

## 1. Introduction

Lorsqu'on exprime les variables de sorties uniquement en fonction des variables d'entrées, le problème à résoudre relève de la logique combinatoire. Autrement dit à chaque combinaison des variables d'entrées correspond toujours une et une seule combinaison des variables de sorties.



## 2. Fonctions et opérateurs logiques

### 21. Fonctions et opérateurs logiques élémentaires

#### 211. Fonction logique OUI

Schéma électrique

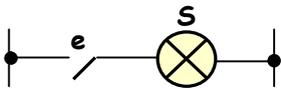


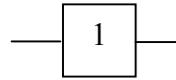
Table de vérité

e	S
0	0
1	1

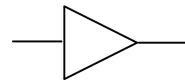
Equation logique

$$S = e$$

Opérateur logique (symbole)



Norme française



Norme américaine

Fonction Traiter

#### 212. Fonction logique NON

Schéma électrique

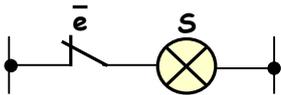


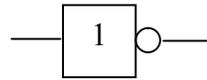
Table de vérité

e	S
0	1
1	0

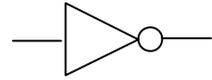
Equation logique

$$S = \bar{e}$$

Opérateur logique (symbole)



Norme française



Norme américaine

#### 213. Fonction logique OU (OR)

Schéma électrique

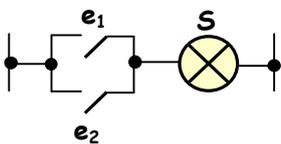


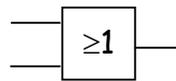
Table de vérité

e <sub>1</sub>	e <sub>2</sub>	S
0	0	0
0	1	1
1	0	1
1	1	1

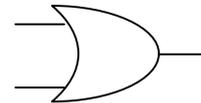
Equation logique

$$S = e_1 + e_2$$

Opérateur logique (symbole)



Norme française



Norme américaine

#### 214. Fonction logique ET (AND)

Schéma électrique

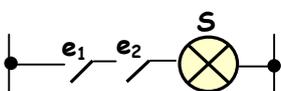


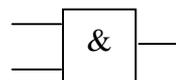
Table de vérité

e <sub>1</sub>	e <sub>2</sub>	S
0	0	0
0	1	0
1	0	0
1	1	1

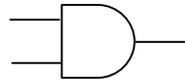
Equation logique

$$S = e_1 \cdot e_2$$

Opérateur logique (symbole)

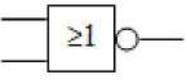
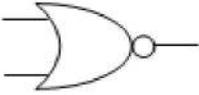
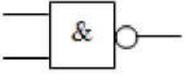
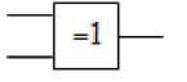
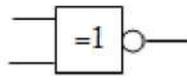
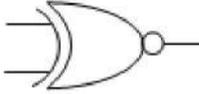


Norme française



Norme américaine

## 22. Fonctions et opérateurs logiques dérivés

Fonction logique	Table de vérité	Equation logique	Opérateur logique (symbole)																
			Norme française	Norme américains															
NON OU (NOR)	<table border="1"> <thead> <tr><th>e<sub>1</sub></th><th>e<sub>2</sub></th><th>S</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	e <sub>1</sub>	e <sub>2</sub>	S	0	0	1	0	1	0	1	0	0	1	1	0	$S = \overline{e_1 + e_2}$		
e <sub>1</sub>	e <sub>2</sub>	S																	
0	0	1																	
0	1	0																	
1	0	0																	
1	1	0																	
NON ET (NAND)	<table border="1"> <thead> <tr><th>e<sub>1</sub></th><th>e<sub>2</sub></th><th>S</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	e <sub>1</sub>	e <sub>2</sub>	S	0	0	1	0	1	1	1	0	1	1	1	0	$S = \overline{e_1 \cdot e_2}$		
e <sub>1</sub>	e <sub>2</sub>	S																	
0	0	1																	
0	1	1																	
1	0	1																	
1	1	0																	
OU EXCLUSIF (EXOR)	<table border="1"> <thead> <tr><th>e<sub>1</sub></th><th>e<sub>2</sub></th><th>S</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	e <sub>1</sub>	e <sub>2</sub>	S	0	0	0	0	1	1	1	0	1	1	1	0	$S = e_1 \oplus e_2$		
e <sub>1</sub>	e <sub>2</sub>	S																	
0	0	0																	
0	1	1																	
1	0	1																	
1	1	0																	
ET EXCLUSIF (EXAND)	<table border="1"> <thead> <tr><th>e<sub>1</sub></th><th>e<sub>2</sub></th><th>S</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	e <sub>1</sub>	e <sub>2</sub>	S	0	0	1	0	1	0	1	0	0	1	1	1	$S = e_1 \otimes e_2$		
e <sub>1</sub>	e <sub>2</sub>	S																	
0	0	1																	
0	1	0																	
1	0	0																	
1	1	1																	

## 3. Opérations booléennes ou logiques

### 31. Propriétés et opérations élémentaires

Commutativité	$S = a \cdot b = b \cdot a$	$S = a + b = b + a$
Associativité	$S = (a \cdot b) \cdot c = a \cdot (b \cdot c)$	$S = (a + b) + c = a + (b + c)$
Distributivité	$S = a \cdot (b + c) = (a \cdot b) + (a \cdot c)$	$S = a + (b \cdot c) = (a + b) \cdot (a + c)$
Complémentation	$S = a \cdot \bar{a} = 0$	$S = a + \bar{a} = 1$
Idempotence	$S = a \cdot a = a$	$S = a + a = a$
Elément neutre	$S = a \cdot 1 = a$	$S = a + 0 = a$
Elément absorbant	$S = a \cdot 0 = 0$	$S = a + 1 = 1$
Absorption	$S = a + (b \cdot a) = a$	$S = a \cdot (b + a) = a$
Involution	$S = \bar{\bar{a}} = a$	$S = \bar{\bar{\bar{a}}} = \bar{a}$
Inclusion	$S = (a \cdot b) + (a \cdot \bar{b}) = a$	

### 32. Relations fondamentales

$S = a + (\bar{a} \cdot b) = a + b$
$S = a + (a \cdot b) = a$
$S = (a \cdot b) + (\bar{a} \cdot c) = (a \cdot b) + (\bar{a} \cdot c) + (b \cdot c)$
$S = a + (b \cdot c) = (a + b) \cdot (a + c)$

### 33. Théorèmes de DEMORGAN

#### 331. Premier théorème de DEMORGAN ou complément d'un produit logique

$$S = a \cdot b \Rightarrow \overline{S} = \overline{a \cdot b} = \overline{a} + \overline{b}$$

#### 332. Deuxième théorème de DEMORGAN ou complément d'une somme logique

$$S = a + b \Rightarrow \overline{S} = \overline{a + b} = \overline{a} \cdot \overline{b}$$

### 4. Représentation d'une fonction logique

Pratiquement, une fonction logique est représentée par :

#### 41. Sa table de vérité

La table de vérité précise toutes les combinaisons d'états logiques possibles des entrées et l'état correspondant de la sortie.

D'une façon générale, on a  $2^n$  combinaisons pour  $n$  variables d'entrées.

#### 42. Son équation logique

L'équation logique n'est qu'une association de sommes et de produits logiques.

#### 43. Son logigramme

Le logigramme ou diagramme logique est une représentation symbolique, sous forme d'un schéma, formé par les différentes liaisons entre les symboles des opérateurs élémentaires.

#### 44. Son chronogramme

Le chronogramme est une représentation graphique qui permet de visualiser en fonction du temps toutes les combinaisons d'états logiques possibles des entrées et l'état correspondant de la sortie.

#### Exemple :

On déduit l'équation logique de la fonction  $F$ , à partir de la table de vérité suivant le raisonnement suivant :

- On cherche les lignes où la fonction  $F$  est égale à 1.
- On note la combinaison des entrées pour chacune de ces lignes.
- On somme logiquement ces combinaisons.

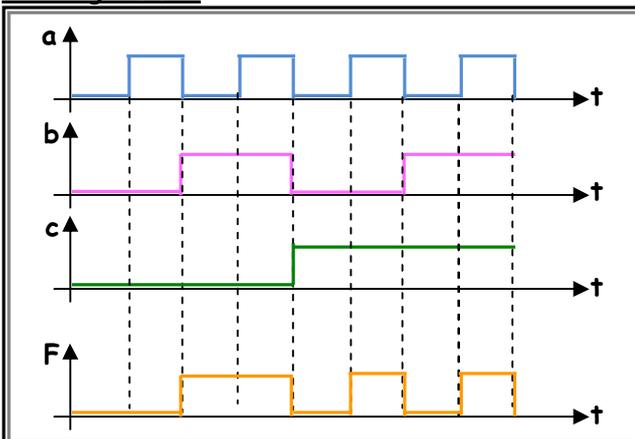
Ainsi, la fonction  $F$  est égale à 1 si  $\overline{a}\overline{b}\overline{c}$  OU  $a\overline{b}\overline{c}$  OU  $a\overline{b}c$  OU  $abc$ , ce qui donne :

$$F = \overline{a}\overline{b}\overline{c} + a\overline{b}\overline{c} + a\overline{b}c + abc = \overline{b}\overline{c} + ac$$

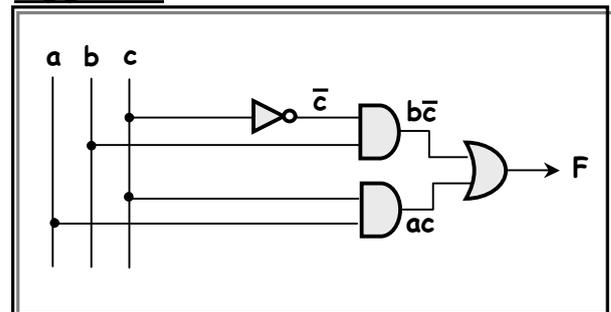
Table de vérité

c	b	a	F
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

#### Chronogrammes



#### Logigramme



## 5. Simplification ou minimisation des fonctions logiques

Le fonctionnement de tout système technique qui relève de la logique combinatoire peut se traduire par une équation logique plus ou moins complexe. La compréhension du fonctionnement du système et la minimisation de son coût de réalisation imposent la recherche de l'équation la plus simple.

### 51. Simplification algébrique

Cette méthode utilise les principes de l'Algèbre de Boole, à savoir : les règles des opérations logiques, des relations fondamentales et des théorèmes de **DEMORGAN**.

**Exemple 1 :**  $F = \bar{a}bc + a\bar{b}c + ab\bar{c} + abc$   
 $\Rightarrow F = \bar{a}bc + a\bar{b}c + ab\bar{c} + abc + abc + abc$   
 $\Rightarrow F = ab + bc + ac$

**Exemple 2 :**  $G = (\bar{x} + \bar{y} + z) \cdot (x + \bar{y} + z) \cdot (\bar{x} + \bar{y})$   
 $\Rightarrow G = \bar{x}z + \bar{y}$

### 52. Simplification graphique par tableau de Karnaugh

#### 521. Règles d'établissement du tableau de Karnaugh

Cette méthode plus simple utilise le tableau de Karnaugh pour simplifier des fonctions booléennes ayant jusqu'à 6 variables. Le tableau de Karnaugh d'une fonction logique est la transformation de sa table de vérité sous forme d'une table contractée à 2 dimensions (lignes et colonnes).

Le passage de la table de vérité au tableau de Karnaugh se fait selon la procédure suivante :

- Chaque ligne (combinaison) de la table de vérité correspond à une case du tableau de Karnaugh.
- Pour passer d'une case à une autre, une seule variable d'entrée change à la fois (code GRAY).
- Deux cases sont dites **adjacentes** si elles ne diffèrent que par une seule variable.

#### Exemple 1 : Représentation des variables dans un tableau de Karnaugh

a	
0	
1	

1 seule variable

a	b	0	1
0			
1			

2 variables

a	bc	00	01	11	10
0					
1					

3 variables

ab	cd	00	01	11	10
00					
01					
11					
10					

4 variables

#### Exemple 2 : Passage de la table de vérité au tableau de Karnaugh

c	b	a	F
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

F	bc	00	01	11	10
a	0	0	0	0	1
1	0	1	1	1	1

### 522. Principe de la simplification à l'aide du tableau de Karnaugh

La méthode de Karnaugh consiste à faire des groupements des cases adjacentes et symétriques du tableau. Ces groupements sont de 2, 4, 8 et 16 cases (généralement la puissance de 2).

En effectuant les groupements, on élimine les variables qui changent d'état et on garde les variables qui restent fixes.

#### 5221. Règles des groupements

##### a. Cases situées en bordure du tableau (cases symétriques)

Dans la recherche des cases adjacentes, on doit utiliser le tableau de Karnaugh comme s'il se refermait sur lui-même, à la fois dans le sens horizontal et dans le sens vertical. Ainsi, deux cases situées en bordure du tableau sur une même ligne ou sur une même colonne peuvent être considérées comme adjacentes.

Exemple :

F	cd	00	01	11	10	
	ab					
	00	1	0	0	1	⇒ $F=bc+\bar{b}\bar{d}$
	01	0	0	1	1	
	11	0	0	1	1	
	10	1	0	0	1	

##### b. Recouvrements

Les groupements les plus grands correspondent aux termes les plus simples, on pourra donc être amené à prendre certaines cases dans plusieurs groupements de façon à ce que ces groupements soient plus grands. Une même case peut être introduite dans plusieurs groupements afin de former des groupements, des cases adjacentes, aussi importants que possible.

Exemple :

F	bc	00	01	11	10	
	a					
	0	1	1	1	1	⇒ $F=\bar{a}+\bar{b}c$
	1	0	1	0	0	

##### c. Cases indifférentes (tableau incomplet)

Il arrive fréquemment que certaines valeurs d'une fonction logique ne soient pas définies, ou indifférentes, pour certaines combinaisons des valeurs des variables d'entrées. Le tableau de Karnaugh correspondant est dit incomplet. Il comporte des cases indéfinies dont les valeurs sont dites indifférentes désignées par X. Ces valeurs indifférentes seront choisies de manière à faciliter la formation des cases adjacentes. On peut attribuer à ces cases soit la valeur 0 soit la valeur 1.

Exemple :

F	cd	00	01	11	10	
	ab					
	00	0	X	X	0	⇒ $F=\bar{a}\bar{b}+\bar{b}d$
	01	1	X	X	1	
	11	0	0	X	X	
	10	0	1	1	0	

## 5222. Applications

Simplifier les équations logiques des fonctions représentées par les tableaux de Karnaugh suivants :

$F_1$	a	bc	00	01	11	10
		0	0	0	1	0
	1		0	1	1	1

$F_1 = \dots\dots\dots$

$F_2$	a	bc	00	01	11	10
		0	0	0	0	1
	1		1	1	1	1

$F_2 = \dots\dots\dots$

$F_3$	a	bc	00	01	11	10
		0	0	1	1	1
	1		1	1	1	1

$F_3 = \dots\dots\dots$

$F_4$	a	bc	00	01	11	10
		0	1	1	1	1
	1		1	1	1	1

$F_4 = \dots\dots\dots$

$F_5$	a	bc	00	01	11	10
		0	1	0	1	0
	1		0	1	1	

$F_5 = \dots\dots\dots$

$F_6$	a	bc	00	01	11	10
		0	1	1	0	1
	1		1	0	1	

$F_6 = \dots\dots\dots$

$F_7$	a	bc	00	01	11	10
		0	0	1	1	0
	1		0	1	1	

$F_7 = \dots\dots\dots$

$F_8$	ab	cd	00	01	11	10
		00	0	1	1	0
		01	1	1	1	1
		11	1	1	1	1
		10	0	1	1	0

$F_8 = \dots\dots\dots$