Lab #4
Adding a Liquid Crystal Display (LCD)
to your Qsys system

EELE 475
HARDWARE AND SOFTWARE ENGINEERING
FOR EMBEDDED SYSTEMS

Assignment Date: 9/20/11

Memo Due Date: 9/27/11

Lab Description

The DE2 board has a LCD on it and today's lab is to create a Qsys system that will use it. Like last week, you will need to use the following Altera tools:

- **Quartus II** - To compile the VHDL that creates your processor hardware.
- **Programmer** - To download your bit stream to the DE2 board. (QuartusII -> Tools -> Programmer)
- **Qsys** - To create your processor system. (QuartusII -> Tools -> Qsys)
- **NIOS II IDE** - To write, compile, and download your C program. (Qsys -> Tools -> Nios II Software Build Tools)

Your starting point will be the system you created last time. Create a new lab4 directory and copy all your files from lab3 into this directory.

Part 1 – Qsys Modification

1. Add the LCD Peripheral, which is found at: Peripherals->Display->Character LCD

2. Name it: lcd_display (This is the name that the diagnostic program expects it named as). Connect up the clock, reset, data, & export the external signals. Name the lcd_display export as lcd.

3. Re-assign the base addresses and re-generate the system.

4. Remember to modify the Nios_Qsys component and instantiation code in your top level VHDL file. You will need to connect the LCD signals coming out of the Nios_Qsys component to the appropriate signal names found in the DE2_Board_top_level.vhdl file. Comment out the LCD signal that are being driven by the default signals.

5. Compile the quartus project and download the bitstream to the DE2 board.

6. Note: you should have the following Qsys modules in your design:

a. cpu (lab 2)
b. sdram (lab 3)
c. jtag_uart (lab 2)
d. LEDs (lab 2)
e. Switches (lab 2)
f. Sysid (optional – this is useful for preventing Eclipse projects from being loaded onto the wrong Qsys hardware design)
g. LCD (lab 4)

<u>Part 4 – Software</u>

7. Open up the NOIS II Eclipse IDE (you can do this from within Qsys -> Tools -> *Nios II Software Build Tools for Eclipse*)

8. Set the workspace to the directory Eclipse_Workspace, which you should have created.

9. Select File->New->*Nios II Application and BSP from Template*.

a. For the SOPC Information File name, browse to the file nios_qsys.sopcinfo found in the DE2_System directory (or where you created your project).

b. Name your Project name “lcd”.

c. Select the Project Template *Board Diagnostics*. In this code you can see how to use the LCD display.

10. Compile and run the diagnostic program. You should see the program write to the LCD display. Don’t worry if it is not scrolling.

11. Strip out the code you don’t need and re-package it in function calls, which you will use later. The idea here is to create lcd print functions that make printing to the lcd easy for all subsequent labs. Make functions that you can use for the next step (#12).

12. For this lab you will need to implement the following functionality:

a. On the DE2 board, if switch SWx (where x can be 0, 1, 2, or 3) is selected, line1 (first line on the LCD) should display: “SW X is active”. If multiple switches are on, the display shouldn’t change until only one switch is on.

b. On line 2 the LCD display should read: “SW X used last”, i.e. it gives the history of the last switch used (what was previously line 1).

13. Explore the system.h header file.

Part 4 – Run the Software

14. Compile and run your program like you did in the previous lab

Deliverables

1. **Have the instructor verification sheet (given below) signed off when you get the LCD printing the messages on the display (i.e. step 12).**
2. **Write up a memo describing the functions that you created for making printing to the LCD easy.**
3. **Load the source code for these LCD printing functions to the D2L site.**

Instructor Verification Sheet
Turn in to get credit for Lab #4

Lab #4
Adding a LCD to your SOPC

EELE 475
HARDWARE AND SOFTWARE ENGINEERING
FOR EMBEDDED SYSTEMS

Name : ________________________

Demo #1 : Show that the LCD displays the appropriate messages reflecting the switch [0:3] positions. Line 1 – current single switch. Line 2 – history.

Verified: ___________________________________ Date:____________________