Les fonctions logiques & l'algèbre de Boole

1 - Algèbre de Boole

Historique : Georges BOOLE, philosophe et mathématicien anglais, publia en 1854 un essai sur les raisonnements logiques portant sur les propositions auxquelles les seules réponses possibles sont **oui** ou **non**.

L'ensemble des opérations découlant de ces propositions forme une structure mathématique, donc une algèbre, appelée « algèbre de BOOLE » .

« L'algèbre de Boole » se caractérise par l'utilisation de variables ne pouvant prendre que deux états distincts. Ces deux états sont représentés par les valeurs **0** et **1**.

A noter : Cette algèbre de Boole est donc utilisée à chaque fois que, dans un système technique, on souhaite traduire, le comportement d'une grandeur physique sous forme de deux états distincts. Par exemple :

- Position de la tige d'un vérin : tige rentrée ou sortie,
- Etat d'un contact électrique : ouvert ou fermé
- Détection présence d'un objet : présent ou absent
- Etat d'un moteur : en rotation ou à l'arrêt

etc. etc...

Ces deux « états logiques » distincts se traduisent généralement par deux « niveaux de tension » distincts : présence ou absence de tension.

2 - Quelques définitions

Variable logique : grandeur, représentée par un identificateur (lettre ou nom) qui peut prendre les seules valeurs **0** ou **1**.

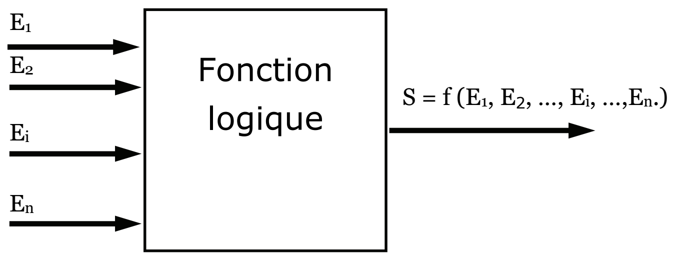
Niveau logique : En électronique, une variable logique est concrétisée par un signal électrique (tension ou courant) qui peut prendre deux **niveaux électriques** (ou niveaux logiques) :
- le niveau logique **Haut (H)** ou **High**
- le niveau logique **Bas (L)** ou **Low**

par convention :

Variable Logique (ou état logique)	Niveau logique convention positive	Niveau logique convention négative
0	**Bas (L)**	**Haut (H)**
1	**Haut (H)**	**Bas (L)**

Algèbre de BOOLE : Ensemble de variables à 2 états, de valeur, ou état "**1**" (vrai) ou "**0**" (faux) et muni d'un petit nombre d'opérateurs fondamentaux : **NON**, **ET**, **OU**.

Fonction logique de n variables binaires : groupe de variables reliées par des opérateurs logiques (NON, ET, OU)



3 - Notion de table de vérité

Une **table de vérité** est une **représentation graphique** (tableau) faisant connaître la réaction du circuit logique, c'est à dire l'état de la sortie S en fonction de toutes les combinaisons de valeurs (0 ou 1) que peuvent prendre les variables binaires d'entrées E_1, E_2, …, E_i, …,E_n.

Exemple d'une table de vérité pour une fonction logique à deux entrées $\mathbf{E_1}$ et $\mathbf{E_2}$ et une sortie **S**

$\mathbf{E_1}$	$\mathbf{E_2}$	**S**
0	0	0
0	1	1
1	0	1
1	1	0

Nombre de variables d'entrées :

Nombre de combinaisons de valeurs possibles pour les variables d'entrées :

4 - Équation logique à 1 d'une sortie

« L'équation logique à 1 d'une sortie » traduit sous forme d'une équation mathématique, le comportement de la sortie de la fonction logique.
Elle consiste en l'**écriture** d'une **équation des cas où la sortie S** de la fonction logique **est égale à « 1 ».**

Nota : - *Par abus de langage et par commodité l'expression « **équation logique à 1 de la sortie** » est souvent réduite à l'expression « équation logique »*

- « L'équation logique » peut être trouvée à partir de la table de vérité d'une fonction.

Exemple : Rechercher l'équation logique de la fonction « OU EXCLUSIF » dont la table de vérité est donnée au paragraphe précédent (c.f. § 3.1)

Ecriture des cas où S est à « 1 » :

Remarque : On verra par la suite qu'il est parfois possible de simplifier une équation logique.

5 - Les fonctions logiques de l'algèbre de Boole

En général, dans un Système Technique Industriel **S.T.I** ;-) la chaîne électronique de traitement de l'information fonctionne avec des **variables binaires**. Cette chaîne de traitement est constituée par un assemblage de **fonctions logiques** représentatives de **l'électronique numérique**.

6 - Les opérateurs logiques de l’algèbre de Boole

6.1 L’opérateur logique NON (NOT)

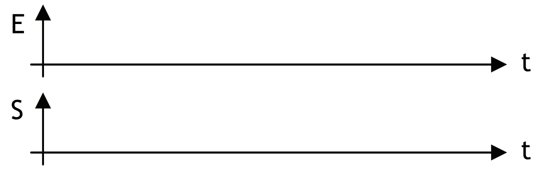
Symbole Européen :

Symbole Américain :

Table de vérité :

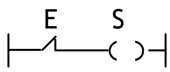
E	S

Chronogrammes d’évolution :



Equation logique à 1 de la sortie :

Schéma à contacts :



6.2 L’opérateur logique OU (OR)

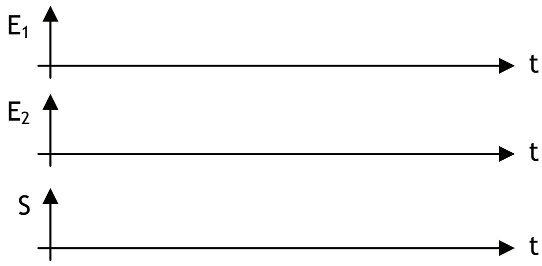
Symbole Européen :

Symbole Américain :

Table de vérité :

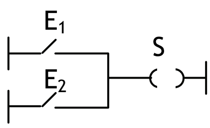
E_1	E_2	S

Chronogrammes d’évolution :



Equation logique à 1 de la sortie :

Schéma à contacts :



6.3 L'opérateur logique ET (AND)

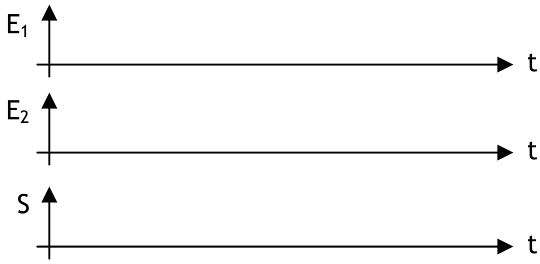
Symbole Européen :

Symbole Américain :

Table de vérité :

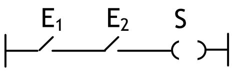
E_1	E_2	S

Chronogrammes d'évolution :



Equation logique à 1 de la sortie :

Schéma à contacts :



7 - Les autres opérateurs logiques

7.1 L'opérateur logique OU-NON* (NOR)

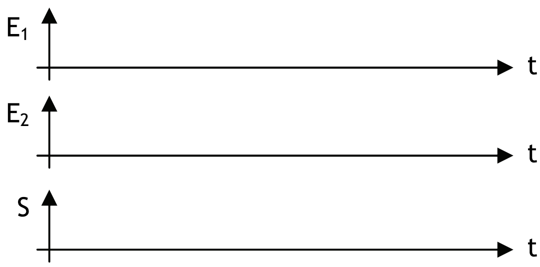
Symbole Européen :

Symbole Américain :

Table de vérité :

E_1	E_2	S

Chronogrammes d'évolution :



Equation logique à 1 de la sortie :

*** A noter :** «OU-NON» est LA bonne traduction de l'acronyme « NOR ». Car il s'agit bien de l'opérateur logique « OU » suivi de l'opérateur « NON » [et pas le contraire]. Cependant par abus de langage on trouve couramment l'expression « NON-OU » dans la littérature technique.

7.2 L'opérateur logique ET-NON* (NAND)

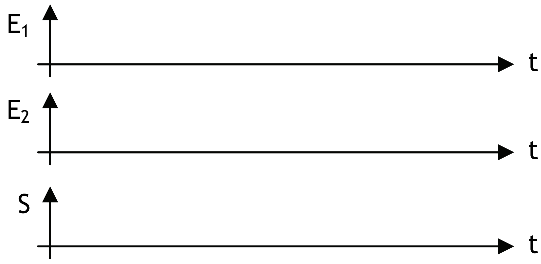
Symbole Européen :

Symbole Américain :

Table de vérité :

E_1	E_2	S

Chronogrammes d'évolution :



Equation logique à 1 de la sortie :

Schéma à contacts :

*** A noter :** «ET-NON» est LA bonne traduction de l'acronyme « NAND ». Car il s'agit bien de l'opérateur logique « ET » suivi de l'opérateur « NON » [et pas le contraire]. Cependant par abus de langage on trouve couramment l'expression « NON-ET » dans la littérature technique.

7.3 L'opérateur logique OU-EXCLUSIF (XOR)

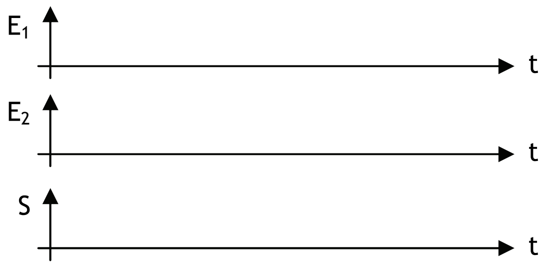
Symbole Européen :

Symbole Américain :

Table de vérité :

E_1	E_2	S

Chronogrammes d'évolution :



Equation logique à 1 de la sortie :

Schéma à contacts :

7.4 L'opérateur logique OUI

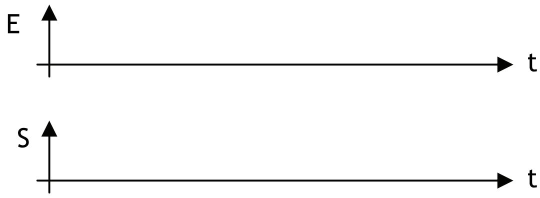
Symbole Européen :

Symbole Américain :

Table de vérité :

E	S

Chronogrammes d'évolution :



Equation logique à 1 de la sortie :

Schéma à contacts :

7.5 Représentation de fonctions logiques complexes : Le logigramme

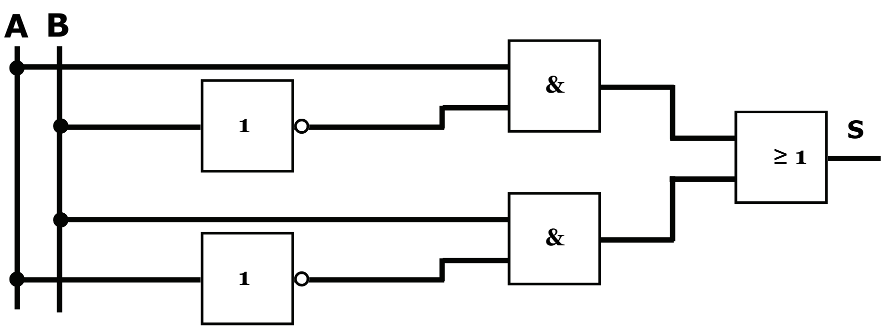
Le logigramme (ou diagramme logique) permet la représentation graphique d'une fonction logique complexe constituée d'un ensemble d'opérateurs interconnectés.
La réalisation d'un logigramme consiste en l'association organisée d'opérateurs logiques traduisant une équation logique sans préjuger de la technologie adoptée.

Exemple 1 :

Dessiner le logigramme correspondant à l'équation logique $S_1 = \overline{(A.B)+(C.D)}$

Exemple 2 :

Trouver l'équation logique correspondant au logigramme ci-dessous :



S =

Exercices :

a) Etablir le logigramme correspondant à l'équation suivante $S_2 = \overline{(\overline{A} + B) \cdot (A + \overline{B})}$

b) Etablir le logigramme correspondant à l'équation suivante $S_3 = \overline{\overline{(A \cdot B)} + (A \cdot B)}$

c) Etablir les tables de vérité correspondantes respectivement aux équations logiques de S_1, S_2 et S_3.

d) Etablir les chronogrammes correspondants respectivement aux équations logiques de S_1, S_2 et S_3.

8.1 Propriétés des opérateurs logiques (propriétés de l'algèbre de Boole)

Les propriétés suivantes permettent d'effectuer des calculs dans l'algèbre de Boole :

propriétés	ET	OU	Application
COMMUTATIVITE	A . B = B . A	A + B = B + A	Les entrées d'un opérateur logique sont interchangeables
ASSOCIATIVITE	(A.B).C = A.B.C = A.(B.C)	(A+B)+C = A + C + B = A + (B+C)	Une fonction ou à 3 entrées peut être réalisée à partir d'opérateurs à 2 entrées.
DISTRIBUTIVITE du OU par rapport au ET et du ET par rapport au OU	A + (B.C) = (A+B) . (A+C) A. (B+C) = (A.B) +(A.C)		
ELEMENT NEUTRE	A . 1 = A	A + 0 = A	
ELEMENTS PRIORITAIRES	A . 0 = 0	A + 1 = 1	
COMPLEMENTATION	$A . \overline{A} = 0$	$A + \overline{A} = 1$	
IDEMPOTENCE	A . A = A	A + A = A	

8.2 Théorèmes de De Morgan :

Historique : **Augustus De Morgan**, mathématicien et logicien britanique, fondateur avec Boole de la logique des classes et des relations. Il a formulé certaines lois du calcul des propositions.

Les 2 théorèmes de De Morgan :

$$\overline{A + B} = \overline{A} \,.\, \overline{B}$$

$$\overline{A \,.\, B} = \overline{A} + \overline{B}$$

Intérêt :
Simplifier et optimiser la conception des structures à base d'opérateurs logiques.

Exemple :

$S = \overline{A} + \overline{B}$

Exercice 1 : A l'aide des propriétés de l'algèbre de Boole, simplifier les expressions suivantes :

1/ $S = a + (b \,.\, a)$

2/ $S = a \,.\, (b + a)$

3/ $S = a + \overline{a} \,.\, b$

4/ $S = (a \,.\, b) + (a \,.\overline{b})$

Exercice 2 : Démontrer les égalités suivantes :

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

2/ $A + \overline{A} \,.\, B = A + B$

3/ $(A + B) . (\overline{A} + B) = B$

4/ $(\overline{A} + \overline{B}) + (\overline{A + B}) . C = \overline{A} + \overline{B}$

9. Conception et optimisation des systèmes à base de logique combinatoire

Définition d'un système dit à « logique combinatoire » : Un système est dit "combinatoire" lorsque qu'à une combinaison des variables binaires d'entrée correspond une (et une seule) combinaison des variables de sorties.

Note : on parle de systèmes combinatoires par opposition aux systèmes séquentiels, dans lesquels les variables de sortie dépendent à la fois des variables d'entrée et de l'état antérieur des variables de sortie.

9.1 Conception de systèmes de nature combinatoire

La réalisation d'un système combinatoire nécessite un cahier des charges dont l'énoncé permet, en détaillant chaque étape du fonctionnement, de dresser un tableau descriptif complet de tous les états binaires . Nous avons déjà vu que ce tableau s'appelle une table de vérité.

De cette table de vérité on peut tirer une expression booléenne qu'il convient de simplifier afin de réduire la complexité de la réalisation.
Il existe plusieurs méthodes d'extraction et de simplification des équations booléennes.

1ère méthode (méthode algébrique) **:**

Pour chaque variable de sortie figurant sur la table de vérité, on écrit la "somme" logique des lignes ou la variable de sortie prend la valeur 1.

note: Lorsque les états "1" sont plus nombreux que les états "0", il est avantageux d'écrire le complément de la somme logique des lignes où la variable de sortie prend la valeur 0.

Puis on simplifie l'expression obtenue en utilisant les propriétés de l'algèbre de BOOLE.

Cette méthode peut convenir pour les cas où le nombre de variables d'entrée ne dépasse pas 2 ou 3.

2ème méthode (méthode graphique) **:**

Dans le cas ou le nombre de variables devient trop important, il est plus avantageux d'utiliser une représentation graphique intitulée « Tableau de Karnaugh » permettant de trouver directement une expression simplifiée de l'équation de sortie d'une fonction logique.

9.2 Expression simplifiée d'une sortie à l'aide de la méthode du « Tableau de Karnaugh »

Définition : Outil graphique de simplification des équations logiques.

Le nombre de case du tableau est égal au nombre de combinaisons possibles pour les entrées soit :

$$\boxed{C = 2^n}$$

Avec
- C : nombre de combinaisons.
- n : nombre de variables (ou entrées).

Exemple : Avec un opérateur OU EXCLUSIF à 2 entrées, le **tableau de karnaugh** comporte nombre de case :

C =

E_2 \ E_1	0	1
0		
1		

Table de vérité « OU EXCLUSIF »

E_2	**E_1**	**S**
0	**0**	**0**
0	**1**	**1**
1	**0**	**1**
1	**1**	**0**

Remarque : 1 équation = 1 tableau de Karnaugh

Méthode pour l'obtention de l'équation simplifiée de la sortie :

1/ Groupement des cases :
Peuvent être réunies les cases adjacentes, contenant des valeurs 1 à condition que le nombre de cases du groupement soit égal à une puissance de 2 (1, 2, 4, 8, 16 …).
Remarques :
- On doit réaliser les plus grands regroupements possibles,
- Il est possible de faire des regroupements par symétrie, c'est-à-dire il est possible de faire des regroupements en regroupant les « 1 » situés de part et d'autre des deux axes de symétrie du tableau.
- Les « 1 » peuvent servir à plusieurs regroupements.
- Tous les « 1 » doivent êtres regroupés (au moins par défaut avec eux-mêmes)

Exemples :

E_3 \ $E_1 E_2$	00	01	11	10
0	*0*	*1*	*0*	*0*
1	*0*	*1*	*0*	*0*

S =

E_3 \ $E_1 E_2$	00	01	11	10
0	*0*	*1*	*1*	*0*
1	*0*	*1*	*1*	*0*

S =

E_3 \ $E_1 E_2$	00	01	11	10
0	*0*	*1*	*0*	*0*
1	*1*	*0*	*0*	*1*

S =

$E_3 E_4$ \ $E_1 E_2$	00	01	11	10
00	*0*	*1*	*0*	*0*
01	*0*	*1*	*0*	*0*
11	*0*	*1*	*0*	*0*
10	*0*	*1*	*0*	*0*

S =

$E_3 E_4$ \ $E_1 E_2$	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 =

$E_3 E_4$ \ $E_1 E_2$	00	01	11	10
00	*0*	*0*	*0*	*0*
01	*0*	*0*	*0*	*0*
11	*1*	*0*	*0*	*1*
10	*1*	*0*	*0*	*1*

S =

2/ Règle de simplification

Lorsque, dans un regroupement, une variable est présente à la fois sous la forme complémentée et non complémenté elle est éliminée.

Exemples :

$E_3 E_4$ \ $E_1 E_2$	00	01	11	10
00	*1*	*1*	*1*	*1*
01	*0*	*0*	*0*	*0*
11	*0*	*0*	*0*	*0*
10	*1*	*1*	*1*	*1*

S =

$E_3 E_4$ \ $E_1 E_2$	00	01	11	10
00	*1*	*0*	*0*	*1*
01	*1*	*0*	*0*	*1*
11	*1*	*0*	*0*	*1*
10	*1*	*0*	*0*	*1*

S =

$E_3 E_4$ \ $E_1 E_2$	00	01	11	10
00	*1*	*0*	*0*	*1*
01	*0*	*0*	*0*	*0*
11	*0*	*0*	*0*	*0*
10	*1*	*0*	*0*	*1*

S =

$E_3 E_4$ \ $E_1 E_2$	00	01	11	10
00	*0*	*1*	*0*	*0*
01	*0*	*1*	*0*	*0*
11	*1*	*1*	*1*	*1*
10	*0*	*1*	*0*	*0*

S =

$E_3 E_4$ \ $E_1 E_2$	00	01	11	10
00	*1*	*1*	*0*	*0*
01	*1*	*1*	*0*	*0*
11	*0*	*0*	*0*	*0*
10	*0*	*0*	*0*	*1*

S =

$E_3 E_4$ \ $E_1 E_2$	00	01	11	10
00	*0*	*1*	*1*	*0*
01	*0*	*1*	*1*	*0*
11	*0*	*1*	*1*	*0*
10	*0*	*1*	*1*	*0*

S =