Indian Institute of Space Science and Technology

Vending Machine Controller Using Verilog

By:
Neha Binny (SC19B090)
Nirbhay Tyagi (SC19B091)

List of Figures

List of Tables

Listings

Contents

Introduction

A vending machine is a machine which dispenses items such as snacks, beverages, lottery tickets, consumer products to customers automatically after the customer inserts currency or credit into the machine. There are vending machines for newspapers, fast food, cokes, coffee, stamps, ticket, change etc. These vending machines can be put to use in these times of COVID-19 to implement social distancing, so that people can easily access these machines without entering the crowded markets.

The purpose of this project is to design a Vending FSM and then implement it using Verilog. We need to design a vending machine which accepts money input in any sequence and delivers the product when the required price has reached and also returns back the change. The Vending machine also provides an additional facility of cancelling the transaction in between by pressing a push button.

For designing the vending machine, we need to first specify the price of the product provided by the vending machine and the possible money inputs(e.g., Re 1 coins, Rs 2 Coins) which are accepted by the vending machine. Then we need to define the states required in the process and then draw a state diagram mentioning the state transitions and input/output relation. On the basis of state diagram, we will write a Verilog code to implement and realize the above designed Vending Machine. A test bench is then written and visualised using GTK Wave to test the functionality of the written code.

1 Implementation

A state diagram is constructed for the vending machine which can vend a mask of ₹7. The machine accepts the following currency coins:

- ₹1 coin
- ₹2 coin
- ₹5 coin

The customer inserts the coin into the vending machine through the input coin slot and the type of inserted coin is then identified by a sensor which generates a electrical signal that is then processed by the vending machine controller. The controller counts the number and the type of coin inserted in the machine and dispenses a mask. If the inserted money is more than ₹7 then, the corresponding change will be returned through the change output slot.

2 Design Methodology

Any sequential digital circuit can be converted into a state machine using state diagram. In a state machine, the circuit's output is defined in a different set of states i.e. each output is a state. There is a state register to hold the state of the machine and a next state logic to decode the next state. There is also a output register that defines the output of the machine. The next state logic is the sequential part of the machine and the output and current state are the register part of the logic.
There are two types of state machines:

MOORE Machine:

In a Moore machine the output state is totally dependent on the present state.

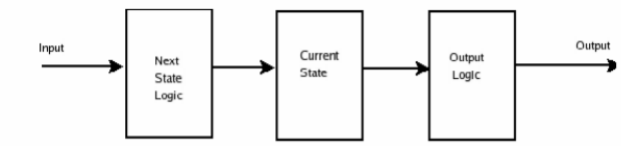

Figure 1: MOORE Machine Model

MEALY Machine:

In a Mealy machine the output depends on the input as well as the present state.

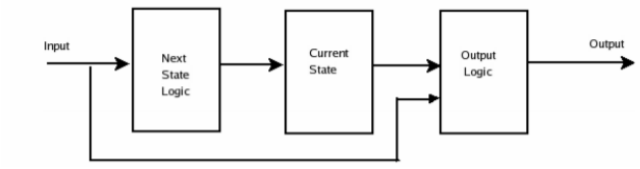

Figure 2: MEALY Machine Model

2.1 State Diagram

We are using mealy machine for the implementation of the Vending Machine controller.

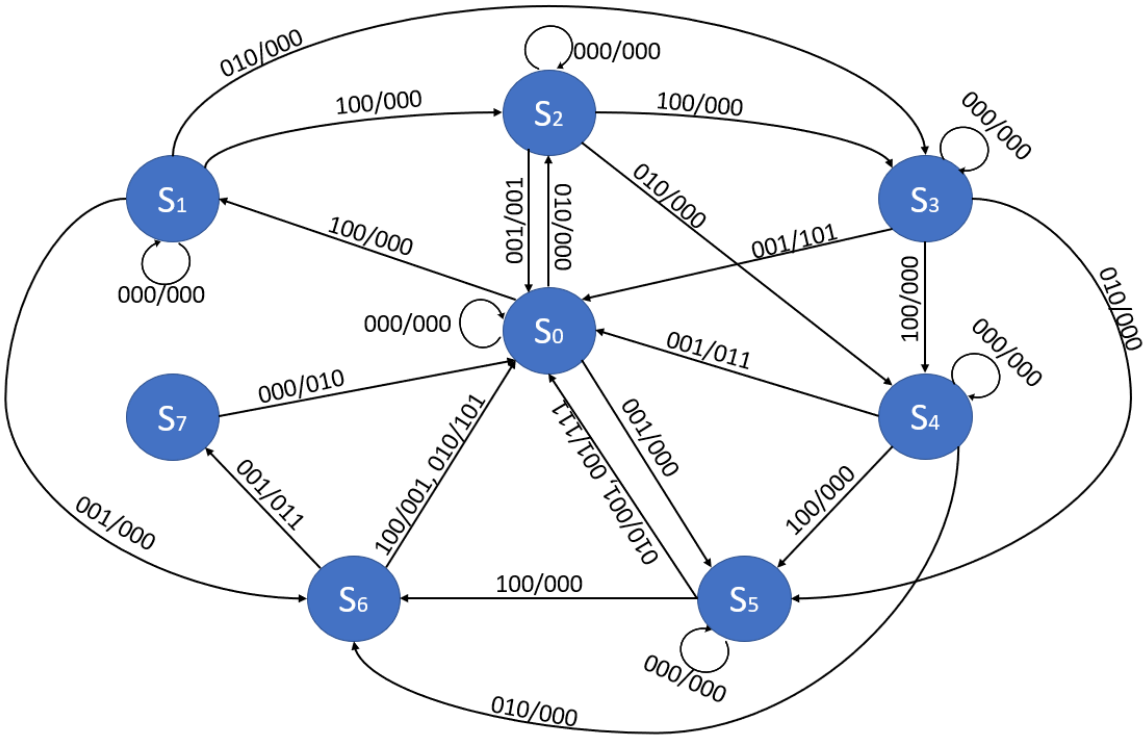

Figure 3: State Diagram - Vending Machine

2.2 Description of States

The State Diagram of the Vending Machine is shown in Figure 3. It has the following states:

1. **S_0:** It is the reset state.
2. **S_1:** The money count at this state is ₹1.
3. **S_2:** The money count at this state is ₹2. This state returns to S_0 when there is an input of ₹5 and dispenses a mask.
4. **S_3:** The money count at this state is ₹3. This state returns to S_0 when there is an input of ₹5 and dispenses a mask along with a change of ₹1.
5. **S_4:** The money count at this state is ₹4. This state returns to S_0 when there is an input of ₹5 and dispenses a mask along with a change of ₹2.
6. **S_5:** The money count at this state is ₹5. This state returns to S_0 and dispenses a mask when there is an input of ₹2 or ₹5. The state also returns a change of ₹2 and ₹1 coins when the input is ₹5.
7. **S_6:** The money count at this state is ₹6. This state returns to S_0 and dispenses a mask when there is an input of ₹1 or ₹2. The state also returns a change of ₹1 coin when the input is

₹2. When the input is ₹5, the state goes to S_7 and dispenses a mask along with a change of ₹2 coin.

8. **S_7:** This state is accessed when total money inserted is ₹11 and returns a change of ₹2.

2.3 State Table

The state table is constructed using the state diagram shown in the Figure 3.

Present State	**one_in**	**two_in**	**five_in**	**Next State**	**one_balance**	**two_balance**	**dispense**
S0	0	0	0	S0	0	0	0
	1	0	0	S1	0	0	0
	0	1	0	S2	0	0	0
	0	0	1	S5	0	0	0
S1	0	0	0	S1	0	0	0
	1	0	0	S2	0	0	0
	0	1	0	S3	0	0	0
	0	0	1	S6	0	0	0
S2	0	0	0	S2	0	0	0
	1	0	0	S3	0	0	0
	0	1	0	S4	0	0	0
	0	0	1	S0	0	0	1
S3	0	0	0	S3	0	0	0
	1	0	0	S4	0	0	0
	0	1	0	S5	0	0	0
	0	0	1	S0	1	0	1
S4	0	0	0	S4	0	0	0
	1	0	0	S5	0	0	0
	0	1	0	S6	0	0	0
	0	0	1	S0	0	1	1
S5	0	0	0	S5	0	0	0
	1	0	0	S6	0	0	0
	0	1	0	S0	0	0	1
	0	0	1	S0	1	1	1
S6	0	0	0	S6	0	0	0
	1	0	0	S0	0	0	1
	0	1	0	S0	1	0	1
	0	0	1	S7	0	1	1
S7	0	0	0	S0	0	1	0

Table 1: State Table - Vending Machine

The terms used in the above tables are:

- one_in: Input ₹1 coin
- two_in: Input ₹2 coin

- five_in: Input ₹5 coin
- one_balance: ₹1 change
- two_balance: ₹2 change
- dispense: dispenses one mask

3 Verilog Code

```
//Digital Electronics Lab Project
//Submitted to: Dr. Sheeba Rani J, Assosiate Professor, IIST
//Submitted by: Neha Binny (Sc19B090), Nirbhay Tyagi (SC19B091)

// Verilog Code - Vending Machine Controller
module vending_machine(one_in,two_in,five_in,clk,reset,one_balance,
   two_balance,dispense);//Module Declaration for vending Machine
   Controller
//State Assignment
parameter S0=8'b0000_0001, S1=8'b0000_0010, S2=8'b0000_0100, S3=8'
   b0000_1000, S4=8'b0001_0000, S5 = 8'b0010_0000, S6 = 8'
   b0100_0000, S7 = 8'b1000_0000;
//
input one_in,  // Input rupee 1 coin
      two_in,  // Input rupee 2 coin
      five_in, // Input rupee 5 coin
      clk,     // Clock of frequency 100MHz
      reset;   // Reset Active High
output reg one_balance,   // Change of rupee 1
           two_balance,   // Change of rupee 2
           dispense;      // Dispenses a mask
reg [7:0] current_state, next_state;
//Next State
always @(posedge clk) begin
    if(reset)
    current_state <=S0;
    else
    current_state<=next_state;
end
//Finite State Machine
always @(one_in, two_in, five_in) begin

    case (current_state)
        S0: begin     // Reset State
          if(one_in==1) begin
            next_state=S1;
            {one_balance, two_balance, dispense} = 3'b000;
```

```
34          end
35          else if(two_in==1) begin
36            next_state=S2;
37            {one_balance, two_balance, dispense} = 3'b000;
38          end
39          else if(five_in==1) begin
40            next_state=S5;
41            {one_balance, two_balance, dispense} = 3'b000;
42          end
43          else begin
44            next_state=S0;
45            {one_balance, two_balance, dispense} = 3'b000;
46          end
47        end
48
49        S1: begin     //Money Count is Rupee 1
50          if(one_in==1) begin
51            next_state=S2;
52            {one_balance, two_balance, dispense} = 3'b000;
53          end
54          else if(two_in==1) begin
55            next_state=S3;
56            {one_balance, two_balance, dispense} = 3'b000;
57          end
58          else if(five_in==1) begin
59            next_state=S6;
60            {one_balance, two_balance, dispense} = 3'b000;
61          end
62          else begin
63            next_state=S1;
64            {one_balance, two_balance, dispense} = 3'b000;
65          end
66        end
67
68        S2: begin     //Money Count is Rupee 2
69          if(one_in==1) begin
70            next_state=S3;
71            {one_balance, two_balance, dispense} = 3'b000;
72          end
73          else if(two_in==1) begin
74            next_state=S4;
75            {one_balance, two_balance, dispense} = 3'b000;
76          end
77          else if(five_in==1) begin
78            next_state=S0;
79            {one_balance, two_balance, dispense} = 3'b001;    //
  Dispenses a mask and No change
```

```
80          end
81          else begin
82            next_state=S2;
83            {one_balance, two_balance, dispense} = 3'b000;
84          end
85        end
86
87        S3: begin     //Money Count is Rupee 3
88          if(one_in==1) begin
89            next_state=S4;
90            {one_balance, two_balance, dispense} = 3'b000;
91          end
92          else if(two_in==1) begin
93            next_state=S5;
94            {one_balance, two_balance, dispense} = 3'b000;
95          end
96          else if(five_in==1) begin
97            next_state=S0;
98            {one_balance, two_balance, dispense} = 3'b101;    //
  Dispenses a mask and 1 Rupee change
99          end
100         else begin
101           next_state=S3;
102           {one_balance, two_balance, dispense} = 3'b000;
103         end
104       end
105
106       S4: begin     //Money Count is Rupee 4
107         if(one_in==1) begin
108           next_state=S5;
109           {one_balance, two_balance, dispense} = 3'b000;
110         end
111         else if(two_in==1) begin
112           next_state=S6;
113           {one_balance, two_balance, dispense} = 3'b000;
114         end
115         else if(five_in==1) begin
116           next_state=S0;
117           {one_balance, two_balance, dispense} = 3'b011;    //
  Dispenses a mask and 2 rupee change
118         end
119         else begin
120           next_state=S4;
121           {one_balance, two_balance, dispense} = 3'b000;
122         end
123       end
124
```

```
125         S5: begin     //Money Count is Rupee 5
126           if(one_in==1) begin
127             next_state=S6;
128             {one_balance, two_balance, dispense} = 3'b000;
129           end
130           else if(two_in==1) begin
131             next_state=S0;
132             {one_balance, two_balance, dispense} = 3'b001;    //
    Dispenses a mask and No change
133           end
134           else if(five_in==1) begin
135             next_state=S0;
136             {one_balance, two_balance, dispense} = 3'b111;    //
    Dispenses a mask and a change of rupee 1 and rupee 2
137           end
138           else begin
139             next_state=S5;
140             {one_balance, two_balance, dispense} = 3'b000;
141           end
142         end
143
144         S6: begin     //Money Count is Rupee 6
145           if(one_in==1) begin
146             next_state=S0;
147             {one_balance, two_balance, dispense} = 3'b001;    //
    Dispenses a mask and No change
148           end
149           else if(two_in==1) begin
150             next_state=S0;
151             {one_balance, two_balance, dispense} = 3'b101;    //
    Dispenses a mask anda change of rupee 1
152           end
153           else if(five_in==1) begin
154             next_state=S7;
155             {one_balance, two_balance, dispense} = 3'b011;    //
    Dispenses a mask and a change of rupee 2
156           end
157           else begin
158             next_state=S6;
159           end
160         end
161
162         S7: begin     // Returns remaining change of rupee 2 for a
    total of rupee 11
163           next_state=S0;
164           {one_balance, two_balance, dispense} = 3'b010;    //
    Dispenses no mask but a change of rupee 2
```

```
        end

        default: begin
          next_state=S0;
          {one_balance, two_balance, dispense} = 3'b000;
        end
    endcase
end
endmodule
```

Listing 1: Verilog Code - Vending Machine

4 Test Bench

```
//Digital Electronics Lab Project
//Submitted to: Dr. Sheeba Rani J, Assosiate Professor, IIST
//Submitted by: Neha Binny (Sc19B090), Nirbhay Tyagi (SC19B091)

//Test Bench - Vending Machine Controller
`timescale 1ns/1ns                  // Time Scale Directive
`include "vending_machine.v"        // Includes the Verilog file
module vending_machine_tb;          //Module Declaration
//DUT Input regs
reg one_in,two_in,five_in,reset;
reg clk=1'b1;
//DUT Output wires
wire one_balance,two_balance,dispense;
//DUT Instntiation
vending_machine DUT(.one_in(one_in),.two_in(two_in),.five_in(
   five_in),.clk(clk),.reset(reset),.one_balance(one_balance),.
   two_balance(two_balance),.dispense(dispense));
//Generating .vcd file
initial begin
    $dumpfile("vending_machine_tb.vcd");
    $dumpvars(0,vending_machine_tb);
    repeat(18)    //Determines Simulation limit
    #5 clk=~clk;  //Clock Generation
end
//Test Vectors
initial begin
    one_in=0;
    two_in=0;
    five_in=0;
    reset=1;
    # 10;

```

```
    reset=0;
    one_in=1;
    two_in=0;
    five_in=0;
    #10;

    reset=0;
    one_in=0;
    two_in=0;
    five_in=1;
    #10;

    reset=0;
    one_in=1;
    two_in=0;
    five_in=0;
    #10;

    one_in=0;
    two_in=0;
    five_in=0;
    #10;

    one_in=0;
    two_in=0;
    five_in=1;
    #10;

    one_in=0;
    two_in=0;
    five_in=0;
    #10;

    one_in=0;
    two_in=0;
    five_in=1;
    #10;

    one_in=0;
    two_in=0;
    five_in=0;
    #10;
end
// Display Output
initial begin
  $monitor("simulation time:%g  Rupee One Input:%b  Rupee Two Input
   :%b  Rupee Five Input:%b  Rupee one change:%b  Rupee two change
```

```
    :%b   Dispense:%b",$time,one_in,two_in,five_in,one_balance,
    two_balance,dispense);
77 end
78 endmodule
```

Listing 2: Test Bench - Vending Machine

5 Simulation Results

5.1 GTK Wave

To visualize the results we used GTK Wave and the simulation results are shown in Figure 4.

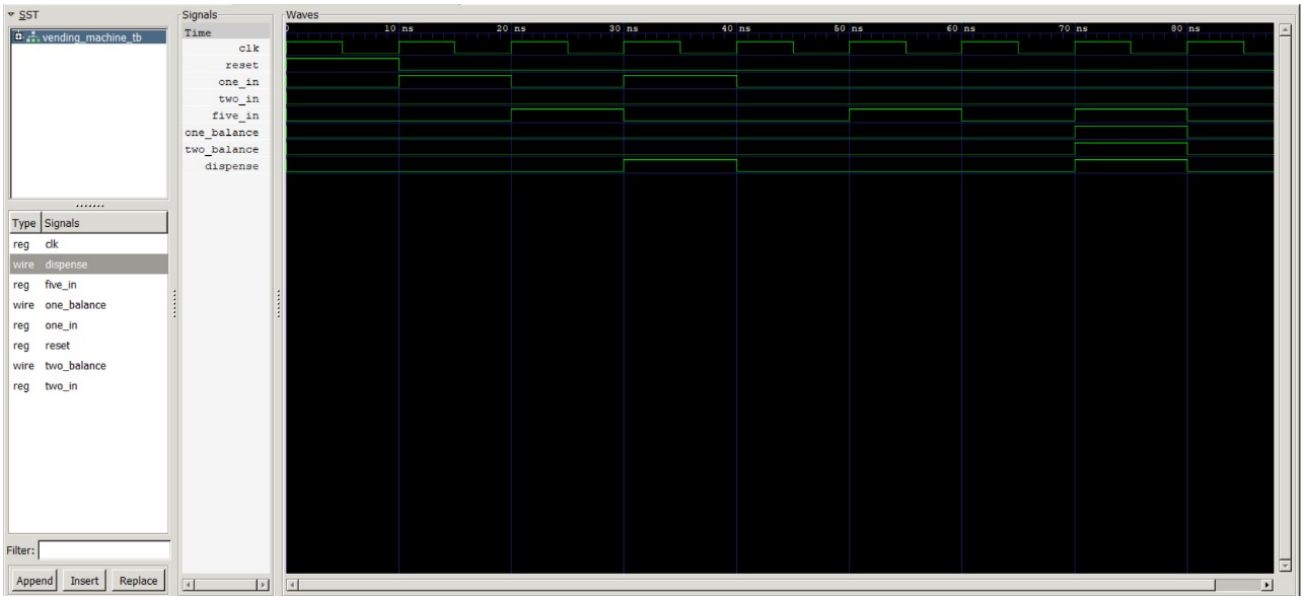

Figure 4: GTK Wave - Vending Machine

5.2 Test bench Output

The display/monitor output is shown in the Figure 5.

```
[Running] vending_machine_tb.v
VCD info: dumpfile vending_machine_tb.vcd opened for output.
simulation time:0  Rupee One Input:0  Rupee Two Input:0  Rupee Five Input:0  Rupee one change:0  Rupee two change:0  Dispense:0
simulation time:10  Rupee One Input:1  Rupee Two Input:0  Rupee Five Input:0  Rupee one change:0  Rupee two change:0  Dispense:0
simulation time:20  Rupee One Input:0  Rupee Two Input:0  Rupee Five Input:1  Rupee one change:0  Rupee two change:0  Dispense:0
simulation time:30  Rupee One Input:1  Rupee Two Input:0  Rupee Five Input:0  Rupee one change:0  Rupee two change:0  Dispense:1
simulation time:40  Rupee One Input:0  Rupee Two Input:0  Rupee Five Input:0  Rupee one change:0  Rupee two change:0  Dispense:0
simulation time:50  Rupee One Input:0  Rupee Two Input:0  Rupee Five Input:1  Rupee one change:0  Rupee two change:0  Dispense:0
simulation time:60  Rupee One Input:0  Rupee Two Input:0  Rupee Five Input:0  Rupee one change:0  Rupee two change:0  Dispense:0
simulation time:70  Rupee One Input:0  Rupee Two Input:0  Rupee Five Input:1  Rupee one change:1  Rupee two change:1  Dispense:1
simulation time:80  Rupee One Input:0  Rupee Two Input:0  Rupee Five Input:0  Rupee one change:0  Rupee two change:0  Dispense:0
[Done] exit with code=0 in 0.363 seconds
```

Figure 5: Testbench Output - Vending Machine

5.3 Explanation of Output

First, we reset the Vending Machine by generating the reset signal which sets the current state to S_0. We take two sets of inputs and observe the corresponding output. The two sets of inputs are explained below:

Set 1:
Input: Two ₹1 coins and One ₹5 coin

When the ₹1 coin is inserted in the machine, the current state changes from $S_0 \rightarrow S_1$. After that, the insertion of ₹5 coin, the current state changes from $S_1 \rightarrow S_6$. Finally, the insertion of ₹1 coin sums up to a total of ₹7 which changes the current state from $S_6 \rightarrow S_0$ thus, resetting the machine and dispenses a mask to the customer.

Set 2:
Input: Two ₹5 coin

When the ₹5 coin is inserted in the machine, the current state changes from $S_0 \rightarrow S_5$. Finally, the insertion of ₹5 coin sums up to a total of ₹10 which changes the current state from $S_5 \rightarrow S_0$ thus, resetting the machine. This also dispenses a mask to the customer and returns a change of ₹3 in the form of one ₹1 coin and one ₹2 coin.

6 Inferences

i.) The Vending Machine Controller is based on Mealy State Machine.

ii.) The output of the Vending Machine controller depends on current state as well as the input.

iii.) The Vending Machine resets itself when provided with a total input $\geq$ ₹7.

iv.) When the total input exceeds ₹7 the Vending Machine returns change which is equal to (Money Inserted - ₹7).

v.) The state S_7 is an exceptional state which is accessed only when the total money inserted amounts to ₹11.

vi.) When the total amount of ₹11 is inserted, a change of ₹4 has to be returned. A change of ₹2 is returned at state S_6 and the remaining change of ₹2 is returned at the state S_7.

7 Result

We have realized a verilog code of a Vending Machine controller using it's state diagram and state tables. The code written dispenses a mask and also returns the change/balance money to the customer. The verilog code has been successfully verified using GTK Wave and the desired outputs have been achieved.