Exercice 0 : Addition & Soustraction binaire

```
   111
   110
+ 1110
= 10100
```

```
    1111
    1111
  10100110
- 00111100
= 01101010
```

Exercice 1 : Nombre relatifs sur machine

- 0. Intervalle des entiers naturels en BCD sur 16 bits :
- 2^{16} éventualités de codage : N = 0 à 2^{16} -1 (0 à +65.535)
- 1. Intervalle des entiers relatifs Complément à 2 (C2) sur 16 bits :
- 2^{16} éventualités de codage : N = -2^{15} à 2^{15} -1 (-32.768 à +32.767)

Exemple sur 3 bits

+3 = 011	-2^2 à 2^2-1
+2 = 010	(-4 à +3)
+1 = 001	
0 = 000	
-1 = 111	
-2 = 110	
-3 = 101	
-4 = 100	

Exercice 1 : Nombre relatifs sur machine

- 2-a. Conversion Hexa et Binaire de 2010_{10}
- 2-b. Conversion en Décimal de $7DA_H$

2010	0
1005	1
502	0
251	1
125	1
62	0
31	1
15	1
7	1
3	1
1	

Binaire : 111 1101 1010

Hexa : 7 D A

Conversion Héxa >> Décimal :

$$7DA = 7x16^2 + Dx16^1 + Ax16^0$$
$$= 7x16^2 + 13x16^1 + 10x16^0$$
$$= 7x256 + 13x16 + 10x1$$
$$= 1792 + 208 + 10$$
$$= \mathbf{2010}$$

Exercice 1 : Nombres relatifs sur machine

- 3. Conversion en Décimal des nombres suivant en C2
- 0110 1100 0001 1011
- 1011 0110 1011 0011

0110 1100 0001 1011 nombre positif

$= 2^{14} + 2^{13} + 2^{11} + 2^{10} + 2^{4} + 2^{3} + 2^{1} + 2^{0}$

$= 16\,384 + 8\,192 + 2048 + 1024 + 16 + 8 + 2 + 1$

= **27 675**

1011 0110 1011 0011 nombre négatif

On calcule sa valeur absolue

N>>C2=C1(N)+1 C2>> -N = C1(C2 - 1)

C1 (C2 + 1111 1111 1111 1111)

1011 0110 1011 0011

1111 1111 1111 1111

1011 0110 1011 0010 >> C1 >> 0100 1001 0100 1101

$= -(2^{14} + 2^{11} + 2^{8} + 2^{6} + 2^{3} + 2^{2} + 2^{0})$

$= -(16\,384 + 2048 + 256 + 64 + 8 + 4 + 1) =$ **- 18 765**

Exercice 1 : Nombres relatifs sur machine

3. Calculer en complément à 2 sur 8 bits les additions suivants :

a) 122 + (-7); b) (-111) + (-17); c) 111 + 17

Faire apparaitre toutes les retenues intermédiaires. Préciser si le résultat est correct ou s'il y a dépassement de capacité. Vérifier les propriétés suivantes de l'addition en Complément à 2 :

- ***Pas de dépassement de capacité si les signes des opérandes sont différents. Il y a dépassement si les signes des opérandes sont égaux avec un changement de signe dans le résultat.***
- ***Dépassement de capacité lors d'une addition de 2 entiers relatifs si et seulement si les deux dernières retenues (de poids le plus élevés) sont différents.***

Conversion des opérandes en Code binaire C2

(C2 sur 8 bits : -12_ à 127)

$122_{(10)} = 0111\ 1010_{(C2)}$

$-7_{(10)} = 1111\ 1001_{(C2)}$

$111_{(10)} = 0110\ 1111_{(C2)}$

$-111_{(10)} = 1001\ 0001_{(C2)}$

$17_{(10)} = 0001\ 0001_{(C2)}$

$-17_{(10)} = 1110\ 1111_{(C2)}$

Exercice 1 : Nombres relatifs sur machine

a) 122 + (-7) :

1 1111 000	(retenues)
0111 1010	(122)
1111 1001	(-7)
0111 0011	(155) ➔ 0111 0011 représente bien 115
	(2 dernières retenues à 1)

b) (-111) + (-17) :

1 1111 111	(retenues)
1001 0001	(-111)
1110 1111	(-17)
1000 0000	($-2^7 = -128$) ➔ 1000 0000 représente bien -128
	(2 dernières retenues identiques)

c) (-111) + 17 :

0 1111 111	(retenues)
0110 1111	(111)
0001 0001	(17)
1000 0000	($-2^7 = -128$) ➔ 1000 0000 représente bien -128
	(erreur ≠ 128 / 2 dernières retenues différentes)

Exercice 2 : Nombre à virgule fixe

1. Donner l'équivalent binaire (parties entière et décimale sur 8 bits) des nombres décimaux suivants :

a) 118,625 ;   b) 1/10 ;   c) 4/3.

2. Exprimez en base 10 le nombre binaire suivant : 100,1011.

a. 118,625

On divise 118 par 2 autant de fois qu'il est possible, les restes successives étant les poids binaires obtenus dans l'ordre des puissances croissantes

Partie entière

118 : 2 = 59 reste 0

59 : 2 = 29 reste 1

29 : 2 = 14 reste 1

14 : 2 = 7 reste 0

7 : 2 = 3 reste 1

3 : 2 = 1 reste 1

1 : 2 = 0 reste 1

D'où : 118 = 111 0110

Partie fractionnaire:

Binaire vers décimal :

2^{-1} = ½ = 0,5; 2^{-2} = ¼ = 0,25; … $2^{-P} = 1/(2^{P})$

Ainsi 0,101 = $1*2^{-1} + 0*2^{-2} + 1*2^{-3}$ = 0,625

Décimal vers binaire : multiplications successives par 2

0,625 x 2 = **1**,25

0,25 x 2 = **0**,5

0,5 x 2 = **1**,0

D'où 0,625 = $1*2^{-1} + 0*2^{-2} + 1*2^{-3}$ = 0,101

0111 0110,1010 0000

Exercice 2 : Nombre à virgule fixe

b. 1/10

0,1 x 2 = **0**,2
0,2 x 2 = **0**,4
0,4 x 2 = **0**,8
0,8 x 2 = **1**,6
0,6 x 2 = **1**,2
0,2 x 2 = **0**,4
0,4 x 2 = **0**,8
0,8 x 2 = **1**,6
0,6 x 2 = **1**,2
…

0000 0000,0001 1001

c. 4/3

4/3 = 1 + 1/3
Partie entière = 1
Partie fractionnaire
0,333… x 2 = **0**,666…
0,666… x 2 = **1**,333…
0,333… x 2 = **0**,666…
…

0000 0001,0101 0101

Exercice 2 : Nombre à virgule fixe

2. Le nombre binaire 101,1011 en base 10

Partie entière :

$101_{(2)} = 1*2^2 + 0*2^1 + 1*2^0 = 5_{(10)}$

Partie fractionnaire

$1011 = 1*2^{-1} + 0*2^{-2} + 1*2^{-3} + 1*2^{-4} = 0,6875_{(10)}$

…

5,6875

Exercice 3 : Nombre à virgule flottante (IEEE754)

1. Donner l'équivalent binaire en simple précision des nombres décimaux suivants :

a) 1,5 ; b) 0,5 ; c) -142,625 ; d) 10 ; e) 1/10 f) -18

2. Faire l'addition en IEEE754 simple précision de 1/10 et 1/10. Reconnaître le résultat.

3. Faire la multiplication en IEEE754 simple précision de -18 par 10.

4. alignement des mantisses : Faire l'addition en IEEE754 simple précision de 1.875 et 15.

Rappel de la norme IEEE754

Décomposition	Signe	Exposant (entier)	Mantisse
Simple précision (32 bits)	**1**	**8**	**23**
Double précision (64 bits)	1	11	52

Nombre normalisé : nombre = $(-1)^{signe} * 1,mantisse * 2^{(exposant - biais)}$

Nombre dénormalisé : nombre = $(-1)^{signe} * 0,mantisse * 2^{(exposant - biais)}$

Biais : 127 (simple précision) et 1023 (double précision)

Cas particuliers :

Signe	Exposant	Mantisse	Valeur
-	0…0	0…0	+/- 0
-	1…1	0…0	+/- ∞
-	1…1	-	NaN
-	0…0	-	Nombre dénormalisé

Exercice 3 : Nombre à virgule flottante (IEEE754)

a) 1,5 = $2^k \cdot (1, \ldots)$ = 1 x 1,5 = 2^0 x (1 + 0,5) = $2^{127-127}$ x (1 + 0,5) ($k \in \mathbf{Z}$)

s (1) = **0**

e (8) = 127 = **0111 1111**

m (23) = 0,5 = 2^{-1} = **100 0000 0000 0000 0000 0000**

sem = **0011 1111 1100 0000 0000 0000 0000 0000** = **3F C0 00 00$_{(H)}$**

b) 0,5 = $2^k \cdot (1, \ldots)$ = 0,5 x 1,0 = 2^{-1} x (1 + 0,0) = $2^{126-127}$ x (1 + 0,0)

s (1) = **0**

e (8) = 126 = **0111 1110**

m (23) = 0 = **000 0000 0000 0000 0000 0000**

sem = **0011 1111 0000 0000 0000 0000 0000 0000** = **3F 00 00 00$_{(H)}$**

c) -142,625 = $2^k \cdot (1, \ldots)$ = - 128 x 1,114 257 8125 = - 2^7 x (1 + 0,114 257 8125) = - $2^{134-127}$ x (1 + 0,114 257 8125)

s (1) = **1**

e (8) = 134 = **1000 0110**

m (23) = 0,114 257 8125 = **000 1110 1010 0000 0000 0000**

sem = **1100 0011 0000 1110 1010 0000 0000 0000** = **C3 0E A0 00$_{(H)}$**

Exercice 3 : Nombre à virgule flottante (IEEE754)

d) 10 = $2^k \cdot (1,\ldots)$ = 8 x 1,25 = 2^3 x (1 + 0,25) = $2^{130-127}$ x (1 + 0,25)

s (1) = **0**

e (8) = 130 = **1000 0010**

m (23) = 0,25 = 2^{-2} = **010 0000 0000 0000 0000 0000**

sem = **0100 0001 0010 0000 0000 0000 0000 0000 = 41 20 00 $00_{(H)}$**

e) 1/10 = $2^k \cdot (1,\ldots)$ = 0,0625 x 1,6 = 2^{-4} x (1 + 0,6) = $2^{123-127}$ x (1 + 0,6)

s (1) = **0**

e (8) = 123 = **0111 1011**

m (23) = 0,6 = **100 1100 1100 1100 1100 1100** (partie fractionnaire infinie)

sem = **0011 1101 1100 1100 1100 1100 1100 1100 = 3D CC CC $CC_{(H)}$**

f) -18 = $2^k \cdot (1,\ldots)$ = - 16 x 1,125 = - 2^4 x (1 + 0,125) = - $2^{131-127}$ x (1 + 0,125)

s (1) = **1**

e (8) = 4 % 127 = 131 = **1000 0011**

m (23) = 0,125 = 2^{-3} = **001 0000 0000 0000 0000 0000** = m_X

sem = **1100 0001 1001 0000 0000 0000 0000 0000 = C1 90 00 $00_{(H)}$**

Exercice 3 : Nombre à virgule flottante (IEEE754)

2. Faire l'addition en IEEE754 simple précision de 1/10 et 1/10. Reconnaître le résultat.

Méthode pour une addition en IEEE 754

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

1/10 = $2^k \cdot (1,\cdots)$ = 0,0625 x 1,6 = 2^{-4} x (1 + 0,6) = $2^{123-127}$ x (1 + 0,6)

s (1) = **0**

e (8) = 123 = **0111 1011**

m (23) = 0,6 = **100 1100 1100 1100 1100 1100** (partie fractionnaire infinie)

sem = **0011 1101 1100 1100 1100 1100 1100 1100 = 3D CC CC CC**$_{(H)}$

1/10 + 1/10 =

```
⇨    1,100 1100 1100 1100 1100 1100 * 2^(123-127)
   + 1,100 1100 1100 1100 1100 1100 * 2^(123-127)
   -----------------------------------------------
    11,001 1001 1001 1001 1001 1000 * 2^(123-127)   → à normaliser
     1,100 1100 1100 1100 1100 1100 * 2^(124-127)   dernier bit (0) perdu - n'importe quel arrondi)
```

soit pour le résultat : **1/10 + 1/10** =

s (1) = **0**

e (8) = 124 = **0111 1100**

m (23) = **100 1100 1100 1100 1100 1100** (partie fractionnaire infinie)

sem = **0011 1110 0100 1100 1100 1100 1100 1100 = 3E 4C CC CC**$_{(H)}$ (même mantisse que 1/10, exposant+1)

A comparer au résultat 2/10 :

2/10 = $2^k \cdot (1,\cdots)$ = 0,125 x 1,6 = 2^{-3} x (1 + 0,6) = $2^{124-127}$ x (1 + 0,6)

s (1) = **0**

e (8) = 124 = **0111 1100**

m (23) = 0,6 = **100 1100 1100 1100 1100 1100** (partie fractionnaire infinie)

sem = **0011 1110 0100 1100 1100 1100 1100 1100 = 3E 4C CC CC**$_{(H)}$ (même résultat que 1/10 + 1/10) OK

Exercice 3 : Nombre à virgule flottante (IEEE754)

3. Faire la multiplication en IEEE754 simple précision de -18 par 10.

Méthode pour une multiplication en IEEE 754

a. Calculer le signe puis la somme des exposants
b. Restaurer le bit de poids fort
c. Effectuer la multiplication des valeurs absolues comme pour les entiers
d. Eventuellement, arrondir, ajuster l'exposant et renormaliser.

X = -18 = $2^k \cdot (1, \dots)$ = - 16 x 1,125 = - 2^4 x (1 + 0,125) = - $2^{131-127}$ x (1 + 0,125)

s (1) = **1**

e (8) = 4 % 127 = 131 = **1000 0011**

m (23) = 0,125 = 2^{-3} = **001 0000 0000 0000 0000 0000** = m_X

sem = **1100 0001 1001 0000 0000 0000 0000 0000 = C1 90 00 00**$_{(H)}$

Y = 10 = $2^k \cdot (1, \dots)$ = 8 x 1,25 = 2^3 x (1 + 0,25) = $2^{130-127}$ x (1 + 0,25)

s (1) = **0**

e (8) = 3 % 127 = 130 = **1000 0010**

m (23) = 0,25 = 2^{-2} = **010 0000 0000 0000 0000 0000** = m_Y

sem = **0100 0001 0010 0000 0000 0000 0000 0000 = 41 20 00 00**$_{(H)}$

Exercice 3 : Nombre à virgule flottante (IEEE754)

Z = X x Y = (-18) x 10 =

signe : 1 (xor entre les 2 signes des 2 opérandes)

exposant : **1000 0011 + 1000 0010 - $(127)_2$ = 1000 0011 + 1000 0010 – 0111 1111**

= 1000 0011 + 1000 0010 + $(-127)_{c2}$ = 1000 0011 + 1000 0010 + 1000 0001 = 0000 0101 + 1000 0001 = 1000 0110

(= 134_{10}) = 7 + 127 (OK : 7 = 4 + 3)

mantisse : Multiplication des mantisses m_X et m_Y :

remarque : $(1 + m_X)(1 + m_Y) = 1 + m_X.m_Y + m_X + m_Y$ → $\mathbf{m_Z = m_X.m_Y + m_X + m_Y}$

```
    1,mX =         1,001  0…0
x   1,mY =       x 1,01   0…0
                 -----------
                   1001
                  0000
                 1001
                 -----------
    1,mZ =       1,01101  0…0
```

(attention à la troncature: le produit de 2 mots de 24 bits a 48 bits)
(attention à la normalisation: les retenues peuvent faire qu'à gauche de la virgule on a plus que 1,)

→ m_Z = 01101 0…0 → m_Z (23) = 011 0100 0000 0000 0000 0000

Interprétation du résultat : **Z**

s (1) : 1

e (8) : 1000 0110 → **e = 134** → **exposant = 134-127 = 7** % 127

m (23) : 011 0100 0000 0000 0000 0000 → **mantisse** = $2^{-2} + 2^{-3} + 2^{-5}$ = 0,40 625

sem = 1100 0011 0011 0100 0000 0000 0000 0000 = C3 34 00 $00_{(H)}$

Z = - 1,40 625 x 2^7 = - 1,40 625 x 128 = - 180

Vérification du résultat : **Z**

Z (exact) = - 180 = $2^k \cdot (1,\cdots)$ = - 128 x 1,40 625 = - 2^7 x (1 + 0,40 625) = - $2^{134-127}$ x (1 + 0,40 625)

s (1) = **1**

e (8) = 134 = **1000 0110** = 7 % 127

m (23) = 0,40 625 = $2^{-2} + 2^{-3} + 2^{-5}$ = **011 0100 0000 0000 0000 0000**

sem = **1100 0011 0011 0100 0000 0000 0000 0000= C3 34 00 $00_{(H)}$**

Exercice 3 : Nombre à virgule flottante (IEEE754)

4. alignement des mantisses : Faire l'addition en IEEE754 simple précision de 1.875 et 15.

X = +1,875 = $+2^k \cdot (1,\cdots)$ = +1 x 1,875 = $+2^0$ x (1 + 0,875) = $+2^{127-127}$ x (1 + 0,875)

s (1) = 0

e (8) = 127 = 0111 1111

m (23) = 0,875 = $2^{-1} + 2^{-2} + 2^{-3}$ = 111 0000 0000 0000 0000 0000 = m_X

sem = 0011 1111 1111 0000 0000 0000 0000 0000 = 3F F0 00 $00_{(H)}$

Y = 15 = $2^k \cdot (1,\cdots)$ = 8 x 1,875 = 2^3 x (1 + 0,875) = $2^{130-127}$ x (1 + 0,875)

s (1) = 0

e (8) = 130 = 1000 0010

m (23) = 0,875 = $2^{-1} + 2^{-2} + 2^{-3}$ = 111 0000 0000 0000 0000 0000 = m_Y

sem = 0100 0001 0111 0000 0000 0000 0000 0000 = 41 70 00 $00_{(H)}$

S = X + Y = (+1,875) + (15) =

```
   1,111 0000 0000 0000 0000 0000 * 2^(127-127)        (+1,875)
+  1,111 0000 0000 0000 0000 0000 * 2^(130-127)        (15)
alignement des mantisses (dénormalisation) :

   0,001 1110 0000 0000 0000 0000 * 2^(130-127)        (+1,875)
+  1,111 0000 0000 0000 0000 0000 * 2^(130-127)        (15)
--------------------------------------------------
  10,000 1110 0000 0000 0000 0000 * 2^(130-127)        Normalisation à faire
   1,000 0111 0000 0000 0000 0000 * 2^(131-127)        Normalisation
```

s (1) = 0

e (8) = 131 = 1000 0011

m (23) = 000 0111 0000 0000 0000 0000

sem = 0100 0001 1000 0111 0000 0000 0000 0000 = 41 87 00 $00_{(H)}$

Vérification : comparaison à S = X+Y = 16,875.