Chapter 2 :

Boolean algebra and logic Gates

Table of contents

Introduction

1- Définitions

Around 1847, George Boole defined an algebra applicable to logical functions of logical variables (Boolean variables). It was 70 years later that Boole's work gained widespread interest, when Claude Shannon made the connection between Boolean algebra and circuit design. Claude Shannon showed that Boolean algebra could be used to optimize circuits.

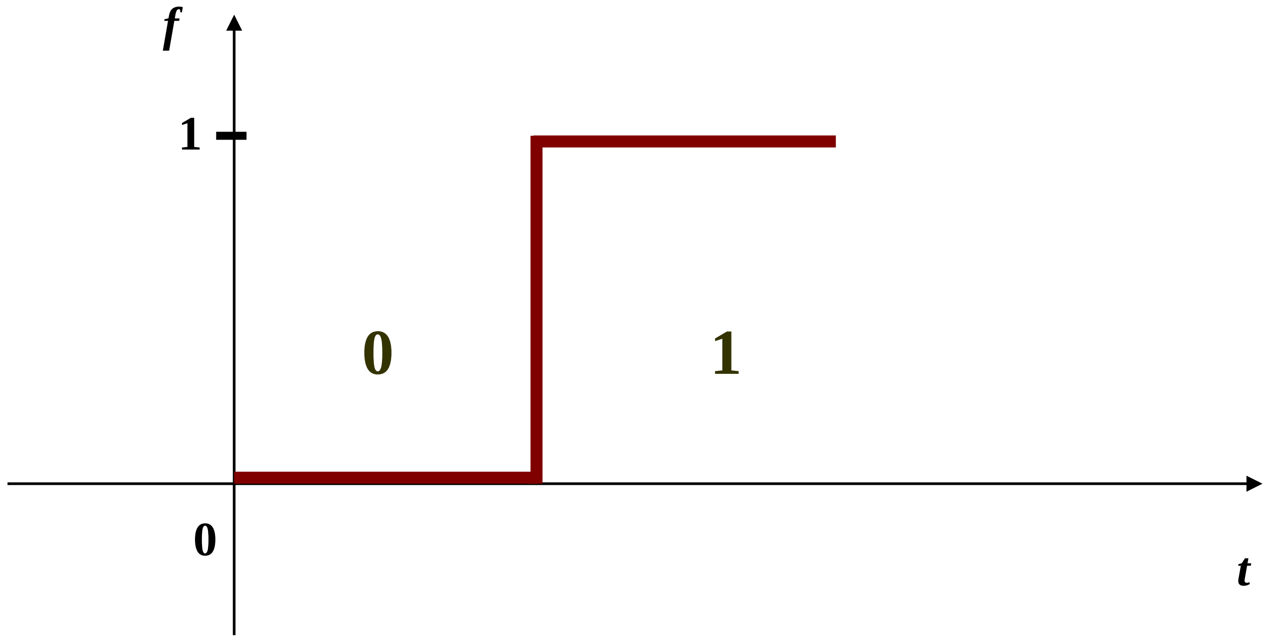
Electronic circuits fall into two categories :

- ***Analog circuits:*** **These are circuits where the electrical signals have a continuously varying amplitude.**
- ***Digital circuits:*** These are circuits in which the electrical signals can have only two levels:

- Level 1

- Level 0

Introduction



George Boole (1815-1864)

Boolean algebra and logic functions

1- Logical variable

A logical variable, often referred to as a Boolean variable, is a variable that can take on one of two possible values: 0 or 1.

2- Logic operators

Logic operators, or boolean operators, are symbols or words used in logic to combine or manipulate logic functions:

- **NOT** ($\overline{a}$) : Returns the opposite of the operand's value.
- **AND** (a.b) :Returns 1 if both operands are 1.
- **OR** (a+b) : Returns 1 if at least one of the operands is 1.

a	b	a.b
0	0	0
0	1	0
1	0	0
1	1	1

a	b	a+b
0	0	0
0	1	1
1	0	1
1	1	1

Boolean algebra and logic functions (2)

Logic functions

Also known or Boolean functions, are mathematical functions that operate on one or more binary input values (0 or 1) and produce a binary output value. These functions are fundamental in the field of logic, computer science, and digital electronics.

Exemple : $f(x) = \overline{x}$

$$f(a,b) = \overline{a} + ab$$

$$f(x,y,z) = xyz + y + x\overline{z}$$

Boolean algebra and logic functions (3)

Logic functions are represented using truth tables or algebraic expressions

5-a°/ using algebraic expressions

In logic, algebraic expressions typically use symbols and operators to represent logical operations.

Example: f Function, of three variables x, y and z.

$$f(x,y,z) = x \cdot y + \overline{y} \cdot z + x \cdot \overline{z}$$

5-b°/ using truth tables

They show all possible input combinations and the corresponding output of a logic function. Let's create truth tables for some common logic functions . Truth tables are especially useful for understanding the behavior of logical functions and evaluating the truth or falsity of complex logical expressions.

✓ There are the input variables, represented in the first columns

✓ The third column represents the result of the logical expression B for each combination of input variables.

Boolean algebra and logic functions (4)

Converting Boolean Expressions to Truth Tables

Certainly! To determine the Boolean value of a function f(x) for each possible combination of variable values, you list all possible combinations of input variables and calculate the corresponding output of the function.

Exemple: $f(a,b,c) = ab + \bar{b}c + a\bar{c}$

a	b	c	$\bar{b}$	$\bar{c}$	ab	$\bar{b}c$	$a\bar{c}$	$f(a,b,c)$
0	0	0	1	1	0	0	0	0
0	0	1	1	0	0	1	0	1
0	1	0	0	1	0	0	0	0
0	1	1	0	0	0	0	0	0
1	0	0	1	1	0	0	1	1
1	0	1	1	0	0	1	0	1
1	1	0	0	1	1	0	1	1
1	1	1	0	0	1	0	0	1

NB: **A Boolean function refers to a function having n number of entries or variables, so it has 2^n number of possible combinations of the given variables**

Postulates of boolean algebra

Postulate 1 : *Fermeture par rapport aux opérateurs OU et AND)*

$\forall x,y \in B, x+y \in B$ **B : l'ensemble des valeurs booléennes**

$\forall x,y \in B, x.y \in B$ **{0,1}**

Postulate 2 : *Identity Law (Idempotent Law)*

$$\boxed{x+x=x, \qquad x\cdot x=x}$$

Postulate 3 : *Null Laws (Annihilation Laws)*

$$x+0=x \quad et \quad x\cdot 1=x$$

Postulate 4 : *Domination Laws*

$$\boxed{x+1=1,\ x\cdot 0=0}$$

Postulate 5 : Double Negation Law

$$\boxed{\overline{\overline{x}}=x}$$

Postulates of boolean algebra

Postulate 6: Complement Laws

$$x+\overline{x}=1 \quad et \quad x\cdot\overline{x}=0$$

Postulate 7: Commutative Laws

$$x.y = y.x \quad or \quad x+y=y+x$$

Postulate 8: Associative Laws

$$(x+y)+z=x+(y+z) \quad et \quad x\cdot(y\cdot z)=(x\cdot y)\cdot z$$

Postulate 9: Distributive Laws

$$x+(y\cdot z)=(x+y)\cdot(x+z) \quad et \quad x\cdot(y+z)=x\cdot y+x\cdot z$$

Theorems of boolean algebra

Théorème 5: De Morgan's Theorems

$$\boxed{\overline{x \cdot y} = \bar{x} + \bar{y}, \qquad \overline{x + y} = \bar{x} \cdot \bar{y}}$$

Théorème 6: De Morgan's theorems genrelise

$$\boxed{\overline{xyz\ldots} = \bar{x} + \bar{y} + \bar{z} + \ldots, \ \overline{x + y + z + \ldots} = \bar{x} \cdot \bar{y} \cdot \bar{z} \cdot \ldots.}$$

Théorème 7: Absorption Theorems

$$\boxed{x + x \cdot y = x, x \cdot (x + y) = x}$$

Théorème 8: Consensus Theorem

$$\boxed{x + \bar{x} \cdot y = x + y, \ \ x \cdot (\bar{x} + y) = x \cdot y}$$

Théorème 9: Consensus Theorem genrelise

$$\boxed{x \cdot y + \bar{x} \cdot z + y \cdot z = x \cdot y + \bar{x} \cdot z, \ (x + y) \cdot (\bar{x} + z) \cdot (y + z) = (x + y) \cdot (\bar{x} + z)}$$

Simplifying logic functions

Simplifying logic functions is a common task in digital design and computer science. The goal is to simplify a Boolean expression or logic circuit while preserving its functionality. There are several techniques for simplifying logic functions, including:

1- **Boolean Algebra:**
Use Boolean algebra laws and theorems to manipulate and simplify expressions. Common laws include the commutative, associative, distributive, identity, and complement laws.

2- **Karnaugh Maps (K-Maps):**
It helps simplify Boolean expressions and minimize logic circuits. Karnaugh Maps are particularly useful for functions with a small number of variables.

Simplifying logic functions (5)

Karnaugh Maps (K-Maps)

❖ **Grid Structure:** A K-Map is represented as a grid or truth table , with cells arranged in rows and columns. The number of rows and columns depends on the number of input variables in the Boolean function.

❖ **Input Variables:** Each cell in the K-Map corresponds to a unique combination of input variable values. For example, in a 2-variable K-Map, you would have four cells representing the combinations (00), (01), (10), (11).

❖**Grouping Technique:** The primary goal is to group adjacent "1"s in the K-Map into rectangles (or squares) that have dimensions of 2^n. You can create groups horizontally, vertically or both.

❖**Minimal Sum-of-Products (SOP):** After grouping the "1"s, you translate these groups into Boolean expressions. Each group corresponds to a term in the minimal Sum-of-Products (SOP) expression.

❖**Minimal Product-of-Sums (POS):** Alternatively, you can create a minimal Product-of-Sums (POS) expression by grouping "0"s instead of "1"s in the K-Map.

Simplifying logic functions (2)

Karnaugh Maps (K-Maps)

2-VARIABLE KARNAUGH MAP

A\B	0	1
0	m0	m1
1	m2	m3

3-VARIABLE KARNAUGH MAP

A\BC	00	01	11	10
0	m0	m1	m3	m2
1	m4	m5	m7	m6

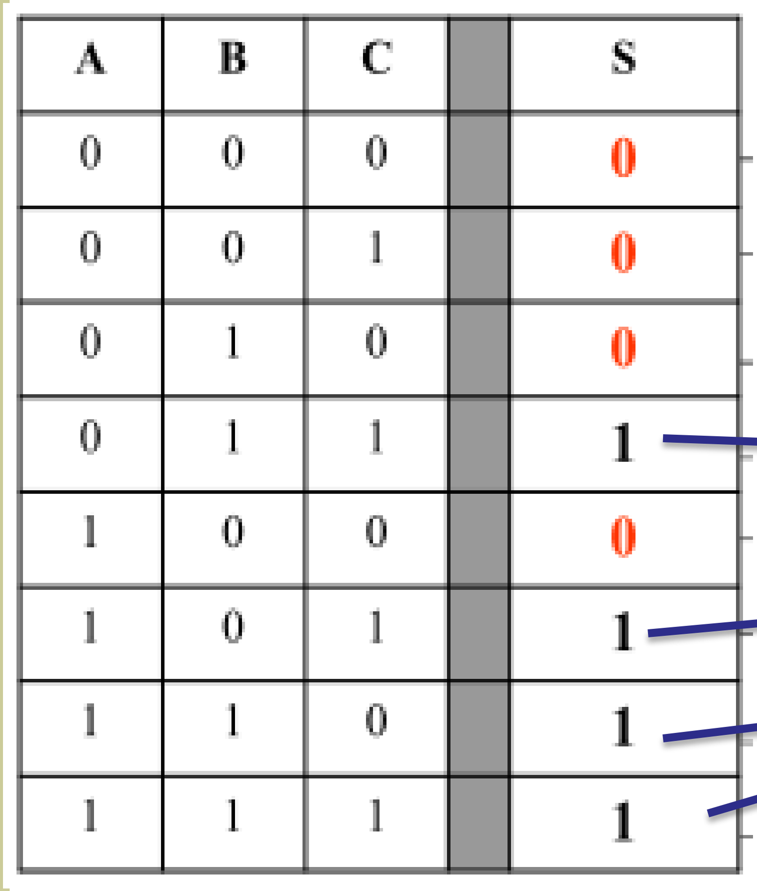
4-VARIABLE KARNAUGH MAP

AB\CD	00	01	11	10
00	m0	m1	m3	m2
01	m4	m5	m7	m6
11	m12	m13	m15	m14
10	m8	m9	m11	m10

Simplifying logic functions (4)

Karnaugh Maps (K-Maps)

EXAMPLE:



A	B	C		S
0	0	0		0
0	0	1		0
0	1	0		0
0	1	1		1
1	0	0		0
1	0	1		1
1	1	0		1
1	1	1		1

1

A\BC	00	01	11	10
0				
1				

Simplifying logic functions (4)

Karnaugh Maps (K-Maps)

adjacent cell in the K-Map

A\BC	**00**	**01**	**11**	**10**
0	🟦	🟥	🟦	🟨
1	🟨	🟦	🟨	🟨

- Adjacent to the red cell are the three blue cells..
- Yellow cells not adjacent to red squares

Simplifying logic functions (5)

Karnaugh Maps (K-Maps)

Methodology

- ❑ Look for groups of '1s' in the Karnaugh Map. These groups can be in the form of squares or rectangles. The goal is to group as many adjacent '1s' as possible.
- ❑ the size of the group should be a power of 2 (8,4,2,and).
- ❑ A '1' can be part of multiple groups.
- ❑The goal is to cover all '1s' using as few terms as possible.

Simplifying logic functions (6)

Karnaugh Maps (K-Maps)

Methodology

❑Once you have identified the groups of '1s', write down the simplified terms for each group. These terms can be derived from the row and column labels associated with the cells in each group.

❑Combine the simplified terms to create the minimized Boolean expression. The simplified expression will have fewer terms than the original expression while preserving the same logic function.

Simplifying logic functions (7)

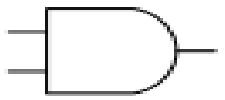
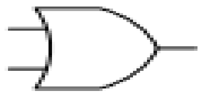
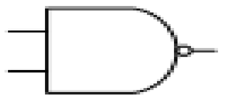
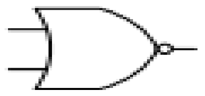
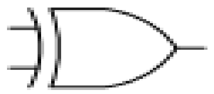
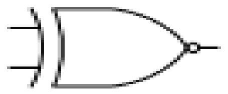
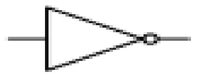
Example

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

A\BC	00	01	11	10
0	0	1	0	0
1	1	1	0	1

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

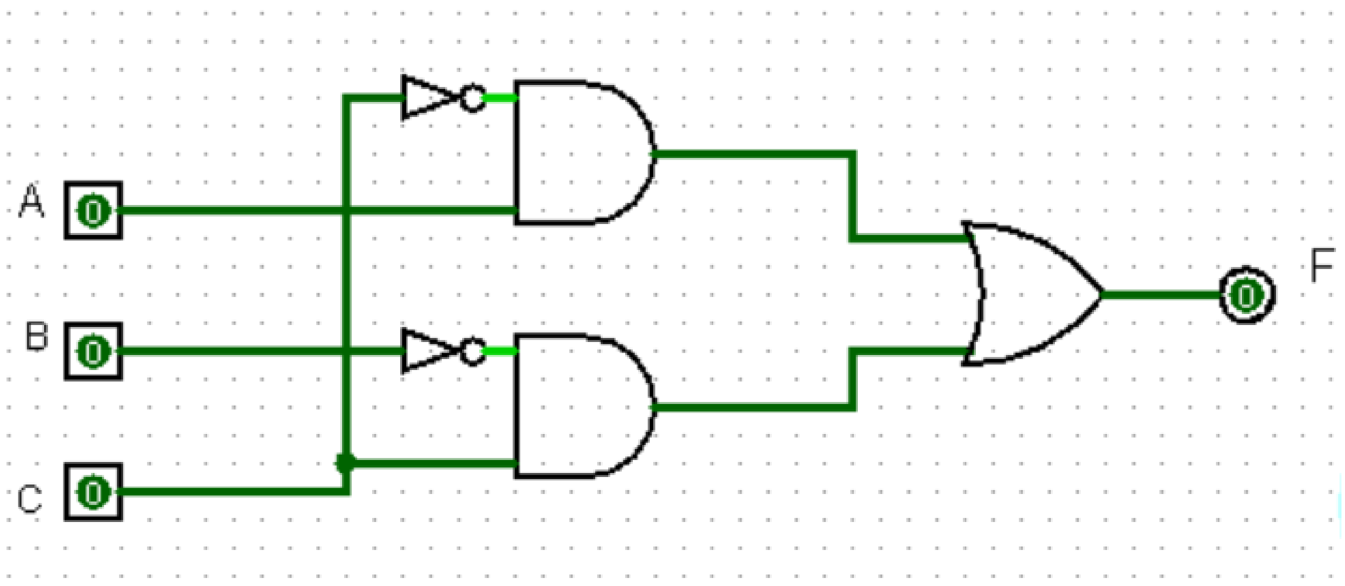
Logic Gates and circuits

Opérateur logique	Nom français	Nom anglais	Symbole	Table de vérité
$A \cdot B$	ET	AND		A B F 0 0 0 0 1 0 1 0 0 1 1 1
$A+B$	OU	OR		A B F 0 0 0 0 1 1 1 0 1 1 1 1
$\overline{A \cdot B}$	NON ET	NAND		A B F 0 0 1 0 1 1 1 0 1 1 1 0
$\overline{A+B}$	NON OU	NOR		A B F 0 0 1 0 1 0 1 0 0 1 1 0
$A \oplus B$	OU exclusif	XOR		A B F 0 0 0 0 1 1 1 0 1 1 1 0
$\overline{A \oplus B} = A \otimes B$	NON OU exclusif	XNOR		A B F 0 0 1 0 1 0 1 0 0 1 1 1
$\overline{A}$	NON (inverseur)	NOT (inverter)		A F 0 1 1 0

Logic Gates and circuits

Example

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



Activity 2

Consider the following Boolean logic function: $F(x,y,z,w) = y\bar{z}\bar{w} + \bar{y}zw + \bar{x}\bar{y}z\bar{w} + \bar{x}yzw + \bar{x}yz\bar{w}$

1) Simplify the function F(x,y,z,w) and draw the equivalent circuit.
2) Which (4-variable) terms need to be added to the initial (non-simplifying) function *F(x,y,z,w)*, to obtain a simplifying function with only two-variable terms.

Solution Activity chapter No.2:

II) $F(x,y,z,w) = y\bar{z}\bar{w} + \bar{y}zw + \bar{x}\bar{y}z\bar{w} + \bar{x}yzw + \bar{x}yz\bar{w}$ **(2.75pt)**

1) Simplification de la fonction F(x,y,z,w) :

A) Method of the laws and theorems of the Boolean algebra :

$F(x,y,z,w) = y\bar{z}\bar{w} + \bar{y}zw + \bar{x}\bar{y}z\bar{w} + \bar{x}yzw + \bar{x}yz\bar{w}$

$F(x,y,z,w) = y\bar{z}\bar{w} + \bar{y}z(w + \bar{x}\bar{w}) + \bar{x}yz(w + \bar{w})$ **(0.25pt)**

$F(x,y,z,w) = y\bar{z}\bar{w} + \bar{y}z(w + \bar{x}) + \bar{x}yz$ **(0.25pt)**

$F(x,y,z,w) = y\bar{z}\bar{w} + \bar{y}zw + \bar{x}\bar{y}z + \bar{x}yz$ **(0.25pt)**

$F(x,y,z,w) = y\bar{z}\bar{w} + \bar{y}zw + \bar{x}z(\bar{y} + y)$ **(0.25pt)**

$\boldsymbol{F(x,y,z,w) = y\bar{z}\bar{w} + \bar{y}zw + \bar{x}z}$ **F simplifier (0.5pt)**

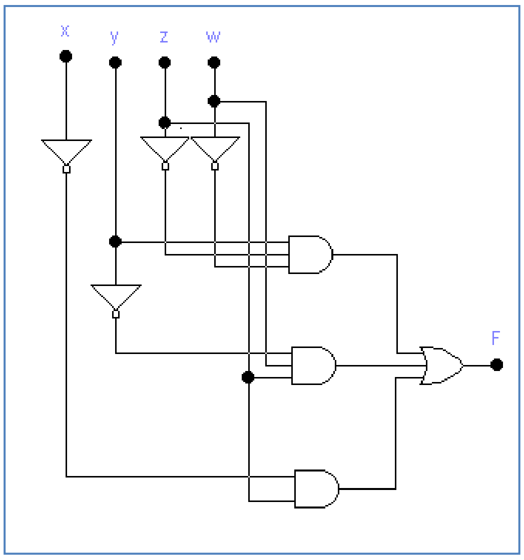
B) Karnaugh Map method:

Tableau : **(1pt)**

xy \ zw	00	01	11	10
00	0	0	1	1
01	1	0	1	1
11	1	0	0	0
10	0	0	1	0

$\boldsymbol{F(x,y,z,w) = y\bar{z}\bar{w} + \bar{y}zw + \bar{x}z}$ **F simplifier (0.5pt)**

- Logic circuit **(0.5pt)**



2) L The 4-variable terms, when added to the initial function F(x,y,z,w) (do not simplify), In order to obtain terms with only two variables in the function F, they are simplified, are:

$\boldsymbol{xyzw}$ **and** $\boldsymbol{xyz\overline{w}}$ **(0,75pt)**

$\boldsymbol{G(x,y,z,w) = y\overline{w} + zw + \overline{x}z}$