Simulation Description

* How did you implement?
  + For each of the bitwise operators, we used the analogous bitwise operator in Verilog. For NOR, we negated the OR output.
  + For the ADD, we created one-bit full adders and chained them together to create a ripple-carry adder.
  + For the mux, we had nested ternary operations that called the appropriate module depending on the given op\_code.
* How did you test it?
  + For each of the bitwise operators, we set X, Y to all permutations of 0x00000000 and 0xFFFFFFFF. This allowed us to test the operation of each bit and make sure our bitwise operators worked correctly.
  + To test the EQUAL and ZERO flags, for every test that we did, we checked if X = Y or if Z = 0 and made sure that the appropriate flags were set.
  + To test ADD, we wrote a full adder test that we did not include in test\_alu. In test\_alu, we created a test to make sure that the full adders were hooked up correctly by bit-shifting X and Y, adding them, and verifying that next significant bit changed correctly due to the carry bit.
  + We did not test the mux.