INTRODUCTION

L'Algèbre de Boole définit en 1847 par Georges Boole (1815-1864), physicien Anglais, c'est une Algèbre applicable au raisonnement logique qui traite des fonctions à variables binaires (seulement deux valeurs) c.a.d ne peut pas être appliqué aux systèmes à plus de deux états. L'algèbre de Boole permet de manipuler des valeurs logiques, il est à noter qu'une valeur logique n'a que deux états possibles : Vrai(1) ou Faux (0).
Plusieurs valeurs logiques peuvent être combinées pour donner un résultat qui est lui aussi une valeur logique

Définitions

Etat: Les états logiques sont représentés par 0 et 1.

Variable: Une grandeur représentée par un symbole, qui peut prendre deux états (0 ou 1).

Fonction: Elle représente un groupe de variables reliées par des opérateurs logiques.

Exemple

On peut associer à un grand nombre de phénomènes physique, un état logique Exemple : (arrêt, marche) (ouvert fermé) (Noir, Blanc) (avant, arrière) (Allumé, etteint).

On associe généralement à l'état logique 1 la situation actionné du composant.

1. LA LOGIQUE COMBINATOIRE

1.1 DEFINITION DE LA LOGIQUE COMBINATOIRE

La logique combinatoire, à l'aide de fonctions logiques, permet la construction d'un système combinatoire.

Un système est dit combinatoire quand il est de type boucle ouverte, c'est-à-dire qu'aucune des sorties n'est bouclée en tant qu'entrée.

A chaque combinaison d'entrée correspond une seule sortie. Les systèmes combinatoires sont les plus simples et peuvent se représenter par une table de vérité indiquant pour chaque état d'entrée quel est l'état de sortie correspondant.

Toute fonction logique peut être réalisée à l'aide d'un petit nombre de fonctions logiques de base appelées **opérateurs logiques** ou **portes**. Il existe deux types d'opérateurs :

- **Opérateurs fondamentaux** : NON, ET, OU
- **Opérateurs dérivés** : NON-ET, NON-OU, OU-EXC, NON-OU-EXC

1.2 PROPRIETES DE L'ALGEBRE DE BOOLE

	Somme de produits (ΣΠ)	**Produit de sommes (ΠΣ)**
AXIOMES		
Associativité	$(a+b)+c = a+(b+c) = a+b+c$	$(a.b).c = a.(b.c) = a.b.c = abc$
Commutativité	$a+b = b+a$	$a.b = b.a$
Distributivité	$a.(b+c) = a.b + a.c$	$a+(b.c) = (a+b).(a+c)$ *
Elément neutre	$a+0 = a$	$a.1 = a$
Complément	$a+\overline{a} = 1$	$a.\overline{a} = 0$
		*: démontrer au moyen de tables de vérité
THEOREMES		
Idempotence	$a+a = a$	$a.a = a$
Absorption	$a+(a.b) = a$	$a.(a+b) = a$
	$a+1 = 1$	$a.0 = 0$
Involution	$\overline{\overline{a}} = a$	$\overline{\overline{a}} = a$
Loi de Morgan	$\overline{a+b} = \overline{a}.\overline{b}$	$\overline{a.b} = \overline{a}+\overline{b}$
Concensus de 1re espèce	$a+\overline{a}.b = a+b$	$a.(\overline{a}+b) = a.b$
Concensus de 2e espèce	$ab+\overline{b}c = ab+\overline{b}c+ac$	$(a+b).(\overline{b}+c) = (a+b).(\overline{b}+c).(a+c)$

DUALITE DE L'ALGEBRE DE BOOLE

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

Exemple :

$A+1=1 \rightarrow A.0=0$

$A+\overline{A}=1 \rightarrow A.\overline{A}=0$

LA LOI DE MORGAN

La somme logique complimentée de deux variables est égale au produit des compléments des deux variables

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

Le produit logique complimenté de deux variables est égale au somme logique des compléments des deux variables.

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

- ***Généralisation du Théorème DEMORGANE à N variables***

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

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

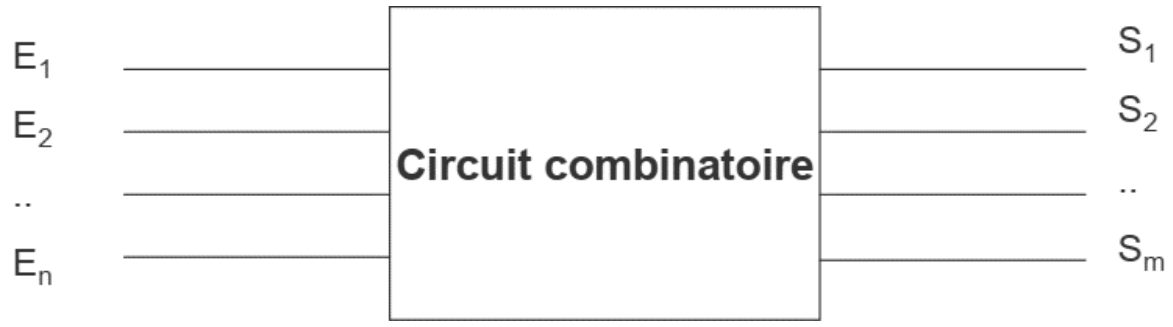
1.3. LES CIRCUITS COMBINATOIRES

On appelle circuit logique (ou circuit combinatoire) un assemblage de portes logiques reliées entre elles pour schématiser une expression algébrique

Un circuit combinatoire est un circuit numérique dont les sorties dépendent uniquement des entrées.

$S_i=F(Ei)$

$S_i=F(E_1,E_2,….,E_n)$



On utilise des circuits combinatoires pour réaliser d'autres circuits plus complexes

FONCTIONS LOGIQUES

C'est une fonction qui relie N variables logiques avec un ensemble d'opérateurs logiques de base.

• Dans l'Algèbre de Boole il existe trois opérateurs de base : NON , ET , OU.

• La valeur d'une fonction logique est égale à 1 ou 0 selon les valeurs des variables logiques.

• Si une fonction logique possède N variables logiques → 2^n combinaisons → la fonction possède 2^n valeurs.

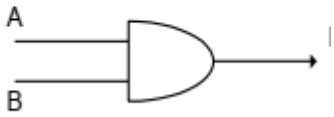
TABLES DE VERITE DES OPERATEURS DE BASE

Fonction logique ET (AND)

Représentation : F= A*B ou A ⋆ B ou AB

Table de vérité

Entrée		*Sortie*
B	A	F
0	0	0
0	1	0
1	0	0
1	1	1



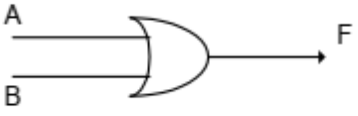
Symbole graphique

Fonction logique Ou (OR)

Représentation :F= A+ B

Table de vérité

Entrée		*Sortie*
B	A	F
0	0	0
0	1	1
1	0	1
1	1	1



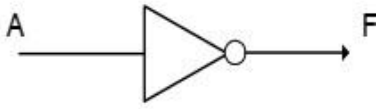
Symbole graphique

Fonction logique Non (Not)

Représentation : $F=\bar{A}$

Table de vérité

Entrée	*Sortie*
A	F
0	1
1	0



Symbole graphique

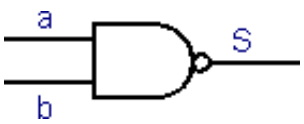
FONCTIONS DÉRIVÉES DES FONCTIONS FONDAMENTALES

Fonction logique NON ET (NAND)

Représentation : F= A/B

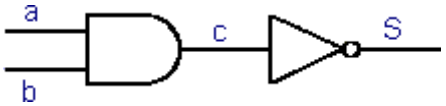
Table de vérité

A	B	A/B
0	0	**1**
0	1	**1**
1	0	**1**
1	1	**0**



Symbole graphique du NAND

Un circuit **NAND** est obtenu en mettant en série une porte **ET** et un inverseur comme représenté ci-dessous



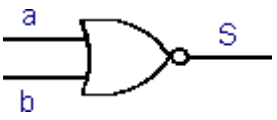
Décomposition d'un circuit NAND

Fonction logique NON OU (NOR)

Représentation : F= A↓B

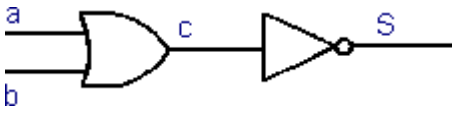
Table de vérité

A	B	A↓B
0	0	**1**
0	1	**0**
1	0	**0**
1	1	**0**



Symbole graphique du NOR

Un circuit **NOR** est obtenu en mettant en série une porte **Ou** et un inverseur comme représenté ci-dessous



Décomposition d'un circuit NOR

FONCTION XOR (OU EXCLUSIF / EXCLUSIVE OR)

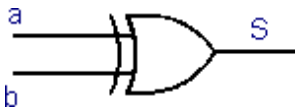
Dans le cas du **OU exclusif**, , pour **S = 1**, il faudra que **a OU b** soit à **1** exclusivement, c'est-à-dire que la sortie sera égale à 1 si les deux entrées sont d'états différentes (l'un égale à 1 et l'autre à zéro)

On écrit alors que **F = a ⊕ b** que l'on énonce **F égal a OU exclusif b.**

Le signe ⊕ est le symbole du XOR (**OU exclusif)** dans les équations logiques.

Table de vérité

A	B	A ⊕ B
0	0	**0**
0	1	**1**
1	0	**1**
1	1	**0**



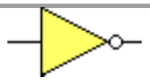
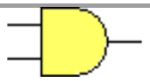
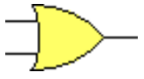
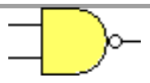
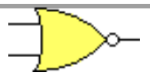
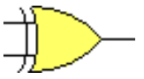
Symbole graphique du XOR

Remarque :

•Les portes ET , OU , NAND , NOR peuvent avoir plus que deux entrées

·Il n'existe pas de OU exclusif à plus de deux entrées

TABLEAU RECAPITULATIF

<table>
<tr><td rowspan="3">Opérateurs de base</td><td>NON (NOT)
Inversion</td><td>–</td><td> 
/a</td><td rowspan="3">NOT, NON
<table><tr><th>A</th><th>/A</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table></td></tr>
<tr><td>ET (AND)</td><td>.</td><td> 
a.b</td></tr>
<tr><td>OU (OR)</td><td>+</td><td> 
a+b</td></tr>
<tr><td rowspan="3">Combinaisons d'opérateurs</td><td>NAND
(NOT-AND)</td><td>/</td><td> 
a/b</td><td rowspan="3">NAND,
ET-NON
<table><tr><th>A</th><th>B</th><th>A/B</th></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
(NOT-OR)</td><td>↓</td><td> 
a↓b</td></tr>
<tr><td>XOR
ou exclusif</td><td>⊕</td><td> 
a ⊕ b</td></tr>
</table>

<table>
<tr><th>A</th><th>B</th><th>A.B</th></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>

AND, ET

<table>
<tr><th>A</th><th>B</th><th>A+B</th></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>

OR, OU

<table>
<tr><th>A</th><th>B</th><th>A↓B</th></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>

NOR,

OU-NON

<table>
<tr><th>A</th><th>B</th><th>A ⊕ B</th></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>

XOR, OUX

.

1.4.1 ETABLISSEMENT DE LA TABLE DE VERITE

Les fonctions logiques peuvent être représentées par des **Tables de vérités,** La table de vérité permet la connaissance de la sortie d'un circuit logique en fonction des diverses combinaisons des valeurs des entrées

- **Le nombre de colonnes est le nombre total d'entrées et de sorties**
- **Le nombre de lignes est 2^N sachant que "N" est le nombre d'entrées,**

Une fonction de 3 entrées et 1 sortie se représente par une table de 4 colonnes et 8 lignes

A	B	C	Résultat
0	0	0	$\bar{A}\bar{B}\bar{C}$
0	0	1	$\bar{A}\bar{B}C$
0	1	0	$\bar{A}B\bar{C}$
0	1	1	$\bar{A}BC$
1	0	0	$A\bar{B}\bar{C}$
1	0	1	$A\bar{B}C$
1	1	0	$AB\bar{C}$
1	1	1	ABC

Exemple : Soit F1 et F2 deux fonctions logiques avec leurs tables de vérité respectives

F1 = *XY + YZ + XZ*

X	Y	Z	XY	YZ	XZ	F1
0	0	0	0	0	0	**0**
0	0	1	0	0	0	**0**
0	1	0	0	0	0	**0**
0	1	1	0	1	0	**1**
1	0	0	0	0	0	**0**
1	0	1	0	0	1	**1**
1	1	0	1	0	0	**1**
1	1	1	1	1	1	**1**

F2 **= X +YZ +** $\bar{Y}\bar{Z}$**W**

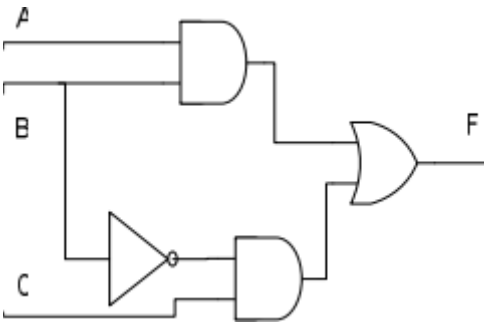
X	*Y*	*Z*	*W*	*YZ*	$\bar{Y}\bar{Z}$W	***F2***
0	*0*	*0*	*0*	***0***	***0***	***0***
0	*0*	*0*	*1*	***0***	***1***	***1***
0	*0*	*1*	*0*	***0***	***0***	***0***
0	*0*	*1*	*1*	***0***	***0***	***0***
0	*1*	*0*	*0*	***0***	***0***	***0***
0	*1*	*0*	*1*	***0***	***0***	***0***
0	*1*	*1*	*0*	***1***	***0***	***1***
0	*1*	*1*	*1*	***1***	***0***	***1***
1	*0*	*0*	*0*	***0***	***0***	***1***
1	*0*	*0*	*1*	***0***	***1***	***1***
1	*0*	*1*	*0*	***0***	***0***	***1***
1	*0*	*1*	*1*	***0***	***0***	***1***
1	*1*	*0*	*0*	***0***	***0***	***1***
1	*1*	*0*	*1*	***0***	***0***	***1***
1	*1*	*1*	*0*	***1***	***0***	***1***
1	*1*	*1*	*1*	***1***	***0***	***1***

SCHEMA D'UN CIRCUIT LOGIQUE (LOGIGRAMME)

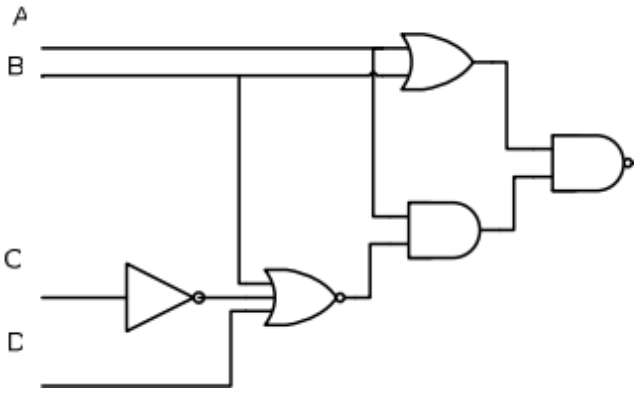
•C'est la traduction de la fonction logique en un schéma électronique.

•Le principe consiste à remplacer chaque opérateur logique par la porte logique qui lui correspond.

F(A,B,C) = AB + $\bar{B}$**C**



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



FORMES CANONIQUES

A	B	C		S		
0	0	0		0	→	$A+B+C$: max terme
0	0	1		0	→	$A+B+\overline{C}$: max terme
0	1	0		0	→	$A+\overline{B}+C$: max terme
0	1	1		1	→	$\overline{A}.B.C$: min terme
1	0	0		0	→	$\overline{A}+B+C$: max terme
1	0	1		1	→	$A.\overline{B}.C$: min terme
1	1	0		1	→	$A.B.\overline{C}$: min terme
1	1	1		1	→	$A.B.C$: min terme

On appel forme canonique d'une fonction la forme ou chaque terme de la fonction comportent toutes les variables.

• Exemple :

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

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

Première forme canonique

• Première forme canonique (forme disjonctive) : somme de produits

• C'est la somme des min termes.

• Une disjonction de conjonctions.

• Exemple :

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

Cette forme est la forme la plus utilisée

Deuxième forme canonique

• Deuxième forme canonique (conjonctive): produit de sommes

• Le produit des max termes

• Conjonction de disjonctions

• Exemple :

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

La première et la deuxième forme canonique sont équivalentes.

Remarque

• On peut toujours ramener n'importe qu'elle fonction logique à l'une des formes canoniques.

• Cela revient à rajouter les variables manquants dans les termes qui ne contiennent pas toutes les variables (les termes non canoniques).

• Cela est possible en utilisant les règles de l'algèbre de Boole :

– Multiplier un terme avec une expression qui vaut 1

– Additionner à un terme avec une expression qui vaut 0

– Par la suite faire la distribution

Exemple

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

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

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

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

2. $F(A, B, C) = AB + C$

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

$= ABC + AB\overline{C} + AC + \overline{A}C$

$= ABC + AB\overline{C} + AC(B + \overline{B}) + \overline{A}C\,(B + \overline{B}\,)$

$= ABC + AB\overline{C} + ABC + A\overline{B}C + \overline{A}BC + \overline{A}\,\overline{B}C$

$= ABC + AB\overline{C} + A\overline{B}C + \overline{A}\;B\;C + \overline{A}\;\overline{B}\;C$

Simplification de fonctions logiques

Dès que l'on dispose de l'expression d'un circuit logique, il peut être possible de la minimiser pour obtenir une équation comptant moins de termes ou de variables par terme. Cette simplification peut se faire de différentes façons :

- par l'utilisation des propriétés des fonctions logiques
- par l'utilisation des tableaux de Karnaugh
- Depuis une table de vérité

Exemple 1

$$Z = (\overline{A}+B).\ (A+B+D)\overline{D}$$

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

$$\uparrow \qquad\qquad \uparrow \qquad\qquad\qquad\qquad \uparrow$$

$$0 \qquad\qquad 0 \qquad\qquad\qquad\qquad 0$$

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

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

$$= B\ .\overline{D}.(\overline{A}+A+1)$$

$$= B.\ \overline{D}$$

Exemple 2

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

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

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

$$= \overline{A} + \overline{B} \qquad \text{car } X + \overline{X}.\ Y = X + Y$$

LE DIGRAMME DE KARNAUGH

Le digramme de Karnaugh est un outil graphique qui permet de simplifier de manière mathématique une expression Booléenne d'un circuit logique.

Les règles à suivre pour l'élaboration d'un tableau de Karnaugh sont les suivantes :

CD / AB	0 0 $\overline{C}$ $\overline{D}$	0 1 $\overline{C}$ D	1 1 C D	1 0 C $\overline{D}$
0 0 $\overline{A}$ $\overline{B}$	$\overline{A}$ $\overline{B}$ $\overline{C}$ $\overline{D}$		$\overline{A}$ $\overline{B}$C D	
0 1 $\overline{A}$ B		A B $\overline{C}$ D		
1 1 A B	A B $\overline{C}$ D		A B CD	
1 0 A $\overline{B}$				A $\overline{B}$ C $\overline{D}$

- *L*e diagramme de Karnaugh d'une fonction logique à n variables est constitué d'un rectangle divisé en 2N cases. Chaque case correspond donc à une combinaison de l'état logique que peut prendre la variable de sortie.
- L'ordre des variables en abscisse et en ordonnée est choisi de telle sorte que lorsqu'on passe d'une case à une case adjacente, une seule variable soit modifiée (on codifie le tableau selon le code Gray **«00,01,11,10 »**).
- Le regroupement des cases doit être égale à une puissance de 2, soit 2, 4, 8…cases. Pour le regroupement des cases périphériques, on suppose que le tableau est de forme cylindrique, et pour le regroupement des cases de coins, on peut supposer que le tableau à la forme d'une sphère.

Etapes de simplification par la méthode de Karnaugh

1. On détermine le nombre de variables d'entrée afin de connaître la taille du tableau.
2. Dresser le tableau de Karnaugh en réspectant le code de Gray
3. Remplir le tableau selon les termes de la fonctions à simplifier
4. Effectuer les regroupements de façon optimale
5. Effectuer les groupements de cases adjacentes.
6. Faire sortir les termes de la fonction simplifiées

A. Cas de deux variables

	$\bar{B}$	B
$\bar{A}$	1	0
A	0	1

$X = A\bar{.}B\bar{} + A.B$

	$\bar{B}$	B
$\bar{A}$	1	1
A	0	1

$X = \bar{A} + B$

B. Cas de trois variables

	$\bar{C}$	C
$\bar{A}\,\bar{.}\,\bar{B}$	0	0
$\bar{A}\,.B$	1	0
A .B	1	0
A $\bar{.}\,\bar{B}$	0	0

$X = B\,\bar{.}\,\bar{C}$

	$\bar{C}$	C
$\bar{A}\,\bar{.}\,\bar{B}$	0	0
$\bar{A}\,.B$	1	1
A .B	0	0
A $\bar{.}\,\bar{B}$	0	0

$X = \bar{A}\,.\,B$

Cas de quatre variables

	$\bar{C}\,\bar{.}\,\bar{D}$	$\bar{C}\,.D$	C .D	C $\bar{.}\,\bar{D}$
$\bar{A}\,\bar{.}\,\bar{B}$	0	0	0	1
$\bar{A}\,.B$	0	1	1	0
A .B	0	1	1	0
A $\bar{.}\,\bar{B}$	0	0	1	0

$X = \bar{A}.\,\bar{B}\,.C\,\bar{.}\,\bar{D} + A\,.C\,D + B\,.D$

	$\bar{C}.\bar{D}$	$\bar{C}.D$	$C.D$	$C.\bar{D}$
$\bar{A}.\bar{B}$	0	1	0	0
$\bar{A}.B$	0	1	1	1
$A.B$	0	0	0	1
$A.\bar{B}$	1	1	0	1

$X = \bar{A}.B.D + B.C.\bar{D} + \bar{B}.\bar{C}.D + A.\bar{B}.\bar{D}$

Cas particulier et élément indéterminé

Il arrive parfois qu'une fonction soit indéfinie pour certaines combinaisons des variables, pour différentes raisons ; la plus courante est que certaines combinaisons des variables étant impossibles, on ne juge pas utile de donner une valeur particulière à la fonction pour ces combinaisons là. 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

F = { 4, 7, 6} + Ø { 1,5}

BC / A	0 0	0 1	1 1	1 0
0	0 (0)	X (1)	0 (3)	0 (2)
1	1 (4)	X (5)	1 (7)	1 (6)

On obtient ici l'expression la plus simple de F en transformant le X de la case 5 en «1» , ce qui permet de regrouper les cases 4, 5, 7, 6 en un groupe de 4 et en laissant le X de la case 1 tel qu'il est Nous aurons donc : **F = A.**

On considéré le « X » comme étant un « 1 » s'il peut nous donner un regroupement meilleur