Maven Lab 3

1)Write behavioral description and test bench for 4:1 MUX

CODE:

module mux\_4\_1(input [3:0]in,input [1:0]sel,output reg out);

always@(\*)

begin

case(sel)

2'b00:out<=in[0];

2'b01:out<=in[1];

2'b10:out<=in[2];

2'b11:out<=in[3];

default : out<=1'b0;

endcase

end

endmodule

TESTBENCH:

module mux\_4\_1\_tb;

reg [3:0]in;

reg [1:0]sel;

wire out;

integer i;

integer j;

mux\_4\_1 dut(in,sel,out);

initial

begin

for(i=0;i<4;i=i+1)

begin

sel=i;

#10;

end

for(j=0;j<16;j=j+1)

begin

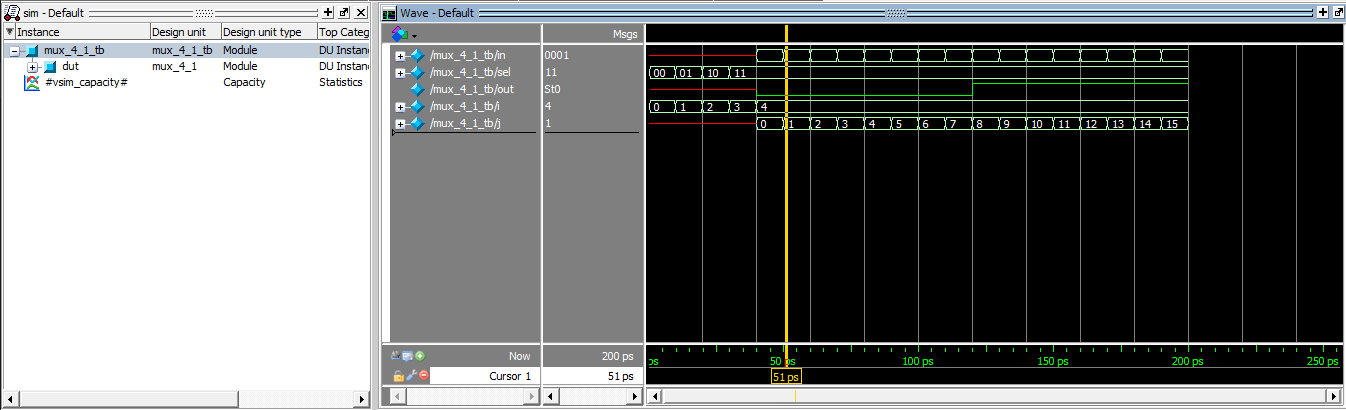
in=j;

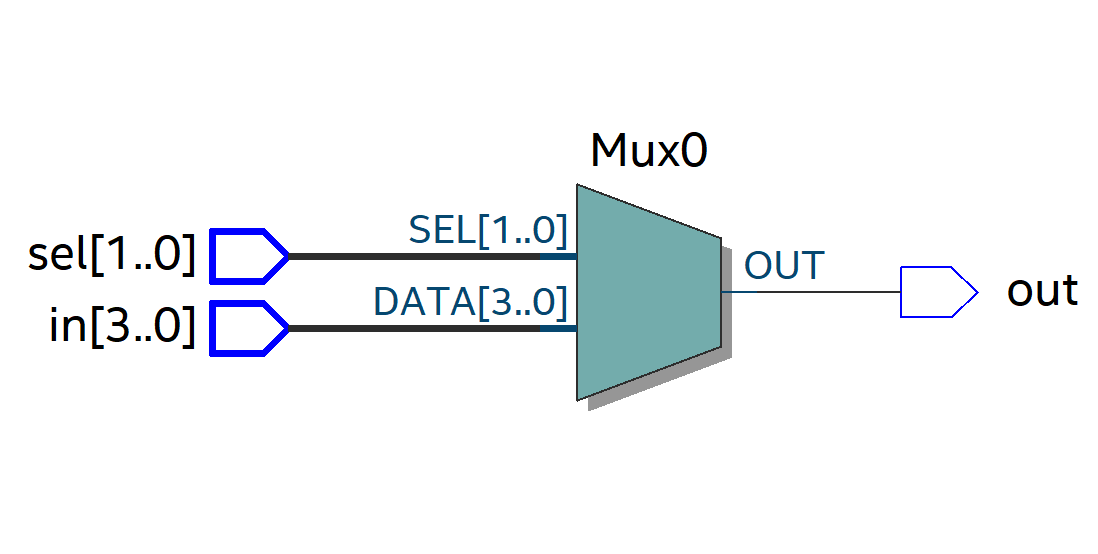
#10;

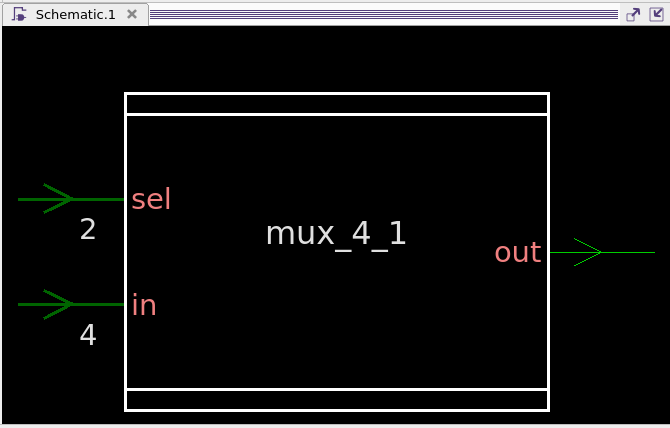
end

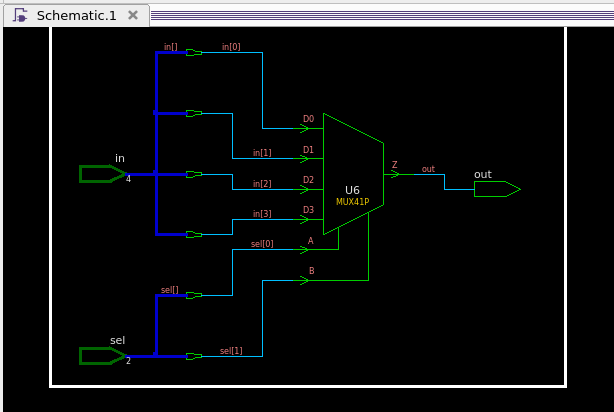
end

endmodule









2. Write a behavioral description for a 3:8 decoder and verify using a test bench.

CODE:

module decoder\_3\_8(input [2:0]in, output reg [7:0]out);

always@(\*)

begin

case(in)

3'b000:out<=8'b0000\_0001;

3'b001:out<=8'b0000\_0010;

3'b010:out<=8'b0000\_0100;

3'b011:out<=8'b0000\_1000;

3'b100:out<=8'b0001\_0000;

3'b101:out<=8'b0010\_0000;

3'b110:out<=8'b0100\_0000;

3'b111:out<=8'b1000\_0000;

default:out<=8'b0000\_0000;

endcase

end

endmodule

TESTBENCH:

module decoder\_3\_8\_tb;

reg [2:0]in;

wire [7:0]out;

integer i;

decoder\_3\_8 dut(in,out);

initial

begin

in=3'b000;

end

initial

begin

for(i=0;i<8;i=i+1)

begin

in=i;

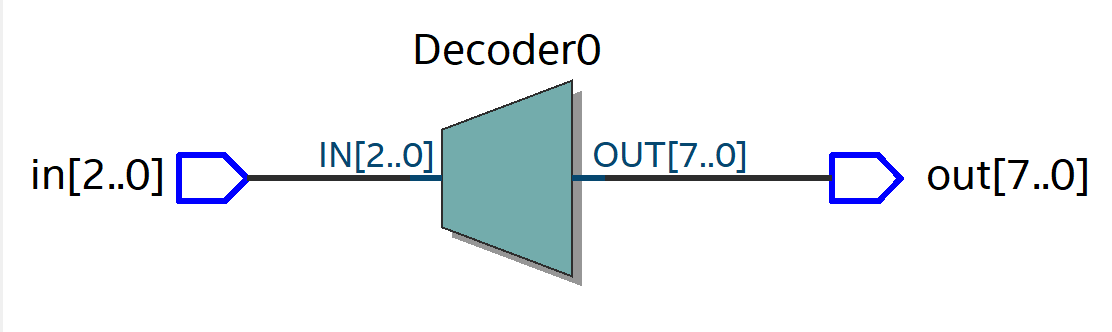
#10;

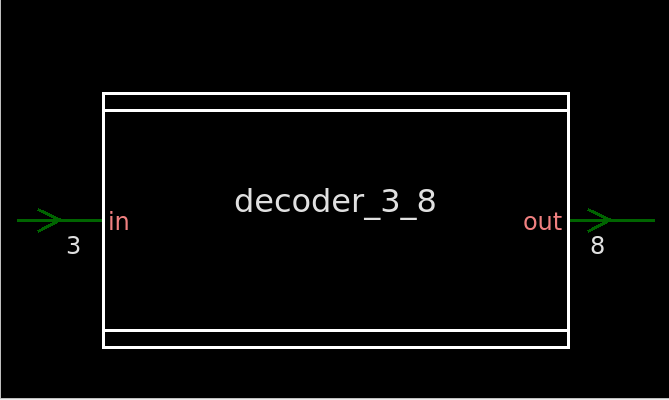
end

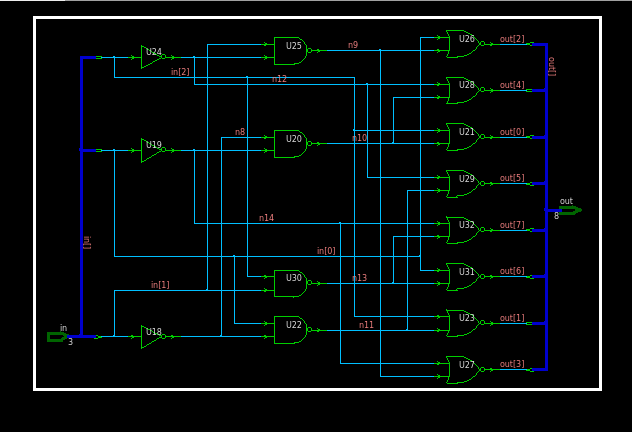
end

endmodule









3.Write a behavioral description  and testbench 8:3 priority encoder.

CODE:

module encoder\_8\_3 (

input [7:0] in,

output reg [2:0] out

);

always @\*

begin

if (in[7] == 1)

out = 3'b111;

else if (in[6] == 1)

out = 3'b110;

else if (in[5] == 1)

out = 3'b101;

else if (in[4] == 1)

out = 3'b100;

else if (in[3] == 1)

out = 3'b011;

else if (in[2] == 1)

out = 3'b010;

else if (in[1] == 1)

out = 3'b001;

else

out = 3'b000;

end

endmodule

TESTBENCH:

module encoder\_8\_3\_tb;

reg [7:0]in;

wire [2:0]out;

integer i;

encoder\_8\_3 dut(in,out);

initial

begin

for(i=0;i<256;i=i+5)

begin

in=i;

#10;

end

end

endmodule

