Le binaire et l'hexadécimal

Laurent Debize

BTS SIO

Mathématiques appliquées à l'informatique

Généralités

Comment dénombrer un troupeau ?

- Sur ses doigts...
- Un bâton par tête...
- Entailles sur un os
- I, II, III, IV, V, VI, VII...

Origine de notre notation 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 :

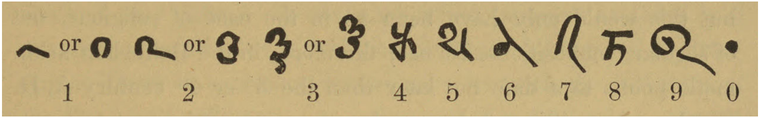
- Apparition en Inde au VIème siècle
- Arrivée en Europe par le moyen-orient au Xème siècle
- Leonardo Fibonacci propage ces notions (début XIIIème)

Avantages ?

- Numérotation de position

Généralités

Nombres utilisés dans le manuscrit de Bakhshali. Il est daté entre le IIe siècle avant n.è. et le IIe siècle de n.è. :



Numération de position, système décimal

Rappels sur les propriétés des puissances :

Quels que soient les réels x, y et les entiers a et b.

- $x^0 = 1$
- $x^1 = x$
- $x^{a+b} = x^a x^b$
- $x^{-a} = \frac{1}{x^a}$, pour $x \neq 0$
- $x^{a-b} = \frac{x^a}{x^b}$, pour $x \neq 0$
- $x^{ab} = (x^a)^b$
- $\left(\frac{x}{y}\right)^a = \frac{x^a}{y^a}$, pour $y \neq 0$

Numération de position, système décimal

La façon que nous avons d'écrire les nombres (la numération) est une numération de position.

- Elle utilise les 10 chiffres : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 (d'où l'appellation « système décimal »).
- Suivant leur position dans le nombre, les chiffres n'ont pas le même poids :

Par exemple, dans le nombre 789 : 9 est le chiffre des unités, 8 celui des dizaines, et 7 celui des centaines.
789 = 7 centaines + 8 dizaines + 9 unités

$$789 = 7 \times 100 + 8 \times 10 + 9 \times 1$$

ou, en utilisant les puissances de 10 :

$$789 = 7 \times 10^2 + 8 \times 10^1 + 9 \times 10^0$$

Numération de position, système décimal

Définition

En arithmétique, la **base** désigne la valeur dont les puissances successives interviennent dans le calcul des nombres.

Dans l'exemple précédent :

$$789 = 7 \times 10^2 + 8 \times 10^1 + 9 \times 10^0$$

La base était 10.

Retenons :

L'écriture que nous utilisons habituellement est appelée **écriture en base 10** ou **écriture décimale**.

D'autres systèmes de numération de position

Système binaire

- C'est la numération utilisée dans les ordinateurs.
- Deux symboles uniquement : 0 et 1.
- Chaque chiffre s'appelle un **bit**
- Un groupement de 8 bits s'appelle un **octet** (Byte en anglais)
- Les nombres ne s'écrivent qu'avec des 0 et des 1
- 1 Mb $\neq$ 1MB

Conversion binaire $\rightarrow$ décimal

- Exemple : $N = (11011)_2$

Chiffres du nombre en binaire	1	1	0	1	1
Poids	2^4	2^3	2^2	2^1	2^0

- Calculs :

$$\begin{aligned} N &= 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \\ &= 16 + 8 + 0 + 2 + 1 \\ &= 27 \end{aligned}$$

- Donc $N = (11011)_2 = (27)_{10}$

Exercice 1

Convertir les nombres suivants :

$(11)_2 =$

$(110)_2 =$

$(101010)_2 =$

Moralité

Le monde se divise en 10 catégories :

- Ceux qui comprennent le binaire
- Et ceux qui ne le comprennent pas

Conversion décimal → binaire

Il existe deux méthodes :

- Méthode intuitive
- Méthode algorithmique

Méthode intuitive

Prenons $n = (90)_{10}$
Utilisons un tableau contenant les puissances de 2 :

128	64	32	16	8	4	2	1
0	1	0	1	1	0	1	0

La plus grande puissance de 2 inférieure à 90 est 64.
Il reste 90-64 = 26.
La plus grande puissance de 2 inférieure à 26 est 16.
Il reste 26-16 = 10.
La plus grande puissance de 2 inférieure à 10 est 8.
Il reste 10-8 = 2.
La plus grande puissance de 2 inférieure à 2 est 2.
Il reste 2-2 = 0. Terminé !

Donc $(90)_{10} = (1011010)_2$

Méthode algorithmique

Comment faisait-on des divisions à l'école primaire ?

Division euclidienne

$$\begin{array}{r|l} 694 & 10 \\ \cline{2-2} 4 & 69 \end{array}$$

Dans cette écriture : 694 s'appelle le **dividende** ; 10 est le **diviseur** ; 69 est le **quotient** et 4 est le **reste**.

On peut établir sans difficulté les deux propriétés suivantes :

$$694 = 10 \times 69 + 4 \quad \text{et} \quad 0 \leq 4 < 10$$

$$\mathit{dividende} = \mathit{diviseur} \times \mathit{quotient} + \mathit{reste} \quad \text{et} \quad 0 \leq \mathit{reste} < \mathit{diviseur}$$

Ce type de division s'appelle une **division euclidienne**.

Le mot « euclidienne » vient du nom du mathématicien grec de l'Antiquité : Euclide.

Notez bien que dans ce type de division :

- n'interviennent que des nombres entiers (et jamais de décimaux) ;
- il est indispensable de bien avoir la relation : $0 \leq \mathit{reste} < \mathit{diviseur}$

Méthode algorithmique

Répétons l'opération sur le quotient jusqu'à trouver un quotient strictement inférieur à 10.

$$
\begin{array}{l|l|l}
694 & 10 & \\
\cline{2-2}
4 & 69 & 10 \\
\cline{3-3}
 & 9 & 6 \\
 & &
\end{array}
$$

Lire le dernier quotient et les restes du bas vers le haut : 6 9 4

Méthode algorithmique

Le principe est le même pour convertir en base 2, sauf qu'il faut faire des divisions successives par 2 au lieu de 10.

90	2	
		2

Exercice 2

Convertir les nombres suivants en binaire (utilisez une méthode différente pour chaque calcul) :

$(72)_{10} =$

$(578)_{10} =$

D'autres systèmes de numération de position

Système binaire

- Exemple : 100110010 ; 1101101 ; etc.
- Le nombre 42 (écriture décimale) va s'écrire : 101010 en binaire
- Conséquence : écriture plus longue

D'autres systèmes de numération de position

Système hexadécimal

- Ecriture binaire trop longue $\Rightarrow$ comment la réduire ?
- Utilisation de la base 16
- Il faut 16 symboles : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F
- où A=10, B=11, . . . , F=15
- C'est le système utilisé pour écrire l'adresse et le contenu des mémoires
- On l'utilise aussi pour coder les couleurs

Conversion binaire $\leftrightarrow$ hexadécimal

- Deux écritures très liées
- En effet, $16 = 2^4$
- Correspondance entre écriture hexadécimale et écriture binaire :

Hexadécimal	Binaire	Hexadécimal	Binaire
0	0000	8	1000
1	0001	9	1001
2	0010	A	1010
3	0011	B	1011
4	0100	C	1100
5	0101	D	1101
6	0110	E	1110
7	0111	F	1111

Conversion binaire $\leftrightarrow$ hexadécimal

Conversion hexadécimal $\rightarrow$ binaire :

Remplacer le symbole hexadécimal par son équivalent binaire d'après le tableau précédent

Exemple

$$(5D3)_{16} = (0101\,1101\,0011)_2$$

Pour donner l'écriture binaire, le premier zéro n'est pas nécessaire, on écrira donc :

$$(5D3)_{16} = (101\,1101\,0011)_2$$

Conversion binaire $\leftrightarrow$ hexadécimal

Conversion binaire $\rightarrow$ hexadécimal :

- Découper l'écriture binaire en « paquets » de 4 chiffres en partant de la droite
- Remplacer chaque paquet par le symbole hexadécimal correspondant

Exemple

$$\begin{array}{rcccccc} (010111001110101)_2 & = & 010 & 1110 & 0111 & 0101 \\ & = & 0010 & 1110 & 0111 & 0101 \\ & = & 2 & E & 7 & 5 \\ & = & (2E75)_{16} \end{array}$$

Exercice 3

Convertir en hexadécimal :
$(1101011)_2 =$

$(10000)_2 =$

convertir en binaire :
$(3E9)_{16} =$

Conversion hexadécimal $\rightarrow$ décimal

- Même principe mais en base 16
- Rappel de la valeur des symboles

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

- Exemple : écrire la valeur décimale de $N = (4A7E)_{16}$

Symboles en base 16	4	A	7	E
poids	16^3	16^2	16^1	16^0

- Calculs :

$$\begin{aligned} N &= 4 \times 16^3 + 10 \times 16^2 + 7 \times 16^1 + 14 \times 16^0 \\ &= 16384 + 2560 + 112 + 14 \\ &= 19070 \end{aligned}$$

- Donc $N = (4A7E)_{16} = (19070)_{10}$

Exercice 4

Convertir en décimal :
$(B7D)_{16} =$

Conversion décimal → hexadécimal

Même principe, mais en faisant des divisions successives par 16. On s'arrête quand le quotient est strictement inférieur à 16

$$\begin{array}{r|l|l}
847 & 16 & \\
\cline{2-2}
15 & 52 & 16 \\
\cline{3-3}
 & 4 & 3 \\
\end{array}$$

$(847)_{10} = (34F)_{16}$

On vérifie ?

$(34F)_{16} = 3 \times 16^2 + 4 \times 16^1 + 15 \times 16^0 = 768 + 64 + 15 = 847$

Exercice 5

Convertir en hexadécimal :

$(6348)_{10} =$

$(255)_{10} =$

$(32)_{10} =$

Les opérations en binaire et en hexadécimal

- L'addition
- La multiplication

Les opérations se font de la même façon qu'en base 10 en tenant compte des retenues

Addition

Table d'addition en base 2

$+$	0	1
0	0	1
1	1	10

Calculer $1101 + 110$

Retenues		1			
		1	1	0	1
	$+$		1	1	0
	1	0	0	1	1

Exercice 6

calculer :

$(1011)_2 + (11001)_2 =$

$(1111\,1111)_2 + (1)_2 =$

Dépassement d'entier (overflow)

Imaginons que notre processeur fait ses calculs sur 8 bits.
Le résultat de l'opération $(1111\,1111)_2 + (1)_2$ pourra-t-il être contenu ?

Multiplication

Table de multiplication en base 2

$\times$	0	1
0	0	0
1	0	1

On pose les multiplications comme en base 10, en tenant compte des retenues

Exemple : calculons 1101×110

$$\begin{array}{ccccccc}
 & & & 1 & 1 & 0 & 1 \\
 & & \times & & 1 & 1 & 0 \\
\hline
 & & & 0 & 0 & 0 & 0 \\
 & & 1 & 1 & 0 & 1 & - \\
 & 1 & 1 & 0 & 1 & - & - \\
\hline
1 & 0 & 0 & 1 & 1 & 1 & 0
\end{array}$$

Exercice 7

Calculer $(11010)_2 \times (1011)_2$

$$
\begin{array}{rccccc}
 & 1 & 1 & 0 & 1 & 0 \\
\times & & 1 & 0 & 1 & 1 \\
\hline
\\
\\
\hline
\end{array}
$$

Les opérations en hexadécimal

Même principe, mais en base 16

Une addition

$(C4D)_{16} + (85)_{16}$

Retenues			1	
		C	4	D
	$+$		8	5
		C	D	2

$(D+5)_{16} = (18)_{10} = (1 \times 16^1 + 2 \times 16^0)_{10} = (12)_{16}$: on pose 2 et on retient 1

$1 + 4 + 8 = (13)_{10} = (D)_{16}$

Finalement : $C4D + 85 = CD2$

Exercice 8

Calculer :

$(3E5)_{16} + (2D)_{16} =$

$(FF)_{16} + (1)_{16} =$

Les opérations en hexadécimal

Une multiplication

$(C4D)_{16} \times (35)_{16}$

		C	4	D
	×		3	5
	3	D	8	1
2	4	E	7	-
2	8	B	F	1

$5 \times D = (65)_{10} = (4 \times 16 + 1)_{10} = (41)_{16}$, on pose 1 on retient 4
$5 \times 4 = (20)_{10}$ sans oublier la retenue :
$(20)_{10} + (4)_{10} = (24)_{10} = 1 \times 16 + 8 = (18)_{16}$ On pose 8 et on retient 1
$5 \times C = (60)_{10} = (3 \times 16 + 12)_{10} = (3C)_{16}$ plus la retenue de 1.
etc.

Exercice 9

Calculer :

$(54C)_{16} \times (10)_{16} =$

$(3E5)_{16} \times (2D)_{16} =$

Le traitement des nombres négatifs

Principe :

Le premier bit stocke le signe :

- 0 : +
- 1 : -

Le traitement des nombres négatifs

Deux méthodes pour représenter des nombres négatifs en informatique

- Le complément à 1
- Le complément à 2

Le complément à 1

Principe

Inverser chaque bit composant la valeur binaire
On code en général les nombres sur 4 bits, 8 bits ou plus

Exemple

Le nombre 7 se code sur 8 bits par : 0000 0111
Son complément à 1 (codage sur 4 bits) sera : 1111 1000

Problème

Le nombre 0 se code de deux façons différentes : 0000 et 1111 !
Il faut réaliser 2 tests pour savoir si le résultat d'une opération est nul

Le complément à 2

Reprenons l'exemple précédent :

Le nombre 7 se code : 0111
Son complément à 1 est : 1000
Ajoutons 1, on obtient : 1001
On appelle ce nombre le complément à 2 de 7 (sur 4 bits)
C'est la façon de représenter le nombre – 7.

Méthode

Codage sur 4 bits de 7	0111
Complément à 1 ($0 \rightleftarrows 1$)	1000
On ajoute 1	$+1$
Complément à 2 $=$ codage sur 4 bits de – 7	1001

Le complément à 2

Vérifions que $7 + (-7) = 0$

Retenues		1	1	1	
		0	1	1	1
	+	1	0	0	1
		0	0	0	0

(On laisse tomber la dernière retenue)

Soustraction

Principe

Il suffit d'ajouter l'opposé, comme en décimal

Exemple : calculons $1101 - 110$

Cherchons sur 4 bits l'opposé de 110
Complément à 1 sur 4 bits : 1001
Complément à 2 : $1001 + 1 = 1010$
Donc : $1101 - 110 = 1101 + 1010 = 0111$
(En laissant tomber la dernière retenue)

Exercice 10

Convertir en base 2 sur 8 bits les nombres suivants :

- $(-56)_{10}$
- $(-128)_{10}$

Déterminer l'intervalle de valeurs codables sur un octet.

Les entiers relatifs en informatique

Le type int

- Sur 32 bits
- De -2^{31} à $2^{31} - 1$
- De $-2\,147\,483\,648$ à $2\,147\,483\,647$

Le type long int

- Sur 64 bits
- De -2^{63} à $2^{63} - 1$
- De $-9\,223\,372\,036\,854\,775\,808$ à $9\,223\,372\,036\,854\,775\,807$

Les entiers naturels en informatique

Le type unsigned int

- Sur 32 bits
- De 0 à $2^{32} - 1$
- De 0 à 4 294 967 295

Le type unsigned long int

- Sur 64 bits
- De 0 à $2^{64} - 1$
- De 0 à 18 446 744 073 709 551 615

Techniques de conversion des nombres réels

Exemple

$$0,275 = 2 \times 0,1 + 7 \times 0,01 + 5 \times 0,001$$

Ou encore :

$$0,275 = 2 \times 10^{-1} + 7 \times 10^{-2} + 5 \times 10^{-3}$$

Même principe avec les bases de 2 ou 16

Conversion binaire → décimal

Exemple : Soit $X = (0,1101)_2$ écrit en base 2

Chiffres	0,	1	1	0	1
poids	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}

Partie entière de X : $0 \times 2^0 = 0$
Partie fractionnaire de X :
$$1 \times 2^{-1} + 1 \times 2^{-2} + 0 \times 2^{-3} + 1 \times 2^{-4} = \frac{1}{2} + \frac{1}{4} + \frac{1}{16} = 0,8125$$
Rappel : pour tout $a \neq 0$: $a^{-n} = \dfrac{1}{a^n}$
Donc $X = (0,8125)_{10}$

Conversion hexadécimal → décimal

En base 16 : technique similaire, mais avec des puissances de 16

Exemple : Soit $Y = (0, A78E)_{16}$ écrit en base 16

Chiffres	0,	A	7	8	E
poids	16^0	16^{-1}	16^{-2}	16^{-3}	16^{-4}

Partie entière de Y : $0 \times 16^0 = 0$
Partie fractionnaire de Y :
$A \times 16^{-1} + 7 \times 16^{-2} + 8 \times 16^{-3} + E \times 16^{-4}$ **Rappel :** $A = 10$ et $E = 14$ donc
Partie fractionnaire de Y : $\frac{10}{16} + \frac{7}{16^2} + \frac{8}{16^3} + \frac{14}{16^4} \approx 0,654510498...$
Finalement $Y \approx (0,654510498...)_{10}$
Uniquement une valeur **approchée**

Exercice 11

Convertir en base 10 les nombres suivants :

- $(0,1011)_2$
- $(0,5FC4)_{16}$

Conversion décimal → binaire

Exemple : Prenons $N = (0,375)_{10}$ et convertissons-le en base 2

Notons l'écriture en base de N de la façon suivante :

$$N = (0, a_{-1}a_{-2}a_{-3} \dots a_{-n})_2$$

Les chiffres $a_{-1}a_{-2}a_{-3} \dots a_{-n}$ sont tous égaux à 0 ou 1. Ce sont les chiffres qui composent l'écriture en base 2 de la partie fractionnaire de N.

On peut donc écrire :

$$N = (0,375)_{10} = \frac{a_{-1}}{2} + \frac{a_{-2}}{2^2} + \frac{a_{-3}}{2^3} + \dots + \frac{a_{-n}}{2^n}$$

$$\downarrow \times 2$$

$$2 \times (0,375)_{10} = a_{-1} + \frac{a_{-2}}{2} + \frac{a_{-3}}{2^2} + \dots + \frac{a_{-n}}{2^{n-1}}$$

a_{-1} est donc la partie entière de $(0,375)_{10} \times 2$

Conversion décimal → binaire

On avait :

$$2 \times (0,375)_{10} = a_{-1} + \frac{a_{-2}}{2} + \frac{a_{-3}}{2^2} + \ldots + \frac{a_{-n}}{2^{n-1}}$$

Pour obtenir le chiffre suivant, on calcule :

$$2 \times (0,375)_{10} - a_{-1} = \frac{a_{-2}}{2} + \frac{a_{-3}}{2^2} + \ldots + \frac{a_{-n}}{2^{n-1}}$$

$$\downarrow \times 2$$

$$2 \times (2 \times (0,375)_{10} - a_{-1}) = a_{-2} + \frac{a_{-3}}{2} + \ldots + \frac{a_{-n}}{2^{n-2}}$$

a_{-2} est la partie entière de ce nombre

Et ainsi de suite : on s'arrête quand on obtient 0 où que l'on a atteint la précision souhaitée.

Conversion décimal → binaire

Règle pratique

- On multiplie N par 2.
- La partie entière du résultat donne le premier chiffre de l'écriture en base 2.
- On soustrait à $2N$ ce premier chiffre et on multiplie par 2.
- La partie entière du résultat fournit le second chiffre de l'écriture binaire.
- On recommence jusqu'à obtenir 0 ou la précision souhaitée.

Notre exemple

$0,375 \times 2 = 0,750 \Rightarrow a_{-1} = 0$

$0,750 - 0 = 0,750$

$0,750 \times 2 = 1,500 \Rightarrow a_{-2} = 1$

$1,500 - 1 = 0,500$

$0,500 \times 2 = 1,000 \Rightarrow a_{-3} = 1$

$1,000 - 1 = 0$ STOP !

D'où : $(0,375)_{10} = (0,011)_2$

Exercice 12

Convertir en base 2 les nombres suivants :

- $(0,625)_{10}$
- $(0,3)_{10}$

Votre ordinateur est-il précis ?

Exercice :

Nous voulons étudier le comportement de la suite (u_n) définie pour tout entier naturel n par :

$$\begin{cases} u_0 = 0,1 \\ u_{n+1} = 11 \times u_n - 1 \end{cases}$$

- Calculer **à la main** u_1.
- A votre avis, comment pourrait-on exprimer le terme général u_n pour tout entier naturel n ?
- Ouvrez votre tableur préféré !
- Dans une case, entrez 0,1.
- Dans la case en dessous, entrez la formule pour calculer u_1
- Étendez la formule sur une vingtaine de cases.
- Que se passe-t-il ?
- Pour comprendre le phénomène, convertissez 0,1 en base 2.

De la précision des conversions...

En général, il faudra se contenter d'une valeur approchée
Prenons par exemple $(0,837)_{10}$

Que ce soit en binaire...

$$
\begin{aligned}
0,837 \times 2 &= 1,674 \Rightarrow a_{-1} = 1\\
0,674 \times 2 &= 1,348 \Rightarrow a_{-2} = 1\\
0,348 \times 2 &= 0,696 \Rightarrow a_{-3} = 0\\
0,696 \times 2 &= 1,392 \Rightarrow a_{-4} = 1\\
0,392 \times 2 &= 0,784 \Rightarrow a_{-5} = 0\\
0,784 \times 2 &= 1,568 \Rightarrow a_{-6} = 1
\end{aligned}
$$

etc.

D'où $(0,837)_{10} \approx (0,110101)_2$

ou en hexadécimal

$$
\begin{aligned}
0,837 \times 16 &= 13,392 \Rightarrow a_{-1} = D\\
0,392 \times 16 &= 6,272 \Rightarrow a_{-2} = 6\\
0,272 \times 16 &= 4,352 \Rightarrow a_{-3} = 4\\
0,352 \times 16 &= 5,632 \Rightarrow a_{-4} = 5\\
0,632 \times 16 &= 10,112 \Rightarrow a_{-5} = A\\
0,112 \times 16 &= 1,792 \Rightarrow a_{-6} = 1
\end{aligned}
$$

etc.

D'où $(0,837)_{10} \approx (0,D645A1)_{16}$

(Si on a besoin d'une précision de 6 chiffres après la virgule)

Conversion décimal $\rightarrow$ hexadécimal

Même principe, mais en utilisant des multiplications par ... 16 !

Reprenons $N = (0,375)_{10}$

$0,375 \times 16 = 6,000 \Rightarrow a_{-1} = 6$

$6,000 - 6 = 0$ STOP !

Sur cet exemple, le calcul se termine très vite et donc :
$(0,375)_{10} = (0,6)_{16}$

Généralisation

Comment convertir un nombre comme $(51,625)_{10}$ en base 2, ou $(101,11)_2$ en base 10 ?

- Prendre la partie entière
  Ex : $(51)_{10}$ et $(101)_2$
- Convertir ce nombre en utilisant la technique de conversion de nombres entiers correspondante
  Ex : $(51)_{10} = (110011)_2$ et $(101)_2 = (5)_{10}$
- Prendre ensuite la partie après la virgule
  Ex : $(0,625)_{10}$ et $(0,11)_2$
- Convertir ce nombre en utilisant la technique de conversion que nous venons de voir.
  Ex : $(0,625)_{10} = (0,101)_2$ et $(0,11)_2 = (0,75)_{10}$
- Ajouter les résultats des deux conversions pour obtenir le résultat final
  Ex : $(51,625)_{10} = (110011,101)_2$ et $(101,11)_2 = (5,75)_{10}$

Représentation des réels dans un ordinateur

La virgule flottante

Prenons le nombre suivant : 52,7653

On peut l'écrire de cette façon : $52,7653 = +527653 \times 10^{-4}$

Le principe de la virgule flottante, c'est d'écrire un nombre de cette façon :

$$x = (-1)^{signe} \times mantisse \times 10^{exposant}$$

où *signe* vaut 0 ou 1, et *mantisse* et *exposant* sont entiers.

A la différence près qu'en machine :

- on n'utilise pas la base 10, mais la base 2
- l'exposant doit être décalé pour pouvoir représenter à la fois les grands nombres que le petits.

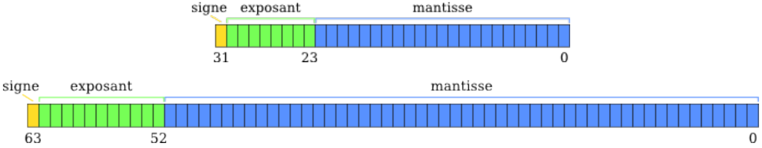
Représentation des réels dans un ordinateur

La norme IEEE 754

La norme IEEE 754 fixe deux types de formats :

Précision	Encodage	*Signe*	*Exposant*	*Mantisse*	Valeur d'un nombre	Précision	Chiffres significatifs
Simple précision	32 bits	1 bit	8 bits	23 bits	$(-1)^S \times M \times 2^{(E-127)}$	24 bits	environ 7
Double précision	64 bits	1 bit	11 bits	52 bits	$(-1)^S \times M \times 2^{(E-1023)}$	53 bits	environ 16

Le tableau ci-dessus indique les bits représentés. Le premier bit de la mantisse d'un nombre normalisé étant toujours 1, il n'est représenté dans aucun de ces deux formats : on parle de bit implicite. Pour ces deux formats, les précisions sont donc respectivement de 24 et de 53 bits.



Exercice 13

1. Convertir en base 10 :

$$(111,101)_2 \qquad (35C,38A)_{16}$$

2. Convertir en binaire :

$$(121,25)_{10}$$

3. Convertir en hexadécimal :

$$(33,242)_{10} \qquad (101,1011\,1100\,1100)_2$$

Arrondi

En base 10

Considérons le nombre $N = 432,617$.
L'arrondi (au plus près) à $\mathbf{10}$ près de N est 430.
L'arrondi (au plus près) à $\mathbf{1}$ près de N est 433.
L'arrondi (par excès) à $\mathbf{0{,}1}$ près de N est 432,7.
L'arrondi (par défaut) à $\mathbf{0{,}01}$ près de N est 432,61.

Arrondi

Généralisation

- L'arrondi **par défaut** correspond à une **troncature** : on coupe l'écriture en enlevant des décimales.
- L'arrondi **par excès** consiste à **ajouter 1** à la dernière décimale conservée.
- Quelle que soit la base, l'arrondi **au plus près** d'un nombre réel x à une certaine précision est le nombre le plus proche de x tel que tous les chiffres allant au-delà de cette précision soient nuls.

Remarque

Par convention, lorsqu'il existe deux nombres possibles, l'arrondi est alors le plus grand.

Arrondi

Exemple 1 :

Considérons $N = (101,10011)_2$

- L'arrondi (au plus près) à $(10)_2$ de N est $(110)_2$
- L'arrondi (au plus près) à $(0,1)_2$ de N est $(101,1)_2$

Exemple 2 :

Considérons $M = (B82A,7AB)_{16}$

- L'arrondi (au plus près) à $(100)_{16}$ de M est $(B800)_{16}$
- L'arrondi (au plus près) à $(0,1)_{16}$ de M est $(B82A,8)_{16}$

Exercice 14

- Quel est l'arrondi (au plus près) de $(1101011)_2$ à $(100)_2$ près ?
- Quel est l'arrondi de $(10,011)_2$ à $(0,1)_2$ près :
  - Par défaut
  - Par excès
  - Au plus près
- Quel est l'arrondi de $(A,BB)_{16}$ à $(0,1)_{16}$ près ?
  - Par défaut
  - Par excès
  - Au plus près

Exercice 15

- Convertir $(0,333)_{10}$ en base 2 en arrondissant à 2^{-3} près :
  - Par défaut
  - Par excès
  - Au plus près
- Convertir $(0,333)_{10}$ en base 16 en arrondissant à 16^{-3} près :
  - Par défaut
  - Par excès
  - Au plus près