NEC

16-Bit Timer/Event Counter 0 (TM0) in Interval Timer Mode

On-Chip Peripheral Program Example August 1999

Description

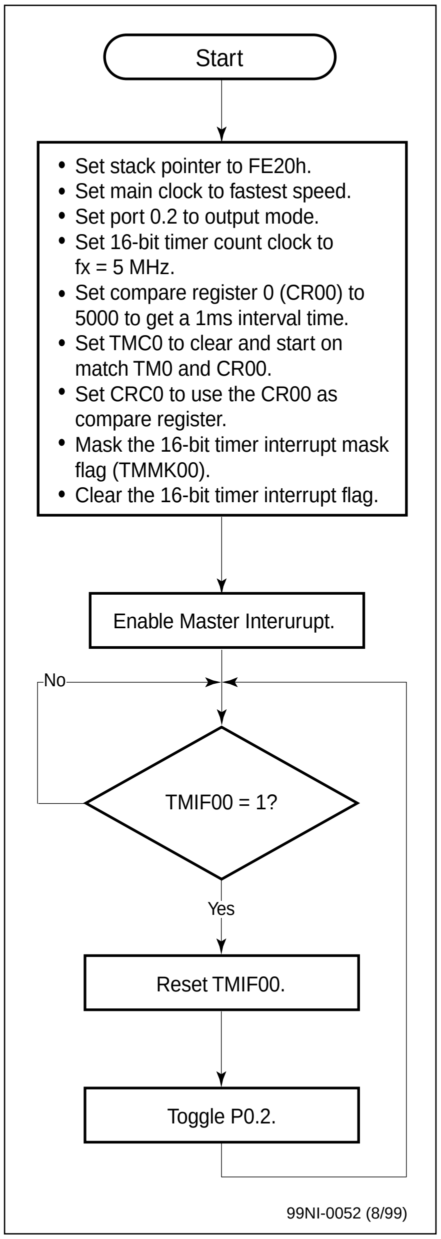
The 16-bit timer/event counter (TM0) in the μPD7805x/78005x subseries can be used as an interval timer, external event counter, pulse-width modulator output, square-wave output, one-shot pulse output or for pulse-width measurement.

This program demonstrates the 16-bit timer/event counter in interval timer mode. When the count value of the 16-bit timer register (TM0) matches the value set to capture/compare register 0 (CR00), counting continues with the TM0 value cleared to 0 and the interrupt request signal (INTTM00) generated. This sets the interrupt flag TMIF00 in the interrupt request flag register (IF0H). Each time the program detects the flag set to one, it toggles port 0.2 and clears the flag.

Program Specifications

- Timer count clock: 5 MHz
- Interval time: 1 ms (2 ms period)
- Square wave frequency: 500 Hz
- Interrupt handling: polling the interrupt request flag TMIF00
- Pins used in program: P02/INTP2 (toggles every 1 ms)

Flowchart


Start
• Set stack pointer to FE20h.
• Set main clock to fastest speed.
• Set port 0.2 to output mode.
• Set 16-bit timer count clock to fx = 5 MHz.
• Set compare register 0 (CR00) to 5000 to get a 1ms interval time.
• Set TMC0 to clear and start on match TM0 and CR00.
• Set CRC0 to use the CR00 as compare register.
• Mask the 16-bit timer interrupt mask flag (TMMK00).
• Clear the 16-bit timer interrupt flag.
Enable Master Interurupt.
No
TMIF00 = 1?
Yes
Reset TMIF00.
Toggle P0.2.
99NI-0052 (8/99)

Assembly Language Program

```
;****************************************************************************
; Date:       07/19/1999
;
; Parameters: - fastest CPU clock
;               (fx=5.00 MHz, 1 CPU clock cycle = 200 ns)
;             - interval time: 1 ms
;             - Count clock:   fx = 5 MHz
;             - Unmask the 16-bit timer interrupt mask bit for
                   polling the timer interrupt flag
;             - Port 0.2 toggles every 1 ms
;
;*****************************************************************************

;=========================================
;=     Specify Interrupt Vectors         =
;=========================================

Res_Vec CSEG AT 0000h            ; Set main program start vector
        DW   Start

;=========================================
;      Main Program                      =
;=========================================

MAIN     CSEG
Start:   DI                      ; Disable interrupts
         MOVW       AX, #0FE20h  ; Load SP address
         MOVW       SP, AX       ; Set Stack Pointer
         MOV        OSMS,#01h    ; Don't use scaler
         MOV        PCC, #00h    ; Main system clock at fastest setting
         CLR1       P0.2         ; Latch port 0.2 low
         CLR1       PM0.2        ; Set port 0.2 as output
         MOV        TCL0,#020h   ; Select counter clock to fx = 5 MHz
         MOVW       CR00,#5000   ; Set compare register to 5000 for 1 ms interval
         MOV        TMC0,#0Ch    ; Set to clear and start of match TM0 and CR00
         CLR1       CRC0.0       ; Set CR00 operation to compare register
         SET1       TMMK00       ; Mask the 16-bit timer interrupt mask bit
         CLR1       TMIF00       ; Clear 16-bit timer interrupt flag
         EI                      ; Enable interrupts
Loop:    BF         TMIF00,$$    ; Branch if timer interrupt flag is high
         CLR1       TMIF00       ; Clear timer interrupt flag
         XOR        P0,#04h      ; Toggle port 0.2
         BR         Loop         ; Loop back
       END
```

C Language Program

```
/**************************************************************************
; Date:       07/19/1999
;
; Parameters: - fastest CPU clock
;              (fx = 5.00 MHz, 1 CPU clock cycle = 200 ns)
;             - interval time: 1 ms
;             - Count clock:   fx = 5  MHz
;             - Unmask the 16-bit timer interrupt mask bit for
                   polling the timer interrupt flag
;             - Port 0.2 toggles every 1 ms
;
;**************************************************************************/
/* extension functions in K0/K0S compiler */
#pragma sfr  /* key word to allow SFR names in C code */
#pragma asm  /* key word to allow ASM statements in C code */
#pragma DI   /* key word for DI instruction in C code */
#pragma EI   /* key word for EI instruction in C code */
/*==========================================
;      Constants/Variables                =
;========================================*/

#define TRUE        1
#define FALSE       0

/*==========================================
;      Main Program                       =
;========================================*/
void main(void)
{
      OSMS = 0x01;               /* Don't use scaler */
      PCC = 0x00;                /* Main system clock at fastest setting */
      P0.2 = 0;                  /* Latch port 0.2 low */
      PM0.2 = 0;                 /* Set port 0.2 as output */
      TCL0 = 0x20;               /* Select counter clock to fx = 5 MHz */
      CR00 = 5000;               /* Set compare register to 5000 for a 1 ms interval */
      TMC0 = 0x0c;               /* Set to clear and start on match of TM0 and CR00 */
      CRC0.0 = 0;                /* Set CR00 operation to compare register */
      TMMK00= 1;                 /* Mask the 16-bit timer interrupt mask bit */
      TMIF00 = 0;                /* Clear timer interrupt flag */
      EI();                      /* Enable interrupts */
      while(TRUE)                /* while loop*/
      {
            while( !TMIF00 );    /* Wait until timer interrupt flag is high */
            TMIF00 = 0;          /* Clear timer interrupt flag */
            P0 ^= 0x04;          /* toggle port 0.2 */
      }                          /* end of while loop */
}                                /* end of function main() */
```

NEC

For literature, call **1-800-366-9782** 7 a.m. to 6 p.m. Pacific time
or FAX your request to **1-800-729-9288**
or visit our web site at **www.necel.com**

50871