ECE 3120: Computer Systems
Chapter 8: Output Compare Function

Manjeera Jeedigunta

http://blogs.cae.tntech.edu/msjeedigun21

Email: msjeedigun21@tntech.edu

Tel: 931-372-6181, Prescott Hall 120

Output Compare Function

- The HCS12 has eight output compare functions.
- Each output compare channel consists of
  - A 16-bit comparator
  - A 16-bit compare register TCx (also used as inout capture register)
  - An output action pin (PTx, can be pulled high, pulled low, or toggled)
  - An interrupt request circuit
  - A forced-compare function (CFOCx)
  - Control logic

Operation of the Output-Compare Function

- One of the applications of the output-compare function is to trigger an action at a specific time in the future.
- To use an output-compare function, the user
  - Makes a copy of the current contents of the TCNT register
  - Adds to this copy a value equal to the desired delay
  - Stores the sum into an output-compare register (TCx, x = 0..7)
- The actions that can be activated on an output compare pin include:
  - Pull up to high
  - Pull down to low
  - Toggle

Operation of the Output-Compare Function

- The action is determined by the Timer Control Register 1 & 2 (TCTL1 & TCTL2):
- A successful compare will set the corresponding flag bit in the TFLG1 register.
- An interrupt may be optionally requested if the associated interrupt enable bit in the TIE register is set.

	7	6	5	4	3	2	1	0
	OM7	OL7	OM6	OL6	OM5	OL5	OM4	OL4
value after reset	0	0	0	0	0	0	0	0

(a) TCTL1 register

	7	6	5	4	3	2	1	0
	OM3	OL3	OM2	OL2	OM1	OL1	OM0	OL0
value after reset	0	0	0	0	0	0	0	0

(b) TCTL2 register

read: anytime

write: anytime

OMn	OLn	: output level
0	0	no action (timer disconnected from output pin)
0	1	toggle OCn pin
1	0	clear OCn pin to 0
1	1	set OCn pin to high

Figure 8.18 Timer control register 1 and 2 (TCTL1 & TCTL2)

Operation of the Output-Compare Function

- **Example 8.4** Generate an active high 1 KHz digital waveform with 30 percent duty cycle from the PT0 pin. Use the polling method to check the success of the output compare operation. The frequency of the E clock is 24 MHz.

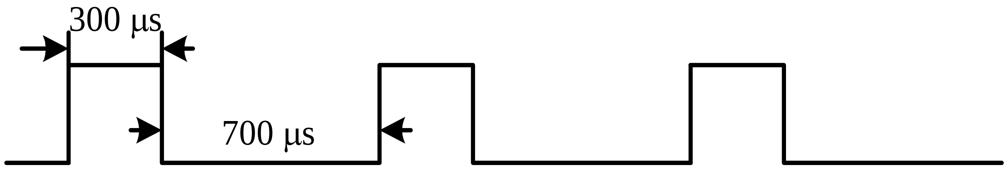
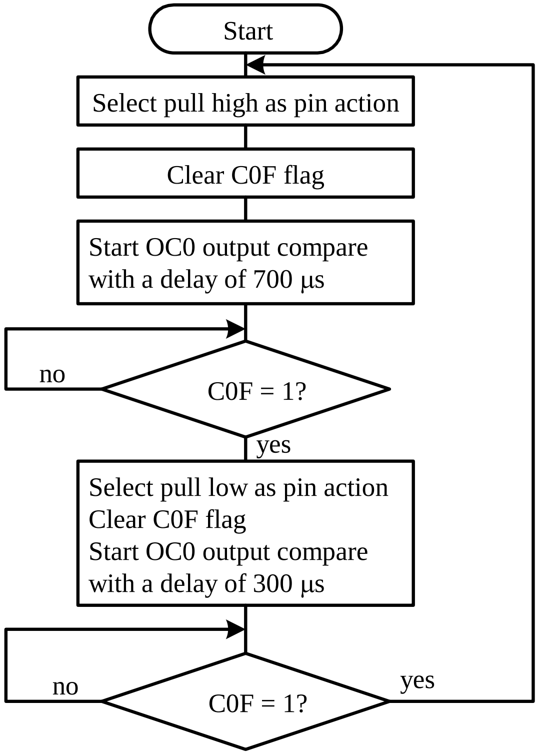


Figure 8.19 1 KHz 30 percent duty cycle waveform

Solution: **An active high 1 KHz waveform with 30 percent duty cycle is shown in Figure 8.19. The logic flow of this problem is illustrated in Figure 8.20.**

Setting the prescaler to the TCNT to 8, then the period of the clock signal to the TCNT will be 1/3 µs. The numbers of clock cycles that the signal is high and low are 900 and 2100, respectively.



Start

Select pull high as pin action

Clear C0F flag

Start OC0 output compare with a delay of 700 µs

C0F = 1?

no

yes

Select pull low as pin action
Clear C0F flag
Start OC0 output compare with a delay of 300 µs

C0F = 1?

no

yes

Figure 8.20 The program logic flow for digital waveform generation

Generating 1 KHz Digital Waveform

```
#include  "c:\miniide\hcs12.inc"
hi_time   equu      900
```

- **Example 8.5** Write a function to generate a time delay which is a multiple of 1 ms. Assume that the E clock frequency is 24 MHz. The number of milliseconds is passed in Y.

- **Solution:** One method to create 1 ms delay is as follows:
  - Set the prescaler to TCNT to 64
  - Perform the number of output-compare operations (given in Y) with each operation creating a 1-ms time delay.
  - The number to be added to the copy of TCNT is 375. (375 × 64 ÷ 24000000 = 1 ms)

```
delayby1ms  pshd
            movb   #$90,TSCR1      ; enable TCNT & fast flags clear
            movb   #$06,TSCR2      ; configure prescaler to 64
            bset   TIOS,OC0        ; enable OC0
            ldd    TCNT
again0      addd   #375            ; start an output-compare operation
            std    TC0             ; with 1 ms time delay
wait_lp0    brclr  TFLG1,COF,wait_lp0
            ldd    TC0
            dbne   y,again0
            puld
            rts
```

Example 8.6 Use an input-capture and an output-compare functions to measure the frequency of the signal connected to the PT0 pin.

- **Solution:** To measure the frequency, we will
  - Use one of the output-compare function to create a one-second time base.
  - Keep track of the number of rising (or falling) edges that arrived at the PT0 pin within one second.

```
#include   "c:\MiniIDE\hcs12.inc"
CR          equu    $0D
LF          equ     $0A
            org     $1000
oc_cnt      rmb     1
frequency   rmb     2
            org     $1500
            movb    #$90,TSCR1      ; enable TCNT and fast timer flags clear
            movb    #$02,TSCR2      ; set prescale factor to 4
            movb    #$02,TIOS       ; enable OC1 and IC0
            movb    #100,oc_cnt     ; prepare to perform 100 OC1 operation, each
                                    ; creates 10 ms delay and total 1 second
            movw    #0,frequency    ; initialize frequency count to 0
            movb    #$01,TCTL4      ; prepare to capture the rising edges of PT0
            movb    #C0F,TFLG1      ; clear the C0F flag
            bset    TIE,IC0         ; enable IC0 interrupt
            cli                     ;       "
```

```
          ldd     TCNT                 ; start an OC1 operation with 10 ms delay
continue  addd    #60000               ;                    "
          std     TC1                  ;     "
w_lp      brclr   TFLG1,C1F,w_lp            ; wait for 10 ms
          ldd     TC1
          dec     oc_cnt
          bne     continue
          ldd     frequency
          pshd
          ldd     #msg
          jsr     [printf,PCR]
          leas    2,sp
          swi
msg       db      CR,LF,"The frequency is %d",CR,LF,0
TC0_isr   ldd     TC0                  ; clear C0F flag
          ldx     frequency            ; increment frequency count by 1
          inx                          ;   "
          stx     frequency            ;
          rti
          org     $3E6E                ; set up interrupt vector number
          fdb     TC0_isr              ; for TC0
          end
```