LAB 08

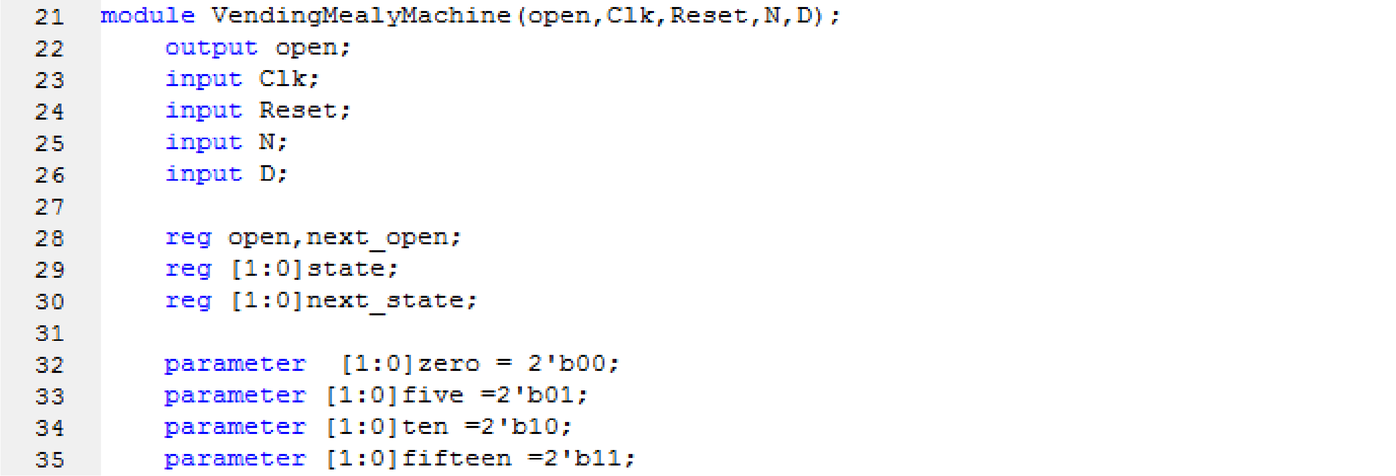
OBJECTIVE

Develop, implement and Simulate Mealy Machine using behavioral modeling.

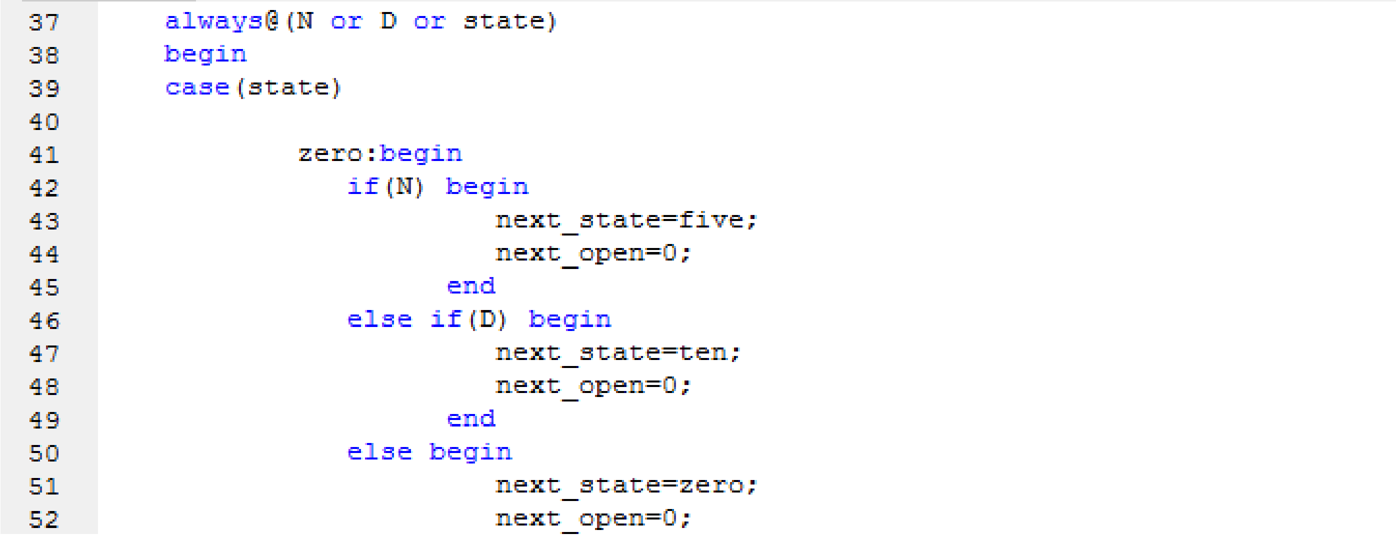
Lab Task 01:

To design, implement and simulate vending machine using mealy machine concept.

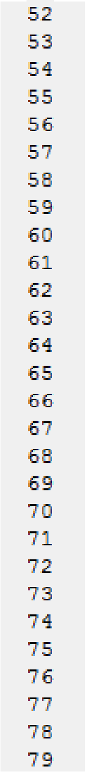
Verilog Module:



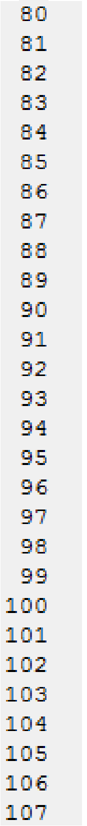
```
module VendingMealyMachine(open,Clk,Reset,N,D);
    output open;
    input Clk;
    input Reset;
    input N;
    input D;

    reg open,next_open;
    reg [1:0]state;
    reg [1:0]next_state;

    parameter  [1:0]zero = 2'b00;
    parameter [1:0]five =2'b01;
    parameter [1:0]ten =2'b10;
    parameter [1:0]fifteen =2'b11;
```



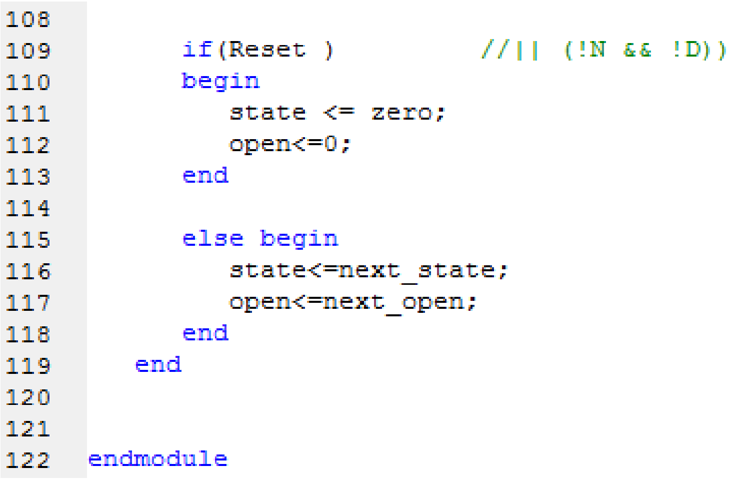
```
    always@(N or D or state)
    begin
    case(state)

            zero:begin
               if(N) begin
                        next_state=five;
                        next_open=0;
                     end
               else if(D) begin
                        next_state=ten;
                        next_open=0;
                     end
               else begin
                        next_state=zero;
                        next_open=0;
```



```
                     next_open=0;
                  end
             end

         five:begin
            if(N) begin
                     next_state=ten;
                     next_open=0;
                  end
            else if(D) begin
                     next_state=fifteen;
                     next_open=1;
                  end
            else begin
                     next_state=five;
                     next_open=0;
                  end
             end

         ten:begin
            if(N) begin
                     next_state=fifteen;
                     next_open=1;
                  end
            else if(D) begin
                     next_state=fifteen;
                     next_open=1;
                  end
```

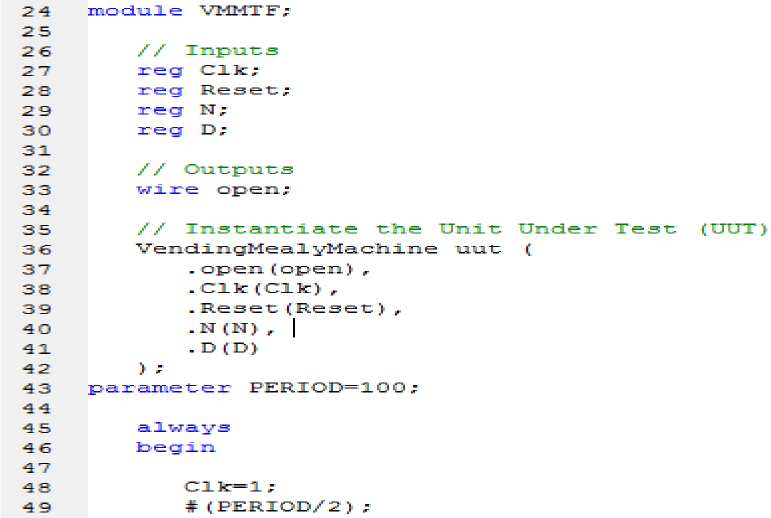


```
            else begin
                     next_state=ten;
                     next_open=0;
                  end
             end

         fifteen:begin

            if(N) begin
                     next_state=five;
                     next_open=0;
                  end
            else if(D) begin
                     next_state=ten;
                     next_open=0;
                  end
            else begin
                     next_state=0;
                     next_open=0;
                  end
             end

endcase
end


always @(posedge Clk)
begin
```

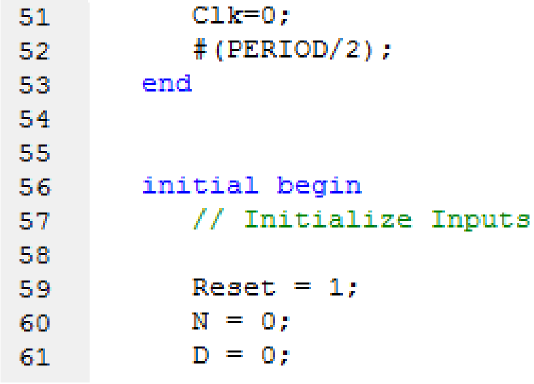


```

      if(Reset )          //|| (!N && !D))
      begin
         state <= zero;
         open<=0;
      end

      else begin
         state<=next_state;
         open<=next_open;
      end
   end


endmodule
```

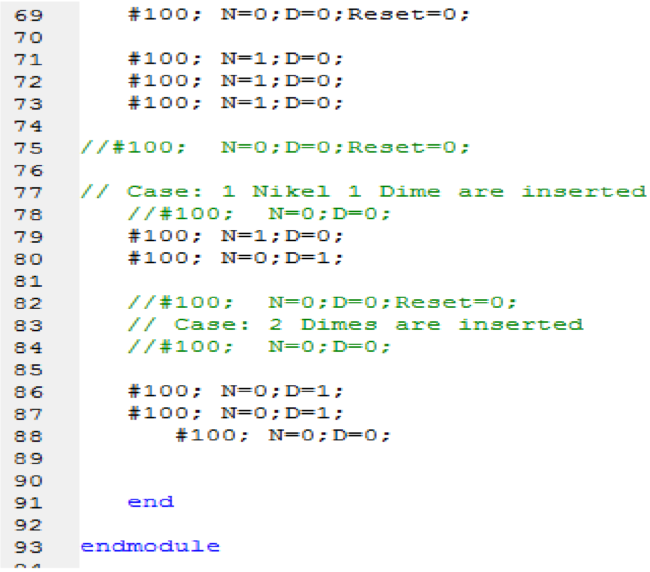
Verilog Test Fixture:



```
module VMMTF;

   // Inputs
   reg Clk;
   reg Reset;
   reg N;
   reg D;

   // Outputs
   wire open;

   // Instantiate the Unit Under Test (UUT)
   VendingMealyMachine uut (
      .open(open),
      .Clk(Clk),
      .Reset(Reset),
      .N(N),
      .D(D)
   );
parameter PERIOD=100;

   always
   begin

      Clk=1;
      #(PERIOD/2);
```

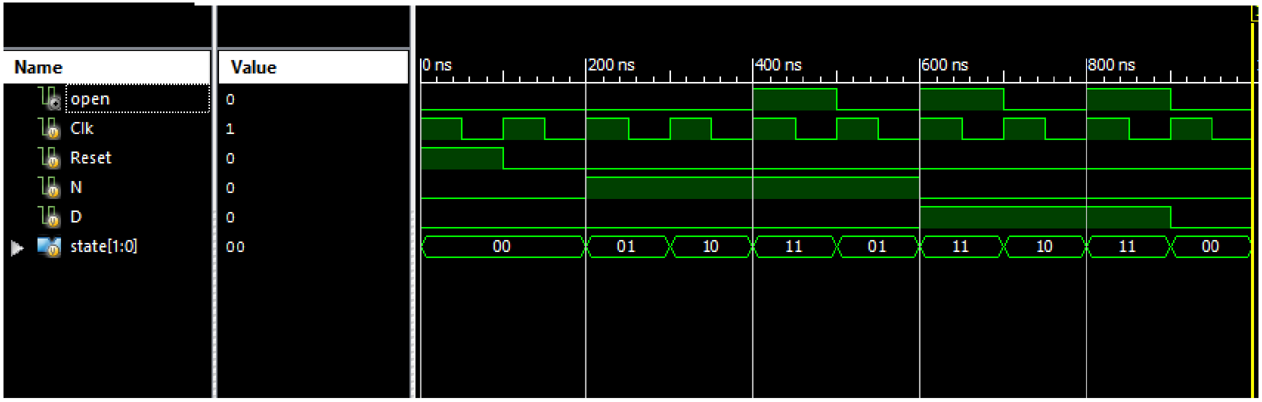


```
      Clk=0;
      #(PERIOD/2);
   end


   initial begin
      // Initialize Inputs

      Reset = 1;
      N = 0;
      D = 0;
```



```
   #100;  N=0;D=0;Reset=0;

   #100;  N=1;D=0;
   #100;  N=1;D=0;
   #100;  N=1;D=0;

//#100;   N=0;D=0;Reset=0;

// Case: 1 Nikel 1 Dime are inserted
   //#100;   N=0;D=0;
   #100;  N=1;D=0;
   #100;  N=0;D=1;

   //#100;   N=0;D=0;Reset=0;
   // Case: 2 Dimes are inserted
   //#100;   N=0;D=0;

   #100;  N=0;D=1;
   #100;  N=0;D=1;
      #100;  N=0;D=0;


   end

endmodule
```

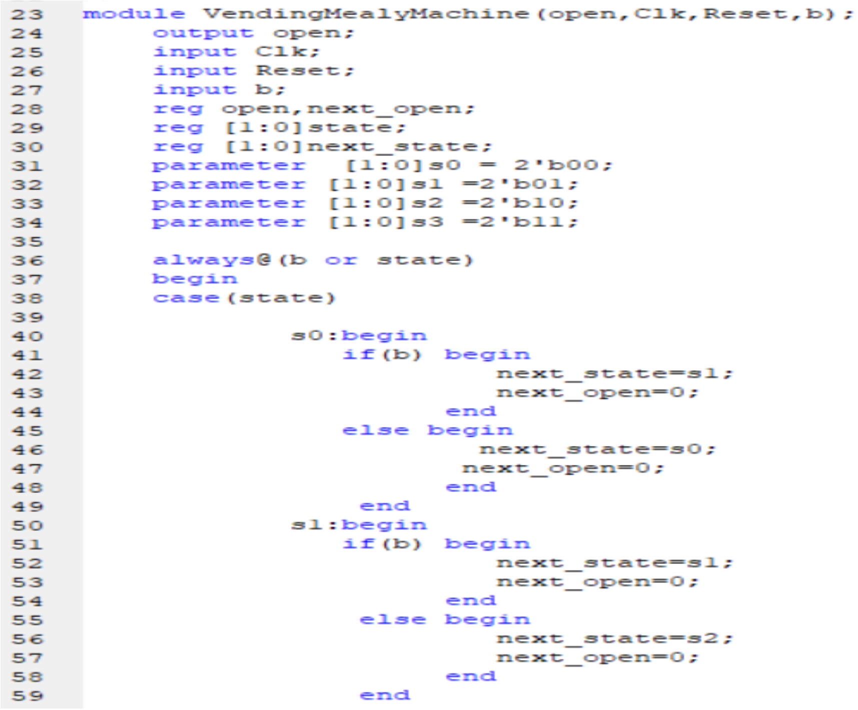
WAVEFORM:



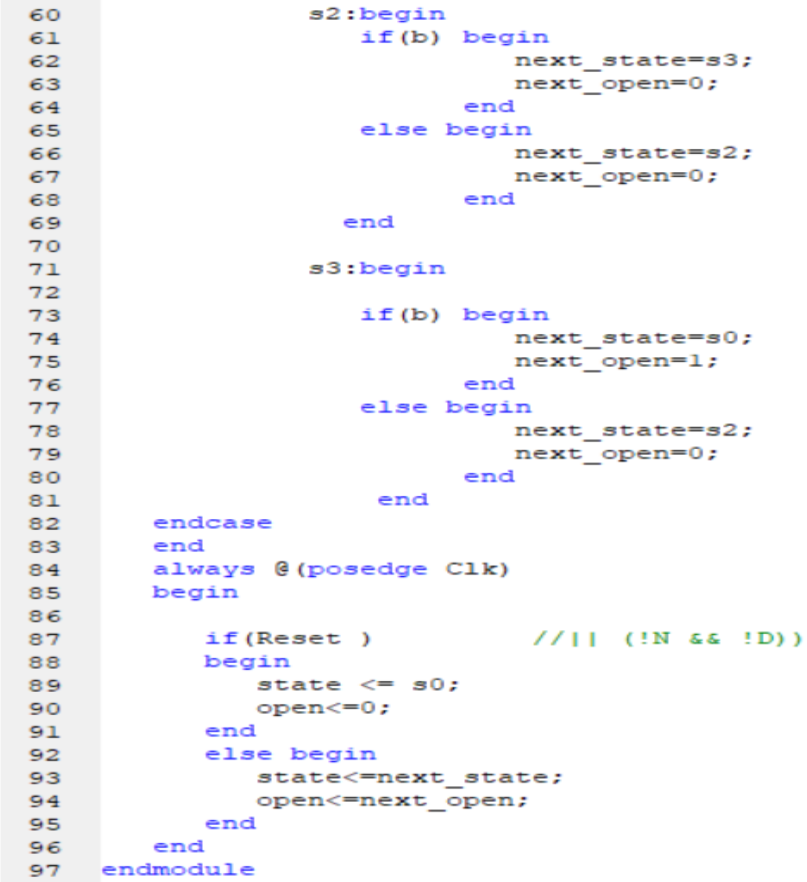
Lab Task 02 / Home Assignment:

To design, implement and simulate sequence detector using mealy machine concept.

Verilog Module:

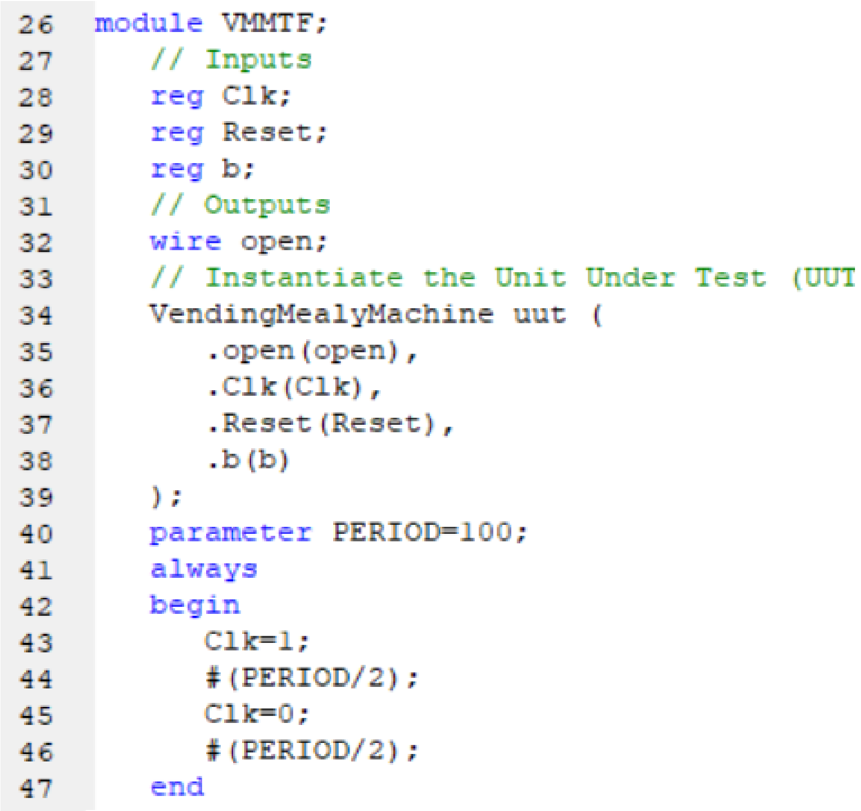


```
module VendingMealyMachine(open,Clk,Reset,b);
    output open;
    input Clk;
    input Reset;
    input b;
    reg open,next_open;
    reg [1:0]state;
    reg [1:0]next_state;
    parameter  [1:0]s0 = 2'b00;
    parameter [1:0]s1 =2'b01;
    parameter [1:0]s2 =2'b10;
    parameter [1:0]s3 =2'b11;

    always@(b or state)
    begin
    case(state)

            s0:begin
               if(b) begin
                        next_state=s1;
                        next_open=0;
                     end
               else begin
                       next_state=s0;
                      next_open=0;
                     end
                end
            s1:begin
               if(b) begin
                        next_state=s1;
                        next_open=0;
                     end
                else begin
                        next_state=s2;
                        next_open=0;
                     end
                end
```

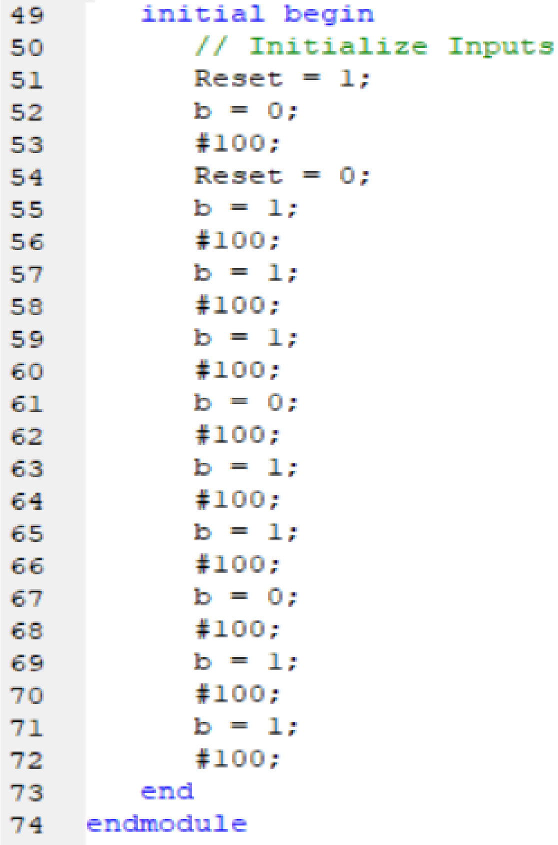


```
            s2:begin
               if(b) begin
                        next_state=s3;
                        next_open=0;
                     end
               else begin
                        next_state=s2;
                        next_open=0;
                     end
              end

            s3:begin

               if(b) begin
                        next_state=s0;
                        next_open=1;
                     end
               else begin
                        next_state=s2;
                        next_open=0;
                     end
                end
    endcase
    end
    always @(posedge Clk)
    begin

       if(Reset )          //|| (!N && !D))
       begin
          state <= s0;
          open<=0;
       end
       else begin
          state<=next_state;
          open<=next_open;
       end
    end
endmodule
```

Verilog Test Fixture:

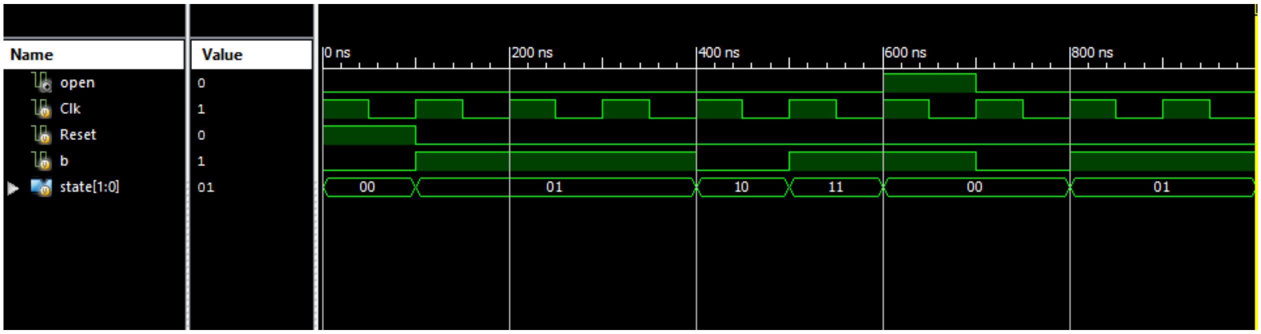


```
module VMMTF;
   // Inputs
   reg Clk;
   reg Reset;
   reg b;
   // Outputs
   wire open;
   // Instantiate the Unit Under Test (UUT
   VendingMealyMachine uut (
      .open(open),
      .Clk(Clk),
      .Reset(Reset),
      .b(b)
   );
   parameter PERIOD=100;
   always
   begin
      Clk=1;
      #(PERIOD/2);
      Clk=0;
      #(PERIOD/2);
   end
```



```
   initial begin
      // Initialize Inputs
      Reset = 1;
      b = 0;
      #100;
      Reset = 0;
      b = 1;
      #100;
      b = 1;
      #100;
      b = 1;
      #100;
      b = 0;
      #100;
      b = 1;
      #100;
      b = 1;
      #100;
      b = 0;
      #100;
      b = 1;
      #100;
      b = 1;
      #100;
   end
endmodule
```

<u>WAVEFORM:</u>



CONCLUSION:

In this lab, we learned how to develop Vending Machine and also develop sequence detector using Mealy machine. Using the ISE Design Suite, We also produced a wave output in which we can see the characteristics of the output after performing these tasks.

Implementing a Mealy machine can be a useful tool in designing and implementing a finite state machine for a specific application. It allows for the output of the machine to be dependent on both the current input and current state, which can be beneficial in certain situations where the output needs to be generated in real-time based on the current input. However, it is important to carefully consider the specific requirements and constraints of the application before deciding whether a Mealy machine is the appropriate solution. It may be necessary to evaluate the trade-offs and limitations of using a Mealy machine, such as the potential for increased complexity in the design and implementation, compared to other types of finite state machines. Overall, the decision to use a Mealy machine should be based on a thorough analysis of the specific needs of the application and the resources available for implementation.