Chapitre 2 LES SYSTEMES DE NUMERATION

M^{me} Zeghdoudi Salima

M^{elle} Bourai Nour

I Notion de BIT (Binary digit)

Dans une mémoire, les informations sont représentées sous forme numérique : grandeur physique avec 2 états 0 et 1. Chaque élément de mémoire ne peut être que 0 ou 1. Il s'appelle le **BIT** C'est la plus petite unité d'information que l'on peut stocker dans une mémoire

II L'OCTET

Un OCTET ou un BYTE (chez les anglo-saxons) est une combinaison de 8 bits
Chaque caractère occupe un octet à l'intérieur (dans la mémoire) de la machine
L'ordinateur reconnaît 256 caractères (soit 2^8)

Il existe des multiples de l'octet

Kilo Octet (KO) = 2^{10} Octets = 1024 Octets

Méga Octet (MO) = 2^{10} KO = 2^{20} Octets

Giga Octets (GO) = 2^{10} MO = 2^{20} KO = 2^{30} Octets

Tera Octet, Petra octet, hexa octet,

III LES SYSTEMES DE NUMERATION

A l'intérieur de l'ordinateur les informations sont représentées sous une forme compréhensible par la machine. En effet, l'ordinateur ne reconnaît que 0 et 1 (appelé le BINAIRE).

Le **BINAIRE** constitue un système de numération dont la base est **2**

On s'intéressera également au système **OCTAL** dont la base est **8** et au système **HEXADECIMAL** dont la base est **16**

La base est le nombre de symboles utilisés dans le système

IV LES CONVERSIONS

Le processus de conversion est un ensemble de procédures permettant de convertir un nombre représenté dans un système de numération vers son équivalent dans un autre système. Il existe différents algorithmes, nous nous intéresserons aux plus accessibles

IV.1 CONVERSION DU SYSTEME DECIMAL AU SYSTEME BINAIRE

IV.1.1 Cas des nombres entiers

- **Algorithme Méthode Modulo B (B= 2)**

On divise successivement par 2 le nombre décimal, jusqu'à l'obtention d'un quotient =0. Les restes obtenus (qui ne peuvent être que 0 ou 1) représentent le résultat soit le nombre en binaire

Exemple : $(83)_{10} = (?)_2$

$$\begin{array}{r} 83 \mid 2 \\ 1 \mid 41 \mid 2 \\ 1 \mid 20 \mid 2 \\ 0 \mid 10 \mid 2 \\ 0 \mid 5 \mid 2 \\ 1 \mid 2 \mid 2 \\ 0 \mid 1 \mid 2 \\ 1 \mid 0 \end{array} \leftarrow \text{condition d'arrêt}$$

$(83)_{10} = (1010011)_2$

IV.1.2 Cas des nombres fractionnaires

- Conversion de la partie entière : On applique la méthode modulo B vue précédemment pour la conversion de la partie entière du nombre
- Conversion de la partie fractionnaire : On procède comme suit ;
 - On multiplie par 2 et on conserve la retenue
 - On répète la multiplication jusqu'à obtention d'un résultat 0

Exemple : $(83.625)_{10} = (?)_2$

- Partie entière : $(83)_{10} = (1010011)_2$ D'après l'exemple précédent
- Partie fractionnaire :

$$\begin{array}{r} 625 \\ \times 2 \\ \hline 1 \quad 250 \\ \times 2 \\ \hline 0 \quad 500 \\ \times 2 \\ \hline 1 \quad 000 \leftarrow \text{Condition d'arrêt} \end{array}$$

Le résultat de la partie fractionnaire est 101

$$(83.625)_{10} = (1010011.101)_2$$

IV.2 CONVERSION DU SYSTEME BINAIRE AU SYSTEME DECIMAL

- **Algorithme Méthode de la somme des produits**

IV.2.1 Cas des nombres entiers

- **Etape 1 :** On part du bit le plus à droite, on le multiplie par 2^0
- **Etape 2 :** On ajoute la valeur du bit suivant multipliée par 2^1
- **Etape 3 :** On ajoute la valeur du bit suivant multipliée par 2^2
- **Etape 4 :** On répète la somme jusqu'au bit (n) multipliée par 2^{n-1}

Exemple : $(100101)_2 = (?)_{10}$

$$1*2^0 + 0*2^1 + 1*2^2 + 0*2^3 + 0*2^4 + 1*2^5 = 1+0+4+0+0+32 = 37$$

$$(100101)_2 = (37)_{10}$$

IV.2.2 Cas des nombres fractionnaires

On applique la méthode de la somme des produits en considérant que les chiffres de la partie fractionnaire portent des puissances négatives décroissantes vers la droite

Exemple : $(100101.101)_2 = (?)_{10}$

$$1*2^0 + 0*2^1 + 1*2^2 + 0*2^3 + 0*2^4 + 1*2^5 + 1*2^{-1} + 0*2^{-2} + 1*2^{-3} =$$
$$37 \qquad \qquad \qquad + 0.5 \quad + \quad 0 \quad + \quad 0.125 \quad = 37.625$$

$$(100101.101)_2 = (37.625)_{10}$$

IV.3 CONVERSION DU SYSTEME DECIMAL AU SYSTEME OCTAL

- **SYSTEME OCTAL** : Le système octal utilise 8 symboles 0,1,2,3,4,5,6,7

La base est 8 . Chaque chiffre est affecté d'une puissance de 8 .

- **Algorithme Méthode Modulo B (B= 8)**

On divise successivement par 8 le nombre décimal , jusqu'à l'obtention d'un quotient =0 . Les restes obtenus (qui ne peuvent être que 0,1,2,3,4,5,6,7) représentent le résultat soit le nombre octal

Exemple : $(247)_{10} = (?)_8$

$$\begin{array}{r} 247 \overline{) 8} \\ \underline{7 } \\ 30 \overline{) 8} \\ \underline{6 } \\ 3 \overline{) 8} \\ \underline{3 } \\ 0 \end{array} \quad \leftarrow \text{condition d'arrêt}$$

$$(247)_{10} = (367)_8$$

IV.4 CONVERSION DU SYSTEME OCTAL AU SYSTEME DECIMAL

On applique la somme des produits

Exemple : $(207)_8 = (?)_{10}$

$$7*8^0 + 0*8^1 + 2*8^2 = 7 + 0 + 128 = 135$$

$$(207)_8 = (135)_{10}$$

IV.5 CONVERSION DU SYSTEME DECIMAL AU SYSTEME HEXADECIMAL

- **SYSTEME HEXADECIMAL** : Le système Hexadécimal utilise 16 symboles

0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

La base =16 . Chaque chiffre est affecté d'une puissance de 16 .

Les lettres A,B,C,D,E,F représentent respectivement les valeurs 10,11,12,13,14,15 du système décimal .

- **Algorithme Méthode Modulo B (B= 16)**

On divise successivement par 16 le nombre décimal , jusqu'à l'obtention d'un quotient =0 . Les restes obtenus (qui ne peuvent être que 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15) représentent le résultat soit le nombre en hexadécimal . Les restes **10,11,12,13,14,15** devront être convertis en équivalents hexadécimal **A,B,C,D,E,F**

Exemple : $(142)_{10} = (?)_{16}$

$$\begin{array}{r} 142 \overline{) 16} \\ \underline{14 } \\ 8 \overline{) 16} \\ \underline{8 } \\ 0 \end{array} \quad \leftarrow \text{condition d'arrêt}$$

$$(142)_{10} = (8 E)_{16}$$

IV.6 CONVERSION DU SYSTEME HEXADECIMAL AU SYSTEME DECIMAL

On applique la somme des produits

Exemple : $(3A4)_{16} = (?)_{10}$

$$4*16^0 + 10*16^1 + 3*16^2 = 4 + 160 + 768 = 932$$

$$(3A4)_{16} = (932)_{10}$$

IV.7 CONVERSION DU SYSTEME BINAIRE AU SYSTEME OCTAL

On applique l'algorithme suivant :

- Etape1 : On part du chiffre le plus à droite
- Etape2 : On sépare les bits par groupe de 3 car $8 = 2^3$, on complète si besoin le dernier groupe par un ou deux 0 à gauche
- Etape3 : On remplace chaque groupe par son équivalent octal

Nombre octal	Equivalent binaire
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Table de correspondance Octal/Binaire

Exemple : $(10111011101)_2 = (?)_8$

010 111 011 101
2 7 3 5

$(10111011101)_2 = (2735)_8$

IV.8 CONVERSION DU SYSTEME OCTAL AU SYSTEME BINAIRE

Chaque chiffre octal est remplacé par son équivalent binaire de 3 bits

Exemple : $(643710)_8 = (?)_2$

6 4 3 7 1 0
110 100 011 111 001 000

$(643710)_8 = (110100011111001000)_2$

IV.9 CONVERSION DU SYSTEME BINAIRE AU SYSTEME HEXADECIMAL

On applique l'algorithme suivant :

- Etape1 : On part du chiffre le plus à droite
- Etape2 : On sépare les bits par groupe de 4 car $16 = 2^4$, on complète si besoin le dernier groupe par un , deux ou trois 0 à gauche
- Etape3 : On remplace chaque groupe par son équivalent hexadécimal

Nombre 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

Table de correspondance Hexadécimal/Binaire

Exemple : $(10111011101)_2 = (?)_{16}$

0101 1101 1101

7 D D

$(10111011101)_2 = (7\ D\ D)_{16}$

IV.10 CONVERSION DU SYSTEME HEXADECIMAL AU SYSTEME BINAIRE

Chaque chiffre hexadécimal est remplacé par son équivalent binaire de 4 bits

Exemple : $(1F7A)_{16} = (?)_2$

1 F 7 A
0001 1111 0111 1010

$(1F7A)_{16} = (0001111101111010)_2$