Michael Marchek & Johnmark Norris

4/27/12

COMP ARCH

Project Final Report **Design of a RISC CPU**

1. What did you learn from this project?

We practiced and improved our VHDL skills, learned how to use the Quartus II IDE, and became very adept at using the ModelSim simulation environment.

We learned a lot about processor architecture, and how a processer actually functions. It was extremely interesting to actually attempt to implement one in hardware, even if our design did not fully perform as expected. It was also intriguing to compare notes with other groups to see what they were doing and how different it was from our design.

The engineering process was very important in this project, due to the amount of work required to finish it and the timescale the project had to be completed over. It was essential to practice basic engineering tenants to ensure that our project stayed on schedule and satisfied our temporary goals. First we had to identify the problem at hand, be that correcting an error, or even choosing out an instruction set for our design. Once we know the problem we needed to figure out the constraints of the problem, often times a method of handling an issue might create others, careful consideration of modifications was essential as they caused a domino effect that would cripple previously functional components. This was part of the reason our design failed to perform as expected in the end, we failed to carefully consider the implications of doubling up our clock cycle to accommodate the memory’s port and clocking requirements. Generating ideas sometimes required stepping away from the design for a while, or even a night. Once we’d considered our ideas we’d chose a plan, and attempt to execute it in our design. Usually while in simulation the testing of a part would leave behind test signals, or other remnants and left overs, and so the final step of our process was to attempt to clean up our codes, test programs, do files and projects files.

1. What would you do differently next time?

We would have settled for 2 memories, instead of attempting to improve our design by having one memory. By doing so we caused a lot more problems than having 2 memories would have caused in the end. Paying attention to requirements is very important. The Harvard architecture would have satisfied the constraints of the project while avoiding many of the timing and clocking errors that we encountered. In short we hurt ourselves by attempting to be too clever and combining single and multi-cycle components theories and setups.

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

Consider carefully what type of design you want to implement. Specifically, what are the requirements for your assignment? We included functions that were completely unnecessary to the design, such as jump return. Carefully consider what devices require clocking and do not require clocking and whether they operate synchronously or asynchronously. Last minute fixes to a design can have unexpected consequences and devastating. Lastly, when testing, pay more attention to the signals (control signals) than the results. Even if the results are correct the signals may not be correct. Ignoring faulty commands and attempting to charge ahead never pays off in the long run. You only end up more confused when you have to go back and edit at a later date. This above all else was an extremely informative learning experience.