HARDWARE LAB 5/DESIGN PROJECT
Finite State Machine
Design of a Vending Machine
Using Xilinx ISE Project Navigator and
Spartan 3E FPGA Development Board with VHDL

Acknowledgements: Developed by Bassam Matar, Engineering Faculty Chandler-Gilbert Community Colleges in Arizona.

Lab Summary: In previous labs, you become familiar with VHDL design and programming. In this lab, you will increase your knowledge by implementing a more complex digital design problem (Vending Machine). In this lab activity, we will 1) use Xilinx to derive the VHDL code for a vending machine and then program it to the Spartan 3E FPGA device; 2) you will apply your understanding of VHDL by implement a vending machines with more options (i.e. Design Project)

Lab Goals: The goals of this lab are to learn how to create a VHDL from state diagrams Xilinx® ISE software by designing a vending machine.

Learning Objectives

1. Develop the state diagram for a vending machine.
2. Develop the VHDL code.
3. Develop a User Constraint File **"ucf"** that maps the input and output signals to the Spartan 3E FPGA
4. Program the vending machine into the FPGA Spartan 3E board
5. Learn how to debug hardware
6. Repeat the same process for a vending machine with more options.

Grading Criteria: Your grade will be determined by your instructor.

Time Required: 2-3 hours

Special Safety Requirements

Static electricity can damage the CPLD or FPGA devices used in this lab. Use appropriate ESD methods to protect the devices. Be sure to wear a grounded wrist-strap at all times while handling the electronic components in this circuit. The wrist strap need not be worn after the circuit construction is complete.

No serious hazards are involved in this laboratory experiment, but be careful to connect the components with the proper polarity to avoid damage.

Lab Preparation

- Read Task1 and 2. Verify the operation to the state diagram for the 15 cents Moore and Mealy vending machines
- Acquire required hardware components/equipment.

Equipment and Materials

Each team of students will need the test equipment, tools, and parts specified below. Students should work in teams of two or three.

Test Equipment and Power Supplies	Quantity
The following items from the Xilinx: • free software ISE *WebPACK (***www.xilinx.com)** that can be installed on your personal computer or full version of Xilinx in your classroom • Spartan-3E Starter Kit, including download cable and power supply	1
ESD Anti-static Wrist Strap	1

Additional References:

1. Xilinx Spartan 3E FPGA Reference Manual and Schematic from:
http://www.digilentinc.com/Products/Detail.cfm?Prod=S3EBOARD&Nav1=Products&Nav2=Programmable

Vending Machine Design

In this Lab, we will look at making a vending machine. Vending machines are more complicated than you might think. We know that a vending machine must remember how much money has been inserted. This means that its outputs are a function of past inputs, so it must be a sequential circuit. Probably the best way to design this circuit is as a state machine. (Designing it using a microprocessor might be better, since microprocessors are so cheap and common. However, this option takes more design work, and we will not consider it here.)

Here is how the control circuitry is supposed to work. The vending machine delivers a package of gum after it has received ***15 cents*** in coins. The machine has a single coin slot that accepts ***nickels and dimes***, one coin at a time. A mechanical sensor indicates to the control whether a dime or a nickel has been inserted into the coin slot. The controller's output causes a single package of gum to be released down a chute to the customer.

One further specification: We will design our machine so it **does not** give change. A customer who pays with two dimes is out 5 cents.

TASK 1. Part 1:

a. Understanding the Problem

The first step in the finite state machine design process is to *understand the problem*.
Here is a block diagram of our design that illustrates inputs and outputs.

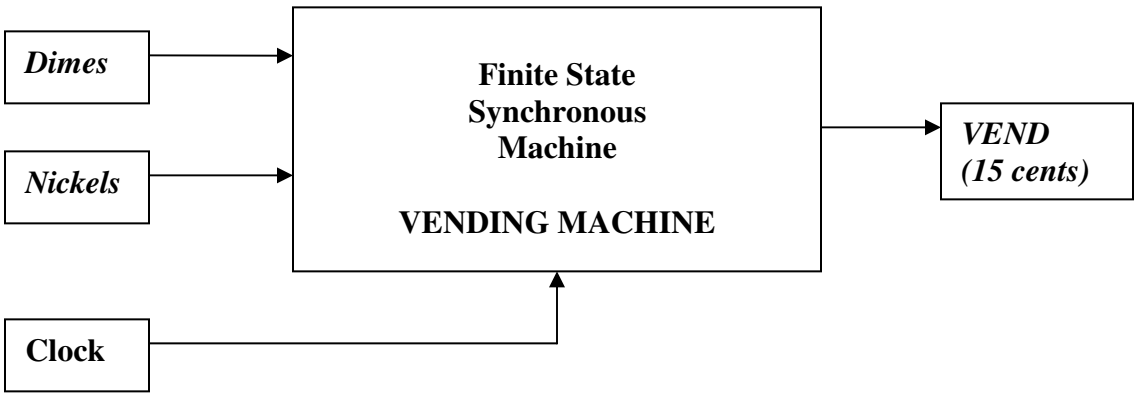


Figure 1. **Vending machine block diagram**

The specifications may not completely define the behavior of the finite state machine. For example,

1. What happens if someone inserts a penny into the coin slot? or
2. What happens after the gum is delivered to the customer?

Some times we have to make reasonable assumptions. For the first questions, we assume that the coin sensor returns any coins it does not recognize, leaving N and D unasserted. For the latter, we will assume that external logic resets the machine after the gum is delivered.

b. Abstract representation

Once you understand the behavior reasonably well, it is time to *map the specification into a more suitable abstract representation*. A good way to begin is by enumerating the possible unique sequences of inputs or configurations of the system. These will help define the states of the finite state machine.

c. Here are all the possible input sequences that lead to an output of releasing a package of gum (15 Cents):

N= ***Nickels***
D= ***Dime***

1. N N N
3. D N
2. N D
4. D D (out 5 cents)
5. N N D (out 5 cents) (yes it is possible)

Assumptions:

- Only N or D can be 1 during any one clock pulse.
- Campus Machine (i.e., it doesn't give change.)
- An external reset signal supplied on "power-up."

The possible input sequences that lead to releasing a package of gum can be represented with a state diagram.

Input

N = Nickle (N=1 means 5 cents deposited)
D = Dime (D=1 means 10 cents deposited)

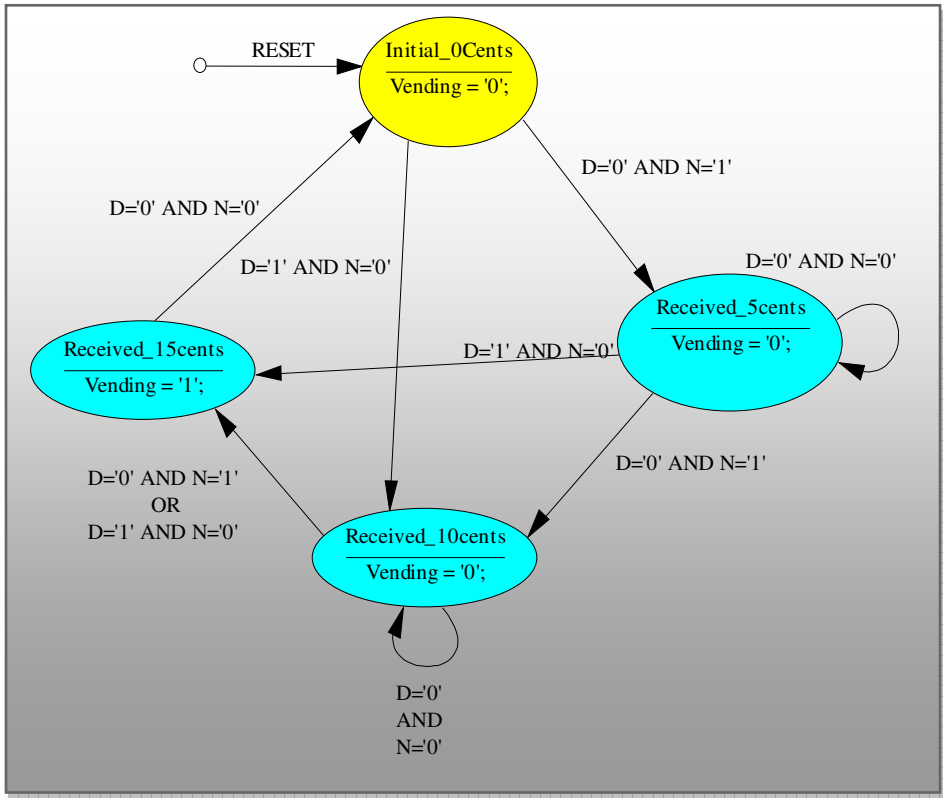
Output

Vend = Package of gum

State Definitions

State	Meaning
Initial	Reset/ No money has been deposited
Received_5cents	The vending machine received 5 cents
Received_10cents	The vending machine received 10 cents
Received_15cents	The vending machine received 15 cents

.

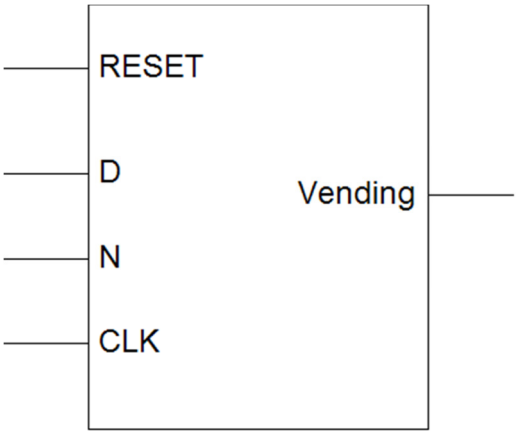


VHDL Code for the Moore State Diagram is available on BlackBoad under Design Project folder.

Part 2 of Task 1: 1n this task, we will interface the vending machine with Spartan 3F FPGA development board.

Based on the information that you have learned in previous lab (Part 2, Task2):

1. Create a **project** for the vending machine
2. Obtain a symbol as shown below



3. Create the "**ucf**" file that maps the inputs (Reset, Clock, Nickel and Dime) and the output (vending).

```
# Inputs:
NET "CLK" LOC = "C9";          # Location of 50 MHZ Clock on Spartan 3E
NET " CLR" LOC = "L14";        # Switch 1 (SW1). Reset the vending machine to state 0 cents.
NET "D" LOC = "H18";           # Switch 2 (SW2) for Dime input
NET "N" LOC = "N17";           # Switch 3 (SW3) for Nickel input

# Outputs:
NET "vending" LOC = "E12";     # LD 1 for 15 cents (VEND)
```

This UCF file is on BlackBoard under Design Project folder.

4. Verify the operation of the 15 cents vending machine using Spartan 3E FPGA evaluation board.

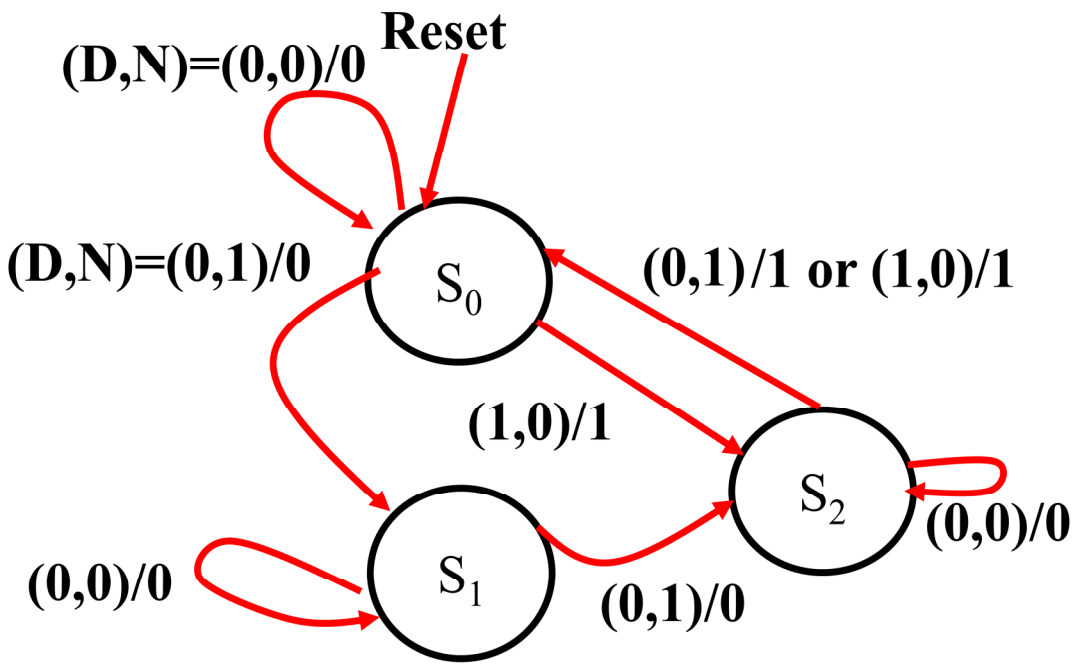
Repeat the same procedures for the mealy machines.

TASK 2:

Mealy Design State Definitions

State	Meaning
Initial	Reset/ No money has been deposited
Received_5cents	The vending machine received 5 cents
Received_10cents	The vending machine received 10 cents

Same Assumption as Moore Machine. One way of designing the Mealy State Diagram:



VHDL Code for the Mealy State Diagram is available on BlackBoad under Design Project folder. Pay attention to the Mealy output VHDL.

DESIGN PROJECT

You are to design two CONCEPUTALLY different synchronous state machines (Mealy and Moore) that perform the task described below.

First, **you will need to implement both designs using Xilinx. Second**, **you will need to implement ONE design using Spartan 3E FPGA Evaluation board you used in earlier lab work and demonstrate to only your instructor that your circuit works correctly**. The instructor will test your design and stamp the design pages of your report after successful demonstration. Your final report will be collect by your instructor at the conclusion of your demonstration. **Bring your report with you to your in-lab hardware demonstration.**
You must complete this demo by the assigned due date.
You have only one chance to demo this to your instructor, so make sure it works before you begin.

In your report, you must:

(1) Explain how each circuit works.
(2) Pick one circuit as the recommended answer.
(3) Explain why you prefer the circuit picked.

Lab project make-ups will NOT be allowed.

You can organize this report, as you think best. One possible organization is:

(1) Problem Statement
(2) First Design: Principles and Description
(3) Second Design: Principles and Description
(4) Design Selection Criteria
(5) Justification for Design Selection
(6) Conclusion (if you think it's needed)

Grading Policy

You may discuss this project with each other; however, you must provide an individual report. In addition:

(1) Individually, you must interface at least one design with FPGA board in order to have your individual report count.

The grade will be allocated as follows:

30% Design Simulation With Xilinx

- 15% for the design of the first circuit and VHDL Code
- 15% for the design of the second circuit and VHDL Code

70% Build One Design in the Hardware Lab and Lab Report

- 30% demonstration of one design in the hardware lab
- 10% for an explanation in the report of how the first circuit performs the application
- 10% for an explanation in the report of how the second circuit performs the application
- 10% for picking a preferred design
- 10% for providing a sensible reason for picking one design as the "Best" design. (The one design you build in the hardware lab does not need to be the "Best" design)

RULES FOR PERFORMANCE

1. You will only be allowed to do the project during the assigned times. Your instructor will provide the hardware demo times. ***See Lab Schedule on BlackBoard.***

2. Hardware Demo: You must have your lab report completed before you start the lab demo since you will need to give it your instructor at the completion of the demo. (Having your report finished will help you complete the lab project in the allotted time

Hardware Demo: You will have ONLY 2 HOURS to complete the lab project in the hardware lab. You must work alone with no lab partners to assist you. You may bring any textbooks, notes, reference books, lab notes, lab experiments, and lab reports.

Notice: If you are thinking of omitting this lab project from your schedule, be reminded that parts of this lab project will show up on the final exam for this course! And It is 10% of your final class grades.

Problem Statement

You are to design a Moore network similar to previous task to control a vending machine. The machine only dispenses product that costs **25 cents.**

The Moore network has 3 inputs: N, D, and Q (nickel, dime and quarter inputs) and three outputs, P, V, X (dispense product, and dispense change in dimes or nickels). V and X (Roman numerals) represent one nickel or one dime dispensed.

The coin detector mechanism in the vending machine is synchronized with the clock in the network you are to design. The coin detector outputs a single 1 for N, D, or Q for each nickel, dime, or quarter, respectively, that the customer inserts. Only one input will be 1 at any time. When the customer has inserted at least 25¢ in any combination of nickels, dimes and quarters, the machine dispenses one yummy piece of candy and gives any change due. The change may be dispensed in nickels or dimes. For each 1 output on V, the customer is issued one nickel, and for each 1 output on X, the customer is issued a dime. For each 1 output on P, the delicious candy is dispensed. After dispensing the product, the network resets.

Notes:

- any number of 0's can occur between the 1's on the inputs
- you may assume that the customer will not insert any more coins once the machine is dispensing change
- nickels and dimes should not be dispensed as change during the same clock cycle; product *may* be dispensed at the same time as a nickel or dime
- If you make any other assumptions, they must be *clearly* stated in your report.
- You should design with VHDL using STATE CAD editor.

Example: The customer inserts a nickel, then a dime, then a quarter.

INPUTS:

N = 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
D = 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
Q = 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0

OUTPUTS:

P = 0 0 0 0 0 0 0 0 0 0 - - - - 0 0 0 0 (product can be dispensed anywhere shown with -'s, but only once)
V = 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 (5¢ in change)
X = 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 (10¢ in change) = 15¢ total

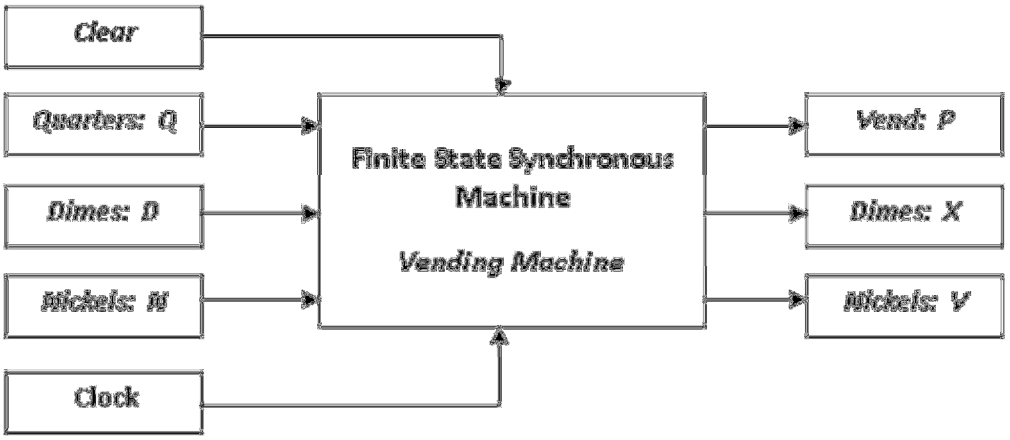
---

Notes:

- Please assume that, once the user has put in enough to receive some yummy candy, they will NOT insert any more change until you are ready for them. In other words, while you are dispensing their change, they won't put in any more coins.
- Be sure that, when dispensing change, **you do NOT dispense two coins at the same time**. Just as the user shouldn't put in 2 coins at once since they may get stuck, you can't dispense 2 at the same time....

Please incorporate the following items in your report:

1. Draw a block diagram
2. List all the possible combinations of Nickels, Dimes and Quarters that will lead you to a vend.
3. Define all your states
4. Draw the state diagram
5. Develop the VHDL code
6. Create a new project for the machine
7. Create a symbol for the machine
8. Create a UCF file
9. Download your design to the Spartan 3E FPGA board,,
10. Test and demo your results to you instructor



Block Diagram

Combinations for Vend

Combinations for Vend		
25 cents	**30 cents**	**40 cents**
N, N, N, N, N	N, N, N, N, D	N, N, N, Q
N, N, N, D	N, D, N, D	N, D, Q
N, N, D, N	D, N, N, D	D, N, Q
N, D, N, N	N, N, D, D	**45 cents**
D, N, N, N	D, D, D	N, N, N, N, Q
D, D, N	N, Q	N, N, D, Q
D, N, D	**35 cents**	D, D, Q
N, D, D	N, N, Q	D, N, N, Q
Q	D, Q	N, D, N, Q