Chapter 4: Boolean Algebra

1. Introduction

It is a binary algebra developed by the English mathematician George Boole (1815-1864) to study logic. The variables, called booleans, can only take two values: TRUE or FALSE, or alternatively, 1 or 0. Operators can be defined on these variables, and the result can be recorded in a TRUTH TABLE. These operators can be implemented by electronic circuits: So they are called LOGICAL GATES

2. Definition and Properties of Boolean Algebra

Let B be a set on which two binary operations + and ×, and a unary operation ' are defined, and let 0 and 1 be two distinct elements of B. Under these conditions, the sextuplet: (B, +, ×, ', 0, 1) forms a Boolean algebra if the following postulates, concerning arbitrary elements a, b, and c of B, are satisfied:

Commutativity	$a + b = b + a$	$a \times b = b \times a$
Distributivity	$a + (b \times c) = (a + b) \times (a + c)$	$a \times (b + c) = (a \times b) + (a \times c)$
Identity	$a + 0 = a$	$a \times 1 = a$
Complementarity	$a + a' = 1$	$a \times a' = 0$

0 is called the null element, 1 is the unit element, and a' (or $\overline{a}$) is the complement of a. The operations + and × are respectively called sum and product. Often, we write the symbol × as . (a simple dot) or we simply ignore it. For example:

$$a \times (b + c) = a . (b + c) = a (b + c)$$

2.1. Operator Precedence “priority”

The highest priority is given to parentheses, then the priority is to the negation ' (or $\overline{\ }$) and then to the × operator, and finally to the + operator.

Example :

a + b × c means a + (b × c) and not (a + b) × c
a × b’ means a× (b’) and not (a × b)’

Note :

Be careful! Although their names and symbols may look similar, do not confuse logical sum and product with arithmetic sum and product as we know them. The former operates on logical values, whereas the latter operates on numbers.

Convention:

Since we will exclusively use Boolean algebra to implement logical circuits using logic gates, from now on, we will call the operator × as logical AND and the operator + as logical OR. As for the complement of a: $\overline{a}$, it represents the logical negation of “a”, i.e., (NOT a).

Example :

Let B be the set {0,1} on which the operators + and × are defined. Let's assume that the complements are defined as $\overline{1}$ = 0 and $\overline{0}$ = 1, So B is a Boolean algebra.

+	**1**	**0**
1	1	1
0	1	0

x	**1**	**0**
1	1	0
0	0	0

2.2. Principle of Duality

The dual of any statement in a Boolean algebra B is the statement obtained by interchanging the operators + and ×, as well as the elements 0 and 1, in the original expression. The dual of a true expression is also true. In other words, if the original expression is true, its dual is also true

Example: The dual of the Boolean expression:

$(1 + a) \times (b + 0) = b$ is $(0 \times a) + (b \times 1) = b$

2.3. Fundamental Theorems

Let a, b, and c elements of a Boolean algebra:

- Idempotent law: $a + a = a$ $\quad a \times a = a$
  $a + 1 = 1$ $\quad a \times 0 = 0$
- Absorption law: $a + (a \times b) = a$ $\quad a \times (a + b) = a$
- associative law: $(a + b) + c = a + (b + c)$ $\quad (a \times b) \times c = a \times (b \times c)$
- Uniqueness of Complement: If $a + x = 1$ and $a \times x = 0$ $\quad x = \overline{a}$
  i.e : $a + \overline{a} = 1$ et $a \times \overline{a} = 0$
- Involution Law: $\overline{\overline{a}} = a$ // $\overline{0} = 1$ et $\overline{1} = 0$
- Morgan's Law: $\overline{(a + b)} = \overline{a} \times \overline{b}$ $\quad \overline{(a \times b)} = \overline{a} + \overline{b}$
- Generalized theorem of morgan's law:

$$\overline{(a + b + c + \ldots)} = \overline{a} \times \overline{b} \times \overline{c} \times \ldots \qquad \overline{(a \times b \times c \times \ldots)} = \overline{a} + \overline{b} + \overline{c} + \ldots$$

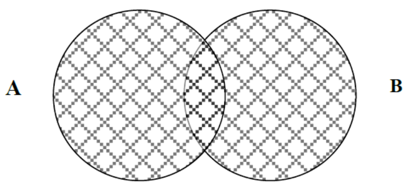
Venn Diagram:

There is a visual way to translate these abstract concepts: it consists of considering Boolean classes as sets. "*The idea is not so foolish since a great English mathematician, John Venn [1834-1923], has preceded us in this regard*".

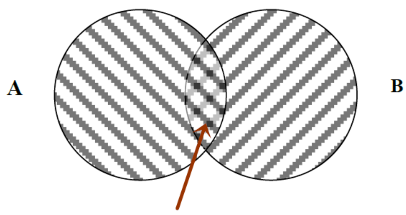
Thus, the fundamental Boolean operations take the form of operators on sets, and we can use Venn diagrams to represent them:

- The logical sum of two classes (A + B) is represented by the union (A U B) of the two corresponding sets.
- The logical product (A . B) is represented by the intersection (A ∩ B) of the two sets.
- The complement $\overline{A}$ of A is represented by the complement of a set A.

By using Venn diagrams, these abstract concepts become more tangible and easier to visualize.

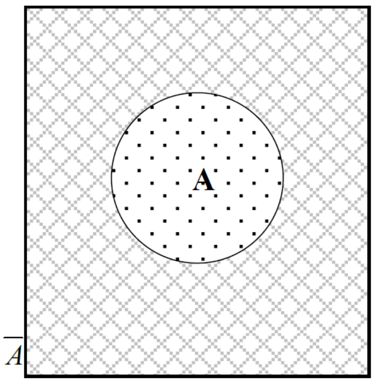


(A ∪ B) ↔ A+B



(A∩B) ↔ A.B

C(A) ↔ $\overline{A}$



Example : Let's simplify the following expressions as much as possible using the various laws we have learned in the course:

- E1= a + ($\overline{a}$. b) et E2= (a + b) × (a + $\overline{b}$)
- E1= a + ($\overline{a}$. b) = (a + $\overline{a}$) . (a + b) = 1 . (a + b) = a + b
- E2= (a + b) . (a + $\overline{b}$) = a . (a + $\overline{b}$) + b . (a + $\overline{b}$) = aa + a$\overline{b}$ + ba + b$\overline{b}$ =
  a + a$\overline{b}$ + ab = a + a($\overline{b}$ + b) = a + a.1 = a + a = a

2.4. Boolean function

$$F(A,B,C) = A\,B\,\overline{C} + A\,\overline{C}\,B + \overline{A}\,B\,C$$

It is a function that relates N logical variables with a set of logical operators. A logical variable (Boolean) is a variable that can take either the value 0 or 1. The value of a logical function is also either 1 or 0, depending on the values of the logical variables. If a logical function has N logical variables, it implies that there can be 2^n combinations of its variables, so this same function can have 2^n values. The 2^n combinations are represented in a table called the truth table (T.T).

Example : Let's consider the logical function

$$F(A,B,C) = \overline{A}\overline{B}C + \overline{A}BC + A\overline{B}C + ABC$$

A	B	C		F
0	0	0		0
0	0	1		1
0	1	0		0
0	1	1		1
1	0	0		0
1	0	1		1
1	1	0		0
1	1	1		1

2.4.1. Canonical form of boolean function

The canonical form of a boolean function is the form where each term of the function includes all the variables.

Example :

The function $F(A,B,C) = A\,B\,\overline{C} + A\,\overline{C}\,B + \overline{A}\,B\,C$ is in canonical form.

2.4.1.1. First Canonical Form (Disjunctive Form)

This form is expressed as a sum of products (or sum of minterms). It is also called a disjunction of conjunctions. This form is the most commonly used.

Example :

The following function is in the first canonical form:

$$F(A,B,C) = \overline{A}.B.C + A.\overline{B}.C + A.B.\overline{C} + A.B.C$$

2.4.1.2. Second Canonical Form (Conjunctive Form)

This form is expressed as a product of sums (or product of maxterms). It is also called a conjunction of disjunctions. Both canonical forms (first and second) are equivalent.

Example :

The following function is in the second canonical form:

$$F(A,B,C) = (A + B + C)(A + B + \overline{C})(A + \overline{B} + C)(\overline{A} + B + C)$$

2.4.2. Extracting Canonical Forms from Truth Table T.T

A function can be expressed in either its first canonical form or its second canonical form based on its truth table. The best way to illustrate this is through an example:

A	B	C		F		
0	0	0		0	→ $A+B+C$	MAX TERMS
0	0	1		0	→ $A+B+\overline{C}$	MAX TERMS
0	1	0		0	→ $A+\overline{B}+C$	MAX TERMS
0	1	1		1	→ $\overline{A} . B . C$	
1	0	0		0	→ $\overline{A}+B+C$	
1	0	1		1	→ $A . \overline{B} . C$	MIN TERMS
1	1	0		1	→ $A . B . \overline{C}$	MIN TERMS
1	1	1		1	→ $A . B . C$	MIN TERMS

We can express the function F in the form of a product of maxterms (second canonical form) by considering the rows containing "0" in the truth table:

$$F(A,B,C) = (A + B + C)(A + B + \overline{C})(A + \overline{B} + C)(\overline{A} + B + C)$$

Or in the form of a sum of minterms (first canonical form) by considering the rows containing "1" in the truth table of F.

$$F(A, B,C) = \overline{A}.B.C + A.\overline{B}.C + A.B.\overline{C} + A.B.C$$

Note:
'How to Retrieve a Canonical Form from a Simplified Equation?'

We can always convert any logical function to the canonical forms. This involves adding the missing variables in the terms that do not contain all the variables (non-canonical terms). This can be achieved using the rules of Boolean algebra:

- Multiply a term with an expression equal to 1,
- Add to a term with an expression equal to,
- Then make the distribution,

In other words, we simply complete the missing variable(s) in each term without changing the function's value. For example, for the logical function H_1 with three variables (a, b, c) and whose "simplified" logical equation is:

$$H_1(a, b, c) = a.\overline{b} + a.b.\overline{c}$$

The first canonical form can be obtained as follows:

$$H_1(a, b, c) = a.\overline{b} + a.b.\overline{c} = a.\overline{b}.(c+\overline{c}) + a.b.\overline{c} = a.\overline{b}.c + a.\overline{b}.\overline{c} + a.b.\overline{c}$$

Note that the added term does not change the function because this term equals 1.

Examples :

- $F(A,B) = A + B$
  $= A (B+\overline{B}) + B (A+\overline{A})$
  $= AB + A\overline{B} + AB + \overline{A}B$
  $= AB + A\overline{B} + \overline{A}B$

- $F(A,B,C) = AB + C$
  $= AB (C + \overline{C}) + C (A + \overline{A})$
  $= ABC + AB\overline{C} + AC + \overline{A}C$
  $= ABC + AB\overline{C} + AC (B + \overline{B}) + \overline{A}C (B + \overline{B})$
  $= ABC + AB\overline{C} + ABC + A\overline{B}C + \overline{A}BC + \overline{A}\overline{B}C$
  $= ABC + AB\overline{C} + A\overline{B}C + \overline{A}BC + \overline{A}\overline{B}C$

2.4.3. Complement of a Boolean Function

The complement of a Boolean function (negation of a function) can be obtained by applying Morgan's law if it is in one of its canonical forms, or by inverting its truth table. In general, we deduce the complement of a function in one of the two canonical forms by replacing OR with AND, AND with OR, and each term with its complement.

Example1 :

$F(A, B, C , D) = A + \overline{A}B + (C + \overline{D})$

$\overline{F}(A, B, C , D) = \overline{A + \overline{A}B + (C + \overline{D})} = \overline{A}.(\overline{\overline{A}} + \overline{B}).\overline{C}.\overline{\overline{D}} = \overline{A}.(A + \overline{B}).\overline{C}.D$

Example2 :

Let’s calculate the complement of the function presented in ***2.4.2***.