- **The 9S12 Output Compare Function**
- Huang Section 8.6
- ECT_16B8C Block User Guide
    - Review of Timer Overflow and Input Capture
    - Making an event happen at a specific time on the HC12
    - The 9S12 Output Compare Function
    - Registers used to enable the Output Compare Function
    - Using the 9S12 Output Compare Function
    - A program to use the 9S12 Output Compare to generate a square wave
    - Setting and clearing bits in the Timer Subsystem
    - Introduction of Pulse Width Modulation

Setting and Clearing Bits in the Timer Subsystem

• Registers in the timer subsystem control multiple timer channels.
– Usually, you want to use ANDS and ORS to change only that channel you are working on.
– For example, to make Channel 2 an output compare, and set it to toggle on compare, do this:

```
TIOS = TIOS | 0x04;                 /* Configure PT2 as Output Compare */
TCTL2 = (TCTL2 | 0x10) & ~0x20; /* Set up PT2 to toggle on compare */
```

– Do not do this:

```
TIOS = 0x04;                        /* Configure PT2 as Output Compare */
TCTL2 = 0x10);                      /* Set up PT2 to toggle on compare */
```

This would set up Channel 2 as an output compare, toggle on successful compare. However, it will force all the other channels for other functions – this may not be what you want to do.

• To clear a flag bit, do not use ORs!
– To clear Timer Channel 2 flag, do the following:

```
TFLG1 = 0x04;
```

This will clear Timer Channel 2 flag, and leave all other flags unaffected.

– Do not do this:

```
TFLG1 = TFLG1 | 0x04;                /* DO NOT DO THIS */
```

This will clear Timer Channel 2 flag, but will also clear any other flag which is set. Suppose, for example, Timer Channel 2 and Timer Channel 3 flags are both set at the same time, so TFLG1 register is 0x0C. You want to deal the Timer Channel 2 first and Timer Channel 3 afterwards.

Program to use output compare subsystem

```
/*
* Program to generate square wave on PT2
* Frequency of square wave is 500 Hz
* Period of square wave is 2 ms
* Set prescale to give 0.667 us cycle
* 2 ms is 3,000 cycles of 1.5 MHz clock
*
*/
#include "hcs12.h"
#include "vectors12.h"

#define PERIOD 3000
#define HALF_PERIOD (PERIOD/2)
#define TRUE 1
#define enable() asm(" cli")

void INTERRUPT toc2_isr(void);

main()
{
        TSCR1 = 0x80;                /* Turn on timer subsystem */
        TSCR2 = 0x04;                /* Set prescaler to 0.666 us */
        TIOS = TIOS | 0x04;          /* Configure PT2 as Output Compare */
        TCTL2 = (TCTL2 | 0x10) & ~0x20; /* Set up PT2 to toggle on compare */
        TFLG1 = 0x04;                /* Clear Channel 2 flag */

        /* Set interrupt vector for Timer Channel 2 */
        UserTimerCh2 = (unsigned short) &toc2_isr;

        /* Enable interrupts on Channel 2 */
        TIE = TIE | 0x04;
        enable();
        while (TRUE)
        {
                asm("wai");
        }
}
```

```
void INTERRUPT toc2_isr(void)
{
        TC2 = TC2 + HALF_PERIOD;
        TFLG1 = 0x04;
}
```

Pulse Width Modulation

• Often you want to control something by adjusting the percentage of time the object is turned on

• For example,
– A DC motor — the higher the percentage, the faster the motor goes
– A light – the higher the percentage, the brighter the light
– A heater – the higher the percentage, the more heat output

• Can use Output Compare to generate a PWM signal, but PWM is used so often the HCS12 has a built-in PWM system

• The PWM system on the HCS12 is very flexible

– It allows you to set a wide range of PWM frequencies
– It allows you to generate up to 8 separate PWM signals, each with a different frequency
– It allows you to generate 8-bit PWM signals or 16-bit PWM signals
– It allows you to select high polarity or low polarity for the PWM signal
– It allows you to select left aligned or center aligned PWM signal

• To simplify the discussion we will only discuss 8-bit, high polarity, left-aligned PWM signals.

Need a way to set the PWM period and duty cycle
The HC12 sets the PWM period by counting from 0 the some maximum count with a special PWM clock

PWM Period = PWM Clock Period x (Max Count + 1)

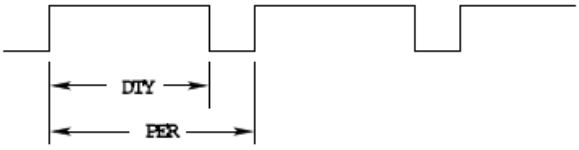
Once the PWM period is selected, the PWM duty cycle is set by telling the HC12 how many counts it should keep the signal high for

PWM Duty Cycle = (Count High + 1)/(Max Count + 1)

The hard part about PWM on the HC12 is figuring out how to set the PWM Period

The HCS12 Pulse Width Modulation System

Control speed of motor by adjusting percent of time power is applied to the motor.
Need to choose period, and have a way to adjust duty cycle



• The HCS12 has a flexible, and complicated, PWM system

• There are eight 8-bit PWM channels
– Two 8-bit channels can be combined into a single 16-bit channel
– We will discuss only 8-bit mode

• You can select high polarity or low polarity
– We will discuss only high polarity mode

• You can select center-aligned or left-aligned PWM
– We will discuss only left-aligned mode

• Full information about the HCS12 PWM subsystem can be found in PWM 8B8C Block User Guide.

• To select 8-bit mode, write a 0 to Bits 7, 6, 5 and 4 of PWMCTL register.

• To select left-aligned mode, write 0x00 to PWMCAE.

• To select high polarity mode, write an 0xFF to PWMPOL register.

• To set the period for a PWM channel you need to program bits in the following PWM registers

– For Channel 0 the registers are PWMCLK, PWMPRCLK, PWMSCLA and PWMPER0
– For Channel 1 the registers are PWMCLK, PWMPRCLK, PWMSCLA and PWMPER1
– For Channel 2 the registers are PWMCLK, PWMPRCLK, PWMSCLB and PWMPER2
– For Channel 3 the registers are PWMCLK, PWMPRCLK, PWMSCLB and PWMPER3
– For Channel 4 the registers are PWMCLK, PWMPRCLK, PWMSCLA and PWMPER4
– For Channel 5 the registers are PWMCLK, PWMPRCLK, PWMSCLA and PWMPER5
– For Channel 6 the registers are PWMCLK, PWMPRCLK, PWMSCLB and PWMPER6
– For Channel 7 the registers are PWMCLK, PWMPRCLK, PWMSCLB and PWMPER7

• To set the duty cycle for a PWM channel you need to write to the PWDTYn register for Channel n.