

(1) Write the behavioral description of a Moore FSM described by Figure 1.

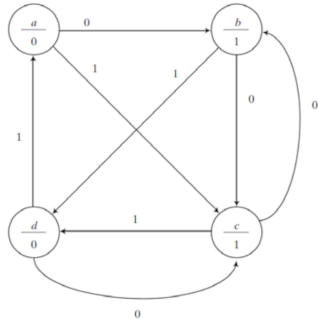


Figure 1. A Moore FSM.

我將a,b,c,d分別設為parameter 00,01,10,11 一開始當input為0時, state會跑到b,output為1...以此類推, 此種machine的输出不受input影響。

code:

```
2  `timescale 1 ns/100ps
3  //moore
4  module moore(in, rst_n, clk, out);
5
6  input in;
7  input rst_n, clk;
8  output out;
9
10
11  reg[1:0] curr_state, next_state;
12  reg out;
13  parameter a = 2'b00, b = 2'b01, c = 2'b10, d = 2'b11;
14
15
16  always@(posedge clk or negedge rst_n) begin
17  if (~rst_n)
18  curr_state <= #1 a;
19  else
20  curr_state <= #1 next_state;
21  end
22
23  always @(curr_state or in) // next state logic
24  begin
25  case (curr_state)
26  a : next_state <= (in == 1'b1) ? c : b;
27  b : next_state <= (in == 1'b1) ? d : c;
28  c : next_state <= (in == 1'b1) ? d : b;
29  d : next_state <= (in == 1'b1) ? a : c;
30  default : next_state = a;
31  endcase
32  end
33
34  always @(curr_state) // output logic
35  begin
36  case (curr_state)
37  a : out = 1'b0;
38  b : out = 1'b1;
39  c : out = 1'b1;
40  d : out = 1'b0;
41  default : out = 1'b0;
42  endcase
43  end
44  endmodule
45
```

(2) Write the behavioral description of a Mealy FSM described by Figure 2

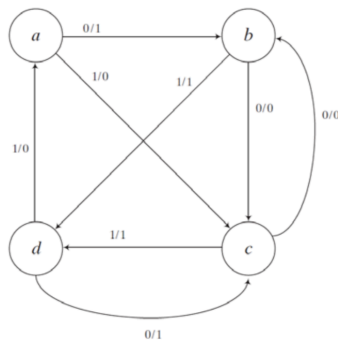


Figure 2. A Mealy FSM.

我將a,b,c,d分別設為parameter 00,01,10,11 一開始當input為1時, state會跑到c, output為0...以此類推, 此種machine的输出受input影響。

code:

```

47 ///////////////////////////////////////////////////
48 `timescale 1 ns/100ps
49 //moore
50
51 module mealy(in, rst_n, clk, out);
52
53   input in;
54   input rst_n, clk;
55   output out;
56
57   reg out, in2;
58   reg[1:0] curr_state, next_state;
59   parameter a = 2'b00, b = 2'b01, c = 2'b10, d = 2'b11;
60
61
62   always@(posedge clk or negedge rst_n) begin
63     if(curr_state == a)
64       in2 = 0;
65     if(curr_state == b)
66       in2 = 0;
67     if(curr_state == c)
68       in2 = 1;
69     if(curr_state == d)
70       in2 = 1;
71     if (~rst_n)
72       curr_state <= #1 a;
73     else
74       curr_state <= #1 next_state;
75   end
76
77   always @(curr_state or in) // next state logic
78   begin
79     case (curr_state)
80       a : next_state <= (in == 1'b1) ? c : b;
81       b : next_state <= (in == 1'b1) ? d : c;
82       c : next_state <= (in == 1'b1) ? d : b;
83       d : next_state <= (in == 1'b1) ? a : c;
84     default : next_state = a;
85   endcase
86   end
87
88   always @(curr_state or in2 or in) // output logic
89   begin
90     case (curr_state)
91       a : out = 0;
92       b : out = ~in2;
93       c : out = in2;
94       d : out = 1;
95     default : out = 0;
96   endcase
97   end
98 endmodule
99

```

test bench:

```

100 //////////////testbench//////////
101 module machine_TB();
102     reg rst_n, clk, in;
103     wire out_moore, out_mealy;
104
105     //
106     moore mo(in, rst_n, clk, out_moore);
107     mealy me(in, rst_n, clk, out_mealy);
108     //
109
110     initial
111     begin
112         clk = 0;
113         repeat (30)
114             #100 clk = ~clk;
115     end
116     initial
117     begin
118         in = 0; rst_n = 0; // a
119         #100 in = 0; rst_n = 1; //b
120         #100 in = 1; rst_n = 1; //c
121         #100 in = 0; rst_n = 1; //b
122         #100 in = 1; rst_n = 1;
123         #100 in = 0; rst_n = 1;
124         #100 in = 1; rst_n = 1;
125         #100 in = 1; rst_n = 1;
126         #100 in = 0; rst_n = 1;
127         #100 in = 0; rst_n = 1;
128     end
129     initial #2000 $finish;
130     initial $dumpvars;
131 endmodule
132

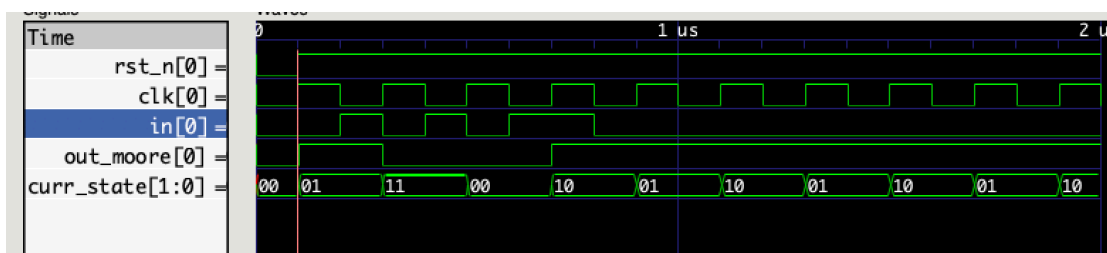
```

waveform:

實作模擬如下圖

### 1. Moore FSM

此 positive edge trigger 一開始透過 rst\_n 講 state initiate 為 a(00)，我們可以透過 curr\_state 來查看 state 的跑動和 input 的關聯。



### 2. Mealy FSM

此 positive edge trigger 一開始透過 rst\_n 講 state initiate 為 a(00)，我們可以透過 curr\_state 來查看 state 的跑動,output 和 input 的關聯。實作結果中，也可以看到 glitch 的發生，因為我有用一個 in2 register 來計算 ouput 的值，而造成一些 delay。

