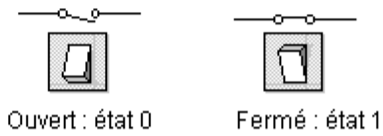
CHAPITRE 2 : ALGEBRE DE BOOLE ET SIMPLIFICATION DES FONCTIONS LOGIQUES

2.1 Introduction

L'algèbre de Boole, ou calcul booléen, lancée en 1854 par le mathématicien britannique George Boole, est une branche de mathématiques qui s'intéresse aux opérations et aux fonctions sur les variables logiques. Le binaire permet de représenter facilement l'état logique d'un système technique ou de ses entrées-sorties. C'est une logique à ***deux états***.

- Une lampe est allumée ou éteinte.
- Un interrupteur est ouvert ou fermé.
- Une tension est élevée ou faible.
- Une pression est présente ou non.

Exemple : cas d'un interrupteur

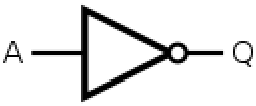


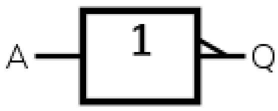
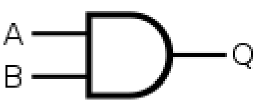
2.2 Fonctions logiques et tables de vérité

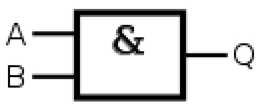
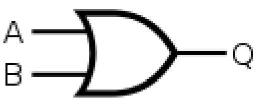
Une table de vérité (des fois appelée fonction de vérité) est une table mathématique utilisée en logique classique pour représenter des expressions logiques et calculer la valeur de leur fonction par rapport aux arguments fonctionnels (combinaison des variables logiques).

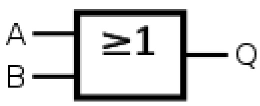
Ils sont généralement utilisés en mathématiques (logique propositionnelle), en électronique (porte logique) et en informatique (tests) selon un code binaire d'entrée (1 / 0, vrai / faux, allumé / éteint, etc.). La sortie représentée sous forme de colonne, formulée sous forme d'état binaire, est le résultat des états d'entrées.

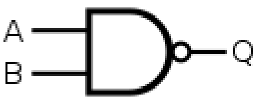
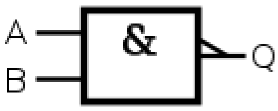
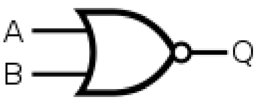
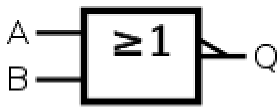
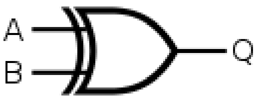
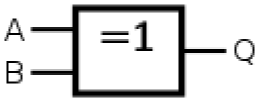
Tableau 2.1: Portes logiques et leurs tables de vérités.

<table>
<tr><th>Norme Américaine (ANSI)</th><th>Norme IEC</th><th>Opération Booléenne</th><th>Table de vérité</th></tr>
<tr><td>

NOT (NON)</td><td>
</td><td>$Q = \overline{A}$</td><td><table><tr><th>Entrée</th><th>Sortie</th></tr><tr><td>A</td><td>Q</td></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table></td></tr>
<tr><td>

AND (ET)</td><td>
</td><td>$Q = A.B$</td><td><table><tr><th colspan="2">Entrées</th><th>Sortie</th></tr><tr><td>A</td><td>B</td><td>Q</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table></td></tr>
<tr><td>

OR (OU)</td><td>
</td><td>$Q = A + B$</td><td><table><tr><th colspan="2">Entrées</th><th>Sortie</th></tr><tr><td>A</td><td>B</td><td>Q</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table></td></tr>
</table>

<table>
<tr>
<td>
NAND (NON-ET)</td>
<td></td>
<td>$Q = \overline{A.B}$</td>
<td><table><tr><th colspan="2">Entrées</th><th>Sortie</th></tr><tr><td>A</td><td>B</td><td>Q</td></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table></td>
</tr>
<tr>
<td>
NOR (NON-OU)</td>
<td></td>
<td>$Q = \overline{A + B}$</td>
<td><table><tr><th colspan="2">Entrées</th><th>Sortie</th></tr><tr><td>A</td><td>B</td><td>Q</td></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table></td>
</tr>
<tr>
<td>
XOR (OU-exclusif)</td>
<td></td>
<td>$Q = A \oplus B$

$Q = A\,\overline{B} + B\,\overline{A}$</td>
<td><table><tr><th colspan="2">Entrées</th><th>Sortie</th></tr><tr><td>A</td><td>B</td><td>Q</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table></td>
</tr>
</table>

2.3 Propriétés de l'algèbre de Boole

a) Commutativité

A + B = B + A

A * B = B * A

b) Associativité

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

(A * B) * C= A *(B * C)

c) Distributivité

A * (B + C)=(A * B) + (A * C)

A + (B * C)=(A + B) * (A + C)

d) Idempotence

A * A= A

A + A= A

e) Complémentarité

$A * \overline{A} = 0$

$A + \overline{A} = 1$

f) Double négation

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

g) Elément neutre

A + 0 = A 0 élément neutre du OU

A * 1 = A 1 élément neutre du ET

h) Absorption

$A * 0 = 0$

$A + 1 = 1$

i) **Théorème de De Morgan**

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

j) **Théorème de consensus**

$$AB + \overline{A}C + BC = AB + \overline{A}C$$
$$(A+B)(\overline{A}+C)(B+C) = (A+B)(\overline{A}+C)$$

k) **Dualité**

Deux expressions logiques sont équivalentes par dualité si on peut obtenir l'une en permutant dans l'autre, les "OU" par des "ET", les "ET" par des "OU", les "0" par des "1 et "les "1" par des "0".

Exemple :

A + 0 = A l'expression duale équivalente A * 1 = A (élément neutre).

A * (A+B) l'expression duale équivalente A + (A * B) = A (distributivité).

2.4 Fonction logique

Une fonction booléenne peut être représentée par :

- Table de vérité
- Expression
- Table de Karnaugh
- Logigramme
- Chronogramme

2.4.1. Représentation par table de vérité

Une fonction à « *n* » variables présente 2^n résultats (sorties). Ainsi, elle sera représentée par une table de vérité de 2^n lignes. Chaque ligne représente la valeur de la fonction pour une combinaison binaire d'entrées donnée.

Exemple: S=*f* (A, B, C) une fonction booléenne qui prend « 0 » si A et B sont égaux, et « 1 » ailleurs.

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

2.4.2 Représentation par expression

Une fonction logique exprimée sous une forme canonique est constituée d'un ou plusieurs termes canoniques.

2.4.1 Terme canonique

Un terme algébrique est appelé canonique s'il contient une occurrence de chacune des variables impliquées dans la forme. (ex. les occurrences de la variable A sont A ou $\overline{A}$).

2.4.2 Forme canonique

La forme canonique d'une fonction booléenne peut être exprimée de l'une de deux façons suivantes :

- ✓ ***La somme des produits (mintermes) :***

  On utilise le complément de la variable aux bits où on a un 0 dans la table de vérité, ensuite on effectue la somme des mintermes où la fonction est égale à « **1** ».

- ✓ ***Le produit des sommes (maxtermes) :***

  On utilise le complément lorsqu'il y a un 1 dans la table de vérité, ensuite on effectue le produit des maxtermes où la fonction est égale à « **0** ».

Exemple 1 :

Considérons l'exemple précédent, la table suivante montre les mintermes et les maxtermes pour chacune des combinaisons d'entrée.

Table 2.2 : Mintermes et maxtermes à 3 bits.

A	B	C	S	Minterme	Maxterme
0	0	0	0	$m_0(\bar{A}\,\bar{B}\,\bar{C})$	$M_0(A+B+C)$
0	0	1	0	$m_1(\bar{A}\,\bar{B}\,C)$	$M_1(A+B+\bar{C})$
0	1	0	1	$m_2(\bar{A}\,B\,\bar{C})$	$M_2(A+\bar{B}+C)$
0	1	1	1	$m_3(\bar{A}\,B\,C)$	$M_3(A+\bar{B}+\bar{C})$
1	0	0	1	$m_4(A\,\bar{B}\,\bar{C})$	$M_4(\bar{A}+B+C)$
1	0	1	1	$m_5(A\,\bar{B}\,C)$	$M_5(\bar{A}+B+\bar{C})$
1	1	0	0	$m_6(A\,B\,\bar{C})$	$M_6(\bar{A}+\bar{B}+C)$
1	1	1	0	$m_7(A\,B\,C)$	$M_7(\bar{A}+\bar{B}+\bar{C})$

- La forme canonique de sortie utilisant les mintermes est :

  $S = m_2 + m_3 + m_4 + m_5 = \bar{A}\,B\,\bar{C} + \bar{A}\,B\,C + A\,\bar{B}\,\bar{C} + A\,\bar{B}\,C$

- La forme canonique de sortie utilisant les maxtermes est :

  $S = M_0 + M_1 + M_6 + M_7 = (A+B+C)(A+B+\bar{C})(\bar{A}+\bar{B}+C)(\bar{A}+\bar{B}+\bar{C})$

Exemple 2:

Soit la fonction $F(A,B,C) = A + \bar{B}C$, cette fonction peut s'écrire sous la première forme canonique (somme des mintermes) comme suit :

Chaque terme doit contenir une occurrence de chaque variable d'entrée :

$$
\begin{aligned}
F(A,B,C) &= A + \bar{B}C \\
&= A(B+\bar{B})(C+\bar{C}) + (A+\bar{A})\bar{B}C \\
&= (AB + A\bar{B})(C+\bar{C}) + (A+\bar{A})\bar{B}C \\
&= AB(C+\bar{C}) + A\bar{B}(C+\bar{C}) + (A+\bar{A})\bar{B}C \\
&= ABC + AB\bar{C} + (A\bar{B}C) + A\bar{B}\bar{C} + (A\bar{B}C) + \bar{A}\bar{B}C \\
&= ABC + AB\bar{C} + A\bar{B}C + A\bar{B}\bar{C} + \bar{A}\bar{B}C \\
&= m_7 + m_6 + m_5 + m_4 + m_1
\end{aligned}
$$

Maintenant on va tenter d'écrire cette fonction sous la deuxième forme canonique (produit des maxtermes):

$$F(A,B,C) = A + \overline{B}C$$
$$= (A + \overline{B})(A + C)$$
$$= (A + \overline{B} + 0)(A + C + 0)$$

On remplace le premier 0 par le terme manquant et son complément $C\overline{C}$ et le deuxième 0 par $B\overline{B}$, ce qui donne:

$$F(A,B,C) = (A + \overline{B} + \overline{C}C)(A + C + B\overline{B})$$
$$= (\mathbf{A} + \overline{\mathbf{B}} + \mathbf{C})(A + \overline{B} + \overline{C})(A + C + B)(\mathbf{A} + \mathbf{C} + \overline{\mathbf{B}})$$
$$= M_2\, M_3\, M_0$$

2.4.3 Représentation par table de Karnaugh

Lorsque le nombre de variables croit, le nombre de lignes croit aussi, (*n* variables → 2^n lignes). On fait recours à la table de Karnaugh qui va permettre d'effectuer des simplifications beaucoup plus rapides (sans manipuler de longues équations).

- Sur les lignes et colonnes, on place l'état des variables d'entrée codées en binaire réfléchi (code Gray).
- Dans chacune des cases, on met l'état de la sortie pour les combinaisons d'entrée correspondante.

Table 2.2 : Table de Karnaugh à 3 variables.

A \ BC	00	01	11	10
0	0	0	1	1
1	1	1	0	0

2.4.4 Représentation par logigramme

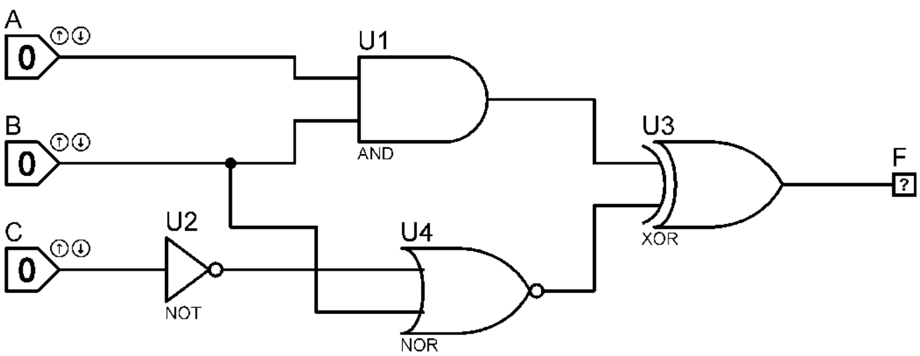


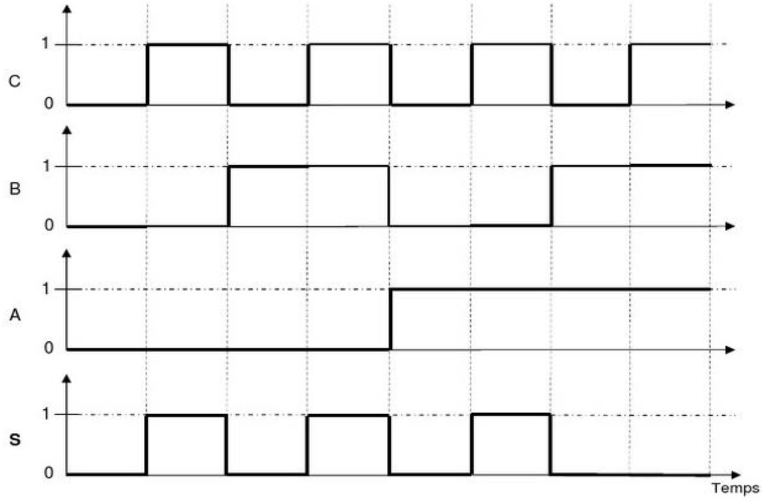
Figure 2.1 : Logigramme logique.

Alors la fonction de sortie est égale :

$$F = AB \oplus \overline{(B + \overline{C})}$$

2.4.5 Représentation par chronogramme

Le chronogramme est une représentation graphique permettant de visualiser, en fonction du temps, toutes les combinaisons d'états logiques possibles des entrées avec l'état correspondant de la sortie.



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

Figure 2.2 : Chronogramme et table de vérité d'une fonction à 3 entrées.

2.5 Simplification des fonctions logiques

Lors de la réalisation d'un circuit électronique d'une fonction logique, on doit simplifier au maximum la fonction à réaliser, ce qui est équivalent à réduire le nombre de ses termes ou le nombre de variables dans un même terme.

Plusieurs méthodes de simplification existent :

- ✓ Méthode algébrique
- ✓ Méthode de Karnaugh

2.5.1 Méthode algébrique

Elle se base sur l'utilisation des propriétés de l'algèbre de Boole.

Exemple :

Soit la fonction à simplifier suivante :

$$\begin{aligned} F(A,B,C) &= AB + \overline{A}C + BC \\ &= AB + \overline{A}C + BC * 1 \\ &= AB + \overline{A}C + BC(A + \overline{A}) \\ &= AB + \overline{A}C + BCA + BC\overline{A} \\ &= AB(1 + C) + \overline{A}C(1 + BC) \\ &= AB(1) + \overline{A}C(1) \\ &= AB + \overline{A}C \end{aligned}$$

2.5.2 Méthode de Karnaugh

La méthode consiste à réaliser des groupements de ***cases adjacentes*** contenant des 1 ou des 0. Un groupement de 1 permet d'obtenir l'équation de F, un groupement de 0 permet d'obtenir l'équation $\overline{F}$.

Règles :

- Le nombre des cases dans un groupement doit être égal à 1, 2 , 4 , …, 2^n.
- Les groupements doivent être les plus grands possibles.
- Les groupements peuvent se chevaucher pour être les plus grands possibles.
- Dans chaque groupement, on ne garde que les variables qui ne changent pas d'état.
- La fonction logique peut être obtenue on effectuant la somme logique (OU logique) de tout les termes trouvés (somme des mintermes).

  Remarques :

  - Un groupement de 1 case n'élimine aucune variable.
  - Un groupement de 2 cases élimine 1 variable.
  - Un groupement de 4 cases élimine 2 variables.
  - Un groupement de 8 cases élimine 3 variables.

Exemples de groupements :

Possibles

0	1	1	0
1	0	0	1
1	0	0	1
0	1	1	0

0	1	0	1
0	0	0	0
1	1	0	0
0	0	0	1

0	1	1	0
1	1	1	1
0	1	1	0
0	1	1	0

1	0	0	1
0	1	1	0
0	1	1	0
1	0	0	1

Impossibles

0	0	0	0
1	1	1	0
1	1	1	0
0	1	1	0

0	1	0	0
0	0	1	0
0	1	0	0
1	0	0	0

Exemple 1 : regroupement à deux

A\BC	00	01	11	10
0	0	0	0	1
A **1**	0	0	0	1

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

A\BC	00	01	11	10
0	0	0	1	1
A **1**	0	0	0	0

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

A\BC	00	01	11	10
0	1	0	0	1
A **1**	0	0	0	0

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

AB\CD	00	01	11	10
00	0	0	1	1
01	0	0	0	0
11	0	0	0	0
10	1	0	0	1

$$F = \overline{A}\overline{B}CD + \overline{A}\overline{B}C\overline{D} + A\overline{B}\overline{C}\overline{D} + A\overline{B}C\overline{D}$$

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

Exemple 2 : regroupement à quatre

AB\CD	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	0	0	0	0
10	0	0	0	0

$F = \overline{A}B$

AB\CD	00	01	11	10
00	0	0	0	0
01	0	1	1	0
11	0	1	1	0
10	0	0	0	0

$F = BD$

AB\CD	00	01	11	10
00	1	0	0	1
01	0	0	0	0
11	0	0	0	0
10	1	0	0	1

$F = \overline{B}\,\overline{D}$

AB\CD	00	01	11	10
00	1	0	0	1
01	1	0	0	1
11	0	0	0	0
10	0	0	0	0

$F = \overline{A}\,\overline{D}$

Exemple 3 : regroupement mixte

AB\CD	00	01	11	10
00	1	1	0	1
01	1	1	0	0
11	0	1	0	0
10	1	0	1	1

$F = \overline{A}\,\overline{B}\,\overline{D} + \overline{A}\,\overline{C} + BD + A\overline{B}C$

AB\CD	00	01	11	10
00	1	0	0	1
01	1	1	0	1
11	1	1	1	0
10	1	0	0	0

$F = \overline{A}\,\overline{D} + B\overline{C} + \overline{C}\,\overline{D} + ABD$

Cas particulier et éléments indéterminés

Des fois une fonction est indéfinie pour certaines combinaisons des variables, la cause la plus courante est que certaines combinaisons des variables étant impossibles, alors on ne donnera pas une valeur particulière à la fonction pour ces combinaisons. Dans les cases correspondantes du tableau de Karnaugh, on placera un signe particulier (Ø : élément indéterminé).

Lors du regroupement des cases, nous transformons le Ø en 0 ou en 1 suivant la convenance ou les simplifications qui peuvent en découler.

Exemple :

A\BC	00	01	11	10
0	1	Ø	Ø	Ø
1	1	Ø	0	0

L'expression la plus simple de F est obtenue en transformant les Ø des cases 001 et 101 en «1» , ce qui permet de regrouper 4 cases et en transformant les Ø des cases 111 et 010 en «0». Nous aurons donc : $F = \overline{B}$