

Assignment 1 Dram and registers

For this assignment I worked alone.

Design choices for registers and dram

- 1) For my general purpose register I had four d flipflops with a two selector bit multiplexor taking in a write, an input, and bit values 0 & 1. This would all me to be able have my write input node be able to store, reset, and load based on the input fed into it
- 2) For the program counter it uses four d flip flops, a control, and a multiplexor taking inputs from the control with constants 1 and 0, each time the clock iterates while the control is set to 1, the output increments, when it is set to 2 the output is reset, and at 3 the output is fully loaded
- 3) For my memory, I used four of my general purpose registers with a 2 selector bit demultiplexer taking an address bit. And the four general purpose registers are being fed a clock, an input, and a control bit that stores, resets, and loads
- 4) the instruction register just takes inputs from memory and feeds into a general purpose register, producing the final output
- 5) In the memory registers dram circuit, the input first begins with the program counter where the output is fed through a splitter so that the bits could match when it goes into the multiplexor. The multiplexor also takes an address input and its output goes into the address for memory which takes in another input. The output of memory is then fed into the instruction register which outputs based on what the control is set to

Testing screenshots

- 1) The first screenshot shows the general purpose register taking in an input of 5 and outputting that value
- 2) The second screenshot shows the program counter having incremented to 14 (E)
- 3) The third screenshot shows the program counter loading a max value, that being 15
- 4) The fourth screenshot shows it being reset to 0
- 5) The fifth and sixth screenshot is of the instruction register outputting different instruction values
- 6) The seventh screenshot is of the dram with everything set before the control for the instruction register allows it to output the final value
- 7) The eighth screenshot is of the instruction register outputting the value after the control was set to let it load







