

Project 1

Simulator

April Gross, and Nick Scheel

October 20, 2017

1 Simulator

1. Set up
 - (a) Check for correct number of arguments from command line
 - (b) Initialize struct with registers 0 through 7 initialized to 0
 - (c) Initialize memory to 65536
 - (d) Initialize program counter (PC) to 0
 - (e) Loop through file and store each line in memory
2. Looping through lines in memory starting at 0
 - (a) Print state
 - (b) PC++
 - (c) Extract opcode from instruction
 - (d) For instruction type assign appropriate labels to appropriate section of bits
 - (e) For each opcode, do operation
 - i. Assign registers values as needed
 - (f) “halt” is reached, print state and exit

2 Multiplication

1. Store values in registers
2. Loop through 14 times
 - (a) Check if the specific bit of the multiplier is a 1 or a 0
 - (b) If a 1, add the mcand to the running total
 - (c) Shift the mcand over the the left one
 - (d) Reset registers if needed

3 Journey

The simulator was fairly easy to code after we learned that we weren't duplicating the circuit. The main trouble we had was with the offset. It took a while for us to figuring out how to tell if it was negative. Once that was solved, everything fell into place.

The multiplier took a lot of thinking. We figured out how to get the mcand shifted to add it to the running total. The issue was determining if the bit we are looking at in the mplier was a one or a zero. We landed on a solution that used a mask and compared the result it to all ones. If it was all ones the bit was a zero if not, it was a one. The mask also gave us issues, we originally calculated the number of bits incorrectly. We then realized we could nand zero with itself to get the appropriate number of ones in all the bits.