1. Implement Sequence recognizer for detecting three successive 1’s using Mealy Model (Behavioral Modeling).

module mealy\_seq(out,s,in,rst,clk);

output reg [2:0]s;

output reg out;

input in,rst,clk;

always @(posedge clk,rst)

if(rst)

begin

out<=0;

s<=3'b000;

end

else

begin

case({s})

3'b000:begin

if(in==1)

s<=3'b001;

end

3'b001:begin

if(in==1)

s<=3'b011;

else

s<=3'b000;

end

3'b011:begin

if(in==1)

s<=3'b111;

else

s<=3'b000;

end

3'b111:begin

s<=3'b000;

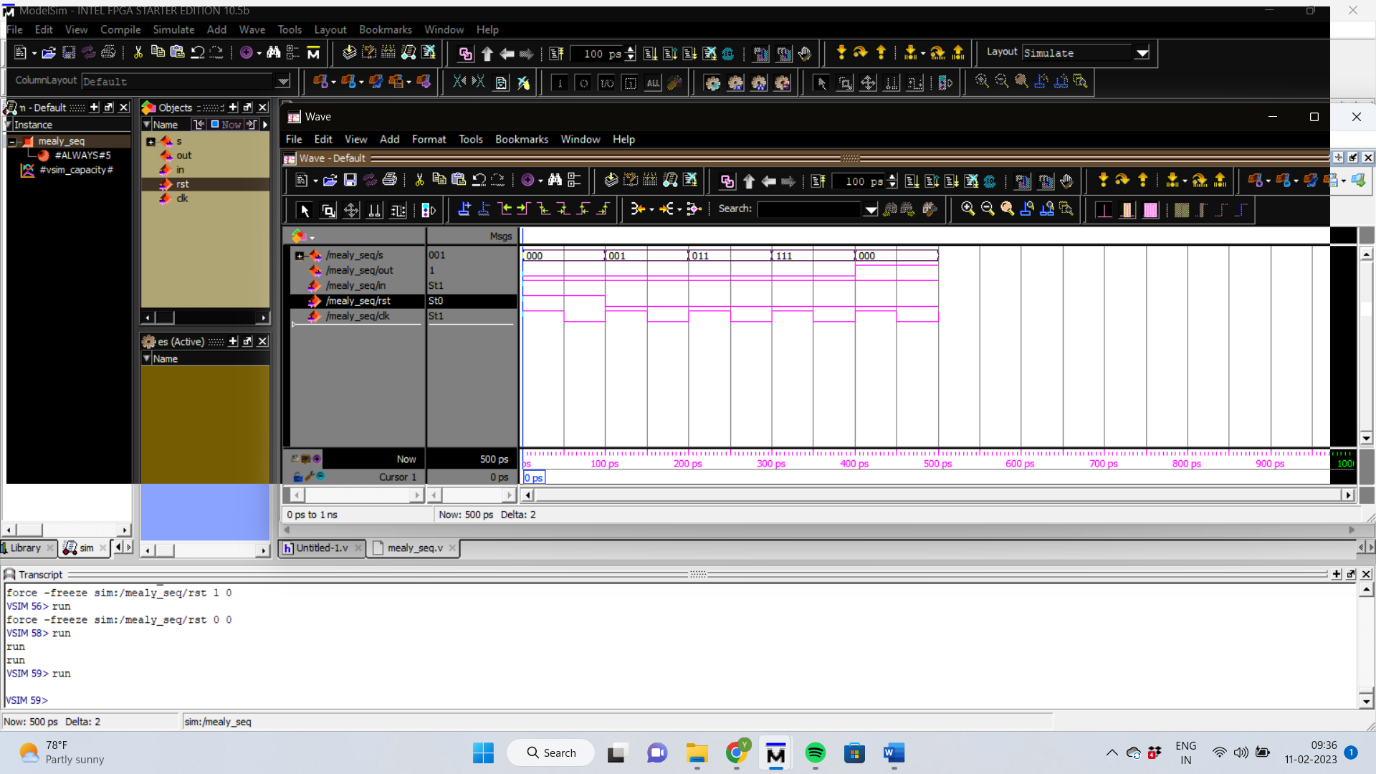
out<=1;

end

endcase

end

endmodule



1. 2) Consider the following state graph of a sequential network which has both Mealy and Moore outputs. The outputs Y1 and Y2 are the Mealy outputs and so should be

conditional outputs. The Ya, Yb, and Yc are the Moore outputs so they should be part of state box. Input X can either be “0” or “1” and hence it should be part of the decision box.

module mealy\_moore\_seq(meout,moout,mealy,moore,in,rst,clk);

output reg [2:0]mealy;

output reg [3:0]moore;

output reg meout,moout;

input in,rst,clk;

always @(posedge clk,rst)

begin

if(rst)

begin

meout<=0;

mealy<=3'b000;

moore<=4'b0000;

moout<=0;

end

else

begin

case({mealy})

3'b000:begin

if(in==1)

mealy<=3'b001;

meout<=0;

end

3'b001:begin

if(in==1)

mealy<=3'b011;

else

mealy<=3'b000;

end

3'b011:begin

if(in==1)

mealy<=3'b111;

else

mealy<=3'b000;

end

3'b111:begin

mealy<=3'b000;

meout<=1;

end

endcase

/////////

///////////

case({moore})

4'b0000:begin

if(in==1)

moore<=4'b0001;

moout<=0;

end

4'b0001:begin

if(in==1)

moore<=4'b0011;

else

moore<=4'b0000;

end

4'b0011:begin

if(in==1)

moore<=4'b0111;

else

moore<=4'b0000;

end

4'b0111:begin

if(in==1)

begin

moore<=4'b1111;

moout<=1;

end

else

moore<=4'b0000;

end

endcase

///////

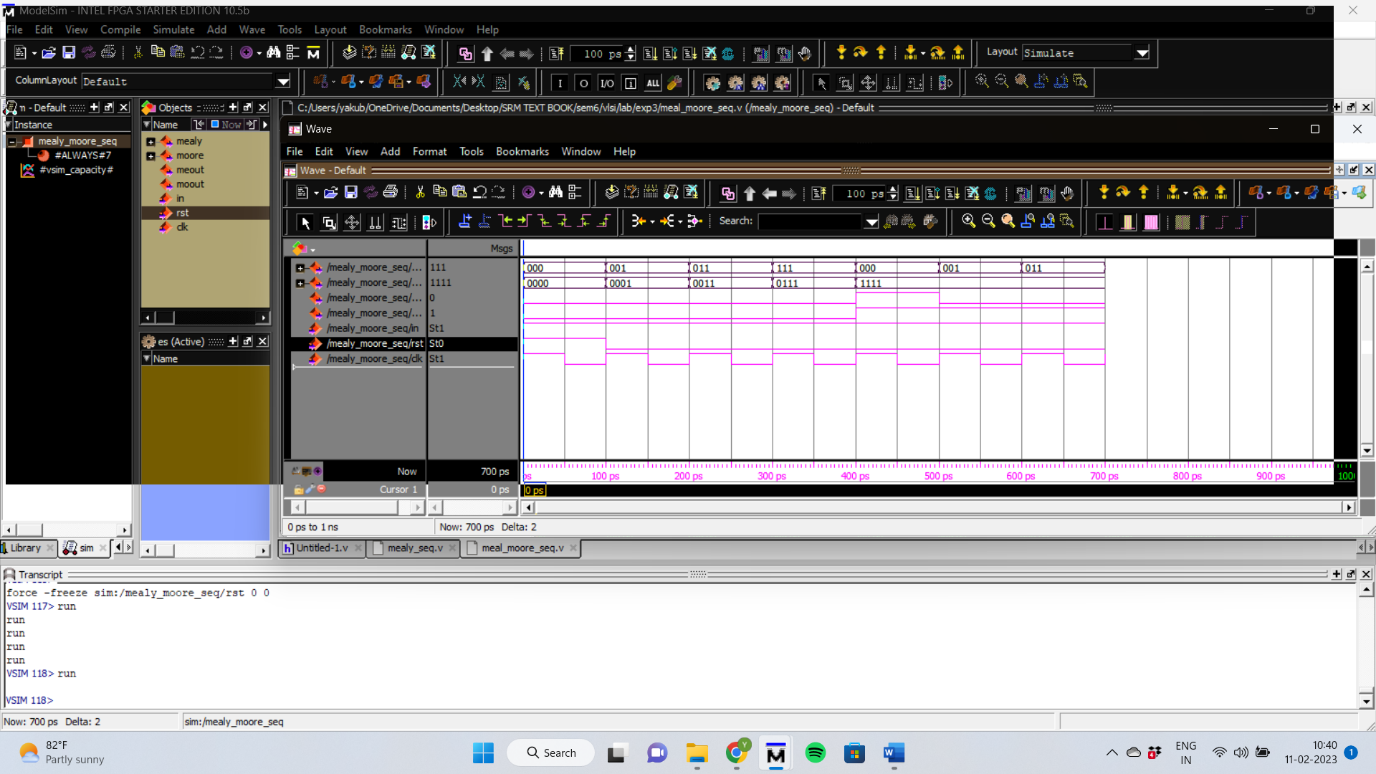
//////

///////

end

end

endmodule



POST LAB

1) Write Verilog code to implement an FSM using Moore Machine.

module moore\_seq(moout,moore,in,rst,clk);

output reg [3:0]moore;

output reg moout;

input in,rst,clk;

always @(posedge clk,rst)

begin

if(rst)

begin

moore<=4'b0000;

moout<=0;

end

else

begin

case({moore})

4'b0000:begin

if(in==1)

moore<=4'b0001;

moout<=0;

end

4'b0001:begin

if(in==1)

moore<=4'b0011;

else

moore<=4'b0000;

end

4'b0011:begin

if(in==1)

moore<=4'b0111;

else

moore<=4'b0000;

end

4'b0111:begin

if(in==1)

begin

moore<=4'b1111;

moout<=1;

end

else

begin

moore<=4'b0000;

moout<=0;

end

end

4'b1111:begin

if(in==0)

begin

moore<=4'b0000;

moout<=0;

end

end

endcase

///////

//////

///////

end

end

endmodule

