What did you learn from this project?

This project was very helpful in showing how everything works together in a CPU. When designing the CPU from scratch I spent a lot of time on how the individual pieces should function and where they should be in the data path. Implementing them wasn’t to big of an issue, but when it came time to create the top level diagram and bring everything together, that was an entirely new experience. Debugging the entire model is quite the chore even when the individual pieces work on their own. Through the debugging process I learned a lot about timing and how it plays a large role on how certain elements take in values at certain times and parts like the PC have the ability to be really complex when it seems so simple.

This project also taught me the importance of choosing the instruction set and data path. When designing my instruction set, I included a jump and link function, but I failed to include a way to pull the item back out of memory and into the PC. Thus this opcode’s ability to save the PC values is effectively useless. The last bit of knowledge that I gained from this project is the importance of time management. The first couple of projects was not too big of an issue when it came to arranging my time because the parts were more theoretical and the semester load was lite. When it came time to create the top level model and bring everything together, a lot more time was needed that was not originally allocated. I am grateful for the extensions that Dr. Agrawal offered throughout the project as they allowed me to work on this project on my own leisure and prevent me from turning in incomleted parts.

What would you do differently next time?

If I were to do this project again, I would probably have done something more complicated. I chose to do the single cycle implementation because I knew that I would not have too much trouble implementing and debugging it compared to a multicycle or pipeline implementation. This was a wise decision based on the time constraint I had this semester working on it, but I would still enjoy the challenge that a more complex data path would have offered.

For this project I also should have really thought my instruction set out more. As I mentioned before I chose to implement a jump and link instruction, but did not include the ability to get the address out of the register and into the PC. I should have seen this well before I started implementing by thinking harder about all of the possibilities that the instruction set had offered. More research could have been done in this field before beginning.

The last thing that I would have done differently would be to have started this project in Quartus rather than Project Navigator. By choosing to start in Quartus, I would have saved a lot of headache and time later on in the semester when it came time to implement the design on the board.

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

My advice to anyone starting this project is to start early and work on it piece by piece. Working on it incrementally will allow anyone to see simple mistakes in data path and instruction set early on. That being said, I would advise future students to spend a lot of time on the datapath and instruction set to prevent headaches and last minute fixes being applied toward the end of the project when it is more difficult to add extra functionality. Another piece of advice would be to not be afraid of asking the TA questions. George in our case was more than helpful in explaining timing issues and possible solutions to problems that would have taken a long time on my own to come up with. Lastly would be not to stress over it. It is a fun project in a class that is very complicated. If things do not make sense the first time you encounter them then take a step back and ask for help or explanation if needed. It is nothing to stress or worry over.