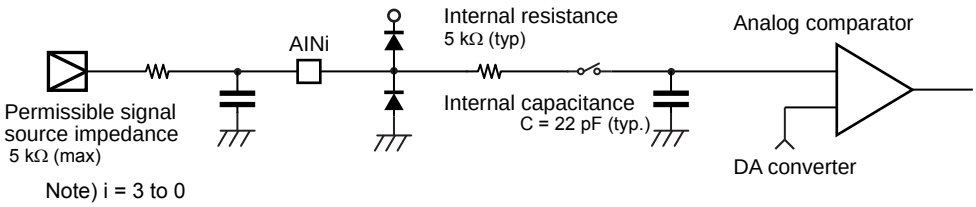
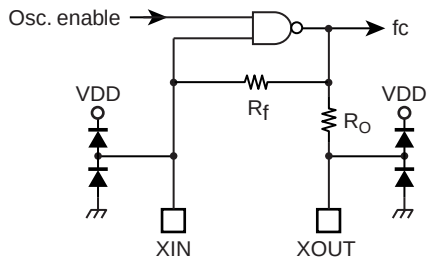
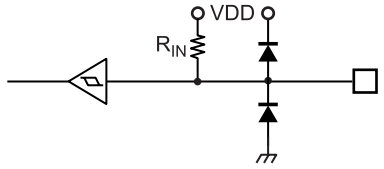
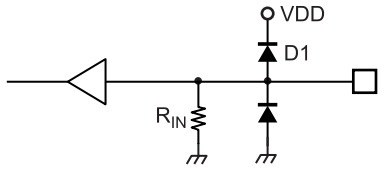


Figure 14-5 Analog Input Equivalent Circuit and Example of Input Pin Processing

15. Input/Output Circuitry

15.1 Control pins

The input/output circuitries of the TMP88CH40NG control pins are shown below.

Control Pin	I/O	Input/Output Circuitry	Remark
XIN XOUT	Input Output		High-frequency resonator connecting pins R_f = 1.2 MΩ (typ.) R_O = 0.5 kΩ (typ.)
$\overline{\text{RESET}}$	Input		Hysteresis input Pullup resistor included R_{IN} = 220 kΩ (typ.)
TEST	Input		Pull-down resistor included R_{IN} = 70 kΩ (typ.) Fix the TEST pin at "L" level in MCU mode.

Note: The TEST pin of TMP88PH40 does not have a pull-down resistor (R_{IN}) and protect diode (D1).
Fix the TEST pin at "L" level in MCU mode.

15.2 Input/output ports

Port	I/O	Input/output Circuit	Remark
P3 P4	I/O	Initial "High-Z" Data output Output control Disable Pin input	Tri-state output Programmable open-drain P3, P4: Large-current port Hysteresis input
P6	I/O	Initial "High-Z" Data output Disable Pin input	Tri-state output
P1	I/O	Initial "High-Z" Data output Disable Pin input	Tri-state output Hysteresis input

16. Electrical Characteristics

16.1 Absolute Maximum Ratings

The Absolute Maximum Ratings stipulate the standards, any parameter of which cannot be exceeded even in an instant. If the device is used under conditions exceeding the Absolute Maximum Ratings, it may break down or degrade, causing injury due to rupture or burning. Therefore, always make sure the Absolute Maximum Ratings will not be exceeded when designing your application equipment.

(V_{SS} = 0 V)

Parameter	Symbol	Pins	Standard	Unit	Remarks
Power supply voltage	V_{DD}		−0.3 to 6.5	V	
Input voltage	V_{IN}		−0.3 to V_{DD} + 0.3		
Output voltage	V_{OUT}		−0.3 to V_{DD} + 0.3		
Output current	I_{OH}	P1, P3, P4, P6	−1.8	mA	
	I_{OL1}	P1, P6	3.2		
	I_{OL2}	P3, P4	30		
Mean output current	ΣI_{OUT1}	P1, P6	16		Total of all ports except large-current ports
	ΣI_{OUT2}	P3	60		Total of 8 pins of large-current ports P30 to 37
	ΣI_{OUT3}	P4	60		Total of 6 pins of large-current ports P40 to 45
Power dissipation	P_D	TMP88CH40NG	300	mW	SDIP
Operating temperature	Topr		−40 to 85	°C	
Soldering temperature (time)	Tsld		260 (10 s)	°C	
Storage temperature	Tstg		−55 to 125	°C	

16.2 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.

(V_{SS} = 0 V, Topr = –40 to 85°C)

Parameter	Symbol	Pins	Condition		Min	Max	Unit
Power supply voltage	V_{DD}		fc = 20 MHz	NORMAL/IDLE	4.5	5.5	V
High level input voltage	V_{IH1}	Normal (P6)	$V_{DD} \geq 4.5$ V		$V_{DD} \times 0.70$	V_{DD}	V
	V_{IH2}	Hysteresis (P1, P3, P4, $\overline{RESET}$)			$V_{DD} \times 0.75$		
Low level input voltage	V_{IL1}	Normal (P6)	$V_{DD} \geq 4.5$ V		0	$V_{DD} \times 0.30$	V
	V_{IL2}	Hysteresis (P1, P3,P4, $\overline{RESET}$)				$V_{DD} \times 0.25$	
Clock frequency	fc	XIN, XOUT	V_{DD} = 4.5 V to 5.5 V		8	20	MHz

16.3 DC Characteristics

(V_{SS} = 0 V, Topr = –40 to 85°C)

Parameter	Symbol	Pins	Condition	Min	Typ.	Max	Unit
Input current	I_{IN1}	TEST	V_{DD} = 5.5 V, V_{IN} = 5.5 V/0 V	–	–	±2	μA
	I_{IN2}	Sink Open Drain, Tri-state					
	I_{IN3}	$\overline{RESET}$					
Input resistance	R_{IN1}	TEST		-	70	-	kΩ
	R_{IN2}	$\overline{RESET}$		90	220	510	
Output leakage current	I_{LO1}	Sink Open Drain	V_{DD} = 5.5 V, V_{IN} = 0.0 V	–	–	2	μA
	I_{LO2}	Tri-state port	V_{DD} = 5.5 V, V_{IN} = 5.5 V/0 V	–	–	±2	
High level output voltage	V_{OH}	Tri-state port	V_{DD} = 4.5 V, I_{OH} = –0.7 mA	4.1	–	–	V
Low level output voltage	I_{OL1}	P1, P6	V_{DD} = 4.5 V, V_{OL} = 0.4 V	1.6	–	–	mA
	I_{OL2}	P3, P4	V_{DD} = 4.5 V, V_{OL} = 1.0 V	–	20	–	
NORMAL mode power supply current	I_{DD}		V_{DD} = 5.5 V, V_{IN} = 5.3 V/0.2 V fc = 20 MHz	–	12	15	
IDLE mode power supply current				–	8	10	

Note 1: Typical values show those at Topr = 25°C, VDD = 5V.

Note 2: Input current (I_{IN1},I_{IN3}); The current through pull-up or pull-down resistor is not included.

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

16.4 AD Conversion Characteristics

(Topr = –40 to 85°C)

Parameter	Symbol	Condition	Min	Typ.	Max		Unit
					8 bit	10 bit	
Analog reference voltage	V_{AREF}	V_{SS} = 0 V, V_{DD} = AV_{DD}	V_{DD} –1.0	–	V_{DD}		V
Analog input voltage range	V_{AIN}		V_{ASS}	–	V_{AREF}		
Analog reference power supply current	I_{REF}	V_{DD} = AV_{DD} = V_{AREF} = 5.0 V V_{SS} = AV_{SS} = 0 V	–	0.5	1.0		mA
Nonlinearity error		V_{DD} = 5 V, V_{SS} = 0 V AV_{DD} = V_{AREF} = 5 V AV_{SS} = 0 V	–	–	±1	±2	LSB
Zero error			–	–	±1	±2	
Full scale error			–	–	±1	±2	
Overall error			–	–	±2	±4	

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

Note 2: Conversion time is different in recommended value by power supply voltage.
About conversion time, please refer to "Register Configuration" in the section of AD converter.

Note 3: Please use input voltage to AIN input pin in limit of V_{AREF} - V_{SS}.
When voltage or range outside is input, conversion value becomes unsettled and gives affect to other channel conversion value.

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

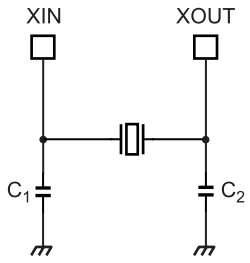
Note 5: When AD converter is not used, fix the AVDD and VAREF pin on the , V_{DD} level.

16.5 AC Characteristics

(V_{SS} = 0 V, V_{DD} = 4.5 to 5.5 V, Topr = –40 to 85°C)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Machine cycle time	tcy	During NORMAL mode During IDLE mode	0.2	–	0.5	μs
High level clock pulse width	t_{WCH}	When operating with external clock (XIN input) fc = 20 MHz	–	25	–	ns
Low level clock pulse width	t_{WCL}					

16.6 Recommended Oscillation Conditions



High-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: For the resonators to be used with Toshiba microcontrollers, we recommend ceramic resonators manufactured by Murata Manufacturing Co., Ltd.
For details, please visit the website of Murata at the following URL:
http://www.murata.com

16.7 Handling Precaution

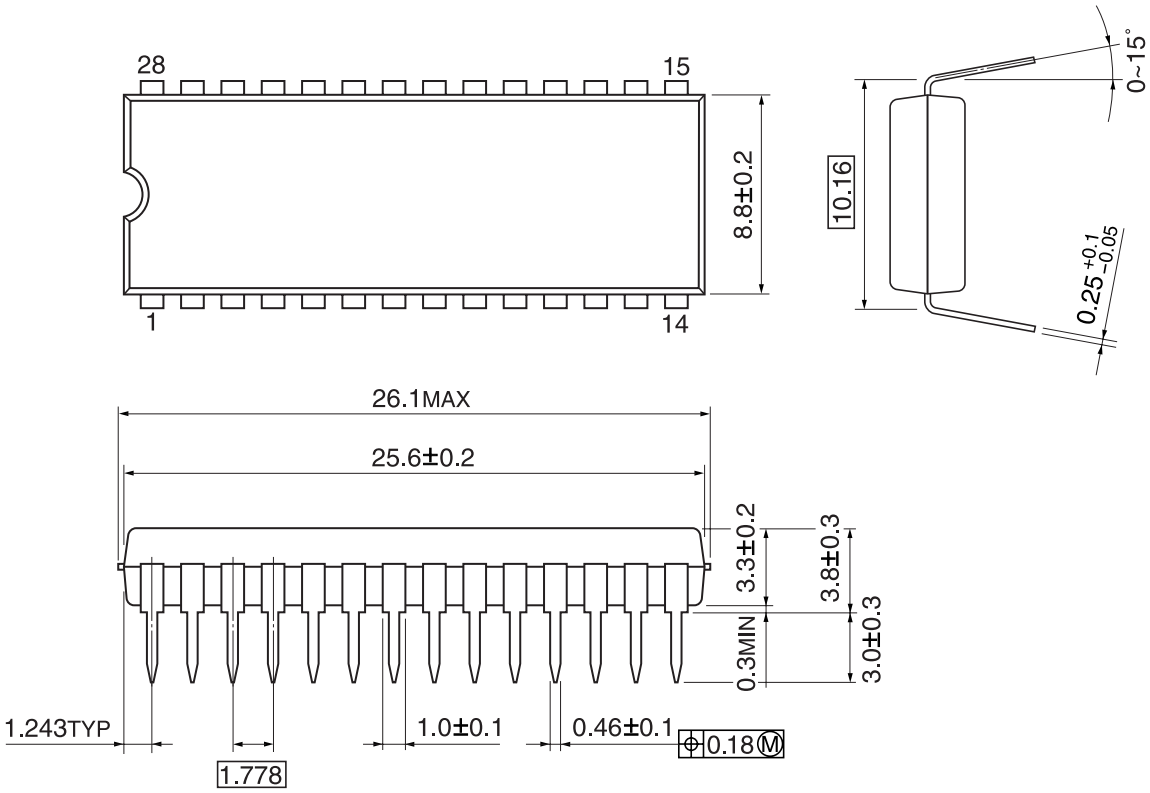
- The solderability test conditions for lead-free products (indicated by the suffix G in product name) are shown below.
  1. When using the Sn-37Pb solder bath
     Solder bath temperature = 230 °C
     Dipping time = 5 seconds
     Number of times = once
     R-type flux used
  2. When using the Sn-3.0Ag-0.5Cu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

  Note: The pass criteron 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.

17. Package Dimensions

SDIP28-P-400-1.78 Rev 01

Unit: mm



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

Toshiba provides a variety of development tools and basic software to enable efficient software development.

These development tools have specifications that support advances in microcomputer hardware (LSI) and can be used extensively. Both the hardware and software are supported continuously with version updates.

The recent advances in CMOS LSI production technology have been phenomenal and microcomputer systems for LSI design are constantly being improved. The products described in this document may also be revised in the future. Be sure to check the latest specifications before using.

Toshiba is developing highly integrated, high-performance microcomputers using advanced MOS production technology and especially well proven CMOS technology.

We are prepared to meet the requests for custom packaging for a variety of application areas.
We are confident that our products can satisfy your application needs now and in the future.