Karnaugh Maps

Bob Harbort ● Bob Brown
Computer Science Department
Southern Polytechnic State University

Simplifying Boolean Expressions

We can use the sum-of-products method to generate any digital logic circuit for which we can write the truth table. While the sum-of-products method will always produce a correct circuit, it usually does not produce the optimal circuit. We want the simplest possible circuit because fewer gates or simpler gates mean reduced cost, improved reliability, and often increased speed. We can simplify a circuit by simplifying the Boolean expression for it, then using the techniques already learned to produce the circuit that is equivalent to the simplified expression. The need to simplify Boolean expressions occurs in programming as well as hardware design, and the techniques discussed here are equally applicable to programming.

One way to simplify a Boolean expression is to apply the laws of Boolean algebra, some of which are summarized in the table on p. 144 of Tanenbaum. We will apply the laws of Boolean algebra to simplify $AB + A\overline{B}$. We choose this expression because it is key to how Karnaugh maps work; other expressions can also be simplified using Boolean algebra.

$AB + A\overline{B}$	*original expression*
$A(B + \overline{B})$	*after applying distributive law*
$A1$	*after applying inverse law*
$1A$	*after applying commutative law*
A	*after applying identity law*

Therefore, $AB + A\overline{B} = A$.

Karnaugh Maps

Karnaugh maps are a graphical way of using the relationship $AB + A\overline{B} = A$ to simplify a Boolean expression and thus simplify the resulting circuit. A Karnaugh map is a completely mechanical method of performing this simplification, and so has an advantage over manipulation of expressions using Boolean algebra. Karnaugh maps are effective for expressions of up to about six variables. For more complex expressions the Quine-McKluskey method, not discussed here, may be appropriate.

The Karnaugh map uses a rectangle divided into rows and columns in such a way that any product term in the expression to be simplified can be represented as the intersection of a row and a column. The rows and columns are labeled with each term in the expression and its complement. The labels must be arranged so that each horizontal or vertical move changes the state of one and only one variable.

To use a Karnaugh map to simplify an expression:

1. Draw a “map” with one box for each possible product term in the expression. The boxes must be arranged so that a one-box movement either horizontal or vertical changes one and only one variable. See Figure 1.
2. For each product term in the expression to be simplified, place a checkmark in the box whose labels are the product's variables and their complements.
3. Draw loops around adjacent pairs of checkmarks. Blocks are "adjacent" horizontally and vertically only, not diagonally. A block may be "in" more than one loop, and a single loop may span multiple rows, multiple columns, or both, so long as the number of checkmarks enclosed is a power of two.
4. For each loop, write an unduplicated list of the terms which appear; *i.e.* no matter how many times A appears, write down only one A.
5. If a term and its complement both appear in the list, *e.g.* both A and $\overline{A}$, delete both from the list.
6. For each list, write the Boolean product of the remaining terms.
7. Write the Boolean sum of the products from Step 5; this is the simplified expression.

Karnaugh Maps for Expressions of Two Variables

Start with the expression $AB + A\overline{B}$. This is an expression of two variables. We draw a rectangle and divide it so that there is a row or column for each variable and its complement.

Next, we place checks in the boxes that represent each of the product terms of the expression. The first product term is AB, so we place a check in the upper left block of the diagram, the conjunction of A and B. The second is $A\overline{B}$, so we place a check in the lower left block. Finally, we draw a loop around adjacent pairs of checks.

The loop contains A, B, A, and $\overline{B}$. We remove one A so that the list is unduplicated. The B and $\overline{B}$ "cancel," leaving only A, which is the expected result: $AB + A\overline{B} = A$.

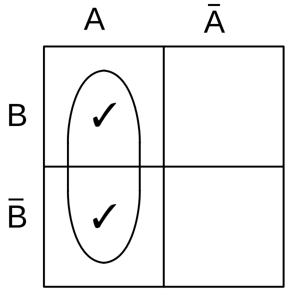


Figure 1. Karnaugh map for $AB + A\overline{B}$.

Let us try a slightly more interesting example: simplify $A\,\overline{B} + \overline{A}\,B + \overline{A}\,\overline{B}$. There are two variables, so the rectangle is the same as in the first example. We perform the following steps:

- Place a check in the $A\overline{B}$ area.
- Place a check in the $\overline{A}B$ area.
- Place a check in the $\overline{A}\,\overline{B}$ area.
- Draw loops around pairs of adjacent checks.

The Karnaugh map appears in Figure 2. Because there are two loops, there will be two terms in the simplified expression. The vertical loop contains $\overline{A}$, B, $\overline{A}$, and $\overline{B}$. We remove one $\overline{A}$ to make an unduplicated list. The B and $\overline{B}$ cancel, leaving the remaining $\overline{A}$. From the horizontal loop we remove the duplicate $\overline{B}$, then remove A and $\overline{A}$ leaving only $\overline{B}$ in the second term. We write the Boolean sum of these, and the result is $\overline{A} + \overline{B}$, so:

$$A\,\overline{B} + \overline{A}\,B + \overline{A}\,\overline{B} = \overline{A} + \overline{B}$$

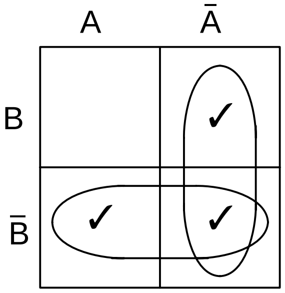


Figure 2. Karnaugh map for $A\,\overline{B} + \overline{A}\,B + \overline{A}\,\overline{B}$

Expressions of Three Variables

Recall that an essential characteristic of a Karnaugh map is that moving one position horizontally or vertically changes one and only one variable to its complement. For expression of three variables, the basic Karnaugh diagram is shown in Figure 3.

As with the diagram for two variables, adjacent squares differ by precisely one literal. The left and right edges are considered to be adjacent, as though the map were wrapped around to form a cylinder.

Now we'll work through a complete example, starting with deriving a circuit from a truth table using the sum of products method, simplifying the sum of products expression, and drawing the new, simpler circuit.

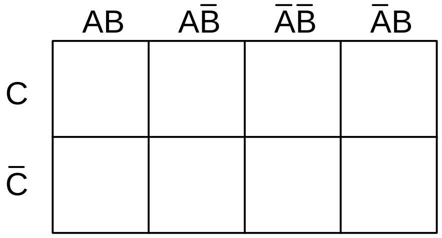
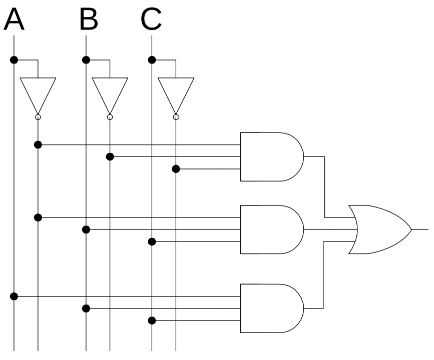


Figure 3. Form of a Karnaugh map for expressions of three variables.

Truth Table | **Product Terms** | **Sum-of-Products Expression** | **Digital Logic Circuit**

A	B	C	X	
0	0	0	1	$\overline{A}\,\overline{B}\,\overline{C}$
0	0	1	0	
0	1	0	0	
0	1	1	1	$\overline{A}\,B\,C$
1	0	0	0	
1	0	1	0	
1	1	0	0	
1	1	1	1	$A\,B\,C$

$$\overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,B\,C + A\,B\,C$$



a) **b)** **c)**

Figure 4. a) A truth table with product terms, b) the resulting sum-of-products expression, and c) the equivalent digital logic circuit.

The truth table in Figure 4a generates an expression with three product terms, as shown in Figure 4b. A measure of the complexity of a digital logic circuit is the number of gate inputs. The circuit in Figure 4c has 15 gate inputs. The Karnaugh map for the expression in Figure 4b is shown at the right. In this Karnaugh map, the large loop surrounds $A\,B\,C$ and $\overline{A}\,B\,C$; note that it "wraps around" from the left edge of the map to the right edge. The A and $\overline{A}$ cancel, so these two terms simplify to BC.

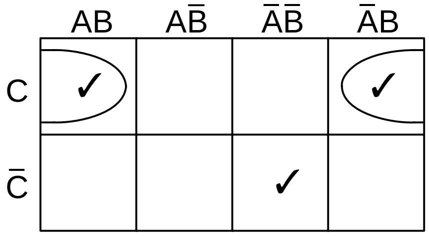


Figure 5. Simplifying $\overline{A}\,\overline{B}\,\overline{C}+\overline{A}\,B\,C+A\,B\,C$ with a Karnaugh map.

$\overline{A}\,\overline{B}\,\overline{C}$ is in a cell all by itself, and so contributes all three of its terms to the final expression. The simplified expression is $B\,C+\overline{A}\,\overline{B}\,\overline{C}$ and the simplified circuit is shown in Figure 6. In the simplified circuit, one three-input AND gate was removed, a remaining AND gate was changed to two inputs, and the OR gate was changed to two inputs, resulting in a circuit with ten gate inputs.

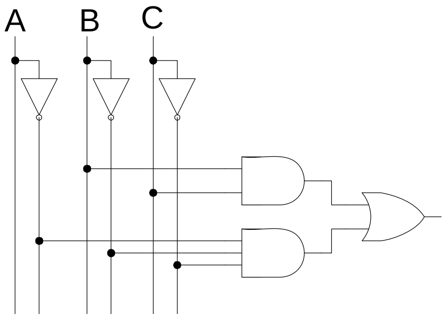


Figure 6. Simplified circuit for the truth table of Figure 4a.

Let's consider another example. The truth table in Figure 7a generates a sum-of-products expression with five product terms of three variables each. The sum-of-products expression is shown in Figure 7b. The digital logic circuit for this expression, shown in Figure 7c, has nine gates and 23 gate inputs. The Karnaugh map for this expression is shown in Figure 8.

Truth Table | **Product Terms**

A	B	C	X	
0	0	0	0	
0	0	1	0	
0	1	0	1	$\overline{A}\,B\,\overline{C}$
0	1	1	1	$\overline{A}\,B\,C$
1	0	0	0	
1	0	1	1	$A\,\overline{B}\,C$
1	1	0	1	$A\,B\,\overline{C}$
1	1	1	1	$A\,B\,C$

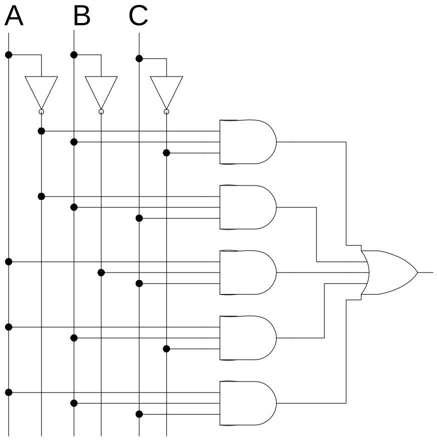
a)

Sum-of-Products Expression

$$\overline{A}\,B\,\overline{C}+\overline{A}\,B\,C+A\,\overline{B}\,C$$
$$+A\,B\,\overline{C}+A\,B\,C$$

b)

Digital Logic Circuit



c)

Figure 7. a) A truth table with product terms, b) the resulting sum-of-products expression, and c) the equivalent digital logic circuit.

After removing duplicates, the large loop contains A and $\overline{A}$ *and also* C and $\overline{C}$; these cancel. All that's left after removing the two complement pairs is B. The small loop contains B and $\overline{B}$, which are removed, so it yields AC. We have simplified the expression in Figure 7b to B+AC.

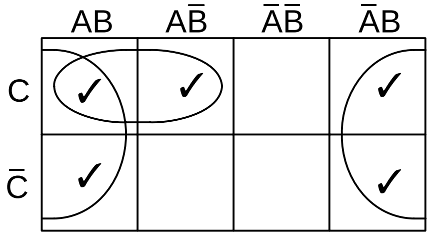


Figure 8. Karnaugh map for the expression of figure 7.

The circuit for B+AC is shown in Figure 9. We have simplified the circuit from nine gates and 23 inputs to two two-input gates. This is a substantial reduction in complexity.

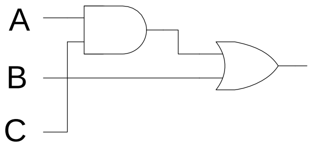


Figure 9. Simplified circuit equivalent to Figure 7c.

Getting the Best Results

For maximum simplification, you want to make the loops in a Karnaugh map as big as possible. If you have a choice of making one big loop or two small ones, choose the big loop. The restriction is that the loop must be rectangular and enclose a number of checkmarks that is a power of two.

When a map is more than two rows deep, *i.e.* when it represents more than three variables, the top and bottom edges can be considered to be adjacent in the same way that the right and left edges are adjacent in the two-by-four maps above.

If all checkmarks in a loop are enclosed within other loops as well, that loop can be ignored because all its terms are accounted for. In the Karnaugh map in Figure 10, the vertical loop is redundant and can be ignored.

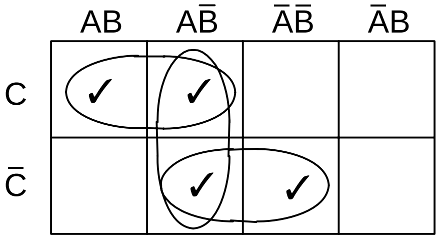


Figure 10. Karnaugh map showing a redundant loop.

Sometimes not all possible combinations of bits represented in a truth table can occur. For example, if four bits are used to encode a decimal digit, combinations greater than 1001 cannot occur. In that case, you can place a “D” (for “don’t care”) in the result column of the truth table. These D’s may be treated as *either* ones or zeroes, and you can place check marks on the map in the D’s positions if doing so allows you to make larger loops.

One General Form for a Karnaugh Map

There are several possible forms for a Karnaugh map, including some three-dimensional versions. All that is required is that a movement of one position changes the value of one and only one variable. We have shown a form for maps of two and three variables. Below are maps for four and five variables.

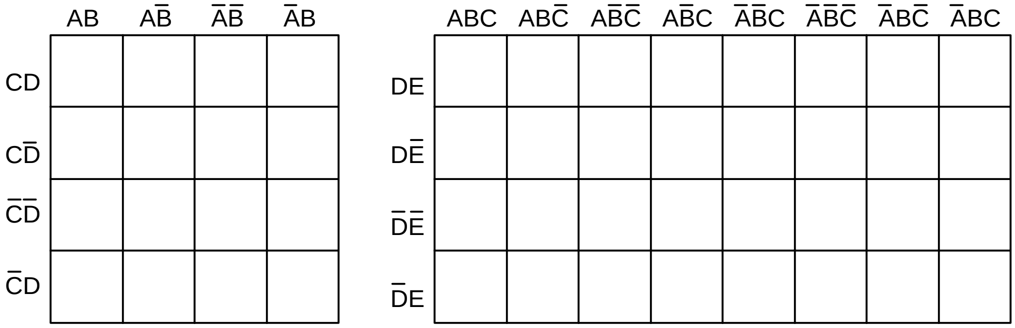

AB
$A\overline{B}$
$\overline{A}\overline{B}$
$\overline{A}B$
CD
$C\overline{D}$
$\overline{C}\overline{D}$
$\overline{C}D$
ABC
$AB\overline{C}$
$A\overline{B}\overline{C}$
$A\overline{B}C$
$\overline{A}\overline{B}C$
$\overline{A}\overline{B}\overline{C}$
$\overline{A}B\overline{C}$
$\overline{A}BC$
DE
$D\overline{E}$
$\overline{D}\overline{E}$
$\overline{D}E$

Figure 11. One form of the Karnaugh map for expressions of four and five variables.

A Notation Reminder

The *Boolean product* of two variables is written as AB, A∧B or A·B; the variables are combined using the AND function.

The *Boolean sum* of two variables is written as A+B or A∨B; the variables are combined using the OR function.

The *complement* of a Boolean variable is written as $\overline{A}$; it is evaluated using the NOT function.

The product, sum, and complement can be applied to expressions as well as single variables. Parentheses can be used to show precedence when needed.

Bibliography

Mendelson, Elliott, *Schaum's Outline of Theory and Problems of Boolean Algebra,* McGraw-Hill, 1970.

Stallings, William, *Computer Organization and Architecture: Designing for performance,* Prentice-Hall, 1996.

Tanenbaum, Andrew S., *Structured Computer Organization,* Prentice-Hall, 2006.

Exercises

1. Verify that the circuit in Figure 9 is equivalent to the circuit in Figure 7c by deriving the truth table for the circuit in Figure 9 and comparing it to Figure 7a.

2. Sketch a Karnaugh map for expressions of six variables. *Hint:* See Figure 11.

3. The truth table for binary addition has three inputs: the addend, the augend, and the carry in. The output has two parts, the sum and the carry out. Write the truth table for the sum part of binary addition. Use a Karnaugh map to simplify the expression represented by this truth table. *Hint:* This is a sneaky question, but you will learn a lot about the power of Karnaugh maps.

4. Write the truth table for the carry part of binary addition. Use a Karnaugh map to simplify the sum-of-products expression which this truth table produces.

5. Use a Karnaugh map to simplify $A\,B\,C + A\,\overline{B}\,\overline{C} + \overline{A}\,\overline{B}\,\overline{C}$.