ELEC 5200 Project

Final Report

Andrew Tomlinson

Zane Franklin

## What did you learn from this project?

In designing a CPU from the ground up, we learned about the operation of the individual components as well as the intricate connections between these components. Additionally, we gained a deep appreciation for well formed VHDL code and descriptive signal and component names. Being one of the largest VHDL projects we have ever constructed, the use of good coding conventions was essential to keep track of individual signals while maintaining a grasp on the big picture.

In order to design and build a processor from start to finish, as well as design each individual component for the data-path, we had to completely understand the function of each part and control the data flow in order to realize the correct function for every instruction in our ISA. It was truly challenging, and we made many mistakes, but together we were able to solve the problems that arose and implement a completely functional CPU.

## What would you do differently next time?

We were on multiple occasions frustrated with having to rely on “pre-made” components for use in our data-path when these components did not have the exact functionality that we needed. For example, we designed a single cycle data-path with the assumption that our memory would have asynchronous read capability. However, the memory which was provided to us by Altera was designed to be synchronous read only. This memory had many other useful features such as on-board re-programmability, so we wanted to use it when putting our CPU design on the FPGA. We solved the issue by creating a two cycle data-path. After doing this we realized that creating a full-blown multi-cycle data-path from the start would have cut down on synchronization issues and data race conditions we experienced throughout our design process.

Additionally, due to the fact that we were required to address a memory with 1k words (which takes 10 bits) and we only were able to use a 16 bit instruction, our ISA was somewhat limited. We did not end up using nearly this much memory, and it would have been better to require the addressing of a smaller memory so more instructions and functionality might be added to the ISA.

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

Don’t be afraid to change. We hit many snags while trying to implement the processor that we originally designed with the ISA that we originally agreed upon. During that time, instructions were thrown out, while other instructions were added. Control signals were added, and sometimes even whole registers were inserted into the data-path that were overlooked in the first design. Multi cycle and pipelined data-paths might seem terrifying at the beginning of this course, but try to design your CPU around one of those architectures. It will be rewarding.