```
1  //~ `New testbench
2  `timescale  1ns / 1ps
3
4  module tb_fsm;
5
6  // fsm Parameters
7  parameter PERIOD  = 10   ;
8  parameter FIRST   = 2'b11;
9  parameter SECOND  = 2'b01;
10 parameter THIRD   = 2'b10;
11
12 // fsm Inputs
13 reg   pause                                = 0 ;
14 reg   restart                              = 0 ;
15 reg   clk                                  = 0 ;
16 reg   rst                                  = 0 ;
17
18 // fsm Outputs
19 wire  [1:0]  state                         ;
20 wire  odd                                  ;
21 wire  even                                 ;
22 wire  terminal                             ;
23
24
25 initial
26 begin
27     forever #(PERIOD/2)  clk=~clk;
28 end
29
30 initial
31 begin
32     #(PERIOD*2) rst  =  0;
33 end
34
35 fsm #(
36     .FIRST  ( FIRST  ),
37     .SECOND ( SECOND ),
38     .THIRD  ( THIRD  ))
39  u_fsm (
40     .pause                   ( pause           ),
41     .restart                 ( restart         ),
42     .clk                     ( clk             ),
43     .rst                     ( rst             ),
44
45     .state                   ( state     [1:0] ),
46     .odd                     ( odd             ),
47     .even                    ( even            ),
48     .terminal                ( terminal        )
49 );
50
51 initial
52 begin
53
54     // TESTCASE: recycle FIRST all the time
55     restart = 1;
56     pause = 1;
57     # 30;
58     restart = 1;
59     pause = 0;
60     # 30;
```

```
61      restart = 0;
62      pause = 1;
63      # 30;
64
65      //TESTCASE: start to second
66      restart = 0;
67      pause = 0;
68      # 10;
69      // we should now be at SECOND state
70      pause = 1;
71      #30;
72
73      // TESTCASE: second to third
74      pause = 0;
75      #10;
76      $assert (u_fsm.state == 2'b10);
77
78      #10;
79      $assert (u_fsm.state == 2'b10) ;
80
81      restart = 1;
82      #10;
83      $assert (u_fsm.state == 2'b11) ;
84
85      $stop;
86  end
87
88  endmodule
```