Jesse Anzelc

4/28/14

EEC 180B

Lab 3 Report

I decided not to implement my RAM as a state machine. It seemed more intuitive at the time to simply test the write enable and select bits and perform the appropriate function. I wanted to implement the RAM modules as verilog modules and instantiate them in a part1 module, but this caused me some problems so I combined everything into one module.

For the booth multiplier, I went straight to programming the FPGA. This made debugging more difficult, but I made it work by using red and green LEDs to keep track of the count and current state. My implementation uses two always blocks: one to set the Done flag (sensitive to count) and another to handle everything else (sensitive to clock and reset). A few things I learned in this lab were when to use nonblocking statements versus blocking and that reset is active low, two things that prevented me from finishing two days earlier.