

## Chapitre 5 : Représentation de l'information

### 1. Système de numération

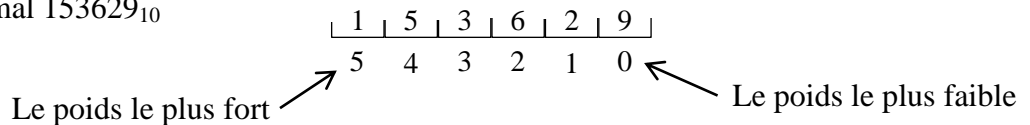
#### 1.1. Introduction

Un système de numération se caractérise par une base et un nombre de symboles qu'il utilise pour exprimer des grandeurs. **Exemple** : dans le système à base 10 ou le système décimal on utilise dix chiffres : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

#### 1.2. Poids d'un symbole

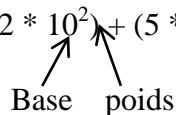
Dans un système de numération l'importance d'un chiffre varie en fonction de sa position (rang) dans le nombre à partir de la droite et en commençant par 0, cette position s'appelle Poids. Le chiffre le plus à droite possède le poids le plus faible et celui le plus à gauche possède le poids le plus fort.

**Exemple** : le nombre décimal  $153629_{10}$



#### 1.3. Le système décimal

C'est le système le plus connu. Dans ce système les symboles de représentation sont de 0 à 9 (10 symboles) et la base est 10. **Exemple** : Soit le nombre 256 exprimé dans le système décimal. Ce nombre peut être exprimé aussi comme suit :  $200 + 50 + 6$  Ou bien encore :  $(2 * 10^2) + (5 * 10^1) + (6 * 10^0)$



Cette représentation nous permet de dire que le chiffre 2, par exemple, a un poids plus important que le chiffre 5. En effet, le chiffre 2 représente les centaines alors que le chiffre 5 représente les dizaines.

#### Règle générale :

Pour tout nombre exprimé en une base B, la valeur d'un chiffre X de poids n, dans ce nombre, est égale à :  $X * B^n$ .

#### 1.4. Le système binaire

Les systèmes informatiques, étant construits à l'aide de circuits intégrés (composants électroniques qui rassemblent sur une puce de silicium quelques millions de transistors), ne peuvent fonctionner que selon une logique à deux états telle que le courant passe ou ne passe pas dans le transistor. Ces deux états logiques, conventionnellement notés 0 et 1, déterminent une logique dite : binaire, qui correspond à deux niveaux électriques que nous donnerons, pour simplifier, comme étant équivalents à 0 volt et à + 5 volts.

Toute information à traiter doit être représentée sous une forme assimilable par la machine, c'est à dire sous forme binaire.

Le passage d'une information, d'un langage compréhensible par l'homme à un langage compréhensible par le système informatique, s'appelle codage ou numérisation. Il existe de nombreuses possibilités de codage d'information : BINAIRE, HEXADICEMAL, BCD, ASCII, ....

Le système binaire est le système à base 2. Il utilise donc deux symboles pour représenter un nombre quelconque. Ces deux symboles sont 0 et 1. Un nombre binaire se présente donc sous la forme d'une combinaison de 0 et 1. **Exemple** : Soit le nombre binaire : 1101. Pour calculer la valeur de ce nombre, nous procédons comme suit :  $1101_2 = (1 * 2^3) + (1 * 2^2) + (0 * 2^1) + (1 * 2^0) = 8 + 4 + 0 + 1 = 13_{10}$ .

### 1.5. Le système octal

Ce système est de base 8 ce qui veut dire qu'il utilise 8 symboles pour la représentation des nombres, ces symboles sont : 0, 1, 2, 3, 4, 5, 6, 7. **Exemple** : le chiffre  $14_{10} = 16_8$  car  $(1 * 8^1) + (6 * 8^0) = 14_{10}$

### 1.6. Le système hexadécimal

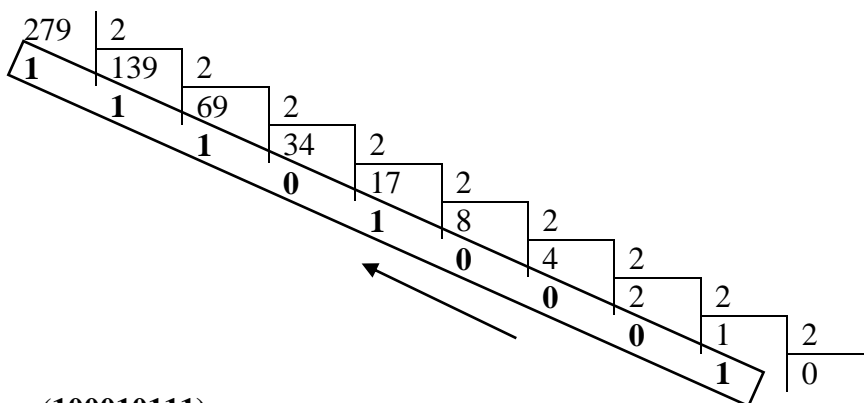
Ce système est de base 16 ce qui veut dire qu'il utilise 16 symboles pour la représentation des nombres, ces symboles sont : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F. **Exemple** : le chiffre  $14_{10} = E_{16}$ .

### 1.7. La conversion inter-bases

La conversion est l'opération qui permet de déterminer, pour tout nombre exprimé dans un système de numération de base quelconque, son équivalent dans une autre base.

#### 1.7.1. Du décimal au binaire

Pour convertir un nombre décimal en un nombre binaire il faut procéder par division successive par 2 du nombre décimal à convertir jusqu'à ce que le quotient soit nul. Le dernier de ces restes représente le chiffre binaire de rang le plus élevé. **Exemple** :  $279_{10} = (?)_2$ :



Donc  $(279)_{10} = (100010111)_2$

#### 1.7.2. Du binaire au décimal

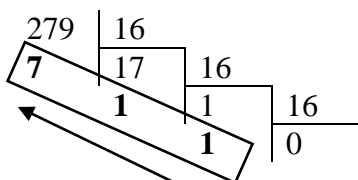
Pour retrouver l'équivalent décimal d'un nombre binaire, il suffit de calculer la somme des poids binaires de tous les chiffres 1 du nombre binaire. **Exemple** :  $(1011)_2 = (?)_{10}$

L'équivalent se calcule comme suit :  $(1 * 2^0) + (1 * 2^1) + (1 * 2^2) = 1 + 2 + 8 = 11$ . Donc :  $(1011)_2 = (11)_{10}$

#### 1.7.3. Du décimal à l'hexadécimal

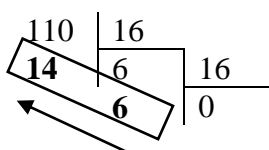
Pour déterminer l'équivalent hexadécimal d'un nombre décimal, on utilise la même méthode utilisée pour passer du décimal au binaire en procédant par division successive par 16 cette fois-ci.

**Exemple 1**:  $(279)_{10} = (?)_{16}$



D'où :  $(279)_{10} = (117)_{16}$

**Exemple 2** :  $(110)_{10} = (?)_{16}$  :



L'équivalent de 14 en hexadécimal est E d'où :  $(110)_{10} = (6E)_{16}$

#### 1.7.4. De l'hexadécimal au décimal

La méthode est la même que celle utilisée pour passer du binaire au décimal.

**Exemple 1 :**  $(117)_{16} = (?)_{10}$

$$(117)_{16} = (1 * 16^2) + (1 * 16^1) + (7 * 16^0) = 256 + 16 + 7 = 279 \text{ Donc : } (117)_{16} = (279)_{10}$$

**Exemple 2 :**  $(1DA)_{16} = (?)_{10}$

$$(1DA)_{16} = (1 * 16^2) + (D * 16^1) + (A * 16^0) = (1 * 256) + (13 * 16) + (10 * 1) = 474 \text{ Donc : } (1DA)_{16} = (474)_{10}$$

#### 1.7.5. Du binaire à l'hexadécimal

La méthode de conversion consiste à découper, de droite à gauche, le nombre binaire en tranches de quatre chiffres, puis faire la correspondance de chaque groupe en symbole hexadécimal.

**Exemple :**  $(00111101)_2 = (?)_{16}$

Le découpant ce nombre en groupes de quatre bits donne : « 0011 et 1101 » ce qui permet de faire la correspondance en hexadécimal :  $(0011)_2 = (3)_{16}$  et  $(1101)_2 = (D)_{16}$  D'où :  $(00111101)_2 = (3D)_{16}$

#### 1.7.6. De l'hexadécimal au binaire

Le passage de l'hexadécimal au binaire s'effectue par remplacement de chaque symbole de la représentation hexadécimale par sa valeur en binaire. **Exemple :**  $(5C3A)_{16} = (?)_2$

L'équivalent de ce nombre en binaire est établi comme suit :

5	C	3	A
0101	1100	0011	1010

$$\text{D'où : } (5C3A)_{16} = (0101110000111010)_2$$

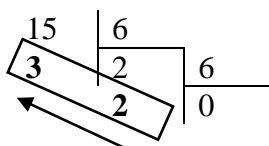
#### 1.7.7. Du décimal à un système de base N

Pour réaliser cette conversion, nous devons accomplir les étapes suivantes :

- 1- Diviser par N le nombre décimal à convertir, puis les quotients successifs obtenus, jusqu'à obtention d'un quotient nul.
- 2- Le nombre recherché sera constitué par le regroupement de tous les restes obtenus, le premier de ces restes occupe le rang 0 et le dernier reste se trouve au rang le plus élevé.

**Exemple :**

L'équivalent du nombre décimal 15 dans le système à base 6 :



$$\text{Donc : } (15)_{10} = (23)_6$$

#### 1.7.8. D'un système à base N au Décimal

Pour convertir un nombre quelconque, d'un système de base N vers le système décimal, il faut suivre les étapes suivantes :

- 1- Multiplier chacun des chiffres qui composent le nombre à convertir par la base (N) puissance le poids du chiffre.
- 2- Le nombre recherché vaut la somme des résultats obtenus.

**Exemple :**  $(1142)_6 = (?)_{10}$

$$(1142)_6 = (1 * 6^3) + (1 * 6^2) + (4 * 6^1) + (2 * 6^0) = 216 + 36 + 24 + 2 = 278 \text{ d'où } (1142)_6 = (278)_{10}$$

### 2. Représentation des nombres entiers

La représentation (ou codification) des nombres est nécessaire afin de les stocker et les manipuler par un ordinateur. Le principal problème est la limitation de la taille du codage : un nombre mathématique peut prendre des valeurs arbitrairement grandes, tandis que le codage dans l'ordinateur doit s'effectuer sur un nombre de bits fixé.

#### 2.1. Entiers naturels

Les entiers naturels (positifs ou nuls) sont codés sur un nombre d'octets fixé (un octet est un groupe de 8 bits). On rencontre habituellement des codages sur 1, 2 ou 4 octets, plus rarement sur 64 bits.

Un codage sur n bits permet de représenter tous les nombres naturels compris entre 0 et  $2^n - 1$ . Par exemple sur 1 octet, on pourra coder les nombres de 0 à  $255 = 2^8 - 1$ .

On représente le nombre en base 2 et on range les bits dans les cellules binaires correspondant à leur poids binaire, de la droite vers la gauche. Si nécessaire, on complète à gauche par des zéros (bits de poids fort).

### 2.3. Entiers relatifs

Il faut ici coder le signe du nombre. On utilise le codage en complément à deux, qui permet d'effectuer ensuite les opérations arithmétiques entre nombres relatifs de la même façon qu'entre nombres naturels.

- **Entiers positifs ou nuls** : On représente le nombre en base 2 et on range les bits comme pour les entiers naturels. Cependant, la cellule de poids fort est toujours à 0 : on utilise donc n - 1 bits. Le plus grand entier positif représentable sur n bits en relatif est donc  $2^{n-1} - 1$ .
- **Entiers négatifs** : Pour obtenir le codage d'un nombre x négatif, on code en binaire sa valeur absolue sur n - 1 bits, puis on complémente (ou inverse) tous les bits et on ajoute 1.

**Exemple** : soit à coder la valeur -2 sur 8 bits. On exprime 2 en binaire, soit 00000010. Le complément à 1 est 11111101. On ajoute 1 et on obtient le résultat : 1111 1110.

#### Remarque :

- (a) Le bit de poids fort d'un nombre négatif est toujours 1;
- (b) Sur n bits, le plus grand entier positif est  $2^{n-1} - 1 = 011\dots1$ ;
- (c) sur n bits, le plus petit entier négatif est  $-2^{n-1}$ .

### 3. Représentation des nombres réels (norme IEEE)

Soit à codifier le nombre 3,25, qui s'écrit en base 2 :  $(11, 01)_2$ .

On va normaliser la représentation en base 2 de telle sorte qu'elle s'écrive sous la forme :  $1, \dots \times 2^n$

Dans notre exemple  $11, 01 = 1, 101 \times 2^1$

La représentation IEEE code séparément le signe du nombre (ici +), l'exposant n (ici 1), et la mantisse (la suite de bits après la virgule), le tout sur 32 bits.

Le codage a la forme : seeeeeemmmmmmmmmmmmmmmmmmmmmmmmmmmmmmm.

- Le signe est représenté sur le bit de poids fort s, + est représenté par 0 et - par 1.
- L'exposant est codé sur les 8 bits e. On code en binaire la valeur n + 127.
- La mantisse est codée sur les 23 bits de poids faibles m.

#### Remarque :

1. Les exposants 00000000 et 11111111 sont interdits :
  - L'exposant 00000000 signifie que le nombre est dénormalisé;
  - L'exposant 11111111 indique que l'on n'a pas affaire à un nombre (on note cette configuration NaN, Not a Number, et on l'utilise pour signaler des erreurs de calculs, comme par exemple une division par 0).
2. Le plus petit exposant est donc -126, et le plus grand +127.

Voyons ce codage sur un exemple :

Soit à coder la valeur 525,5.

525,5 est positif donc le 1er bit sera 0.

Sa représentation en base 2 est la suivante : 1000001101,1

En normalisant, on trouve :  $1,0000011011 \times 2^9$

On ajoute 127 à l'exposant qui vaut 9 ce qui donne 136, soit en base 2 : 10001000

La mantisse est composée de la partie décimale de 525,5 en base 2 normalisée, c'est-à-dire 0000011011.

Comme la mantisse doit occuper 23 bits, il est nécessaire d'ajouter des zéros pour la compléter :

00000110110000000000000

La représentation du nombre 525,5 en binaire avec la norme IEEE est donc :

0 1000 1000 00000110110000000000000

Voici un autre exemple avec un réel négatif :

Soit à coder la valeur -0,625.

Le bit s vaut 1 car 0,625 est négatif

0,625 s'écrit en base 2 de la façon suivante : 0,101

On souhaite l'écrire sous la forme  $1.01 \times 2^{-1}$

Par conséquent l'exposant vaut 1111110 car  $127 - 1 = 126$  (soit 1111110 en binaire)

La mantisse est 010000000000000000000000 (seuls les chiffres après la virgule sont représentés, le nombre entier étant toujours égal à 1)

La représentation du nombre 0,625 en binaire avec la norme IEEE est :

1 1111 1110 010000000000000000000000

#### 4. Représentation des caractères

Les caractères sont des données non numériques : il n'y a pas de sens à additionner ou multiplier deux caractères. Par contre, il est souvent utile de comparer deux caractères, par exemple pour les trier dans l'ordre alphabétique.

Les caractères, appelés symboles alphanumériques, incluent les lettres majuscules et minuscules, les symboles de ponctuation (& ~ , . ; # " - etc...), et les chiffres.

Un texte, ou chaîne de caractères, sera représentée comme une suite de caractères.

Le codage des caractères est fait par une table de correspondance indiquant la configuration binaire représentant chaque caractère. Le code le plus connu est le code ASCII (American Standard Code for Information Interchange). Le code ASCII représente chaque caractère sur 7 bits (on parle parfois de code ASCII étendu, utilisant 8 bits pour coder des caractères supplémentaires).

Le code ASCII original, défini pour les besoins de l'informatique en langue anglaise) ne permet pas la représentation des caractères accentués, et encore moins des caractères chinois ou arabes. Pour ces langues, d'autres codages existent, utilisant 16 bits, voire plus, par caractères : le UNICODE.

#### 5. Représentation des images et vidéos :

La représentation la plus simple d'une image est d'utiliser une suite d'octets représentant les pixels de l'image :

- Pour une image en **noir et blanc**, un seul bit est nécessaire par pixel (00110 : Noir, Noir, Blanc, Blanc, Noir), la taille de l'image est donc : **largeur x hauteur** (en bits).

**Exemple** : soit pour une image  $800 \times 600 = 480\,000$  bits, soit 60 000 octets, soit environ 58 Ko.

- Pour une représentation en **niveau de gris** on utilisera généralement un octet par pixel (255 niveaux de gris possibles), la taille de l'image est donc : **largeur x hauteur** (en octets).

**Exemple** : soit pour une image  $800 \times 600 = 480\,000$  octets, soit 468,75 Ko.

- Pour une image en couleur, le codage d'une couleur est généralement donné par trois composantes : Rouge, Vert et Bleu d'où le terme RVB ou RGB en anglais. L'idée est donc d'utiliser un certain nombre d'octets par composantes. Par exemple, si l'on utilise 1 octet par composante (le cas général), la couleur 0,0,0 représente le noir et 255,255,255 le blanc.

**Exemple** : soit pour une image  $800 \times 600 \times 3 = 1\,440\,000$  octets, soit 1406,25 Ko, soit environ 1,4 Mo.

Une vidéo est une succession d'images à une certaine cadence. L'œil humain a comme caractéristique d'être capable de distinguer environ 20 images par seconde. Ainsi, en affichant plus de 20 images par seconde, il est possible de tromper l'œil et de lui faire croire à une image animée. On caractérise la fluidité d'une vidéo par le nombre d'images par secondes (en anglais frame rate), exprimé en FPS (Frames per second, en français trames par seconde).

#### 6. Notion de fichier

**Un fichier** est une collection d'informations numériques réunies sous un même nom, enregistrées sur un support de stockage et manipulées comme une seule unité. La dénomination des noms de fichiers suit des règles qui varient d'un système d'exploitation à un autre.

Sous Windows, la désignation d'un fichier est composée du nom de fichier suivi d'un point et d'une extension (**Nom.extension**). L'extension indique le type du fichier voir l'application qui a servi à sa création. Ainsi, le fichier « page web.html » est une page web et « cours.doc » est un document Word.

En vue de faciliter leur organisation, les fichiers sont disposés dans des **répertoires** ou **dossiers** eux-mêmes organisés selon le même principe de manière à former une hiérarchie arborescente. Les fichiers et répertoires sont souvent stockés sur des volumes (unités de disque) représentés par une lettre. Par convention, la lettre A désigne la disquette. La lettre C est réservée au disque principal. Les autres lettres peuvent représenter d'autres unités de stockage comme un disque secondaire, un lecteur CD ...

Les volumes sont divisés en dossier qui de façon récursive peuvent contenir d'autres dossiers. Ainsi le fichier « page web.html » peut se trouver dans le dossier « site web », qui se situe dans le dossier « Mes documents » qui se situe à la racine du disque D de mon ordinateur. Le chemin d'accès à ce fichier s'exprime de la façon suivante : D:\Mes documents \ site web \ page web.html.

Pour ouvrir un fichier, l'application correspondante doit être chargée. Sous Windows, c'est l'extension du fichier qui permet au système de connaître l'application à utiliser pour ouvrir ce fichier.

### **Exemple d'extension de fichiers :**

- ☞ Les .doc et .docx sont des documents créés par Word.
- ☞ Les .xls et .xlsx sont des documents créés par Excel.
- ☞ Les .bmp, gif, tiff, jpg, png sont des fichiers d'images.
- ☞ Les .mov, avi, mp4 sont des vidéos.
- ☞ Les wav, aif, mid, mp3, ram sont des fichiers de sons.
- ☞ Les .htm, .html, .css, .js, .php sont les fichiers diffusés par les serveurs Web.
- ☞ Les .exe sont des applications exécutables.