Architecture des Ordinateurs

TD1 : Représentation des données - Codage

ING1 Informatique

Année 2023–2024

Attention : L'ensemble des exercices ci-dessous ne seront pas tous corrigés en cours !

Exercice 1 : Entiers positifs

1) Convertir $(25)_{10}$ en octal et en hexadécimal
   $(25)_{10} = (31)_8$
   $(25)_{10} = (19)_{16}$

2) Convertir $(72)_8$ en base 10
   $(72)_8 = (58)_{10}$

3) Convertir $(A2F3)_{16}$ en base 10
   $(A2F3)_{16} = A*16^3 + 2*16^2 + F*16^1 + 3*16^0 = (41715)_{10}$

4) Soustraction : Calculer 10100110 – 00111100
   01101010

Exercice 2 : Entiers négatifs

1) Donner le Complément à 1 et le Complément à 2 de -32, sur 8 bits.
   $(32)_{10} = (00100000)_2 = 2^5$
   En complément à 1 : $(-32)_{10} = (1101\ 1111)_2$
   En complément à 2 : $(-32)_{10} = (1110\ 0000)_2$

2) Convertir -7 en Complément à 2, sur 8 bits.
   $(+7)_{10} = (0000\ 0111)_2$
   En complément à 1 : $(-7)_{10} = (1111\ 1000)_2$
   En complément à 2 : $(-7)_{10} = (1111\ 1001)_2$

3) Addition : Calculer en Complément à 2 (sur 8 bits) : 122 + (-7).
   $(122)_{10} = (0111\ 1010)_2 + (-7)_{10} = (1111\ 1001)_2 = (1\ 0111\ 0011)_2$
   la retenue en rouge n'entre pas dans la réponse finale
   => $(0111\ 0011)_2$ représente bien $(115)_{10}$

4) Addition : Calculer en Complément à 2 (sur 8 bits) : (-111) + (-17).
   $(-111)_{10} = (1001\ 0001)_2 + (-17)_{10} = (1110\ 1111)_2 = (1000\ 0000)_2 = (-128)_{10} = -2^7$
   => 2 dernières retenues sont identiques ; regarder règle par rapport à l'addition en C à 2

5) Soustraction : Calculer en Complément à 2 (sur 4 bits) : 2 - (-3). Donner le résultat en décimal.
   $(2)_{10} = (0010)_2 – (-3)_{10} = (1101)_2 = (0101)_2$ représente bien $(5)_{10}$
   $(3)_{10} = (0011)_2$
   En complément à 1 : $(-3)_{10} = (1100)_2$
   En complément à 2 : $(-3)_{10} = (1101)_2$

Exercice 3 : Nombres à virgule fixe

1) Donner l'équivalent binaire (sur 8 bits) de 18,3125.
   1) Donner l'équivalent binaire sur 8 bits de 18,3125
   0001 0010,0101 000

2) Convertir $(2607,75)_{10}$ en hexadécimal.
   partie entière : $(2607)_{10} = (A2F)_{16}$
   partie fractionnaire :
   $0.75 * 16 = 12 = C$
   $(0.75)_{10} = (0.C)_{16}$
   $(A2F.C)_{16}$

3) Convertir $(67,2)_8$ en décimal.
   $(67.2)_8 = 6*8^1 + 7*8^0 + 2*8^{-1} = (55.25)_{10}$

Exercice 4 : Nombres à virgule flottante

1) Coder en binaire et en hexadécimal les nombres décimaux suivants en utilisant le mécanisme IEEE 754 (simple précision, 32 bits) :

   a) 1,5

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

   $s\,(1) = \mathbf{0}$

   $e\,(8) = 127 = \mathbf{0111\ 1111}$

   $m\,(23) = 0.5 = 2^{-1} = \mathbf{100\ 0000\ 0000\ 0000\ 0000\ 0000}$

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

   b) −142,625

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

   $s\,(1) = \mathbf{1}$

   $e\,(8) = 134 = \mathbf{10000\ 0110}$

   $m\,(23) = 0{,}114\ 257\ 8125 = \mathbf{000\ 1110\ 1010\ 0000\ 0000\ 0000}$

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

2) $(3EE00000)_{16}$ et $(3D800000)_{16}$ sont codés suivant la norme IEEE 754. Calculer leur somme et donner le résultat en suivant la norme IEEE754.

Somme de 3EE00000 et 3D800000

Hexadécimal	3	E	E	0	0	0	0	0
Binaire	0 ⋮ 011	1110	1 ⋮ 110	0000	0000	0000	0000	0000
IEEE 774	+ Exp biaisé : 125 Exp : 125–127 = -2		Pseudo mantisse : 110 0000 0000 0000 0000 0000 Mantisse : 1, 110 0000 0000 0000 0000 0000					
	+ 1, 110 x 2^{-2} (=> 0,4375 en décimal)							

Hexadécimal	3	D	8	0	0	0	0	0
Binaire	0 ⋮ 011	1101	1 ⋮ 000	0000	0000	0000	0000	0000
IEEE 774	+ Exp biaisé : 123 Exp : 123–127 = -4		Pseudo mantisse : 000 0000 0000 0000 0000 0000 Mantisse : 1, 000 0000 0000 0000 0000 0000					
	+ 1 , 0 x 2^{-4} (=> 0,0625 en décimal)							

$$(1{,}110 \times 2^{-2}) + (1{,}0 \times 2^{-4}) = (1{,}110 \times 2^{-2}) + (0{,}010 \times 2^{-2})$$
$$= (1{,}110 + 0{,}010) \times 2^{-2} = 10{,}0 \times 2^{-2} = 1{,}0 \times 2^{-1}$$

IEEE 774	**+ 1, 0 x 2^{-1}** (=> 0, 5 en décimal)							
	+ Exp : = -1 Biaisé :-1+127 = **126**		Mantisse : 1, **0** Pseudo mantisse : 000 0000 0000 0000 0000 0000					
Binaire	0 ⋮ 011	1111	0 ⋮ 000	0000	0000	0000	0000	0000
Hexadécimal	3	F	0	0	0	0	0	0

3) Multiplier -18 par 10 en suivant la norme IEE754 (simple précision, 32 bits).

4)

-18 = $(-1)^1 \times 2^k \times (1,\ldots) = -16 \times 1{,}125 = -\ 2^4 \times (1{,}125)$

$= -\ 2^{131-127} \times (1 + 0{,}125)$

s (1) = **1**

e (8) = 131 = **1000 0011**

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

sem = **1100 0001 1001 0000 0000 0000 0000 0000 = C1 90 00 00** $_{16}$

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

s (1) = **0**

e (8) = 130 = 1**000 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** $_{16}$