```
1  //~ `New testbench
2  `timescale  1ns / 1ps
3
4  module tb_fsm;
5
6  // fsm Parameters
7  parameter PERIOD  = 10                    ;
8  parameter FIRST   = 12'b000_011_010_1_0_0;
9  parameter SECOND  = 12'b001_101_100_0_1_0;
10 parameter THIRD   = 12'b010_010_111_1_0_0;
11 parameter FOURTH  = 12'b011_110_011_0_1_0;
12 parameter FIFTH   = 12'b100_101_010_1_0_1;
13
14 // fsm Inputs
15 reg   restart                             = 0 ;
16 reg   clk                                 = 0 ;
17 reg   rst                                 = 0 ;
18 reg   pause                               = 0 ;
19 reg   goto_third                          = 0 ;
20
21 // fsm Outputs
22 wire  [2:0]  out1                          ;
23 wire  [2:0]  out2                          ;
24 wire  even                                ;
25 wire  odd                                 ;
26 wire  terminal                            ;
27
28
29 initial
30 begin
31     forever #(PERIOD/2)  clk=~clk;
32 end
33
34
35
36 fsm #(
37     .FIRST  ( FIRST  ),
38     .SECOND ( SECOND ),
39     .THIRD  ( THIRD  ),
40     .FOURTH ( FOURTH ),
41     .FIFTH  ( FIFTH  ))
42  u_fsm (
43     .restart                ( restart           ),
44     .clk                    ( clk               ),
45     .rst                    ( rst               ),
46     .pause                  ( pause             ),
47     .goto_third             ( goto_third        ),
48
49     .out1                   ( out1        [2:0] ),
50     .out2                   ( out2        [2:0] ),
51     .even                   ( even              ),
52     .odd                    ( odd               ),
53     .terminal               ( terminal          )
54 );
55
56 initial
57 begin
58     rst = 1;
59     #10;
60     rst = 0;
```

```
61      restart = 0;
62      pause = 0;
63      assert (u_fsm.state == FIRST) ;
64      #10;
65      assert (u_fsm.state == SECOND);
66      pause = 1;
67      #30;
68      assert (u_fsm.state == SECOND);
69      pause = 0;
70      #10;
71      assert (u_fsm.state == THIRD);
72      #10;
73      assert (u_fsm.state == FOURTH);
74      restart = 1;
75      #10;
76      restart = 0;
77      assert (u_fsm.state == FIRST);
78      #10;
79      #10;
80      #10;
81      assert (u_fsm.state == FOURTH);
82      #10;
83      assert (u_fsm.state == FIFTH);
84      goto_third = 1;
85      #10;
86      goto_third = 0;
87      assert (u_fsm.state == THIRD);
88      #100;
89      assert (u_fsm.state == FIFTH);
90
91      $stop;
92  end
93
94  endmodule
95
```