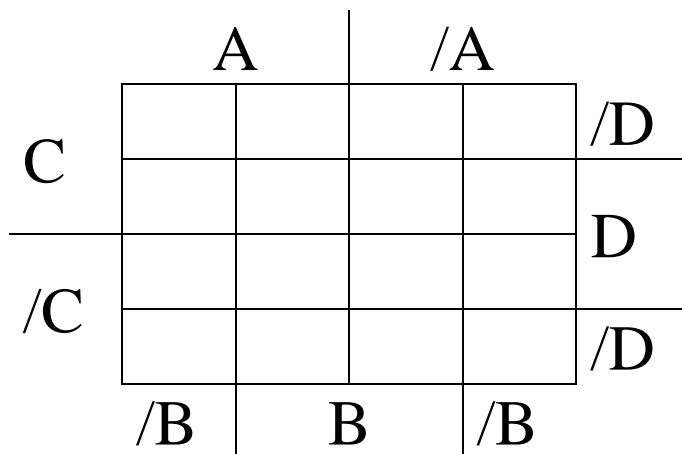


Karnaugh Maps

Karnaugh maps, or K-maps enable the simplification of Boolean expressions to be performed graphically. A K-map is a diagram on which **the sums-of-products** of Boolean expressions are plotted and which allows the elimination and reduction of terms to be made in a straight forward visual manner.

A K-map for n variables consists of 2^n squares or cells, so that **every input combination is shown**. Each cell may be regarded as having a co-ordinate formed from the combinations of input variables. Thus, for the four-variable K-map, the upper-left cell has the co-ordinate **A./B.C./D**, and the cell to the right of this has the co-ordinate **A.B.C./D**.

Write in the co-ordinates on the diagram below.



The cell co-ordinate system is arranged in such a way that **only one variable changes** (egg. from A to /A and vice versa) **on going from one square to the next horizontally or vertically**.

Furthermore, **the cell at the top left is also adjacent to the cell at the top right as well as the cell at bottom left**. The map may be thought of as being wrapped around cylinders both horizontally and vertically.

There is one K-map for every output in a design. So, if your circuit has three outputs, there will be three K-maps.

The procedure for filling in the K-map and then using it to reduce a Boolean expression is as follows:

- 1: A 1 is written into each cell of the map corresponding to each term in the original Sum of Products expression. In other words, a 1 is written into the K-Map for each row in the truth table for which the output is a 1. In certain designs, "don't care" terms may appear, in which case these may be entered onto the map as "x's".
- 2: Adjacent cells containing 1's (or 1's and "x's") are then combined. Cells can only be combined in groups of powers of two, for example 2,4,8,16,etc. Groups with sides 3

cells long are not allowed. There are three basic rules which must be followed when combining cells:

- a: every cell containing a 1 must be included in at least 1 group;
- b: each group should be as large as possible;
- c: 1's must may be included in more than one group if they increase the size of a group.

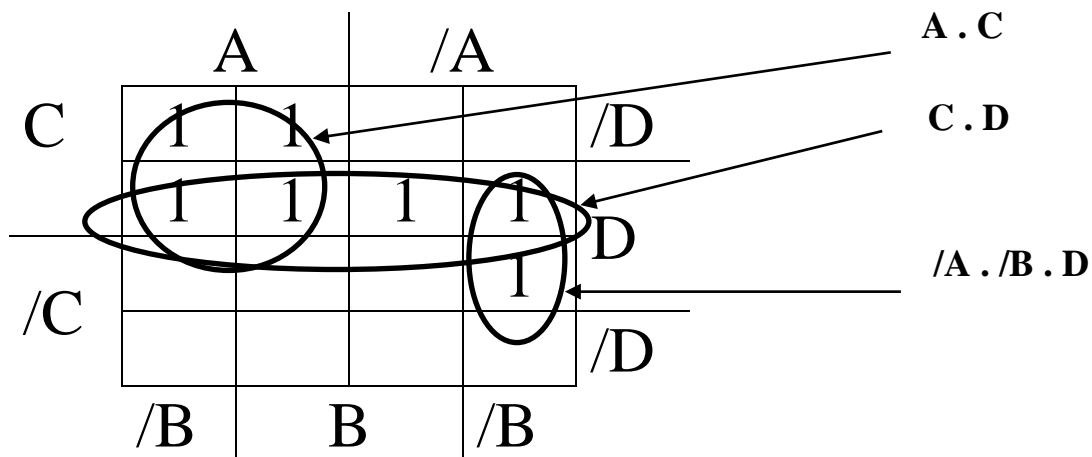
- 3:** Once the groupings have been made, the K-map may be read. Each group corresponds to a new minterm in the new expression, the value of which is found by considering those variables within each group which do not change.

Once the simplified expression has been obtained, further (easy) simplification may be possible. Such simplification may be made by applying the algebraic rules.

Example

For expression:

$$Z = A.B.C.D + A.B.C./D + A./B.C.D + A./B.C./D + /A.B.C.D + /A./B.C.D + /A./B./C.D$$



So the expression can be written as:

$$Z = A.C + C.D + /A.B.D$$