

**WEEK 4 LAB 4**

**1:** Write and simulate the Verilog code for a BCD to Excess 3 code converter using 8 to 1

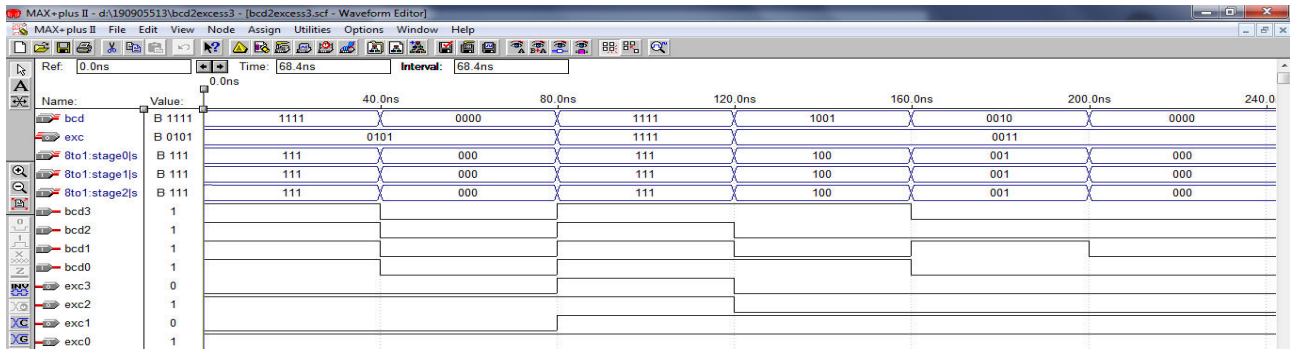
multiplexers and other necessary gates.

Program:

```
module bcd2excess3(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

module mux8to1(w,s,f);
input [0:7]w;
input [2:0]s;
output f;
reg f;
always @(w or s or f)
begin
case(s)
0:f=w[0];
1:f=w[1];
2:f=w[2];
3:f=w[3];
4:f=w[4];
5:f=w[5];
6:f=w[6];
7:f=w[7];
endcase
end
endmodule
```

**OUTPUT :**



**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 decode2to4(w,y,en);
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 dec4to16(w,y,en);
input en;

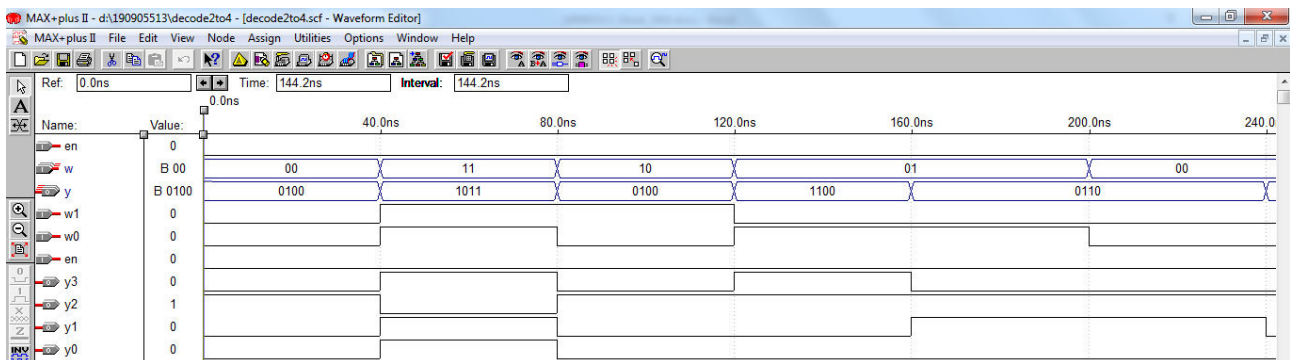
```

```

input [3:0]w;
output [15:0]y;
reg [15:0]f;
wire [3:0]x;
decode2to4 stage0(w[3:2],x[3:0],en);
decode2to4 stage1(w[1:0],f[3:0],~x[0]);
decode2to4 stage2(w[1:0],f[7:4],~x[1]);
decode2to4 stage3(w[1:0],f[11:8],~x[2]);
decode2to4 stage4(w[1:0],f[15:12],~x[3]);
assign y=~f;
endmodule

```

## OUTPUT :



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

```

module priorityencoder16to4(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

```

## OUTPUT :

