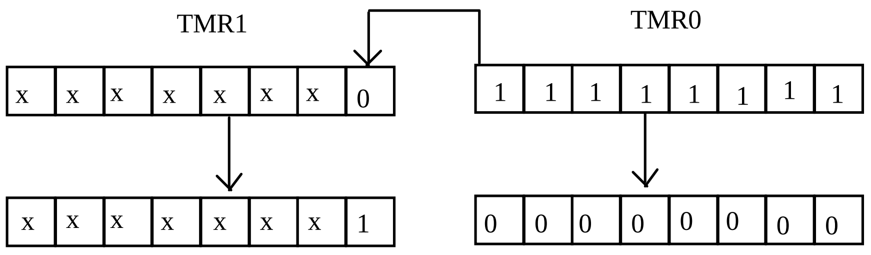
Hitachi America, Ltd. TN-0126
Application Engineering
TechNote June, 1993

Creating a 16-bit Timer from Two 8-bit Timers

The H8/300 series micro controllers include one 16-bit Free Running Timer and two 8-bit Counter\Timers. The two 8-bit timers can be easily combined through software and a single hardware jumper to form a second 16-bit timer. Timer TMR0 forms the lower byte and is set to toggle output TMO0 when an overflow occurs. This output acts as an external clock input to increment timer 1. Thus by cascading the two timers we can achieve minimum and maximum pulse duration's equivalent to the 16-bit FRT. The figure below diagrams the concept:



Timer 0 output toggles on overflow to serve as a clock input to Timer 1

Timer 1 compare match register contains the upper byte of the desired 16-bit count. When the upper timer byte reaches the set match value, TMR1 sets a compare/match flag and issues an interrupt, if enabled. Software should respond to the interrupt or flag by resetting TMR0 to generate the lower byte count.

For example:

To produce a 0.50 second timer, driven by the internal clock, pre-scaled by 1024, use a bit count = H'1312 with a 10 MHz CPU clock. Hardwire TMO0 to TMCI1 clock input.

TMR0: clock source = F/1024, clear TCNT on compare-match A, toggle TMO0 output on match A, compare-match register A= FF.

TMR1: clock source= external clock counted on both the rising and falling edges, compare-match register A= H'13

When TMR1 reaches H'13, the timer sets the compare-match flag and issues an interrupt, if enabled. The CPU responds by re-setting TMR0 to generate the lower byte count. The routine re-loads TMR0 compare match register A with the value H'12 and clears the compare-match flag. Software can either enable the TMR0 compare-match interrupt A or poll the compare-match flag to determine the final time-out.

A driver routine for the above example, designed to run on the H8/330 Evaluation Board is given below:

A similar technique can be used to cascade both 8-bit timers with the 16-bit timer to produce a 32-bit clock.

```
.include "c:\demos\h8338.inc"

;This routine uses two 8-bit timers to generate a 16-bit clock to count 0.25 s.
;Timer 0 clock source is the 10MHz system clock, pre-scaled by 1024. A 16-bit timer
;set to H'1312 will time a 0.25 s period. Timer 1 counts the upper byte, Timer 0
;counts the lower byte. On the H8/330 EVB the port 6.0 LED is toggled to signal the
;end of the period.  At the end of the period this routine repeats the clock cycle
;to cause LED 0 to blink at 0.5 Hz.  The routine can be modified to return to the
;main calling routine by changing the line 'GO2: BRA GO' to 'GO2: RTS'

        .org    h'8500

INI:    mov.w   #h'ff80,r7      ;initialize the stack pointer
        mov.b   #h'02,r0l
        mov.b   r0l,@p4_ddr     ;set P4.1 as output
        mov.b   #h'ff,r0l
        mov.b   r0l,@p6_ddr     ;set port 6 as outputs
        mov.b   #h'00,r0l
        mov.b   r0l,@p6_dr      ;output lows


;Initialize Timer 0

        mov.b   #h'00,r0l
        mov.b   r0l,@tmr0_tcnt    ;clear the counter
        mov.b   #h'ff,r0l
        mov.b   r0l,@tmr0_tcora   ;TMR0 initial compare-match value = h'FF
        mov.b   #h'03,r0l                       ;
        mov.b   r0l,@tmr0_tcsr    ;toggle output TMO0 with C/M A
        mov.b   #h'12,r0l                       ;
        mov.b   r0l,@h'fd80       ;store the lower byte count at the first RAM
                                  ;location

;Initialize Timer 1

        mov.b   #h'00,r0l
        mov.b   r0l,@tmr1_tcnt    ;clear the counter
        mov.b   #h'13,r0l
        mov.b   r0l,@tmr1_tcora   ;TMR1 compare-match value = h'13
;************************************************************
start:  jsr @timer                ;start the timer
main:   bra main

;initializing the TCR registers starts the counters.  Start Timer 0 first.

timer:  push    r0

        mov.b   #h'03,r0l                       ;
        mov.b   r0l,@tmr0_tcr   ;clock source= CPU clock / 1024, no interrupts
        mov.b   #h'17,r0l                       ;
        mov.b   r0l,@tmr1_tcr   ;clock source= external on both edges
        pop r0

go:     jsr    @tst
go2:    bra go                  ;change BRA GO to RTS to return to the main
                                ;calling routine
        nop
        nop

TST:    btst #6,@h'ffd1         ;TEST FOR CMFA FROM TIMER 1
        BEQ     TST
;
```

```
        push    r1

timer1_CMA:
        mov.b   @H'fb80,r1l
        mov.b   r1l,@tmr0_tcora         ;re-set timer 0 compare-match value
        bclr    #6,@tmr1_tcsr           ;clear timer 1 CMFA flag
        bclr    #h'6,@tmr0_tcsr         ;clear timer 0 CMFA flag
        bclr    #h'6,@tmr1_tcsr         ;clear compare-match flag
cma:    btst    #6,@tmr0_tcsr           ;look for the match
        bclr    #6,@tmr0_tcsr           ;clear timer 0 CMFA flag
        beq     cma
        bnot    #0,@P6_DR               ;TOGGLE LED 0

        pop r1
        rts
```

written by: Carol Jacobson, Application Engineering

Hitachi America, Ltd. Semiconductor and I.C. Division
2000 Sierra Point Parkway, Brisbane, CA 94005-1819 (415)589-8300