Architecture des ordinateurs

3 – Système de numération de position

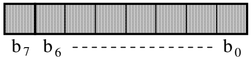
Philippe Darche
IUT Paris Descartes

Règles de calcul sur les puissances

- Elévation d'un nombre A à la puissance n :
- $f(n) = A^n = A \times A \times \ldots \times A$
  - (n-1) multiplications pour $n \geq 2$
  - $f(1) = A$
  - $f(0) = 1$
- $A^x \times A^y = A^{x+y}$
- $(A^x)^y = A^{x \times y}$
- $\frac{A^x}{A^y} = A^{x-y}$ avec, en particulier, $\frac{1}{A^x} = A^{-x}$

Format d'un nombre

- Format n d'un nombre
  = nombre n de chiffres (*digit*) composant ce mot
- En base 2, le chiffre se nomme le bit,
  contraction de *binary digit* (chiffre binaire)
- 2^n valeurs disponibles
  - exemple : $n = 8 \Rightarrow 256$ valeurs possibles



- Notion importante car la largeur de stockage dans un ordinateur est finie
  - notion associée : le dépassement de capacité

Notation indicielle

- Format n chiffres
- Nombre A exprimée dans une base B
  - $A = (a_{n-1}\ a_{n-2} \bullet\bullet\bullet\ a_1\ a_0)_B$
- Base sous-entendue
  - $A = a_{n-1}\ a_{n-2} \bullet\bullet\bullet\ a_1\ a_0$
  - exemple: A = 1023 (base 10 implicite)
    - $a_3 = 1$
    - $a_2 = 0$
    - $a_1 = 2$
    - $a_0 = 3$

Système de numération

- Le problème
  - comment représenter une quantité
- (Système de) Numération
  - abstraction pour représenter une quantité
  - méthode de représentation d'un nombre
  - types
    - figurée
    - parlée
    - écrite

Type de numération

- Numération figurée
  - un objet = un nombre
  - exemples : os, cordelette à nœuds, etc.
- Numération parlée
  - un nom = un nombre
  - exemples : un, deux, etc.
- Numération écrite
  - utilise des symboles (écriture) pour représenter la quantité
    - le chiffre
  - à ce système est associé un ensemble d'opérations

Aspects complémentaires du nombre

- Fonction cardinale
  - évaluation de la quantité
  - exemple : le mois de janvier a 31 jours
- Fonction ordinale
  - notion de succession des nombres (ordre)
  - exemple : le 31 janvier = le 31$^{\text{ème}}$ jour du mois

Système de numération écrite

- Numération additive
  - la valeur du nombre est égale à la somme des valeurs de chaque chiffre
- Numération de position
  - la valeur du chiffre dépend de sa position dans l'écriture du nombre
  - notion de base de numération

Observation d’un nombre en base 10 !

- Par exemple : prenons 123
  - deux notions transparaissent :
    - le concept de base
    - le concept de poids

Base de numération B

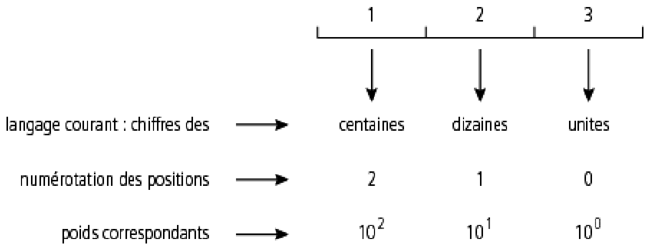
- B symboles de 0 à 9 puis A, B, etc. (par convention)
  - d'où $a_i < B$
  - les bases usuelles : les bases décimale (dix), hexadécimale (16), octale (8) et binaire (2)
- Utilisation d'une base dans un système de numération
  - système de numération de position

La base 10, notre référence

- Référence aux doigts des deux mains
- Représentation « naturelle » des nombres
- Mais difficile à implémenter
  - dix états physiques, donc 10 niveaux de détection
    - à interpréter
    - à mémoriser
    - à transmettre
  - un exemple : l’ENIAC

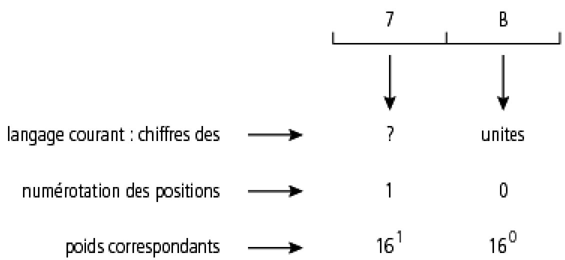
Poids d’un chiffre

- B^i = poids du chiffre a de rang ou position i
- Reprenons notre nombre



Le même nombre dans une autre base

- Laquelle ?



Quelques puissances de 2 positives et négatives

n	2^n	2^{-n}
0	1	1
1	2	0,5
2	4	0,25
3	8	0,125
4	16	0,0625
5	32	0,03125
6	64	0,015625
7	128	0,0078125
8	256	0,003906250
9	512	0,001953125
10	1024	0,000976562
11	2048	0,000488281
12	4096	0,000244141
13	8192	0,000122070

Décomposition d'un nombre A dans la base B

- Somme pondérée de ces chiffres

$$A = a_{n-1} \times B^{n-1} + a_{n-2} \times B^{n-2} + \ldots + a_1 \times B^1 + a_0 \times B^0 = \sum_{i=0}^{n-1} \left(a_i \times B^i \right)$$

- Système de numération simple de position

Exemples

- Base 2 ou base binaire
  - $A_2 = 1101$
  - $A_{10} = 1 \times 2^0 + 0 \times 2^1 + 1 \times 2^2 + 1 \times 2^3 = 13$
- Base 10 ou base décimale
  - $A = 324 = 4 \times 10^0 + 2 \times 10^1 + 3 \times 10^2$
    $= 4 + 20 + 300$

Un contre-exemple

- Le système de numération additif
  - un exemple : le système de numération romain
    - les chiffres romains : I, V, X, L, C, D et M
    - pour représenter respectivement les valeurs 1, 5, 10, 50, 100, 500 et 1000
  - exemples:
    - XIII = 13
    - XXIV = 24

Correspondance entre quelques bases classiques

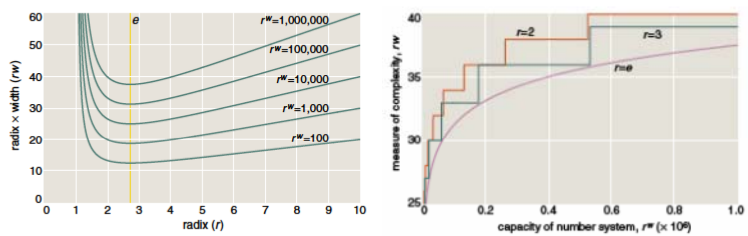
base → nombre base dix ↓	2	3	4	5	6	7	8	9	11	12	16
0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1
2	10	2	2	2	2	2	2	2	2	2	2
3	11	10	3	3	3	3	3	3	3	3	3
4	100	11	10	4	4	4	4	4	4	4	4
5	101	12	11	10	5	5	5	5	5	5	5
6	110	20	12	11	10	6	6	6	6	6	6
7	111	21	13	12	11	10	7	7	7	7	7
8	1000	22	20	13	12	11	10	8	8	8	8
9	1001	100	21	14	13	12	11	10	9	9	9
10	1010	101	22	20	14	13	12	11	A	A	A
11	1011	102	23	21	15	14	13	12	10	B	B
12	1100	110	30	22	20	15	14	13	11	10	C
13	1101	111	31	23	21	16	15	14	12	11	D
14	1110	112	32	24	22	20	16	15	13	12	E
15	1111	120	33	30	23	21	17	16	14	13	F
16	10000	121	100	31	24	22	20	17	15	14	10

La base optimale

- Quelle est la base qui permet de stocker une étendue de valeurs dans un minimum de place ?
- Le produit base $\times$ format représente le coût du matériel
  - ex. : soit à représenter 999 999.
    Le format n minimum nécessaire est :
    - base 10 → n = 6
    - base 2 → n = 20
    - base 3 → n = 13
- Minimisation de cette fonction de coût
  = minimisation du matériel

La base optimale

- Réponse : la base e
  - arrondie → la base 2 ou 3



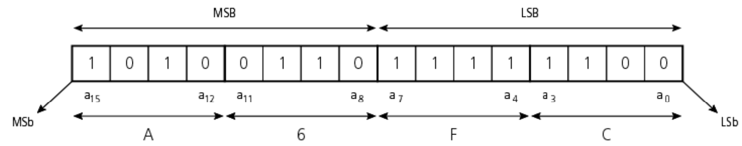
[Hayes 01]

Implémentation de la base ternaire

- Trois symboles
  - 0,1 et 2
  - -1, 0 et +1 (équilibrée – *balanced ternary*)
- Unités : trit et tryte (*ternary digit* et *byte* à 3 bits !)
- Quelques machines
  - les machines à calculer de Thomas Fowler (1840)
  - la série des Setun (1958) en URSS
- Trop complexe à implémenter malgré de grands supporters
  - « *The .. simple arithmetic of this number system will prove to be quite important some day when the "flip-flop" is replaced by a "flip-flap-flop"* » [Knuth 69]

Un peu de vocabulaire

- Représentation interne du nombre A = A6FCh
  - ⇒ format n = 16 bits

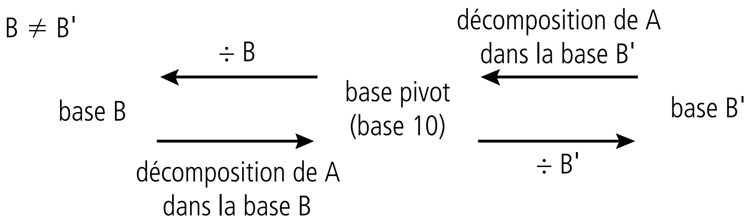


Un peu de vocabulaire (suite)

- MSb : *Most Significant <u>bit</u>*
  - le bit le plus significatif (*i.e.* de poids le plus fort)
- LSb : *Least Significant <u>bit</u>*
  - le bit le moins significatif (*i.e.* de poids le plus faible)
- MSB : *Most Significant <u>Byte</u>*
  - l'octet le plus significatif
- LSB : *Least Significant <u>Byte</u>*
  - l'octet le moins significatif
- Par extension, MSD et LSD (D pour *Digit*)

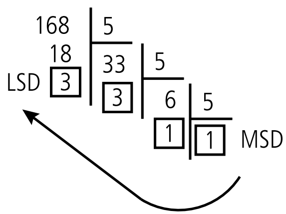
Conversion d'un nombre entier naturel

- Schéma général de conversion



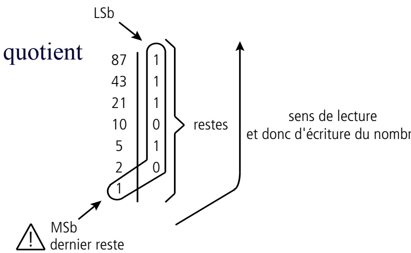
Méthode par divisions entières successives

- Passage base 10 à base quelconque
- Condition d'arrêt : un quotient devenant décimal (Q < D)
- Un premier exemple
  - $168_{10} = 1133_5$



Méthode par divisions entières successives

- Diviseur implicite, pour simplifier l'écriture
- Un exemple (B = 2)
  - $87_{10} = 1010111_2$
  - ne pas oublier le dernier quotient



Autre méthode

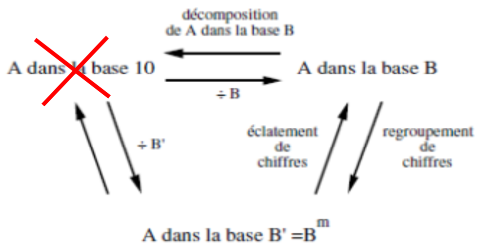
- Recherche des poids dans le nombre
  - plus particulièrement adaptée à la base 2
    - le nombre est vu comme une somme de puissance de 2

Conversion inverse

- *i.e.* d'une base quelconque à la base 10
- Formule de décomposition du nombre dans sa base initiale
- Voir exemples du transparent 10

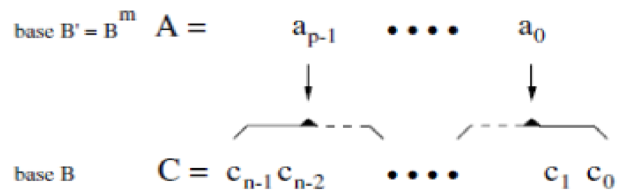
Méthodes particulières

- Par regroupement ou par éclatement



Méthodes par éclatement de chiffres

- Condition : B' = B^m

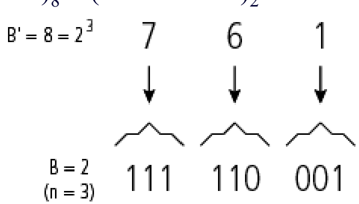


Exemple : passage base 8 → base 2

- Constat : $8 = 2^3$

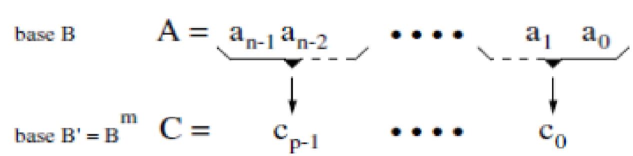
⇒ Conversion de chaque chiffre dans la base 2 au format n = 3 chiffres

- Exemple: $(761)_8 = (111\ 110\ 001)_2$



Méthodes par regroupement de chiffres

- Condition : B' = B^m



Exemple : passage base 2 → base 16

- Constat : $16 = 2^4$

⇒ Regroupement par paquet de 4 bits et conversion de chaque paquet en son équivalent hexadécimal

- Exemple : $(1011001110)_2 = (2CE)_{16}$

$$\underbrace{0\,0\,1\,0}_{2}\quad\underbrace{1\,1\,0\,0}_{C}\quad\underbrace{1\,1\,1\,0}_{E}$$

Tableau de conversion !

- Limitation par la taille de la table
- Par contre, utile pour les deux méthodes précédentes
- Exemple :
  - table pour les bases 10, 2 et 16

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

Le schéma de Horner

- Partie entière

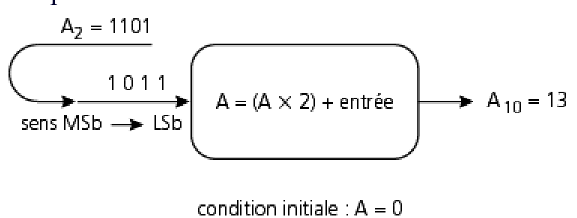
$$A_e = ((\ldots((a_{n-1} \times B + a_{n-2}) \times B + a_{n-3}) \times B + \ldots) \times B + a_1) \times B + a_0$$

- Partie fractionnaire

$$A_f = ((\ldots((a_{-f} / B + a_{-f+1}) / B + a_{-f+2}) / B + \ldots) / B + a_{-1}) / B$$

Le schéma de Horner

- Exemple



- Pour une partie fractionnaire, ordre inverse et ÷ à la place de ×

Les compléments à la base

- Utilisés en particulier pour les représentations entières signées
- La notion de format n intervient
- Le complément vrai, à la base B ou à B^n

$$\hat{A}_B = B^n - A_B$$

- Le complément restreint, diminué ou à B-1

$$\overline{A}_B = (B^n - 1) - A_B$$

Exemples (format n = 3)

- Le complément vrai, à la base B ou à B^n
  - base 10, A = 010 $\Rightarrow \widehat{A}_{10} = 990$
  - base 2, A = 111 $\Rightarrow \widehat{A}_2 = 001$
- Le complément restreint, diminué ou à B-1
  - base 10, A = 210 $\Rightarrow \overline{A}_{10} = 789$
  - base 2, A = 101 $\Rightarrow \overline{A}_2 = 010$

Compléments en base binaire

- Complément à la base diminuée (ou restreint)
  - calcul de $\overline{A}$:
    - complémentation de tous les bits (0 → 1 et 1 → 0)
- Complément à la base ou vrai

$$\hat{A} = \overline{A} + 1$$

- Opérations symétriques

$$\overline{\overline{A}} = A \text{ et } \hat{\hat{A}} = A$$

Conclusions

- Le format de travail n est important car il détermine l'étendue des valeurs numériques en machine
- Le choix de la base 2 pour représenter un nombre en machine n'est pas forcément un mauvais choix !
- La notion de compléments permet de représenter les nombres négatifs et donc de réaliser des additions algébriques

Bibliographie

- [Buchholz 62] : "Planning a Computer System - Project Stretch". Edited by Werner Buchholz. McGraw-Hill Book Company 1962.
- [Hayes 01] : Brian Hayes : "Third Base". Computing Science. American Scientist, Vol. 89, N° 6, pp. 490-494. November–December 2001.