

2.6 Arithmétique binaire.

L'arithmétique binaire est la manière dont on mène les calculs en base 2 (système binaire).

C'est un concept essentiel de l'informatique et l'électronique. En effet, les processeurs des ordinateurs sont composés de millions de transistors (imprimés sur un circuit électronique) qui chacun ne gère que des bits 0 (" le courant ne passe pas ") et 1 (" le courant passe ").

Un calcul informatique n'est donc qu'une suite d'opération sur des paquets de 0 et de 1, appelés byte(8bits), word(16bits),double word(32bits)...etc.

Sur cette base, il est possible de réaliser toutes les opérations arithmétiques addition, soustraction, multiplication, division. La première que nous aborderons est l'addition.

2.6.1 Opération d'addition

Avant de travailler en binaire, il est intéressant de se rappeler comment l'addition se réalise en calcul écrit. Considérons comme premier exemple 133+164

```

  1 3 3  << premier naturel
+ 1 6 4  << second naturel
-----
  2 9 7

```

Pour des nombres simples comme celui repris ci-dessus, l'addition s'effectue « chiffre par chiffre ». Vous avez aussi appris qu'il faut parfois faire des reports lorsqu'une addition « chiffre par chiffre » donne un résultat qui est supérieur à 10.

L'addition avec des nombres en représentation binaire peut se faire de la même façon. Considérons l'exemple suivant :

```

  0 0 1 0  << premier nombre binaire (2 en décimal)
+ 0 1 0 1  << second nombre binaire (5 en décimal)
-----
  0 1 1 1  << 7 en décimal

```

Comme avec l'addition des naturels, il est possible d'avoir des reports lorsque l'on réalise une addition entre des nombres binaires voir l'exemple :

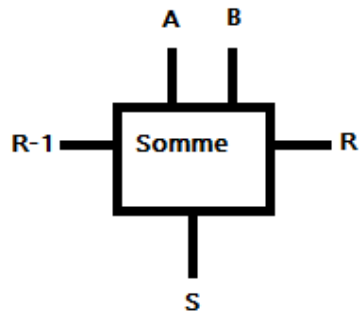
```

  1 1 1      << reports
  1 0 1 0  << premier nombre binaire (10 en décimal)
+  0 1 1 1  << second nombre binaire (7 en décimal)
-----
  1 0 0 0 1

```

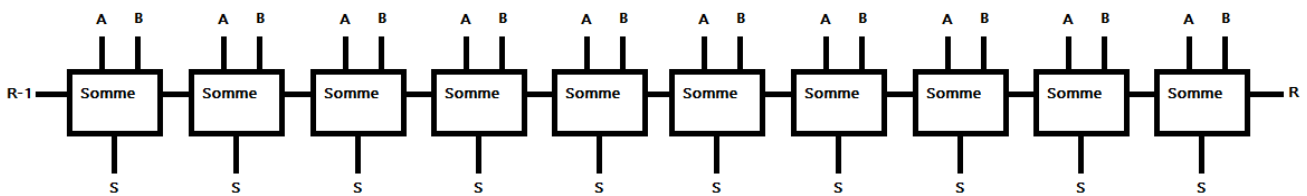
En remarque qu'il est possible de construire des fonctions booléennes qui permettent de réaliser l'opération d'addition. Considérer l'addition entre deux bits. En tout généralité, cette addition peut donner comme résultat un nombre stocké sur deux bits, le bit de poids fort (report) et le bit de poids faible (somme). Si les deux bits des nombres à additionner (A,B) et le bit de report (R-1) de l'étage précédent on peut facilement vérifier que cette addition correspond à la table de ci-dessous.

R-1	A	B	S	R
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	0	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



Cette table est appelé table de vérité d'une opération d'addition complet (full-adder en anglais). Il s'agit d'une fonction à trois entrées (a, b et r-1) et deux sorties (r et s) (en va l'implémenter les prochains chapitres).

On combine en cascade la fonction somme pour construire un additionneur qui est capable d'additionner deux nombres binaires sur n bits. La figure suivante présente l'addition de deux nombre de 10 bits.



2.6.2 Opération de soustraction

Pour l'opération de soustraction entre (A - B) revient à faire l'addition entre (A + -B) avec B coder en compliment à 2

$$01010 - 00111 \text{ (10-7)}$$

$$01010 + (11001) \text{ (11001 est le Cà2 de -7)}$$

$$\begin{array}{r}
 01010 \\
 + 11001 \\
 \hline
 \textcircled{1}00011
 \end{array}$$

Le un encerclé est un débordement de l'opération, dans le cas de complément à 2 il sera écarté.

2.6.3 Opération de multiplication

Pour la multiplication, en rappelle comment la multiplication se réalise en calcul écrit. Considérons comme exemple 13 X 5

X	0 1 1 0 1		0 1 1 0 1
	1 0 1		1 0 1
	0 1 1 0 1		0 1 1 0 1
	+ 0 0 0 0 0 .		+ 0 0 0 0 0 0
	0 1 1 0 1 . .		0 1 1 0 1 0 0
	0 1 0 0 0 0 1	ou	0 1 0 0 0 0 1

D'où (01101 X 101 = 01101X1 + 01101X10X0 +01101X100X1)

Donc la multiplication est une addition de zéro et le multiplicateur, avec à chaque étape, il faut décaler le multiplicateur d'un bit vers la gauche.

2.6.4 Opération de division

La quatrième opération arithmétique de base est la division euclidienne. Cette division prend deux arguments : un dividende et un diviseur. Elle retourne deux entiers : un quotient et un reste. la relation entre ces trois entiers est : dividende=diviseur X quotient + reste

Le principe de la division binaire est semblable à celui de la division décimal, mais aussi simple du fait que chaque quotient partiel est soit égal à 1 (division possible) ou à 0 (division impossible). Cette étape est répétée jusqu'à épuisement des bits du dividende.

2.7 Logique binaire.

En 1847, George Boole un mathématicien anglais, a défini l'algèbre de bool qui s'applique à des fonctions logiques.

2.8 Notion de base de l'algèbre de Boole

2.8.1 Variable logique

La variable logique est une variable qui peut prendre l'une de deux valeurs : vrai ou faux, 1 ou 0, aussi appelée variable booléenne.

2.8.2 Fonction logique

Une fonction logique est le résultat de la combinaison d'une ou de plusieurs variables logiques reliées entre elles par des opérations logiques. Le résultat d'une fonction logique est 0 ou 1.

2.8.3 Fonction à n variables logiques

Comme les fonctions à plusieurs variables, une fonction logique peut avoir plusieurs variables. chaque variable prend la valeur 0 ou 1, pour n variables on obtient 2^n combinaisons.

2.8.4 Table de vérité

Puisque le nombre de cas est limité, on peut représenter tous les cas dans une table, appelé table de vérité. Elle nous permet de connaître la réaction de la fonction logique.

2.8.5 Opérateurs logiques de base

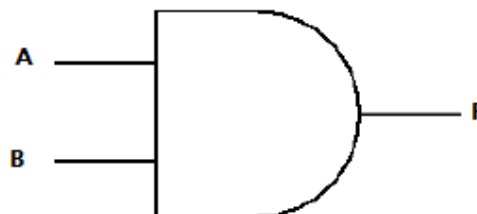
Il y a trois fonctions (opérateurs) logiques élémentaires en algèbre de Boole :

La conjonction ET logique ("multiplication"), la disjonction OU logique ("addition"), la négation NON logique ("négation").

2.8.5.1 ET logique :

L'opérateur ET (en anglais AND) est une multiplication logique symbolisée par « . », Sa table de vérité :

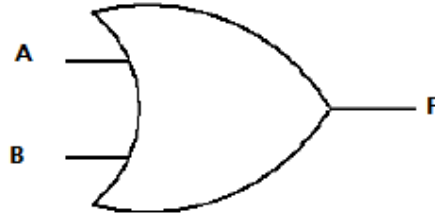
A	B	F=A and .B
0	0	0
0	1	0
1	0	0
1	1	1



2.8.5.2 OU logique :

L'opérateur OU (en anglais OR) est une addition symbolisée par « + », Sa table de vérité :

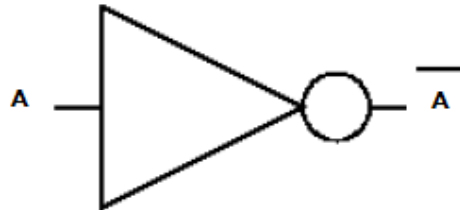
A	B	F=A or .B
0	0	0
0	1	1
1	0	1
1	1	1



2.8.5.3 NON logique :

L'opérateur NON (en anglais NOT) est une fonction unaire qui inverse la valeur en entrée. Symbolisée par « - », Sa table de vérité :

A	F=-A
0	1
1	0



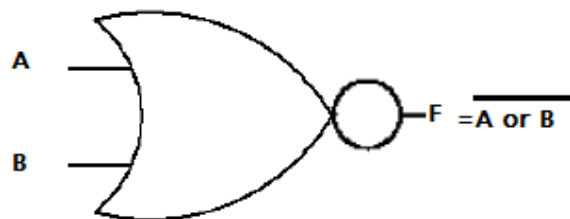
2.8.6 Autres Operateurs logiques :

Il existe d'autres operateurs logiques en algèbre de Boole

2.8.6.1 NON-OU(NOR):

L'opérateur NON-OU (en anglais NOR) est une fonction qui combine le OU logique et le NON logique, Sa table de vérité :

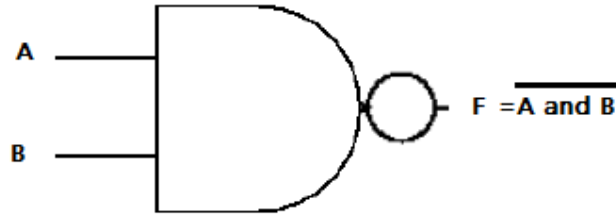
A	B	F=A Nor .B
0	0	1
0	1	0
1	0	0
1	1	0



2.8.6.2 NON-ET(NAND):

L'opérateur NON-ET (en anglais NAND) est une fonction qui combine le ET logique et le NON logique, Sa table de vérité :

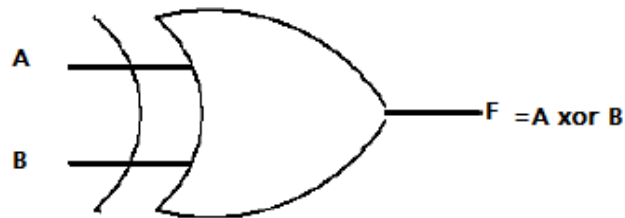
A	B	F=A Nand .B
0	0	1
0	1	1
1	0	1
1	1	0



2.8.6.3 OU-Exclusif (XOR):

L'opérateur OU exclusif (en anglais XOR) est une fonction qui combine le ET logique et OU logique, Sa table de vérité :

A	B	F=A XOR .B
0	0	0
0	1	1
1	0	1
1	1	0



2.8.6.4 NON OU-Exclusif (NXOR):

L'opérateur NON OU exclusif ou équivalence (en anglais equivalence) est une fonction qui combine le ET logique et OU logique, Sa table de vérité :

A	B	F=A NXOR .B
0	0	1
0	1	0
1	0	0
1	1	1



2.8.7 Propriétés de l'algèbre de Boole

Soient a, b et c, trois variables logiques.

2.8.7.1 Associativité

Comme avec les opérations algébriques :

$$(A+B)+C = A+(B+C)$$

$$(A.B).C = A.(B.C)$$

2.8.7.2 Commutativité

$$A+B = B+A$$

$$A.B = B.A$$

2.8.7.3 Distributivité

$$A.(B+C) = (A.B)+(A.C)$$

$$A+(B.C) = (A+B).(A+C)$$

2.8.7.4 Élément neutre

$$A.1=A$$

$$A+0=A$$

2.8.7.5 Élément absorbant

$$A.0=0$$

$$A+1=1$$

2.8.7.6 Involution

$$\text{Not}(\text{Not}(A))=A$$

2.8.7.7 Complémentation

$$A.\text{Not}(A)=0$$

$$A+\text{Not}(A)=1$$

2.8.7.8 Idempotence

$$A.A.A.A.A.....A=A$$

$$A+A+A+A++++A=A$$

2.8.7.9 Théorème de MORGANE

Le compliment de la somme logique de deux variables est égale au produit des compléments de ces deux variables : $\text{Not}(A \text{ or } B) = \text{Not}(A) \text{ and } \text{Not}(B)$

Le compliment de produit logique de deux variables est égale à la somme des compléments de ces deux variables : $\text{Not}(A \text{ and } B) = \text{Not}(A) \text{ or } \text{Not}(B)$

2.8.7.10 Forme canonique :

On a deux expressions logiques à partir table de vérité

2.8.7.10.1 Première Forme canonique

La Première forme canonique est une forme disjonctive dite aussi SOP (Sum Of Products) somme des mintermes, en prenant compte que les « 1 » de la fonction f

Un minterme est le produit logique des variables de la même ligne de la table de vérité.

2.8.7.10.2 Deuxième Forme canonique

La Deuxième Forme canonique est une forme conjonctive dite aussi POS (Product Of Sums) produit des maxtermes, en prenant compte que les « 0 » de la fonction f

Un maxterme est la somme logique des variables sous forme inversée de la même ligne de la table de vérité.

a	b	c	F(a,b,c)
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

1^{er} forme

SOP : $f(a, b, c) = \bar{a}\bar{b}c + \bar{a}b\bar{c} + a\bar{b}c + ab\bar{c} + abc$

2^{ème} forme

POS : $f(a, b, c) = (a + b + c) \cdot (a + \bar{b} + \bar{c}) \cdot (\bar{a} + b + c)$