Lab Assignment 2

1. If a = 4'b1111 and b = -5'b00010, write the program to check what the values with 4 bits are when you want to calculate "a (+/-/*/%) b". How about b = -5'b01xz ?

```
//q1 by Angela Lo

module q1;
reg [5:0] a;
reg [5:0] b;
        initial
        begin
        a=4'b1111;
        b=-5'b00010;
        $display ("a=4'b1111, b=-5'b00010");
        $display ("a+b=%b",a+b);
        $display ("a-b=%b",a-b);
        $display ("a*b=%b",a*b);
        $display ("a%sb=%b","%",a%b);
        b=-5'b01xz;
        $display ("a=4'b1111, b=-5'b01xz");
        $display ("a+b=%b",a+b);
        $display ("a-b=%b",a-b);
        $display ("a*b=%b",a*b);
        $display ("a%sb=%b","%",a%b);
        end
endmodule
```

Simulation Results:

```
a=4'b1111, b=-5'b00010
a+b=001101
a-b=010001
a*b=100010
a%b=001111
a=4'b1111, b=-5'b01xz
a+b=xxxxxx
a-b=xxxxxx
a*b=xxxxxx
a%b=xxxxxx
            V C S   S i m u l a t i o n   R e p o r t
```

Lab Assignment 2

2. When a = 2'b1z and b = 3'b11z, verify the values for (a>b), (a>=b), (a<b) and (a<=b) by a program. If a = 4'b01xz, check again.

```
//q2 by Angela Lo

module q2;
reg [5:0] a;
reg [5:0] b;
        initial
        begin
        a=2'b1z;
        b=3'b11z;
        $display ("a=2'b1z, b=3'b11z");
        $display ("a>b=%b",a>b);
        $display ("a>=b=%b",a>=b);
        $display ("a<b=%b",a<b);
        $display ("a<=b=%b",a<=b);
        b=4'b01xz;
        $display ("a=4'b01xz,b=3'b11z");
        $display ("a>b=%b",a>b);
        $display ("a>=b=%b",a>=b);
        $display ("a<b=%b",a<b);
        $display ("a<=b=%b",a<=b);
        end
endmodule
```

Simulation Results:

```
Compiler version E-2011.03; Runtime version E-2011.03;  Oct 12 23:55 2016

a=2'b1z, b=3'b11z
a>b=x
a>=b=x
a<b=x
a<=b=x
a=4'b01xz,b=3'b11z
a>b=x
a>=b=x
a<b=x
a<=b=x
          V C S   S i m u l a t i o n   R e p o r t
```

Lab Assignment 2

3. Write a program to see results for 4 questions on "Equality Operators" page in the handout.

```
//q3 by Angela Lo

module q3();
        initial
        begin
        $display ("3'b11z===3'b11z=%b", (3'b11z===3'b11z)); // Case equality
        $display ("3'bx1z===3'b11z=%b", (3'bx1z===3'b11z));
        $display ("3'b01z!==3'b01x=%b", (3'b01z!==3'b01x)); // Case inequality
        $display ("3'b01z!==3'b01z=%b", (3'b01z!==3'b01z));
        $display ("3==5=%b", (3==5));       //Logic equality
        $display ("3==3=%b", (3==3));
        $display ("3!=5=%b", (3!=5));       //Logic inequality
        $display ("3!=3=%b", (3!=3));
        end
endmodule

~
~
```

Simulation Result:

```
Compiler version E-2011.03; Runtime version E-2011.03;  Oct 13 00:58 2016

3'b11z===3'b11z=1
3'bx1z===3'b11z=0
3'b01z!==3'b01x=1
3'b01z!==3'b01z=0
3==5=0
3==3=1
3!=5=1
3!=3=0
          V C S   S i m u l a t i o n   R e p o r t
```

Lab Assignment 2

4. Verify the results by a program for the following "A's values".

A	!A
1'bx	
1'bz	
2'b1z	
2'b0z	
2'bxz	
3'bxxx	
3'b1xx	
3'b0xx	

```
//q4 by angela Lo
`timescale 10ns/1ps

module q4;
reg [1:0]a;
reg [1:0]b;
reg [2:0]c;

        initial begin
        a=1'bx; b=2'b1z;c=3'bxxx;
#1      a=1'bz; b=2'b0z;c=3'b1xx;
#1      a=1'b1; b=2'bxz;c=3'b0xx;
#1       $finish;
end
        initial begin
        $monitor ($time, "abar=%0b, bbar=%0b, cbar=%0b",!a,!b,!c);

        end
endmodule
```

Simulation Report

```
Compiler version E-2011.03; Runtime version E-2011.03;  Oct 19 21:51 2016

                  0abar=x, bbar=0, cbar=x
                  1abar=x, bbar=x, cbar=0
                  2abar=0, bbar=x, cbar=x
$finish called from file "q4.v", line 13.
$finish at simulation time                30000
          V C S   S i m u l a t i o n   R e p o r t
```

Lab Assignment 2

5. Write a program to see what you will get for 1'bx && 2'bxz , 2'b0x || 1'bz , 2'b00 && 2'b1z and 2'b0z || 4'b01xz.

```
//q5 by Angela Lo

module q5();

        initial
        begin
        $display ("%g 1'bx&&2'bxz=%b",$time,(1'bx&&2'bxz));
        #1 $display ("%g 2'b0x||1'bz=%b",$time, (2'b0x||1'bz));
        #1 $display ("%g 2'b001&&2'b1z=%b",$time, (2'b001&&2'b1z));
        #1 $display ("%g 2'b0z||4'b01xz=%b",$time, (2'b0z||4'b01xz));
        end
endmodule
```

Simulation Report:

```
Compiler version E-2011.03; Runtime version E-2011.03;  Oct 18 23:02 2016

0 1'bx&&2'bxz=x
1 2'b0x||1'bz=x
2 2'b001&&2'b1z=1
3 2'b0z||4'b01xz=1
          V C S   S i m u l a t i o n   R e p o r t
Time: 3
```

Lab Assignment 2

6. What are the results in the following operations and verify them by Verilog code?

~4'b01xz = ?
4'b01xz & 4'bzx01 = ?
4'b01xz | 4'bzx01 = ?
4'b01xz ^ 4'bzx01 = ?
4'b01xz ^~ 4'bzx01 = ?
4'b01xz ^~ 2'bz1 = 4'b?
4'b01xz ^~ 2'bz1 = 2'b?

```
//q6 by Angela Lo

module q6();

        initial
        begin
        $display ("%g ~4'b01xz=%b",$time,(~4'b01xz));
        #1 $display ("%g 4'b01xz&4'bzx01=%b",$time,(4'b01xz&4'bzx01));
        #1 $display ("%g 4'b01xz|4'bzx01=%b",$time,(4'b01xz|4'bzx01));
        #1 $display ("%g 4'b01xz^4'bzx01=%b",$time,(4'b01xz^4'bzx01));
        #1 $display ("%g 4'b01xz^~4'bzx01=%b",$time,(4'b01xz^~4'bzx01));
        #1 $display ("%g 4'b01xz^~2'bz1=%b",$time,(4'b01xz^~2'bz1));
        #1 $display ("%g 4'b01xz^~2'bz1=%b",$time,(4'b01xz^~2'bz1));
        end
endmodule
```

Simulation Report:

```
Compiler version E-2011.03; Runtime version E-2011.03;  Oct 18 23:04 2016

0 ~4'b01xz=10xx
1 4'b01xz&4'bzx01=0x0x
2 4'b01xz|4'bzx01=x1x1
3 4'b01xz^4'bzx01=xxxx
4 4'b01xz^~4'bzx01=xxxx
5 4'b01xz^~2'bz1=10xx
6 4'b01xz^~2'bz1=10xx
          V C S   S i m u l a t i o n   R e p o r t
```

Lab Assignment 2

7. What are you going to get for "& 4'b01xz" , "~| 4'b01xz" , "^ 4'b01xz" and "~^ 4'b01xz".

```
//q7 by Angela Lo

module q7();

        initial
        begin
        $display ("%g &4'b01xz=%b",$time,(&4'b01xz));
        #1 $display ("%g ~|4'b01xz=%b",$time,(~|4'b01xz));
        #1 $display ("%g ^4'b01xz=%b",$time,(^4'b01xz));
        #1 $display ("%g ~^4'b01xz=%b",$time,(~^4'b01xz));
        end
endmodule
```

Simulation Report:

```
Compiler version E-2011.03; Runtime version E-2011.03;  Oct 18 23:06 2016

0 &4'b01xz=0
1 ~|4'b01xz=0
2 ^4'b01xz=x
3 ~^4'b01xz=x
          V C S   S i m u l a t i o n   R e p o r t
```

Lab Assignment 2

8. What are the new values after bit shifting for “4'b01xz << 1'bz” and “4'b01xz >>2'bxx” ?

```
//q8 by Angela Lo

module q8();

        initial
        begin
        $display ("%g 4'b01xz<<1'bz=%b",$time,(4'b01xz<<1'bz));
        #1 $display ("%g 4'b01xz>>2'bxx=%b",$time,( 4'b01xz>>2'bxx));
        end
endmodule
```

Simulation Result:

```
Compiler version E-2011.03; Runtime version E-2011.03;  Oct 18 23:08 2016

0 4'b01xz<<1'bz=xxxx
1 4'b01xz>>2'bxx=xxxx
          V C S   S i m u l a t i o n   R e p o r t
```

Lab Assignment 2

9. In this expression A = B ? 4'b1100 : 5'b11ZX0 and if B = 2'b1x, What is A(4-bit number)? How about B= 3'b1xz? Write a program to verify your answers.

```
//q9 by Angela Lo

module q9();
wire A;
reg [2:0]B;


        assign A = B?4'b1100:5'b11ZX0;
        initial
        begin
        B=2'b1x
        $display ("%g A=%b",$time,A);
        B=3'b1xz
        $display ("%g B=%b",$time,B);
        end
endmodule

~
```

Report:

```
Parsing design file 'q9.v'

Error-[SE] Syntax error
  Following verilog source has syntax error :
  "q9.v", 12: token is '$display'
        $display ("%g A=%b",$time,A);
                 ^
```

Lab Assignment 2

10 . Output strength

```
//q10
//diaplay output strength
// Angela Lo
`timescale 10ns/100ps
module teststrength1();
        reg a,b;
        wire y;
        buf (strong1,weak0) g1 (y,a);
        buf (weak1,strong0) g2 (y,b);

        initial
        begin
        a=1;
        b=1;
        $display (" y=%b,a=%b,b=%b",y,a,b);
        end
endmodule

module teststrength2();
        reg il,i2,ctrl;
        wire y;
        bufif0 (strong1,weak0) g1 (y,il,ctrl);
        bufif0 (strong1,weak0) g2 (y,i2,ctrl);

        initial
        begin
        ctrl=1;
        il=0;
        il=1;
        $display (" y=%b,il=%b,i2=%b,ctrl=%b",y,il,i2,ctrl);
        end
endmodule
```

Simulation

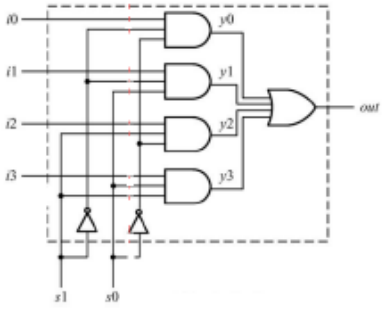
Report

```
Compiler version E-2011.03; Runtime version E-2011.03;  Oct 19 22:51 2016

 y=1,a=1,b=1
 y=z,il=1,i2=x,ctrl=1
          V C S   S i m u l a t i o n   R e p o r t
Time: 000 ps
```

Lab Assignment 2

11. Design Verilog program for 4 to 1 mux in gate level and write the testbench to verify.




```
//q11   4 to 1 mux
// uSing Gate level
//code by Angela Lo
//`timescale 10ns/1ps
module mux_4to1 (I0,I1,I2,I3,Sel0,Sel1,OUT);
        input I0,I1,I2,I3,Sel0,Sel1;
        output OUT;
wire y0,y1,y2,y3,Sel0b,Sel1b;
wire d0,d1,d2,d3,k0,k1,OUT;

not n0(Sel0b,Sel0);
not n1(Sel1b,Sel1);
and x0(y0,I0,d0);
and x1(y1,I1,d1);
and x2(y2,I2,d2);
and x3(y3,I3,d3);
and a0(do,Sel1b,Sel0b);   //decoder
and a1(d1,Sel1b,Sel0 );
and a2(d2,Sel1 ,Sel0b);
and a3(d3,Sel1 ,Sel0 );
or  b0(k0,y0,y1);
or  b1(k1,y2,y3);
or  o1(OUT,k0,k1);
endmodule
```

Lab Assignment 2

Question 11: 4 to 1 mux testbench

```
//teStbench mux_4to1

module   tb_mux_4to1 ;
         reg I0,I1,I2,I3,Sel0,Sel1;
         wire OUT;
mux_4to1 dut (I0,I1,I2,I3,Sel0,Sel1,OUT);
         initial
         begin
         I0=0;
         I1=0;
         I2=0;
         I3=0;
         Sel0=0;
         Sel1=0;
         #5 I0=0;I1=0;I2=0;I3=0;
         #5 I0=1;I1=0;I2=0;I3=0;
         #5 I0=0;I1=1;I2=0;I3=0;
         #5 I0=1;I1=1;I2=0;I3=0;
         #5 I0=0;I1=0;I2=1;I3=0;
         #5 I0=1;I1=0;I2=1;I3=0;
         #5 I0=0;I1=1;I2=1;I3=0;
         #5 I0=1;I1=1;I2=1;I3=0;
         #10 $finish;
         end
initial
         $monitor($time,, I3,I2,I1,I0,, Sel1,Sel0,, OUT);
endmodule
~
~
~
~
```

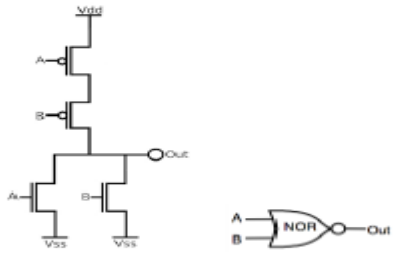
Simulation Result:

```
Compiler version E-2011.03; Runtime version E-2011.03;  Oct 19 22:56 2016

                   0 0000 00 0
                  10 0001 00 x
                  15 0010 00 0
                  20 0011 00 x
                  25 0100 00 0
                  30 0101 00 x
                  35 0110 00 0
                  40 0111 00 x
$finish called from file "tb_mux_4to1.v", line 23.
```

Lab Assignment 2

12. Write nor gate verilog module using switch devices and testbench to verify it.




```
//q12 nor gate using switch device
//Angela Lo

module mynor (input a,b,output f);
supply1 vdd;    //voltage hi
supply0 gnd;    //voltage lo

pmos upmos1(n1,vdd,a);
pmos upmos2(f,n1,b);
nmos unmos1(f,gnd,a);
nmos unmos2(f,gnd,b);
endmodule
~
~
```

Testbench

```
//tbq12.v
//Angela Lo
`include "q12.v"
module tbq12();
reg a_r,b_r;
wire f_w;
mynor dut(
          .a(a_r),
          .b(b_r),
          .f(f_w)
          );
initial begin
          a_r=1'b0; b_r=1'b0;
#1        a_r=1'b0; b_r=1'b1;
#1        a_r=1'b1; b_r=1'b0;
#1        a_r=1'b1; b_r=1'b1;
#1        $finish;
end
initial begin
          $monitor ("a=%0b,b=%0b,f_w=%0b", a_r,b_r,f_w);
end
endmodule
```