Jaci Reichenberger, A20131719

Stephen Potter, A11341625

ECEN 4243: Computer Architecture

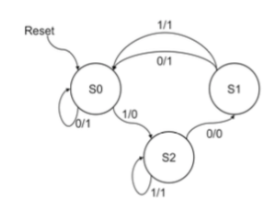
8 February 2021

Lab 1: Revisiting Digital Logic and Verilog Simulation

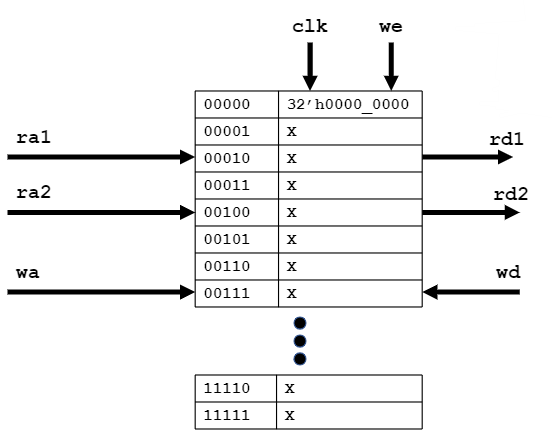
**Section 1: Introduction**

The purpose of this lab is to introduce the Verilog tools that will be utilized in this class. Using code already provided, this lab introduced how to use ModelSim to view the output waveforms of an FSM. Students were able to read through the code and learn how to utilize a DO file in order to see the waveform inputs and outputs in the application. The second part of this lab introduced how to write code, a testbench, and modify a DO file in ModelSim in order to show both the input and the output waveforms for a register. The code for the register was created following guidelines given for this lab. The testbench code was written to test the various guidelines given, and the DO file was only slightly changed from the FSM’s DO file. All in all, this lab introduced students to ModelSim and reminded students how to code in Verilog.

**Section 2: Baseline Design**

Part 1: FSM Simulation

The figure above is a basic Mealy Finite State Machine (FSM). The code was given for this part, but all the code did was explain what this diagram does. This FSM works with inputs and outputs. Based on whether the input is a one or a zero will determine if this FSM moves on to the next state or not, and its output depends on the input as well. For example, in the above FSM, if the input is a zero, then it will stay in S0 with an output of one. If the input is a one, however, it will move from S0 to S2 and output a zero. All this is seen using the given testbench and the DO file that output the various waveforms for the various states of this FSM.

Part 2: Register File and Simulation

The second part of the lab was to create the code, testbench, and DO file for a register. A file was given that contained the variables needed to write the code in order implement the register. For the inputs, there are two 5-bit source register numbers (ra1, ra2), one 5-bit destination register number (wa3), one 32-bit wide date port for writes (wd3), one write enable signal (we3), a clock (clk), and two 32-bit register values (rd1, rd2) for the outputs.

Baseline design and implementation. Figures (FSM). How figure works. (<1pg)

**Section 3: Design**

Overview of Design. Why you chose to implement that way

**Section 4: Testing Strategy**

Tb. What happened in tb. Table with test results. Explain why it proves design works

**Section 5: Evaluation**

Simulation results. Explain them. Big picture summary.