# Boolean Algebra \& Minimization-2 Study Notes for GATE \& Computer Science Engineering Exams 

Boolean algebra and minimization techniques are crucial aspects of digital logic design, enabling engineers to represent complex logic functions efficiently. These concepts find extensive application in various fields.

Here we have provided the Boolean Algebra \& Minimization-2 Study Notes that covers canonical expressions, POS (Product of Sum) form, SOP (Sum of Product) form, simplified expressions, and canonical form.

## Boolean Algebra \& Minimization

Boolean algebra is a branch of mathematical logic or algebra in which the value of the variables is binary, i.e. it can be either 0 or 1 , also denoted as false and true respectively. Boolean algebra is used to simplify and analyze digital logical circuits. Boolean algebra was introduced by George Boole in 1854.

Any digital circuit can be implemented and represented using Boolean functions, but the issue is that literals used in original Boolean function are usually high, and if that function is implemented using logic gates it will increase the complexity and cost of the system. Boolean functions are directly related to algebraic expression's complexity through which we were implementing the function.

To decrease the complexity, we need to have a simplified version of algebraic Boolean expression. So, the process of minimizing and simplifying an algebraic expression is known as minimization. Minimization is important to cut down cost and complexity of the circuit.

## Ways of representing the functions

## - Canonical Expressions

- POS form
- SOP form
- Simplified Expressions


## Canonical Form

Before moving towards actual expression, let us understand the basic terms

## Minterm

It is the product term which contains all the variables in the given function, and for that combination, the function output must be 1 .

- It is represented by m .
- A minterm is also called a canonical product term.
- A minterm is a product term, but a product term may or may not be a minterm.
- The SOP Expression is written in the form of minterms.


## Maxterm

- Each sum term is known as the maximum term that contains all of the variables used in the function.
- It is a sum term, containing all the variables in the given function either complementary or un-complementary form, for that combination, function output must be 0 .
- A maxterm is a sum term of all variables in which each variable is either in complemented form or in uncomplemented form.
- A maxterm is also called a canonical sum term.
- A maxterm is a sum term, but a sum term may or may not be a maxterm.


## Canonical SOP Expression

- SOP stands for sum of product terms.
- To obtain the SOP expression, Refer to the combination of truth table whose value is 1
- If a variable $x$ is 1 , represent it as $x$
- if variable $x$ is 0 , represent it as $x^{\prime}$.
- In this manner, obtain All the minterm for the given function.
- Add all the minterms to get the SOP expression for the given function.


## Canonical POS Expression

POS stands for product of sum terms.

To obtain the POS expression, Refer to those combinations of the truth table whose value is 0 .

- If a variable $X$ is 1 , represent it as $x^{\prime}$.
- if a variable $X$ is 0 , represent it as $x$,
- Represent all the variables as sum terms
- In this manner obtain all the maxterm of the given function
- Finally multiply all the product terms obtained to get the POS expression.

The following are examples of product term, minterm, sum term, and maxterm for a function of three variables $a, b$, and $c$ :
product terms: a, ac, b'c, abc, a'bc, a'b'c', ...
minterms: ab'c, abc, a'b'c, a'b'c', ...
sum terms: $a,(a+b),(b+c),\left(a^{\prime}+b\right),\left(a^{\prime}+b^{\prime}\right), \ldots$
maxterms: $(a+b+c),\left(a+b^{\prime}+c\right),\left(a^{\prime}+b^{\prime}+c^{\prime}\right), \ldots$

## Representations of Minterm and Maxterm

| $x$ | $y$ | $z$ | Minterm | Minterm <br> Notation | Maxterm | Maxterm <br> Notation |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 0 | 0 | 0 | $x^{\prime} y^{\prime} z^{\prime}$ | $m 0$ | $x+y+z$ | $M 0$ |
| 0 | 0 | 1 | $x^{\prime} y^{\prime} z$ | $m 1$ | $x+y+z^{\prime}$ | $M 1$ |
| 0 | 1 | 0 | $x^{\prime} y z^{\prime}$ | $m 2$ | $x+y^{\prime}+z$ | $M 2$ |
| 0 | 1 | 1 | $x^{\prime} y z$ | $m 3$ | $x+y^{\prime}+z^{\prime}$ | $M 3$ |
| 1 | 0 | 0 | $x y^{\prime} z^{\prime}$ | $m 4$ | $x^{\prime}+y+z$ | $M 4$ |
| 1 | 0 | 1 | $x y^{\prime} z$ | $m 5$ | $x^{\prime}+y+z^{\prime}$ | $M 5$ |
| 1 | 1 | 0 | $x y z^{\prime}$ | $m 6$ | $x^{\prime}+y^{\prime}+z$ | $M 6$ |
| 1 | 1 | 1 | $x y z$ | $m 7$ | $x^{\prime}+y^{\prime}+z^{\prime}$ | $M 7$ |

Note: With' n' variables maximum possible minimum and maximum terms $=2 n$

With' n' variables maximum possible logic expression $=2^{2^{n}}$

## SOP (Sum of Product)

A sum of product expression is two or more AND functions or functions together.

SOP expression is used when output becomes logic 1.

Example:
$\mathrm{ABC}+\overline{\mathrm{ABC}}+\mathrm{AB} \overline{\mathrm{C}}$
three minterms are there in the expression

## POS (Product of Sum)

It is the AND function of two or more OR functions.
POS expression is used when output is logic ' 0 '.

Example:

$$
(\mathrm{A}+\mathrm{B}+\mathrm{C}) \cdot(\mathrm{A}+\overline{\mathrm{B}}+\mathrm{C}) \cdot(\overline{\mathrm{A}}+\overline{\mathrm{B}}+\overline{\mathrm{C}})
$$

Three maxterms are there in the expression

Example: Sum of Product and Product Of Sum Equivalences for function F and Its Inverse Function F'.


Duality Theorem: Duality Theorem used to change positive logic into negative logic and vice-versa, we make use of dual function.

$$
\mathrm{A} \cdot \overline{\mathrm{~B}} \cdot \mathrm{C}+\overline{\mathrm{A}} \cdot \mathrm{~B} \cdot \mathrm{C}+\mathrm{AB} \overline{\mathrm{C}} \stackrel{\text { dual }}{\longleftrightarrow}(\mathrm{A}+\overline{\mathrm{B}}+\mathrm{C}) \cdot(\overline{\mathrm{A}}+\mathrm{B}+\mathrm{C}) \cdot(\mathrm{A}+\mathrm{B}+\overline{\mathrm{C}})
$$

Change each AND sign by OR sign and vice versa ( $\leftrightarrow+$ )
Complement any 0 or I appearing in expression.

Keep variable as it is.
Example:

# $\mathrm{A} \cdot \overline{\mathrm{B}} \cdot \mathrm{C}+\overline{\mathrm{A}} \cdot \mathrm{B} \cdot \mathrm{C}+\mathrm{AB} \overline{\mathrm{C}} \xrightarrow{\text { dual }}(\mathrm{A}+\overline{\mathrm{B}}+\mathrm{C}) \cdot(\overline{\mathrm{A}}+\mathrm{B}+\mathrm{C}) \cdot(\mathrm{A}+\mathrm{B}+\overline{\mathrm{C}})$ 

## Minimization of Boolean Expressions

The following two approaches can be used for simplification of a Boolean expression:

- Algebraic method (using Boolean algebra rules)
- Karnaugh map method


## Minimization using boolean algebra properties

All the properties are used to minimize the obtained expression, but it is not always possible to minimize the expression completely just by using the boolean algebra properties. There are the chances that expression cant be reduced further but it is not in minimal form, using boolean algebra properties. Such a form of expression is known as Irreducible form/ irredundant expression.

But there are some other ways to reduce an expression completely, by using K- map.

## Representation of K-map

With n-variable Karnaugh-map, there are 2 n cells.

## Example:

2 -variable K Map:


## Two variable $K$-map with cell number

3 -variable K Map:

| Here $n=3$, number of cells $=2^{3}=8$ |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: |
|  | 00 | 01 | 11 | 10 |
| 0 | 0 | 1 | 3 | 2 |
| 1 | 4 | 5 | 7 | 6 |

Three variable $K$-map with cell number

4 -variable K Map:

| $A B \underbrace{C D}_{0}$ | Here $n=4$, number of cells $=2^{n}={ }^{n} 16$ |  |  |  |
| :---: | :---: | :---: | :---: | :---: |
|  | 00 | 01 | 11 | 10 |
| 00 | 0 | 1 | 3 | 2 |
| 01 | 4 | 5 | 7 | 6 |
| 11 | 12 | 13 | 15 | 14 |
| 10 | 8 | 9 | 11 | 10 |

Four variaive $K$-map with cell number

NOTE: Once the you fill Karnaugh map with 1s, 0s and Xs as specified the remaining task is to group adjacent terms of the same state (generally 1 ) in groups of $2^{\wedge} n$, where $n=$ any rational number i.e. $1,2,4,8,16$, and so on. The bigger the group of similar adjacent terms the simpler the final expression. It is also possible for groups to overlap. This happens to achieve a larger group size, hence we get a more simplified final expression.

## Minimization Procedure of Boolean Expression using K-map

- Construct a K-map.
- Find all groups of horizontal or vertical adjacent cells that contain 1.
- Each group must be either rectangular or square with 1, 2, 4, 8, or 16 cells.
- Each group should be as large as possible.
- Each cell with 1 on the K-map must be covered at least once. The same cell can be included in several groups if necessary.
- Select the least number of groups so as to cover all the 1's.
- Adjacency applies to both vertical and horizontal borders.
- Translate each group into a product term. (Any variable whose value changes from cell to cell drops out from the term)
- Sum all the product terms.

Note: Don't care conditions can be used to provide more simplified version of a Boolean Expression.

EXAMPLE: Simplify the following equation using a K-map (Karnaugh-map):

$$
\begin{aligned}
& \mathrm{X}=\overline{\mathrm{A} B C \bar{D}+\overline{\mathrm{A}} \overline{\mathrm{BC}} \overline{\mathrm{D}}+\overline{\mathrm{A}} \overline{\mathrm{~B} C \bar{D}}+\overline{\mathrm{A}} \overline{\mathrm{~B}} \overline{\mathrm{C}} \overline{\mathrm{D}}+\mathrm{A} \overline{\mathrm{~B} C \overline{\mathrm{D}}+}} \begin{array}{l}
\mathrm{m}_{6} \quad \mathrm{~m}_{4} \quad \mathrm{~m}_{2} \quad \mathrm{~m}_{0} \\
\mathrm{~A} \overline{\mathrm{~B}} \overline{\mathrm{C}} \overline{\mathrm{D}}+\overline{\mathrm{A}} \mathrm{~B} \overline{\mathrm{C}} \mathrm{D}+\overline{\mathrm{A}} \overline{\mathrm{~B}} \overline{\mathrm{C}}+\overline{\mathrm{A}} \mathrm{BCD}+\overline{\mathrm{A}} \overline{\mathrm{~B} C D}+\mathrm{ABCD}
\end{array}
\end{aligned}
$$

$$
\mathrm{m}_{8} \quad \mathrm{~m}_{5} \quad \mathrm{~m}_{1} \quad \mathrm{~m}_{7} \quad \mathrm{~m}_{3} \quad \mathrm{~m}_{15}
$$

SOLUTION: Draw the K-map and minimize.


Prime Implicants: It is the smallest possible product term on the given function, from which elimination of any of the literal is not possible.

Essential Prime Implicants:

- It should be a prime implicant.
- It should cover at least one minterm which is not covered by any one of the prime implicants.

Thanks!

BYJU'S Exam Prep - The Most Comprehensive App!

