**Course Code: EE461**

**Assignment**

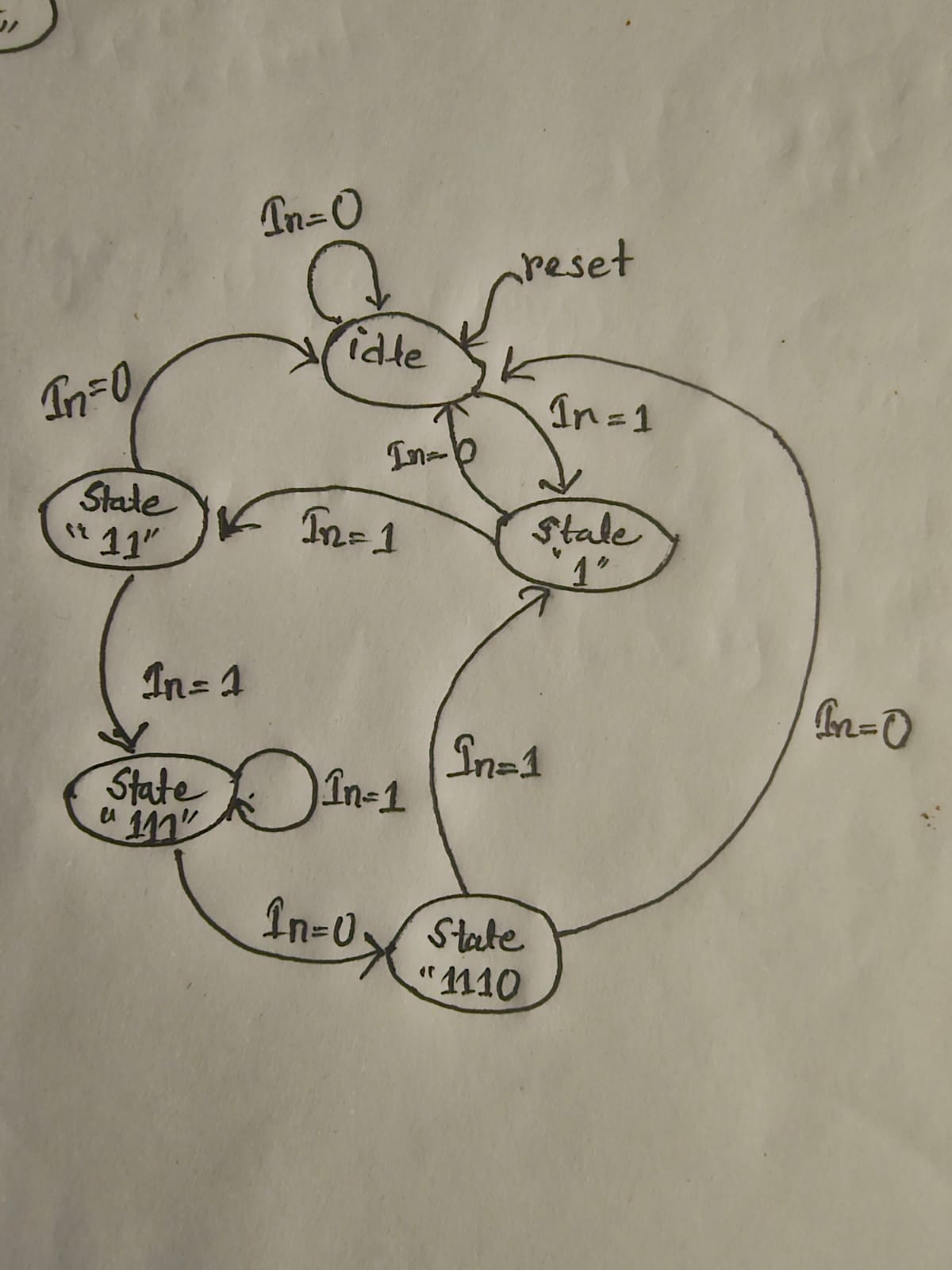
**PREPARED BY**

**Khandoker Samiul Hoque**

**Student Id: 19837**

**Github Url:** <https://github.com/KhandokerSamiulHoque/EE461_Assign5.git>

**1 No Answer:**



**Design:**

module bitstrmptrn (input in, clk ,rst ,

output reg out);

localparam [1:0]

state0 = 2'b00,

state1 = 2'b01,

state2 = 2'b10,

state3 = 2'b11;

reg [1:0] state;reg out\_flag;

always @(posedge clk, posedge rst)

begin

if (rst) begin

state <= state0;

out\_flag <= 0;

end

else begin

case (state)

state0: if (in) state <= state1; else state <= state0;

state1: if (in) state <= state2; else state <= state0;

state2: if (in) state <= state3; else state <= state0;

state3: begin

if

(in) state <= state1;

else begin

state <= state0;

out\_flag <= 1;

end

end

endcase

end

end

always @(posedge clk) begin

out <= out\_flag;

end

endmodule

**TestBench:**

module bitstrmptrn\_tb;

reg in, clk, rst;

wire out;

bitstrmptrn Assign5\_prob1(.in(in),.clk(clk),.rst(rst),.out(out));

initial begin

clk = 0;

forever

#3

clk = ~clk;

end

initial begin

$dumpfile("bitstrmptrn.vcd");

$dumpvars(0,bitstrmptrn\_tb);

rst = 1;

in = 0;

#8

rst = 0;

#8

in = 1;

#8

in = 1;

#8

in = 1;

#8

in = 0;

#8

in = 1;

#8

in = 0;

#8

in = 0;

#8

$finish();

end

always @(posedge clk) begin

$display("in=%b flag=%b", $time, in, out);

end

endmodule

**2 No Answer:**

**Design :**

module bitstreamdiv7(input in, input clk, rst, output reg out);

parameter pIdle1 = 3'b000;

parameter pRem0 = 3'b001;

parameter pRem1 = 3'b010;

parameter pRem2 = 3'b011;

parameter pRem3 = 3'b100;

parameter pRem4 = 3'b101;

parameter pRem5 = 3'b110;

parameter pRem6 = 3'b111;

reg [2:0] cutSt;

reg [2:0] nxtSt;

always @(posedge clk) begin

if (rst) cutSt <= #1 pIdle1;

else cutSt <= #1 nxtSt;

end

always @(\*) begin

out = 0;

case (cutSt)

pIdle1: begin

if (in) nxtSt = pRem1;

else nxtSt = pIdle1;

end

pRem1: begin

if (in) nxtSt = pRem2;

else nxtSt = pRem6;

end

pRem2: begin

if (in) nxtSt = pRem3;

else nxtSt = pRem5;

end

pRem3: begin

if (in) nxtSt = pRem4;

else nxtSt = pRem4;

end

pRem4: begin

if (in) nxtSt = pRem5;

else nxtSt = pRem3;

end

pRem5: begin

if (in) nxtSt = pRem6;

else nxtSt = pRem2;

end

pRem6: begin

if (in) nxtSt = pRem1;

else begin

out = 1;

nxtSt = pIdle1;

end

end

default: nxtSt = pIdle1;

endcase

end

endmodule

module TOP(input clk, rst,input in,output reg cnt);

wire out\_cntIn;

bitstreamdiv7 Dividedby7\_inst (.in(in),.clk(clk),.rst(rst),.out(out\_cntIn));

always @(posedge clk) begin

if (rst) cnt <= 0;

else if (out\_cntIn) cnt <= cnt + 1;

end

endmodule

**Testbench:**  
  
module bitstreamdiv7\_tb;

reg in;

reg clk;

reg rst;

wire out;

bitstreamdiv7 Assign5\_prob2(.clk(clk),.rst(rst),.in(in),.out(out));

always

#1

clk = ~clk;

initial begin

$dumpfile("bitstreamdiv7.vcd");

$dumpvars(0,bitstreamdiv7\_tb);

clk = 1'b1;

rst = 1'b1;

in = 1'b0;

#2

rst = 1'b0;

#4

in = 1'b1;

#6

in = 1'b0;

#7

in = 1'b1;

#8

in = 1'b0;

#8

rst = 1'b1;

#9

$finish;

end

endmodule