Finite State Machine Design–A Vending Machine

You will learn how turn an informal sequential circuit description into a formal finite-state machine model, how to express it using ABEL, how to simulate it, and how to implement it and test it on the logic board.

1.0 Introduction

Design of sequential circuits is quite a bit more involved than that of combinational circuits. There are many opportunities for things to go wrong, and debugging a malfunctioning circuit can be tedious at best and downright frustrating at worst. Fortunately, by adopting a structured approach to the design and debug of sequential circuits, and by carefully documenting the design, many of these problems become manageable. The analogy to software development here is quite apt; if you write “spaghetti” code you are more likely to be frustrated when things don’t work, and won’t get much sympathy when you seek help. The reward for discipline, on the other hand, is that you’ll spend less time chasing after hard-to-find bugs and more enjoying the fruits of your creativity. Design *can* be fun!

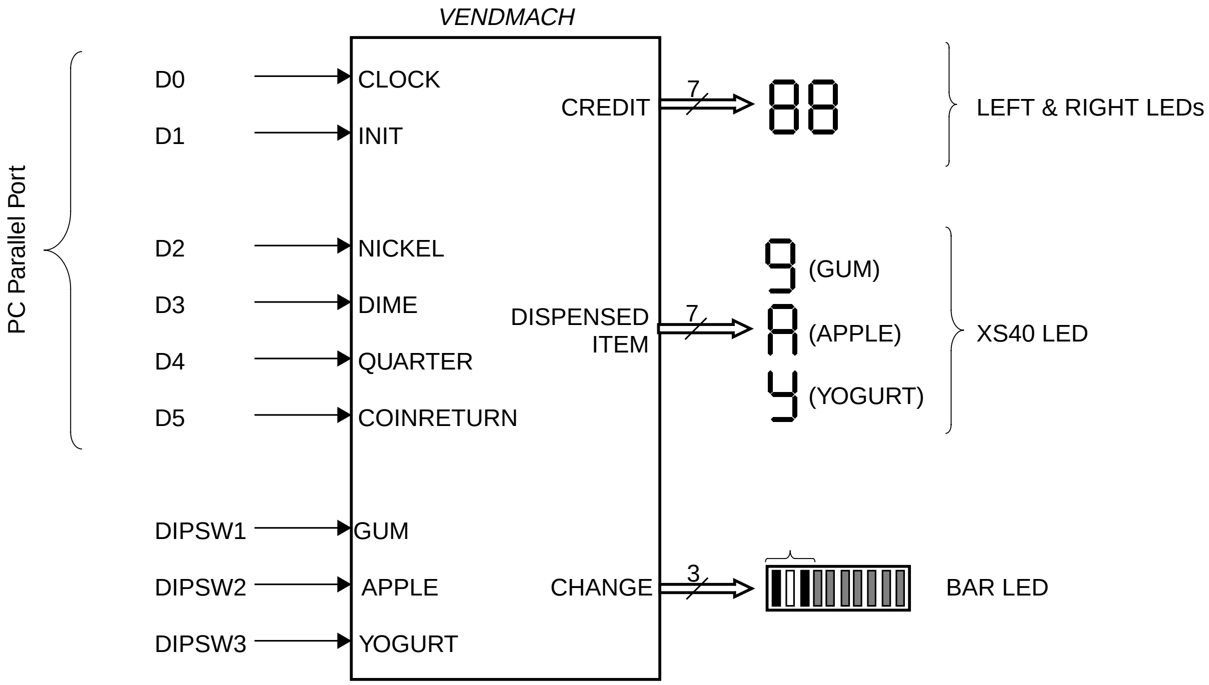
2.0 Preparation

Read Chapter 7 of Wakerly, especially Sec. 7.6 on state machine synthesis using transition lists and Sec. 7.11 on the sequential features of the ABEL HDL; additional help on ABEL is available through the on-line documentation of the Foundation software.

3.0 Design Specifications

VENDMACH is a vending machine that accepts nickels, dimes, and quarters, and dispenses gum, apple, or yogurt. A gum pack costs 15¢, an apple is 20¢, and yogurt is 25¢.

FIGURE 8. Top-level schematic of *VENDMACH*



The top-level schematic of *VENDMACH* is shown in Figure 8. The machine has the following 1-bit inputs:

- NICKEL: a signal that becomes 1 when a nickel is deposited in the coin slot.
- DIME: a signal that becomes 1 when a dime is deposited in the coin slot.
- QUARTER: a signal that becomes 1 when a quarter is deposited in the coin slot.
- COINRETURN: a signal that becomes 1 when the coin return button is pressed.
- GUM: a signal that becomes 1 when the gum selection button is pressed.
- APPLE: a signal that becomes 1 when the apple selection button is pressed.
- YOGURT: a signal that becomes 1 when the yogurt selection button is pressed.

In addition to these “user” inputs, the machine has two control inputs:

- CLOCK: a timing signal that sequences the state transitions of the machine.
- INIT: an initialization signal that resets the machine to a suitable starting state.

The machine has three outputs:

- CREDIT: the amount of money deposited so far and available to make a purchase; CREDIT, in cents, should be displayed on the LEFT and RIGHT LED digits.
- DISPENSED ITEM: the item that was just purchased should be displayed on the XS40 LED: g for gum, A for apple, and y for yogurt, as indicated in Figure 8.

- CHANGE: the amount of money returned in change, or as a result of pressing the coin return button. The machine returns change using only nickels; the number of nickels returned should be displayed as a binary number using the three left-most BAR LED segments (bright segment for 1, dark segment for 0.)

The machine should behave in accordance with the following specifications:

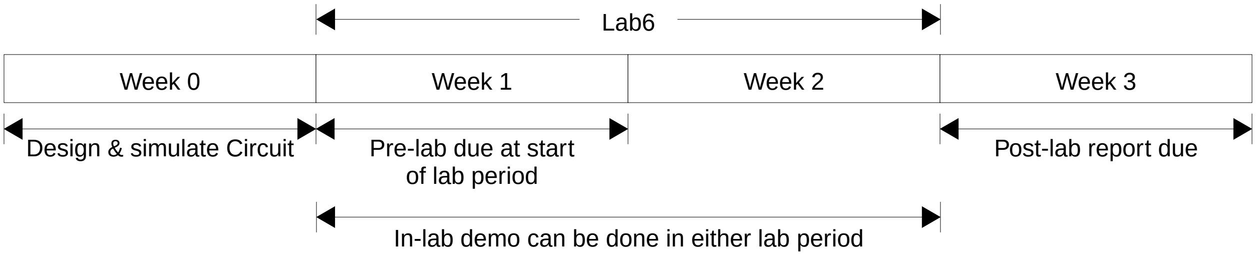
1. A customer needs to deposit a sufficient amount of money *before or at the same time* that he or she is selecting an item for purchase. If the item costs less than the deposited amount, the item is dispensed and the correct change is returned. If the item costs more than the deposited amount, the machine waits for more coins to be inserted or for a cheaper item to be selected.
2. The coin slot in the machine will only accept a single coin. Attempts to jam two or more coins cannot affect the state of the machine and are ignored. Similarly, attempts to make multiple selections (e.g. selecting apple and yogurt at the same time) are ignored.
3. At any time, pressing the coin return button causes the credit amount to be returned as change in nickels. Furthermore, when credit is equal to 25¢, any additional coins that are inserted are immediately returned as change. Finally, the return button closes the coin insertion slot; thus attempting to deposit a coin while at the same time pressing the return button prevents the coin from being inserted.

4.0 Design Notes and Hints

- This circuit has too many inputs and state variables to be designed using an explicit state diagram model. Instead, it is best approached by constructing a “transition list” that gives symbolic transition expressions for each pair of present and next states. Construct this transition list and code it using ABEL’s state diagram construct. Make sure that your transition expressions for every possible present state are mutually exclusive and all-inclusive.
- Use a one-hot state encoding. The FPGA has plenty of flip-flops, and a sparse one-hot encoding of the machine states is more appropriate than an encoding that uses the fewest number of flip-flops.
- Sequential circuits are a lot harder to design and debug than combinational circuits. Make sure you provide a means to initialize the flip-flops in your design. Whenever you notice signals whose value is X or Z, check to make sure that the flip-flops driving them are properly initialized.
- While we have not paid much attention to electrical issues in this lab, sometimes reality hits and we must face it. Signals that have high fan-out (i.e. a lot of connections) such as clocks must be buffered sufficiently and treated specially by the wire routing tools (to use high-speed interconnect layers, e.g.) to avoid timing problems. There are a number of “global” clock buffers that you can instantiate to help deal with these problems. For the clock input, it is suggested that you insert a BUFGLS after the IBUF you normally use on input signals.

5.0 Deliverables

This is a two-week lab with the following timeline:


Lab6
Week 0
Week 1
Week 2
Week 3
Design & simulate Circuit
Pre-lab due at start of lab period
Post-lab report due
In-lab demo can be done in either lab period

The pre-lab is due at the beginning of your lab period in week 1. The in-lab demonstration can be done during either of the two lab periods; if you successfully demonstrate a working design in the first lab period, you don't have to come to the second lab period.

5.1 Pre-Lab (25 Points)

1. Hardcopy of *VENDMACH's* schematic, ABEL, and UCF files.
2. Hardcopy of simulation results for the sequence of inputs shown below:

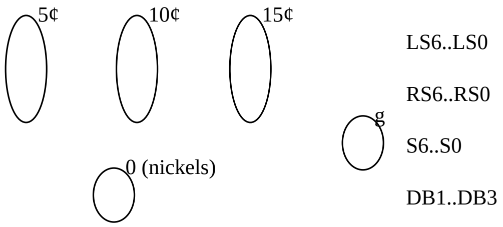
Cycle	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Inputs	N	N	N	G	Q	A	Q	Q	N	N	D	N	N	Q	A	R	Q	Q	D	Y
							G	R	D		G		G		Y	A				
								Y	Q											

In the above table, N, D, Q, R, G, A, Y stand for asserting (i.e. setting to 1) the nickel, dime, quarter, coin return, gum, apple, and yogurt inputs. You should make sure the circuit is initialized properly before applying the above input sequence. Display the simulation traces using the format shown in the Figure 9; annotate your waveforms (either using the comment feature of the waveform viewer, or by marking your hardcopy) to facilitate their interpretation. Provide a table showing the correspondence between the hexadecimal values displayed for the "bus" signals and their interpretation as displayed on the LEDs. For example, the hexadecimal values 6D and 14 on LS6..LS0 and RS6..RS0 display as the decimal digits 15, meaning a 15-cent credit.

5.2 In-Lab (30 Points)

Generate the .bit file for *VENDMACH*, download it to the XESS board, and verify that your implementation is working properly. Note that applying input to the FPGA requires that you "issue" three stimulus commands: a) set the inputs to their required values (while the clock is low); b) assert the clock (make it go high); and c) de-assert both the input and the clock. When you are satisfied that you have a correctly-function-

FIGURE 9. Format for displaying simulation results



ing circuit, demonstrate its operation to your lab GSI and have him sign your experiment’s cover sheet.

5.3 Post-Lab (45 Points)

Prepare your lab report as described in the *EECS270 Laboratory Overview* handout. Make sure you complete and include all parts of the report including the *Cover Sheet*, the *Design Narrative* section, and the *Design Documentation* section. Include as part of your design documentation all corrected pre-lab requirements. DO NOT INCLUDE THE MAP AND PLACE & ROUTE IMPLEMENTATION REPORT FILES. Instead, consult the report files (and any other files in the project directory) to find the following information about the design:

- Utilization of the FPGA resources: number of CLBs used, number of flip-flops used, and number of 3- and 4-input look-up tables (LUTs) used.
- Worst case net delay.
- Minimum clock period and maximum clock frequency for reliable operation.
- Minimum and maximum fan-in (number of first-level AND gates) over the set of implemented output functions.
- Number of product terms used in the implementation.

In your design narrative, provide enough detail about the choices you made to realize your particular implementation of this circuit. Specifically, document any flexibility in the design specification that you exploited to make particular design decisions.