Chapter II

Boolean function representation and minimization techniques

- Standard and canonical representation
- Minimization of Boolean expressions using Karnaugh map

Standard forms of Boolean Expressions

All Boolean expressions, regardless of their form, can be converted into either of two standard forms:

- The Sum of Products (SOP) Form
- The Product of Sums (POS) Form

The Sum of Products (SOP) Form

- A product term is defined as a term consisting of the product (Boolean multiplication) of literals.
- When two or more product terms are summed by Boolean addition, the resulting expression is sum-of-products (SOP).

Examples:

AB + ABC

ABC+DCE+B’C’F

Domain of a Boolean Expression is the set of variables contained in either complemented or uncomplemented form.

Example: The domain of AB'+AB'C is the set of variables A, B, C.

Implementation of an SOP expression simply requires ORing the outputs of two or more AND gates.

Any logic expression can be converted into SOP form by applying Boolean algebra techniques.

Example: A(B+CD) can be converted to SOP form by applying distributive law:

$A(B+CD) = AB+ACD$

The Standard SOP Form

All the product terms do not contain all of the variables in the domain of the expression. Example: A'BC'+ AB'D + A'BC'D has a domain made up of variables A,B,C and D.

But the complete set of variables in the domain is not represented in the first two terms of the expression, i.e. D or D' is missing from the first term and C or C' from the second term.

A standard SOP expression is one in which all the variables in the domain appear in each product term in the expression.

Example: AB'CD+ A'B‘CD' + A'BC'D is a standard SOP expression.

Standard SOP forms are important in constructing the truth tables.

Any non standard SOP expression can be converted to standard SOP expression using Boolean algebra.

Converting Product Terms to Standard SOP

Following steps are involved:

1. Multiply each nonstandard product term by a term made up of the sum of a missing variable and its complement. This results in two product terms.
2. Repeat step 1 until all resulting product terms contain all variables in the domain in either complemented or uncomplemented form. In converting a product term to standard SOP form, the number of product term is doubled for each missing variable.

Example

Convert the following Boolean expression into standard SOP form

AB'C + A'B' +ABC'D

The domain of this SOP expression is A,B,C,D.

Consider 1st term : AB'C –in this D or D' is missing, so multiply the first term by D +D' as

AB'C = AB'C(D +D') = AB'CD + AB'CD'.

Thus two product terms have resulted here.

Similarly consider the 2nd term:

A'B' has two missing variables, C or C' and D or D'.

First multiply by C+C' as

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

The two resulting terms have missing variables D or D'.

So A'B'C(D+D') + A'B'C'(D+D') = A'B'CD + A'B'CD' + A'B'C'D+A'B'C'D'.

The third term is already in standard form.

So the complete SOP form for the above expression is

AB'C + A'B' +ABC'D = AB'CD + AB'CD' +A'B'CD + A'B'CD' + A'B'C'D+A'B'C'D'+ ABC'D.

Binary Representation of a Standard Product Term

A standard product term is equal to 1 for only one combination of variable values. E.g. the product term AB'CD' is equal to 1 when A=1,B=0,C=1,D=0 and is 0 for all other combinations. The product term has a value of 1010(10 in decimal).

An SOP expression is equal to 1 only if one or more of the product terms in the expression is equal to 1.

The Product of Sums (POS) Form

A sum term is defined as a term consisting of the sum of the literals. When two or more sum terms are multiplies, the resulting expression is product-of-sums (POS) form.

Example:

(A+B)(A+B+C ')

A POS expression can contain a single term.

Implementing a POS expression simply requires ANDing the outputs of two or more OR gates.

A standard POS expression is the one in which all the variables in the domain appear in each sum term in the expression.

To convert a nonstandard POS expression to standard POS expression:

1. Add to each nonstandard product term a term made up of the product of missing variable and its complement.
2. Apply the rule A+BC = (A+B)(A+C).
3. Repeat step 1 until the resulting term contains all the variables in complemented or uncomplemented form.

Example

Convert the following expression to standard POS form

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

The domain of this POS is A,B,C,D.

1st term A+B'+C– to it add DD‘.

A+B'+C = A+B'+C+DD’ = (A+B'+C+D)(A+B'+C+D’)

Thus the final expression will be

(A+B'+C+D)(A+B'+C+D’)(A+B'+C+D')(A’+B'+C+D')(A+B'+C'+D)

Converting standard SOP to standard POS

1. Evaluate each product term in the given standard SOP expression , i.e. determine the binary numbers that represent the product terms.
2. Determine all of the binary numbers not included in the step1.
3. Write equivalent sum term for each binary number from step 2 and express in POS form.

Example

Convert A’B’C’+A’BC’+AB’C+ABCto standard POS form

The evaluation is done as

000+010+011+101+111

Remaining 3 combinations which must be included in POS are 001,100,110.

(A+B+C’)(A’+B+C)(A’+B’+C)

Boolean expressions and truth tables

All Boolean expressions can be converted into truth tables.

A truth table is a common way of presenting in a concise format, the logical operation of a circuit.

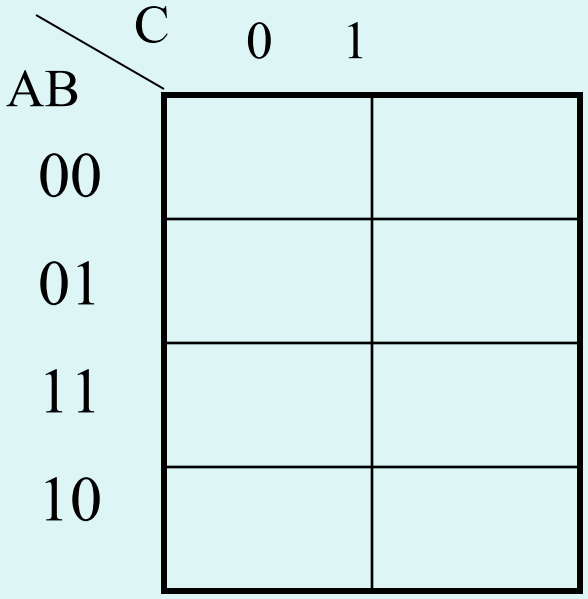
The Karnaugh Map

- A Karnaugh map (K-map) is similar to a truth table because it presents all of the possible values of input variables and the resulting output for each value.
- Instead of being organized into columns and rows like a truth table, the K-map is an array of cells in which each cell represents the binary value of the input variable.
- The cells are arranged in such a way that the simplification of a given expression is simply a matter of properly grouping the cells.

- The number of cells in a K-map is equal to the total number of possible input variable combinations as is the number of rows in a truth table.
- Example: for 2 variable, 4 cells will be there.
- For 3—8 cells
- For 4 --- 16 cells and so on.

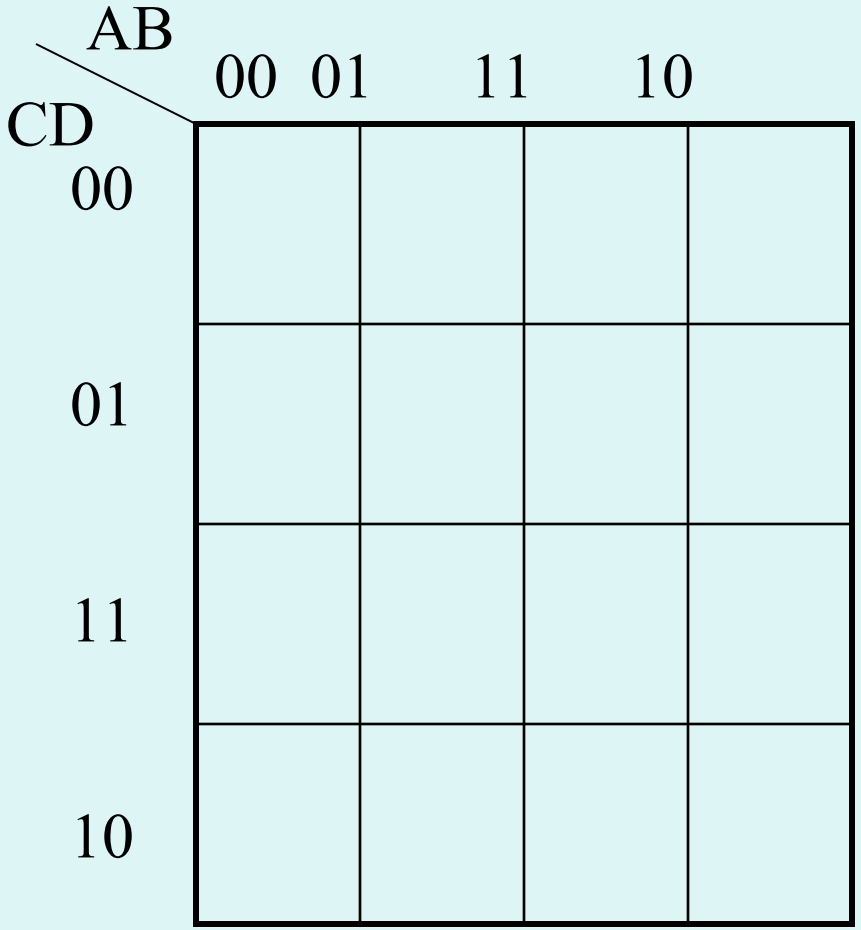
The 3-Variable Karnaugh Map

- A 3-variable K-map is an array of eight cells.
- Binary values of A and B are on the left side and the values of C are at the top of the same column.



The 4-Variable Karnaugh Map

- 4-Variable K-map is an array of 16 cells.
- A and B and C and D can be interchanged.
- All the cells contain binary values of 0000 to 1111.



Mapping a standard SOP Expression

For an SOP expression in the standard form, a 1 is placed on the K-map for each product term in the expression.

Each 1 is placed in the cell corresponding to the value of the product term.

Example

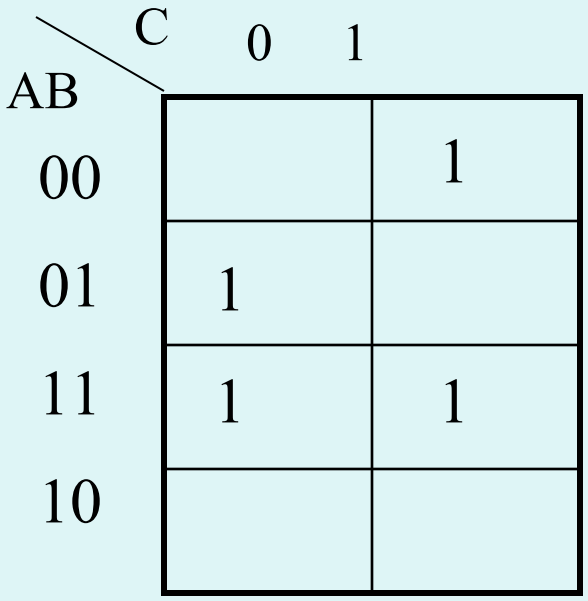
A’B’C+A’BC’+ABC’+ABC

It will be a 3-variable K-map

The expression is

001 010 110 111

The Boolean expression must be in standard or canonical form. If the expression is not in standard for, it has to be converted to the standard form before entering the values in K-map.



K-map Simplification of SOP Expressions

The process that results in an expression containing the fewest possible terms with the fewest possible variables is called **minimization**.

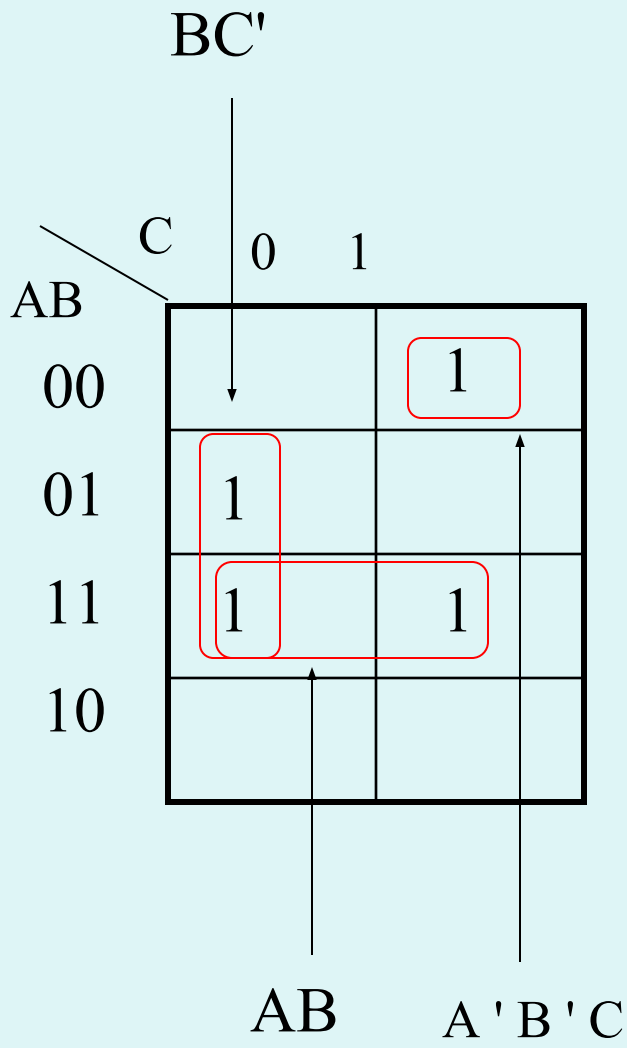
After mapping the SOP expression in K-map, following steps are to be followed to obtain the minimized SOP expression.

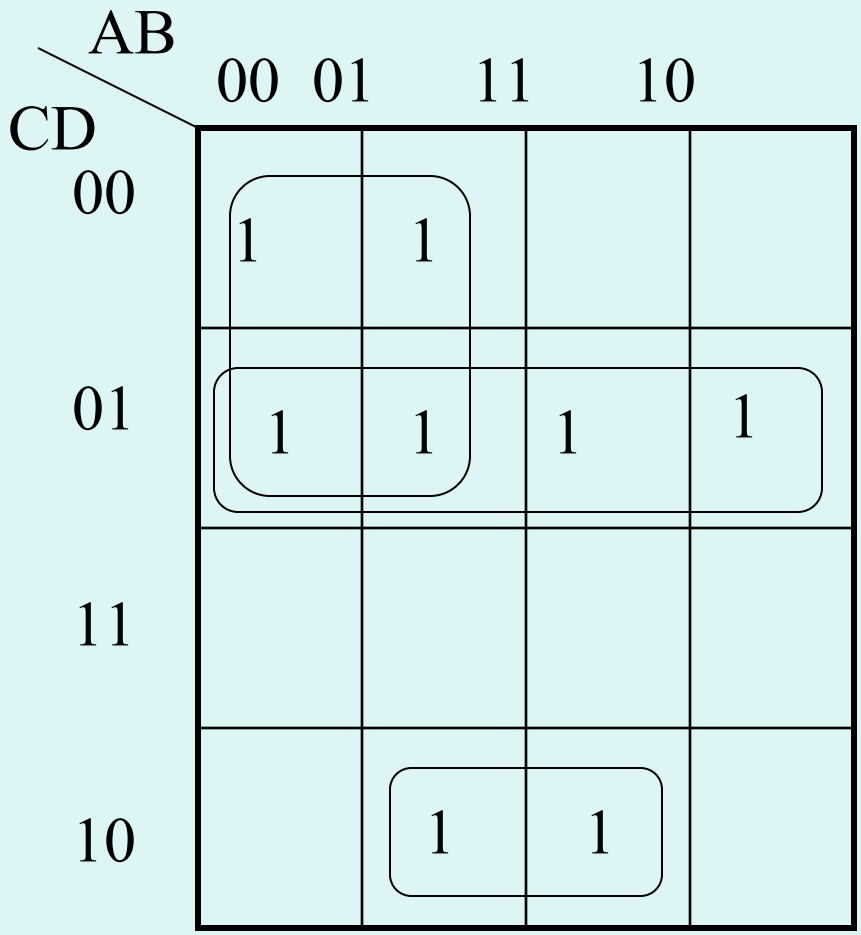
1s are grouped in the K-map using the following steps.

The main aim is to maximize the size of group and to minimize the number of groups

- A group must contain either 1,2,4,8 or 16 cells, which are all powers of 2. Example: In a 3-variable K-map, $2^3=8$ is the maximum group.
- Each cell in the group must be adjacent to one or more cells in that same group, but all cells in the group do not have to be adjacent to each other.
- Always include largest possible number of 1s in a group.
- Each 1 on the map must be included in at least one group. Overlapping can be done as long as this contains the non common 1s

Examples


BC'
C
0
1
AB
00
01
11
10
1
1
1
1
AB
A ' B ' C


AB
00
01
11
10
CD
00
01
11
10
1
1
1
1
1
1
1
1
1
1

Determining the minimum SOP expression

1. Group the cells that have 1s. Each group of the cells containing 1s creates one product term composed of all variables that occur in only one form within the group. Variables that occur both uncomplemented and complemented are eliminated.
2. Determine the minimum product term for each group.

I. For a 3-variable K-map

a) A 1-cell group yields a 3-variable product term.
b) A 2-cell group yields a 2-variable product term.
c) A 4-cell group yields a 1-variable product term.
d) A 8-cell group yields a value 1.

II. For a 4-variable K-map

a) A 1-cell group yields a 4-variable product term.

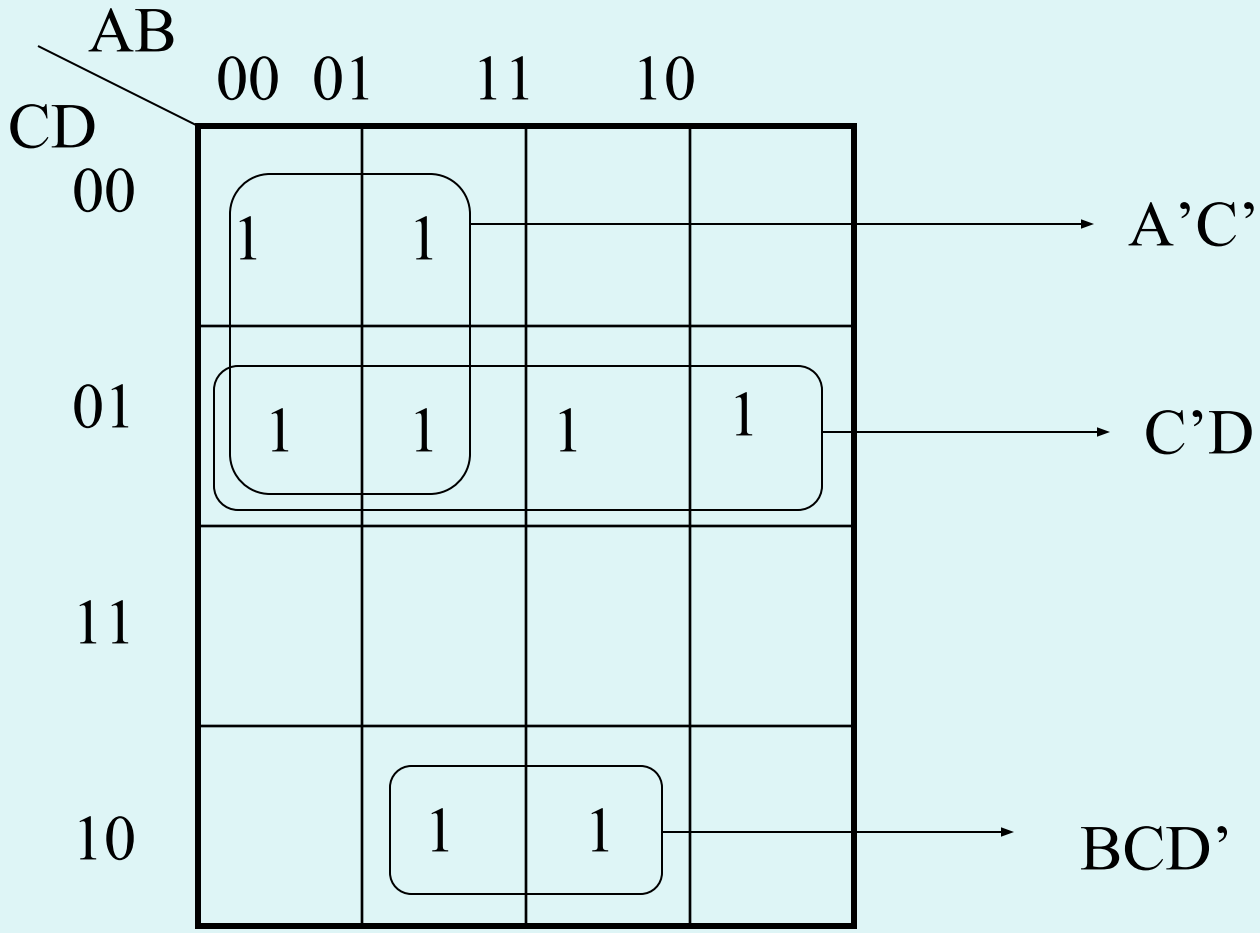
b) A 2-cell group yields a 3-variable product term.

c) A 4-cell group yields a 2-variable product term.

d) A 8-cell group yields a 1-variable product term.

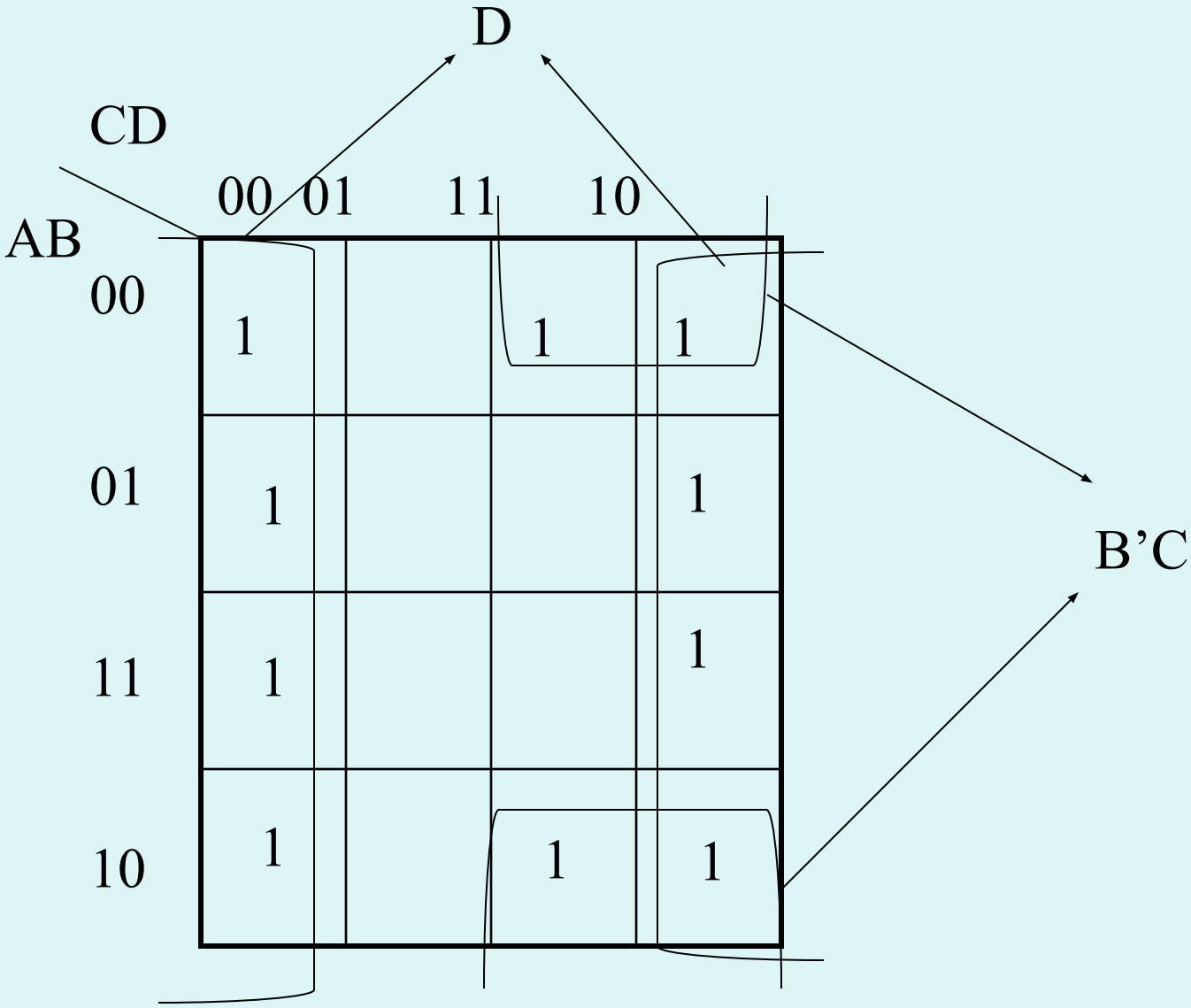
e) A 16-cell group yields a value 1.

3. When all the minimum product terms are derived form the K-map, they are summed to form the minimum SOP expression.


AB
00 01 11 10
CD
00
01
11
10
1 1
1 1 1 1
1 1
A’C’
C’D
BCD’

Use K-map to minimize the following expression

B’C’D+A’BC’D+ABC’D’+A’B’CD+AB’CD +A’B’CD’+A’BCD’+ABCD’+AB’CD’


D
CD
00 01 11 10
AB
00
01
11
10
1 1 1
1 1
1 1
1 1 1
1 1 1
B’C

Karnaugh Maps

- Karnaugh maps (K-maps) -- convenient tool for representing switching functions of up to six variables.
- K-maps form the basis of useful heuristics for finding MSOP and MPOS representations.
- An *n*-variable K-map has 2^n cells with each cell corresponding to a row of an *n*-variable truth table.
- K-map cells are labeled with the corresponding truth-table row.
- K-map cells are arranged such that adjacent cells correspond to truth rows that differ in only one bit position (*logical adjacency*).
- Switching functions are mapped (or plotted) by placing the function's value (*0,1,d*) in each cell of the map.

Don’t care conditions

Sometimes a situation arises in which some input variable combinations are not allowed. For example, the six invalid combinations in BCD (1010….1111).

Since these unallowed states will never occur in an application involving the BCD code, they are treated as “don’t care” terms with respect to their effect on the output.

So for these don’t care terms, either 1 or 0 may be assigned to the output, it really does not matter.

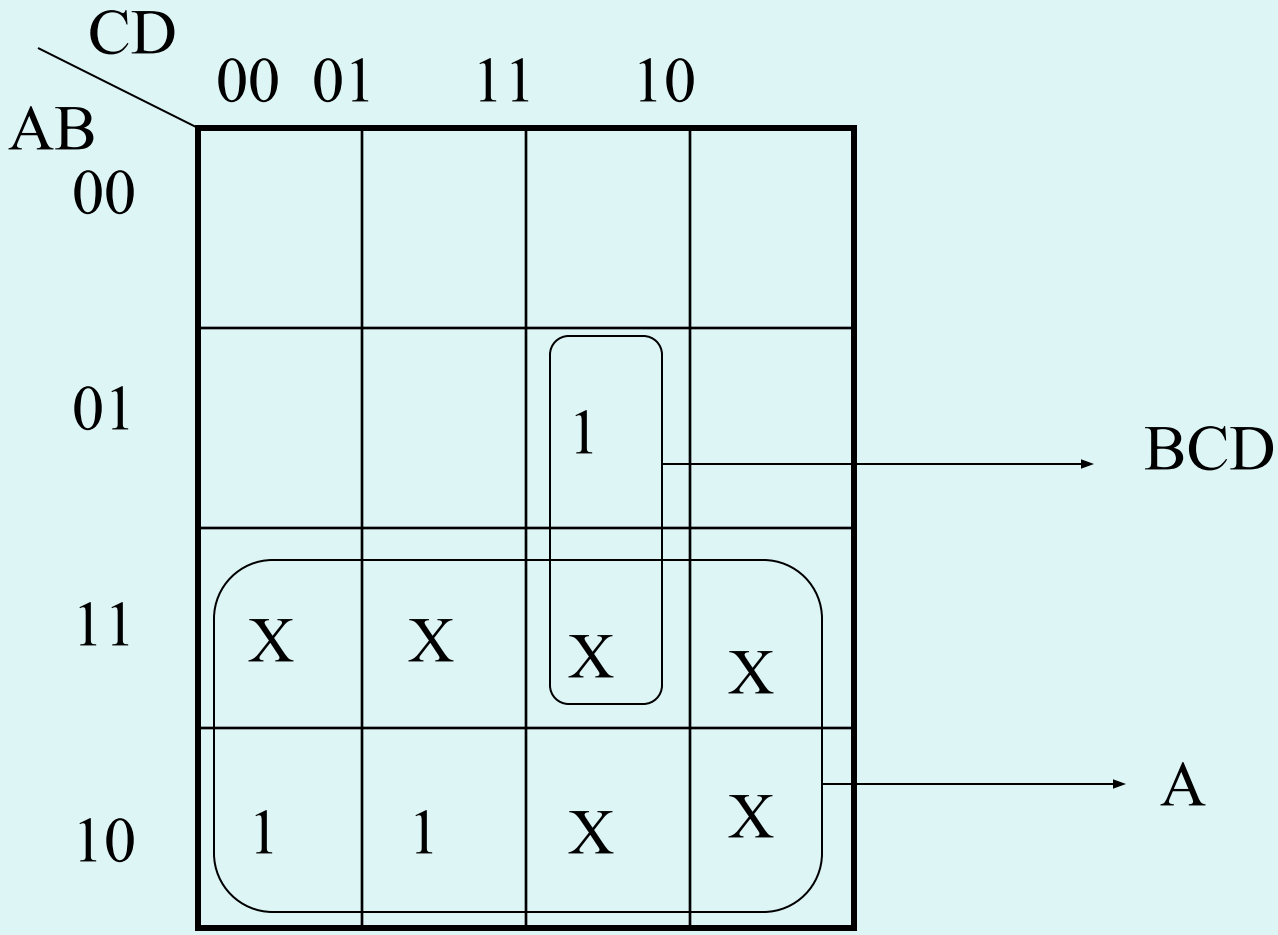
These don’t care terms can be used to advantage on K-maps.

Example:

$Y(A,B,C,D)= \Sigma m(7,8,9)+d(10,11,12,13,14,15)$

In this example, map the SOP expression in K-map and for all the don’t care values, mark ‘X’.

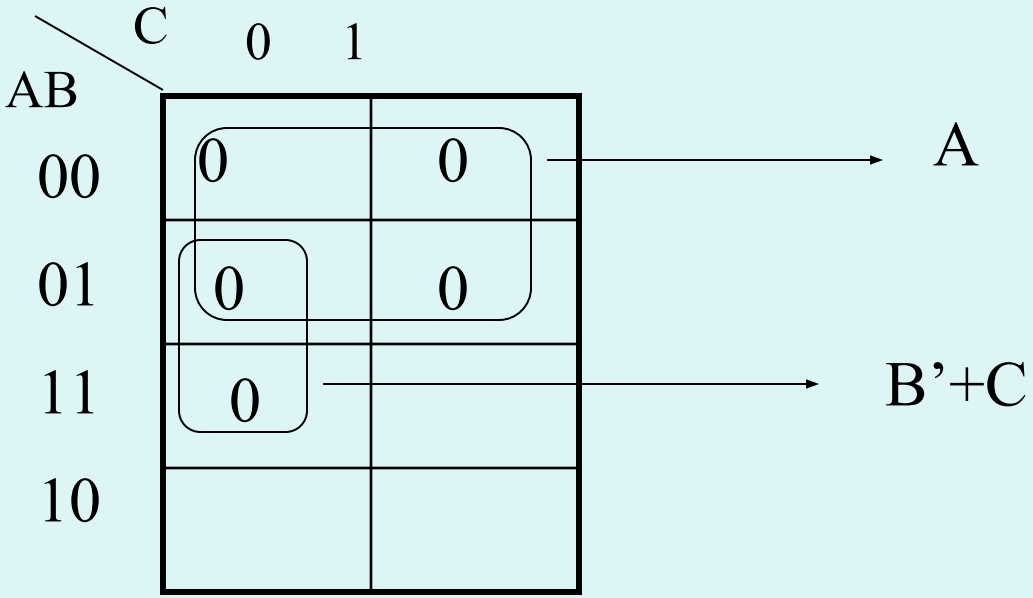
And then try to group these don’t cares also.


CD
00 01 11 10
AB
00
01
11
10
1
X X X X
1 1 X X
BCD
A

K-Map POS Minimization

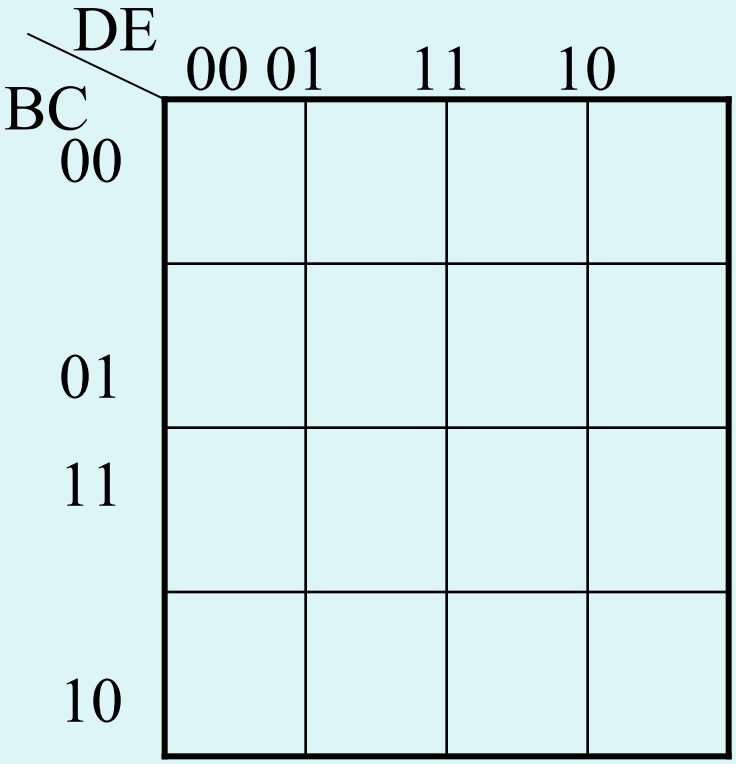
1. Determine the binary value of each sum in the standard POS expression. This is the binary value that makes the term equal to 0.
2. As each sum term is evaluated, place a 0 on the K-map in the corresponding cell.

(A+B+C)(A+B+C')(A+B'+C)(A+B'+C')(A'+B'+C)

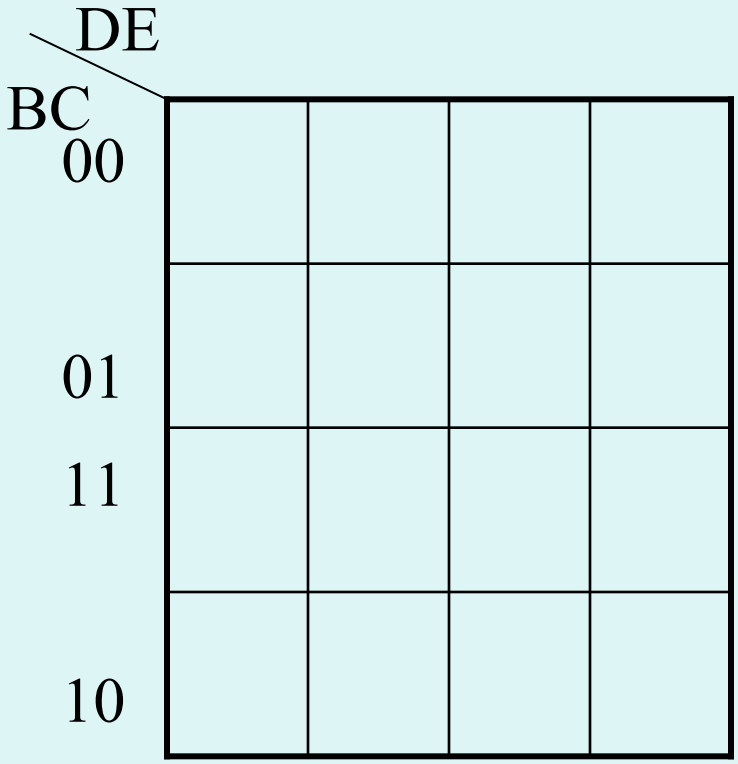


Final Expression is A(B'+C)

Five variable K-map



A=0



A=1

Y= Σm(0,1,4,8,12,13,15,16,17,23,29,31)