BOOLEAN ALGEBRA SIMPLIFICATION

**Minimization using Algebraic Manipulation –**

This method is the simplest of all methods used for minimization. It is suitable for medium sized expressions involving 4 or 5 variables. Algebraic manipulation is a manual method, hence it is prone to human error.  
Common Laws used in algebraic manipulation :

Simplify: C + (BC)’:

|  |  |
| --- | --- |
| Expression | Rule(s) Used |
| C + (BC)’ | Original Expression |
|  |  |
| C + (B’ + C’) | DeMorgan's Law. |
| (C + C’) + B’ | Commutative, Associative Laws. |
| 1 + B’ | Complement Law. |
| 1 | Identity Law. |

Simplify: (AB)’(A’ + B)(B’ + B):

|  |  |
| --- | --- |
| Expression | Rule(s) Used |
| (AB)’(A’ + B)(B’ + B) | Original Expression |
|  |  |
| (AB)’(A‘B’+ A’B+BB’+BB) | Complement law, Identity law. |
| (A ‘+ B’)(A’(B’+B) +0+ B)  (A ‘+ B’)(A’.1 +0+ B)  (A ‘+ B’)(A’+ B)  AA’+AB+A’B’+BB’  AB+A’B’ |  |
| Simplify: (A + C)(AD + AD’) + AC + C:   |  |  | | --- | --- | | Expression | Rule(s) Used | | (A + C)(AD + AD’) + AC + C |  | |  |  | | (A + C)A(D + D’) + AC + C |  | | (A + C)A + AC + C  AA+AC+AC+C  A+AC+C  A(1+C)+C |  | | A+C |  | |  |  | |  |  | |  |  | |  |  | |  |
|  |  |

Simplify: A’(A + B) + (B + AA)(A + B’):

|  |  |
| --- | --- |
| Expression |  |
| A’(A + B) + (B + AA)(A + B’) |  |
| A’(A + B) + (B + A)(A + B’)  A’A+A’B+AB+BB’+AA+AB’  A’B+AB+A+AB’  A(1+B+B’)+A’B  A+A’B  (A+A’)(A+B)  A+B |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Boolean functions

Boolean Functions

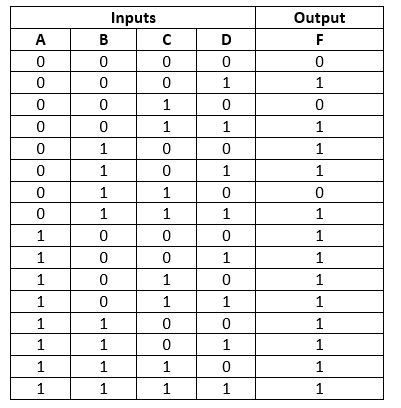
The binary variables and logic operations are used in Boolean algebra. The algebraic expression is known as **Boolean Expression**, is used to describe the **Boolean Function**. The Boolean expression consists of the constant value 1 and 0, logical operation symbols, and binary variables.

**Example 1: F=xy' z+p**

We defined the Boolean function **F=xy' z+p** in terms of four binary variables x, y, z, and p. This function will be equal to 1 when x=1, y=0, z=1 or z=1.

**Example: F(A,B,C,D)=A+BC'+D**

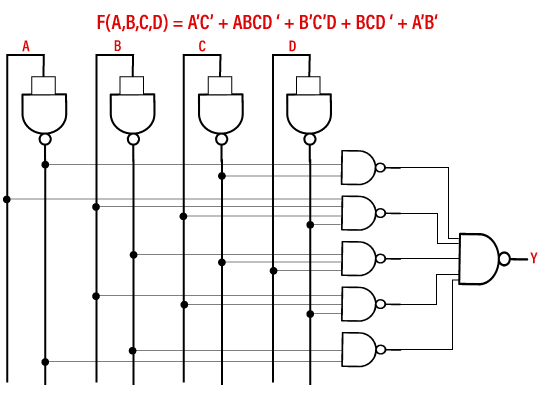
The output will be high when A=1 or BC'=1 or D=1 or all are set to 1. The truth table of the above example is given below. The 2n is the number of rows in the truth table. The n defines the number of input variables. So the possible input combinations are 23=8.



### **NAND gates realization**

Apart from the K-map, we can also use the NAND gate for simplifying the Boolean functions. Let's see an example:

Example 1: F(A,B,C,D)=A' C'+ABCD'+B' C' D+BCD'+A'B'



# Minterm and Maxterm

There are two ways in which we can put the Boolean function. These ways are the minterm canonical form and maxterm canonical form.

## **Literal**

A Literal signifies the Boolean variables including their complements. Such as B is a boolean variable and its complements are ~B or B', which are the literals.

## **Minterm**

The product of all literals, either with complement or without complement, is known as **minterm**.

**Example**

The minterm for the Boolean variables A and B is:

1. A.B'
2. A'.B

## **Minterm from values**

Using variable values, we can write the minterms as:

1. If the variable value is 1, we will take the variable without its complement.
2. If the variable value is 0, take its complement.

**Example**

Let's assume that we have three Boolean variables A, B, and C having values

A=1  
B=0  
C=0

Now, we will take the complement of the variables B and C because these values are 0 and will take A without complement. So, the minterm will be:

Minterm=A.B'C'

Let's take another example in which we have two variables B and C having the value

B = 0  
C = 1

Minterm=B'C

## **Maxterm**

The sum of all literals, either with complement or without complement, is known as **maxterm**.

**Example:**

The maxterm for the Boolean variables A and B will be:

1. A+B
2. A+~B
3. ~A+B

We know that the complement variables ~A and ~B can be written as A' and B' respectively. So, the above maxterm can be written as

1. A+B'
2. A'+B

## **Maxterm from values**

Using the given variable values, we can write the maxterm as:

1. If the variable value is 1, then we will take the variable without a complement.
2. If the variable value is 0, take the complement of the variable.

**Example**

Let's assume that we have three Boolean variables A, B., and C having values

A=1  
B=0  
C=0

Now, we will take the complement of the variables B and C because these values are 0 and will take A without complement. So, the maxterm will be:

Maxterm=A+B'+C'

Let's take another example in which we have two variables B and C having the value

B = 0  
C = 1

Maxterm=B'+C

# Sum of product(SOP)

A canonical sum of products is a boolean expression that entirely consists of minterms. The Boolean function F is defined on two variables X and Y. The X and Y are the inputs of the boolean function F whose output is true when any one of the inputs is set to true. The truth table for Boolean expression F is as follows:

|  |  |  |
| --- | --- | --- |
| **Inputs** | | **Output** |
| **X** | **Y** | **F** |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

In our previous section, we learned about how we can form the minterm from the variable's value. Now, a column will be added for the minterm in the above table. The complement of the variables is taken whose value is 0, and the variables whose value is 1 will remain the same.

|  |  |  |  |
| --- | --- | --- | --- |
| **Inputs** | | **Output** | **Minterm** |
| **X** | **Y** | **F** | **M** |
| 0 | 0 | 0 | X'Y' |
| 0 | 1 | 1 | X'Y |
| 1 | 0 | 1 | XY' |
| 1 | 1 | 1 | XY |

Now, we will add all the minterms for which the output is true to find the desired canonical SOP(Sum of Product) expression.

F=X' Y+XY'+XY

# Product of Sum (POS)

A canonical product of sum is a boolean expression that entirely consists of maxterms. The Boolean function F is defined on two variables X and Y. The X and Y are the inputs of the boolean function F whose output is true when only one of the inputs is set to true. The truth table for Boolean expression F is as follows:

|  |  |  |
| --- | --- | --- |
| **Inputs** | | **Output** |
| **X** | **Y** | **F** |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

In our minterm and maxterm section, we learned about how we can form the maxterm from the variable's value. A column will be added for the maxterm in the above table. The complement of the variables is taken whose value is 0, and the variables whose value is 1 will remain the same.

Now, we will multiply all the minterms for which the output is false to find the desired canonical POS(Product of sum) expression.

F=(X'+Y').(X+Y)