République Tunisienne
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Institut Supérieur d'Informatique et des Mathématiques de Monastir
Université de Monastir

Cours:
Systèmes Logiques et Architecture des Ordinateurs

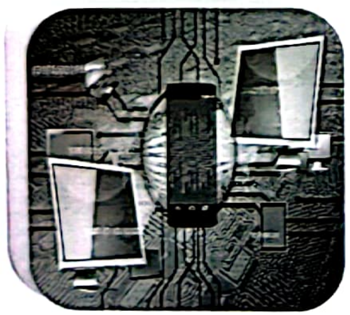
Dr. Safa Teboulbi

Année universitaire : 2024-2025

Chapitre 1

Systèmes de Numération et Codage Des Informations

Systèmes de Numération



- Les systèmes numériques sont des systèmes qui utilisent des nombres et des symboles pour représenter et manipuler des informations.

- Les systèmes numériques sont utilisés dans de nombreux domaines, de l'informatique à l'électronique en passant par les télécommunications.

Systèmes de Numération

- Pour qu'une information numérique soit traitée par un circuit, elle doit être mise sous forme adaptée à celui-ci.
  Pour cela, il faut choisir un système de numération de base B (B un nombre entier naturel >= 2).

- Les systèmes de numération les plus utilisés en technologie numérique sont les systèmes:
  - Décimal (base 10)
  - Binaire (base 2)
  - Octal (base 8)
  - Hexadécimal (base 16)

- Les systèmes numériques sont fondamentaux pour la conception et le fonctionnement des ordinateurs.

Systèmes de Numération

Représentation d'un nombre

Dans un système de numération en base B, un nombre noté $N_{(B)}$ égal à :

$$N_{(B)} = \sum_{k=0}^{n-1} a_k \times B^k = a_{n-1} \times B^{n-1} + a_{n-2} \times B^{n-2} + \dots\dots\dots\dots + a_1 \times B^1 + a_0 \times B^0{}_{(B)}$$

B Base du système de numération, elle représente le nombre des différents chiffres qu'utilise ce système de numération

K Rang du chiffre a_k

a_k Un chiffre (ou digit) parmi les chiffres de la base du système de numération du rang k

B^k Pondération associée à a_k

Systèmes de Numération

Système Décimal (Base 10)

- Le système décimal comprend 10 chiffres qui sont {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}.
- C'est un système qui s'est imposé tout naturellement à l'homme qui possède 10 doigts.
  - Un nombre décimal peut s'écrire sous la forme polynomiale.

<u>*Exemples*</u>

$(7239)_{10} = 7 \times 10^3 + 2 \times 10^2 + 3 \times 10^1 + 9 \times 10^0$

$(546)_{10} = 5 \times 10^2 + 4 \times 10^1 + 6 \times 10^0$

$(239{,}537)_{10} = 2 \times 10^2 + 3 \times 10^1 + 9 \times 10^0 + 5 \times 10^{-1} + 3 \times 10^{-2} + 7 \times 10^{-3}$

Systèmes de Numération

Système Binaire (Base 2)

- Dans ce système de numération il n'y a que deux chiffres possibles (0, 1) qui sont souvent appelés bits « <u>*binary digit*</u> ».
  - Un nombre binaire peut s'écrire sous la forme polynomiale.

<u>*Exemples*</u>

$(111011)_2 = 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$

$(11110010)_2 = 1 \times 2^7 + 1 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$

$(10011{,}1101)_2 = 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} + 0 \times 2^{-3} + 1 \times 2^{-4}$

Systèmes de Numération

Système Octal (Base 8)

- Le système octal ou base 8 comprend huit chiffres qui sont {0, 1, 2, 3, 4, 5, 6, 7}. Les chiffres 8 et 9 n'existent pas dans cette base.
  - Un nombre octal peut s'écrire sous la forme polynomiale.

<u>*Exemples*</u>

$(4527)_8 = 4 \times 8^3 + 5 \times 8^2 + 2 \times 8^1 + 7 \times 8^0$

$(562)_8 = 5 \times 8^2 + 6 \times 8^1 + 2 \times 8^0$

$(1274{,}632)_8 = 1 \times 8^3 + 2 \times 8^2 + 7 \times 8^1 + 4 \times 8^0 + 6 \times 8^{-1} + 3 \times 8^{-2} + 2 \times 8^{-3}$

Systèmes de Numération

Système Hexadécimal (Base 16)

- Le système Hexadécimal ou base 16 contient seize éléments qui sont {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F}.
- Les chiffres A, B, C, D, E, et F représentent respectivement 10, 11, 12, 13, 14 et 15.
  - Un nombre hexadécimal peut s'écrire sous la forme polynomiale.

Exemples

$(3256)_{16} = 3 \times 16^3 + 2 \times 16^2 + 5 \times 16^1 + 6 \times 16^0$

$(9C4F)_{16} = 9 \times 16^3 + 12 \times 16^2 + 4 \times 16^1 + 15 \times 16^0$

$(A2B,E1)_{16} = 10 \times 16^2 + 2 \times 16^1 + 11 \times 16^0 + 14 \times 16^{-1} + 1 \times 16^{-2}$

Changement de Base

- Il s'agit de la conversion d'un nombre écrit dans une base B_1 à son équivalent dans une autre base B_2

Conversion d'un nombre N de base B en un nombre décimal

- La valeur décimale d'un nombre N, écrit dans une base B, s'obtient par sa forme polynomiale décrite précédemment.

Exemples

$(1011101)_2 = 1 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = (93)_{10}$

$(7452)_8 = 7 \times 8^3 + 4 \times 8^2 + 5 \times 8^1 + 2 \times 8^0 = (3882)_{10}$

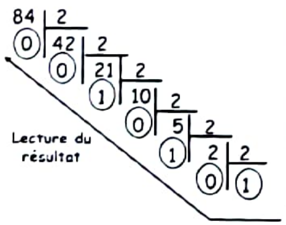
$(D7A)_{16} = 13 \times 16^2 + 7 \times 16^1 + 10 \times 16^0 = (3450)_{10}$

Conversion d'un nombre décimal entier

- Pour convertir un nombre décimal entier en un nombre de base B quelconque, il faut faire des divisions entières successives par la base B et conserver à chaque fois le reste de la division.
- On s'arrête lorsqu'on obtient un résultat *inférieur* à la base B.

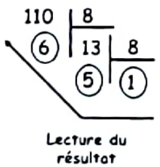
Exemples

$(84)_{10} = (?)_2$



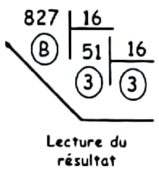
$(3256)_{10} = (1010100)_2$

$(110)_{10} = (?)_8$



$(110)_{10} = (156)_8$

$(827)_{10} = (?)_{16}$



$(827)_{10} = (33B)_{16}$

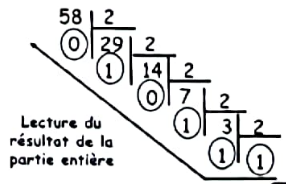
Conversion d'un nombre décimal à virgule

- Pour convertir un nombre décimal à virgule dans une base B quelconque, il faut :
- Convertir la partie entière en effectuant des divisions successives par B (comme nous l'avons vu précédemment).
- Convertir la partie fractionnaire en effectuent des multiplications successives par B et en conservant à chaque fois le chiffre devenant entier.

Exemple

$(58{,}625)_{10} = (?)_2$

Conversion de la partie entière



Conversion de la partie fractionnaire

$0.625 \times 2 = \boxed{1}.25$

$0.25 \times 2 = \boxed{0}.5$

$0.5 \times 2 = \boxed{1}.0$

Lecture du résultat de la partie fractionnaire

$(58{,}625)_{10} = (111010{,}101)_2$

Remarques

- Parfois en multipliant la partie fractionnaire par la base B on n'arrive pas à convertir toute la partie fractionnaire. Ceci est dû essentiellement au fait que le nombre à convertir n'a pas un équivalent exacte dans la base B et sa partie fractionnaire est cyclique.

Exemples

$(0,15)_{10} = (?)_2$

0,15 × 2 = 0 ,3
0,3 × 2 = 0 ,6
0,6 × 2 = 1 ,2
0,2 × 2 = 0 ,4
0,4 × 2 = 0 ,8
0,8 × 2 = 1 ,6
0,6 × 2 = 1 ,2
0,2 × 2 = 0 ,4
0,4 × 2 = 0 ,8
0,8 × 2 = 1 ,6

Le nombre $(0,15)_{10}$ est cyclique dans la base *2* de période ***1001***.

$(0,15)_{10} = (0,0010011001)_2 = (0,00\underline{1001})_2$

Autres Conversions

- Pour faire la conversion d'un nombre d'une base quelconque B_1 vers une autre base, il faut passer par la base 10.
- Mais si la base B_1 et B_2 s'écrivent respectivement sous la forme d'une puissance de 2 on peut passer par la base 2 (binaire) :
  - Base octale (base 8) : $8 = 2^3$: Chaque chiffre octal se convertit tout seul sur 3 bits.
  - Base hexadécimale (base 16) : $16 = 2^4$: Chaque chiffre hexadécimal se convertit tout seul sur 4 bits.

Conversion d'un nombre binaire vers l'Octal

$(101010)_2 = (101\ 010)_2 = (52)_8$

Conversion d'un nombre Octal vers le binaire

$(6530)_8 = (110\ 101\ 011\ 000)_2$

Conversion d'un nombre binaire vers l'Hexadécimal

$(110101110001)_2 = (1101\ 0111\ 0001)_2 = (D71)_{16}$

Conversion d'un nombre Hexadécimal vers le binaire

$(9\ A\ 2\ C)_{16} = (1001\ 1010\ 0010\ 1100)_2$

- Pour la conversion Octal-Hexadécimal et Hexadécimal-Octal, la plus simple méthode est de passer par *le système Binaire*.

Conversion d'un nombre Octal vers l'Hexadécimal

$(51024)_8 = (101001000010100)_2 = (5214)_{16}$

Conversion d'un nombre Hexadécimal vers l'Octal

$(92BD5A)_{16} = (100100101011110101011010)_2 = (44536532)_8$

Représentation des nombres signés

- La représentation des nombres signés est une méthode utilisée en informatique pour représenter à la fois des nombres positifs et négatifs dans un système numérique.
- Il existe plusieurs méthodes pour représenter des nombres signés.
- Les méthodes les plus couramment utilisées sont:

La représentation en valeur absolue et signe

- *Cest* une méthode de représentation des nombres signés qui utilise un bit pour indiquer le signe (0 pour positif, 1 pour négatif) et les bits restants pour représenter la valeur absolue du nombre.

Exemple

La représentation en valeur absolue et signe d'un nombre -5 sur 4 bits :

- Le signe est négatif, donc le bit de signe est 1.
- La valeur absolue du nombre est 5, qui s'écrit en binaire comme 101.

⟶ La représentation en valeur absolue et signe de -5 sur 4 bits est : 1101.

La représentation en complément à 1

- Est une méthode de représentation des nombres signés.
- Dans cette méthode, les nombres négatifs sont obtenus en inversant tous les bits (en changeant les 0 en 1 et vice versa) du nombre positif correspondant.

<u>Exemple</u>

La représentation en complément à un sur 4 bits du nombre -5 :

- -5 en binaire : 0101
- Complément à un : 1010

La représentation en complément à 2

- Est la méthode la plus couramment utilisée pour représenter des nombres signés en binaire.
- Pour représenter un nombre positif, il suffit de le convertir en binaire normalement.
- Pour représenter un nombre négatif, on effectue les étapes suivantes :
  - Prendre le complément à un du nombre binaire (inversion des bits : 0 devient 1 et vice versa).
  - Ajouter 1 au résultat obtenu à l'étape précédente.

<u>Exemple</u>

Pour représenter -5 en complément à deux sur 4 bits :

- 5 en binaire : 0101
- Complément à un : 1010
- Ajouter 1 : 1011 (représentation de -5 en complément à deux sur 4 bits)

Les opérations dans les bases

L'Addition

0 + 0 = 0

0 + 1 = 1

1 + 0 = 1

1 + 1 = 10

Effectuer une retenue comme dans le cas d'une addition décimale

<u>Exemples</u>

$(101101)_2 + (10010)_2 = (111111)_2$

$(11011)_2 + (1101)_2 = (101000)_2$

$(243)_8 + (112)_8 = (355)_8$

$(567)_8 + (432)_8 = (1221)_8$

$(12A)_{16} + (3B2)_{16} = (4DC)_{16}$

La Soustraction

0 - 0 = 0

0 - 1 = 1 avec un retenue de <u>1</u>

1 - 0 = 1

1 - 1 = 0

<u>Exemples</u>

$(1110110)_2 - (110101)_2 = (1000001)_2$

$(1010)_2 - (0111)_2 = (0011)_2$

$(213)_8 - (167)_8 = (024)_8$

$(A37)_{16} - (216)_{16} = (821)_{16}$

$(A210)_{16} - (2BCF)_{16} = (7641)_{16}$

La Multiplication

0 × 0 = 0

0 × 1 = 0

1 × 0 = 0

1 × 1 = 1

<u>Exemples</u>

$(1011)_2 \times (1101)_2 = (10001111)_2$

$(237)_8 \times (63)_8 = (17655)_8$

$(A928)_{16} \times (7D3)_{16} = (52B83F8)_{16}$

La division

- La division en Binaire, Octal et Hexadecimal suit essentiellement les <u>mêmes</u> principes que la division en Décimal (Base 10).
- Cependant, en fonction de la base dans laquelle vous effectuez la division, le processus peut <u>varier</u> légèrement.

Exemples

$$(10110)_2 \div (11)_2 = (0111)_2$$

$$(651)_8 \div (3)_8 = (215)_8$$

$$(24328)_{16} \div (2B)_{16} = (D78)_{16}$$

Codage de l'Information

On appelle codage, l'opération qui consiste à faire correspondre à tout caractère (lettre, chiffre, signe, ..) un symbole où un ensemble de symboles particuliers appelés mot de code

Le codage de l'information est nécessaire pour le traitement automatique de celui-ci

Les codes binaires purs

- C'est une représentation numérique des nombres dans la base 2.

Exemples

$(0)_{10} = (0000)_2$

$(1)_{10} = (0001)_2$

$(2)_{10} = (0010)_2$

$(3)_{10} = (0011)_2$

$(4)_{10} = (0100)_2$

Ce code présente l'inconvénient de changer <u>plus qu'un seul bit</u> quand on passe d'un nombre à un autre immédiatement supérieur.

Le Code Binaire Réfléchi ou Le Code GRAY

- Dans le cas des capteurs - codeurs (de position par exemple), où l'utilisation du code binaire pur peut amener la lecture de codes erronés au moment des transitions d'une position à la suivante.

- On utilise le code GRAY qui est un code binaire réfléchi dans lequel un seul bit change d'état au changement d'un nombre au nombre suivant.

Code en Décimal	Code en Binaire Gray	Code en Décimal	Code en Binaire Gray
0	0000	5	0111
1	0001	6	0101
2	0011	7	0100
3	0010	8	1100
4	0110	9	1101

Conversion du Binaire Naturel <u>Vers</u> Binaire Réfléchi

Il s'agit de comparer les bits b_{n+1} et le bit b_n du <u>binaire naturel</u>.
Le résultat est b_r du binaire réfléchi
$b_r = 0$ si $b_{n-1} = b_n$
$= 1$ sinon
Le premier bit à gauche reste inchangé.

Exemple

$$\begin{array}{l}(6)_{BN} = 1 \leftrightarrow 1 \leftrightarrow 0 \\ \quad\quad\quad\;\; \downarrow \quad \downarrow \quad \downarrow \\ (6)_{BR} = 1 \quad 0 \quad 1\end{array}$$

$$\boxed{(6)_{10} = (110)_{BN} = (101)_{BR}}$$

Conversion du Binaire Réfléchi <u>Vers</u> Binaire Naturel

Il s'agit de comparer le bit b_{n-1} du <u>binaire naturel</u> et le bit b_n du <u>binaire réfléchi</u>.
Le résultat est b_n du binaire naturel
$b_n = 0$ si $b_{n-1} = b_n$
$= 1$ sinon.
Le premier bit à gauche reste inchangé.

Exemple

$$\begin{array}{l}(10)_{BR} = 1 \quad 1 \quad 1 \quad 1 \\ (10)_{BN} = 1 \quad 0 \quad 1 \quad 0\end{array}$$

$$\boxed{(10)_{10} = (1111)_{BR} = (1010)_{BN}}$$

Le Code DCB (Décimal Codé Binaire)

- Les <u>systèmes digitaux</u> utilisent tous des <u>nombres binaires</u> pour leurs <u>opérations internes</u>.
- La numération décimale étant la plus couramment utilisée par les humains, cela suppose une <u>conversion</u> entre la <u>numérotation en décimal</u> et <u>la numérotation en binaire</u>.

- Le code DCB, en Anglais BCD (Binary Coded Decimal), consiste à représenter <u>chaque chiffre</u> d'un <u>nombre décimal</u> par son <u>équivalent binaire</u> <u>sur 4 bits</u>.

Exemples

$(7239)_{10} = (0111\ 0010\ 0011\ 1001)_{BCD}$

$(1001\ 0011)_{BCD} = (93)_{10}$

<u>Remarque</u>

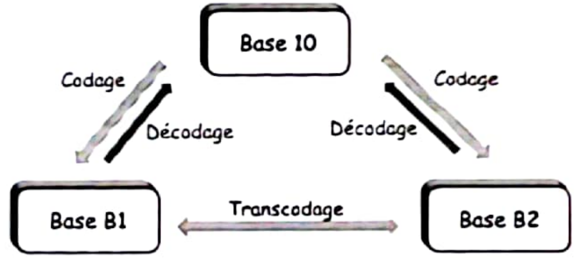
- Le code binaire pur prend le nombre décimal complet et le représente en binaire.
- Le code BCD convertit chaque digit décimal individuellement en binaire.

Exemple

$(137)_{10} = (10001001)_2 = (0001\ 0011\ 0111)_{BCD}$

Le Transcodage

- Une des applications liée au codage des informations est le passage d'un code à un autre.
- Cette opération est appelée <u>transcodage</u>.



- Le codage des informations se fait au moyen d'un circuit combinatoire appelé <u>Codeur</u>.
- Le décodage des informations se fait au moyen d'un circuit combinatoire appelé <u>Décodeur</u>.
- Un transcodeur est un <u>Décodeur</u> associé à un <u>Codeur</u>.