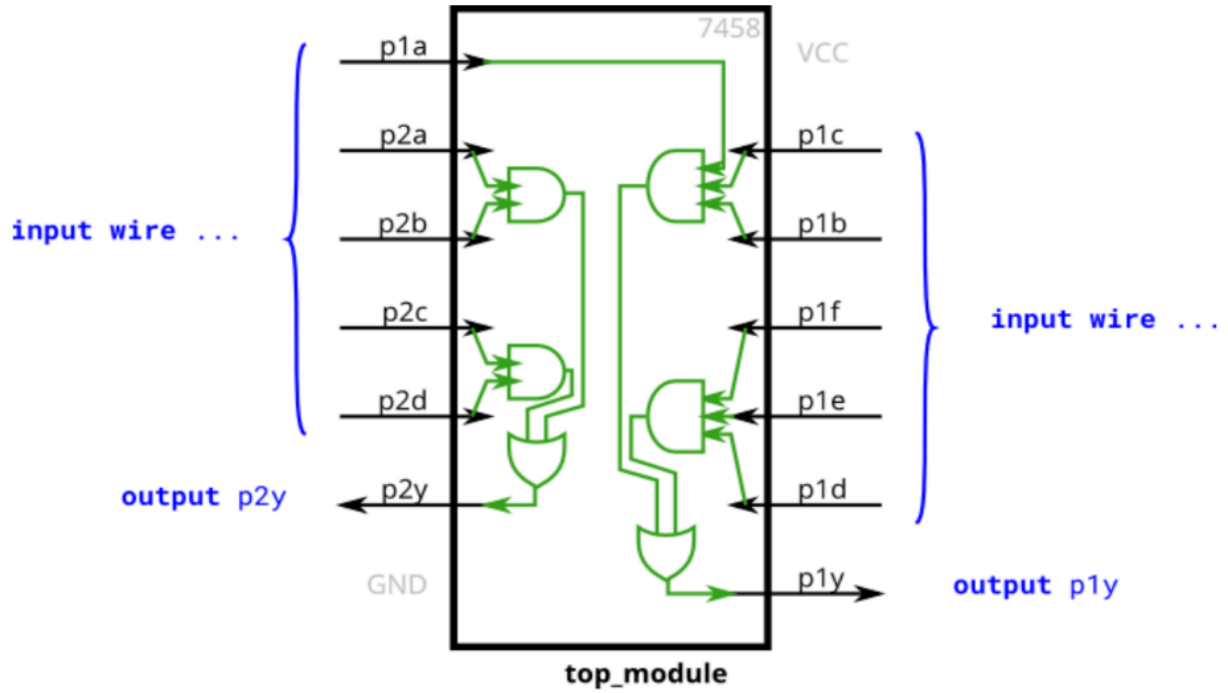
KARTHICK RAJA B
126004121

Given:

The 7458 is a chip with four AND gates and two OR gates. Create a module with the same functionality as the 7458 chip. It has 10 inputs and 2 outputs.



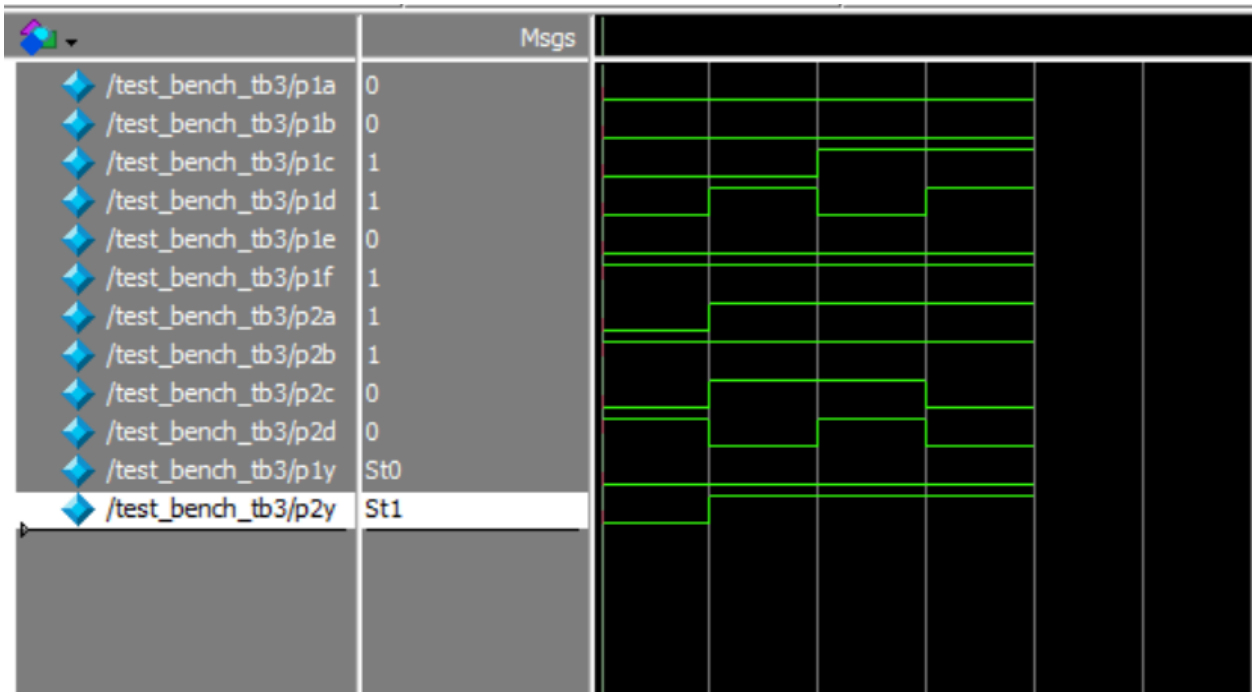
Verilog Code:

```
module RTL_Day_3(p1y,p2y,p1a,p1b,p1c,p1d,p1e,p1f,p2a,p2b,p2c,p2d);
output p1y,p2y;
input p1a,p1b,p1c,p1d,p1e,p1f,p2a,p2b,p2c,p2d;
wire w1,w2,w3,w4;
assign w1=(p1a & p1b & p1c);
assign w2=(p1d & p1e & p1f);
assign p1y=(w1 | w2);
assign w3=(p2a & p2b);
assign w4=(p2c & p2d);
assign p2y=(w3 | w4);
endmodule
```

Testbench code :

```
module test_bench_tb3();
reg p1a,p1b,p1c,p1d,p1e,p1f,p2a,p2b,p2c,p2d;
wire p1y,p2y;

RTL_Day_3 t3(p1y,p2y,p1a,p1b,p1c,p1d,p1e,p1f,p2a,p2b,p2c,p2d);
initial
begin

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

// i/p = 000001
p1a=1'b0; p1b=1'b0; p1c=1'b0; p1d=1'b0; p1e=1'b0; p1f=1'b1; #50;
// i/p = 000101
p1a=1'b0; p1b=1'b0; p1c=1'b0; p1d=1'b1; p1e=1'b0; p1f=1'b1; #50;
// i/p = 001001
p1a=1'b0; p1b=1'b0; p1c=1'b1; p1d=1'b0; p1e=1'b0; p1f=1'b1; #50;
// i/p = 001101
p1a=1'b0; p1b=1'b0; p1c=1'b1; p1d=1'b1; p1e=1'b0; p1f=1'b1; #50;

// I have given another four combinations for 2nd o/p

p2a=1'b0; p2b=1'b1; p2c=1'b0; p2d=1'b1; #50;    // i/p = 0101
p2a=1'b1; p2b=1'b1; p2c=1'b1; p2d=1'b0; #50;   // i/p = 1110
p2a=1'b1; p2b=1'b1; p2c=1'b1; p2d=1'b1; #50;   // i/p = 1111
p2a=1'b1; p2b=1'b1; p2c=1'b0; p2d=1'b0;          // i/p = 1100
end
endmodule
```

Functional Simulation:


Msgs
/test_bench_tb3/p1a
/test_bench_tb3/p1b
/test_bench_tb3/p1c
/test_bench_tb3/p1d
/test_bench_tb3/p1e
/test_bench_tb3/p1f
/test_bench_tb3/p2a
/test_bench_tb3/p2b
/test_bench_tb3/p2c
/test_bench_tb3/p2d
/test_bench_tb3/p1y
/test_bench_tb3/p2y
0
0
1
1
0
1
1
1
0
0
St0
St1