= ABC + ABC′ + AB′C + AB′C′ + ABC + A′BC
= ABC + ABC′ + AB′C + AB′C′ + A′BC [Canonical SOP]

Min Term Designation

Min term designation is denoted with the letter M along with a subscript obtained by using the following steps:

(*i*) Find binary pattern by writing 1 for the variable and 0 for the complement of the variable.
(*ii*) Find decimal equivalent of the binary pattern.
(*iii*) Use decimal number as a subscript of the letter M.

For example,

- **To find Min Term Designation of ABC′.**
  Binary pattern of input values of ABC′ : 110
  Decimal equivalent of binary value 110 : 6
  Hence, min term designation of ABC′ is : M_6
- **To find Min Term Designation of A′B′C.**
  Binary pattern of input values of A′B′C : 001
  Decimal equivalent of binary value 001 : 1
  Hence, min term designation of A′B′C is : M_1
- **To find Min Term Designation of ABC.**
  Binary pattern of input values of ABC : 111
  Decimal equivalent of binary value 111 : 7
  Hence, min term designation of ABC is : M_7

Obtaining Min Term Expression from Truth Table

To obtain min term expression from the truth table, use the steps as given below:

(*i*) See the input bit pattern of the variables which results in 1.
(*ii*) If the input bit is 0, use complement of the variable.
(*iii*) If the input bit is 1 then use the variable in the same form.
(*iv*) Find their product to obtain min term.
(*v*) Finally add all the min terms to get min term expression or canonical SOP expression.

Consider the truth tables given below:

A	B	Y	MIN TERMS	MIN TERM DESIGNATION
0	0	0		
0	1	1	A′B (A′ for 0 and B for 1)	M_1
1	0	1	AB′ (A for 1 and B′ for 0)	M_2
1	1	1	AB (A for 1 and B for 1)	M_3

Hence, min term expression (Y) = A′B + AB′ + AB.

Min term expression by using min term designation: $M_1 + M_2 + M_3$

The boolean expression obtained can also be represented as:

$F(A \cdot B) = \Sigma(1, 2, 3)$, where 1, 2 and 3 are the subscripts of the min term designations and Σ stands for summation.

A	B	C	Y	MIN TERM	MIN TERM DESIGNATION
0	0	0	0		
0	0	1	1	A′B′C	M_1
0	1	0	1	A′BC′	M_2
0	1	1	0		
1	0	0	0		
1	0	1	1	AB′C	M_5
1	1	0	1	ABC′	M_6
1	1	1	1	ABC	M_7

Hence, min term expression Y = A'B'C + A'BC' + AB'C + ABC' + ABC

Min Term expression by using min term designation: $M_1 + M_2 + M_5 + M_6 + M_7$

The boolean expression obtained can also be represented as:

F(A·B, C) = Σ(1, 2, 5, 6, 7), where 1, 2, 5, 6 and 7 are the subscripts of the min term designations.

Example: Obtain min term expression from the boolean function as given below:
F(A, B, C) = Σ(5, 1, 3, 6).

Ans.

M_5 represents 101: AB'C

M_1 represents 001: A'B'C

M_3 represents 011: A'BC

Hence, min term expression is: AB'C + A'B'C + A'BC

Product of Sum (POS)

This type of expression is formed by the product of the sum terms of the variables.

For example, A.B can be written as (A + 0)(B + 0) and

A.(B + C) can be written as (A + 0)(B + C).

In the first example shown above A + 0 and B + 0 are the two sum terms multiplied together to form the expression. Similarly, in the next example A + 0 and B + C are multiplied to form the expression. Hence, they are product of sum expressions.

Canonical Product of Sum (POS) Expression

A product of sum expression which is formed by the product of all the sum terms in such a way that each term possesses all the variables (may be complement or non-complement forms) applied in the system.

For example, (A + B)(A + B')(A' + B').

The expression shown above contains each term with the combination of both the variables used in the expression. Hence, it is a canonical POS expression.

You must note that each sum term of the expression containing all the variables is also termed as max term. Hence, the expression is the product of the max terms and is also referred as max term expression.

Conversion into Canonical POS Expression

To convert a given expression into canonical POS or max term expression, the following steps can be used:

(*i*) Find the term where any variable is missing.

(*ii*) Add the product of missing variable and its complement with the existing variable.

(*iii*) Simplify the expression to obtain canonical POS expression.

Example: Convert the following expressions into its equivalent canonical POS form:

(*i*) (A.B)

Ans. First term is missing B whereas second term is missing A.

Hence, (A + B.B')(A.A' + B)

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

= (A + B) (A + B') (A' + B) [Canonical POS]

(*ii*) A(B + C)

Ans. First term is missing A and B both the variables whereas second term is missing only A.

Hence, (A + BB' + CC')(AA' + B + C)

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

= (A + B + C)(A + B + C')(A + B' + C)(A + B' + C') (A' + B + C) [Canonical POS]

Max Term Designation

Max Term designation is denoted with the letter M along with a subscript obtained by using the following steps:

(i) Find binary pattern by writing 0 for the variable and 1 for the complement of the variable.
(ii) Find decimal equivalent of the binary pattern.
(iii) Use decimal number as a subscript of the letter M.

For example,

- **To find Max Term Designation of (A + B + C′).**
  Binary pattern of input values of (A + B + C′) : 001
  Decimal Equivalent of binary value 001 : 1
  Hence, max term designation of (A + B + C′) is : M_1
- **To find Max Term Designation of (A′ + B′ + C).**
  Binary pattern of input values of (A′ + B′ + C) : 110
  Decimal Equivalent of binary value 110 : 6
  Hence, max term designation of A′B′C is : M_6
- **To find Max Term Designation of (A + B + C).**
  Binary pattern of input values of (A + B + C) : 000
  Decimal Equivalent of binary value 000 : 0
  Hence, max term designation of (A + B + C) is : M_0

Obtaining Max Term Expression from Truth Table

To obtain max term expression from the truth table, use the steps given below:
(i) See the input bit pattern of the variables which results in 0.
(ii) If the input bit is 1, use complement of the variable.
(iii) If the input bit is 0 then use the variable in the same form.
(iv) Find their sum to obtain max term.
(v) Finally multiply all the max terms to get max term expression or canonical POS expression.

Consider the truth tables as given below:

A	B	Y	MAX TERMS	MAX TERM DESIGNATION
0	0	0	A + B (A for 0 and B for 0)	M_0
0	1	0	A + B′(A for 0 and B′ for 1)	M_1
1	0	0	A′ + B (A′ for 1 and B for 0)	M_2
1	1	1		

Hence, max term expression Y = (A + B)(A + B′)(A′ + B)
Max term expression by using designation $= M_0 \cdot M_1 \cdot M_2$
The boolean expression obtained can also be represented as:
F(A.B) = π(0, 1, 2), where 0, 1 and 2 are the subscripts of the max term designations and π stands for product.

A	B	C	Y	MAX TERM	MAX TERM DESIGNATION
0	0	0	0	A + B + C	M_0
0	0	1	1		
0	1	0	1		
0	1	1	0	A + B′ + C′	M_3
1	0	0	0	A′ + B + C	M_4
1	0	1	1		
1	1	0	1		
1	1	1	1		

Hence, max term expression Y = (A + B + C) (A + B′ + C′) (A′ + B + C)
Max term expression by using designation $= M_0 \cdot M_3 \cdot M_4$
The boolean expression obtained can also be represented as:
F(A·B, C) = π(0, 3, 4), where 0, 3 and 4 are the subscripts of the max term designations.

Example: Obtain max term expression for the following boolean function:
F(A, B, C) = π(7, 0, 3, 5)

Ans.

M_7 represents: 111: $(A' + B' + C')$
M_0 represents: 000: $(A + B + C)$
M_3 represents: 011: $(A + B' + C')$
M_5 represents: 101: $(A' + B + C')$

Hence, Max Term expression: $(A' + B' + C')(A + B + C)(A' + B + C')$.

KARNAUGH MAP

You have come across simplification of boolean expression by using postulates or laws. This is feasible only if the expression is short and can be simplified by using limited steps. Sometimes, it becomes very difficult to simplify an expression when its derivation is longer.

Maurice Karnaugh, a great mathematician, developed a tabular or matrix representation which can enable you to obtain an expression in its simplest form. This tabular representation is termed as **K-map** or **Karnaugh map**.

The advantage of using K-map is that the expression can be simplified to its most simplest form without any algebraic derivation. But this process can only be applied to reduce any canonical SOP or POS expression.

Two Variables K-map

A K-map can be designed to simplify an expression which is canonical SOP or POS in two variables by using the following steps:

(*i*) Use one variable to represent rows and other to represent columns.
(*ii*) Consider two rows corresponding to the row variable and two columns for column variable.
(*iii*) Represent first row or column with complement of the variable followed by non-complement of the variable.

Let two variables be A and B then K-map can be designed as:

	$\overline{B}$	B
$\overline{A}$	$\overline{A}\overline{B}$	$\overline{A}B$
A	$A\overline{B}$	AB

Fig. 1

	0	1
0	0	1
1	2	3

Fig. 2

Two variables K-map for SOP expression

As shown in the Fig. 1 and 2 above, a K-map in two variables is a 2×2 matrix form. Rows or columns can also be represented by 0 or 1 where 0 may be applied corresponding to complement of the variable and 1 corresponding to non-complement variable. The cells, so formed, can be numbered with decimal number equivalent to the binary representation of the cell (i.e. 0 for row number 0 and column number 0, 1 for row number 0 and column number 1 and so on).

This system of formation of K-map is applied only to simplify SOP expression. In order to form a K-map in two variables to simplify POS expression, the complement of the variable is changed with non-complement and vice versa. Binary representation of rows and columns remains same. Hence, cell numbers do not get affected.

	B	$\overline{B}$
A	$A+B$	$A+\overline{B}$
$\overline{A}$	$\overline{A}+B$	$\overline{A}+\overline{B}$

Fig. 3

	0	1
0	0	1
1	2	3

Fig. 4

Two variables K-map for POS expression

Three Variables K-map

Let three variables be A, B and C. To design a K-map in three variables, the following steps can be taken:

(*i*) Use variable A in row and two variables B and C in column.
(*ii*) Use two rows for variable A and four columns for the variables B and C.
(*iii*) Mark the rows or columns by the variables as shown in the Fig. 5 and 6 below:

	$\overline{B}\overline{C}$	$\overline{B}C$	BC	$B\overline{C}$
$\overline{A}$	$\overline{A}\overline{B}\overline{C}$	$\overline{A}\overline{B}C$	$\overline{A}BC$	$\overline{A}B\overline{C}$
A	$A\overline{B}\overline{C}$	$A\overline{B}C$	ABC	$AB\overline{C}$

Fig. 5

	00	01	11	10
0	0	1	3	2
1	4	5	7	6

Fig. 6

Three variables K-map for SOP expression

You can observe the way of marking the columns by the variables BC. The first column starts with B′C′, then complement rotates and goes out from C. Hence, next column is B′C. After further rotation, complement moves out from B also and third column is marked as BC. Finally, complement resumes back on C to represent fourth column as BC′.

As shown in the Fig. 5 and 6 above, a K-map in three variables is a 2 × 4 matrix form. Rows or columns can also be represented by 0 or 1 where 0 may be applied corresponding to complement of the variable and 1 corresponding to non-complement variable. The cells, so formed, can be numbered with decimal number equivalent to the binary representation of the cell (i.e. 0 for row number 0 and column number 00, 1 for row number 0 and column number 01 and so on).

This system of designing a K-map is applied only to simplify SOP expression. In order to form a K-map in three variables to simplify POS expression, the complement of the variable is changed with non-complement and vice versa. Apply + sign in between the column variables representing each column. Binary representation of rows and columns remains same. Hence, cell numbers do not get affected.

	$B+C$	$B+\overline{C}$	$\overline{B}+\overline{C}$	$\overline{B}+C$
A	$A+B+C$	$A+B+\overline{C}$	$A+\overline{B}+\overline{C}$	$A+\overline{B}+C$
$\overline{A}$	$\overline{A}+B+C$	$\overline{A}+B+\overline{C}$	$\overline{A}+\overline{B}+\overline{C}$	$\overline{A}+\overline{B}+C$

Fig. 7

	00	01	11	10
0	0	1	3	2
1	4	5	7	6

Fig. 8

Three variables K-map for SOP expression

Four Variables K-Map

Let four variables used in the expression be A, B, C and D. A K-map can be designed to simplify SOP expression by using the following steps:

(*i*) Use variables A and B to represent rows and C and D to represent columns.
(*ii*) Draw four rows corresponding to variables AB and four columns for CD.
(*iii*) A 4×4 matrix will appear. Mark rows and columns by the variables as shown below:

	$\overline{C}\overline{D}$	$\overline{C}D$	CD	$C\overline{D}$
$\overline{A}\overline{B}$	$\overline{A}\overline{B}\overline{C}\overline{D}$	$\overline{A}\overline{B}\overline{C}D$	$\overline{A}\overline{B}CD$	$\overline{A}\overline{B}C\overline{D}$
$\overline{A}B$	$\overline{A}B\overline{C}\overline{D}$	$\overline{A}B\overline{C}D$	$\overline{A}BCD$	$\overline{A}BC\overline{D}$
AB	$\overline{A}\overline{B}CD$	$AB\overline{C}D$	$ABCD$	$ABC\overline{D}$
$A\overline{B}$	$A\overline{B}\overline{C}\overline{D}$	$A\overline{B}\overline{C}D$	$A\overline{B}CD$	$A\overline{B}C\overline{D}$

Fig. 9

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

Fig. 10

Four variables K-map for SOP expression

In order to design a four variables K-map for POS expression, change complement to non-complement and vice versa and apply + sign in between the variables representing each row and each column.

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

Fig. 11

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

Fig. 12

Four variables K-map for POS expression

Marking Cells

Let us take a SOP expression A'B + AB for our consideration. It is a two variable expression. Hence, K-map can be designed as a matrix form 2×2 as shown below:

	$\overline{B}$	B
$\overline{A}$	0	1 (1)
A	2	1 (3)

Fig. 13

Mark 1's in the cells represented by first variable of each term as row and second variable as column.

For a three variable expression A'B'C' + A'B'C + A'BC + AB'C' + ABC, you can use a K-map having two rows and four columns as shown in the Fig. 14 below:

	$\overline{B}\overline{C}$	$\overline{B}C$	BC	$B\overline{C}$
$\overline{A}$	1 (0)	1 (1)	1 (3)	2
A	1 (4)	5	1 (7)	6

Fig. 14

Here, mark 1's in the cells for each term of the expression representing row with the first variable and column with other two variables. Similarly, you can mark 0's in the cells to reduce POS expression.

Framing Groups

After marking 1's in the cells, you can go for framing groups. A group is a combination of 1's placed in consecutive cells horizontally or vertically. Groups can be formed in the following ways:

(*i*) **Pair:** A set of two one's in consecutive horizontal or vertical cells can be marked under a group termed as a **pair**.

	$\overline{C}\overline{D}$	$\overline{C}D$	$C\cancel{\overline{D}}$	$C\overline{D}$
$\overline{A}\overline{B}$	0	1 (1)	1 (3)	2
$\overline{A}B$	1 (4)	5	7	6
AB	1 (12)	13	1 (15)	1 (14)
$A\overline{B}$	8	9	11	10

Fig. 15

Pair formation

Pair Reduction Scheme

K-Map in Fig. 5 contains three pairs:

Pair I: $M_1 \cdot M_3$

Rows representation of pair I: A′B′

Columns representation of pair I: C′D + CD

Here, C′ and C are available in opposite form. Hence, get cancelled.

Variable obtained from columns is D.

Hence, reduced term is: A′B′D

Pair II: $M_4 . M_{12}$

Rows representation of pair II: A′B + AB

Here, A′ and A are in opposite form. Hence, get cancelled.

Variable obtained from rows is B.

Columns representation of pair II: C′D′

Hence, reduced term is: BC′D′

Pair III: $M_{15} . M_{14}$

Rows representation of pair I: AB

Columns representation of pair I: CD + CD′

Here, D and D′ are available in opposite form. Hence, get cancelled.

Variable obtained from columns is D.

Hence, reduced term is: ABD

Most simplified expression = Sum of the reduced terms = A′B′D + BC′D′ + ABD

(iii) **Quad:** A set of four one's placed together can form a group termed as a **quad**.

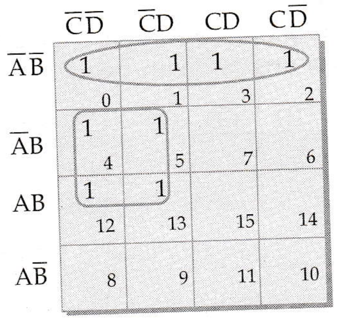


Fig. 16

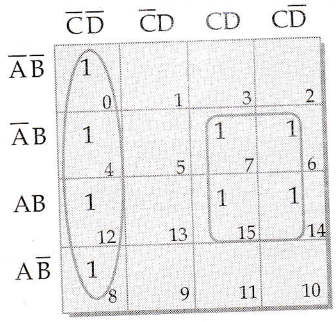


Fig. 17

Quad formation

Quad Reduction Scheme

The Fig. 16 shown above contains two quads:

Quad I: $M_0M_1M_3M_2$

Rows representation of Quad I: A′B′

Columns representation of Quad I: C′D′ + C′D + CD + CD′

Here all the variables are opposite to each other. Hence, get cancelled.

Term obtained: A′B′

Quad II: $M_4M_5M_{12}M_{13}$

Rows representation of Quad II: A′B + AB = B [A′ and A are opposite to each other. Hence, get cancelled.]

Columns representation of Quad II: C′D′ + C′D = C′ [D′ and D are opposite to each other. Hence, get cancelled.]

Term obtained: BC

Most simplified expression: A′B′ + BC

Similarly, you can obtain reduced expression from Fig. 17.

(*iii*) Octet: A set of eight one's placed together can form a group termed as **octet**.

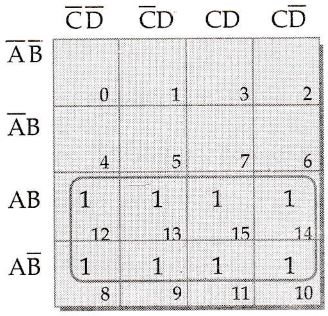


Fig. 18

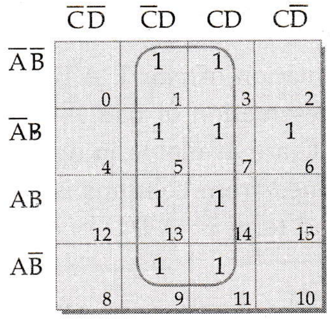


Fig. 19

Octet formation

Octet Reduction Scheme

In the Fig. 18 shown above, only one octet is formed.

Octet I: $M_{12}M_{13}M_{15}M_{14}M_8M_9M_{11}M_{10}$

Rows representation of Octet I: AB + AB′ = A [B and B′ are in opposite form and hence, get cancelled.]

Columns representation of Octet I: C′D′ + C′D + CD + CD′ = 1 [All variables are in opposite form of each other. Hence, get cancelled.]

Hence, reduced expression is A·1 = A

Similarly, you can get reduced expression from Fig. 19.

Group Formation By Folding or Rolling a K-map

Group can also be formed by folding or rolling the map (i.e. considered horizontal or vertical joining of the map edge to edge). Some illustrations are as given below:

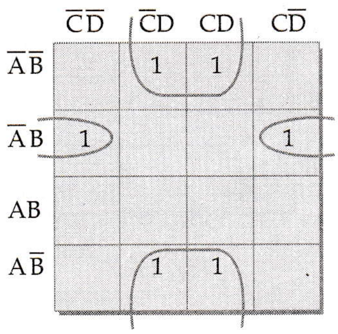


Fig. 20
Formation of pair and quad

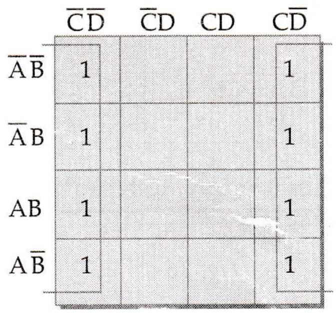


Fig. 21
Formation of octet

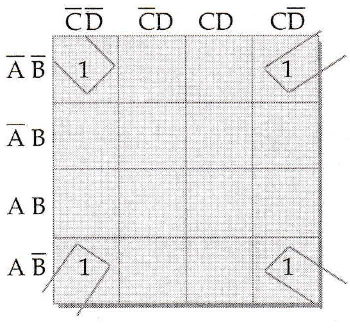


Fig. 22
Formation of quad

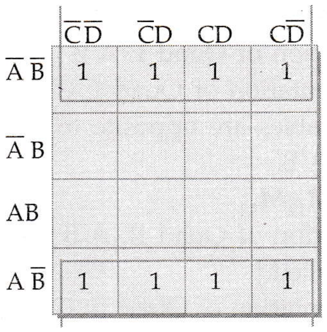


Fig. 23
Formation of octet

Formation of Group by Sharing 1's

Sometimes, it becomes difficult to form a group due to unavailability of 1's consecutively together. In such situation, you can frame a group even by sharing 1's of two different groups. Framing groups by sharing 1's are as shown below:

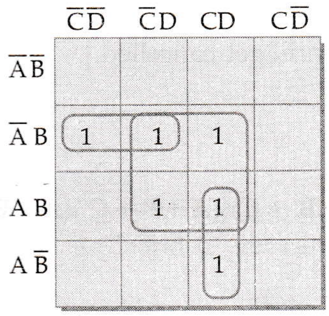


Fig. 24

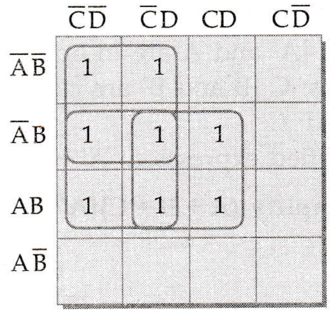


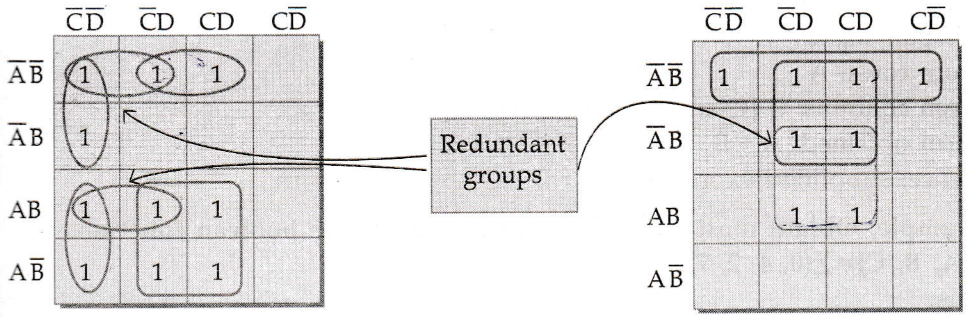
Fig. 25

Sharing of 1's to form a group

Redundant Groups

A group is said to be redundant if all its 1's are shared by other groups. Such group is unwanted and hence, must be eliminated from the K-map. The existence of redundant groups may destruct to obtain expression in the most simplified form.

Some examples to illustrate redundant groups are as shown below:



Formation of redundant pairs *Redundant quad*

Fig. 26

Points to be kept in the mind while grouping:

(i) You must try to accommodate all 1's in the groups.
(ii) You must try to go for higher group first (octet then quad and then pair), if possible.
(iii) Groups can be formed even by sharing 1's.
(iv) You can frame a group by rolling or folding a map.
(v) Avoid redundancy.

Reducing Expressions

Example: Simplify A'B'C' + A'B'C + A'BC + AB'C + A'BC' + ABC.

Ans.

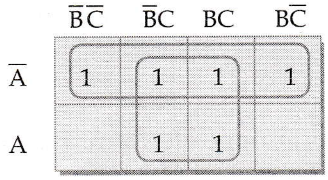


Fig. 27

For quad $M_0M_1M_3M_2$:
From rows: A′
From columns: 1 [Both the variables are in opposite form. Hence, get cancelled.]
Term obtained: A′
For quad $M_1M_3M_5M_7$:
From rows: 1 [A′ and A are in opposite form. Hence, get cancelled.]
From columns: C [B and B′ are cancelled.]
Term obtained: C
Hence, simplified expression: A′ + C

Example: Simplify (A + B + C)(A′ + B + C) + (A′ + B′ + C)((A + B + C′)(A + B′ + C)
Ans.

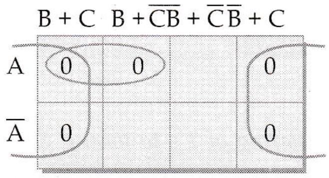


Fig. 28

For quad $M_0M_2M_4M_6$:
From rows: O [A and A′ are in opposite form. Hence, get cancelled.]
From columns: C [B and B′ get cancelled.]
Term obtained: C
For pair M_0M_1:
From rows: A
From columns: B [C and C′ are cancelled.]
Term obtained: A + B
Hence, simplified expression: A + B + C

Example: Obtain most simplified expression for the boolean function:
F(A, B, C) = Σ(0, 4, 2, 7, 5)
Ans.

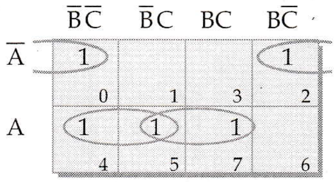


Fig. 29

For pair M_0M_2:
From rows: A′
From columns: C′ [B and B′ get cancelled.]
Term obtained: A′C′
For pair M_4M_5:
From rows: A
From columns: B′ [C and C′ get cancelled.]
Term obtained: AB′
For pair M_5M_7:
From rows: A
From columns: C [B and B′ are in opposite form. Hence, get cancelled.]
Term obtained: AC
Simplified expression: A′C′ + AB′ + AC

Example: Obtain the most simplified expression for the boolean function given below:
$F(A, B, C) = \pi(0, 1, 4, 5, 2, 6)$

Ans.

	$B + C$	$B + \overline{C}$	$\overline{B} + \overline{C}$	$\overline{B} + C$
A	0 [0]	0 [1]	[3]	0 [2]
$\overline{A}$	0 [4]	0 [5]	[7]	0 [6]

Fig. 30

For quad $M_0M_1M_4M_5$:
From rows: 0 [A and A′ get cancelled.]
From columns: B [C and C′ get cancelled.]
Term obtained: B
For quads $M_0M_2M_4M_6$:
From rows: 0 [A and A′ get cancelled.]
From columns: C [B and B′ get cancelled.]
Term obtained: C
Most simplified expression: B + C

Example: Obtain most simplified expression for:
A′BC′D′ + ABC′D′ + A′B′C′D + A′B′CD + A′BCD′ + ABCD′ + AB′C′D + AB′CD

Ans.

	$\overline{C}\overline{D}$	$\overline{C}D$	CD	$C\overline{D}$
$\overline{A}\overline{B}$	[0]	1 [1]	1 [3]	[2]
$\overline{A}B$	1 [4]	[5]	[7]	1 [6]
AB	1 [12]	[13]	[15]	1 [14]
$A\overline{B}$	[8]	1 [9]	1 [11]	[10]

Fig. 31

For quad $M_4M_{12}M_6M_{14}$:
Form rows: B [A and A′ get cancelled]
From columns: D′ [C′ and C get cancelled]
Term obtained: BD′
For quads $M_1M_3M_9M_{10}$:
From rows: B′ [A and A′ get cancelled]:
From columns: D [C and C′ get cancelled]
Term obtained: B′D
Most simplified expression: BD′ + B′D

Example: Obtain most simplified expression for:
(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′) (A + B′ + C′ + D) (A′ + B′ + C′ + D)

Ans.

	$C+D$	$C+\overline{D}$	$\overline{C}+\overline{D}$	$\overline{C}+D$
$A + B$	[0]	0 [1]	[3]	[2]
$A + \overline{B}$	0 [4]	0 [5]	0 [7]	0 [6]
$\overline{A} + \overline{B}$	0 [12]	0 [13]	[15]	0 [14]
$\overline{A} + B$	[8]	0 [9]	[11]	[10]

Fig. 32

For quad $M_4M_{12}M_6M_{14}$:
Form rows: B′ [A and A′ get cancelled.]
From columns: D [C and C′ get cancelled.]
Term obtained: B′ + D
For quads $M_1M_5M_{13}M_9$:
From rows: 0 [A and A′, B and B′ both get cancelled.]
Form columns: C + D′
Term obtained: C + D′
For quads $M_4M_5M_7M_6$:
From rows: A + B′
From columns: 0 [C and C′, D and D′ get cancelled.]
Term obtained: A + B′
Hence, most simplified expression: (B′ + D)(C + D′)(A + B′)

Solved Exercises

1. **Simplify the following boolean expression using laws of boolean algebra.** *[ISC 2010]*
   $a \cdot b \cdot (b \cdot c + a \cdot b)$

Ans.

$$ab(bc + ab)$$
$= ab \cdot bc + ab \cdot ab$ [Taking ab common]
$= ab \cdot bc + ab$ [bc + 1 = 1]
$= ab(bc + 1)$
$= ab \cdot 1$
[bc + 1 = 1]
$= ab \cdot 1$ [ab·1 = ab]
$= ab$

2. **State the Demorgan's laws. Verify one of them using truth table.** *[ISC 2010]*

Ans.

Demorgan's laws state that:
(*i*) $(a + b)' = a' \cdot b'$
(*ii*) $(a \cdot b)' = a' + b'$
Verification of first law:

a	b	(a + b)′	a′·b′
0	0	1	1
0	1	0	0
1	0	0	0
1	1	0	0

3rd and 4th columns have same entries.
Hence proved.

3. **Convert the following SOP expression into its corresponding product of sum form:** *[ISC 2010]*
   $F(O, V, W) = O' \cdot V' \cdot W' + O' \cdot V' \cdot W + O' \cdot V \cdot W + O \cdot V' \cdot W$

Ans.

F(O, V, W) = O′V′W′ + O′V′W + O′VW + OV′W
Binary values of the max terms.
000 + 001 + 011 + 101
Min term designation = $M_0 + M_1 + M_3 + M_5$
Max term designation other then Min terms:
$= M_2 + M_4 + M_6 + M_7$
Binary values of the max terms:
010 + 100 + 110 + 111

Hence, max term expression

$F(O, V, W) = (O + V' + W)\ (O' + V + W) + (O' + V' + W)\ (O' + V' + W')$

4. Find the complement of

$F(a, b, c, d) = [a + \{(b + c) \cdot (b' + d')\}]$ *[ISC 2010]*

Ans.

Complement of the expression:

$[a + \{(b + c) \cdot (b' + d')\}]'$

$= [a' \cdot \{(b + c) \cdot (b' + d')\}']$ [Using Demorgan's Theorem]

$= [a' \cdot \{(b + c)' + (b' + d')'\}]$ [Using Demorgan's Theorem]

$= [a' \cdot \{b'c' + (b \cdot d)\}]$ [Using Demorgan's Theorem]

$= [a' \cdot \{b'c' + b \cdot d\}]$

$= a'b'c' + a'bd$

5. What do you mean by min terms and max terms? State the relation between them. State the two compliment properties of boolean algebra and verify one of them using truth table. *[ISC 2008]*

Ans.

Min term is the product of the boolean variables in which variables may be in complement and non-complement form. Max term is the sum of the variables comprising a term of the expression in which the variables may be in complement as well as in non-complement form. Both the terms are complement to each other.

Complement property:

- **$A + A' = 1$**
- **$A \cdot A' = 0$**

A	A′	A + A′
0	1	1
1	0	1

The truth table shown above verifies the first property in which the outcome is always 1 for any value of A and A′.

6. Reduce the following to its simplest from using laws of boolean algebra. At each step state clearly the law used for simplification $C \cdot D' + A + A + C' \cdot D' + A \cdot B$.

Ans.

$C \cdot D' + A + A + C' \cdot D' + A \cdot B$

$= C \cdot D' + A + C' \cdot D' + A \cdot B$ [Idempotent law]

$= C \cdot D' + C' \cdot D' + A \cdot AB$

$= C \cdot D' + C' \cdot D' + 1$ [Absorption law]

$= C \cdot D' + 1$ [Variable plus 1 = 1]

$= 1$ [$C \cdot D' + 1 = 1$]

7. Simplify the following boolean expressing stating clearly the laws used for the simplification at each step: $x \cdot y + x,\ z + x \cdot y \cdot z$ *[ISC 2008]*

Ans. $x \cdot y + x,\ z + x \cdot y \cdot z$

$= x \cdot z + x \cdot y(1 + z)$ [Taking $x \cdot y$ common]

$= x \cdot z + xy$ [$1 + z = 1$]

$= xy + xz$

8. Find the complement of:

$x(y' \cdot z' + y \cdot z)$ *[ISC 2008]*

Ans.

Complement of $x \cdot (y' \cdot z' + y \cdot z)$

$= [x(y' \cdot z' + y \cdot z)]'$

$= [x' + (y'z' + y \cdot z)']$

$= [x' + \{(y'z')' \cdot (yz)'\}]$
$= [x' + (y + z)\ (y' + z')]$
$= (x' + y + z) . (x' + y' + z')$

9. Given the following function F (a, b, c) = ∑(0. 1.2.4.5.6). Obtain most simplified expression by using K-map.

Ans.

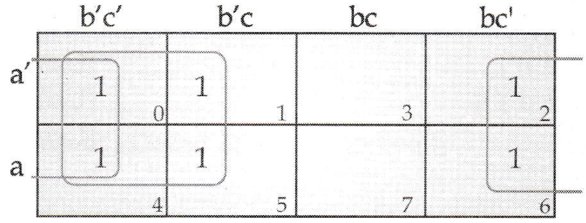


Fig. 1

From quad (0, 1, 4, 5):
Rows representing the quad: $a + a' = 1$
Columns representing the quad: $b'c' + b'c = b$
[c and c' are in opposite form hence, get cancelled]
Term obtained: $b \cdot 1 = b$
From quad 0, 2, 4, 6:
Rows representing the quad: $a + a' = 1$
Columns representing the quad: $b'c' + bc' = c'$
[b and b' are in opposite form hence, get cancelled]
Term obtained $= c' \cdot 1 = c'$
Hence, most simplified expression $= b + c'$

10. For the given boolean function X(a, b, c, d) = ∑(0, 1, 2, 3, 5, 7, 13, 15, 8, 9, 10, 11) obtain the most simplifies expression by using K-map.

Ans.

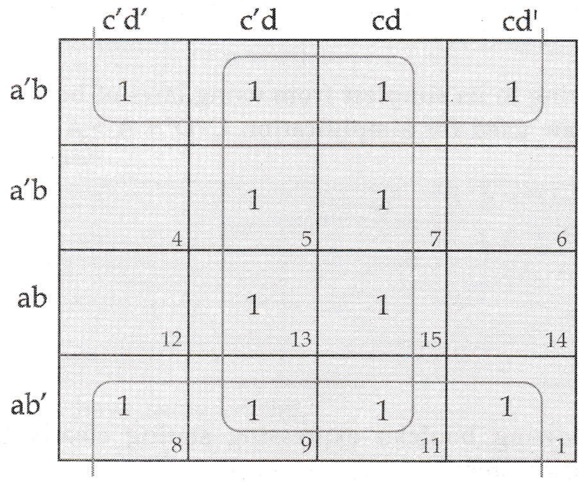


Fig. 2

From octet (1, 3, 5, 7, 13, 15, 9, 11):
Rows representing the octet: All the rows with variables in opposite form of each other. Hence, all the variables get cancelled.
Columns representation of octet: $= c'd + cd = d$ [c and c' get cancelled]
Term obtained: $d \cdot 1 = d$
From octet (0, 1, 3, 2, 8, 9, 11, 10):
Rows representation of octet: $a'b' + ab' = b'$ [a and a' get cancelled]
Columns representation of octet: All columns having variables in opposite form of each other. Hence, all the variables get cancelled.

Term obtained: b′
Most simplified expression: b′ + d

11. For the given boolean function F (X, Y, Z) = π (0·1·2·3·4·6) obtain the most simplified expression by using K-map.

Ans.

	Y + Z	Y′ + Z′	Y′ + Z′	Y′ + Z
X	0 (0)	0 (1)	0 (3)	0 (2)
X′	0 (4)	(5)	(7)	0 (6)

Fig. 3

From quad (0, 1, 2, 3):
Rows representation of the quad: X
Columns representation of the quad: All columns having variables in opposite form of each other. Hence, get cancelled.

Term obtained = (X + 0): X
From quad (0, 2, 4, 6):
Rows representation of quad: $X \cdot X' = 0$
Columns representation of the quad: (Y + Z) (Y′ + Z) = Z
[Y and Y′ get cancelled due to opposite forms]
Term obtained: (0 + Z) = Z
Most simplified expression: $X \cdot Z$

12. Obtain the most simplified boolean expression by using K-map for the following function:
F(X, Y, Z, W) = π(5, 7, 12, 13, 14, 15, 8, 10)

Ans.

	Z + W	Z + W′	Z′ + W′	Z′ + W
X+Y	(0)	(1)	(3)	(2)
X+Y′	(4)	0 (5)	0 (7)	(6)
X′+Y′	0 (12)	0 (13)	0 (15)	0 (14)
X′+Y	0 (8)	(9)	(11)	0 (10)

Fig. 4

From quad (5, 7, 13, 15):
Rows representation of quad: ((X + Y′)(X′ + Y′) = Y′
[X and X′ get cancelled as they are in opposite forms]
Columns representation of quad: (Z + W′) (Z′ + W) = W′
[Z and Z′ get cancelled as they are in opposite forms]
Term obtained: W′ + Y′
From quad (12, 14, 8, 10):
Rows representation of the quad: (X′ + Y′) (X′ + Y) = X′
[Y and Y′′ get cancelled as they are in opposite forms]

Term obtained: b′
Most simplified expression: b′ + d

11. For the given boolean function F (X, Y, Z) = π (0·1·2·3·4·6) obtain the most simplified expression by using K-map.

Ans.

	Y + Z	Y′ + Z′	Y′ + Z′	Y′ + Z
X	0 (0)	0 (1)	0 (3)	0 (2)
X′	0 (4)	(5)	(7)	0 (6)

Fig. 3

From quad (0, 1, 2, 3):
Rows representation of the quad: X
Columns representation of the quad: All columns having variables in opposite form of each other. Hence, get cancelled.

Term obtained = (X + 0): X
From quad (0, 2, 4, 6):
Rows representation of quad: $X \cdot X' = 0$
Columns representation of the quad: (Y + Z) (Y′ + Z) = Z
[Y and Y′ get cancelled due to opposite forms]

Term obtained: (0 + Z) = Z
Most simplified expression: $X \cdot Z$

12. Obtain the most simplified boolean expression by using K-map for the following function:
F(X, Y, Z, W) = π(5, 7, 12, 13, 14, 15, 8, 10)

Ans.

	Z + W	Z + W′	Z′ + W′	Z′ + W
X+Y	(0)	(1)	(3)	(2)
X+Y′	(4)	0 (5)	0 (7)	(6)
X′+Y′	0 (12)	0 (13)	0 (15)	0 (14)
X′+Y	0 (8)	(9)	(11)	0 (10)

Fig. 4

From quad (5, 7, 13, 15):
Rows representation of quad: ((X + Y′)(X′ + Y′) = Y′
[X and X′ get cancelled as they are in opposite forms]
Columns representation of quad: (Z + W′) (Z′ + W) = W′
[Z and Z′ get cancelled as they are in opposite forms]
Term obtained: W′ + Y′
From quad (12, 14, 8, 10):
Rows representation of the quad: (X′ + Y′) (X′ + Y) = X′
[Y and Y′′ get cancelled as they are in opposite forms]

Column representation of quad: (Z + W) (Z′ + W) = W

[Z and Z′ get cancelled as they are in opposite forms]

Term obtained = X′ + W

Most simplified expression: (W′ + Y′) (X′ + W).

13. On a railway station, there are three tracks and an alarm signal. The alarm signal sets high if at least two tracks have trains on them. Using inputs A, B and C that represent railway tracks and output D as the alarm clock in such a way that

A = 1,	**Means first track has train on it.**
A = 0,	**Means no train on first track.**
B = 1,	**Means second track has train on it.**
B = 0,	**Means no train on the second track**
C = 1,	**Means third track has train on it.**
C = 0,	**Means no train on third track.**
D = 1,	**Means alarm signal sets high.**
C = 0,	**Means alarm signal sets low.**

Perform the following operations:
- **Draw a truth table for the aforementioned situation.**
- **Write SOP expression from the truth table.**
- **Simplify the expression by using K-map.**

Ans.

Truth table to realize the given situation:

A	B	C	D	Min Terms
0	0	0	0	
0	0	1	0	
0	1	0	0	
0	1	1	1	A′.B.C
1	0	0	0	
1	0	1	1	A.B′.C
1	1	0	1	A.B.C′
1	1	1	1	A.B.C

D = A′BC + AB′C + ABC′ + ABC

K-map to reduce the expression:

	B′C′	B′C	BC	BC′
A′	0	1	1 3	1 2
A	4	1 5	1 7	1 6

Fig. 5

From pair (3, 7):

Rows representation of pair: A′ + A = 1

Columns representation of pair: BC

Term obtained = BC

From pair (5, 7):

Rows representation of pair: A

Columns representation of pair: B′C + BC = C

[B and B′ get cancelled being in opposite forms]

Term obtained: AC

From pair (7, 6):
Rows representation of pair: A
Columns representation of pair: $BC' + BC = B$ [C and C′ get cancelled being in opposite forms]

Term obtained: AB
Hence, most simplified expression $= AB + BC + AC$.

14. Convert the following product of sums form into its corresponding sum of products form: $F(x, y, z) = \prod(2, 4, 6, 7)$. *[ISC 2008]*

Ans.

Given POS expression is:
$F(x, y, z) = \prod(2, 4, 6, 7)$
Max term designation $= M_2, M_4, M_6, M_7$
Binary equivalent value representing max terms = (010, 100, 110, 111)
Binary equivalent value representing min terms = (101, 011, 001, 000)
Min term designation $= M_0, M_1, M_3, M_5$
Hence, sum of product form $F(x, y, z) = \sum(0, 1, 3, 5)$

15. Convert $A \cdot B + B \cdot C'$ to is canonical SOP form using boolean algebra. *[ISC 2008]*

Ans.

Expression is: $A.B + B.C'$
$= AB(C + C') + BC'(A + A')$
$= ABC + ABC' + ABC' + A'BC'$
$= ABC + ABC' + A'BC'$.

16. Verify if $X'.Y.Z + X.Y'.Z + X.Y.Z' + X.Y.Z = X.Y + Y.Z + Z.X$
At each step state clearly the law used for simplification. *[ISC 2007]*

Ans.

$X' \cdot Y \cdot Z + X \cdot Y' \cdot Z + X \cdot Y \cdot Z' + X \cdot Y \cdot Z = X \cdot Y + Y \cdot Z + Z \cdot X$
L.H.S:
$X' \cdot Y \cdot Z + X \cdot Y' \cdot Z + X \cdot Y \cdot Z' + X \cdot Y \cdot Z$
$= X'YZ + XY'Z + XY(Z + Z')$ [Taking XY common]
$= X'YZ + XY'Z + XY$ $[Z + Z' = 1]$
$= X'YZ + X(Y'Z + Y)$
$= X'YZ + X[(Y' + Y)(Z + Y)]$ [By using distributive law]
$= X'YZ + X[1(Z + Y)]$ $[Y' + Y = 1]$
$= X'YZ + XZ + XY$
$= Z(X'Y + X) + XY$ [Taking Z common]
$= Z[(X' + X)\ (X + Y)] + XY$ [By using distributive law]
$= Z(X + Y) + XY$ $[X + X' = 1]$
$= XZ + YZ + XY$
$= XY + YZ + XZ$
Hence proved.

17. State whether the following is true or false:
$(x + y) \cdot (y + z) \cdot (x + z) = x \cdot (y + z) + y \cdot z$ *[ISC 2007]*

Ans.

L.H.S.: $(x + y) \cdot (y + z) \cdot (x + z)$
$= (xy + xz + y + yz)(x + z)$
$= xy + xyz + xz + xz + xy + yz + xyz + yz$
$= xy(1 + z) + xz + yz + xyz$
$= xy + xz + yz(1 + z)$
$= xy + xz + yz$
$= x(y + z) + yz$
Hence, the given statement is true.

18. Prove that $[(p' + q) \cdot (q' + r)]' + (p' + r) = 1$ *[ISC 2006]*

Ans.

L.H.S.:

$[(p' + q) \cdot (q' + r)]' + (p' + r)$

$= [(p' + q)' + (q' + r)'] + (p' + r)$ [By using Demorgan's theorem]

$= [(p.q' + q.r')] + (p' + r)$ [By using Demorgan's theorem]

$= pq' + p' + qr' + r$

$= (p + p')(p' + q') + (q + r)(r' + r)$

$= p' + q' + q + r$ [$p + p' = 1$ and $r + r' = 1$]

$= p' + 1 + r$ [1 added to any value results in 1]

$= 1$

Hence proved.

19. Find the complement of $F(m, n, o) = m' \cdot n \cdot o' + m' \cdot n' \cdot o$ *[ISC 2006]*

Ans.

Complement of $= m' \cdot n \cdot o' + m' \cdot n' \cdot o = (m' \cdot n \cdot o' + m' \cdot n' \cdot o)'$

$= (m' \cdot n \cdot o')' \cdot (m' \cdot n' \cdot o)'$

$= (m + n' + o) \cdot (m + n + o')$

20. Using the truth table, prove that $(A + B)' + (A + B')' = A'$ *[ISC 2000, 2006]*

Truth table of: $(A + B)' + (A + B')' = A'$

A	B	(A + B)′	(A + B′)′	(A + B)′ + (A + B′)′
0	0	1	0	1
0	1	0	1	1
1	0	0	0	0
1	1	0	0	0

Entries in 1st and 5th columns show opposite values. Hence proved.

21. Verify the following law by using the truth table:
(A XNOR B) XNOR C = A XNOR (B XNOR C) *[ISC 2003]*

Ans.

A	B	C	A XNOR B	(A XNOR B) XNOR C	B XNOR C	A XNOR (B XNOR C)
0	0	0	1	0	1	0
0	0	1	1	1	0	1
0	1	0	0	1	0	1
0	1	1	0	0	1	0
1	0	0	0	1	1	1
1	0	1	0	0	0	0
1	1	0	1	0	0	0
1	1	1	1	1	1	1

5th and 7th columns have same entries. Hence proved.