

Chapitre I: Systèmes de numération

I) Introduction: L'automaticien est amené à manipuler des valeurs exprimées dans différentes bases et notamment au sein des systèmes informatiques; les instruments de mesure... et il est essentiel de posséder quelques notions sur les systèmes de numération en général et sur les systèmes binaire, octal et hexadécimal en particulier. ainsi que les différents types de codage et l'utilité de chacun d'entre eux, et aussi les opérations arithmétiques et la conversion entre base.

II) Quelques définitions:

II.1 Définition: une base B caractérise un système de numération dans lequel tout nombre N peut s'écrire:

$$N(B) = a_{n-1} B^{n-1} + a_{n-2} B^{n-2} + \dots + a_1 B^1 + a_0 B^0 + a_{-1} B^{-1} + \dots + a_{-m} B^{-m}$$

ou bien:

$$N = (a_{n-1} a_{n-2} \dots a_1 a_0 a_{-1} a_{-2} \dots a_{-m})_B$$

on peut écrire (1) sous sa forme générale: $N(B) = \sum_{i=-m}^{n-1} a_i \times B^i$
avec:

B: la Base du système de numération

a_i : le chiffre de l'indice i

i: est le rang du chiffre a_i

B^i : est le poids du chiffre a_i

Exemple:

$$(2008)_{10} = \underset{\substack{\uparrow \\ \text{Base } B}}{10} \quad \begin{matrix} \text{rang } i \\ \swarrow \quad \searrow \\ \text{chiffre } a_i \end{matrix} \quad \begin{matrix} 7 \times 10^3 & + & 0 \times 10^2 & + & 0 \times 10^1 & + & 8 \times 10^0 \\ \text{poids } B^i \end{matrix}$$

$$(37,51)_8 = 1 \times 8^1 + 5 \times 8^0 + 7 \times 8^{-1} + 3 \times 8^{-2}$$

II-2) Les bases de numération les plus utilisées:

la base 10: ^(décimale) le système de numération que nous employons couramment utilise 10 chiffres: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. on l'appelle pour cela le système décimal; $B=10$ ou système à base 10: $(2007)_{10}$

la base 2 (binaire)

le système décimal est malheureusement difficile à adapter aux mécanismes numériques, car il est difficile de concevoir du matériel électronique fonctionnant sur dix plages de tensions différentes. on lui préfère donc le système binaire caractérisé par:

- base $B=2$

- deux symboles: {0, 1} appelés « éléments binaires » ou « bits » (bit = Binary digit).

- le système binaire est pondéré par 2: les poids sont les puissances de 2: $(10010)_2 = 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 10$.
Conversion en binaire des chiffres de 0 à 15

en décimal	³ 2	⁴ 2	¹ 2	⁰ 2
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0

— poids

la base 8 (octal): Autrefois très utilisée, elle tend aujourd'hui à disparaître au profit de la base 16 suite à l'évolution technologique des composants (16 bits+). Les caractéristiques du système octal sont les suivantes:

- * la Base: $B=8$,
- * huit symboles: $\{0, 1, 2, 3, 4, 5, 6, 7\}$,
- * le système octal est pondéré par 8: les poids sont les puissances de 8.

$$(1000)_8 = 0 \times 8^0 + 0 \times 8^1 + 0 \times 8^2 + 1 \times 8^3 = (512)_{10}$$

L'intérêt de ce système est que la base est une puissance de $(8=2^3)$ et il nous facilite ainsi la conversion entre le système décimal et binaire.

la base 16 (hexadécimal): est apparue avec la logique microprogrammée et les microprocesseurs. Elle permet de traduire plus facilement un nombre binaire et autorise une représentation plus conviviale de grands nombres. Les caractéristiques du système hexadécimal sont les suivantes:

- * la base $B=16$,
- * seize symboles: $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$,
- * le système hexadécimal est pondéré par 16: les poids sont les puissances de 16.

$$(A9B)_{16} = \underbrace{11}_{11} \times \underbrace{16^0}_{1} + \underbrace{9}_{9} \times \underbrace{16^1}_{16} + \underbrace{10}_{10} \times \underbrace{16^2}_{256} = (2715)_{10}$$

III) Conversion d'un système de numération à l'autre:

III.1 Base B vers base 10: le passage d'une base quelconque en base 10 ne pose aucun problème, il suffit d'utiliser la représentation polynomiale.

$$\begin{aligned} (a_{n-1} a_{n-2} \dots a_0 \dots a_{-m+1} a_{-m})_B &= a_{n-1} \cdot B^{n-1} + a_{n-2} \cdot B^{n-2} + \dots + a_0 \cdot B^0 + \dots + a_{-m+1} \cdot B^{-m+1} + a_{-m} \cdot B^{-m} \\ &= (b_{n-1} b_{n-2} \dots b_{-m+1} b_{-m})_{10} \end{aligned}$$

Exemple: passage de la base 2 à la base 10.

$$(1001,1)_2 = \cancel{1 \cdot 2^3} + \cancel{0 \cdot 2^2} + \cancel{0 \cdot 2^1} + \cancel{1 \cdot 2^0} + 1 \cdot 2^{-1} = \cancel{8} + \cancel{0} + \cancel{0} + \cancel{1} + 0,5 = (9,5)_{10}$$

passage de la base 16 à la base 10.

$$(A1E.B)_{16} = 11 \times 16^{-1} + E \times 16^0 + 1 \times 16^1 + 10 \times 16^E = (2578.6875)_{10}$$

III.2 Base 10 vers base B

a) Première méthode: la première méthode consiste à soustraire successivement la plus grande puissance de B inférieure au reste du nombre en base 10.

Exemple:

conversion du nombre fractionnaire 100.75 en binaire.

$$100.75 = 1 \cdot 2^6 + 36.75$$

$$36.75 = 1 \cdot 2^5 + 4.75$$

$$4.75 = 1 \cdot 2^2 + 0.75$$

$$0.75 = 1 \cdot 2^{-1} + 0.25$$

$$0.25 = 1 \cdot 2^{-2} + 0.00$$

$$2^7 = 128 > 100.75$$

donc on doit commencer

par $2^6 = 64$

6 est la plus grande puissance

la puissance
la plus grande

6	5	4	3	2	1	0	-1	-2
↓	↓	↓	↓	↓	↓	↓	↓	↓
1	1	0	0	1	0	0	1	1

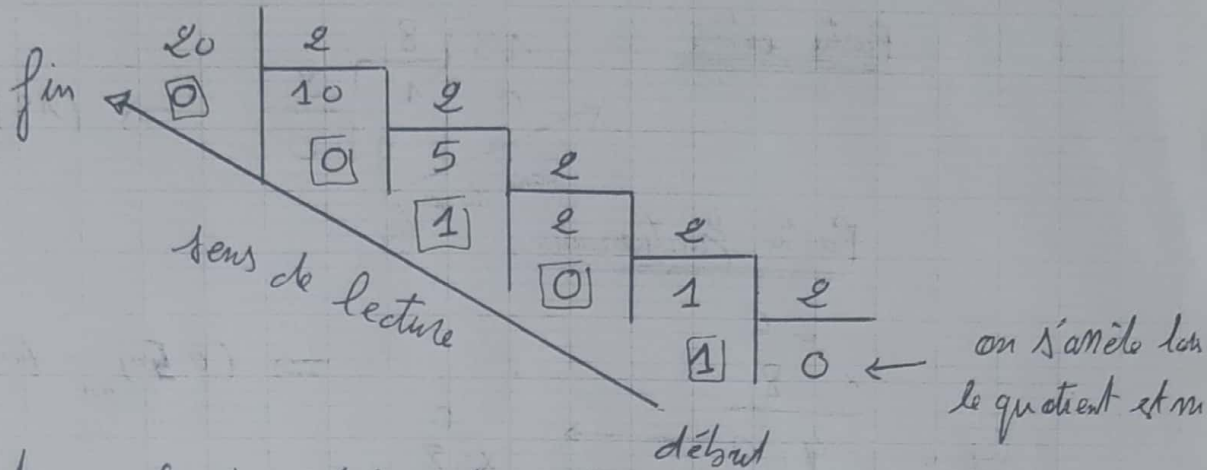
la puissance
la moins faible

$$\text{donc : } (100.75)_{10} = (1100100,11)_2$$

b) Deuxième méthode:

cette méthode consiste à diviser par B autant de fois que cela est nécessaire pour obtenir un quotient nul. Ensuite on écrit les restes successifs dans l'ordre inverse de celui dans lequel ils ont été obtenus.

Exemple: $(20)_{10} = ? ()_2$

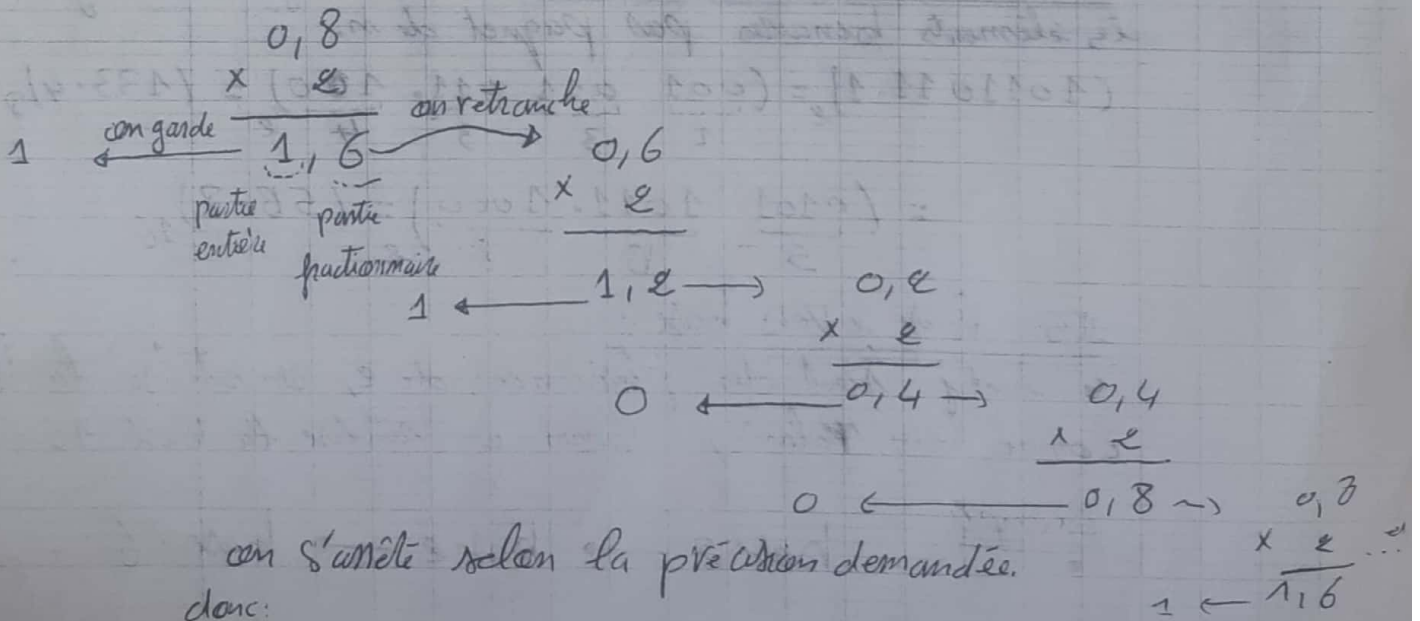


donc: $(20)_{10} = (10100)_2$

pour la partie fractionnaire on multiplie par 2 la partie fractionnaire. si le résultat est un nombre plus grand ou égale à 1, on garde la valeur, sinon le résultat intermédiaire est nul.

on retranche ensuite la valeur gardée à la partie fractionnaire restante et on recommence autant de fois qu'il faut pour obtenir la précision demandée.

Exemple:



donc:

$(0,8)_{10} = (0,11001)_2$

Exemple: $(13,5)_{10} \stackrel{?}{=} (15,40)_8$

Partie entière:

$$\begin{array}{c|c|c} 13 & 8 & \\ \hline 5 & 1 & 8 \\ & 1 & 0 \end{array} \Rightarrow (13)_{10} = (15)_8$$

sens de lecture

Partie fractionnaire:

$$\begin{array}{r} 0,5 \\ \times 8 \\ \hline 4,0 \end{array} \Rightarrow (0,50)_{10} = (0,40)_8$$

(4) ← 4,0 → (0) ← 0

III.3 Base 2^n vers base 2:

chaque symbole de la base $B = 2^n$ peut être représenté par n éléments binaires (bits).

$$(349)_{16} = (0011 \quad \underline{1010} \quad 1001)_2$$

$\downarrow$ 4 éléments

$$(742,5)_8 = (111 \quad \underline{100} \quad 010 \quad 101)_2$$

$\downarrow$ 3 éléments

III.4: Base 2 vers base 2^n : Il suffit de regrouper les éléments binaires par paquet de n .

$$\begin{aligned} (1011011,1)_2 &= (\underline{001} \quad \underline{011} \quad \underline{011} \quad \underline{100})_4 = (133,4)_4 \\ &= (\underline{0101} \quad \underline{1011} \quad \underline{1000})_8 = (5B,8)_{16} \end{aligned}$$

$\downarrow$ 5 4 4 4

III.5 Base i vers base j :

si i et j sont des puissances de 2, on utilise la base 2 comme un relais, sinon on utilise la base 10.

Exemple:

$$\begin{aligned} \text{base } 8 &\rightarrow \text{base } 2 \rightarrow \text{base } 16 \\ \text{ou} \quad \text{base } 5 &\rightarrow \text{base } 10 \rightarrow \text{base } 2 \end{aligned}$$

V) Les opérations arithmétiques.

V.1) Addition binaire: L'opération d'addition en binaire se fait de la même manière qu'en décimal à la seule différence que le binaire utilise deux symboles. on peut utiliser l'algorithme suivant.

$$0+0=0$$

$$1+1=0 \text{ et retenue } 1$$

$$0+1=1$$

$$1+0=1$$

$$1+1+1=1 \text{ et retenue } 1$$

Exemple:

$$\begin{array}{r} 6 \\ * + 7 \\ \hline 13 \end{array} \Rightarrow \begin{array}{r} 10110 \\ + 0111 \\ \hline 1101 \end{array}$$

$$*(A8ce)_{16} + (14BD)_{16} \stackrel{?}{=} (BD7E)_{16}$$

$$\begin{array}{cccc} 1010 & 1000^1 & 1100 & 0010 \\ + 0001 & 0100 & 1011 & 1101 \\ \hline \underline{1011} & \underline{1101} & \underline{0111} & \underline{1111} \\ B & D & 7 & F \end{array}$$

$$*(9e37)_{BCD} + (81)_{BCD} \stackrel{?}{=} (9318)_{BCD}$$

$$\begin{array}{cccc} 1001 & 0010 & 0011 & 111^1 \\ + 0000 & 0000 & 1000 & 0001 \\ \hline \underline{1001} & \underline{0010^1} & \underline{1011} & \underline{1000} \\ 9 & 2 \rightarrow 3 & + 0110 & 8 \\ & & \underline{0001} & \end{array}$$

Remarque: 1100^1 ne correspond à aucun chiffre décimal codé en BCD, puisque le BCD ne représente que