Week 4

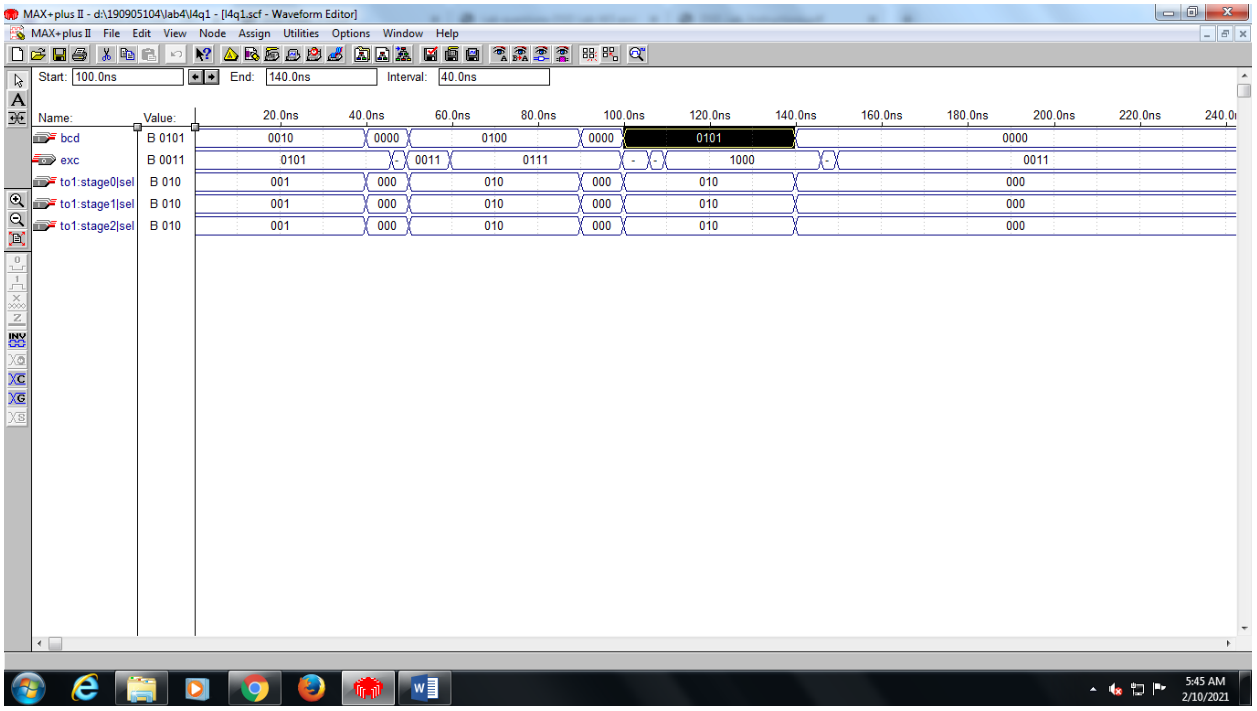
1. Write and simulate the Verilog code for a BCD to Excess 3 code converter using 8 to 1 multiplexers and other necessary gates.

```
module mux8to1(d, sel, f);

input [0:7]d;

input [2:0] sel;

output f;

reg f;

always@(sel)

begin

case(sel)

3'b000: f=d[0];

3'b001: f=d[1];

3'b010: f=d[2];

3'b011: f=d[3];

3'b100: f=d[4];

3'b101: f=d[5];

3'b110: f=d[6];

3'b111: f=d[7];

endcase

end

endmodule


module l4q1(bcd, exc);

input [3:0]bcd;

output [3:0]exc;


mux8to1 stage0({1'b0,1'b0,bcd[0],1'b1,1'b1,X,X,X}, bcd[3:1], exc[3]);

mux8to1 stage1({bcd[0],1'b1,~bcd[0],1'b0,bcd[0],X,X,X}, bcd[3:1], exc[2]);
```

mux8to1 stage2({~bcd[0],bcd[0],~bcd[0],bcd[0],~bcd[0],X,X,X}, bcd[3:1], exc[1]);

assign exc[0]=~bcd[0];

endmodule



```
MAX+plus II - d:\190905104\lab4\l4q1 - [l4q1.scf - Waveform Editor]

Start: 100.0ns    End: 140.0ns    Interval: 40.0ns

Name:            Value:   20.0ns   40.0ns   60.0ns   80.0ns   100.0ns   120.0ns   140.0ns   160.0ns   180.0ns   200.0ns   220.0ns   240.0ns
bcd              B 0101   0010     0000     0100     0000     0101      0000
exc              B 0011   0101     -        0011     0111     -  -      1000      -         0011
to1:stage0|sel   B 010    001      000      010      000      010       000
to1:stage1|sel   B 010    001      000      010      000      010       000
to1:stage2|sel   B 010    001      000      010      000      010       000
```

2. Write behavioral Verilog code for a 2 to 4 decoder with active low enable input and active high output using case statement. Using this, design a 4 to 16 decoder with active low enable input and active high output and write the Verilog code for the same.

module dec2to4(w, en, y);

input [1:0]w;

input en;

output [3:0]y;

reg [3:0]y;

always @(w or en)

begin

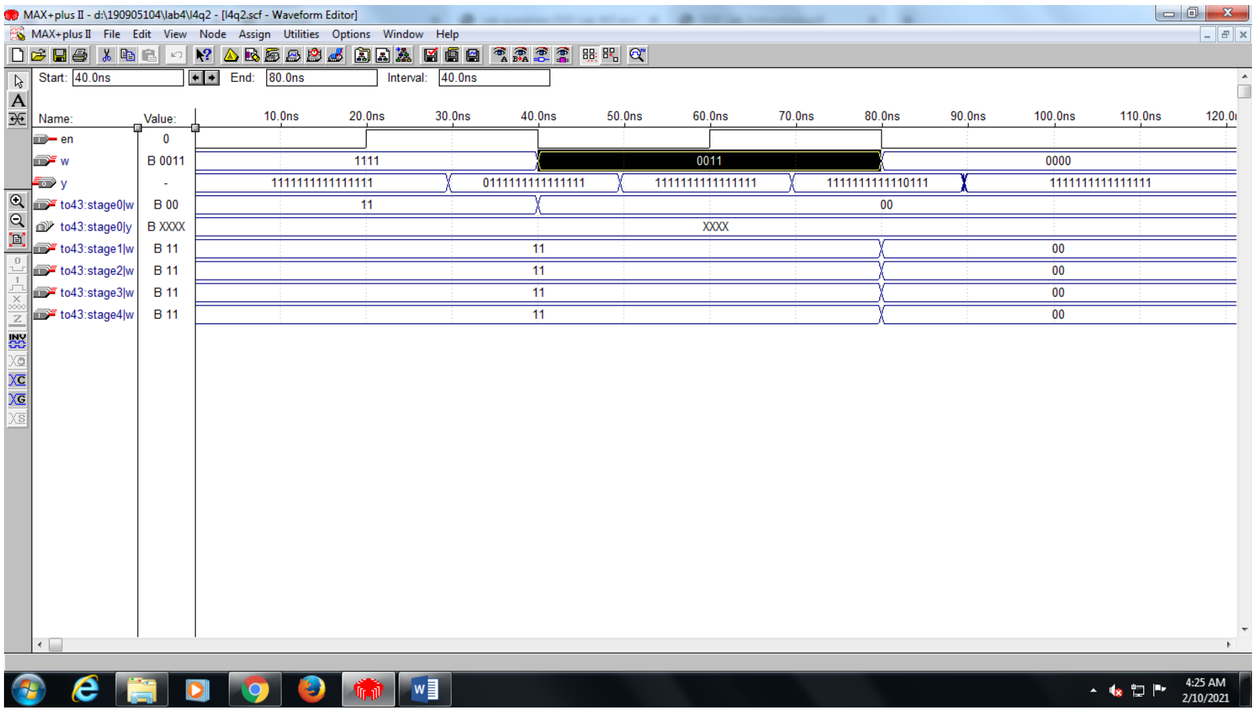
if(en == 0)

y = 15;

else

begin

```
case (w)
0:y = 14;
1:y = 13;
2:y = 11;
3:y = 7;
endcase
end
end
endmodule


module l4q2(w,y,en);
input en;
input [3:0]w;
output [15:0]y;
wire [3:0]x;
dec2to4 stage0(w[3:2],en,x[3:0]);
dec2to4 stage1(w[1:0],~x[0],y[3:0]);
dec2to4 stage2(w[1:0],~x[1],y[7:4]);
dec2to4 stage3(w[1:0],~x[2],y[11:8]);
dec2to4 stage4(w[1:0],~x[3],y[15:12]);
endmodule
```

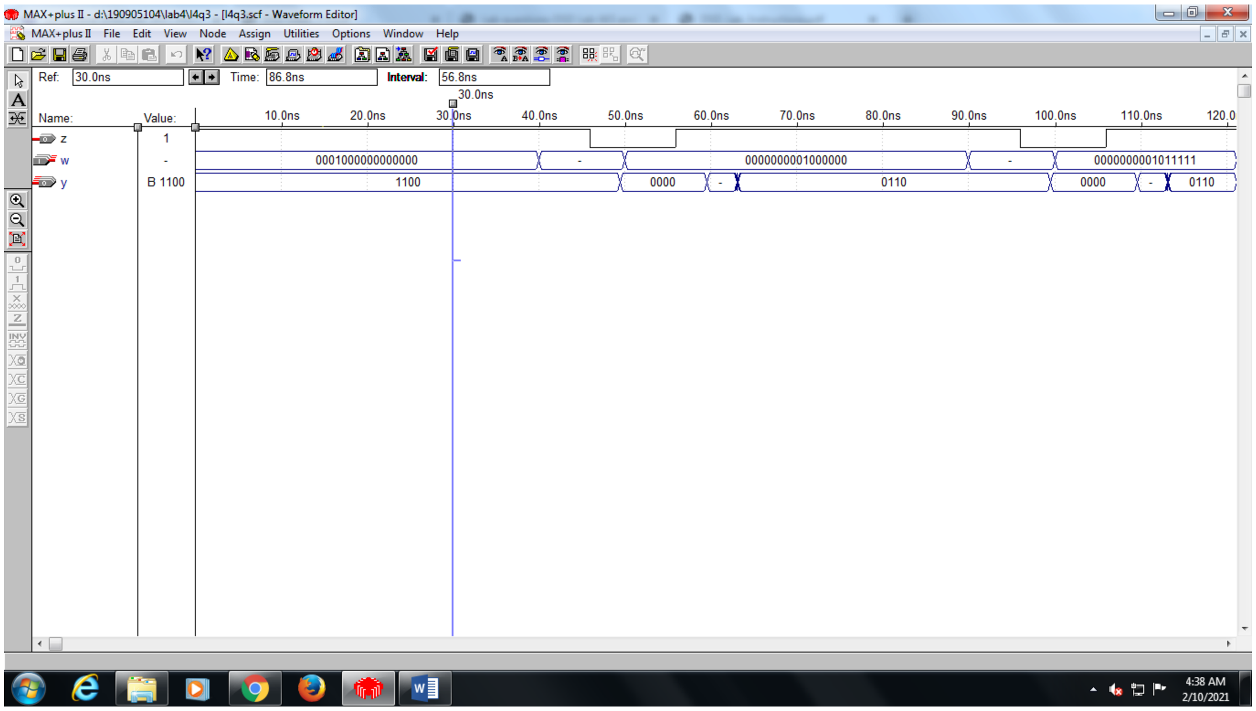

MAX+plus II - d:\190905104\lab4\l4q2 - [l4q2.scf - Waveform Editor]
MAX+plus II File Edit View Node Assign Utilities Options Window Help
4:25 AM
2/10/2021

```
Start: 40.0ns   End: 80.0ns   Interval: 40.0ns

Name:            Value:   10.0ns 20.0ns 30.0ns 40.0ns 50.0ns 60.0ns 70.0ns 80.0ns 90.0ns 100.0ns 110.0ns 120.0
en               0
w                B 0011   1111 | 0011 | 0000
y                -        1111111111111111 | 0111111111111111 | 1111111111111111 | 1111111111110111 | 1111111111111111
to43:stage0|w    B 00     11 | 00
to43:stage0|y    B XXXX   XXXX
to43:stage1|w    B 11     11 | 00
to43:stage2|w    B 11     11 | 00
to43:stage3|w    B 11     11 | 00
to43:stage4|w    B 11     11 | 00
```

3. Write behavioral Verilog code for 16 to 4 priority encoder using for loop

```
module l4q3(w, y, z);

input [15:0]w;

output [3:0]y;

output z;

reg [3:0]y;

reg z;

integer k;

always @(w)

begin

z = 0;

if(w == 0)

y = 0;

else

begin

for(k = 0 ; k < 16 ; k = k + 1)
```

```
begin
if(w[k] == 1)
y = k;
end
z = 1;
end
end
endmodule
```



MAX+plus II - d:\190905104\lab4\l4q3 - [l4q3.scf - Waveform Editor]

Ref: 30.0ns Time: 86.8ns Interval: 56.8ns

```
Name:   Value:
z       1
w       -       0001000000000000 | - | 0000000001000000 | - | 0000000001011111
y       B 1100  1100 | 0000 | - | 0110 | 0000 | - | 0110
```