

**ECE 368—Fall 2006, Instructor: Prof. Shantanu Dutt****Lab 5 : Due Thurs., March 19****Purpose**

The purpose of this lab is to: (1) familiarize yourself with VHDL behavioral descriptions and especially such descriptions of sequential circuits (i.e., finite state machines or FSMs); (2) Describe a clocked system, i.e., a circuit that operate in synchrony with a clock. You will achieve these by behaviorally describing an FSM that is triggered on the positive edge of a clock.

**Design Problem**

You need to design a Moore FSM for a vending machine with the following specifications:

1. It dispenses two items: gum and chips.
2. Gum costs 20 cents, and a packet of chips costs 40 cents.
3. Quarters (Q), dimes (D) and nickels (N) can be inserted into the vending machine.
4. The customer also presses either the gum (G) or chips (C) button to indicate his/her choice. The choice button is to be pressed after the needed coins are inserted.
5. The machine needs to return change, and you need to determine the different change amount the machine may need to return based on the coins inserted and the choice button pressed. At the same time, allowance has to be made for assuming “unreasonable” behavior by the customer in terms of coin insertion. This is to be done by returning the last coin inserted by the customer, if the the amount of money inserted before the last coin is  $\geq$  the price of the most expensive item (in our case, 40 cents). Note that the “last-coin-return” strategy is independent of the choice made (in fact, it should occur before the choice is made).

For example, if the customer inserts the following sequence of coins: Q, Q, D, the last dime is to be returned right away (why?). However, if he/she inserts the sequence Q, D, Q, there is no last coin return involved (why?).

6. The return of money is to be done using a coin return mechanism that can return Qs, Ds and Ns. Coins are to be returned sequentially, and a strategy of returning the minimum number of coins should be followed. So if 20 cents are to be returned two D's should be returned (as opposed to, say, one D and two N's). Assume that all the needed coins are always available.
7. There is also a Cancel button, which when pressed should result in all coins inserted by the customer (if any) to be returned.
8. The latency for the electro-mechanical system to return a single coin is 3 secs, while the latency of the system to dispense a product is 10 secs.
9. The machine also needs to have a timeout mechanism of 20 secs after each coin insertion. After the last coin insertion, if 20 secs elapse without the customer inserting another coin or pressing a choice or Cancel button, the customer is to be prompted with "Do you need more time?". The customer needs to respond to this using a Yes or No button. After pressing the Yes button, the customer is allowed another 20 secs, and 10 secs after pressing No. If no action is performed by the customer by the time either time units elapse, all the money inserted by the customer is to be returned. Similarly, if the customer does not respond to the above message in 20 secs, all his/her money is to be returned.
10. If the customer presses a choice button without sufficient money inserted, he/she is to be given the message "Insufficient money inserted" that stays until more money is inserted, or an appropriately different choice is pressed or the Cancel button is pressed or 20 secs elapse (in the latter case, the above action is to be performed by your machine).
11. If you think some specification is missing, please incorporate that specification and clearly state what the added specs are.

Provide your design as follows:

- A block-level system diagram (similar to that of the vending machine system in the FSM notes) showing: (a) all inputs to the system (properly labeled indicating what they are for); (b) all system outputs (properly labeled indicating what they are for); (c) all timers; (d) all FSM modules, and the various outputs of the output module (properly labeled indicating what they are for); (e) the display module(s); (f) any other modules needed.
- A neatly drawn state-transition diagram with mnemonically labeled states that indicate the meaning of the state (also, be sure to label the *reset* state as such). Separately provide a description of the

purpose/functionality of each state. Finally, provide a concise rationale and description of your FSM design (refer to the state labels to describe your design).

400

## VHDL Description

Describe your designed FSM in VHDL using multiple processes as discussed in class. The FSM should be positively edge triggered. See the lecture notes on “Sequential Circuits (FSMs) and Their VHDL Descriptions” for FSM descriptions in VHDL.

You need to have appropriate signal assignment statements with delays to simulate a state-transition logic delay of 5 ns, output logic delay of 3 ns and a FF delay of 2 ns.

You also need to have a clock process that generates a clock with a period of 10 ns and a duty cycle of 50% (i.e., high period of 5 ns). Such a clock process is given below.

```
architecture behav of pattern_det_fsm is
  signal clk: std_logic := 0;
  other declarations

  clock: process is
  begin
    clk <= '0','1' after 5 ns;
    wait for 10 ns ;
  end process clock;

  :
end architecture behav;
```

Each display should be implemented by a process using the **report** statement in VHDL. A display process is triggered whenever the corresponding output from the FSM goes high. When there is nothing to be displayed, a blank line is to be actually displayed.

400

## Testbench

Design a test bench that uses a hard-coded set of symbolic input sequences given to you by the TA. The input sequences will have symbols for the various inputs, possibly alternating with the time between two

inputs; when there is no time between two inputs, it means that they occur simultaneously. E.g., "Q, 5 sec, D, 4 sec, D, 7 sec, C" is a symbolic sequence which means a customer inserts a quarter waits for 5 secs, inserts a dime, and so forth and finally presses the C selection button. There will be multiple such symbolic sequences provided to you, and for each, one complete simulation of your design needs to be performed.

The TB needs to feed the appropriate input bit vector to your system based on the symbolic inputs given by the TA; consecutive bit input vectors need to be separated by the amount of time given in the current symbolic input sequence being simulated. So for a "Q, 5 sec, D, ..." sequence, your TB will need to generate a bit vector  $b_1$  in which the  $Q$  input is 1 and all other inputs are 0, then wait for 5 secs, then generate another bit vector  $b_2$  in which the  $D$  input is 1 and all others 0, and so forth.

Also, each bit vector needs to be provided to your system by the TB at every negative edge of the clock. Note that by feeding the input to the FSM every negative edge, we ensure that the FSM will be able to reliably sample it at the clock's positive edge.

Provide the timing graphics of the inputs and the outputs corresponding to the sequence of symbolic inputs provided to you.