TOSHIBA

8 Bit Microcontroller
TLCS-870/X Series

TMP88F846UG

TOSHIBA CORPORATION

Revision History

Date	Revision	
2008/2/8	1	First Release
2009/9/3	2	Contents Revised

Table of Contents

18. Flash Memory

19. Serial PROM Mode

CMOS 8-Bit Microcontroller

TMP88F846UG

The TMP88F846UG is a single-chip 8-bit high-speed and high-functionality microcomputer incorporating 8192 bytes of Flash Memory. It is pin-compatible with the TMP88CH41UG (Mask ROM version). The TMP88F846UG can realize operations equivalent to those of the TMP88CH41UG by programming the on-chip Flash Memory.

Product No.	ROM (FLASH)	RAM	Package	MaskROM MCU
TMP88F846UG	8192 bytes	512+128 bytes	LQFP44-P-1010-0.80B	TMP88CH41UG

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. 26 interrupt sources (External : 6 Internal : 20)
3. Input / Output ports (33 pins)
   Large current output: 16pins (Typ. 20mA), LED direct drive
4. Power-on reset circuit
5. Voltage detection circuit(VLTD)
6. Prescaler
   - Time base timer
     Divider output function (DVO)
7. Watchdog Timer
   Select of "internal reset request" or "interrupt request".
8. 16-bit timer counter: 1 ch
   - Timer, External trigger, Window, Pulse width measurement,
     Event counter, Programmable pulse generate (PPG) modes
9. 16-bit timer/counter(CTC): 1ch
   - CTC:Timer,event counter or PPG (Programmable Pulse) output
10. 8-bit timer counter : 1 ch
    - Timer, Event counter, Capture modes
11. 8-bit timer counter : 1 ch
    - Timer, Event counter, Pulse width modulation (PWM) output,
      Programmable divider output (PDO) modes
12. Programmable motor driver (PMD) : 1 ch
    - Sine wave drive circuit (built-in sine wave data-table RAM)
      Rotor position detect function
      Motor control timer and capture function
      Overload protective function
      Auto commutation and auto position detection start function

13. 8-bit UART/SIO : 1 ch
14. 10-bit successive approximation type AD converter
    - Analog input: 8 ch
15. Clock oscillation circuit : 1 set
16. Low power consumption operation (2 modes)
    - STOP mode: Oscillation stops. (Battery/Capacitor back-up.)
    - IDLE mode: CPU stops.

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

    4.5 V to 5.5 V at 20MHz

1.2 Pin Assignment

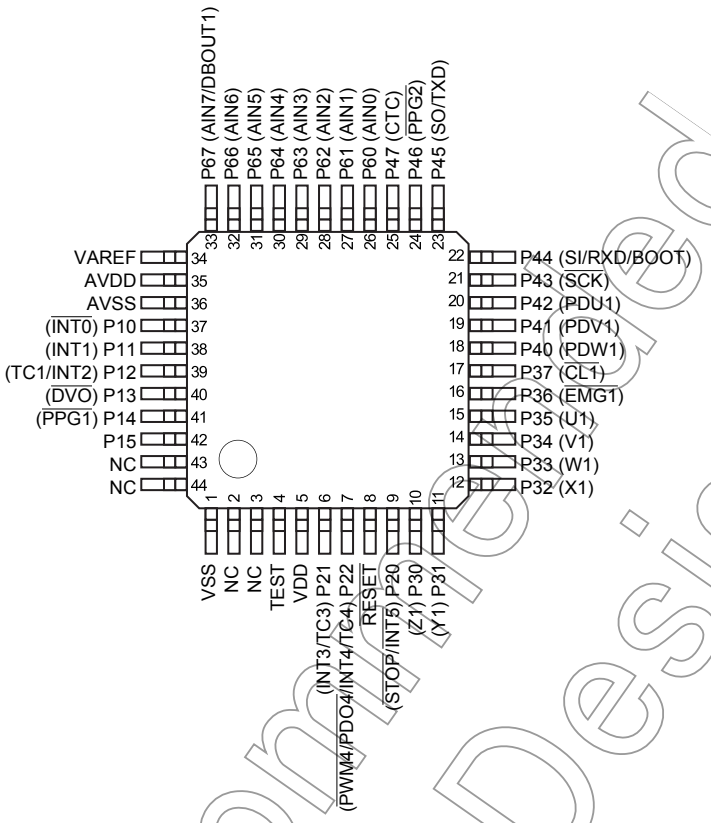


Figure 1-1 Pin Assignment

1.3 Block Diagram

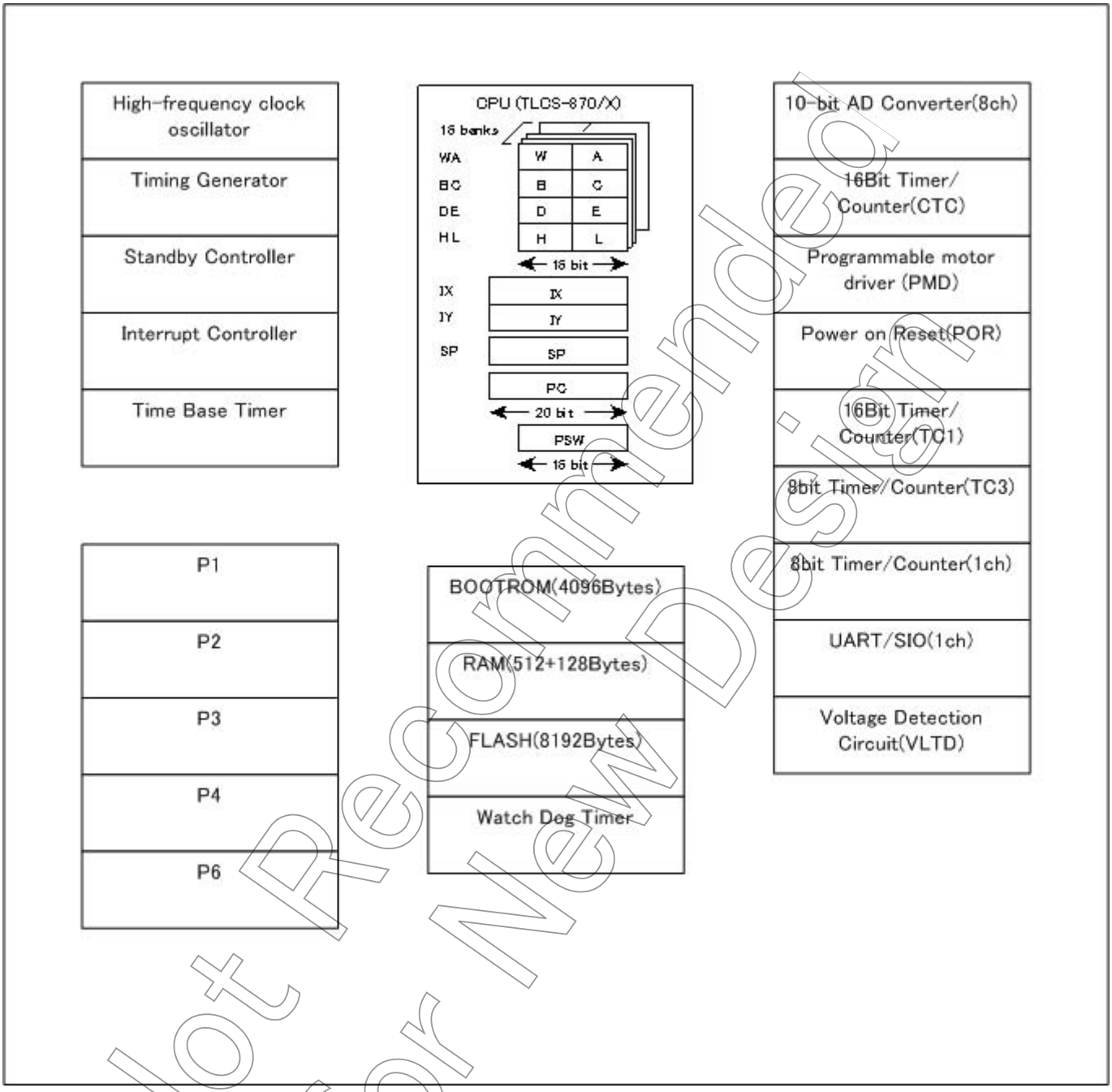


Figure 1-2 Block Diagram

1.4 Pin Names and Functions

The TMP88F846UG has MCU mode, parallel PROM mode, and serial PROM mode. Table 1-1 shows the pin functions in MCU mode. The serial PROM mode is explained later in a separate chapter.

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

Pin Name	Pin Number	Input/Output	Functions
P15	42	IO	PORT15
P14 $\overline{\text{PPG1}}$	41	IO O	PORT14 PPG1 output
P13 $\overline{\text{DVO}}$	40	IO O	PORT13 Divider Output
P12 INT2 TC1	39	IO I I	PORT12 External interrupt 2 input TC1 input
P11 INT1	38	IO I	PORT11 External interrupt 1 input
P10 $\overline{\text{INT0}}$	37	IO I	PORT10 External interrupt 0 input
P22 TC4 INT4 $\overline{\text{PWM4/PDO4}}$	7	IO I I O	PORT22 TC4 input External interrupt 4 input PWM4/PDO4 output
P21 TC3 INT3	6	IO I I	PORT21 TC3 pin input External interrupt 3 input
P20 $\overline{\text{INT5}}$ $\overline{\text{STOP}}$	9	IO I I	PORT20 External interrupt 5 input STOP mode release signal input
P37 $\overline{\text{CL1}}$	17	IO I	PORT37 PMD over load protection input1
P36 $\overline{\text{EMG1}}$	16	IO I	PORT36 PMD emergency stop input1
P35 U1	15	IO O	PORT35 PMD control output U1
P34 V1	14	IO O	PORT34 PMD control output V1
P33 W1	13	IO O	PORT33 PMD control output W1
P32 X1	12	IO O	PORT32 PMD control output X1
P31 Y1	11	IO O	PORT31 PMD control output Y1
P30 Z1	10	IO O	PORT30 PMD control output Z1

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

Pin Name	Pin Number	Input/Output	Functions
P47 CTC	25	IO I	PORT47 CTC input
P46 $\overline{PPG2}$	24	IO I	PORT46 PPG2 output
P45 SO TXD	23	IO O O	PORT45 Serial Data Output UART data output
P44 SI RXD BOOT	22	IO I I I	PORT44 Serial Data Input UART data input Serial PROM mode control input
P43 $\overline{SCK}$	21	IO IO	PORT43 Serial Clock I/O
P42 PDU1	20	IO I	PORT42 PMD control input U1
P41 PDV1	19	IO I	PORT41 PMD control input V1
P40 PDW1	18	IO I	PORT40 PMD control input W1
P67 AIN7 DBOUT1	33	IO I O	PORT67 Analog Input7 PMD debug output1
P66 AIN6	32	IO I	PORT66 Analog Input6
P65 AIN5	31	IO I	PORT65 Analog Input5
P64 AIN4	30	IO I	PORT64 Analog Input4
P63 AIN3	29	IO I	PORT63 Analog Input3
P62 AIN2	28	IO O	PORT62 Analog Input2
P61 AIN1	27	IO I	PORT61 Analog Input1
P60 AIN0	26	IO I	PORT60 Analog Input0
$\overline{RESET}$	20	IO	Reset signal
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	34	I	Analog Base Voltage Input Pin for A/D Conversion
AVDD	35	I	Analog Power Supply

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

Pin Name	Pin Number	Input/Output	Functions
AVSS	36	I	Analog Power Supply
VDD	24	IO	+5V
VSS	1	I	0(GND)
NC	43	I	Non Connection
NC01	44	I	Non Connection
NC02	2	I	Non Connection
NC03	3	I	Non Connection

2. Functional Description

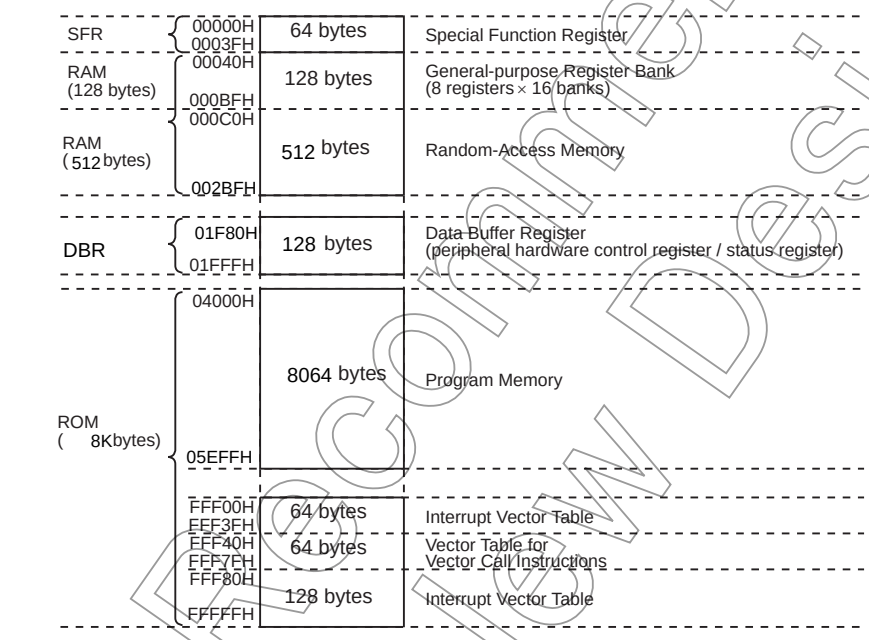
2.1 Functions of the CPU Core

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 TMP88F846UG.

2.1.1 Memory Address Map

The memory of the TMP88F846UG 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 TMP88F846UG.



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 TMP88F846UG contains 8Kbytes program memory (Flash) located at addresses 04000H to 05EFFH and addresses FFF00H to FFFFFH.

2.1.3 Data Memory (RAM)

The TMP88F846UG 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 TMP88F846UG (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 an On-Chip Oscillator, timing generator, and standby control circuit.

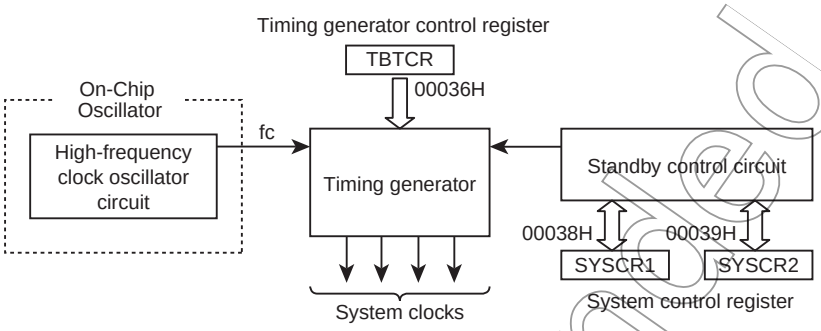


Figure 2-2 System Clock Control Circuit

2.1.4.1 On-Chip Oscillator

The On-Chip Oscillator the fundamental clock which serves as the reference for the system clocks supplied to the CPU core and peripheral hardware units.

The frequency clock (frequency fc) be fixed at 20MHz, and it is changed by temperature or many years past.

Measurement of 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 measured 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 a divider output ($\overline{DVO}$) pulse
2. Generate the source clock for the time base timer
3. Generate the source clock for the watchdog timer
4. Generate the internal source clock for the timer counter
5. Generate a warm-up clock when exiting STOP mode

(1) Configuration of the Timing Generator

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

When reset and when entering/exiting STOP mode, the prescaler and dividers are cleared to 0.

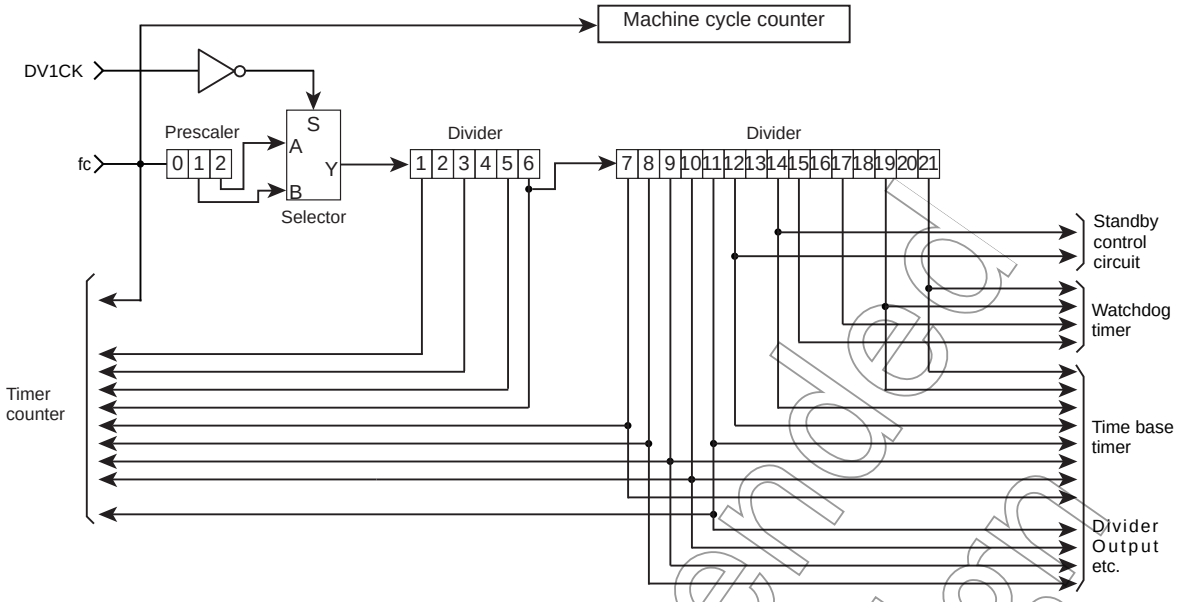


Figure 2-3 Configuration of the Timing Generator

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.

Timing Generator Control Register

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

Note 1: Be sure to write "0" to TBTCR Register bit 4.

(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 "machine 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.

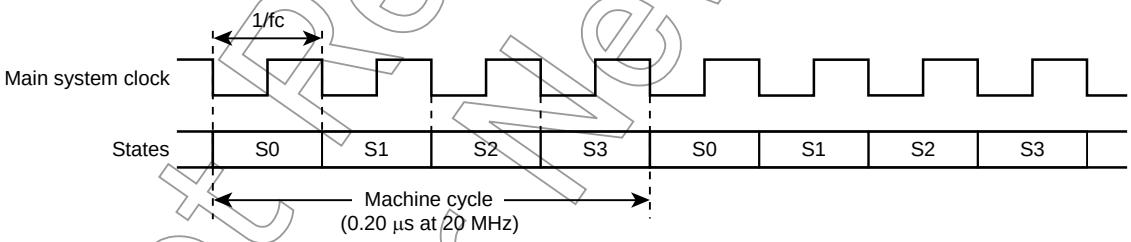


Figure 2-4 Machine Cycles

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 (SYSCR1, SYSCR2) are used to control operation modes of this circuit. Figure 2-5 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 TMP88F846UG 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.

3. STOP mode

   The entire system operation including the oscillator circuit is halted, retaining the internal state immediately before being stopped, with a minimal amount of power consumed.

   STOP mode is entered into by using System Control Register 1, and is exited by $\overline{\text{STOP}}$ pin input (level or edge selectable). After an elapse of the warm-up time, the device restarts execution beginning with the instruction next to one that placed it in STOP mode.

Table 2-1 Single Clock Mode

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

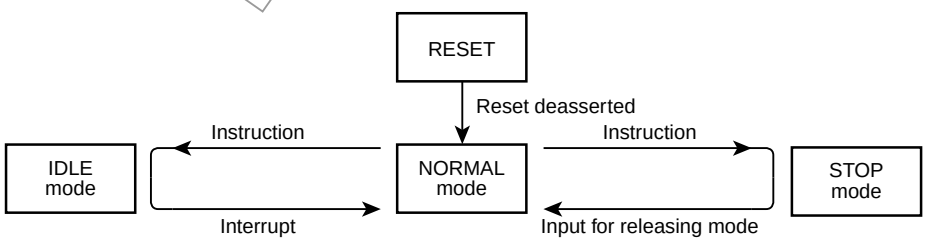


Figure 2-5 Operation Mode Transition Diagram

System Control Register 1

SYSCR1 (0038H)

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

(Initial value: 0000 00**)

STOP	Place the device in STOP mode	0: Keep the CPU core and peripheral hardware operating 1: Stop the CPU core and peripheral hardware (placed in STOP mode)		R/W
RELM	Select method by which the device is released from STOP mode	0: Released by a rising edge on $\overline{STOP}$ pin input 1: Released by a high level on $\overline{STOP}$ pin input		
RETM	Select operation mode after exiting STOP mode	0: Returns to NORMAL mode 1: Reserved		
OUTEN	Select port output state during STOP mode	0: High-impedance state 1: Hold output		
WUT	Unit of warm-up time when exiting STOP mode		When Returning to NORMAL Mode	
			DV1CK = 0	DV1CK = 1
		00	3×2^{16}/fc	3×2^{17}/fc
		01	2^{16}/fc	2^{17}/fc
		10	2^{14}/fc	2^{15}/fc
		11	Reserved	Reserved

Note 1: When entering from NORMAL mode into STOP mode, always be sure to set SYSCR1<RETM> to 0.

Note 2: When the device is released from STOP mode by $\overline{RESET}$ pin input, it always returns to NORMAL mode regardless of how SYSCR1<RETM> is set.

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

Note 4: The values of the SYSCR1 Register bits 1 and 0 are indeterminate when read.

Note 5: When placed the device in STOP mode, make sure to set "1" to SYSCR1<OUTEN>.

Note 6: Releasing the device from the STOP mode causes the STOP bit to be automatically cleared to "0".

Note 7: Select an appropriate value for the warm-up time according to the characteristic of the resonator used.

System Control Register 2

SYSCR2 (0039H)

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

(Initial value: 1000 ****)

XEN	Control high-frequency oscillator	0: Stop oscillation 1: Continue or start oscillating	R/W
SYSCK	Select (write)/monitor (read) system clock	0: High-frequency clock (NORMAL/IDLE) 1: Reserved	R/W
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

RETM	Operation Mode after Releasing STOP Mode	XEN	SYSCK
0	NORMAL mode	1	0
1	No operation	0	1

Note 1: When exiting STOP mode, SYSCR2<XEN and SYSCK> are automatically rewritten according to SYSCR1<RETM>.

Note 2: When SYSCR2<XEN>is cleared to 0, the device is reset.

Note 3: WDT: Watchdog Timer, *: Don't care

Note 4: Be sure to write "0" to SYSCR2 Register bit6.

Note 5: The values of the SYSCR2 Register bits 3 to 0 are indeterminate when read.

Note 6: Change the operation mode after disabling external interrupts. If interrupts are enabled after changing operation mode, clear interrupt latches as appropriate in advance.

2.1.4.4 Controlling Operation Modes

(1) STOP mode

STOP mode is controlled by System Control Register 1 (SYSCR1) and the $\overline{\text{STOP}}$ pin input. The $\overline{\text{STOP}}$ pin is shared with P20 port and $\overline{\text{INT5}}$ (external interrupt input 5). STOP mode is entered into by setting STOP (SYSCR1 Register bit 7) to 1. During STOP mode, the device retains the following state.

1. Stop oscillation, thereby stopping operation of all internal circuits.
2. The data memory, register, program status word, and port output latch hold the state in which they were immediately before entering STOP mode.
3. Clear the prescaler and divider for the timing generator to 0.
4. The program counter holds the instruction address two instructions ahead the one that placed the device in STOP mode (e.g., "SET (SYSCR1).7").

The device is released from STOP mode by the active level or edge on STOP pin input as selected by SYSCR1<RELM>.

Note: Before entering STOP mode, be sure to disable interrupts. This is because if the signal on an external interrupt pin changes state during STOP (from entering STOP mode till completion of warm-up) the interrupt latch is set to 1, so that the device may accept the interrupt immediately after exiting STOP mode. Also, when enabling interrupts after exiting STOP mode, be sure to clear the unnecessary interrupt latches beforehand.

a. Released by level (when RELM = 1)

The device is released from STOP mode by a high level on $\overline{\text{STOP}}$ pin input.

Any instruction to place the device in STOP mode is ignored when executed while $\overline{\text{STOP}}$ pin input level is high, and the device immediately goes to a release sequence (warm-up) without entering STOP mode. Therefore, before STOP mode can be entered while RELM = 1, the $\overline{\text{STOP}}$ pin input must be verified to be low in a program. There are following methods to do this verification.

1. Testing the port status
2. INT5 interrupt (interrupt generated at a falling edge on $\overline{\text{INT5}}$ pin input)

Example 1 :Entering STOP mode from NORMAL mode by testing P20 port

```
        LD      (SYSCR1), 01010000B     ; Select to be released from STOP mode by level
SSTOPH: TEST    (P2DR) . 0              ; Wait until STOP pin input goes low
        JRS     F, SSTOPH
        DI                              ; IMF ← 0
        SET     (SYSCR1) . 7            ; Place the device in STOP mode
```

Example 2 :Entering STOP mode from NORMAL mode by INT5 interrupt

```
PINT5 : TEST    (P2DR) . 0              ; Do not enter STOP mode if P20 port input level is
                                        ; high, to eliminate noise
        JRS     F, SINT5                ; Do not enter STOP mode if P20 port input level is
                                        ; high, to eliminate noise
        LD      (SYSCR1), 01010000B     ; Select to be released from STOP mode by level
        DI                              ; IMF ← 0
        SET     (SYSCR1) . 7            ; Place the device in STOP mode
SINT5 : RETI
```

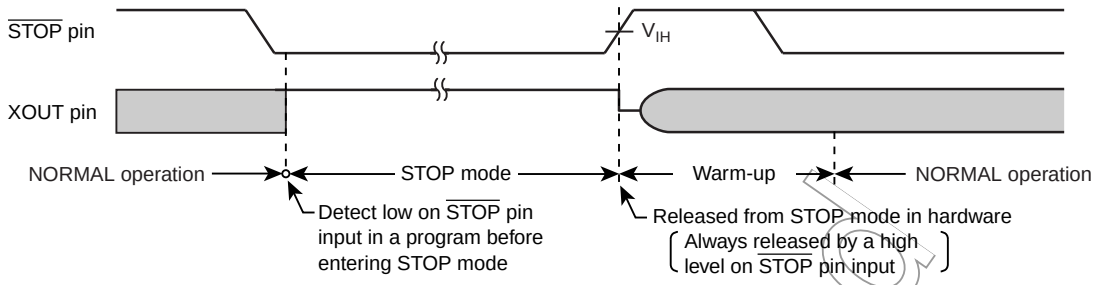


Figure 2-6 Released from STOP Mode by Level

Note 1: Once warm-up starts, the device does not return to STOP mode even when the STOP pin input is pulled low again.

Note 2: If RELM is changed to 1 (level mode) after being set to 0 (edge mode), STOP mode remains unchanged unless a rising edge on STOP pin input is detected.

a. Released by edge (when RELM = 0)

The device is released from STOP mode by a rising edge on STOP pin input. This method is used in applications where a relatively short time of program processing is repeated at certain fixed intervals. Apply a fixed-period signal (e.g., clock from the low-power oscillating source) to the STOP pin. When RELM = 0 (edge mode), the device is placed in STOP mode even when the STOP pin input level is high.

Example :Entering STOP mode from NORMAL mode

```
DI                          ; IMF ← 0
LD      (SYSCR1) , 10010000B  ; Set to be released by edge when entering STOP mode
```

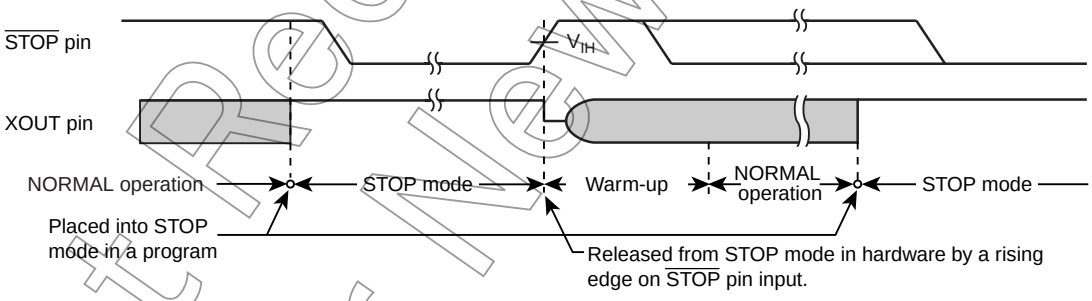


Figure 2-7 Released from STOP Mode by Edge

The device is released from STOP mode following the sequence described below.

1. Only the high-frequency oscillator is oscillating.
2. A warm-up time is inserted in order to allow for the clock oscillation to stabilize. During warm-up, the internal circuits remain idle. The warm-up time can be selected from three choices according to the oscillator characteristics by using SYSCR1<WUT>.
3. After an elapse of the warm-up time, the device restarts normal operation beginning with the instruction next to one that placed it in STOP mode. At this time, the prescaler and divider for the timing generator start from the zero-cleared state.

Table 2-2 Warm-up Time (Example: fc = 20 MHz)

WUT	Warm-up Time [ms]	
	When Returning to NORMAL Mode	
	DV1CK = 0	DV1CK = 1
00	9.831	19.662
01	3.277	6.554
10	0.819	1.638
11	Reserved	Reserved

Note: Because the warm-up time is obtained from the fundamental clock by dividing it, if the oscillation frequency fluctuates while exiting STOP mode, the warm-up time becomes to have some error. Therefore, the warm-up time must be handled as an approximate value.

The device can also be released from STOP 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: When exiting STOP mode while the device is retained at low voltage, the following caution is required.

Before exiting STOP mode, the power supply voltage must be raised to the operating voltage. At this time, the $\overline{\text{RESET}}$ pin level also is high and rises along with the power supply voltage. If the device has a time-constant circuit added external to the chip, the voltage on $\overline{\text{RESET}}$ pin input does not rise as fast as the power supply voltage. Therefore, if the voltage level on $\overline{\text{RESET}}$ pin input is below the $\overline{\text{RESET}}$ pin's noninverted, high-level input voltage (hysteresis input), the device may be reset.

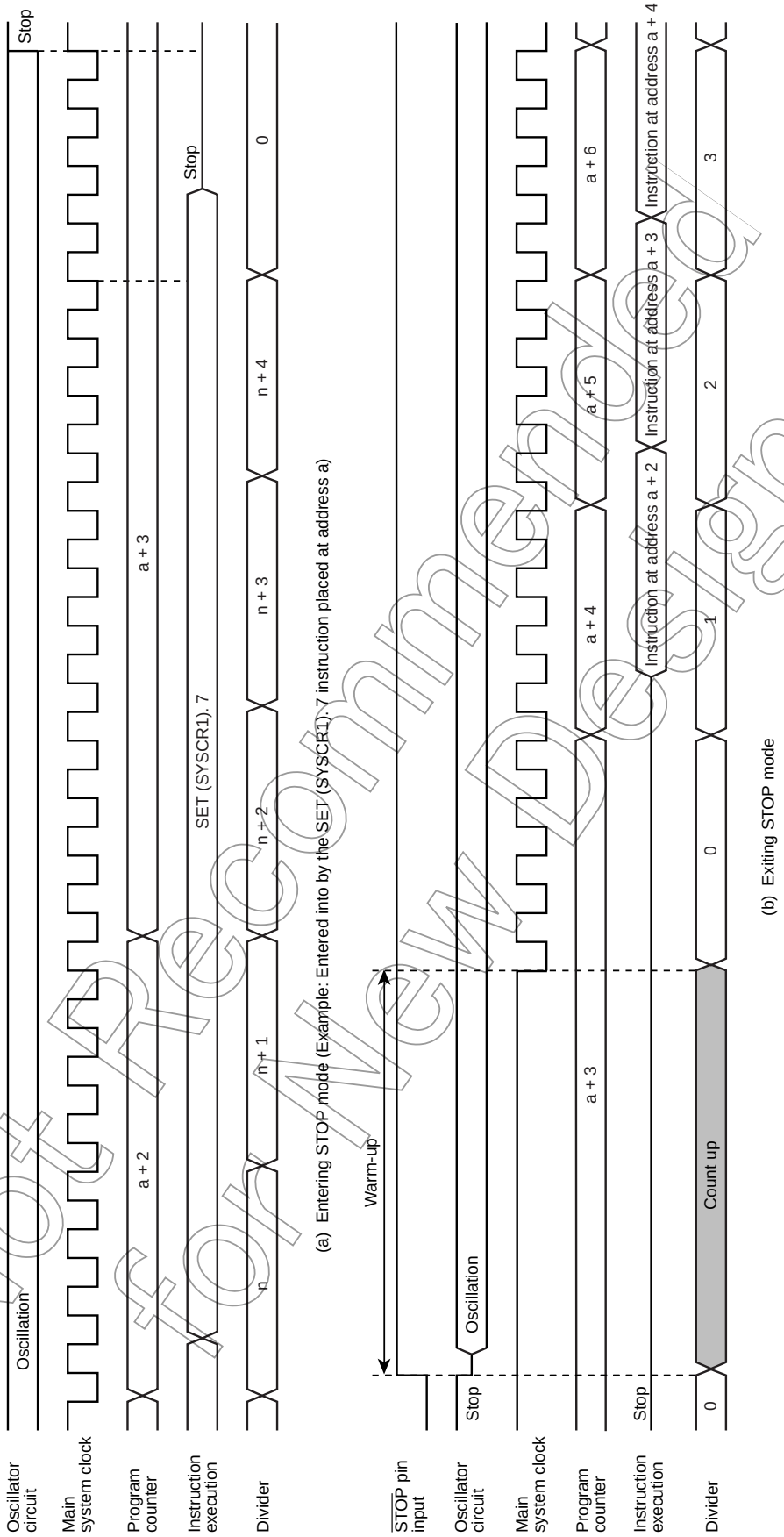


Figure 2-8 Entering and Exiting STOP Mode (when DV1CK = 0)

(2) 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
```

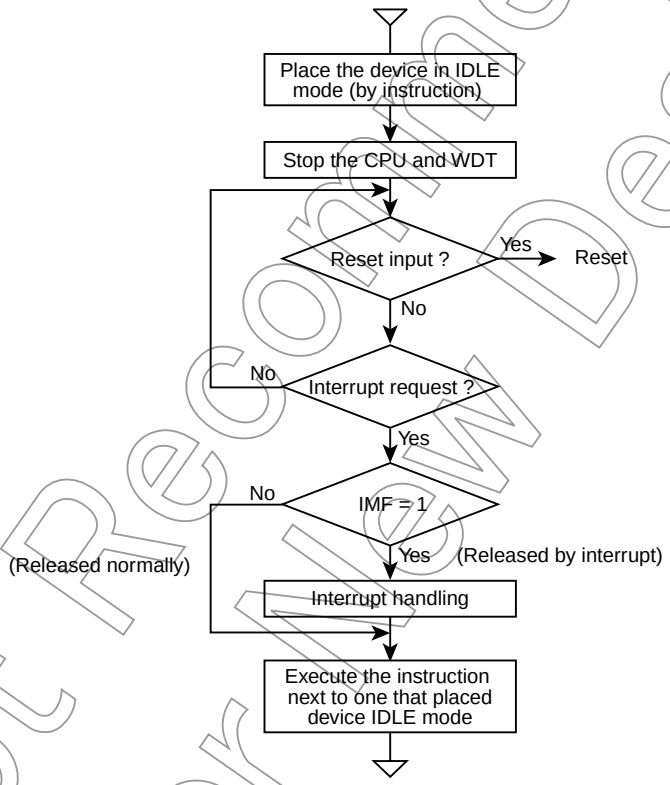


Figure 2-9 IDLE Mode

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.

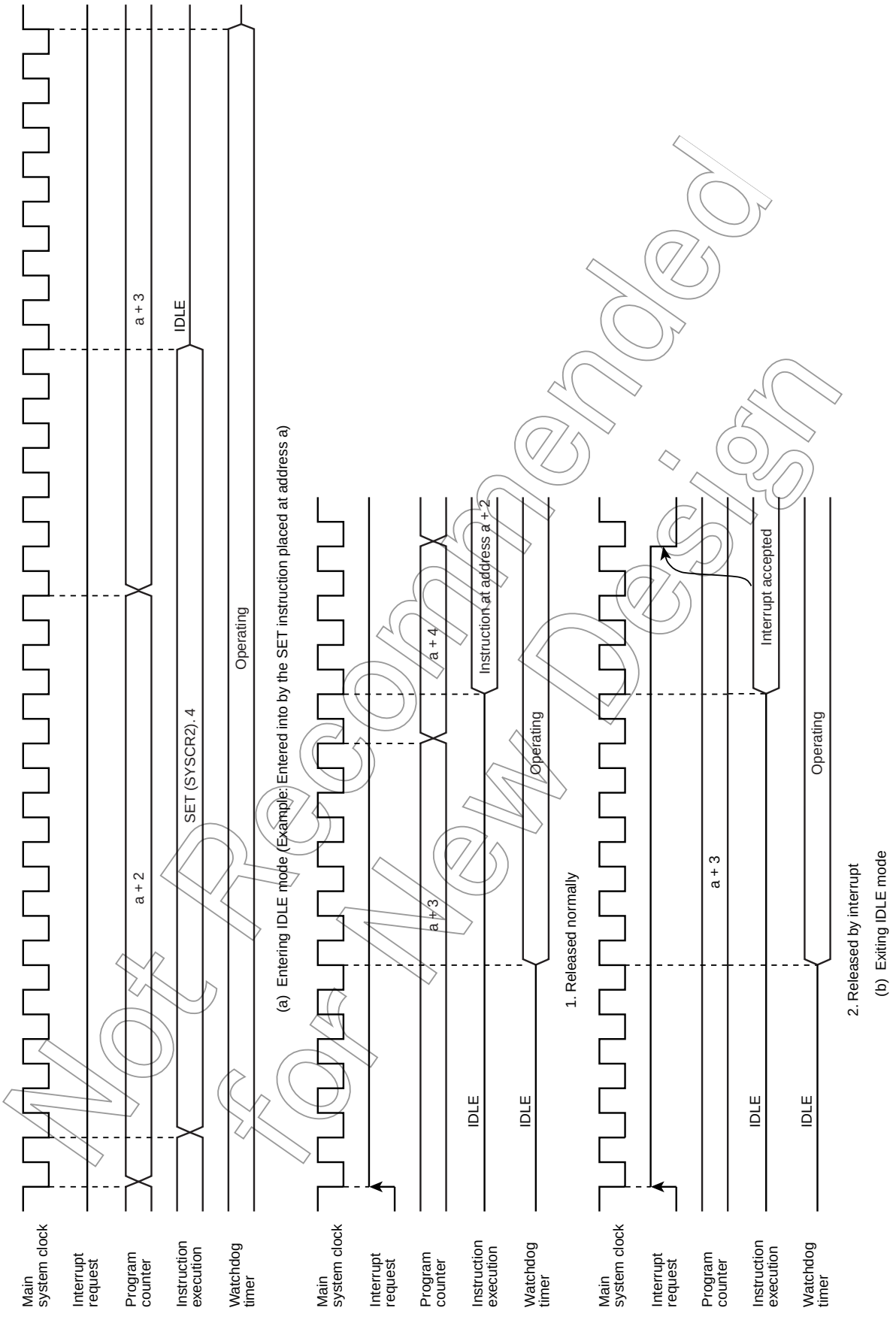


Figure 2-10 Entering and Exiting IDLE Mode

2.1.5 Reset Circuit

The TMP88F846UG has seven ways to generate a reset: external reset input, address trap reset, watchdog timer reset , system clock reset, voltage detect reset, power on reset and trimming data reset.

Table 2-3 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.

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

Table 2-3 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.

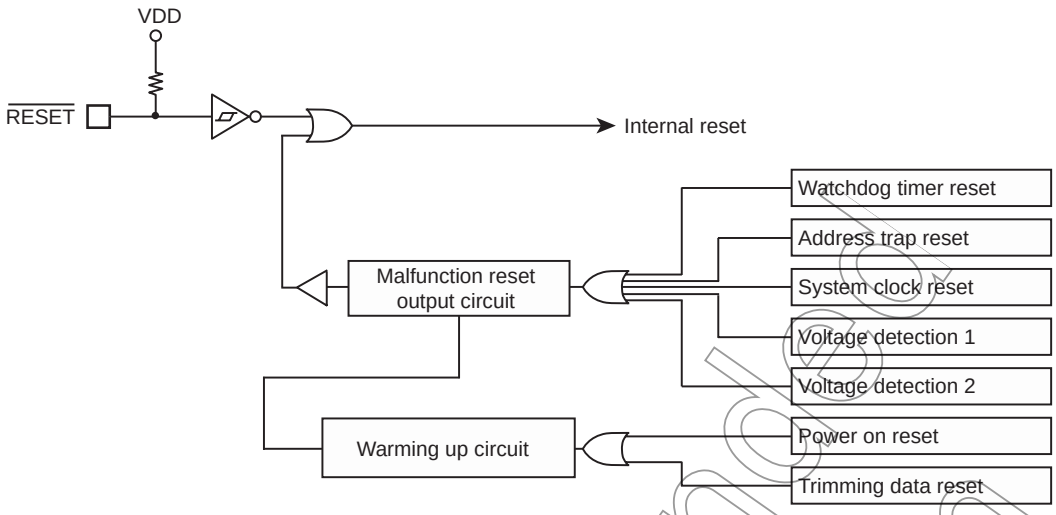


Figure 2-11 Reset Circuit

2.1.5.2 Address 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 generates an internal reset.

The address trap permission/prohibition is set by the address trap reset control register (ATAS,ATKEY). The address trap is permitted 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<XEN> is cleared to 0 or when SYSCR2<XEN> is cleared to 0 while SYSCR2<SYSCK> = 0, the system clock is turned off, causing the CPU to become locked up. To prevent this problem, upon detecting SYSCR2<XEN> = 0, SYSCR2<XEN> = SYSCR2<SYSCK> = 0 or SYSCR2<SYSCK> = 1, the device automatically generates an internal reset signal to let the system clock continue oscillating.

2.1.5.5 Voltage detection reset

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

Refer to Section "Voltage detection circuit".

2.1.5.6 Power-on Reset

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

Refer to Section "Power on reset".

2.1.5.7 Trimming data reset

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

3. Interrupt Control Circuit

The TMP88F846UG has a total of 26 interrupt sources excluding reset. Interrupts can be nested with priorities. Two of the internal interrupt sources are pseudo non-maskable while the rest are maskable.

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

Interrupt Factors		Enable Condition	Interrupt Latch	Vector Address	Priority
Internal/External	(Reset)	Non-maskable	-	FFFFC	High 0
Internal	INTSWI (Software interrupt)	Pseudo non-maskable	-	FFFF8	1
Internal	INTWDT (Watchdog timer interrupt)	Pseudo non-maskable	IL2	FFFF4	2
External	$\overline{\text{INT0}}$ (External interrupt 0)	IMF・EF3 = 1, INT0EN = 1	IL3	FFFF0	3
-	Reserved	IMF・EF4 = 1	IL4	FFFEC	4
External	INT1 (External interrupt 1)	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
External	INT5 (External interrupt 5)	IMF・EF15 = 1	IL15	FFFC0	15
Internal	INTPDC1 (ch1 Position detect interrupt)	IMF・EF16 = 1	IL16	FFFBC	16
-	Reserved	IMF・EF17 = 1	IL17	FFFB8	17
Internal	INTPWM1 (ch1 Waveform generator interrupt)	IMF・EF18 = 1	IL18	FFFB4	18
-	Reserved	IMF・EF19 = 1	IL19	FFFB0	19
Internal	INTEDT1 (ch1 Electric 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
Internal	INTCTC1 (CTC interrupt)	IMF・EF27 = 1	IL27	FFF90	27
-	Reserved	IMF・EF28 = 1	IL28	FFF8C	28
External	INT2 (External interrupt 2)	IMF・EF29 = 1	IL29	FFF88	29
External	INT3 (External interrupt 3)	IMF・EF30 = 1	IL30	FFF84	30
External	INT4 (External interrupt 4)	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	38
Internal	INTVLTD	IMF・EF39 = 1	IL39	FFF20	Low 39

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 (IL39 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), 00000000B    ; IL2 to IL7 ← 0
LD      (ILH), 00000000B    ; IL8 to IL15 ← 0
LD      (ILE), 00000000B    ; IL16 to IL23 ← 0
LD      (ILD), 00000000B    ; IL24 to IL31 ← 0
LD      (ILC), 00000000B    ; IL32 to IL39 ← 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 (EF39 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 (EF39 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 *00*0000)

ILH,ILL (003DH, 003CH)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IL15	-	-	IL12	-	IL10	-	IL8	-	IL6	IL5	-	IL3	IL2	INF	

ILH (003DH) ILL (003CH)

(Initial value: 000*00*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
IL31	IL30	IL29	-	IL27	IL26	-	IL24	-	IL22	-	IL20	-	IL18	-	IL16

ILD (002FH) ILE (002EH)

(Initial value: 00*00000)

ILC (002BH)

7	6	5	4	3	2	1	0
IL39	IL38	-	IL36	IL35	IL34	IL33	IL32

ILE (002BH)

			Read	Write	
IL39 to IL2	Interrupt latches	0:	No interrupt request	Clears the interrupt request (Note1)	R/W
		1:	Interrupt request	(Unable to set interrupt latch)	
INF	Interrupt Nesting Flag	00:	Out of interrupt service	Reserved	
		01:	On interrupt service of level 1	Clear the nesting counter	
		10:	On interrupt service of more than level 2	Count-down 1 step for the nesting counter (Note2)	
		11:	On interrupt service of more than level 3	Reserved	

Note 1: IL2 cannot alone be cleared.

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 *00*0**0)

EIRH,EIRL (003BH, 003AH)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EF15	-	-	EF12	-	EF10	-	EF8	-	EF6	EF5	-	EF3			IMF

EIRH (003BH) EIRL (003AH)

(Initial value: 000*00*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
EF31	EF30	EF29	-	EF27	EF26	-	EF24	-	EF22	-	EF20	-	EF18	-	EF16

EIRD (002DH) EIRE (002CH)

(Initial value: 00*00000)

EIRE (002AH)

7	6	5	4	3	2	1	0
EF39	EF38	-	EF36	EF35	EF34	EF33	EF32

EIRE (002AH)

EF39 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 (EF39 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".

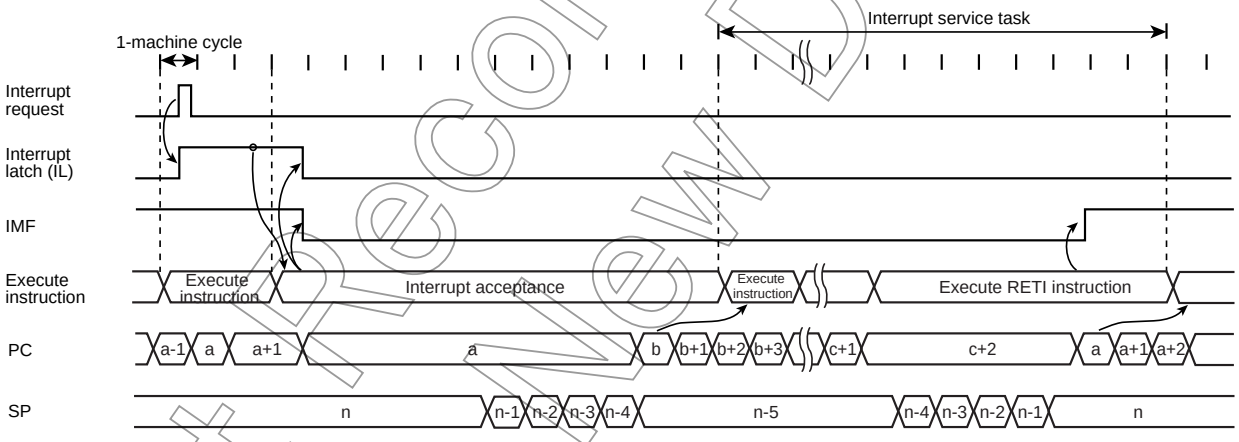
3.3 Interrupt Sequence

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 selector (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

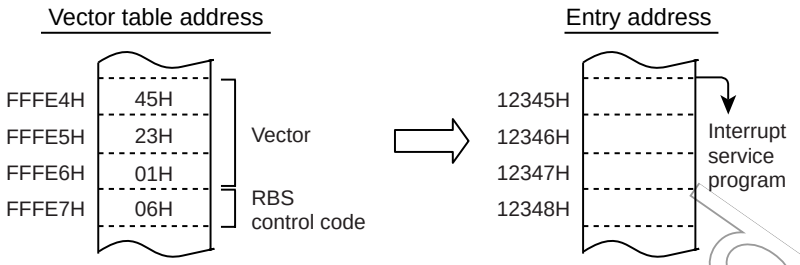


Figure 3-2 Vector table address, Entry address

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 switching

By switching to non-use register bank, it can restore the general-purpose register at high 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 interrupt 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 high 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
```

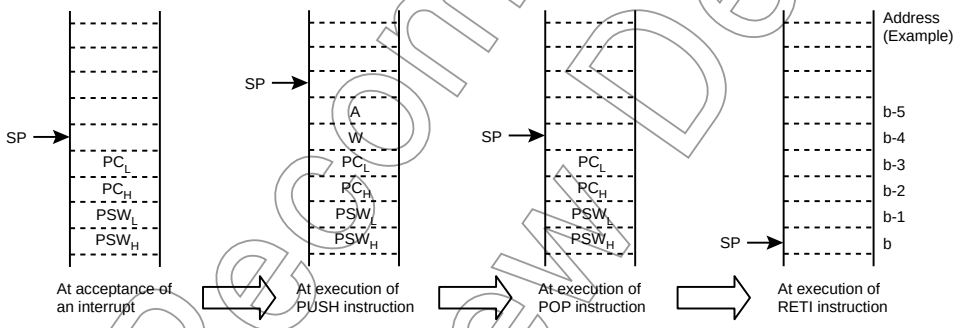


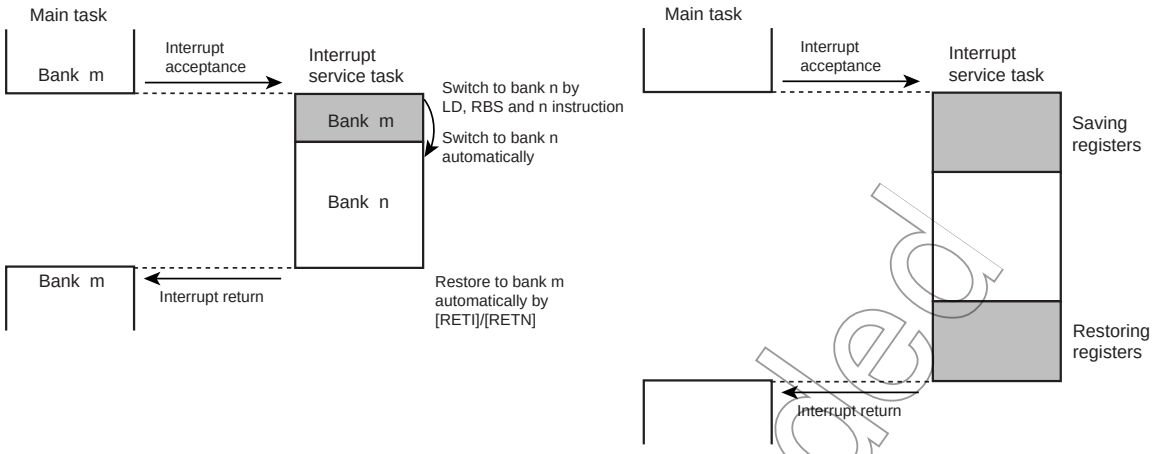
Figure 3-3 Save/store register using PUSH and POP instructions

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.	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.	2.	The stack pointer is incremented 5 times.
3.	The interrupt master enable flag is set to "1".	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.	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 interrupt 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 TMP88F846UG has 6 external interrupt inputs. These inputs are equipped with digital noise reject circuits (Pulse inputs of less than a certain time are eliminated as noise).

Edge selection is also possible with INT1 to INT4. The $\overline{\text{INT0}}$/P10 pin can be configured as either an external interrupt input pin or an input/output port, and is configured as an input port during reset.

Edge selection, and 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).
INT1	INT1	P11	IMF + EF5 = 1	Falling edge or Rising edge	Pulses of less than 15/fc or 63/fc [s] are eliminated as noise. Pulses of 48/fc or 192/fc [s] or more are considered to be signals. (at CGCR<DV1CK>=0).
INT2	INT2	P12/TC1	IMF + EF29 = 1		Pulses of less than 7/fc [s] are eliminated as noise. Pulses of 24/fc [s] or more are considered to be signals.(at CGCR<DV1CK>=0).
INT3	INT3	P21/TC3	IMF + EF30 = 1		
INT4	INT4	P22/TC4	IMF + EF31 = 1		
INT5	$\overline{\text{INT5}}$	P20/$\overline{\text{STOP}}$	IMF + EF15 = 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.

Note 1: In NORMAL or IDLE mode, if a signal with no noise is input on an external interrupt pin, it takes a maximum of "signal establishment time" from the input signal's edge to set the interrupt latch.
(1) INT1 pin 49/fc [s] (at EINTCR<INT1NC> = "1") , 193/fc [s] (at EINTCR<INT1NC> = "0")
(2) INT2 to INT4 pins 25/fc [s]

Note 2: When EINTCR<INT0EN> = "0", IL3 is not set even if a falling edge is detected on the $\overline{\text{INT0}}$ pin input.

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

External Interrupt Control Register

EINTCR (0037H)

7	6	5 4	3	2	1	0
INT1NC	INT0EN	INT4ES	INT3ES	INT2ES	INT1ES	

(Initial value: 0000 000*)

INT1NC	Noise reject time select	0: Pulses of less than 63/fc [s] are eliminated as noise 1: Pulses of less than 15/fc [s] are eliminated as noise	R/W
INT0EN	P10/$\overline{\text{INT0}}$ pin configuration	0: P10 input/output port 1: $\overline{\text{INT0}}$ pin (Port P10 should be set to an input mode)	R/W
INT4 ES	INT4 edge select	00: Rising edge 01: Falling edge 10: Rising edge and Falling edge 11: H level	R/W
INT3 ES	INT3 edge select	0: Rising edge 1: Falling edge	R/W
INT2 ES	INT2 edge select		
INT1 ES	INT1 edge select		

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).

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

Note 4: In case $\overline{\text{RESET}}$ pin is released while the state of INT4 pin keeps "H" level, the external interrupt 4 request is not generated even if the INT4 edge select(EINTCR<INT4ES>) is specified as "H" level. The rising edge is needed after $\overline{\text{RESET}}$ pin is released.

4. Special Function Register

The TMP88F846UG 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 TMP88F846UG.

4.1 SFR

Address	Read	Write
0000H	Reserved	
0001H	P1DR	
0002H	P2DR	
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	CTC1CR1	
0015H	CTC1CR2	
0016H	-	CTC1DRL
0017H	-	CTC1DRH
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	SYSCR1	
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		VDCR1	
1FCDH		VDCR2	
1FCEH		Reserved	
to		:	
1FFDH		Reserved	
1FFEH		SPCR	
1FFFH		FLSCR	

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

Note 2: −: Cannot be accessed.

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

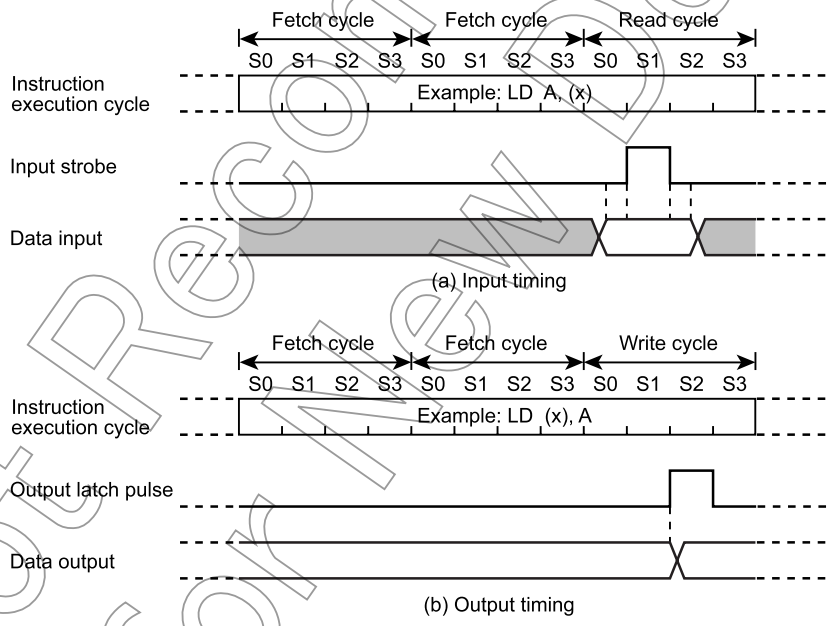
5. Input/Output Ports

The TMP88F846UG contains 5 input/output ports comprised of 33 pins.

	Primary Function	Secondary Functions
Port P1	6-bit I/O port	External interrupt input, Timer/counter input/output, divider output
Port P2	3-bit I/O port	External interrupt input, Timer/counter input/output, STOP mode release signal input
Port P3	8-bit I/O port	Motor control input/output
Port P4	8-bit I/O port	Timer/counter output, Serial interface input/output, motor control circuit input, Serial PROM mode control input
Port P6	8-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 (P15 to P10)

Port P1 is an 6-bit input/output port shared with external interrupt input, timer/counter input/output, and divider output. 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.

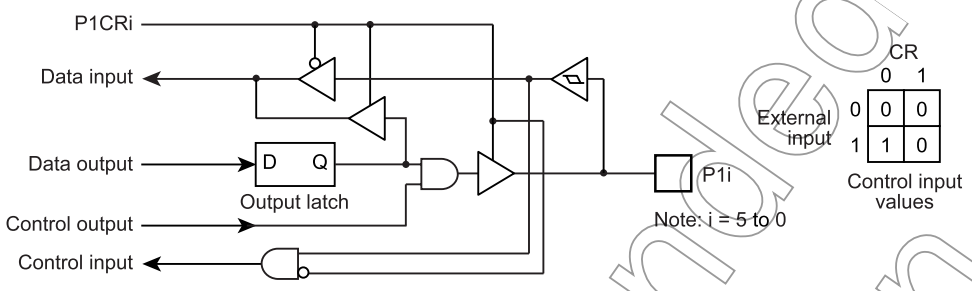


Figure 5-2 Port P1

P1 Port Input/Output Register

	7	6	5	4	3	2	1	0	
P1DR (00001H)			P15	P14 / PPG1	P13 / DVO	P12 / INT2 / TC1	P11 / INT1	P10 / INT0	Read/Write (Initial value: **00 0000)
P1CR (0000BH)	7	6	5	4	3	2	1	0	(Initial value: **00 0000)

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

Note 1: When a read instruction is excuted on P1 port, indeterminate values are read in from bits 7 to 6.

Note 2: *: Don't care

5.2 Port P2 (P22 to P20)

Port P2 is a 3-bit input/output port shared with external interrupt input and STOP mode release signal. When using this port as these functional pins or an input port, set the output latch to 1. When reset, the output latch is initialized to 1.

We recommend using the P20 pin as external interrupt input, STOP mode release signal input, or input port. When using this port as an output port, note that the interrupt latch is set by a falling edge of output pulse. And note that outputs on this port during STOP mode go to a high-impedance state even if SYSCR1<OUTEN> is set "1" , because P20 port is also used as $\overline{\text{STOP}}$ pin.

When a read instruction is executed on P2 port, indeterminate values are read in from bits 7 to 3.

When any read-modify-write instruction is executed on P2 port, the content of the output latch is read out. When any other instruction is executed, the external pin state is read out.

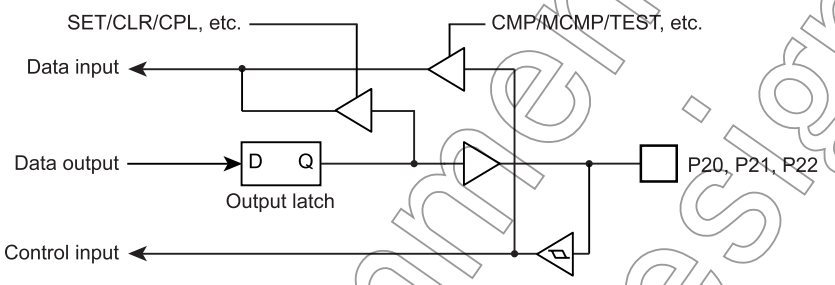


Figure 5-3 Port P2

P2 Port Input/Output Registers

	7	6	5	4	3	2	1	0	
P2DR (00002H)						P22 TC4 INT4 $\overline{\text{PWM4}}$ $\overline{\text{PDO4}}$	P21 TC3 INT3	P20 $\overline{\text{INT5}}$ $\overline{\text{STOP}}$	Read/Write (Initial value: **** *111)

Note 1: When a read instruction is executed on P2 port, indeterminate values are read in from bits 7 to 3.

Note 2: Port P20 is used as $\overline{\text{STOP}}$ pin. Therefore, when stop mode is started, SYCR1<OUTEN> does not affect to P20, and P20 becomes High-Z mode.

Note 3: *: Don't care

5.3 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.

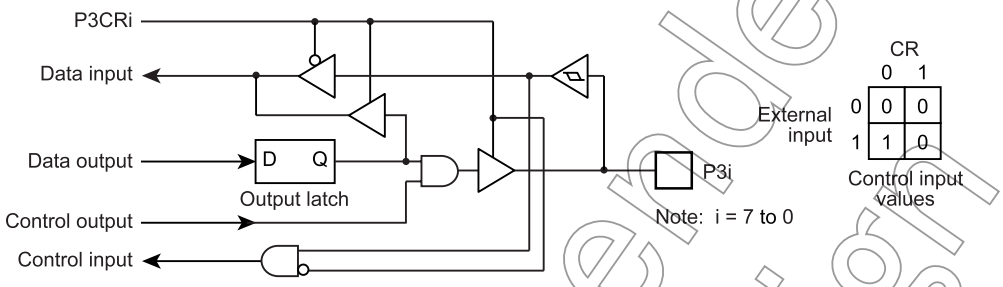


Figure 5-4 Port P3

P3 Port Input/Output Registers

	7	6	5	4	3	2	1	0	
P3DR (00003H)	P37 $\overline{CL1}$	P36 $\overline{EMG1}$	P35 U1	P34 V1	P33 W1	P32 X1	P31 Y1	P30 Z1	Read/Write (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: Read-modify-write (RMW) operation executes at open-drain mode is selected, read out the output latch states. When any other instruction is executed, external pin states is 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.

5.4 Port P4 (P47 to P40)

Port P4 is an 8-bit input/output port shared with serial interface input/output and serial PROM mode control input. 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.

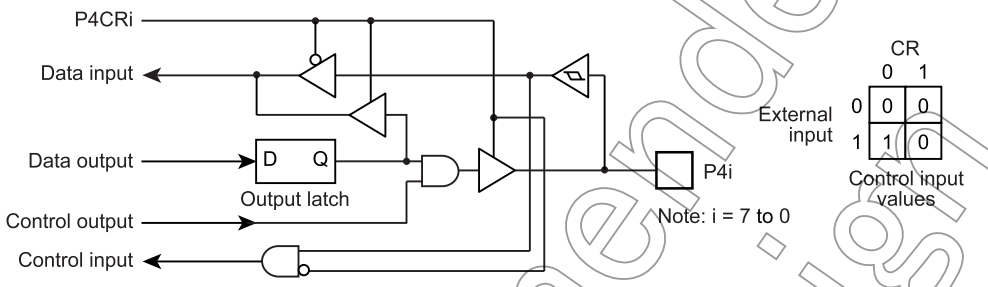


Figure 5-5 Port P4

P4 Port Input/Output Registers

P4DR (00004H)

7	6	5	4	3	2	1	0
P47 CTC	P46 $\overline{PPG2}$	P45 SO TXD	P44 SI RXD BOOT	P43 $\overline{SCK}$	P42 PDU1	P41 PDV1	P40 PDW1

(Initial value: 0000 0000)

P4CR (01F8AH)

7	6	5	4	3	2	1	0

(Initial value: 0000 0000)

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

P4ODE (01F84H)

7	6	5	4	3	2	1	0

(Initial value: 0000 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: Read-modify-write (RMW) operation executes at open-drain mode is selected, read out the output latch states. When any other instruction is executed, external pin states is read out.

Note 3: When using the CTC 16-bit timer (CTC) as an ordinary timer, set P47 (CTC) for output mode.

5.5 Port P6 (P67 to P60)

Port P6 is an 8-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 P67 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.

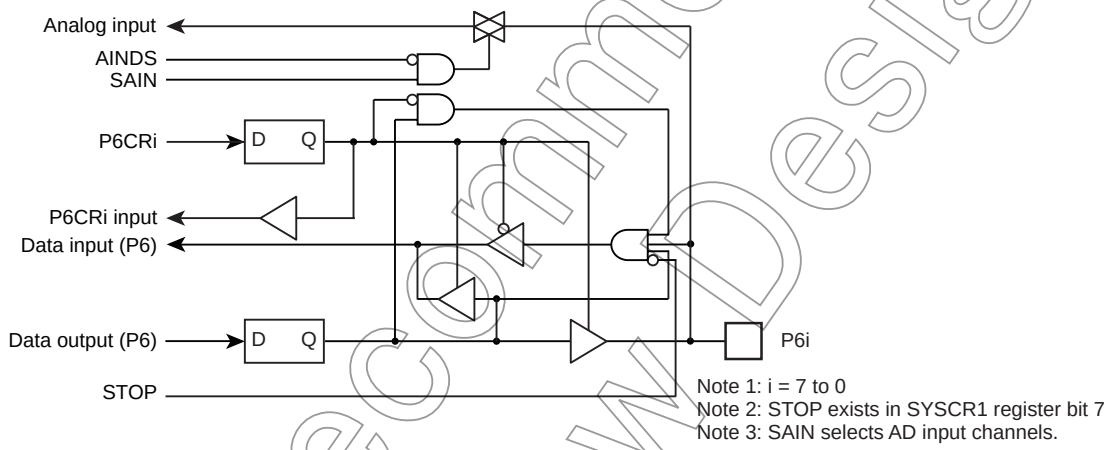


Figure 5-6 Port P6

P6 Port Input/Output Registers

	7	6	5	4	3	2	1	0	
P6DR (00006H)	P67 AIN7 DBOUT1	P66 AIN6	P65 AIN5	P64 AIN4	P63 AIN3	P62 AIN2	P61 AIN1	P60 AIN0	Read/Write (Initial value: 0000 0000)
P6CR (01F8CH)									(Initial value: 0000 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				

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 DBOUT1 output, set the P6DR (P67) output latch to 1.

Note 4: 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. Power-on reset circuit

The power-on reset circuit generates a reset when the TMP88F846UG is powered on. It also generates a power-on reset signal if the supply voltage drops below the detection voltage of the power-on reset circuit.

6.1 Configuration

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

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

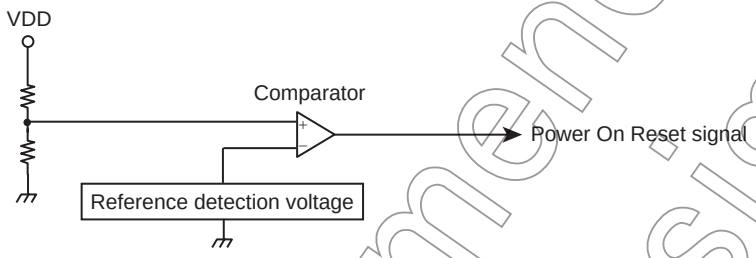


Figure 6-1 Power-on reset circuit

6.2 function

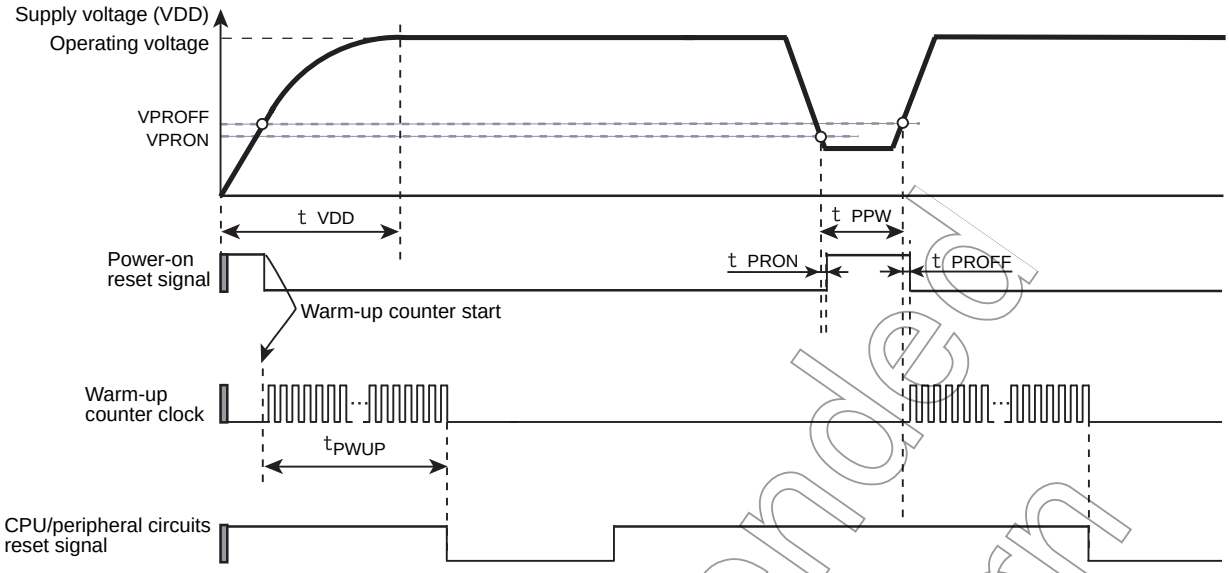
When the TMP88F846UG is powered on, the power-on reset circuit generates a power-on reset signal while the supply voltage is below the power-on reset release voltage. The power-on reset signal is released when the supply voltage rises above the power-on reset release voltage.

When the TMP88F846UG is shut off, the power-on reset circuit generates a power-on reset signal when the supply voltage drops below the power-on reset detection voltage.

While the power-on reset signal is generated, the warm-up counter circuit, CPU and peripheral circuits are reset.

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

After the supply voltage reaches the power-on reset release voltage level, it must be raised to the operating range before the power-on warm-up time expires. If the supply voltage is not in the operating range at the completion of the power-on warm-up time, the TMP88F846UG cannot operate properly.



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

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

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

Figure 6-2 Operation timing of the power-on reset circuit

7. Voltage detection circuit (VLTD)

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

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

7.1 Configuration

The voltage detection circuit is comprised of a reference voltage generator, a detection voltage select circuit, a comparator and two control registers.

The supply voltage (VDD) is divided by the ladder resistor and input to the detection voltage select circuit. The detection voltage select circuit selects a voltage according to the specified detection voltage (VD1LVL), and the comparator compares it with the reference voltage. When the comparator detects the selected voltage, a voltage detection reset signal or an INTVLTD interrupt request signal can be generated.

Whether to generate a voltage detection reset signal or an INTVLTD interrupt request signal can be programmed by software. In the former case, a voltage detection reset signal is generated when the supply voltage (VDD) becomes lower than the detection voltage (VD1LVL). In the latter case, an INTVLTD interrupt request signal is generated when the supply voltage (VDD) falls to the detection voltage level.

Note: Since the comparators used for voltage detection do not have a hysteresis structure, INTVLTD interrupt request signals may be generated frequently if the supply voltage (VDD) is close to the detection voltage (VDxLVL). INTVLTD interrupt request signals may be generated not only when the supply voltage falls to the detection voltage but also when it rises to the detection voltage.

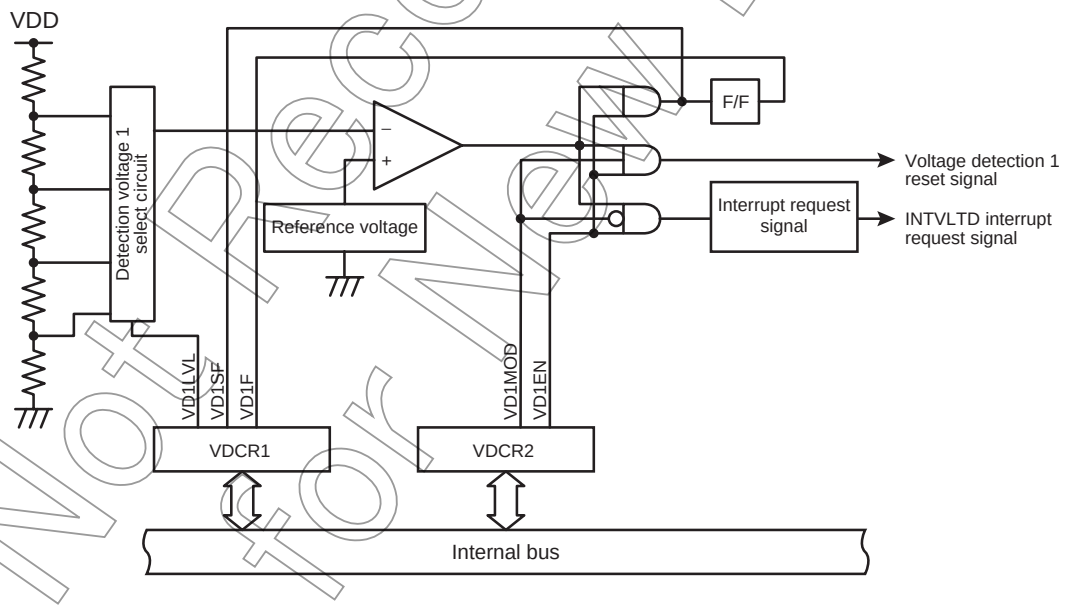


Figure 7-1 Voltage detection circuit diagram

Note: In the "development tool" of this device, since operation of the voltage detection circuit is not supported, even if it detects the set-up voltage, interrupt is not generated. Please check using a real device about actual operation.

7.2 Control

The voltage detection circuit is controlled by the Voltage Detect Control Register 1 and Voltage Detect Control Register 2.

voltage detection control register1

VDCR1	7	6	5	4	3	2	1	0	
(1FCCH)	0	0	0	1	VD1F	VD1SF	VD1LVL		(Initial: 0*10 0000)

VD1F	Voltage detection 1 flag (VDD<VD1LVL When with becoming, keeping state) (Note 2)	Read 0 : VDD ≥ VD1LVL 1 : VDD < VD1LVL Write 0 : Clearing the flag 1 : -(Cannot write "1"by the software)	R/W
VD1SF	Voltage detection 1 status flag (Large and something related to VDD and VD1LVL of the point in time when it reads out)	0 : VDD ≥ VD1LVL 1 : VDD < VD1LVL	Read only
VD1LVL	Voltage detection 1 level select	00 : 4.3 to 4.7V 01 : 4.0 to 4.4V 10 : Reserved 11 : Reserved	R/W

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

Note 2: If the VD1F bit is cleared by software at the same time as it is set by detection of a low-voltage condition, the setting operation overrides the clearing operation so that the bit is set to 1.

Note 3: Be sure to write “0” to VDCR1 Register bits 7 and 5, and write "1" to VDCR1 Register bit 4.

Note 4: The values of the VDCR1 Register bit 6 is indeterminate when read.

voltage detection control register 2

VDCR2	7	6	5	4	3	2	1	0	
(1FCDH)					0	0	VD1MOD	VD1EN	(Initial: **** 0000)

VD1MOD	Voltage detection 1 operation mode select	0 : INTVLTD interrupt 1 : Voltage detection 1 reset signal occurrence	R/W
VD1EN	Voltage detection 1 operation enable/disable	0 :Voltage detection 1 disable 1 : Voltage detection 1 enable	R/W

Note 1: Note: The VDCR2 is only initialized by a power-on reset or an external reset input.

Note 2: Be sure to write “0” to VDCR2 Register bits 3 and 2.

7.3 Function

The voltage detection circuit allows detection voltage level to be specified. The detection voltage, whether to enable or disable voltage detect operation and the action to be taken when the supply voltage (VDD) falls to or below the detection voltage (VD1LVL) can be programmed by software.

7.3.1 Enabling/Disabling Voltage Detection Operation

Setting the VDCR2<VD1EN> bit to 1 enables voltage detect operation and clearing this bit to 0 disables it. Immediately after release of a power-on reset,VDCR2<VD1EN>, is cleared to 0.

Note: Setting VDCR2<VD1EN> to 1 while the supply voltage is below the detection voltage (VDD < VD1LVL) immediately causes an INTVLTD interrupt or a voltage detect x reset signal to be generated.

7.3.2 Selecting the Voltage Detect Operating Mode

When VDCR2<VD1MOD> is set to "0", the voltage detection operation mode is set to generate INTVLTD interrupt request signals. When VDCR2<VD1MOD> is set to "1", the operation mode is set to generate voltage detection reset signals.

- When the operation mode is set to generate INTVLTD interrupt signals (VDCR2<VD1MOD>="0")

  When VDCR2<VD1EN>="1", an INTVLTD interrupt request signal is generated when the supply voltage (VDD) falls to the detection voltage (VD1LVL).

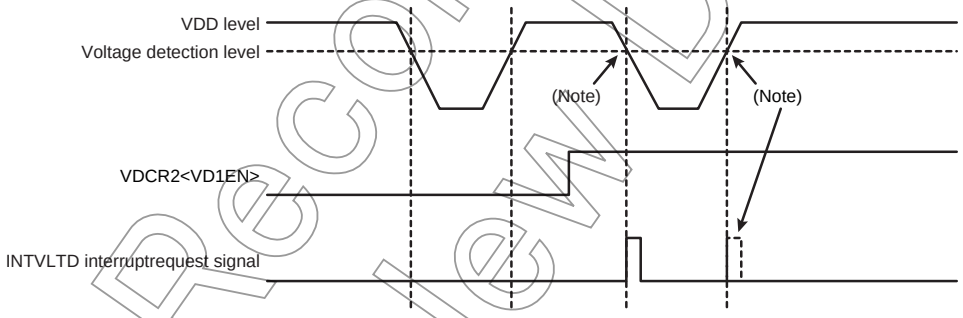


Figure 7-2 Voltage Detection Internal Request

Note: Since the comparators used for voltage detection do not have a hysteresis structure, INTVLTD interrupt request signals may be generated frequently when the supply voltage (VDD) is close to the detection voltage (VD1LVL). INTVLTD interrupt request signals may be generated not only when the supply voltage falls to the detection voltage but also when it rises to the detection voltage.

- When the operation mode is set to generate voltage detection reset signals (VDCR2<VD1MOD>="1")

  When VDCR2<VD1EN> = "1", a voltage detection reset signal is generated when the supply voltage (VDD) becomes lower than the detection voltage (VD1LVL).

  VDCR1 and VDCR2 are initialized by a power-on reset or an external reset input only. A voltage detection reset signal is generated continuously as long as the supply voltage (VDD) is lower than the detection voltage (VDxLVL).

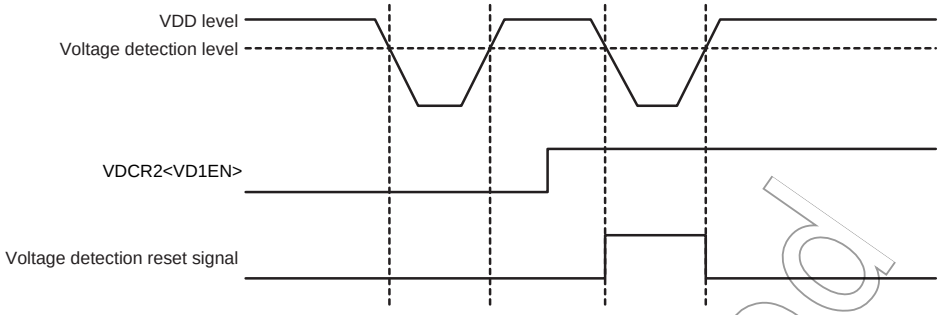


Figure 7-3 Voltage Detection Reset Signal

7.3.3 Detection voltage level section

The detection voltage level is selected by programming the VDCR1<VD1LVL> bits.

7.3.4 Voltage detection flag and voltage detection status flag

The voltage detect flag (VDCR1<VD1F>) and voltage detect status flag (VDCR1<VD1SF>) indicate the relation between the supply voltage (VDD) and the detection voltage (VD1LVL), i.e, whether VDD is above or equal to VD1LVL or VDD is below VD1LVL.

When VDCR2<VD1EN>=1, a drop of the supply voltage (VDD) below the detection voltage (VD1LVL) causes the VDCR1<VD1F> flag to be set. This flag remains set until it is cleared by software. VDCR1<VD1F> is not cleared to 0 when the supply voltage (VDD) becomes equal to or higher than the detection voltage (VD1LVL).

Once VDCR1<VD1F> has been set to 1, this state is retained even if VDCR2<VD1EN> is cleared to 0. VDCR1<VD1F> can only be cleared by writing a 0.

When VDCR2<VD1EN>=1, a drop of the supply voltage (VDD) below the detection voltage (VD1LVL) also causes the VDCR1<VD1SF> flag to be set to 1. When VDD rises above VD1LVL, the VDCR1<VD1SF> flag is cleared to 0.

Unlike the VDCR1<VD1F> flag, the VDCR1<VD1SF> flag does not remain set and changes its state according to the relation between VDD and VD1LVL.

Note: Depending on the voltage detect timing, the voltage detect status flag (VDCR1<VD1SF>) may be set one machine cycle before the voltage detect flag (VDCR1<VD1F>) is set.

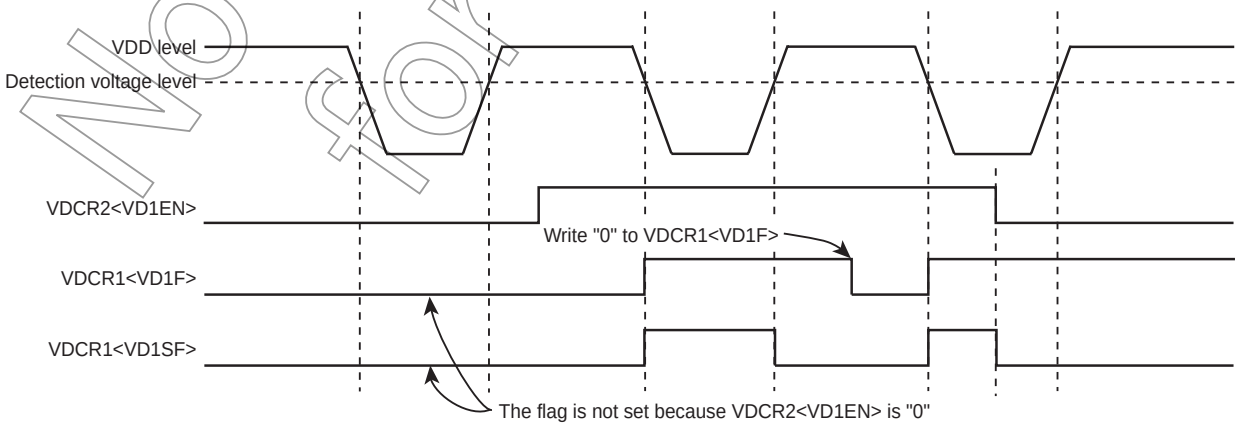


Figure 7-4 Change of voltage detection flag and voltage detection status flag

7.4 Setting of register

7.4.1 Setting Procedure for Generate an interrupt

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

1. Clear the INTVLTD interrupt enable flag <EF4> to 0.
2. Select the detection voltage by programming the VDCR1<VD1LVL> bits.
3. Clear the VDCR2<VD1MOD> bit to 0 to generate an INTVLTD interrupt upon detection of a low-voltage condition.
4. Set the VDCR2<VD1EN> bit to 1 to enable voltage detect operation.
5. Wait for 5 μs until the voltage detection circuit stabilizes.
6. Ensure that the VDCR1<VD1SF> is 0.
7. Clear the INTVLTD interrupt latch <IL4> to 0 and set the interrupt enable flag <EF4> to 1 to enable interrupts.

Note: When the supply voltage (VDD) is close to the detection voltage (VD1LVL), voltage detection request signals may be generated frequently. If this may pose any problem, execute appropriate wait processing depending on fluctuations in the system power supply and clear the interrupt latch before returning from the INTVLTD interrupt service routine.

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

1. Clear the INTVLTD interrupt enable flag <EF4> to 0.
2. Clear the VDCR2<VD1EN> bit to 0 to disable voltage detect operation.

Note: If the voltage detection circuit is disabled without clearing INTVLTD interrupt enable flag <EF4>, unexpected interrupt request may occur.

7.4.2 Setting procedure to generate a reset

The following shows the setting procedure for generating a voltage detect 1 reset signal upon detection of a low-voltage condition.

1. Clear the INTVLTD interrupt enable flag <EF4> to 0.
2. Select the detection voltage by programming the VDCR1<VD1LVL> bits.
3. Clear the VDCR2<VD1MOD> bit to 0 to generate an INTVLTD interrupt upon detection of a low-voltage condition.
4. Set the VDCR2<VD1EN> bit to 1 to enable voltage detect operation.
5. Wait for 5 μs until the voltage detection circuit stabilizes.
6. Ensure that the VDCR1<VD1SF> bit is 0.
7. Clear the VDCR1<VD1F> bit to 0.
8. Set the VDCR2<VD1MOD> bit to 1 to generate a voltage detect x reset signal upon detection of a low-voltage condition.

Note 1: The VDCR1 and VDCR2 are only initialized by a power-on reset or an external reset input. Therefore, at the time when a low-voltage detect reset is released, if the supply voltage (VDD) is found to be below the detection voltage (VD1LVL) before VDCR2<VD1EN> is cleared to 0, a voltage detect reset will immediately be generated.

Note 2: The voltage detect reset signal remains asserted while the supply voltage (VDD) is below the detection voltage (LD1LVL).

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

1. Clear the INTVLTD interrupt enable flag <EF4> to 0.
2. Clear the VDCR2<VD1MOD> bit to 0 to generate an INTVLTD interrupt upon detection of a low-voltage condition.
3. Clear the VDCR2<VD1EN> bit to 0 to disable the voltage detect operation.

Note: If the voltage detection circuit is disabled without clearing INTVLTD interrupt enable flag <EF4>, unexpected interrupt request may occur.

8. Time Base Timer (TBT) and Divider Output ($\overline{DVO}$)

8.1 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 8-2).

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

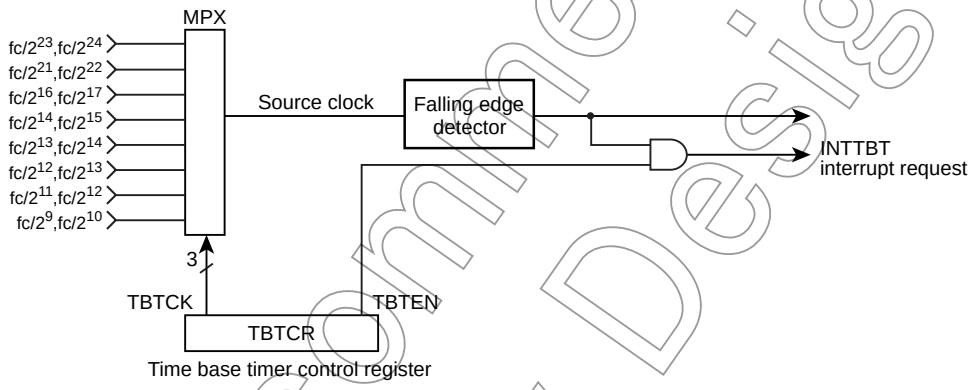


Figure 8-1 Time Base Timer configuration

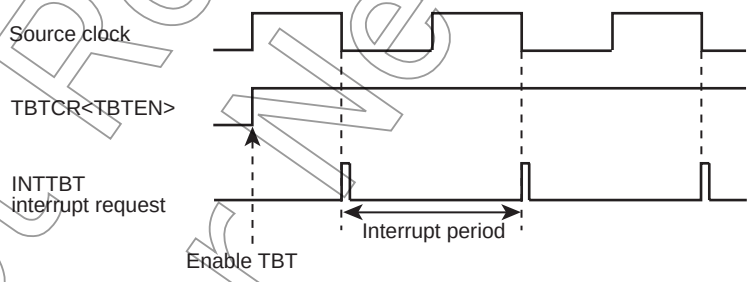


Figure 8-2 Time Base Timer Interrupt

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)	(DVOEN)	(DVOCK)		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 on TBTCR register.

Table 8-1 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

8.2 Divider Output ($\overline{DVO}$)

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

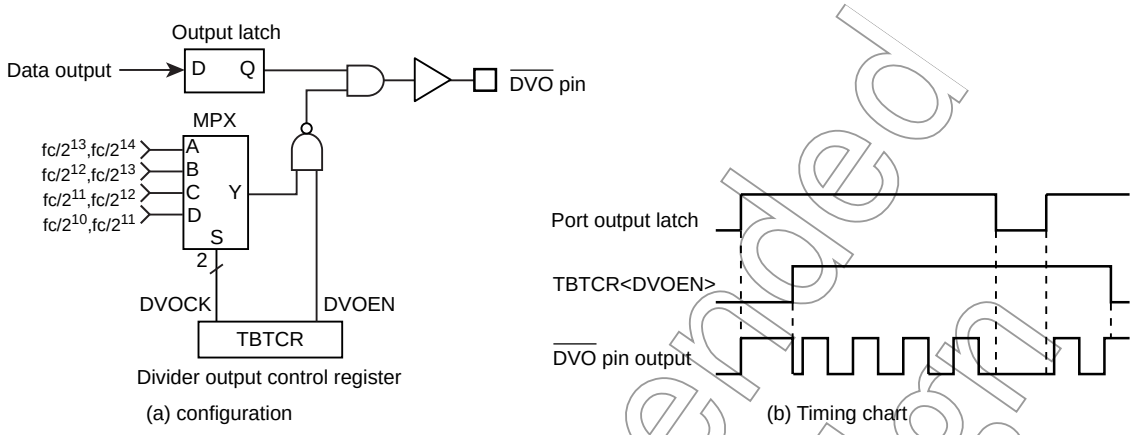


Figure 8-3 Divider Output

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

Time Base Timer Control Register

	7	6	5	4	3	2	1	0	
TBTCR (00036H)	DVOEN	DVOCK		"0"	(TBTEN)	(TBTCK)			(Initial value: 0000 0000)

DVOEN	Divider output enable / disable	0: Disable 1: Enable		R/W
DVOCK	Divider Output ($\overline{DVO}$) frequency selection: [Hz]	NORMAL, IDLE Mode		R/W
		DV1CK=0	DV1CK=1	
		00	$fc/2^{13}$	$fc/2^{14}$
		01	$fc/2^{12}$	$fc/2^{13}$
		10	$fc/2^{11}$	$fc/2^{12}$
		11	$fc/2^{10}$	$fc/2^{11}$

Note 1: Selection of divider output frequency (DVOCK) must be made while divider output is disabled (DVOEN="0"). Also, in other words, when changing the state of the divider output frequency from enabled (DVOEN="1") to disable(DVOEN="0"), do not change the setting of the divider output frequency.

Note 2: In case of using $\overline{DVO}$ output, set output mode by P1CR register after setting the related port output latch to "1" by P1DR register.

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

Note 4: Be sure to write "0" to TBTCR Register bit 4.

Example : 2.44 kHz pulse output (fc = 20.0 MHz)

```
                Port setting

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

Table 8-2 Divider Output Frequency (Example : fc = 20.0 MHz)

DVOCK	Divider Output Frequency [Hz]	
	NORMAL, IDLE Mode	
	DV1CK=0	DV1CK=1
00	2.4415 k	1.22075 k
01	4.8825 k	2.4415 k
10	9.765 k	4.8825 k
11	19.5325 k	9.765 k

9. 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 non-maskable 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.

9.1 Watchdog Timer Configuration

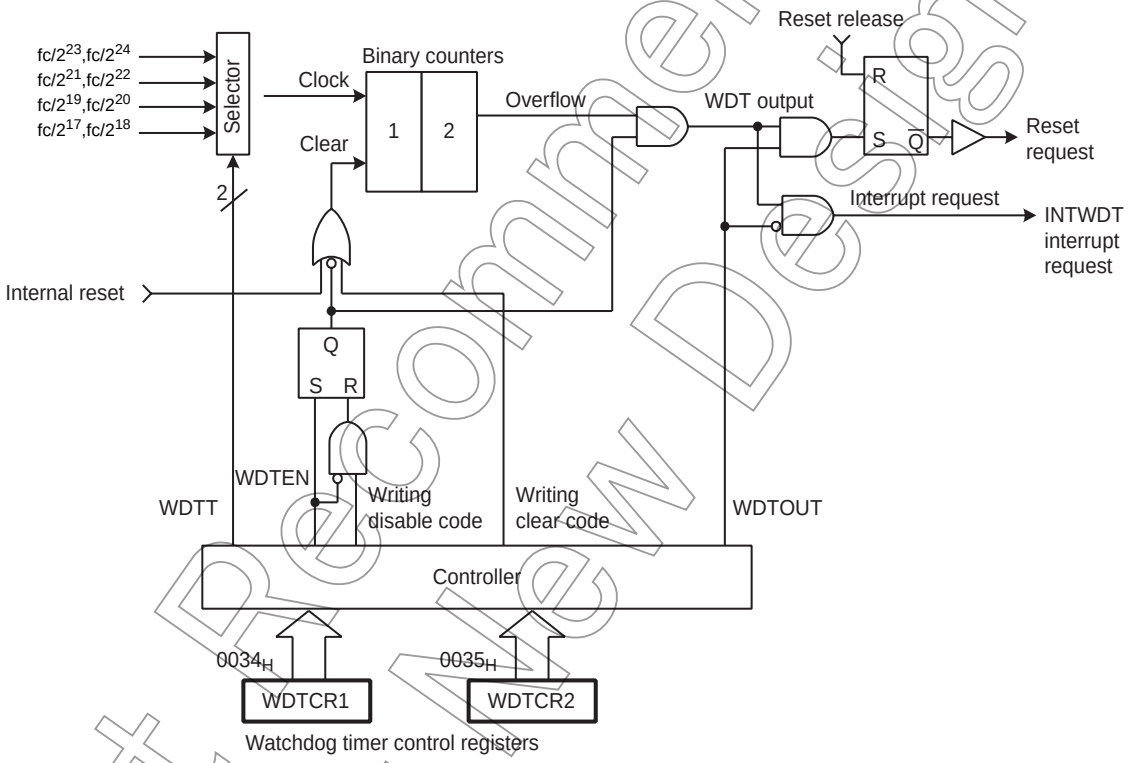


Figure 9-1 Watchdog Timer Configuration

9.2 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.

9.2.1 Malfunction Detection Methods Using the Watchdog Timer

The CPU malfunction is detected, as shown below.

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

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

The watchdog timer temporarily stops counting in the STOP mode including the warm-up or IDLE mode, and automatically restarts (continues counting) when the STOP/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 activate the STOP mode, disable the watchdog timer or clear the counter immediately before entering the STOP mode. After clearing the counter, clear the counter again immediately after the STOP mode is inactivated.

Note 5: To clear WDTCR1<WDTEN>, set the register in accordance with the procedures shown in “9.2.3 Watchdog Timer Disable”.

Note 6: 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 7: 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.

9.2.2 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.

9.2.3 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 counter
LDW     (WDTCR1), 0B101H    : WDTEN ← 0, WDTCR2 ← Disable code
EI                          : IMF ← 1
```

Table 9-1 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.

9.2.4 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
```

9.2.5 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).

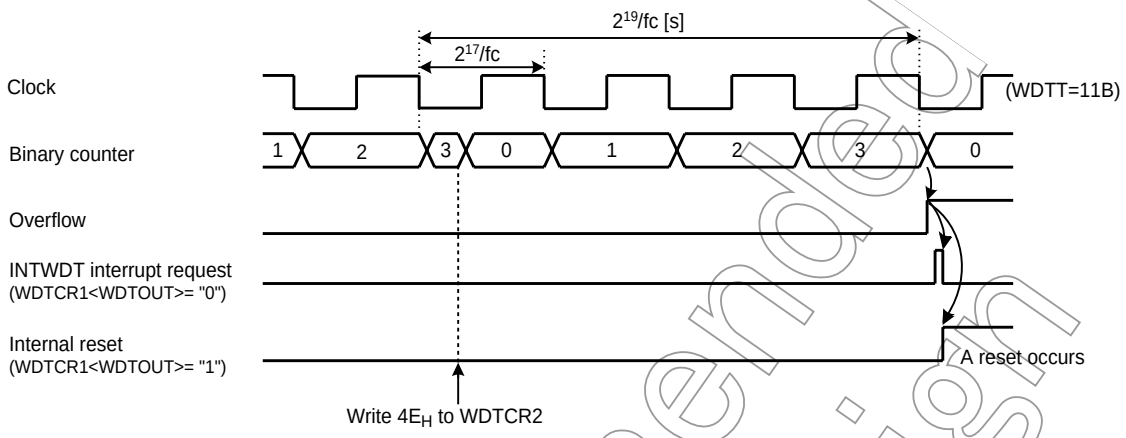


Figure 9-2 Watchdog timer Interrupt and Reset

10. 16-Bit TimerCounter 1 (TC1)

10.1 Configuration

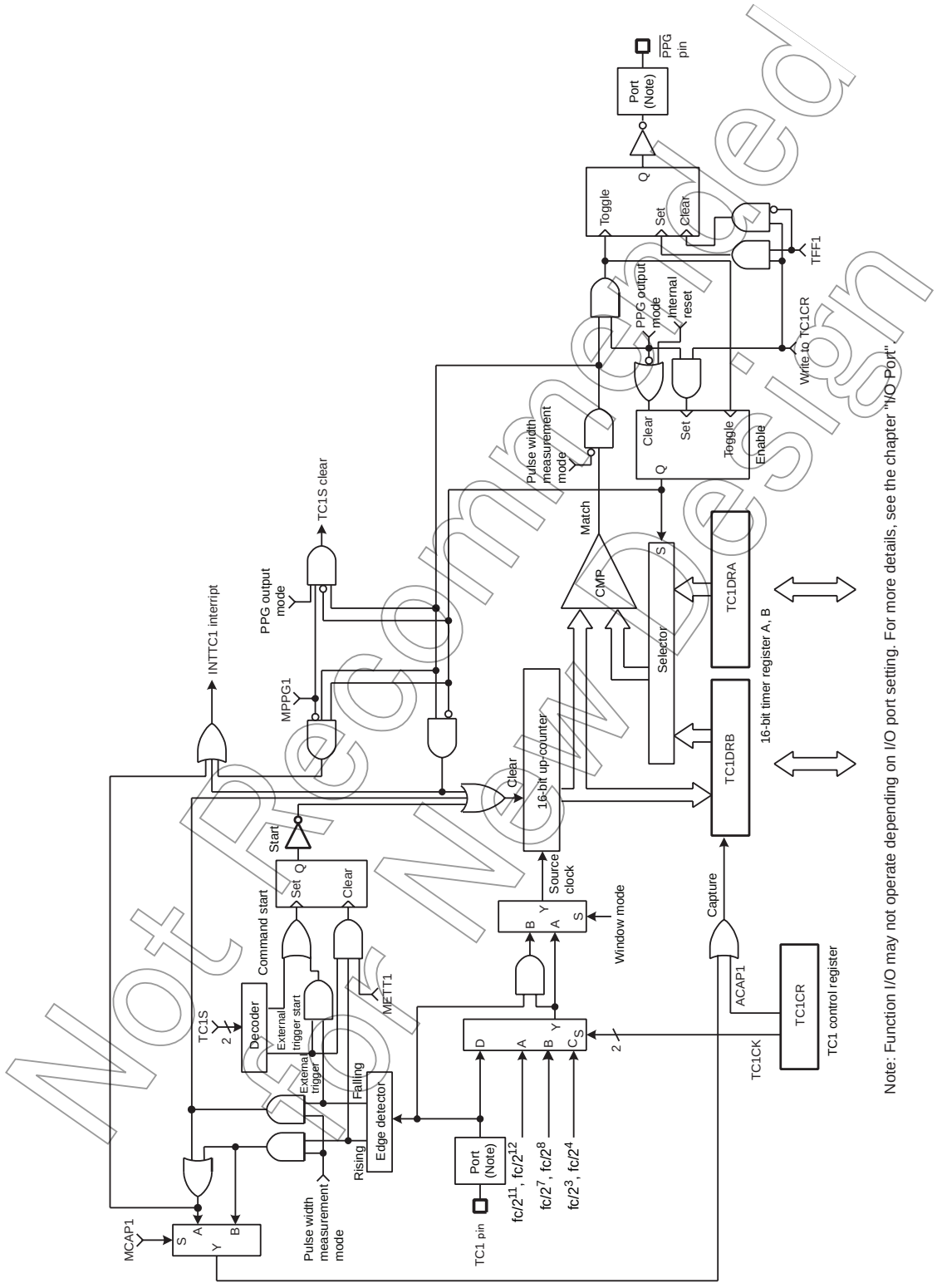


Figure 10-1 TimerCounter 1 (TC1)

10.2 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/Write (Write enabled only in the PPG output mode)							

TimerCounter 1 Control Register

	7	6	5	4	3	2	1	0	
TC1CR (000FH)	TFF1	ACAP1 MCAP1 METT1 MPPG1	TC1S		TC1CK		TC1M		Read/Write (Initial value: 0000 0000)

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

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

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

Note 3: To set the mode, source clock, PPG output control and timer F/F control, write to TC1CR during TC1CR<TC1S>=00. Set the timer F/F1 control until the first timer start after setting the PPG mode.

Note 4: Auto-capture can be used only in the timer, event counter, and window modes.

Note 5: To set the timer registers, the following relationship must be satisfied.
TC1DRA > TC1DRB > 1 (PPG output mode), TC1DRA > 1 (other modes)

Note 6: Set TC1CR<TFF1> to “0” in the mode except PPG output mode.

Note 7: Set TC1DRB after setting TC1CR<TC1M> to the PPG output mode.

Note 8: When the STOP mode is entered, the start control (TC1CR<TC1S>) is cleared to “00” automatically, and the timer stops. After the STOP mode is exited, set the TC1CR<TC1S> to use the timer counter again.

Note 9: Use the auto-capture function in the operative condition of TC1. A captured value may not be fixed if it's read after the execution of the timer stop or auto-capture disable. Read the capture value in a capture enabled condition.

Note 10: Since the up-counter value is captured into TC1DRB by the source clock of up-counter after setting TC1CR<ACAP1> to "1". Therefore, to read the captured value, wait at least one cycle of the internal source clock before reading TC1DRB for the first time.

10.3 Function

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

10.3.1 Timer mode

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

Table 10-1 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
```

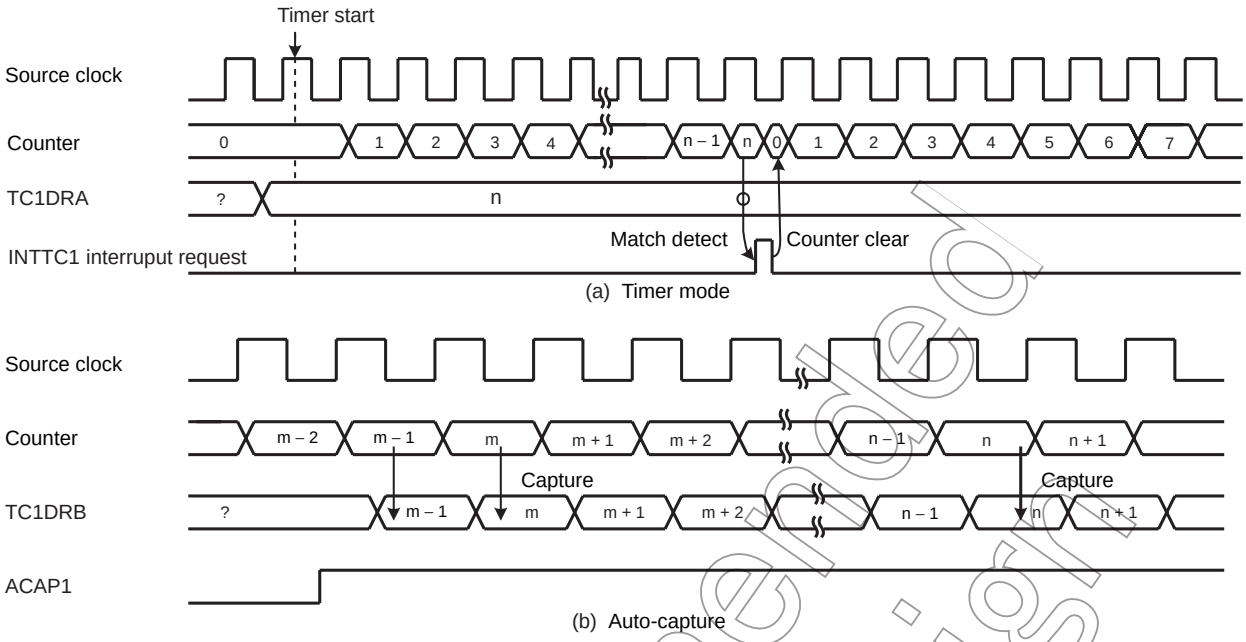


Figure 10-2 Timer Mode Timing Chart

10.3.2 External Trigger Timer Mode

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

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

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

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

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

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

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

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

Since the TC1 pin input has the noise rejection, pulses of 4/fc [s] or less are rejected as noise. A pulse width of 12/fc [s] or more is required to ensure edge detection.

Example 1 :Generating an interrupt 1 ms after the rising edge of the input pulse to the TC1 pin
(fc = 20 MHz, CGCR<DV1CK> = "1")

```
LDW     (TC1DRA), 007DH     ; 1ms ÷ 2^7/fc = 7DH
DI                          ; IMF= "0"
SET     (EIRD). 2           ; Enables INTTC1 interrupt
EI                          ; IMF= "1"
LD      (TC1CR), 00001000B  ; Selects the source clock and mode
LD      (TC1CR), 00111000B  ; Starts TC1 external trigger, METT1 = 0
```

Example 2 :Generating an interrupt when the low-level pulse with 4 ms or more width is input to the TC1 pin
(fc = 20 MHz, CGCR<DV1CK> = "1")

```
LDW     (TC1DRA), 0138H     ; 4 ms ÷ 2^8/fc = 0138H
DI                          ; IMF= "0"
SET     (EIRD). 2           ; Enables INTTC1 interrupt
EI                          ; IMF= "1"
LD      (TC1CR), 00000100B  ; Selects the source clock and mode
LD      (TC1CR), 01110100B  ; Starts TC1 external trigger, METT1 = 0
```

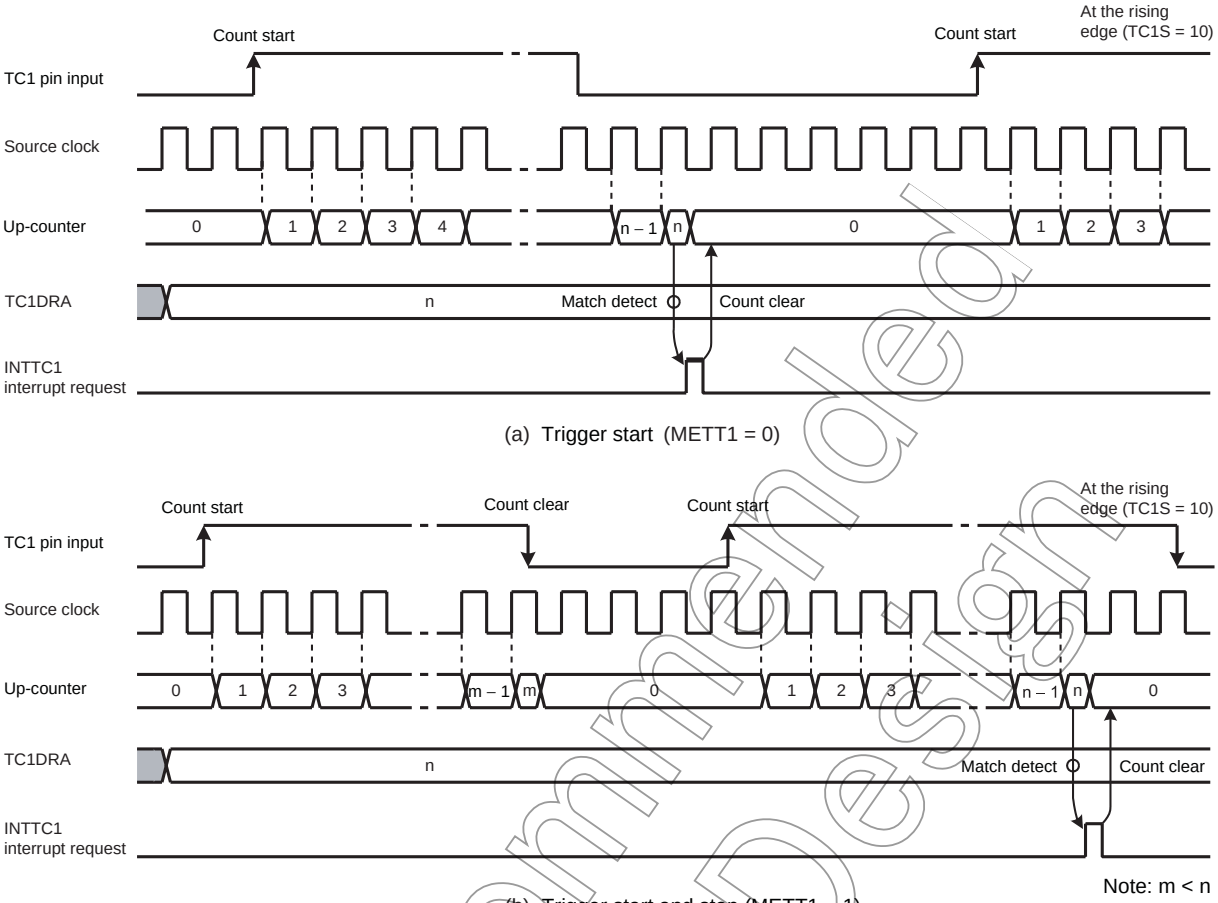


Figure 10-3 External Trigger Timer Mode Timing Chart

10.3.3 Event Counter Mode

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

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

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

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

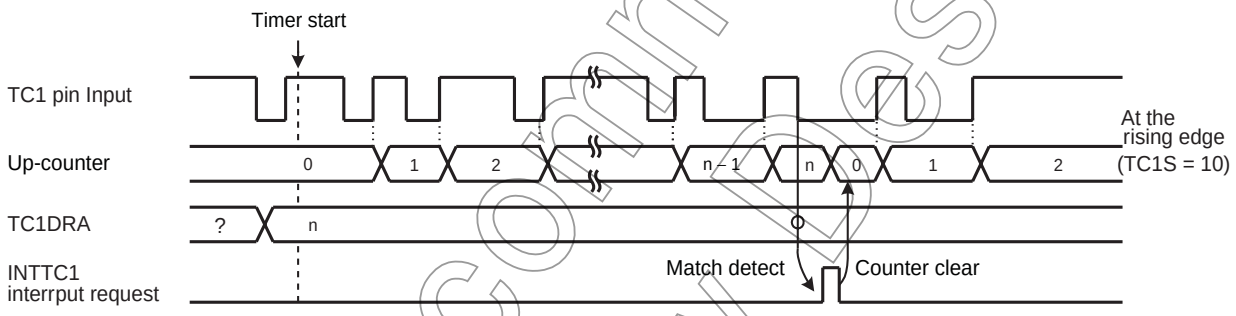


Figure 10-4 Event Counter Mode Timing Chart

Table 10-2 Input Pulth Width to TC1 Pin

	Minimum Pulse Width [s]
	NORMAL, IDLE Mode
High-going	2^3/fc
Low-going	2^3/fc

10.3.4 Window Mode

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

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

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

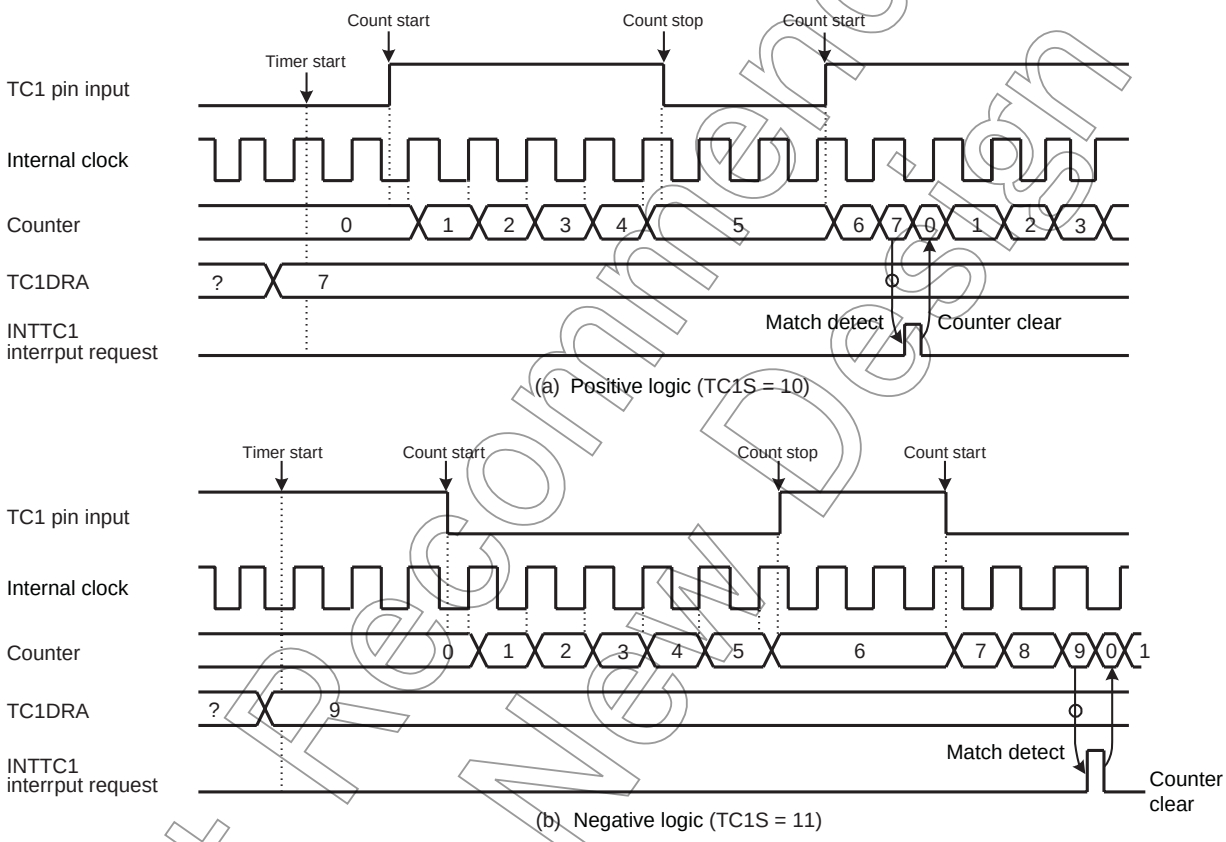


Figure 10-5 Window Mode Timing Chart

10.3.5 Pulse Width Measurement Mode

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

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

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

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

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

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

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

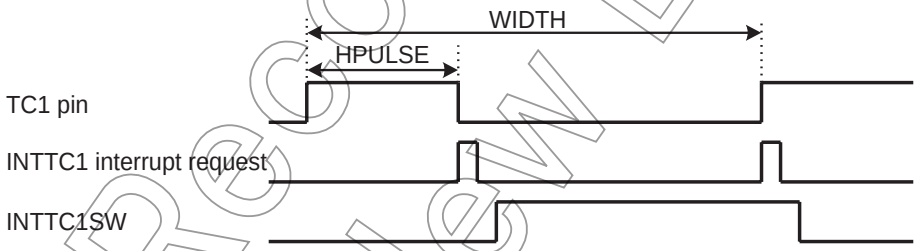
Note 1: The captured value must be read from TC1DRB until the next trigger edge is detected. If not read, the captured value becomes a don't care. It is recommended to use a 16-bit access instruction to read the captured value from TC1DRB.

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

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

Example :Duty measurement (resolution fc/2^7 [Hz], CGCR<DV1CK> = “0”)

```
                CLR     (INTTC1SW). 0       ; INTTC1 service switch initial setting
                                            ; Address set to convert INTTC1SW at each INTTC1
                LD      (TC1CR), 00000110B  ; Sets the TC1 mode and source clock
                DI                          ; IMF= “0”
                SET     (EIRD). 2           ; Enables INTTC1
                EI                          ; IMF= “1”
                LD      (TC1CR), 00100110B  ; Starts TC1 with an external trigger at MCAP1 = 0
                :
PINTTC1:        CPL     (INTTC1SW). 0       ; INTTC1 interrupt, inverts and tests INTTC1 service switch
                JRS     F, SINTTC1
                LD      A, (TC1DRBL)        ; Reads TC1DRB (High-level pulse width)
                LD      W,(TC1DRBH)
                LD      (HPULSE), WA        ; Stores high-level pulse width in RAM
                RETI
SINTTC1:        LD      A, (TC1DRBL)        ; Reads TC1DRB (Cycle)
                LD      W,(TC1DRBH)
                LD      (WIDTH), WA         ; Stores cycle in RAM
                :
                RETI                        ; Duty calculation
                :
VINTTC1:        DW      PINTTC1             ; INTTC1 Interrupt vector
```



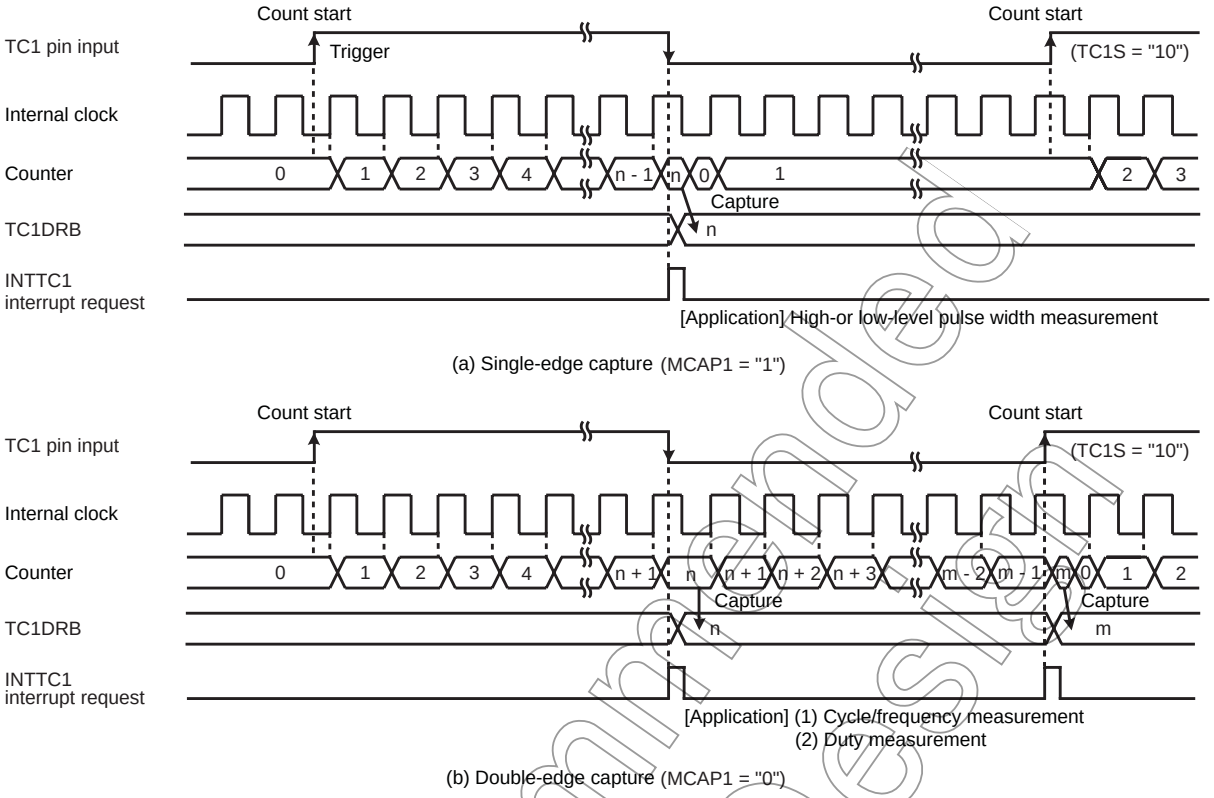


Figure 10-6 Pulse Width Measurement Mode

10.3.6 Programmable Pulse Generate (PPG) Output Mode

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

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

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

  When TC1CR<TC1S> is cleared to "00" during PPG output, the $\overline{PPG}$ pin retains the level immediately before the counter stops.

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

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

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

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

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

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

TC1DRA > TC1DRB

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

Example :Generating a pulse which is high-going for 800 µs and low-going for 200 µs
(fc = 20 MHz, CGCR<DV1CK> = "0")

```
                Setting port
LD      (TC1CR), 10001011B      ; Sets the PPG mode, selects the source clock
LDW     (TC1DRA), 04E2H         ; Sets the cycle (1 ms ÷ 2⁴/fc µs = 04E2H)
LDW     (TC1DRB), 00FAH         ; Sets the low-level pulse width (200 µs ÷ 2⁴/fc = 00FAH)
LD      (TC1CR), 10010111B      ; Starts the timer
```

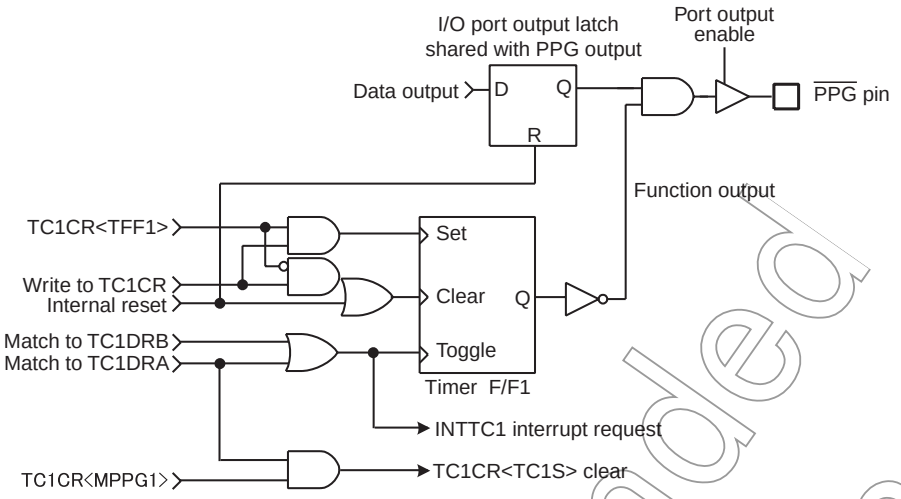


Figure 10-7 $\overline{\text{PPG}}$ Output

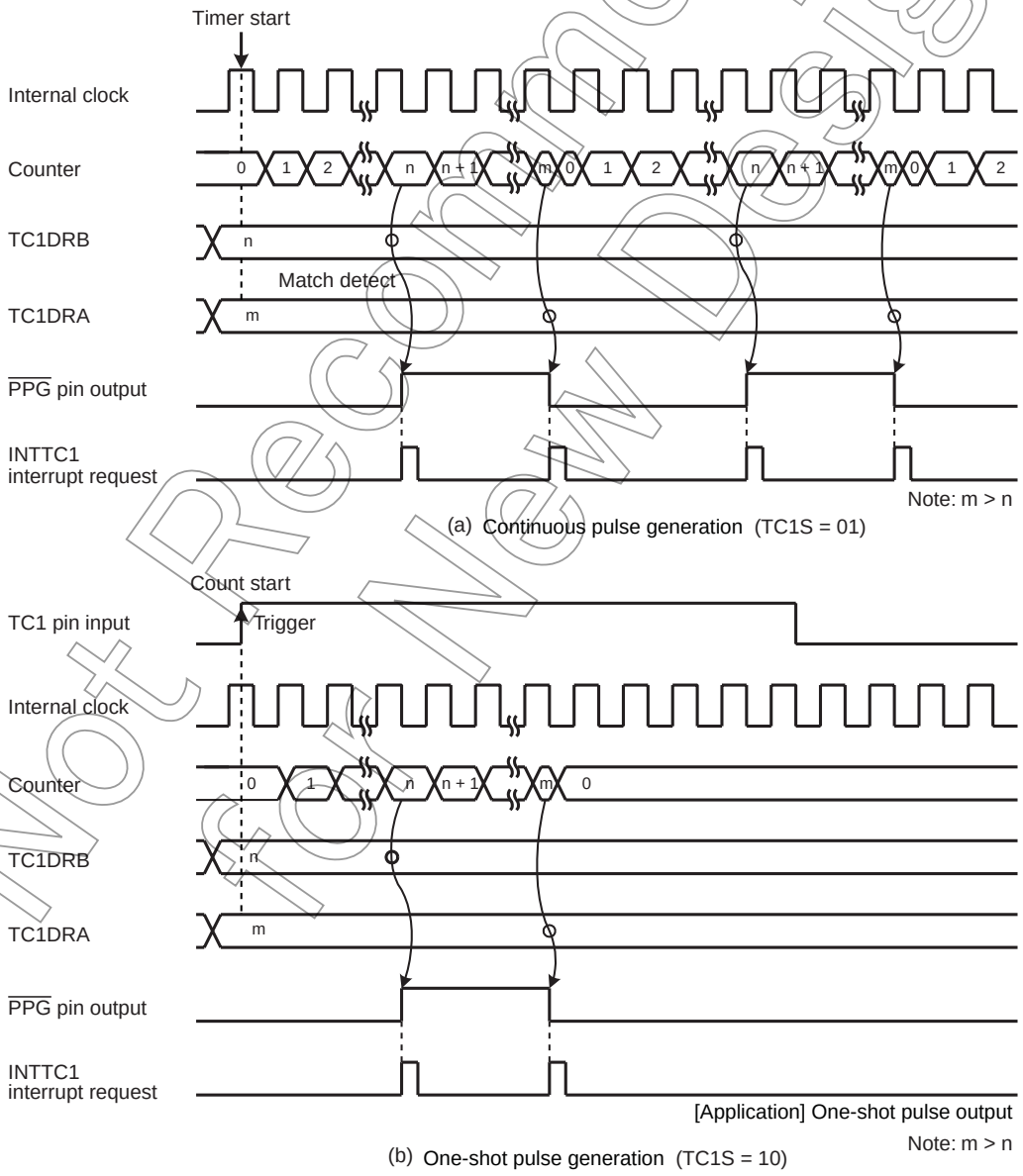


Figure 10-8 PPG Mode Timing Chart

11. 16-Bit Timer (CTC)

11.1 Configuration

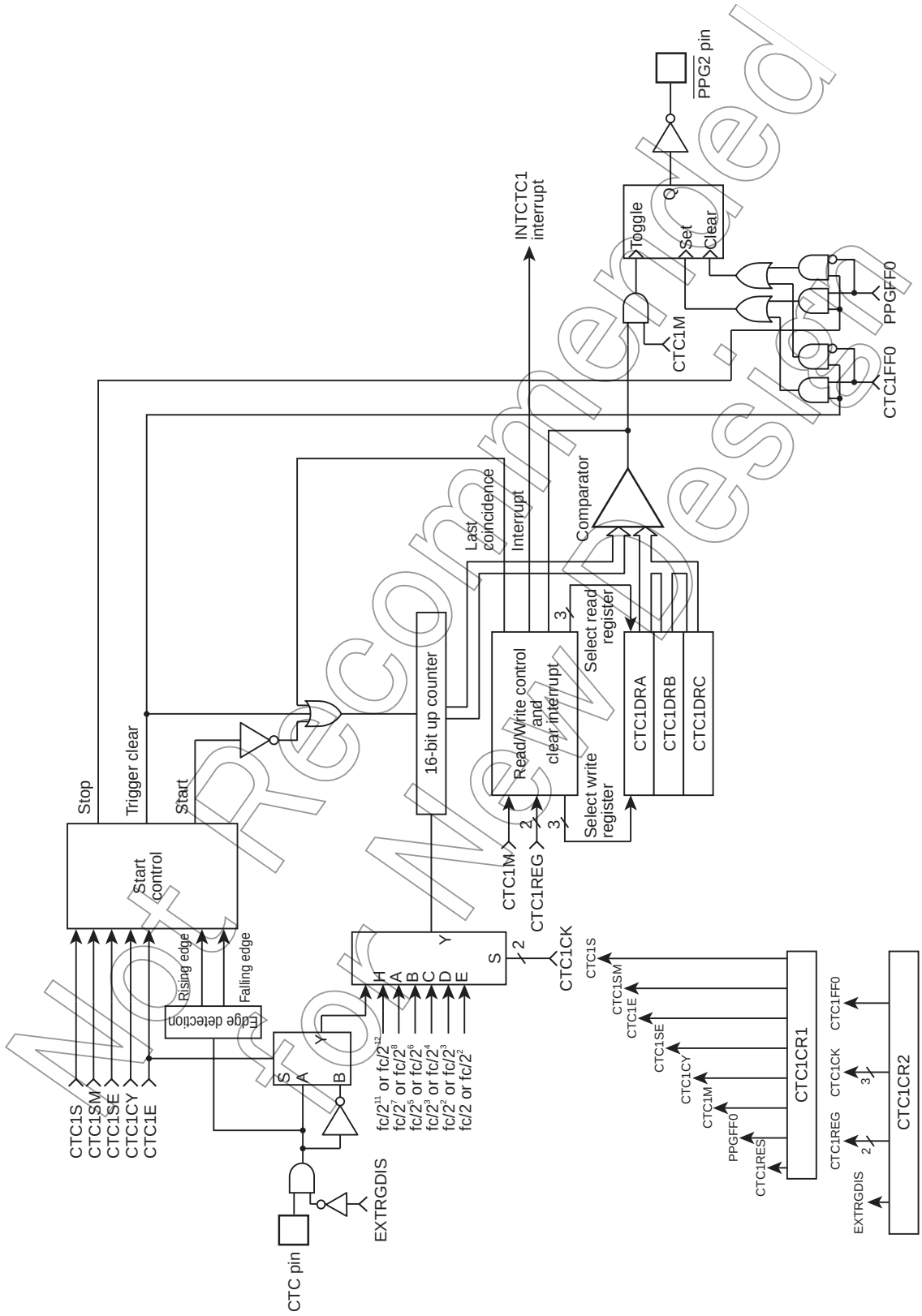


Figure 11-1 CTC Block Diagram

11.2 Control

Compare timer/counter 1 is controlled using Compare timer/counter 1 Control Registers (CTC1CR1 and CTC1CR2), as well as three 16-bit Timer Registers (CTC1DRA, CTC1DRB, and CTC1DRC).

Compare Timer Registers (CTC1DRH: 00017h, CTC1DRL: 00016h)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
CTC1DRA	CTC1DRAH								CTC1DRAL								Write only (Initial value: ******** ********)
CTC1DRB	CTC1DRBH								CTC1DRBL								Write only (Initial value: ******** ********)
CTC1DRC	CTC1DRCH								CTC1DRCL								Write only (Initial value: ******** ********)

Note: CTC1DRA, CTC1DRB, and CTC1DRC are write-only registers and must not be used with any of the read-modify-write instructions such as SET, CLR, etc.

Compare Timer/Counter 1 Control Registers (CTC1CR2: 00015h, CTC1CR1: 00014h)

	7	6	5	4	3	2	1	0	
CTC1CR1 lower address	CTC1RES	PPGFF0	CTC1M	CTC1CY	CTC1SE	CTC1E	CTC1SM	CTC1S	R/W (Initial value: 00000000)
CTC1CR2 upper address	*	EX-TRGDIS	CTC1REG		CTC1CK			CTC1FF0	R/W (Initial value: *0000000)

Note 1: *: Don't care

Note 2: The CTC1CR1<CTC1RES> is 0 when read.

Note 3: Use the LDW instruction for write to the CTC1DR H/L Registers. Set a value equal to or greater than 2.

Note 4: Write to CTC1DR H/L A, B, and C Registers as many as set with the CTC1CR2 Register CTC1REG bit.

Note 5: Data are written to CTC1DR H/L Registers in order of CTC1DRA, CTC1DRB, and CTC1DRC.

Setting-up the CTC1CR1 Register

			Timer	Event	PPG	
CTC1S	Control start	0: Stop and clear counter 1: Command start	o o	o o	o o	R/W
CTC1SM	Select start	0: Software start 1: External trigger start	o o	o ×	o o	
CTC1E	Select external trigger edge	0: Enable one edge 1: Enable both edges	o o	o ×	o o	
CTC1SE	Select external trigger start edge	0: Rising edge 1: Falling edge	o o	o o	o o	
CTC1CY	Select cycle	0: Successive 1: One shot	o o	o ×	o o	
CTC1M	Set operation mode	0: Timer/Event counter modes 1: PPG (programmable pulse generator) output mode				
PPGFF0	Select PPG output	0: Forward output immediately after start 1: Reverse output immediately after start				
CTC1RES	Reset all	0: Normal operation 1: CTC1 reset				

Setting-up the CTC1CR2 Register

CTC1FF0	Control timer output F/F0	0: Clear 1: Set					R/W
CTC1CK	Select timer/counter clock source Unit: Hz		NORMAL and IDLE Modes				
			DV1CK = 0	DV1CK = 1	Timer	Event	PPG
		000	$fc/2^{11}$	$fc/2^{12}$	o	−	×
		001	$fc/2^{7}$	$fc/2^{8}$	o	−	×
		010	$fc/2^{5}$	$fc/2^{6}$	o	−	×
		011	$fc/2^{3}$	$fc/2^{4}$	o	−	×
		100	$fc/2^{2}$	$fc/2^{3}$	o	−	o Note3
		101	fc/2	$fc/2^{2}$	o	−	o
		110	−	−	×	×	×
		111	External clock input (CTC1 pin input)		-	○	×
CTC1REG	Set registers used by timer/ counter	00: CTC1DRA 1REG 01: CTC1DRA + CTC1DRB 2REG 10: CTC1DRA + CTC1DRB + CTC1DRC 3REG 11: Reserved					
EXTRGDIS	External trigger input Note4	0: Enable external trigger input 1: Disable external trigger input					

Note 1: fc: Clock [Hz]

Note 2: Make sure the timer/counter is idle (CTC1CR1<CTC1SM, CTC1S> = 00) before setting operation mode, edge, start, source clock, external trigger timer mode control, and PPG output control.

Note 3: When DV1CK=1, CTC1CR2<CTC1CK>=100 cannot be used.

Note 4: When CTC1 input is not used in the CTC1 timer, external trigger input must be disabled (CTC1CR2<EXTRGDIS> = 1) regardless of the selected mode.

Note 5: The CTC1DRB and CTC1DRC Registers cannot be accessed for write unless they are set for PPG output mode and specified with CTC1CR2<CTC1REG>.

Note 6: CTC1CR1<CTC1E> is effective only when using an external clock as trigger (CTC1CR1<CTC1SM>).

Note 7: Data must be written to as many data registers as set with CTC1CR2<CTC1REG>.

Note 8: To write data to CTC1DRA/B/C, use the LDW instruction, or use the LD instruction writing in order of L, H.

Note 9: Data register values must be written to the respective registers before starting. To modify the values after starting, write the new data within an interval from an INTCTC1 interrupt to the next INTCTC1.

Note 10: Specifying CTC1CR1<CTC1RES> = 1 causes all conditions to be reset. Even when the CTC circuit is operating, they are reset, and the PPG output becomes “0”. However, only the INTCTC1 signal is not reset if the signal is being generated.

Note 11: For event counter mode (when CTC pin input is selected in timer mode), the active edge of the external trigger to count can be selected with CTC1CR1<CTC1SE>.

Note 12: Disabling external trigger input with CTC1CR2<EXTRGDIS> creates the 0 input state.

Note 13: To stop the counter by software at trigger start, set CTC1CR2<CTC1SM, CTC1S> = 00.

Note 14: The number of registers set and the values set in the timer registers must meet the conditions shown below.

Number of Registers		Timer Register Value Conditions
CTC1REG	1 Register	CTC1DRA ≥ 2
	2 Register	CTC1DRB > CTC1DRA + 1, and CTC1DRA ≥ 2
	3 Register	CTC1DRC > CTC1DRB + 1, CTC1DRB > CTC1DRA + 1, and CTC1DRA ≥ 2

11.3 Function

Compare timer/counter 1 has three modes: timer, event counter, and programmable pulse generator output modes.

11.3.1 Timer mode with software start

In this mode, the timer/counter (16-bit counter) counts up synchronously with the internal clock. When the counter value and the set value of Compare Timer Register 1A (CTC1DRA) match, an INTCTC1 interrupt is generated and the counter is cleared. After the counter is cleared, it restarts and continues counting up.

Table 11-1 Internal Clock Source for Compare Timer/Counter 1 (Example: fc = 20 MHz)

CTC1CK	NORMAL and IDLE Modes			
	DV1CK = 0		DV1CK = 1	
	Resolution [μs]	Maximum Setting Time [s]	Resolution [μs]	Maximum Setting Time [s]
000	102.4	6.71	204.8	13.42
001	6.4	0.419	12.8	0.839
010	1.6	0.105	3.2	0.210
011	0.4	26.21 m	0.8	52.43 m
100	0.2	13.11 m	0.4	26.21 m
101	0.1	6.55 m	0.2	13.11 m
110	-	-	-	-

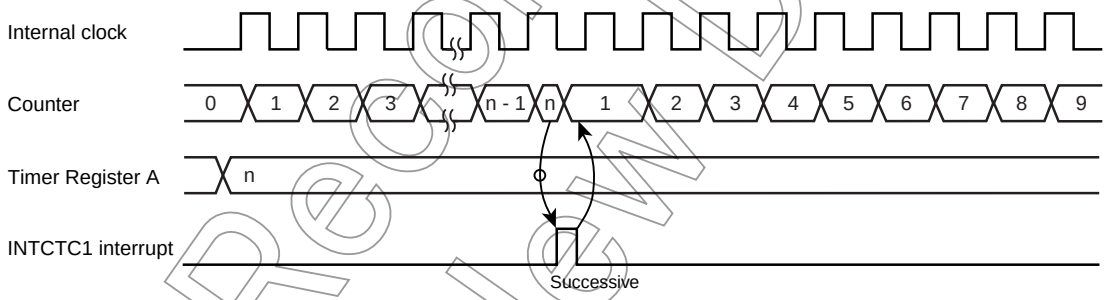


Figure 11-2 Timer Mode Timing Chart

Note: If the CTC input port (P47) is set for input mode, the timer/counter is reset by an input edge on port. When using the timer/counter as an ordinary timer, set CTC1CR2<EXTRGDIS> to 1 or set P47 for output mode.

11.3.2 Timer mode with external trigger start

In this timer mode, the timer/counter starts counting as triggered by input on CTC pin (rising or falling edge selected with CTC1CR1<CTC1SE>). The source clock is an internal clock. For successive cycles, when the counter value and the set value of the CTC1DRA Register match, an INTCTC1 interrupt is generated and the counter is cleared and then restarted. The counter is stopped by a trigger input on CTC pin and restarted by the next trigger input. For a one-shot cycle, when the counter value and the set value of the CTC1DRA Register match, an INTCTC1 interrupt is generated and the counter is cleared and stopped. The counter restarts counting up by input on CTC pin. When CTC1CR1<CTC1E> = 1, the counter is cleared and stops counting at an edge on CTC pin input opposite the active edge that triggers the counter to start counting. In this mode, an interrupt can be generated by entering a pulse which has a certain width. When CTC1CR1<CTC1E> = 0, opposite edges on CTC input are ignored.

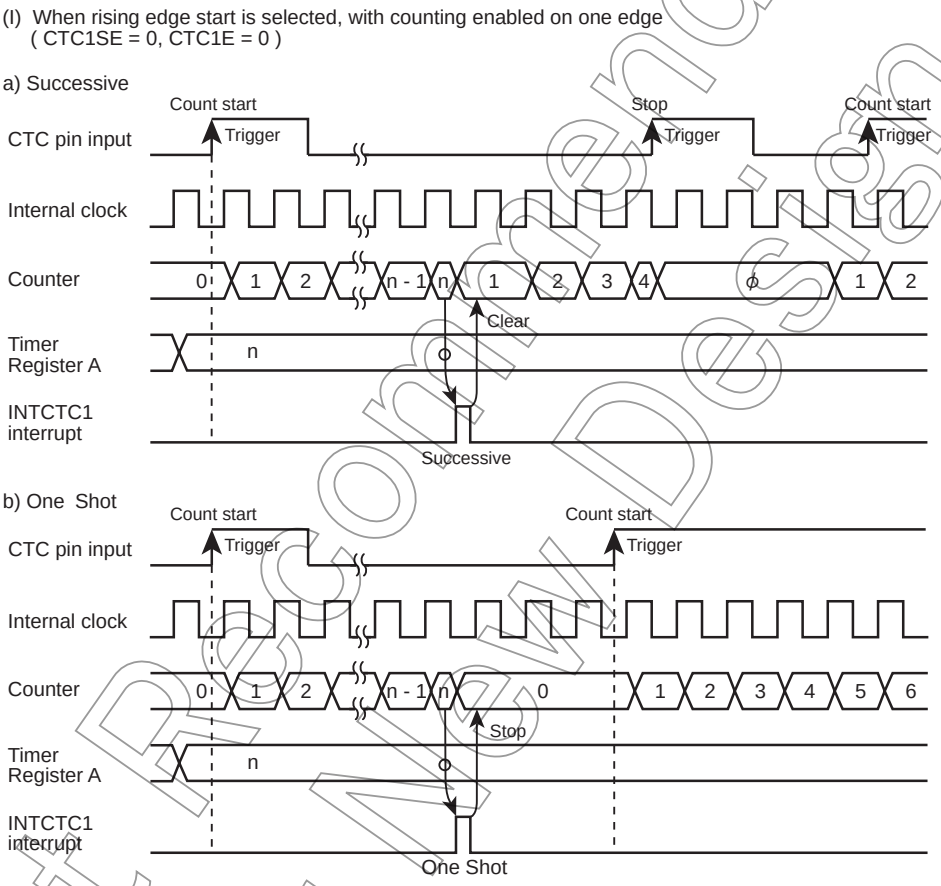


Figure 11-3 External Trigger Mode Timing Chart

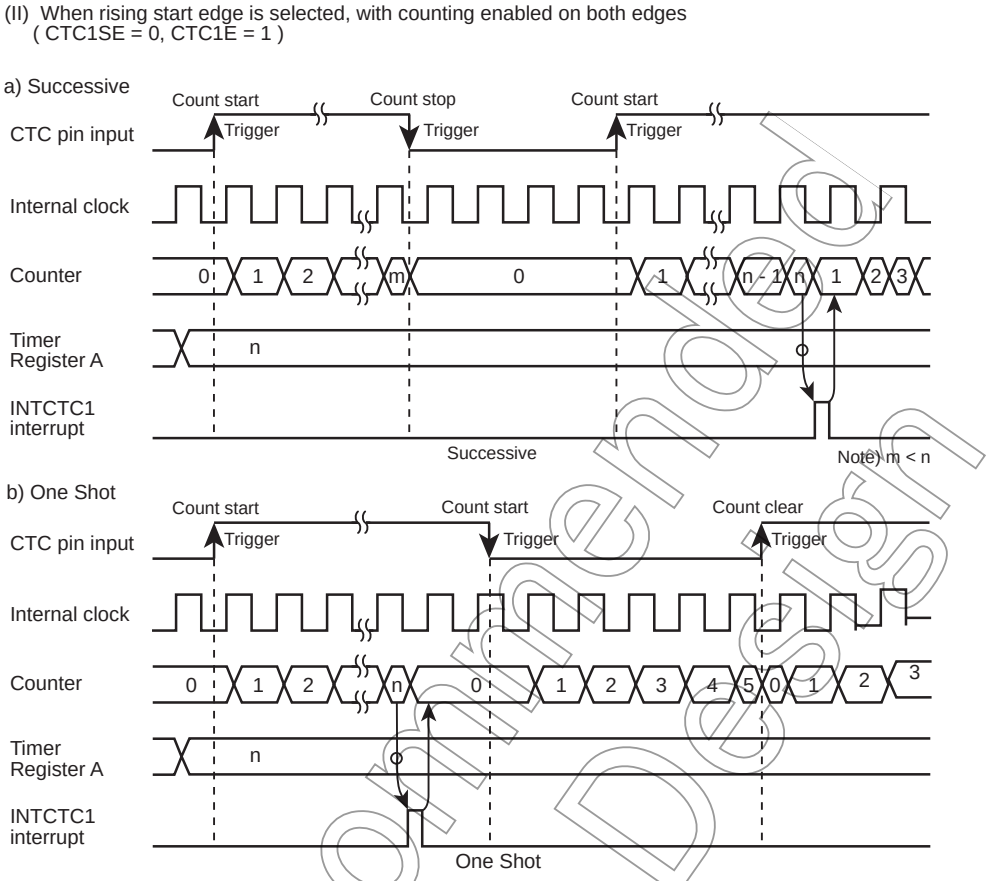


Figure 11-4 External Trigger Mode Timing Chart

11.3.3 Event counter mode

In this mode, the timer/counter counts up at the active edge on CTC pin input (rising or falling edge selected with the CTC1CR1<CTC1SE> which is provided for selecting external trigger edge). When the counter value and the set value of the CTC1DRA Register match, an INTCTC1 interrupt is generated and the counter is cleared. After the counter is cleared, it restarts and continues counting up at each edge on CTC pin input. The maximum applied frequency is shown in the table below. Because coincidence detection is made at an edge opposite the selected edge, the external clock signal on CTC pin must always be entered.

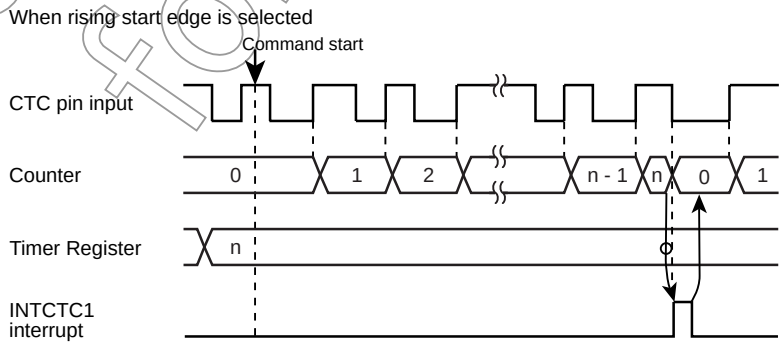


Figure 11-5 Event Counter Mode Timing Chart

Table 11-2 External Clock Source for Compare Timer/Counter 1

	NORMAL and IDLE Modes
Maximum applied frequency [Hz]	Up to fc/2^2
Minimum pulse width	2^2/fc and over

11.3.4 Programmable Pulse Generate (PPG) output mode

The timer/counter starts counting as a command or edge on CTC pin input (rising/falling edge and one/both edges respectively selected with the CTC1CR1<CTC1SE> and CTC1CR1<CTC1E>). The source clock is an internal clock. When matched with the CTC1DR A/B/C Registers, the timer output F/F corresponding to each mode is inverted. When matched with the CTC1DR A/B/C Registers next time, the timer output F/F is inverted again. An INTCTC1 interrupt request is generated when the counter value matches the maximum register value set by CTC1CR2<CTCREG>. The timer output F/F is cleared to 0 when reset. Because CTC1CR2<CTC1FF0> can be used to set the initial value for the timer output F/F, an active-high or active-low pulse whichever is desired can be output. The CTC1DRB and CTC1DRC Registers cannot be accessed for write unless they are set for PPG output mode and the registers used are selected with CTC1CR2<CTC1REG>. The number of registers set can be altered during operation. In this case, however, be sure to set the number of registers used and write values to the data registers before the next CTC1INIT1 is output after the first CTC1INIT1 output. Even when only altering the data register values while leaving the number of registers unchanged, be sure to do this within the same period of time.

Table 11-3 Internal Clock Source for Compare Timer/Counter 1 (Example: fc = 20 MHz)

CTC1CK	NORMAL and IDLE Modes			
	DV1CK = 0		DV1CK = 1	
	Resolution [μs]	Maximum Setting Time [s]	Resolution [μs]	Maximum Setting Time [s]
000	−	−	−	−
001	−	−	−	−
010	−	−	−	−
011	−	−	−	−
100	0.2	13.11 m	−	−
101	0.1	6.55 m	0.2	13.11m
110	−	−	−	−

Note: When Port P47 is set as a CTC input port, an edge input resets the timer/counter. when PPG output mode is selected and external trigger start is not used, set CTC1CR2<EXTRGDIS> to "1" or set P47 as an output port.

(I) One register used (CTC1REG = 00)
When set to command start.

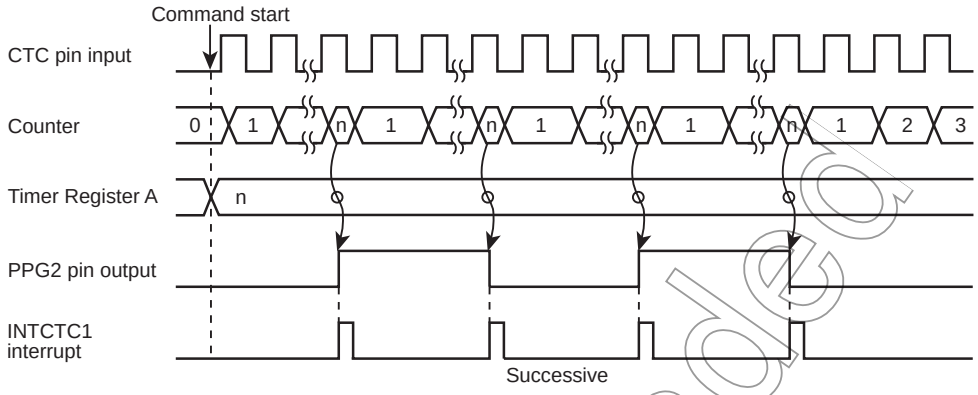
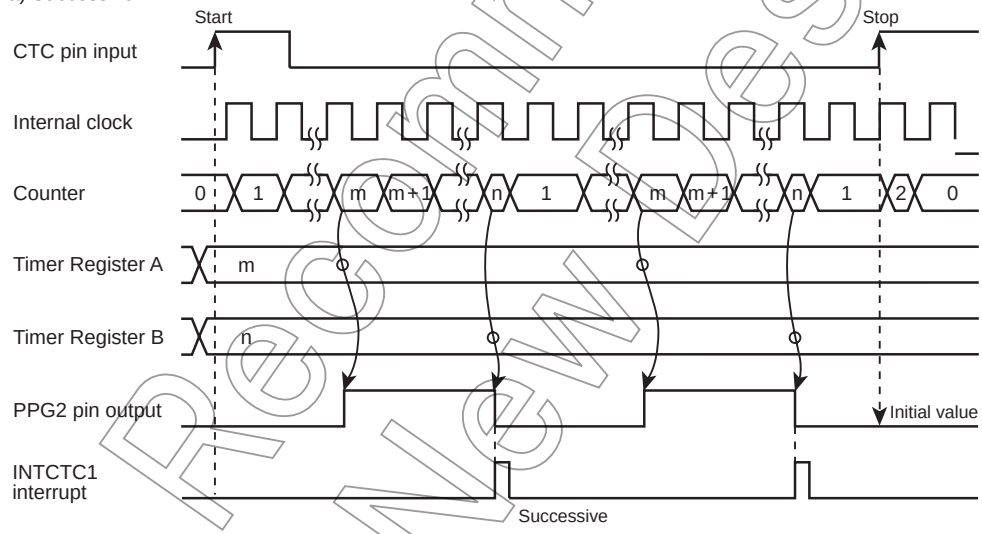


Figure 11-6 One Register Command Start Mode Timing Chart

(II) Two registers used (CTC1REG = 01)
When set to the external trigger rising edge start and the one edge enable.

a) Successive



b) One shot

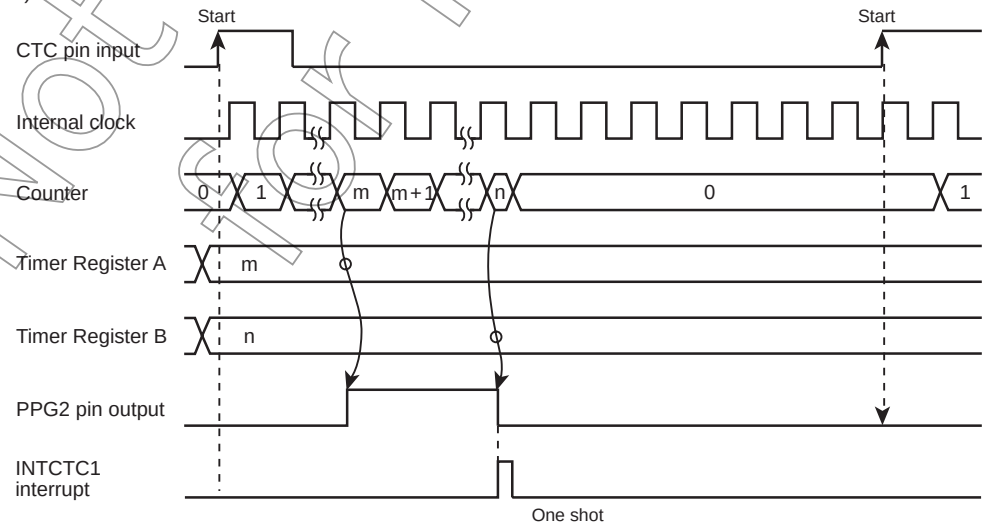


Figure 11-7 Two Register One Edge Trigger Start Mode Timing Chart

When set to the external trigger rising edge start and the both edges enable.

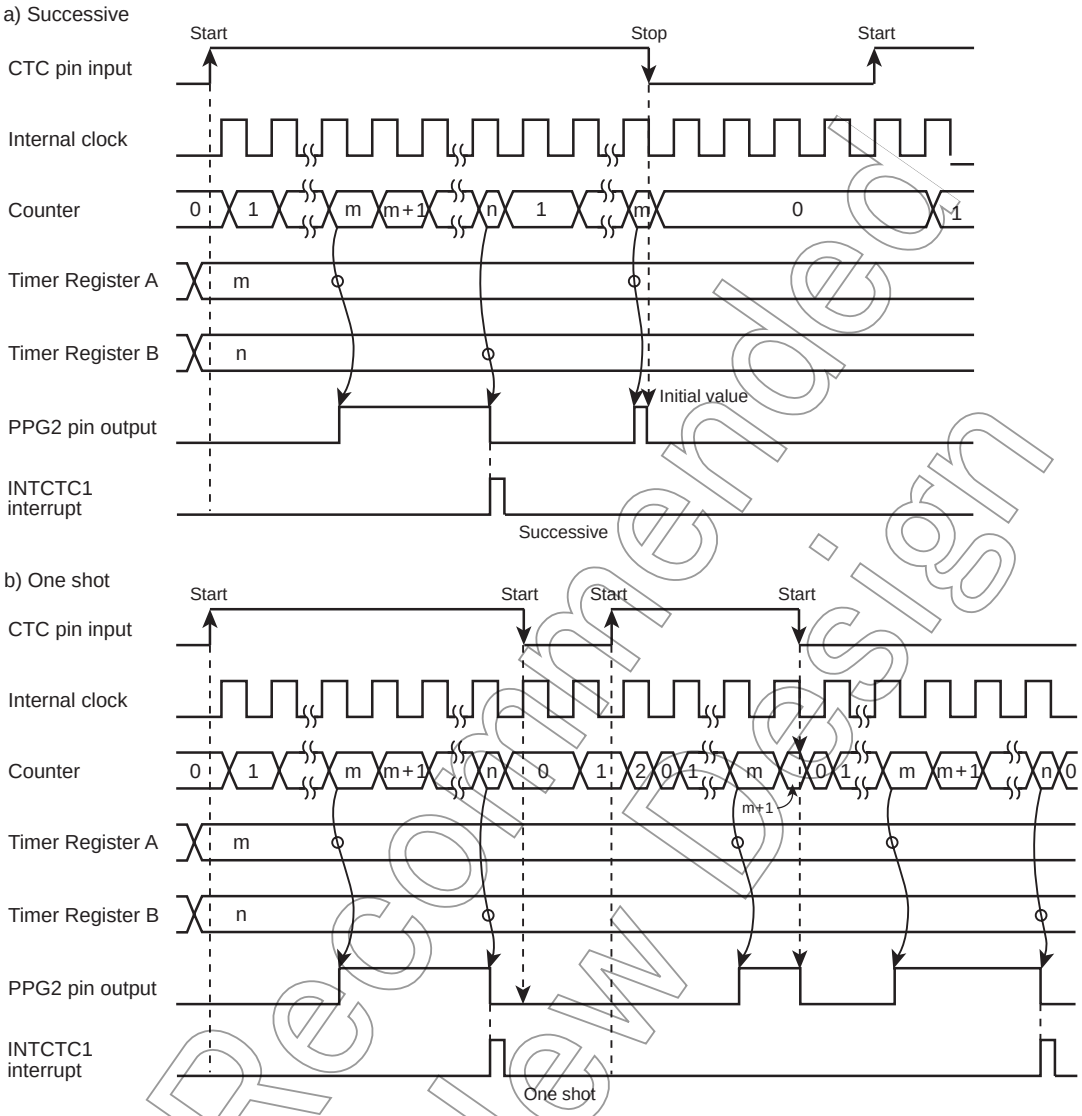
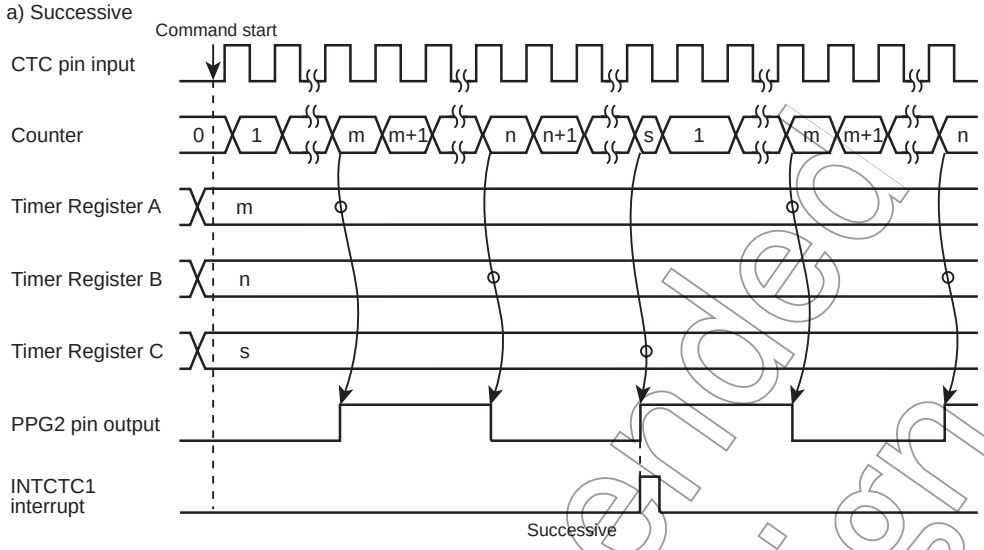


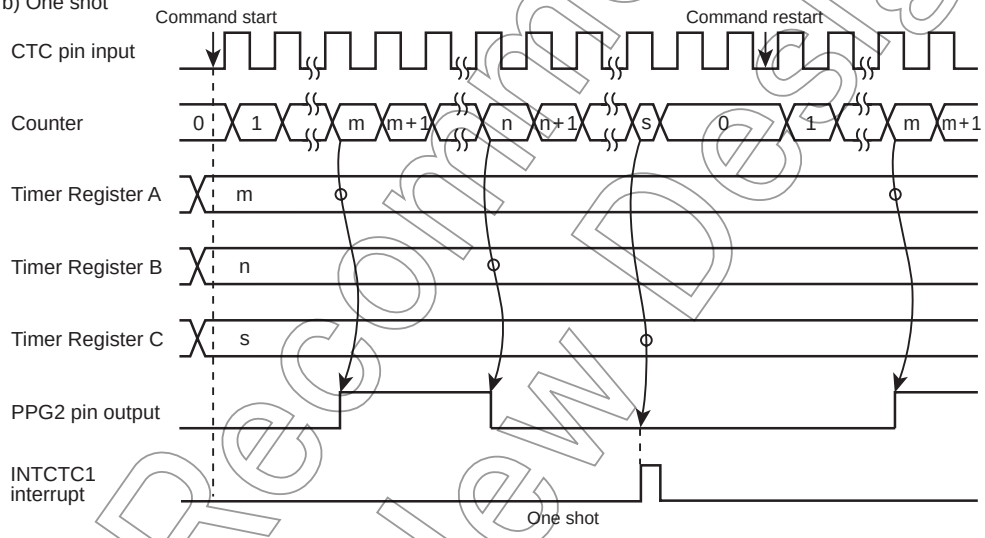
Figure 11-8 Two Register Both edges Trigger Start Mode Timing Chart

(III) Three registers used (CTC1REG = 10)
When set to command start.

a) Successive



b) One shot

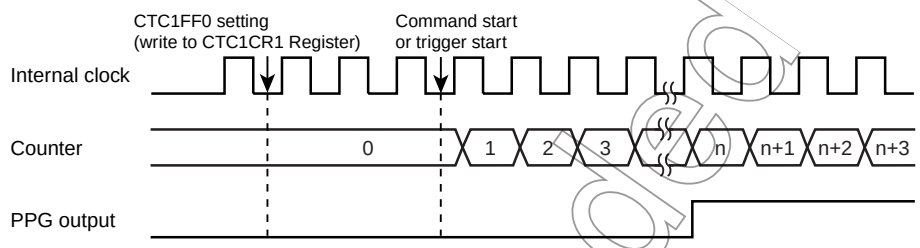
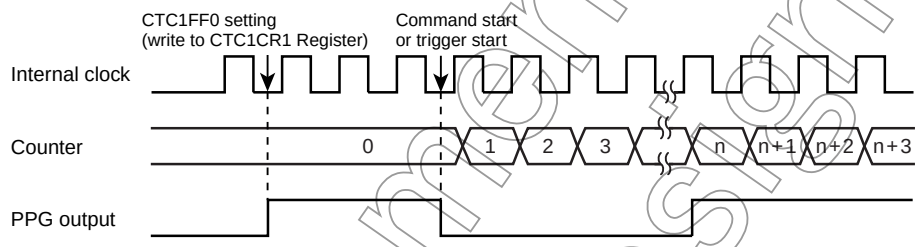
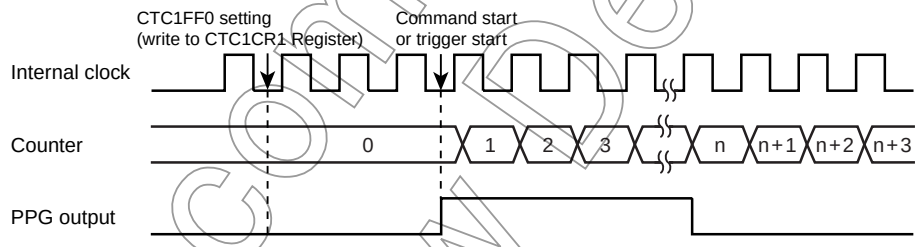
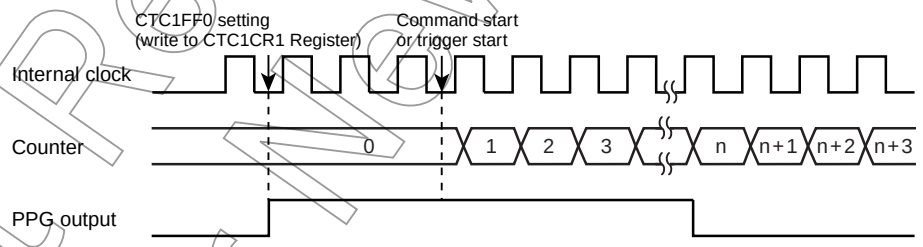


Note: In the single-shot mode, the PPG pin output is not toggled at the last register match; it stays at the value specified with CTC1CR2<CTC1FF0>.

Figure 11-9 Three Register Command Start Mode Timing Chart

Detail operation at start that varies depending on how CTC1CR2<CTC1FF0> and CTC1CR1<PPGFF0> are set during PPG output.

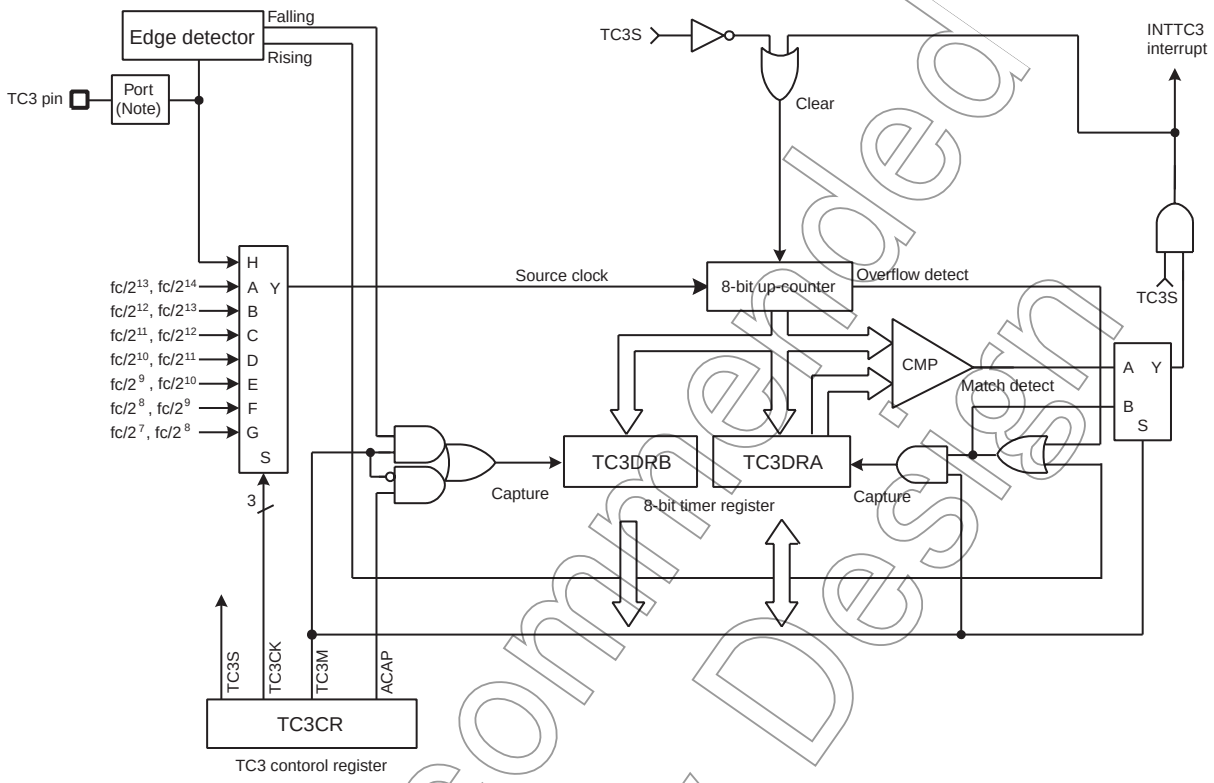
Table 11-4 Varying PPG Output Timing Depending on Settings

CTC1FF0 = 0 PPGFF0 = 0	
CTC1FF0 = 1 PPGFF0 = 0	
CTC1FF0 = 0 PPGFF0 = 1	
CTC1FF0 = 1 PPGFF0 = 1	

By changing the port-shared output for PPG output before the counter starts counting after setting CTC1CR2<CTC1FF0>, it is possible to determine the initial value of PPG output.

12. 8-Bit TimerCounter 3 (TC3)

12.1 Configuration



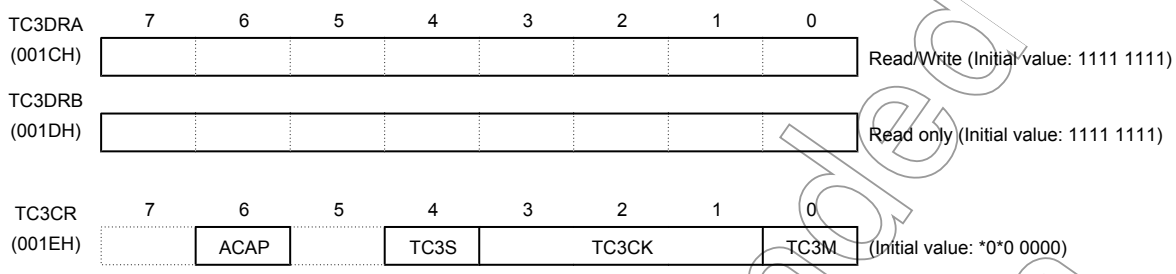
Note: Function input may not operate depending on I/O port setting. For more details, see the chapter "I/O Port".

Figure 12-1 TimerCounter 3 (TC3)

12.2 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



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	External clock (TC3pin input)		
TC3M	TC3 operating mode select	0: Timer/event counter mode 1: Capture mode			R/W

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

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

Note 3: To set the timer registers, the following relationship must be satisfied.
TC3DRA > 1 (Timer/event counter mode)

Note 4: Auto-capture (TC3CR<ACAP>) can be used only in the timer and event counter modes.

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

Note 6: Do not program TC3DRA when the timer is running (TC3CR<TC3S> = 1).

Note 7: When the STOP mode is entered, the start control (TC3CR<TC3S>) is cleared to 0 automatically, and the timer stops. After the STOP mode is exited, TC3CR<TC3S> must be set again to use the timer counter.

12.3 Function

TimerCounter 3 has three types of operating modes: timer, event counter and capture modes.

12.3.1 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 12-2)

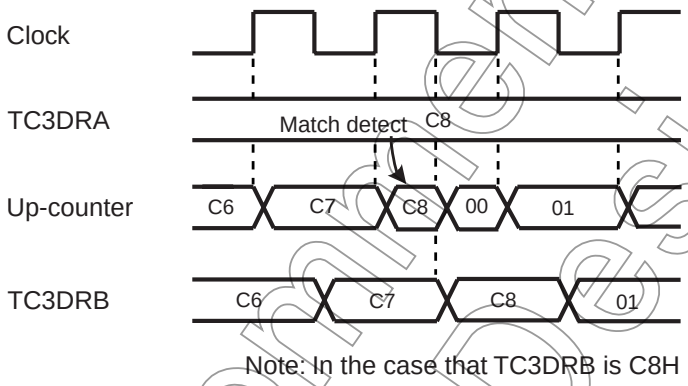


Figure 12-2 Auto-Capture Function

Table 12-1 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

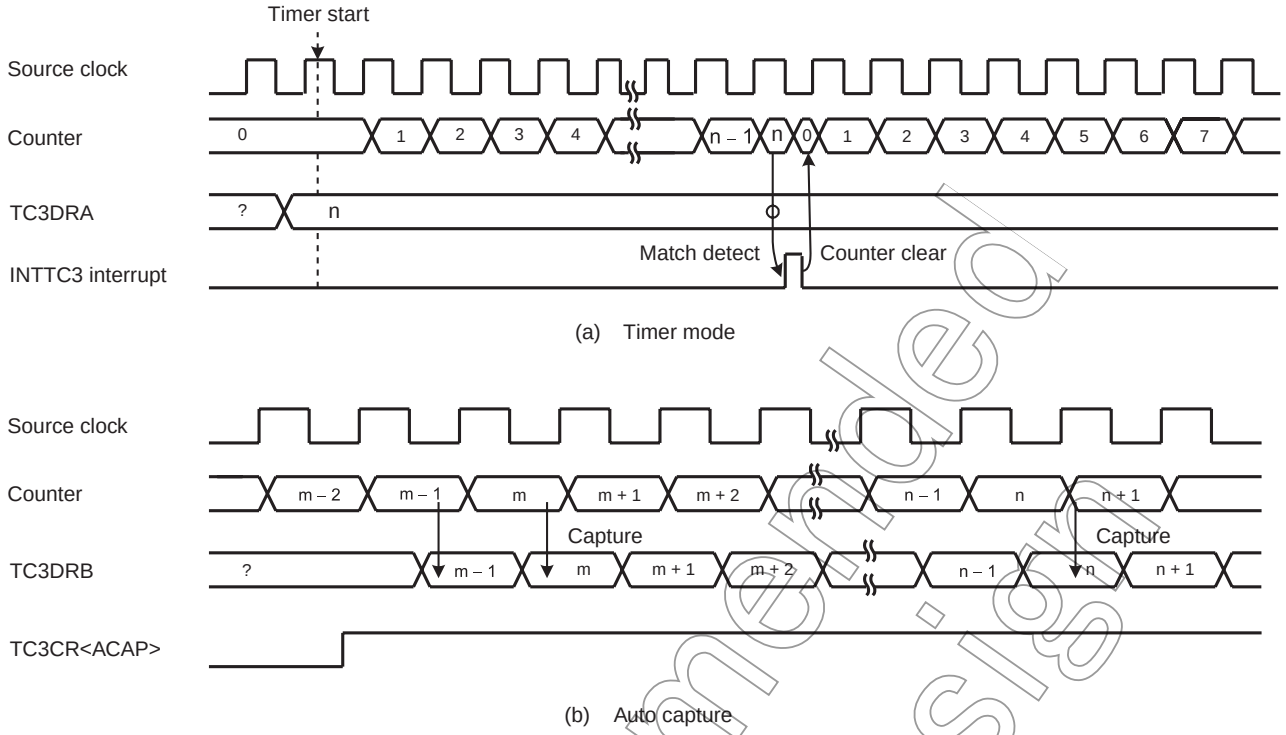


Figure 12-3 Timer Mode Timing Chart

12.3.2 Event Counter Mode

In the event counter mode, the up-counter counts up at the rising edge of the input pulse to the TC3 pin.

When a match between the up-counter and TC3DRA value is detected, an INTTC3 interrupt is generated and up-counter is cleared. After being cleared, the up-counter restarts counting at each rising edge of the input pulse to the TC3 pin. Since a match is detected at the falling edge of the input pulse to TC3 pin, an INTTC3 interrupt request is generated at the falling edge immediately after the up-counter reaches the value set in TC3DRA.

The maximum applied frequencies are shown in Table 12-2. The pulse width larger than one machine cycle is required for high-going and low-going pulses.

Setting TC3CR<ACAP> to 1 captures the up-counter value into TC3DRB with the auto-capture function. The count value during a timer operation can be checked by 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 12-2)

Example :Inputting 50 Hz pulse to TC3, and generating interrupts every 0.5 s

```
LD      (TC3CR), 00001110B      : Sets the clock mode
LD      (TC3DRA), 19H           : 0.5 s ÷ 1/50 = 25 = 19H
LD      (TC3CR), 00011110B      : Starts TC3.
```

Table 12-2 Maximum Frequencies Applied to TC3

	Minimum Pulse Width
	NORMAL, IDLE mode
High-going	2^2/fc
Low-going	2^2/fc

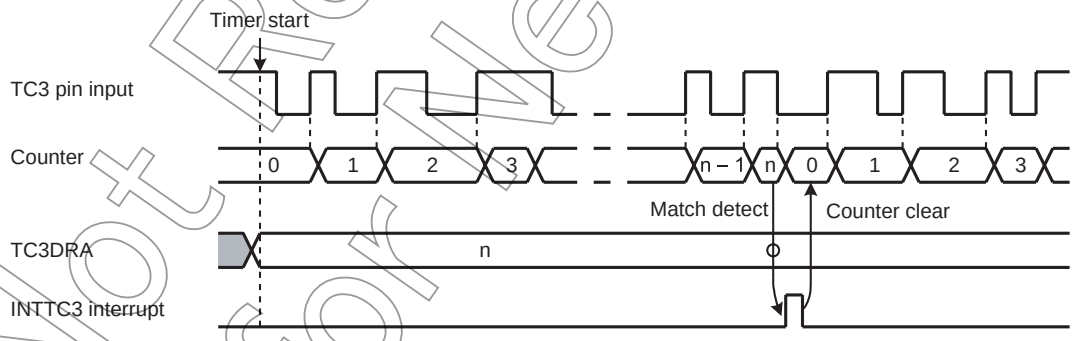


Figure 12-4 Event Counter Mode Timing Chart

12.3.3 Capture Mode

In the capture mode, the pulse width, frequency and duty cycle of the pulse input to the TC3 pin are measured with the internal clock. The capture mode is used to decode remote control signals, and identify AC50/60 Hz.

When the falling edge of the TC3 input is detected after the timer starts, the up-counter value is captured into TC3DRB. Hereafter, whenever the rising edge is detected, the up-counter value is captured into TC3DRA and the INTTC3 interrupt request is generated. The up-counter is cleared at this time. Generally, read TC3DRB and TC3DRA during INTTC3 interrupt processing. After the up-counter is cleared, counting is continued and the next up-counter value is captured into TC3DRB.

When the rising edge is detected immediately after the timer starts, the up-counter value is captured into TC3DRA only, but not into TC3DRB. The INTTC3 interrupt request is generated. When the read instruction is executed to TC3DRB at this time, the value at the completion of the last capture (FF immediately after a reset) is read.

The minimum input pulse width must be larger than one cycle width of the source clock programmed in TC3CR<TC3CK>.

The INTTC3 interrupt request is generated if the up-counter overflow (FFH) occurs during capture operation before the edge is detected. TC3DRA is set to FFH and the up-counter is cleared. Counting is continued by the up-counter, but capture operation and overflow detection are stopped until TC3DRA is read. Generally, read TC3DRB first because capture operation and overflow detection resume by reading TC3DRA.

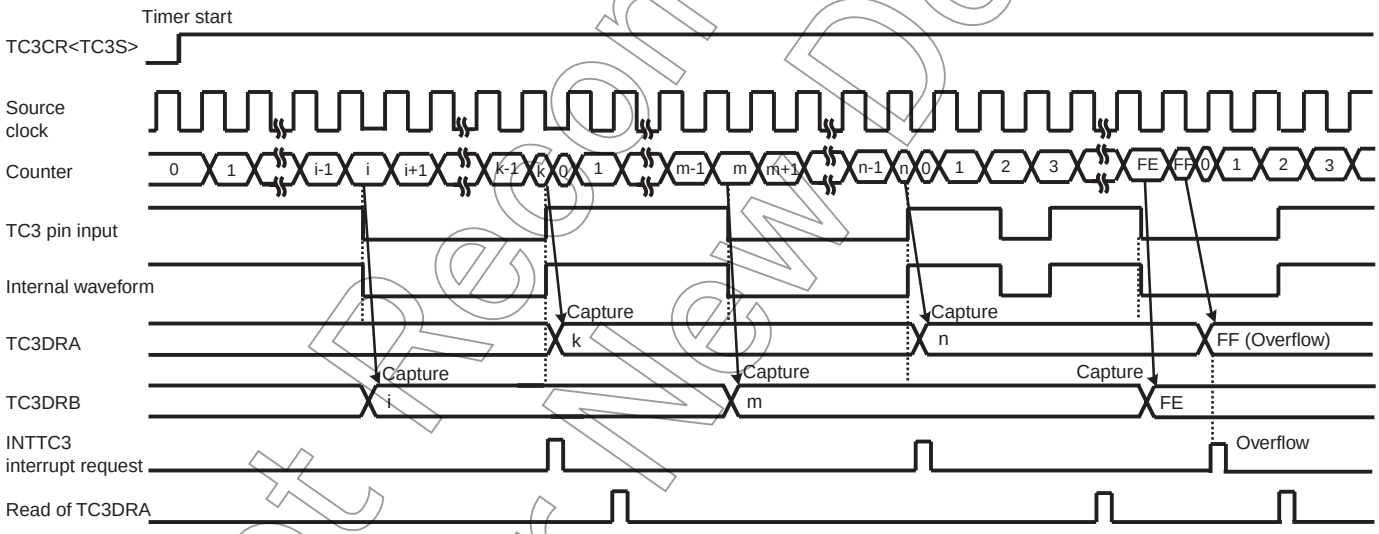
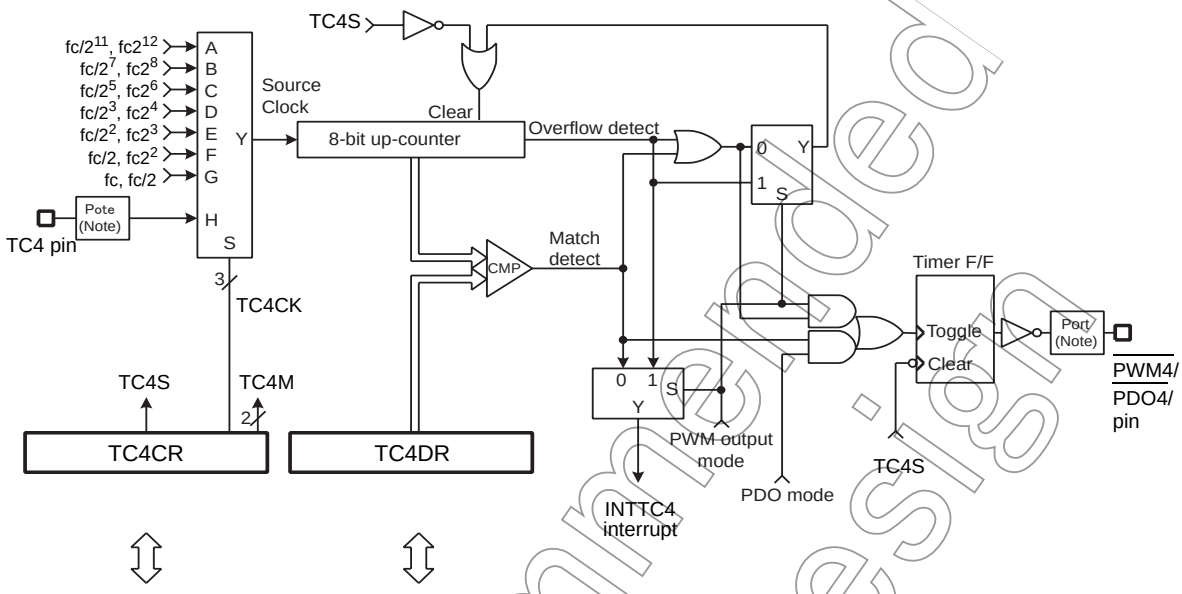


Figure 12-5 Capture Mode Timing Chart

13. 8-Bit TimerCounter 4 (TC4)

13.1 Configuration



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

Figure 13-1 TimerCounter 4 (TC4)

13.2 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	$fc/2^{14}$	$fc/2^{12}$	
		001	$fc/2^{7}$	$fc/2^{8}$	
		010	$fc/2^{5}$	$fc/2^{6}$	
		011	$fc/2^{3}$	$fc/2^{4}$	
		100	$fc/2^{2}$	$fc/2^{3}$	
		101		$fc/2^{2}$	
		110	(fc)Note8	(fc/2)Note8	
		111	External clock (TC4 pin input)		
TC4M	TC4 operating mode select	00: Timer/event counter mode 01: Reserved 10: Programmable divider output (PDO) mode 11: Pulse width modulation (PWM) output mode			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 ≤ TC4DR ≤ 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 event counter and PWM output modes are used only in the NOMAL and IDLE modes.

Note 5: When the STOP mode is entered, the start control (TC4S) is cleared to "0" automatically.

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

Note 7: In the timer, event counter and PDO modes, do not change the TC4DR setting when the timer is running.

Note 8: When the high-frequency clock fc exceeds 10 MHz, do not select the source clock of TC4CR< TC4CK> = 110.

Note 9: For available source clocks depending on the operation mode, refer to the following table.

		Timer Mode	Event Counter Mode	PDO Mode	PWM Mode
TC4CK	000	O	−	O	−
	001	O	−	O	−
	010	O	−	O	−
	011	O	−	−	O
	100	−	−	−	O
	101	−	−	−	O
	110	−	−	−	O
	111	−	O	−	×

Note: O : Available source clock

13.3 Function

TimerCounter 4 has four types of operating modes: timer, event counter, programmable divider output (PDO), and pulse width modulation (PWM) output modes.

13.3.1 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 13-1 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

13.3.2 Event Counter Mode

In the event counter mode, the up-counter counts up at the rising edge of the input pulse to the TC4 pin.

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 at rising edge of the TC4 pin. Since a match is detected at the falling edge of the input pulse to the TC4 pin, the INTTC4 interrupt request is generated at the falling edge immediately after the up-counter reaches the value set in TC4DR.

The minimum pulse width applied to the TC4 pin are shown in Table 13-2. The pulse width larger than two machine cycles is required for high- and low-going pulses.

Note: The event counter mode can used in the NORMAL and IDLE modes only.

Table 13-2 External Source Clock for TimerCounter 4

	Minimum Pulse Width
	NORMAL, IDLE mode
High-going	2^3/fc
Low-going	2^3/fc

13.3.3 Programmable Divider Output (PDO) Mode

The programmable divider output (PDO) mode is used to generated a pulse with a 50% duty cycle by counting with the internal clock.

When a match between the up-counter and the TC4DR value is detected, the logic level output from the $\overline{\text{PDO4}}$ pin is switched to the opposite state and INTTC4 interrupt request is generated. The up-counter is cleared at this time and then counting is continued. When a match between the up-counter and the TC4DR value is detected, the logic level output from the $\overline{\text{PDO4}}$ pin is switched to the opposite state again and INTTC4 interrupt request is generated. The up-counter is cleared at this time, and then counting and PDO are continued.

When the timer is stopped, the $\overline{PDO4}$ pin is high. Therefore, if the timer is stopped when the $\overline{PDO4}$ pin is low, the duty pulse may be shorter than the programmed value.

Example :Generating 1024 Hz pulse (fc = 20.0 MHz and CGCR<DV1CK> = 0)

```
LD      (TC4CR), 00000110B      : Sets the PDO mode. (TC4M = 10, TC4CK = 001)
SET     (P2DR), 2               : Sets the P22 output latch to 1.
LD      (TC4DR), 4CH            : 1/1024 ÷ 2^7/fc ÷ 2 (half cycle period) = 4CH
LD      (TC4CR), 00100110B      : Start TC4
```

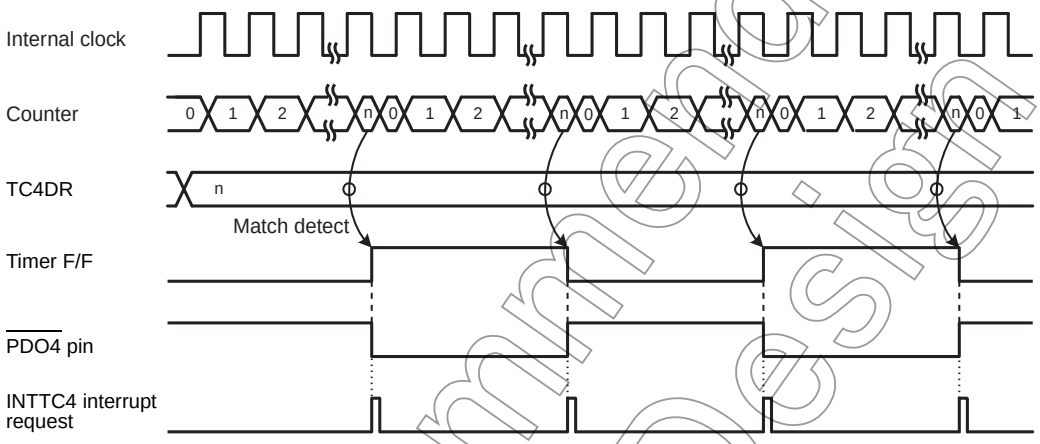


Figure 13-2 PDO Mode Timing Chart

13.3.4 Pulse Width Modulation (PWM) Output Mode

The pulse width modulation (PWM) output mode is used to generate the PWM pulse with up to 8 bits of resolution by an internal clock.

When a match between the up-counter and the TC4DR value is detected, the logic level output from the $\overline{PWM4}$ pin becomes low. The up-counter continues counting. When the up-counter overflow occurs, the $\overline{PWM4}$ pin becomes high. The INTTC4 interrupt request is generated at this time.

When the timer is stopped, the $\overline{PWM4}$ pin is high. Therefore, if the timer is stopped when the $\overline{PWM4}$ pin is low, one PMW cycle may be shorter than the programmed value.

TC4DR is serially connected to the shift register. If TC4DR is programmed during PWM output, the data set to TC4DR is not shifted until one PWM cycle is completed. Therefore, a pulse can be modulated periodically. For the first time, the data written to TC4DR is shifted when the timer is started by setting TC4CR<TC4S> to 1.

Note 1: The PWM output mode can be used only in the NORMAL and IDEL modes.

Note 2: In the PWM output mode, program TC4DR immediately after the INTTC4 interrupt request is generated (typically in the INTTC4 interrupt service routine.) When the programming of TC4DR and the INTTC4 interrupt occur at the same time, an unstable value is shifted, that may result in generation of pulse different from the programmed value until the next INTTC4 interrupt request is issued.

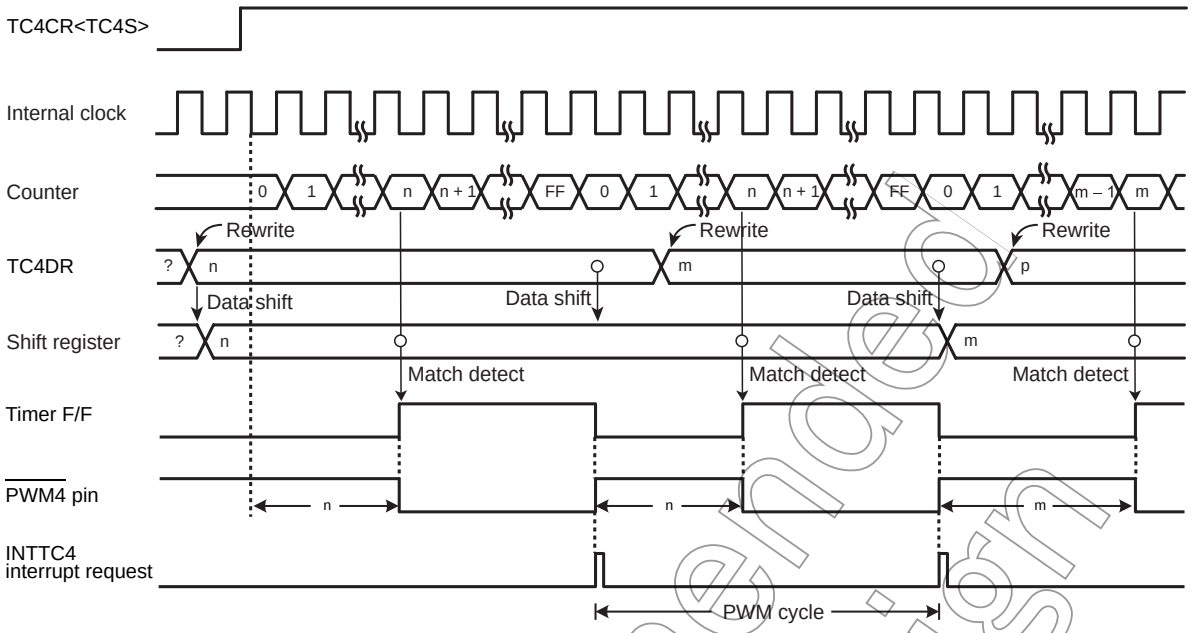


Figure 13-3 PWM output Mode Timing Chart (TC4)

Table 13-3 PWM Mode (Example: fc = 20 MHz)

TC4CK	NORMAL, IDLE Mode			
	DV1CK = 0		DV1CK = 1	
	Resolution [ns]	Cycle [μs]	Resolution [ns]	Cycle [μs]
000	-	-	-	-
001	-	-	-	-
010	-	-	-	-
011	400	102.4	800	204.8
100	200	51.2	400	102.4
101	100	25.6	200	51.2
110	-	-	-	-

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

The TMP88F846UG 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

14.1 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 14-1, 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 14-1 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 14-4. 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 14-4.

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.

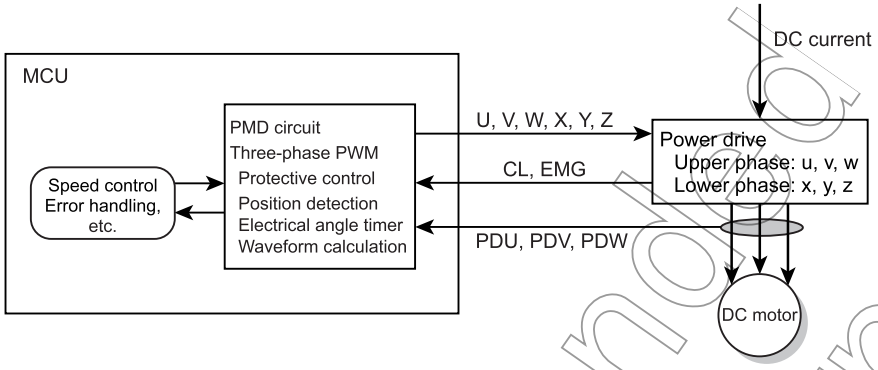


Figure 14-1 Conceptual Diagram of DC Motor Control

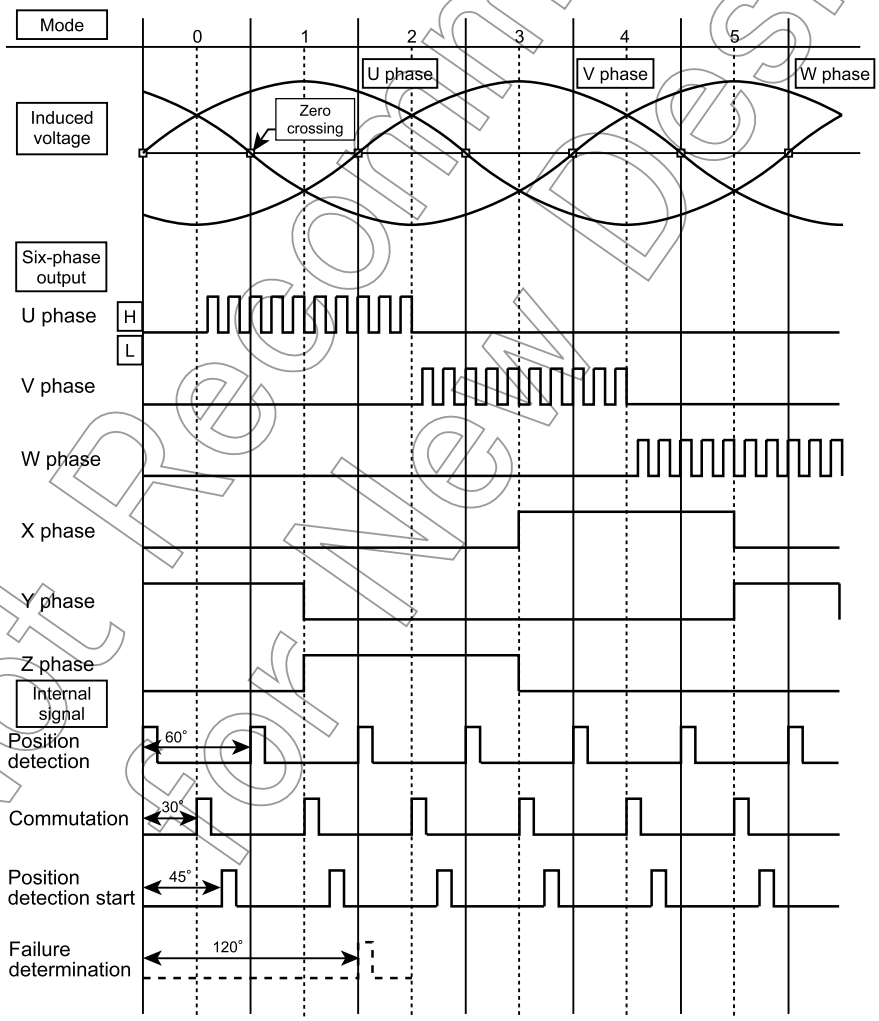


Figure 14-2 Example of Sensorless DC Motor Control Timing Chart

14.2 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.

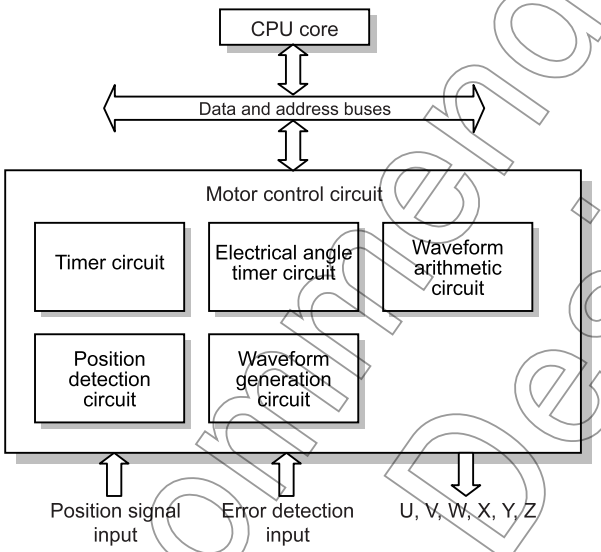


Figure 14-3 Block Diagram of the Motor Control Circuit

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.

14.3 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 14-2 from the timing chart in Figure 14-2. 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 14-2 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

14.3.1 Configuration of the position detection unit

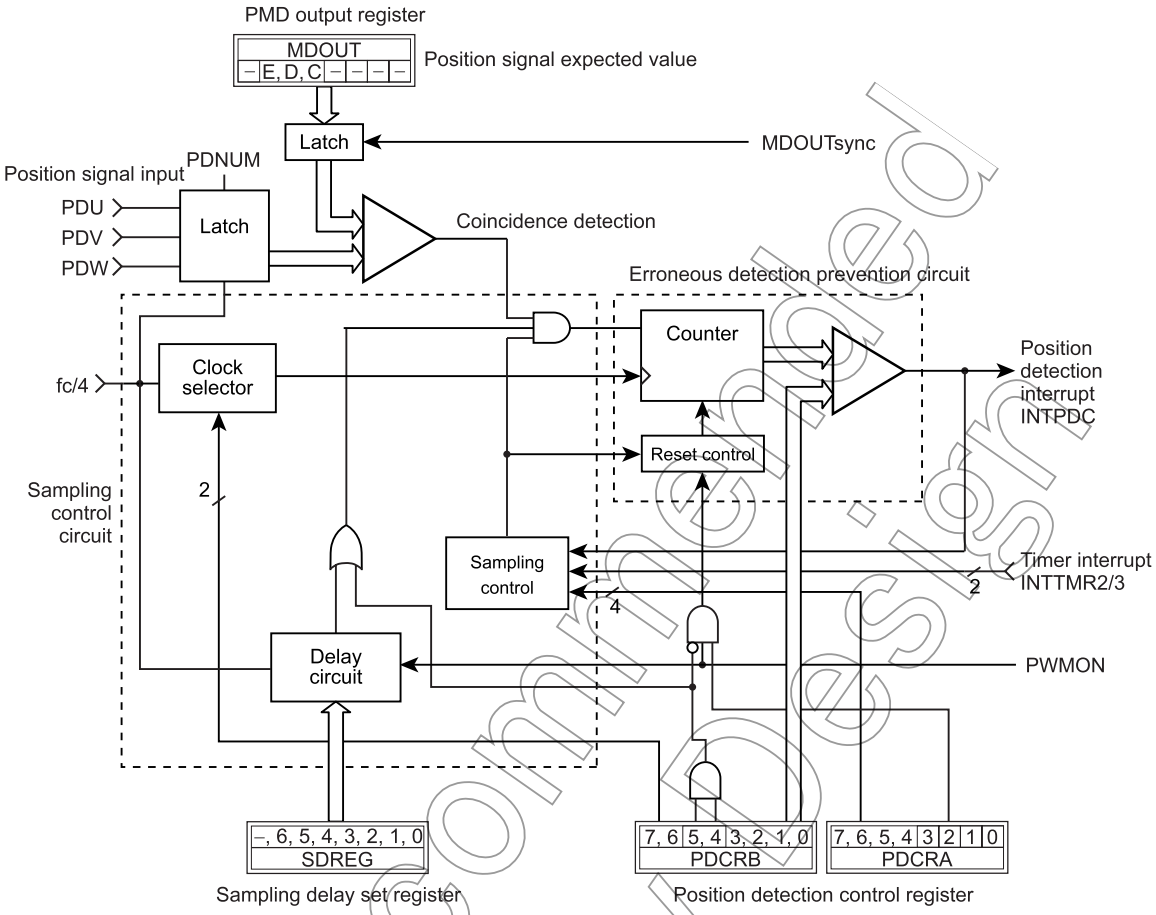


Figure 14-4 Configuration of the Position Detection Circuit

- 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).

14.3.2 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 14-5)

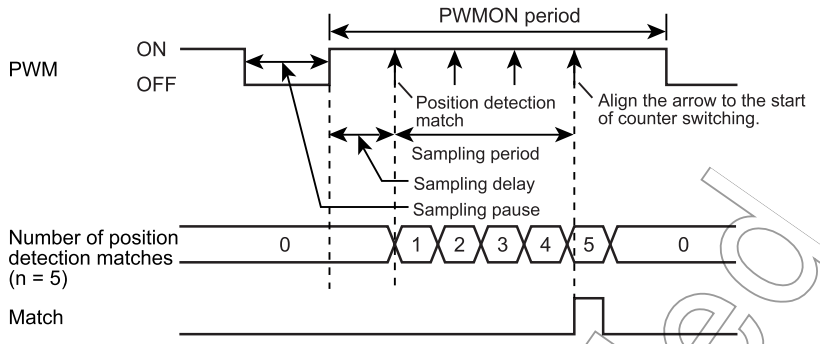


Figure 14-5 Position Detection Sampling Timing with the PWMON Period Selected

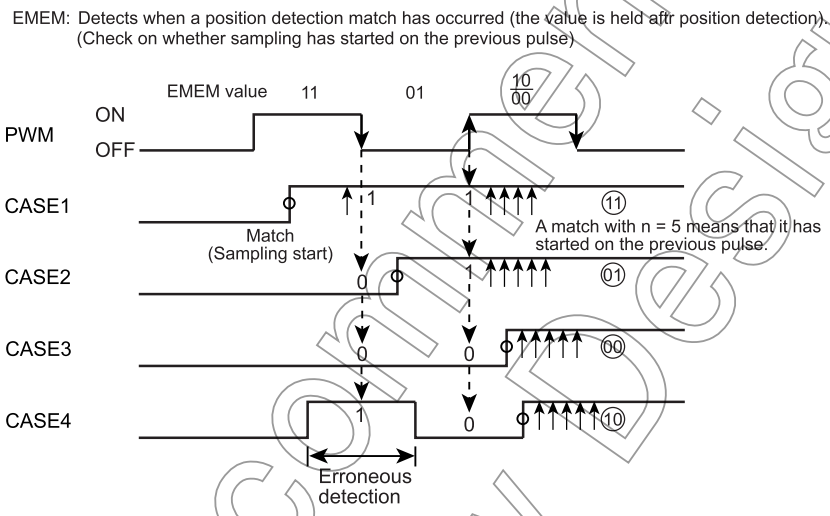


Figure 14-6 Detection Timing of the Position Detection Position

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 status	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 matching 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 detection function	0: Disable 1: Enable (Sampling starts)	

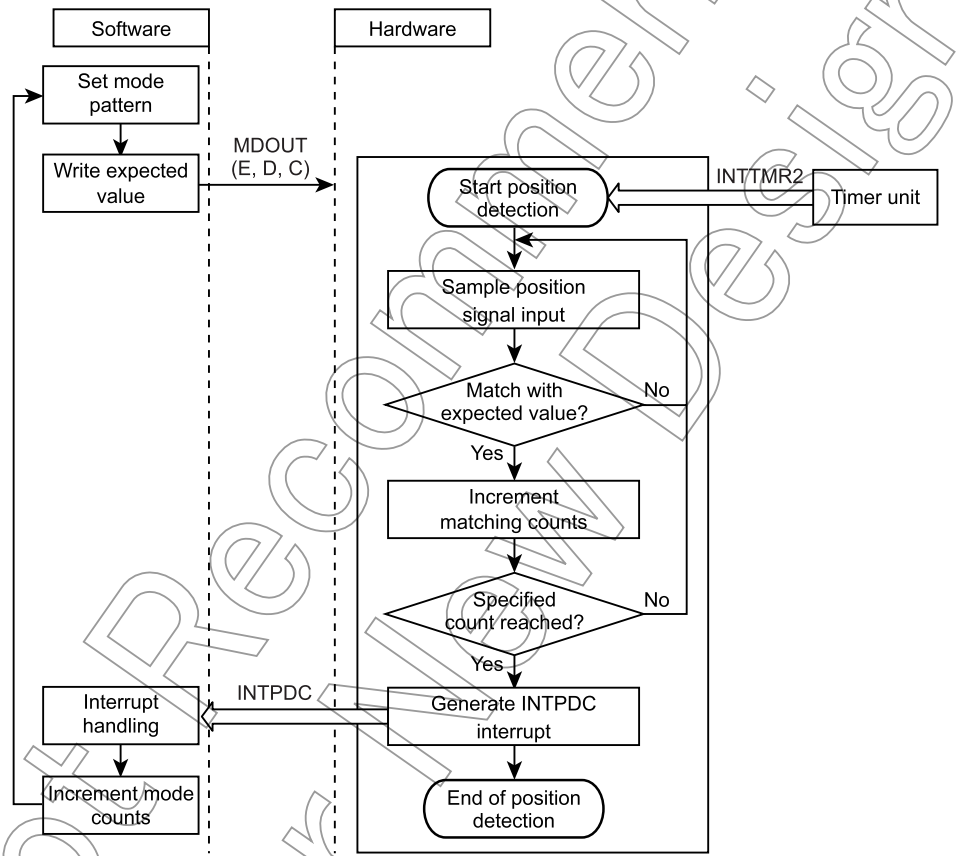
Note: Read-modify-write instructions, such as a bit manipulation instruction, cannot access the PDCRA because it contains a write only bit.

	7	6	5	4	3	2	1	0	
SDREG (01FA3H)	-	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).

14.3.3 Outline Processing in the Position Detection Unit



14.4 Timer Unit

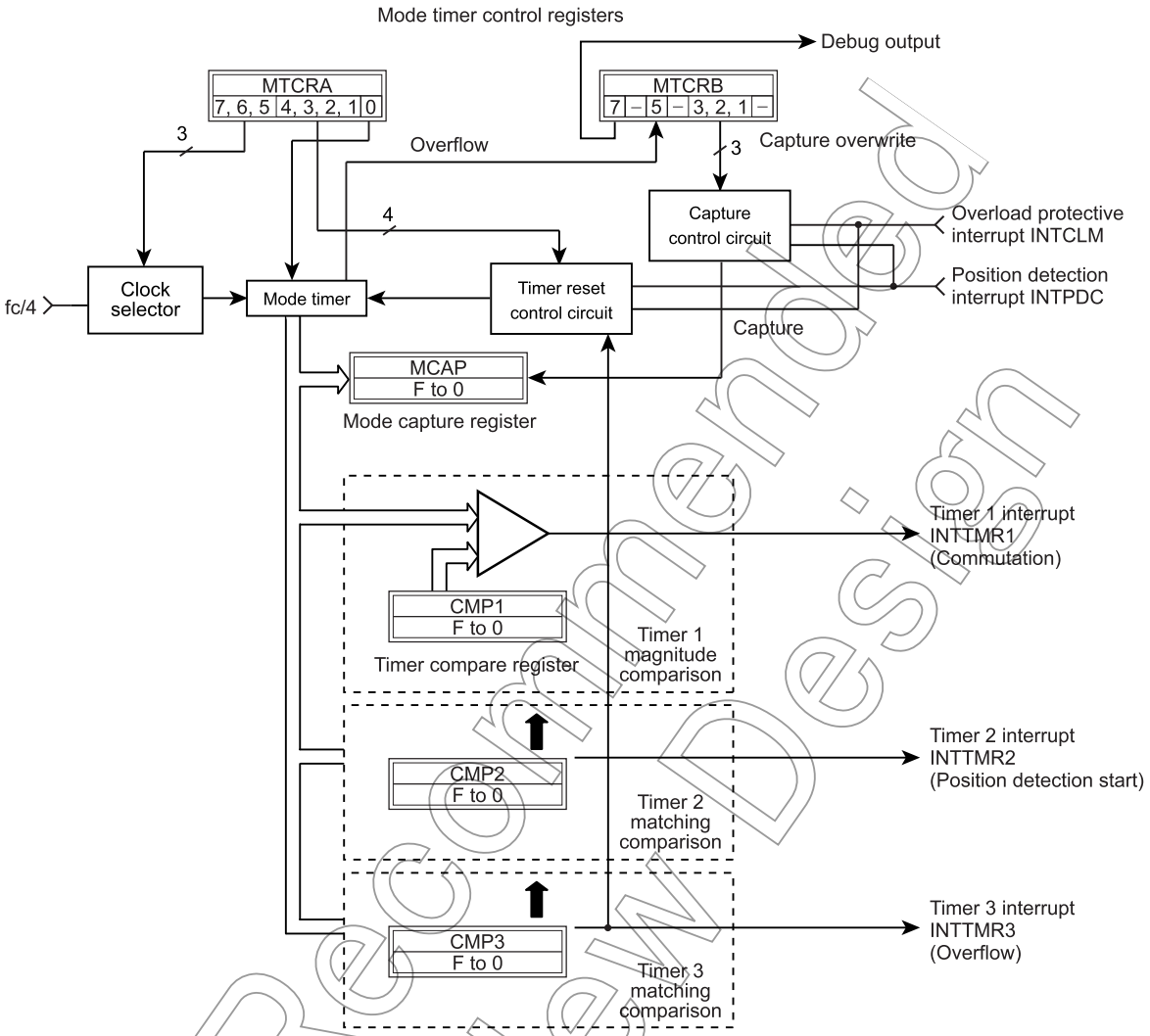


Figure 14-7 Timer Circuit Configuration

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.

14.4.1 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.

14.4.1.1 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 14-8). 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)	

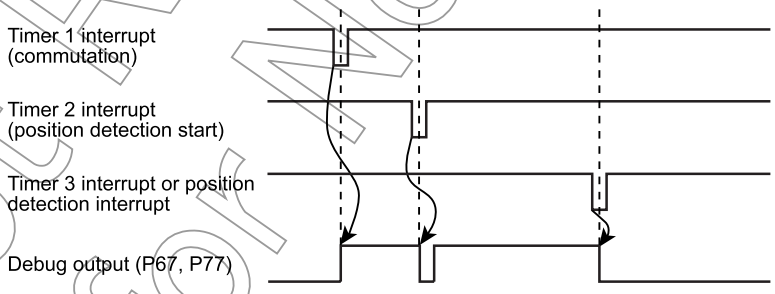


Figure 14-8 DBOUT Debug Output Diagram

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 overload 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)

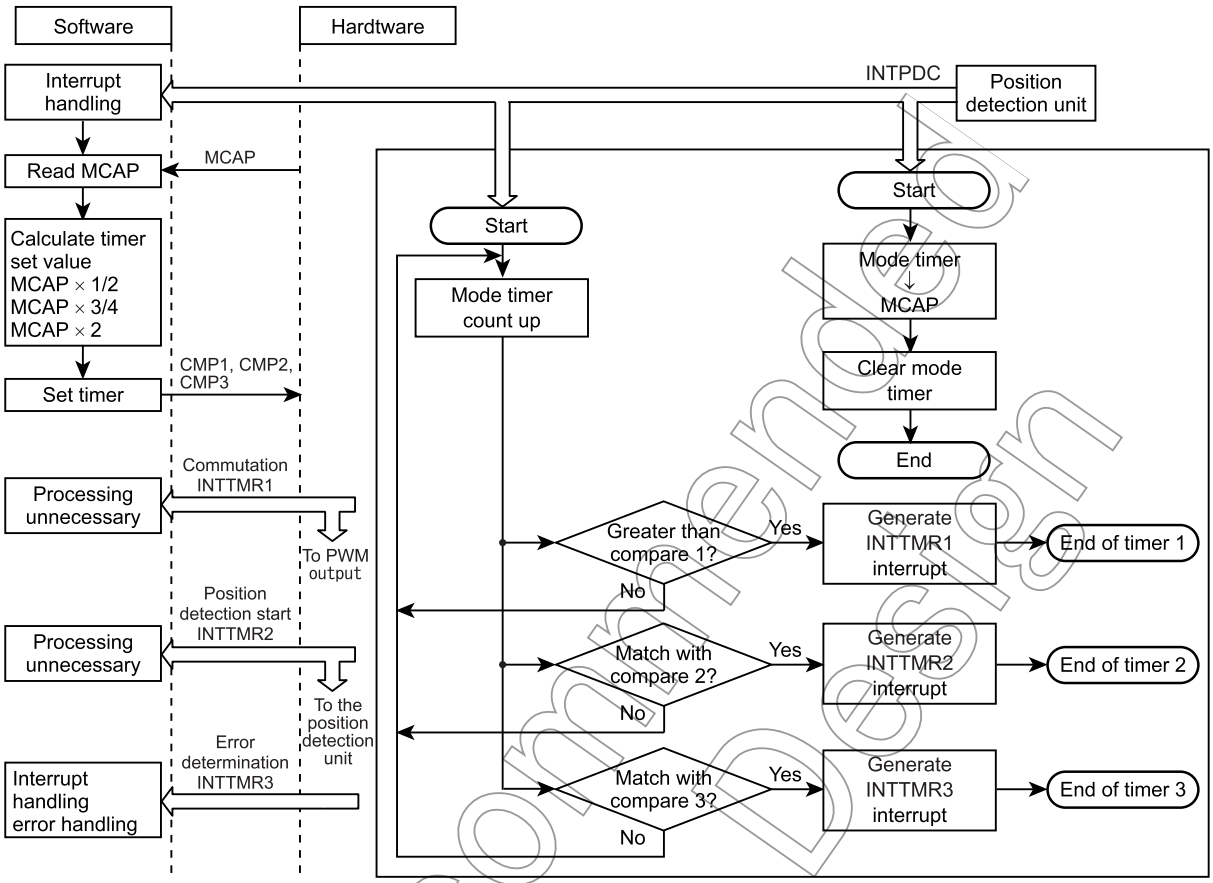
F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
DF	DE	DD	DC	DB	DA	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

(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.

14.4.1.2 Outline Processing in the Timer Unit



14.5 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.

14.5.1 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.

14.5.1.1 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 2), 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.

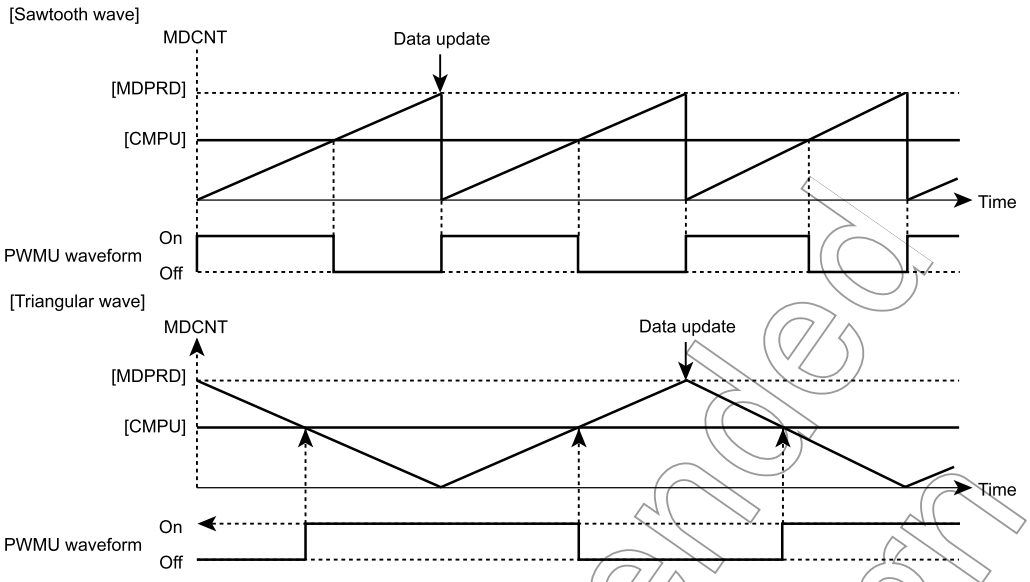


Figure 14-9 PWM Waveforms

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.

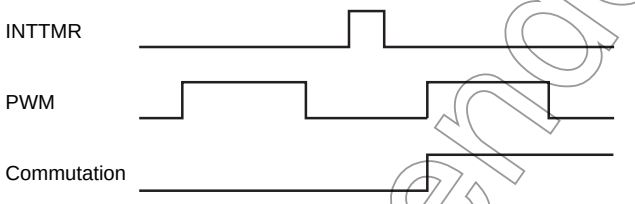
1. Three-phase independent mode: Values are set independently in the three-phase PMD Compare Registers to produce three-phase independent PWM waveforms. This method may be used to produce sinusoidal or any other desired drive waveforms.
2. Three-phase common mode: A value is set in only the U-phase PMD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

14.5.1.2 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.

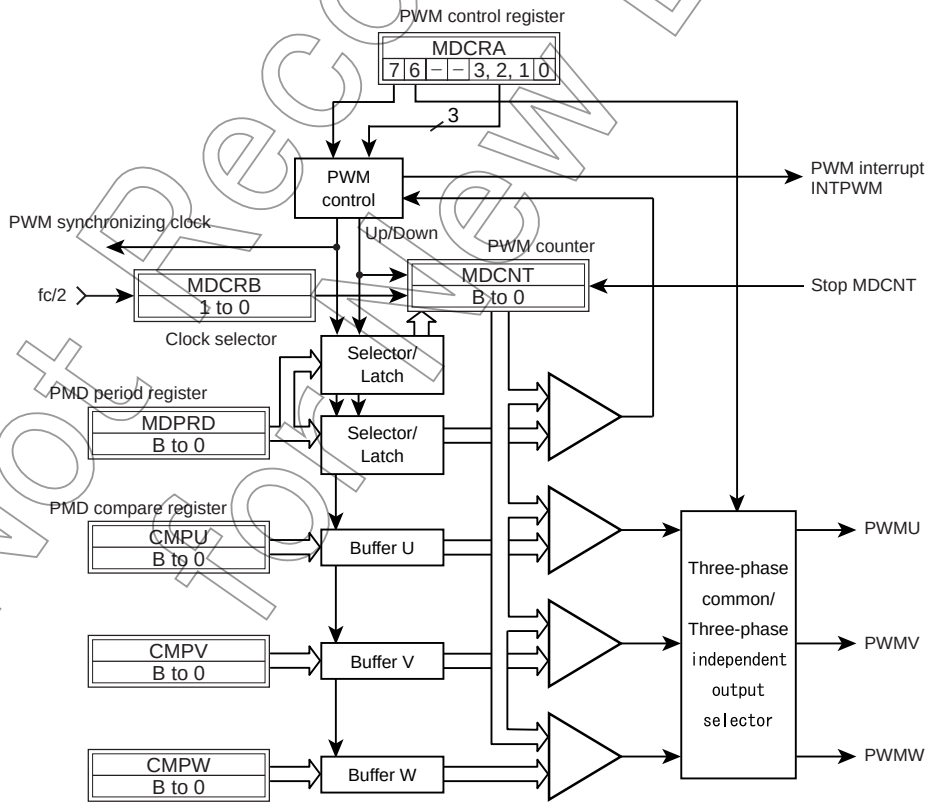


Figure 14-10 Pulse Width Modulation Circuit

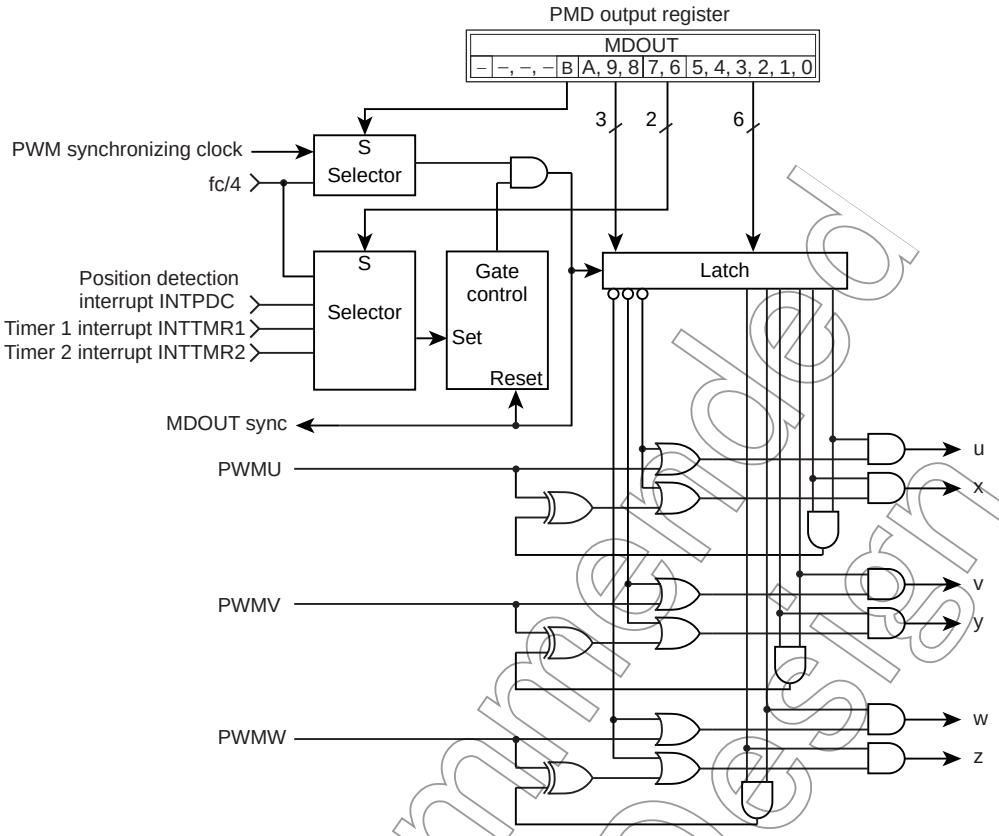


Figure 14-11 Commutation Control Circuit

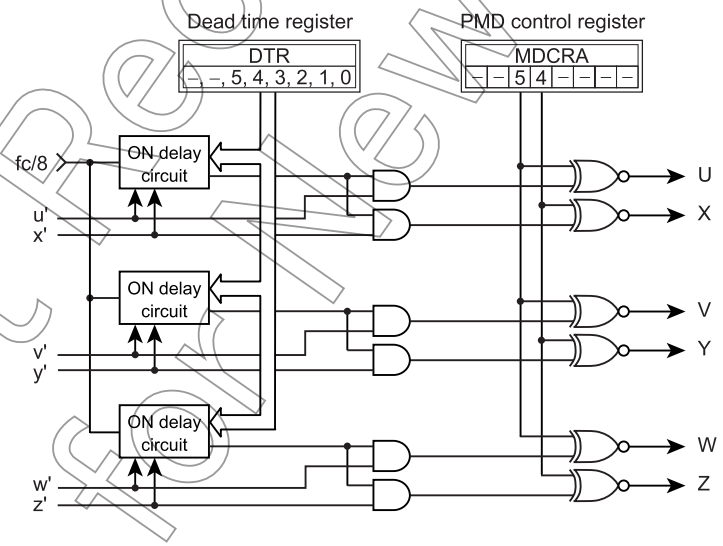


Figure 14-12 Dead Time Circuit

14.5.2 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 14-3)
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 14-3)

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 counter Select 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		

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

Table 14-3 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	PWM	PWM	L	L
0 1	L	PWM	L	H
1 0	PWM	L	H	L
1 1	PWM	$\overline{\text{PWM}}$	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	$\overline{\text{PWM}}$	H	H
0 1	H	$\overline{\text{PWM}}$	H	L
1 0	$\overline{\text{PWM}}$	H	L	H
1 1	$\overline{\text{PWM}}$	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	

14.5.4 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.

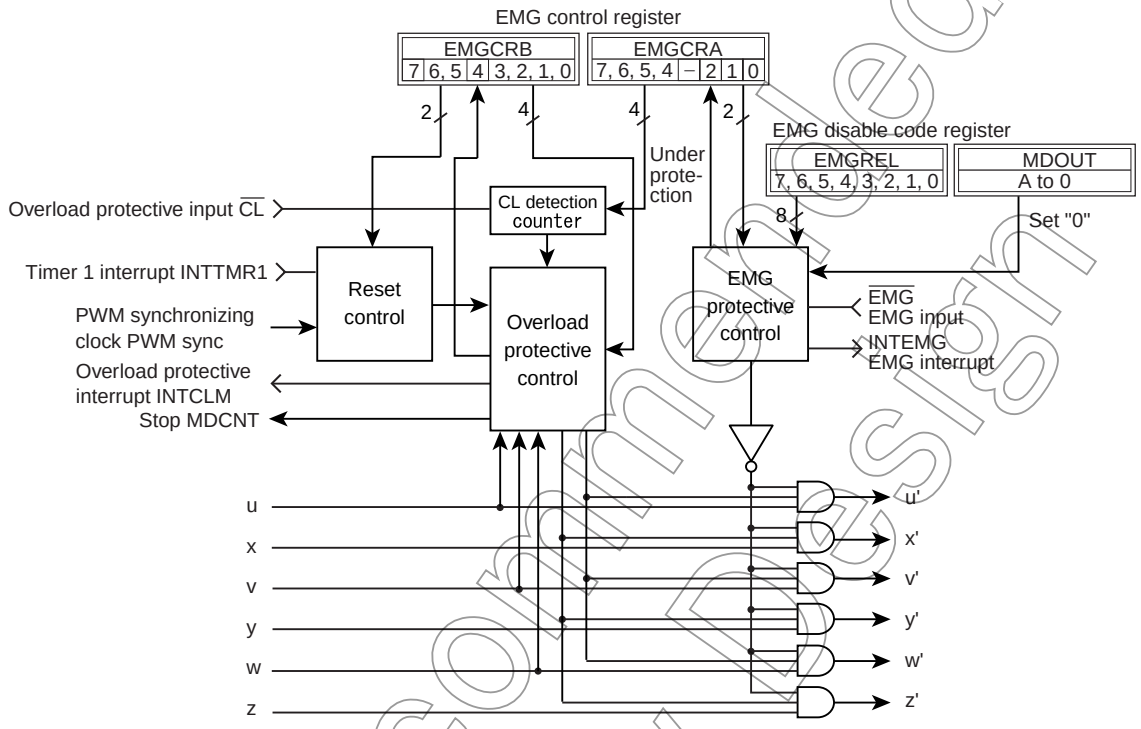


Figure 14-13 Configuration of the Protective Circuit

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.

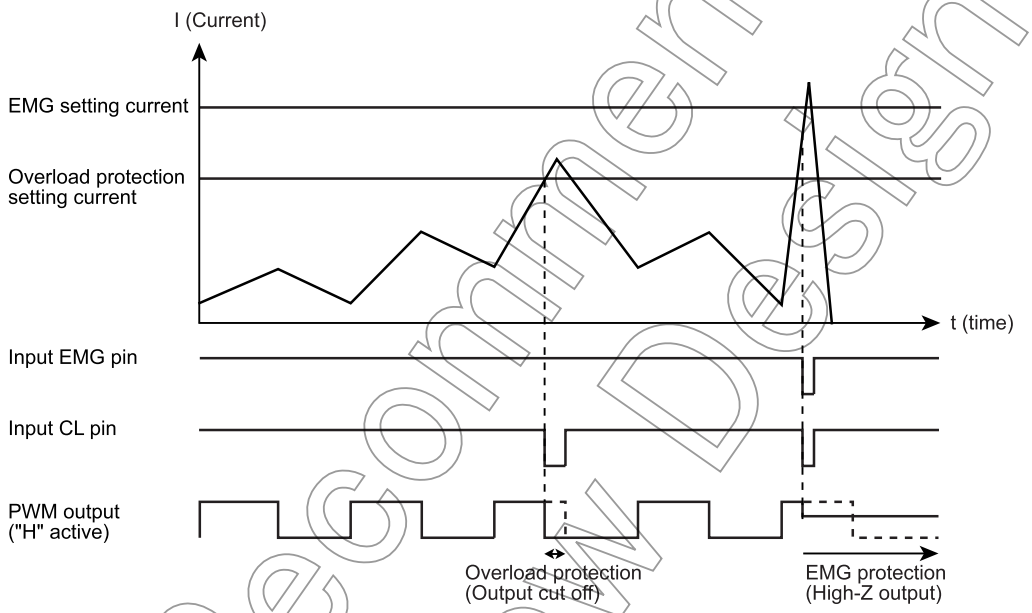


Figure 14-14 Example of Protection Circuit Operation

14.5.5 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.	$2^2/fc \times n$ (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 $\overline{EMG}$ 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.

14.6 Electrical Angle Timer and Waveform Arithmetic Circuit

Electrical Angle Timer

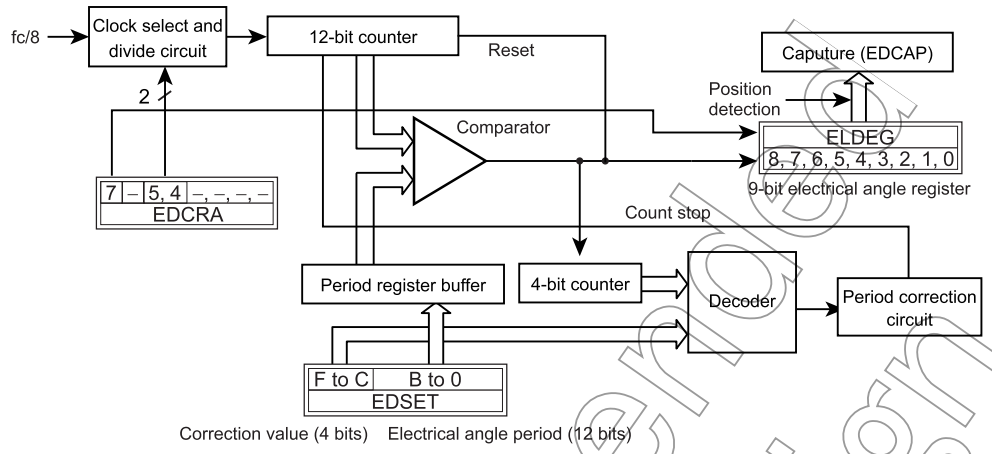


Figure 14-15 Electrical Angle Timer Circuit

Waveform Arithmetic Circuit

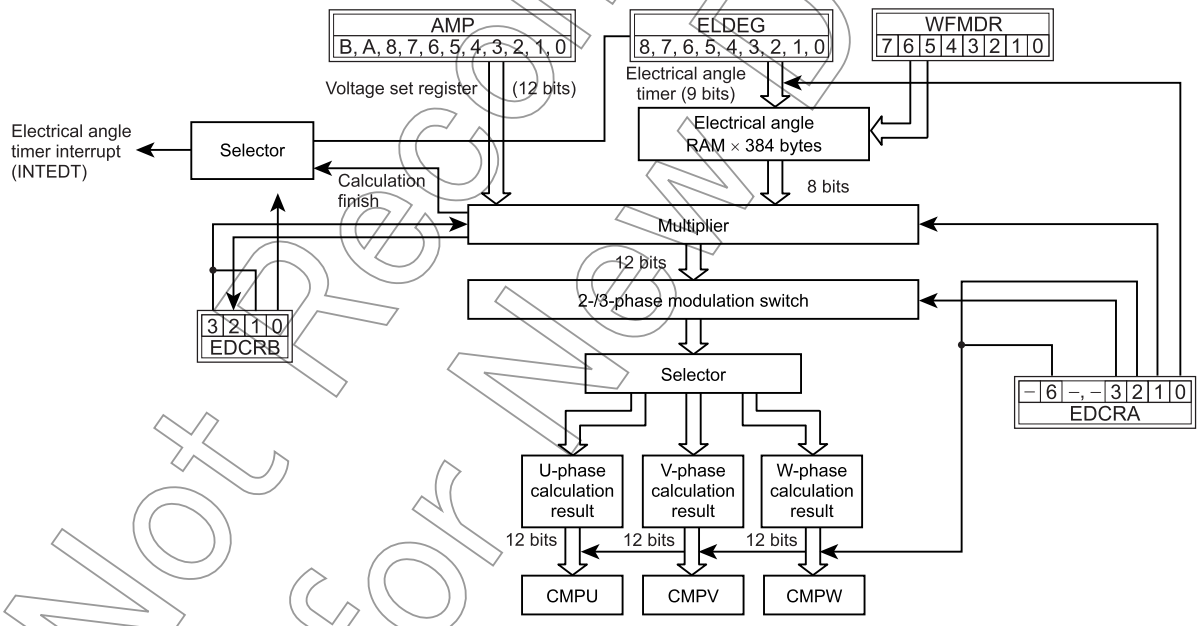


Figure 14-16 Waveform Arithmetic Circuit

14.6.1 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 (-120) degrees relative to the U phase, and the W phase whose electrical angle is +240 (-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.

14.6.1.1 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 $\text{Three-phase modulation: DATA} = \frac{\text{MOPRD}}{2} \pm \frac{\text{ramdata(ELDEG)} \times \text{AMP}}{2}$ 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.

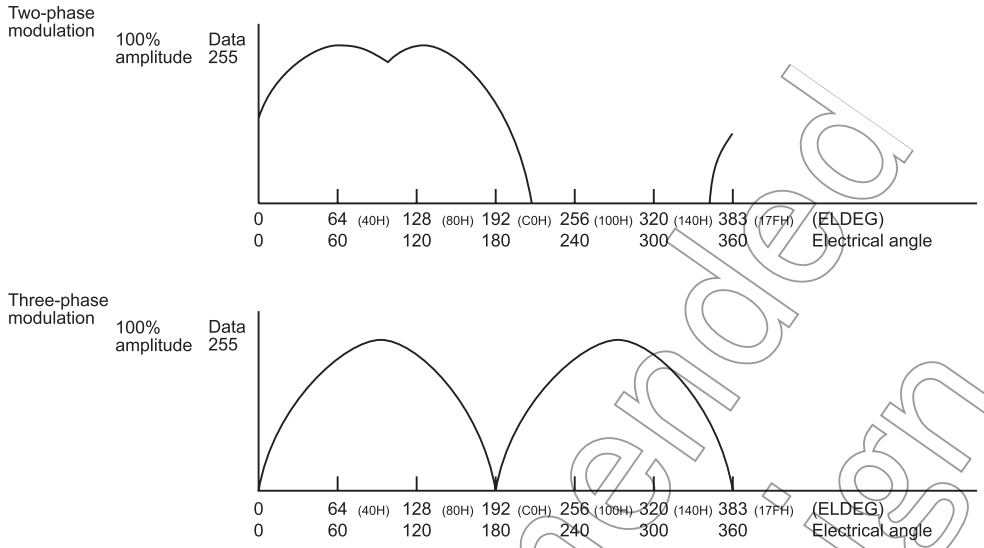
ELDEG	Electrical angle	Read the electrical angle. This register can also be set to initialize or correct the angle while counting. Any value greater than 17FH cannot be set.

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 14-17 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 not calculation 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 angle Enable/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.

14.6.1.2 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: Recount 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. $2^2/fc \times n$ (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 function 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 beforehand.)

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 calculation
		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.

15. Asynchronous Serial interface (UART)

The TMP88F846UG 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.

When using UART of TMP88F846UG, it might be impossible to communicate because of the error of the oscillation frequency. Moreover, the oscillation frequency changes and it is impossible to communicate by the temperature change etc. Therefore, we recommend you to adjust the transmission rate by the communication protocol as follows.

[For Example]

1. Fixed data (ex: 05AH) is transmitted at head byte of transmission data or constant intervals from the host side whose oscillation is stable.
2. The host side keeps transmitting fixed data until receiving the ACK from remote side. The usual communication is restarted after ACK reception.
3. Remote side adjusts a transfer rate finely until receiving fixed data. When receiving fixed data, an ACK is transmitted to host and communication is usually restarted.

15.1 Configuration

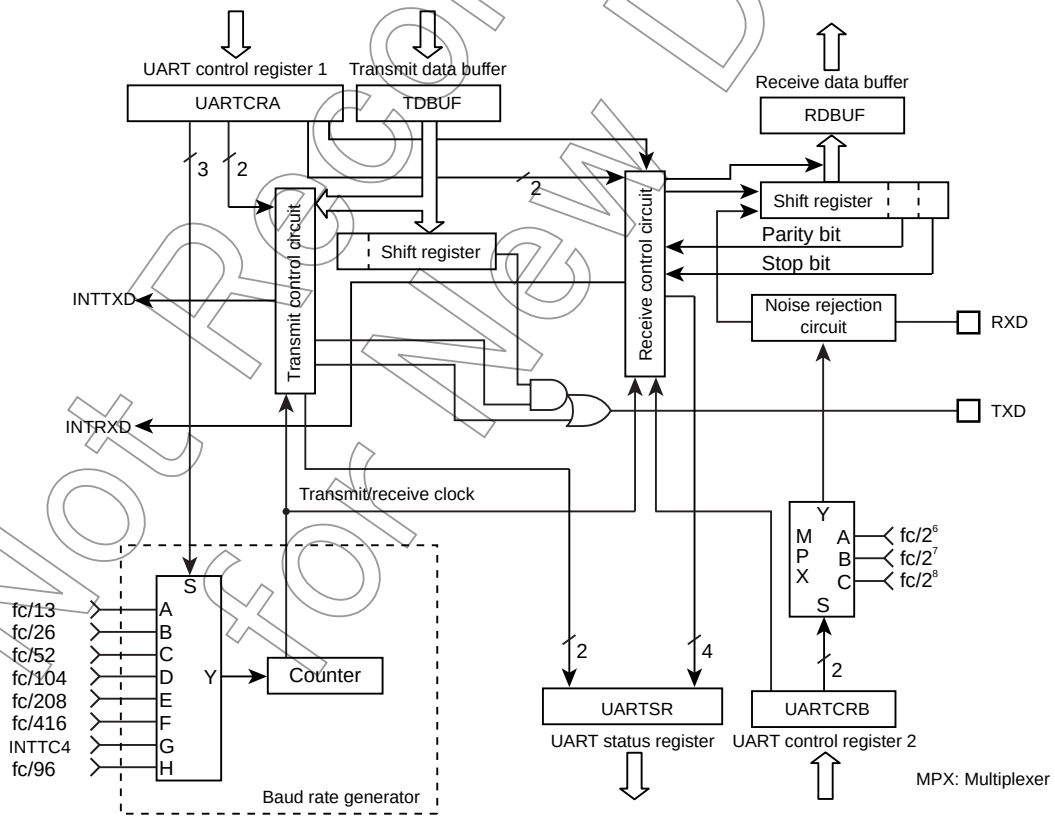


Figure 15-1 UART (Asynchronous Serial Interface)

15.2 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 1: Settings of RXDNC are limited depending on the transfer clock specified by BRG. The combination "O" is available but please do not select the combination "-". The transfer clock is calculated by the following equation : Transfer clock [Hz] =INTTC4 sourcec clock [Hz] ÷ TC4DR set value

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

UART Status Register

UARTSR (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)

15.3 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.

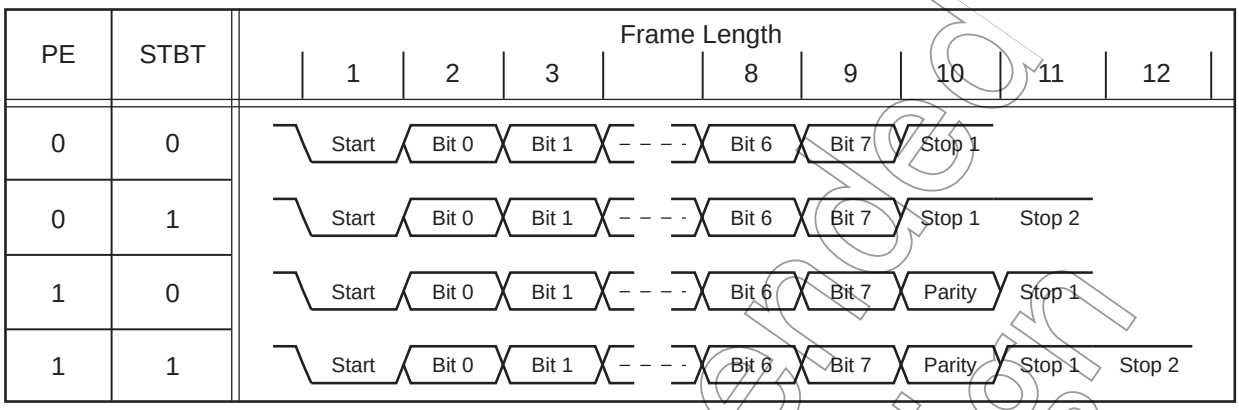


Figure 15-2 Transfer Data Format

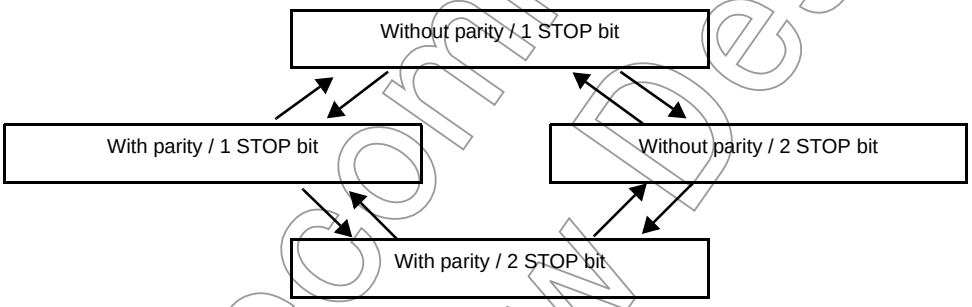


Figure 15-3 Caution on Changing Transfer Data Format

Note: In order to switch the transfer data format, perform transmit operations in the above Figure 15-3 sequence except for the initial setting.

15.4 Transfer Rate

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

Table 15-1 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

15.5 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).

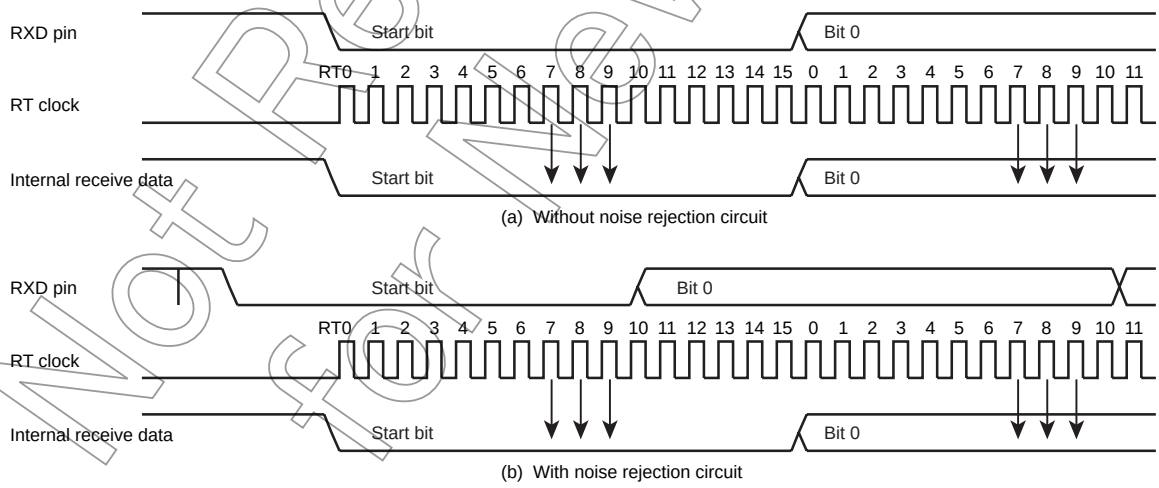


Figure 15-4 Data Sampling Method

15.6 STOP Bit Length

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

15.7 Parity

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

15.8 Transmit/Receive Operation

15.8.1 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.

15.8.2 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.

15.9 Status Flag

15.9.1 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.

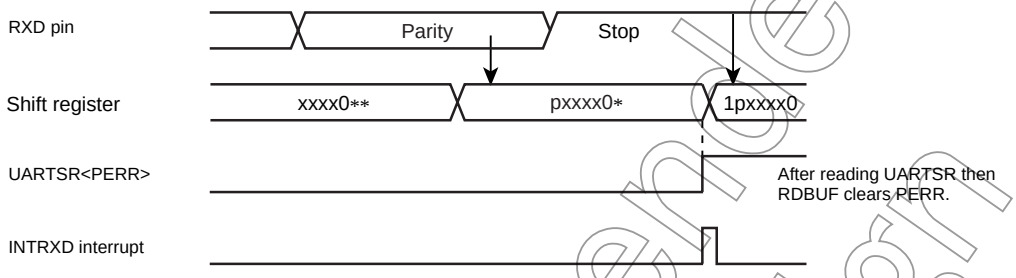


Figure 15-5 Generation of Parity Error

15.9.2 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.

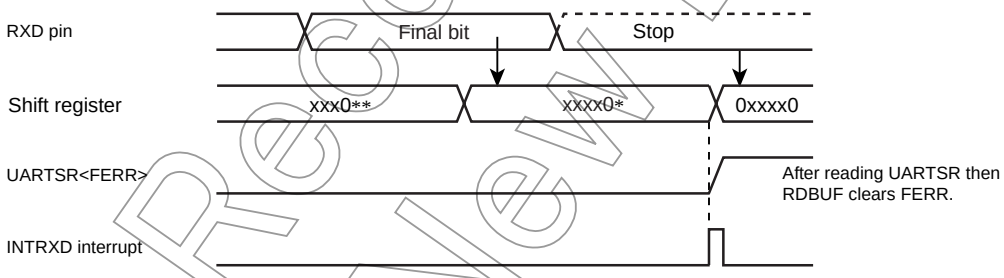


Figure 15-6 Generation of Framing Error

15.9.3 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.

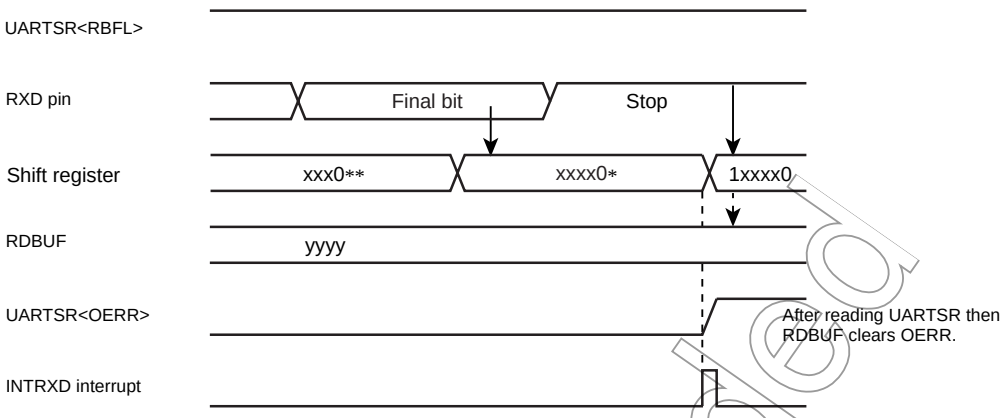


Figure 15-7 Generation of Overrun Error

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

15.9.4 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.

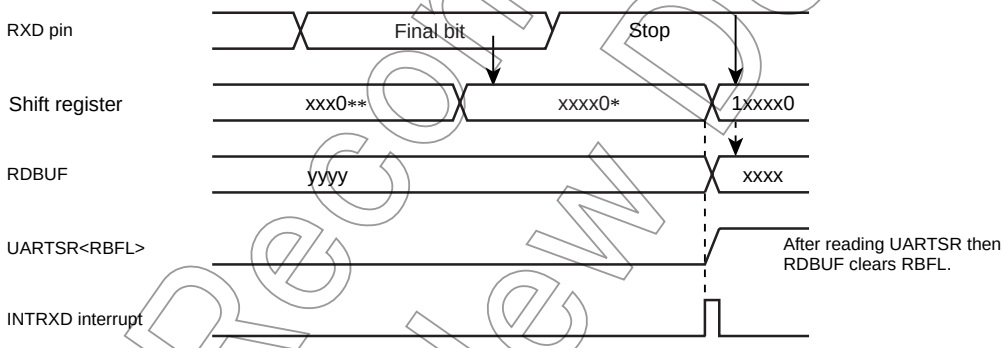


Figure 15-8 Generation of Receive Data Buffer Full

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.

15.9.5 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.

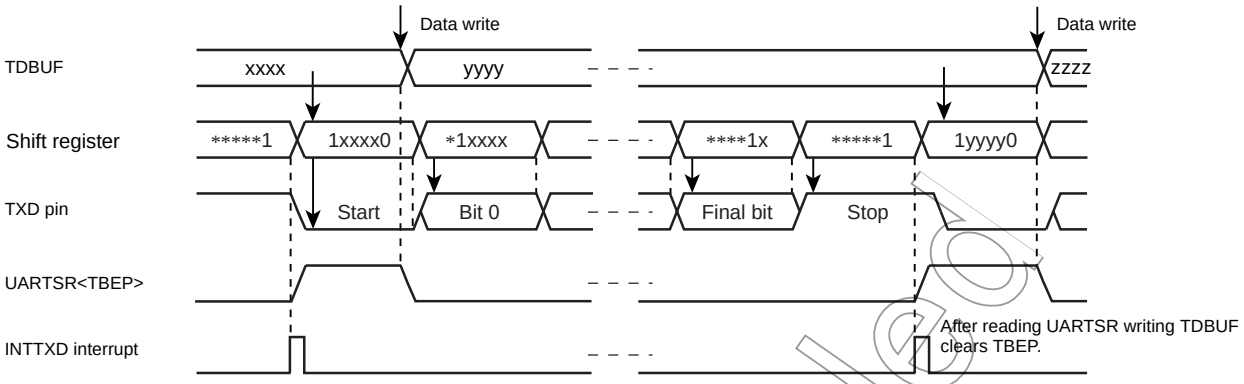


Figure 15-9 Generation of Transmit Data Buffer Empty

15.9.6 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.

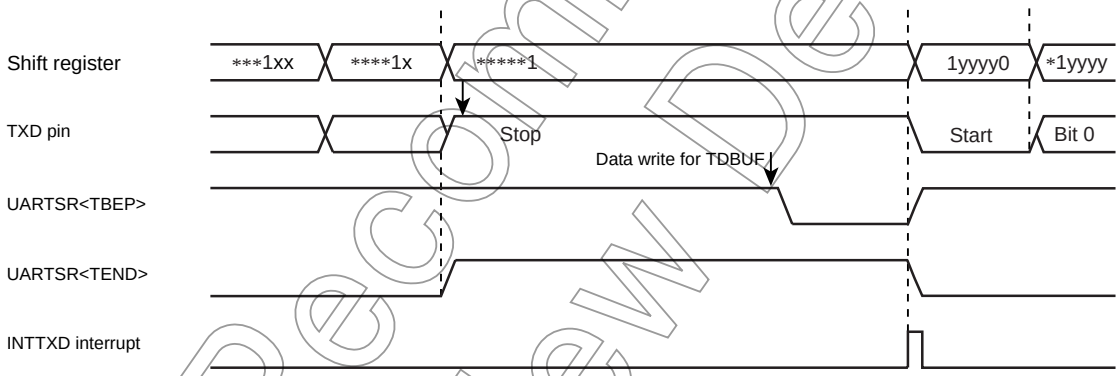


Figure 15-10 Generation of Transmit End Flag and Transmit Data Buffer Empty

16. Synchronous Serial Interface (SIO)

The TMP88F846UG has a clocked-synchronous 8-bit serial interface. Serial interface has an 8-byte transmit and receive data buffer that can automatically and continuously transfer up to 64 bits of data.

Serial interface is connected to outside peripheral devices via SO, SI, $\overline{SCK}$ port.

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

16.1 Configuration

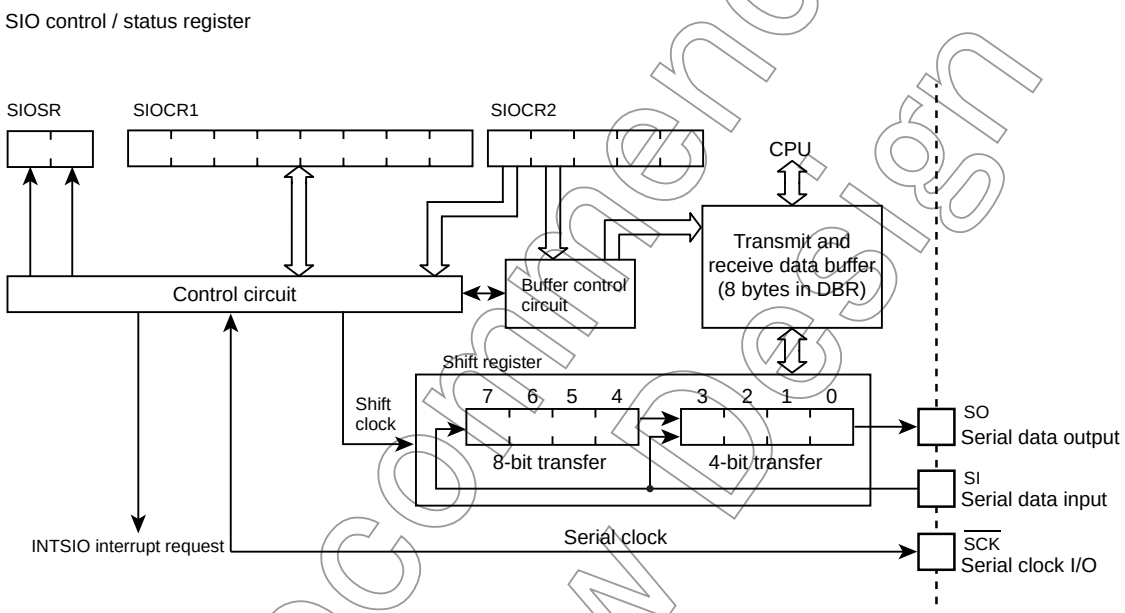


Figure 16-1 Serial Interface

16.2 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 (1F96H)

7	6	5	4	3	2	1	0
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

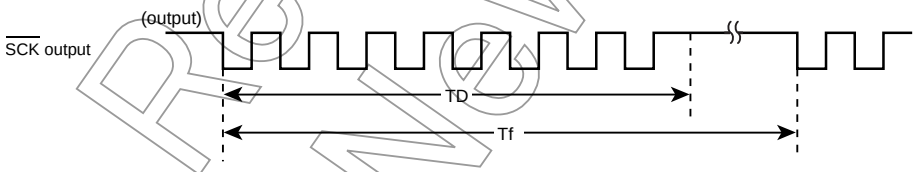


Figure 16-2 Frame time (T_f) and Data transfer time (T_D)

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".

16.3 Serial clock

16.3.1 Clock source

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

16.3.1.1 Internal clock

Any of six frequencies can be selected. The serial clock is output to the outside on the $\overline{SCK}$ pin. The $\overline{SCK}$ 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 16-1 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)

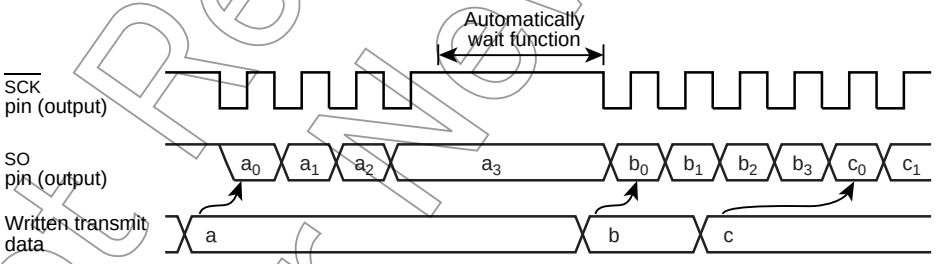


Figure 16-3 Automatic Wait Function (at 4-bit transmit mode)

16.3.1.2 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.

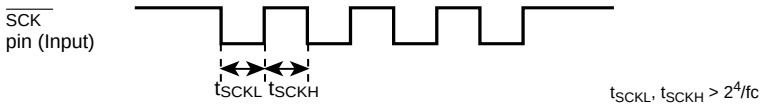


Figure 16-4 External clock pulse width

16.3.2 Shift edge

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

16.3.2.1 Leading edge

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

16.3.2.2 Trailing edge

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

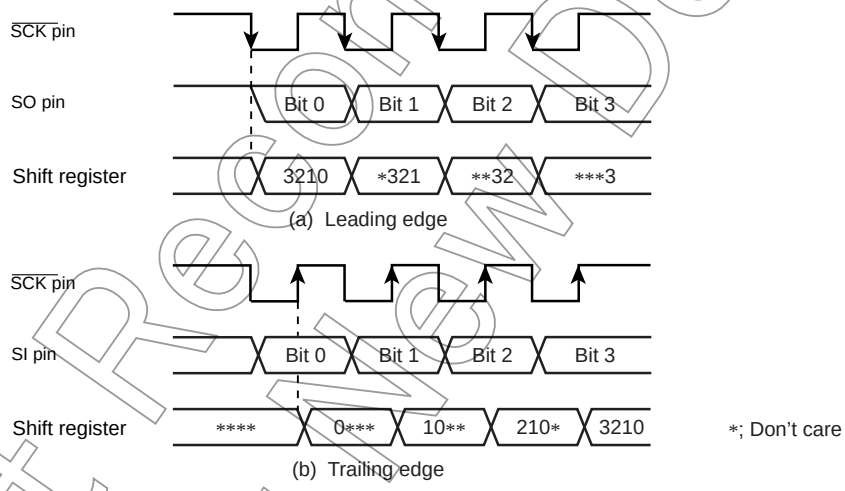


Figure 16-5 Shift edge

16.4 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).

16.5 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.

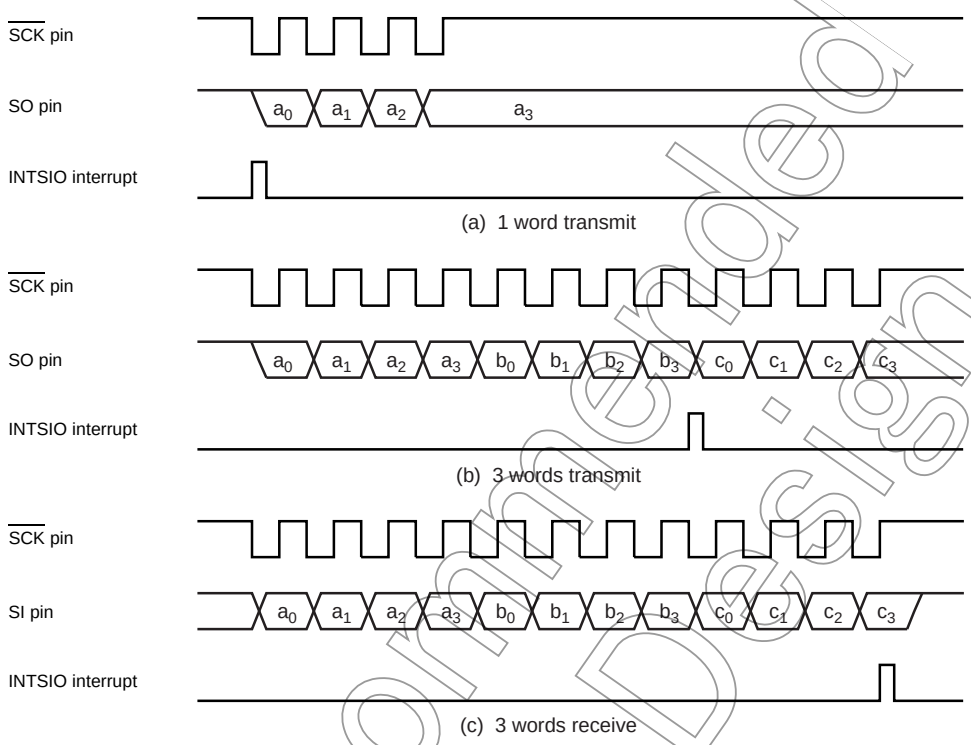


Figure 16-6 Number of words to transfer (Example: 1word = 4bit)

16.6 Transfer Mode

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

16.6.1 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”.

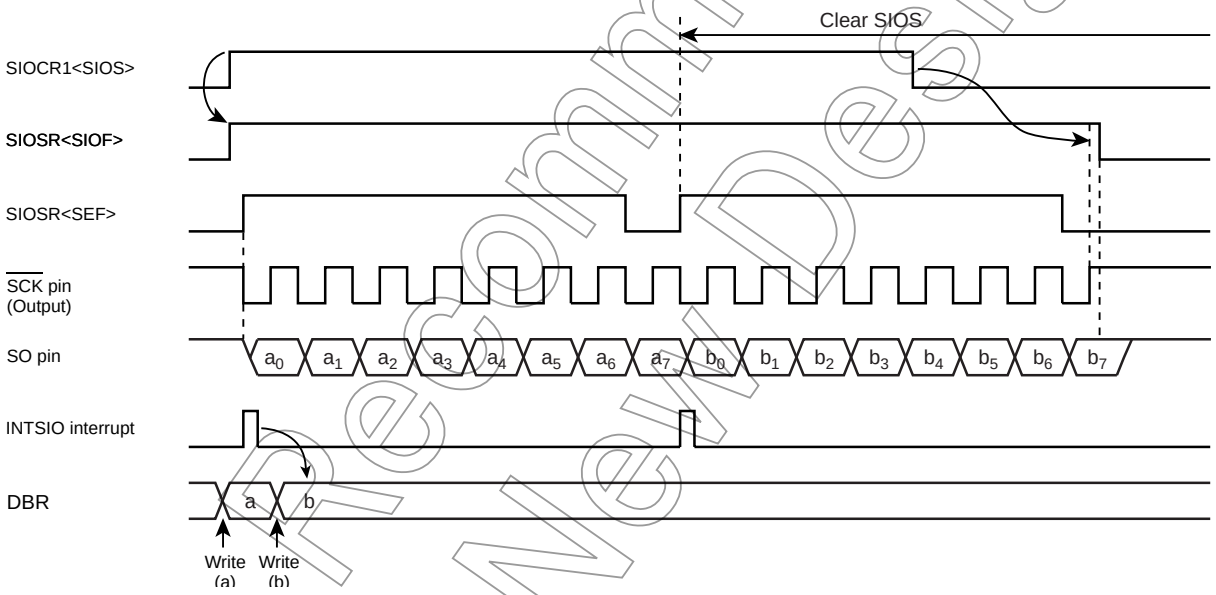


Figure 16-7 Transfer Mode (Example: 8bit, 1word transfer, Internal clock)

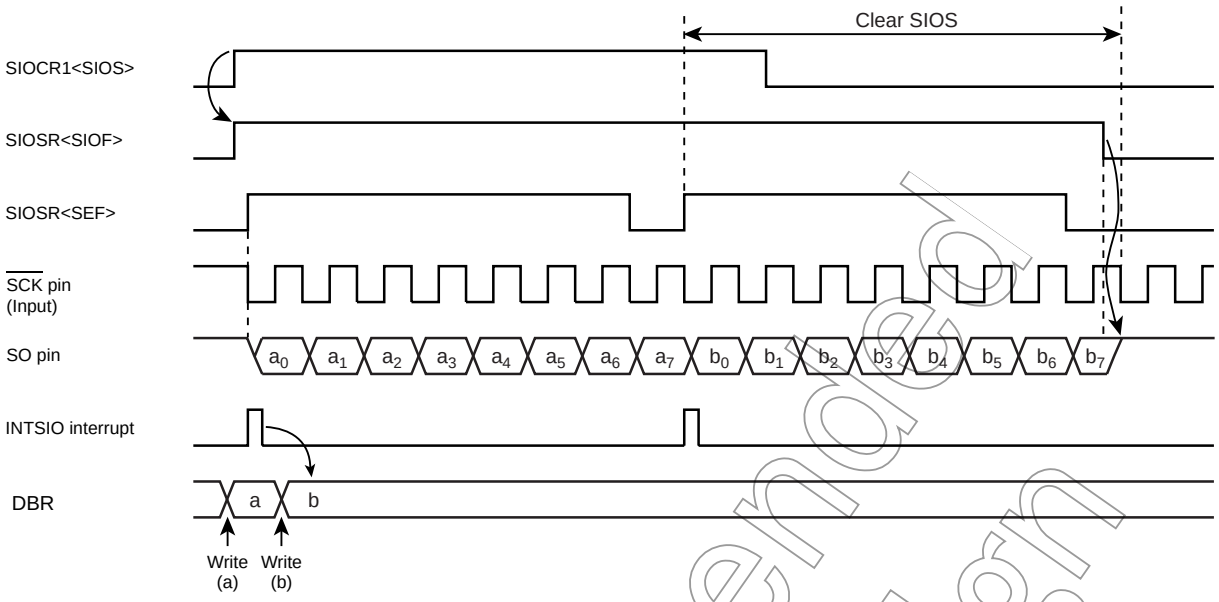


Figure 16-8 Transfer Mode (Example: 8bit, 1word transfer, External clock)

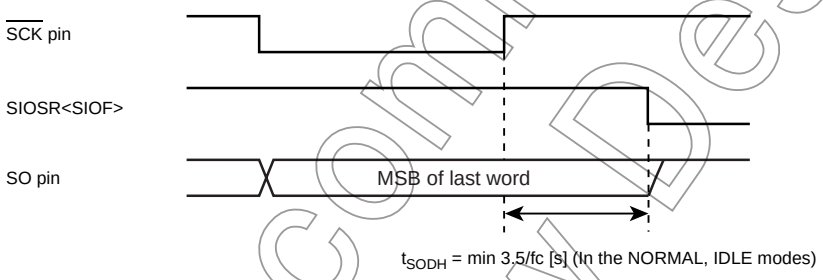


Figure 16-9 Transmitted Data Hold Time at End of Transfer

16.6.2 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.

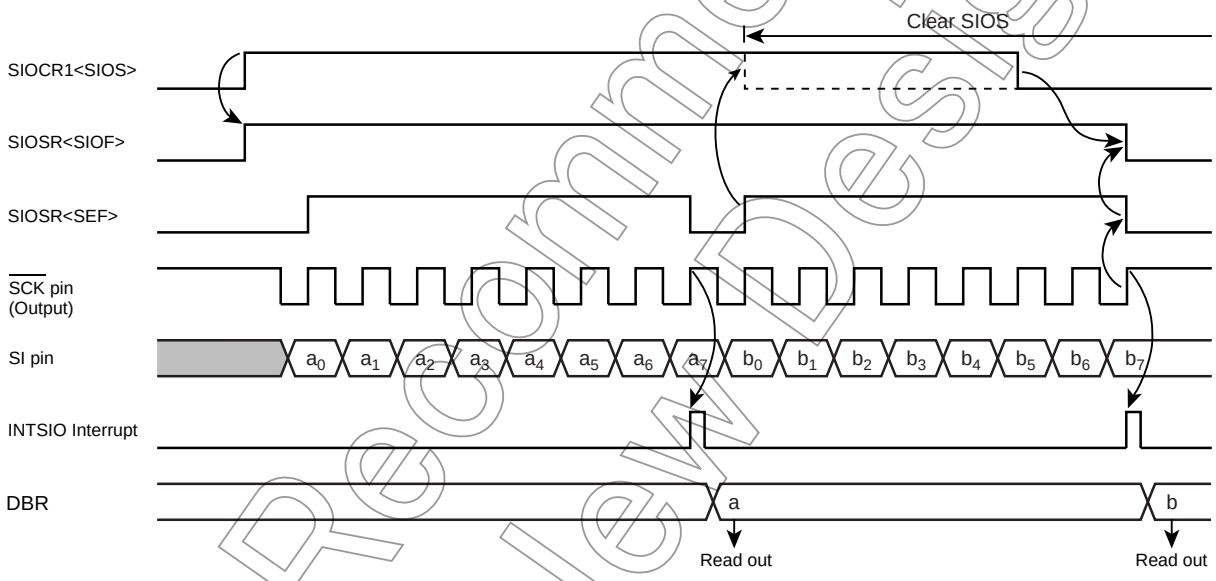


Figure 16-10 Receive Mode (Example: 8bit, 1word transfer, Internal clock)

16.6.3 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.

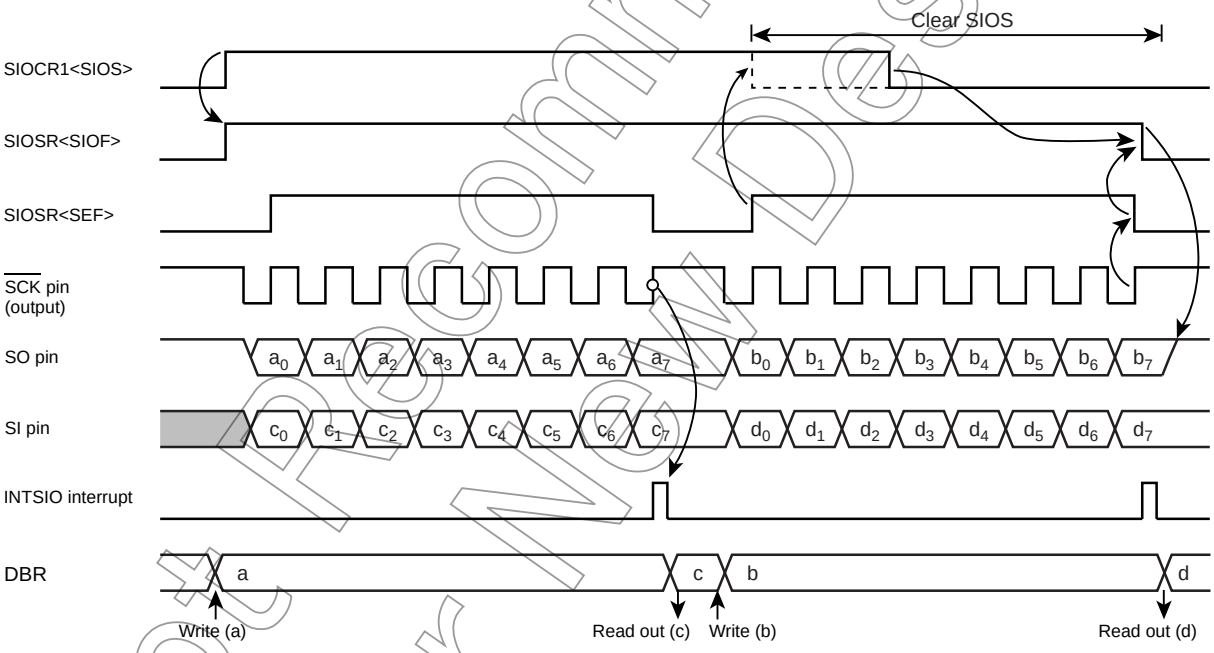


Figure 16-11 Transfer / Receive Mode (Example: 8bit, 1word transfer, Internal clock)

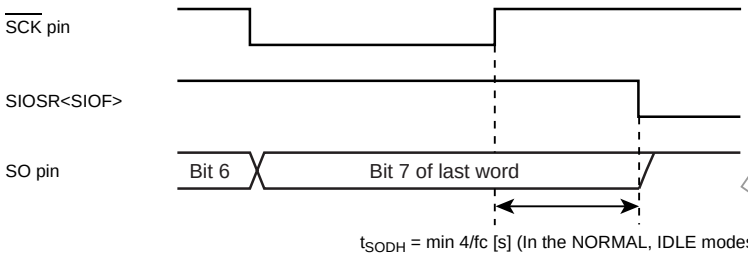


Figure 16-12 Transmitted Data Hold Time at End of Transfer / Receive

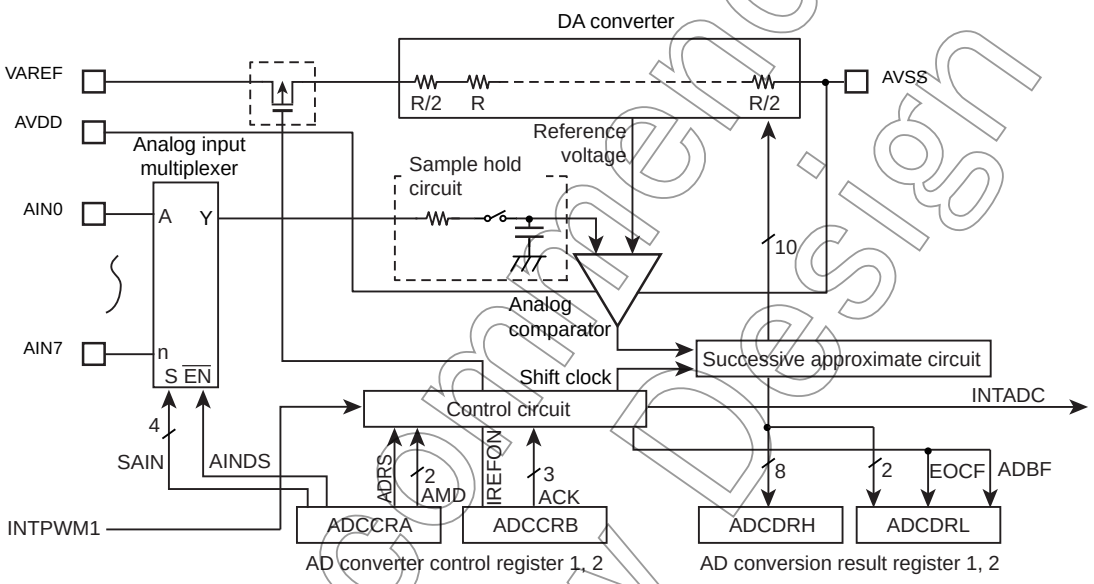
17. 10-bit AD Converter (ADC)

The TMP88F846UG have a 10-bit successive approximation type AD converter.

17.1 Configuration

The circuit configuration of the 10-bit AD converter is shown in Figure 17-1.

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 combining a analog input port. For details, see the section on "I/O ports".

Figure 17-1 10-bit AD Converter

17.2 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: INTPWM start mode 11: Repeat mode	
AINDS	Analog input control	0: Analog input enable 1: Analog input disable	
SAIN	Analog input channel select	0000: AIN0 0001: AIN1 0010: AIN2 0011: AIN3 0100: AIN4 0101: AIN5 0110: AIN6 0111: AIN7 1000: Reserved 1001: Reserved 1010: Reserved 1011: Reserved 1100: Reserved 1101: Reserved 1110: Reserved 1111: Reserved	

Note 1: Select analog input channel during AD converter stops (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 STOP mode is started, AD converter control register1 (ADCCRA) is all initialized and no data can be written in this register. Therefore, to use AD converter again, set the ADCCRA newly after returning to NORMAL mode.

Note 7: After RESET, ADCCRA<SAIN> is initialized Reserved setting. Therefore, set the appropriate analog input channel to ADCCRA<SAIN> when use AD converter.

Note 8: 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	(Initial value: **0* 000*)
			IREFON	"1"	ACK			"0"	

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.

Note 3: After STOP mode is started, AD converter control register2 (ADCCRB) is all initialized and no data can be written in this register. Therefore, to use AD converter again, set the ADCCRB newly after returning to NORMAL mode.

Table 17-1 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 17-2 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. Therefore, 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. It also is cleared upon entering STOP mode.

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

17.3 Function

17.3.1 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).

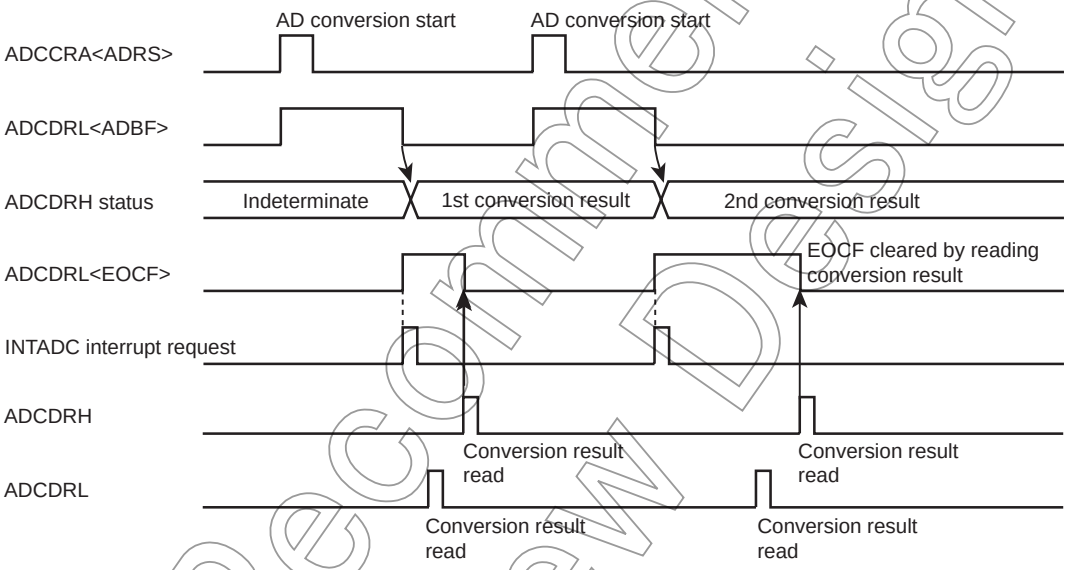


Figure 17-2 Software Start Mode

17.3.2 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.

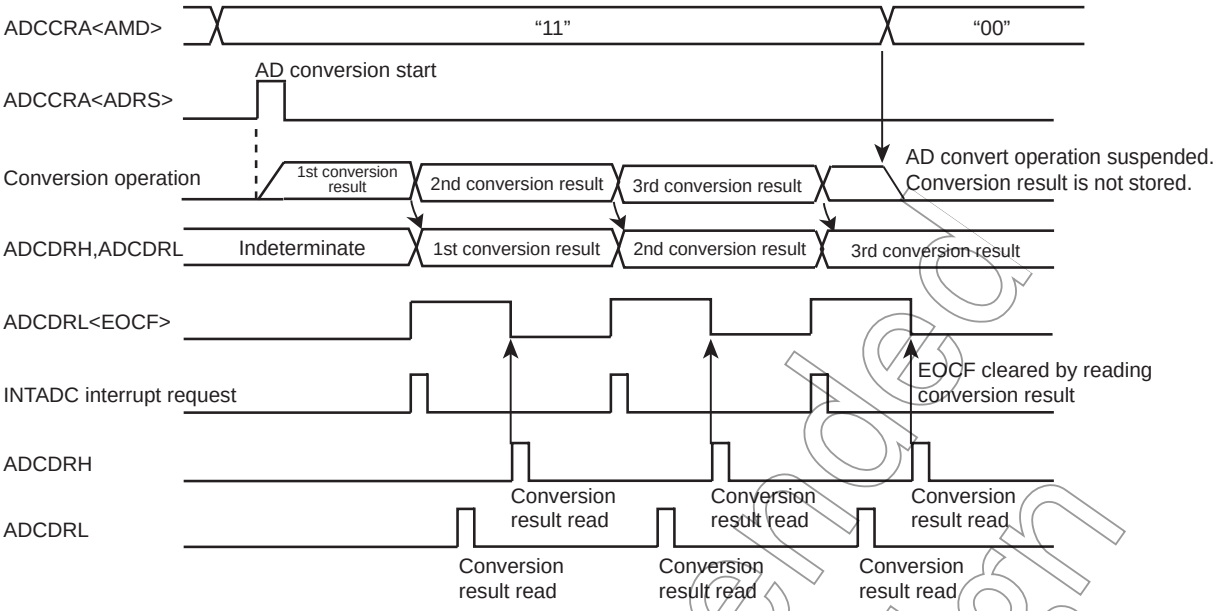


Figure 17-3 Repeat Mode

17.3.2.1 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 Table 17-1, Table 17-2 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 appropriately 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
```

17.3.3 INTPWM Start Mode

AD conversion can be started by the PWM interrupt (INTPWM) provided as part of the PMD function.

To use INTPWM start mode, set ADCCRA<AMD> to "10" to put the AD converter in a standby state. While the AD converter is in this state, generation of an INTPWM interrupt set ADCCRA<ADRS> to "1" and starts AD conversion.

After the AD conversion operation is completed, the conversion result is stored in the AD conversion value registers (ADCDRH, ADCDRL). At the same time, ADCDRL<EOCF> is set to "1" and an end-of-conversion interrupt (INTADC) is generated.

To perform the next AD conversion in INTPWM start mode, set ADCCRA<AINDS> to "1" (to analog input disable) and then set ADCCRA<AINDS> to "0" (to analog input enable) to put the AD converter in a standby state.

To stop AD conversion, write "0" to ADCCRA<AMD>. This will immediately stop the AD conversion operation in progress. At this time, the conversion value is not stored in the AD conversion value registers, and the AD converter immediately exits a standby state.

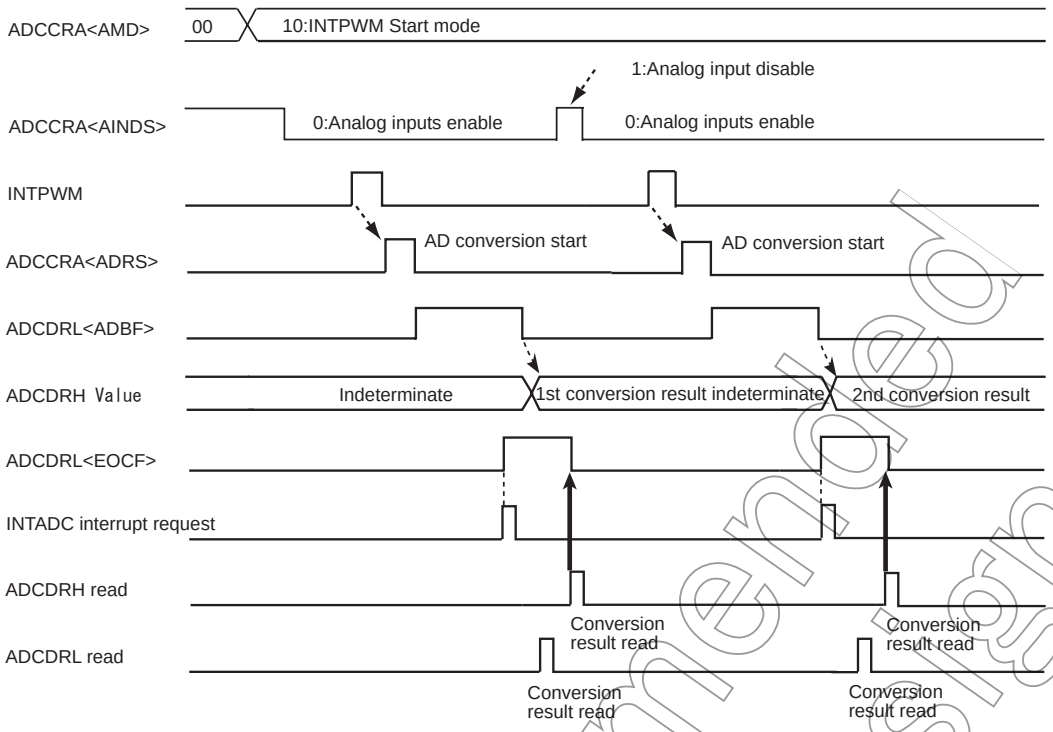


Figure 17-4 INTPWM Start Mode

17.3.3.1 Register Setting

1. Set the AINDS (analog input control) bit to "1" to disable analog input.
2. Set the AD converter control register 1 (ADCCRA) as follows:
   - Select the AD conversion channel to be used in the SAIN (analog input channel select) field.
   - Set the AMD (AD operating mode) field to "10" to select INTPWM start mode.
3. Set the AD converter control register 2 (ADCCRB) as follows:
   - Specify the AD conversion time in the ACK field. For details, see Table 17-1 and Table 17-2.
   - Set the IREFON (DA converter connection control) bit as appropriate.
4. After making the above settings, set ADCCRA<AINDS> (analog input control) to "0" to put the AD converter in a standby state.
5. Upon generation of an INTPWM interrupt, ADCCRA<ADRS> (AD conversion start) is set to "1" AD conversion is immediately started.
6. After completion of the AD conversion operation, the end-of-conversion flag (EOCF) in the ADCDRL is set to "1" and the AD conversion result is stored in the AD conversion value registers 1 and 2 (ADCDRH and ADCDRL). At this time, an INTADC interrupt is generated.
7. When the conversion result is read from the AD conversion value register 1 (ADCDRH), the EOCF flag is cleared to "0". If the next AD conversion is performed before the ADCDRH is read, the EOCF flag is cleared to "0", but the previous conversion result is retained until the next conversion is completed.
8. After end of conversion in INTPWM start mode, it dose not put the AD converter in a standby state. To start the next AD conversion in INTPWM start mode, first set ADCCRA<AINDS> to "1" to disable analog input, then clear this bit again to put the AD converter in a standby state.

   To change the AD conversion operation, first clear ADCCRA<AMD> to "00" to disable AD operation, and then set ADCCRA again.

17.4 STOP mode during AD Conversion

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

17.5 Analog Input Voltage and AD Conversion Result

The analog input voltage is corresponded to the 10-bit digital value converted by the AD as shown in Figure 17-5.

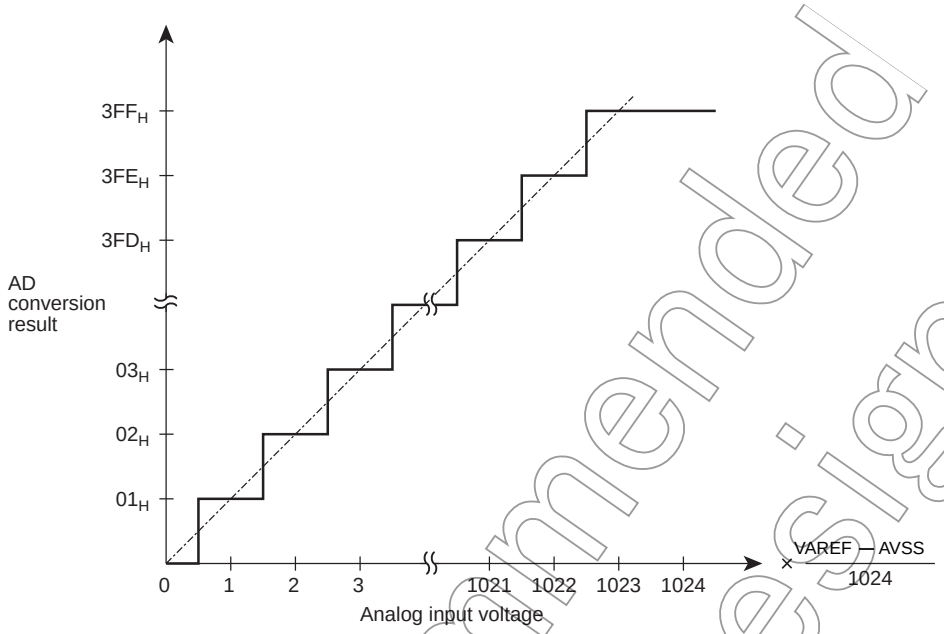


Figure 17-5 Analog Input Voltage and AD Conversion Result (Typ.)

17.6 Precautions about AD Converter

17.6.1 Analog input pin voltage range

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

17.6.2 Analog input shared pins

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

17.6.3 Noise Countermeasure

The internal equivalent circuit of the analog input pins is shown in Figure 17-6. 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.

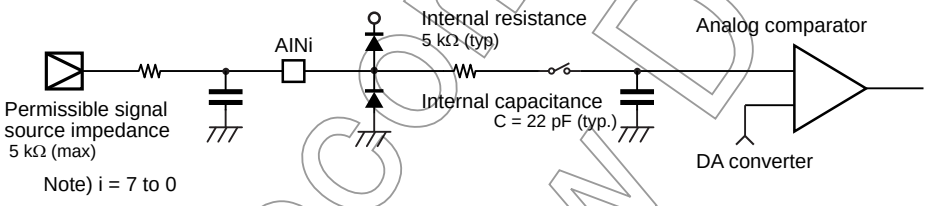


Figure 17-6 Analog Input Equivalent Circuit and Example of Input Pin Processing

18. Flash Memory

TMP88F846UG has 8192 bytes flash memory (address: 04000H to 05EFFH and FFF00H to FFFFH). The write and erase operations to the flash memory are controlled in the following three types of mode.

- MCU mode

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

- Serial PROM mode

  The flash memory is accessed by the CPU control in the serial PROM mode. Use of the serial interface (UART) enables the flash memory to be controlled by the small number of pins. TMP88F846UG in the serial PROM mode supports on-board programming which enables users to program flash memory after the microcontroller is mounted on a user board.

- Parallel PROM mode

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

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

18.1 Flash Memory Control

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

Flash Memory Control Register

FLSCR	7	6	5	4	3	2	1	0	
(01FFFH)	FLSMD								(Initial value : 1100 ****)

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

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

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

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

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

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

18.2 Command Sequence

The command sequence in the MCU and the serial PROM modes consists of six commands (JEDEC compatible), as shown in Table 18-1.

Table 18-1 Command Sequence

	Command Sequence	1st Bus Write Cycle		2nd Bus Write Cycle		3rd Bus Write Cycle		4th Bus Write Cycle		5th Bus Write Cycle		6th Bus Write Cycle	
		Address	Data	Address	Data	Address	Data	Address	Data	Address	Data	Address	Data
1	1 Byte program	04555H	AAH	04AAAH	55H	04555H	A0H	BA (Note 1)	Data (Note 1)	-	-	-	-
2	Sector Erase (4-kbyte Erase)	04555H	AAH	04AAAH	55H	04555H	80H	04555H	AAH	04AAAH	55H	SA (Note 2)	30H
3	Chip Erase (All Erase)	04555H	AAH	04AAAH	55H	04555H	80H	04555H	AAH	04AAAH	55H	04555H	10H
4	Product ID Entry	04555H	AAH	04AAAH	55H	04555H	90H	-	-	-	-	-	-
5	Product ID Exit	XXH	F0H	-	-	-	-	-	-	-	-	-	-
	Product ID Exit	04555H	AAH	04AAAH	55H	04555H	F0H	-	-	-	-	-	-
6	Security Program	04555H	AAH	04AAAH	55H	04555H	A5H	04F7FH	00H	-	-	-	-

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

Note 2: Set the address which is the specified sector. (The area to be erased is specified with the upper 8 bits of the address.)
(Example) In case " Sector 1 ", SA is " 0x05yyy ".

18.2.1 Byte Program

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

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

18.2.2 Sector Erase (4-kbyte Erase)

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

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

18.2.3 Chip Erase (All Erase)

This command erases the entire flash memory in approximately 30 ms. The next command sequence cannot be executed until the erase operation is completed. To check the completion of the erase operation, perform read

operations repeatedly for data polling until the same data is read twice from the same address in the flash memory. During the erase operation, any consecutive attempts to read from the same address is reversed bit 6 of the data (toggling between 0 and 1). After the chip is erased, all bytes contain FFH.

18.2.4 Product ID Entry

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

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

Cell	Address	Meaning	Read Value
1	04000H	Vendor ID	98H
	04001H	Flash macro ID	41H
	04002H	Flash size	1DH: 120 kbytes 0EH: 60 kbytes 0BH: 48 kbytes 07H: 32 kbytes 05H: 24 kbytes 03H: 16 kbytes 01H: 8 kbytes 00H: 4 kbytes
	04F7FH	Security program status	FFH: Security program disabled Other than FFH: Security program enabled

Note: The value at address 04002H (flash size) depends on the size of flash memory incorporated in each product. For example, if the product has 16-kbyte flash memory, "03H" is read from address 04002H.

18.2.5 Product ID Exit

This command is used to exit the Product ID mode.

18.2.6 Security Program

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

To disable the security program setting, it is necessary to execute the chip erase command sequence. Whether or not the security program is enabled can be checked by reading 04F7FH in the Product ID mode. For details, see Table 18-2.

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

18.3 Toggle Bit (D6)

After the byte program, chip erase, and security program command sequence is executed, any consecutive attempts to read from the same address is reversed bit 6 (D6) of the data (toggling between 0 and 1) until the operation is completed. Therefore, this toggle bit provides a software mechanism to check the completion of each operation.

Usually perform read operations repeatedly for data polling until the same data is read twice from the same address in the flash memory. After the byte program, chip erase, or security program command sequence is executed, the initial read of the toggle bit always produces a "1".

18.4 Access to the Flash Memory Area

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

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

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

18.4.1 Flash Memory Control in the Serial PROM Mode

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

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

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

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

1. Transfer the write control program to the RAM area in the RAM loader mode.
2. Jump to the RAM area.
3. Disable (DI) the interrupt master enable flag (IMF←"0").
4. Set FLSCR<FLSMD> to "0011B" (to enable command sequence execution).
5. Execute the erase command sequence.
6. Read the same flash memory address twice.

   (Repeat step 6 until the same data is read by two consecutive reads operations.)
7. Execute the write command sequence.
8. Read the same flash memory address twice.

   (Repeat step 8 until the same data is read by two consecutive reads operations.)
9. Set FLSCR<FLSMD> to "1100B" (to disable command sequence execution).

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

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

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

```
                DI                          : Disable interrupts (IMF← "0")
                LD      (FLSCR),0011_1000B  : Enable command sequence execution.
                LD      IX,04555H
                LD      IY,04AAAH
                LD      HL,0F000H
; #### Flash Memory Chip erase Process ####
                LD      (IX),0AAH           : 1st bus write cycle
                LD      (IY),55H            : 2nd bus write cycle
                LD      (IX),80H            : 3rd bus write cycle
                LD      (IX),0AAH           : 4th bus write cycle
                LD      (IY),55H            : 5th bus write cycle
                LD      (IX),10H            : 6th bus write cycle
sLOOP1:         LD      A,(IX)
                CMP     A,(IX)
                JR      NZ,sLOOP1           : Loop until the same value is read.
; #### Flash Memory Write Process ####
                LD      (IX),0AAH           : 1st bus write cycle
                LD      (IY),55H            : 2nd bus write cycle
                LD      (IX),0A0H           : 3rd bus write cycle
                LD      (HL),3FH            : 4th bus write cycle, (F000H) = 3FH
sLOOP2:         LD      A,(HL)
                CMP     A,(HL)
                JR      NZ,sLOOP2           : Loop until the same value is read.
                LD      (FLSCR),1100_1000B  : Disable command sequence execution.
sLOOP3:         JP      sLOOP3
```

18.4.2 Flash Memory Control in the MCU mode

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

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

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

1. Transfer the write control program to the RAM area.
2. Jump to the RAM area.
3. Disable (DI) the interrupt master enable flag (IMF←"0").
4. Disable the watchdog timer, if it is used.
5. Set FLSCR<FLSMD> to "0011B" (to enable command sequence execution).
6. Execute the erase command sequence.
7. Read the same flash memory address twice.

   (Repeat step 7 until the same data is read by two consecutive read operations.)
8. Execute the write command sequence.
9. Read the same flash memory address twice.

   (Repeat step 9 until the same data is read by two consecutive read operations.)
10. Set FLSCR<FLSMD> to "1100B" (to disable command sequence execution).
11. Jump to the flash memory area.

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

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

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

```
                DI                          : Disable interrupts (IMF←"0")
                LD      (WDTCR2),4EH        : Clear the WDT binary counter.
                LDW     (WDTCR1),0B101H     : Disable the WDT.
                LD      (FLSCR),0011_1000B  : Enable command sequence execution.
                LD      IX,04555H
                LD      IY,04AAAH
                LD      HL,0E000H
; #### Flash Memory Sector Erase Process ####
                LD      (IX),0AAH           : 1st bus write cycle
                LD      (IY),55H            : 2nd bus write cycle
                LD      (IX),80H            : 3rd bus write cycle
                LD      (IX),0AAH           : 4th bus write cycle
                LD      (IY),55H            : 5th bus write cycle
                LD      (HL),30H            : 6th bus write cycle
sLOOP1:         LD      A,(IX)
                CMP     A,(IX)
                JR      NZ,sLOOP1           : Loop until the same value is read.
; #### Flash Memory Write Process ####
                LD      (IX),0AAH           : 1st bus write cycle
                LD      (IY),55H            : 2nd bus write cycle
                LD      (IX),0A0H           : 3rd bus write cycle
                LD      (HL),3FH            : 4th bus write cycle, (E000H)=3FH
sLOOP2:         LD      A,(HL)
                CMP     A,(HL)
                JR      NZ,sLOOP2           : Loop until the same value is read.
                LD      (FLSCR),1100_1000B  : Disable command sequence execution.
                JP      XXXXH               : Jump to the flash memory area.
```

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

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

19. Serial PROM Mode

19.1 Outline

The TMP88F846UG has a 4096 byte BOOTROM (Mask ROM) for programming to flash memory. The BOOT-ROM is available in the serial PROM mode, and controlled by TEST, BOOT, $\overline{\text{RESET}}$ and some pins. Communication is performed via UART. The serial PROM mode has seven types of operating mode: Flash memory writing, RAM loader, Flash memory SUM output, Product ID code output, Flash memory status output, Flash memory erasing and Flash memory security program setting. Memory address mapping in the serial PROM mode differs from that in the MCU mode. Figure 19-1 shows memory address mapping in the serial PROM mode.

Table 19-1 Operating Range in the Serial PROM Mode

Parameter	Min	Max	Unit
Power supply	4.5	5.5	V
High frequency (Note)	8	20	MHz

Note 1: Though included in above operating range, some of high frequencies are not supported in the serial PROM mode. For details, refer to “Table 19-5”.

Note 2: As for TMP88F846UG, high frequencies becomes 20MHz fixation by a built-in oscillator. Moreover, there is a case where communication becomes impossible, since the frequency of a built-in oscillator is changed with temperature etc. Please use in normal temperature (around 25 degrees) as much as possible.

19.2 Memory Mapping

The Figure 19-1 shows memory mapping in the Serial PROM mode and MCU mode.

In the serial PROM mode, the BOOTROM (Mask ROM) is mapped in addresses from 3000H to 3FFFH.

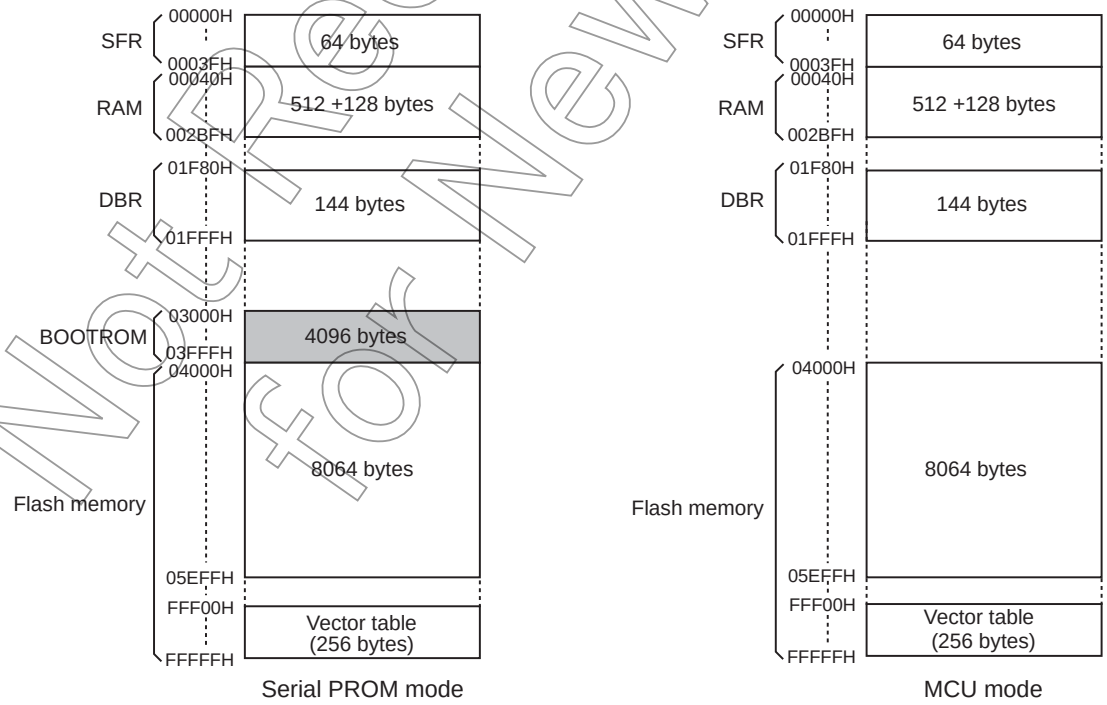


Figure 19-1 Memory Address Maps

19.3 Serial PROM Mode Setting

19.3.1 Serial PROM Mode Control Pins

To execute on-board programming, activate the serial PROM mode. Table 19-2 shows pin setting to activate the serial PROM mode.

Table 19-2 Serial PROM Mode Setting

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

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

19.3.2 Pin Function

In the serial PROM mode, TXD (P45) and RXD (P44) are used as a serial interface pin.

Table 19-3 Pin Function in the Serial PROM Mode

Pin Name (Serial PROM Mode)	Input/ Output	Function		Pin Name (MCU Mode)
TXD	Output	Serial data output (Note 3)	(Note 1)	P45
BOOT/RXD	Input/Input	Serial PROM mode control/Serial data input		P44
$\overline{\text{RESET}}$	Input	Serial PROM mode control		$\overline{\text{RESET}}$
TEST	Input	Fixed to high		TEST
VDD, AVDD	Power supply	4.5 to 5.5 V		
VSS, AVSS	Power supply	0 V		
VAREF	Power supply	Leave open or apply input reference voltage.		
I/O ports except P45, P44	I/O	These ports are in the high-impedance state in the serial PROM mode. The input level is fixed to the port inputs with a hardware feature to prevent overlap current. (The port inputs are invalid.) To make the port inputs valid, set the pin of the SPCR register to “1” by the RAM loader control program.		

Note 1: During on-board programming with other parts mounted on a user board, be careful no to affect these communication control pins.

Note 2: TXD port is set a open-drain port in seriarl PROM mode and it needs pull-up resistor.

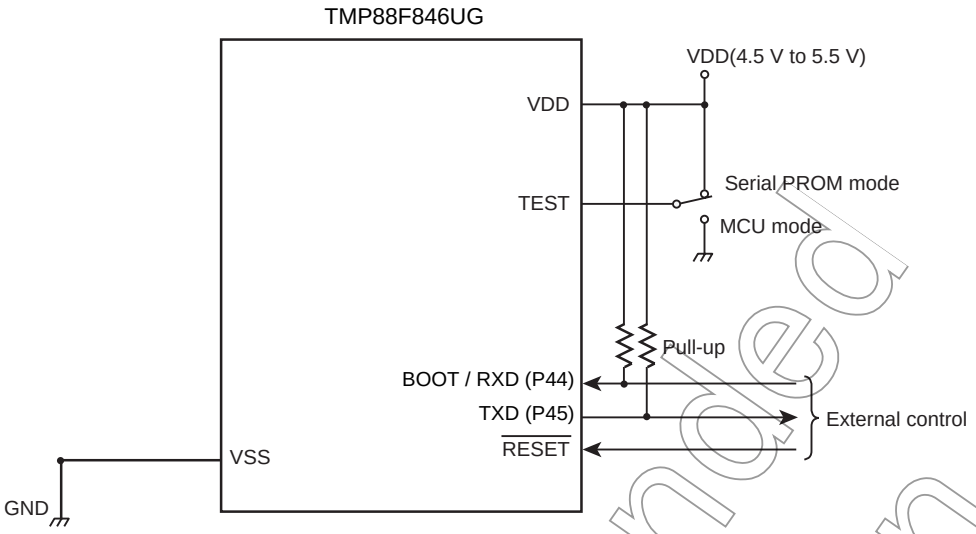


Figure 19-2 Serial PROM Mode Pin Setting

Note 1: For connection of other pins, refer to "Table 19-3 Pin Function in the Serial PROM Mode".

19.3.3 Example Connection for On-Board Writing

Figure 19-3 shows an example connection to perform on-board wring.

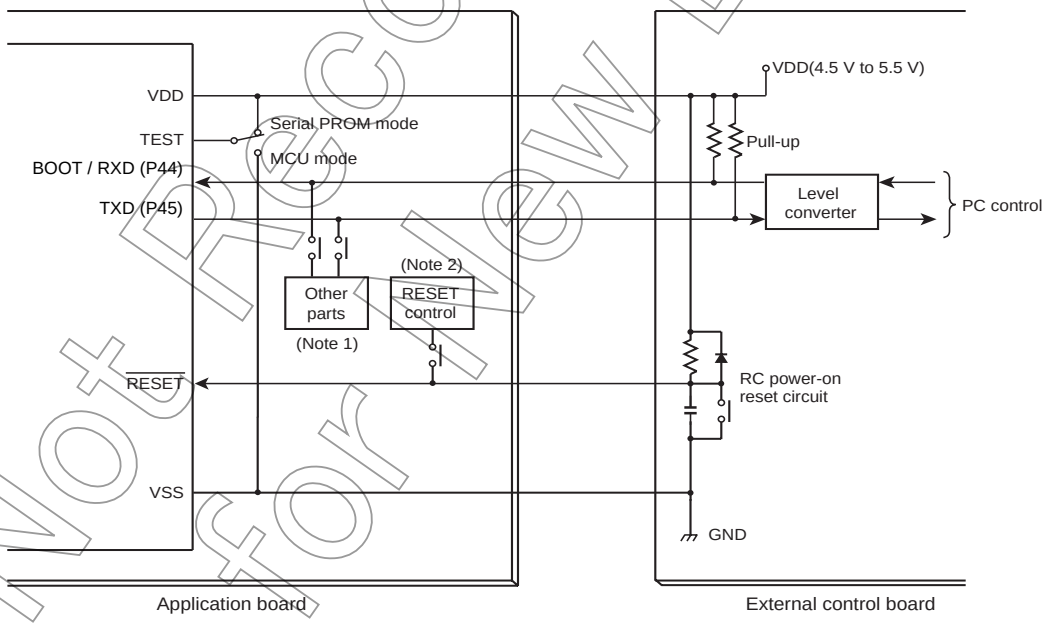


Figure 19-3 Example Connection for On-Board Writing

Note 1: When other parts on the application board effect the UART communication in the serial PROM mode, isolate these pins by a jumper or switch.

Note 2: When the reset control circuit on the application board effects activation of the serial PROM mode, isolate the pin by a jumper or switch.

Note 3: For connection of other pins, refer to "Table 19-3 Pin Function in the Serial PROM Mode".

19.3.4 Activating the Serial PROM Mode

The following is a procedure to activate the serial PROM mode. "Figure 19-4 Serial PROM Mode Timing" shows a serial PROM mode timing.

1. Supply power to the VDD pin.
2. Set the $\overline{\text{RESET}}$ pin to low.
3. Set the TEST pin and BOOT/RXD pins to high.
4. Wait until the power supply and clock oscillation stabilize.
5. Set the $\overline{\text{RESET}}$ pin to high.
6. Input the matching data (5AH) to the BOOT/RXD pin after setup sequence. For details of the setup timing, refer to "19.16 UART Timing".

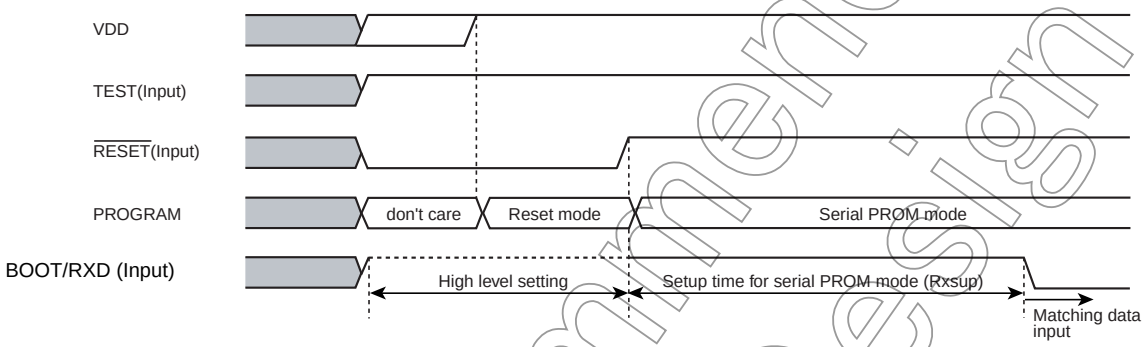


Figure 19-4 Serial PROM Mode Timing

19.4 Interface Specifications for UART

The following shows the UART communication format used in the serial PROM mode.

To perform on-board programming, the communication format of the write controller must also be set in the same manner.

The default baud rate is 9600 bps regardless of operating frequency of the microcontroller. The baud rate can be modified by transmitting the baud rate modification data shown in Table 19-4 to TMP88F846UG. The Table 19-5 shows an operating frequency and baud rate. The frequencies which are not described in Table 19-5 can not be used.

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

Table 19-4 Baud Rate Modification Data

Baud rate modification data	04H	05H	06H	07H	0AH	18H	28H
Baud rate (bps)	76800	62500	57600	38400	31250	19200	9600

Table 19-5 Operating Frequency and Baud Rate in the Serial PROM Mode

(Note 3)	Reference Baud Rate (bps)		76800		62500		57600		38400		31250		19200		9600	
	Baud Rate Modification Data		04H		05H		06H		07H		0AH		18H		28H	
	Ref. Frequency (MHz)	Rating (MHz)	Baud rate (bps)	(%)	(bps)	(%)	(bps)	(%)	(bps)	(%)	(bps)	(%)	(bps)	(%)	(bps)	(%)
7	20	18.80 to 20.64	76923	+1.73	-	-	-	-	39063	+0.16	31250	0.00	19531	+1.73	9766	+1.73

Note 1: “Ref. Frequency” and “Rating” show frequencies available in the serial PROM mode. Though the frequency is supported in the serial PROM mode, the serial PROM mode may not be activated correctly due to the frequency difference in the external controller (such as personal computer) and oscillator, and load capacitance of communication pins.

Note 2: It is recommended that the total frequency difference is within ±3% so that auto detection is performed correctly by the reference frequency.

Note 3: The external controller must transmit the matching data (5AH) repeatedly till the auto detection of baud rate is performed. This number indicates the number of times the matching data is transmitted for each frequency.

19.5 Operation Command

The eight commands shown in Table 19-6 are used in the serial PROM mode. After reset release, the TMP88F846UG waits for the matching data (5AH).

Table 19-6 Operation Command in the Serial PROM Mode

Command Data	Operating Mode	Description
5AH	Setup	Matching data. Execute this command after releasing the reset.
F0H	Flash memory erasing	Erases the flash memory area (address 4000H to 05EFFH, and FFF00 to FFFFFH).
30H	Flash memory writing	Writes to the flash memory area (address 4000H to 05EFFH, and FFF00 to FFFFFH) .
60H	RAM loader	Writes to the specified RAM area (address 0040H to 10BFH).
90H	Flash memory SUM output	Outputs the 2-byte checksum upper byte and lower byte in this order for the entire area of the flash memory (address 4000H to 05EFFH, and FFF00 to FFFFFH).
C0H	Product ID code output	Outputs the product ID code (13-byte data).
C3H	Flash memory status output	Outputs the status code (7-byte data) such as the security program condition.
FAH	Flash memory security program setting	Enables the security program.

19.6 Operation Mode

The serial PROM mode has seven types of modes, that are (1) Flash memory erasing, (2) Flash memory writing, (3) RAM loader, (4) Flash memory SUM output, (5) Product ID code output, (6) Flash memory status output and (7) Flash memory security program setting modes. Description of each mode is shown below.

1. Flash memory erasing mode

   The flash memory is erased by the chip erase (erasing an entire flash area) or sector erase (erasing sectors in 4-kbyte units). The erased area is filled with FFH. When the security program is enabled, the sector erase in the flash erasing mode can not be performed. To disable the security program, perform the chip erase. Before erasing the flash memory, TMP88F846UG checks the passwords except a blank product. If the password is not matched, the flash memory erasing mode is not activated.

2. Flash memory writing mode

   Data is written to the specified flash memory address for each byte unit. The external controller must transmit the write data in the Intel Hex format (Binary). If no error is encountered till the end record, TMP88F846UG calculates the checksum for the entire flash memory area (4000H to 05EFFH, and FFF00H to FFFFFH), and returns the obtained result to the external controller. When the security program is enabled, the flash memory writing mode is not activated. In this case, perform the chip erase command beforehand in the flash memory erasing mode. Before activating the flash memory writing mode, TMP88F846UG checks the password except a blank product. If the password is not matched, flash memory writing mode is not activated.

3. RAM loader mode

   The RAM loader transfers the data in Intel Hex format sent from the external controller to the internal RAM. When the transfer is completed normally, the RAM loader calculates the checksum. After transmitting the results, the RAM loader jumps to the RAM address specified with the first data record in order to execute the user program. When the security program is enabled, the RAM loader mode is not activated. In this case, perform the chip erase beforehand in the flash memory erasing mode. Before activating the RAM loader mode, TMP88F846UG checks the password except a blank product. If the password is not matched, flash RAM loader mode is not activated.

4. Flash memory SUM output mode

   The checksum is calculated for the entire flash memory area (4000H to 05EFFH, and FFF00H to FFFFFH), and the result is returned to the external controller. Since the BOOTROM does not support the operation command to read the flash memory, use this checksum to identify programs when managing revisions of application programs.

5. Product ID code output mode

   The code used to identify the product is output. The code to be output consists of 21-byte data, which includes the information indicating the area of the ROM incorporated in the product. The external controller reads this code, and recognizes the product to write.

   (In the case of TMP88F846UG, the addresses from 4000H to 05EFFH, and FFF00H to FFFFFH become the ROM area.)

6. Flash memory status output mode

   The status of the area from FFFE0H to FFFFFH, and the security program condition are output as 7-byte code. The external controller reads this code to recognize the flash memory status.

7. Flash memory security program setting mode

   This mode disables reading and writing the flash memory data in parallel PROM mode. In the serial PROM mode, the flash memory writing and RAM loader modes are disabled. To disable the flash memory security program, perform the chip erase in the flash memory erasing mode.

19.6.1 Flash Memory Erasing Mode (Operating command: F0H)

Table 19-7 shows the flash memory erasing mode.

Table 19-7 Flash Memory Erasing Mode

	Transfer Byte	Transfer Data from the External Controller to TMP88F846UG	Baud Rate	Transfer Data from TMP88F846UG to the External Controller
BOOT ROM	1st byte 2nd byte	Matching data (5AH) -	9600 bps 9600 bps	- (Automatic baud rate adjustment) OK: Echo back data (5AH) Error: No data transmitted
	3rd byte 4th byte	Baud rate change data (Table 19-4) -	9600 bps 9600 bps	- OK: Echo back data Error: A1H × 3, A3H × 3, 62H × 3 (Note 1)
	5th byte 6th byte	Operation command data (F0H) -	Modified baud rate Modified baud rate	- OK: Echo back data (F0H) Error: A1H × 3, A3H × 3, 63H × 3 (Note 1)
	7th byte 8th byte	Password count storage address bit 23 to 16 (Note 4, 5)	Modified baud rate Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	9th byte 10th byte	Password count storage address bit 15 to 08 (Note 4, 5)	Modified baud rate Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	11th byte 12th byte	Password count storage address bit 07 to 00 (Note 4, 5)	Modified baud rate Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	13th byte 14th byte	Password comparison start address bit 23 to 16 (Note 4, 5)	Modified baud rate Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	15th byte 16th byte	Password comparison start address bit 15 to 08 (Note 4, 5)	Modified baud rate Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	17th byte 18th byte	Password comparison start address bit 07 to 00 (Note 4, 5)	Modified baud rate Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	19th byte : m'th byte	Password string (Note 4, 5) -	Modified baud rate Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	n'th - 2 byte	Erase area specification (Note 2)	Modified baud rate	-
	n'th - 1 byte	-	Modified baud rate	OK: Checksum (Upper byte) (Note 3) Error: Nothing transmitted
	n'th byte	-	Modified baud rate	OK: Checksum (Lower byte) (Note 3) Error: Nothing transmitted
	n'th + 1 byte	(Wait for the next operation command data)	Modified baud rate	-

Note 1: "xxH × 3" indicates that the device enters the halt condition after transmitting 3 bytes of xxh.

Note 2: Refer to "19.13 Specifying the Erasure Area".

Note 3: Refer to "19.8 Checksum (SUM)".

Note 4: Refer to "19.10 Passwords".

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

Note 6: When a password error occurs, TMP88F846UG stops UART communication and enters the halt mode. Therefore, when a password error occurs, initialize TMP88F846UG by the $\overline{\text{RESET}}$ pin and reactivate the serial PROM mode.

Note 7: If an error occurs during transfer of a password address or a password string, TMP88F846UG stops UART communication and enters the halt condition. Therefore, when a password error occurs, initialize TMP88F846UG by the $\overline{\text{RESET}}$ pin and reactivate the serial PROM mode.

Description of the flash memory erasing mode

1. The 1st through 4th bytes of the transmitted and received data contain the same data as in the flash memory writing mode.
2. The 5th byte of the received data contains the command data in the flash memory erasing mode (F0H).
3. When the 5th byte of the received data contains the operation command data shown in Table 19-6, the device echoes back the value which is the same data in the 6th byte position of the received data (in this case, F0H). If the 5th byte of the received data does not contain the operation command data, the device enters the halt condition after sending 3 bytes of the operation command error code (63H).
4. The 7th thorough m'th bytes of the transmitted and received data contain the same data as in the flash memory writing mode. In the case of a blank product, do not transmit a password string. (Do not transmit a dummy password string.)
5. The n’th - 2 byte contains the erasure area specification data. The upper 4 bits and lower 4 bits specify the start address and end address of the erasure area, respectively. For the detailed description, see "19.13 Specifying the Erasure Area" .
6. The n’th - 1 byte and n’th byte contain the upper and lower bytes of the checksum, respectively. For how to calculate the checksum, refer to "19.8 Checksum (SUM)" . Checksum is calculated unless a receiving error or Intel Hex format error occurs. After sending the end record, the external controller judges whether the transmission is completed correctly by receiving the checksum sent by the device.
7. After sending the checksum, the device waits for the next operation command data.

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

Table 19-8 shows flash memory writing mode process.

Table 19-8 Flash Memory Writing Mode Process

	Transfer Byte	Transfer Data from External Controller to TMP88F846UG	Baud Rate	Transfer Data from TMP88F846UG to External Controller
BOOT ROM	1st byte 2nd byte	Matching data (5Ah) -	9600 bps 9600 bps	- (Automatic baud rate adjustment) OK: Echo back data (5AH) Error: Nothing transmitted
	3rd byte 4th byte	Baud rate modification data (See Table 19-4) -	9600 bps 9600 bps	- OK: Echo back data Error: A1H × 3, A3H × 3, 62H × 3 (Note 1)
	5th byte 6th byte	Operation command data (30H) -	Modified baud rate Modified baud rate	- OK: Echo back data (30H) Error: A1H × 3, A3H × 3, 63H × 3 (Note 1)
	7th byte 8th byte	Password count storage address bit 23 to 16 (Note 4)	Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	9th byte 10th byte	Password count storage address bit 15 to 08 (Note 4)	Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	11th byte 12th byte	Password count storage address bit 07 to 00 (Note 4)	Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	13th byte 14th byte	Password comparison start address bit 23 to 16 (Note 4)	Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	15th byte 16th byte	Password comparison start address bit 15 to 08 (Note 4)	Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	17th byte 18th byte	Password comparison start address bit 07 to 00 (Note 4)	Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	19th byte : m'th byte	Password string (Note 5) -	Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	m'th + 1 byte : n'th - 2 byte	Intel Hex format (binary) (Note 2)	Modified baud rate	- -
	n'th - 1 byte	-	Modified baud rate	OK: SUM (Upper byte) (Note 3) Error: Nothing transmitted
	n'th byte	-	Modified baud rate	OK: SUM (Lower byte) (Note 3) Error: Nothing transmitted
	n'th + 1 byte	(Wait state for the next operation command data)	Modified baud rate	-

Note 1: "xxH × 3" indicates that the device enters the halt condition after sending 3 bytes of xxH. For details, refer to "19.7 Error Code".

Note 2: Refer to "19.9 Intel Hex Format (Binary)".

Note 3: Refer to "19.8 Checksum (SUM)".

Note 4: Refer to "19.10 Passwords".

Note 5: If addresses from FFFE0H to FFFFFH are filled with "FFH", the passwords are not compared because the device is considered as a blank product. Transmitting a password string is not required. Even in the case of a blank product , it is required to specify the password count storage address and the password comparison start address. Transmit these data from the external controller. If a password error occurs due to incorrect password count storage address or password

comparison start address, TMP88F846UG stops UART communication and enters the halt condition. Therefore, when a password error occurs, initialize TMP88F846UG by the $\overline{\text{RESET}}$ pin and reactivate the serial ROM mode.

Note 6: If the security program is enabled or a password error occurs, TMP88F846UG stops UART communication and enters the halt confition. In this case, initialize TMP88F846UG by the $\overline{\text{RESET}}$ pin and reactivate the serial ROM mode.

Note 7: If an error occurs during the reception of a password address or a password string, TMP88F846UG stops UART communication and enters the halt condition. In this case, initialize TMP88F846UG by the $\overline{\text{RESET}}$ pin and reactivate the serial PROM mode.

Note 8: Do not write only the address from FFFE0H to FFFFFH when all flash memory data is the same. If only these area are written, the subsequent operation can not be executed due to password error.

Note 9: To rewrite data to Flash memory addresses at which data (including FFH) is already written, make sure to erase the existing data by "sector erase" or "chip erase" before rewriting data.

Description of the flash memory writing mode

1. The 1st byte of the received data contains the matching data. When the serial PROM mode is activated, TMP88F846UG (hereafter called device), waits to receive the matching data (5AH). Upon reception of the matching data, the device automatically adjusts the UART's initial baud rate to 9600 bps.
2. When receiving the matching data (5AH), the device transmits an echo back data (5AH) as the second byte data to the external controller. If the device can not recognize the matching data, it does not transmit the echo back data and waits for the matching data again with automatic baud rate adjustment. Therefore, the external controller should transmit the matching data repeatedly till the device transmits an echo back data. The transmission repetition count varies depending on the frequency of device. For details, refer to Table 19-5.
3. The 3rd byte of the received data contains the baud rate modification data. The five types of baud rate modification data shown in Table 19-4 are available. Even if baud rate is not modified, the external controller should transmit the initial baud rate data (28H: 9600 bps).
4. Only when the 3rd byte of the received data contains the baud rate modification data corresponding to the device's operating frequency, the device echoes back data the value which is the same data in the 4th byte position of the received data. After the echo back data is transmitted, baud rate modification becomes effective. If the 3rd byte of the received data does not contain the baud rate modification data, the device enters the halts condition after sending 3 bytes of baud rate modification error code (62H).
5. The 5th byte of the received data contains the command data (30H) to write the flash memory.
6. When the 5th byte of the received data contains the operation command data shown in Table 19-6, the device echoes back the value which is the same data in the 6th byte position of the received data (in this case, 30H). If the 5th byte of the received data does not contain the operation command data, the device enters the halt condition after sending 3 bytes of the operation command error code (63H).
7. The 7th byte contains the data for 23 to 16 bits of the password count storage address. When the data received with the 7th byte has no receiving error, the device does not send any data. If a receiving error or password error occurs, the device does not send any data and enters the halt condition.
8. The 9th byte contains the data for 15 to 8 bits of the password count storage address. When the data received with the 9th byte has no receiving error, the device does not send any data. If a receiving error or password error occurs, the device does not send any data and enters the halt condition.
9. The 11th byte contains the data for 7 to 0 bits of the password count storage address. When the data received with the 11th byte has no receiving error, the device does not send any data. If a receiving error or password error occurs, the device does not send any data and enters the halt condition.
10. The 13th byte contains the data for 23 to 16 bits of the password comparison start address. When the data received with the 13th byte has no receiving error, the device does not send any data. If a receiving error or password error occurs, the device does not send any data and enters the halt condition.
11. The 15th byte contains the data for 15 to 8 bits of the password comparison start address. When the data received with the 15th byte has no receiving error, the device does not send any data. If a receiving error or password error occurs, the device does not send any data and enters the halt condition.
12. The 17th byte contains the data for 7 to 0 bits of the password comparison start address. When the data received with the 17th byte has no receiving error, the device does not send any data. If a receiving error or password error occurs, the device does not send any data and enters the halt condition.
13. The 19th through m'th bytes contain the password data. The number of passwords becomes the data (N) stored in the password count storage address. The external password data is compared with N-byte data from the address specified by the password comparison start address. The external controller should

send N-byte password data to the device. If the passwords do not match, the device enters the halt condition without returning an error code to the external controller. If the addresses from FFFE0H to FFFFFH are filled with “FFH”, the passwords are not conpared because the device is considered as a blank product.

14. The m’th + 1 through n’th - 2 bytes of the received data contain the binary data in the Intel Hex format. No received data is echoed back to the external controller. After receiving the start mark (3AH for “:”) in the Intel Hex format, the device starts data record reception. Therefore, the received data except 3AH is ignored until the start mark is received. After receiving the start mark, the device receives the data record, that consists of data length, address, record type, write data and checksum. Since the device starts checksum calculation after receiving an end record, the external controller should wait for the checksum after sending the end record. If a receiving error or Intel Hex format error occurs, the device enters the halts condition without returning an error code to the external controller.
15. The n’th - 1 and n’th bytes contain the checksum upper and lower bytes. For details on how to calculate the SUM, refer to "19.8 Checksum (SUM)". The checksum is calculated only when the end record is detected and no receiving error or Intel Hex format error occurs. After sending the end record, the external controller judges whether the transmission is completed correctly by receiving the checksum sent by the device.
16. After transmitting the checksum, the device waits for the next operation command data.

19.6.3 RAM Loader Mode (Operation Command: 60H)

Table 19-9 shows RAM loader mode process.

Table 19-9 RAM Loader Mode Process

	Transfer Bytes	Transfer Data from External Controller to TMP88F846UG	Baud Rate	Transfer Data from TMP88F846UG to External Controller
BOOT ROM	1st byte 2nd byte	Matching data (5AH) -	9600 bps 9600 bps	- (Automatic baud rate adjustment) OK: Echo back data (5AH) Error: Nothing transmitted
	3rd byte 4th byte	Baud rate modification data (See Table 19-4) -	9600 bps 9600 bps	- OK: Echo back data Error: A1H × 3, A3H × 3, 62H × 3 (Note 1)
	5th byte 6th byte	Operation command data (60H) -	Modified baud rate Modified baud rate	- OK: Echo back data (60H) Error: A1H × 3, A3H × 3, 63H × 3 (Note 1)
	7th byte 8th byte	Password count storage address bit 23 to 16 (Note 4)	Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	9th byte 10th byte	Password count storage address bit 15 to 08 (Note 4)	Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	11th byte 12th byte	Password count storage address bit 07 to 00 (Note 4)	Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	13th byte 14th byte	Password comparison start address bit 23 to 16 (Note 4)	Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	15th byte 16th byte	Password comparison start address bit 15 to 08 (Note 4)	Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	17th byte 18th byte	Password comparison start address bit 07 to 00 (Note 4)	Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	19th byte : m'th byte	Password string (Note 5) -	Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	m'th + 1 byte : n'th - 2 byte	Intel Hex format (Binary) (Note 2)	Modified baud rate Modified baud rate	- -
	n'th - 1 byte	-	Modified baud rate	OK: SUM (Upper byte) (Note 3) Error: Nothing transmitted
	n'th byte	-	Modified baud rate	OK: SUM (Lower byte) (Note 3) Error: Nothing transmitted
RAM	-	The program jumps to the start address of RAM in which the first transferred data is written.		

Note 1: "xxH × 3" indicates that the device enters the halt condition after sending 3 bytes of xxH. For details, refer to "19.7 Error Code".

Note 2: Refer to "19.9 Intel Hex Format (Binary)".

Note 3: Refer to "19.8 Checksum (SUM)".

Note 4: Refer to "19.10 Passwords".

Note 5: If addresses from FFFE0H to FFFFFH are filled with "FFH", the passwords are not compared because the device is considered as a blank product. Transmitting a password string is not required. Even in the case of a blank product , it is required to specify the password count storage address and the password comparison start address. Transmit these data

from the external controller. If a password error occurs due to incorrect password count storage address or password comparison start address, TMP88F846UG stops UART communication and enters the halt condition. Therefore, when a password error occurs, initialize TMP88F846UG by the $\overline{\text{RESET}}$ pin and reactivate the serial ROM mode.

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

Note 7: If the security program is enabled or a password error occurs, TMP88F846UG stops UART communication and enters the halt condition. In this case, initialize TMP88F846UG by the $\overline{\text{RESET}}$ pin and reactivate the serial PROM mode.

Note 8: If an error occurs during the reception of a password address or a password string, TMP88F846UG stops UART communication and enters the halt condition. In this case, initialize TMP88F846UG by the $\overline{\text{RESET}}$ pin and reactivate the serial PROM mode.

Note 9: To re-write data for the address of flash memory which has already written data (include "FF"), make sure to erase the data first by the sector erase or chip erase, and then write new data to the flash memory.

Description of RAM loader mode

1. The 1st through 4th bytes of the transmitted and received data contains the same data as in the flash memory writing mode.
2. In the 5th byte of the received data contains the RAM loader command data (60H).
3. When th 5th byte of the received data contains the operation command data shown in Table 19-6, the device echoes back the value which is the same data in the 6th byte position (in this case, 60H). If the 5th byte does not contain the operation command data, the device enters the halt condition after sending 3 bytes of operation command error code (63H).
4. The 7th through m'th bytes of the transmitted and received data contain the same data as in the flash memory writing mode.
5. The m'th + 1 through n'th - 2 bytes of the received data contain the binary data in the Intel Hex format. No received data is echoed back to the external controller. After receiving the start mark (3AH for ":") in the Intel Hex format, the device starts data record reception. Therefore, the received data except 3AH is ignored until the start mark is received. After receiving the start mark, the device receives the data record, that consists of data length, address, record type, write data and checksum. The writing data of the data record is written into RAM specified by address. Since the device starts checksum calculation after receiving an end record, the external controller should wait for the checksum after sending the end record. If a receiving error or Intel Hex format error occurs, the device enters the halts condition without returning an error code to the external controller.
6. The n'th - 1 and n'th bytes contain the checksum upper and lower bytes. For details on how to calculate the SUM, refer to "19.8 Checksum (SUM)". The checksum is calculated only when the end record is detected and no receiving error or Intel Hex format error occurs. After sending the end record, the external controller judges whether the transmission is completed correctly by receiving the checksum sent by the device.
7. After transmitting the checksum to the external controller, the boot program jumps to the RAM address that is specified by the first received data record.

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

Table 19-10 shows flash memory SUM output mode process.

Table 19-10 Flash Memory SUM Output Process

	Transfer Bytes	Transfer Data from External Controller to TMP88F846UG	Baud Rate	Transfer Data from TMP88F846UG to External Controller
BOOT ROM	1st byte 2nd byte	Matching data (5AH) -	9600 bps 9600 bps	- (Automatic baud rate adjustment) OK: Echo back data (5AH) Error: Nothing transmitted
	3rd byte 4th byte	Baud rate modification data (See Table 19-4) -	9600 bps 9600 bps	- OK: Echo back data Error: A1H × 3, A3H × 3, 62H × 3 (Note 1)
	5th byte 6th byte	Operation command data (90H) -	Modified baud rate Modified baud rate	- OK: Echo back data (90H) Error: A1H × 3, A3H × 3, 63H × 3 (Note 1)
	7th byte	-	Modified baud rate	OK: SUM (Upper byte) (Note 2) Error: Nothing transmitted
	8th byte	-	Modified baud rate	OK: SUM (Lower byte) (Note 2) Error: Nothing transmitted
	9th byte	(Wait for the next operation command data)	Modified baud rate	-

Note 1: “xxH × 3” indicates that the device enters the halt condition after sending 3 bytes of xxH. For details, refer to "19.7 Error Code".

Note 2: Refer to "19.8 Checksum (SUM)".

Description of the flash memory SUM output mode

1. The 1st through 4th bytes of the transmitted and received data contains the same data as in the flash memory writing mode.
2. The 5th byte of the received data contains the command data in the flash memory SUM output mode (90H).
3. When the 5th byte of the received data contains the operation command data shown in Table 19-6, the device echoes back the value which is the same data in the 6th byte position of the received data (in this case, 90H). If the 5th byte of the received data does not contain the operation command data, the device enters the halt condition after transmitting 3 bytes of operation command error code (63H).
4. The 7th and the 8th bytes contain the upper and lower bits of the checksum, respectively. For how to calculate the checksum, refer to "19.8 Checksum (SUM)".
5. After sending the checksum, the device waits for the next operation command data.

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

Table 19-11 shows product ID code output mode process.

Table 19-11 Product ID Code Output Process

	Transfer Bytes	Transfer Data from External Controller to TMP88F846UG	Baud Rate	Transfer Data from TMP88F846UG to External Controller	
BOOT ROM	1st byte 2nd byte	Matching data (5AH) -	9600 bps 9600 bps	- (Automatic baud rate adjustment) OK: Echo back data (5AH) Error: Nothing transmitted	
	3rd byte 4th byte	Baud rate modification data (Table 19-4) -	9600 bps 9600 bps	- OK: Echo back data Error: A1H × 3, A3H × 3, 62H × 3 (Note 1)	
	5th byte 6th byte	Operation command data (C0H) -	Modified baud rate Modified baud rate	- OK: Echo back data (C0H) Error: A1H × 3, A3H × 3, 63H × 3 (Note 1)	
	7th byte		Modified baud rate	3AH	Start mark
	8th byte		Modified baud rate	12H	The number of transfer data (from 9th to 26th bytes)
	9th byte		Modified baud rate	03H	Length of address (3 bytes)
	10th byte		Modified baud rate	3DH	Reserved data
	11th byte		Modified baud rate	00H	Reserved data
	12th byte		Modified baud rate	00H	Reserved data
	13th byte		Modified baud rate	00H	Reserved data
	14th byte		Modified baud rate	02H	ROM block count (2 block)
	15th byte		Modified baud rate	00H	First address of ROM block 1 (Upper byte)
	16th byte		Modified baud rate	40H	First address of ROM block 1 (Middle byte)
	17th byte		Modified baud rate	00H	First address of ROM block 1 (Lower byte)
	18th byte		Modified baud rate	00H	End address of ROM block 1 (Upper byte)
	19th byte		Modified baud rate	5EH	End address of ROM block 1 (Middle byte)
	20th byte		Modified baud rate	FFH	End address of ROM block 1 (Lower byte)
	21st byte		Modified baud rate	0FH	First address of ROM block 2 (Upper byte)
	22nd byte		Modified baud rate	FFH	First address of ROM block 2 (Middle byte)
	23rd byte		Modified baud rate	00H	First address of ROM block 2 (Lower byte)
	24th byte		Modified baud rate	0FH	End address of ROM block 2 (Upper byte)
	25th byte		Modified baud rate	FFH	End address of ROM block 2 (Middle byte)
	26th byte		Modified baud rate	FFH	End address of ROM block 2 (Lower byte)
	27th byte		Modified baud rate	06H	Checksum of transferred data (9th through 26th byte)
	28th byte	(Wait for the next operation command data)	Modified baud rate	-	

Note: “xxH × 3” indicates that the device enters the halt condition after sending 3 bytes of xxH. For details, refer to "19.7 Error Code".

Description of Product ID code output mode

1. The 1st through 4th bytes of the transmitted and received data contain the same data as in the flash memory writing mode.
2. The 5th byte of the received data contains the product ID code output mode command data (C0H).
3. When the 5th byte contains the operation command data shown in Table 19-6, the device echoes back the value which is the same data in the 6th byte position of the received data (in this case, C0H). If the 5th byte data does not contain the operation command data, the device enters the halt condition after sending 3 bytes of operation command error code (63H).
4. The 9th through 26th bytes contain the product ID code. For details, refer to "19.11 Product ID Code".
5. After sending the checksum, the device waits for the next operation command data.

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

Table 19-12 shows Flash memory status output mode process.

Table 19-12 Flash Memory Status Output Mode Process

	Transfer Bytes	Transfer Data from External Controller to TMP88F846UG	Baud Rate	Transfer Data from TMP88F846UG to External Controller
BOOT ROM	1st byte 2nd byte	Matching data (5AH) -	9600 bps 9600 bps	- (Automatic baud rate adjustment) OK: Echo back data (5AH) Error: Nothing transmitted
	3rd byte 4th byte	Baud rate modification data (See Table 19-4) -	9600 bps 9600 bps	- OK: Echo back data Error: A1H × 3, A3H × 3, 62H × 3 (Note 1)
	5th byte 6th byte	Operation command data (C3H) -	Modified baud rate Modified baud rate	- OK: Echo back data (C3H) Error: A1H × 3, A3H × 3, 63H × 3 (Note 1)
	7th byte		Modified baud rate	3AH Start mark
	8th byte		Modified baud rate	04H Byte count (from 9th to 12th byte)
	9th byte		Modified baud rate	00H to 03H Status code 1
	10th byte		Modified baud rate	00H Reserved data
	11th byte		Modified baud rate	00H Reserved data
	12th byte		Modified baud rate	00H Reserved data
	13th byte		Modified baud rate	Checksum 2's complement for the sum of 9th through 12th bytes 9th byte Checksum 00H: 00H 01H: FFH 02H: FEH 03H: FDH
	14th byte	(Wait for the next operation command data)	Modified baud rate	-

Note 1: "xxH × 3" indicates that the device enters the halt condition after sending 3 bytes of xxH. For details, refer to "19.7 Error Code".

Note 2: For the details on status code 1, refer to "19.12 Flash Memory Status Code".

Description of Flash memory status output mode

1. The 1st through 4th bytes of the transmitted and received data contain the same data as in the Flash memory writing mode.
2. The 5th byte of the received data contains the flash memory status output mode command data (C3H).
3. When the 5th byte contains the operation command data shown in Table 19-6, the device echoes back the value which is the same data in the 6th byte position of the received data (in this case, C3H). If the 5th byte does not contain the operation command data, the device enters the halt condition after sending 3 bytes of operation command error code (63H).
4. The 9th through 13th bytes contain the status code. For details on the status code, refer to "19.12 Flash Memory Status Code".
5. After sending the status code, the device waits for the next operation command data.

19.6.7 Flash Memory Security Program Setting Mode (Operation Command: FAH)

Table 19-13 shows Flash memory security program setting mode process.

Table 19-13 Flash Memory Security Program Setting Mode Process

	Transfer Bytes	Transfer Data from External Controller to TMP88F846UG	Baud Rate	Transfer Data from TMP88F846UG to External Controller
BOOT ROM	1st byte 2nd byte	Matching data (5AH) -	9600 bps 9600 bps	- (Automatic baud rate adjustment) OK: Echo back data (5AH) Error: Nothing transmitted
	3rd byte 4th byte	Baud rate modification data (See Table 19-4) -	9600 bps 9600 bps	- OK: Echo back data Error: A1H × 3, A3H × 3, 62H × 3 (Note 1)
	5th byte 6th byte	Operation command data (FAH) -	Modified baud rate Modified baud rate	- OK: Echo back data (FAH) Error: A1H × 3, A3H × 3, 63H × 3 (Note 1)
	7th byte 8th byte	Password count storage address 23 to 16 (Note 2)	Modified baud rate Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	9th byte 10th byte	Password count storage address 15 to 08 (Note 2)	Modified baud rate Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	11th byte 12th byte	Password count storage address 07 to 00 (Note 2)	Modified baud rate Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	13th byte 14th byte	Password comparison start address 23 to 16 (Note 2)	Modified baud rate Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	15th byte 16th byte	Password comparison start address 15 to 08 (Note 2)	Modified baud rate Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	17th byte 18th byte	Password comparison start address 07 to 00 (Note 2)	Modified baud rate Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	19th byte : m'th byte	Password string (Note 2) -	Modified baud rate Modified baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	n'th byte	-	Modified baud rate	OK: FBH (Note 3) Error: Nothing transmitted
	n'+1th byte	(Wait for the next operation command data)	Modified baud rate	-

Note 1: "xxH × 3" indicates that the device enters the halt condition after sending 3 bytes of xxH. For details, refer to "19.7 Error Code".

Note 2: Refer to "19.10 Passwords".

Note 3: If the security program is enabled for a blank product or a password error occurs for a non-blank product, TMP88F846UG stops UART communication and enters the halt mode. In this case, initialize TMP88F846UG by the $\overline{\text{RESET}}$ pin and reactivate the serial PROM mode.

Note 4: If an error occurs during reception of a password address or a password string, TMP88F846UG stops UART communication and enters the halt mode. In this case, initialize TMP88F846UG by the $\overline{\text{RESET}}$ pin and reactivate the serial PROM mode.

Description of the Flash memory security program setting mode

1. The 1st through 4th bytes of the transmitted and received data contain the same data as in the Flash memory writing mode.
2. The 5th byte of the received data contains the command data in the flash memory status output mode (FAH).
3. When the 5th byte of the received data contains the operation command data shown in Table 19-6, the device echoes back the value which is the same data in the 6th byte position of the received data (in this case, FAH). If the 5th byte does not contain the operation command data, the device enters the halt condition after transmitting 3 bytes of operation command error code (63H).
4. The 7th through m’th bytes of the transmitted and received data contain the same data as in the flash memory writing mode.
5. The n'th byte contains the status to be transmitted to the external controller in the case of the successful security program.

19.7 Error Code

When detecting an error, the device transmits the error code to the external controller, as shown in Table 19-14.

Table 19-14 Error Code

Transmit Data	Meaning of Error Data
62H, 62H, 62H	Baud rate modification error.
63H, 63H, 63H	Operation command error.
A1H, A1H, A1H	Framing error in the received data.
A3H, A3H, A3H	Overrun error in the received data.

Note: If a password error occurs, TMP88F846UG does not transmit an error code.

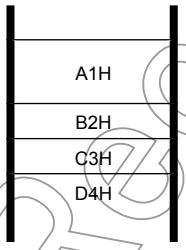
19.8 Checksum (SUM)

19.8.1 Calculation Method

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

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

Example:



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

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

The checksum which is transmitted by executing the flash memory write command, RAM loader command, or flash memory SUM output command is calculated in the manner, as shown above.

19.8.2 Calculation data

The data used to calculate the checksum is listed in Table 19-15.

Table 19-15 Checksum Calculation Data

Operating Mode	Calculation Data	Description
Flash memory writing mode	Data in the entire area of the flash memory	Even when a part of the flash memory is written, the checksum of the entire flash memory area (04000H to 05EFFH, and FFF00H to FFFFFH) is calculated. The data length, address, record type and checksum in Intel Hex format are not included in the checksum.
Flash memory SUM output mode		
RAM loader mode	RAM data written in the first received RAM address through the last received RAM address	The length of data, address, record type and checksum in Intel Hex format are not included in the checksum.
Product ID Code Output mode	9th through 18th bytes of the transferred data	For details, refer to "19.11 Product ID Code".
Flash Memory Status Output mode	9th through 12th bytes of the transferred data	For details, refer to "19.12 Flash Memory Status Code"
Flash Memory Erasing mode	All data in the erased area of the flash memory (the whole or part of the flash memory)	When the sector erase is executed, only the erased area is used to calculate the checksum. In the case of the chip erase, an entire area of the flash memory is used.

19.9 Intel Hex Format (Binary)

1. After receiving the checksum of a data record, the device waits for the start mark (3AH ":") of the next data record. After receiving the checksum of a data record, the device ignores the data except 3AH transmitted by the external controller.
2. After transmitting the checksum of end record, the external controller must transmit nothing, and wait for the 2-byte receive data (upper and lower bytes of the checksum).
3. If a receiving error or Intel Hex format error occurs, the device enters the halt condition without returning an error code to the external controller. The Intel Hex format error occurs in the following case:
   - When the record type is not 00H, 01H, or 02H
   - When a checksum error occurs
   - When the data length of an extended record (record type = 02H) is not 02H
   - When the data length of the end record (record type = 01H) is not 00H

19.10 Passwords

The consecutive eight or more-byte data in the flash memory area can be specified to the password. TMP88F846UG compares the data string specified to the password with the password string transmitted from the external controller. The area in which passwords can be specified is located at addresses 04000H to 05EFFH. The area from FFF00H to FFFFFH can not be specified as the passwords area.

If addresses from FFFE0H through FFFFFH are filled with "FFH", the passwords are not compared because the product is considered as a blank product. Even in this case, the password count storage addresses and password comparison start address must be specified. Table 19-16 shows the password setting in the blank product and non-blank product.

Table 19-16 Password Setting in the Blank Product and Non-Blank Product

Password	Blank Product (Note 1)	Non-Blank Product
PNSA (Password count storage address)	04000H ≤ PNSA ≤ 05EFFH	04000H ≤ PNSA ≤ 05EFFH
PCSA (Password comparison start address)	04000H ≤ PCSA ≤ 05EFFH	04000H ≤ PCSA ≤ 05F00H - N
N (Password count)	*	8 ≤ N
Password string setting	Not required (Note 5)	Required (Note 2)

Note 1: When addresses from FFFE0H through FFFFFH are filled with "FFH", the product is recognized as a blank product.

Note 2: The data including the same consecutive data (three or more bytes) can not be used as a password. (This causes a password error data. TMP88F846UG transmits no data and enters the halt condition.)

Note 3: *: Don't care.

Note 4: When the above condition is not met, a password error occurs. If a password error occurs, the device enters the halt condition without returning the error code.

Note 5: In the flash memory writing mode or RAM loader mode, the blank product receives the Intel Hex format data immediately after receiving PCSA without receiving password strings. In this case, the subsequent processing is performed correctly because the blank product ignores the data except the start mark (3AH ":") as the Intel Hex format data, even if the external controller transmits the dummy password string. However, if the dummy password string contains "3AH", it is detected as the start mark erroneously. The microcontroller enters the halt mode. If this causes the problem, do not transmit the dummy password strings.

Note 6: In the flash memory erasing mode, the external controller must not transmit the password string for the blank product.

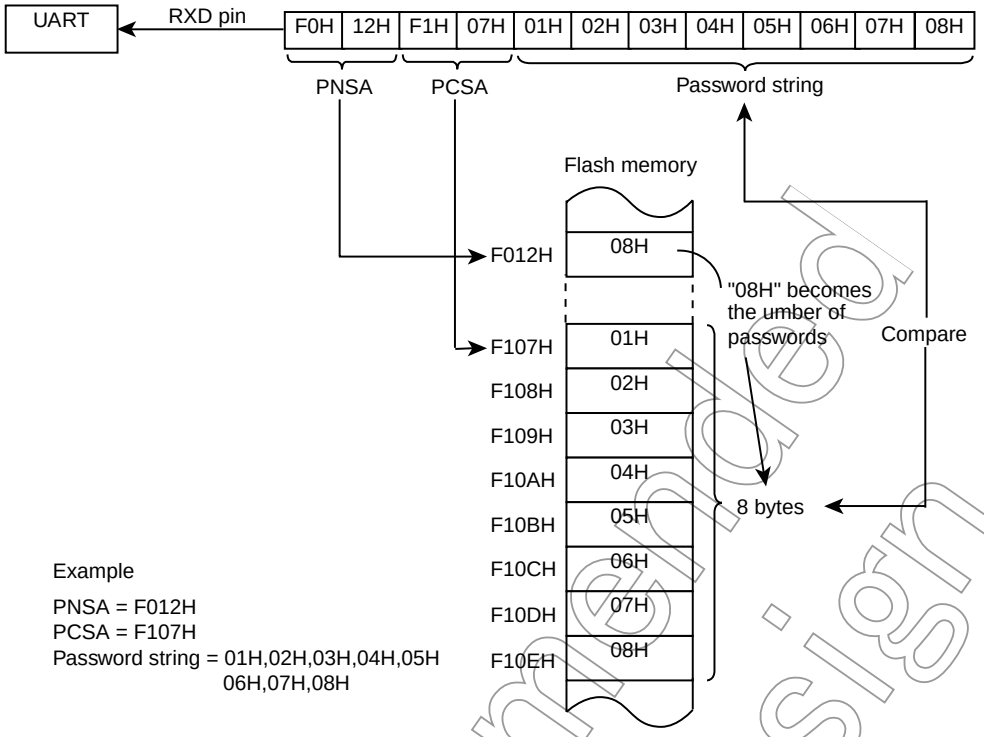


Figure 19-5 Password Comparison

19.10.1 Password String

The password string transmitted from the external controller is compared with the specified data in the flash memory. When the password string is not matched to the data in the flash memory, the device enters the halt condition due to the password error.

19.10.2 Handling of Password Error

If a password error occurs, the device enters the halt condition. In this case, reset the device to reactivate the serial PROM mode.

19.10.3 Password Management during Program Development

If a program is modified many times in the development stage, confusion may arise as to the password. Therefore, it is recommended to use a fixed password in the program development stage.

Example :Specify PNSA to F000H, and the password string to 8 bytes from address F001H
(PCSA becomes F001H.)

```
Password Section code abs = 0F000H
        DB    08H             : PNSA definition
        DB    “CODE1234”      : Password string definition
```

19.11 Product ID Code

The product ID code is the 21-byte data containing the start address and the end address of ROM. Table 19-17 shows the product ID code format.

Table 19-17 Product ID Code Format

Data	Description	In the Case of TMP88F846UG
1st	Start Mark (3AH)	3AH
2nd	The number of transfer data (18 bytes from 3rd to 20th byte)	12H
3rd	Address length (3 bytes)	03H
4th	Reserved data	3DH
5th	Reserved data	00H
6th	Reserved data	00H
7th	Reserved data	00H
8th	ROM block count	02H
9th	The first address of ROM block 1 (Upper byte)	00H
10th	The first address of ROM block 1 (Middle byte)	40H
11th	The first address of ROM block 1 (Lower byte)	00H
12th	The end address of ROM block 1 (Upper byte)	00H
13th	The end address of ROM block 1 (Middle byte)	5EH
14th	The end address of ROM block 1 (Lower byte)	FFH
15th	The first address of ROM block 2 (Upper byte)	0FH
16th	The first address of ROM block 2 (Middle byte)	FFH
17th	The first address of ROM block 2 (Lower byte)	00H
18th	The end address of ROM block 2 (Upper byte)	0FH
19th	The end address of ROM block 2 (Middle byte)	FFH
20th	The end address of ROM block 2 (Lower byte)	FFH
21th	Checksum of the transferred data (2's compliment for the sum of 3rd through 20th bytes)	06H

19.12 Flash Memory Status Code

The flash memory status code is the 7-byte data including the security program status and the status of the data from FFFE0H to FFFFFH. Table 19-18 shows the flash memory status code.

Table 19-18 Flash Memory Status Code

Data	Description	In the Case of TMP88F846UG
1st	Start mark	3AH
2nd	Transferred data count (3rd through 6th byte)	04H
3rd	Status code	00H to 03H (See figure below)
4th	Reserved data	00H
5th	Reserved data	00H
6th	Reserved data	00H
7th	Checksum of the transferred data (2's compliment for the sum of 3rd through 6th data)	3rd byte checksum 00H 00H 01H FFH 02H FEH 03H FDH

Status Code 1

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

RPENA	Flash memory read security program status	0: 1:	Security program is disabled. Security program is enabled.
BLANK	The status from FFFE0H to FFFFFH.	0: 1:	All data is FFH in the area from FFFE0H to FFFFFH. The value except FFH is included in the area from FFFE0H to FFFFFH.

Some operation commands are limited by the flash memory status code 1. If the security program is enabled, flash memory writing mode command and RAM loader mode command can not be executed. Erase all flash memory before executing these command.

RPENA	BLANK	Flash Memory Writing Mode	RAM Loader Mode	Flash memory SUM Output Mode	Product ID Code Output Mode	Flash Memory Status Output Mode	Flash Memory Erasing Mode	Security program Setting Mode
0	0	O	O	O	O	O	O	×
0	1	Pass	Pass	O	O	O	Pass	Pass
1	0	×	×	O	O	O	O	×
1	1	×	×	O	O	O	Pass	Pass

Note: O : The command can be executed.
Pass : The command can be executed with a password.
× : The command can not be executed.
(After echoing the command back to the external controller, TMP88F846UG stops UART communication and enters the halt condition.)

19.13 Specifying the Erasure Area

In the flash memory erasing mode, the erasure area of the flash memory is specified by n−2 byte data.

The address of an erasure area is specified by ERAREA.

The sector erase (flash memory erasing every 4K bytes) is performed if the address data from "00H" to "1DH" is specified by ERAREA.

The chip erase (all flash memory erasing) is performed if the address data from "C0H" to "FFH" is specified by ERAREA. At the same time, the security program for flash memory is also disabled.

Therefore, make sure to execute the chip erase (not sector erase) to disable the security program for flash memory. Executing the sector erase while the security program is enabled results in an infinite loop.

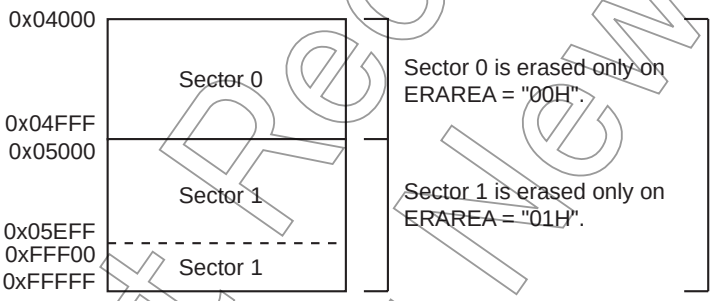
Erasure Area Specification Data (n−2 byte data)

7	6	5	4	3	2	1	0
ERASTA							

ERAREA	Select the erase area	00H: Sector 0 erase 01H: Sector 1 erase C0H: Chip erase : : FFH: Chip erase

Note 1: When the sector erase is executed for the area containing no flash cell, TMP88F846UG stops the UART communi cation and enters the halt condition.

Note 2: Executing the sector erase while the security program is enabled results in an infinite loop.



19.14 Port Input Control Register

In the serial PROM mode, the input level is fixed to the all ports except P45 and P44 ports with a hardware feature to prevent overlap current to unused ports. (All port inputs and peripheral function inputs shared with the ports become invalid.) Therefore, to access to the flash memory in the RAM loader mode without UART communication, port inputs must be valid. To make port inputs valid, set the pin of the port input control register (SPCR) to "1".

The SPCR register is not operated in the MCU mode.

Port Input Control Register

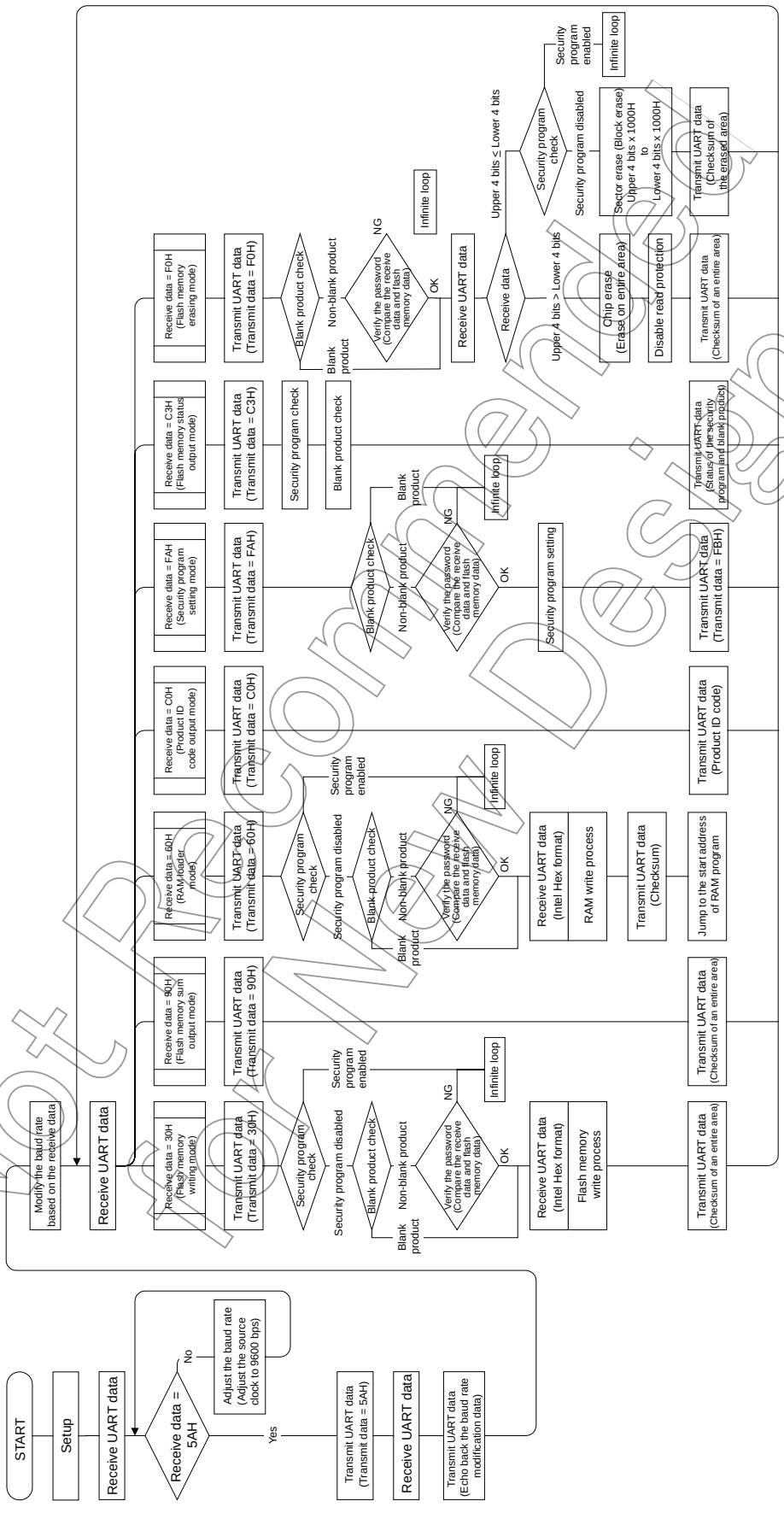
SPCR (01FFEH)	7	6	5	4	3	2	1	0	
								PIN	(Initial value: **** ***0)

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

Note 1: The SPCR register can be read or written only in the serial PROM mode. When the write instruction is executed to the SPCR register in the MCU mode, the port input control can not be performed. When the read instruction is executed for the SPCR register in the MCU mode, read data of bit7 to 1 are unstable.

Note 2: All I/O ports except P45 and P44 ports are controlled by the SPCR register.

19.15 Flowchart



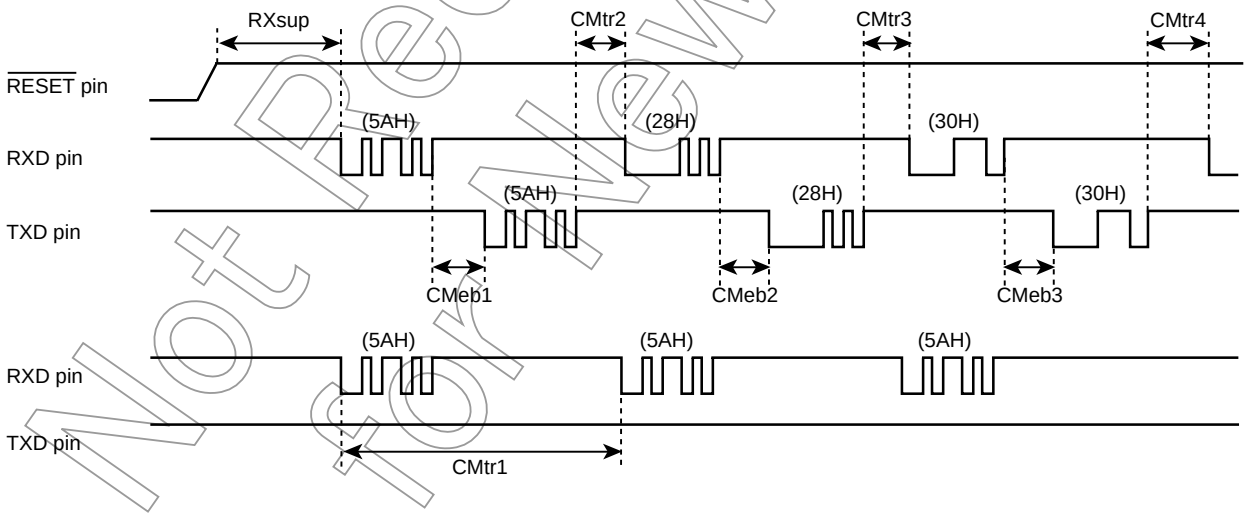
19.16 UART Timing

Table 19-19 UART Timing-1 (VDD = 4.5 to 5.5 V, fc = 20 MHz, Topr = -10 to 40˚C)

Parameter	Symbol	Clock Frequency (fc)	Minimum Required Time
			At fc = 20 MHz
Time from matching data reception to the echo back	CMeb1	Approx. 930	46.5 µs
Time from baud rate modification data reception to the echo back	CMeb2	Approx. 980	49.0 µs
Time from operation command reception to the echo back	CMeb3	Approx. 800	40 µs
Checksum calculation time	CKsm	Approx. 7864500	0.39 s
Erasure time of an entire flash memory	CEall	-	30 ms
Erasure time for a sector of a flash memory (in 4-kbyte units)	CEsec	-	15 ms

Table 19-20 UART Timing-2 (VDD = 4.5 to 5.5 V, fc = 20 MHz, Topr = -10 to 40˚C)

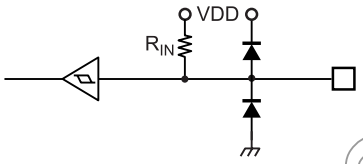
Parameter	Symbol	Clock Frequency (fc)	Minimum Required Time
			At fc = 20 MHz
Time from the reset release to the acceptance of start bit of RXD pin	RXsup	2100	105.0 ms
Matching data transmission interval	CMtr1	28500	1.43 ms
Time from the echo back of matching data to the acceptance of baud rate modification data	CMtr2	380	19.0 µs
Time from the echo back of baud rate modification data to the acceptance of an operation command	CMtr3	650	32.5 µs
Time from the echo back of operation command to the acceptance of password count storage addresses (Upper byte)	CMtr4	800	40 µs



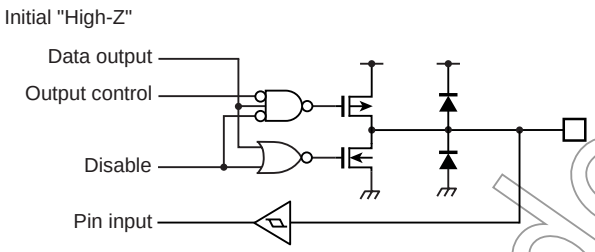
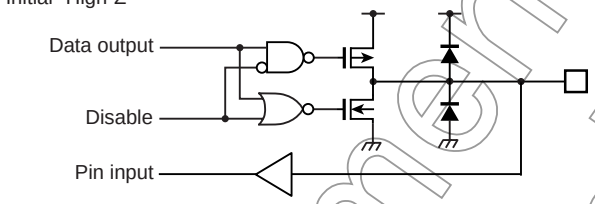
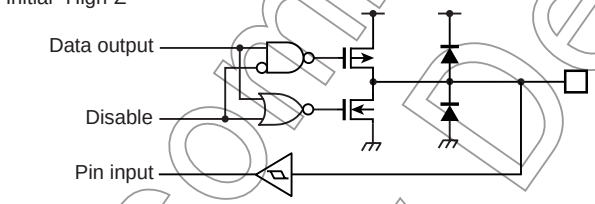
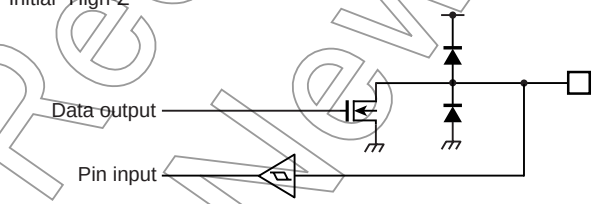
20. Input/Output Circuitry

20.1 Control pins

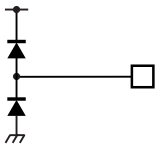
The input/output circuitries of the TMP88F846UG control pins are shown below.

Control Pin	I/O	Input/Output Circuitry	Remark
$\overline{\text{RESET}}$	Input		Hysteresis input Pull-up resistor included R_{IN} = 220 kΩ (typ.)
TEST	Input		Without pull-down resistor Fix the TEST pin at “L” level in MCU mode.

20.2 Input/output ports

Port	I/O	Input/Output Circuit	Remark
P3 P4	I/O		Three-state output Programmable open-drain P3, P4: Large-current port Hysteresis input
P6	I/O		Three-state output
P1	I/O		Three-state output Hysteresis input
P2	I/O		Open-drain output Hysteresis input

20.3 NC pins

Port	I/O	Input/Output Circuit	Remark
LQFP44 NC pins 2 pin, 3 pin 43 pin 44 pin	NC		NC pins

Note: The NC pins of TMP88PH41UG does not have a protect diode.

21. Electrical Characteristics

21.1 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 6.0	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, P2, P6	3.2		
	I_{OL2}	P3, P4	30		
Mean output current	Σ I_{OUT1}	P1, P2, P6	60		Total of all ports except large-current ports
	Σ I_{OUT2}	P3	60		Total of 8 pins of large-current ports P30 to 7
	Σ I_{OUT3}	P4	60		Total of 8 pins of large-current ports P40 to 7
Power dissipation	P_D	TMP88F846UG	350	mW	LQFP
Soldering temperature (time)	Tsld		260 (10 s)	°C	
Operating Temperature	Topr		−40 to 85		
Storage temperature	Tstg		−55 to 125		

21.2 Operating Conditions

The Operating Conditions show the conditions under which the device be used in order for it to operate normally while maintaining its quality. If the device is used outside the range of Operating Conditions (power supply voltage, operating temperature range, or AC/DC rated values), it may operate erratically. Therefore, when designing your application equipment, always make sure its intended working conditions will not exceed the range of Operating Conditions.

21.2.1 MCU mode (Flash Programming or erasing)

(V_{SS} = 0 V, Topr = −10 to 40°C)

Parameter	Symbol	Pins	Ratings	Min	Typ	Max	Unit
Supply voltage	V_{DD}		NORMAL mode	4.5	5.0	5.5	V
Input High level	V_{IH1}	Normal input (P6)	V_{DD} ≥ 4.5 V	V_{DD} × 0.70	−	V_{DD}	
	V_{IH2}	Hysteresis input (P1, P2, P3, P4, $\overline{RESET}$)		V_{DD} × 0.75	−		
Input Low level	V_{IL1}	Normal input (P6)	V_{DD} ≥ 4.5 V	0	−	V_{DD} × 0.30	
	V_{IL2}	Hysteresis input (P1, P2, P3, P4, $\overline{RESET}$)				V_{DD} × 0.25	
Clock Frequency	fc		V_{DD} = 4.5 V to 5.5 V	−	20.0	−	MHz

21.2.2 MCU mode (Except Flash Programming or erasing)

(V_{SS} = 0 V, Topr = -40 to 85°C)

Parameter	Symbol	Pins	Ratings		Min	Typ	Max	Unit
Supply voltage	V_{DD}		fc=20MHz	NORMAL / IDLE modes	4.5	5.0	5.5	V
			STOP mode					
Input high level	V_{IH1}	Normal input (P6)	V_{DD} ≥ 4.5 V		V_{DD} × 0.70	−	V_{DD}	
	V_{IH2}	Hysteresis input (P1,P2,P3,P4, $\overline{RESET}$)			V_{DD} × 0.75	−		
Input low level	V_{IL1}	Normal input (P6)	V_{DD} ≥ 4.5 V		0	−	V_{DD} × 0.30	
	V_{IL2}	Hysteresis input (P1,P2,P3,P4, $\overline{RESET}$)					V_{DD} × 0.25	
Clock frequency	fc		V_{DD} = 4.5 to 5.5 V		−	20.0	−	MHz

21.2.3 Serial PROM mode

(V_{SS} = 0 V, Topr = -10 to 85°C)

Parameter	Symbol	Pins	Ratings	Min	Typ	Max	Unit
Supply voltage	V_{DD}		NORMAL mode	4.5	5.0	5.5	V
Input high level	V_{IH1}	Normal input (P6)	V_{DD} ≥ 4.5 V	V_{DD} × 0.70	–	V_{DD}	
	V_{IH2}	Hysteresis input (P1,P2,P3,P4, $\overline{RESET}$)		V_{DD} × 0.75	–		
Input low level	V_{IL1}	Normal input (P6)	V_{DD} ≥ 4.5 V	0	–	V_{DD} × 0.30	
	V_{IL2}	Hysteresis input (P1,P2,P3,P4, $\overline{RESET}$)				V_{DD} × 0.25	
Clock frequency	fc		V_{DD} = 4.5 to 5.5 V	–	20.0	–	MHz

21.3 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, Three-state					
	I_{IN3}	$\overline{RESET}$, $\overline{STOP}$					
Input resistance	R_{IN2}	$\overline{RESET}$		90	220	510	kΩ
Output leakage current	I_{LO}	Sink open drain, Three-state	V_{DD} = 5.5 V, V_{IN} = 5.5 V/0 V	-	-	±2	μA
High level output voltage	V_{OH}	Three-state port	V_{DD} = 4.5 V, I_{OH} = −0.7 mA	4.1	-	-	V
Low level output current	I_{OL1}	P0, P1, P2, 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}		When a program operates on flash memory (Note4,5)	-	22.0	38.0	mA
			V_{DD} = 5.5 V V_{IN} = 5.3 V/0.2 V fc = 20 MHz				
IDLE mode power supply current				-	20	35	
STOP mode power supply current				-	2	100	μA
Peak current of intermittent operation (Note4,5)	I_{DDP-P}		V_{DD} = 5.5 V	-	10.0	-	mA

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

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

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

Note 4: When a program is executing in the flash memory or when data is being read from the flash memory, the flash memory operates in an intermittent manner, causing peak currents in the operation current, as shown in Figure 21-1. In this case, the supply current I_{DD} (in NORMAL mode) is defined as the sum of the average peak current and MCU current.

Note 5: When designing the power supply, make sure that peak currents can be supplied.

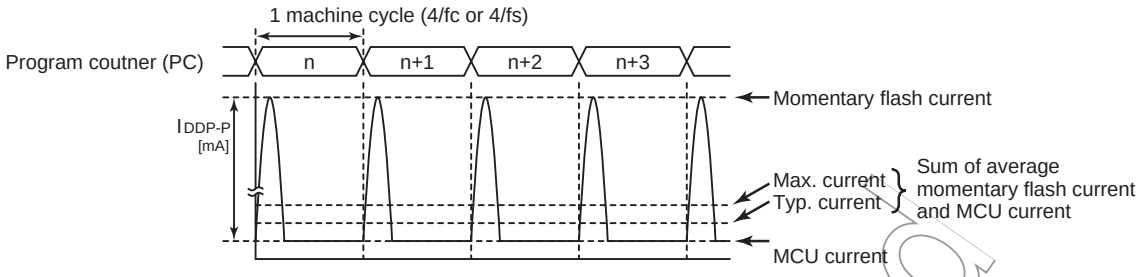


Figure 21-1 Intermittent Operation of Flash Memory

21.4 On-Chip Oscillator Characteristics

(Topr = −40 to 85°C)

Parameter	Symbol	Condition	Min	Typ	Max	Unit
Normal Oscillation frequency	f_{NOM}	V_{DD} = 5.0V, Topr = 25°C	19.8	20	20.2	MHz
Frequency deviation	Δf	with respect to f_{NOM} V_{DD} = 5.0V, Topr = −40 to 10°C	-4	-	+4	%
		with respect to f_{NOM} V_{DD} = 5.0V, Topr = −10 to 85°C	-4	-	+4	%

21.5 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, AV_{DD} = V_{DD}	V_{DD} −1.0	-	V_{DD}		V
Analog Input voltage range	V_{AIN}		AV_{SS}	-	V_{AREF}		V
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 AV_{DD} pin and V_{AREF} pin on the V_{DD} level.

21.6 Power On Reset circuit (POR)

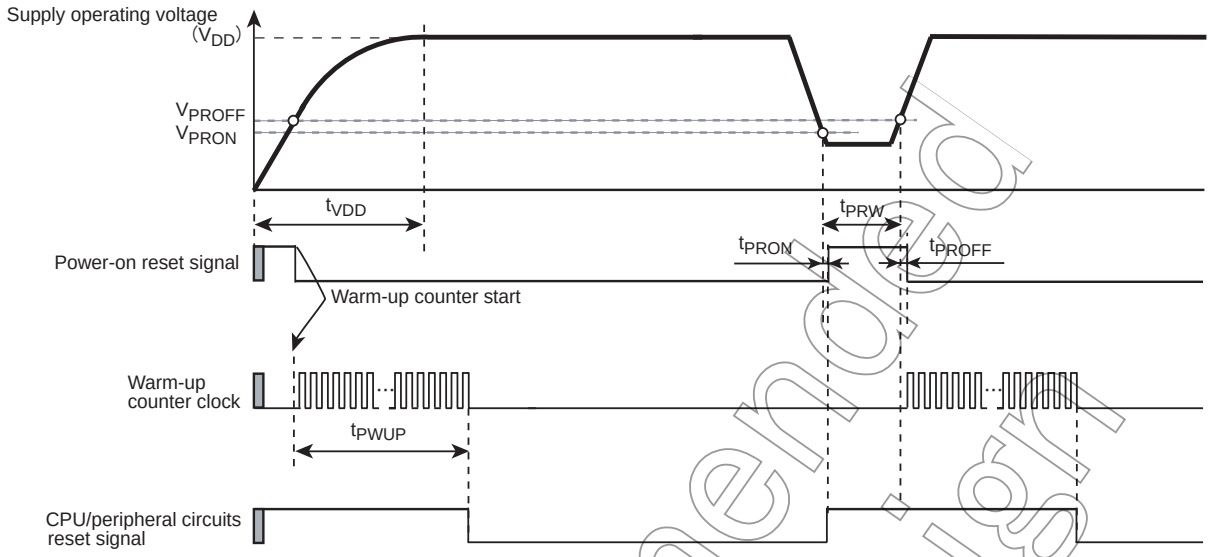


Figure 21-2 Operation of the Power-on reset circuit

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

(V_{SS} = 0 V, Topr = −40 to 85°C)

Symbol	Parameter	Min	Typ	Max	Unit
V_{PROFF}	Power-on reset release voltage	2.2	2.4	2.6	V
V_{PRON}	Power-on reset detection voltage	2.0	2.2	2.3	
t_{PROFF}	Power-on reset release response time	-	0.01	0.1	ms
t_{PRON}	Power-on reset generation response time	-	0.01	0.1	
t_{PRW}	Power-on reset minimum pulse width	1.0	-	-	
t_{PWUP}	Warming-up time after releasing RESET	14.0	-	30.0	
t_{VDD}	Power supply raising time	-	-	7	

Note 1: Because the power-on reset releasing voltage and the power-on reset detecting voltage change relative to one another, the detected voltage will never become inverted.

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

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

21.7 Voltage Detection circuit (VLTD)

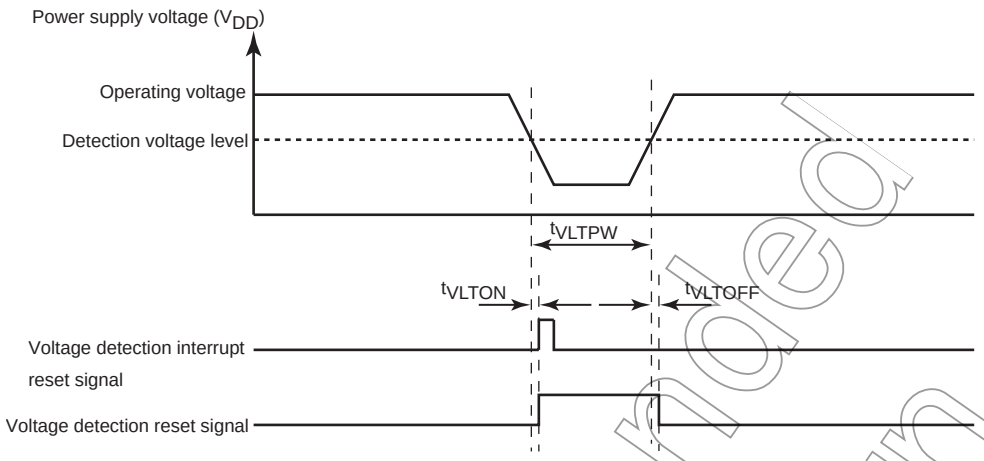


Figure 21-3 Operation timing of Voltage detection circuit

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

(V_{SS} = 0 V, Topr = −40 to 85°C)

Symbol	Parameter	Min	Typ	Max	Unit
t_{VLTOFF}	Voltage detection release response time	-	0.01	0.1	ms
t_{VLTON}	Voltage detection response time	-	0.01	0.1	ms
t_{VLTPW}	Voltage detection minimum pulse width	1.0	-	-	ms

21.8 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	0.2	-	0.5	μs
		During IDLE mode				

21.9 Flash Characteristics

Write Characteristics

(V_{SS} = 0 V, 4.5 V ≤ V_{DD} ≤ 5.5 V)

Parameter	Condition	Min	Typ	Max	Unit
Number of guaranteed writes to flash memory	Topr = -10 to 40°C	-	-	100	Times

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

21.10 Handling Precaution

- The solderability test conditions are shown below.

  1. When using the Sn-37Pb solder bath

  Solder bath temperature = 230 °C

  Dipping time = 5 seconds

  Number of times = once

  R-type flux used

  2. When using the Sn-3.0Ag-0.5Cu solder bath

  Solder bath temperature = 245 °C

  Dipping time = 5 seconds

  Number of times = once

  R-type flux used

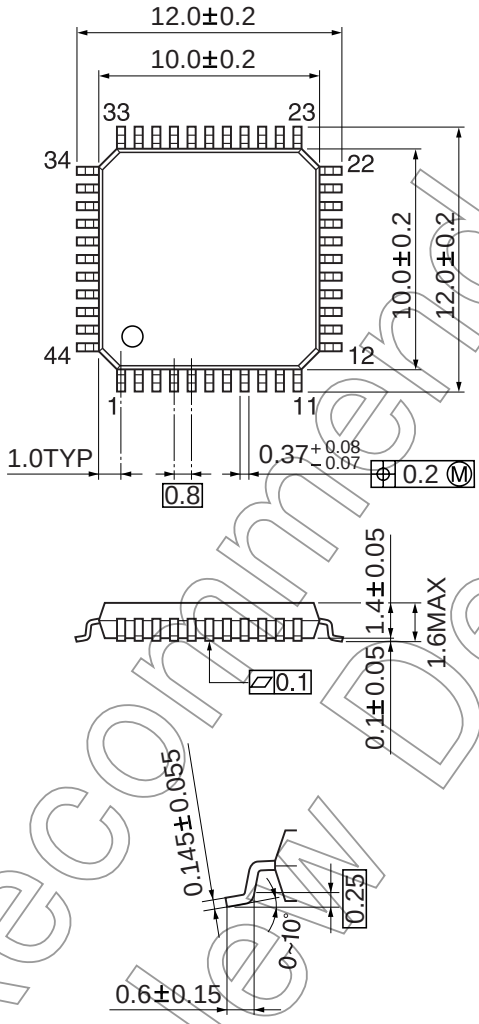
  Note: The pass criterion of the above test is as follows: Solderability rate until forming ≥95%

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

22. Package Dimensions

LQFP44-P-1010-0.80B Rev 01

Unit: mm



RESTRICTIONS ON PRODUCT USE

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