2.3 K-MAP (KARNAUGH MAP)

Karnaugh map method or K-map method is the pictorial representation of the Boolean equations and Boolean manipulations are used to reduce the complexity in solving them. These can be considered as a special or extended version of the 'Truth table'.

By using Karnaugh map technique, we can reduce the Boolean expression containing any number of variables, such as 2-variable Boolean expression, 3-variable Boolean expression, 4-variable Boolean expression and even 7-variable Boolean expressions, which are complex to solve by using regular Boolean theorems and laws.

Minimization with Karnaugh Maps and advantages of K-map

- K-maps are used to convert the truth table of a Boolean equation into minimized SOP form.
- Easy and simple basic rules for the simplification.
- The K-map method is faster and more efficient than other simplification techniques of Boolean algebra.
- All rows in the K-map are represented by using a square shaped cells, in which each square in that will represent a minterm.
- It is easy to convert a truth table to k-map and k-map to Sum of Products form equation.

There are 2 forms in converting a Boolean equation into K-map:

1. Un-optimized form
2. Optimized form

- Un-optimized form: It involves in converting the number of 1's into equal number of product terms (min terms) in an SOP equation.
- Optimized form: It involves in reducing the number of min terms in the SOP equation.

Grouping of K-map variables

- There are some rules to follow while we are grouping the variables in K-maps. They are

- The square that contains ‘1’ should be taken in simplifying, at least once.
- The square that contains ‘1’ can be considered as many times as the grouping is possible with it.
- Group shouldn’t include any zeros (0).
- A group should be the as large as possible.
- Groups can be horizontal or vertical. Grouping of variables in diagonal manner is not allowed.

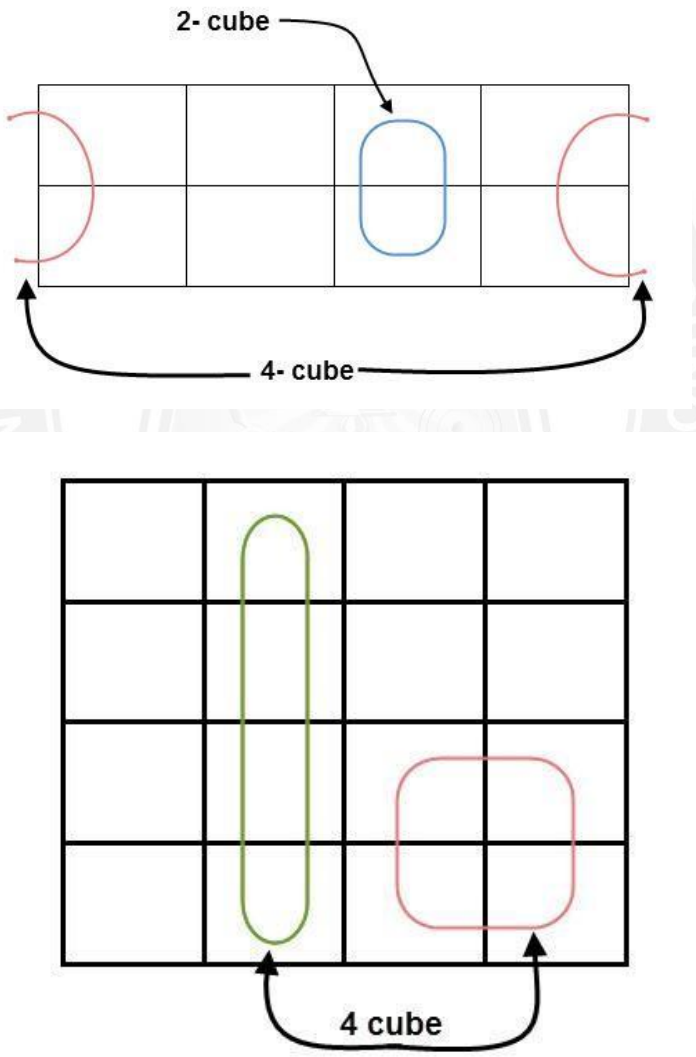


Figure 2.3.1 K-map

[Source: https://www.electronicshub.org/k-map-karnaugh-map/]

- If the square containing ‘1’ has no possibility to be placed in a group, then it should be added to the final expression.
- Groups can overlap.

- The number of squares in a group must be equal to powers of 2, such as 1, 2, 4, 8 etc.
- Groups can wrap around. As the K-map is considered as spherical or folded, the squares at the corners (which are at the end of the column or row) should be considered as they adjacent squares.
- The grouping of K-map variables can be done in many ways, so the obtained simplified equation need not to be unique always.
- The Boolean equation must be in must be in canonical form, in order to draw a K-map.

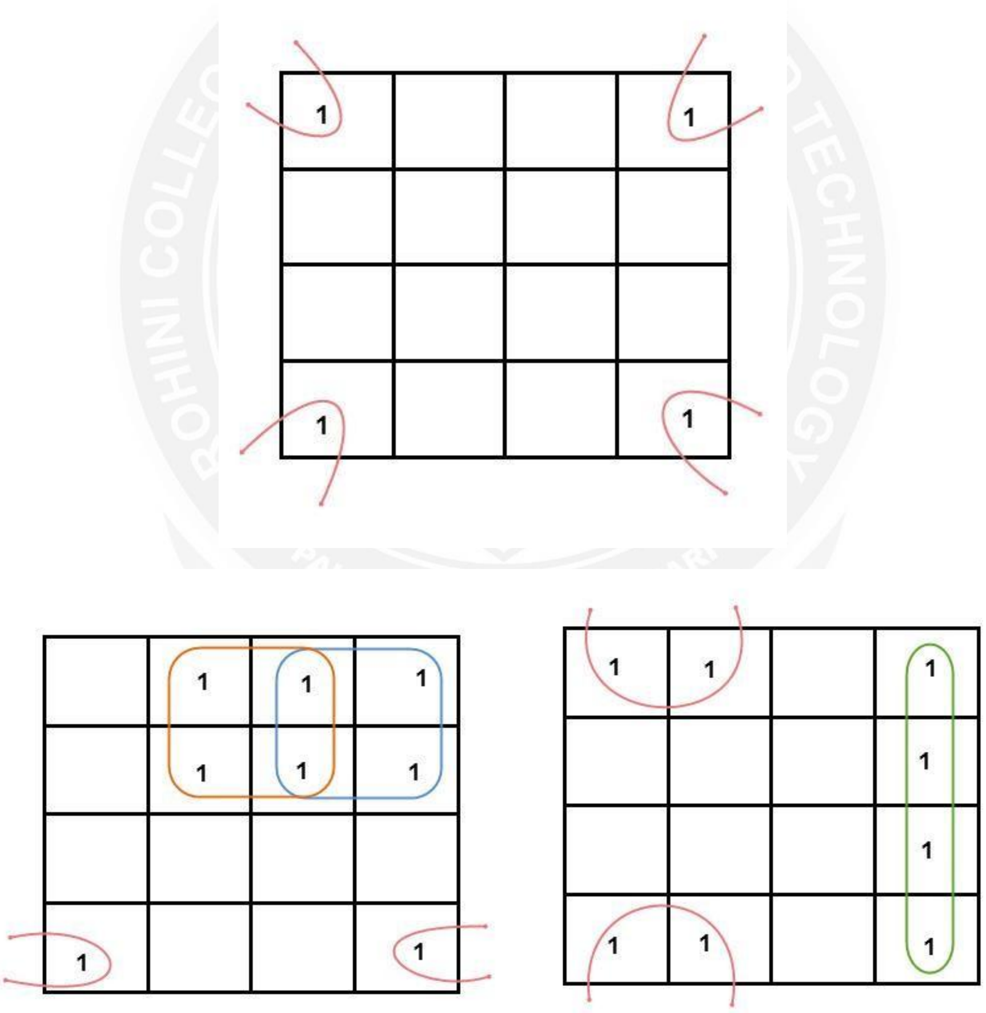
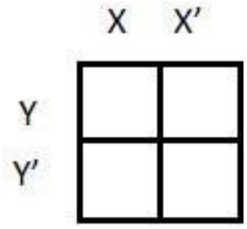


Figure 2.3.2 K-map Combination

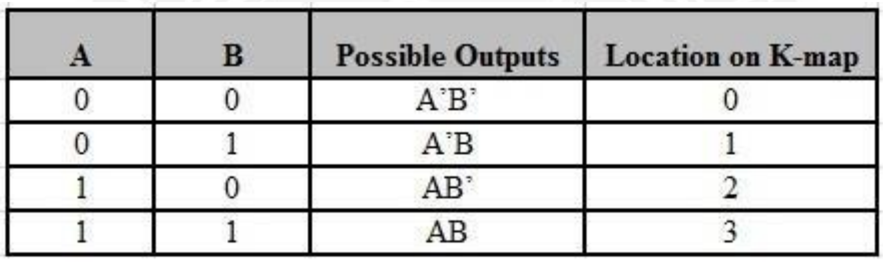
[Source: https://www.electronicshub.org/k-map-karnaugh-map/]

2 variable K-maps

There are 4 cells (22) in the 2-variable k-map. It will look like (see below image)

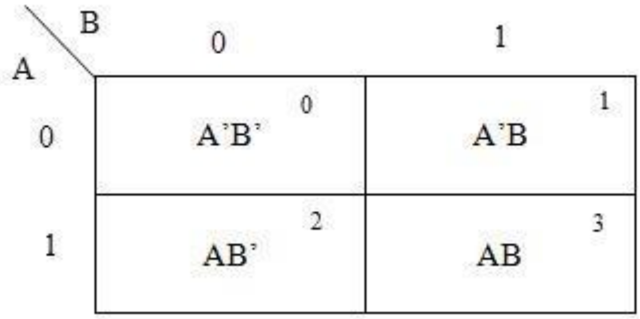


The possible min terms with 2 variables (A and B) are A.B, A.B’, A’.B and A’.B’. The conjunctions of the variables (A, B) and (A’, B) are represented in the cells of the top row and (A, B’) and (A’, B’) in cells of the bottom row. The following table shows the positions of all the possible outputs of 2-variable Boolean function on a K-map.



A	B	Possible Outputs	Location on K-map
0	0	A'B'	0
0	1	A'B	1
1	0	AB'	2
1	1	AB	3

A general representation of a 2 variable K-map plot is shown below.



When we are simplifying a Boolean equation using Karnaugh map, we represent the each cell of K-map containing the conjunction term with 1. After that, we group the adjacent cells with possible sizes as 2 or 4. In case of larger k-maps, we can group the variables in larger sizes like 8 or 16.

The groups of variables should be in rectangular shape, that means the groups must be formed by combining adjacent cells either vertically or horizontally. Diagonal shaped or

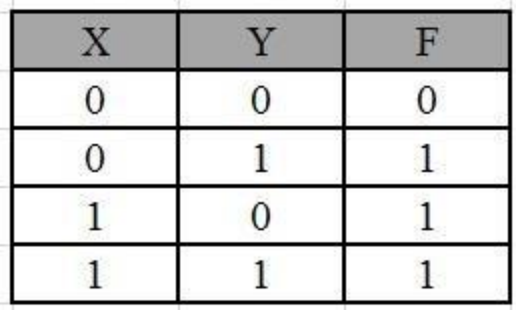
L-shaped groups are not allowed. The following example demonstrates a K-map simplification of a 2-variable Boolean equation.

Example

Simplify the given 2-variable Boolean equation by using K-map.

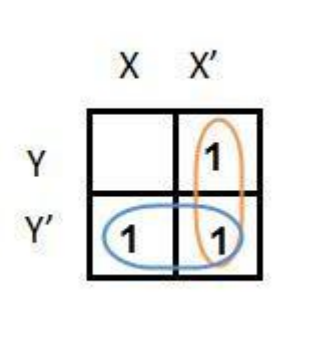
F = X Y’ + X’ Y + X’Y’

First, let’s construct the truth table for the given equation,



X	Y	F
0	0	0
0	1	1
1	0	1
1	1	1

We put 1 at the output terms given in equation.



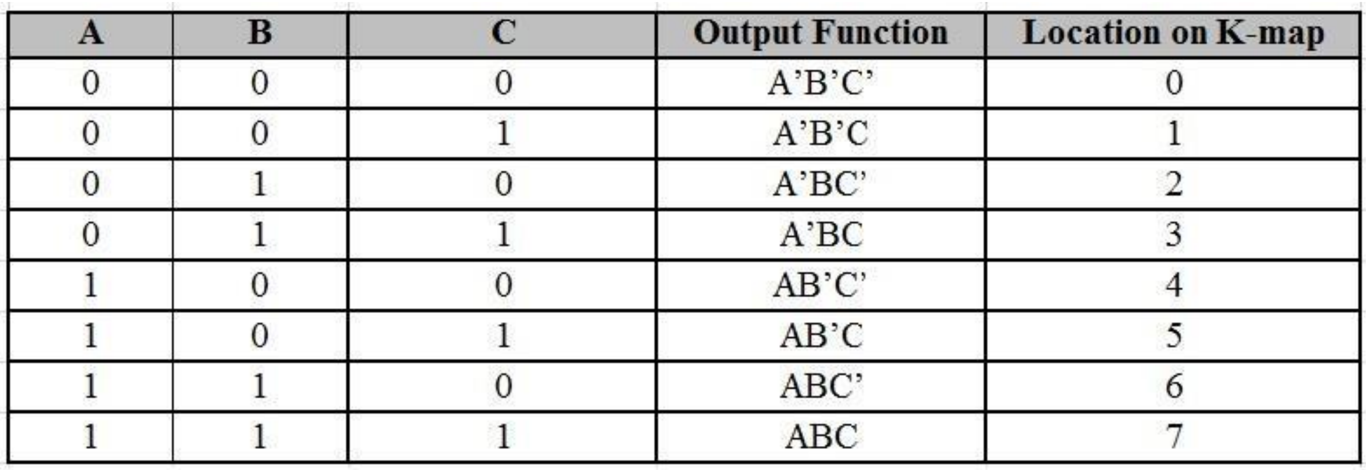
In this K-map, we can create 2 groups by following the rules for grouping, one is by combining (X’, Y) and (X’, Y’) terms and the other is by combining (X, Y’) and (X’, Y’) terms. Here the lower right cell is used in both groups.

After grouping the variables, the next step is determining the minimized expression.

By reducing each group, we obtain a conjunction of the minimized expression such as by taking out the common terms from two groups, i.e. X’ and Y’. So the reduced equation will be X’ +Y’.

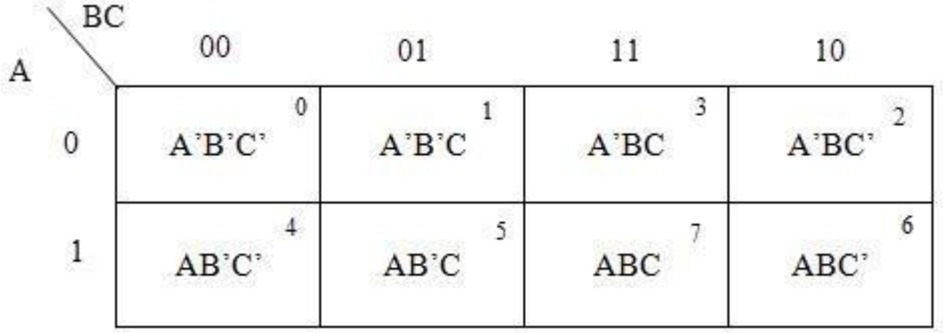
3 variable K-maps

For a 3-variable Boolean function, there is a possibility of 8 output min terms. The general representation of all the min terms using 3-variables is shown below.



A	B	C	Output Function	Location on K-map
0	0	0	A’B’C’	0
0	0	1	A’B’C	1
0	1	0	A’BC’	2
0	1	1	A’BC	3
1	0	0	AB’C’	4
1	0	1	AB’C	5
1	1	0	ABC’	6
1	1	1	ABC	7

A typical plot of a 3-variable K-map is shown below. It can be observed that the positions of columns 10 and 11 are interchanged so that there is only change in one variable across adjacent cells. This modification will allow in minimizing the logic.



A \ BC	00	01	11	10
0	A’B’C’ 0	A’B’C 1	A’BC 3	A’BC’ 2
1	AB’C’ 4	AB’C 5	ABC 7	ABC’ 6

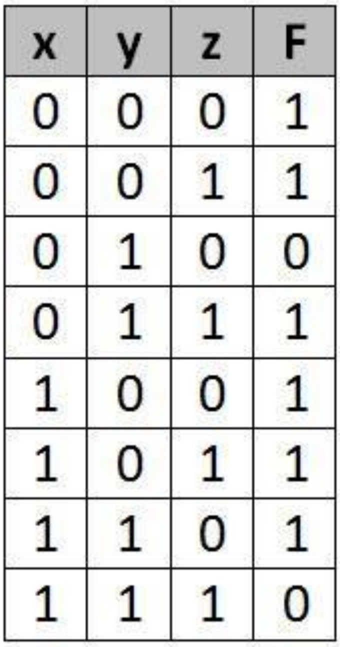
Up to 8 cells can be grouped in case of a 3-variable K-map with other possibilities being 1,2 and 4.

Example

Simplify the given 3-variable Boolean equation by using k-map.

F = X’ Y Z + X’ Y’ Z + X Y Z’ + X’ Y’ Z’ + X Y Z + X Y’ Z’

First, let’s construct the truth table for the given equation,

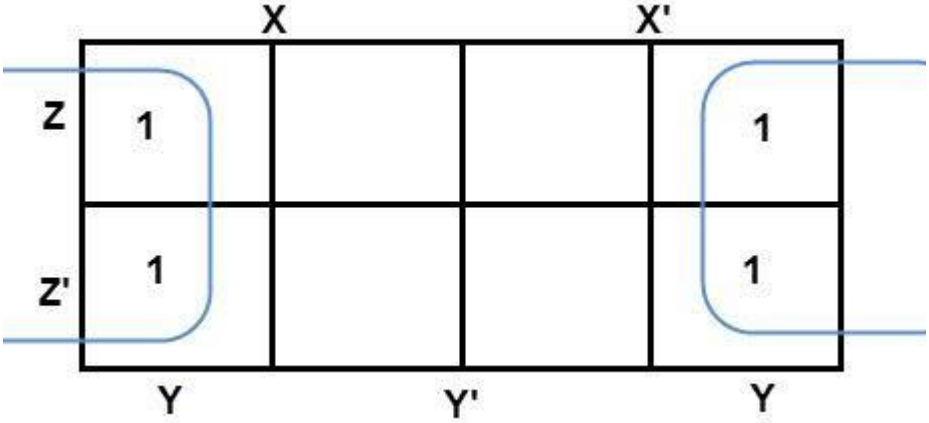


x	y	z	F
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

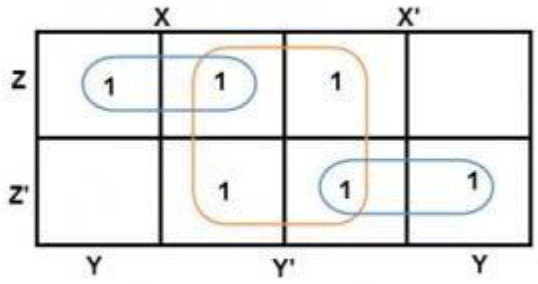
We put 1 at the output terms given in equation.

There are 8 cells (23) in the 3-variable k-map. It will look like (see below image).

The largest group size will be 8 but we can also form the groups of size 4 and size 2, by possibility. In the 3 variable Karnaugh map, we consider the left most column of the k-map as the adjacent column of rightmost column. So the size 4 group is formed as shown below.



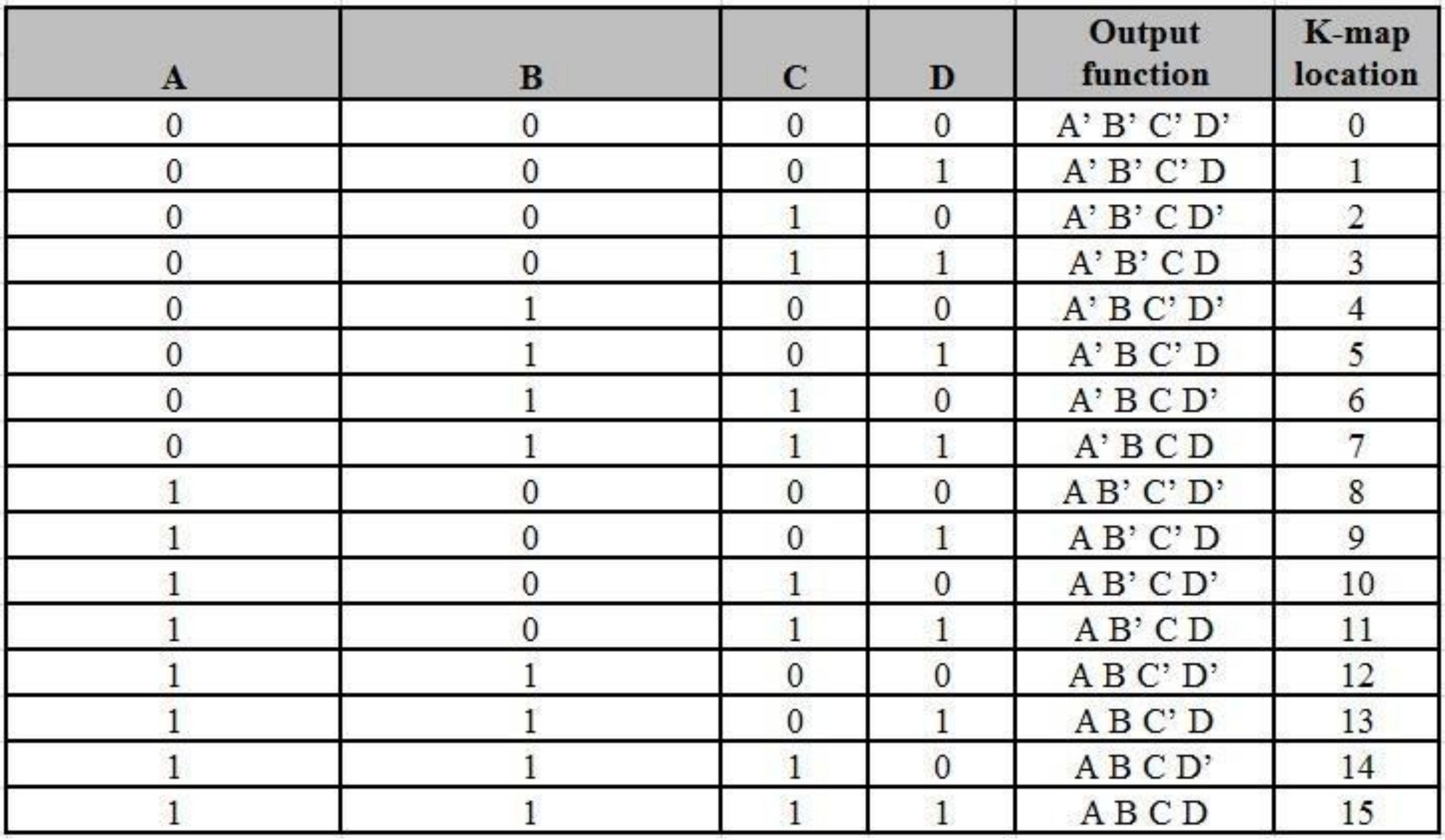
And in both the terms, we have 'Y' in common. So the group of size 4 is reduced as the conjunction Y. To consume every cell which has 1 in it, we group the rest of cells to form size 2 group, as shown below.



The 2 size group has no common variables, so they are written with their variables and its conjugates. So the reduced equation will be X Z’ + Y’ + X’ Z. In this equation, no further minimization is possible.

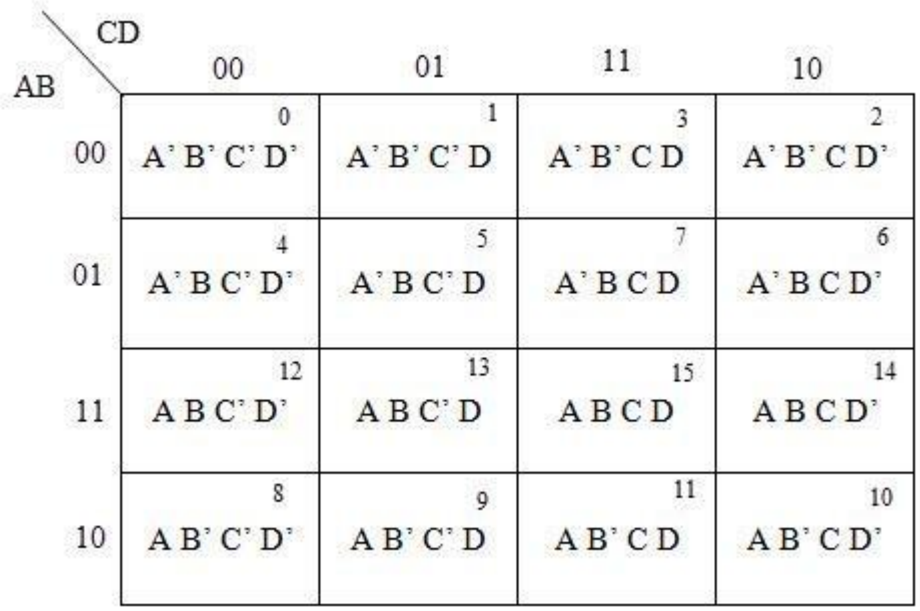
4 variable K-maps

There are 16 possible min terms in case of a 4-variable Boolean function. The general representation of minterms using 4 variables is shown below.



A	B	C	D	Output function	K-map location
0	0	0	0	A’ B’ C’ D’	0
0	0	0	1	A’ B’ C’ D	1
0	0	1	0	A’ B’ C D’	2
0	0	1	1	A’ B’ C D	3
0	1	0	0	A’ B C’ D’	4
0	1	0	1	A’ B C’ D	5
0	1	1	0	A’ B C D’	6
0	1	1	1	A’ B C D	7
1	0	0	0	A B’ C’ D’	8
1	0	0	1	A B’ C’ D	9
1	0	1	0	A B’ C D’	10
1	0	1	1	A B’ C D	11
1	1	0	0	A B C’ D’	12
1	1	0	1	A B C’ D	13
1	1	1	0	A B C D’	14
1	1	1	1	A B C D	15

A typical 4-variable K-map plot is shown below. It can be observed that both the columns and rows of 10 and 11 are interchanged.



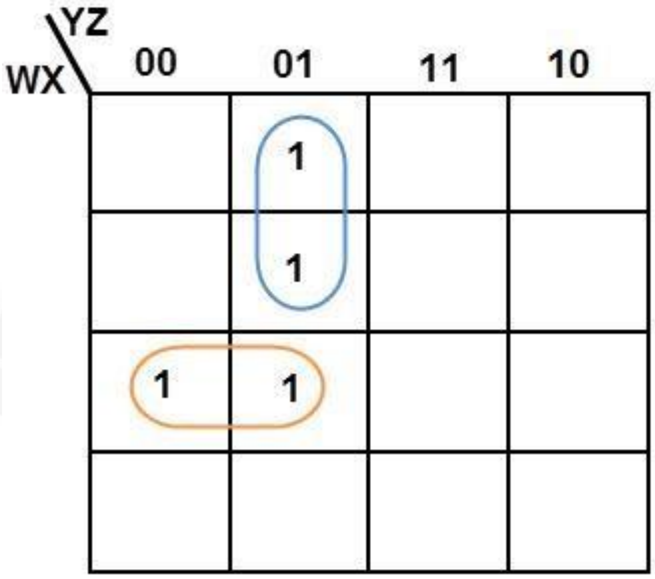
AB \ CD	00	01	11	10
00	0 A' B' C' D'	1 A' B' C' D	3 A' B' C D	2 A' B' C D'
01	4 A' B C' D'	5 A' B C' D	7 A' B C D	6 A' B C D'
11	12 A B C' D'	13 A B C' D	15 A B C D	14 A B C D'
10	8 A B' C' D'	9 A B' C' D	11 A B' C D	10 A B' C D'

The possible number of cells that can be grouped together are 1, 2, 4, 8 and 16.

Example

Simplify the given 4-variable Boolean equation by using k-map. F (W, X, Y, Z) = (1, 5, 12, 13)

Sol: F (W, X, Y, Z) = (1, 5, 12, 13)



WX \ YZ	00	01	11	10
		1		
		1		
	1	1		

By preparing k-map, we can minimize the given Boolean equation as

F = W Y’ Z + W ‘Y’ Z