Hitachi America, Ltd.

#TN-0057

Application Engineering TechNotes

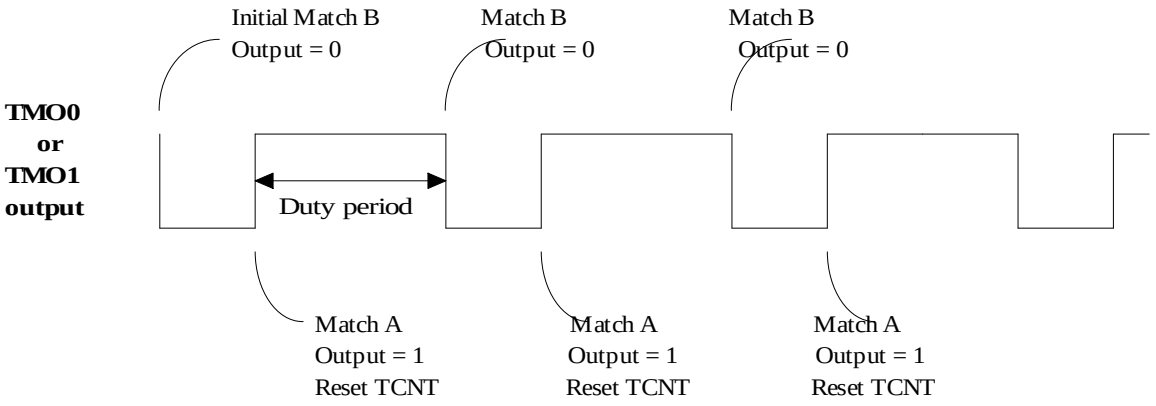
Aug, 1992

Using the H8/3XX series to generate uneven duty cycle pulses

One of the most common applications of the 8-bit timer module in the H8/300 series is to produce a rectangular wave output with an arbitrary duty factor. The pulse frequency depends on the system clock (SCLK), the timer clock, and the counter-match value programmed into one of the 2 Time Constant Registers, TCORA or TCORB. The system clock is provided by an external crystal oscillator whose frequency is halved by an internal divider; its frequency can range between 0.5 and 10MHz. The timer clock is generated internally by scaling down the SCLK to 3 possible values, SCLK/8, SCLK/64, and SCLK/1024; bits 2-0 of the Timer Control Register (TCR) select the desired option. TCORA and TCORB contain the total counter-match value corresponding to the rectangular-wave frequency, and the match value corresponding to the high pulse duration, i.e. the duty factor. These values are programmed into TCORA and TCORB according to the formulas below:

Total counter-match value = Desired wave period/Timer clock period
Duty factor match value = High pulse duration/Timer clock period

During the counting process, the Timer Counter Register (TCNT) contents are incremented on each consecutive falling edge of the timer clock. Programming bits 4 and 3 of the TCR to 01 or 10 clears the counter register when its contents match the total counter-match value, and counting resumes again. The state of the timer output (TMO0 or TMO1) following a match between TCORA and TCNT, and between TCORB and TCNT, is changed to 0 or 1 according to the values programmed into bits 3-0 of the Timer Control/Status Register (TCSR). Hence, the desired pulse could be generated as shown below:



Example:

Create a program that will generate a rectangular wave with a period of 24us and a duty cycle of 33%. Use TMO0 to output it. Refer to the figure above.

Code:

```
;       Specify port addresses

        P4_DDR          .equ    h'ffb5
        P4_DR           .equ    h'ffb7
        P6_DDR          .equ    h'ffb9
        P6_DR           .equ    h'ffbb

;       Specify register addresses

        ISCR            .equ    h'ffc6
        IER             .equ    h'ffc7
        TMR0_TCORA      .equ    h'ffca
        TMR0_TCORB      .equ    h'ffcb
        TMR0_TCNT       .equ    h'ffcc
        TMR0_TCR        .equ    h'ffc8
        TMR0_TCSR       .equ    h'ffc9

;       Specify interrupt vector addresses and program start address

        .org    h'1e
        .data.w MATCH_A
        .org    h'08
        .data.w MATCH_B
        .org    h'500

;       Perform all initializations

        mov.w   #h'ff80,R7              ;initialize stackpointer
        bset    #0,@ISCR                ;IRQ0 sensed on the falling edge
        ldc     #0,CCR                  ;unmask all interrrupts
        mov.b   #h'ff,R0h
        mov.b   R0h,@P4_DDR             ;enable P4 as an output for TMO0
        mov.b   R0h,@P4_DR              ;initialize TMO0 high
        mov.b   #0,R0l
        mov.b   R0l,@P6_DDR             ;enable P6 as input  for IRQ0
        mov.b   R0l,@P6_DR              ;disable pull-ups

;       Use timer 0 to produce the uneven duty cycle waveform

        mov.b   #h'14,R1l
        mov.b   R1l,@TMR0_TCORB     ;load compare-match B reg w/ high pulse duration value
        add.b   #h'0A,R1l
        mov.b   R1l,@TMR0_TCORA     ;load compare-match A reg w/ low pulse duration value
        mov.b   #h'14,R0h
        mov.b   R0h,@TMR0_TCNT      ;start clock w/ match B output high
        mov.b   #h'09,R1h
        bset    #0,@IER             ;enable interrupt MATCH_B to start counter
        mov.b   #h'06,R1l
        mov.b   R1l,@TMR0_TCSR      ;on match B output goes low, on match A it goes high
        nop                         ;wait for counter start
        nop
```

```
ADD:    adds    #1,R4                   ;wait loop during counter timing
        bra     add

;       Start timer 0
MATCH_B:
        mov.b   R1h,@TMR0_TCR           ;clock is /8 and counter cleared on match A
        bset    #6,@TMR0_TCR            ;enable interrupt on MATCH_A
        rte

;       Timer 0 runs to match A value and timer counter is cleared
MATCH_A:
        bclr    #6,@TMR0_TCSR           ;clear compare-match A interrupt flag
        rte

        .end
```

Notes:

1. The timer output pin TMO0 is connected to the IRQ0 pin, and is initialized to a high state at the beginning of the code. By programming bits 4 and 0 of the IRQ Sense Control Register (ISCR) to 01, the MATCH_A interrupt routine is initiated at the first falling edge detected at the IRQ0 pin. Loading the TCSR will cause the timer output to switch low, and therefore this instruction is executed **after** the IRQ0 interrupt is enabled.

2. The port and register addresses can be specified in a separate "include" file that is normally invoked at the start of the program using the command: **include "file path and name".**

written by: Cristian Tomescu, Application Engineering

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