# Computer Architecture Project – Part 6

# Hardware Implementation

# Sean Tyler Atwell

## What did I learn from this project

First and foremost, I learned how a processor works. I failed to gain a full understanding of how all the components worked together to process instructions, but putting together an actual system really brought everything home for me.

I also gained a newfound respect for developing Instruction Sets. I gave myself an untold amount of trouble by poorly defining my ISA; largely because I thought I could define things fairly arbitrarily without any repercussions.

## What would I do differently next time

I would completely revamp my ISA. I did a lot of things in the name of “getting it to work” (e.g. I declared a part of my R type function argument as a global control signal) that just killed the efficiency of my machine. I would worry less about reducing my total number of opcodes (since I only ended up with 8 or so by cleverly using the function argument) and worry more about what each opcode would need to influence on the board.

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

One, if you declare a hard zero register in your register file, do not try to write things to it. You won’t get the output you expect and you will waste 2 hours trying to figure out what is going on.

Don’t collate opcodes for functions with intermediate arguments with opcodes for functions with register arguments.

Definitely use the Single Cycle datapath design. Anything more interesting gets out of hand really quickly.