**ELEC 5200/6200 Final Project Report**

Myers Hawkins and George Conover

**What did you learn from this project?**

 In this project, we learned how to create a CPU. This meant creating an instruction set, choosing a datapath and architecture, and implementing everything in a hardware description language. Eventually, we implemented the CPU in an FPGA, testing and debugging our design. We choose a multi cycle datapath, whereupon we found the complexity in multi cycle design not apparent in the lectures. The problems we encountered were in branching and memory access functions. We found that the data needed to be written into the program counter or the memory came a clock cycle after we thought it should. Adding another cycle in the process fixed this problem, but added to execution time. The only unresolved issues were the load function (where we suspect the problem was the same as it was for the store function) and the branch if greater than function. The branch issue only gave an incorrect result if the two numbers compared were equal; otherwise all of the branch functionality (including other comparisons for branch if greater than) worked correctly.

**What would you do differently next time?**

 If asked to do the project again, we likely would choose a different datapath. A single cycle datapath would seem to be much less complex, despite the added architecture. A pipeline datapath would also be a possibility, as we now know how to create a multi cycle one and a pipeline would simply be building on what we have already done. We would also be more conscious of how we designed the instruction set, so that implementation of the hardware would be more straightforward.

**What is your advice to someone who is going to work on a similar project?**

 The first thing to do is to look ahead. Try to see where the decisions you make now will impact the project later on. Give yourself some time to analyze the different datapaths before it comes time to choose which one you will use. Try to see how the binary code you use for the instructions will be implemented in hardware. You should not be afraid to change things along the way. The binary code you choose early on is not necessarily the setup you will need in the end. In the In the real world, changing things decided on early will not always be possible, so take advantage of this while you are able. We would also recommend finding someone to work with on the project. Not only does this split the work load, but one person may see a problem the other wouldn’t, saving a considerable amount of time. Just be sure you understand the work your partner has done, so that you can both understand the how the project works. This helps a lot when debugging.