Nem Negash

Project 2 report

The logic for the state machine was derived from a state diagram. I figured out I need to have a total of twelve different states and four D-flip flops. Using the state diagram, I made a state table with each state’s next state and the output depending on the input. After having the next states, I assigned each index of the state to a flop. I was then able to use Karnaugh maps to figure out the Boolean equations for each flop and the two outputs. The equations were all sums of products, therefore it’s a group of “and” gates that are then all put as inputs into an “or” gate. For the Verilog code, I divided up the equation for each segment into the products and used “and” gates to get the output. This output was stored in wires and all the wires that held the outputs of the “and” gates for that flip-flop. These outputs were then put as inputs for an “or” gate which was stored in the final output for that flip-flop. This process was repeated for the remaining flops and the two outputs in the coin-count module. The rest of the Verilog code was provided by the professor.







