LAB #6: DIGITAL STOPWATCH DESIGN WITH PROGRAMMABLE TIMERS

INTRODUCTION

Real-time systems often require precise timing for data sampling and control operations. This is usually achieved through the use of a programmable timer circuit that periodically interrupts the main processor at precise times. To study timed interrupt-driven operation, this lab will require you to design and implement a digital stopwatch using one of the HCS12 programmable timers to interrupt the CPU.

Two sets of 4 LEDs on the *Waveforms Static I/O Instrument* are to be used to display (in BCD format) time in seconds and tenths of a second (0.0 to 9.9, and then roll over to 0.0 and continue). The stopwatch is to be controlled with two buttons on the keypad interfaced in the previous lab (preferred), or with two push buttons on the *Waveforms Static I/O Instrument*:

Button 0: start/stop timing
Button 1: clear the display to 0

Depending on how you implemented the keypad, either a loop in the main program is to continuously test these buttons, or a keypad interrupt routine should determine the pressed button, to determine when to start the timer, stop the timer, and reset (clear) the display. The initial design should be done prior to lab. The HCS12 timer module "output compare" function is to provide the time base. Whenever the stopwatch is running, the timer interrupt service routine should update the display every 0.1 second. When the stopwatch is stopped, the display should freeze at the last time value. If the clear button is pressed while the stopwatch is stopped, the display should reset to 0.0. The clear button should have no effect while the stopwatch is running. When the stopwatch is restarted, the time should continue from the displayed value.

HCS12 FAMILY PROGRAMMABLE TIMER FUNCTIONS

Refer to your course notes from ELEC 2220, the Cady text, and the class presentation slides for an overview of HCS12 timer functions. In this lab, the timer output compare function is to be used for generating timer interrupts. If time permits, you might experiment with modifying your design to use timer overflow interrupts or the HCS12 real-time interrupt module.

PRE-LAB ASSIGNMENT

Reading

Review Chapter 14 of the Cady text, 2nd edition, which describes the MCS12 timers. Link to ELEC 2220 lectures on programmable timers and interrupt operation are available on the ELEC 3040/3050 web page.

Software Design

Design a C program that initializes the timer module, initializes the stopwatch to 0.0, and then begins sampling the keypad, if using program-controlled I/O for the keypad, or enters a "do-nothing" loop to wait for a keypad interrupt. Operation is to be as follows.

1. When the start/stop button is pressed for the first time, timer output compare interrupts should be enabled and the stopwatch should begin running, updating the display every 0.1 seconds. You may use any of the eight timer channels. *The timing must be precise.*
2. When the start/stop button is pressed again, timer output compare interrupts should be disabled, effectively stopping the timer and freezing the display at the time corresponding to the instant at which the button was pressed.
3. Pressing the start/stop button again should resume timing at the current time on the display; the time should be reset to 0.0 only if the display has been cleared.
4. The clear button should be acknowledged if and only if the stopwatch is stopped, resetting the display to 0.0 if pressed.

LAB PROCEDURE

1. Implement your hardware design, adding a second set of 4 LEDs to your previous circuit using Port AD, pins 7-4. If you wish, you might also find it useful to connect another LED or two that can be used for debugging (Ports A and B have one bit each that can be used.)

2. Compile and download your test program. To begin, verify that you can start and stop the timer and trigger timer interrupts. (Consider using the logic analyzer to capture the signals from the keypad and those going to your seven-segment displays.) Then verify that the clear operation works. Demonstrate a working system to lab instructor.

DELIVERABLES THIS WEEK

Lab notebooks are to be submitted to the GTA at the end of your lab session.