	Marwadi University **Faculty of Technology** **Department of Information and Communication Technology**	
Subject: Digital Design using Verilog (01CT0619)	**Aim: Write the Verilog code to implement logic functions using continuous assignments.**	
Experiment No: 01	**Date:**	**Enrollment No: 92100133020**

<u>Aim:</u> Write the Verilog code to implement given logic functions using continuous assignments.

g= x1x3 + x2x4

h = (x1 + x3 bar)(x2bar + x4)

f = g + h

IDE: EDA Playground

Theory:

Binary operation performs by any digital circuit with the set of elements 0 and 1 ,are called logical operations or logic functions. The algebra used to symbolically represent the logic function is called Boolean algebra. It is a two state algebra invented by George Boole in 1854.

Thus, a Boolean algebra is a system of mathematics logic for the analysis and designing of digital systems.

A variable or function of variables in Boolean algebra can assume only two values ,either a ‘0’ or a ‘1’.Hence,there are no fractions, no negative numbers, no square roots, no cube roots, no logarithms etc.

Pre Lab Exercise:

a. What is logic function?

b. What is the minimum number of two-input NAND gates used to perform the function of two input OR gate ?

c. A student makes a mistake somewhere in the process of simplifying the following Boolean expression:

$$AB + A(B + C)$$

$$AB + AB + C$$

$$AB + C$$

Determine where the mistake was made, and what the proper sequence of steps should be to simplify the original expression.

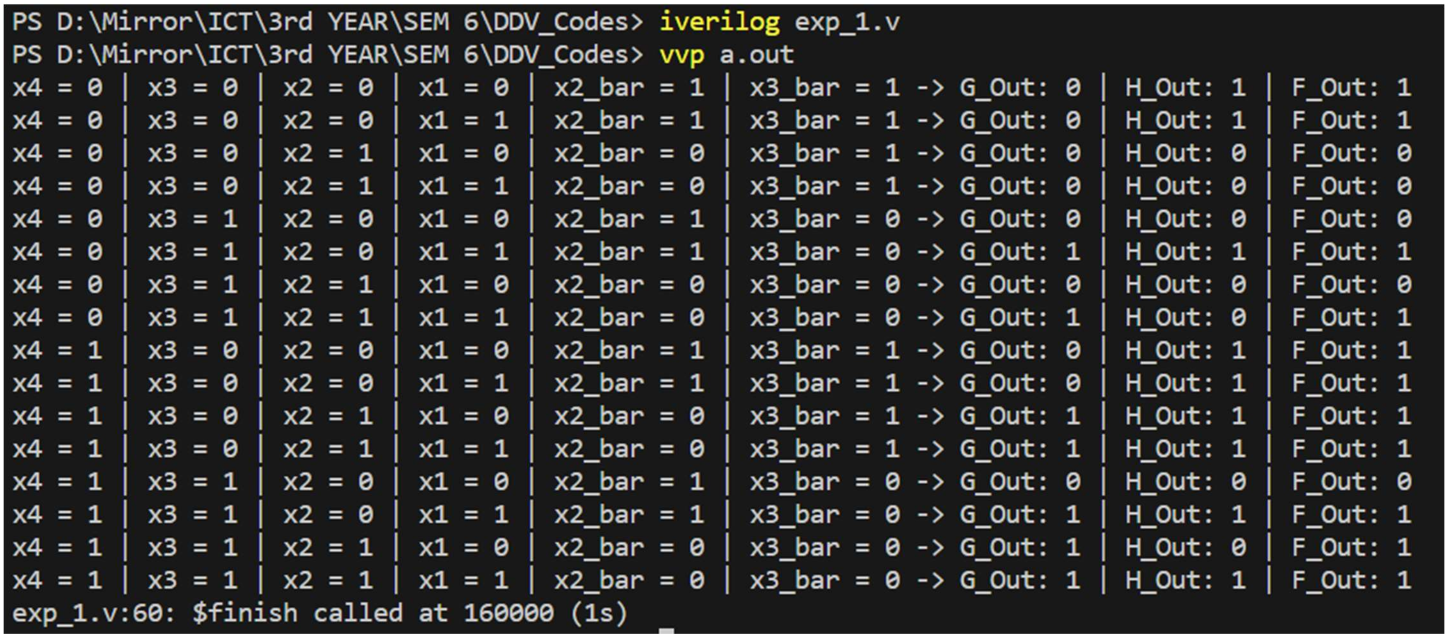
Verilog Code:

```
// Experiment 1

// Write the Verilog code to implement given logic functions using continuous assignments.

// g= x1x3 + x2x4
// h = (x1 + x3 bar)(x2bar + x4)
// f = g + h

module and_gate (in1,in2,out);
  input in1;
  input in2;
  output out;
  and(out,in1,in2);
endmodule

module or_gate (in1,in2,out);
  input in1;
  input in2;
  output out;
  or(out,in1,in2);
endmodule

module not_gate (in,out);
  input in;
  output out;
  not(out,in);
endmodule
```

Testbench:

```
// Testbench

module and_gate_tb;
  reg x1=0, x2=0, x3=0, x4=0;
  wire g_out, h_out, f_out, out1, out2, x3_bar, x2_bar, out3, out4;
  reg [4:0] inputs; // 4-bit binary counter

  and_gate a(x1, x3, out1);
  and_gate b(x2, x4, out2);
  or_gate c(out1, out2, g_out);

  not_gate d(x3, x3_bar);
  not_gate e(x2, x2_bar);

  or_gate f(x1, x3_bar, out3);
  or_gate g(x2_bar, x4, out4);
  and_gate h(out3, out4, h_out);

  or_gate i(g_out, h_out, f_out);

  initial begin
    for (inputs = 0; inputs < 16; inputs = inputs + 1) begin
        x1 = inputs[0]; // 0th bit for x1
        x2 = inputs[1]; // 1st bit for x2
        x3 = inputs[2]; // 2nd bit for x3
        x4 = inputs[3]; // 3rd bit for x4
        #10000; // Delay of 1 second

        // Display inputs and corresponding outputs
        $display("x4 = %b | x3 = %b | x2 = %b | x1 = %b | x2_bar = %b | x3_bar = %b -> G_Out: %b | H_Out: %b | F_Out: %b", x4, x3, x2, x1, x2_bar, x3_bar, g_out, h_out, f_out);
    end
    $finish;
  end
endmodule
```

Results:



```
PS D:\Mirror\ICT\3rd YEAR\SEM 6\DDV_Codes> iverilog exp_1.v
PS D:\Mirror\ICT\3rd YEAR\SEM 6\DDV_Codes> vvp a.out
x4 = 0 | x3 = 0 | x2 = 0 | x1 = 0 | x2_bar = 1 | x3_bar = 1 -> G_Out: 0 | H_Out: 1 | F_Out: 1
x4 = 0 | x3 = 0 | x2 = 0 | x1 = 1 | x2_bar = 1 | x3_bar = 1 -> G_Out: 0 | H_Out: 1 | F_Out: 1
x4 = 0 | x3 = 0 | x2 = 1 | x1 = 0 | x2_bar = 0 | x3_bar = 1 -> G_Out: 0 | H_Out: 0 | F_Out: 0
x4 = 0 | x3 = 0 | x2 = 1 | x1 = 1 | x2_bar = 0 | x3_bar = 1 -> G_Out: 0 | H_Out: 0 | F_Out: 0
x4 = 0 | x3 = 1 | x2 = 0 | x1 = 0 | x2_bar = 1 | x3_bar = 0 -> G_Out: 0 | H_Out: 0 | F_Out: 0
x4 = 0 | x3 = 1 | x2 = 0 | x1 = 1 | x2_bar = 1 | x3_bar = 0 -> G_Out: 1 | H_Out: 1 | F_Out: 1
x4 = 0 | x3 = 1 | x2 = 1 | x1 = 0 | x2_bar = 0 | x3_bar = 0 -> G_Out: 0 | H_Out: 0 | F_Out: 0
x4 = 0 | x3 = 1 | x2 = 1 | x1 = 1 | x2_bar = 0 | x3_bar = 0 -> G_Out: 1 | H_Out: 0 | F_Out: 1
x4 = 1 | x3 = 0 | x2 = 0 | x1 = 0 | x2_bar = 1 | x3_bar = 1 -> G_Out: 0 | H_Out: 1 | F_Out: 1
x4 = 1 | x3 = 0 | x2 = 0 | x1 = 1 | x2_bar = 1 | x3_bar = 1 -> G_Out: 0 | H_Out: 1 | F_Out: 1
x4 = 1 | x3 = 0 | x2 = 1 | x1 = 0 | x2_bar = 0 | x3_bar = 1 -> G_Out: 1 | H_Out: 1 | F_Out: 1
x4 = 1 | x3 = 0 | x2 = 1 | x1 = 1 | x2_bar = 0 | x3_bar = 1 -> G_Out: 1 | H_Out: 1 | F_Out: 1
x4 = 1 | x3 = 1 | x2 = 0 | x1 = 0 | x2_bar = 1 | x3_bar = 0 -> G_Out: 0 | H_Out: 0 | F_Out: 0
x4 = 1 | x3 = 1 | x2 = 0 | x1 = 1 | x2_bar = 1 | x3_bar = 0 -> G_Out: 1 | H_Out: 1 | F_Out: 1
x4 = 1 | x3 = 1 | x2 = 1 | x1 = 0 | x2_bar = 0 | x3_bar = 0 -> G_Out: 1 | H_Out: 0 | F_Out: 1
x4 = 1 | x3 = 1 | x2 = 1 | x1 = 1 | x2_bar = 0 | x3_bar = 0 -> G_Out: 1 | H_Out: 1 | F_Out: 1
exp_1.v:60: $finish called at 160000 (1s)
```

Observation and Result Analysis:

Write the final observation and Analysis

Post Lab Exercise:

a. An engineer hands you a piece of paper with the following Boolean expression on it, and tells you to build a gate circuit to perform that function:

$$A\overline{B} + \overline{C}(A + B)$$

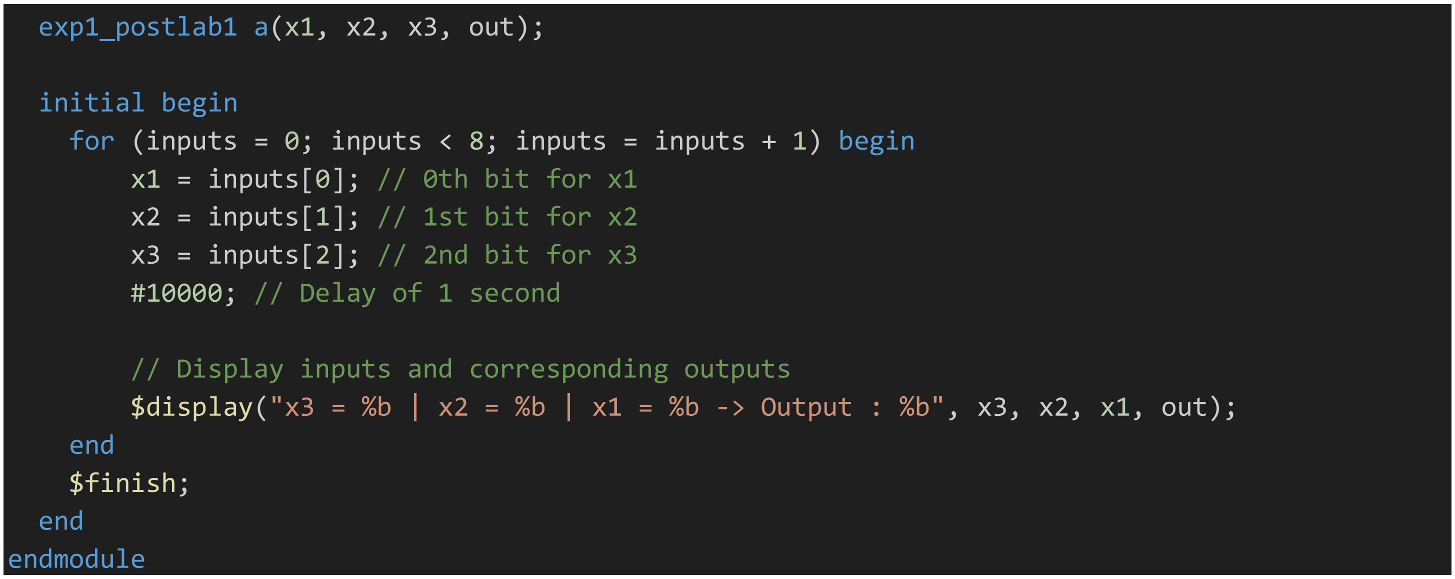
Write Verilog code and test bench for the same. Also Mention Output for the same.

Verilog Code:

```verilog
// Lab Experiment 1 - Post Lab 1
// Equation is: AB' + C'(A + B)

module exp1_postlab1(inA, inB, inC, out);
    input inA, inB, inC;
    wire w1, w2, w3, w4, w5;
    output out;

    or(w1, inA, inB);
    not(w2, inC);
    and(w3, w1, w2);
    not(w4, inB);
    and(w5, inA, w4);
    or(out, w3, w5);
endmodule
```

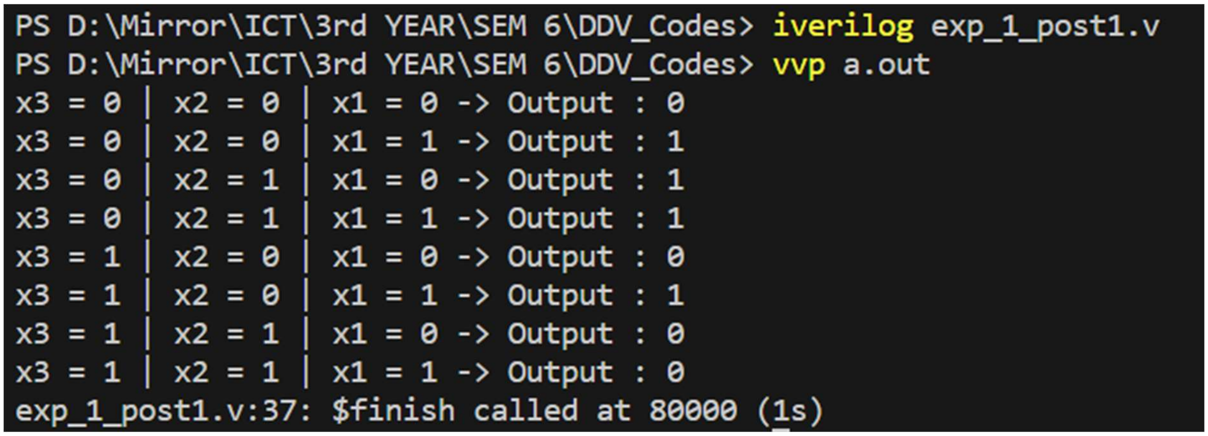
Testbench:

```verilog
// Testbench

module exp1_postlab1_tb;
  reg x1=0, x2=0, x3=0;
  wire out;
  reg [3:0] inputs; // 3-bit binary counter
```



```
  exp1_postlab1 a(x1, x2, x3, out);

  initial begin
    for (inputs = 0; inputs < 8; inputs = inputs + 1) begin
        x1 = inputs[0]; // 0th bit for x1
        x2 = inputs[1]; // 1st bit for x2
        x3 = inputs[2]; // 2nd bit for x3
        #10000; // Delay of 1 second

        // Display inputs and corresponding outputs
        $display("x3 = %b | x2 = %b | x1 = %b -> Output : %b", x3, x2, x1, out);
    end
    $finish;
  end
endmodule
```

Results:



```
PS D:\Mirror\ICT\3rd YEAR\SEM 6\DDV_Codes> iverilog exp_1_post1.v
PS D:\Mirror\ICT\3rd YEAR\SEM 6\DDV_Codes> vvp a.out
x3 = 0 | x2 = 0 | x1 = 0 -> Output : 0
x3 = 0 | x2 = 0 | x1 = 1 -> Output : 1
x3 = 0 | x2 = 1 | x1 = 0 -> Output : 1
x3 = 0 | x2 = 1 | x1 = 1 -> Output : 1
x3 = 1 | x2 = 0 | x1 = 0 -> Output : 0
x3 = 1 | x2 = 0 | x1 = 1 -> Output : 1
x3 = 1 | x2 = 1 | x1 = 0 -> Output : 0
x3 = 1 | x2 = 1 | x1 = 1 -> Output : 0
exp_1_post1.v:37: $finish called at 80000 (1s)
```

b. The Boolean function (x + y) (x̄ + z) (y+ z) is equal to which one of the following expressions?

(a) (x + y) (y + z) (b) (x̄ + z) (y+ z) (c) (x + y) (x̄ + z) (d) (x + y) (x̄ + z)

Write Verilog code for given Boolean function and the answer which you have selected from options, Comment on the output for the same.

Here the answer is option C and D, because both the options are same and after implementation of the code the outputs are also same.

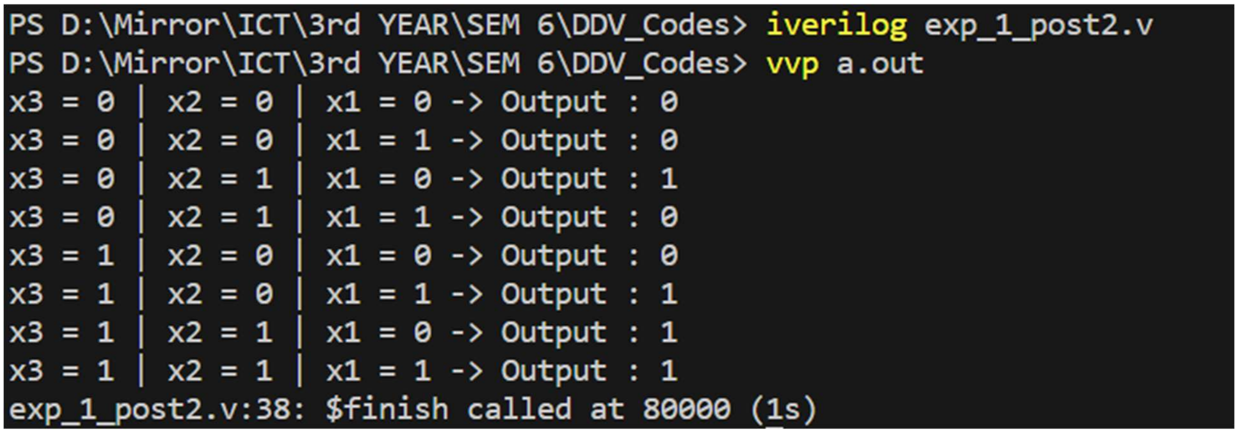
Verilog Code:

```
// Lab Experiment 1 - Post Lab 2
/* The Boolean function ( x + y) ( x̄ + z) ( y+ z) is equal to which one of the following
 expressions?
(a) (x + y)(y+z)          (b) (x̄ + z)(y+ z)          (c) (x + y)(x̄ + z)   (d) (x + y) (x̄ + z)
Write verilog code for given Boolean function and the answer which you have selected from
options, Comment  on the output for the same.*/

module exp1_postlab2(inX, inY, inZ, out);
    input inX, inY, inZ;
    wire w1, w2, w3, w4, w5;
    output out;

    not(w2, inX);
    or(w1, inX, inY);
    or(w3, inZ, w2);
    or(w4, inY, inZ);
    and(w5, w3, w4);
    and(out, w5, w1);
endmodule
```

Testbench:

```
// Testbench

module exp1_postlab2_tb;
  reg x1=0, x2=0, x3=0;
  wire out;
  reg [3:0] inputs; // 3-bit binXry counter

  exp1_postlab2 a(x1, x2, x3, out);

  initial begin
    for (inputs = 0; inputs < 8; inputs = inputs + 1) begin
        x1 = inputs[0]; // 0th bit for x1
        x2 = inputs[1]; // 1st bit for x2
        x3 = inputs[2]; // 2nd bit for x3
        #10000; // Delay of 1 second

        // Display inputs and corresponding outputs
        $display("x3 = %b | x2 = %b | x1 = %b -> Output : %b", x3, x2, x1, out);
    end
    $finish;
  end
endmodule
```

Results:



```
PS D:\Mirror\ICT\3rd YEAR\SEM 6\DDV_Codes> iverilog exp_1_post2.v
PS D:\Mirror\ICT\3rd YEAR\SEM 6\DDV_Codes> vvp a.out
x3 = 0 | x2 = 0 | x1 = 0 -> Output : 0
x3 = 0 | x2 = 0 | x1 = 1 -> Output : 0
x3 = 0 | x2 = 1 | x1 = 0 -> Output : 1
x3 = 0 | x2 = 1 | x1 = 1 -> Output : 0
x3 = 1 | x2 = 0 | x1 = 0 -> Output : 0
x3 = 1 | x2 = 0 | x1 = 1 -> Output : 1
x3 = 1 | x2 = 1 | x1 = 0 -> Output : 1
x3 = 1 | x2 = 1 | x1 = 1 -> Output : 1
exp_1_post2.v:38: $finish called at 80000 (1s)
```