

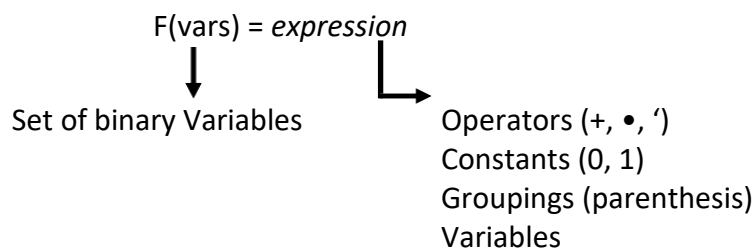
# Boolean Function

## Boolean Function

Boolean algebra is an algebra that deals with binary variables and logic operations.

A Boolean function described by an algebraic expression consists of binary variables, the constants 0 and 1, and the logic operation symbols.

For a given value of the binary variables, the function can be equal to either 1 or 0.



Consider an example for the Boolean function

$$F1 = x + y'z$$

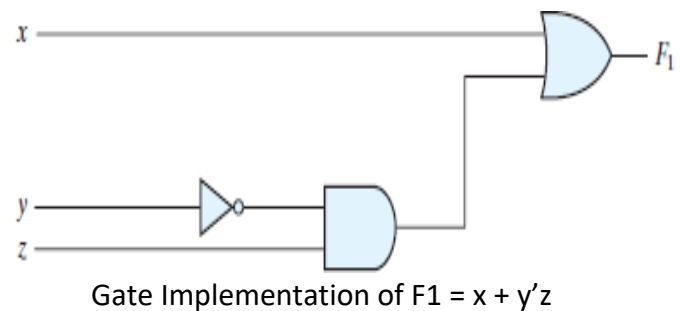
The function  $F_1$  is equal to 1 if  $x$  is equal to 1 or if both  $y'$  and  $z$  are equal to 1.  $F_1$  is equal to 0 otherwise. The complement operation dictates that when  $y' = 1$ ,  $y = 0$ . Therefore,  $F_1 = 1$  if  $x = 1$  or if  $y = 0$  and  $z = 1$ .

A Boolean function expresses the logical relationship between binary variables and is evaluated by determining the binary value of the expression for all possible values of the variables.

A Boolean function can be represented in a truth table. The number of rows in the truth table is  $2^n$ , where  $n$  is the number of variables in the function. The binary combinations for the truth table are obtained from the binary numbers by counting from 0 through  $2^n - 1$ .

Truth Table for  $F_1$

x	y	z	$F_1$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1



**Note:**

Q: Let a function  $F()$  depend on  $n$  variables. How many rows are there in the truth table of  $F()$  ?

A:  $2^n$  rows, since there are  $2^n$  possible binary patterns/combinations for the  $n$  variables.

Truth Tables

- Enumerates all possible combinations of variable values and the corresponding function value
- Truth tables for some arbitrary functions  
 $F_1(x,y,z)$ ,  $F_2(x,y,z)$ , and  $F_3(x,y,z)$  are shown to the below.

x	y	z	$F_1$	$F_2$	$F_3$
0	0	0	0	1	1
0	0	1	0	0	1

0	1	0	0	0	1
0	1	1	0	1	1
1	0	0	0	1	0
1	0	1	0	1	0
1	1	0	0	0	0
1	1	1	1	0	1

- Truth table: a unique representation of a Boolean function
- If two functions have identical truth tables, the functions are equivalent (and vice-versa).
- Truth tables can be used to prove equality theorems.
- However, the size of a truth table grows exponentially with the number of variables involved, hence unwieldy. This motivates the use of Boolean Algebra.

### Boolean expressions-NOT unique

Unlike truth tables, expressions representing a Boolean function are NOT unique.

- Example:
  - $F(x,y,z) = x' \cdot y' \cdot z' + x' \cdot y \cdot z' + x \cdot y \cdot z'$
  - $G(x,y,z) = x' \cdot y' \cdot z' + y \cdot z'$
- The corresponding truth tables for F() and G() are to the right. They are identical.
- Thus,  $F() = G()$

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

### Algebraic Manipulation (Minimization of Boolean function)

- Boolean algebra is a useful tool for simplifying digital circuits.
- Why do it? Simpler can mean cheaper, smaller, faster.
- Example: Simplify  $F = x'yz + x'yz' + xz$ .

$$\begin{aligned}
 F &= x'yz + x'yz' + xz \\
 &= x'y(z+z') + xz \\
 &= x'y \cdot 1 + xz
 \end{aligned}$$

$$= x'y + xz$$

- Example: Prove

$$x'y'z' + x'yz' + xyz' = x'z' + yz'$$

- **Proof:**

$$\begin{aligned} x'y'z' + x'yz' + xyz' \\ &= x'y'z' + x'yz' + x'yz' + xyz' \\ &= x'z'(y'+y) + yz'(x'+x) \\ &= x'z' \cdot 1 + yz' \cdot 1 \\ &= x'z' + yz' \end{aligned}$$

### Complement of a Function

- The complement of a function is derived by interchanging ( $\bullet$  and  $+$ ), and (1 and 0), and complementing each variable.
- Otherwise, interchange 1s to 0s in the truth table column showing F.
- The *complement* of a function IS NOT THE SAME as the *dual* of a function.

Example

- Find  $G(x,y,z)$ , the complement of  $F(x,y,z) = xy'z' + x'yz'$

$$\begin{aligned} \text{Ans: } G &= F' = (xy'z' + x'yz')' \\ &= (xy'z')' \bullet (x'yz')' \quad \text{DeMorgan} \\ &= (x'+y+z) \bullet (x+y'+z') \quad \text{DeMorgan again} \end{aligned}$$

**Note:** The complement of a function can also be derived by finding the function's *dual*, and then complementing all of the literals

### Canonical and Standard Forms

We need to consider formal techniques for the simplification of Boolean functions.

Identical functions will have exactly the same canonical form.

- Minterms and Maxterms
- Sum-of-Minterms and Product-of- Maxterms
- Product and Sum terms
- Sum-of-Products (SOP) and Product-of-Sums (POS)

### Definitions

**Literal:** A variable or its complement

**Product term:** literals connected by  $\bullet$

**Sum term:** literals connected by  $+$

**Minterm:** a product term in which all the variables appear exactly once, either complemented or uncomplemented.

**Maxterm:** a sum term in which all the variables appear exactly once, either complemented or uncomplemented.

**Canonical form:** Boolean functions expressed as a sum of Minterms or product of Maxterms are said to be in canonical form.

**Minterm**

- Represents exactly one combination in the truth table.
- Denoted by  $m_j$ , where  $j$  is the decimal equivalent of the minterm’s corresponding binary combination ( $b_j$ ).
- A variable in  $m_j$  is complemented if its value in  $b_j$  is 0, otherwise is uncomplemented.

Example: Assume 3 variables (A, B, C), and  $j=3$ . Then,  $b_j = 011$  and its corresponding minterm is denoted by  $m_j = A'BC$

**Maxterm**

- Represents exactly one combination in the truth table.
- Denoted by  $M_j$ , where  $j$  is the decimal equivalent of the maxterm’s corresponding binary combination ( $b_j$ ).
- A variable in  $M_j$  is complemented if its value in  $b_j$  is 1, otherwise is uncomplemented.

Example: Assume 3 variables (A, B, C), and  $j=3$ . Then,  $b_j = 011$  and its corresponding maxterm is denoted by  $M_j = A+B'+C'$

**Truth Table notation for Minterms and Maxterms**

- Minterms and Maxterms are easy to denote using a truth table.

Example: Assume 3 variables  $x,y,z$  (order is fixed)

x	y	z	Minterm	Maxterm
0	0	0	$x'y'z' = m_0$	$x+y+z = M_0$
0	0	1	$x'y'z = m_1$	$x+y+z' = M_1$
0	1	0	$x'yz' = m_2$	$x+y'+z = M_2$
0	1	1	$x'yz = m_3$	$x+y'+z' = M_3$
1	0	0	$xy'z' = m_4$	$x'+y+z = M_4$
1	0	1	$xy'z = m_5$	$x'+y+z' = M_5$
1	1	0	$xyz' = m_6$	$x'+y'+z = M_6$
1	1	1	$xyz = m_7$	$x'+y'+z' = M_7$

**Canonical Forms**

- Every function  $F()$  has two canonical forms:
  - Canonical Sum-Of-Products (sum of minterms)
  - Canonical Product-Of-Sums (product of maxterms)

Canonical Sum-Of-Products:

The minterms included are those  $m_j$  such that  $F() = 1$  in row  $j$  of the truth table for  $F()$ .

Canonical Product-Of-Sums:

The maxterms included are those  $M_j$  such that  $F() = 0$  in row  $j$  of the truth table for  $F()$ .

### Example

Consider a Truth table for  $f_1(a,b,c)$  at right

The canonical sum-of-products form for  $f_1$  is

$$f_1(a,b,c) = m_1 + m_2 + m_4 + m_6$$

$$= a'b'c + a'bc' + ab'c' + abc'$$

The canonical product-of-sums form for  $f_1$  is

$$f_1(a,b,c) = M_0 \cdot M_3 \cdot M_5 \cdot M_7$$

$$= (a+b+c) \cdot (a+b'+c') \cdot (a'+b+c') \cdot (a'+b'+c')$$

- Observe that:  $m_j = M_j'$

a	b	c	$f_1$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

## Shorthand: $\Sigma$ and $\Pi$

- $f_1(a,b,c) = \Sigma m(1,2,4,6)$ , where  $\Sigma$  indicates that this is a sum-of-products form, and  $m(1,2,4,6)$  indicates that the minterms to be included are  $m_1$ ,  $m_2$ ,  $m_4$ , and  $m_6$ .
- $f_1(a,b,c) = \Pi M(0,3,5,7)$ , where  $\Pi$  indicates that this is a product-of-sums form, and  $M(0,3,5,7)$  indicates that the maxterms to be included are  $M_0$ ,  $M_3$ ,  $M_5$ , and  $M_7$ .
- Since  $m_j = M_j'$  for any  $j$ ,  
 $\Sigma m(1,2,4,6) = \Pi M(0,3,5,7) = f_1(a,b,c)$
- 

## Conversion between Canonical Forms

- Replace  $\Sigma$  with  $\Pi$  (or *vice versa*) and replace those  $j$ 's that appeared in the original form with those that do not.

- Example:

$$\begin{aligned}f_1(a,b,c) &= a'b'c + a'bc' + ab'c' + abc' \\ &= m_1 + m_2 + m_4 + m_6 \\ &= \Sigma(1,2,4,6) \\ &= \Pi(0,3,5,7) \\ &= (a+b+c) \cdot (a+b'+c') \cdot (a'+b+c') \cdot (a'+b'+c')\end{aligned}$$

## Standard Forms

Another way to express Boolean functions is in standard form. In this configuration, the terms that form the function may contain one, two, or any number of literals.

There are two types of standard forms: the sum of products and products of sums.

The sum of products is a Boolean expression containing AND terms, called product terms, with one or more literals each. The sum denotes the ORing of these terms. An example of a function expressed as a sum of products is

$$F1 = y' + xy + x'yz'$$

The expression has three product terms, with one, two, and three literals. Their sum is, in effect, an OR operation.

A product of sums is a Boolean expression containing OR terms, called sum terms. Each term may have any number of literals. The product denotes the ANDing of these terms. An example of a function expressed as a product of sums is

$$F2 = x(y' + z)(x' + y + z')$$

This expression has three sum terms, with one, two, and three literals. The product is an AND operation.

## Conversion of SOP from standard to canonical form

### Example-1.

Express the Boolean function  $F = A + B'C$  as a sum of minterms.

Solution: The function has three variables: A, B, and C. The first term A is missing two variables; therefore,

$$A = A(B + B') = AB + AB'$$

This function is still missing one variable, so

$$\begin{aligned} A &= AB(C + C') + AB'(C + C') \\ &= ABC + ABC' + AB'C + AB'C' \end{aligned}$$

The second term  $B'C$  is missing one variable; hence,

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

Combining all terms, we have

$$\begin{aligned} F &= A + B'C \\ &= ABC + ABC' + AB'C + AB'C' + A'B'C \end{aligned}$$

But  $AB'C$  appears twice, and according to theorem ( $x + x = x$ ), it is possible to remove one of those occurrences. Rearranging the minterms in ascending order, we finally obtain

$$\begin{aligned} F &= A'B'C + AB'C + AB'C + ABC' + ABC \\ &= m_1 + m_4 + m_5 + m_6 + m_7 \end{aligned}$$

When a Boolean function is in its sum-of-minterms form, it is sometimes convenient to express the function in the following brief notation:

$$F(A, B, C) = \sum m(1, 4, 5, 6, 7)$$

### Example-2.

Express the Boolean function  $F = xy + x'z$  as a product of maxterms.

Solution: First, convert the function into OR terms by using the distributive law:

$$\begin{aligned} F &= xy + x'z = (xy + x')(xy + z) \\ &= (x + x')(y + x')(x + z)(y + z) \\ &= (x' + y)(x + z)(y + z) \end{aligned}$$

The function has three variables: x, y, and z. Each OR term is missing one variable; therefore,

$$\begin{aligned} x' + y &= x' + y + zz' = (x' + y + z)(x' + y + z') \\ x + z &= x + z + yy' = (x + y + z)(x + y' + z) \\ y + z &= y + z + xx' = (x + y + z)(x' + y + z) \end{aligned}$$

Combining all the terms and removing those which appear more than once, we finally obtain

$$\begin{aligned} F &= (x + y + z)(x + y' + z)(x' + y + z)(x' + y + z) \\ F &= M_0 M_2 M_4 M_5 \end{aligned}$$

A convenient way to express this function is as

$$\text{follows: } F(x, y, z) = \pi M(0, 2, 4, 5)$$

The product symbol,  $\pi$ , denotes the ANDing of maxterms; the numbers are the indices of the maxterms of the function.