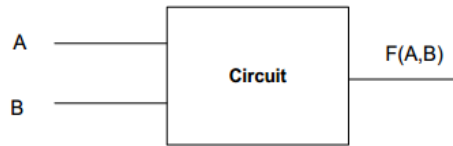


Chapitre 0 : Algèbre de Boole

1. Introduction

- Les machines numériques sont constituées d'un ensemble de circuits électroniques.
- Chaque circuit fournit une fonction logique bien déterminée (addition, comparaison,...).



La fonction $F(A, B)$ peut être : la somme de A et B , ou le résultat de la comparaison de A et B ou une autre fonction

- Pour concevoir et réaliser ce circuit on doit avoir un modèle mathématique de la fonction réalisée par ce circuit. Ce modèle doit prendre en considération le système binaire.
- Le modèle mathématique utilisé est celui de Boole

2. Algèbre de Boole

C'est une algèbre binaire définie en 1847 par Georges Boole (1815-1864), physicien Anglais

- Algèbre applicable au raisonnement logique qui traite des fonctions à variables binaires (deux valeurs).
- Ne s'applique pas aux systèmes à plus de deux états d'équilibre.
- Permet d'étudier les circuits logiques (un système logique sert à modifier des signaux).

Exemple de systèmes à deux états

- Un interrupteur est ouvert ou non ouvert (fermé)
- Une lampe est allumée ou non allumée (éteinte)
- Une porte est ouverte ou non ouverte (fermée)

3. Définitions

3.1. Variable logique (booléenne)

- Une variable logique (booléenne) est une variable qui peut prendre soit la valeur 0 ou 1 .
- Généralement elle est exprimée par un seul caractère alphabétique en majuscule ($A, B, S, \dots$)

Exemple

- Une lampe : allumée $L = 1$ éteinte $L = 0$

3.2. Fonction logique

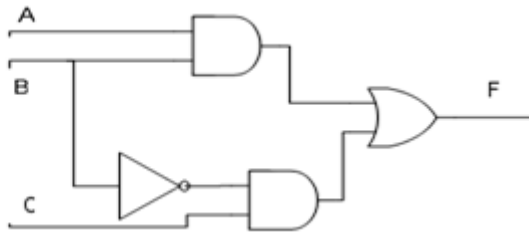
- C'est une fonction qui relie **N variables logiques** avec un ensemble **d'opérateurs logiques** de base.
- Dans l'Algèbre de Boole il existe trois opérateurs de base : **NON , ET , OU**.
- La valeur d'une fonction logique est **égale à 1 ou 0** selon les valeurs des variables logiques.
- Si une fonction logique possède N variables logiques $\longrightarrow 2^n$ combinaisons $\longrightarrow$ la fonction possède 2^n valeurs.
- Les 2^n combinaisons sont représentées dans une table qui s'appelle table de vérité (TV).

3.3. Schéma 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.

Exemple

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



4. Opérateurs logiques de base

Une porte logique est un circuit électronique élémentaire qui permet de réaliser la fonction d'un **opérateur logique de base**.

4.1. Fonction logique NON (NOT : négation) : à pour rôle d'inverser la valeur d'une variable.

$$F(A) = \text{Non } A = \bar{A}$$

(lire : A barre)

Table de vérité

Entrée	Sortie
A	F
0	1
1	0



Symbole graphique

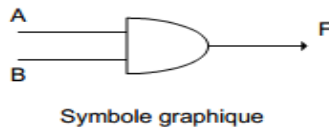
4.2. Fonction logique ET (AND) : à pour rôle de réaliser le Produit logique entre deux variables booléennes.

- Le ET fait la conjonction entre deux variables.

Le ET est défini par $F(A,B) = A * B$ ou $A \cdot B$ ou AB

Table de vérité

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



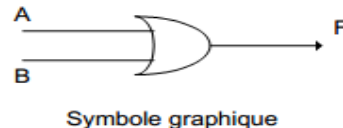
4.3. Fonction logique OU (OR) : à pour rôle de réaliser la somme logique entre deux variables logiques.

- Le OU fait la disjonction entre deux variables

Le OU est Défini par $F(A,B) = A + B$

Table de vérité

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



4.4. Précédence des opérateurs (priorité des opérateurs)

- Pour évaluer une expression logique (fonction logique) :
 - on commence par évaluer les sous expressions entre les parenthèses. puis le complément (NON) , en suite le produit logique (ET). Enfin la somme logique (OU)

Exercice

Trouver la table de vérité de la fonction $F(A, B, C) = (\overline{A} \cdot \overline{B}) \cdot (C + B) + A \cdot \overline{B} \cdot C$?

Solution

- Pour trouver la table de vérité de la fonction F, il faut trouver sa valeur pour chaque combinaisons des trois variables A, B , C
 - 3 variables $\rightarrow 2^3 = 8$ combinaisons

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

$$F(0,0,0) = (\overline{0} \cdot \overline{0}) \cdot (0 + 0) + 0 \cdot \overline{0} \cdot 0 = 0$$

$$F(0,0,1) = (\overline{0} \cdot \overline{0}) \cdot (1 + 0) + 0 \cdot \overline{0} \cdot 1 = 1$$

$$F(0,1,0) = (\overline{0} \cdot \overline{1}) \cdot (0 + 1) + 0 \cdot \overline{1} \cdot 0 = 1$$

$$F(0,1,1) = (\overline{0} \cdot \overline{1}) \cdot (1 + 1) + 0 \cdot \overline{1} \cdot 1 = 1$$

$$F(1,0,0) = (\overline{1} \cdot \overline{0}) \cdot (0 + 0) + 1 \cdot \overline{0} \cdot 0 = 0$$

$$F(1,0,1) = (\overline{1} \cdot \overline{0}) \cdot (1 + 0) + 1 \cdot \overline{0} \cdot 1 = 1$$

$$F(1,1,0) = (\overline{1} \cdot \overline{1}) \cdot (0 + 1) + 1 \cdot \overline{1} \cdot 0 = 0$$

$$F(1,1,1) = (\overline{1} \cdot \overline{1}) \cdot (1 + 1) + 1 \cdot \overline{1} \cdot 1 = 0$$

A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

5. Autres opérateurs logiques

5.1. OU exclusif (XOR)

$$F(A, B) = A \oplus B$$

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

A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0



5.2. NAND (NON ET)

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

$$F(A, B) = A \uparrow B$$

A	B	$\overline{A \cdot B}$
0	0	1
0	1	1
1	0	1
1	1	0



5.3. NOR (NON OU)

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

$$F(A, B) = A \downarrow B$$

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



5.4. NAND et NOR sont des opérateurs universels

- En utilisant les NAND et les NOR on peut exprimer n'importe quelle expression (fonction) logique.
- Pour cela, Il suffit d'exprimer les opérateurs de base (NON , ET , OU) avec des NAND et des NOR.

5.4.1. Réalisation des opérateurs de base avec des NOR

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

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

$$A \cdot B = \overline{\overline{A \cdot B}} = \overline{\overline{A} + \overline{B}} = \overline{\overline{A} \downarrow \overline{B}} = (A \downarrow A) \downarrow (B \downarrow B)$$

5.4.2. Propriétés des opérateurs NAND et NOR

$$\left. \begin{array}{l} A \uparrow 0 = 1 \\ A \uparrow 1 = \overline{A} \\ A \uparrow B = B \uparrow A \\ (A \uparrow B) \uparrow C \neq A \uparrow (B \uparrow C) \end{array} \right| \begin{array}{l} A \downarrow 0 = \overline{A} \\ A \downarrow 1 = 0 \\ A \downarrow B = B \downarrow A \\ (A \downarrow B) \downarrow C \neq A \downarrow (B \downarrow C) \end{array}$$

6. Lois fondamentales de l'Algèbre de Boole

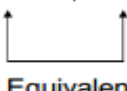
Fermeture	Si A et B sont des variables booléennes, alors $A+B$, AB sont aussi des variables booléennes
Commutativité	$A+B = B+A$ $A \cdot B = B \cdot A$
Associativité	$A+(B+C) = (A+B)+C$ $A \cdot (B \cdot C) = (A \cdot B) \cdot C$
Distributivité	ET / OU $A(B+C) = AB+AC$ OU / ET $A+(BC) = (A+B)(A+C)$
Idempotence	$A+A = A$ $A \cdot A = A$
Complémentarité	$A+\bar{A} = 1$ $A \cdot \bar{A} = 0$
Identités remarquables	$1+A = 1$ $1 \cdot A = A$ $0+A = A$ $0 \cdot A = 0$
Distributivité interne	$A+(B+C) = (A+B)+(A+C)$ $A \cdot (B \cdot C) = (A \cdot B) \cdot (A \cdot C)$

7. Théorème de DE-MORGANE

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

Vérification :

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

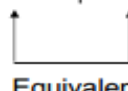


Equivalent

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

Vérification :

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



Equivalent

8. Simplification des fonctions logiques

- L'objectif est de :
 - réduire le nombre de termes dans une fonction
 - et de réduire le nombre de variables dans un terme
- Cela afin de réduire le nombre de portes logiques utilisées → réduire le coût du circuit
- Plusieurs méthodes existent pour la simplification:
 - La Méthode algébrique(en utilisant des propriétés et des théorèmes)
 - Les Méthodes graphiques : (ex : table de karnaugh)
 - Les méthodes programmables

8.1. Méthode algébrique

• Le principe consiste à appliquer les règles de l'algèbre de Boole afin d'éliminer des variables ou des termes. Mais il n'y a pas une démarche bien spécifique.

• Voici quelques règles les plus utilisées

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

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

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

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

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

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

• **Règles 1** : regrouper des termes à l'aide des règles précédentes

Exemple

$$\begin{aligned} ABC + AB\overline{C} + A\overline{B}CD &= AB(C + \overline{C}) + A\overline{B}CD \\ &= AB + A\overline{B}CD \\ &= A(B + \overline{B}(CD)) \\ &= A(B + CD) \\ &= AB + ACD \end{aligned}$$

• **Règles 2** : Rajouter un terme déjà existant à une expression

Exemple

$$\begin{aligned} ABC + \overline{A}BC + A\overline{B}C + AB\overline{C} &= \\ ABC + \overline{A}BC + ABC + A\overline{B}C + ABC + AB\overline{C} &= \\ BC + AC + AB & \end{aligned}$$

• **Règles 3** : il est possible de supprimer un terme superflu (un terme en plus), c'est-à-dire déjà inclus dans la réunion des autres termes.

Exemple

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

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

- **Règles 4** : il est préférable de simplifier la forme canonique ayant le nombre de termes minimum.

Exemple

$$\begin{aligned}
 F(A, B, C) &= R(2, 3, 4, 5, 6, 7) \\
 \overline{F(A, B, C)} &= R(0, 1) = \overline{A} \cdot \overline{B} \cdot \overline{C} + \overline{A} \cdot \overline{B} \cdot C \\
 &= \overline{A} \cdot \overline{B} (\overline{C} + C) \\
 &= \overline{A} \cdot \overline{B} = \overline{A + B} \\
 F(A, B, C) &= \overline{\overline{F(A, B, C)}} = \overline{\overline{A + B}} = A + B
 \end{aligned}$$

8.2 Simplification par la table de Karnaugh

- La méthode consiste à mettre en évidence par une méthode graphique (un tableau) tous les termes qui sont adjacents (qui ne diffèrent que par l'état d'une seule variable).
- La méthode peut s'appliquer aux fonctions logiques de 2, 3, 4, 5 et 6 variables.
- Un tableau de Karnaugh comporte 2^n cases (N est le nombre de variables)

A

B

0

1

0		
1		

Tableau à 2 variables

AB

C

00

01

11

10

0				
1				

Tableaux à 3 variables

AB

CD

00

01

11

10

00				
01				
11				
10				

Tableau à 4 variables

AB

CD

00

01

11

10

00				
01				
11				
10				

U = 0

AB

CD

00

01

11

10

00				
01				
11				
10				

U = 1

Exemple de termes adjacents

Ces termes sont adjacents

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

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

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

Ces termes ne sont pas adjacents

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

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

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

Dans un tableau de Karnaugh, chaque case possède un certain nombre de cases adjacentes.

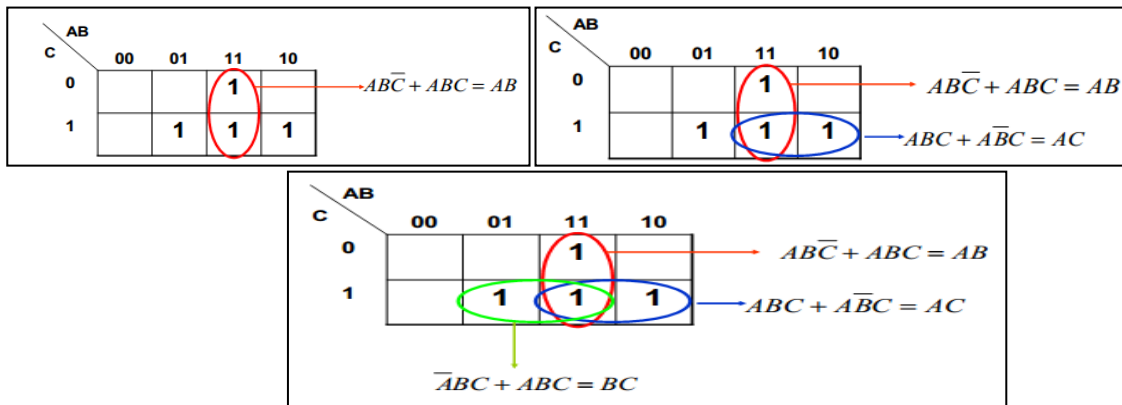
AB \ C	00	01	11	10
0	•	•	•	
1		•		

Les trois cases bleues sont des cases adjacentes à la case rouge

AB \ CD	00	01	11	10
00		•		
01	•	•	•	
11		•		
10				

9. Des exemples de la Méthode de simplification par la table de Karnaugh

Exemple 0 : 3 variables



• La fonction final est égale à la réunion (somme) des termes après simplification.

$$F(A, B, C) = AB + AC + BC$$

