

```

1  /*-----
2  Name Lamin Jammeh
3  Class: EE417 Summer 2024
4  Lesson 04 HW Question 1
5  Group: Ron Kalin/ Lamin Jammeh
6  Project Description: This is the main module for a Mealy FSM that receives a
7  set of bits and checks if they are equal to Z1=010 or Z2=100 and takes a decision
8  on whether to reset the machine or keep looking for receiving data and comparing them
9  -----*/
10 module Sequence_100_Detector_Mealy (z2, z1, clk, S_reset, x_in);
11
12 //define the inputs and outputs of the system
13 output reg z2, z1;
14 input      clk, S_reset;
15 input      x_in;
16
17 //define the states
18 reg [2:0] present_state, next_state;
19
20 //define the possible parameter
21 parameter S_000 = 3'b000;
22 parameter S_001 = 3'b001;
23 parameter S_010 = 3'b010; //z1
24 parameter S_011 = 3'b011;
25 parameter S_100 = 3'b100; //z2
26 parameter S_101 = 3'b101;
27 parameter S_110 = 3'b110;
28 parameter S_111 = 3'b111;
29
30 //sequential logic updating the state register (flip flop)
31 always @(posedge clk)
32     if (S_reset) present_state <= S_000;
33     else present_state <= next_state;
34
35 //combination logic determining the next_state and the output state
36 always @ * //trigger the block when there is any change in the signals
37     used in the block
38     case (present_state)
39         S_000 : begin
40             z1 = 1'b0;
41             z2 = 1'b0;
42             if (x_in) next_state = S_001;
43             else next_state = S_000;
44         end
45         S_001 : begin
46             z1 = 1'b0;
47             z2 = 1'b0;
48             if (x_in) next_state = S_010;
49             else next_state = S_001;
50         end
51         S_010 : begin
52             z1 = 1'b1;
53             z2 = 1'b0;
54             if (x_in) next_state = S_011;
55             else next_state = S_010;
56         end
57         S_011 : begin
58             z1 = 1'b0;
59             z2 = 1'b0;
60             if (x_in) next_state = S_100;
61             else next_state = S_011;
62         end
63         S_100 : begin
64             z1 = 1'b0;
65             z2 = 1'b1;
66             if (x_in) next_state = S_101;
67             else next_state = S_100;
68         end
69         S_101 : begin

```

```
69         z1 = 1'b0;
70         z2 = 1'b0;
71         if (x_in) next_state = S_110;
72         else      next_state = S_100;
73     end
74     S_110 : begin
75         z1 = 1'b0;
76         z2 = 1'b0;
77         if (x_in) next_state = S_111;
78         else      next_state = S_110;
79     end
80     S_111 : begin
81         z1 = 1'b0;
82         z2 = 1'b0;
83         if (x_in) next_state = S_111;
84         else      next_state = S_111;
85     end
86 endcase
87 endmodule
```