

TD d'architecture des ordinateurs I

Arithmétique entière, bases de numération

Exercice 1 : Conversions de bases

1. Écrire les nombres suivants dans chacune des bases 2, 8, 10 et 16 : $7F_{(16)}$, $11000001_{(2)}$, $1000001_{(2)}$, $13_{(10)}$, $755_{(8)}$, $1100000011011110_{(2)}$.

Réponse _____

- $1111111_{(2)}$, $177_{(8)}$, $127_{(10)}$, $7F_{(16)}$
- $11000001_{(2)}$, $301_{(8)}$, $193_{(10)}$, $C1_{(16)}$
- $1000001_{(2)}$, $101_{(8)}$, $65_{(10)}$, $41_{(16)}$
- $1101_{(2)}$, $15_{(8)}$, $13_{(10)}$, $0D_{(16)}$
- $111101101_{(2)}$, $755_{(8)}$, $493_{(10)}$, $1ED_{(16)}$
- $1100000011011110_{(2)}$, $140336_{(8)}$, $49374_{(10)}$, $C0DE_{(16)}$



Exercice 2 : Arithmétique

1. Calculer les tables d'addition et de multiplication pour les bases 2 et 8.

Réponse _____

+	0	1
0	0	1
1	1	10

×	0	1
0	0	0
1	0	1

+	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7	10
2	2	3	4	5	6	7	10	11
3	3	4	5	6	7	10	11	12
4	4	5	6	7	10	11	12	13
5	5	6	7	10	11	12	13	14
6	6	7	10	11	12	13	14	15
7	7	10	11	12	13	14	15	16

×	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7
2	0	2	4	6	10	12	14	16
3	0	3	6	11	14	17	22	25
4	0	4	10	14	20	24	30	34
5	0	5	12	17	24	31	36	43
6	0	6	14	22	30	36	44	52
7	0	7	16	25	34	43	52	61



2. Calculez les opérations suivantes : $1111001_{(2)} + 100101_{(2)}$, $101011_{(2)} * 11011_{(2)}$, $53262_{(8)} - 4323_{(8)}$, la division euclidienne de $3046_{(8)}$ par $56_{(8)}$.

Réponse _____

- $1111001_{(2)} + 100101_{(2)} = 10011110_{(2)}$
- $101011_{(2)} * 11011_{(2)} = 10010001001_{(2)}$
- $53262_{(8)} - 4323_{(8)} = 46737_{(8)}$
- $3046_{(8)} = 42_{(8)} * 56_{(8)} + 12_{(8)}$



Exercice 3 : Entiers relatifs en machine

1. Calculer le complémenté à 9 de 2006 sur 4 chiffres décimaux.

2. Calculer ensuite le complémenté à 10 de 2006 sur 4 chiffres décimaux. Vérifier que leur somme est bien égale à zéro (en complément à 10 sur 4 chiffres).

Remarque 1 Le nom « complément à 10 (en base 10) » est un abus de langage (compris de tous les informaticiens). Il s'agit en fait du « complément à 9 plus 1 ». Il en est de même en base 2, le complément à 2 en base 2 (complément à 1 plus 1) n'a rien à voir avec le complément à 2 en base 3.

Remarque 2 On codera les nombres en complément à 2 sur 16 bits dans toute la suite.

3. Quel est l'intervalle des entiers relatifs représentables dans ce système de codage ?

Réponse _____
De $-2^{15} = -32768$ à $2^{15} - 1 = 32767$.



4. Exprimez en base 10 les nombres dont le codage en complément à 2 sur 16 bits est le suivant : 0110110000011011, 1011011010110011.

Réponse _____
27675 et -18765 .



5. Montrer que le complémenté à 2 d'un entier n (sur 16 bits) est égal à $2^{16} - n$.

Réponse _____
Par définition le complémenté à 2 de n est égal à $1111111111111111 - n + 1 = 2^{16} - n$.



Remarque 3 Travailler avec des nombres en complément à 2 sur 16 bits revient à poser implicitement que $2^{16} = 0$: c'est à dire que nous effectuons les opérations usuelles modulo 2^{16} .

6. Dédurre de la question précédente que le complément à 2 est involutif (le complémenté à 2 du complémenté à 2 est l'entier relatif de départ).

Réponse _____
Le complémenté à 2 est donc $2^{16} - (2^{16} - n) = n$.



Remarque 4 Autrement dit, l'opposé de l'opposé de l'entier relatif n est bien égal à n , propriété bien connue de la soustraction.

7. Comment exprimer les conditions de dépassement de capacité lors d'une addition avec des entiers relatifs en complément à 2 (sur 16 bits) ?

Réponse _____
Pas de dépassement si les signes sont différents. Dépassement si les signes sont égaux et avec un changement de signe dans le résultat. (Raisonnement en représentant les nombres sur 17 bits et en laissant tomber le bit de poids fort, on a alors une représentation modulo 2^{16} avec des nombres dans l'intervalle $[-2^{15}, 2^{15} - 1]$.)



8. Montrez qu'on peut aussi exprimer cette condition comme suit :

Il y a dépassement de capacité lors d'une addition de deux entiers relatifs en complément à deux si et seulement si les deux dernières retenues (de poids le plus élevé) sont différentes.

- Addition de 2 nombres positif : les deux bits de poids fort sont 0, donc la retenue sortante est forcément 0, on aura dépassement de capacité ssi le bit de poids fort de la somme est 1 (nombre négatif), ce qui revient à avoir une retenue d'avant dernier rang égale à 1 ;
- addition de 2 nombres négatifs : les deux bits de poids fort sont 1, donc la retenue sortante est forcément 1, on aura dépassement de capacité ssi le bit de poids fort de la somme est 0 (nombre positif), ce qui revient à avoir une retenue d'avant dernier rang égale à 0 ;
- addition de deux nombres de signes différents : un des bits de poids fort est 0 et l'autre 1, on n'a donc que deux cas : il n'y a pas de retenue d'avant dernier rang, et donc pas de retenue finale ou il y a une retenue d'avant dernier rang et elle est propagée. Dans tous les cas les deux retenues ont la même valeur et il n'y a jamais de dépassement de capacité.



1 Rappels de cours

Les machines les plus répandues utilisent une version (en général réduite) de la norme IEEE 754. Vous trouverez ici une version simplifiée de cette norme (en simple précision) :

signe 1 bit	Exposant 8 bits $E = e + 127$	mantisse M de 23 bits $m = 1, \underbrace{m_0 m_1 \dots m_{22}}_{M_{(2)}}$	valeur
s	11111111	00000000000000000000000	$(-1)^s \times \infty$
s	E	M	$(-1)^s \times m \times 2^e$
?	00000000	00000000000000000000000	0
?	11111111	M	NaN
?	00000000	M	$(-1)^s \times 0, M \times 2^e$

1.1 L'addition en virgule flottante

On suit *méticuleusement* les étapes suivantes :

- Ramener les deux nombres au même exposant.
- Restaurer le bit de poids fort.
- Effectuer l'addition ou la soustraction des valeurs absolues comme pour des entiers.
- Renormaliser le résultat (arrondi, bit de poids fort, exposant).

1.2 La multiplication en virgule flottante

On suit *méticuleusement* les étapes suivantes :

- Calculer le signe et la somme des exposants (attention au biais).
- Restaurer le bit de poids fort.
- Effectuer la multiplication des valeurs absolues comme pour des entiers.
- Eventuellement, arrondir, ajuster l'exposant et renormaliser.

Exercice 4 : Questions

1. Quelle est la représentation du plus petit réel représentable en simple précision ?

Réponse _____
 Le "piège" est qu'il faut prendre le plus "grand" des négatifs.
 Le codage est $1|111\ 1111\ 0|111\ 1111\ 1111\ 1111\ 1111\ 1111 \Rightarrow FF7FFFFF$
 et la valeur $-(2 - 2^{-23}) \times 2^{127}$ soit approximativement $-3,4028235 \cdot 10^{38}$ (ou, à l'anglo-saxonne, $-3.4028235E38$).



2. Quelle est la représentation de $-\infty$ en simple précision?

Réponse _____
 Le codage est $1|111\ 1111\ 1|000\ 0000\ 0000\ 0000\ 0000\ 0000 \Rightarrow FF800000$



3. Quelle est la représentation du plus petit réel strictement positif représentable en simple précision?

Réponse _____
 Le codage normalisé est $0|00000001|000000000000000000000000 \Rightarrow 00800000_{(16)}$
 et la valeur 2^{-126} soit approximativement $1,1754944 \cdot 10^{-38}$. En codage dénormalisé (exposant nul, le biais est alors de 126 au lieu de 127), mentionné ici pour la culture, on a aussi $0|000\ 0000\ 0|000\ 0000\ 0000\ 0000\ 0000\ 0001 \Rightarrow 00000001_{(16)}$
 et la valeur 2^{-149} soit approximativement $1,4012985 \cdot 10^{-45}$.



4. Quelle est la représentation de 1 en simple précision?

Réponse _____
 Le codage est $0|011\ 1111\ 1|000\ 0000\ 0000\ 0000\ 0000\ 0000 \Rightarrow 3F800000_{(16)}$



5. Quelle est la représentation du plus grand réel représentable en simple précision strictement inférieur à 1?

Réponse _____
 Le codage normalisé est $0|011\ 1111\ 0|111\ 1111\ 1111\ 1111\ 1111\ 1111 \Rightarrow 3F7FFFFF_{(16)}$ et la valeur $0,99999994$.



6. Quelle est la plus grande valeur représentable en simple précision qui peut être ajoutée à 1 telle qu'après arrondi de la somme, le résultat est toujours égal à 1?

Réponse _____
 Ce cas correspond à l'addition

$$\begin{array}{r} 1,000\ 0000\ 0000\ 0000\ 0000\ 0000 \\ + \quad 0,000\ 0000\ 0000\ 0000\ 0000\ 0111\ 1111\ 1111\ 1111\ 1111\ 1 \end{array}$$

Le codage normalisé du deuxième nombre est $0|011\ 0011\ 0|111\ 1111\ 1111\ 1111\ 1111\ 1111 \Rightarrow 337FFFFF_{(16)}$ et sa valeur approximativement $5,9604641 \cdot 10^{-8}$. En fait nous avons fait l'hypothèse implicite que l'arrondi n'était pas calculé avec plus d'un bit après le 23^e bit de la mantisse. Dans la réalité, les processeurs de calcul en flottants effectuent les arrondis avec plus de bits (en nombre variable selon les processeurs) et l'arrondi aurait été quand même effectué à 1 plutôt qu'à 0.



Exercice 5 : Opérations

1. Trouver la représentation IEEE 754 simple précision de $\frac{1}{3}$, $\frac{1}{5}$ et $\frac{1}{10}$.

Réponse _____
 Comme $\frac{1}{3} = \frac{01_{(2)}}{2^2-1}, \frac{1}{3} = 0,010101\dots$. De même $\frac{1}{5} = \frac{0011_{(2)}}{2^4-1}$, donc $\frac{1}{5} =$

$0,0011\underline{0011}\dots$ et $\frac{1}{10} = 0,00011\underline{0011}\dots$. Les étudiants ne sont pas censés connaître cette méthode, il faut en tout cas bien leur faire remarquer dans l'extraction de la représentation en base 2 que l'on retrouve les mêmes restes au bout d'un moment et donc que le développement est périodique à partir de ce point.

A ce point, leur expliquer que pour des raisons d'arrondi on choisit pour $\frac{1}{5}$ la mantisse $1,1001 \dots 1001 \mathbf{101}$ et non $1,1001 \dots 1001 \mathbf{100}$ (c'est important pour la suite). La mantisse de $\frac{1}{10}$ est identique. Pour $\frac{1}{3}$, il faut également arrondir le 23^e chiffre à 1 et non 0.

Les solutions sont donc dans l'ordre :

- $0|011\ 1110\ 1|010\ 1010\ 1010\ 1010\ 1011 \Rightarrow 3EAAAAAB_{(16)}$
- $0|011\ 1110\ 0|100\ 1100\ 1100\ 1100\ 1100\ 1101 \Rightarrow 3E4CCCCC_{(16)}$
- $0|011\ 1101\ 1|100\ 1100\ 1100\ 1100\ 1100\ 1101 \Rightarrow 3DCCCCCC_{(16)}$

Remarquer que contrairement à la base 10, les développements de $\frac{1}{5}$ et $\frac{1}{10}$ sont infinis (mais quand même périodiques).



2. Un nombre flottant est connu avec 6 chiffres binaires significatifs. À combien de chiffres décimaux correspondent-ils environ ? À combien de chiffres décimaux correspondent donc les $24 = 23 + 1$ bits de la mantisse ?

Réponse _____

Leur expliquer ou leur faire trouver que pour obtenir 6 chiffres significatifs en base 2, il en faut environ $6 \cdot \frac{\ln 2}{\ln 10} = 6 \times 0,3 = 1,8$ chiffres en base 10. 24 bits correspondent donc à environ $24 \cdot \frac{\ln 2}{\ln 10} = 7,2$ chiffres.



3. Trouver la représentation IEEE 754 simple précision de 10 et $-142,625$.

Réponse _____

Les solutions sont dans l'ordre (observer que $-142,625 = -\frac{1141}{8}$) :

- $0|100\ 0001\ 0|010\ 0000\ 0000\ 0000\ 0000\ 0000 \Rightarrow 41200000_{(16)}$
- $1|100\ 0011\ 0|000\ 1110\ 1010\ 0000\ 0000\ 0000 \Rightarrow C30EA000_{(16)}$



4. Faire l'addition IEEE 754 de $\frac{1}{10}$ et $\frac{1}{10}$. Que se passe-t-il ?

Réponse _____

Ils doivent vraiment faire l'addition, pour se faire la main. Bien leur faire remarquer que comme le développement est périodique, ils n'ont que quelques chiffres à calculer.

La mantisse de $\frac{1}{10}$ est $1,1001 \dots 1001101$. Donc pour $2 \cdot \frac{1}{10}$: $1,1001 \dots 10011010$. Résultat, on trouve bien $\frac{1}{5}$: on ajoute juste 1 à l'exposant.



5. Calculer $\frac{1}{3} - \frac{1}{5}$ en IEEE 754 simple précision. Reconnaître le résultat.

Réponse _____

Bien faire utiliser la périodicité pour éviter de passer des heures sur ce calcul. On trouve

$$|0|011\ 1110\ 0|000\ 1000\ 1000\ 1000\ 1000\ 1001|,$$

ce qui est bien $\frac{2}{15}$ (avec l'arrondi correct).



6. Calculer $11 \times -142,625$ et $10 \times \frac{1}{10}$ en IEEE 754 simple précision. Que remarque-t-on ? Réponse _____

Pénible mais faisable... mêmes remarques méthodologiques que précédemment. Faire observer les problèmes de chiffres significatifs sur le second exemple.



7. Calculer $-142,625 \times \frac{1}{3}$.