EEEE-220-Lab7

1) From mycourses download on your desktop this pdf file. This lab consists of two parts.
2) **Objective**: The objective of this lab exercise is to execute the paper and pencil design of the fmlRISC data path following the template and example covered in class.
3) **Part 1**: fmlRISC – DP – P&P – Design
4) Make sure you have the last three lecture handouts.
5) Convert the decimal number you have received in the lecture to a 7-bit binary number. Enter it in the appropriate table in your handout. Circle your individual design specifications in the next table. For this step you are required to use a pen (any color) and not a pencil.
6) As in the lecture, use one or more landscape oriented letter sheets to draw your DP schematic. Use two ASM templates to capture the machine cycles of your instructions. Use the control signals table to capture the control signal values during each machine cycle.
7) Your TA will have to check your design before you start implementing it in Lab8.
8) Each student will have to meet with the instructor for ~15 min. before the lab next week to show her/his work, even if it won't be complete at that time. **Without this meeting the grade for this lab will be 0. More details to follow in the lecture.**
9) If you choose to do so, you can use a drawing program for the schematic and/or ASM Chart (Microsoft Visio is an example), and a Word or Excel table for the control signal table. However, this may be more time consuming and it doesn't carry any points. Nonetheless, neat drawing and writing is expected and required.
10) **Part 2:** About DEBUGGING and the debugging process.
11) Go to Wikipedia and read about "debugging".
12) Debugging is the systematic process during which:
    a. In the case of verification, one detects and fixes design errors. These errors can have different causes:
        i. Syntax errors
        ii. Semantic errors
        iii. Real design (structural, data flow, or behavioral) errors.
    b. In the case of testing a real product implementation, be it hardware or software, one detects and fixes faults and/or errors. Note that a fault may exist, but may go undetected if it doesn't result in an identifiable error during the debugging or testing process. Faults and errors can different causes:
        i. Design errors – rare if the verification was rigorous and the design specification were met or didn't change in the course of the design.
        ii. Manufacturing or implementation errors – most often. Examples:

1. Open wires or connections (due to broken wires or "cold" solder/contact points)
2. Short-circuits between wires or connections.

iii. We will cover fault types and models in the testing chapter.

13) As you are already aware, from a subjective/human point of view, the debugging process tends to be very frustrating and time consuming. The only way one can shorten the debug time is by being disciplined and rigorous during the capture phase of the design, i.e. entering the debug phase with as few errors as possible.
14) To alleviate the anxiety related to the debugging process, I recommend considering the following parallel example activity.
15) Based on mass-media statistics, a majority of TV watchers like movies and documentaries that cover police detective work.
16) Debugging is similar to the work a detective does. You are the detective. The DUT, i.e. your design, is the environment (location, society) you operate in. The "bugs" are the reasons why some parts of your design don't work.
17) You start debugging by investigating the source of the errors you see. In the case of digital hardware, you apply known stimuli, and check to see if the circuit outputs the expected output values.
18) If not, you have to trace it down, from outputs to inputs, or vice-versa, and find the location in your circuit that doesn't function as expected, i.e. the "culprit".
19) Once you discover the "culprit", you return to the design (code) and make the necessary changes.
20) The nice thing about debugging is that nobody gets hurt and no one goes to jail. That's for the movies.
21) No matter what kind of electrical engineering design work you will perform in the future, you cannot escape the debugging process. Thus, you'll have to find a way to "live with it".
22) There are also rewards in the debug process. The satisfaction of having debugged and completed a new and complex design is not unlike the satisfaction of completing other creative works. As Irving Stone put it: "The Agony and the Ecstasy".
23) **Grading**:
    a. 20 points for the following ten instructions: ADD, SUB, INC, DEC, NOT, AND, OR, SHR, SWAP, CPY. Per instruction: 1 point for the ASM Chart portion and 1 point for the hardware schematic necessary to implement it.
    b. 18 points for the remaining six instructions. Per instruction: 1 point for the ASM Chart portion and 2 points for the hardware schematic necessary to implement it.
    c. 2 points for the control signal table.
24) Show your complete design to the TA, preferably during her/his OH, i.e. before the due date and time. Within the first hour of next week's lab she/he won't have time to check all 16-22 designs.

25) Furthermore, remember: Each student will have to meet with the instructor for ~10 min. before the lab next week to show her/his work, even if it is not yet complete. **Without this meeting the grade for this lab will be 0. More details to follow in the lecture.**
26) Write your report and upload it along with scanned or photographed copies of your ASM Chart and schematics in the dropbox on mycourses, as described in the lab policy.
27) Even if you don't complete your entire design before the due date and time, you should still upload whatever you have up to that point on mycourses!
28) This concludes this week's lab.