Introduction to Digital Logic and CPU Design

Nicholas Miehlbradt

August 26, 2022

Quick Recap

- Finished our register file
- Started fitting everything together
- Included some memory to store our program and more data
- Made a program counter circuit to keep track of where the CPU is up to in the program
- Identified our control lines

If you were not here last week then I've uploaded my work to GitHub.

The missing parts

There are two[1] key parts that are missing before we can run a program.
What are they?

[1] Also a bunch of 'glue' logic to interconnect the parts.

Detecting zero

One thing we also need our CPU to keep track of is whether the last result our ALU produced was zero. This is the **Zero flag**. We can use this to conditionally jump.

A bit of glue logic

We also need a few more ways to control our CPU.

We need to be able to put a value into a register, no ALU or memory involved.

We need to be able to add an offset to our PC value

Decoding logic

Now we need a way to decode our instructions into control lines that tell each individual part of our CPU what to do.
Fortunately this is all combinational logic, we can go line by line and figure it out.
The truth tables for these can be bigger than its worth, there are many patterns that we can exploit to minimise our work.

A bit more glue

Now we just need to hook everything together.

You can use tunnels if you think its easier, just remember to rename them all correctly.

Lets write a program!

We're going to write a program in assembly and then hand assemble it into machine language.

Then we can put it into our CPU's memory and run it.

All Done!

Congratulations! You've implemented your own CPU.

Where to go from here

There are lots of things we didn't have time to cover

- Subtraction and how computers represent negative numbers
- More ALU operations
- Register-Immediate operations
- More flags, more jump conditions
- Function call instructions

All these things can be implemented while maintaining backwards compatibility.

More advanced things

- How CPUs go fast
  - Pipelining
  - Out of order
  - Branch prediction and speculation
  - SIMD
  - Multi-core
- Memory systems
  - Caches
  - Virtual memory
- Interrupts
- IO

Useful Links

- Digital (Logic Simulator)
- Digital Design and Computer Architecture (Textbook)