

ARCHITECTURE DES ORDINATEURS

Plan Du Cours

Chapitre 1 : Introduction à la technologie des ordinateurs

Chapitre 2 : Représentation de l'information

Chapitre 3 : Fonction et opérateurs logiques

Chapitre 4 : Les circuits intégrés et combinatoires

Chapitre 1 : Introduction à la technologie des ordinateurs

I. Introduction

L'homme a toujours eu besoin de compter, il a donc eu à développer selon les civilisations des systèmes de numérations (les doigts de la main et des pieds, les cailloux, les articulations des doigts ou des phalanges). Les doigts ont servi à nos ancêtres pour compter et pour effectuer toutes sortes d'opérations arithmétiques. La plus naturelle et la plus répandue des systèmes de numération est celle qui compte en **base (10)**. Dans la nouvelle science, on parle de calcul digital. Le mot **digit** ayant le sens de chiffre d'une numération de position. Dans les ordinateurs, on utilise des **digits binaires** ou **bit** (contraction du mot anglais « binary digits » : unité élémentaire d'information). C'est l'écriture des nombres comportent que les deux chiffres **0 et 1**. Le tableau suivant montre les principales bases utilisées :

Base 1	Comptage avec les doigts, cailloux, bâtonnets
Base 2	Système Binaire, logique symboliques : utilisés par les ordinateurs
Base 5	Système quinaire : civilisation Aztèque
Base 7	Les jours de la semaine
Base 8	Système Octal : utilisés par les ordinateurs
Base 10	Système Décimal : l'homme
Base 12	Heures, les mois de l'année
Base 16	Système Hexadécimal (0,..., 9, A, B, C, D, E, F)
Base 20	Comptage des mains et pieds
Base 24	Heures du jour
Base 60	Système Sexagésimal : minutes, seconde, degrés

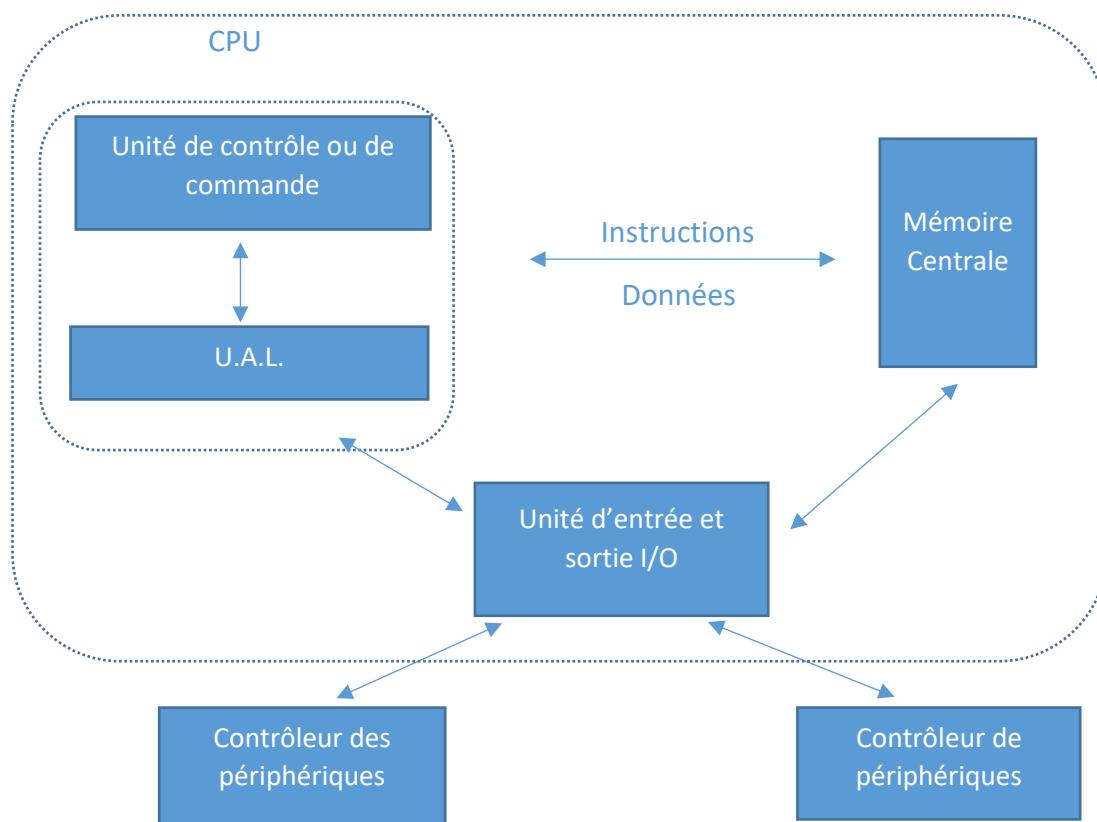
II. Naissance de l'ordinateur

Parallèlement à l'évolution des signes, des chiffres, des calculs mentaux et manuels on a assisté au développement d'outils, de systèmes, de machines pour simplifier et accélérer les tâches. C'est ainsi en **1945** que **John Von Neumann** (Mathématicien d'origine Hongroise de la Hongrie) proposa la machine modèle de l'ordinateur telle qu'il est conçu actuellement. On peut résumer les caractéristiques de l'ordinateur selon John Von Neumann :

- Machine universelle qui est contrôlée par un programme
- Les instructions du programme sont codées sous formes numériques (Binaire et enregistré en mémoire)
- Le programme peut modifier ses propres instructions qui sont normalement exécuté en séquence
- Des instructions qui existent permettant des ruptures de séquences.

Van Neumann décrit en **1945** les **5 composants essentiels** de ce qui aller être **l'Architecture de Von Neumann** :

- L'Unité Arithmétique et Logique (UAL)
- L'Unité de Commandes
- La Mémoire Centrale
- L'Unité d'Entrée
- L'Unité de Sortie



III. Le système informatique

L'informatique (Computer Science) est définie comme la science du traitement automatique de l'information. L'information est la matière à manipuler et traiter par les ordinateurs, ils doivent être capable de la comprendre tout comme l'homme comprend une langue et des concepts. Le système, c'est l'ensemble des moyens logiciels (**Software**) et matériels (**Hardware**) nécessaire pour satisfaire les besoins informatiques des utilisateurs.

Un système informatique est donc un système de traitement à qui l'on remet des informations en **entrée** qui exécute un traitement sur ces informations et qui restitue en fin des informations en **sortie**. L'ordinateur est une machine électronique spécialement conçue pour permettre le traitement de l'information. Une telle machine accepte à l'entrée des données (numérique ou non) effectue le traitement en fonction d'un programme enregistré dans sa mémoire et produit des résultats sous forme compréhensible à l'utilisateur. On peut établir en parallèle entre son fonctionnement et la façon dont un employé de bureau.



Pour exécuter son travail, l'employeur peut avoir besoin d'une unité de calcul (Calculatrice) d'un programme de travail (instructions données par le patron) qu'il doit mémoriser (mémoire vive RAM et la mémoire morte ROM).

D'autres informations dont l'employeur peut avoir besoin pour son travail sont stockées dans des fichiers ou des mémoires auxiliaires.

La configuration d'un ordinateur correspond à l'organisation adaptée pour mettre ensemble et faire fonctionner les divers éléments matériels d'un ordinateur.

1. Processeur Central (CPU : Central Processing Unit)

Appelé aussi **microprocesseur** ou **puce** correspond au moteur de la machine (situé sur la carte mère au fond d'un panier). Le processeur central contient deux unités : **l'unité de commande** et **l'unité de calcul**.

L'unité de commande s'occupe de gérer l'exécution des instructions d'un programme. Elle contient le registre d'instruction (**RI**) qui contient l'instruction et le compteur ordinal (**CO**) qui contient l'adresse de la prochaine instruction à exécuter.

2. La mémoire centrale

Contient principalement deux types d'information :

- **Les instructions des différents programmes** ;
- **Les données nécessaires à l'exécution d'un programme.**

Les instructions sont stockées sous forme de codes machines. La mémoire centrale est divisée physiquement en cellule. Chaque cellule correspond à une mémoire. C'est l'unité d'information adressable (la longueur d'un mot mémoire varie d'une machine à l'autre). A chaque mot mémoire est associé :

- **Une adresse (unique) indiquant la position en mémoire**
- **Un contenu d'instructions ou données**

3. Les unités d'entrées et sorties

Ce sont éléments qui permettent de transférer des informations entre les unités centrales et les périphériques. Les unités d'entrées et de sorties les plus courants sont les Bus.

4. Les unités périphériques

Elles se répartissent en deux classes :

- **Les unités qui permettent à l'ordinateur d'échanger les données avec l'extérieur (écran, clavier, modem ...)**
- **Les mémoires auxiliaires qui permettent de stocker de manière permanente beaucoup d'information en moindre coût (disque magnétique, CD, DVD).**

A chaque catégorie d'unités périphériques est associé un contrôleur de périphériques qui s'occupe de la gestion de ces unités et de l'interface avec les entrées et les sorties.

Conclusion

Pour résoudre un problème à l'aide d'un ordinateur on doit utiliser un algorithme exprimé sous la forme d'un programme. Les ordinateurs arrivent à des vitesses de l'ordre de plusieurs d'opérations arithmétiques par seconde alors que le cerveau humain est plus lent. Pourtant certaines opérations complexes telles que la reconnaissance d'un objet sont réalisés beaucoup plus rapidement par le cerveau que par un ordinateur. Ceci est dû à une différence fondamentale de structures entre l'ordinateur et le cerveau humain.

Chapitre 2 : Représentation de l'information

I. Introduction

La tâche principale d'un ordinateur est le traitement de l'information numérique et celle-ci peut être représenté sur plusieurs formes (binaire, octal, décimal, hexadécimal). L'objectif de ce chapitre est de montrer différentes formes de représentations binaires pour les données numériques.

Règles générales :

Les systèmes de numération utilisent tous certains symboles pour exprimer des nombres. Chaque système de numération est défini selon une **base B** et comprend un ensemble de nombre entiers compris entre 0 et B-1. Soit **N** un nombre entier si B est la base alors $N = \sum_{i=0}^n a_i b^i$ Avec $a_i \in [0; B - 1]$.

a_0 est le type le moins significatif ou la byte de poids faibles (**LSB : Least Significant Bit**)

a_n est le byte le plus significatif ou la byte de poids fort (**MSB : Most Significant Bit**)

II. Cas des nombres entiers

1. Système de numération binaire

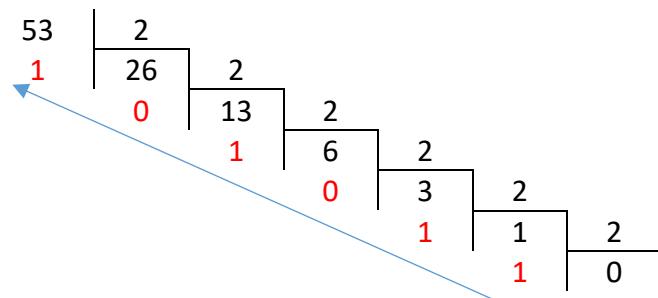
Le système de numération binaire dont la base est 2 exprime les nombres à l'aide de 2 chiffres (**0 et 1**). Le système binaire est un système à poids positionnel puisque chaque chiffre binaire est affecté d'un poids exprimé comme une puissance de 2.

- Conversion – Décimal – Binaire

Soit A un nombre décimal à représenter en base B la méthode utilisée est celle de la division euclidienne

- Si $A < B$ il n'a pas besoin d'être converti
- Si $A \geq B$ on peut diviser A par B et on divise successivement les différents quotients q obtenus par la base B

Exemple : Conversion de $(53)_{10}$ en base 2 :



$$(53)_{10} = (110101)_2$$

Autre méthode :

0	1	3	6	13	26	53
	1	1	0	1	0	1

- Conversion Binaire – Décimale

$$(11100101)_2 = (1 * 2^7 + 1 * 2^6 + 1 * 2^5 + 0 * 2^4 + 0 * 2^3 + 1 * 2^2 + 0 * 2^1 + 1 * 2^0)_{10} \\ = (229)_{10}$$

- Autre Exemple

$$(11100101)_2 \Rightarrow \begin{aligned} 1 * 2 + 1 &= 3 & 28 * 2 + 1 &= 57 \\ 3 * 2 + 1 &= 7 & 57 * 2 + 0 &= 114 \\ 7 * 2 + 0 &= 14 & 114 * 2 + 1 &= 229 \\ 14 * 2 + 0 &= 28 \end{aligned}$$

- Conversion Décimale – Octale

$$(266)_{10} = (412)_8$$

0	4	33	266
	4	1	2

- Conversion Octale – Décimale

$$(372)_8 = 3 * 8^2 + 7 * 8^1 + 2 * 8^0 = (250)_{10}$$

- Conversion Octale – Binaire

Pour convertir un nombre octal en son équivalent binaire il suffit de transformer chaque chiffre du nombre octal en son équivalent binaire de trois chiffres. Le tableau suivant exprime chacun des huit symboles octaux en binaire :

Chiffre Octal	Equivalent Binaire
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Exemple :

$$(372)_8 = ?$$

On a :

$$3 \rightarrow 011$$

$$7 \rightarrow 111$$

$$2 \rightarrow 010$$

$$\text{Donc } (372)_8 = (011\ 111\ 010)_2$$

- Conversion Binaire – Octale

La conversion binaire en nombre octal se fait en groupage du nombre binaire en paquet de trois bits en partant du bit de poids le plus faible (LSB) puis en convertissant les triplets en leur équivalent octal

Exemple

$$(100\ 110\ 101)_2 = ?$$

On a :

$$100 \rightarrow 4$$

$$110 \rightarrow 6$$

$$101 \rightarrow 5$$

$$\text{Donc } (100\ 110\ 101)_2 = (465)_8$$

2. Système de numération Hexadécimal

Il requiert 16 symboles pour exprimer les chiffres. Par convention les 10 premiers symboles sont empruntés au système de numération décimal (0, ..., 9) et les 6 derniers sont les lettres de l'alphabet latin (A, ..., F).

- Conversion Décimale – Hexadécimale

$$\text{Exemple } (423)_{10} = (?)_{16}$$

0	1	26	423
	1	A	7

$$\text{Donc } (423)_{10} = (1A7)_{16}$$

- **Conversion Hexadécimale – Décimale**

$$\text{Exemple } (F5B)_{16} = (15 * 16^2 + 5 * 16^1 + 11 * 16^0)_{10} \\ = (3931)_{10}$$

- **Conversion Hexadécimale – Binaire**

Pour écrire un nombre hexadécimal en son équivalent binaire il suffit de transformer chaque chiffre du nombre hexadécimal en son équivalent binaire sur 4 chiffres.

Le tableau suivant exprime chacun des 16 symboles en son équivalent binaire :

Chiffre Hexadécimal	Equivalent Binaire
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

Exemples :

$$(D73A)_{16} = (?)_2$$

$$D \rightarrow 1101$$

$$7 \rightarrow 0111$$

$$3 \rightarrow 0011$$

$$A \rightarrow 1010$$

$$(D73A)_{16} = (1101\ 0111\ 0011\ 1010)_2$$

- **Conversion Binaire – Hexadécimale**

Même procédure de la conversion octale sauf qu'il faut des paquets de 4 bites

Exemple :

$$(0110\ 0011\ 0101\ 1100)_2 = (?)_{16}$$

$$0110 \rightarrow 6$$

$$0011 \rightarrow 3$$

$$0101 \rightarrow 5$$

$$1100 \rightarrow C$$

$$\text{Donc } (0110\ 0011\ 0101\ 1100)_2 = (635C)_{16}$$

III. Cas des nombres fractionnaires

La conversion des nombres fractionnaires se fait selon les mêmes principes

1. Conversion sur le système décimal

Quelque soit M un nombre fractionnaire si b est la base :

$$M = a_m b^m + a_{m-1} b^{m-1} + \dots + a_1 b^1 + a_0 b^0 + a_{-1} b^{-1} + a_{-2} b^{-2} + \dots \\ \text{avec } a_i \in [0, b - 1]$$

Méthode 1 :

$$(1001, 1101)_2 = (?)_{10}$$

$$(1001, 1101)_2 = (1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 1 \cdot 2^{-2} + 0 \cdot 2^{-3} + 1 \cdot 2^{-4})_{10}$$

$$= (9,8125)_{10}$$

Méthode 2 :

Partie entière :

$$1 \cdot 2 + 0 = 2$$

$$2 \cdot 2 + 0 = 4$$

$$4 \cdot 2 + 1 = 9$$

Partie décimale :

$$1/2 + 0 = 0,5$$

$$0,5/2 + 1 = 1,25$$

$$1,25/2 + 1 = 1,625$$

$$1,625/2 = 0,8125$$

$$(1001, 1101)_2 = (9 + 0,8125)_{10}$$

$$= (9,8125)_{10}$$

2. Conversion d'un nombre décimal en nombre binaire :

Exemple :

$$(27,3)_{10} = (?)_2$$

Partie entière :

0	1	3	6	13	27
	1	1	0	1	1

Partie décimale :

$$0,3 \cdot 2 = 0,6 \rightarrow 0$$

$$0,6 \cdot 2 = 1,2 \rightarrow 1$$

$$0,2 \cdot 2 = 0,4 \rightarrow 0$$

$$0,4 \cdot 2 = 0,8 \rightarrow 0$$

$$(27,3)_{10} = (1101, 0100)_2$$

3. Conversion d'un nombre binaire en un nombre Octal ou hexadécimal

Exemple 1 :

$$(010\ 111\ 110\ 101,\ 011\ 010)_2 = (?)_8$$

$$010 \rightarrow 2$$

$$111 \rightarrow 7$$

$$110 \rightarrow 6$$

$$101 \rightarrow 5$$

$$011 \rightarrow 3$$

$$010 \rightarrow 2$$

$$(010\ 111\ 110\ 101,\ 011\ 010)_2 = (2765,32)_8$$

Exemple 2 :

$$(0101\ 1111\ 0101\ 0110\ 1000)_2 = (?)_{16}$$

$$0101 \rightarrow 5$$

$$1111 \rightarrow F$$

$$0101 \rightarrow 5$$

$$0110 \rightarrow 6$$

$$1000 \rightarrow 8$$

$$(0101\ 1111\ 0101\ 0110\ 1000)_2 = (5F568)_{16}$$

IV. Opérations Arithmétiques binaires :

Les opérations arithmétiques élémentaires sur les nombres binaires d'un seul bit sont assez simples et accessibles à la table d'addition binaire

Table d'addition Binaire :

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 10$$

Table de soustraction binaire :

$$0 - 0 = 0$$

$$1 - 0 = 1$$

$$10 - 1 = 1$$

Table de multiplication binaire

$$0 * 0 = 0$$

$$0 * 1 = 0$$

$$1 * 0 = 0$$

$$1 * 1 = 1$$

Exemple : $(19)_{10} + (23)_{10} = (42)_{10}$

$$(19)_{10} = (?)_2$$

0	1	2	4	9	19
	1	0	0	1	1

$$(23)_{10} = (?)_2$$

0	1	2	5	11	23
	1	0	1	1	1

$$\begin{array}{r}
 & 1 & 0 & 0 & 1 & 1 \\
 + & & & & & \\
 & 1 & 0 & 1 & 1 & 1 \\
 \hline
 & 1 & 0 & 1 & 0 & 1 & 0
 \end{array}$$

Exemple : $(26)_{10} - (17)_{10} = (9)_{10}$

$$(26)_{10} = (?)_2$$

0	1	3	6	13	26
	1	1	0	1	0

$$(17)_{10} = (?)_2$$

0	1	2	4	8	17
	1	0	0	0	1

$$\begin{array}{r}
 & 1 & 1 & 0 & 1 & 0 \\
 - & & & & & \\
 & 1 & 0 & 0 & 0 & 1 \\
 \hline
 & 0 & 1 & 0 & 0 & 1
 \end{array}$$

Exemple : $(11)_{10} * (13)_{10} = (143)_{10}$

$$(11)_{10} = (?)_2$$

0	1	2	5	11
	1	0	1	1

$$(13)_{10} = (?)_2$$

0	1	3	6	13
	1	1	0	1

$$\begin{array}{r}
 * \\
 & 1 & 1 & 0 & 1 \\
 & 1 & 0 & 1 & 1 \\
 & 0 & 0 & 0 & 0 & . \\
 & 1 & 0 & 1 & 1 & . & . \\
 \hline
 & 1 & 0 & 1 & 1 & . & . & . \\
 \hline
 & 1 & 0 & 0 & 0 & 1 & 1 & 1
 \end{array}$$

Exemple : $(173)_{10} / (5)_{10} = (34)_{10} + \text{reste } (3)_{10}$

$$(26)_{10} = (?)_2$$

0	1	2	5	10	21	43	86	173
1	0	1	0	1	1	0	1	

10101101	101
0000110	<u>100010</u>
00000011	

$$(5)_{10} = (?)_2$$

0	1	2	5
1	0	1	

$$(10101101)_2 / (101)_2 = (100010)_2 + \text{reste } (11)_2$$

V. Cas des nombres signés

Quelques méthodes sont généralement utilisées pour représenter les nombres signés, **complément à 1** (CA1), **complément à 2** (CA2).

1. Le binaire signé :

Ce codage permet de représenter des nombres entiers relatifs. Dans cette représentation le bit le plus à gauche est utilisé pour le signe (MSB)

0 correspond +

1 correspond -

Ainsi un nombre négatif se forme en changeant le bit de signe du nombre positif i.e., 0 à 1

Exemple : Pour un mot de 8 bits

$(+12)_{10} = (00001100)_2$ en binaire signé sur un mot de 8 bits

$(-12)_{10} = (10001100)_2$ en binaire signé sur un mot de 8 bits

Pour un mot de 8 bits, on peut écrire 256 nombres.

Notons qu'avec cette méthode il y a deux représentations pour le zéro :

$(+0)_{10} = (00000000)_2$

$(-0)_{10} = (10000000)_2$

Ainsi seul $2^8 - 1 = 255$ peuvent être représentés.

2. Le complément à 1 ou complément restreint :

Le complément à 1 d'un nombre binaire s'obtient en convertissant tous les 1 en 0 et tous les 0 en 1

Exemple :

$(+12)_{10} = (00001100)_2$ en complément à 1 sur un mot de bits donne

$(-12)_{10} = (11110011)_2$

- On note une fois de plus deux représentations pour le nombre zéro donc 255 nombres qu'on peut écrire en un complément à 1 pour un mot de 8 bits.

En Complément à 1, la soustraction sera obtenue par l'addition du complément restreint du nombre à soustraire et le report de la retenue

Exemple :

$$(45)_{10} - (23)_{10} = (22)_{10}$$

$$(45)_{10} = (?)_2$$

0	1	2	5	11	22	45
1	0	1	1	1	0	1

00101101

- → +

00010111
00010110

$$(23)_{10} = (?)_2$$

0	1	2	5	11	23
1	0	1	1	1	1

00101101

- → +

11101000
1 00010101

+

1
00010110

S'il y a absence de report issu du dernier rang cela signifie que le résultat est négatif et qu'il se présente sous une forme complémentée (Complément restreint). Il suffira donc dans trouver le complément restreint pour retrouver la valeur cherchée.

Exemple :

$$(23)_{10} - (45)_{10} = (-22)_{10}$$

$$\begin{array}{r} 00010111 \\ - 00101101 \\ \hline \end{array} \quad \rightarrow \quad \begin{array}{r} 00010111 \\ + 11010010 \\ \hline 11101001 \end{array}$$

On fait un complément à 1 du résultat qui est un nombre négatif donc on obtient :

$$(11101001)_2 \text{ CA1} \rightarrow (00010110)_2 \rightarrow (-22)_{10}$$

3. Le complément à 2 :

Le complément à 2 d'un nombre binaire s'obtient en faisant d'abord le complément à 1 du nombre puis **en ajoutant 1** au nombre obtenu s'il existe une retenue après addition avec le bit plus significatif alors on l'ignore.

Exemple :

$$(+12)_{10} = (00001100)_2 \text{ CA1} \rightarrow \quad \quad \quad 11110011$$

$$+ \frac{1}{11110100} \rightarrow (-12)_{10}$$

Pour un mot de 8 bits : $(0)_{10} = (00000000)_2$ CA1 → 11111111

$$+\frac{1}{00000000} \rightarrow (0)_{10}$$

On peut écrire 256 nombres en complément à 2 pour un mot de 8 bits.

Chapitre 3 : Fonctions et opérations logiques

I. Algèbre de Boole

1. Définition

L'algèbre de **Boole** ou l'algèbre **booléien** est une méthode mathématique créée par le mathématicien anglais **Georges Boole** dans son ouvrage **The Laws Of Thought (1844)**.

- **Les états logiques**

Ils sont représentés par les chiffres **0** et **1**

- **Une variable logique**

C'est une grandeur représentée par un symbole (**lettre ou signe**) qui peut prendre que les seules valeurs **0, 1** suivant certaines conditions.

- **Une fonction logique**

Elle est représentée par des groupes de variables reliées par des opérateurs logiques.

Ils existent trois opérateurs logiques élémentaire : **NON, ET, OU** (cité par ordre de priorité dans une opération). Une fonction logique tout comme les variables logiques ne peut prendre que deux valeurs **0** ou **1**.

2. Représentation des variables et des fonctions

➤ La table de vérité

Si une fonction possède **n variables** la table de vérité relative à cette fonction est un **tableau de $n+1$ colonnes et 2^n lignes**. Chaque ligne est représentative d'une combinaison de variables parmi les **2^n ligne**. Une dernière colonne est réservée à la valeur de la fonction pour la combinaison des variables correspondantes.

Exemple : $F(A, B) = A \text{ OU } B$

A	B	$F(A, B)$
0	0	0
0	1	1
1	0	1
1	1	1

➤ La table de Karnaugh

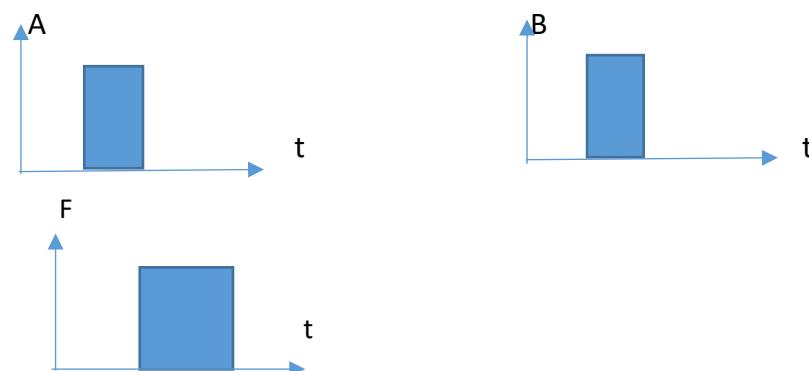
Exemple : $F(A, B) = A \text{ OU } B$

A \ B	0	1
0	0	1
1	1	1

➤ Le chronogramme

C'est le graphe de l'évolution temporel des variables et des fonctions logiques

Exemple $F(A, B) = A \text{ OU } B$

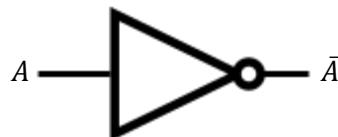


3. Fonctions logiques élémentaires

Trois fonctions élémentaires suffisent pour définir l'algèbre de Boole : La fonction **NON** (complément), la fonction **ET** (produit logique) et la fonction **OU** (addition logique)

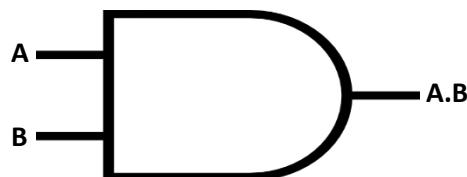
a. La fonction Non (complément)

A	$F(A) = \bar{A}$
0	1
1	0



b. La fonction ET (produit logique)

A	B	$F(A) = A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1



c. La fonction OU (addition logique)

A	B	$F(A) = A + B$
0	0	0
0	1	1
1	0	1
1	1	1



➤ Théorèmes constants

$$A + 0 = A$$

$$A + 1 = 1$$

$$A \cdot 0 = 0$$

$$A \cdot 1 = A$$

➤ Idempotence

$$A + A = A$$

$$A \cdot A = A$$

➤ Complémentation

$$A + \bar{A} = 1$$

$$A \cdot \bar{A} = 0$$

$$\bar{\bar{A}} = A$$

➤ **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**

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

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

Exemple : Démontrer que $A + \bar{A}B = A + B$

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

$$= 1 \cdot (A + B)$$

$$= A + B$$

➤ **Théorème de Morgan : Complément d'une expression**

On a :

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

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

A	B	\bar{A}	\bar{B}	$A \cdot B$	$\overline{A \cdot B} = \bar{A} + \bar{B}$	$A + B$	$\overline{A + B} = \bar{A} \cdot \bar{B}$
0	0	1	1	0	1	0	1
0	1	1	0	0	1	1	0
1	0	0	1	0	1	1	0
1	1	0	0	1	0	1	0

Les théorèmes de Morgan peuvent être appliqués à des produits logiques (à des sommes logiques) comportant un nombre quelconque de variables

$$\overline{A + B + C + \dots + Z} = \bar{A} \cdot \bar{B} \cdot \bar{C} \cdot \dots \cdot \bar{Z}$$

$$\overline{A \cdot B \cdot C \cdot \dots \cdot Z} = \bar{A} + \bar{B} + \bar{C} + \dots + \bar{Z}$$

➤ **Autres relations**

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

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

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

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

II. Représentation des fonctions logiques

Une fonction logique se présente comme une association de sommes et de produits logiques

- Si l'expression est une somme de produits la forme est dite **disjonctive**

Exemple : $F(A, B, C) = AB + AC + \bar{A}B$

- Si l'expression est un produit de sommes la forme est dite **conjonctive**

Exemple : $F(A, B, C) = (A + B) \cdot (A + \bar{C}) \cdot (B + C)$

- Une fonction logique est sous forme **normale** ou **canonique** si chaque terme contient toutes les variables

- **Forme normale disjonctive**

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

Chaque terme est appelé **intersection de base minterm**

- Forme normale conjonctive de vase maxterm

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

Lorsque la forme n'est pas normale on dit qu'elle est sous la **forme simplifiée**

III. Les formes normales

Les formes normales sont obtenues par application directe des théorèmes d'expression de Shannon

1. Forme normale disjonctive

Cette forme normale fait appel au 1^{er} théorème de Shannon. Toute fonction logique peut être développée par rapport à l'une des variables sous forme d'une somme de 3 produits logiques telle que

$$F(A, B, \dots, Z) = A \cdot F(1, B, \dots, Z) + \bar{A} \cdot F(0, B, \dots, Z)$$

Exemple :

$$F(A, B) = A \cdot F(1, B) + \bar{A} \cdot F(0, B)$$

$$F(A, B) = A \cdot (B \cdot F(1, 1) + \bar{B} \cdot F(1, 0)) + \bar{A} \cdot (B \cdot F(0, 1) + \bar{B} \cdot F(0, 0))$$

$$F(A, B) = AB \cdot F(1, 1) + A\bar{B} \cdot F(1, 0) + \bar{A}B \cdot F(0, 1) + \bar{A}\bar{B} \cdot F(0, 0)$$

Pour chaque couple (i, j) le terme $F(i, j)$ représente la valeur particulière $F(A, B)$ dans le problème à résoudre

Exemple :

$$\begin{aligned} F(A, B, C) = & ABC \cdot F(1, 1, 1) + AB\bar{C} \cdot F(1, 1, 0) + A\bar{B}C \cdot F(1, 0, 1) + \\ & A\bar{B}\bar{C} \cdot F(1, 0, 0) + \bar{A}BC \cdot F(0, 1, 1) + \bar{A}B\bar{C} \cdot F(0, 1, 0) + \\ & \bar{A}\bar{B}C \cdot F(0, 0, 1) + \bar{A}\bar{B}\bar{C} \cdot F(0, 0, 0) \end{aligned}$$

D'une manière générale le développement comporte une réunion de 2nd terme chaque terme est le produit des variables et des valeurs particulières de la somme qui peut prendre les valeurs (**0 et 1**)

Exemple : Soit la fonction de 3 variables A, B et C définie par la table de vérité suivante

N°	A	B	C	$F(A, B, C)$	
0	0	0	0	0	$F(0, 0, 0)$
1	0	0	1	1	$F(0, 0, 1)$
2	0	1	0	1	$F(0, 1, 0)$
3	0	1	1	1	$F(0, 1, 1)$
4	1	0	0	0	$F(1, 0, 0)$
5	1	0	1	1	$F(1, 0, 1)$
6	1	1	0	0	$F(1, 1, 0)$
7	1	1	1	1	$F(1, 1, 1)$

Formule normale disjonctive de F : $A = 1$ et $\bar{A} = 0$

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

2. Forme normale conjonctive

Cette forme utilise le 2nd théorème d'expression de Shannon. Toute fonction logique peut être développée par rapport à l'une des variables sous forme d'un produit de 2 sommes logiques tel que

$$F(A, B, \dots, Z) = (\bar{A} + F(1, B, \dots, Z)) \cdot (A + F(0, B, \dots, Z))$$

Exemple :

$$F(A, B) = (\bar{A} + F(1, B)) \cdot (A + F(0, B))$$

$$F(A, B) = (\bar{A} + (\bar{B} + F(1, 1)) \cdot (B + F(1, 0))) \cdot (A + (\bar{B} + F(0, 1)) \cdot (B + F(0, 0)))$$

$$F(A, B) = (\bar{A} + \bar{B} + F(1, 1)) \cdot (\bar{A} + B + F(1, 0)) \cdot (A + \bar{B} + F(0, 1)) \cdot (A + B + F(0, 0))$$

Exemple : Soit la fonction de 3 variables A, B et C définie par la table de vérité suivante

N°	A	B	C	F(A, B, C)	
0	0	0	0	0	F(0, 0, 0)
1	0	0	1	1	F(0, 0, 1)
2	0	1	0	1	F(0, 1, 0)
3	0	1	1	1	F(0, 1, 1)
4	1	0	0	0	F(1, 0, 0)
5	1	0	1	1	F(1, 0, 1)
6	1	1	0	0	F(1, 1, 0)
7	1	1	1	1	F(1, 1, 1)

Formule normale conjonctive de F : $A = 0$ et $\bar{A} = 1$

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

3. Passage d'une forme normale à une autre

Pour le passage d'une forme normale à une autre, on a une complémentation successive indispensable.

- La première est effectuée manuellement,
- La seconde en effectuant algébriquement le théorème de Morgan

Exemple : Soit la fonction de 3 variables A, B et C définie par la table de vérité suivante

N°	A	B	C	F(A, B, C)	$\bar{F}(A, B, C)$
0	0	0	0	0	1
1	0	0	1	1	0
2	0	1	0	1	0
3	0	1	1	1	0
4	1	0	0	0	1
5	1	0	1	1	0
6	1	1	0	0	1
7	1	1	1	1	0

Formule normale disjonctive de \bar{F} :

$$\bar{F}(A, B, C) = \bar{A}\bar{B}\bar{C} + A\bar{B}\bar{C} + A\bar{B}\bar{C} \text{ En recomplémentant, on a :}$$

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

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

4. Les formes numériques

Pour condenser l'écriture, il est possible de repérer chaque terme par un numéro qui peut être l'équivalent décimal du nombre binaire représenté

Exemple :

N°	A	B	C	F (A, B, C)
0	0	0	0	0
1	0	0	1	1
2	0	1	0	1
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	0
7	1	1	1	1

- Forme numérique disjonctive

$$F(A, B, C) = R(1, 2, 3, 5, 7)$$

R → Réunion ensemble

- Forme numérique conjonctive

$$F(A, B, C) = I(0, 4, 6)$$

I → Intersection ensemble

IV. Simplification des fonctions logiques

Afin de fabriquer un système à moindre coup, il faut simplifier les fonctions logiques à réaliser. Pour cela nous utilisons les méthodes de simplification suivantes :

- La méthode algébrique
- Le tableau de Karnaugh

Ces méthodes permettent d'obtenir une forme minimale d'une fonction logique. On appelle forme minimale d'une expression logique l'expression sous forme réduite qui comporte un nombre minimal de terme et le nombre minimal de variables de chaque terme. Cette méthode consiste à appliquer les principes de l'**algèbre de Boole**. Les relations sont souvent utilisées.

1. La méthode algébrique

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

$$A + AB = A$$

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

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

$$A(A + B) = A$$

Règles de simplification :

- Règle 1 :

On peut simplifier une fonction logique en regroupant les termes identiques à l'aide des identités ci-dessous

Exemple :

$$F(A, B, C, D) = ABC + AB\bar{C} + A\bar{B}CD$$

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

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

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

$$F(A, B, C, D) = A(B + CD)$$

- Règle 2 :

On peut ajouter un terme déjà existant à une expression logique :

Exemple :

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

$$F(A, B, C, D) = ABC + \bar{A}BC + ABC + A\bar{B}C + ABC + AB\bar{C}$$

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

- **Règle 3 :**

On peut supprimer un terme superflu c'est-à-dire déjà inclus dans la réunion des autres termes

Exemple :

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

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

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

- **Règle 4 :**

Il est apriori préférable de simplifier la forme canonique ayant le nombre minimal de terme

Exemple :

$$F(A, B, C) = R(2, 3, 4, 5, 6, 7) \text{ avec le poids de } A = 4 \text{ } B = 2 \text{ } C = 1$$

$$F(A, B, C, D) = \bar{A}\bar{B}\bar{C} + \bar{A}BC + A\bar{B}\bar{C} + A\bar{B}C + A\bar{C} + ABC$$

$$F(A, B, C, D) = R(0, 1)$$

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

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

2. Méthode graphique Tableau de Karnaugh

Équivalence en décimal

- Tableau de Karnaugh à deux variables

A \ B	0	1
0	0	1
1	2	3

- Tableau de Karnaugh à trois variables

A \ BC	00	01	11	10
0	0	1	3	2
1	4	5	7	6

- Tableau de Karnaugh à quatre variables

AB \ CD	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

- Tableau de Karnaugh cinq variables

AB \ CDE	000	001	011	010	110	111	101	100
00								
01								
11								
10								

- Représentation d'une fonction dans un tableau de Karnaugh

- A partir d'une forme canonique disjonctive

Exemple :

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

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

- A partir de la première forme numérique

Exemple :

$$F(A, B, C) = R(1, 3, 7)$$

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

- A partir de la seconde forme numérique

Exemple :

$$F(A, B, C) = I(0, 2, 4, 5, 6)$$

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

- A partir d'une table de vérité :

Exemple :

N°	A	B	C	F (A, B, C)
0	0	0	0	0
1	0	0	1	1
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	0
6	1	1	0	0
7	1	1	1	1

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

3. Règles de simplification

- Procédure

On regroupe les cases comportant un même état logique (le "1") en recherchant le maximum d'axes de symétrie, en respectant les critères d'enroulement

- Règle 1 :

On ne peut regrouper qu'un nombre de cases correspondants à une puissance de 2 exacte

- Règle 2 :

Le regroupement de 2^n cases doit être en ligne, en colonne, en carré, en rectangle dans un tableau

- Règle 3 :

Il faut utiliser tous les "1" au moins une fois dans les regroupements le résultat est donné par la réunion des différents regroupements

- Règle 4 :

Il faut rechercher les groupements les plus grands, il en résultat un nombre minimal de variables

- Règle 5 : Détermination du terme produit associé à un groupement donné :

Une variable (respectivement son complément) appartient au produit si tous les "1" du groupement appartiennent à la moitié du tableau pour lequel cette variables (respectivement son complément), prend la valeur de "1"

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

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

4. Cas des fonctions incomplètement définis

Si pratiquement certaines combinaisons de variables ne peuvent jamais exister, il est possible d'utiliser ces combinaisons pour simplifier la fonction. Ces combinaisons sont appelées des conditions disponibles à la simplification et sont notés ϕ dans le tableau de Karnaugh.

Exemple :

Soit un chiffre N du système décimal traduit en binaire à l'aide de variable A, B, C et D. La sortie S du système à étudier prend la valeur 1 si N est inférieur ou égal à 5. Déterminer S sous la forme canonique disjonctive

Nº	A	B	C	D	F (A, B, C , D)
0	0	0	0	0	1
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	1
4	0	1	0	0	1
5	0	1	0	1	1
6	0	1	1	0	0
7	0	1	1	1	0
8	1	0	0	0	0
9	1	0	0	1	0
10	1	0	1	0	0
11	1	0	1	1	0
12	1	1	0	0	0
13	1	1	0	1	0
14	1	1	1	0	0
15	1	1	1	1	0

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

$$S = \bar{A}\bar{C} + \bar{A}\bar{B}$$