**CPU Project Report**

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

The project exposed us to the design considerations involved in realizing a MultiCycle datapath in VHDL. Each part of the project provided us with different lessons. First, the ISA should be kept simple. The placement of register operands in the instruction format are important and will affect hardware implementations in later parts. The size of the opcode will determine how many unique instructions the RISC CPU can realize. Second, implementations of hardware components should be kept as simple as possible. This includes the control unit. The control unit will need to be modified as the project progresses, so making it simple is quite important. Third, isolating and testing each component mitigates the number of problems that can come up in later parts of the project. The most crucial components (control unit and ALU) should be tested thoroughly using a testbench. We learned how to implement test benches that would read test vectors from a file. Having the test vectors in a file enabled us make quick changes to the inputs being applied and the outputs being asserted for the unit under test. In addition, some basic lessons learned include: the importance of keeping code simple/readable and building independent components for a multicycle datapath. The most important lesson we learned was that all design decisions made in beginning portions of the project will affect design options later. These decisions can affect the number of hardware components needed to realize the datapath. They can also cause problems that need to be handled in later parts.

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

Each part of the project had its own challenges. Some of these problems could be anticipated, but others appeared without warning. We have learned quite a bit from these challenges. First, a design consideration made in the beginning can and will affect design considerations later. The ISA should be kept simple to avoid some of these problems. If time became an absolute issue (which it came very close) we would have implemented a single cycle datapath rather than multicycle. We would have adopted the convention of placing the destination register bits in the last register location of the instruction format. This would have reduced the need of extra MUXs and extra control signals. In addition, we would have started the project earlier.

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

First, a design consideration made in the beginning can and will affect design considerations later. If there are problems with that decision, the problem will propagate to later portions of the project. Keep the ISA simple and start designing it early. It will determine more than you think. Some things it will affect are: number of MUXs in your datapath, component implementations, number of registers you can use. Keep your VHDL code neat, it will be beneficial when you have to debug everything. Keep in mind that every part is connected to all the previous parts. Do not over complicate something that can be made simple. The two most important pieces of advice would be: start early and make your control unit easily debuggable. Most solutions to problems in the datapath will require modifying the control unit. If it is difficult to determine which control signals are being sent to the datapath, then it will also be difficult to debug the datapath.