Lab 3: Memories

Worth 50 points (40 lab work + 10 lab report)
Lab source and report txt files submitted on ecommons
Check offs will use the submitted source files and can happen the week after submitting

Lab Objective:

You now have had a chance to play around with some basic combinational logic circuits. Let's move on to something a little more interesting, sequential logic. From class you learned that combinational logic was just a function of current inputs and that sequential logic was a function not only of current input, but some past sequence of inputs. To store that past sequence information we need some storage devices. In this 2 part lab you will look into the basic storage element used in the LC-3, the D-latch, and modify it to support a reset signal. You will then design and build some logic that will allow you to read and write some registers (which are just a group of D-latches that are read and written together).

Part A: The basic storage element

In this part you will modify the D-latch example in Multimedia Logic to support an asynchronous reset signal. Asynchronous means that it does not need to be qualified with the clock. Thus the reset should set the "Q" to 0 and the "QN" to 1 whenever it goes HIGH.

- Start by playing with the D-Latch file in examples\Basic\DLFF.LGI. The label says "D-Latch Flip Flop" which is not completely correct; it is just a D-Latch. Simulate the design and notice how the "Q" and "Not Q" outputs only change to reflect what "D" is when the "C" button is pressed.
- Create a new schematic called Lab3A.LGI.
- Figure out how you can "reset" this circuit to have "Q" be a 0 and "Not Q" be a 1 by adding a new button labeled "R" to the circuit, this way whenever you press the reset button the D-Latch goes back to Q=0 and QN=1.
- For some hints on how to figure this out look in the text book p64-67 which cover R-S Latches and D-Latches.
- Build a truth table for the functionality you want as well. This should be in your write up.

It might take some experimentation to figure this out. The instructor's solution (and there can be more than one) only required changes to some of the existing gates input sizes.

Part B: A simple register file

Now that you know a bit about the LC-3 Architecture and how registers work and how inputs are selected lets apply that to do some reading and writing. You will be building a 2 by 4 memory called a "register file". This means you will have 2 registers, each 4-bits in size. You will either read from one of the two registers or write to one of them (thus reads and writes are always 4-bits in size). If you are reading, the result should be stored in a set of 4 latches, the output of which should be connected to a 7-segment display. Thus the display should only change when a read is done, not when a write is done. To help you understand this a bit more look at the file "Block_Diagram.pdf". This shows a block diagram for what your logic could look like. The TA/tutors will go over this in lab section.

- Open a new schematic and call it Lab3B.LGI.
- Start with the file "input.lgi" for the keypad input that will be stored when doing a write (maybe add a 7-segment to this so you always know what will be written). Feel free to change this as you like.
- Use the 7-segment tool to display contents when doing a read (remember you will be need 4 latches to hold the value read).
- Use 1 switch for your Read/Write signal and 1 switch for your address select, since only 2 locations to address.
- Use a button for your clock/control input, only when this button (or the reset button) is pressed should a value be written or read out of the register file.
- Use a button to add in a global reset signal so you can reset all the registers, including the display latches to 0.
- Use multiple pages; the first page should have your clock & reset buttons, the keypad, the address & Read/Write switches, and the 7-segment display. Another page (or more, you decide) should have the address decode logic, register file and remaining logic. See the file examples\Devices\signal.lgi on how to use the "signal sender" and "signal receiver" tools if you have not done so yet.

Your circuit should function as follows:

- State changes ONLY when the clock or reset buttons are pressed.
- The 7-segment output display ONLY changes when a read occurs or you are resetting the circuit.

Feel free to add LEDs or other devices to your schematic to make it more understandable or fun for you.

Lab Report and Documenting

You should include your truth table from Part A and a discussion of how you created your basic storage element. For Part B you should discuss how you are controlling the

read/write of one “word” of the register file at a time. What is the word size? Address space? How about the addressability? Explain the over flow of data through the circuit. If you have a need for a figure please reference your schematics or include images separately.
Your schematics should have the same block comments and labels as past labs.
Collaboration: ***You are allowed to discuss this lab with other students on this lab but all the work must be your own.***

Files to Submit:

- **Lab3A.LGI**
- **Lab3B.LGI**
- **Lab3_report.pdf**

Check-off: You should demonstrate your lab to the TA/Tutor when finished to get it checked off.
General grading template (total points possible: 50):
Sign off is worth at most 40 points:
A: 36-40 points. Have a completely functional circuit that is labeled well and laid out neatly using multiple sheets
B: 32-35 points. Have a completely function circuit that is messy, has minimal comments, and may not use enough sheets
C: 28-31 points. Circuit that mostly works with some functional issues or minor glitches,
D: 24-27 points. Circuit looks to have had some effort put into it but is not complete nor functional.
Lab report is worth up to 10 points.

Happy Designing!!