Project 2
Single Cycle Processor

Jonathan Egbert

Work Division:

I worked on this project with Adam Yost. He wrote and debugged the assembler on his own. I did most of the verilog code for the process but he helped with several components and fixed many bugs.

Approach:

The way we approached this project is by breaking each component of our single cycle processor into different modules that took all of the output from other components that it needed as input. Each of these modules would handle the input it recieved by looking at the opcode passed from the controller. Our DMem is a slightly modified IntrMem. The DMem module takes in all the hardware references that the projects takes in so that it can map them to memory locations. For our ALU, we output a 3 bit value that can be checked to know if the last operation resulted in a positive, negative, or zero value. We use this in PC to determine if a branch should be taken. The PC module takes in all output that it needs from other modules to correctly increment the program counter depending on the current instruction.

Problems:

A did not run into many problems in the assignment other than trying to understand the assignment. Once we had a good understanding, most of the issues we ran into were forgetting to take into account certain instructions. This meant we had to visit “finished” modules multiple times to add functionality to handle them.