**Project Part 6**

Alex Dyke

Robert Skelton

1. What did you learn from this project?

Throughout the entirety of building a 16-bit MIPS microprocessor in VHDL, we learned more about how the different parts of a computer system work and how they interact. Our VHDL skills were much sharper afterwards, and we learned how to notice simple syntax errors quickly.

1. What would you do differently next time?

If we were to do this project again, we would definitely start earlier so that we didn’t fall behind. We would still stick with the single-cycle datapath for it’s ease of implementation, and might have tried to find out how to use the Altera Megafunctions. The Altera Megafunction Plugin Wizard was not included on any computers in Broun 310, so we chose to create our own data and instruction memories and attach them manually.

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

Start early! Also pay attention in Computer Systems in Sophomore year, most of your basics for this class will be covered there and it is crucial to understand them. Assembly is only covered in that course, so you will want a good foundation in assembly to be able to do well in Computer Architecture. Review old projects from Digital System Design (ELEC 4200) to get you started in this semester. Make your design as simple as possible, because it is hard to debug a complicated system. Draw your architecture on a piece of paper and label all signals clearly so that you’re staying consistent. When you have to edit large chunks of your code, do not use the Xilinix IDE, open your code in Notepadd+ and make your edits, then move it back over. This saved us lots of time by using the find and replace and multi-line edit features that Xilinix does not have.