module FSM\_1101(input clk,input rst,input in,output reg out);

parameter s0=0,s1=1,s11=2,s110=3;

reg [1:0] state;

always @(posedge clk)begin

if(rst == 1'b1)

out <= 0;

state <= s0;

begin

case(state)

s0 : begin

state <=in ? s1:s0;

out <= 0;

end

s1 : begin

state <= in ? s11 : s0;

out <= 0;

end

s11 : begin

state <=in ? s11 : s110;

out <= 0;

end

s110 : begin

state <= in ? s0:s0;

out <= 0 ? 1 : 0;

end

default : begin

state <= s0;

out <= 0;

end

endcase

end

end

endmodule

// Code your testbench here

// or browse Examples

module FSM\_1101\_tb();

reg CLK,RST,IN;

wire OUT;

FSM\_1101 dut(

.clk(CLK),

.rst(RST),

.in(IN),

.out(OUT)

);

initial begin

CLK = 1'b0;

RST = 1'b1;

#15 RST = 1'b0;

end

initial forever begin

#5 CLK = ~CLK;

end

initial begin

$monitor ("%t || in = %b || out = %b || state = %d",$time,IN,OUT,DUT.state);

$dumpfile("FSM\_1101.vcd");

$dumpvars;

end

initial begin

#15 IN =0;

#10 IN =1; #10 IN = 1;#10 IN =0;#10 IN =1;#10 IN =1;#10 IN =0;

#20 $finish;

end

endmodule