DSD LAB

Week 3

Udeet Mittal

CSE C3

Roll Number 64

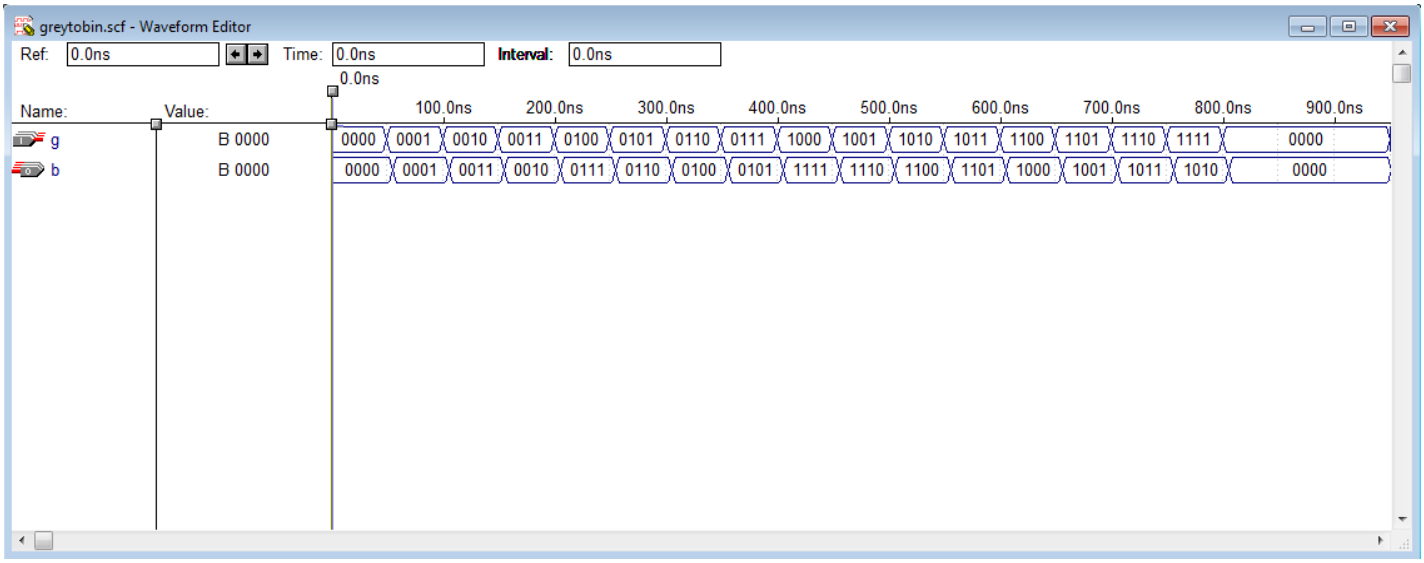
1.

1. Using **for** loop, write behavioral Verilog code to convert an N bit grey code into equivalent binary code.

Source Code:

```
module greytobin(g,b);
parameter n=4;
input [n-1:0]g;
output [n-1:0]b;
reg [n-1:0]b;
integer i;
always @(g)
begin
        b[n-1]=g[n-1];
        for(i=n-2;i>=0;i=i-1)
                b[i]=b[i+1]^g[i];
end
endmodule
```

Output Waveform:



2.

2. Write and simulate the Verilog code for a 4-bit comparator using 2-bit comparators.

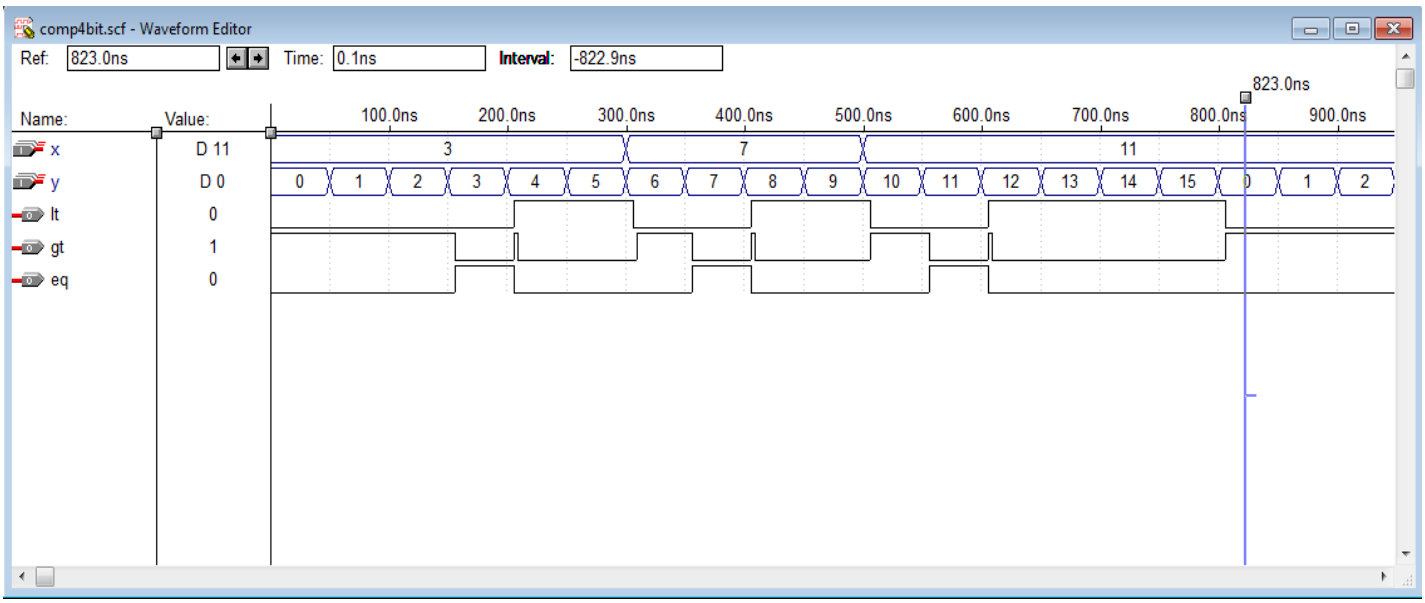
Source Code:

```
module comp4bit(x,y,eq,lt,gt);
input [3:0]x,y;
output eq,lt,gt;
wire e1,e2,l1,l2,g1,g2;
comp2bit bits32(x[3:2],y[3:2],e1,l1,g1);
comp2bit bits10(x[1:0],y[1:0],e2,l2,g2);
assign eq=e1&e2;
assign gt=g1|(e1&g2);
assign lt=l1|(e1&l2);
endmodule

module comp2bit(x,y,eq,lt,gt);
```

```
input [1:0]x,y;
output eq,lt,gt;
wire i1,i0;
assign i1=~(x[1]^y[1]);
assign i0=~(x[0]^y[0]);
assign eq=i1&i0;
assign gt=(x[1]&~y[1])|(i1 & x[0] & ~y[0]);
assign lt=~(gt|eq);
endmodule
```

Output Waveform:



3. Write behavioral Verilog code for
   - an 8 to 1 multiplexer using **case** statement
   - a 2 to 1 multiplexer using the **if-else** statement.

   Using the above modules write the hierarchical code for a 16 to 1 multiplexer.

3.

Source Code:

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

```
module mux2to1(w,s,out);
input [1:0]w;
wire [1:0]w;
input s;
output out;
reg out;
always @(w or s)
begin
if(s)
        out=w[1];
else
        out=w[0];
end
endmodule

module mux16to1(w,s,out);
input [15:0]w;
input [3:0]s;
wire [1:0]op;
output out;
mux8to1 m1(w[7:0],s[2:0],op[0]);
mux8to1 m2(w[15:8],s[2:0],op[1]);
mux2to1 m3(op,s[3],out);
endmodule
```

Output Waveform: