Digital Logic
CSC 244L
Laboratory 5
Vending Machine FSM 2: Electric Boogaloo

1 Objectives

1. Learn and gain experience using *procedural* SystemVerilog (SV);
2. Implement a finite state machine (FSM) directly from the state table;
3. Compare results derived from procedural SV to those derived via behavioral and structural SV; and
4. Explore difference between a Moore and Mealy FSM.

2 Pre-Laboratory Procedure

Illegible work and files that do not open in the D2L dropbox will result in reduced grades. It is your responsibility to ensure that what you turn in is readable by the TAs. Please read the submission instructions and follow them to ensure you receive all points. The circuit diagrams for pre-lab may be *neatly* hand drawn. To be completed *before* your lab meets (individually):

2.1 Read the entire lab procedure, and Chapters 4.4–4.6 in your book;

2.2 The following will not need to be turned in as pre-lab, but it is up to you to have these completed prior to your lab section (**otherwise you will not finish in time**). Complete these items (in this lab manual) for your pre-lab:

- 3.1.1, 3.1.2, 3.1.3

2.3 Bring soft copies of all your SV modules to lab to be compiled and loaded onto the FPGA for testing and demonstration.

By now you should have downloaded and installed Quartus Prime Lite. You should come to lab with **SystemVerilog modules that compile the very first time**. You will ensure that your files compile correctly by compiling them at home with Quartus. A proactive student would also test that the compiled SV works on their DE10-Lite board, which would make for quite a short lab period.

3 Laboratory Procedure

3.1 Part 2: Moore and Mealy Sequence Detector FSMs

The second part of the lab will have you implementing the sequence detector (sequence = '1010') FSM as both a *Moore* and *Mealy* machine using *procedural* SV. You will compare the operation of the Moore and Mealy machine, as shown in Fig. 1. Use a *debounced* push-button as your FSM clock.

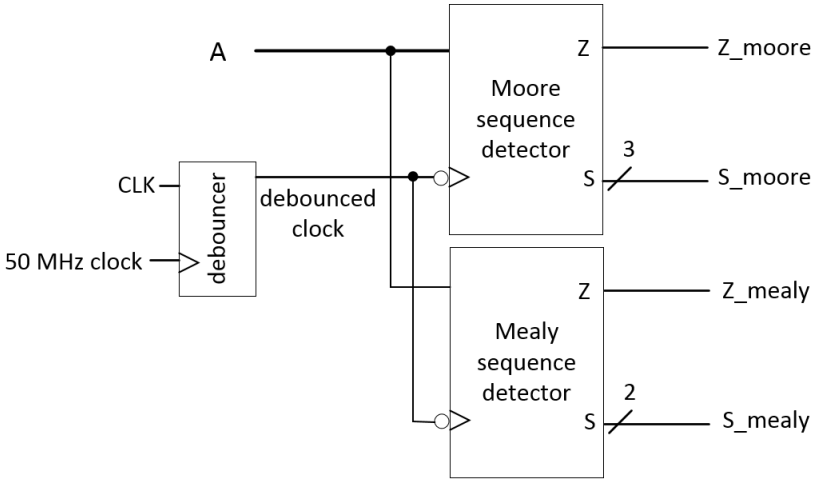

Figure 1: A `1010` sequence detector FSM implemented as both a Moore and Mealy style machine.

3.1.1 Create a SV file named "sequenceDetectorMoore.sv" that contains one SV `module` named sequenceDetectorMoore. Write ***procedural*** SV to describe the operation of the Moore-style `1010` sequence detector from class (**You must use parameters/enums to label the state inputs**).

3.1.2 Create a SV file named "sequenceDetectorMealy.sv" that contains one SV `module` named sequenceDetectorMealy. Write ***procedural*** SV to describe the operation of the Mealy-style `1010` sequence detector from class (**You must use parameters/enums to label the state inputs**).

3.1.3 Create a SV file named "machineComparison.sv" that contains one SV `module` named machineComparison. Write ***structural*** SV to combine the two FSM modules together as shown in Fig. 1.

3.1.4 **Optional, but recommended:** Compile the Moore FSM using Quartus Prime Light. Assign the *Moore* inputs and outputs to the DE10-Lite I/O using the values in Table 1.

3.1.5 **Optional, but recommended:** Load just the Moore sequence detector FSM to your FPGA. Verify the operation of your FSM by checking enough input-state combinations and checking the output.

3.1.6 **Optional, but recommended:** Compile the Mealy FSM using Quartus Prime Light. Assign the *Mealy* inputs and outputs to the DE10-Lite I/O using the values in Table 1.

3.1.7 **Optional, but recommended:** Load just the Mealy sequence detector FSM to your FPGA. Verify the operation of your FSM by checking enough input-state combinations and checking the output. **What happens to the output if you change the input during the 'Z=1' transition between clock pulses?**

3.1.8 Compile the machineCompare module using Quartus Prime Light. Assign the inputs and outputs to the DE10-Lite I/O using the values in Table 1.

3.1.9 Load the machineCompare module to your FPGA. Verify the operation and compare the differences between Moore and Mealy FSMs of your FSM by checking enough input-state combinations and checking the output. **Pay special attention to the output state, and what happens if you change the input during this transition between the two machines!** Show your working FPGA module to your TA, including the asynchronous behavior of the Mealy machine, and have them sign off on your lab sheet.

Table 1: Required DE10-Lite Input/Output

Signal	**DE10-Lite I/O**
CLK Input	KEY0
A Input	SW[0]
Mealy Z Output	LEDR[0]
Mealy State Outputs	LEDR[3:2]
Moore Z Output	LEDR[9]
Moore State Outputs	LEDR[7:5]

4 Report

4.1 This lab is demonstration only, no report is required to be submitted.

Laboratory 5 Signoff Sheet

Student Name (Print in Black Ink):

TA Signature	**Section**
	3.1.9

Student Signature: Date: