ECE263

Lab 6

Spring 2013

Enhanced Capture Timer Module
ECT

Revised: 12 Nov 2012 DAL

(intentionally left blank)

Object:

In this lab you will be using the input capture and output compare functions of the Enhanced Capture Timer (ECT) module of the MC9S12DG256 to perform some simple timing functions. You will use the logic analyzer to verify the correct operation of your programs.

Material:

CodeWarrior software package
Dragon 12 Plus Development Board
P&E USB Multilink
CPU12 Reference Guide (CPU12RG)
Agilent 16801A Logic Analyzer
One flying lead logic analyzer probes

Background

The ECT contains eight channels each of which can be configured to perform either the input capture function or the output compare function and all eight channel share the same 16-bit free running counter. In order to conserve pins, the ECT shares the I/O pins of the MC9S12DG256 with parallel Port T.

When configured for the output compare function, the I/O pin associated with a channel is set to be an output and can be commanded to go to a high state, a low state or to toggle (i.e. take on the opposite state from the current state) at some time in the future when the contents of the free running counter match the value contained in the channel register. In this mode the channel acts like the alarm function on an alarm clock: you select the time that you want the alarm to sound (channel register) and then when the real time (free running counter) counts up to reach the value contained in the alarm register, the buzzer goes off (pin takes on commanded state). The ECT has an 8-bit status register which contains one bit for each channel. When the output event occurs on the pin, the status bit for the channel is set true so the program can determine that the time period has elapsed and the output pin has changed state. The channel can be programmed for only one event at a time which is the next event that will occur at some time in the future. If you need to generate some arbitrary waveform that consists of a series of transitions at various times (frequencies) then you program the events in series: program the channel for the first event, wait for it to occur by monitoring the status bit, when it occurs, reset the status bit and then program for the next event.

When configured for the input capture function, the I/O pin associated with the channel is set to be an input and when it detects a specific transition, either high state to low state, low state to high state or toggle (i.e. a change from either state to the opposite state), the channel register is loaded with the contents of the free running counter and the status bit for the channel is set. In this mode, the channel acts like a stopwatch: the watch is counting up in time and when the user pushes the stop button (when the commanded state is detected), the current time is captured on the display (the free running counter is loaded into the channel register). Like the output compare function, the input compare function can be programmed for only one operation at a time. If you need to detect a series of transitions, they must be programmed one at a time: program the transition desired, wait for it to occur by monitor the status bit, when it becomes set reset it, then read the channel register and save it in memory, and finally program for the next event. Projects that require complex timing tasks can combine the operations of multiple ECT channels by programming various combinations of functions.

Design Description

The program for this lab will perform two different tasks using three separate channels of the ECT. It should be written so that both programs can be running at the same time without interfering with each other. One program is to generate an arbitrary output waveform on a timer channel and the second program is to use two channels working together to generate an output pulse on one channel after every low to high transition of an input signal on a second channel. The program(s) should use interrupt service routines to control the ECT.

Arbitrary waveform

You are to use a channel of the ECT to generate an arbitrary waveform output that has 8 separate high/low components. The length of each part is to be contained in a table of 16-bit values representing the delta counts of the free running counter TCNT. The table should be located in RAM so the values can be varied at execution time For the sake of simplicity assume the bus clock is 4MHz and the pre-scalar is set to divide by 1 which gives 250ns/clock tick. Also assume that all deltas will be between 0.1ms to 16msec. For all even entries of the table, the output should be in the high state and for all odd entries the output should be in the low state. After all table entries have been used the program should revert back to the first entry and repeat the process so that the waveform repeats forever.

Output Pulse

This program is going link the operation of two ECT channels together. One channel will be connected to an external signal and will be set to look for low to high transitions. Every time a transition is detected, the program will generate a low going pulse after a specified delay. The length of the delay and the length of the pulse will be stored as 16-bit global variables containing the delta counts of TCNT. Again for simplicity you can assume that each clock tick is 250nsec and that the delay and pulse width will be between 0.1ms to 16msec. In addition you can ignore any additional lo-hi transitions of the input until after the output pulse is complete.

Preparation:

1. Draw a timing diagram of the arbitrary output waveform for Part 1 of the program
2. Draw a timing diagram of the input signal and output pulse for Part 2 of the program.
3. Generate a flow chart or pseudo code to show the design of your program(s).
4. Calculate the deltas for the following times: 0.2ms, 0.5ms, 1ms, 2ms, 3ms, 4ms 5ms, and 10ms

Prelab:

Do the steps above **BEFORE** coming to lab. I guarantee that you will not have time to design the program, write the code, compile it and debug it all in the time allocated for the lab.

Procedure: Working with your lab partner(s), complete the following steps:

1. Create a new project and enter your program as *main.asm*. Generate the executable object file using the *Make* facility.
2. Set up the logic analyzer to monitor the three bits of Port T that you used as timer channels for your program.
3. Set up the function generator:
    a. Set the frequency at 100 KHz.
    b. Enable the SYNC output (SYNC ON from Utilities menu)
4. Attach the SYNC of the function generator to pin of Port T you used as the input channel.
5. Download your program to the Dragon 12 Plus.
6. Set the 8 values in the waveform table as follows: 2ms, 0.5ms, 1ms, 10ms, 4ms, 0.5ms, 3ms, 0.2ms, and 5ms.
7. Set the global variables for Part 2 of your program as follows:
    a. delay from input detect to start of output pulse = 4ms
    b. pulse width of output pulse = 0.5ms

<u>Test Part 1</u>

8. Start testing the waveform program by triggering the analyzer on output pulse and verify that it is generating the correct timing. Plot some traces for the lab report.
9. Correct your program as necessary.
10. Modify the table entries and recheck your results and get more analyzer traces for the lab report.

<u>Test Part 2</u>

11. Verify that the output pulse is being generated correctly by triggering on the output of the function generator (i.e. input channel) and measuring the timing of the output pulse. Plot some traces for the lab report
12. Correct you program as necessary.
13. Change the frequency of the function generator and recheck your results.
14. Modify the two global variables and recheck your results and get more analyzer traces for the lab report.

15. Demonstrate the operation of the program to the instructor or the TA and get the Lab Verification sheet signed.