**James Sentell**

**Project Part 6 – Hardware Implementation and a Working Processor Demo**

Due to time constraints and workloads imposed by other classes, I have elected to complete the report portion of Part 6 of the project only. As demonstrated in previous reports, the CPU was fully functional in simulation. Thus, this report contains the answers to the three required questions:

1. I learned a great deal from this project. Coming in to this class, I did not understand how a processor actually executed instructions. Other classes had given me a somewhat fuzzy picture of some aspects of a processor, but I did not feel that I could trace the execution flow of a program down into the CPU level. After completing this project, I now understand how a processor executes instructions. I can walk through the block-level diagram of a (albeit simple) processor and understand every step. I feel that this project was a good capstone of sorts – enabling me to now gain a very basic understanding of the entirety of a computer. This project also served as an excellent refresher for VHDL modeling. The only prior experience I had gained with VHDL was in the ELEC 4200 Digital System Design course. This project was an excellent opportunity to revisit my VHDL programming abilities and further hone my skills in that area. Another lesson I learned was the importance of the instruction set to hardware organization. A well-designed, well-planned, consistent instruction set encoding scheme can result in a much simpler and more efficient hardware design.
2. If I had the opportunity to start the project over from scratch, I would spend much more time in the instruction set design phase. While my CPU was fully functional, I felt that it became rather “messy” due to the number of multiplexers required. Several of the instructions included in my final instruction set were late additions, and this did not have the same level of thought and care in their design that the majority of my instructions did. This resulted in many multiplexers having to be added all over my design just to support a relatively small number of instructions. As previously stated in my answer to question 1, a well-designed instruction set often results in a much better-designed and more efficient hardware design.
3. My advice to someone working on a similar project would be essentially the same as my answer to question 2: take great care designing the instruction set. Make sure that you fully understand all that is required by **all** the different types of instructions before starting on a hardware design. Additionally, it would probably be wise to go through multiple design iterations. Often times, a “first-draft” of a design can be greatly improved with multiple iterations of re-work. At times, this project seemed a representation of the classic “chicken and egg” problem. One has to fully understand a CPU to design a good CPU, yet one also has to design a CPU to fully understand a CPU. Thus, initial design efforts should be directed toward developing a full understanding of the system that you are attempting to design. Successive iterations can be focused on “good design principles” and efficiency refinements.