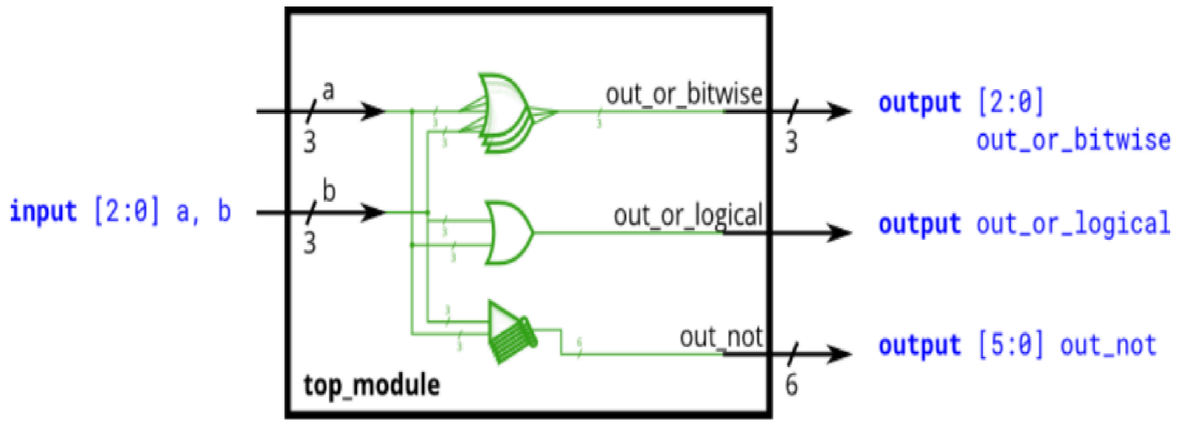
KARTHICK RAJA B
126004121

Given:
Build a circuit that has two 3-bit inputs that computes the bitwise-OR of the two vectors, the logical-OR of the two vectors, and the inverse (NOT) of both vectors. Place the inverse of b in the upper half of out_not (i.e., bits [5:3]), and the inverse of a in the lower half.

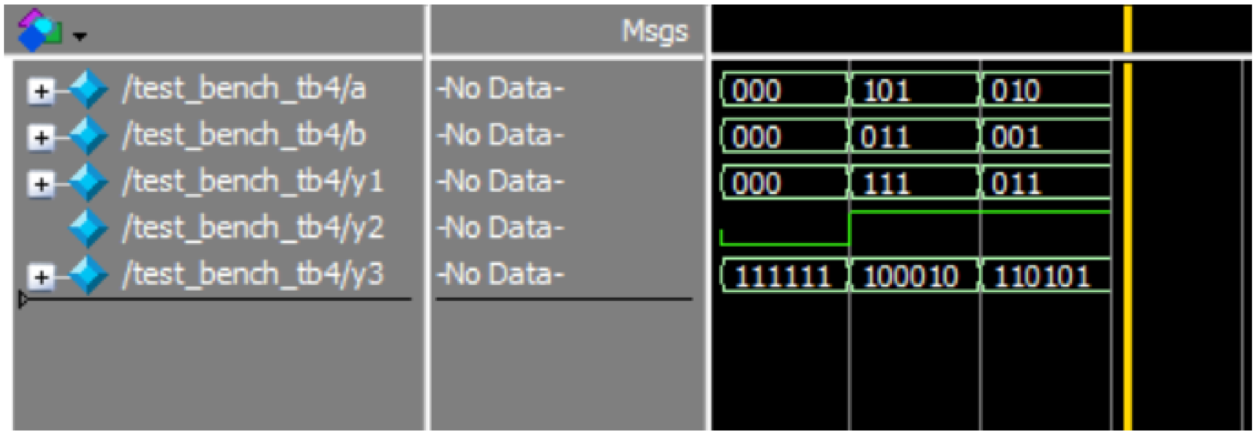


Verilog Code:
```
module RTL_Day_4(y1,y2,y3,a,b);
input[2:0] a,b;
output[2:0] y1;
output y2;
output[5:0] y3;
assign y1=(a | b);   // bitwise OR
assign y2=(a || b);  // logical OR
not n1(y3[0],a[0]);
not n2(y3[1],a[1]);
not n3(y3[2],a[2]);
not n4(y3[3],b[0]);
not n5(y3[4],b[1]);
not n6(y3[5],b[2]);
endmodule
```

Testbench Code:

```
module test_bench_tb4();
reg [2:0]a,b;
wire[2:0] y1;
wire y2;
wire[5:0] y3;

RTL_Day_4 t4(y1,y2,y3,a,b);
initial
begin

// I have given two combinations for 1st o/p

a[0]=1'b0; b[0]=1'b0; a[1]=1'b0; b[1]=1'b0; a[2]=1'b0; b[2]=1'b1; #100;
a[0]=1'b0; b[0]=1'b0; a[1]=1'b0; b[1]=1'b1; a[2]=1'b0; b[2]=1'b1; #100;

// I have given two combinations for 2nd o/p
a[0]=1'b0; b[0]=1'b0; a[1]=1'b1; b[1]=1'b0; a[2]=1'b0; b[2]=1'b1; #100;
a[0]=1'b0; b[0]=1'b0; a[1]=1'b1; b[1]=1'b1; a[2]=1'b0; b[2]=1'b1; #100;

// I have given another two  combinations for 3rd o/p

a[0]=1'b0; b[0]=1'b1; a[1]=1'b0; b[1]=1'b1; a[2]=1'b0; b[2]=1'b1; #100;
a[0]=1'b1; b[0]=1'b1; a[1]=1'b1; b[1]=1'b0; a[2]=1'b0; b[2]=1'b1;

end
endmodule
```

Functional Simulation:


Msgs
/test_bench_tb4/a
/test_bench_tb4/b
/test_bench_tb4/y1
/test_bench_tb4/y2
/test_bench_tb4/y3
-No Data-
-No Data-
-No Data-
-No Data-
-No Data-
000
101
010
000
011
001
000
111
011
111111
100010
110101