```
1   module FSM (clk, reset, in, enter, exit);
2
3      input logic clk, reset;
4      input logic [1:0] in;
5      output logic enter, exit;
6
7      logic [1:0] ps, ns;
8
9      parameter A = 2'b00;
10     parameter B = 2'b10;
11     parameter C = 2'b01;
12     parameter D = 2'b11;
13
14     assign enter = ~in[0] & ~in[1] & ~ps[0] & ps[1];
15     assign exit = ~in[0] & ~in[1] & ps[0] & ~ps[1];
16     assign ns[0] = in[0];
17     assign ns[1] = in[1];
18
19     always_ff @(posedge clk) begin
20        if (reset) ps <= A;
21        else ps <= ns;
22     end
23
24  endmodule
25
26  module FSM_testbench();
27
28     logic clk, reset;
29     logic [1:0] in;
30     logic enter, exit;
31
32     FSM dut (.clk, .reset, .in, .enter, .exit);
33
34     parameter CLOCK_PERIOD = 100;
35     initial clk = 1;
36     always begin
37        #(CLOCK_PERIOD / 2);
38        clk = ~clk;
39     end
40
41     initial begin
42                       @(posedge clk);
43     reset <= 1;       @(posedge clk);
44                       @(posedge clk);
45     reset <= 0;       @(posedge clk);
46                       @(posedge clk);
47           in[0] = 0; in[1] = 0;   @(posedge clk);
48           in[0] = 0; in[1] = 0;   @(posedge clk);
49           in[0] = 1; in[1] = 0;   @(posedge clk);
50           in[0] = 1; in[1] = 0;   @(posedge clk);
51           in[0] = 1; in[1] = 1;   @(posedge clk);
52           in[0] = 1; in[1] = 1;   @(posedge clk);
53           in[0] = 0; in[1] = 1;   @(posedge clk);
54           in[0] = 0; in[1] = 1;   @(posedge clk);
55           in[0] = 0; in[1] = 0;   @(posedge clk);
56           in[0] = 0; in[1] = 0;   @(posedge clk);
57           in[0] = 0; in[1] = 1;   @(posedge clk);
58           in[0] = 0; in[1] = 1;   @(posedge clk);
59           in[0] = 1; in[1] = 1;   @(posedge clk);
60           in[0] = 1; in[1] = 1;   @(posedge clk);
61           in[0] = 1; in[1] = 0;   @(posedge clk);
62           in[0] = 1; in[1] = 0;   @(posedge clk);
63           in[0] = 0; in[1] = 0;   @(posedge clk);
64           in[0] = 0; in[1] = 0;   @(posedge clk);
65        $stop;
66     end
67
68  endmodule
69
```