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:

```
21  module VendingMealyMachine(open,Clk,Reset,N,D);
22      output open;
23      input Clk;
24      input Reset;
25      input N;
26      input D;
27
28      reg open,next_open;
29      reg [1:0]state;
30      reg [1:0]next_state;
31
32      parameter  [1:0]zero = 2'b00;
33      parameter [1:0]five =2'b01;
34      parameter [1:0]ten =2'b10;
35      parameter [1:0]fifteen =2'b11;
37      always@(N or D or state)
38      begin
39      case(state)
40
41              zero:begin
42                 if(N) begin
43                          next_state=five;
44                          next_open=0;
45                       end
46                 else if(D) begin
47                          next_state=ten;
48                          next_open=0;
49                       end
50                 else begin
51                          next_state=zero;
52                          next_open=0;
```

```
52                          next_open=0;
53                      end
54                  end
55
56              five:begin
57                  if(N) begin
58                          next_state=ten;
59                          next_open=0;
60                      end
61                  else if(D) begin
62                          next_state=fifteen;
63                          next_open=1;
64                      end
65                  else begin
66                          next_state=five;
67                          next_open=0;
68                      end
69                  end
70
71              ten:begin
72                  if(N) begin
73                          next_state=fifteen;
74                          next_open=1;
75                      end
76                  else if(D) begin
77                          next_state=fifteen;
78                          next_open=1;
79                      end
80                  else begin
81                           next_state=ten;
82                           next_open=0;
83                       end
84                   end
85
86              fifteen:begin
87
88                  if(N) begin
89                           next_state=five;
90                           next_open=0;
91                       end
92                  else if(D) begin
93                           next_state=ten;
94                           next_open=0;
95                       end
96                  else begin
97                           next_state=0;
98                           next_open=0;
99                       end
100                  end
101
102     endcase
103     end
104
105
106     always @(posedge Clk)
107     begin
```

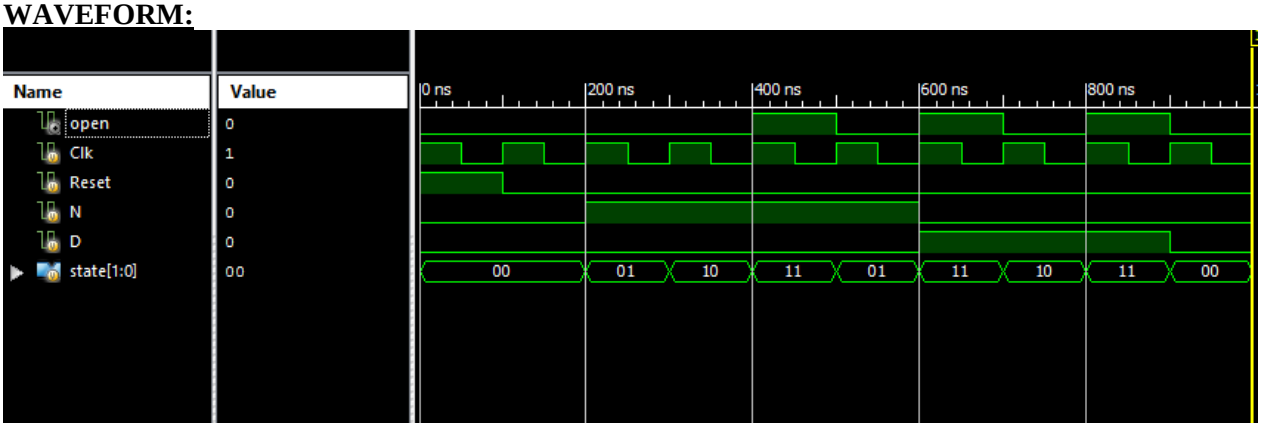
```
108
109         if(Reset )          //|| (!N && !D))
110         begin
111            state <= zero;
112            open<=0;
113         end
114
115         else begin
116            state<=next_state;
117            open<=next_open;
118         end
119      end
120
121
122   endmodule
```

Verilog Test Fixture:

```
24    module VMMTF;
25
26       // Inputs
27       reg Clk;
28       reg Reset;
29       reg N;
30       reg D;
31
32       // Outputs
33       wire open;
34
35       // Instantiate the Unit Under Test (UUT)
36       VendingMealyMachine uut (
37          .open(open),
38          .Clk(Clk),
39          .Reset(Reset),
40          .N(N),
41          .D(D)
42       );
43    parameter PERIOD=100;
44
45       always
46       begin
47
48          Clk=1;
49          #(PERIOD/2);
51          Clk=0;
52          #(PERIOD/2);
53       end
54
55
56       initial begin
57          // Initialize Inputs
58
59          Reset = 1;
60          N = 0;
61          D = 0;
```

```
69      #100; N=0;D=0;Reset=0;
70
71      #100; N=1;D=0;
72      #100; N=1;D=0;
73      #100; N=1;D=0;
74
75  //#100;   N=0;D=0;Reset=0;
76
77  // Case: 1 Nikel 1 Dime are inserted
78      //#100;  N=0;D=0;
79      #100; N=1;D=0;
80      #100; N=0;D=1;
81
82      //#100;  N=0;D=0;Reset=0;
83      // Case: 2 Dimes are inserted
84      //#100;  N=0;D=0;
85
86      #100; N=0;D=1;
87      #100; N=0;D=1;
88          #100; N=0;D=0;
89
90
91      end
92
93  endmodule
```

WAVEFORM:



Lab Task 02 / Home Assignment:

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

Verilog Module:

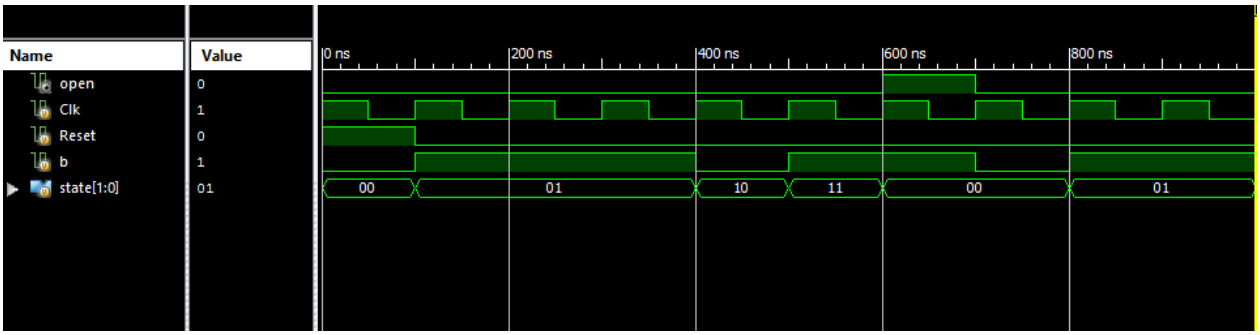
```
23  module VendingMealyMachine(open,Clk,Reset,b);
24      output open;
25      input Clk;
26      input Reset;
27      input b;
28      reg open,next_open;
29      reg [1:0]state;
30      reg [1:0]next_state;
31      parameter   [1:0]s0 = 2'b00;
32      parameter [1:0]s1 =2'b01;
33      parameter [1:0]s2 =2'b10;
34      parameter [1:0]s3 =2'b11;
35
36      always@(b or state)
37      begin
38      case(state)
39
40                s0:begin
41                   if(b) begin
42                             next_state=s1;
43                             next_open=0;
44                         end
45                   else begin
46                            next_state=s0;
47                           next_open=0;
48                         end
49                    end
50                s1:begin
51                   if(b) begin
52                             next_state=s1;
53                             next_open=0;
54                         end
55                    else begin
56                             next_state=s2;
57                             next_open=0;
58                         end
59                    end
```

```
60                s2:begin
61                   if(b) begin
62                             next_state=s3;
63                             next_open=0;
64                         end
65                   else begin
66                             next_state=s2;
67                             next_open=0;
68                         end
69                  end
70
71                s3:begin
72
73                   if(b) begin
74                             next_state=s0;
75                             next_open=1;
76                         end
77                   else begin
78                             next_state=s2;
79                             next_open=0;
80                         end
81                    end
82      endcase
83      end
84      always @(posedge Clk)
85      begin
86
87          if(Reset )          //|| (!N && !D))
88          begin
89              state <= s0;
90              open<=0;
91          end
92          else begin
93              state<=next_state;
94              open<=next_open;
95          end
96      end
97  endmodule
```

Verilog Test Fixture:

```
26  module VMMTF;
27      // Inputs
28      reg Clk;
29      reg Reset;
30      reg b;
31      // Outputs
32      wire open;
33      // Instantiate the Unit Under Test (UUT
34      VendingMealyMachine uut (
35          .open(open),
36          .Clk(Clk),
37          .Reset(Reset),
38          .b(b)
39      );
40      parameter PERIOD=100;
41      always
42      begin
43          Clk=1;
44          #(PERIOD/2);
45          Clk=0;
46          #(PERIOD/2);
47      end
49     initial begin
50         // Initialize Inputs
51         Reset = 1;
52         b = 0;
53         #100;
54         Reset = 0;
55         b = 1;
56         #100;
57         b = 1;
58         #100;
59         b = 1;
60         #100;
61         b = 0;
62         #100;
63         b = 1;
64         #100;
65         b = 1;
66         #100;
67         b = 0;
68         #100;
69         b = 1;
70         #100;
71         b = 1;
72         #100;
73     end
74  endmodule
```

WAVEFORM:



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.