

Chapitre 1 : Système de numération et codage de l'information

Systèmes de numération:

L'ensemble des outils informatiques sont basés sur les mêmes principes de calcul (loi de tout ou rien). Les calculs habituels sont effectués dans le système de numération décimal, par contre le calculateur électronique ne peut pas utiliser ce système car le circuit électronique ne permet pas de distinguer 10 états. Le système de numération binaire ne comportera que 2 états 0 et 1.

1/ Système décimal: (base 10: 10 éléments de 0 à 9)

- Exemples: $9817 = 9.10^3 + 8.10^2 + 1.10^1 + 7.10^0$

$297,45 = 2.10^2 + 9.10^1 + 7.10^0 + 4.10^{-1} + 5.10^{-2}$

2/ Système binaire: (base 2: 2 éléments 0 et 1 => 2 bits)

Conversion binaire-décimal:

- **Exemple n°1:** $101101_{(2)}$

Bits	1	0	1	1	0	1
Puissance	2^5	2^4	2^3	2^2	2^1	2^0
Pondération	32	0	8	4	0	1

Somme des pondérations: $32+8+4+1 = 45$

Donc : $101101_{(2)} = 45_{(10)}$

- **Exemple n°2:** $1011,011_{(2)}$

Bits	1	0	1	1	0	1	1
Puissance	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
Pondération	8	0	2	1	0	0,25	0,125

Somme des pondérations: $8+2+1+0,25+0,125 = 11,375$

Donc : $1011,011_{(2)} = 11,375_{(10)}$

- Exemple n°1: $49_{(10)} = ?_{(2)}$

Quotient	Reste(/2)
49	
24	1
12	0
6	0
3	0
1	1
0	1

Donc: $49_{(10)} = 110001_{(2)}$

3/ *Système octal: (base 8: 8 éléments de 0 à 7)*

- Conversion octal-décimal:

- Exemple: $476_{(8)} = 4.8^2 + 7.8^1 + 6.8^0 = 256 + 56 + 6 = 318_{(10)}$

- Conversion décimal-octal:

On a 2 méthodes: « passage par la base 2 » ou « divisions successives par 8 »

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

- Exemple: $928_{(10)} = ?_{(8)}$

$928_{(10)} = 1\ 110\ 100\ 000_{(2)} = 1\ 6\ 4\ 0_{(8)}$

Ou bien:

Quotient	Reste (/8)
928	
116	0
14	4
1	6
0	1

Donc: $928_{(10)} = 1640_{(8)}$

Vérification: $1640_{(8)} = 1.8^3 + 6.8^2 + 4.8^1 + 0.8^0 = 512 + 384 + 32 = 928_{(10)}$

4/ Système hexadécimal: (base 16: 16 éléments, 0..9, A, B, C, D, E et F)

- Conversion hexadécimal-décimal:

- Exemple: $4CA2_{(16)} = ?_{(10)}$

$$4CA2_{(16)} = 4.16^3 + 12.16^2 + 10.16^1 + 2.16^0 = 16384 + 3072 + 160 + 2 = 19618_{(10)}$$

- Conversion décimal-hexadécimal:

On a 2 méthodes: « passage par la base 2 » ou « divisions successives par 16 »

Base 10	Base 16	Base 2
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

- Exemple: $469_{(10)} = ?_{(16)}$

Quotient	Reste (/16)
469	
29	5
1	13 (D)
0	1

Donc: $469_{(10)} = 1D5_{(16)}$

- Conversion hexadécimal-octal: (Passage par la base 2)

- Exemple: $AF9,D1_{(16)} = ?_{(10)}$

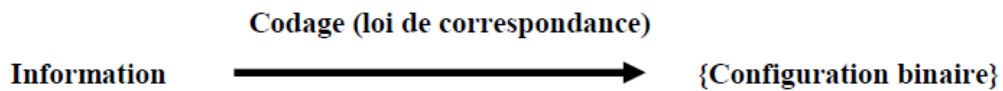
$AF9,D1_{(16)} = 1010\ 1111\ 1001,1101\ 0001_{(2)}$

$= 101\ 011\ 111\ 001,110\ 100\ 010_{(2)}$

$= 5\ 3\ 7\ 1,6\ 4\ 2_{(8)}$

Notions de codage:

Un dispositif logique ou numérique est destiné à manipuler des informations diverses qui doivent être traduites par un ensemble de 0 et 1, obtenu suivant une loi de correspondance préétablie: c'est l'opération de codage de l'information.



- Dans une machine, toutes les informations sont codées sous forme d'une suite de "0" et de "1" (langage binaire). Mais l'être humain ne parle généralement pas couramment le langage binaire.
- Il doit donc tout "traduire" pour que la machine puisse exécuter les instructions relatives aux informations qu'on peut lui donner.
- Le codage étant une opération purement humaine, il faut produire des algorithmes qui permettront à la machine de traduire les informations que nous voulons lui voir traiter. C'est une opération établissant une bijection entre une information et une suite de " 0 " et de " 1 " qui sont représentables en machine.

Codage de l'information

On peut coder n'importe quoi avec des bits (des octets), en particulier des nombres

- _ Idée : tout représenter avec des nombres
- _ Lettres → nombres
- _ Couleurs → nombres
- _ Texte simple → lettres → nombres
- _ Images → couleurs → nombres
- _ Sons → échantillons → nombres
- _ Vidéo → images + sons → nombres
- _ Texte mis en page → texte simple + instructions de mise en page
- _ Texte simple + texte instructions → nombres

Exemples de codes:

* Code ASCII: (**A**merican **S**tandard **C**ode for **I**nformation **I**nterchange) chaque touche du clavier est codée sur 8 bits, donc on peut coder 256 caractères.

Exemple: Touche 'A' ==> en code ASCII = « 01000001 » ??

* Code DCB (**D**écimal **C**odée en **B**inaire): utilisé uniquement pour les chiffres décimaux. Ce code est obtenu en remplaçant individuellement chacun des chiffres du nombre à représenter par son équivalent binaire pur.

Exemple: $2458_{(10)} = 0010\ 0100\ 0101\ 1000_{(DCB)}$

- Avantages: Représentation plus simple et très utile pour les systèmes d'affichage à 7 segments.

- Inconvénient: nécessite plus de bits.

Codage des caractères en ASCII

• Pour coder des caractères

○ on code le caractère en ASCII

○ on transforme le code ASCII en code binaire

caractère	ascii	décomposition de l'ascii	binaire
E	69	64 + 4 + 1	01 00 01 01
x	120	64 + 32 + 16 + 8	01 11 10 00
e	101	64 + 32 + 4 + 1	01 10 01 01
m	109	64 + 32 + 8 + 4 + 1	01 10 11 01
p	112	64 + 32 + 16	01 11 00 00
l	108	64 + 32 + 8 + 4	01 10 11 00
e	101	64 + 32 + 4 + 1	01 10 01 01
		$2^6\ 2^5\ 2^4\ 2^3\ 2^2\ 2^1\ 2^0$	

Octets, kilo-octets, méga-octets, giga-octets, ...

1 octet = 1 byte (en) = 8 bits

1 ko = 1024 octets

1 Mo = 1024 ko

1 Go = 1024 Mo

1 To = 1024 Go

Codage des images

Une image est un tableau de pixels (h x l)

- _ coder chaque pixel par une couleur, ligne par ligne
- _ Poids d'une image = taille occupée en mémoire
- _ image RGB, 3 octets par couleur, 100 x 100
- _ $100 \times 100 \times 3 = 30000 \text{ o} = 29,3 \text{ ko}$

Exemples

Définition de l'image	Noir et blanc (1 bit)	256 couleurs (8 bits)	65000 couleurs (16 bits)	True color (24 bits)
320x200	7.8 Ko	62.5 Ko	125 Ko	187.5 Ko
640x480	37.5 Ko	300 Ko	600 Ko	900 Ko
800x600	58.6 Ko	468.7 Ko	937.5 Ko	1.4 Mo
1024x768	96 Ko	768 Ko	1.5 Mo	2.3 Mo

Taille en octets d'une image

La taille en octets d'une image se définit ainsi :

$$\underbrace{\text{Nb_de_colonnes} * \text{Nb_de_lignes}}_{= \text{Nb de pixels}} * \text{Nb_de_bits_par_pixel} / 8$$

Exemple de trouver la taille d'une image

En fonction des différents paramètres du tableau et on appliquant la formule on peut définir la taille d'une image en octets.

Exemple : une image de dimension (640x480) en 256 couleurs aura pour taille:

Nombre de colonnes=640

Nombre de lignes 480

256 couleurs correspondent dans le tableau à **8** bits

Donc la taille est:

$$\mathbf{T = 640 \times 480 \times 8 / 8 = 307200 \text{ octets} \approx 300 \text{ koctets}}$$