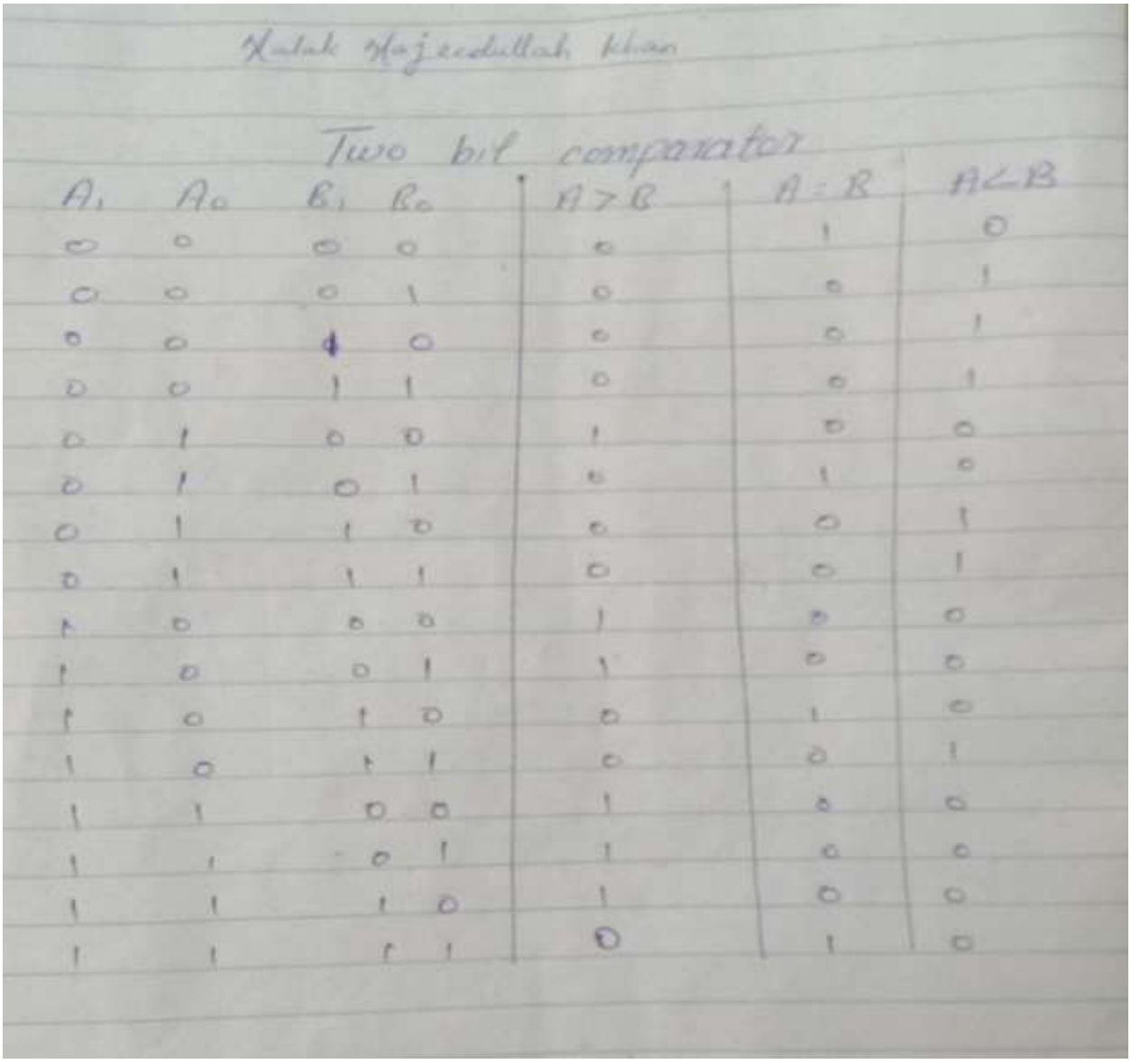
Lab 02

Handout#02: Verilog Data Flow Modelling for Basic Combinational Circuits

1. A *magnitude comparator* checks if one number is greater than or equal to or less than another number.

A 4-bit magnitude comparator takes two 4-bit numbers *A*, and *B* as input.

a. Drive Boolean expressions for the design. Provide the handwritten work.



Kalak Hajeedullah Khan

Two bit comparator

A_1	A_0	B_1	B_0	$A>B$	$A=B$	$A<B$
0	0	0	0	0	1	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	1	0	0	1	0	0
0	1	0	1	0	1	0
0	1	1	0	0	0	1
0	1	1	1	0	0	1
1	0	0	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	0	1	0
1	0	1	1	0	0	1
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	0	1	0

for $A > B$

B_1B_0 / A_1A_0	00	01	11	10
00	0	0	0	0
01	1	0	0	0
11	1	1	0	1
10	1	1	0	0

$A > B = A_1\bar{B}_1 + A_0\bar{B}_1\bar{B}_0 + A_0A_1\bar{B}_0$

for $(A < B) = \bar{A}_0\bar{A}_1B_0 + \bar{A}_1B_1 + \bar{A}_0B_0B_1$

B_1B_0 / A_1A_0	00	01	11	10
00	0	1	1	1
01	0	0	1	1
11	0	0	0	0
10	0	0	1	0

For $A = B$

B_1B_0 / A_1A_0	00	01	11	10
00	1	0	0	0
01	0	1	0	0
11	0	0	1	0
10	0	0	0	1

$A = B \Rightarrow \bar{A}_0\bar{A}_1\bar{B}_0\bar{B}_1 + A_0\bar{A}_1\bar{B}_0B_1 + A_0A_1B_0B_1 + \bar{A}_0A_1\bar{B}_0B_1$

b. Write the Verilog description of the module *magnitude_comparator* using bitwise operators.
Provide the codes with name signature.

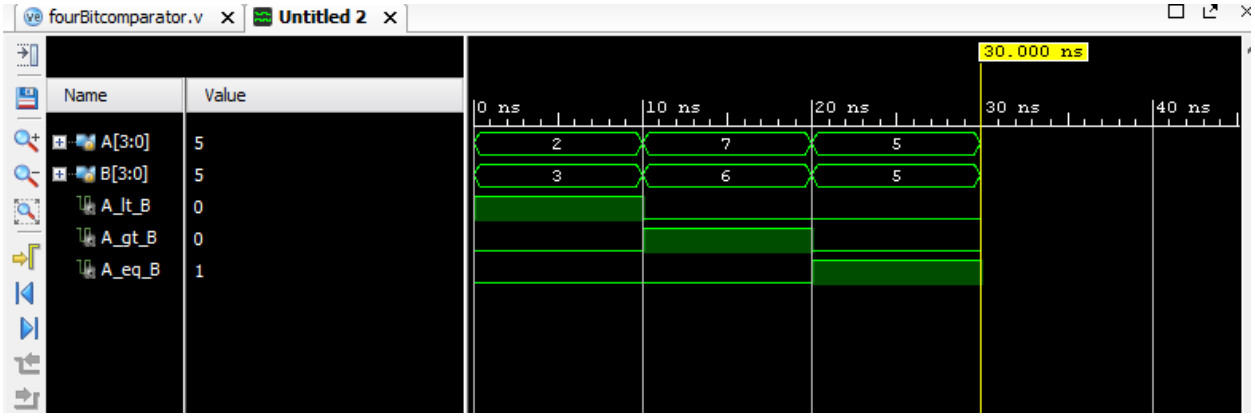
```
// Malak Majeedullah khan

module fourBitcomparator(A_lt_B, A_gt_B, A_eq_B, A, B); //declaring module
input [3:0] A, B; // declaring aray input
output A_lt_B, A_gt_B, A_eq_B; // declaring array output
assign A_lt_B = A < B; // less than operator
assign A_gt_B = A > B; // greater than operator
assign A_eq_B = A == B; // equal to operator
endmodule
```

c. Test the design using Verilog test bench. Provide the codes with the name signature, along with
tcl console and timing diagram results.

```
module Comparator_4bit_TB();
reg [3:0] A, B;
wire A_lt_B, A_gt_B, A_eq_B;
// instantiation
fourBitcomparator inst(A_lt_B, A_gt_B, A_eq_B, A, B);
//
initial
begin
#00 A = 2; B = 3; // A < B condition
#10 A = 7; B = 6 ; // A > B condition
#10 A = 5; B = 5; // A = B condition
#10 $stop;
end
initial
begin
$display("Author:Malak Majeedullah khan Khan\n");

$monitor($time,"ns -- A=%d, B=%d, A_lt_B=%b, A_gt_B=%b,A_eq_B=%b" , A,B,A_lt_B, A_gt_B, A_eq_B);
end
endmodule
```

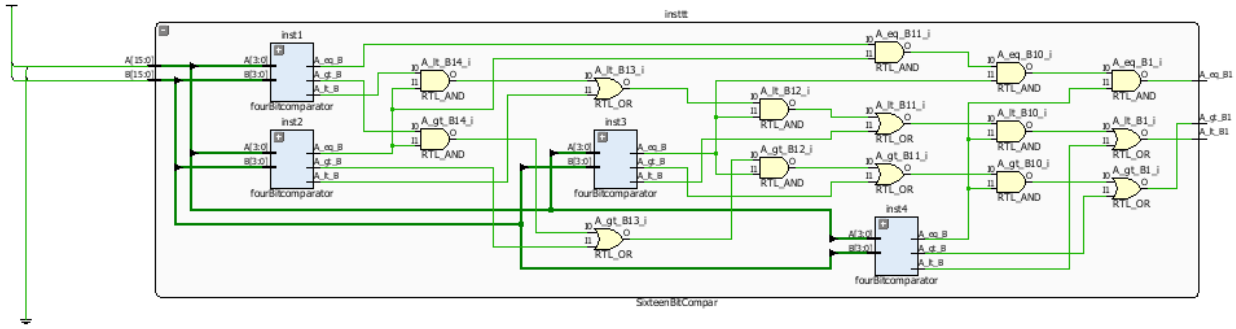
```
# }
# run 1000ns
Author:Malak Majeedullah khan Khan

                   0ns -- A= 2, B= 3, A_lt_B=1, A_gt_B=0,A_eq_B=0
                  10ns -- A= 7, B= 6, A_lt_B=0, A_gt_B=1,A_eq_B=0
                  20ns -- A= 5, B= 5, A_lt_B=0, A_gt_B=0,A_eq_B=1
xsim: Time (s): cpu = 00:00:07 ; elapsed = 00:00:06 . Memory (MB): peak = 860.563 ; gain = 0.000
INFO: [USF-XSim-96] XSim completed. Design snapshot 'Comparator_4bit_TB_behav' loaded
```

d. Make use of comments and indentations while coding.



2. Design 16-bit comparator by cascading method.

a. Provide the design schematic



b. Implement the design and provide the codes.

F:/8th semster/DSD/lab 01/lab_02/lab_02.srcs/sources_1/new/SixteenBitCompar.v

```
21
22      // malak Majeedullah khan
23
24      module SixteenBitCompar(A_lt_B1, A_gt_B1, A_eq_B1, A, B); //declering module
25
26      input [15:0] A, B; // declaring array input
27      output A_lt_B1, A_gt_B1, A_eq_B1; // declaring array output
28      wire [3:0] A_lt_B, A_gt_B, A_eq_B ;
29
30
31      // instantiating the five fourbitComparartor functions
32      fourBitcomparator inst1(A_lt_B[0], A_gt_B[0], A_eq_B[0], A[3:0], B[3:0]);
33      fourBitcomparator inst2(A_lt_B[1], A_gt_B[1], A_eq_B[1], A[7:4], B[7:4]);
34      fourBitcomparator inst3(A_lt_B[2], A_gt_B[2], A_eq_B[2], A[11:8], B[11:8]);
35      fourBitcomparator inst4(A_lt_B[3], A_gt_B[3], A_eq_B[3], A[15:12], B[15:12]);
36
37
38      assign A_eq_B1 =(A_eq_B[0] & A_eq_B[1] & A_eq_B[2] & A_eq_B[3]);
39      assign A_lt_B1 = ( ( ( ( ((A_lt_B[0]& A_eq_B[1]) | (A_lt_B[1])) & A_eq_B[2] ) | (A_lt_B[2]) ) & (A_eq_B[3])) | (A_lt_B[3]) );
40      assign A_gt_B1 = ( ( ( ( ((A_gt_B[0]& A_eq_B[1]) | (A_gt_B[1])) & A_eq_B[2] ) | (A_gt_B[2]) ) & (A_eq_B[3])) | (A_gt_B[3]) );
41
42
43      endmodule
44
```

c. Write down the test bench and provide tcl console and timing diagrams.

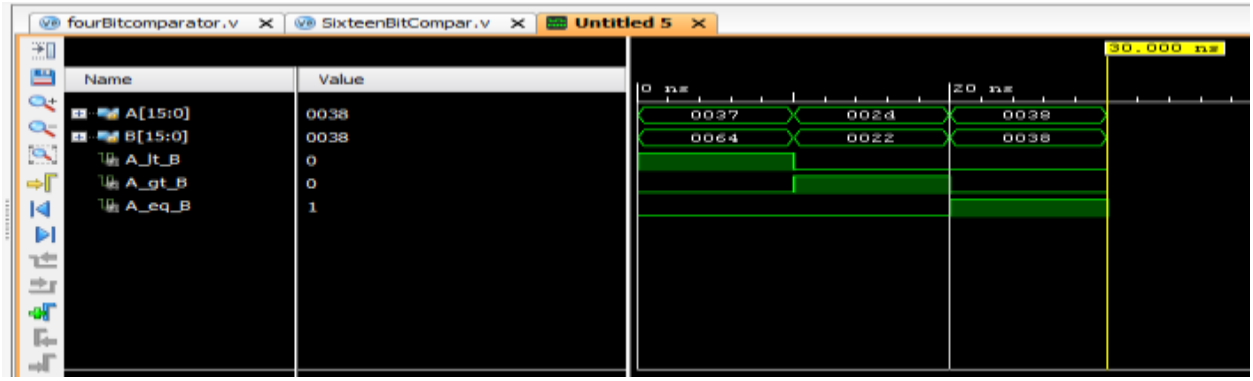
fourBitcomparator.v | SixteenBitCompar.v | Untitled 4

F:/8th semster/DSD/lab 01/lab_02/lab_02.srcs/sources_1/new/SixteenBitCompar.v

```
50
51  module sixteenBitTb();
52  reg [15:0] A, B;
53  wire A_lt_B, A_gt_B, A_eq_B;
54  // instantiation
55  SixteenBitCompar insttt(A_lt_B, A_gt_B, A_eq_B, A, B);
56  //
57  initial
58  begin
59  // #00 A = 16'b1000110011011101; B = 16'b1000110011011111; // A < B condition
60  // #10 A = 16'b1000110011011111; B = 16'b1000110011011101; // A > B condition
61  // #10 A = 16'b1000110011011111; B = 16'b1000110011011111;; // A = B condition
62  #00 A = 55; B = 100; // A < B condition
63  #10 A = 45; B = 34; // A > B condition
64  #10 A = 56; B = 56;; // A = B condition
65  #10 $stop;
66  end
67  initial
68  begin
69  $display("Author:Malak Majeedullah khan\n");
70  $monitor($time,"ns -- A=%d, B=%d, A_lt_B=%b, A_gt_B=%b,A_eq_B=%b" , A,B,A_lt_B, A_gt_B, A_eq_B);
71  end
72  endmodule
```

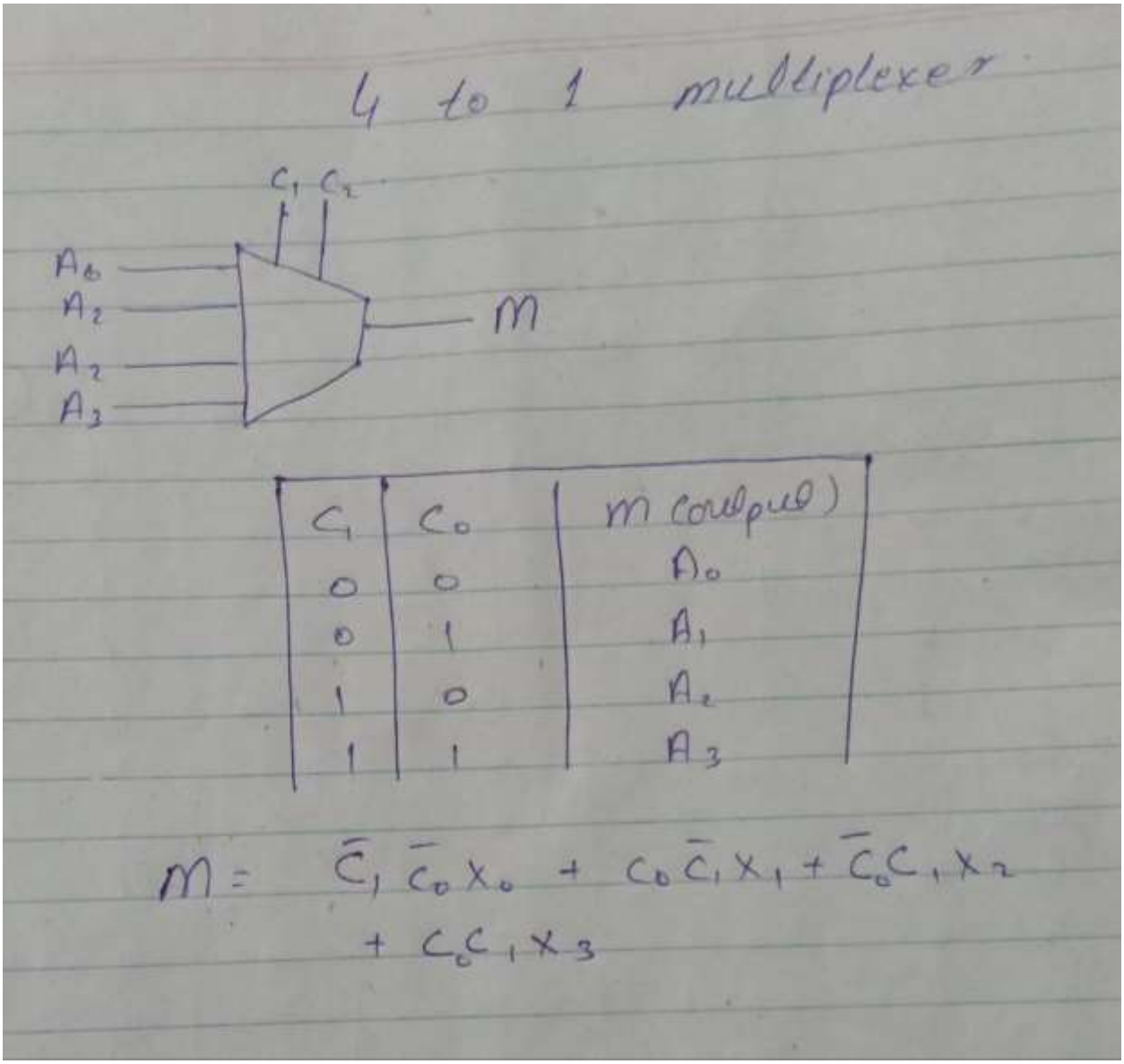
```
# run 1000ns
Author:Malak Majeedullah khan Khan

                 0ns -- A= 2, B= 3, A_lt_B=1, A_gt_B=0,A_eq_B=0
                10ns -- A= 7, B= 6, A_lt_B=0, A_gt_B=1,A_eq_B=0
                20ns -- A= 5, B= 5, A_lt_B=0, A_gt_B=0,A_eq_B=1
xsim: Time (s): cpu = 00:00:07 ; elapsed = 00:00:06 . Memory (MB): peak = 860.563 ; gain = 0.000
INFO: [USF-XSim-96] XSim completed. Design snapshot 'Comparator_4bit_TB_behav' loaded.
```

d. Make use of comments and indentations while coding.



3. A full subtracter has three 1-bit inputs x,y, and z (previous borrow) and two 1-bit outputs D (difference) and B (borrow).

a. Draw the truth table for the given circuit.



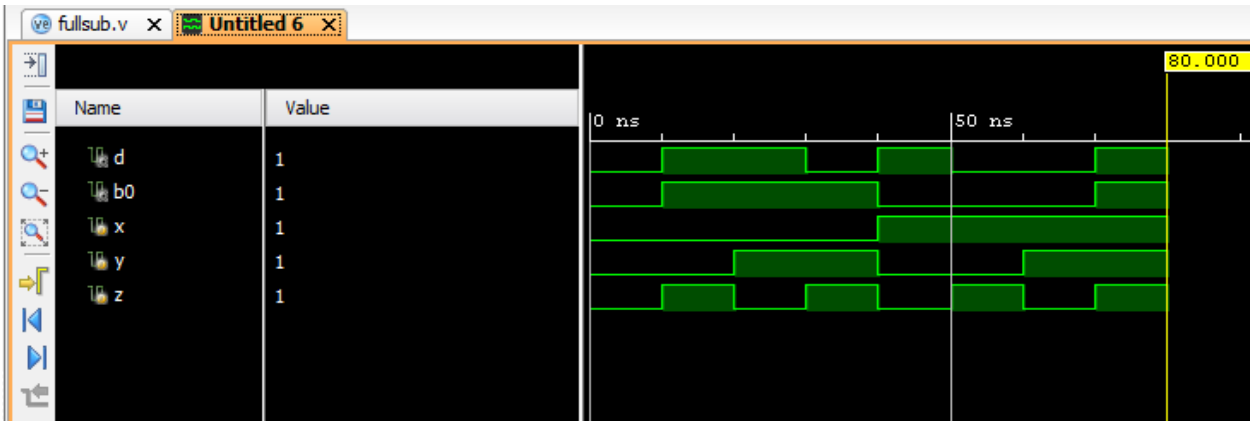
b. Drive the simplified Boolean expression.
Ans: simplified equation is given in the page

c. Write the Verilog description for the subtracter module.

```
// Malak Majeed Ullah khan

module fullsub(d, b0, x, y, z); // declaring the fullsub module
output d, b0; // declaring the outputs
input x, y, z; //declaring inputs

assign d=x^y^z; // assigning function for output d
assign b0=(((~(x))&y)|(((~(x))|y)&z)); // assigning function for output b0

endmodule
```

d. Instantiate the design module in the test module and provide the tcl console and timing diagram
outputs.

```
module fullsubTb();
wire d, b0;
reg x, y, z;
fullsub abc(d, b0, x, y, z);
initial
begin
#00 x=1'b0; y=1'b0; z=1'b0;
#10 x=1'b0; y=1'b0; z=1'b1;
#10 x=1'b0; y=1'b1; z=1'b0;
#10 x=1'b0; y=1'b1; z=1'b1;
#10 x=1'b1; y=1'b0; z=1'b0;
#10 x=1'b1; y=1'b0; z=1'b1;
#10 x=1'b1; y=1'b1; z=1'b0;
#10 x=1'b1; y=1'b1; z=1'b1;
#10 $finish;
end
initial
begin
$display("Auther: Malak Majeed Ullah khan\n");
$monitor($time, "ns -- x=%b, y=%b, z=%b, d=%b, b0=%b", x, y, z, d, b0);
end
endmodule
```




```
Auther: Malak Majeed Ullah khan

                  0ns -- x=0, y=0, z=0, d=0, b0=0
                 10ns -- x=0, y=0, z=1, d=1, b0=1
                 20ns -- x=0, y=1, z=0, d=1, b0=1
                 30ns -- x=0, y=1, z=1, d=0, b0=1
                 40ns -- x=1, y=0, z=0, d=1, b0=0
                 50ns -- x=1, y=0, z=1, d=0, b0=0
                 60ns -- x=1, y=1, z=0, d=0, b0=0
                 70ns -- x=1, y=1, z=1, d=1, b0=1
$finish called at time : 80 ns : File "F:/8th semster/DSD/lab 01/lab_02/lab_02.srcs/sources_1/new/fullsub.v" Line 48
INFO: [USF-XSim-96] XSim completed. Design snapshot 'fullsubTb behav' loaded.
```