



DIRO
IFT 1215

INTRODUCTION AUX SYSTÈMES INFORMATIQUES

SYSTÈMES DE NOMBRE

Max Mignotte

Département d'Informatique et de Recherche Opérationnelle
Http: [//www.iro.umontreal.ca/~mignotte/](http://www.iro.umontreal.ca/~mignotte/)
E-mail: mignotte@iro.umontreal.ca

SYSTÈMES DE NOMBRE

SOMMAIRE

Introduction	2
Représentation en Base n	3
Conversion des nombres entiers	5
Conversion entre Base	10
Représentation des nombres fractionnaires ...	11
Conversion des nombres fractionnaires	14
Troncature	15

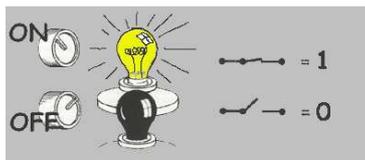
SYSTÈMES DE NOMBRE

INTRODUCTION

Pourquoi La Représentation Binaire (Base 2) ?

La base d'un système de nombre est le nombre de chiffres (*digit*) incluant le zéro qui existe dans le système de nombre

- Les humains utilisent pour représenter les nombres la numérotation décimale (base 10 ou 10 digits) (pas pour compter le temps qui s'écoule ...)
 - ▷ Origine: compter sur les doigts
 - ▷ *Digits* du mot latin *digitus* = doigts
- Les ordinateurs précédant 1945 (Ex: Mark 1, Eniac, etc.) utilisant la représentation décimale (base 10)
- **John V. Neumann** qui proposa la numérotation binaire (1945), pourquoi ?
 - ▷ Simplifiait l'architecture des ordinateurs (arithmétique, architecture, etc.)
 - ▷ Utilisé à la fois pour les instructions et les données (homogénéité, simplicité, etc.)
 - ▷ Relation naturel entre le calcul logique Booléen et la commutation ON/OFF des transistors (dispositif physique disponible, rapidité, stabilité, simplicité)



SYSTÈMES DE NOMBRE

INTRODUCTION

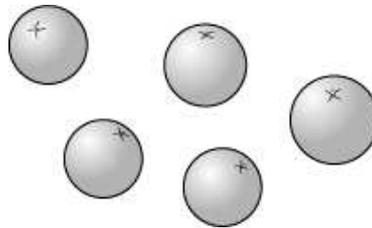
Nombre : Représentation Physique

- Différents système de représentation mais même nombre d'orange :

- ▷ Anciennes civilisation : IIIII
- ▷ Chiffres Romains : V
- ▷ Chiffres Arabes : 5

- Différentes bases mais même nombre d'orange :

- ▷ Base 10: 5_{10}
- ▷ Base 2: 101_2
- ▷ Base 3 : 12_3
- ▷ etc.



Conversion Possibles

Pour un humain, représentation possible de -13.2125_{10}

Pour un ordinateur :

- ▷ Seulement 2 digits (0,1)
- ▷ Pas de signe - (moins)
- ▷ Pas de point décimal (-13.2125_{10})

SYSTÈMES DE NOMBRE

REPRÉSENTATION EN BASE N

Représentation en Base n

Base 10 (valeurs permises pour les digits : 0,1,2,3,4,5,6,7,8,9)

$$527 = (5 \times 10^2) + (2 \times 10^1) + 7$$

Base 8 (valeurs permises pour les digits : 0,1,2,3,4,5,6,7)

$$\begin{aligned} 527_{10} &= (1 \times 8^3) + (0 \times 8^2) + (1 \times 8^1) + (7 \times 8^0) \\ &= 1017_8 \end{aligned}$$

Base 16 (valeurs permises : 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F)

$$\begin{aligned} 527_{10} &= (2 \times 16^2) + (0 \times 16^1) + 15 \\ &= 20F_{16} \end{aligned}$$

Base 2 (valeur permise pour les digits (*bits*) : 0,1)

$$\begin{aligned} 527_{10} &= (1 \times 2^9) + (0 \times 2^8) + (0 \times 2^7) + (0 \times 2^6) + (0 \times 2^5) \\ &\quad + (0 \times 2^4) + (1 \times 2^3) + (1 \times 2^2) + (1 \times 2^1) + (1 \times 2^0) \\ &= 100001111_2 \end{aligned}$$

Compter en Base 2

NUMBER	EQUIVALENT	DECIMAL EQUIVALENT
0	0×2^0	0
1	1×2^0	1
10	$1 \times 2^1 + 0 \times 2^0$	2
11	$1 \times 2^1 + 1 \times 2^0$	3
100	1×2^2	4
101	$1 \times 2^2 + 1 \times 2^0$	5
110	$1 \times 2^2 + 1 \times 2^1$	6
111	$1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$	7
1000	1×2^3	8
1001	$1 \times 2^3 + 1 \times 2^0$	9
1010	$1 \times 2^3 + 1 \times 2^1$	10

SYSTÈMES DE NOMBRE

CONVERSION DES NOMBRES ENTIERS

Conversion base $n \rightarrow$ base 10

$$\text{Valeur}_{\text{décimale}} = \sum_{i=0}^{n_b-1} \text{Digit}_i \times \text{Base}^i$$

Exemples:

$$\begin{aligned} 541_{10} &= (5 \times 10^2) + (4 \times 10^1) + (1 \times 10^0) \\ 541_8 &= (5 \times 8^2) + (4 \times 8^1) + (1 \times 8^0) \end{aligned}$$

Nombre de représentations possibles:

$$\text{Range} = \text{Base}^{(N_b \text{ de Digits})}$$

Exemples:

Base 10, 2 digits \triangleright 100 nombres différents (0 : 99)

Base 2, 16 digits \triangleright 65536 nombres différents

Base 2, 32 digits \triangleright 4294967296 nombres différents

Pour les ordinateurs:
(nb de cases mémoires)

- 1 K (Kilo) = 2^{10}
- 1 M (Mega) = 2^{20}
- 1 G (Giga) = 2^{30}
- 1 T (Tera) = 2^{40}

8 bits = 1 Octet [Byte]

SYSTÈMES DE NOMBRE

CONVERSION DES NOMBRES ENTIERS

Conversion base $n \rightarrow$ base 10

▷ Ex. 1 : Convertir 13754_8 en Base 10

$$\begin{array}{r} 1 \\ \times 8 \\ \hline 8 + 3 = 11 \\ \times 8 \\ \hline 88 + 7 = 95 \\ \times 8 \\ \hline 760 + 5 = 765 \\ \times 8 \\ \hline 6120 + 4 = 6124 \end{array}$$

Ou en appliquant directement

$$\text{Valeur}_{\text{décimale}} = \sum_{i=0}^{n_b-1} \text{Digit}_i \times \text{Base}^i$$

$$\begin{aligned} 13754_8 &= (1 \times 8^4) + (3 \times 8^3) + (7 \times 8^2) + (5 \times 8^1) + (4 \times 8^0) \\ &= (1 \times 4096) + (3 \times 512) + (7 \times 64) + (5 \times 8) + (4 \times 1) \\ &= 6124 \end{aligned}$$

SYSTÈMES DE NOMBRE

CONVERSION DES NOMBRES ENTIERS

Conversion base 10 → base n

MÉTHODE DES POIDS

▷ Ex. 1 : Convertir 6124 en Base 5

Les poids associés à chaque digit sont :

15625	3125	625	125	25	5	1
-------	------	-----	-----	----	---	---

$$1 \times 3125 = 3125 \quad \text{Reste } 2999$$

$$4 \times 625 = 2500 \quad \text{Reste } 499$$

$$3 \times 125 = 375 \quad \text{Reste } 124$$

$$4 \times 25 = 100 \quad \text{Reste } 24$$

$$4 \times 5 = 20 \quad \text{Reste } 4$$

$$4 \times 1 = 4 \quad \text{Reste } 0$$

$$\boxed{6124_{10} = 143444_5}$$

▷ Ex. 2 : Convertir 114 en Base 2

128	64	32	16	8	4	2	1
-----	----	----	----	---	---	---	---

$$1 \times 64 = 64 \quad \text{Reste } 50$$

$$1 \times 32 = 32 \quad \text{Reste } 18$$

$$1 \times 16 = 16 \quad \text{Reste } 2$$

$$0 \times 8 = 0 \quad \text{Reste } 2$$

$$0 \times 4 = 0 \quad \text{Reste } 2$$

$$1 \times 2 = 2 \quad \text{Reste } 0$$

$$0 \times 1 = 0 \quad \text{Reste } 0$$

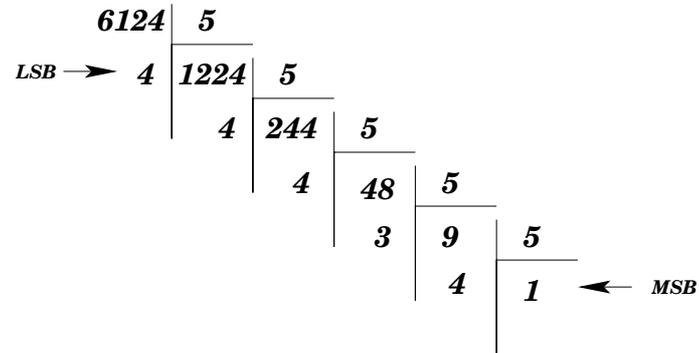
$$\boxed{114_{10} = 1110010_2}$$

SYSTEMES DE NOMBRE

CONVERSION DES NOMBRES ENTIERS

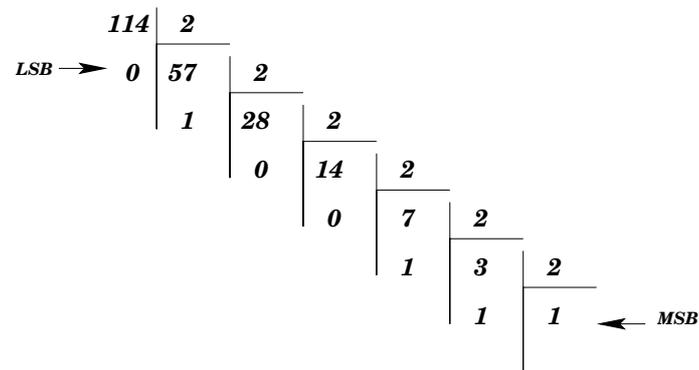
MÉTHODE DU RESTE DIVISÉ

▷ Ex. 1 : Convertir 6124 en Base 5



$$6124_{10} = 143444_5$$

▷ Ex. 2 : Convertir 114 en Base 2



$$114_{10} = 1110010_2$$

SYSTÈMES DE NOMBRE

CONVERSION DES NOMBRES ENTIERS

▷ Ex. 3 : Convertir 8151_{10} en Base 16

[valeur permise : 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F]

$$\begin{array}{r|l} 8151 & 16 \\ \hline \text{LSB} \rightarrow 7 & 509 \\ & 13 \\ & 31 \\ & 15 \\ & 1 \leftarrow \text{MSB} \end{array}$$

$$8151_{10} = 1FD7_{16}$$

SYSTÈMES DE NOMBRE

CONVERSION ENTRE BASE

Conversion Binaire ◀▶ Octal ◀▶ Hexa

- ▷ Ex. 1 : Convertir 11010111011000_2 en Base 16
(cad Base 2^1 en Base 2^4)

```
110101110110002
0011 0101 1101 10002
 3 5 D 816
```

- ▷ Ex. 2 : Convertir 11010111011000_2 en Base 8
(cad Base 2^1 en Base 2^3)

```
110101110110002
011 010 111 011 0002
 3 2 7 3 08
```

Cela marche dans le sens inverse

- ▷ Ex. 3 : Convertir 275331_8 en Base 2
(cad Base 2^3 en Base 2^1)

```
2753318
010 111 101 011 011 0012
```

SYSTÈMES DE NOMBRE

REPRÉSENTATION DES NOMBRES FRACTIONNAIRES

Nombres Fractionnaires

▷ Ex.: Comment représenter 0.1_{10} ou $1/10$ en Binaires ?

MÉTHODE DES POIDS

$$\text{Valeur}_{\text{décimale}} = \sum_{i=-m}^{n-1} \text{Digit}_i \times \text{Base}^i$$

Les poids associés à chaque bit (après la virgule) sont

$.B_1$	B_2	B_3	B_4	B_5
\swarrow	\swarrow	\downarrow	\swarrow	\swarrow
$1/2$	$1/4$	$1/8$	$1/16$	$1/32$
$0 \times (1/2) = 0.0$				Reste 0.1
$0 \times (1/4) = 0.0$				Reste 0.1
$0 \times (1/8) = 0.0$				Reste 0.1
$1 \times (1/16) = 0.0625$				Reste 0.0375
$1 \times (1/32) = 0.03125$				Reste 0.00625
$0 \times (1/64) = 0$				Reste 0.00625
				...

$0.1_{10} = 0.000110\dots_2$

Remarque

La fraction décimale 0.1_{10} ne peut être représenté parfaitement en notation binaire

La fraction $(1/3)$ non plus en base décimale ...

SYSTÈMES DE NOMBRE

REPRÉSENTATION DES NOMBRES FRACTIONNAIRES

MÉTHODE DU RESTE DIVISÉ

▷ Ex. : Comment représenter 0.1_{10} ou $1/10$ en Binaire ?

$$\begin{array}{r}
 \text{MSB} \\
 \downarrow \\
 0.1 \times 2 = 0.2 \\
 \leftarrow \uparrow \\
 .2 \times 2 = 0.4 \\
 \leftarrow \uparrow \\
 .4 \times 2 = 0.8 \\
 \leftarrow \uparrow \\
 .8 \times 2 = 1.6 \\
 \leftarrow \uparrow \\
 .6 \times 2 = 1.2 \\
 \leftarrow \uparrow \\
 .2 \times 2 = 0.4 \\
 \leftarrow \uparrow \\
 \text{LSB}
 \end{array}$$

...

$$0.1_{10} = 0.0001100110011\dots_2$$

▷ Ex. : Comment représenter 0.375_{10} en Binaire ?

$$\begin{array}{r}
 \text{MSB} \\
 \downarrow \\
 0.375 \times 2 = 0.75 \\
 \leftarrow \uparrow \\
 .75 \times 2 = 1.5 \\
 \leftarrow \uparrow \\
 .5 \times 2 = 1.0 \\
 \leftarrow \uparrow \\
 \text{LSB}
 \end{array}$$

$$0.375_{10} = 0.011\dots_2$$

SYSTÈMES DE NOMBRE

REPRÉSENTATION DES NOMBRES FRACTIONNAIRES

▷ Ex. : Comment représenter 114.1_{10} en Binaire ?

On a vu que $114_{10} = 1110010_2$,

et que $0.1_{10} = 0.0001\overline{10011}_2$

Donc

$$114.1_{10} = 1110010.0001\overline{10011}_2$$

▷ Ex. : Comment représenter 527.375_{10} en Binaire ?

On a vu que $527_{10} = 100001111_2$,

et que $0.375_{10} = 0.011_2$

Donc

$$527.375_{10} = 100001111.011_2$$

Cela Marche pour toute les bases ...

▷ Ex. : Comment représenter 527.375_{10} en Base 8 ?

On a vu que $527_{10} = 1017_8$,

Comment représenter 0.375_{10} en Base 8

$$0.375 \times 8 = 3.0$$

$$\begin{array}{c}
 \downarrow \quad \quad \quad \uparrow \\
 (.0 \times 8 = 0)
 \end{array}$$

...

Donc

$$527.375_{10} = 1017.3_8$$

SYSTÈMES DE NOMBRE

CONVERSION DES NOMBRES FRACTIONNAIRES

Remarque

1390_{10} est 10 fois plus grand que 139.0_{10}

De même,

11001.110_2 est 2 fois plus grand que 1100.1110_2

Mais,

11001.110_2 est 2 fois moins grand que 110011.10_2

Bouger le point d'un espace vers la droite
multiplie le chiffre par sa base

Bouger le point d'un espace vers la gauche
divise le chiffre par sa base

Conversion Binaire ◀▶ Octal ◀▶ Hexa

▷ Ex. : Comment convertir $0.00011001100110011\dots_2$
en Base 16 ?

$0.0001\ 1001\ 1001\ 1001\ \dots_2$

$0.1\ 9\ 9\ 9\ \dots_{16}$

Nota :

-1- Pour une autre conversion, on utilise la base 10
comme base intermédiaire

-2- Quand une représentation ou une conversion est
nécessaire et faite par un ordinateur, une troncation ou
un arrondi est effectué

SYSTÈMES DE NOMBRE

TRONCATURE

Exemple de Problèmes liés à la troncature

Défaillance du missile US patriote

Le 25/02/1991, durant la guerre du golf, un missile patriote US à Dharan manque un scud Iraquien et frappe une caserne US tuant 28 soldats



► Cause de ce problème : erreur d'estimation du temps écoulé depuis la mise en tension de l'ordinateur du missile (100 heures), causée par une simple erreur d'affectation.

Tps de l'horloge interne est mesuré en dixième de secondes puis multiplié par la valeur $1/10 = 1/2^4 + 1/2^5 + 1/2^8 + 1/2^9 + \dots = (0.000110011001\dots)_2$ (stocké dans un registre de 24 bits).

► introduit une erreur de $\epsilon = (0. - 23 \times 0 - 11001100\dots)_2$ ou $(0.000000095)_{10}$. Soit une erreur totale $= \epsilon \times 100 \times 60 \times 60 \times 10 = 0.34$ secondes.