NEC

8-Bit Timer/Event Counters (TM1 and TM2) in 16-Bit Timer Counter Mode

On-Chip Peripheral Program Example August 1999

Description

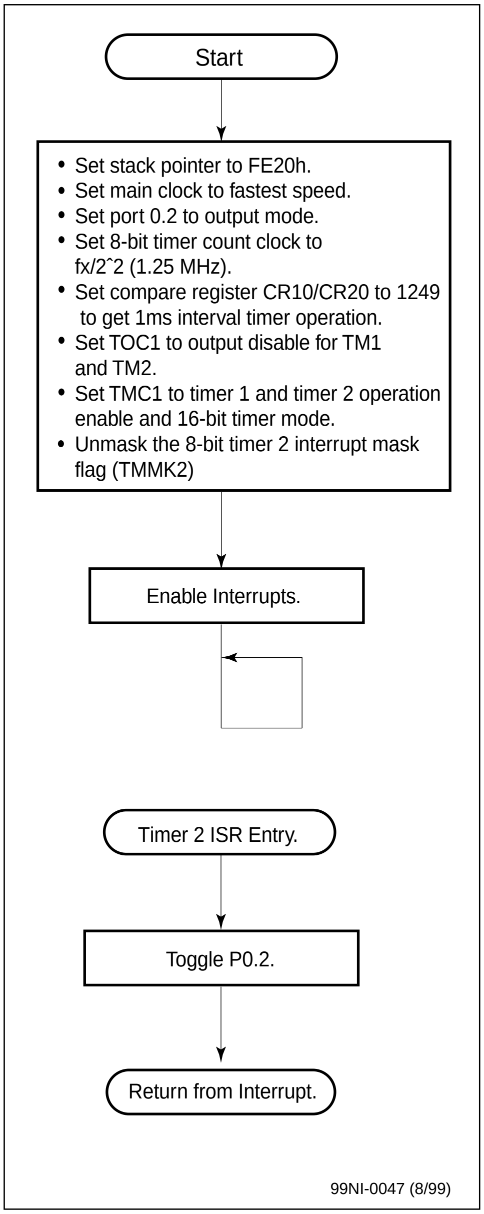
The 8-bit timer/event counters (TM1 and TM2) in the µPD7805x/78005x subseries can be cascaded to become one 16-bit timer/counter, which can be used as a 16-bit interval timer, 16-bit external event counter, or 16-bit square-wave output.

This program demonstrates how the TM1 and TM2 operate in 16-bit interval timer mode. The count clock clocks TM1 and the overflow signal from TM1 clocks TM2. When the count value of TM1 matches the value set to compare register CR10 and the count value of TM2 matches the value set to compare register CR20, the interrupt request flag (TMIF2) is set to 1, causing a vectored interrupt that toggles port pin 0.2 in the interrupt service routine. At the same time, TM1 and TM2 are cleared to 0.

Program Specifications

- ❑ Count clock frequency: 1.25 MHz at 5 MHz main system clock
- ❑ Square wave frequency: 500 Hz (2 ms period)
- ❑ Pins used in program: P02/INTP2 (port pin 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 8-bit timer count clock to fx/2ˆ2 (1.25 MHz).
• Set compare register CR10/CR20 to 1249 to get 1ms interval timer operation.
• Set TOC1 to output disable for TM1 and TM2.
• Set TMC1 to timer 1 and timer 2 operation enable and 16-bit timer mode.
• Unmask the 8-bit timer 2 interrupt mask flag (TMMK2)
Enable Interrupts.
Timer 2 ISR Entry.
Toggle P0.2.
Return from Interrupt.
99NI-0047 (8/99)

Assembly Language Program

```
;**********************************************************************
; Date:        06/16/1999
; Parameters: - fastest CPU clock
;               (fx = 5.00 MHz, 1 CPU clock cycle = 200 ns)
;             - Count clock:    fx (1.25 MHz)
;             - Interval time:  1 ms (2 ms period)
;             - enable interrupt handling for timer 2
;             - port 0.2 toggles every 1 ms
;**********************************************************************

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

Res_Vec      CSEG   AT 0000h      ; Set main program start vector.
             DW     Start
             ORG    0026h         ; Set interrupt vector
             DW     TM2_ISR       ; for 8-bit timer 2

;========================================
;=           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 to output mode
             MOV    TCL1,#07h     ; Select counter clock to fx(1.25 MHz)
             MOV    CR10,#0E1h    ; Set compare register 1 (low byte) to E1h for 1 ms interval
             MOV    CR20,#04h     ; Set compare register 2 (high byte) to 04h for 1 ms interval
             MOV    TOC1,#00h     ; Disable output function
             MOV    TMC1,#07h     ; Set TM1 and TM2 operational enable and 16-bit timer mode
             CLR1   TMMK2         ; Unmask the 8-bit timer 2 interrupt bit
             EI                   ; Enable interrupts
Loop1:       BR     $Loop1        ; Endless loop

;========================================
;=      8-bit timer 2 ISR               =
;========================================

ISR          CSEG
TM2_ISR:     XOR    P0,#04h       ; Toggle port 0.2
             RETI                 ; Return from interrupt
       END
```

C Language Program

```
/**************************************************************************
; Date:       06/16/1999
; Parameters: - fastest CPU clock
;               (fx=5.00 MHz, 1 CPU clock cycle = 200 ns)
;             - Count clock: fx (1.25 MHz)
;             - Interval time: 1 ms (2 ms period)
;             - Enable interrupt handling for timer 2
;             - 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 EI   /* key word for EI instruction in C code */

/*;======================================
;=     Specify Interrupt vectors        =
;======================================*/
/* Set interrupt vector for the 8-bit Timer 0 */
#pragma interrupt INTTM2 TM2_ISR

/*;======================================
;=     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 output low */
       PM0.2 = 0;             /* Set port 0.2 output mode*/
       TCL1 = 0x07;           /* Select counter clock to fx(1.25 MHz) */
       CR10 = 0xE1;           /* Set compare register 1(low byte) for 1 ms interval */
       CR20 = 0x04;           /* Set compare register 2(high byte) for 1 ms interval */
       TOC1 = 0x00;           /* Disable output function */
       TMC1 = 0x07;           /* Set TMC1 to TM1 and TM2 operational enable and
                                  16-bit timer mode */
       TMMK2= 0;              /* Unmask the 8-bit timer 2 interrupt bit */
       EI();                  /* Enable interrupts */
       while(TRUE);           /* loop here */
 }                            /* end of function main() */
/*=======================================
;=     8-bit TIMER 2 ISR                =
;======================================*/
void TM2_ISR(void)
{
       P0 ^= 0x04;             /* toggle port 0.2 */
}
```

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**

In North America: No part of this document may be copied or reproduced in any form or by any means without the prior written consent of NEC Electronics Inc. (NECEL). The information in this document is subject to change without notice. All devices sold by NECEL are covered by the provisions appearing in NECEL Terms and Conditions of Sales only. Including the limitation of liability, warranty, and patent provisions. NECEL makes no warranty, express, statutory, implied or by description, regarding information set forth herein or regarding the freedom of the described devices from patent infringement. NECEL assumes no responsibility for any errors that may appear in this document. NECEL makes no commitments to update or to keep current information contained in this document. The devices listed in this document are not suitable for use in applications such as, but not limited to, aircraft control systems, aerospace equipment, submarine cables, nuclear reactor control systems, and life support systems. “Standard” quality grade devices are recommended for computers, office equipment, communication equipment, test and measurement equipment, machine tools, industrial robots, audio and visual equipment, and other consumer products. For automotive and transportation equipment, traffic control systems, anti-disaster and anti-crime systems, it is recommended that the customer contact the responsible NECEL salesperson to determine the reliability requirements for any such application and any cost adder. NECEL does not recommend or approve use of any of its products in life support devices or systems or in any application where failure could result in injury or death. If customers wish to use NECEL devices in applications not intended by NECEL, customer must contact the responsible NECEL salespeople to determine NECEL's willingness to support a given application.