Computer Organization

Karnaugh Map

Karnaugh Map

The Karnaugh map, also known as the K-map, is a method to simplify boolean algebra expressions.

The Karnaugh map reduces the need for extensive calculations by taking advantage of humans' pattern-recognition capability.

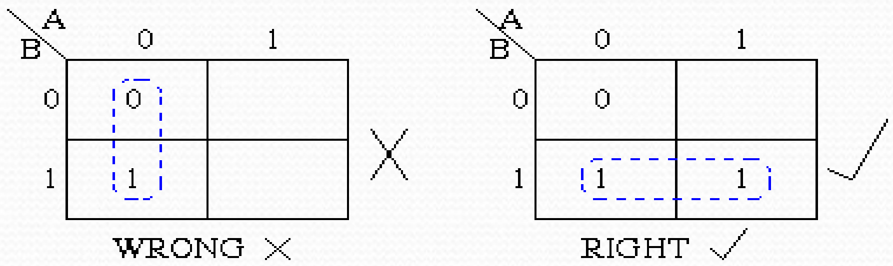
The required boolean results are transferred from a truth table onto a two-dimensional grid where the cells are ordered in Gray code, and each cell position represents one combination of input conditions, while each cell value represents the corresponding output value. Optimal groups of 1s or 0s are identified.

These terms can be used to write a minimal boolean expression representing the required logic.

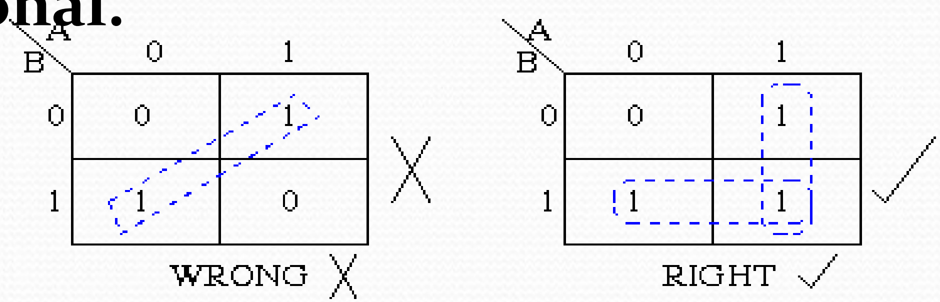
Karnaugh Maps - Rules of Simplification

The Karnaugh map uses the following rules for the simplification of expressions by grouping together adjacent cells containing *ones*.

1.Groups may not include any cell containing a zero.



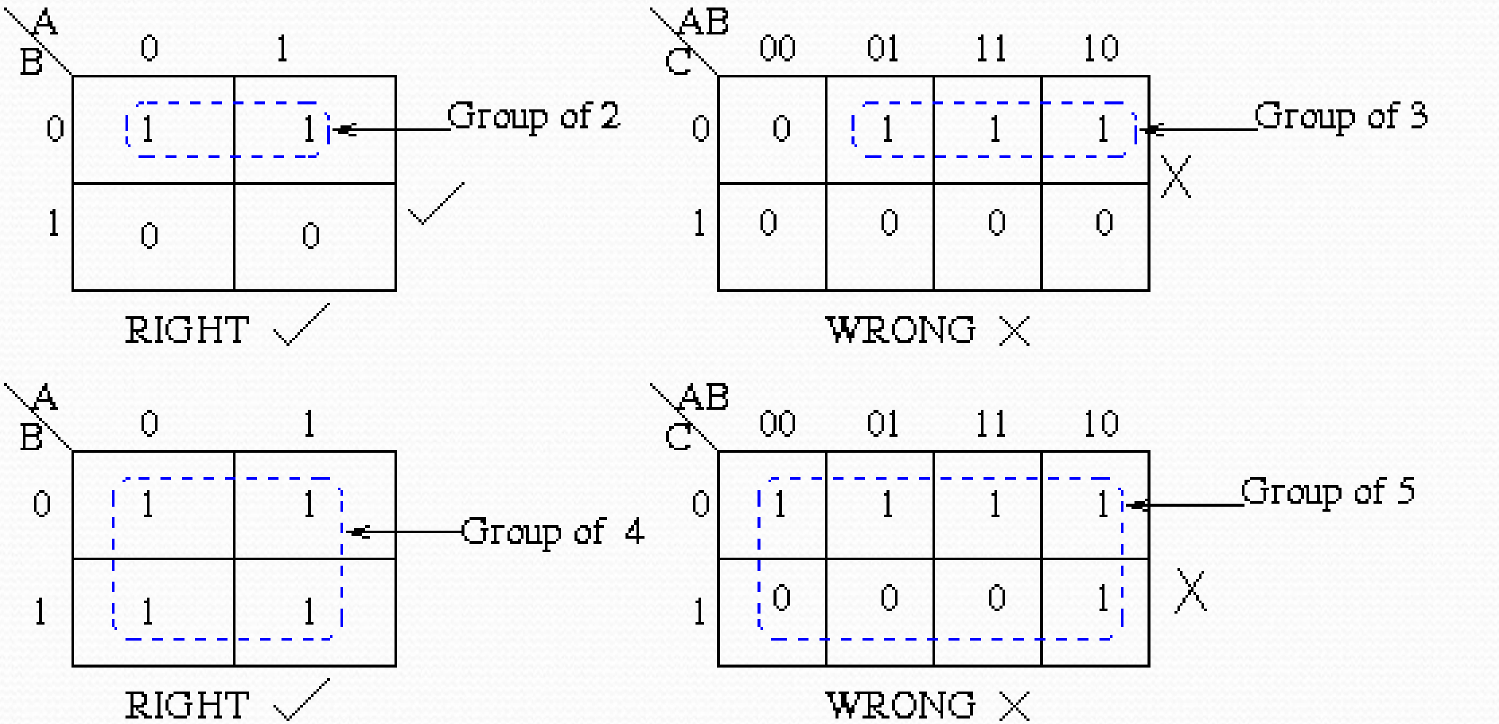
2.Groups may be horizontal or vertical, but not diagonal.



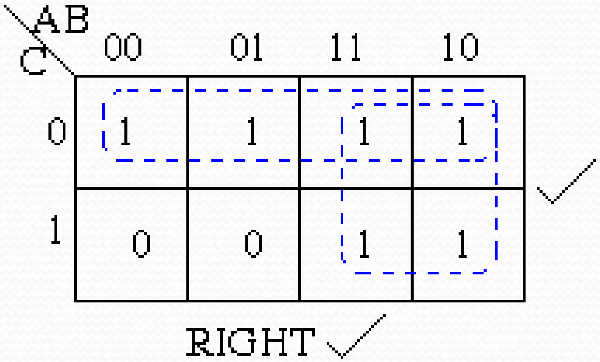
3. Groups must contain 1, 2, 4, 8, or in general 2^n cells.

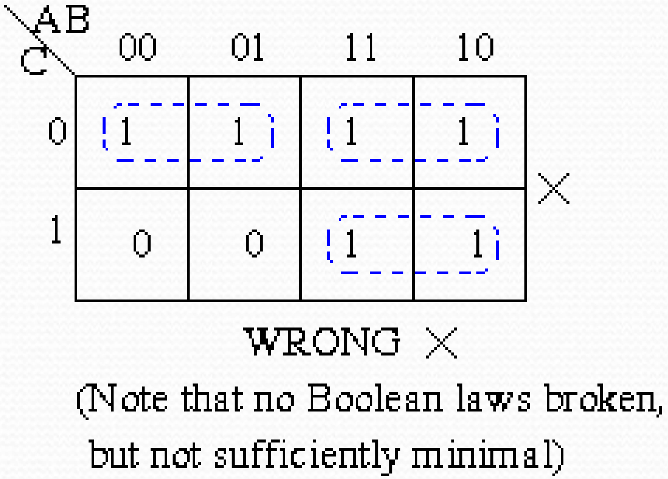
That is if n = 1, a group will contain two 1's since $2^1 = 2$.

If n = 2, a group will contain four 1's since $2^2 = 4$.

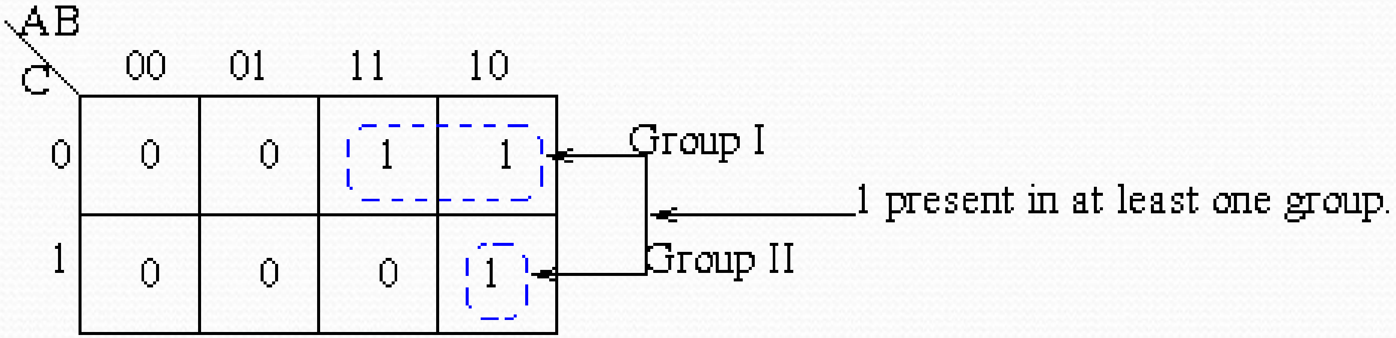


4. Each group should be as large as possible.

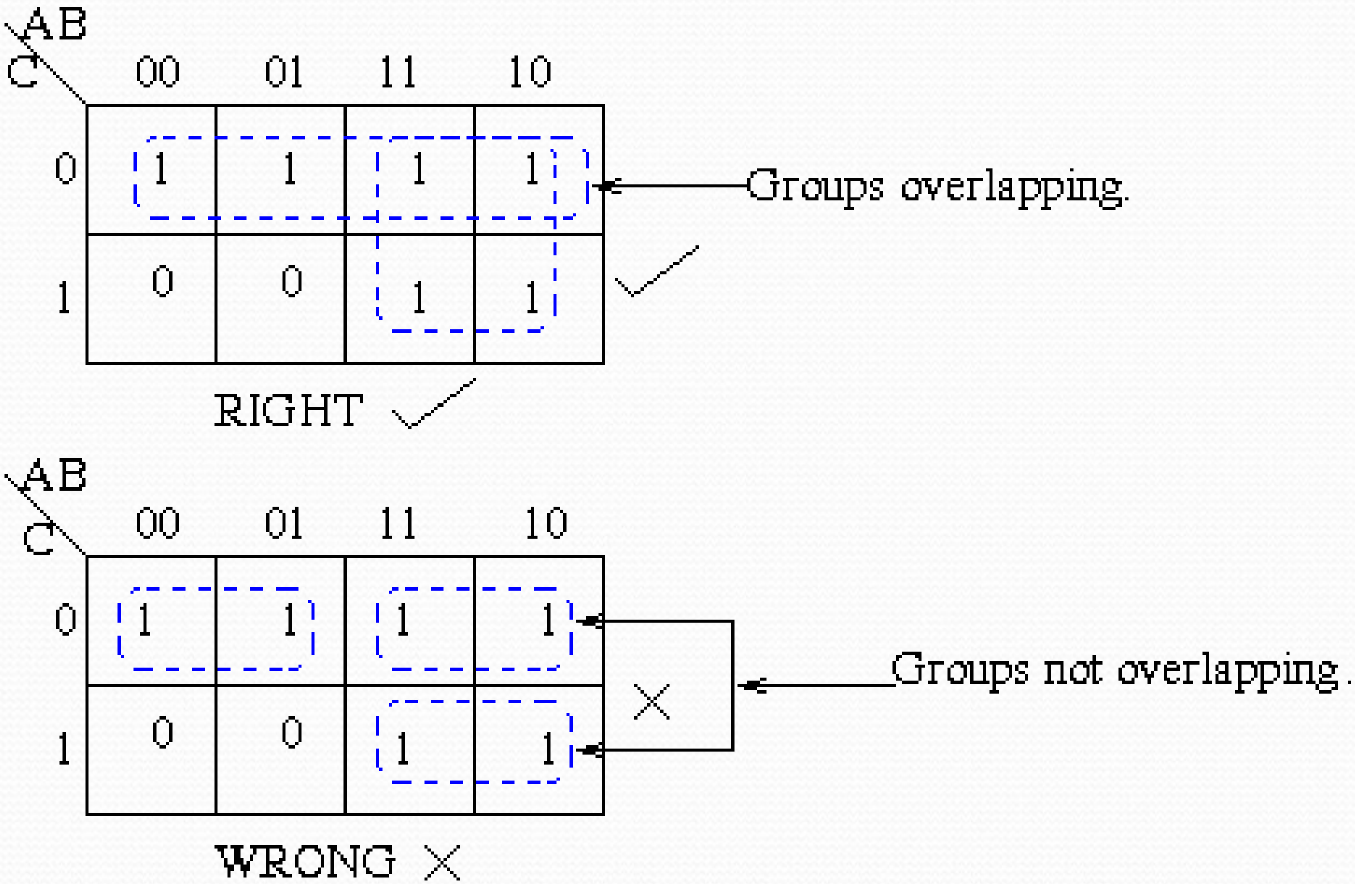




5.Each cell containing a one must be in at least one group.

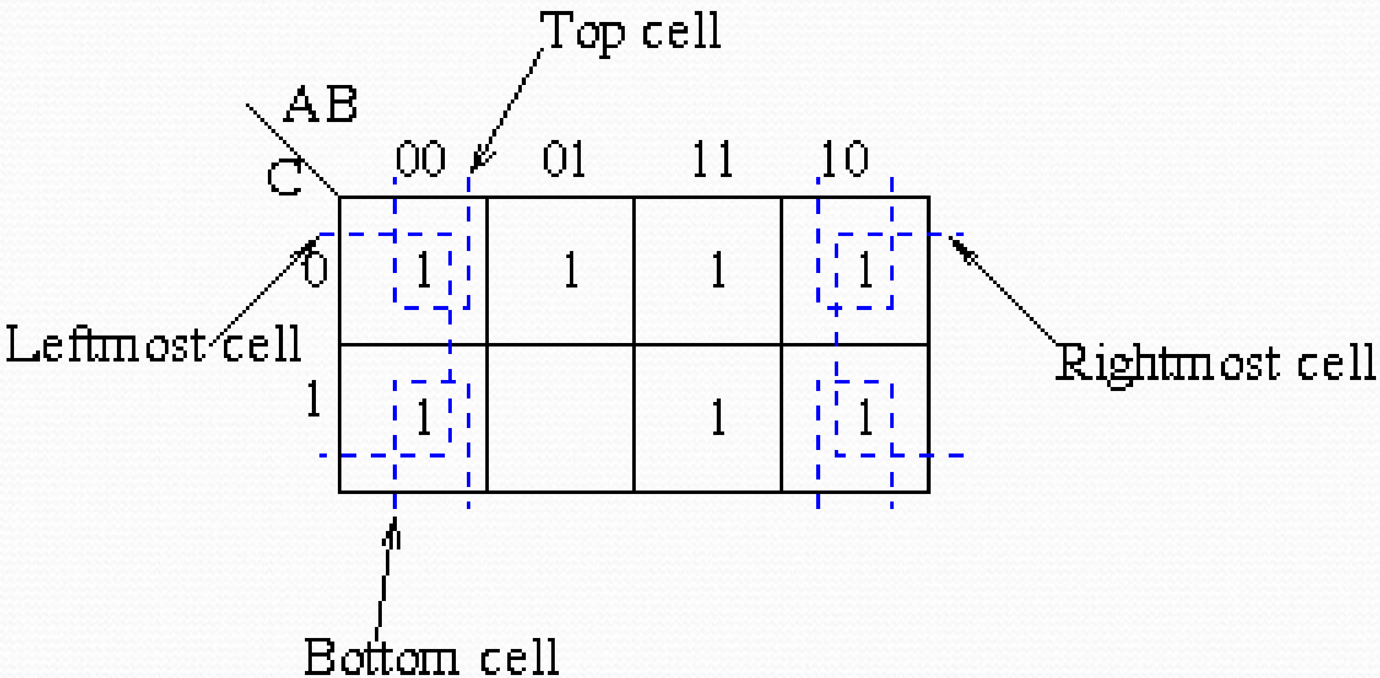


6.Groups may overlap.

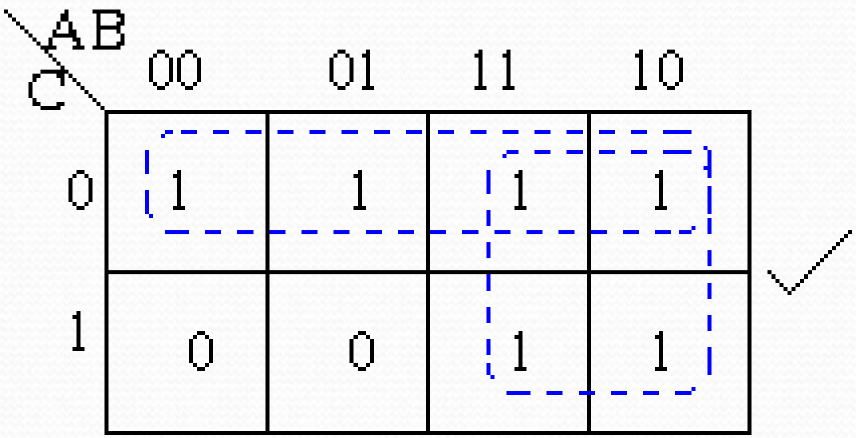


7.Groups may wrap around the table.

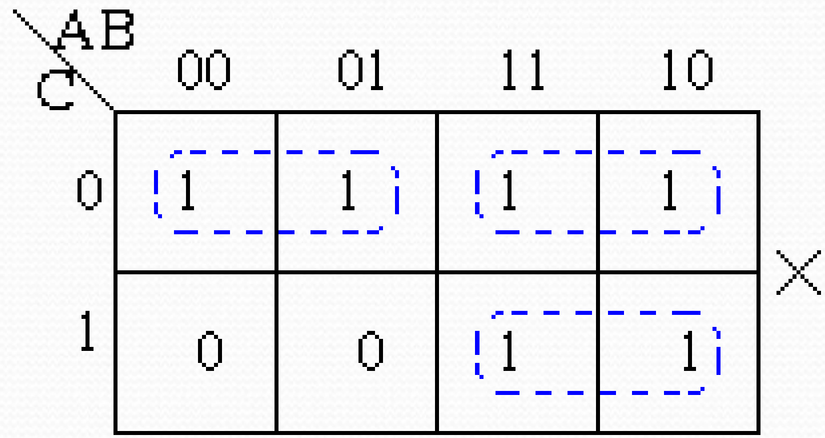
The leftmost cell in a row may be grouped with the rightmost cell and the top cell in a column may be grouped with the bottom cell.



8. There should be as few groups as possible, as long as this does not contradict any of the previous rules.



RIGHT ✓



WRONG ×

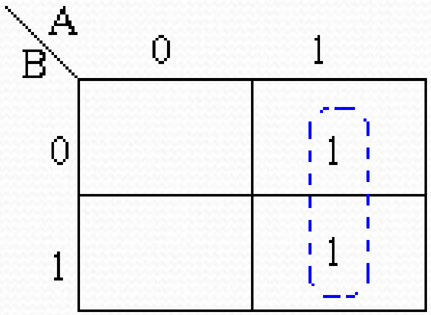
Summmary:

- No zeros allowed.
- No diagonals.
- Only power of 2 number of cells in each group.
- Groups should be as large as possible.
- Every one must be in at least one group.
- Overlapping allowed.
- Wrap around allowed.
- Fewest number of groups possible.

Examples

Example 1:

Consider the following map. The function plotted is: Z = f(A,B) = A + AB.



Note that values of the input variables form the rows and columns.
That is the logic values of the variables A and B (with one denoting true form and zero denoting false form) form the head of the rows and columns respectively.

The map displayed is a one dimensional type which can be used to simplify an expression in two variables.

There is a two-dimensional map that can be used for up to four variables, and a three-dimensional map for up to six variables.

Referring to the map above, the two adjacent 1's are grouped together.

Through inspection it can be seen that variable B has its true and false form within the group.

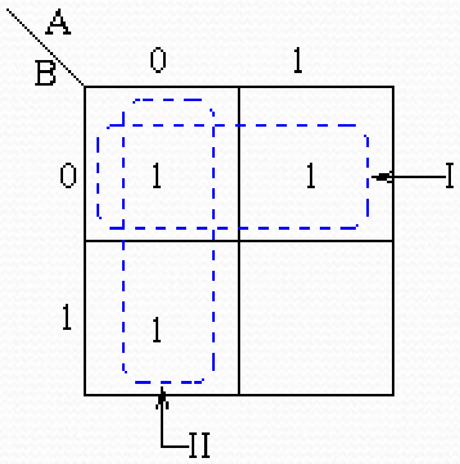
This eliminates variable B leaving only variable A which only has its true form.

The minimized answer therefore is Z = A.

Example 2

Consider the expression Z = f(A,B) = + A + B plotted on the Karnaugh map:

Pairs of 1's are grouped as shown and the simplified answer is obtained by using the following steps:



Note that two groups can be formed for the example given above, the largest rectangular clusters that can be made consist of two 1s.

Notice that a 1 can belong to more than one group The first group labeled I, consists of two 1s which correspond to A = 0, B = 0 and A = 1, B = 0.

In another way, all squares in this example that correspond to the area of the map where B = 0 contains 1s, independent of the value of A. So when B = 0 the output is 1.

The expression of the output will contain the term $\overline{B}$

For group labelled II corresponds to the area of the map where A = 0. The group can therefore be defined as . This implies that when A = 0 the output is 1. The output is therefore 1 whenever B = 0 and A = 0

Hence the simplified answer is Z = $\overline{A} + \overline{B}$

Following are two different notations describing the same function in unsimplified Boolean algebra, using the Boolean variables A,B,C,D and their inverses.

It is used to minimize number of logic gates required.

Example 3: Take the Boolean or binary function described by the truth table.

	A	B	C	D	f(A, B, C, D)
0	0	0	0	0	0
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	0
6	0	1	1	0	1
7	0	1	1	1	0
8	1	0	0	0	1
9	1	0	0	1	1
10	1	0	1	0	1
11	1	0	1	1	1
12	1	1	0	0	1
13	1	1	0	1	1
14	1	1	1	0	1
15	1	1	1	1	0

$f(A,B,C,D) = \sum m(6,8,9,10,11,12,13,14)$ The values inside

$\sum$ are the minterms to map (i.e. rows which have output 1 in the truth table).

$$f(A,B,C,D) = (\overline{A}BC\overline{D})+(A\overline{B}\,\overline{C}\,\overline{D})+(A\overline{B}\,\overline{C}D)+(A\overline{B}C\overline{D})+(A\overline{B}CD)+(AB\overline{C}\,\overline{D})+(AB\overline{C}D)+(ABC\overline{D})$$

In this case, the four input variables can be combined in 16 different ways, so the truth table has 16 rows, and the Karnaugh map has 16 positions.

The Karnaugh map is therefore arranged in a 4 × 4 grid.The row and column values (shown across the top, and down the left side of the Karnaugh map) are ordered in Gray code rather than binary numerical order.

Gray code ensures that only one variable changes between each pair of adjacent cells. Each cell of the completed Karnaugh map contains a binary digit representing the function's output for that combination of inputs

After the Karnaugh map has been constructed it is used to find one of the simplest possible forms—a canonical form—for the information in the truth table.

Adjacent 1s in the Karnaugh map represent opportunities to simplify the expression. The minterms ('minimal terms') for the final expression are found by encircling groups of 1s in the map.

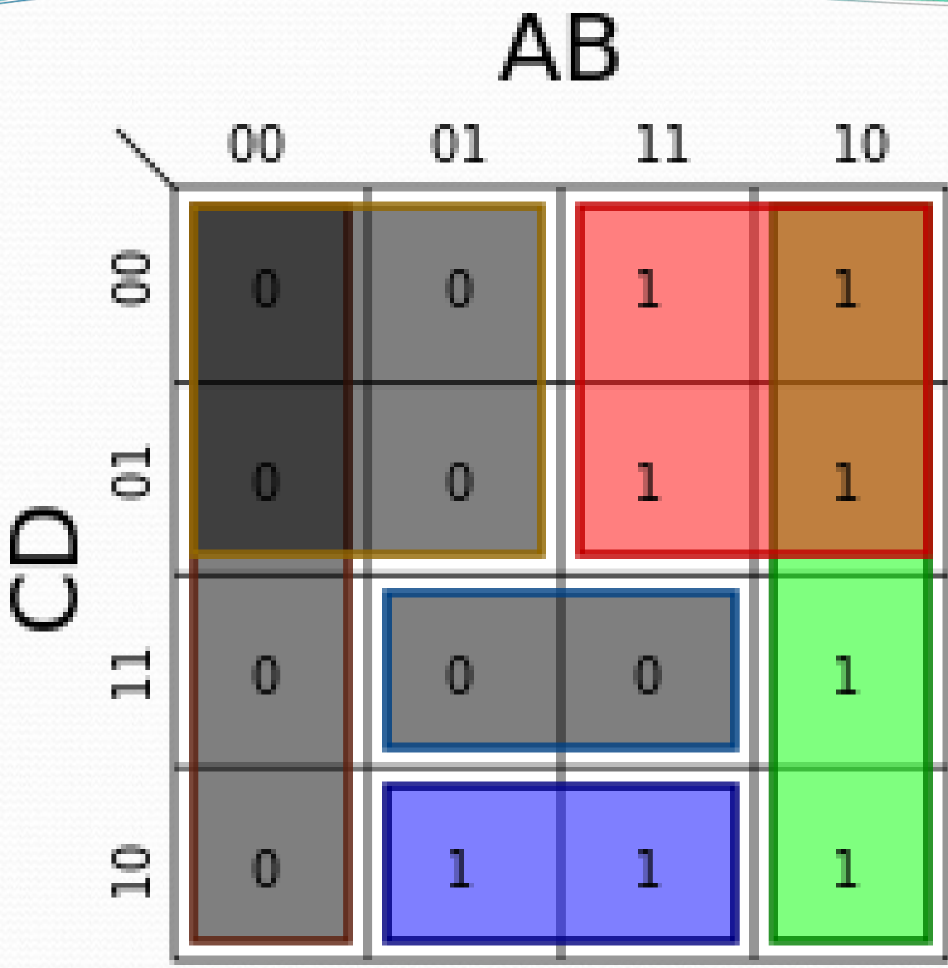
Minterm groups must be rectangular and must have an area that is a power of two (i.e. 1, 2, 4, 8…). Minterm rectangles should be as large as possible without containing any 0s. Groups may overlap in order to make each one larger.

The grid is toroidally connected, which means that rectangular groups can wrap across the edges. Cells on the extreme right are actually 'adjacent' to those on the far left. similarly, so are those at the very top and those at the bottom.

Therefore can be a valid term—it includes cells 12 and 8 at the top, and wraps to the bottom to include cells 10 and 14—as is , which includes the four corners.

CD \ AB	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

ABCD	ABCD
0000 - 0	1000 - 8
0001 - 1	1001 - 9
0010 - 2	1010 - 10
0011 - 3	1011 - 11
0100 - 4	1100 - 12
0101 - 5	1101 - 13
0110 - 6	1110 - 14
0111 - 7	1111 - 15



f(A,B,C,D) = E(6,8,9,10,11,12,13,14)

F=AC'+AB'+BCD'

F=(A+B)(A+C)(B'+C'+D')

- Once the Karnaugh map has been constructed and the adjacent 1s linked by rectangular and square boxes, the algebraic minterms can be found by examining which variables stay the same within each box.
- For the red grouping:
- The variable *A* is the same and is equal to 1 throughout the box, therefore it should be included in the algebraic representation of the red minterm.
- Variable *B* does not maintain the same state (it shifts from 1 to 0), and should therefore be excluded.
- *C* does not change. It is always 0 so its complement, NOT-C, should be included thus, .
- *D* changes, so it is excluded as well.
- Thus the first minterm in the Boolean sum-of-products expression is .
- For the green grouping, *A* and *B* maintain the same state, while *C* and *D* change. *B* is 0 and has to be negated before it can be included. Thus the second term is .
- In the same way, the blue grouping gives the term .
- The solutions of each grouping are combined .

- Thus the Karnaugh map has guided a simplification of

$$f(A,B,C,D) = (\overline{A}BC\overline{D}) + (A\overline{B}\,\overline{C}\,\overline{D}) + (A\overline{B}\,\overline{C}D) + (A\overline{B}C\overline{D}) + (A\overline{B}CD) + (AB\overline{C}\,\overline{D}) + (AB\overline{C}D) + (ABC\overline{D})$$

to

$$f(A,B,C,D) = A\overline{C} + A\overline{B} + BC\overline{D}$$

Inverse

The inverse of a function is solved in the same way by grouping the os instead.
The three terms to cover the inverse are all shown with grey boxes with different colored borders. This yields the inverse:

$$\overline{F} = \overline{A}\,\overline{B} + \overline{A}\,\overline{C} + BCD$$

Through the use of De Morgan law the product of sums can be determined:

$$\overline{\overline{F}} = \overline{\overline{A}\,\overline{B} + \overline{A}\,\overline{C} + BCD}$$

$$F = (A+B)(A+C)\left(\overline{B} + \overline{C} + \overline{D}\right)$$

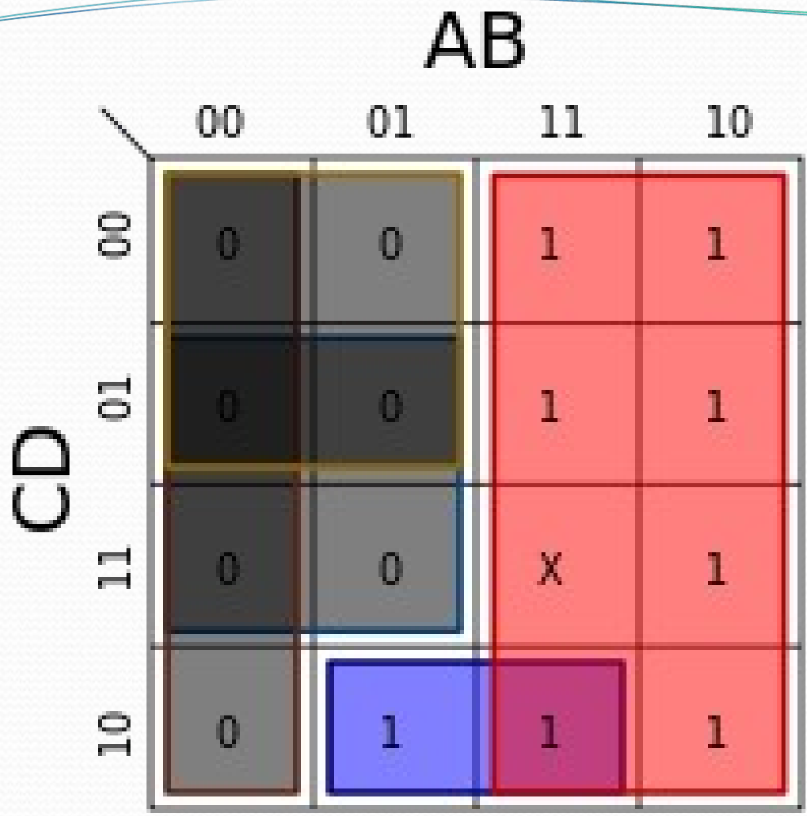
Don't cares are usually indicated on the map with a dash or X.

Karnaugh maps also allow easy minimizations of functions whose truth tables include “don’t care” conditions.

A "don't care" condition is a combination of inputs for which the designer doesn't care what the output is.

Therefore "don't care" conditions can either be included in or excluded from any circled group, whichever makes it larger. They are usually indicated on the map with a dash or X.

The example on the right is the same as the example above but with the value of *F* for *ABCD* = 1111 replaced by a "don't care". This allows the red term to expand all the way down and, thus, removes the green term completely.



f(A,B,C,D) = E(6,8,9,10,11,12,13,14)

F=A+BCD'

F=(A+B)(A+C)(A+D')

This yields the new minimum equation:

$$F = A + BC\overline{D}.$$

- Note that the first term is just A not AC . In this case, the don't care has dropped a term (the green); simplified another (the red); and removed the race hazard (the yellow as shown in a following section).

- The inverse case is simplified as follows (SOP)

$$\overline{F} = \overline{A}\overline{B} + \overline{A}\overline{C} + \overline{A}D$$

By
S.Vanitha, Chennai