Travis Keller
ELEC 5200
CPU Design Project Part 6

This CPU Design project taught me a number of things and gave me valuable practice in skills I don't often get to exercise, primarily patience. Prior to this project, I had experiece with HDL languages, but was not nearly as comfortable as I am now with Vivado Design Suite and writing testbenches. This project solidified and improved those knowledge sets. If given another design projct in the future to complement this one, it would certainly prove to be much easier to accomplish. Additionally, I also now have a much greater understanding of how elementary CPUs work and the effort required to get one to execute a simple instruction set in HDLs.

Were I to change anything the next time around, I, like many students who enroll in this course, would start my projects a little earlier. This, however, would have been much easier said than done since much of my time was competing assignments in other courses. I made the most substantial progress on my design over the Thanksgiving break when I could finally take the time to focus on it. If I could do it over again, I might have started on the project and all its parts from the beginning of the semester since they were all provided at that time. Underestimating the time required to complete the project was my biggest mistake as it often takes me two or three times as long to complete an assignment when compared with my traditional peers.

Furthermore, I would have likely gone with a pipelined design so that all parts of my CPU would be driven by clock edges. This would have simplified timing and propagation delay. When I constructed my original model however, I had no knowledge of multi-cycle or pipelining, so it would have been a tall order to design anythying other than a single-cycle datapath and control unit.

Were I to offer advice to someone else about the project, I would of course say start as early as possible. That advice will likely fall on deaf ears because, well, we are undergraduate college students. I would also recommend using clock edges to trigger as many register changes as possible because then register transfers are easy to predict and observe while single-stepping through the project. Much of my confusion with the project came from debugging and single-stepping through it, and not easily comprehending what was going on in the project. I would tell other students that you may think that the programming is the bulk of the project, but the reality is the programming is only a minor part wheras the majority of the time spent on it is consumed by trying to comprehend what, why, and when things are happening.