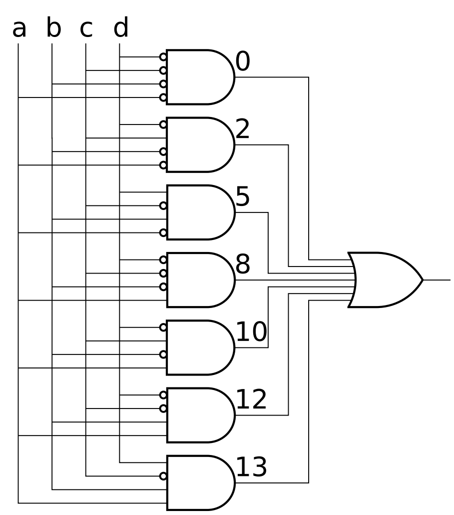
Problem 1: Consider the following logic function in canoncial form:
$\sum_{a,b,c,d} m(0,2,5,8,10,12,13)$.

(*a*) Draw a truth table for this logic function.

```
SOLUTION:
row  a b c d !  f
-------------+----
0    0 0 0 0 !  1
1    0 0 0 1 !  0
2    0 0 1 0 !  1
3    0 0 1 1 !  0
4    0 1 0 0 !  0
5    0 1 0 1 !  1
6    0 1 1 0 !  0
7    0 1 1 1 !  0
8    1 0 0 0 !  1
9    1 0 0 1 !  0
10   1 0 1 0 !  1
11   1 0 1 1 !  0
12   1 1 0 0 !  1
13   1 1 0 1 !  1
14   1 1 1 0 !  0
15   1 1 1 1 !  0
```

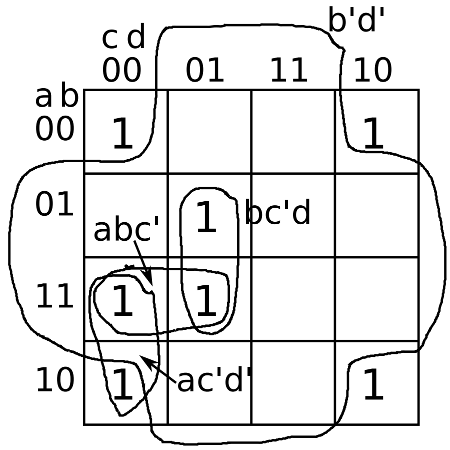
(*b*) Draw a logic circuit for this function. Do not simplify.

Diagram appears below. The and gates are labeled with minterm numbers.



(*c*) Draw a Karnaugh map for the logic function.

Solution appears below, the prime implicants are circled.



(d) List the prime implicants.

As shown in the Karnaugh map above, the prime implicants are: $b'd'$, abc', $bc'd$, and $ac'd'$.

(e) List the essential prime implicants.

The essential prime implicants are $b'd'$ (because of minterms 0, 2, and 10), and $bc'd$ (because of minterm 5).

(f) List all of the minimum cost sum-of-product expressions.

There are only two,

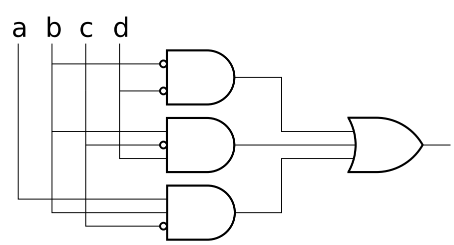
$$b'd' + bc'd + ac'd'$$

and

$$b'd' + bc'd + abc'.$$

(g) Draw a logic diagram for your favorite one.

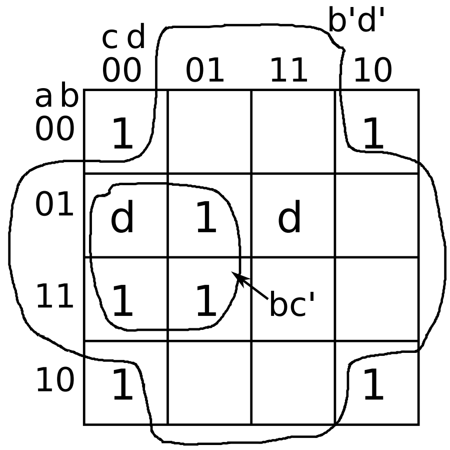
Diagram appears below.



Problem 2: Consider again the logic function from the previous problem, $\sum_{a,b,c,d} m(0, 2, 5, 8, 10, 12, 13)$. This time however suppose the outputs are *don't care* for two sets of inputs, $a = 0$, $b = 1$, $c = 0$, $d = 0$ (corresponding to row (minterm) 4) and $a = 0$, $b = 1$, $c = 1$, $d = 1$ (corresponding to row (minterm) 7).

(a) Draw a Karnaugh map, include the don't cares.

Solution appears below. The d for minterm 4 (the one on the left) is set to 1, the other is set to zero. This enables the minterms in three "real" non-corner minterms to be covered by just one prime implicant, bc', which happens to be an essential prime implicant.

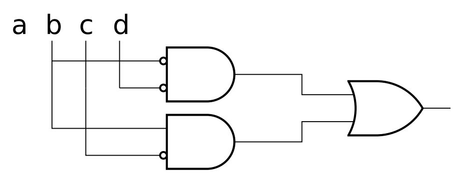


(*b*) Find a minimum-cost sum-of-products expression making the best use of the don't cares.

Based on the diagram above the two essential prime implicants cover the whole expression. So the minimum cost expression is $b'd' + bc'$.

(*c*) Draw a logic diagram corresponding to the minimum-cost expression.

Solution appears below.



Problem 3: The *population* of an n-bit quantity is the number of bits with value 1. For example, the population of 4-bit quantity `0101` is 2, the population of `1101011` is 5.

(a) Show a truth table for a Boolean function with an output that's logic 1 if the population of 2-bit input a_1a_0 is the same as the population of 2-bit input b_1b_0. (The function has four inputs, a_1, a_0, b_1, and b_0.)

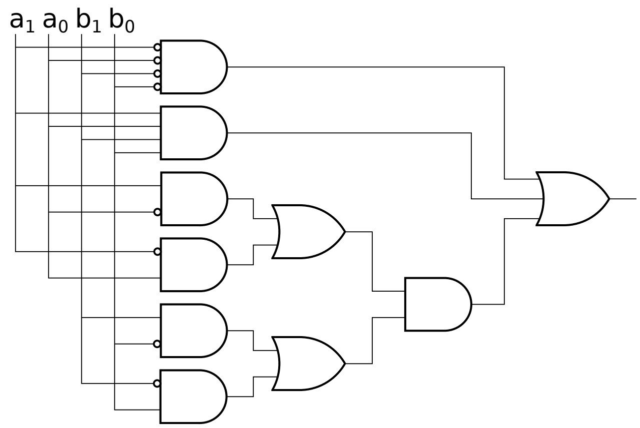
```
SOLUTION:
row  a1 a0 b1 b0
----------------+----
0     0  0  0  0 !  1
1     0  0  0  1 !  0
2     0  0  1  0 !  0
3     0  0  1  1 !  0
4     0  1  0  0 !  0
5     0  1  0  1 !  1
6     0  1  1  0 !  1
7     0  1  1  1 !  0
8     1  0  0  0 !  0
9     1  0  0  1 !  1
10    1  0  1  0 !  1
11    1  0  1  1 !  0
12    1  1  0  0 !  0
13    1  1  0  1 !  0
14    1  1  1  0 !  0
15    1  1  1  1 !  1
```

(b) Derive a Boolean algebraic expression for the same function without using the truth table. Use the following approach: derive an expression that's logic 1 when the population of a_1a_0 is zero. Derive similar expressions for when the population is 1 and when the population is 2. Then pair such expressions for a and b.

The three cases are population 0, 1, and 2. For population 0 both bits of a and b must be `0`, that is true when $a_1'a_0'b_1'b_0'$ is true. For population 2 both bits of a and b must be `1`, that happens when $a_1a_0b_1b_0$ is true. Population 1 can happen two ways, either $a_1 = 1$ and $a_0 = 0$ or $a_1 = 0$ and $a_0 = 1$. The expression for that for a is $a_1a_0' + a_1'a_0$. The population for both a and b will be 1 when both expressions are true: $(a_1a_0' + a_1'a_0)(b_1b_0' + b_1'b_0)$. Combining all of these conditions yields the solution: $a_1'a_0'b_1'b_0' + a_1a_0b_1b_0 + (a_1a_0' + a_1'a_0)(b_1b_0' + b_1'b_0)$.

(c) Draw a logic diagram for either the hand-derived expression (the previous part) or if you couldn't do the previous part, an expression based on the truth table.

The diagram below is for the hand-derived expression:



(d) Try simplifying the Boolean expressions using the exclusive or ($\oplus$) operator ($a \oplus b = ab' + a'b$). If successful, draw a logic diagram based on the simplified expressions.

Note that $a_1a_0' + a_1'a_0 = a_1 \oplus a_0$. Using that we get for the entire expression:
$a_1'a_0'b_1'b_0' + a_1a_0b_1b_0 + (a_1 \oplus a_0)(b_1 \oplus b_0)$.
The logic diagram appears below.