

- ✓ Acquérir les bases de la numération et du codage de l'information

SYNTHESE CODAGE



1. La numération

1.1. Les systèmes de numération et leur base

Un système de numération décrit la façon avec laquelle les nombres sont représentés.

Un système de numération est défini par :

- **Un alphabet** : ensemble de symboles ou de chiffres,
- **Des règles d'écritures des nombres** : Juxtaposition de symboles, système additif, soustractif...

1.2. Les principes de la numération actuelle

La numération actuelle permet de représenter un nombre, par juxtaposition de symboles (numération de position), appelés chiffres, appartenant à un ensemble.

Lorsqu'un nombre est écrit, la position respective des chiffres (**rang**) détermine leur **poids**. Ainsi dans le système décimal, nous avons les poids mille, cent, dix, unité, etc...

Le **poids** (p) d'un symbole, dans une juxtaposition, est lié à la **base** de numération et au **rang** (r) qu'occupe le symbole par la relation suivante :

$$p = \text{base}^r$$

Il est donc possible d'exprimer n'importe quel nombre dans le système décimal sous sa **forme canonique** :

$$(a_{n-1}a_{n-2} \dots a_2a_1a_0)_B = \sum_{i=0}^{i=n-1} a_i \cdot B^i = a_{n-1} \cdot B^{n-1} + a_{n-2} \cdot B^{n-2} + \dots + a_2 \cdot B^2 + a_1 \cdot B^1 + a_0 \cdot B^0 = (N)_{10}$$

1.3. Les principaux systèmes de numération

Les principaux systèmes de numérations utilisés en technologie sont **le décimal, le binaire et l'hexadécimal**.

Pour représenter ces systèmes de numération, il est nécessaire d'utiliser une notation pertinente :

Le système décimal	Le système binaire	Le système hexadécimal
$(N)_{10}$, $(N)_{\text{dec}}$ ou $N_{(10)}$.	$(N)_2$, $(N)_{\text{bin}}$, $N_{(2)}$, $0bN$ ou $\%N$.	$(N)_{16}$, $(N)_{\text{hex}}$, $N_{(16)}$, $0xN$, $\$N$, &hN ou $\#N$

Des opérations arithmétiques et logiques sont possibles dans ces différents systèmes de numération en appliquant les règles du système utilisé.

1.4. Les changements de base

Le transcodage (ou changement de base) est l'opération qui permet de passer de la représentation d'un nombre exprimé dans une base à la représentation du même nombre mais exprimé dans une autre base.

1.4.1. La conversion base X → base 10 (décimal)

Exemple base 2 (binaire) vers base 10 (décimal)

$$(1011)_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 8 + 2 + 1 = (11)_{10}$$

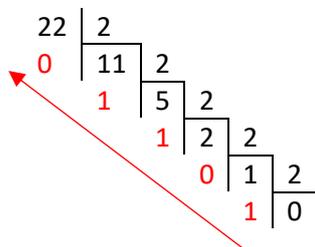
Exemple base 16 (hexadécimal) vers base 10 (décimal)

$$(1A3)_{16} = 1 \times 16^2 + 10 \times 16^1 + 3 \times 16^0 = 256 + 160 + 3 = (419)_{10}$$

1.4.2. La conversion base 10 (décimal) → base X

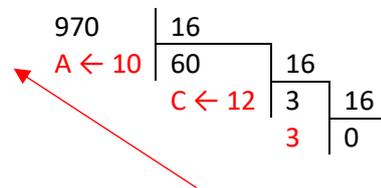
Exemple base 10 (décimal) vers base 2 (binaire)

$$(22)_{10} = (10110)_2$$



Exemple base 10 (décimal) vers base 16 (hexadécimal)

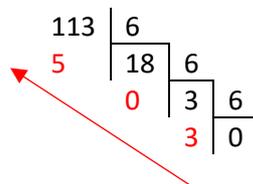
$$(970)_{10} = (3CA)_{16}$$



1.4.3. La conversion base X → base Y

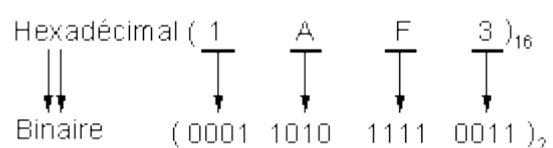
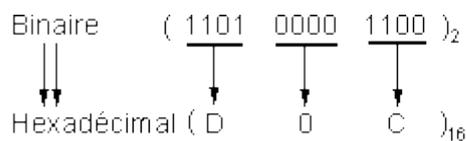
Exemple base 5 vers base 6

$$(423)_5 = 4 \times 5^2 + 2 \times 5^1 + 3 \times 5^0 = 100 + 10 + 3 = (113)_{10} = (305)_6$$



1.4.4. La conversion base 16 (hexadécimal) ↔ base 2 (binaire)

Chaque nombre en base 16 correspond à 4 chiffres en base 2 (en commençant depuis la droite)



2. Le codage de l'information

Le **codage de l'information** permet d'établir une correspondance qui permet sans ambiguïté de passer d'une représentation (dite externe) d'une information à une autre représentation (dite interne : sous forme binaire) de la même information, suivant un ensemble de règles précises.

2.1. L'unité de codage

L'unité de codage de l'information est un élément ne pouvant prendre que les valeurs 0 ou 1 ; le **bit**.

Il est possible de traiter des ensembles de plusieurs bits (4 bits → **quartet (nibble)** ; 8 bits → **octet (byte)** ; 16 bits → **mot (word)** ; 32 bits → **double mot (dword)** ; 64 bits → **quadruple mot (qword)**).

Dans un ensemble binaire :

- Le bit situé le plus à gauche est le bit le plus significatif, appelé **Bit de poids fort ou MSB** (Most Significant Bit)
- Celui situé le plus à droite est le bit le moins significatif, appelé **Bit de poids faible ou LSB** (Less Significant Bit).

2.1. Le codage des nombres

2.1.1. Le codage des entiers positifs (non signés)

Un codage sur n bits d'entiers positifs permet de représenter les nombres de **0 à $2^n - 1$**

Il est possible d'effectuer une addition entre deux nombres positifs codés sur n bits. Il faut par contre faire attention car il peut y avoir un **débordement**.

Il est aussi possible d'effectuer une multiplication entre deux nombres positifs codés sur n bits. Le résultat devra par contre être codé sur **$2 \times n$ bits**.

2.1.2. Le codage des entiers négatifs (signés)

Pour coder un nombre entier négatif il y a plusieurs possibilités mais celle qui est utilisée aujourd'hui par l'ensemble des ordinateurs est celle du **complément à 2**.

Un codage sur n bits d'entiers positifs ou négatifs avec la méthode du complément à 2 permet de représenter les nombres de **-2^{n-1} à $2^{n-1} - 1$** et le MSB d'un nombre positif est **0** et le MSB d'un nombre négatif est **1**.

2.1.3. Les codages spéciaux de nombres entiers

Le BCD (Binary Coded Decimal)

Le **code BCD** est utilisé principalement lorsque l'on utilise des afficheurs 7 segments.

Il faut ici coder les chiffres décimaux individuellement afin d'obtenir pour chaque chiffre décimal son équivalent codé en binaire sur 4 bits (quartet).

Le code GRAY (ou binaire réfléchi)

Le **code GRAY** ou binaire réfléchi est très utilisé, notamment dans les codeurs absolus de position que vous verrez dans le chapitre sur les capteurs.

Dans ce codage, un seul bit change d'état entre deux valeurs voisines. Il ne peut pas exister de résultat temporaire aberrant entre deux valeurs voisines.

Il est par contre impossible d'effectuer des opérations arithmétiques (addition, soustraction, ...) avec ce code.

Binaire réfléchi

0	0	0	0
0	0	0	1
0	0	1	1
0	0	1	0
0	1	1	0
0	1	1	1
0	1	0	1
0	1	0	0
1	1	0	0
1	1	0	1
1	1	1	1
1	1	1	0
1	0	1	0
1	0	1	1
1	0	0	1
1	0	0	0

Symétrie

2.2. Le codage des caractères

2.2.1. Le code ASCII

Le **code ASCII** (American Standard Code for Information Interchange) est principalement utilisé pour la transmission en série des caractères alphanumériques (Chiffres, lettres, caractères spéciaux) dans les systèmes informatiques.

Le code ASCII utilisé actuellement est le code étendu. Il permet le codage de caractères sur 8 bits, soit 256 caractères possibles.

2.2.2. L'UTF-8 (Unicode)

Pour coder de manière universelle l'ensemble des symboles utilisés quel que soit la langue (anglais, français, grec, chinois, ...) il faut attribuer à tout caractère ou symbole de n'importe quel système d'écriture de langue un nom et un identifiant numérique, et ce de manière unifiée, quelle que soit la plate-forme informatique ou le logiciel.

C'est ce que propose la norme **Unicode**.

Chaque symbole d'écriture est représenté par un nom et une valeur hexadécimale préfixée par « U+ ».

Pour stocker sur un support informatique un texte constitué de caractères Unicode, il faut encore choisir un procédé transformant chaque définition Unicode en une suite d'octets et réciproquement... C'est le processus d'encodage.

Actuellement un des systèmes d'encodage couramment utilisés (Unix, Internet, ...) est **UTF-8** (Unicode Transformation Format).

L'avantage du codage UTF-8 est qu'il est compatible avec le code ASCII.

2.3. Le codage des images

Dans le domaine du numérique, les images sont constituées d'une **matrice L x H** (Largeur x Hauteur) de points élémentaires que l'on nomme généralement des **pixels** (abréviation de Picture Element).

La **définition** d'une image est caractérisée par le nombre total de pixels dans l'image. Elle est définie par le nombre de pixels sur la hauteur de l'image multiplié par le nombre de pixels sur la largeur de l'image.

La **résolution** d'une image se caractérise par le nombre de pixels par unité de longueur, c'est en fait la densité de pixels de l'image. Celle-ci n'a rien à voir avec le nombre de pixels. La résolution s'exprime en ppp (pixel par pouce).

Chaque pixel a une couleur codée sur un nombre plus ou moins grand de bits. C'est le terme de **profondeur** d'image est utilisé pour spécifier le nombre de bits alloué au codage de chaque pixel.

- 1 (image noir et blanc),
- 8 (256 couleurs ou niveaux de gris)
- 24 (RVB : true color)
- 32 (RVBA : true color avec canal alpha).

