Le binaire et l'hexadécimal

Laurent Debize

TIIS 2
Outils Mathématiques pour l'informatique

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 1

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 $a \neq 0$: $a^{-n} = \frac{1}{a^n}$
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 : $\dfrac{10}{16} + \dfrac{7}{16^2} + \dfrac{8}{16^3} + \dfrac{14}{16^4} \approx 0,654510498...$
Finalement $Y \approx (0,654510498...)_{10}$
Uniquement une valeur **approchée**

Exercice 2

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 $\rightarrow$ 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 3

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 :

$$\left\{ \begin{array}{r} u_0 = 0,1 \\ u_{n+1} = 11 \times u_n - 1 \end{array} \right.$$

- 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...

$0,837 \times 2 = 1,674 \Rightarrow a_{-1} = 1$
$0,674 \times 2 = 1,348 \Rightarrow a_{-2} = 1$
$0,348 \times 2 = 0,696 \Rightarrow a_{-3} = 0$
$0,696 \times 2 = 1,392 \Rightarrow a_{-4} = 1$
$0,392 \times 2 = 0,784 \Rightarrow a_{-5} = 0$
$0,784 \times 2 = 1,568 \Rightarrow a_{-6} = 1$
etc.
D'où $(0,837)_{10} \approx (0,110101)_2$

ou en hexadécimal

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

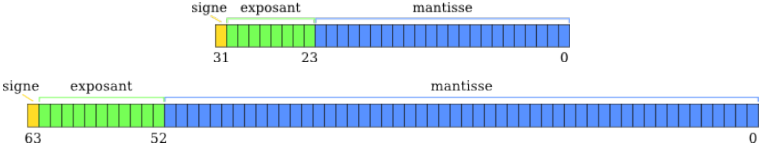
Représentation des réels dans un ordinateur

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 4

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 **10** près de N est 430.
L'arrondi (au plus près) à **1** près de N est 433.
L'arrondi (par excès) à **0,1** près de N est 432,7.
L'arrondi (par défaut) à **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 5

- 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 6

- 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