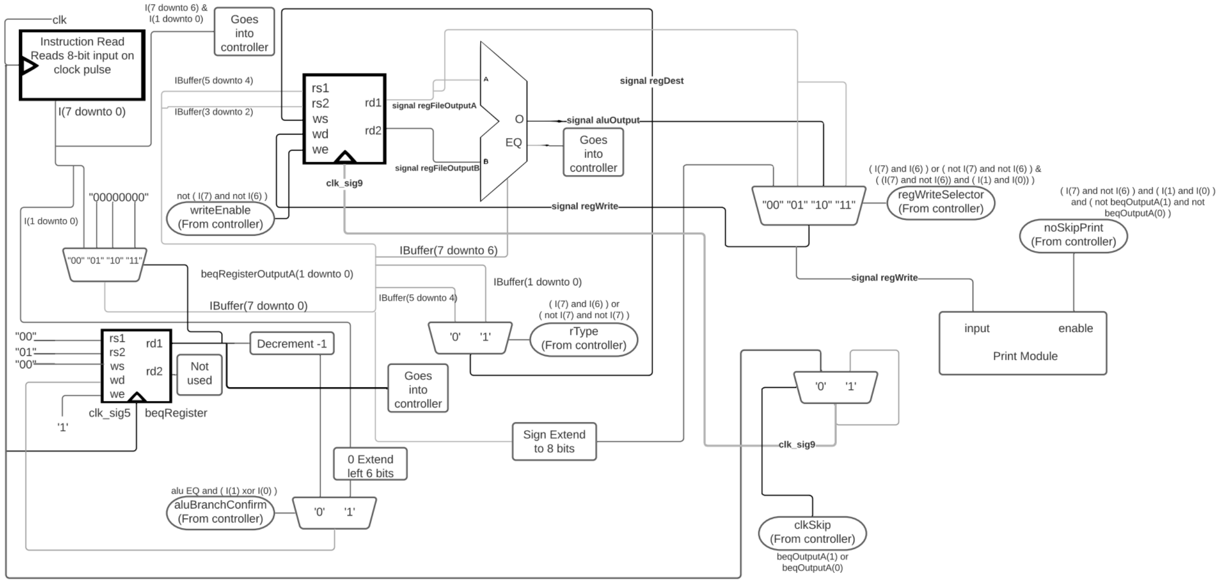

Above is pictured our circuit diagram where we started with an instruction read that reads the binary values from a text file and passes values through the circuit in order to complete the objective. We start by reading the op code to tell what we have to do and select the correct mux's accordingly. When the correct control's are set we move to the register file that fetches the data from the input registers based on the 8 bit code. Then these values are passed to the ALU which decides what to do with that data and whether or not to write data back. The alu can take either branch input or directly from the regFile. The cycle will complete if the regFile decides to write it back and it will start at the next cycle. If the print module is invoked it will print the value of the register you enter as input. There is a second registerFile that holds the number of instructions (with each instruction corresponding to a clock cycle) to skip in the output rd1. The output of this registerFile decrements each clock cycle until it reaches zero. The controller takes in the opcode, the immediate field, the "eq" output of the alu, and the beq registerFile rd2 output to determine the mux selectors.