Représentation approximative des réels - corrigé

Exercice 0 Donner la représentation en base 10 des nombres **signés** suivant $(00100010, 00001000)_2$ et $(11000001, 1110011)_2$, où la représentation en complément à deux a été utilisé pour encoder la partie entière.

Corrigé

- $(00100010, 00001000)_2 = 1 \times 2^5 + 1 \times 2^1 + 1 \times 2^{-5} = 34,03125$
- $(11000001, 1110011)_2$:
  - décodage de la partie entière m : le bits de poids le plus fort de $m = (11000001)_2$ est à 1. Il s'agit donc de l'encodage en complément à deux d'un nombre négatif. Soit $|m| = 2^8 - (1 \times 2^7 + 1 \times 2^6 + 1 \times 2^1) = 63$. Donc $m = -63$.
  - décodage de la partie fractionnaire d : $d = (0,1110011)_2 = 1 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3} + 1 \times 2^{-6} + 1 \times 2^{-7}$

Exercice 1 Effectue la division de $(0101)_2$ par $(1000)_2$ puis de $(10)_2$ par $(1010)_2$ en la posant **dans le système binaire**, pour obtenir respectivement la représentation de $\frac{5}{8}$ et de $\frac{2}{10}$ en base 2. Que remarques-tu dans le second cas ?

Corrigé

$$\begin{array}{ccccccc|cccc}
 & 1 & 0 & 1 & 0 & & & 1 & 0 & 0 & 0 \\
\cline{8-11}
- & 1 & 0 & 0 & 0 & & & 0, & 1 & 0 & 1 \\
\cline{1-5}
 & 0 & 0 & 1 & 0 & 0 & 0 & & & & \\
 & & - & 1 & 0 & 0 & 0 & & & & \\
\cline{4-7}
 & & & 0 & 0 & 0 & 0 & & & &
\end{array}$$

On obtient $\frac{5}{8} = (0,101)_2$.

Vérification : $(0,101)_2 = 2^{-1} + 2^{-3} = \frac{1}{2} + \frac{1}{8} = \frac{5}{8}$.

$$\begin{array}{cccccccc|cccc}
 & 1 & {}^{1}0 & {}^{1}0 & {}^{1}0 & 0 & & & 1 & 0 & 1 & 0 \\
\cline{9-12}
- & {}^{1} & {}^{1}1 & {}^{1}0 & 1 & 0 & & & 0, & 0 & 0 & 1 & 1 \\
\cline{2-6}
 & 0 & 0 & 1 & 1 & {}^{1}0 & 0 & & & & & \\
 & & - & 1 & {}^{1}0 & 1 & 0 & & & & & \\
\cline{3-7}
 & & & 0 & 0 & 1 & 0 & 0 & & & &
\end{array}$$

On obtient $\frac{2}{10} = (0,0011001100110011...)_2$.

Le nombre $\frac{2}{10}$ n'est donc pas représentable exactement en base 2.

Rappel : les retenues de la 1ère ligne d'une addition posée valent 2, celles de la seconde ligne valent 1.

Exercice 2 Donne l'encodage en virgule fixe de 122,6875 sur 16 bits (8 bits pour la partie entière et 8 bits pour la partie fractionnaire).

Corrigé

- encodage de la partie entière : $122 = 64 + 32 + 16 + 8 + 2 = (01111010)_2$
- encodage de la partie fractionnaire : On utilise l'algorithme des multiplications successives par deux de la partie fractionnaire.

$$\begin{array}{rl}
0,6875 \times 2 = 1,375 & 0,1 \\
0,375 \times 2 = 0,750 & 0,10 \\
0,75 \times 2 = 1,5 & 0,101 \\
0,5 \times 2 = 1,0 & 0,1011
\end{array}$$

La partie fractionnaire est nulle, l'algorithme se termine donc.

On obtient donc, sur 16 bits : $122,6875 = (01111010, 10110000)_2$

Exercice 3 Quelles sont les parties entières représentables en virgule fixe sur 32 *bits* ? Quelle est la plus petite partie fractionnaire non nulle représentable en virgule fixe sur 32 bits ?

Corrigé
En virgule fixe sur 32bits, la partie entière **signée** est représentée sur 16 *bits*, ce qui correspond aux entiers de -2^{15} à $2^{15}-1$ ou de -32768 à 32767 (on utilise le complément à deux).

La plus petite partie fractionnaire non nulle représentable sur 16 bits est $(0,0000000000000001)_2 = 2^{-16}$.

On considère l'écriture permettant de représenter un nombre en **virgule flottante** suivante :

$$s \times (1+m) \times 2^e$$

- s représente le signe ;
- m représente la mantisse (partie fractionnaire) ;
- e représente l'exposant.

Exercice 4 Écris $122,6875$ en virgule flottante.

Corrigé
D'après l'exercice 2, on a

$$\begin{aligned} 122,6875 &= (01111010,10110000)_2 \\ &= (1,11101010110000)_2 \times 2^6 \\ &= +(1+0,11101010110000)_2 \times 2^6 \end{aligned}$$

Ce qui donnerait l'encodage suivant sur 32 *bits*, avec un exposant décalé de 127 ($6+127=133=(10000101)_2$) :

s	$e+127$								m																						
0	1	0	0	0	0	1	0	1	1	1	1	0	1	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Exercice 5 On donne la représentation binaire en virgule flottante sur 32 bit suivante :

s	$e+127$								m																						
0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	0	0	1	1	1	1	1	1	0	1	1	0	1	1

- le bit de poids le plus fort représente s (0 pour les positifs, 1 pour les négatifs) ;
- les 8 bits suivants représentent l'exposant décalé $e+127$;
- les 23 derniers bits représentent la mantisse m.

Donne l'écriture décimale du nombre correspondant ?

Corrigé

- signe : le bit de poids le plus fort est à 0, c'est donc un nombre positif ;
- l'exposant décalé $e+127$ vaut $(10000000)_2 = 128$, donc $e = 128-127 = 1$;
- la mantisse m vaut $2^{-1}+2^{-4}+2^{-7}+2^{-12}+2^{-13}+2^{-14}+2^{-15}+2^{-16}+2^{-17}+2^{-19}+2^{-20}+2^{-22}+2^{-23} = 0,57079637050628662109375$.

On obtient donc $+(1+0,5707963705062866)*2^1 = 3.14159274101257324218 75$