**Harshit Goyal and Tausif Muzaffar**

**ELEC 5220**

**27 April 2012**

**What did we learn from this project?**

This project showed up how a CPU actually functions. It makes sense that things are stored in memory and it’s up to central components to retrieve and use these instructions, but this is the first time we actually had to do it ourselves. Now we had a greater appreciation and understanding to how to create more efficient code. With this project the concept that multi-cycle allows for a fast running of application is the code written does not create a large amount of hazards, because if there are lot of hazards the error checking that the system has to do takes away from the speed. To that same point we now understand why using a lot of jumps in a program is a poor style of programming, because it is terribly inefficient for a speed of a program.

Aside the concepts of a CPU we also learn some more about FPGAs. Despite already studying in ELEC 4200 using the Altera software compare to the Spartan III is much different. While the board is more robust, software is straighter; the model simulation seems to be more difficult on the Altera board compared to the Spartan. So we learned what are preferences would be when working with FPGAs but we also learned how to work with different software which can be useful in different environments.

**What we do differently next time?**

If I were to choose something to do differently, I would try to use single cycle rather than multi-cycle. Because of the way we setup our code we could not run our clock on a switch, which means when we ran the program it was hard to debug what we had on the board or if our command were working the way we intended them to work.

Also we would create a better hazard detection system which could help our system to stop our program from crashing. This could have helped because our program would crash constantly and since we could debug on the board well it would be best to just cut it out.

**What would we recommend?**

Work on this early and constantly. The program may simulate differently than it runs on the board so make sure you continue to work through all the programs that you have. Also use single cycle, although you may have clocking issues because it is one cycle, we couldn’t implement a push button clock and it could have been really useful.