DSDL

Week 5

Udeet Mittal

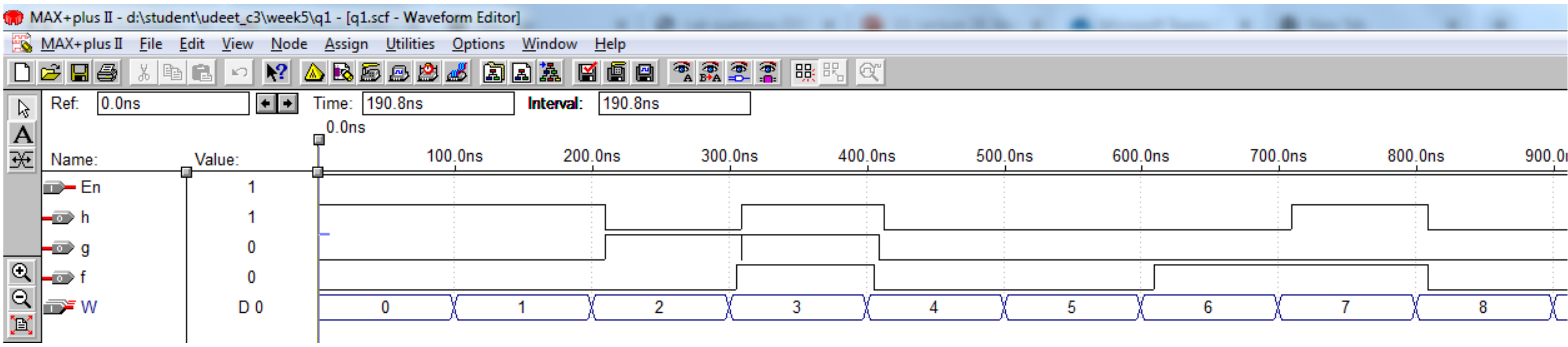
CSE C3

Roll Number 64

1. 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

```
module decoder2to4(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
```

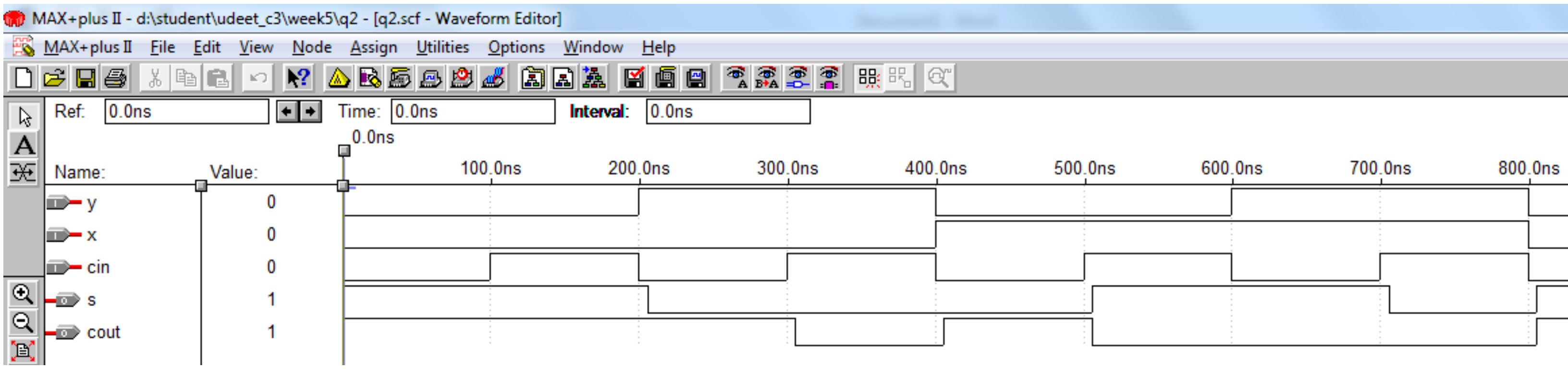
```
endmodule

module decoder4to16(W,En,Y);
input [3:0]W;
input En;
output [0:15]Y;
wire[0:3]Z;
decoder2to4 d1(W[3:2],En,Z[0:3]);
decoder2to4 d2(W[1:0],Z[0],Y[0:3]);
decoder2to4 d3(W[1:0],Z[1],Y[4:7]);
decoder2to4 d4(W[1:0],Z[2],Y[8:11]);
decoder2to4 d5(W[1:0],Z[3],Y[12:15]);
endmodule

module q1(W, En, f, g, h);
input [3:0] W;
input En;
output f, g, h;
wire [0:15]Y;
decoder4to16 d1(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
```



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

```
module dec2to4(En,w,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=8;
3'b001:y=4;
3'b010:y=2;
3'b001:y=1;
default:y=0;
endcase
end
endmodule
```

```
module q2(x,y,cin,cout,s);

input x,y,cin;

output s,cout;

wire En1,En2,En;

wire [1:0]ip1;

wire [7:0]z;

assign En=1;

assign En1=(~x&En);

assign En2=(x&En);

assign ip1={y,cin};

dec2to4 stage1(En1,ip1,z[3:0]);

dec2to4 stage2(En2,ip1,z[7:4]);

assign s=z[1]|z[2]|z[4]|z[7];

assign cout=z[3]|z[5]|z[6]|z[7];

endmodule
```



MAX+plus II - d:\student\udeet_c3\week5\q2 - [q2.scf - Waveform Editor]

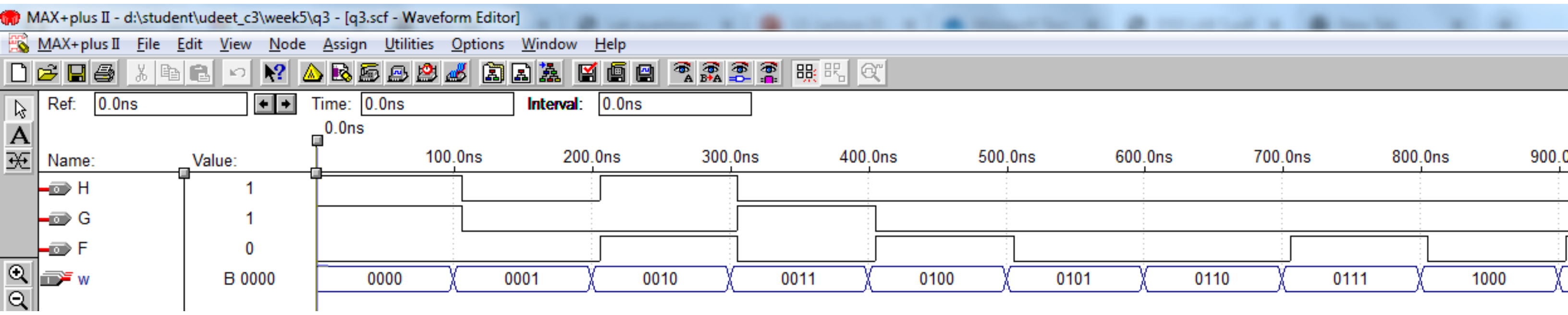
Ref: 0.0ns Time: 0.0ns Interval: 0.0ns

Name:	Value:
y	0
x	0
cin	0
s	1
cout	1

3. 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)

```
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

module q3(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
```



MAX+plus II - d:\student\udeet_c3\week5\q3 - [q3.scf - Waveform Editor]

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

Name:	Value:
H	1
G	1
F	0
w	B 0000