Chapitre 2

Fonctions et Opérateurs Logiques

George Boole (1815-1864)

I. Introduction

Les systèmes logiques fonctionnent en mode binaire ; les variables d'entrée et de sortie ne prennent que deux valeurs : « **0** » ou « **1** ». Ces deux valeurs (états) correspondent à des plages définies à l'avance.

Concrètement, lors de la réalisation de circuits électroniques numériques, les 2 niveaux logiques sont constitués par **2 tensions différentes**. La tension correspondant au niveau 0 est en généra l0 V. La tension correspondant au niveau 1 dépend de la technologie utilisée. Une norme couramment répandue est la norme TTL :

- niveau logique **0** ↔ **0** à **0,8** V
- niveau logique **1** ↔ **2,4** à **5** V

Dans le domaine de la logique numérique, on utilise d'autres expressions qui sont synonymes de 0 et de 1 :

Niveau logique 0	Niveau logique 1
Non	Oui
Faux	Vrai
Ouvert	Fermé
Arrêt	Marche
Bas	Haut
Éteint	Allumé

La logique binaire basée sur l'algèbre de Boole permet de décrire dans un modèle mathématique les manipulations et traitement des informations binaires, et d'analyser les systèmes numériques.

En général, la logique utilisée est la logique **positive**, dans laquelle le niveau dit actif est le **niveau 1**. Ça sera le cas dans la totalité de ce cours. Mais il existe également la **logique négative**, dans laquelle il s'agit du **0**.

II. Variables et fonctions logiques

Cette partie abordera la représentation des fonctions sous forme algébrique, sous forme de table de vérité puis sous forme de schémas, et leur simplification au moyen des règles de l'algèbre de Boole et des tableaux de Karnaugh.

II.1. Algèbre de Boole

George Boole, philosophe et mathématicien irlandais du 19ème siècle est l'auteur d'une théorie sur l'art de construire un raisonnement logique au moyen de propositions qui ont une seule réponse **OUI** (**VRAI**) ou **NON** (**FAUX**). Il est le premier, avec de Morgan, à essayer de fonder la logique mathématique indépendamment de la philosophie. Pour cela, il crée une algèbre binaire n'acceptant que deux valeurs numériques, **0** ou **1**. '**1**' pour une proposition toujours vraie, et '**0**' une proposition fausse. Il définit des lois (**ET** et **OU**) sur cette algèbre satisfaisant un certain nombre de propriétés (distributivité, commutativité,...). L'ensemble des opérations formelles appliquées à ces propositions forme une structure mathématique appelée algèbre de Boole. A son époque, il s'agissait de développement purement théorique car on ignorait l'importance qu'allait prendre cette algèbre avec l'informatique.

Les concepts de la logique booléenne ont été ensuite appliqués aux circuits électroniques par ***Claude Shannon*** (**1916-2001**). Cette algèbre est applicable à l'étude des systèmes possédant deux états s'excluant mutuellement. Dans la logique positive (la plus couramment utilisée), on associe **OUI** à **1** et **NON** à **0**. Dans la logique négative, c'est l'inverse. L'algèbre booléenne binaire est à la base de la mise en œuvre de tous les systèmes numériques : ordinateurs, systèmes numériques portables, systèmes de communication numériques, etc. Elle permet entre autres de simplifier les fonctions logiques, et donc les circuits électroniques associés.

II.2. Variables logiques

Ce sont des variables ne pouvant prendre que deux valeurs distinctes : « **0** » ou « **1** ». Une variable binaire peut représenter n'importe quel dispositif binaire (contact, Interrupteur, lampe, électrovanne,...).

L'association de ces variables dites booléennes ou logiques munie d'un certain nombre d'opérateurs donne naissance à des fonctions logiques.

II.3. Fonctions logiques

Une fonction logique est une fonction d'une ou plusieurs variables logiques, combinées entre elles par **3** fonctions **élémentaires** simples : **NON**, **OU** et **ET**.

Il existe également des fonctions élémentaires composées de fonctions élémentaires simples : NON-ET, NON-OU, OU-EXCLUSIF, NON-OU-EXCLUSIF.

Tout circuit logique peut être décrit par des fonctions logiques et/ou une table de vérité, et être réalisé à partir des opérateurs logiques élémentaires.

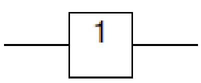
III. Opérateurs logiques de base

Les fonctions logiques peuvent être représentées schématiquement par des opérateurs logiques, encore appelés portes logiques. Chaque **opérateur** est représenté par un **symbole** et sa **fonction** est définie par une **table de vérité**.

III.1. Opérateur identité 'Oui'

Cette fonction fait intervenir une seule variable d'entrée (***A***). Le niveau logique de la sortie est égal au niveau logique de l'entrée : ***S* = *A***. On peut exprimer cette propriété sous forme d'un tableau entrée/sortie, appelé table de vérité.

Symbole européen



Symbole américain

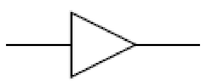


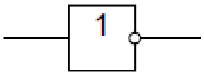
Table de vérité

A	$S = A$
0	0
1	1

III.2. Opérateur inverseur 'Non' ou 'Not'

Elle fait également intervenir une seule variable d'entrée. Le niveau logique de sortie est l'inverse de celui présent à l'entrée.

Symbole européen



Symbole américain

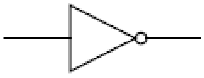


Table de vérité

A	$S=\overline{A}$
0	1
1	0

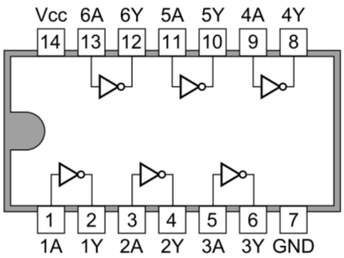
Soit $\boldsymbol{A}$ une variable quelconque. La fonction complément de la variable $\boldsymbol{A}$ est notée : $\boldsymbol{S} = \boldsymbol{f}(\boldsymbol{A}) = \overline{\boldsymbol{A}}$ (Prononcer "**A barre**", "**non A**", ou encore "**complément de A**").

❖ ***<u>Remarque 2.1 :</u>***

Le cercle est en général utilisé pour indiquer une complémentation. On appelle souvent cet opérateur "inverseur".

❖ ***<u>Exemple de Portes logiques NOT</u>***

Le circuit intégré de référence : **<u>74LS04</u>**

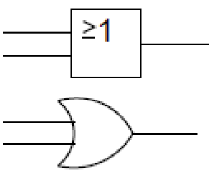


- Ce circuit comporte **14 broches** (**pin** ou **pattes**).
- L'alimentation en tension est faite entre les pattes **7** et **14**; respectivement pour la masse (**GND**) et **5V** (**VCC**).
- Il contient **6 portes** logiques NOT indépendantes ayant **2 entrées chacune**. Les entrées de chacune de ces portes NOT sont identifiées par la broche **A** et **B**, tandis que la sortie est représenté par **Y**.

III.3. Fonction OU (ou somme logique)

La fonction logique OU est également appelée "somme logique", ou "union logique". Sa notation algébrique utilise le symbole de la somme arithmétique. Pour 2 variables A et B, on a :

$$\boldsymbol{f}(\boldsymbol{A},\boldsymbol{B}) = \boldsymbol{A} + \boldsymbol{B}$$



A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

Propriétés du OU :

$x + 1 = 1 \qquad x + \overline{x} = 1$

$x + 0 = x \qquad x + x = x$

Élément neutre : 0

Élément absorbant : 1

Le terme anglais est "OR".

III.3.1 Représentation d'Euler ou de Venn de la fonction OR

Si l'ensemble $\boldsymbol{A}$ est l'ensemble pour lequel la variable $\boldsymbol{A = 1}$, et l'ensemble $\boldsymbol{B}$ celui pour lequel la variable $\boldsymbol{B = 1}$, $\boldsymbol{f}$ est **l'union** de $\boldsymbol{A}$ et $\boldsymbol{B}$, c'est-à-dire l'ensemble des valeurs de $\boldsymbol{f}$ égales à $\boldsymbol{1}$ (Figure 2. 1). Donc L'ensemble dans lequel les variables **A** ou **B** sont à **1**, ou **somme logique**, sera la surface formée de la réunion des deux régions précédentes.

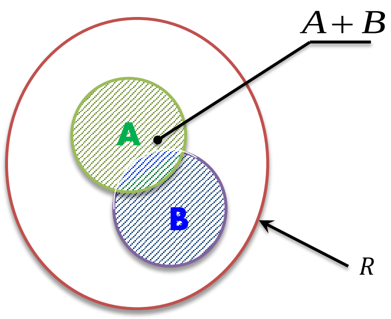
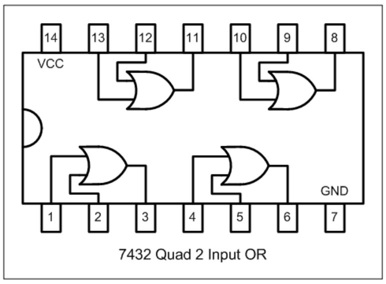


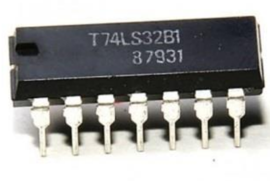
Figure 2. 1. Représentation d'Euler de la fonction OR.

III.3.2 Exemple de Portes logiques OR

Le circuit intégré de Référence : 74LS32

- Ce circuit comporte **14 broches** (**pin** ou **pattes**).
- L'alimentation en tension est faite entre les pattes **7** et **14**; respectivement pour la masse (**GND**) et **5V** (**VCC**).
- Il contient **4 portes** logiques OR indépendantes ayant **2 entrées chacune** et une sortie.

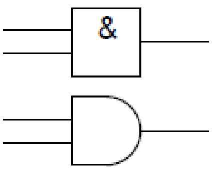




III.4. Fonction ET (ou produit logique)

La fonction ***ET*** est également appelée "produit logique", ou "intersection logique". Sa notation algébrique utilise le symbole de la multiplication arithmétique :

$$f(A,B) = A \cdot B$$



A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

Propriétés du ET :

$x.1 = x \quad x.\overline{x} = 0$

$x.0 = 0 \quad x.x = x$

Élément neutre : 1

Élément absorbant : 0

Le terme anglais est "***AND***".

III.4.1 Représentation d'Euler ou de Venn de la fonction ET

Si l'ensemble ***A*** est l'ensemble pour lequel la variable ***A = 1***, et l'ensemble ***B*** celui pour lequel la variable ***B = 1***, ***f*** est **l'intersection** de ***A*** et ***B***, c'est-à-dire l'ensemble des valeurs de ***f*** égales à ***1*** (Figure 2. 2**)**. Donc L'ensemble dans lequel les variables **A** ou **B** sont à **1**, ou **produit logique**, sera la surface formée de l'intersection des deux régions précédentes.

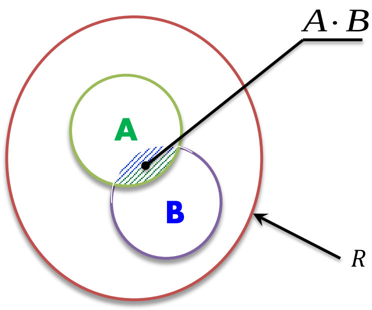


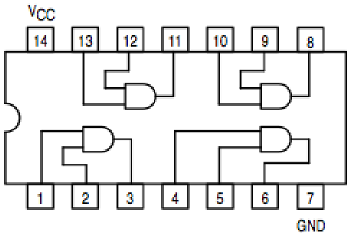
Figure 2. 2. Représentation d'Euler de la fonction AND.

❖ ***Remarque 2.2 :***

Avec la fonction ET à plus de 2 entrées, le seul cas où la sortie serait à 1 serait le cas où toutes les entrées sont à 1.

III.4.2 Exemple de Portes logiques AND

Le circuit intégré de Référence : **74LS08**



- Ce circuit comporte **14 broches** (**pin** ou **pattes**).
- L'alimentation en tension est faite entre les pattes **7** et **14**; respectivement pour la masse (**GND**) et **5V** (**VCC**).
- Il contient **4 portes** logiques **AND** indépendantes ayant **2 entrées chacune** et une sortie.

❖ ***Remarque 2.3 :***

Les opérateurs {ET, OU, NON} permettent à eux trois de réaliser n'importe quelle fonction logique : on dit qu'ils forment un groupe complet.

Le théorème de De Morgan permet de dire que les groupes {ET, NON} et {OU, NON} sont également des groupes complets.

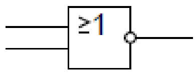
IV. Opérateurs composés

Les fonctions élémentaires composées (ou combinées, ou induits) sont obtenues en combinant entre eux les fonctions élémentaires simples **NON**, **ET** et **OU**. L'ensemble des fonctions élémentaires simples et des fonctions élémentaires combinées **NON-ET**, **NON-OU**, **OU-EXCLUSIF**, **NON-OU-EXCLUSIF** définissent un ensemble complet d'opérateurs.

IV.1. Opérateur Non-Ou (NOR)

Les deux opérateurs OU et NON peuvent être combinés en un seul opérateur NON-OU : NON-OU est donc un opérateur complet.

Symbole européen



Symbole américain

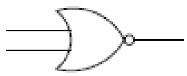


Table de vérité

A	B	$S = \overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

Si variable ***A*** est représentée par l'ensemble ***A***, et la variable ***B*** est représentée par l'ensemble ***B***, ***f*** est **le complément de l'union** de ***A*** et ***B***, c'est-à-dire l'ensemble des valeurs de ***f*** égales à ***1*** **(**Figure 2. 3**).**

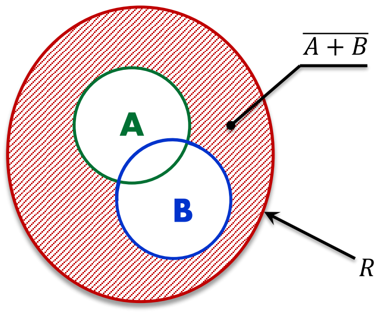
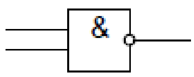


Figure 2. 3. Représentation d'Euler de la fonction NOR.

IV.2. Opérateur Non-Et (NAND)

Les deux opérateurs **ET** et **NON** peuvent être combinés en un seul opérateur NON-ET : NON-ET est donc un opérateur complet.

Symbole européen



Symbole américain

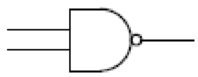


Table de vérité

A	B	$S = \overline{A.B}$
0	0	1
0	1	1
1	0	1
1	1	0

Si variable ***A*** est représentée par l'ensemble ***A***, et la variable ***B*** est représentée par l'ensemble ***B***, ***f*** est **le complément de l'intersection** de ***A*** et ***B***, c'est-à-dire l'ensemble des valeurs de ***f*** égales à ***1*** **(**Figure 2. 3**).**

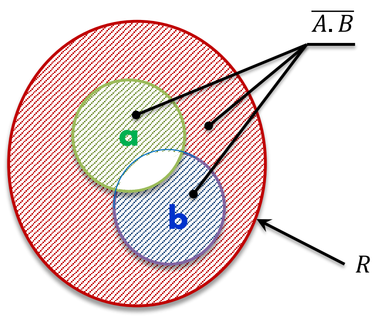


Figure 2. 4. Représentation d'Euler de la fonction NAND.

IV.3. Opérateur Ou exclusif (XOR)

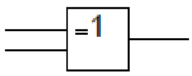
Pour **2** variables **A** et **B**, La sortie d'une porte OU-exclusif est au niveau haut seulement lorsque les deux entrées sont à des niveaux logiques différents, la fonction OU EXCLUSIF est définie par :

$$\boldsymbol{f(A,B) = A\overline{B} + \overline{A}B}$$

On la note également :

$$\boldsymbol{f(A,B) = A \oplus B}$$

Symbole européen



Symbole américain

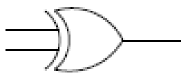


Table de vérité

A	B	$S = A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

La représentation d'Euler de la fonction XOR est la suivante (Figure 2. 5) :

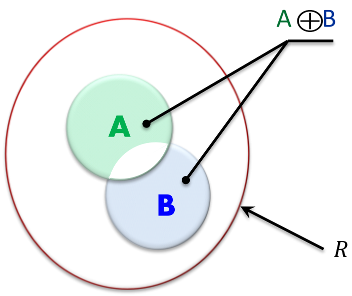
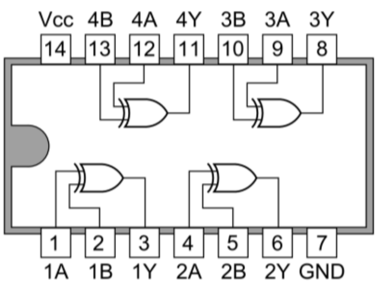


Figure 2. 5. Représentation d'Euler de la fonction XOR.

❖ ***Exemple de Portes logiques XOR***

Le circuit intégré de référence : **74LS86**



- Ce circuit comporte **14 broches** (**pin** ou **pattes**).
- L'alimentation en tension est faite entre les pattes **7** et **14**; respectivement pour la masse (**GND**) et **5V** (**VCC**).
- Il contient **4 portes** logiques XOR indépendantes ayant **2 entrées chacune**. Les entrées de chacune de ces portes XOR sont identifiées par la broche **A** et **B**, tandis que la sortie est représenté par **Y**.

IV.4. Fonction NON-OU EXCLUSIF (XNOR)

La fonction NON-OU EXCLUSIF est obtenue en complémentant la sortie d'un OU EXCLUSIF, c'est à dire en appliquant la sortie de la fonction OU EXCLUSIF à la fonction NON. Pour 2 variables A et B, elle est notée :

$$\boldsymbol{f(A,B) = \overline{A \oplus B}}$$

On la note également :

$$\boldsymbol{f(A,B) = A \odot B}$$

La représentation d'Euler de la fonction XNOR est la suivante (Figure 2. 6) :

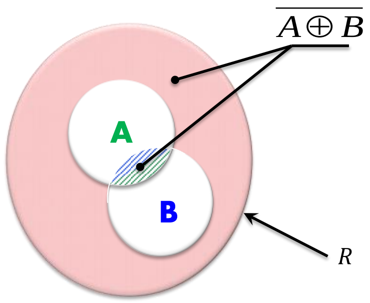
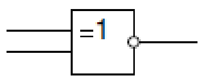


Figure 2. 6. Représentation d'Euler de la fonction XNOR.

❖ ***Remarque 2.4 :***

Pour obtenir la table de vérité du NON-OU EXCLUSIF, il suffit d'inverser les valeurs de sortie dans la table de vérité du OU EXCLUSIF.

Symbole européen



Symbole américain

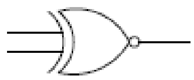
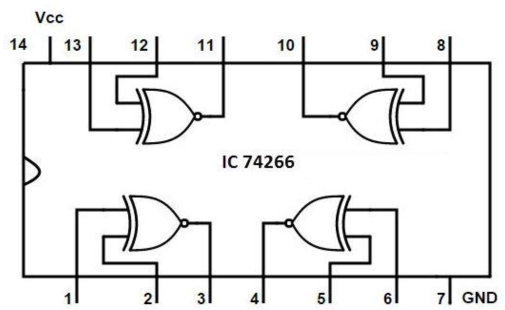


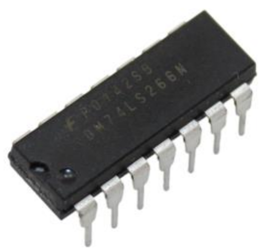
Table de vérité

A	B	$S = A \odot B$
0	0	1
0	1	0
1	0	0
1	1	1

❖ ***Exemple de Portes logiques XNOR***

Le circuit intégré de référence : **74LS266**





- Ce circuit comporte **14 broches** (**pin** ou **pattes**).
- L'alimentation en tension est faite entre les pattes **7** et **14**; respectivement pour la masse (**GND**) et **5V** (**VCC**).
- Il contient **4 portes** logiques XNOR indépendantes ayant **2 entrées chacune** et une sortie.

V. Propriétés et théorèmes

V.1. Propriétés monovariables

V.1.1 Elément neutre

À chaque opérateur correspond un élément neutre qui, lorsqu'il est opéré avec une variable quelconque ***A***, donne un résultat identique à cette variable.

$$\boldsymbol{A+0=A} \qquad \boldsymbol{A\cdot 1=A}$$

V.1.2 Elément nul

À chaque opérateur correspond un élément nul (appelé aussi élément absorbant) qui, lorsqu'il est opéré avec une variable quelconque ***A***, donne un résultat identique à cet élément nul.

$$\boldsymbol{A+1=1} \qquad \boldsymbol{A\cdot 0=0}$$

V.1.3 Idempotence

Le résultat d'une opération entre une variable ***A*** et elle-même est égal à cette variable.

$$\boldsymbol{A+A=A} \qquad \boldsymbol{A\cdot A=A}$$

V.1.4 Complémentation

$$\boldsymbol{A+\overline{A}=1} \qquad \boldsymbol{A\cdot \overline{A}=0}$$

V.1.5 Involution

Le complément du complément d'une variable ***A*** est égal à cette variable.

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

V.2. Propriétés multivariables

V.2.1 Associativité

Les opérations $+$, $\cdot$, et $\oplus$ sont associatives :

$$A+B+C=(A+B)+C=A+(B+C)$$
$$A\cdot B\cdot C=(A\cdot B)\cdot C=A\cdot(B\cdot C)$$
$$A\oplus B\oplus C=(A\oplus B)\oplus C=A\oplus(B\oplus C)$$

V.2.2 Commutativité

Les opérations $+$, $\cdot$, et $\oplus$ sont commutatives :

$$A+B=B+A$$
$$A\cdot B=B\cdot A$$
$$A\oplus B=B\oplus A$$

V.2.3 Distributivité

Chacune des opérations + et $\cdot$ est distributive sur l'autre :

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

❖ ***Remarque 2.5 :***

On peut remarquer que cette propriété est particulière dans l'algèbre booléenne puisqu'ici les deux expressions sont vraies, alors que seule la première l'est dans l'algèbre ordinaire.

V.2.4 Absorption

$$A + A \cdot B = A$$
$$A \cdot (A + B) = A$$

V.2.5 Dualité

Deux expressions sont dites **duales** si l'on obtient l'une en **changeant** dans l'autre, les **ET** par des **OU**, les **OU** par des **ET**, les '**1**' par des '**0**' et les '**0**' par des '**1**'.

Conséquences de Principe de dualité :

*Toute expression logique reste **vrais** si on remplace le **ET** par le **OU**, le **OU** par le **ET**, les **1** par des **0** et les **0** par des **1**.*

V.2.6 Théorème de Consensus

Le **théorème du consensus** ou la **règle du consensus** sont les identités :

$$A \cdot B + \bar{A} \cdot C = A \cdot B + \bar{A} \cdot C + \boldsymbol{B} \cdot \boldsymbol{C}$$
$$(A + B) \cdot (\bar{A} + C) = (A + B) \cdot (\bar{A} + C) \cdot (\boldsymbol{B} + \boldsymbol{C})$$

La 2$^{\text{ème}}$ égalité est donc le dual de la 1$^{\text{ère}}$; on peut donc l'obtenir par application du principe de dualité.

Les termes $\boldsymbol{B} \cdot \boldsymbol{C}$ et $(\boldsymbol{B} + \boldsymbol{C})$ sont appelés les consensus des termes $A \cdot B$ et $\bar{A} \cdot C$ pour la 1$^{\text{ère}}$ expression et $(A + B)$ et $(\bar{A} + C)$ pour la 2$^{\text{ème}}$.

V.2.7 Théorème de De Morgan

❶ Le complément d'un produit est égal à la somme des termes complémentés :

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

❷ Le complément d'une somme est égal au produit des termes complémentés :

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

Généralisation de théorème de De Morgane :

$$\overline{f(a, b, c, \dots, \cdot, +)} = f\left(\bar{a}, \bar{b}, \bar{c}, \dots, +, \cdot\right)$$

VI. Représentation des fonctions logiques

VI.1. Formes canonique d'une fonction

Les expressions booléennes peuvent être manipulées sous différentes formes.

Une expression est sous sa **forme canonique** si **chaque terme** de la fonction comporte **toutes les variables**. Lorsqu'une équation est écrite à partir de sa table de vérité, elle est dans sa forme canonique.

Ils existent plusieurs formes canoniques : les plus utilisées sont la **première** et la **deuxième forme**.

VI.1.1 1ère Forme Canonique ou Forme disjonctive (Sommes de Produits)

On appelle «**minterme**» de **n** variables, l'un des **produits** de ces variables ou de leurs complémentaires.

EXEMPLE 2.1 :

La table suivante donne les **mintermes** d'une fonction de deux variables :

		m_0	m_1	m_2	m_3
A	B	$\bar{A}\cdot\bar{B}$	$\bar{A}\cdot B$	$A\cdot\bar{B}$	$A\cdot B$
0	**0**	**1**	0	0	0
0	**1**	0	**1**	0	0
1	**0**	0	0	**1**	0
1	**1**	0	0	0	**1**

Une fonction booléenne peut être représentée sous forme d'une **somme de produits** (forme disjonctive) utilisant les mintermes. Ces mintermes sont représentés par des '**1**' dans une table de vérité.

EXEMPLE 2.2 :

$$f(A,B,C) = \underbrace{\bar{A}\cdot\bar{B}\cdot C}_{Minterme} + \underbrace{\bar{A}\cdot B\cdot C}_{\dots} + \underbrace{A\cdot B\cdot C}_{\dots}$$

$$f(A,B,C) = \sum m(1,3,7)$$

VI.1.2 2ème Forme Canonique ou Forme conjonctive (Produits de Sommes)

On appelle «**maxterme**» de **n** variables, l'une des **sommes** de ces variables ou de leurs complémentaires.

Une fonction booléenne peut être représentée sous forme d'un **produit de sommes** (forme **conjonctive**) utilisant les maxtermes. Ces maxtermes sont représentés par des '**0**' dans une table de vérité.

EXEMPLE 2.3 :

La table suivante donne les **maxtermes** d'une fonction de deux variables :

		M_0	M_1	M_2	M_3
A	B	$A+B$	$A+\bar{B}$	$\bar{A}+B$	$\bar{A}+\bar{B}$
0	**0**	**0**	1	1	1
0	**1**	1	**0**	1	1
1	**0**	1	1	**0**	1
1	**1**	1	1	1	**0**

EXEMPLE 2.4 :

$$f(A,B,C) = \underbrace{(A+B+C)}_{Maxterme} \cdot \underbrace{(\bar{A}+B+C)}_{...} \cdot \underbrace{(\bar{A}+\bar{B}+C)}_{...}$$

$$f(A,B,C) = \sum M(0,4,6)$$

VI.1.3 Représentation d'une fonction sous forme de mintermes et maxtermes

Une fonction logique peut être représentée sous :

❶ Sa 1ère forme canonique (Σ de mintermes) : on développe la fonction sous la forme d'une somme de produits (SDP) puis on prend chaque terme avec pour variable manquante **X** et on applique un ET logique avec $\boldsymbol{X + \bar{X}}$;

❷ Sa 2ème forme canonique (Π de maxtermes) : on développe la fonction sous la forme d'un produit de sommes (PDS) puis on prend chaque terme avec pour variable manquante **X** et on applique un OU logique avec $\boldsymbol{X.\bar{X}}$.

EXEMPLE 2.5 :

1. $f(A,B) = A + B$

$$\begin{aligned} f(A,B) &= A \cdot (B+\bar{B}) + B \cdot (A+\bar{A}) \\ &= A \cdot B + A \cdot \bar{B} + B \cdot A + B \cdot \bar{A}) \\ &= \bar{A} \cdot B + A \cdot \bar{B} + A \cdot B \end{aligned}$$

$$f(A,B) = \sum m(1,2,3)$$

2. $f(A,B,C) = A \cdot B + C$

$$\begin{aligned} f(A,B,C) &= (A+C) \cdot (B+C) \\ &= (A+C+B \cdot \bar{B}) \cdot (B+C+A \cdot \bar{A}) \\ &= (A+C+B) \cdot (A+C+\bar{B}) \cdot (B+C+A) \cdot (B+C+\bar{A}) \end{aligned}$$

$$f(A,B,C) = (A+B+C) \cdot (A+\bar{B}+C) \cdot (\bar{A}+B+C)$$

$$f(A,B,C) = \prod M(0,2,4)$$

VI.1.4 Décomposition de Shannon

Lorsque ***Shannon*** introduisit l'algèbre en vue d'une utilisation dans le cadre des circuits à relais, il ajouta une notion importante, connue aujourd'hui sous le nom de la ***décomposition de Shannon***. La décomposition de Shannon est très utile pour *la simplification des fonctions logiques*, et elle nous servira pour mieux comprendre le fonctionnement de certains circuits usuels vus plus loin dans ce cours.

La décomposition de Shannon s'énonce comme suit :

Soient $\boldsymbol{f}$ une fonction logique de $\boldsymbol{n}$ paramètres $\boldsymbol{x_0, x_1, \ldots x_{n-1}}$ alors :

$$f(\boldsymbol{x_0, x_1, \ldots x_{n-1}}) = \boldsymbol{\overline{x_0}}.f(\boldsymbol{0, x_1, \ldots x_{n-1}}) + \boldsymbol{x_0}.f(\boldsymbol{1, x_1, \ldots x_{n-1}})$$

Notons ici que la variable « $\boldsymbol{x_0}$ » peut être remplacée par n'importe quelle autres variable parmi les $\boldsymbol{n}$ variables restantes. La décomposition peut également être appliquée récursivement sur l'ensemble des variables. De plus, en appliquant le principe de ***dualité***, on peut trouver que, pour toute fonction logique $\boldsymbol{f}$ de de $\boldsymbol{n}$ paramètres $\boldsymbol{x_0}, \boldsymbol{x_1}, \; \dots \boldsymbol{x_{n-1}}$, il est possible d'écrire :

$$f(\boldsymbol{x_0}, \boldsymbol{x_1}, \; \dots \boldsymbol{x_{n-1}}) = \left(\overline{\boldsymbol{x_0}} + f(\boldsymbol{1}, \boldsymbol{x_1}, \; \dots \boldsymbol{x_{n-1}})\right).\left(\boldsymbol{x_0} + f(\boldsymbol{0}, \boldsymbol{x_1}, \; \dots \boldsymbol{x_{n-1}})\right)$$

VI.2. Représentations tabulaires

VI.2.1 Table de vérité

Une table de vérité est l'écriture des valeurs d'une fonction logique pour toutes les combinaisons possibles de ses variables. Chaque ligne présente la combinaison des variables d'entrée ainsi que la ou les sorties correspondante(s).

EXEMPLE 2.6 :

On définit la fonction logique $\boldsymbol{f}\ (\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}) = \boldsymbol{1}$ si $(\boldsymbol{A}\ \boldsymbol{B}\ \boldsymbol{C})_{\boldsymbol{2}} > \boldsymbol{4}$. La table de vérité correspondante est :

$\boldsymbol{A}$	$\boldsymbol{B}$	$\boldsymbol{C}$	$\boldsymbol{f}(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C})$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Le principal inconvénient de la table de vérité est qu'elle devient rapidement très encombrante lorsque le nombre de variables d'entrée augmente.

VI.2.2 Diagramme de Karnaugh et termes adjacents

Quand une équation logique est établie, il faut la simplifier si possible car ceci diminue le nombre de circuits électroniques à utiliser. Cette simplification s'effectue à l'aide de l'algèbre de Boole ou d'un diagramme de Karnaugh. Ce diagramme représente l'état des variables à l'intérieur de cases, pour $\boldsymbol{n}$ variables il y a $\boldsymbol{2^n}$ cases, donc pour 2, 3 et 4 variables il faut 4, 8 et 16 cases, pour 5 variables il faut deux diagrammes de 16 cases.

Deux termes sont adjacents quand ils ne diffèrent l'un de l'autre que par une seule variable (ABC et $AB\bar{C}$ sont adjacents). Un diagramme – ou tableau – de Karnaugh est une table d'implication logique disposée de telle manière que deux termes logiquement adjacents soient également adjacents géométriquement. Afin d'exploiter la notion ***d'adjacence*** entre les termes, les cases doivent être ordonnées selon le ***code binaire réfléchi***, au lieu du code binaire naturel.

❖ ***<u>Remarque 2.6 :</u>***

Les tableaux de Karnaugh se présentent comme des cylindres fermés dans les deux sens (Figure 2. 7).

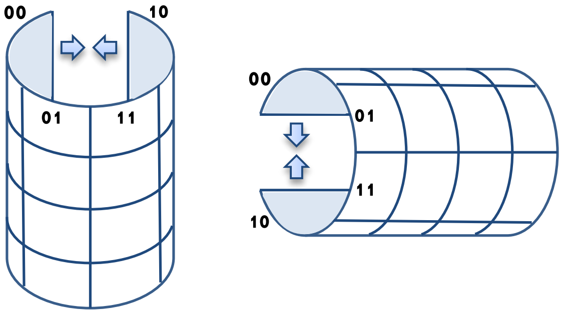


Figure 2. 7. Représentation cylindrique d'un tableau de Karnaugh à 4 variables.

EXEMPLE 2.7 :

Tableau de Karnaugh d'une fonction à **3 variables** :

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

→

f

	C		B	
	0	1	0	1
A	1	0	1	0

VI.3. Représentations graphiques

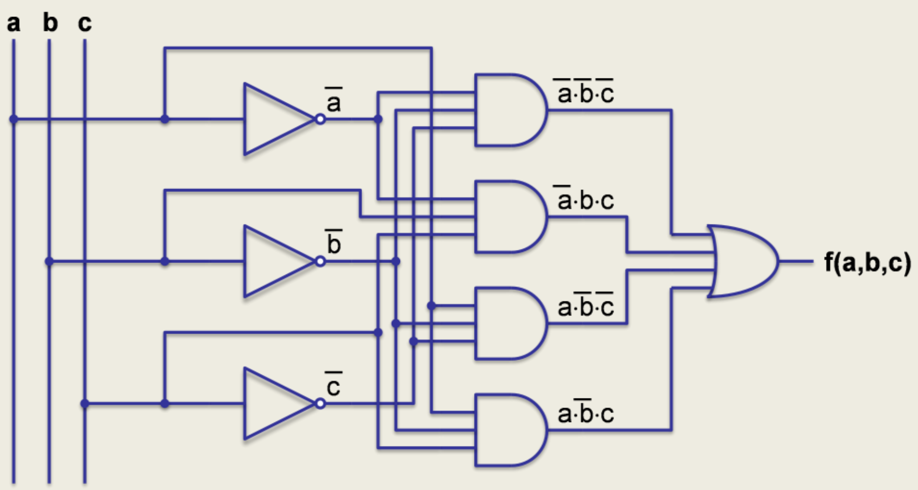
VI.3.1 Logigramme

Un logigramme est un schéma illustrant l'expression d'une fonction logique sans tenir compte des constituants technologiques. Le principe consiste à remplacer chaque opérateur logique par la porte logique qui lui correspond.

EXEMPLE 2.8 :

Soit la fonction logique $f(a,b,c)=\overline{a}\,\overline{b}\,\overline{c}+\overline{a}bc+a\overline{b}\,\overline{c}+a\overline{b}c$

Le Logigramme ***f*** est :

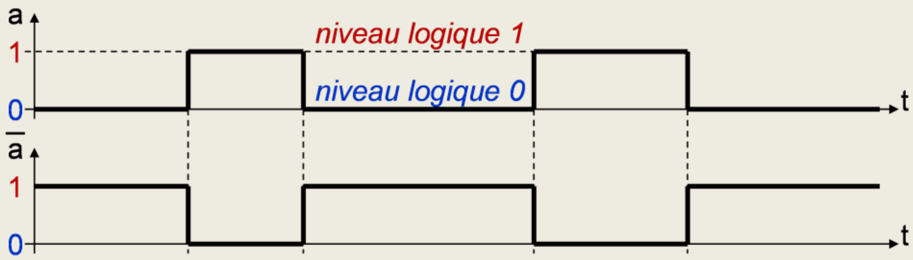


VI.3.2 Chronogramme

C'est le graphe d'évolution temporelle des variables et des fonctions logiques.

EXEMPLE 2.9 :

Le chronogramme de la fonction **NON**



VII. Simplification des fonctions logiques

La simplification d'une fonction logique est son écriture sous forme d'une expression contenant le minimum de lettres et de termes.

VII.1. Simplification algébrique

On réalise cette simplification en utilisant l'ensemble des propriétés et théorème de l'algèbre de Boole et en particulier l'absorption et le théorème de consensus.

VII.1.1 Simplification par mise en facteur commun

On peut faire cette simplification si on a une variable dans un terme et son inverse dans l'autre et si le reste des variables est **identique**.

EXEMPLE 2.10 :

$$\begin{aligned}
f &= \overline{a}\,\overline{b}\,\overline{c}\,\overline{d} + \overline{a}b\overline{c}\,\overline{d} + \overline{a}\,\overline{b}\,\overline{c}d + \overline{a}bcd + ab\overline{c}d + abcd + a\overline{b}c\overline{d} \\
&= \overline{a}\,\overline{c}\,\overline{d}(\overline{b} + b) \quad + \quad \overline{a}bd\left(c + \overline{c}\right) \quad + \quad abd\left(c + \overline{c}\right) \quad + \quad a\overline{b}c\overline{d} \\
&= \quad \overline{a}\,\overline{c}\,\overline{d} \quad + \quad \overline{a}bd \quad + \quad abd \quad + \quad a\overline{b}c\overline{d} \\
&= \overline{a}\,\overline{c}\,\overline{d} + bd + a\overline{b}c\overline{d}
\end{aligned}$$

VII.1.2 Simplification par Consensus et absorption

On peut faire cette simplification si les deux termes n'ont pas le même nombre de variables et s'il y a une variable dans un terme et sont inverse dans l'autre.

EXEMPLE 2.11 :

$$f = \overline{a}\,\overline{b} \quad + \quad a\,b\,c \quad = \overline{a}\,\overline{b} \quad + \quad a\,b\,c \quad + \quad \underbrace{\overline{a}\;c}_{\text{Terme Consensus}}$$

$$= \overline{a}\,\overline{b} \quad + \quad \overline{a}c$$

VII.2. Simplification par tableau de Karnaugh

VII.2.1 Principe

Le diagramme de Karnaugh est un outil graphique qui permet de simplifier une équation. Soit une fonction définie par un tableau de Karnaugh, on peut simplifier la fonction en effectuant de groupement des cases adjacentes contenant la valeur 1 (ou encore la valeur 0).

VII.2.2 Groupement de 2 cases adjacentes

Prenons l'exemple de la fonction S1 :

S1 (a \ b)	0	1
0	x 1	0
1	y 1	0

Les cases **x** et **y** sont adjacentes :

$$\left.\begin{array}{l} x = \bar{a}\,\bar{b} \\ y = a\,\bar{b} \end{array}\right\} \Rightarrow \quad x + y = \bar{a}\,\bar{b} + a\bar{b} = (\bar{a} + a)\bar{b} = \bar{b} \qquad \text{d'où} \qquad \boldsymbol{S_1 = \bar{b}}$$

Le groupement de **deux cases** adjacentes contenant la valeur 1 correspond à deux termes qui différent d'une variable complémentée dans un terme et non complémentée dans l'autre. Le terme résultant du groupement ne comporte pas cette variable qui change d'état.

VII.2.3 Groupement de 4 cases adjacentes

Prenons l'exemple de la fonction **S$_2$** :

S2 (ab \ cd)	$\bar{c}\bar{d}$ 00	$\bar{c}d$ 01	cd 11	$c\bar{d}$ 10	
$\bar{a}\bar{b}$ 00	0	0	0	0	
$\bar{a}b$ 01	0	1	1	0	A
ab 11	0	1	1	0	B
$a\bar{b}$ 10	1	0	0	1	C

$$A = \bar{a}b\,(\bar{c}d + cd) = \bar{a}bd$$
$$B = ab\,(\bar{c}d + cd) = abd \quad \Rightarrow \quad A + B = bd(\bar{a} + a) = bd$$
$$C = a\bar{b}\bar{d} \qquad \text{d'où} \qquad \boldsymbol{S_2 = A + B + C = bd + a\bar{b}\bar{d}}$$

Le groupement de **4 cases** adjacentes contenant la valeur 1 conduit à un terme réduit dans lequel deux variables disparaissent.

D'une manière générale, le **groupement** de **2^n** (2, 4, 8, 16, …) cases adjacentes conduit à un terme réduit dans lequel **n variables disparaissent**.

❖ *Principe de la simplification graphique :*

La simplification graphique consiste à faire apparaître sur le tableau de Karnaugh des groupements en **puissance de 2**, aussi ***importants*** que possible, de cases adjacentes contenant la valeur '**1**'. Une même case peut faire partie de plusieurs groupements. L'écriture simplifiée de la fonction est la somme des termes engendrés par chaque groupement.

❖ *Remarque 2.7 :*

- L'adjacence existe aussi sur les extrémités de tableau :

S ab \ cd	00	01	11	10
00	1	1	0	0
01	0	0	0	0
11	0	0	0	0
10	1	1	0	0

$S = \bar{b}\bar{c}$

S 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

$S = \bar{b}\bar{d}$

- Pour représenter la fonction sous forme de produits de sommes, on procède par groupement des '**0**' :

F_1 c \ ab	00	01	11	10
0	0	0	0	0
1	1	1	0	0

B

A

$$A = (a+b+c)(a+\bar{b}+c) = (a+c)$$
$$B = (\bar{a}+c)(\bar{a}+\bar{c}) = \bar{a}$$
$$\Rightarrow \quad AB = \bar{a}(a+c)$$

d'où $\boldsymbol{F_1 = AB = \bar{a}c}$

VII.2.4 Fonctions incomplètement définies

Dans des cas pratiques, certains combinaisons de variables n'ont aucun sens physique et n'apparaissent jamais dans la réalité. Il est donc inutile de spécifier la valeur de la fonction pour de telles combinaisons.

Dans ce cas, le concepteur peut à sa convenance attribuer à ces cases la valeur **0** ou **1** de manière à obtenir le maximum de groupements.

EXEMPLE 2.12 : *F* est définie par ce tableau de Karnaugh :

ab \ cd	00	01	11	10
00	1	0	0	0
01	1	1	Φ	0
11	0	Φ	1	0
10	Φ	Φ	0	1

$\bar{a}\bar{c}\bar{d}$

bd

$a\bar{b}\bar{d}$

$$F = \bar{a}\bar{c}\bar{d} + a\bar{b}\bar{d} + bd$$