```
//~ `New testbench
`timescale  1ns / 1ps

module tb_fsm;

// fsm Parameters
parameter PERIOD  = 10   ;
parameter FIRST   = 2'b11;
parameter SECOND  = 2'b01;
parameter THIRD   = 2'b10;

// fsm Inputs
reg   pause                                = 0 ;
reg   restart                              = 0 ;
reg   clk                                  = 0 ;
reg   rst                                  = 0 ;

// fsm Outputs
wire  [1:0]  state                         ;
wire  odd                                  ;
wire  even                                 ;
wire  terminal                             ;


initial
begin
    forever #(PERIOD/2)  clk=~clk;
end

initial
begin
    #(PERIOD*2) rst  =  0;
end

fsm #(
    .FIRST  ( FIRST  ),
    .SECOND ( SECOND ),
    .THIRD  ( THIRD  ))
 u_fsm (
    .pause                   ( pause           ),
    .restart                 ( restart         ),
    .clk                     ( clk             ),
    .rst                     ( rst             ),

    .state                   ( state     [1:0] ),
    .odd                     ( odd             ),
    .even                    ( even            ),
    .terminal                ( terminal        )
);

initial
begin

    // TESTCASE: recycle FIRST all the time
    restart = 1;
    pause = 1;
    # 30;
    restart = 1;
    pause = 0;
    # 30;
    restart = 0;
    pause = 1;
    # 30;

    //TESTCASE: start to second
    restart = 0;
    pause = 0;
    # 10;
    // we should now be at SECOND state
    pause = 1;
    #30;

    // TESTCASE: second to third
    pause = 0;
    #10;
    $assert (u_fsm.state == 2'b10);

    #10;
    $assert (u_fsm.state == 2'b10) ;

    restart = 1;
    #10;
    $assert (u_fsm.state == 2'b11) ;

    $stop;
end

endmodule
```