1 Algèbre de Boole

1.1 Définition

Soit un ensemble $\mathcal{A} = \{0, 1\}$ pour lequel on a $0 \leq 1$. On définit alors les opérations suivantes sur $\mathcal{A}$:
- l'addition : $a + b = max(a, b)$
- la multiplication : $a \cdot b = min(a, b)$
- la complémentation : $\bar{0} = 1$ et $\bar{1} = 0$

Le résultat des opération $+$ et $\cdot$ est le suivant :

a	b	$a+b$	$a \cdot b$
0	0	0	0
0	1	1	0
1	0	1	0
1	1	1	1

Le n-uplet $(\mathcal{A}, +, \cdot, ^{-})$ est appelé *algèbre de Boole* s'il respecte les axiomes suivants :

1. l'addition et la multiplication sont commutatives et associatives :

$$\begin{aligned}
&a + b = b + a \\
&a \cdot b = b \cdot a \\
&(a + b) + c = a + (b + c) \\
&(a \cdot b) \cdot c = a \cdot (b \cdot c)
\end{aligned}$$

2. 0 est élément neutre pour l'addition et 1 est élément neutre pour la multiplication :

$$\begin{aligned}
&0 + a = a \\
&1 \cdot a = a
\end{aligned}$$

3. l'addition et la multiplication sont distributives l'une par rapport à l'autre :

$$\begin{aligned}
&(a + b) \cdot c = a \cdot c + b \cdot c \\
&(a \cdot b) + c = (a + c) \cdot (b + c)
\end{aligned}$$

4. la complémentation est telle que $\bar{\bar{a}} = a$ et vérifie les propriétés suivantes :

$$\begin{aligned}
&a + \bar{a} = 1 \\
&a \cdot \bar{a} = 0
\end{aligned}$$

1.2 Fonction booléenne, table de vérité

On appelle fonction *booléenne* une application de $\mathcal{A}^n$ dans $\mathcal{A}$:

$$(x_1, x_2, \cdots, x_n) \longrightarrow f(x_1, x_2, \cdots, x_n)$$

La manière la plus simple de définir une fonction boolénne est de donner la valeur pour tous les n-uplets de valeurs $(x_1, x_2, \cdots, x_n)$ grâce à une table de vérité. Ainsi pour une fonction de n variables on donne les 2^n valeurs définissant la fonction.

Exemple 1 - Soit la fonction f_1 de 3 variables A, B, C définie par :

A	B	C	f_1
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

A partir de la table de la table de vérité d'une fonction on est capable de donner une expression de la fonction sous forme d'une somme de monomes : il suffit d'exprimer les monomes pour lesquels la fonction est égale à 1. Avec l'exemple précédent on obtient :

$$f_1(A,B,C) = \bar{A}.\bar{B}.\bar{C} + \bar{A}.B.C + A.B.\bar{C} + A.B.C$$

Dans le cas particulier des fonctions à 2 variables $f(A,B)$, on peut définir 16 fonctions différentes dont certaines ont été baptisées :

A	B	or	and	xor	nor	nand
0	0	0	0	0	1	1
0	1	1	0	1	0	1
1	0	1	0	1	0	1
1	1	1	1	0	0	0

Il existe un moyen plus simple et plus rapide de décrire la table de vérité d'une fonction booléenne en indiquant les lignes de la table de vérité qui comportent des 1. Ainsi la fonction f_1 vue précédemment est décrite par : $f_1 = (3,4,5,6,7)$ ou encore par $f_1 = (3-7)$.

2 Simplification des fonctions booléennes

2.1 Règles de simplification algèbriques

Deux fonctions booléennes sont dites identiques si elles possèdent la même table de vérité. Cette propriété nous permet d'établir un certain nombre d'identité et de règles de simplification :

Loi		Forme +		Forme $\cdot$
élément neutre	(1)	$X+0=X$	(2)	$X\cdot 1=X$
d'idempotence	(3)	$X+X=X$	(4)	$X\cdot X=X$
d'inversion	(5)	$X+\overline{X}=1$	(6)	$X\cdot\overline{X}=0$
d'absorption	(7)	$X+XY=X$	(8)	$X\cdot(X+Y)=X$
de De Morgan	(9)	$\overline{X+Y}=\overline{X}\cdot\overline{Y}$	(10)	$\overline{X\cdot Y}=\overline{X}+\overline{Y}$
de commutativité	(11)	$X+Y=Y+X$	(12)	$X\cdot Y=Y\cdot X$
d'associativité	(13)	$X+(Y+Z)=(X+Y)+Z$	(14)	$X(YZ)=(XY)Z$
de distributivité	(15)	$X(Y+Z)=XY+XZ$	(16)	$X+YZ=(X+Y)(X+Z)$

L'algèbre de Boole est un outil qui permet de simplifier les circuits électroniques. Considérons par exemple la fonction :

$$F = \overline{X}YZ + \overline{X}Y\overline{Z} + XZ$$

Cette fonction peut être réécrite sous une forme simplifiée en :

$$\begin{aligned} F &= \overline{X}YZ + \overline{X}Y\overline{Z} + XZ \\ &= \overline{X}Y(Z+\overline{Z}) + XZ \\ &= \overline{X}Y\cdot 1 + XZ \\ &= \overline{X}Y + XZ \end{aligned}$$

La fonction est ainsi réduite à deux termes et est plus facile à concevoir sous forme de schéma électronique. On utilise également d'autres formules de simplification parmi lesquelles :

$$\begin{array}{lrcl} (17) & X+XY & = & X \\ (18) & XY+X\overline{Y} & = & X \\ (19) & X+\overline{X}Y & = & X+Y \\ (20) & XY+\overline{X}Z+YZ & = & XY+\overline{X}Z \end{array}$$

2.2 Méthode des tableaux de Karnaugh

La méthode des tableaux de Karnaugh est un procédé de simplification visuel pratique qui ne s'applique qu'à des fonctions booléennes composées au maximum de 6 variables. Elle consiste à représenter sous une forme particulière la table de vérité de la fonction à simplifier ce qui permet de manipuler simplement les différentes formes polynomiales d'une fonction. La simplification se réalise en deux étapes :

1. création d'un tableau de Karnaugh de la fonction à simplifier,
2. simplification de l'expression par réunion de 2^n-uplets adjacents, c'est à dire par groupe de 1,2,4,8 ou 16.

2.2.1 Création d'un tableau de Karnaugh

Considérons la formule suivante $F = \overline{A}B\overline{C} + AB\overline{C} + \overline{A}BC$. Le tableau de Karnaugh de cette fonction est représenté figure 1.

	$\overline{A}\overline{B}$	$\overline{A}B$	AB	$A\overline{B}$
$\overline{C}$		1	1	
C		1		

FIG. 1 – Tableau de Karnaugh de $F = \overline{A}B\overline{C} + AB\overline{C} + \overline{A}BC$

2.2.2 Simplification par réunion des 2^n-uplets

Pour simplifier la fonction il suffit de réunir les 2^n-uplets de valeurs adjacentes. On construit ainsi une nouvelle expression de la fonction en donnant pour chaque 2^n-uplet les valeurs des variables booléennes qui ne changent pas. Pour notre exemple, on obtient :

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

2.3 Représentation des portes logiques

Les portes logiques sont généralement représentées par des symboles composés d'une ou plusieurs entrées et d'une sortie. Pour relier deux portes il suffit de relier la sortie de l'une á l'une des entrées de la seconde.

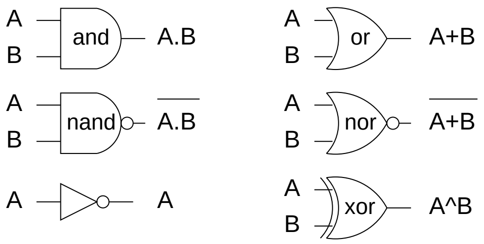

FIG. 2 – Représentation des portes logiques

La fonction précédente $F = \overline{X}YZ + \overline{X}Y\overline{Z} + XZ$ est représentée figure 3.

2.4 Universalité des portes NAND et NOR

Les expressions booléennes se résument à des combinaisons entre trois types de portes couramment utilisées : NOT, AND, et OR. On remarquera que ces trois portes sont modélisables en fonction d'un seul type de porte : la porte NAND.

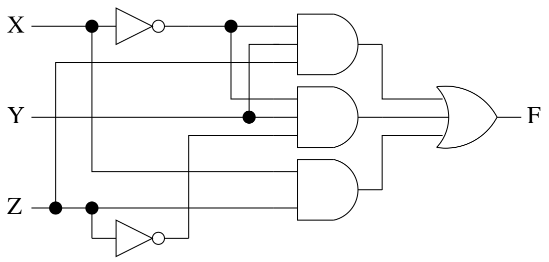

FIG. 3 – Conception de la fonction F

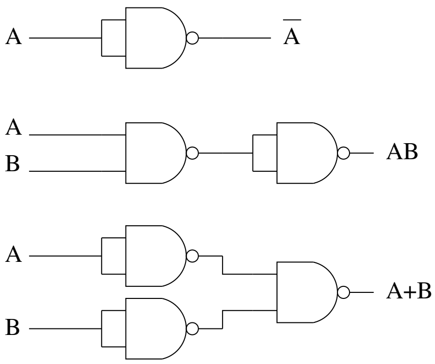

FIG. 4 – Portes NOT, AND et OR en fonction de portes NAND

$$\overline{A} = \overline{\overline{A} \cdot \overline{A}}$$

$$A \cdot B = \overline{\overline{AB} \cdot \overline{AB}}$$

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

On peut également écrire les portes NOT, AND et OR en fonction de portes NOR (cf. figure 5).

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

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

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

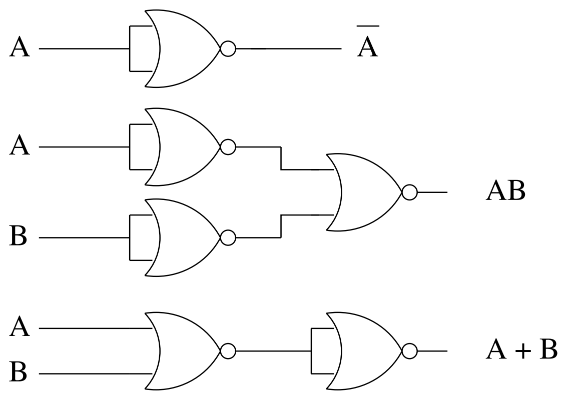

FIG. 5 – Portes NOT, AND et OR en fonction de portes NOR