NC State University
Department of Electrical and Computer Engineering

ECE 310 – Fall 2025

Lab 5

Summary

In this lab, you will create a Moore FSM model for a vending machine.

Deliverables

1. Code for model and testbench as separate .v files uploaded to Moodle.
2. Short report describing the theory of operation, design/diagrams, snippets of code as needed, results, and a discussion. Include a screenshot of simulation waveforms for the design with some representative values (not trivial values). Justify why you used these values. Upload the report to Gradescope.
3. Extra Credit – up to 10 additional points will be added for typesetting your report in LaTeX, depending on usage of LaTeX elements (e.g. figures, tables, sections, etc).

Instructions

1. Create a new project in Vivado named **vending_25c_moore**.

2. Design a Moore FSM that implements a candy vending machine where each candy costs 25 cents. The vending machine only accepts dimes (input/port **D**, 10 cents) and quarters (input/port **Q**, 25 cents), only one coin at a time. The input format is DQ (this helps in using binary values in the state diagram). If both a D and a Q are entered (“11”), the state machine does not advance, that is, it stays in the same state. Same for when no D or Q are entered (“00”). Once 25 cents or more is entered, the vending machine will output a product (binary output/port **P**) and wait in that state for a new user input (i.e. it does not unconditionally reset unless the input is “00” or “11”). If the user enters more than 25 cents total, the vending machine will return some change (binary output/port **C**). Output C will be raised High (‘1’) regardless of how much change is required (assume some other FSM deals with the return of the change). Use state names and assignments following the usual sequences (i.e. S0, S1, etc, and 00…00, 00..01, etc, respectively). Use the minimum number of bits required for state assignments, any unused state assignments will have the next state and outputs set to zeros. States are a function of the total amount entered so far and are in order of the next smallest increment in total sum (e.g. if total sum is 20 cents, the next state is 25 cents, not 30 cents). Obtain the list of inputs and outputs, state encoding table, state transition diagram, state transition table, and output table for this FSM. You do not need to extract equations to implement the FSM in Verilog.

3. Implement the FSM above in Verilog in a module named **vending_25c_moore** with the above port names. Additionally, add a clock signal (**clock**) and an active-low synchronous reset signal (**reset_n**). The reset signal will be used to reset the FSM to S0 at the beginning of simulation. Implement a testbench to test your model with at least 2 input vectors: (a) best-case scenario,

and (b) a sequence of inputs that do not give the best-case scenario. Submit a screenshot of your simulation results showing the 2 patterns above and a description of your test inputs used with justification/explanation.

Grading Breakdown:

- Pen-and-paper design of the Moore FSM with all details listed in Step 2. – 25 pts
- Design and implementation of the Moore FSM in Verilog – 35 pts
- Testbench implementation – 10 pts
- Justification for at least 2 input test vectors used – 5 pts
- Lab report with as much information present as possible, see Deliverables section. A state transition diagram and descriptions for the diagram is required. – 20 pts
- Upload of .v model file and testbench (2 separate files) to Moodle and a .pdf report to Gradescope – 5 pts