# 12- Algèbre de Boole

2 SM-B-; 1 STM; (Doc : élève)

## 1- ALGÈBRE DE BOOLE :

L'algèbre de Boole est l'ensemble *B* = {0 ; 1} muni de la loi **OU** notée $a + b$, de la loi **ET** notée $a . b$ et de la négation **NON** notée $\bar{a}$, où *a* et *b* sont deux variables d'entrées ou sorties.

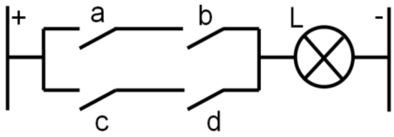
Les éléments **0** et **1** de *S* peuvent être appelés ***vrai*** ou ***faux***, ***oui*** ou ***non***.

**Remarque :**

- L'algèbre de Boole sert à simplifier les équations logiques complexes.
- Lorsque les systèmes automatisés deviennent complexes (grand nombre d'entrées et de sorties, utilisation de temporisations, mémoires, compteurs, etc.) l'algèbre de Boole n'est plus envisageable. Les automatismes sont alors définis, décrits et expliqués plus facilement à partir des diagrammes logiques (GRAFCET, etc.).

## 2- TABLE DE VÉRITÉ ET ÉQUATION LOGIQUE :

Nous avons ***4 variables d'entrées*** (a,b, c et d) pour allumer la lampe L du schéma électrique ci-contre ; donc nous aurons $2^4$ ***lignes*** ou ***états*** pour la variable de sortie L




| a | b | c | d | **L** |
|---|---|---|---|---|
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

## 3- ÉQUATIONS CARACTÉRISTIQUES DE L'ALGÈBRE DE BOOLE :

| | Propriétés du produit | Propriétés de la somme |
|---|---|---|
| Commutativité | $a . b = b . a$ | $a + b = b + a$ |
| Associativité | $a . (b . c) = (a . b) . c$ | $a + (b + c) = (a + b) + c$ |
| Distributivité | $a . (b + c) = a . b + a . c$ | $a + (b . c) = (a + b) . (a + c)$ |
| Élément neutre | $a . 1 = 1 . a = a$ | $a + 0 = 0 + a = a$ |
| Élément absorbant | $a . 0 = 0 . a = 0$ | $a + 1 = 1 + a = 1$ |
| Élément symétrique (Complémentarité) | $a . \bar{a} = 0$ | $a + \bar{a} = 1$ |
| Idempotence | $a . a = a$ | $a + a = a$ |
| Absorption | $a . (a + b) = a$ | $a + a . b = a$ |
| Multiple de complément | $a . (\bar{a} + b) = a . b = b . (a + \bar{b})$ | $a + \bar{a} . b = a + b = b + a . \bar{b}$ |
| Théorème de Morgan | $\overline{a . b} = \bar{a} + \bar{b}$ | $\overline{a + b} = \bar{a} . \bar{b}$ |
| Négation | $\bar{0} = 1 ; \bar{1} = 0 ; \bar{\bar{a}} = a$ | |

## 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 :

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

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

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

### 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 logiciel de simplification).

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

| Équation | $S_1 = \overline{a}.\overline{b} + a.\overline{b}$ | $S_2 = a.\overline{b} + a.b$ | $S_3 = \overline{a}.\overline{b} + a.\overline{b} + a.b$ | $S_4 = \overline{a}.\overline{b} + a.b$ | $S_5 = \overline{a}.\overline{b} + \overline{a}.b + a.\overline{b} + a.b$ |
|---|---|---|---|---|---|
| Karnaugh | $\begin{array}{c\|c\|c} b \backslash a & 0 & 1 \\ \hline 0 & 1 & 1 \\ 1 & 0 & 0 \end{array}$ | $\begin{array}{c\|c\|c} b \backslash a & 0 & 1 \\ \hline 0 & 0 & 1 \\ 1 & 0 & 1 \end{array}$ | $\begin{array}{c\|c\|c} b \backslash a & 0 & 1 \\ \hline 0 & 1 & 1 \\ 1 & 0 & 1 \end{array}$ | $\begin{array}{c\|c\|c} b \backslash a & 0 & 1 \\ \hline 0 & 1 & 0 \\ 1 & 0 & 1 \end{array}$ | $\begin{array}{c\|c\|c} b \backslash a & 0 & 1 \\ \hline 0 & 1 & 1 \\ 1 & 1 & 1 \end{array}$ |
| Équation simplifée | | | | | |

***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 |

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

Table de Karnaugh

| c \ ab | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 |

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

#### 4.2.3- Karnaugh à quatre variables :

| $S_1$ (c.d \ a.b) | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | 0 | 1 | 0 |
| 01 | 1 | 1 | 1 | 1 |
| 11 | 0 | 0 | 1 | 0 |
| 10 | 0 | 0 | 1 | 0 |
| $S_1$ = | | | | |

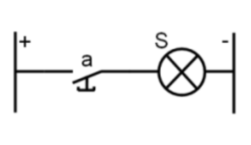
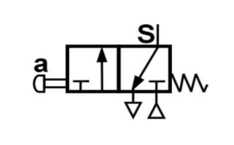
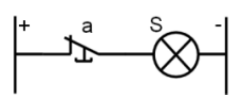
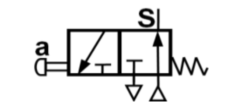
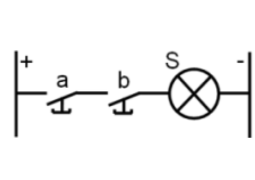
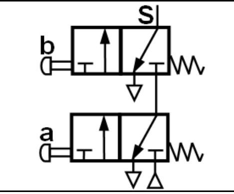
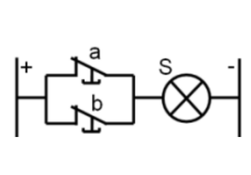
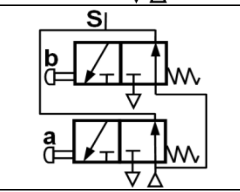
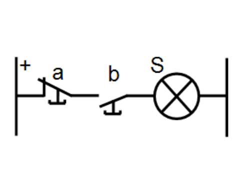
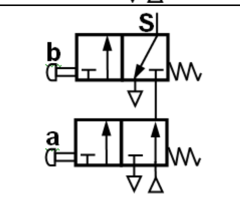
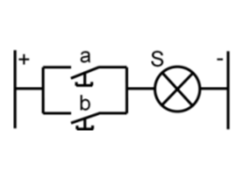
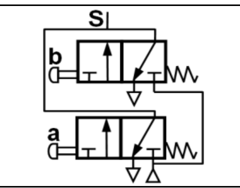
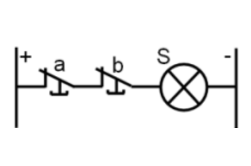
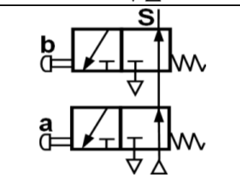
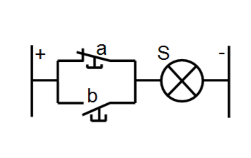
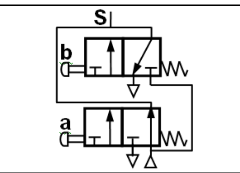
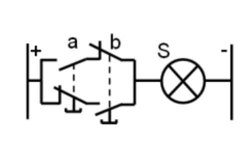
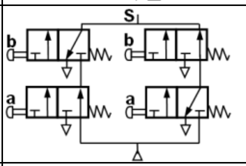
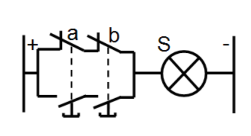
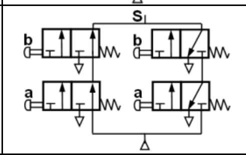
| $S_2$ (c.d \ a.b) | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 | 1 | 1 | 1 |
| 01 | 1 | 1 | 1 | 1 |
| 11 | 0 | 0 | 0 | 0 |
| 10 | 0 | 0 | 0 | 0 |
| $S_2$ = | | | | |

| $S_3$ (c.d \ a.b) | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 | 0 | 1 | 1 |
| 01 | 1 | 0 | 0 | 0 |
| 11 | 0 | 0 | 0 | 0 |
| 10 | 0 | 1 | 1 | 0 |
| $S_3$ = | | | | |

| $S_4$ (c.d \ a.b) | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 | 1 | 0 | 0 |
| 01 | 1 | 1 | 0 | 0 |
| 11 | 0 | 0 | 1 | 1 |
| 10 | 0 | 0 | 1 | 1 |
| $S_4$ = | | | | |

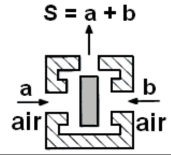
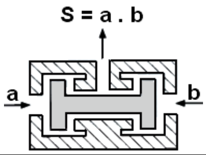
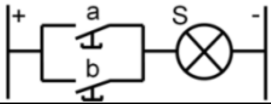
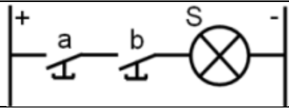
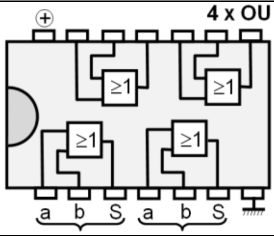
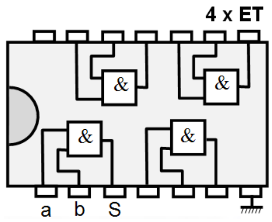
## 5- OPERATEURS, FONCTIONS OU PORTES LOGIQUES DE BASE :

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 AFNOR | Symbole + équation USA | Schéma Électrique | Schéma Pneumatique | Table de vérité |
|---|---|---|---|---|---|
| **OUI** ou **égalité** (identité) | Symbole de l'opérateur<br>a [1] S = a<br>Entrée Sortie | S = a |  |  | a S<br>0 0<br>1 1<br>S = a |
| **NON** ou **Négation** (not) | a [1] $S = \bar{a}$<br>Négation | $S = \bar{a}$ |  |  | a S<br>0 1<br>1 0<br>$S = \bar{a}$ |
| **ET** ou **INTERSECTION** (conjonction) (produit logique) (and) | [&] S = a.b | S = a.b |  |  | a b S<br>0 0 0<br>0 1 0<br>1 0 0<br>1 1 1<br>S = a.b |
| **NON ET** (nand) | [&] $S = \overline{a.b}$<br>$S = \bar{a}+\bar{b}$ | $S = \overline{a.b}$<br>$S = \bar{a}+\bar{b}$ |  |  | a b S<br>0 0 1<br>0 1 1<br>1 0 1<br>1 1 0<br>$S = \overline{a.b}$ |
| **IN** ou **INHIBITION** | [&] $S = \bar{a}.b$ | $S = \bar{a}.b$ |  |  | a b S<br>0 0 0<br>0 1 1<br>1 0 0<br>1 1 0<br>$S = \bar{a}.b$ |
| **OU** ou **OU INCLUSIF** ou **RÉUNION** (somme logique) (or) | [≥1] S = a+b | S = a+b |  |  | a b S<br>0 0 0<br>0 1 1<br>1 0 1<br>1 1 1<br>S = a + b |
| **NON OU** (nor) | [≥1] $S = \overline{a+b}$<br>$S = \bar{a}.\bar{b}$ | $S = \overline{a+b}$<br>$S = \bar{a}.\bar{b}$ |  |  | a b S<br>0 0 1<br>0 1 0<br>1 0 0<br>1 1 0<br>$S = \overline{a + b}$ |
| **IMP** ou **IMPLICATION** | [≥1] $S = \bar{a}+b$ | $S = \bar{a}+b$ |  |  | a b S<br>0 0 1<br>0 1 1<br>1 0 0<br>1 1 1<br>$S = \bar{a}+b$ |
| **OU EXCLUSIF** (xor) (Accepte deux entrées) | [=1] $S=a.\bar{b}+\bar{a}.b$<br>$S = a \oplus b$ | $S = a \oplus b$ |  |  | a b S<br>0 0 0<br>0 1 1<br>1 0 1<br>1 1 0<br>$S = a \oplus b$ |
| **ET INCLUSIF** ou **NON OU EXCLUSIF** (xnor) | [=1] $S = a \otimes b$<br>$S = a.b+\bar{a}.\bar{b}$ | $S = a \otimes b$<br>$S = a.b+\bar{a}.\bar{b}$ |  |  | a b S<br>0 0 1<br>0 1 0<br>1 0 0<br>1 1 1<br>$S = a \otimes b$ |

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

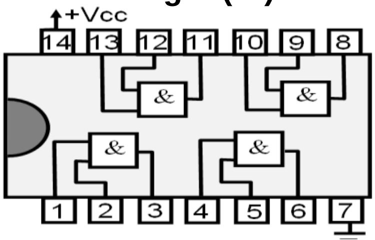
Sur le plan pratique, les opérateurs logiques existent dans des technologies très diverses.
Ils sont utilisés pour réaliser concrètement les équations logiques sous forme de circuits.
En électronique numérique ces opérateurs existent sous forme de ***circuit integrés*** dont les prix sont très économiques.

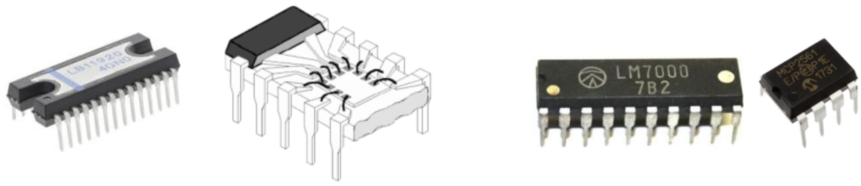
| | OU | ET |
|---|---|---|
| **Pneumatique** |  |  |
| **Électrique** |  |  |
| **Électronique** |  |  |

♦ 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)**



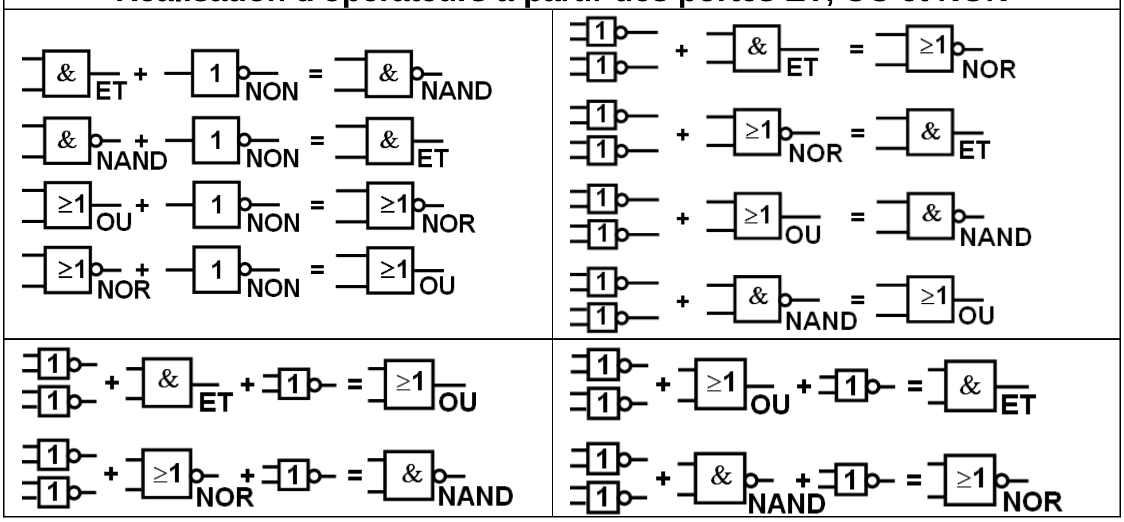




Les millions de transistors des circuits intégrés sont réalisés sur un carré de matériau semi-conducteur (**le silicium**) de la taille d'un ongle. Les exemples les plus connus sont le microprocesseur et la mémoire dynamique (DRAM) qui sont au cœur des micro-ordinateurs.
Ce circuit intégré nécessite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

**Remarque :**
N'importe quelle opérateur logique peut être réalisée à partir de la combinaison des portes
**ET**, **OU** et **NON**

**Réalisation d'opérateurs à partir des portes ET, OU et NON**

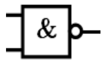
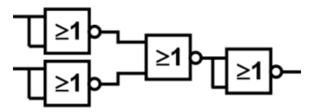
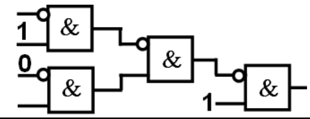
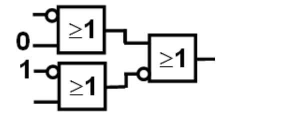
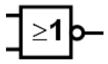
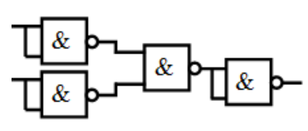
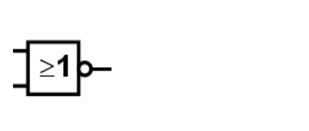
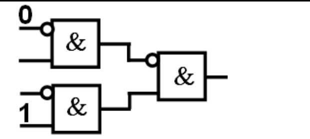
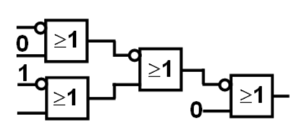
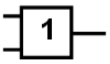
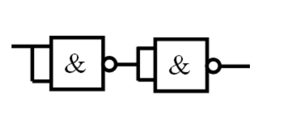
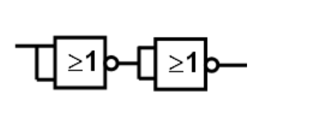
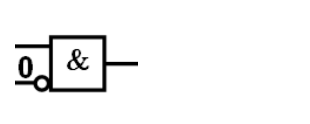
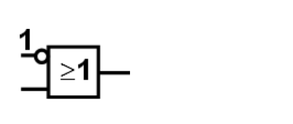
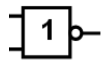
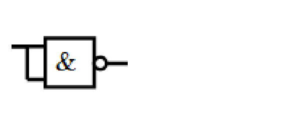
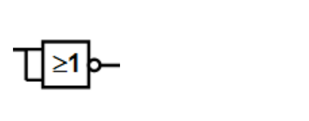
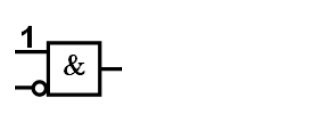
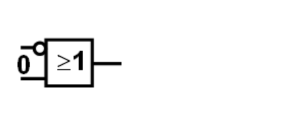
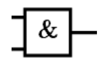
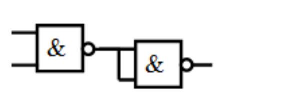
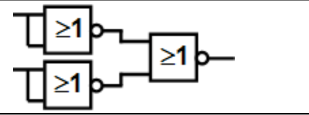
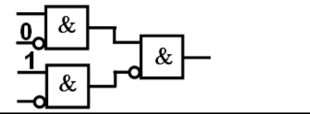
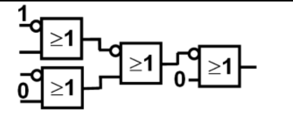
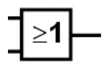
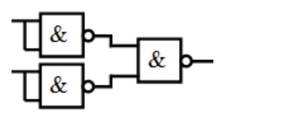
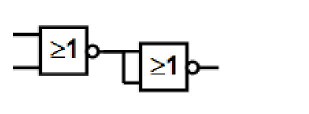
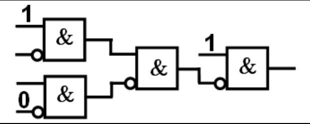
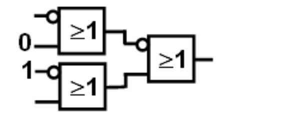


# 12- Algèbre de Boole



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 cicuit.

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

**Opérateur ET**

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

**Logigramme de S**

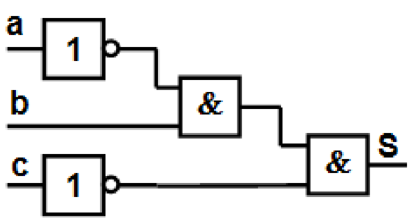
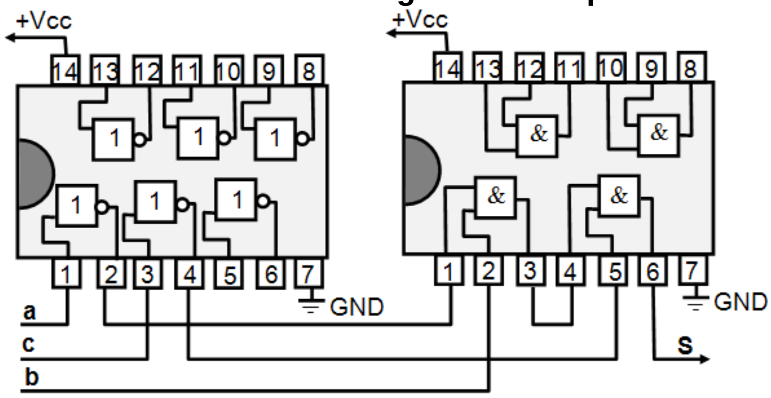



**Schéma de câblage électronique**



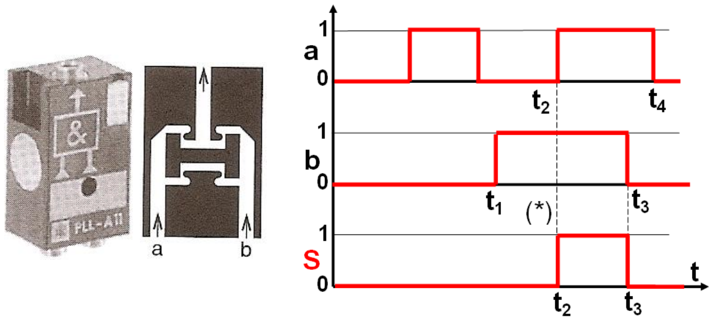
## 7- CHRONOGRAMMES :

Ils représentent graphiquement l'évolution des variables (entrée et sortie) au cours du temps.
Le temps est représenté par l'axe horisontal (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**

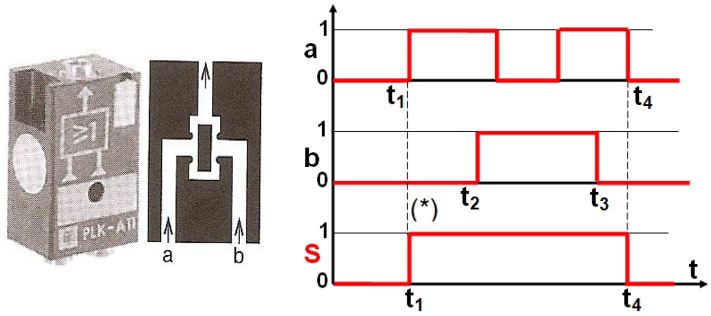
***S* = *a* . *b***

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**

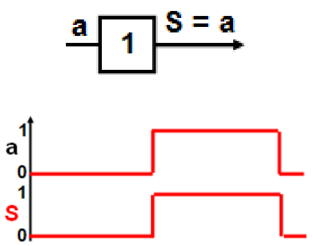
***S* = *a* + *b***



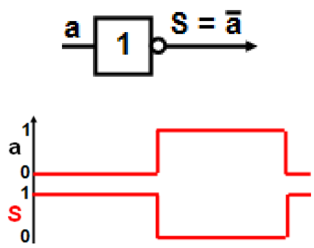

(*) : Le trait interrompu représente le synchronisme du passage à l'état **1** de la variable d'entrée **a** et **b** et de la variable de sortie **S**.

### *Chronogrammes des opérateurs logiques :*

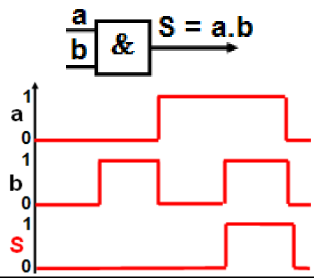
**OUI**



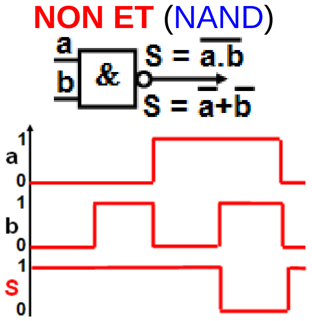

**NON**



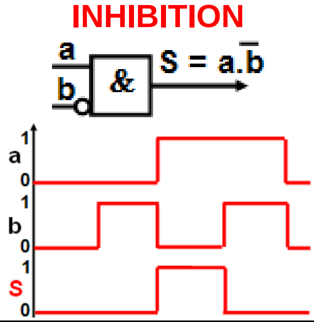

**ET**



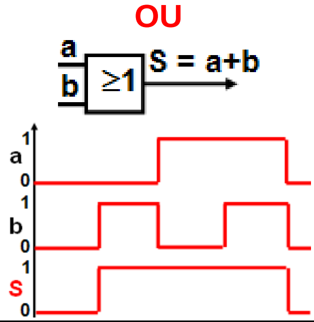

**NON ET** (NAND)



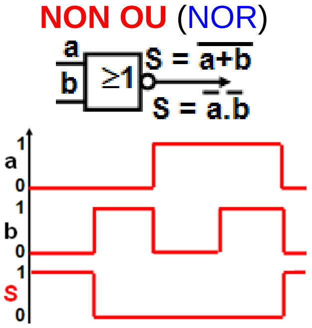

**INHIBITION**



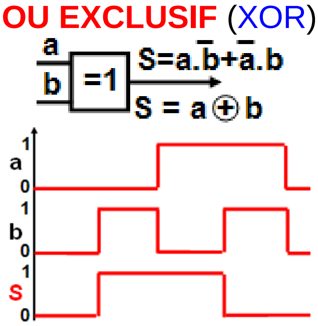

**OU**




**NON OU** (NOR)




**OU EXCLUSIF** (XOR)




**Identité logique** (XNOR)