



## 4- SIMPLIFICATION DES EXPRESSIONS BOOLÉENNES :

Une simplification est nécessaire pour diminuer le nombre et la nature des opérateurs des équations logiques et par là les équipements indispensables aux applications.

### 4.1- Simplification par la méthode algébrique :

La simplification est obtenue par calcul algébrique en utilisant les propriétés et les équations caractéristiques de l'algèbre de Boole.

**EX1 : Simplifier** les équations suivantes par la méthode algébrique :

$$S = \bar{W}.A.\bar{C} + \bar{W}.A.C + W.A.\bar{C} + W.A.C$$

$$S' = \bar{R}.\bar{C}.A + R.\bar{C}.\bar{A} + R.C.A$$

$$S'' = \bar{M}.A.S + M.A.\bar{S} + M.A.S$$

### 4.2- Simplification à partir des tableaux de Karnaugh : (ou diagrammes K)

Ces tableaux sont intéressants pour simplifier les équations logiques ayant un nombre élevé de variables d'entrées dont la forme est constituée d'un ensemble de produits ; c'est le cas d'une équation écrite à partir d'une table de vérité. (Il y a aussi des logiciels de simplification).

#### 4.2.1- Principe de simplification à partir des tableaux de Karnaugh, cas de deux variables

Équation	$S_1 = \bar{a}.\bar{b} + a.\bar{b}$	$S_2 = a.\bar{b} + a.b$	$S_3 = \bar{a}.\bar{b} + a.\bar{b} + a.b$	$S_4 = \bar{a}.\bar{b} + a.b$	$S_5 = \bar{a}.\bar{b} + \bar{a}.b + a.\bar{b} + a.b$
Karnaugh	$\begin{array}{c cc} & a & \\ \hline & 0 & 1 \\ \hline b & 0 & 1 \\ \hline & 1 & 0 \end{array}$	$\begin{array}{c cc} & a & \\ \hline & 0 & 1 \\ \hline b & 0 & 1 \\ \hline & 1 & 0 \end{array}$	$\begin{array}{c cc} & a & \\ \hline & 0 & 1 \\ \hline b & 0 & 1 \\ \hline & 1 & 0 \end{array}$	$\begin{array}{c cc} & a & \\ \hline & 0 & 1 \\ \hline b & 0 & 1 \\ \hline & 1 & 0 \end{array}$	$\begin{array}{c cc} & a & \\ \hline & 0 & 1 \\ \hline b & 0 & 1 \\ \hline & 1 & 1 \end{array}$
Équation simplifiée					

**Remarque :**

#### 4.2.2- Karnaugh à trois variables :

a	b	c	S
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

$S = \bar{a}.\bar{b}.\bar{c} + \bar{a}.\bar{b}.c + \bar{a}.b.\bar{c} + \bar{a}.b.c$

Table de Karnaugh ab

	00	01	11	10
c=0	1	1	0	0
c=1	0	0	1	1

Équation simplifiée :  $S = \bar{a}.\bar{c} + a.c$

#### 4.2.3- Karnaugh à quatre variables :

	a.b			
S <sub>1</sub>	00	01	11	10
00	0	0	1	0
01	1	1	1	1
11	0	0	1	0
10	0	0	1	0
S <sub>1</sub> =				

	a.b			
S <sub>2</sub>	00	01	11	10
00	1	1	1	1
01	1	1	1	1
11	0	0	0	0
10	0	0	0	0
S <sub>2</sub> =				

	a.b			
S <sub>3</sub>	00	01	11	10
00	1	0	1	1
01	1	0	0	0
11	0	0	0	0
10	0	1	1	0
S <sub>3</sub> =				

	a.b			
S <sub>4</sub>	00	01	11	10
00	1	1	0	0
01	1	1	0	0
11	0	0	1	1
10	0	0	1	1
S <sub>4</sub> =				

## 5- OPERATEURS, FONCTIONS OU PORTES LOGIQUES DE BASE :

Ils permettent de manipuler les variables booléennes précédentes et de réaliser les diverses opérations de l'algèbre de Boole. Mis à part les opérateurs **OUI** et **NON**, tous les autres peuvent traiter deux ou plusieurs variables d'entrées. Dans tous les cas on obtient une seule variable de sortie.

Fonction (Opérateur)	Symbole + équation		Schéma		Table de vérité															
	AFNOR	USA	Électrique	Pneumatique																
<b>OUI</b> ou <b>égalité</b> (identité)					<table border="1"> <thead> <tr><th>a</th><th>S</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td></tr> </tbody> </table> $S = a$	a	S	0	0	1	1									
a	S																			
0	0																			
1	1																			
<b>NON</b> ou <b>Négation</b> (not)					<table border="1"> <thead> <tr><th>a</th><th>S</th></tr> </thead> <tbody> <tr><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td></tr> </tbody> </table> $S = \bar{a}$	a	S	0	1	1	0									
a	S																			
0	1																			
1	0																			
<b>ET</b> ou <b>INTERSECTION</b> (conjonction) (produit logique) (and)					<table border="1"> <thead> <tr><th>a</th><th>b</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>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> $S = a.b$	a	b	S	0	0	0	0	1	0	1	0	0	1	1	1
a	b	S																		
0	0	0																		
0	1	0																		
1	0	0																		
1	1	1																		
<b>NON ET</b> (nand)					<table border="1"> <thead> <tr><th>a</th><th>b</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> $S = \overline{a.b}$	a	b	S	0	0	1	0	1	1	1	0	1	1	1	0
a	b	S																		
0	0	1																		
0	1	1																		
1	0	1																		
1	1	0																		
<b>IN</b> ou <b>INHIBITION</b>					<table border="1"> <thead> <tr><th>a</th><th>b</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>0</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table> $S = \bar{a}.b$	a	b	S	0	0	0	0	1	1	1	0	0	1	1	0
a	b	S																		
0	0	0																		
0	1	1																		
1	0	0																		
1	1	0																		
<b>OU</b> ou <b>OU INCLUSIF</b> ou <b>RÉUNION</b> (somme logique) (or)					<table border="1"> <thead> <tr><th>a</th><th>b</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>1</td></tr> </tbody> </table> $S = a + b$	a	b	S	0	0	0	0	1	1	1	0	1	1	1	1
a	b	S																		
0	0	0																		
0	1	1																		
1	0	1																		
1	1	1																		
<b>NON OU</b> (nor)					<table border="1"> <thead> <tr><th>a</th><th>b</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> $S = \overline{a + b}$	a	b	S	0	0	1	0	1	0	1	0	0	1	1	0
a	b	S																		
0	0	1																		
0	1	0																		
1	0	0																		
1	1	0																		
<b>IMP</b> ou <b>IMPLICATION</b>					<table border="1"> <thead> <tr><th>a</th><th>b</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>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table> $S = \bar{a} + b$	a	b	S	0	0	1	0	1	1	1	0	0	1	1	1
a	b	S																		
0	0	1																		
0	1	1																		
1	0	0																		
1	1	1																		
<b>OU EXCLUSIF</b> (xor) (Accepte deux entrées)					<table border="1"> <thead> <tr><th>a</th><th>b</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> $S = a \oplus b$	a	b	S	0	0	0	0	1	1	1	0	1	1	1	0
a	b	S																		
0	0	0																		
0	1	1																		
1	0	1																		
1	1	0																		
<b>ET INCLUSIF</b> ou <b>NON OU EXCLUSIF</b> (xnor)					<table border="1"> <thead> <tr><th>a</th><th>b</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> $S = a \otimes b$	a	b	S	0	0	1	0	1	0	1	0	0	1	1	1
a	b	S																		
0	0	1																		
0	1	0																		
1	0	0																		
1	1	1																		

## 6- MATÉRIALISATION DES OPERATEURS LOGIQUES : (Logigramme)

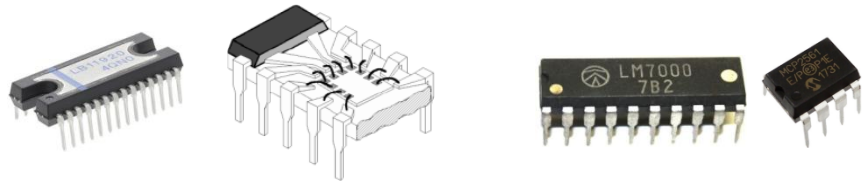
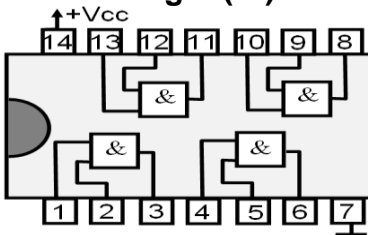
Sur le plan pratique, les opérateurs logiques existent dans des technologies très diverses. Ils sont utilisés pour réaliser concrètement les équations logiques sous forme de circuits. En électronique numérique ces opérateurs existent sous forme de **circuit intégrés** dont les prix sont très économiques.



	OU	ET
<b>Pneumatique</b>	$S = a + b$ 	$S = a \cdot b$ 
<b>Électrique</b>		
<b>Électronique</b>		

- En logique programmée, le microprocesseur effectue les opérations arithmétiques et logiques. Certaines de ces fonctions peuvent être obtenues avec d'autres circuits numériques spécialisés : les portes logiques. Ce sont des **Circuits Intégrés (CI)** permettant de réaliser des fonctions logiques fondamentales ; ils contiennent généralement 4 exemplaires identiques présents par milliers dans les ordinateurs, ils permettent d'obtenir une très grande rapidité de calcul.
- Concevoir un circuit logique, c'est relier électriquement entre eux (**câbler**) un certain nombre de composants : on parle alors de **logique câblée**.

### Le circuit intégré (CI)



Les millions de transistors des circuits intégrés sont réalisés sur un carré de matériau semi-conducteur (**le silicium**) de la taille d'un ongle. Les exemples les plus connus sont le microprocesseur et la mémoire dynamique (DRAM) qui sont au cœur des micro-ordinateurs. Ce circuit intégré nécessite 14 branches pour réaliser 4 opérations **ET** :

- Deux sont réservées à l'alimentation (une branche au potentiel +Vcc et l'autre à la masse) ;
- Les douze autres branches réalisant les 4 portes **ET** (2 entrées et une sortie pour chaque porte).

Réalisation d'opérateurs à partir des portes ET, OU et NON	
$\&_{ET} + \text{---}1\text{---}_{NON} = \&_{NAND}$	$\text{---}1\text{---} + \&_{ET} = \geq 1_{NOR}$
$\&_{NAND} + \text{---}1\text{---}_{NON} = \&_{ET}$	$\text{---}1\text{---} + \geq 1_{NOR} = \&_{ET}$
$\geq 1_{OU} + \text{---}1\text{---}_{NON} = \geq 1_{NOR}$	$\text{---}1\text{---} + \geq 1_{OU} = \&_{NAND}$
$\geq 1_{NOR} + \text{---}1\text{---}_{NON} = \geq 1_{OU}$	$\text{---}1\text{---} + \&_{NAND} = \geq 1_{OU}$
$\text{---}1\text{---} + \&_{ET} + \text{---}1\text{---} = \geq 1_{OU}$	$\text{---}1\text{---} + \geq 1_{OU} + \text{---}1\text{---} = \&_{ET}$
$\text{---}1\text{---} + \geq 1_{NOR} + \text{---}1\text{---} = \&_{NAND}$	$\text{---}1\text{---} + \&_{NAND} + \text{---}1\text{---} = \geq 1_{NOR}$

### Remarque :

N'importe quelle opérateur logique peut être réalisée à partir de la combinaison des portes

**ET, OU et NON**

Une fonction logique est appelée **fonction universelle** lorsqu'elle permet, à elle seule, de réaliser les fonctions de base **OUI**, **NON**, **ET** et **OU**.

Les fonctions **NAND**, **NOR**, **INIBITION** ou **IMPLICATION** sont des fonctions universelles.

Portes	Avec portes NAND	Avec porte NOR	Avec porte Inibition	Avec porte Implication
NAND 				
NOR 				
OUI 				
NON 				
ET 				
OU 				

Les portes **NAND** fonctionnent en logique négative (de - 12V à 0V) et les portes **NOR** fonctionnent en logique positive (de 0V à 12V).

Ne pas mélanger des NAND et des NOR dans un circuit.

En pneumatique, on utilise les portes **OUI**, **NON**, **ET**, **OU**.

En électronique, on utilise les portes **NAND**, **NOR**

**Remarque :** Dans un schéma logique, l'information se propage de la gauche vers la droite ou de haut en bas ; dans le cas contraire, il faut placer des flèches sur les accès concernés.



Opérateur NON

**Exemple :** logigramme de la fonction :  $s = \bar{a} \cdot b \cdot \bar{c}$

Opérateur ET

L'équation logique nécessite quatre opérateurs logiques : deux opérateurs **NON** et deux opérateurs **ET**.

Logigramme de S

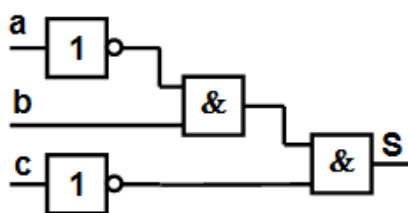
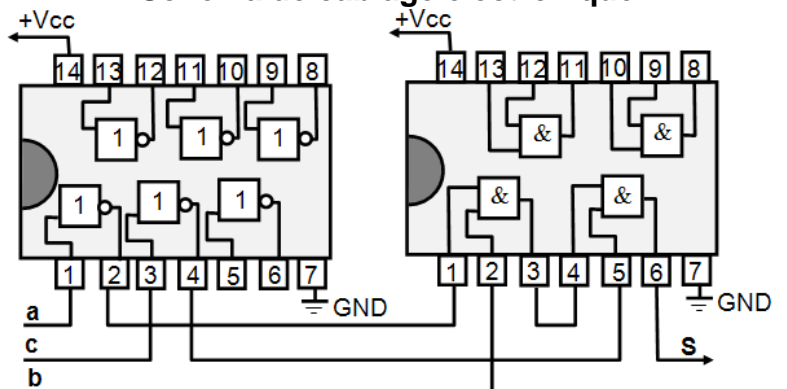


Schéma de câblage électronique



**7- CHRONOGRAMMES :**

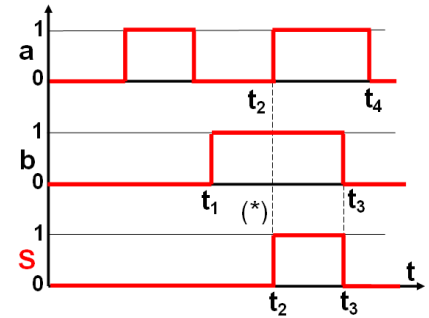
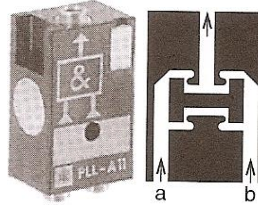
Ils représentent graphiquement l'évolution des variables (entrée et sortie) au cours du temps. Le temps est représenté par l'axe horizontal (abscisse) et l'état logique (0 ou 1) par l'axe vertical (ordonnée). Les graphes des diverses variables sont en général placés les uns au-dessous des autres avec la même échelle de temps (synchronisation).

Le chronogramme de l'opérateur logique **ET**

$$S = a \cdot b$$

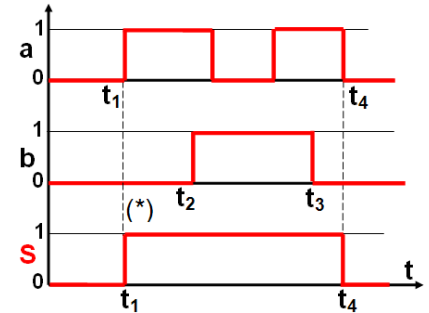
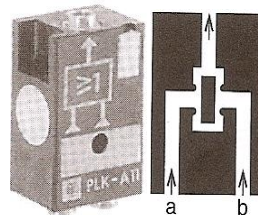
Sur le graphe, on fait varier **a** de 0 à 1 à un instant  $t_2$ , puis de 1 à 0 à un instant  $t_4$ .

Il en va de même pour **b** à des instants  $t_1$  et  $t_3$ . On visualise alors le comportement de la sortie **S** de la porte **ET** sur le diagramme temporel (appelé **chronogramme**), égale à 1 entre les instants  $t_2$  et  $t_3$ .



Le chronogramme de l'opérateur logique **OU**

$$S = a + b$$



(\*) : Le trait interrompu représente le synchronisme du passage à l'état 1 de la variable d'entrée **a** et **b** et de la variable de sortie **S**.

**Chronogrammes des opérateurs logiques :**

<p><b>OUI</b></p> <p><math>a \rightarrow [1] \rightarrow S = a</math></p>	<p><b>NON</b></p> <p><math>a \rightarrow [1] \rightarrow S = \bar{a}</math></p>	<p><b>ET</b></p> <p><math>a, b \rightarrow [ \&amp; ] \rightarrow S = a \cdot b</math></p>
<p><b>NON ET (NAND)</b></p> <p><math>a, b \rightarrow [ \&amp; ] \rightarrow S = \overline{a \cdot b}</math></p> <p><math>S = \bar{a} + \bar{b}</math></p>	<p><b>INHIBITION</b></p> <p><math>a, b \rightarrow [ \&amp; ] \rightarrow S = a \cdot \bar{b}</math></p>	<p><b>OU</b></p> <p><math>a, b \rightarrow [ \geq 1 ] \rightarrow S = a + b</math></p>
<p><b>NON OU (NOR)</b></p> <p><math>a, b \rightarrow [ \geq 1 ] \rightarrow S = \overline{a + b}</math></p> <p><math>S = \bar{a} \cdot \bar{b}</math></p>	<p><b>OU EXCLUSIF (XOR)</b></p> <p><math>a, b \rightarrow [ = 1 ] \rightarrow S = a \cdot \bar{b} + \bar{a} \cdot b</math></p> <p><math>S = a \oplus b</math></p>	<p><b>Identité logique (XNOR)</b></p> <p><math>a, b \rightarrow [ = 1 ] \rightarrow S = \overline{a \oplus b}</math></p> <p><math>S = a \cdot b + \bar{a} \cdot \bar{b}</math></p>