Open | alu.v ~/Desktop/alu | Save

```
1 `timescale 1ns/1ps
2 module alu (a,b,c,n);
3 input [31:0]a;
4 input [31:0]b;
5 output reg [31:0] c;
6 input [3:0] n;
7 always@(*) begin
8 $display("Desr user enter 1 for additon operation");
9 $display("Dear user enter 2 for subtraction operation");
10 $display("Dear user enter 3 for multiplication  operation");
11 $display("Dear user enter 4 for division Quotient  operation");
12 $display("Dear user enter 5 for logical AND operation");
13 $display("Dear user enter 6 for  logical OR operation");
14 $display("Dear user enter 7 for logical NAND operation");
15 $display("Dear user enter 8 for logical NOR operation");
16 $display("Dear user enter  9 for logical XOR  operation");
17 $display (" Dear user enter 10 for equality operation");
18 if(n==1) begin
19 c=a+b;
20 $monitor("sum of the two munbers is %d",c);
21 end
22 else if(n==2) begin
23 c=a-b;
24 $monitor("Difference of two numbers is %d",c); end
25 else if(n==3) begin
26 c=a*b;
27 $monitor("Product of two numbers is %d ",c);
28 end
29 else if(n==4)begin
30 c=a/b;
```

Verilog | Tab Width: 8 | Ln 33, Col 20 | INS

Open | alu.v ~/Desktop/alu | Save

```
30 c=a/b;
31 $monitor(" a divided by b is %d",c);
32 end
33 else if(n==5) begin
34 c=a&b;
35 $monitor( "The value returns for logical AND operation between A and B is %b",c);
36 end
37 else if(n==6) begin
38 c=a|b;
39 $monitor( "The value returns for logical OR operation between A and B is %b",c);
40 end
41 else if (n==7) begin
42 c=~(a&b);
43 $monitor( "The value returns for logical NAND operation between A and B is %b",c);
44 end
45 else if(n==8)begin
46 c=~(a|b);
47 $monitor( "The value returns for logical NOR operation between A and B is %b",c);
48 end
49 else if (n==9) begin
50 c=a^b;
51 $monitor( "The value returns for logical XOR operation between A and B is %b",c);
52 end
53 else if (n==10) begin
54 c=(a==b);
55 $monitor( "The value returns for A is equal to B is  %b",c);
56 end
57 else
58 begin
59 $monitor("Invalid operation");
```

Verilog | Tab Width: 8 | Ln 33, Col 20 | INS

Open

alu.v
~/Desktop/alu

Save

```
end
else if(n==6) begin
c=a|b;
$monitor( "The value returns for logical OR operation between A and B is %b",c);
end
else if (n==7) begin
c=~(a&b);
$monitor( "The value returns for logical NAND operation between A and B is %b",c);
end
else if(n==8)begin
c=~(a|b);
$monitor( "The value returns for logical NOR operation between A and B is %b",c);
end
else if (n==9) begin
c=a^b;
$monitor( "The value returns for logical XOR operation between A and B is %b",c);
end
else if (n==10) begin
c=(a==b);
$monitor( "The value returns for A is equal to B is  %b",c);
end
else
begin
$monitor("Invalid operation");
end
end
endmodule
```

Verilog Tab Width: 8 Ln 33, Col 20 INS

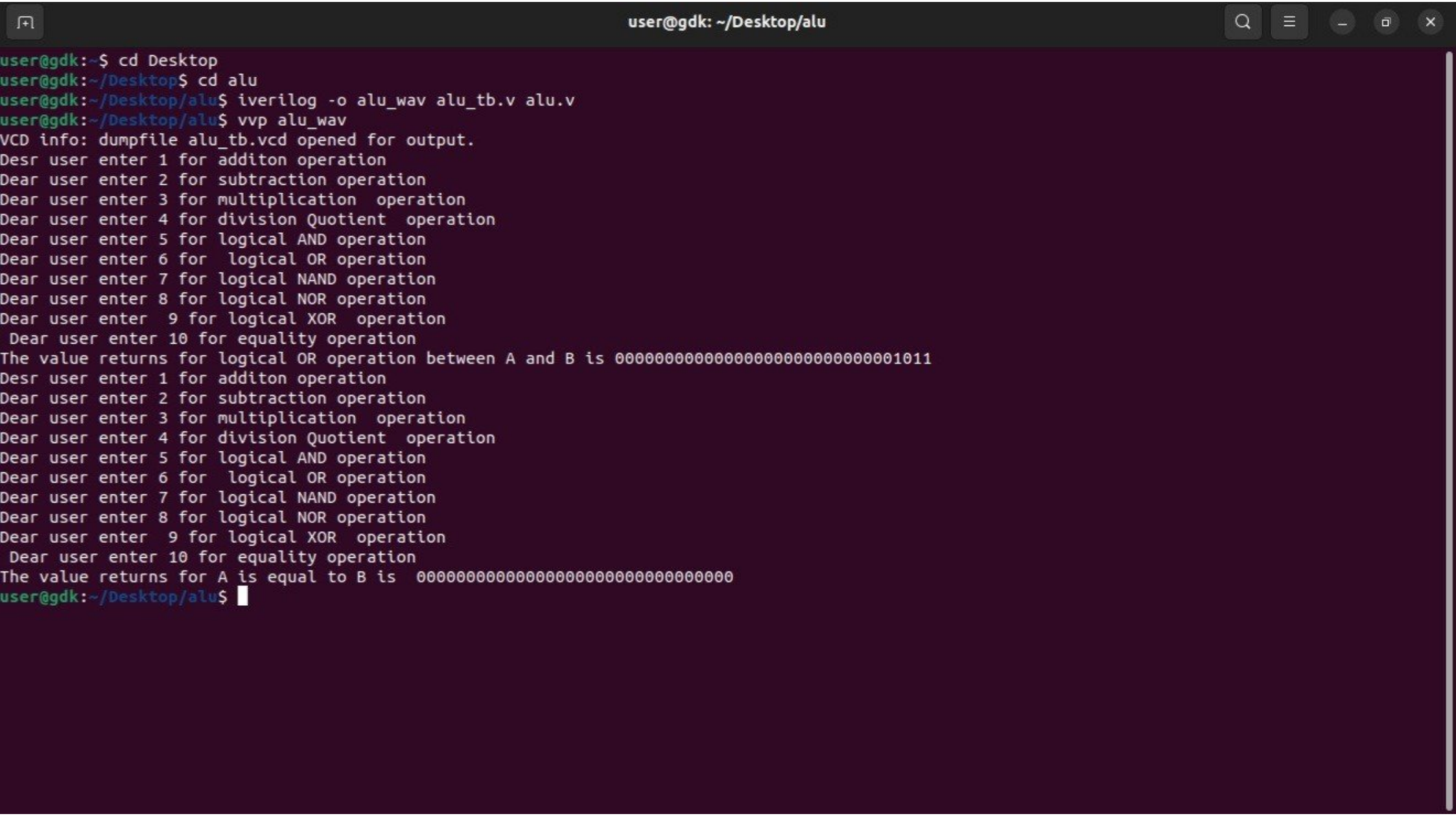
Open

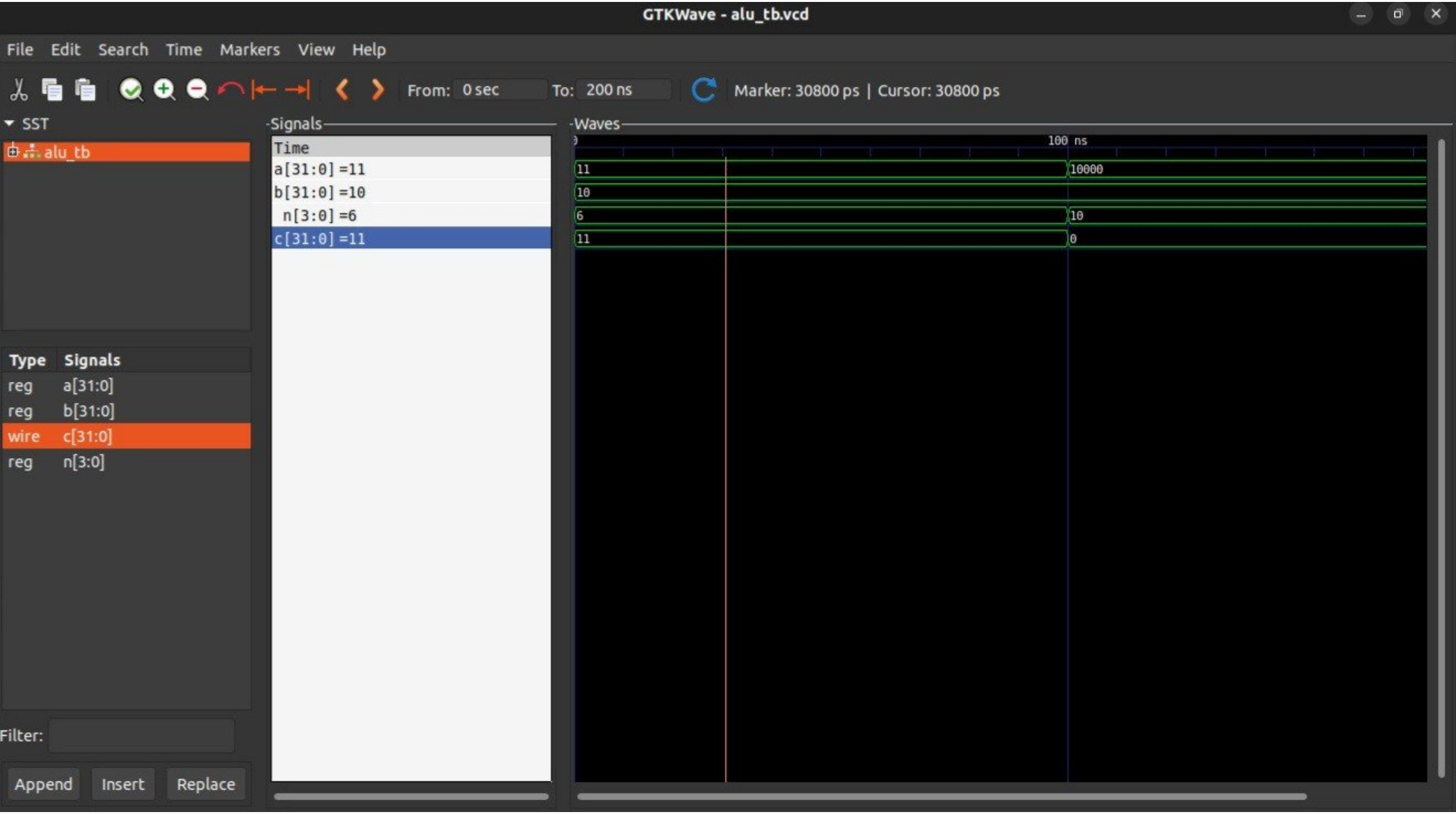
alu_tb.v
~/Desktop/alu

Save

```
1 `timescale 1ns/1ps
2 module alu_tb;
3 wire [31:0] c;
4 reg [31:0] a,b;
5 reg [3:0] n;
6 alu u0 (
7         .a(a),
8         .b(b),
9         .c(c),
10        .n(n));
11        initial begin
12        $dumpfile("alu_tb.vcd");
13        $dumpvars(0,alu_tb);
14        n=4'd6;a=32'b1011 ;b=32'b1010;#100;
15        n=4'd10;a=32'd10000;b=32'd10;#100;
16        end
17        endmodule
```

Verilog Tab Width: 8 Ln 1, Col 19 INS


user@gdk: ~/Desktop/alu
user@gdk:~$ cd Desktop
user@gdk:~/Desktop$ cd alu
user@gdk:~/Desktop/alu$ iverilog -o alu_wav alu_tb.v alu.v
user@gdk:~/Desktop/alu$ vvp alu_wav
VCD info: dumpfile alu_tb.vcd opened for output.
Desr user enter 1 for additon operation
Dear user enter 2 for subtraction operation
Dear user enter 3 for multiplication operation
Dear user enter 4 for division Quotient operation
Dear user enter 5 for logical AND operation
Dear user enter 6 for logical OR operation
Dear user enter 7 for logical NAND operation
Dear user enter 8 for logical NOR operation
Dear user enter 9 for logical XOR operation
Dear user enter 10 for equality operation
The value returns for logical OR operation between A and B is 00000000000000000000000000001011
Desr user enter 1 for additon operation
Dear user enter 2 for subtraction operation
Dear user enter 3 for multiplication operation
Dear user enter 4 for division Quotient operation
Dear user enter 5 for logical AND operation
Dear user enter 6 for logical OR operation
Dear user enter 7 for logical NAND operation
Dear user enter 8 for logical NOR operation
Dear user enter 9 for logical XOR operation
Dear user enter 10 for equality operation
The value returns for A is equal to B is 00000000000000000000000000000000
user@gdk:~/Desktop/alu$


GTKWave - alu_tb.vcd
File Edit Search Time Markers View Help
From: 0 sec
To: 200 ns
Marker: 30800 ps | Cursor: 30800 ps
SST
alu_tb
Type Signals
reg a[31:0]
reg b[31:0]
wire c[31:0]
reg n[3:0]
Filter:
Append Insert Replace
Signals
Time
a[31:0] =11
b[31:0] =10
n[3:0] =6
c[31:0] =11
Waves
100 ns
11
10000
10
6
10
11
0