Chapitre I

REPRESENTATION DES NOMBRES

1. SYSTEMES DE NUMERATION

1.1 Différents systèmes de numération

Il existe de nombreux systèmes de numération tels que le décimal, le binaire, l'octal, l'hexadécimal, le romain etc. Tous ces systèmes reposent sur l'utilisation d'un nombre de symboles dits de base et de leur règle d'utilisation. Leur appellation est généralement liée au nombre de symboles de base : dix (0, 1, 2, 3, 4, 5, 6, 7, 8, 9) pour le décimal, deux (0,1) pour le binaire, huit (0, 1, 2, 3, 4, 5, 6, 7) pour l'octal, seize (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F) pour l'hexadécimal…

- Le système décimal, considéré comme système universel est le plus communément utilisé dans tous les calculs scientifiques, commerciaux, et aussi dans notre activité quotidienne.

- Le système binaire est le système de base de l'algèbre de Boole. Il est utilisé par l'ensemble des technologies numériques.

- Les systèmes octal (base 8) et hexadécimal (base 16), issus du binaire, permettent de représenter une valeur binaire de manière plus compacte et offrent une conversion facile avec le système binaire. L'octal a quasiment disparu en faveur de l'hexadécimal.

- Le système duodécimal (base 12) a l'avantage d'avoir plus de diviseur que le décimal (2, 3, 4 et 6) et offre davantage de possibilités. Néanmoins son utilisation n'est pas courante. Il sert actuellement à compter les mois dans une année et les heures dans une journée.

- Le système vigésimal (base 20) était très utilisé auparavant. Quelques traces subsistent encore dans la numérotation actuelle : soixante-dix, quatre-vingt et quatre-vingt-dix. Leurs équivalents décimaux seraient respectivement : septante, octante et nonante, si la langue française avait suivi l'évolution des nombres.

- Le système sexagésimal (base 60) est utilisé actuellement dans la mesure du temps et des angles. Sur un cercle, il y a 360° qui sont divisés chacun en 60 minutes, divisées chacune en 60 secondes. Il en va de même pour chacune des heures de la journée qui sont divisées chacune en 60 minutes etc.

- Le système romain repose sur l'utilisation des sept symboles suivants avec leurs valeurs respectives :

$$M = 1000,\ D = 500,\ C = 100,\ L = 50,\ X = 10,\ V = 5,\ I = 1.$$

I, C, M sont appelés symboles principaux et V, L, D (multiples de 5), symboles secondaires. Les nombres sont ensuite formés par addition et/ou soustraction et aussi des puissances de mille, selon certaines règles :

1) Tout chiffre romain placé à la droite d'un autre qui lui est égal ou supérieur s'ajoute.

Exemple : VI = 5 + 1 = 6

2) Tout chiffre romain placé à la gauche d'un autre qui lui est supérieur se retranche.

Exemple : IX = 10 - 1 = 9

Il est possible d'ajouter jusqu'à trois chiffres, mais il n'est possible d'en retrancher qu'un seul à la fois.

Exemple : 8 s'écrit VIII mais pas IIX.

Tout chiffre placé entre deux chiffres plus grands, se retranche de celui de droite.

On ne peut pas écrire un symbole secondaire ou plusieurs symboles principaux à gauche d'un symbole plus grand.

Exemple : 45 s'écrit XLV (-10 + 50 + 5) et non pas VL.

3) Les valeurs sont regroupées en ordre décroissant (de droite à gauche) sauf pour les valeurs à retrancher selon la règle n° 2.

4) On ne peut pas répéter une même lettre plus de trois fois de suite à l'exception de M et I (quatre fois).

5) Un nombre surmonté d'un trait est égal à des milliers, deux traits à des millions etc.

Exemple : $\overline{\text{VIII}}$ = 8000

Le système romain est abandonné à cause de sa lourdeur au profit du système décimal dont la règle de composition est simplifiée par la notion des rangs des chiffres (unités, dizaines...). Ainsi, le nombre décimal 396 est la somme de 3 centaines, 9 dizaines et 6 unités.

1.2 Système de numération à base quelconque

De façon générale, un nombre N exprimé à l'aide d'une base b et des symboles a_i représentant les chiffres de la base peut être considéré comme un polynôme en puissance de b :

$$(N)_b = a_n \cdot b^n + a_{n-1} \cdot b^{n-1} + \ldots + a_1 \cdot b^1 + a_0 \cdot b^0 + a_{-1} \cdot b^{-1} + \ldots + a_{-m} \cdot b^{-m}$$

$$= \sum_{i=n}^{0} a_i \cdot b^i + \sum_{i=-1}^{-m} a_i \cdot b^i \qquad b > 1 \quad \text{et} \quad 0 \leq a_i \leq b-1$$

Il faut souligner que le chiffre calculé est en décimal. La notation $(N)_b$ indique seulement que les symboles seront exprimés en base b.

Dans l'usage courant, N s'écrit sous la forme : $a_n \ldots a_i \ldots a_0 , a_{-1} \ldots a_{-m}$.

Le chiffre de droite a_{-m} s'appelle le chiffre de poids le plus faible (LSB, Least Significant Bit). Le chiffre de gauche a_n s'appelle le chiffre de poids le plus fort (MSB, Most Significant Bit). L'élément a_i est le chiffre (digit) de rang i. La position respective des chiffres détermine leurs poids : Centaines, dizaines, unités, dixièmes, centièmes, en décimal. La virgule sépare le nombre en deux parties :

$a_n \ldots a_i \ldots a_0$, appelée partie entière et $a_{-1} \ldots a_{-m}$, la partie fractionnaire.

Quatre systèmes de numération sont fréquemment utilisés.

1.3 Système de numération décimal ou à base 10

C'est le système de numération que nous utilisons habituellement. Il est appelé ainsi car il utilise 10 symboles différents : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

Exemple :

$$(N)_{10} = 2012{,}65 = 2 . 10^3 + 0 . 10^2 + 1 . 10^1 + 2 . 10^0 + 6 . 10^{-1} + 5 . 10^{-2}.$$

En décimal, on note souvent N sans indice

1.4 Système de numération binaire ou à base 2

Le système de numération binaire a été introduit par Leibniz au 17ème siècle. Il s'applique à tout dispositif (mécanique, électrique, électronique etc.) ayant deux états d'équilibres stables (interrupteur ouvert ou fermé, courant passe ou ne passe pas, lampe allumée ou éteinte, transistor bloqué ou saturé…). Ceci est appelé par convention état 1 et état 0. On dispose donc de deux symboles {0, 1} encore appelés éléments binaires (bits), par contraction de l'expression anglaise binary digit (chiffre binaire).
L'expression générale de l'équivalence binaire d'un nombre $(N)_{10}$ est de la forme :

$$(N)_2 = a_n . 2^n + a_{n-1} . 2^{n-1} + \ldots + a_1 . 2^1 + a_0 . 2^0 + a_{-1} . 2^{-1} + \ldots + a_{-m} . 2^{-m}$$

Exemple :

$$(11010, 01)_2 = 1 . 2^4 + 1 . 2^3 + 0 . 2^2 + 1 .2^1 + 0 . 2^0 + 0 . 2^{-1} + 1 . 2^{-2}$$

$$= 16 + 8 + 0 + 2 + 0 + 0 + 0, 25$$

$$= (26, 25)_{10}$$

Le système binaire est le plus utilisé en électronique numérique. Il est aussi la base de calcul élémentaire des unités de calcul (cœur de toutes les unités arithmétiques et logiques des microprocesseurs*). Le bit est utilisé pour la mesure de la quantité d'informations et de la capacité d'une mémoire **. L'unité de cette mesure s'exprime en puissance de 2 ou en multiple de 2^{10} comme l'indique le tableau I-1. Néanmoins, son inconvénient majeur est qu'il faut en moyenne 3 à 4 fois plus de bits en binaire qu'il ne faut de chiffres en décimal pour dénombrer une quantité donnée d'unités.

---

* Microprocesseur : micro = intégration en un seul composant de faible encombrement.
Processeur = commande d'une unité de logique programmée. Le microprocesseur permet d'effectuer des opérations logiques ou arithmétiques.
** Mémoire : dispositif capable de stocker des informations de telle sorte que l'organe qui les utilise puisse à n'importe quel moment accéder à plusieurs unités de traitement.

- Les multiples des unités dans le système binaire

Le bit (0 ou 1) est la plus petite unité d'information manipulable par une machine numérique. L'octet (byte) est une unité composée de huit bits. Il permet de stocker un caractère tel qu'une lettre ou un chiffre. On définit ainsi :

- un word (mot) = 16 bits
- un dword (double mot) = 32 bits
- un qword (quadruple mot) = 64 bits
- un dqword (double quadruple word) = 128 bits

D'habitude, les préfixes de l'octet (kilo, méga, etc.) représentent (à tort) des multiples de $2^{10} = 1024$, au lieu de $10^3 = 1000$. A partir de 1998, la CEI (Commission Electrotechnique Internationale) a introduit une nouvelle norme pour représenter les préfixes informatiques binaires : « kibi » pour kilobinaire, « mébi » pour mégabinaire etc. Le tableau suivant représente les unités standardisées (système international ; SI) et les symboles en français (F).

Décimal			**Binaire**		
Symbole (SI)	Symbole (F)	Valeur en octet	Symbole (SI)	Symbole (F)	Valeur en octet
Kb (kilobyte)	**Ko** (kilooctet)	10^3	**Kib** (kibibyte)	**Kio** (kibioctet)	2^{10}
Mb (megabyte)	**Mo** (mégaoctet)	10^6	**Mib** (mebibyte)	**Mio** (mébioctet)	2^{20}
Gb (gigabyte)	**Go** (gigaoctet)	10^9	**Gib** (gibibyte)	**Gio** (gibioctet)	2^{30}
Tb (terabyte)	**To** (téraoctet)	10^{12}	**Tib** (tebibyte)	**Tio** (tébioctet)	2^{40}
Pb (petabyte)	**Po** (pétaoctet)	10^{15}	**Pib** (pebibyte)	**Pio** (pébioctet)	2^{50}
Eb (exabyte)	**Eo** (exaoctet)	10^{18}	**Eib** (exbibyte)	**Eio** (exbioctet)	2^{60}
Zb (zetabyte)	**Zo** (zétaoctet)	10^{21}	**Zib** (zebibyte)	**Zio** (zébioctet)	2^{70}
Yb (yottabyte)	**Yo** (yottaoctet)	10^{24}	**Yib** (yobibyte)	**Yio** (yobioctet)	2^{80}

Tableau I-1 *Unités d'informations*

1.5 Système de numération octal ou à base 8

Ce système est dérivé du binaire, mais dont la base comporte 8 chiffres, 0 1 2 3 4 5 6 7. L'intérêt de ce système est lié au fait que sa base est une puissance de 2, ($8 = 2^3$). Cette particularité lui confère des propriétés particulières de relation avec le système binaire (§1.4).

Exemple :

$$(256,1)_8 = 2 . 8^2 + 5 . 8^1 + 6 . 8^\circ + 1 . 8^{-1}$$

$$= 2 . 64 + 5 . 8 + 6 . 1 + 1 . 0,125$$

$$= (174, 125)_{10}$$

Dans la plupart des cas, ce système n'utilise pas plus de chiffres que le système décimal pour exprimer une même quantité d'unités, ce qui rappelons le, était un des inconvénients reproché au système binaire. Il est actuellement remplacé par des systèmes de base supérieurs (16, 32, etc.).

1.6 Système de numération hexadécimal, ou à base 16

C'est également un système de numération dérivé du binaire mais dont la base comporte 16 symboles. Puisqu'on n'a l'habitude que des 10 chiffres décimaux, ceux qui manquent seront indiqués par les premières lettres de l'alphabet : A B C D E F, avec

A = 10, B = 11, C = 12, D = 13, E = 14, F = 15.

L'ensemble des chiffres de la base 16 est donc :

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

Comme pour la numération octale, la base de ce système est une puissance de 2 ($16 = 2^4$) et présente le même intérêt que l'octal (§ 1.5).

Exemple :

$$(3D1, E)_{16} = 3 . 16^2 + 13 . 16^1 + 1 . 16^\circ + 14 . 16^{-1}$$
$$= 3 . 256 + 13 . 16 + 1 . 1 + 14 . 0{,}0625$$
$$= 768 + 208 + 1 + 0{,}875$$
$$= (977{,}875)_{10}$$

Le système hexadécimal nécessite au moins autant de chiffres que le système décimal pour exprimer la même quantité d'unités. On emploie le système hexadécimal dans le traitement de l'information car les nombres qu'il utilise sont plus courts et plus clairs que les nombres binaires. Les tableaux de l'annexe A-I donnent quelques puissances successives de 2, 3, 8 et 16.

Remarque : plus la base est faible, plus il faut de chiffres pour représenter le même nombre.

2. PASSAGE D'UNE BASE A UNE AUTRE

Il s'agit d'exprimer le même nombre N dans des bases différentes tel que : $N = (a_n \ldots, \ldots a_{-m})_{b1} = (c_p \ldots\ldots c_{-k})_{b2}$. On examine les cas généralement utilisés.

2.1 Conversion d'un nombre décimal $(N)_{10}$ dans une base b quelconque (dit aussi codage)

Supposons que le nombre N a une partie entière N_e et une partie fractionnaire N_f. L'étude sera faite en séparant les deux parties et en se limitant à l'ordre 4 des polynômes N_e et N_f.

2.1.1 Première méthode

a) Partie entière, méthode de divisions successives

Soit $N_e = a_3 . b^3 + a_2 . b^2 + a_1 . b^1 + a_0 . b^0$.

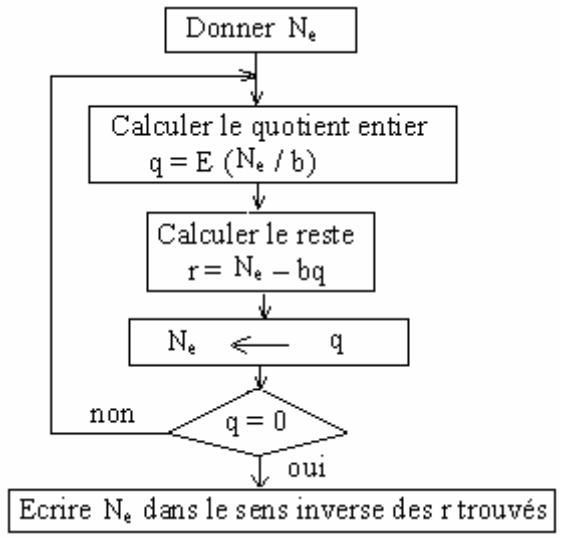
Ce nombre peut être décomposé de la façon suivante :

$$N_e = b\,(a_3 . b^2 + a_2 . b^1 + a_1 . b^0) + a_0$$
$$= b . q_1 + a_0$$

Factorisation qui revient à une division, avec :

$$q_1 = b\,(a_3 . b^1 + a_2 . b^0) + a_1 = b . q_2 + a_1$$
$$q_2 = b . (a_3) + a_2 = b . q_3 + a_2$$
$$q_3 = b . 0 + a_3 = 0 + a_3$$

La conversion s'obtient donc par une succession de divisions par b en appliquant les règles suivantes :
On divise le nombre N_e donné en décimal par la base b, le quotient q obtenu devient dividende, le reste est inscrit à droite et constitue le LSB de l'équivalent en base b. Les quotients obtenus sont à leur tour divisés par b jusqu'à ce qu'un zéro soit obtenu et on écrit de droite à gauche les restes r de ces divisions. Ces restes sont stockés en tableau et constituent le nombre cherché dans la nouvelle base. Cette conversion est illustrée par l'organigramme suivant :



A l'ordre 4, $(N_e)_b = a_3\ a_2\ a_1\ a_0$.

La procédure se généralise facilement à l'ordre supérieur.

- Exemple de représentation d'un chiffre décimal entier dans la base 2 :

$(25)_{10}$ en base 2

25	2				
1	12	2			
	0	6	2		
		0	3	2	
			1	1	2
				1	0
Restes	1	0	0	1	1

$(25)_{10} = (1\ 1\ 0\ 0\ 1)_2$

- Exemple de représentation d'un chiffre décimal entier dans la base 8

$(25)_{10}$ en base 8

25	8	
1	3	8
	3	0
Restes	1	3

$(25)_{10} = (31)_8$

De la même façon, on trouve : $(25)_{10} = (19)_{16}$

b) Partie fractionnaire, méthode de multiplications successives

Soit maintenant N_f la partie fractionnaire. En se limitant à l'ordre 4 on aura :

$N_f = a_{-1} . b^{-1} + a_{-2} . b^{-2} + a_{-3} . b^{-3} + a_{-4} . b^{-4}$
$b . N_f = a_{-1} + a_{-2} . b^{-1} + a_{-3} . b^{-2} + a_{-4} . b^{-3} = a_{-1} + P_1$
$b . P_1 = a_{-2} + a_{-3} . b^{-1} + a_{-4} . b^{-2} = a_{-2} + P_2$
$b . P_2 = a_{-3} + a_{-4} . b^{-1} = a_{-3} + P_3$
$b . P_3 = a_{-4} = a_{-4} + 0$

On multiplie plusieurs fois le nombre fractionnaire par la base b jusqu'à ce que le produit obtenu soit égal à 1,0. On supprime à chaque fois la partie entière du résultat obtenu. La suite des parties entières supprimées écrites de gauche à droite constitue la fraction dans la base b. La conversion de N_f s'obtient donc par une succession de multiplications de N_f par b.

- Exemples de représentation d'un chiffre décimal fractionnaire dans la base 2

1) **$(0,3125)_{10}$ en base 2**

$0,3125 . 2 = 0,625 \quad a_{-1} = 0$
$0,625 . 2 = 1,25 \quad a_{-2} = 1$
$0,25 . 2 = 0,5 \quad a_{-3} = 0$
$0,5 . 2 = 1,0 \quad a_{-4} = 1$
la partie fractionnaire est = 0, la conversion est terminée.

$(0,3125)_{10} = (0,0101)_2$

2) **$(0,24)_{10}$ en base 2**

$0,24 . 2 = 0,48 \quad a_{-1} = 0$
$0,48 . 2 = 0,96 \quad a_{-2} = 0$
$0,96 . 2 = 1,92 \quad a_{-3} = 1$
$0,92 . 2 = 1,84 \quad a_{-4} = 1$
$0,84 . 2 = 1,68 \quad a_{-5} = 1$
$0,68 . 2 = 1,36 \quad a_{-6} = 1$
$0,36 . 2 = 0,72 \quad a_{-7} = 0$
$0,72 . 2 = 1,44 \quad a_{-8} = 1$
$0,44 . 2 = 0,88 \quad a_{-9} = 0$
$0,88 . 2 = 1,76 \quad a_{-10} = 1$
$0,76 . 2 = 1,52 \quad a_{-11} = 1$ etc.

Dans ce cas, la partie fractionnaire ne s'annule jamais.

$(0, 24)_{10} = (0, 00111101011\ldots)_2$

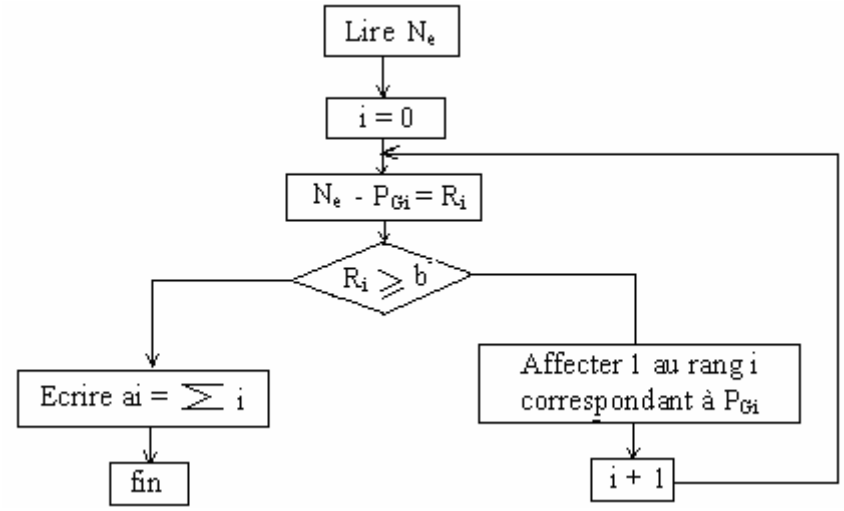
2.1.2 Deuxième méthode, méthode de soustractions successives

a) Partie entière

Nous avons vu précédemment que la partie entière d'un nombre N en base b s'écrit :

$(N_e)_b = a_n . b^n + a_{n-1} . b^{n-1} + \ldots + a_1 . b^1 + a_0 . b^0 \qquad b > 1 \text{ et } 0 \leq a_i \leq b-1$

Il s'agit de déterminer les valeurs des a_i. Pour cela, on cherche la plus grande puissance entière de b contenue dans Ne, retrancher cette quantité de Ne, considérer ensuite le reste obtenu et recommencer le processus. A chaque valeur b_i soustraite correspond un bit « 1 » de rang i, la valeur de a_i est égale à la somme de ces « 1 ». L'organigramme suivant résume cette procédure.



- Exemple de conversion d'un chiffre décimal entier en base 8

$(1086)_{10}$ en base 8

1ère étape : on a $8^4 > 1086 > 8^3 = 512$, c'est-à-dire que la plus grande puissance entière de 8, juste inférieure à 1086 est $8^3 = 512$.

2ère étape, on retranche cette quantité de 1086

$$\begin{array}{l} 1086 \\ \underline{-\ 512\ } \rightarrow 1.8^3 \\ = 574 \end{array}$$

On applique les mêmes étapes au reste de la soustraction, $8^4 > 574 > 8^3 = 512$

$$\begin{array}{l} 574 \\ \underline{-\ 512\ } \rightarrow 1.8^3 \\ = 62 \end{array} \qquad 8^2 > 62 > 8^1 . 7 = 56$$