

Verilog Code

```
`timescale 1ns/10ps
module SeqRcgn (out,x,clk,reset_) ;
// Port declarations
output out ;
input x,clk,reset_ ;

reg out ;
reg[2:0] state, next_state ;

parameter A= 3'b000, B=3'b001, C=3'b011, D=3'b010, E=3'b110;

always@ (posedge clk or negedge reset_)
    if(!reset_) // an asynchronous reset_
        state = A ;
    else
        state = next_state ;

always@ (state or x)
    case(state) // state transition
        A:
            if(x)
                next_state = A ;
            else
                next_state = B ;
        B:
            if(x)
                next_state = A ;
            else
                next_state = C ;
        C:
            if(x)
                next_state = D ;
            else
                next_state = C ;
        D:
            if(x)
                next_state = A ;
            else
                next_state = E ;
        E:
            if(x)
                next_state = A ;
            else
                next_state = C ;
        default:
            next_state = 3'bxxx ;
    endcase

always@ (state)
    case (state)
        A, B, C, D:
            out = 0 ;
        E:
            out = 1 ;
        default:
            out = 1'bx ;
    endcase

endmodule

module test_SeqRcng ;

reg x, clk, reset_ ;

//SeqRcgn instance
SeqRcgn SR1 (out, x, clk, reset_) ;

//Stimulus
initial
begin
    clk = 1'b0 ;
    repeat(30)
        #10 clk = ~clk ;

end

initial
begin
    x = 0 ; reset_ = 0;
    #15 reset_ = 1; x = 1 ;
    #20 x = 0 ;
    #20 x = 1 ;
    #20 x = 0 ;
    #20 x = 0 ;
    #20 x = 1 ;
    #20 x = 0 ;
    #20 x = 0 ;
    #20 x = 1 ;
    #20 x = 0 ;
    #20 x = 0 ;
    #20 x = 1 ;
    #20 x = 1 ;
    #20 x = 1 ;
    #20 x = 0 ;
    #20 x = 1 ;

end

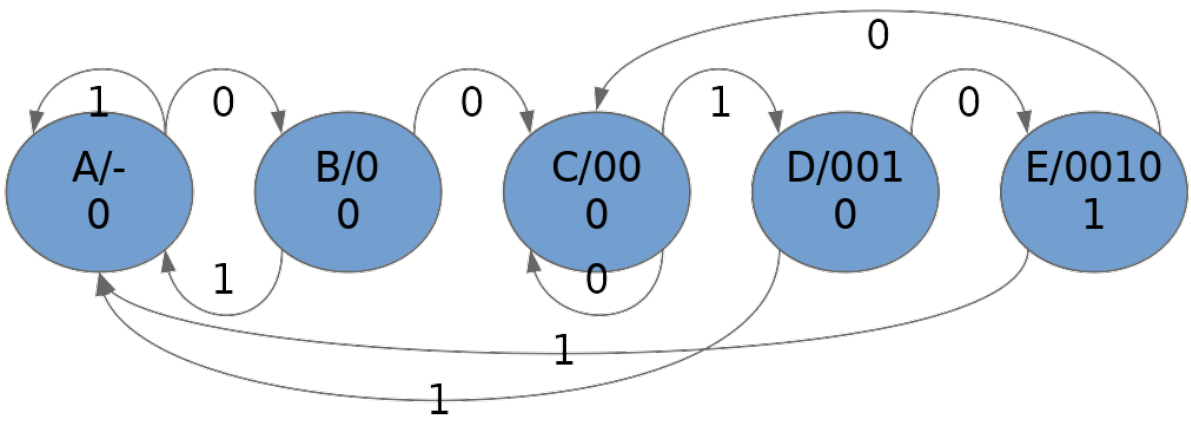
//Display results
initial
begin
    $display("          time out x  clk reset_");
    $monitor($time, "  %b  %b  %b  %b", out, x, clk, reset_) ;
end

initial
begin
    $dumpfile("SeqRcgn.vcd");
    $dumpvars;
end

endmodule
```

State transitions

Target: 0010 (2)



Timing diagram

