Combinational Design

Your friend is rather picky about what pizzas they are willing to eat. From your observations, they will only eat pizzas that conform to the following rules:

1. The pizza should have exactly one vegetable (either mushrooms or peppers).
2. The pizza should not have bacon unless it also has mushrooms.
3. The pizza shouldn't have both sausage and peppers.

Write a boolean expression, L, that evaluates to true for any combination of ingredients that your friend is willing to eat on pizza. *(Use B = bacon, M = mushrooms, P = peppers, S = sausage.)* A blank truth table is provided for your convenience.

$L =$

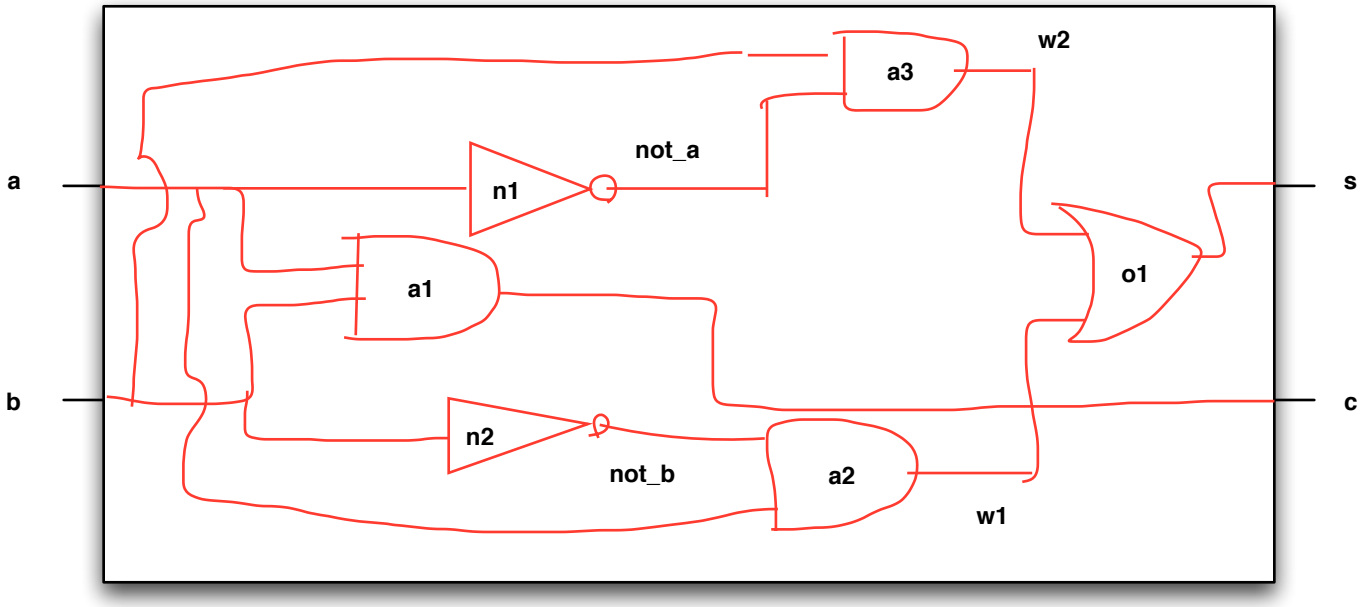
Verilog Modules

```
module sc_block(s, c, a, b);
   output s, c;
   input  a, b;
   wire   w1, w2, not_a, not_b;

   // the "c" output is just the AND of the two inputs
   and a1(c, a, b);

   // the "s" output is 1 only when exactly one of the inputs is 1
   not n1(not_a, a);
   not n2(not_b, b);
   and a2(w1, a, not_b);
   and a3(w2, b, not_a);
   or  o1(s, w1, w2);

endmodule // sc_block
```

Truth Table

a	b	s	c
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Draw a circuit diagram for the Verilog module above:



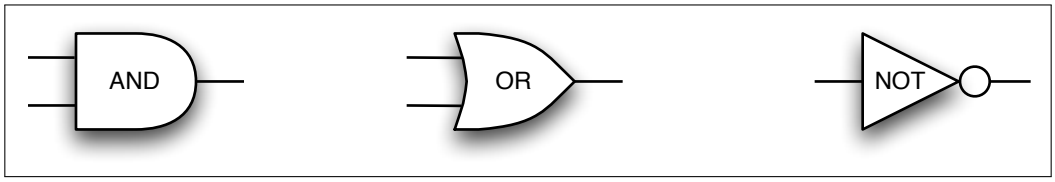
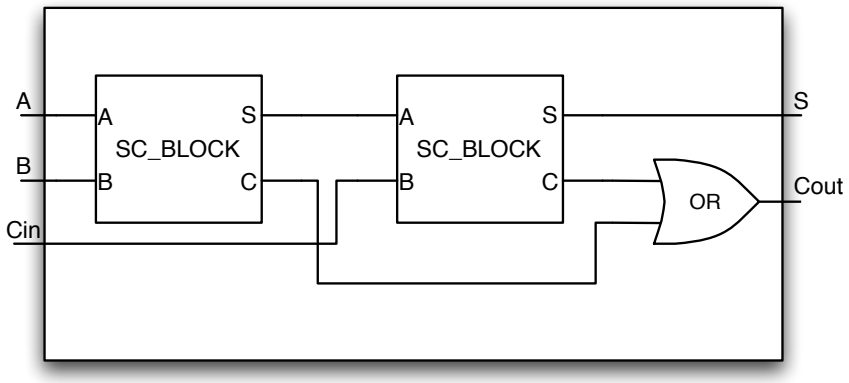


Figure 1. AND, OR, and NOT gates.

Hierarchical design: building modules from modules



A	B	Cin	S	Cout
0	0	0	0	0
0	0	1		
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1		

Figure 2. The sc2_block, which is implemented using sc_block, and its partially completed truth table.

Complete the truth table and write the Verilog for the `sc2_block` module: