```
1   `timescale 1ns / 1ps
2   //////////////////////////////////////////////////////////////////////////////////
3   // Company:
4   // Engineer:
5   //
6   // Create Date:    15:44:42 05/13/2016
7   // Design Name:
8   // Module Name:    FSM
9   // Project Name:
10  // Target Devices:
11  // Tool versions:
12  // Description:
13  //
14  // Dependencies:
15  //
16  // Revision:
17  // Revision 0.01 - File Created
18  // Additional Comments:
19  //
20  //////////////////////////////////////////////////////////////////////////////////
21  module FSM(
22     input pb0,
23     input pb1,
24     //input pb2,
25     input crash,
26     input sec8,
27     input endGame, //high when sw = round+1
28     input round, //high when ready to go to next round
29     input [4:0] Q,
30
31     output resetTime,
32     output resetScore,
33     output flashAll,
34     output flashSlug,
35     output freezeSlug, //high when we want to freeze frame
36     output expand, //high when in expand phase
37     output victoryD, //high when victory dance
38     output [4:0] D,
39     output initaleyes,
40     output freezeM
41
42      );
43
44     assign D[0] = Q[0]&~pb0 | Q[4]&pb1 | Q[2]&pb1;
45     assign D[1] = Q[1]&~crash&~endGame&~round | Q[0]&pb0 | Q[3]&sec8;
46     assign D[2] = Q[2]&~pb1 | Q[1]&crash&~round&~endGame; //crash state
47     assign D[3] = Q[3]&~sec8 | Q[1]&round&~crash&~endGame;
48     assign D[4] = Q[4]&~pb1 | Q[1]&endGame;
49
50     assign resetTime =  Q[1]&round&~crash;
51     assign resetScore = Q[4]&pb1 | Q[2]&pb1;
52     assign flashAll = Q[4];//Game over, you win
53     assign flashSlug = Q[2] | Q[3];
54     assign freezeSlug = Q[2] | Q[4] | Q[3];
55     assign freezeM = Q[3] | Q[2] | Q[0] | Q[4];
56     assign victoryD = Q[4];
57     assign expand = Q[3];
58     assign initaleyes = Q[0] | Q[4]&pb1 | Q[2]&pb1;
```

```
59
60    endmodule
61
```