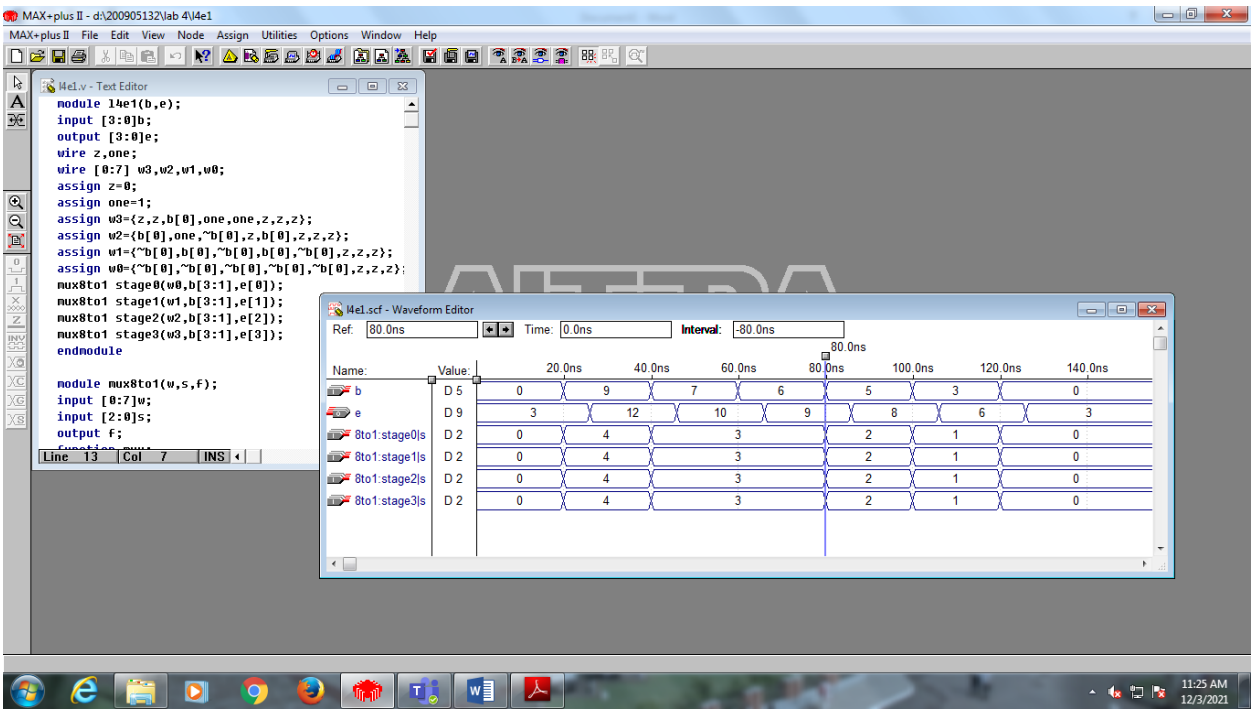
DSD Lab 4

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

Solution:

```
module l4e1(b,e);
input [3:0]b;
output [3:0]e;
wire z,one;
wire [0:7] w3,w2,w1,w0;
assign z=0;
assign one=1;
assign w3={z,z,b[0],one,one,z,z,z};
assign w2={b[0],one,~b[0],z,b[0],z,z,z};
assign w1={~b[0],b[0],~b[0],b[0],~b[0],z,z,z};
assign w0={~b[0],~b[0],~b[0],~b[0],~b[0],z,z,z};
mux8to1 stage0(w0,b[3:1],e[0]);
mux8to1 stage1(w1,b[3:1],e[1]);
mux8to1 stage2(w2,b[3:1],e[2]);
mux8to1 stage3(w3,b[3:1],e[3]);
endmodule


module mux8to1(w,s,f);
input [0:7]w;
input [2:0]s;
output f;
function mux;
input [0:7]x;
input [2:0]ss;
```

```
case(ss)
0:mux=x[0];
1:mux=x[1];
2:mux=x[2];
3:mux=x[3];
4:mux=x[4];
5:mux=x[5];
6:mux=x[6];
7:mux=x[7];
endcase
endfunction
assign f=mux(w,s);
endmodule
```



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.

Solution:

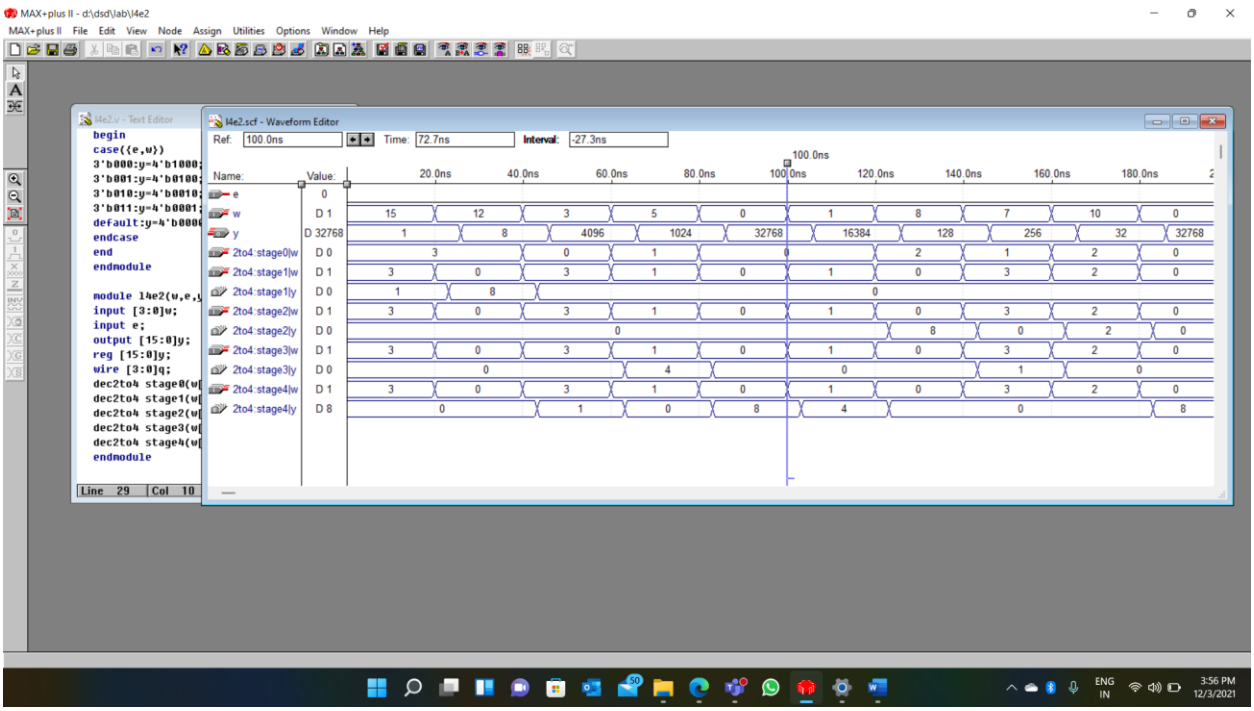
```
module dec2to4(w,e,y);
input [1:0]w;
input e;
output [3:0]y;
reg [3:0]y;
always @(w or e)
begin
case({e,w})
3'b000:y=4'b1000;
3'b001:y=4'b0100;
3'b010:y=4'b0010;
3'b011:y=4'b0001;
default:y=4'b0000;
endcase
end
endmodule

module l4e2(w,e,y);
input [3:0]w;
input e;
output [15:0]y;
reg [15:0]y;
wire [3:0]q;
dec2to4 stage0(w[3:2],e,q);
dec2to4 stage1(w[1:0],~q[0],y[3:0]);
```

dec2to4 stage2(w[1:0],~q[1],y[7:4]);

dec2to4 stage3(w[1:0],~q[2],y[11:8]);

dec2to4 stage4(w[1:0],~q[3],y[15:12]);

endmodule



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

module l4e3(w,z,y);

input [15:0]w;

output z;

output [3:0]y;

reg [3:0]y;

reg z;

integer k;

always @(w)

begin

z=0;

```
y=4'bx;
for(k=0;k<16;k=k+1)
begin
if(w[k])
begin
y=k;
z=1;
end
end
end
endmodule
```