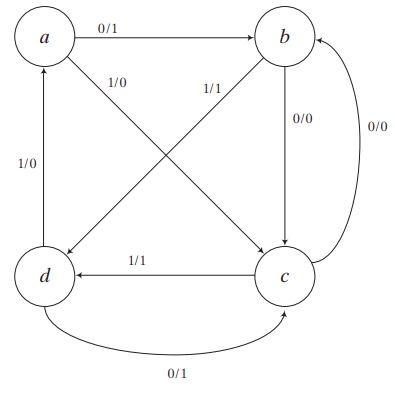
**Assignment 8**

**Assignment** All assignments are to be submitted strictly before start of next lab session through online only. Late assignments will not be entertained and will be awarded ‘0’ marks.

1. Write a Verilog model of the Mealy FSM described by below the state diagram. Develop a test bench and demonstrate that the machine state transitions and output correspond to its state diagram. <https://www.edaplayground.com/x/6pWs>



1. Draw the state diagram of the machine described by the Verilog model given below.

module Prob\_1 ( output reg y\_out, input x\_in, clk, reset);

parameter s0 = 2'b00, s1 = 2'b01, s2 = 2'b10, s3 = 2'b11;

reg [1:0] state, next\_state;

always @ ( posedge clk, negedge reset) begin

if (reset == 1'b0) state <= s0;

else

state <= next\_state;

always @(state, x\_in) begin

y\_out = 0;

next\_state = s0;

case (state)

s0: begin y\_out = 0; if (x\_in) next\_state = s1; else next\_state = s0; end

s1: begin y\_out = 0; if (x\_in) next\_state = s2; else next\_state = s1; end

s2: begin y\_out = 1; if (x\_in) next\_state = s3; else next\_state = s2; end

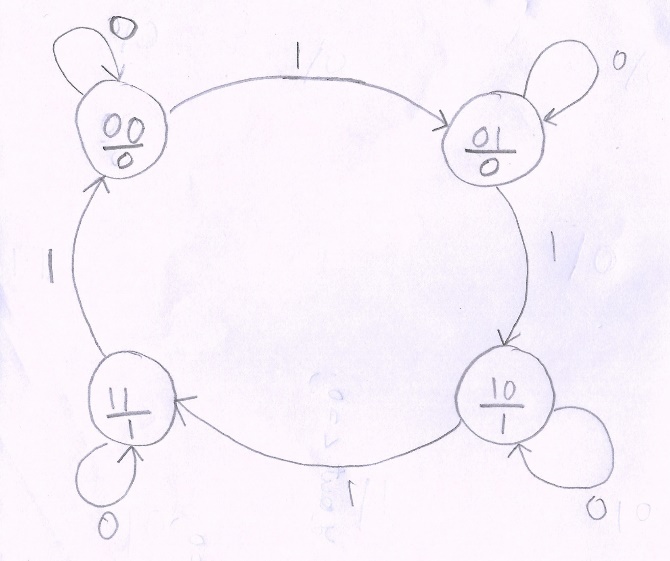
s3: begin y\_out = 1; if (x\_in) next\_state = s0; else next\_state = s3; end

default: next\_state = s0;

endcase

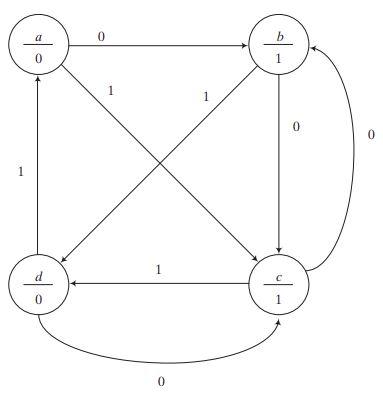
end

endmodule



**Self-Practice and self-evaluation**

1. Write a Verilog model of the Moore FSM described by the below state diagram. Develop a test bench and demonstrate that the machine's state transitions and output correspond to its state diagram.



1. Draw the state diagram of the machine described by the Verilog model given below.

module Prob\_2 ( output reg y\_out, input x\_in, clk, reset);

parameter s0 = 2'b00, s1 = 2'b01, s2 = 2'b10, s3 = 2'b11;

reg [1:0] state, next\_state;

always @ ( posedge clk, negedge reset) begin

if (reset == 1'b0) state <= s0;

else

state <= next\_state;

always @(state, x\_in) begin

y\_out = 0;

next\_state = s0;

case (state)

s0: if x\_in = 1 begin y\_out = 0; if (x\_in) next\_state = s1; else next\_state = s0; end

s1: if x\_in = 1 begin y\_out = 0; if (x\_in) next\_state = s2; else next\_state = s1; end

s2: if x\_in = 1 if (x\_in) begin next\_state = s3; y\_out = 0; else begin next\_state = s2; y\_out = 1; end

s3: if x\_in = 1 begin y\_out = 1; if (x\_in) next\_state = s0; else next\_state = s3; end

default: next\_state = s0;

endcase

end

endmodule