UNIVERSITY OF CALIFORNIA, SANTA CRUZ
BOARD OF STUDIES IN COMPUTER ENGINEERING

CMPE12/L: COMPUTER SYSTEMS AND ASSEMBLY LANGUAGE

Lab 5: Introduction to Uno32 and MIPS
Worth 60 points (50 lab + 10 report)

Lab Objectives:

With our time spent in the LC-3, it is time to move on to real hardware, and an assembly language that is used today. This lab will introduce you to the basics of MIPS and running code on an embedded processor. We will do so by writing code, in MIPS assembly, that will light LED's based off button presses and bounce LED's back and forth.

Part 1:

Please refer to the getting started guide on how to set up the starter file. Once the hello world message has been started you can continue to work on the rest of the lab. The first part of the lab will involve working out what buttons connect to what pins. Refer to both the uC32/uno32 manual and the IO Shield manual to build your mapping. Once you have your mapping you should be able read the status of the switches along with set the values of the LEDs. Remember that you will need to use the TRIS register to do so.

Once you are comfortable reading buttons and turning on LED's program a continuous loop that does the following:

- Read the value of each of the 4 switches (SW1 – SW4) and turn on LD1-LD3 if the corresponding switch is "on".
- Read the state of the 4 buttons (BTN1 – BTN3) and turn on LD4-LD8 if the corresponding button is being pressed. Unlike the switches, the buttons are "momentary", meaning they are have to be held down to be active

Part 2:

In this part you will be working with the delays (software and hardware based) of the Uno32 to come up with 16-levels of delay for the cycling through the LEDs. Start by making a function called mydelay which uses a software loop to "waste" time. Send in an argument in $a0 register which will be a multiple of the base delay. Basically make a for-loop in your mydelay which repeats $a0 times. Print out a greeting message.

Use this delay function to do the following:

- Read in the value of the 4 switches (0-15 to match to 1-16 multiplier)

- Turn on LED1. Wait for a 1-16x multiple of the mydelay, so a lower setting on the switches will result in a faster turning off. After the delay turn the LED1 off.
- Turn on LED2. Repeat.
- Got through all the LEDs.
- When you have reached the last LED's, reverse directions and continue the process.
- Continue to bounce the LED back and forth forever.

We are writing code to make delays. There is hardware on board that can do this delay for us but we won't be using it for this class. While waiting like this could be bad in some situations, many simple situations this makes a lot of sense.

Lab Requirements

You must do the following for this lab:

- Determine the mapping between the LED's and the PIC32 pins.
- Demonstrate the ability to turn on and off LED's using the hardware switches and buttons.
- Walk an LED pattern using software delays.

Files to Submit on Canvas

- Lab5Part1.asm
- Lab5Part2.asm
- Lab5_report.txt

Check-off

You should demonstrate your lab to the TA/Tutor when finished to get it checked off.

Point Breakdown

- 5 pts: Correctly reads in the 4 switches
- 5 pts: Correctly displays the 4 switches to the LEDs
- 5 pts: Correctly reads in the 4 buttons
- 5 pts: Correctly displays the 4 buttons state to the LEDs

- 4 pts: Reads in the 4 switches and puts the number 1-16 into $a0
- 10 pts: Has a mydelay procedure which takes $a0 as an argument and delays for some based period * $a0
- 4 pts: uses the mydelay procedure to control the turning on and off an LED
- 10 pts: Does a standard pattern of turning on and off an LED and moving to the next LED based upon the switches settings
- 2 pts: Have required comments in the code (block personal info, line comments, register usage)
- 10 pts: Lab Report