ELL 201 Group Project Report
Vending Machine

Devesh Yadav 2022MT11281
Piyush Abnave 2022MT11289
Ishant Yadav 2022MT11397
Lokesh Varadraj 2022PH11858

April 2024

1 Introduction

Our topic for this project is An Ideal Vending Machine. A vending machine is an automated machine that dispenses items such as snacks, beverages, cigarettes and lottery tickets to consumers after cash, a credit card, or other forms of payment are inserted into the machine. The vending machine we designed for this project is an ideal machine(explained in the "Working of the Machine" section).

2 Concepts Used

The project is based on the concept of a clocked synchronous digital Finite State Machine (FSM). The FSM can change from one state to another in response to some inputs. The change from one state to another is called a transition. The FSM we use here is the Mealy Machine. Mealy Machine is a finite state machine whose output values are determined solely by its current state and inputs. We make use of a state transition diagram. The Mealy Machine State Diagram gives two outputs; one signifies the state it is going to, and the other signifies the output it gives at the initial state. The Mealy FSM is used to make vending machines.

3 Working of The Machine

The only denominations our vending machine accepts are 5 Rs and 10 Rs. The machine can only accept up to 15 Rs, which signifies the complete transaction. The machine also gives back the change after each transaction. Following is the state diagram of our machine:

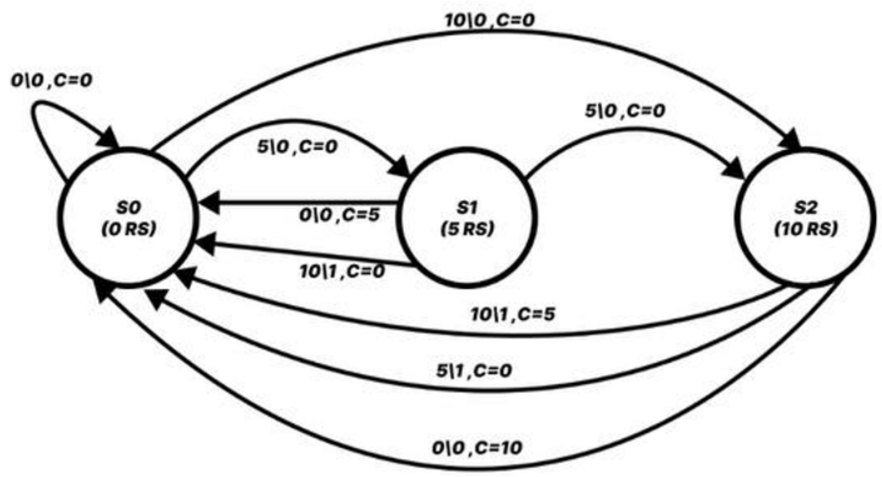
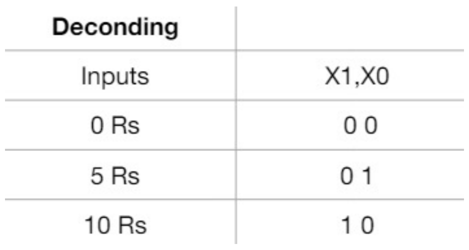


Figure 1: State Diagram

We have three states: S0 for 0 Rs, S1 for 5 Rs and S2 for 10 Rs. We require 15 Rs for a complete transaction. We transition between states depending on the input we give at any state (5 Rs or 10 Rs). A reset button is also present to reset the model, which takes us back to the 0 Rs state. We get two outputs. One output signifies the completion of a transaction, i.e. when the system receives 15 Rs or more than 15 Rs. This is signified by the ignition of one LED. In case the system receives more than 15 Rs (For example, the system has 10 Rs in it, and we input another 10 Rs into it), the necessary change is signified by two LEDs, one for 5 Rs change and the other for 10 Rs change. The vending machine we designed is also ideal: for example, if we input 10 Rs into the machine and don't wish to transact further, the machine returns us the 10 Rs in the form of change by displaying it in the output. In this way, transactions are done.

4 Characteristic Equations

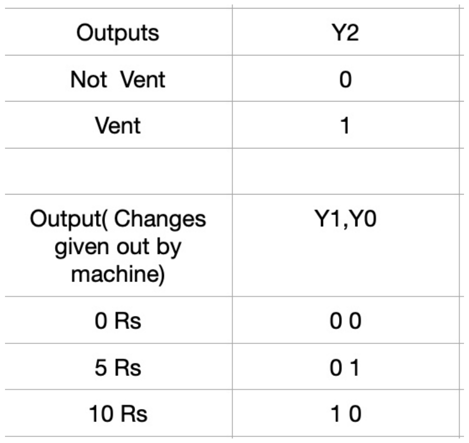
Two-bit inputs are given into the system. The inputs are of three types, 0 Rs, 5 Rs and 10 Rs, as mentioned below:



Deconding	
Inputs	X1,X0
0 Rs	0 0
5 Rs	0 1
10 Rs	1 0

Figure 2: Inputs

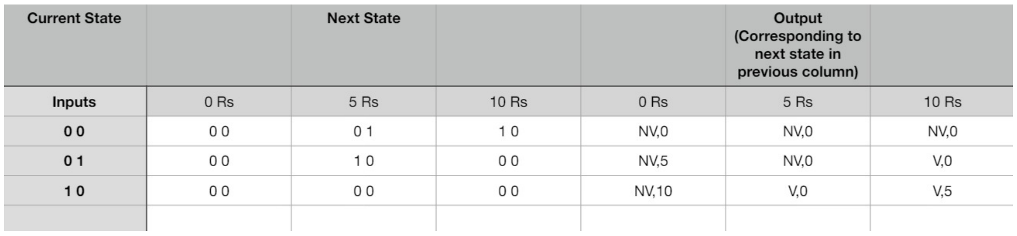
Three-bit outputs are received. One output signifies when the transaction is complete. The other two outputs signify the change received from the machine (5 Rs or 10 Rs). They are shown as follows:



Outputs	Y2
Not Vent	0
Vent	1
Output(Changes given out by machine)	Y1,Y0
0 Rs	0 0
5 Rs	0 1
10 Rs	1 0

Figure 3: Outputs

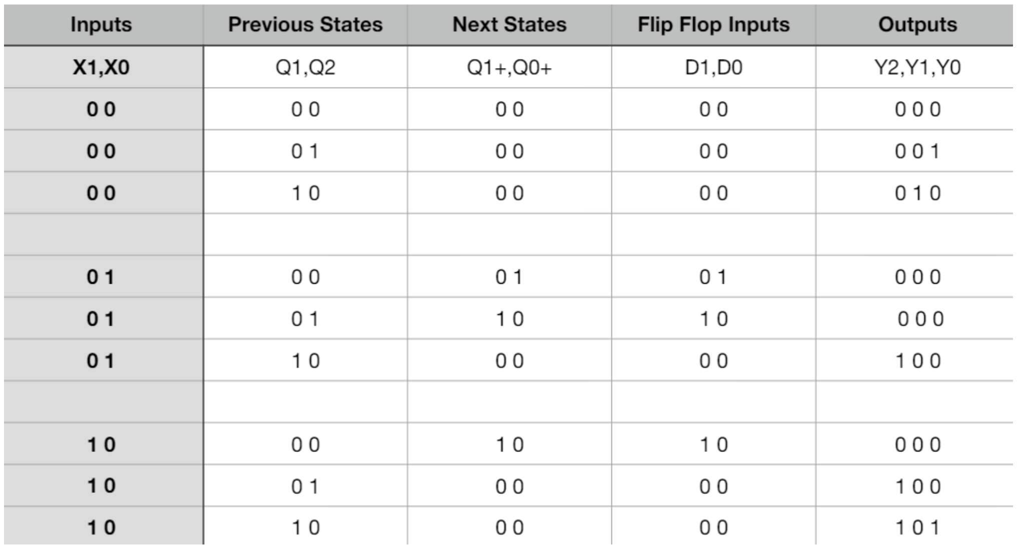
In accordance with the inputs and the outputs, the state excitation table can be given as follows:



Current State		Next State			Output (Corresponding to next state in previous column)	
Inputs	0 Rs	5 Rs	10 Rs	0 Rs	5 Rs	10 Rs
0 0	0 0	0 1	1 0	NV,0	NV,0	NV,0
0 1	0 0	1 0	0 0	NV,5	NV,0	V,0
1 0	0 0	0 0	0 0	NV,10	V,0	V,5

Figure 4: State Excitation Table

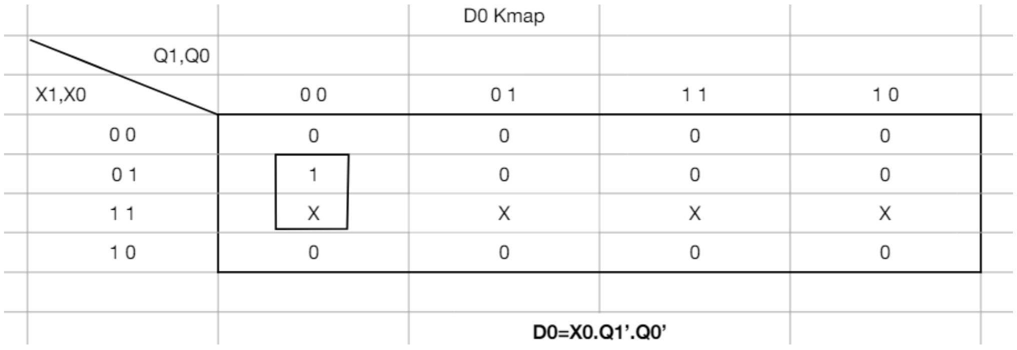
For this project, our flip-flop choice is the D flip-flop. The conversion to D flip-flop of the Mealy FSM is shown as follows:



Inputs	Previous States	Next States	Flip Flop Inputs	Outputs
X1,X0	Q1,Q2	Q1+,Q0+	D1,D0	Y2,Y1,Y0
0 0	0 0	0 0	0 0	0 0 0
0 0	0 1	0 0	0 0	0 0 1
0 0	1 0	0 0	0 0	0 1 0
0 1	0 0	0 1	0 1	0 0 0
0 1	0 1	1 0	1 0	0 0 0
0 1	1 0	0 0	0 0	1 0 0
1 0	0 0	1 0	1 0	0 0 0
1 0	0 1	0 0	0 0	1 0 0
1 0	1 0	0 0	0 0	1 0 1

Figure 5: Conversion to D Flip-Flop

We now write the characteristic equations for D0, D1, Y0, Y1 and Y2, respectively. The K-Maps and their characteristic equations are as follows:

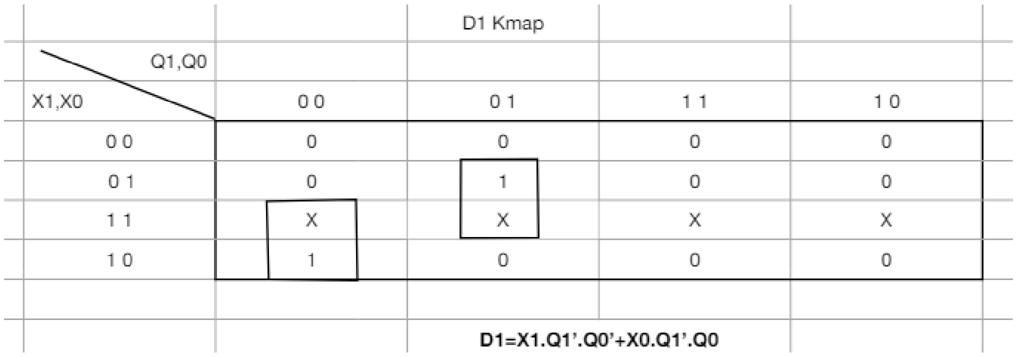


D0 Kmap

X1,X0 \ Q1,Q0	0 0	0 1	1 1	1 0
0 0	0	0	0	0
0 1	1	0	0	0
1 1	X	X	X	X
1 0	0	0	0	0

D0=X0.Q1'.Q0'

Figure 6: K-Map for D0

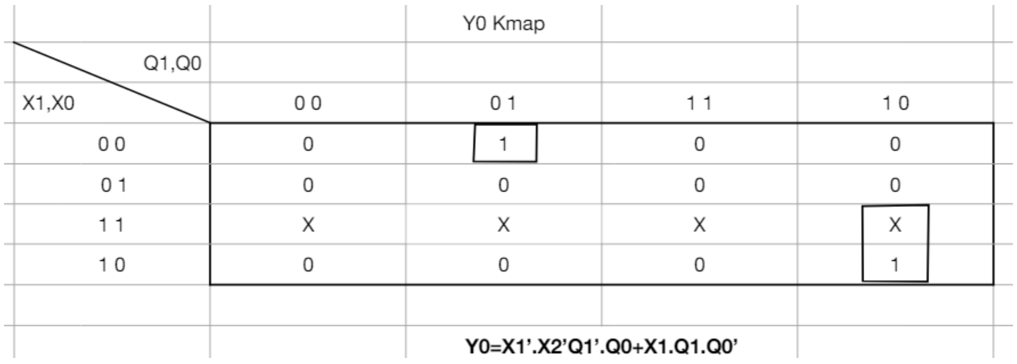


D1 Kmap

X1,X0 \ Q1,Q0	0 0	0 1	1 1	1 0
0 0	0	0	0	0
0 1	0	1	0	0
1 1	X	X	X	X
1 0	1	0	0	0

D1=X1.Q1'.Q0'+X0.Q1'.Q0

Figure 7: K-Map for D1

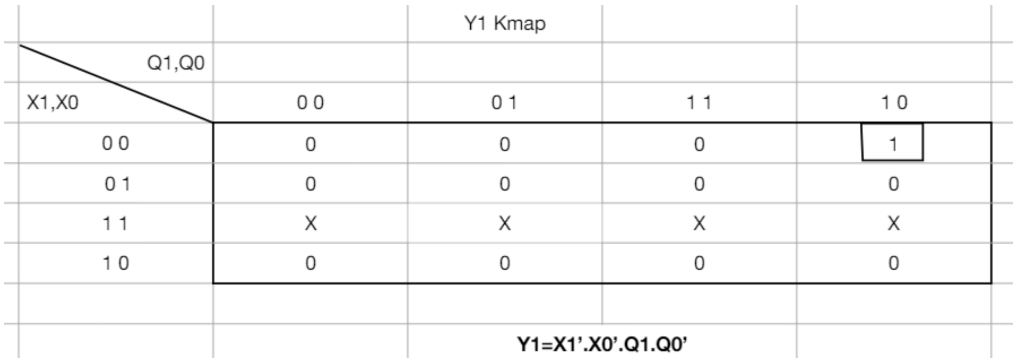


Y0 Kmap

X1,X0 \ Q1,Q0	0 0	0 1	1 1	1 0
0 0	0	1	0	0
0 1	0	0	0	0
1 1	X	X	X	X
1 0	0	0	0	1

Y0=X1'.X2'Q1'.Q0+X1.Q1.Q0'

Figure 8: K-Map for Y0

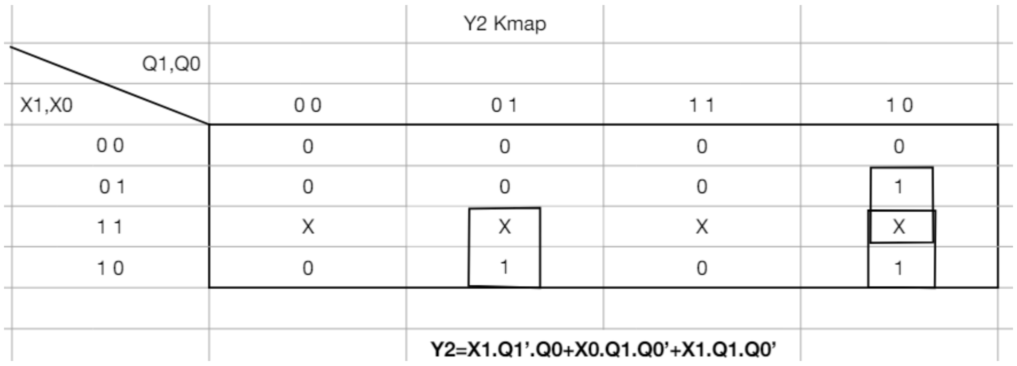


Y1 Kmap

X1,X0 \ Q1,Q0	0 0	0 1	1 1	1 0
0 0	0	0	0	1
0 1	0	0	0	0
1 1	X	X	X	X
1 0	0	0	0	0

Y1=X1'.X0'.Q1.Q0'

Figure 9: K-Map for Y1



Y2 Kmap

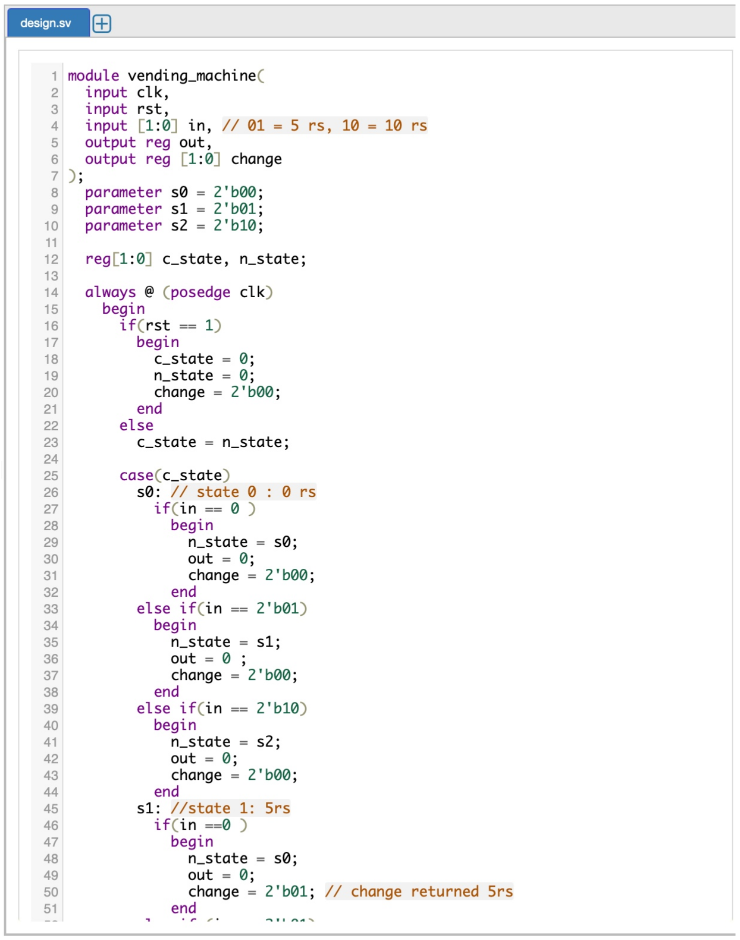
X1,X0 \ Q1,Q0	0 0	0 1	1 1	1 0
0 0	0	0	0	0
0 1	0	0	0	1
1 1	X	X	X	X
1 0	0	1	0	1

Y2=X1.Q1'.Q0+X0.Q1.Q0'+X1.Q1.Q0'

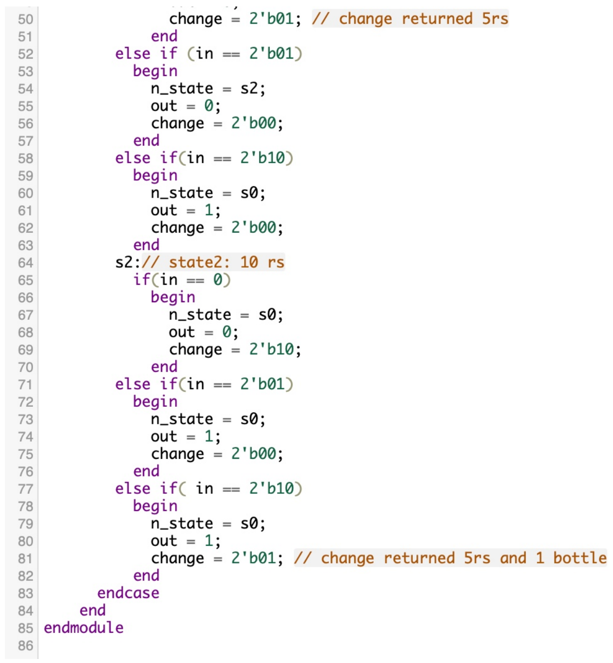
Figure 10: K-Map for Y2

5 Verilog Code and Waveforms

The following is the Verilog Code:


design.sv

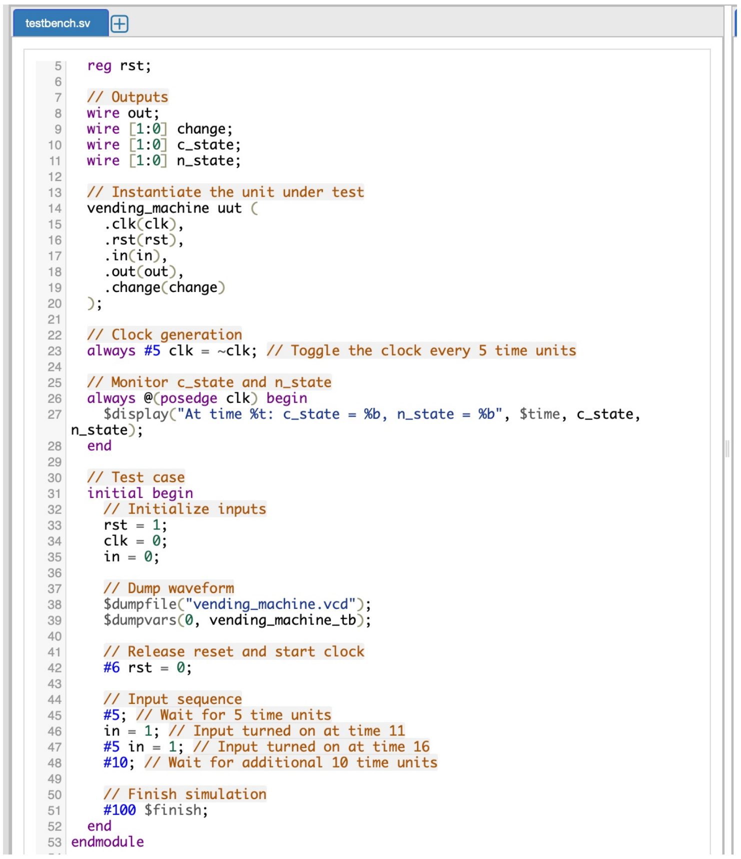
```
module vending_machine(
  input clk,
  input rst,
  input [1:0] in, // 01 = 5 rs, 10 = 10 rs
  output reg out,
  output reg [1:0] change
);
  parameter s0 = 2'b00;
  parameter s1 = 2'b01;
  parameter s2 = 2'b10;

  reg[1:0] c_state, n_state;

  always @ (posedge clk)
    begin
      if(rst == 1)
        begin
          c_state = 0;
          n_state = 0;
          change = 2'b00;
        end
      else
        c_state = n_state;

      case(c_state)
        s0: // state 0 : 0 rs
          if(in == 0 )
            begin
              n_state = s0;
              out = 0;
              change = 2'b00;
            end
        else if(in == 2'b01)
          begin
            n_state = s1;
            out = 0 ;
            change = 2'b00;
          end
        else if(in == 2'b10)
          begin
            n_state = s2;
            out = 0;
            change = 2'b00;
          end
        s1: //state 1: 5rs
          if(in ==0 )
            begin
              n_state = s0;
              out = 0;
              change = 2'b01; // change returned 5rs
            end
```



```
       change = 2'b01; // change returned 5rs
      end
    else if (in == 2'b01)
     begin
      n_state = s2;
      out = 0;
      change = 2'b00;
     end
    else if(in == 2'b10)
     begin
      n_state = s0;
      out = 1;
      change = 2'b00;
     end
    s2:// state2: 10 rs
     if(in == 0)
      begin
       n_state = s0;
       out = 0;
       change = 2'b10;
      end
    else if(in == 2'b01)
     begin
      n_state = s0;
      out = 1;
      change = 2'b00;
     end
    else if( in == 2'b10)
     begin
      n_state = s0;
      out = 1;
      change = 2'b01; // change returned 5rs and 1 bottle
     end
   endcase
  end
endmodule
```

We make a module vending machine and declare our variables there. Our input and outputs are decoded as used in the characteristic equation. Out is for y2, change[0]=y1 and change[0]=y0. Three parameters are declared, denoting the states. Then, the current state and next state are stored in some variables. Then, a loop begins for every positive clock edge. According to the current state of the Mealy machine, s0, s1 or s2 cases are defined. Inside the cases, according to the state transition diagram, if-else conditions are used to define the transitions of the machine from one state to another.

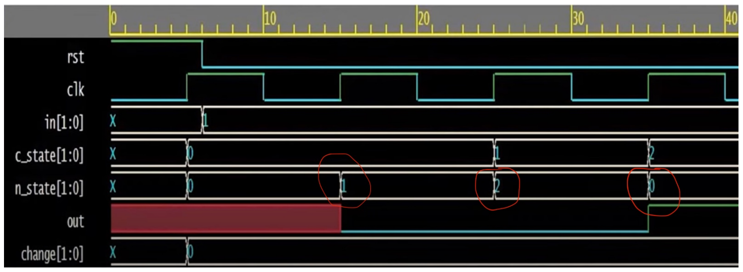
The testbench used is as follows:



```
  reg rst;

  // Outputs
  wire out;
  wire [1:0] change;
  wire [1:0] c_state;
  wire [1:0] n_state;

  // Instantiate the unit under test
  vending_machine uut (
    .clk(clk),
    .rst(rst),
    .in(in),
    .out(out),
    .change(change)
  );

  // Clock generation
  always #5 clk = ~clk; // Toggle the clock every 5 time units

  // Monitor c_state and n_state
  always @(posedge clk) begin
    $display("At time %t: c_state = %b, n_state = %b", $time, c_state,
n_state);
  end

  // Test case
  initial begin
    // Initialize inputs
    rst = 1;
    clk = 0;
    in = 0;

    // Dump waveform
    $dumpfile("vending_machine.vcd");
    $dumpvars(0, vending_machine_tb);

    // Release reset and start clock
    #6 rst = 0;

    // Input sequence
    #5; // Wait for 5 time units
    in = 1; // Input turned on at time 11
    #5 in = 1; // Input turned on at time 16
    #10; // Wait for additional 10 time units

    // Finish simulation
    #100 $finish;
  end
endmodule
```

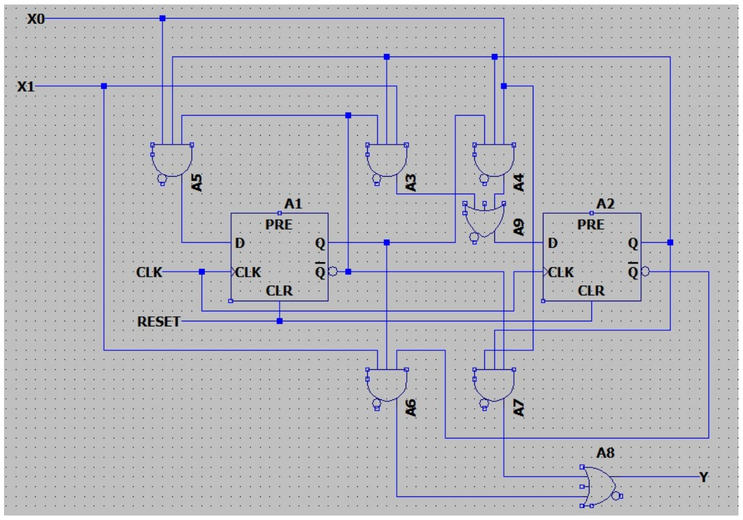
First, we initialise the inputs and output ports needed. Then, the module for the testbench is defined. Inside the module, reset is set to 1 at 0 time so that the garbage values are erased. At the 6 time, the unit reset is turned off. The input of y1 = 0, y0 = 1, i.e. 5 Rs, is given. Then, at intervals of 5 times, a unit input of 0.1 (5 Rs) is again given twice. One at 16 sec and the other at 25 seconds. So, This represents the case where 5 Rs is given 3 times. So, the transaction is complete. Out for venting machine gives output 1, and no changes are returned. This represents that the machine goes from state s0 to s2 to s3 and finally to s0. After that, the simulation is stopped.

The waveform obtained is as follows:



You can see that when the machine's state was changed, we added red circles. The n_state variable is changed from 0 to 1 and then 3. At 3, the transition is complete. So the out wave becomes 1. The change waves remain zero all the time as no changes are given out in our example.

The following is the circuit diagram for our Verilog code. Please note that only the Y1 output is shown due to the complexity issue of our diagram (the diagram was too large in size).



6 Conclusion

In this way, our project for a vending machine was modelled using an FSM and realised on a CPLD board using Verilog. The K-Maps and characteristic equations were made. The Verilog Code, its implementation, the waveform received and the circuit diagram are explained.