Ayush Goyal

190905522

DSD Lab 5 (Session 3)

Q1) Design and simulate a 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

Code:

```
module q51(w,en,F,G,H);
        input [3:0]w;
        input en;
        wire [15:0]decOut;
        output F,G,H;
        dec4t16 d1(w[3:0],en,decOut);
        assign F = decOut[3]|decOut[6]|decOut[7]|decOut[10]|decOut[11]|decOut[14];
        assign G = decOut[2]|decOut[3]|decOut[10]|decOut[14];
        assign H = decOut[0]|decOut[1]|decOut[3]|decOut[7]|decOut[14]|decOut[15];
endmodule
```

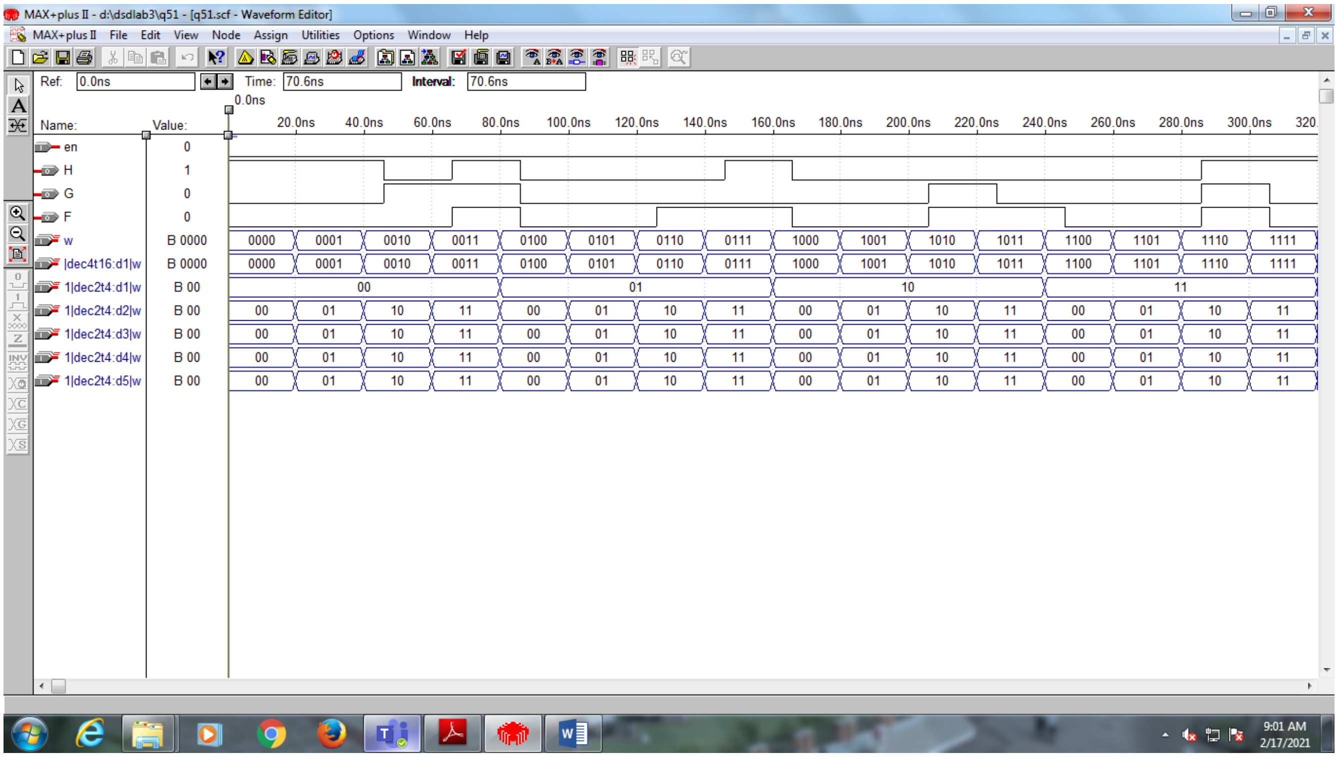
```
module dec4t16(w,en,y);
        input [3:0]w;
        input en;
        output [15:0]y;
        wire [3:0]temp;
        dec2t4 d1(w[3:2],en,temp);
```

```
            dec2t4 d2(w[1:0],~temp[0],y[3:0]);
            dec2t4 d3(w[1:0],~temp[1],y[7:4]);
            dec2t4 d4(w[1:0],~temp[2],y[11:8]);
            dec2t4 d5(w[1:0],~temp[3],y[15:12]);
endmodule



module dec2t4(w,en,y);
            input [1:0]w;
            input en;
            output [3:0]y;
            reg [3:0]y;
            always @(w|en)
            begin
            case({en,w})
            3'b000: y=4'b0001;
            3'b001: y=4'b0010;
            3'b010: y=4'b0100;
            3'b011: y=4'b1000;
            default: y=4'b0000;
            endcase
            end
endmodule
```

OUTPUT:



```
Ref: 0.0ns    Time: 70.6ns    Interval: 70.6ns

Name:             Value:
en                0
H                 1
G                 0
F                 0
w                 B 0000   0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111
|dec4t16:d1|w     B 0000   0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111
1|dec2t4:d1|w     B 00     00 01 10 11
1|dec2t4:d2|w     B 00     00 01 10 11 00 01 10 11 00 01 10 11 00 01 10 11
1|dec2t4:d3|w     B 00     00 01 10 11 00 01 10 11 00 01 10 11 00 01 10 11
1|dec2t4:d4|w     B 00     00 01 10 11 00 01 10 11 00 01 10 11 00 01 10 11
1|dec2t4:d5|w     B 00     00 01 10 11 00 01 10 11 00 01 10 11 00 01 10 11
```

Q2) Design and implement a full adder using 2 to 4 decoder(s) and other gates.

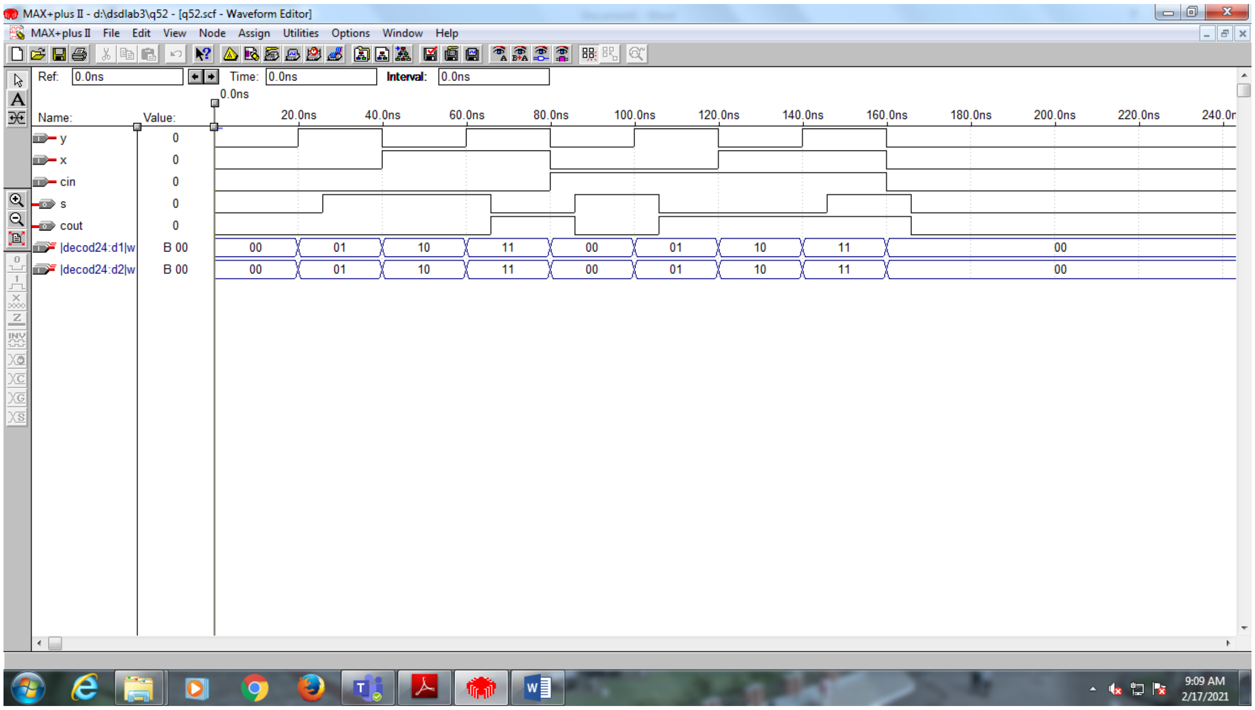
Code:

```
module q52(x,y,cin,s,cout);
        input x,y,cin;
        output s,cout;
        wire [7:0]decOut;
        decod24 d1({x,y},~cin,decOut[7:4]);
        decod24 d2({x,y},cin,decOut[3:0]);
        assign s = decOut[1]|decOut[2]|decOut[4]|decOut[7];
        assign cout = decOut[3]|decOut[5]|decOut[6]|decOut[7];
endmodule


module decod24(w,en,y);
```

```
        input [1:0]w;
        input en;
        output [3:0]y;
        reg [3:0]y;
        always @(w|en)
        begin
        case({en,w})
        3'b000: y=4'b0001;
        3'b001: y=4'b0010;
        3'b010: y=4'b0100;
        3'b011: y=4'b1000;
        default: y=4'b0000;
        endcase
        end
endmodule
```

OUTPUT:


MAX+plus II - d:\dsdlab3\q52 - [q52.scf - Waveform Editor]
MAX+plus II File Edit View Node Assign Utilities Options Window Help
Ref: 0.0ns
Time: 0.0ns
Interval: 0.0ns
0.0ns
20.0ns 40.0ns 60.0ns 80.0ns 100.0ns 120.0ns 140.0ns 160.0ns 180.0ns 200.0ns 220.0ns 240.0r
Name:
Value:
y 0
x 0
cin 0
s 0
cout 0
|decod24:d1|w B 00
|decod24:d2|w B 00
00 01 10 11 00 01 10 11 00
00 01 10 11 00 01 10 11 00
9:09 AM
2/17/2021

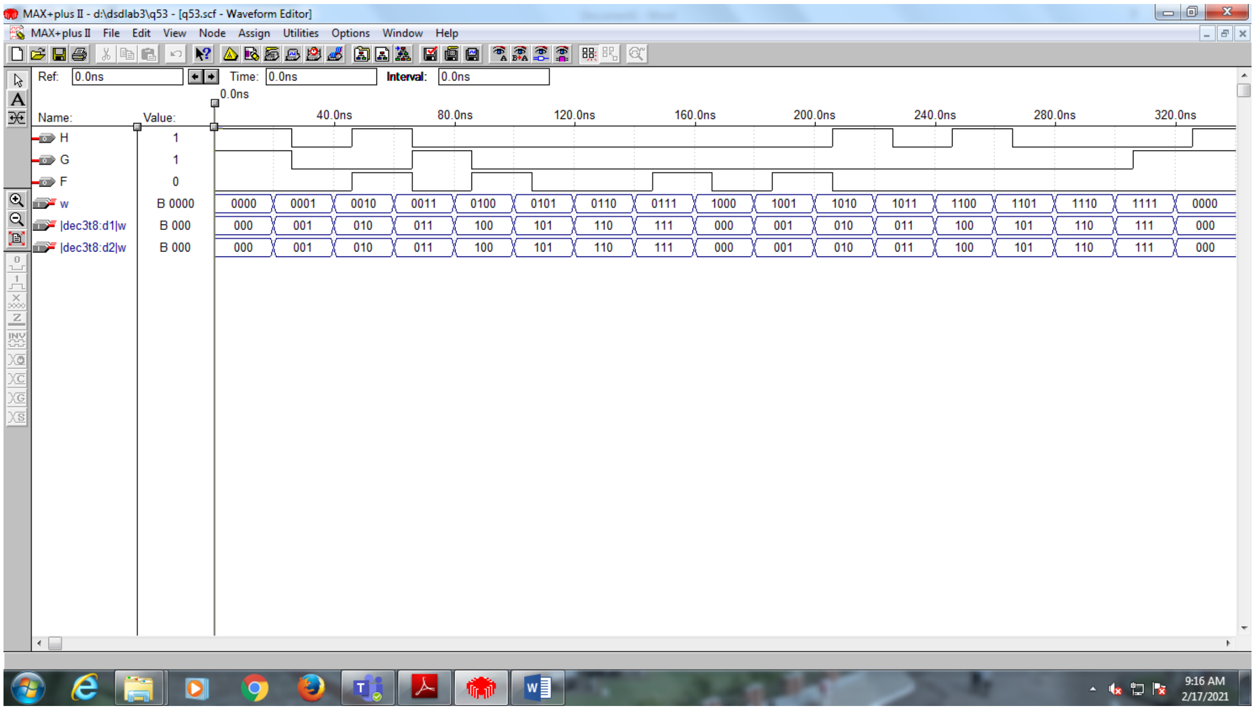
Q3) Design and simulate the circuit with 3 to 8 decoder(s) and external gates to implement the functions below.
F(a, b, c, d)= Σm(2,4,7,9) G (a, b, c, d)= Σm (0,3,15) H(a, b, c, d)= Σm(0,2,10,12)

Code:

```
module q53(w,F,G,H);
        input [3:0]w;
        wire [15:0]decOut;
        output F,G,H;
        dec3t8 d1(w[2:0],w[3],decOut[15:8]);
        dec3t8 d2(w[2:0],~w[3],decOut[7:0]);
        assign F = decOut[2]|decOut[4]|decOut[7]|decOut[9];
        assign G = decOut[0]|decOut[3]|decOut[15];
        assign H = decOut[0]|decOut[2]|decOut[10]|decOut[12];
endmodule


module dec3t8(w,en,y);
        input [2:0]w;
        input en;
        output [7:0]y;
        reg [7:0]y;
        always @(w or en)
        begin
        if(en)
        begin
        y=0;
        case(w)
        3'b000: y=8'b00000001;
        3'b001: y=8'b00000010;
        3'b010: y=8'b00000100;
        3'b011: y=8'b00001000;
        3'b100: y=8'b00010000;
        3'b101: y=8'b00100000;
        3'b110: y=8'b01000000;
        3'b111: y=8'b10000000;
        default: y=0;
        endcase
        end
        else
        y=0;
        end
endmodule
```

OUTPUT:



```
Ref: 0.0ns        Time: 0.0ns        Interval: 0.0ns

                0.0ns        40.0ns       80.0ns       120.0ns       160.0ns       200.0ns       240.0ns       280.0ns       320.0ns
Name:          Value:
H               1
G               1
F               0
w               B 0000   0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111 0000
|dec3t8:d1|w    B 000    000  001  010  011  100  101  110  111  000  001  010  011  100  101  110  111  000
|dec3t8:d2|w    B 000    000  001  010  011  100  101  110  111  000  001  010  011  100  101  110  111  000
```