Sum-of-Products form

A Boolean expression is in **sum-of-products** form (**SOP**) when it's written as the sum of one or more products. In SOP form, an overbar cannot extend over more than one variable.

Examples of expressions in SOP form:

$$\overline{A}\,\overline{B}\,\overline{C} + A\,B \qquad A\,B\,\overline{C} + \overline{C}\,\overline{D} \qquad AB + AC + \overline{D}$$

The book also discusses **product-of-sums** form (**POS**), in which two or more sum terms are multiplied, as in the following examples:

$$(A + B)(\overline{A} + C) \qquad (A + B + C)(\overline{B} + D) \qquad (A + B)(\overline{B} + C)D$$

SOP form is far more useful than POS form.

SOP and POS forms

Many Boolean expressions are in neither SOP form nor POS form.

Examples: $\overline{AB + C}$ and $AB + C(\overline{A}D + B\overline{D})$

But **every** expression can be converted to SOP form by applying the distributive law and DeMorgan's theorems.

REVIEW: Writing the SOP Expression for any Truth Table

Given the truth table for a circuit, it's easy to write an SOP-form expression for that circuit.

Step 1. For each of the truth table's rows with a 1 in the output column, list the corresponding product term of the input variables.

Step 2. Add all of the product terms from Step 1.

See example on next slide…

Example: Writing the SOP Expression for a Truth Table

A	*B*	*C*	*X*
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

Writing the Truth Table for any SOP Expression

Given an SOP expression , it's easy to write the truth table.

Step 1. Based on the number of input variables, build the truth table's input columns.

Step 2. For each product term in the SOP expression, place a 1 in the truth table's output column for all rows that make the product term a 1.

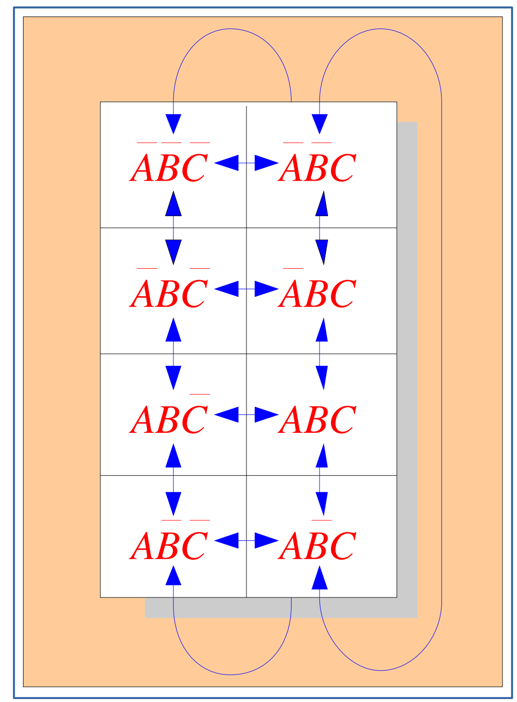
Step 3. After completing Step 2 for all product terms in the SOP expression, place a 0 in the output column for all other rows.

Karnaugh maps

The Karnaugh map (K-map) is a tool for simplifying expressions with 3 or 4 variables. For 3 variables, 8 cells are required (2^3).

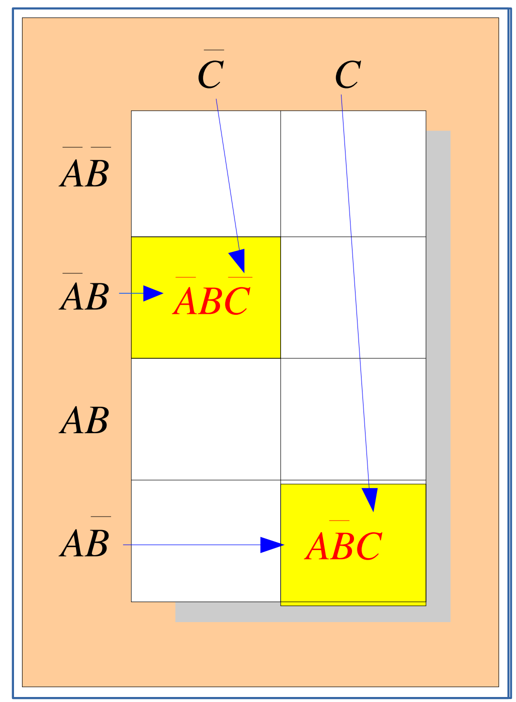
The map shown is for three variables labeled *A*, *B*, and *C*. Each cell represents one possible product term.

Each cell differs from an adjacent cell by only one variable.



Karnaugh maps

Cells in a K-map must be labeled in the order shown.



Example Read the terms for the yellow cells.

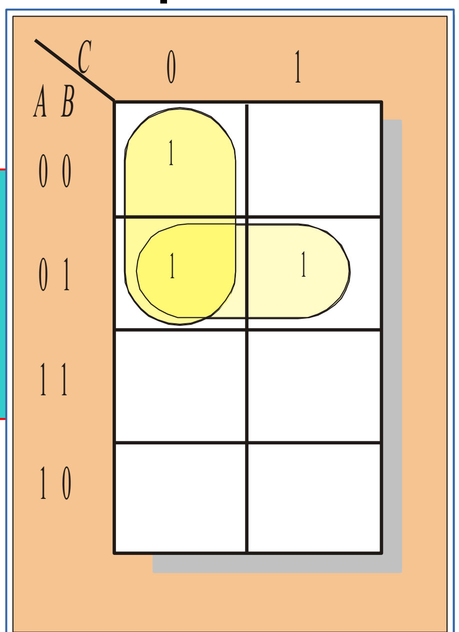
Solution

The cells are $\overline{A}B\overline{C}$ and $A\overline{B}C$.

Karnaugh maps

K-maps can simplify combinational logic by grouping cells and eliminating variables that change.

Example Group the 1's on the map and read the minimum logic.



Solution

1. Group the 1's into two overlapping groups as indicated.
1. Read each group by eliminating any variable that changes across a boundary.
1. The vertical group is read $\overline{A}\overline{C}$.
1. The horizontal group is read $\overline{A}B$.

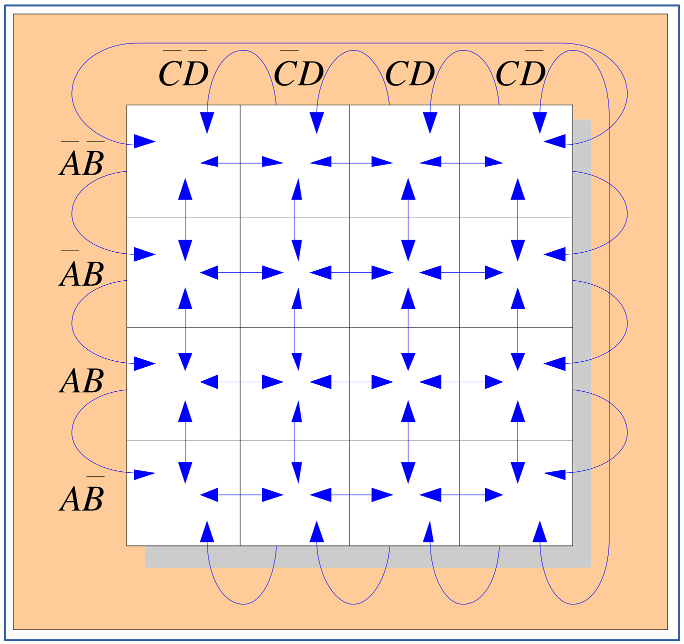
$$X = \overline{A}\,\overline{C} + \overline{A}B$$

Karnaugh Map Procedure

1. If you're starting with a Boolean expression that is not in SOP form, convert it to SOP form.
2. Set up the K-map, labeling its rows and columns.
3. Place 1s in the appropriate squares.
4. Group adjacent 1s in groups of 8, 4, 2, or 1. You want to **maximize the size of the groups** and **minimize the number of groups**. Follow this order:
    a. Circle any octet.
    b. Circle any quad that contains one or more 1s that haven't already been circled, using the minimum number of circles.
    c. Circle any pair that contains one or more 1s that haven't already been circled, using the minimum number of circles.
    d. Circle any isolated 1s that haven't already been circled.
5. Read off the term for each group by including only those complemented or uncomplemented variables that do not change throughout the group.
6. Form the OR sum of the terms generated in Step 5.

Karnaugh maps

A 4-variable map has an adjacent cell on each of its four boundaries as shown.



Each cell is different only by one variable from an adjacent cell.

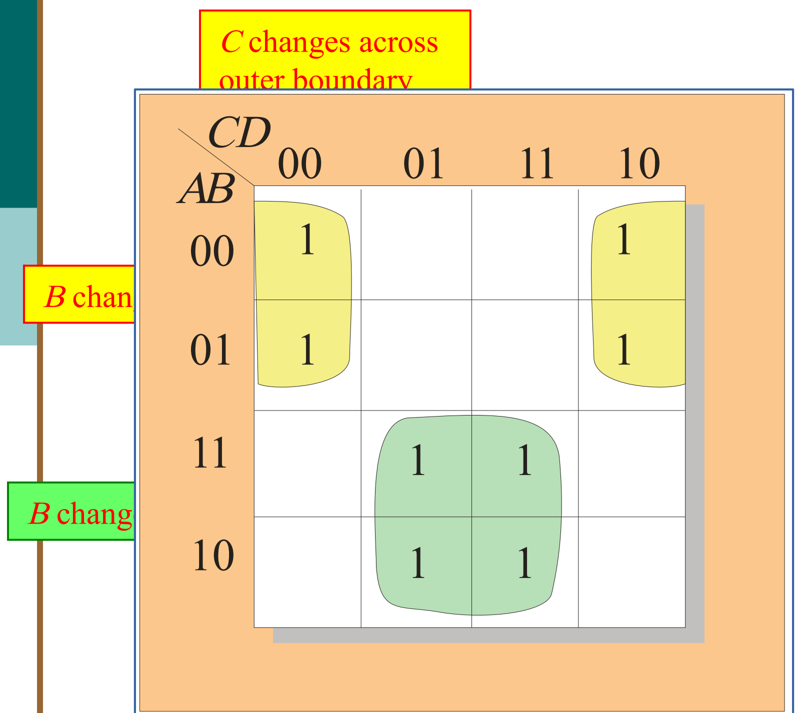
Grouping follows the rules given in the text.

The following slide shows an example of reading a four variable map using binary numbers for the variables…

Karnaugh maps

Example

Group the 1's on the map and read the minimum logic.



X

Solution

1. Group the 1's into two separate groups as indicated.
1. Read each group by eliminating any variable that changes across a boundary.
1. The upper (yellow) group is read as $\overline{A}\,\overline{D}$.
1. The lower (green) group is read as AD.

$$X = \overline{A}\,\overline{D} + AD$$