Timer I for the 83/87C748/749 and the 83/87C751/752 (non-I²C applications) microcontrollers

AN427

The small package 83/87C748, 83/87C749, 83/87C751 and 83/87C752 microcontrollers include two hardware-implemented timers: a 16-bit programmable timer, and a 10-bit fixed-rate timer. The programmable timer is available for the application program, and its operation is similar to the timer/counter of the 80C51 timer in mode 2. The fixed-rate timer, Timer I, is typically employed as a watchdog timer for the I²C port communications and is not available for other uses.

In applications which do not take advantage of the I²C communications capability, the "silicon real estate" taken by Timer I is not necessarily lost—it can be used as a fixed-rate timer by the application. This timer can become useful in various cases, such as simple control applications that need a delay while doing some software activities in parallel, or generating a free-running repetitive waveform where the exact timing is not important. Another type of application is a watchdog timer prompting the user about unexpected operation of a system or its hardware, or resetting a program that "lost track."

TIMER I IMPLEMENTATION

Timer I is clocked once per machine cycle, which is the oscillator frequency divided by 12. The timer operation is enabled by setting the TIRUN bit (bit 4) in the I2CFG register. Writing a 0 into the TIRUN bit will stop and clear the timer. The timer is 10 bits wide, and when it reaches the terminal count of 1024 it carries out and sets the Timer I interrupt flag. An interrupt will occur if the Timer I interrupt is enabled by bit ETI (bit 4) of the Interrupt Enable (IE) register, and global interrupts are enabled by bit EA (bit 7) of the same IE register.

The vector address for the Timer I interrupt is 1B hex, and the interrupt service routine must start at this address. As with all 8051 family microcontrollers, only the Program Counter is pushed onto the stack upon interrupt (other registers that are used both by the interrupt service routine and elsewhere must be explicitly saved). The Timer I interrupt flag is cleared by setting the CLRTI bit (bit 5) of the I2CFG register.

Note that when the I²C interface is not operating—SLAVEN, MASTRQ, and MASTER bits are all 0—the I²C hardware does not affect Timer I. The SCL and SDA pins can be used as I/O pins, and the activity of these pins will not cause the timer to run, stop, or reset. Upon hardware reset of the microcontroller, the SLAVEN, MASTRQ, and MASTER bits are all reset, so the programmer does not have to worry about interaction between the SDA/SCL pins and the timer.

FIXED-RATE TIMER

The first programming example demonstrates simple fixed-rate operation. Upon reset, interrupts are enabled, and Timer I is started. A wait loop simulates the "application" program. The demonstration service routine simply sets a flag—in real life it could do something more useful, such as toggling an output pin. Note that the interrupt flag is cleared by setting CLRTI prior to returning from the service routine. Upon overflow, the timer will go on running, as the TIRUN bit is still set, so the interrupts will be spaced exactly 1024 clock cycles apart. If the service routine would toggle an output pin instead of setting a flag, its output would be a square wave with a period of 2048 cycles. For an application that demands a "one-shot" delay only, the service routine should clear the TIRUN bit in order to avoid subsequent interrupts.

WATCHDOG TIMER

A watchdog timer mechanism is typically applied in order to detect "abnormal" behavior of hardware. If the microcontroller operates in a very noisy environment, there might be a fear of the program "running wild" as a result of extremely violent EMI interference. In such a case, a watchdog may take care to reset the microcontroller when the Timer I interrupt occurs. This could be applied in application programs with a repetitive nature—the software needs to reset the timer within 1024 machine cycles of the last reset.

In a system where something is supposed to occur regularly—for example, an interrupt for an external event—the watchdog is designed to "bite" when the hardware "sleeps" and the expected "something" does not happen for too long a time. The timer is allowed to run continuously, but when the expected event occurs, it resets the timer back to 0. When the timer is reset within 1024 cycles of the last reset, the application runs normally. If the event does not occur, the Timer I interrupt service routine will be activated to take care of the exception.

The second programming example demonstrates the watchdog. Upon Reset, the TIRUN bit, ETI, and global interrupts are enabled. The watchdog timer is reset and restarted by the small subroutine WdRst. The application is simulated by a loop of delays. Delay 1 is less than 1024 cycles, and when WdRst is called within Delay 1 intervals, no Timer I interrupt occurs. This represents normal operation of a "real life" application. When the delay from last reset is greater than 1024 cycles—representing a hardware exception—the interrupt will occur. The service routine for the watchdog is somewhat unusual, as it does not return to the program location where the interrupt occurred. Instead, the operation of the microcontroller is restarted at Reset. Upon entering the service routine, the interrupt is cleared and the timer is reset. Because execution does not return to the interrupted program with a RETI instruction, the interrupt pending flag is cleared by a call to a dummy subroutine XRETI. The program is restarted at Reset with a regular AJMP instruction. The stack pointer is explicitly reinitialized for the warm reset, so there is no danger of stack overflow upon repeated watchdog invocations.

```
 TIINT                          Timer I Fixed Rate Timer                          11/06/90   PAGE 1


                 1 ;
                    ************************************************************************
                 2
                 3 ;                     Timer I Fixed Rate Timer Usage
                 4
                 5 ;This program demonstrates how to activate Timer I on the 8XC748/8XC749/83C751
                 6 ;or 83C752 microcontrollers as a fixed rate timer when the I2C port is not
                 7 ;used. Once activated, Timer I will generate an interrupt every 1024
                 8 ;machine cycles. The I2C bus pins SCL and SDA may be used as open drain
                 9 ;outputs.
                10
                11 ;
                    ************************************************************************
                12
                13  $MOD7 751
                14  $Title(Timer I Fixed Rate Timer)
                15  $Date(11/06/90)
                16  $Debug
                17
                18
0020            19  Flags     DATA  20h             ;Flag byte
0000            20  TstFlag   BIT   Flags.0         ;Timer I flag.
                21
                22
0000            23            ORG   0
0000 0120       24            AJMP  Reset
                25
001B            26            ORG   1Bh             ;Timer I interrupt.
001B D200       27  TimerI:   SETB  TstFlag         ;Set flag to indicate a Timer I interrupt.
001D D2DD       28            SETB  CLRTI           ;Clear Timer I to allow it to restart.
001F 32         29            RETI
                30
                31
0020 D2AB       32  Reset:    SETB  ETI             ;Enable Timer I interrupt.
0022 D2AF       33            SETB  EA              ;Enable global interrupts.
0024 D2DC       34            SETB  TIRUN           ;Start Timer I.
                35
0026 C200       36  Loop:     CLR   TstFlag         ;Initialize interrupt flag.
0028 3000FD     37  Wait:     JNB   TstFlag,Wait    ;Wait for Timer I interrupt.
002B 80F9       38            SJMP  Loop
                39
                40            END

ASSEMBLY COMPLETE, 0 ERRORS FOUND
```

```
TIINT                           Timer I Fixed Rate Timer                        11/06/90   PAGE 2

CLRTI. . . . . . . . . . . . . . .  B ADDR  00DDH  PREDEFINED
EA . . . . . . . . . . . . . . . .  B ADDR  00AFH  PREDEFINED
ETI. . . . . . . . . . . . . . . .  B ADDR  00ABH  PREDEFINED
FLAGS. . . . . . . . . . . . . . .  D ADDR  0020H
LOOP . . . . . . . . . . . . . . .  C ADDR  0026H
RESET. . . . . . . . . . . . . . .  C ADDR  0020H
TIMERI . . . . . . . . . . . . . .  C ADDR  001BH  NOT USED
TIRUN. . . . . . . . . . . . . . .  B ADDR  00DCH  PREDEFINED
TSTFLAG. . . . . . . . . . . . . .  B ADDR  0000H
WAIT . . . . . . . . . . . . . . .  C ADDR  0028H
```

```
TIWD                                Timer I Watchdog                          11-06-90   PAGE 1

                1 ;*******************************************************************************
                2
                3 ;                   Timer I Watchdog Timer Usage
                4
                5 ;This program demonstrates how to use Timer I on the 83C751 or 83C752
                6 ;microcontrollers as a watchdog timer when the I2C port is not used.
                7 ;Once started, Timer I must be cleared more often than once every 1024
                8 ;machine cycles. If Timer I is allowed to overflow, a Timer I
                9 ;interrupt will be generated. Thus, if global interrupts or the Timer
               10 ;I interrupt are inhibited, the watchdog function will be disabled.
               11 ;Also, if the watchdog interrupt occurs during another interrupt
               12 ;service, it will be delayed until an RETI (return from interrupt)
               13 ;instruction is executed. The I2C bus pins SCL and SDA may be used as
               14 ;open drain outputs.
               15
               16;*******************************************************************************
               17
               18    $MOD751
               19    $Title(Timer I Watchdog)
               20    $Date(11-06-90)
               21    $Debug
               22
0000           23              ORG     0
0000 0126      24              AJMP    Reset
               25
001B           26              ORG     1Bh         ;Timer I interrupt.
001B C2AF      27    TimerI:   CLR     EA          ;Get here only if watchdog overflows.
001D C2DC      28              CLR     TIRUN       ;Turn off Timer I.
001F D2DD      29              SETB    CLRTI       ;Clear Timer I interrupt.
0021 1125      30              ACALL   XRETI       ;Force interrupt pending to clear.
0023 0126      31              AJMP    Reset       ;Do a warm start.
0025 32        32    XRETI:    RETI
               33
0026 758107    34    Reset:    MOV     SP,#7h      ;Initialize the stack pointer.
               35
               36 ;Note: it is important to force the stack pointer to a particular
               37 ;starting value in this application because we may be re-starting
               38 ;after a watchdog interrupt, with the stack in an unknown condition.
               39
0029 75D800    40    MOV     I2CFG,#0              ;Initialize I2CFG (set up CT0, CT1).
002C D2DC      41    SETB    TIRUN                 ;Enable Timer I run.
002E D2AB      42    SETB    ETI                   ;Enable Timer I interrupt.
0030 D2AF      43    SETB    EA                    ;Enable interrupt system.
               44
               45
               46 ;The following is a "dummy" main program to test the watchdog timer.
               47
0032 1153      48    Loop:     ACALL   Delay1      ;Wait 901 machine cycles.
0034 114E      49              ACALL   WdRst       ;Reset Watchdog.
0036 1153      50              ACALL   Delay1      ;Wait 901 machine cycles.
0038 114E      51              ACALL   WdRst       ;Reset Watchdog.
003A 1153      52              ACALL   Delay1      ;Wait 901 + 4 for ACALL & prior RET.
003C 1157      53              ACALL   Delay2      ;Wait 108 + 2 for ACALL.
003E 00        54              NOP     ;1016
003F 00        55              NOP     ;1017
0040 00        56              NOP     ;1018
0041 00        57              NOP     ;1019
0042 00        58              NOP     ;1020
```

```
TIWD                            Timer I Watchdog                                11-06-90   PAGE 2

0043 00       59                NOP     ;1021
0044 00       60                NOP     ;1022
0045 00       61                NOP     ;1023
0046 00       62                NOP     ;1024           : Should get 'bitten' here.
0047 00       63                NOP     ;1025
0048 00       64                NOP     ;1026
0049 00       65                NOP     ;1027
004A 00       66                NOP     ;1028
004B 00       67                NOP     ;1029
004C 0132     68                AJMP    Loop            ;Should never get here.
              69
004E C2DC     70    WdRst:      CLR     TIRUN           ;Reset Watchdog timer (Timer I).
0050 D2DC     71                SETB    TIRUN
0052 22       72                RET
              73
0053 7480     74    Delay1:     MOV     A,#128          ;Wait 901 machine cycles (1).
0055 8002     75                SJMP    DLoop           ;(2)
0057 740F     76    Delay2:     MOV     A,#15           ;Wait 108 machine cycles (1).
0059 A3       77    DLoop:      INC     DPTR            ;Delay = (ACC * 7) + 2 mach. cyc (2).
005A A3       78                INC     DPTR            ;(2)
005B 14       79                DEC     A               ;(1)
005C 70FB     80                JNZ     Dloop           ;(2)
005E 22       81                RET                     ;(2)
              82
              83                END

ASSEMBLY COMPLETE, 0 ERRORS FOUND
```

```
TIWD                              Timer I Watchdog                          11-06-90   PAGE 3

CLRTI. . . . . . . . . . . . . .  B ADDR  00DDH  PREDEFINED
DELAY1 . . . . . . . . . . . . .  C ADDR  0053H
DELAY2 . . . . . . . . . . . . .  C ADDR  0057H
DLOOP. . . . . . . . . . . . . .  C ADDR  0059H
EA . . . . . . . . . . . . . . .  B ADDR  00AFH  PREDEFINED
ETI. . . . . . . . . . . . . . .  B ADDR  00ABH  PREDEFINED
I2CFG. . . . . . . . . . . . . .  D ADDR  00D8H  PREDEFINED
LOOP . . . . . . . . . . . . . .  C ADDR  0032H
RESET. . . . . . . . . . . . . .  C ADDR  0026H
SP . . . . . . . . . . . . . . .  D ADDR  0081H  PREDEFINED
TIMERI . . . . . . . . . . . . .  C ADDR  001BH  NOT USED
TIRUN. . . . . . . . . . . . . .  B ADDR  00DCH  PREDEFINED
WDRST. . . . . . . . . . . . . .  C ADDR  004EH
XRETI. . . . . . . . . . . . . .  C ADDR  0025H
```