L'ALGEBRE DE BOOLE

George Boole (1815-1864), mathématicien anglais, a développé une algèbre permettant de manipuler les propositions logiques au moyen d'équations mathématiques où les énoncés VRAI et FAUX sont représentés par les valeurs 1 et 0, tandis que les opérateurs ET et OU deviennent des opérateurs algébriques de multiplication et d'addition.

1. LES EQUATIONS LOGIQUES

L'algèbre de Boole est constituée de :

- Un ensemble E
- Deux éléments particuliers de E : **0** et **1** (correspondant respectivement à FAUX et VRAI)
- Deux opérations binaires sur E :

  Addition : (+) et **multiplication : (·)** correspondant respectivement au **OU** et **ET** logiques.
- Une opération unaire sur E : $\bar{\ }$ (barre) correspondant au **NON**logique.

  Une opération unaire est une opération qui n'a besoin que d'un seul opérateur. Par exemple : la valeur absolue, l'opposé, l'inverse…

1.1 Les postulats

1.	0.0 = 0	5.	1 + 1 = 1
2.	0.1 = 1.0 = 0	6.	1+ 0 = 0 + 1 = 1
3.	1.1 = 1	7.	0 + 0 = 0
4.	$\bar{0}$ = 1 (0barre = 1)	8.	$\bar{1}$ = 0 (1barre = 0)

Ces postulats correspondent, aux définitions des opérations logiques sur les éléments 0 et 1 De l'ensemble E. On pourra noter que dans le cas des opérations binaires, multiplication **(·)** et addition **(+)** , ces postulats correspondent aux résultats de l'arithmétique usuelle, sauf dans le cas du postulat (5) où 1+ 1=1.

De ces postulats découlent les axiomes suivants :

1.2 Les axiomes

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 + 0 = a	a.1 = a
Complémentation :	a + $\bar{a}$ = 1	a . $\bar{a}$ = 0

1.3 Principe de dualité

Chaque *axiome* et chaque postulat possède un équivalent dual, où les éléments 0 sont remplacés par des 1, les 1 par des 0, les (•)par des (+) et vice et versa.
Aussi, tout *théorème* de l'algèbre de Boole a son équivalent dual. Le théorème dual est formulé à partir du théorème de base en remplaçant les éléments 0 par des 1, les 1 par des 0 et les (•) par des (+) et vice et versa.

Par exemple, considérons le théorème suivant : $a + a = a$

La preuve de ce théorème va comme suit :

$a + a = 1.a + 1.a$	d'après l'axiome de l'élément neutre
$a + a = a . (1+ 1)$	d'après l'axiome de distributivité
$a + a = a . 1$	d'après le postulat (5)
$a + a = a$	d'après l'axiome de l'élément neutre

Aussi, on peut immédiatement déduire de ce théorème son dual, ce dernier s'exprimant comme suit : $a . a = a$

Il a suffi simplement de changer l'opérateur (+) par l'opérateur (·). Bien sûr, il est possible de démontrer ce second théorème de nouveau, mais le fait d'avoir prouvé le premier suffit pour accepter le second en se référant au principe de dualité.

La preuve de ce second théorème montre d'ailleurs la symétrie avec la démonstration précédente, le principe de dualité s'appliquant ici à chaque ligne :

$a . a = (0+ a). (0+ a)$	d'après l'axiome de l'élément neutre
$a . a = a+ (0 . 0)$	d'après l'axiome de distributivité
$a . a = a + 0$	d'après le postulat (1)
$a . a = a$	d'après l'axiome de l'élément neutre

1.4 Les théorèmes

	Forme 1	**Forme 2 (dualité)**	**Nom éventuel**
1	$\overline{\overline{a}} = a$	—	Involution
2	$a+a = a$	$a \cdot a = a$	Idempotence
3	$a+1 = 1$	$a \cdot 0 = 0$	Élément absorbant
4	$a+a \cdot b = a$	$a \cdot (a+b) = a$	Absorption
5	$\overline{a+b} = \overline{a} \cdot \overline{b}$	$\overline{a \cdot b} = \overline{a} + \overline{b}$	de Morgan
6	$a+ \overline{a} \cdot b = a+b$	$a \cdot (\overline{a} + b) = a \cdot b$	—

2. L'ADDITION BINAIRE. (D'après http://www.electronique-et-informatique.fr/Electronique-et-Informatique)

2.1 Addition de 2 chiffres binaires

Dans le système binaire, on peut représenter n'importe quel nombre comme dans le système décimal et l'on peut effectuer les quatre opérations arithmétiques élémentaires : addition, soustraction, multiplication et division.
Les trois dernières opérations peuvent être toutes ramenées à l'addition qui est donc la plus importante. Nous allons tout d'abord rappeler l'addition de deux nombres binaires de **1 bit**, nous obtenons les 4 sommes suivantes :

0 + 0 = 0   0 + 1 = 1   1 + 0 = 1   1 + 1 = 10

1er cas : les deux chiffres sont 0 et la somme est 0.

0 + 0 = 0

$$\begin{array}{rl} 0 & \leftarrow \text{1er terme} \\ +\,0 & \leftarrow \text{2ème terme} \\ \hline 0 & \leftarrow \text{Résultat ou somme} \end{array}$$

2ème et 3ème cas : un chiffre vaut 0, l'autre vaut 1 : la somme vaut 1.

0 + 1 = 1

et

1 + 0 = 1

$$\begin{array}{r} 0 \\ +\,1 \\ \hline 1 \end{array} \quad \text{ou} \quad \begin{array}{rl} 1 & \leftarrow \text{1er terme} \\ +\,0 & \leftarrow \text{2ème terme} \\ \hline 1 & \leftarrow \text{Résultat ou somme} \end{array}$$

4ème cas : Les deux chiffres valent 1 et la somme vaut 10 ($2_{(10)}$).

1 + 1 = 10

$$\begin{array}{rrl} & 1 & \leftarrow \text{1er terme} \\ & +\,1 & \leftarrow \text{2ème terme} \\ \hline \text{Retenue} \rightarrow & 10 & \leftarrow \text{Résultat ou somme} \end{array}$$

On remarque que dans les trois premiers cas, il suffit d'un seul chiffre binaire (ou bit) pour indiquer le résultat.
Dans le quatrième cas, il faut deux chiffres : celui situé le plus à droite est le **résultat** (ici 0) et l'autre est la **retenue** (ici 1).
La procédure décrite est la même que pour l'addition dans le système décimal. Elle diffère seulement par la quantité de chiffres mise en jeu : les deux chiffres binaires contre les dix décimaux.

La figure 1 montre les additions des chiffres 0 et 1 relatives aux deux systèmes.

Nous remarquons que les résultats sont les mêmes, bien que dans le système binaire il faille tenir compte de la retenue pour exprimer le résultat deux.

a)

Termes		Somme	
A	B	C*	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

b)

Termes		Somme
A	B	S
0	0	0
0	1	1
1	0	1
1	1	2

C* est l'initiale de Carry (retenue en anglais)

Fig. 1. - Addition des chiffres 0 et 1 dans les deux systèmes binaire a) et décimal b).

2.2 Circuit Additionneur pour 2 chiffres binaires

Puisque nous connaissons les règles de l'addition binaire, nous allons voir à présent comment cette opération peut être réalisée par des circuits logiques.
Il faut réaliser un circuit combinatoire (figure 2) dont les deux entrées A et B et les sorties S et C répondent à la table de vérité de la figure 1.

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

Fig. 2. - Dans un additionneur à 2 bits, la somme du bit A et du bit B donne le résultat S et la retenue C.

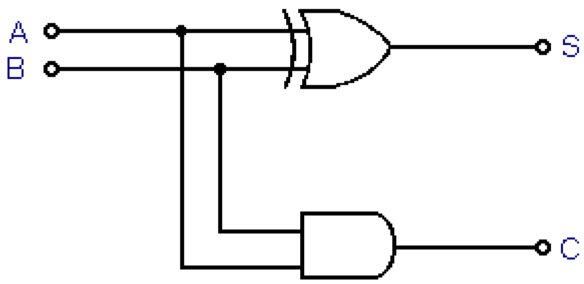
On remarque que S est à l'état 1 si une seule des entrées est à l'état 1.

Nous avons donc affaire à la fonction logique OU Exclusif, soit : $S = A \oplus B$

D'autre part, on remarque que C est à l'état 1 uniquement dans le cas où A et B sont à l'état 1.

Nous avons donc affaire à la fonction logique ET, soit : $C = A . B$

Le circuit qui effectue la somme de deux bits peut être obtenue en associant une porte OU Exclusif et une porte ET comme le montre la figure 3.



3 Fig. 3. - Circuit effectuant la somme de 2 bits A et B.

2.2 Application 1 :

Vérifier le résultat de cette association en utilisant les fonctionnalités d'excel

2.3 Circuit additionneur pour une addition à plusieurs chiffres

Le circuit additionneur examiné précédemment est en mesure d'additionner deux nombres binaires d'un seul chiffre. Pour cette raison, il est appelé demi-additionneur.

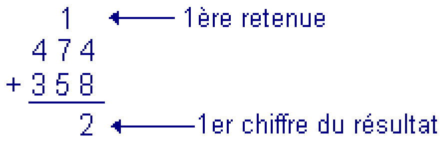
En effet, lorsque l'on doit additionner des nombres de plus d'un chiffre, il devient nécessaire de disposer de circuits qui tiennent compte de la retenue de la somme effectuée sur les chiffres de rang immédiatement inférieur.

Pour comprendre cela, nous allons examiner, comment on effectue l'addition de deux nombres décimaux, par exemple :

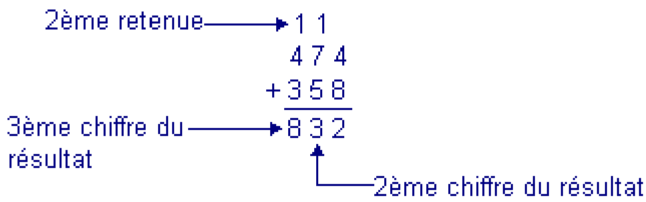
$$\begin{array}{r} 474 \\ +\ \underline{358} \\ = \end{array}$$

Cette opération s'effectue par étapes successives : on additionne d'abord les chiffres de droite, puis les suivants en ajoutant l'éventuelle retenue.

Dans un premier temps, on fait l'addition de 4 et 8 dont la somme est 12 ; on écrit le résultat 2 et on retient 1.



Dans l'étape suivante, on doit faire une addition de 3 chiffres parce qu'on doit tenir compte de la retenue (ici 1). La somme de 7 et 5 plus la retenue 1, donne 13 ; on écrit donc 3 et on retient 1.



La dernière étape est semblable à la précédente. La somme de 4 et 3 plus la retenue 1, donne 8 est cependant sans retenue.

La même procédure s'applique aussi aux nombres binaires.

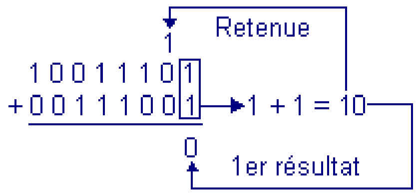
Il faut donc réaliser un circuit qui puisse additionner les deux chiffres de même rang d'un nombre binaire avec la retenue de l'étage précédent, soit trois chiffres binaires.

L'additionneur complet dispose donc de **trois** entrées, **deux** pour les termes et **une** pour la retenue.

Soit à effectuer la somme des deux nombres binaires de 8 bits suivants :

```
  10011101
+ 00111001
=
```

On part de la dernière position à droite, où se trouvent deux 1. On effectue la somme de ces deux chiffres, ce qui donne comme résultat 0 et comme retenue 1.



A l'étape suivante, on doit additionner 3 chiffres alors que la table de la figure 1 se limite à la somme de 2 chiffres.

Nous allons donc construire une table indiquant la somme de 3 chiffres.

Avec 3 chiffres, il y a 8 possibilités qui vont de 0 + 0 + 0 à 1 + 1 + 1.

Pour chacune de ces possibilités, il est facile de relever la somme.

Par exemple :

- 0 + 0 + 0 = 0 (résultat 0, retenue 0)
- 0 + 1 + 1 = 10= $2_{(10)}$ (résultat 0, retenue 1)
- 1 + 1 + 1 = 11= $3_{(10)}$ (résultat 1, retenue 1).

La table de la figure 4 résume toutes les combinaisons possibles.

Dans cette table, Ai et Bi sont les termes de rang.

Ci est la retenue relative à la somme de Ai + Bi

Ci + 1 est la retenue relative à la somme de Ai + Bi + Ci.

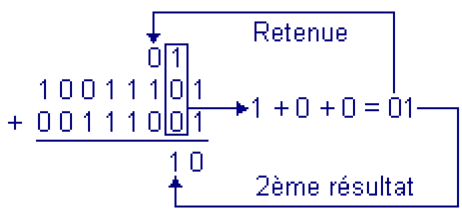
Si est le résultat de la somme Ai + Bi + Ci.

Ai	Bi	Ci	Ci + 1	Si
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

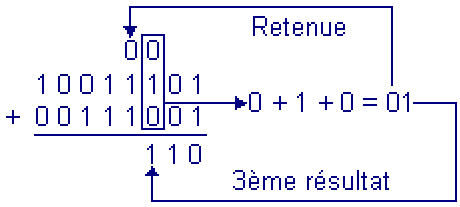
Fig. 4. - Table de la somme de 3 bits.

Revenons maintenant à la somme prise en exemple ; en utilisant la table de la figure 4, on obtient pour les termes de rang 2 :

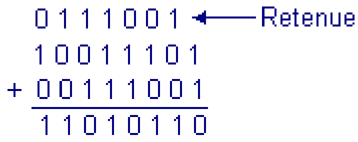
1 + 0 + 0 = 1 avec une retenue égale à 0.



Si l'on additionne les chiffres suivants, on a :



Et ainsi de suite jusqu'au résultat final :



Il faut donc réaliser un circuit qui corresponde à la table de vérité de la figure 4, on obtient le schéma de la figure 5 qui représente un additionneur complet.

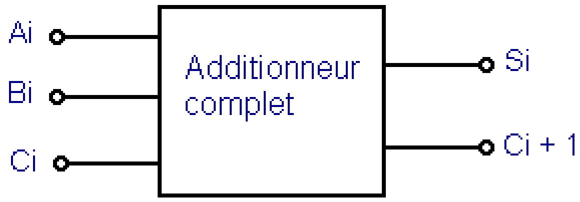


Fig. 5. - Représentation schématique d'un additionneur complet.

Le schéma logique d'un additionneur complet représenté à la figure 7.

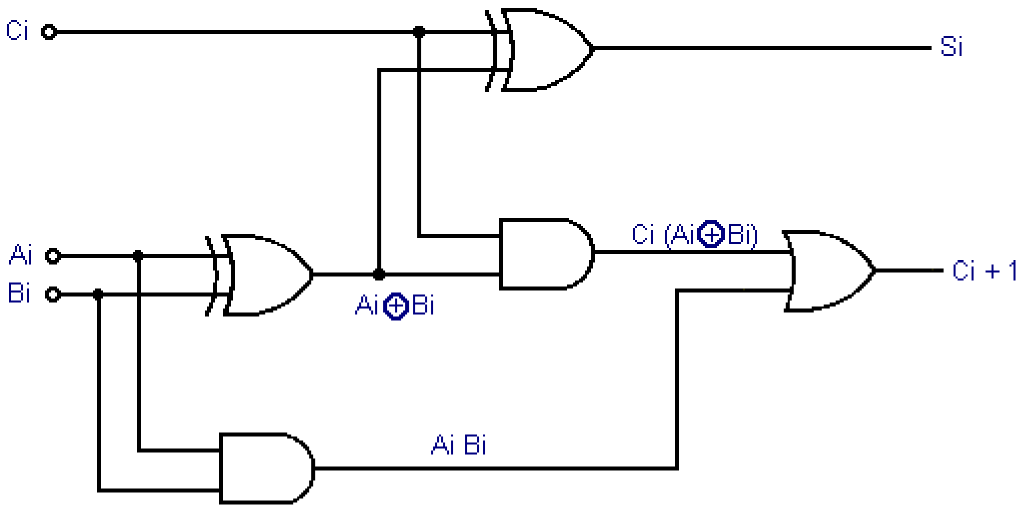


Fig. 7. - Exemple de schéma logique d'un addtionneur complet.

L'additionneur complet est le circuit de base pour effectuer la somme de nombres de plusieurs bits.

Il existe deux méthodes d'addition des nombres binaires.

La première utilise un seul additionneur complet auquel on présente les chiffres de même rang des nombres à additionner. Il s'agit de la somme en série.

La deuxième fait appel à autant d'additionneurs complets qu'il y a de chiffres dans les nombres à ajouter. Il s'agit de la somme en parallèle.

2.4 Application 2 :

Vérifier le résultat de cette association en utilisant les fonctionnalités d'excel