

CSE 140L Hardware Lab Three Winter 2006

Thomas Y. Lee

Finite State Machine

Due time: March 1, 2006

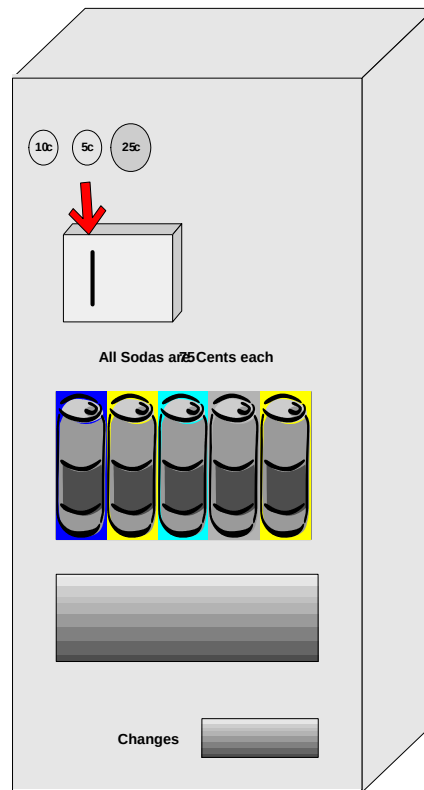
Objective

In this lab, we will design a Vending Machine. We first use the state diagram to specify the system as a finite state machine. We then use either VHDL or Verilog for the implementation. We will try different state assignments, use Mealy and Moore machines, and compare the results.

Note that in this lab, we use Altera FPGA Family: Stratix, Device Type: EP1S, Package Type: 144 pin TQFP.

Part 1. Vending Machine

You are request to design a Vending Machine to be used in UCSD campus. In this machine, user can insert coins, which can be any combination of a quarter, a dime, or a nickel. There are three types of Soda to choose from: (1) Coke (2) Diet Coke (3) Pepsi. The drinks are 60 cents each. After user inserts the enough money for drinks, the machine delivers the selected drinks and makes changes; the changes are in coins only. You must design the system to limit to only one type of money input per clock. In the following state machine simulation, you can simulate the clock in 100ns period, but in reality, the clock may run as high as 10 seconds. To simulate the actual output, you need to put in three quarters to purchase a Coke.



Part 2. Mealy Machine

- i. Implement the Vending Machine as a Mealy machine. Try two different codings (Binary Coding and One-Hot Coding) for state assignments. (hint: Use a state diagram to specify the machine. Convert the specification to a VHDL or Verilog program. Edit the program and synthesize the machine as a macro cell.)
- ii. Display the timing diagrams of the three implementations. (hint: Initialize the state to start the simulation)
- iii. Change the input sequence to demonstrate the glitch of the Mealy machine. (5 percent bonus if you show how to eliminate the identified glitch)

Part 3. Moore Machine

- i. Implement the Vending Machine as a Moore machine. Use two different codings for state assignments.
- ii. Given the same input sequence as in Part 1.ii, display the timing diagrams of the three implementations.
- iii. Demonstrate an input sequence (if any) that causes a glitch in the Moore machine.

Part 4. Comparison

Use a table to compare the four different implementations (two different state assignments for both Mealy and Moore machines), in terms of the number of states,

number of flip-flops, number of lines of VHDL or Verilog codes, and the number of Logic Array Blocks (LABs: boxes in layout).

Report

Title page:

- Names of students, **student ID** and due date.
- Title of the lab and objective.
- A brief description of each person's contribution.

Contents:

Part 1: A summary for the design and implementation process in the lab with less than 80 words.

Part 2: i. The state diagram, two VHDL or Verilog programs, and two FPGA layout figures in timing closure floorplan. ii. Two timing simulation diagrams with worst delay marked. iii. A timing simulation diagram with glitch, and a short explanation why the glitch happens.

Part 3: i. The state diagram, two VHDL or Verilog programs, and two FPGA layout figures in timing closure floorplan. ii. Two timing simulation diagrams with worst delay marked. iii. A timing simulation diagram with glitch if you can find one, or an explanation why there is no glitch.

Part 4: The comparison table.

Grading

90% will be based on the completeness and correctness of the report.

10% will be based on the neatness, organization, and following instruction of the report format.

You will get 5 percent bonus if you show how to eliminate the glitch you identified for the mealy machine in Part 2.