190905514 MOHAMMAD TOFIK

**WEEK 5 LAB 5**

**Q1.**Design and simulatea combinational circuit with external gates and a 4 to 16 decoder built using a decoder tree of 2 to 4 decoders to implement the functions below

.F= ab’c + a’cd + bcd’ ,

G=acd’ + a’b’c and

H=a’b’c’ + abc + a’cd

module functions(W, En, f, g, h);

input [3:0]W;

input En;

output f,g,h;

wire [0:15]Y;

dec4to16 dec1(W, En, Y);

or(f, Y[3], Y[6], Y[7], Y[10], Y[11], Y[14]);

or(g, Y[2], Y[3], Y[10], Y[14]);

or(h, Y[0], Y[1], Y[3], Y[7], Y[14], Y[15]);

endmodule

module dec4to16(W,En,Y);

input [3:0]W;

input En;

output [0:15]Y;

wire [0:3]M;

dec2to4 dec1(W[3:2],En,M[0:3]);

dec2to4 dec2 (W[1:0],M[0],Y[0:3]);

dec2to4 dec3 (W[1:0],M[1],Y[4:7]);

dec2to4 dec4 (W[1:0],M[2],Y[8:11]);

dec2to4 dec5 (W[1:0],M[3],Y[12:15]);

endmodule

module dec2to4(W,En,Y);

input[1:0]W;

input En;

output[0:3]Y;

reg[0:3]Y;

always@(W or En)

begin

if(En==1)

case(W)

0: Y=4'b1000;

1: Y=4'b0100;

2: Y=4'b0010;

3: Y=4'b0001;

endcase

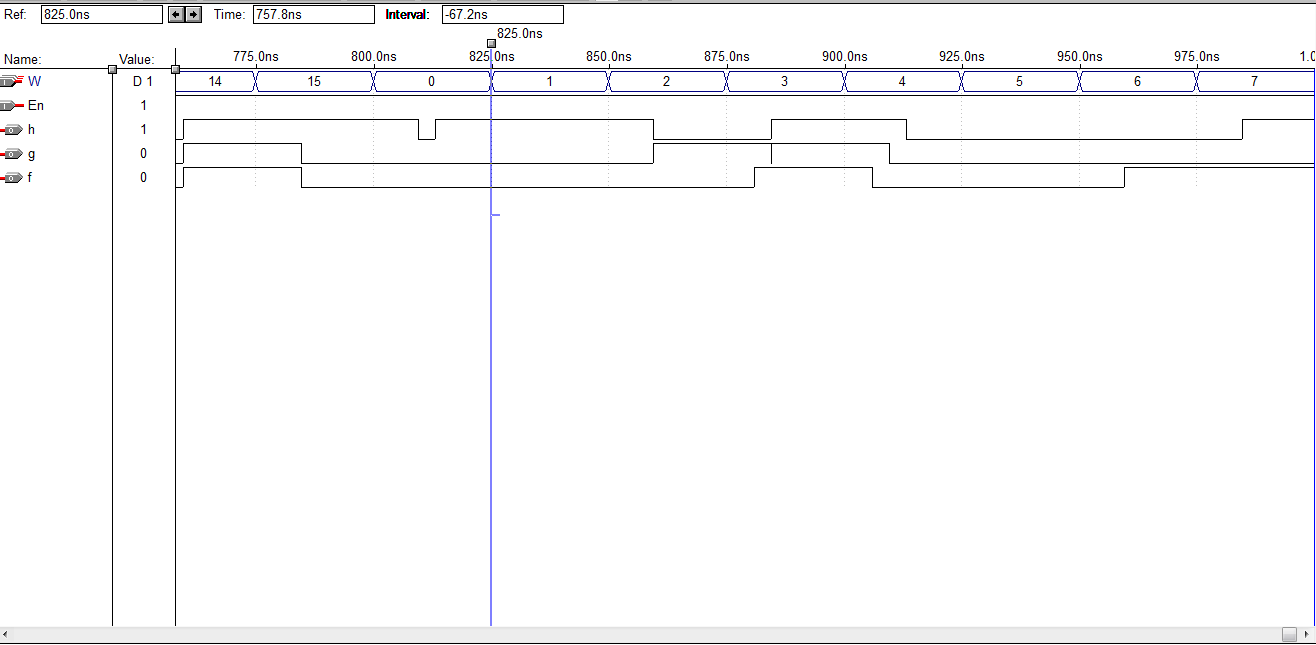
else

Y=4'b0000;

end

endmodule

**OUTPUT :**



**Q2.**

Design and implement a full adder using 2 to 4 decoder(s)and other gates.

module fadder(x,y,c\_in,sum,c\_out);

input x,y,c\_in;

output sum,c\_out;

wire [0:3]dec0w;

wire [0:3]dec1w;

wire [0:3]dec2w;

dec2to4 dec0({1'b0,x},1'b1,dec0w);

dec2to4 dec1({c\_in,y},dec0w[1],dec1w);

dec2to4 dec2({c\_in,y},dec0w[0],dec2w);

or(c\_out,dec2w[3],dec1w[1],dec1w[2],dec1w[3]);

or(sum,dec2w[1],dec2w[2],dec1w[0],dec1w[3]);

endmodule

module dec2to4(W,En,Y);

input[1:0]W;

input En;

output [0:3]Y;

reg [0:3]Y;

always@(W or En)

begin

if(En==1)

case(W)

0:Y=4'b1000;

1:Y=4'b0100;

2:Y=4'b0010;

3:Y=4'b0001;

endcase

else

Y=4'b0000;

end

endmodule

**OUTPUT :**

