

CPU Design Project Part 6 Report by William C. Hodges

What did you learn from this project?

This project gave me an improved understanding of the inner workings of a CPU. I had a small amount of exposure to CPU design in the past, but this project allowed me to really get my hands dirty with VHDL and take into account every little aspect of a basic CPU. The project requires you to be careful in your design all the way from the ISA to timing issues that you will inevitably face while testing the final implementation.

One of my bigger realizations while working on the project was how much of an impact the compiler can have on processor performance; not just in the sense that the compiler can optimize and improve performance, but also that it can cover up shortcomings or issues in the hardware. (Got some major timing issues? Throw some nops in there!) For instance, my ISA only supported immediate values between 0 and 15. This can be quite a problem when you need to, say, load the stack pointer register with the max memory address. With a clever compiler/assembler, this issue could be hidden from the high-level programmer entirely. The fact that I took an independent study on compilers this semester was doubly helpful in driving this point home. Taking both classes at the same time allowed me to get more out of each class than I would have had I taken them individually.

Perhaps the most useful thing you will learn during the course of the project is how all of the little components of the CPU can work together to produce various effects and implement various instructions, such as loading a word from memory. Actions that the programmer tends to take for granted can require a fair amount of complexity from the perspective of the hardware designer.

What would you do differently next time?

I would probably go with a pipelining datapath next time. It seems like more work, but I think it really pays off, especially when you consider that practically all modern processors use pipelining. If you want to really understand how they work, trying to implement a pipelining architecture yourself couldn't hurt. The only reason I didn't do it is because I was drowning in work all semester...

What is your advice to someone working on a similar project?

My advice would be to do some serious reading on the subject... if you have time. It's a terribly interesting field, but those books can also eat up all of your spare time. The class textbook is something of a must-read in the field (*Computer Organization & Design: The Hardware/Software Interface*), but my personal recommendation would probably be *Structured Computer Organization* by Andrew Tanenbaum. I've read a couple of other books by the author, and he has a definite talent for introducing readers to new subjects. He also has a good, dry sense of humor that will keep you from bashing your brains out with the book. We wouldn't want that.