Représentation des nombres réels en binaire

Site Internet : **www.gecif.net**	Type de document : **Cours**	Intercalaire :	Date :

I – Rappel sur l'écriture d'un nombre décimal

Un nombre est appelé nombre décimal lorsqu'il contient une partie entière et **une partie décimale non nulle**. La partie décimale d'un nombre décimal possède un nombre fini de chiffres (contrairement à certains nombres réels).

La virgule flottante est une méthode d'écriture de nombres réels fréquemment utilisée dans les ordinateurs ou sur une calculatrice. Elle consiste à représenter un nombre réel par :

- un signe **S** (égal à -1 ou 1),
- une mantisse **M** (un nombre décimal positif)
- et un exposant **E** (un nombre entier relatif)

Un tel triplet représente le nombre décimal sous la forme $\mathbf{S \times M \times b^E}$ où b est la base du système de numération (b=10 pour l'écriture en décimal). Exemple : le nombre décimal 57,418 peut s'écrire $1\times 57{,}418\times 10^0$, ou alors $1\times 5{,}7418\times 10^1$, ou alors $1\times 5741{,}8\times 10^{-2}$, ou encore $1\times 0{,}057418\times 10^3$ et de bien d'autres manières. On voit qu'en fonction de l'ordre de grandeur de la mantisse il suffit d'adapter l'exposant pour obtenir le même nombre au final. Dans toutes ces écritures équivalentes seul le signe est constant (il vaut toujours 1 dans cet exemple puisque le nombre est positif), et la différence entre ces écritures est la position de la virgule dans la mantisse : virgule flottante

On appelle **écriture scientifique** l'écriture du nombre sous la forme $\mathbf{S \times M \times 10^E}$ avec **une mantisse M comprise dans l'intervalle [1 ;10[**, soit un nombre décimal avec un seul chiffre non nul pour sa partie entière.

L'écriture scientifique du nombre 57,418 est $1\times 5{,}7418\times 10^1$: S=1 M=5,7418 et E=1, soit 5,7418.10

Comment s'écrivent les nombres décimaux suivants en écriture scientifiques ?

26937,1495 = ... - 0,0000081275463 = ...

II – Ecriture en binaire d'un nombre décimal

Pour passer de la représentation binaire à l'écriture en décimal (en base 10) d'un nombre décimal (un nombre à virgule) il suffit d'utiliser les puissances de 2 avec des puissances de 2 positives pour les bits de la partie entière (comme pour les nombres entiers écrits en binaire naturel), et **des puissances de 2 négatives pour les bits de la partie décimale**. Exemples :

$101{,}1101_{(2)} = 1.2^2+0.2^1+1.2^0+1.2^{-1}+1.2^{-2}+0.2^{-3}+1.2^{-4} = 2^2+2^0+2^{-1}+2^{-2}+2^{-4} = 6{,}8125_{(10)}$

$1100{,}001_{(2)} = 2^3+2^2+2^{-3} = 12 + 2^{-3} = 12{,}125_{(10)}$: seuls les bits de valeurs 1 sont significatifs

Comment s'écrivent en décimal les nombres réels suivants exprimés ici en binaire ?

$1110011{,}0010110011_{(2)}$ = ...

$1100101{,}00101110_{(2)}$ = ...

$1111{,}1111_{(2)}$ = ...

$100000{,}000000001_{(2)}$ = ...

Voyons maintenant l'opération réciproque : comment représenter le nombre décimal $5{,}1875_{(10)}$ en binaire ?
Il nous faut déjà représenter la partie entière 5 en binaire. Pour cela aucun problème : $5_{(10)} = 101_{(2)}$
Comment représenter maintenant la partie décimale ",1875" ? Voici la méthode :

- on multiplie 0,1875 par 2 : 0,1875 x 2 = 0,375. On obtient 0,375 que l'on écrira **0** + 0,375
- on multiplie 0,375 par 2 : 0,375 x 2 = 0,75. On obtient 0,75 que l'on écrira **0** + 0,75
- on multiplie 0,75 par 2 : 0,75 x 2 = 1,5. On obtient 1,5 que l'on écrira **1** + 0,5 (quand le résultat de la multiplication par 2 est supérieur à 1, on garde uniquement la partie décimale)
- on multiplie 0,5 par 2 : 0,5 x 2 = 1,0. On obtient 1,0 que l'on écrira **1** + 0,0 (la partie décimale est égale à 0, on arrête le processus)

On obtient une succession de "a + 0,b" ("**0** + 0,375", "**0** + 0,75", "**1** + 0,5" et "**1** + 0,0"). Il suffit maintenant de "prendre" tous les "a" (dans l'ordre de leur obtention) afin d'obtenir la partie décimale de notre nombre : 0011
Nous avons donc $101{,}0011_{(2)}$ qui est la représentation binaire du nombre décimal $5{,}1875_{(10)}$

Comment s'écrivent en binaire les nombres réels suivants exprimés ici en décimal ?

$14{,}863_{[10]}$ = ...

$205{,}49786_{[10]}$ = ...

III – La notation scientifique en binaire

En base dix, il est possible d'écrire les très grands nombres et les très petits nombres grâce aux "puissances de 10" (exemples "$6{,}02.10^{23}$" ou "$6{,}67.10^{-11}$"). Il est possible de faire exactement la même chose avec une représentation binaire, puisque nous sommes en base 2, nous utiliserons des "puissances de 2" à la place des "puissances de 10" (exemple "$101{,}1101.2^{10}$"). La mantisse comme l'exposant sont écrit ici en base 2.

Pour passer d'une écriture sans "puissance de deux" à une écriture avec "puissance de deux", il suffit de décaler la virgule : "$1101{,}1001 = 1{,}1011001.2^{11}$" pour passer de "1101,1001" à "1,1011001" nous avons décalé la virgule de 3 rangs vers la gauche d'où le "2^{11}" (attention de ne pas oublier que nous travaillons en base 2 le "11" correspond bien à un décalage de 3 rangs de la virgule : $3_{[10]} = 11_{[2]}$). Si l'on désire décaler la virgule vers la gauche, il va être nécessaire d'utiliser des "puissances de deux négatives" "$0{,}0110 = 1{,}10.2^{-10}$", nous décalons la virgule de 2 rangs vers la droite, d'où l'exposant en binaire qui vaut "-10".

Comment s'écrivent en notation scientifique binaire les nombres réels suivants exprimés ici en décimal ?

$14{,}863_{[10]}$ = ...

$205{,}49786_{[10]}$ = ...

IV – La norme IEEE-754

A la fin des années 1970, chaque ordinateur avait sa propre représentation pour les nombres à virgule flottante. Il y a donc eu la nécessité de normaliser le codage des nombres flottants. La norme IEEE 754 est la norme la plus employée pour la représentation des nombres à virgule flottante dans le domaine informatique. La première version de cette norme date de 1985. L'avantage de l'écriture en virgule flottante est de permettre de représenter des nombres très grands et très petits sans s'encombrer de zéros. Caractéristiques de la norme IEEE 754 :

- le signe + est représenté par 0 et le signe – par 1 (donc sur 1 bit)
- l'exposant est un entier relatif et il est établi de manière à ce que la mantisse soit de la forme « 1, xxxx »
- la mantisse appartient à l'intervalle [1; 2[

La norme IEEE 754 existe en 2 formats : simple précision sur 32 bits (Bit du signe 1 bit + Exposant 8 bits + Mantisse 23 bits) et double précision sur 64 bits (Bit du signe 1 bit + Exposant 11 bits + Mantisse 52 bits).

Exemple de codage en binaire avec la norme IEEE 754 en simple précision (32 bits soit 4 octets) :

L'exposant E sur 8 bits peut être positif ou négatif : de -126 à +127 (avec les valeurs extrêmes -127 et +128 réservées pour les cas particuliers : l'infini, NaN, etc.). L'exposant (entier signé) n'est pas représenté en complément à 2 mais sous la forme « biaisée » : Sur 8 bits, la valeur biaisée 127 correspond au zéro (non-biaisé). Les valeurs biaisées qui vont de 1 à 126 correspondent aux valeurs non-biaisées de -126 à -1 et les valeurs biaisées qui vont de 128 à 254 correspondent aux valeurs non-biaisées de 1 à 126. On effectue donc la somme E + 127 afin de coder l'exposant E en binaire. La mantisse M (sur 23 bits) va de 1 à $2\text{-}2^{-23}$

Codons par exemple le nombre $-6{,}\ 625$:

- écrivons le nombre en binaire : $6{,}\ 625_{[10]} = 110{,}\ 1010_{[2]}$
- décalons la virgule pour que la mantisse soit de forme 1,xxxx : $110{,}\ 1010 = 1{,}\ 101010 \times 2^2$
- la mantisse (partie décimale de 1,xxxx) sur 23 bits vaut : 10101000000000000000000
- ajoutons 127 à l'exposant puis codons-le en binaire sur 8 bits : $127 + 2 = 129_{[10]} = 10000001_{[2]}$
- et voici la représentation sur 32 bits du nombre $-6{,}\ 625$:1 10000001 10101000000000000000000
- soit en hexadécimal sur 4 octets : C0 D4 00 00

Comment s'écrivent en format IEEE 754 simple précision les nombres réels suivants exprimés ici en décimal ?

$14{,}863_{[10]}$ = ...

....

$-97{,}1685_{[10]}$ = ...

....