`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 = B ;

        else

            next\_state = A ;

    B:

        if(x)

            next\_state = B ;

        else

            next\_state = C ;

    C:

        if(x)

            next\_state = B ;

        else

            next\_state = D ;

    D:

        if(x)

            next\_state = E ;

        else

            next\_state = A ;

    E:

        if(x)

            next\_state = B ;

        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 = 0 ;

  #20  x = 1 ;

  #20  x = 0 ;

  #20  x = 0 ;

  #20  x = 1 ;

  #20  x = 0 ;

  #20  x = 1 ;

  #20  x = 1 ;

  #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

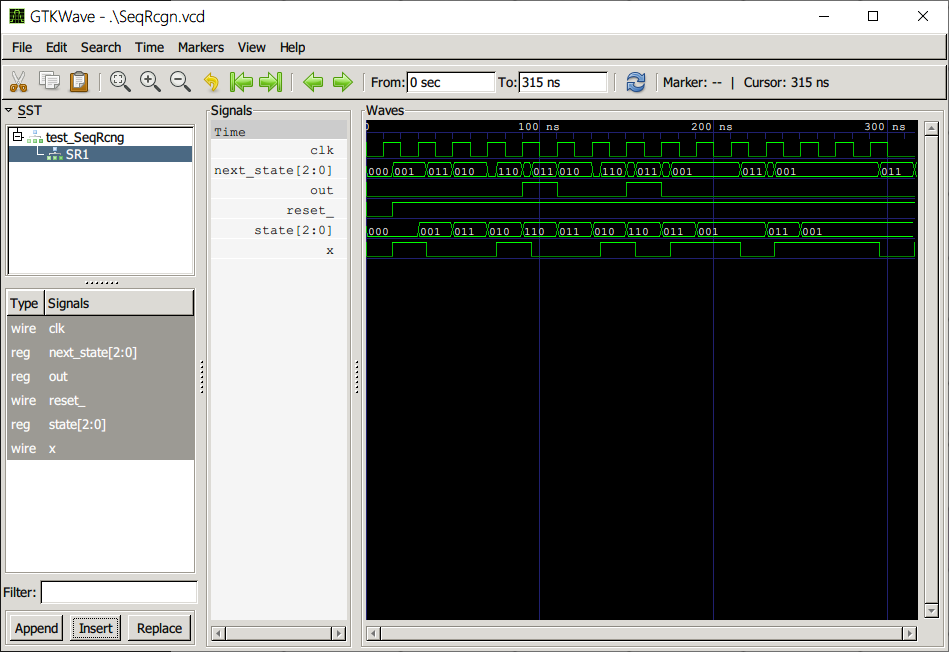
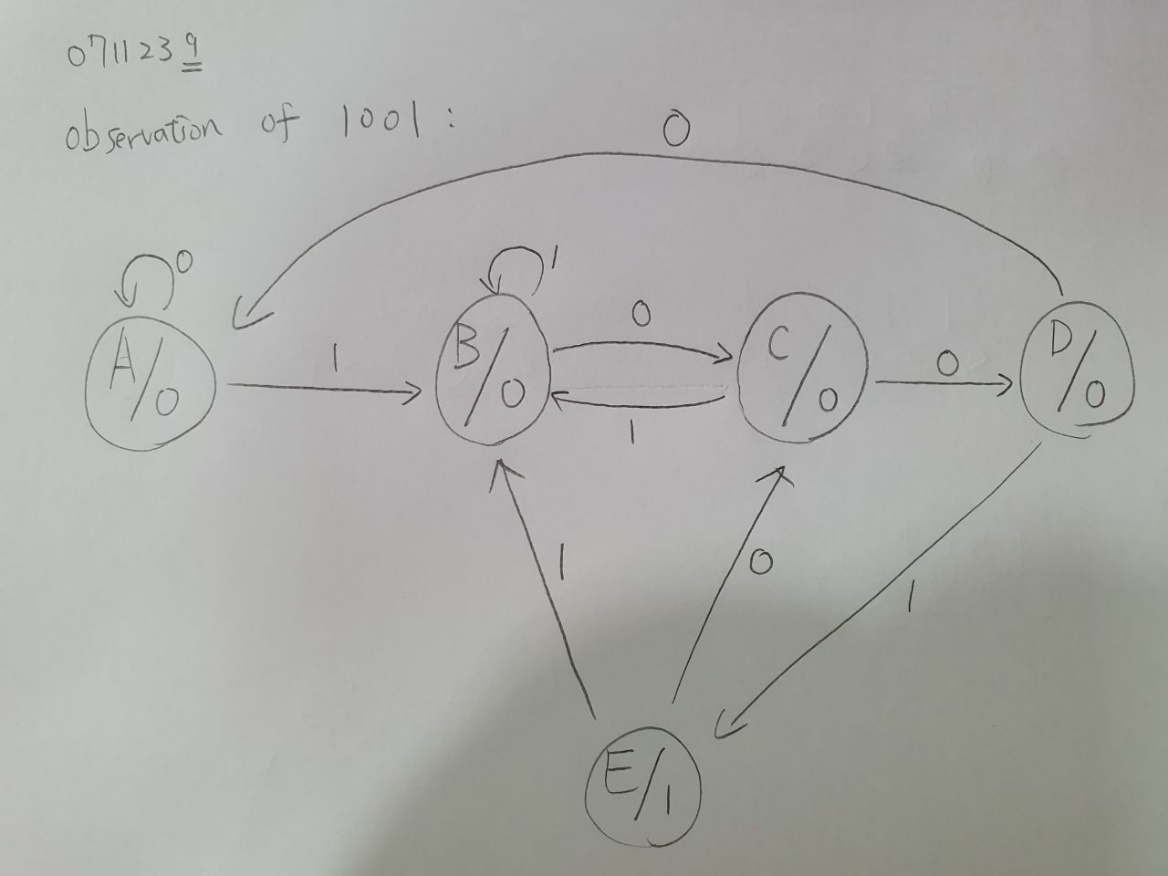
begin

$dumpfile("SeqRcgn.vcd");

$dumpvars;

end

endmodule

state diagram:

simulation: