

CS120 Fall 2018 Lab 5

In this lab, we'll write some machine code, and use a tiny processor in Logisim. You can grab the processor from the course web page.

The processor has four registers, and can do four math operations. Instructions are formatted as <Destination> <Source 1> <Source 2> <Operation>.

For example RegisterD = RegisterA + RegisterB maps to

11 (register D)

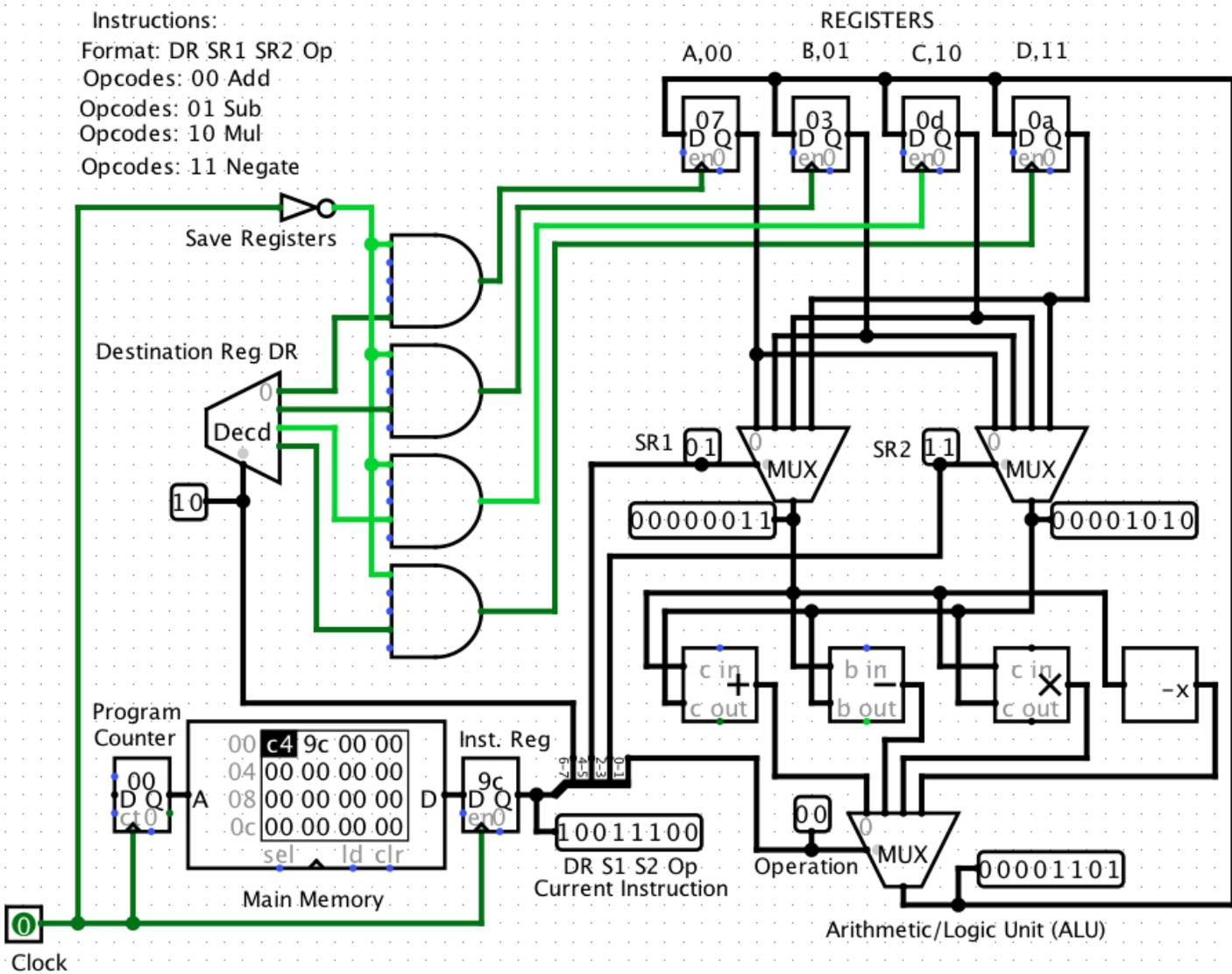
00 (register A)

01 (register B)

00 (Addition operation)

The operations are Add (00), Subtract (01), Multiply (10), and Negate (11)

The entire instruction is 11000100 (binary), or C8 (hex)



The processor operates by using the program counter to send an instruction from main memory to the instruction register, when the clock goes from 0 to 1. That causes the MUXes and decoder to send numbers through the math units, and back towards the registers. When the clock goes from 1 to 0, the result is stored back in the register. The program counter increments automatically.

You can insert numbers into memory and the registers.

For the lab... Convert the following program into machine code, load it into memory, and then set register A to 3, and register B to 4. Single step through the program; make sure that the right number ends up in register D.

$C = A + B$

$D = C + B$

$D = D * D$

If everything works out, you should have 121 (decimal) in register D....

Next — move the program counter back to zero, try things again with a few other numbers.

The next program: figure out the best way to use the registers, and the sequence of instructions, to compute

$D = (A * ((2 * B) + A)) - B * B$

You might need to store some temporary values in register C. If you can avoid changing the values in registers A and B, that would be better.

The C compiler normally does this work for you, figuring out which registers to use, how to sequence the individual instructions, and a bunch of other stuff.