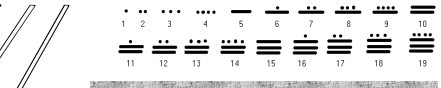
NF92
Représentation des nombres et conversion

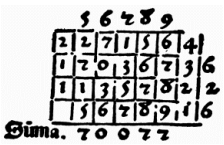
Pavol BARGER

Introduction

- Représentation des nombres et concepts
- Principes de conversion
- Bits, horloge, discrétisation, échantillonnage et binarisation
- Notion de mémoire
- Conventions/formats de représentation
- Arithmétique

Représentation de nombres





Concepts

Numérotation par position, ex : romains (I, V, X, L, C, D, M) + règles gérant la valeur par position :

•Toute lettre placée à la droite d'une autre figurant une valeur supérieure ou égale à la sienne s'ajoute à celle-ci.
•Toute lettre **d'unité** placée immédiatement à la gauche d'une lettre plus forte qu'elle, indique que le nombre qui lui correspond doit être retranché au nombre qui suit.
•Les valeurs sont groupées en ordre décroissant, sauf pour les valeurs à retrancher selon la règle précédente.
•La même lettre ne peut pas être employée quatre fois consécutivement sauf M.
•Lettre **d'unité** : I est une unité pour V et X, X est une unité pour L et C, C est une unité pour D et M.

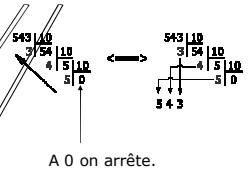
TOP

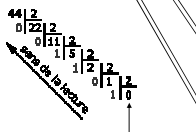
Concepts

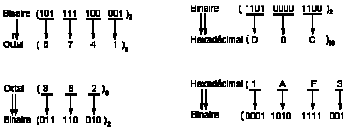
- Signe et ses glyphes
- Valeur associée simple
- Base de calcul régulière
- Numérotation par position régulière
- Ex :
  - base 10
  - signes/valeurs {0,1,2,3,4,5,6,7,8,9}
  - 2734 = 2*1000+7*100+3*10+4*1

TOP

Principes de conversion







Conversion

- Base 10 vers base 2
- Base 2 vers base 10
- Base 10 vers base 8
- Base 8 vers base 10
- Base 10 vers base 16
- Base 16 vers base 10
- Base 2 vers base 8 et 16
- Base 8 vers 2 et 16
- Base 16 vers base 2, 8

7

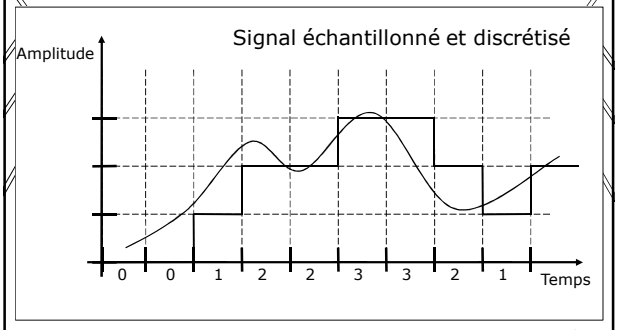
Bit & horloge

- Unité de base dans l'informatique
- Binary digit
  - Impossibilité de fabriquer une machine à trois (ou plus) états stables
- Toute opération dans l'ordinateur se fait sur un top d'horloge

TOP

8

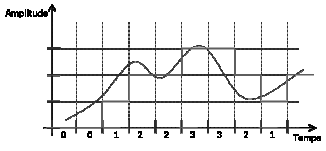
Discrétisation & échantillonnage



9

Binarisation

Décimal	Binaire
0	0
1	1
2	10
3	11
4	100
5	101



Suite originale 0 0 1 2 2 3 3 2 1
Suite binarisée 0 0 1 10 10 11 11 10 1

Résultat 001101011101

Impossible de revenir à l'original

10

Binarisation : correction

Décimal	Binaire
0	000
1	001
2	010
3	011
4	100
5	101

Mots de la même longueur p
Ici p=3
Chaque nombre est représenté comme une série de p bits

Essentiel : p doit toujours être connu

P limite la valeur max. qu'on peut représenter à 2^p-1

Suite originale 0 0 1 2 2 3 3 2 1
Suite binarisée 000 000 001 010 010 011 011 010 01

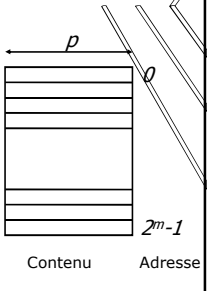
Résultat 000000001010010011011010001

Sous condition que p=3

11

Notion de mémoire

- Un espace de mots
  - Mot : une suite de bits de longueur p (tous la même)
- Chaque mot se trouve à un endroit spécifique
  - Adresse : la position du mot par rapport au début de la mémoire
- Taille de mémoire
  - Nombre de mots (2^m-1) fois taille de mots (p)



12

Mémoire : exemple

Suite originale 0 0 1 2 2 3 3 2 1
Résultat 000|000|001|010|010|011|011|010|001
Sous condition que p=3

0	0	0	0
0	0	0	1
0	0	1	2
0	1	0	3
0	1	0	4
0	1	1	5
0	1	1	6
0	1	0	7
0	0	1	8

Contenu Adresse

13

Mémoire : ATTENTION

La mémoire n'est jamais vide :

Il y a toujours des 0 et des 1 dedans

Conséquence : ne jamais utiliser le contenu d'un mot mémoire sans l'avoir préalablement initialisé correctement

14

Conventions/formats de représentation

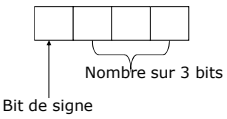
Représentation de nombres

- Entiers positifs (déjà vu avant)
- Négatifs
- Virgule fixe
- Virgule flottante

15

Nombres négatifs

- - (moins) n'existe pas en informatique
  - on utilise alors un bit de signe (0 si positif, 1 si négatif)



- Comment représenter sur 4 bits: 10, -2 ?
- Combien vaut 2+(-3) ?
  - Solution : représentation complémentée à 2

16

Que faire entre 0 et 1 ? (retour sur les conversions)

- Partie fractionnaire
- Le nombre est exprimé en puissances négatives de 2

2^0	1
2^{-1}	0.5
2^{-2}	0.25
2^{-3}	0.125

- exemple : $(0.625)_{10}=0.5+0.125=(0.101)_2$

17

Exemple : 0,347

0,347 . 2 = 0,694 < 1 je pose 0 : 0,347 = 0,0…
0,694 . 2 = 1,388 > 1 je pose 1 : 0,347 = 0,01…
0,388 . 2 = 0,766 < 1 je pose 0 : 0,347 = 0,010…
0,766 . 2 = 1,552 > 1 je pose 1 : 0,347 = 0,0101…
0,552 . 2 = 1,104 > 1 je pose 1 : 0,347 = 0,01011…
0,104 . 2 = 0,208 < 1 je pose 0 : 0,347 = 0,010110…
0,208 . 2 = 0,416 < 1 je pose 0 : 0,347 = 0,0101100…
0,416 . 2 = 0,832 < 1 je pose 0 : 0,347 = 0,01011000…
0,832 . 2 = 1,664 > 1 je pose 1 : 0,347 = 0,010110001…
0,664 . 2 = 1,328 > 1 je pose 1 : 0,347 = 0,0101100011…

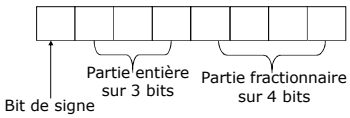
Continuer cette méthode jusqu'à atteindre la précision souhaitée
Ici : 0,347 décimal est arrondi à 0,0101100011 binaire
avec une précision absolue de 2 à la puissance -10

http://pagesperso-orange.fr/arsene.perez-mas/numeration/reels.htm
http://www.iut-info.univ-lille1.fr/~place/ASR1/Cours-ASR1-Codage.pdf

18

Nombres (réels) à virgule fixe

- La virgule n'existe pas
  - réservation de places

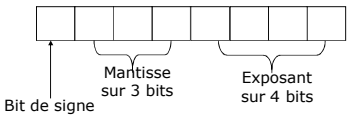


- Problème : Compromis entre l'étendu et la précision

19

Nombres (réels) à virgule flottante

- La virgule n'existe pas
  - réservation de places



- Exemple : 1, 2, 3, 2.2, 0.5, 16, 17
- Problème : Échelle à résolution variable

20

IEEE 754

- Un format standardisé pour représenter des nombres
- Formats de nombres flottants :
  - 32 bits : 1 bit de signe, 8 bits d'exposant (-126 à 127), 23 bits de mantisse
  - 64 bits : 1 bit de signe, 11 bits d'exposant (-1022 à 1023), 52 bits de mantisse
  - 80 bits : 1 bit de signe, 15 bits d'exposant (-16382 à 16383), 64 bits de mantisse
- Les quatre modes d'arrondi :
  - Vers moins l'infini
  - Vers plus l'infini
  - Vers zéro
  - Au plus proche

21

Placement en mémoire

65 41 101 A A $=(01000001)_2$

Adresse	Mémoire								Contenu
00h	0	1	0	0	0	0	0	1	A
01h	0	1	1	0	1	1	1	0	n
02h	0	1	1	0	1	1	1	0	n
03h	0	1	1	0	0	1	0	1	e
04h									
05h									
.....									
FFh									

Pour afficher un char. :
- Lire l'octet à l'adresse
- Le traduire en le comparant avec la table ASCII
- Afficher le caractère
- Lire l'octet suivant

Attention: C'est à vous de lui dire comment interpréter l'octet!

22

Arithmétique

- Soustraire c'est additionner, retour sur la complémentation à la base +1
- Multiplication d'un nombre entier (en binaire) par 2
- Division d'un nombre entier (en binaire) par 2
- Alignement de mantisse en base 10
- Alignement de mantisse en base 2
- Multiplication en base 10
- Multiplication en base 2
- Division en base 10
- Division en base 2

23