Université Batna 2 **1ère année Math &INF**

Socle commun Math & INF **Module : SM1**

TD N°2

Exercice 1 :

1) Remplir le tableau suivant: (tous les nombres sont codés sur 8 bits)

Nombre	Décimal	SVA	Cà1	Cà2
N_1	(+34)	00100010	00100010	00100010
N_2	(+53)	00110101	00110101	00110101
N_3	(–28)	10011100	11100011	11100100
N_4	(–40)	10101000	11010111	11011000
N_5	(–1)	10000001	11111110	11111111

2) Calculer $N_2 + N_3$ en Cà1.

$^{1}0^{1}0\ 1\ 1^{1}0^{1}1^{1}0\ 1$

\+ 1 1 1 0 0 0 1 1

= 1 0 0 0 1 1 0 0 0

\+ 1

= 0 0 0 1 1 0 0 1

3) Calculer $N_1 + N_5$ en Cà2.

$^{11}0^{1}0^{1}1^{1}0^{1}0^{1}0\ 1\ 0$

\+ 1 1 1 1 1 1 1 1

= ~~1~~0 0 1 0 0 0 0 1

4) Calculer $(+74)_{10} + (103)_{10}$ en Cà2. Que remarquez-vous ?

$^{1}0\ 1\ 0^{1}0^{1}1^{1}0\ 1\ 0$ + 74

\+ 0 1 1 0 0 1 1 1 +103

= 1 0 1 1 0 0 0 1 ≠ –79

Il y a un dépassement de capacité puisque si on convertie les nombres en décimal on obtient un résultat faux

Exercice 2 :

1) On définit une représentation simplifiée de la virgule flottante comme suit : un nombre fractionnaire est représenté au total sur 12 bits :1 bit pour le signe, 4 bits pour l'exposant biaisé et 7 bits pour la mantisse.

- Représenter $X_1 = (+8{,}625)_{10}$ et $X_2 = (–4{,}35)_{10}$ selon cette représentation.

On a $X_1 = (+8{,}625)_{10} = (+1000{,}101)_2 = (+0{,}1000101 \times 2^{4})$.

M = 1000101, $E_{réel} = 4$

- Le nombre X_1 est positif alors **S = 0**
- **$Eb = E_{réel} + 2^{4-1} = 4 + 2^3 = 4 + 8 = (12)_{10} = (1100)_2$**
- $X_1 =$ | 0 | 1100 | 1000101 |

On a $X_2 = (-4{,}35)_{10} = (-100{,}01011001100\ldots\ldots)_2 = (-0{,}10001011001100\ldots.. \times 2^{4})$.

- **M=**10001011001100, $E_{réel} = 3$

On remarque ici que M dépasse le nombre de bits réservés à la mantisse sur cette machine (7 bits) alors on prend seulement 7 chiffres : M = 1000101. Dans ce cas on perd un peu de la précision du nombre X_2.

- Le nombre X_2 est négatif alors **S = 1**
- **$Eb = E_{réel} + 2^{4-1} = 3 + 2^3 = 3 + 8 = (11)_{10} = (1011)_2$**
- $X_2 =$ | 1 | 1011 | 1000101 |

Calculer $X_1 + X_2$ et $X_1 \div X_2$ (effectuer les opérations en binaire).

$X_1 + X_2 = (+0{,}1000101 \times 2^{4}) + (-0{,}1000101\ldots.. \times 2^{3})$.

$= (+0{,}1000101 \times 2^{4}) + (-0{,}01000101\ldots.. \times 2^{4})$

$= (+0{,}01000101 \times 2^{4})$

$= (+0{,}1000101 \times 2^{4})$

M = 1000101 (aussi on prend seulement 7 chiffres)**. S = 0** (le résultat est positif).

$Eb = E_{réel} + 2^{4-1} = 3 + 2^3 = 3 + 8 = (11)_{10} = (1011)_2$

$X_1 + X_2 =$ | 0 | 1011 | 1000101 |

$X_1 \div X_2 = (+0{,}1000101 \times 2^{4}) + (-0{,}1000101\ldots.. \times 2^{3})$.

$= (-1 \times 2^{4})$

$= (-0{,}1 \times 2^{5})$

M = 1000000, **S = 1** (le résultat est négatif), **$Eb = E_{réel} + 2^{4-1} = 2 + 2^3 = 2 + 8 = (10)_{10} = (1010)_2$**

$X_1 \div X_2 =$ | 1 | 1010 | 1000000 |

2) Soient X_3 et X_4 deux nombres codés en hexadécimal selon la norme IEEE 754 en simple précision tel que : X_3 = 2AF05000 et X_4 = 3E60D000.

- Calculer $X_3 - X_4$ et $X_3 \times X_4$. Donner le résultat selon la norme IEEE 754 en simple précision et en décimal.

Remarque : La solution de cet exercice est longue, utilisez les données suivantes pour résoudre ce problème : X_3 = C0E7A000 et X_4 = C0600000.

$X_3 = (C0E7A000)_{16} = (1100\ 0000\ 1110\ 0111\ 1010\ 0000\ 0000\ 0000)_2$

$X_3 =$ | 1 | 10000001 | 11001111010000000000000 |

SM = 1 alors X_3 est négatif.

$Eb = (10000001)_2 = (129)_{10}$ donc $E_{réel} = Eb - 127 = 129 - 127 = 2$

On obtient $X_3 = -1,1100111101 \times 2^2$

(Juste pour vérifier le résultat des opérations après) $X_3 = -111,00111101 = -(2^0 + 2^1 + 2^2 + 2^{-3} + 2^{-4} + 2^{-5} + 2^{-6} + 2^{-8}) = (-7,23828125)_{10}$

$X_4 = (C0600000)_{16} = (1100\ 0000\ 0110\ 0000\ 0000\ 0000\ 0000\ 0000)_2$

$X_4 =$ | 1 | 10000000 | 11000000000000000000000 |

SM = 1 alors X_4 est négatif.

$Eb = (10000000)_2 = (128)_{10}$ donc $E_{réel} = Eb - 127 = 128 - 127 = 1$

On obtient $X_4 = -1,11 \times 2^1$

(Juste pour vérifier le résultat des opérations après) $X_4 = -1,11 \times 2^1 = -11,1 = -(2^0 + 2^1 + 2^{-1}) = (3,5)_{10}$

$X_3 - X_4 = -1,1100111101 \times 2^2 - (-1,11 \times 2^1)$

$= -1,1100111101 \times 2^2 + 0,111 \times 2^2$

$= (-1,1100111101 + 0,111) \times 2^2$

$= -0,1110111101 \times 2^2$

$X_3 - X_4$ selon la norme IEEE 754 en simple précision (**32** bits : **1** bit pour le signe, **8** bits pour l'exposant biaisé et **23** bits pour la mantisse) :

$X_3 - X_4 = -1,110111101 \times 2^1$

M = 11011110100000000000000 ; **S = 1** (le résultat est négatif).

$Eb = E_{réel} + 2^{8-1}-1 = 1 + 2^7-1 = 1 + 128 -1 = (128)_{10} = (10000000)_2$

$X_3 - X_4$ = | 1 | 10000000 | 11011110100000000000000 |

$X_3 - X_4$ en décimal :

$= -0,1110111101 \times 2^2$

$= -11,10111101 = -(2^0 + 2^1 + 2^{-1} + 2^{-3} + 2^{-4} + 2^{-5} + 2^{-6} + 2^{-8})$

$= (-3,73828125)_{10}$

$X_3 \times X_4 = -1,1100111101 \times 2^2 \times (-1,11 \times 2^1)$

$= (-1,1100111101 \times -1,11) \times 2^{2+1}$

$= +11,001010101011 \times 2^3$

$X_3 \times X_4$ selon la norme IEEE 754 en simple précision :

$X_3 \times X_4 = +1,1001010101011 \times 2^{1}$

M = 10010101010110000000000 ; **S = 0** (le résultat est positif).

$Eb = E_{réel} + 2^{8-1}-1 = 4 + 2^7-1 = 4 + 128 -1 = (131)_{10} = (10000011)_2$

$X_3 \times X_4$ = | 1 | 10000011 | 10010101010110000000000 |

$X_3 \times X_4$ en décimal :

$= +11,001010101011 \times 2^3 = +11001,010101011 = +(2^0 + 2^3 + 2^4 + 2^{-2} + 2^{-4} + 2^{-6} + 2^{-8} + 2^{-9})$

$= (+25,333984375)_{10}$

Exercice 3 :

- Coder le nombre $(2097)_{10}$ en DCB.

$(2097)_{10} = (0010\ 0000\ 1001\ 0111)_{DCB}$

- Coder en ASCII (en hexadécimal) le mot suivant : COVID-19.

COVID-19 = 43 4F 56 49 44 2D 31 39

- Décoder le mot suivant (le mot est représenté en ASCII et en hexadécimal) :

53 74 72 75 63 74 75 72 65 20 4D 61 63 68 69 6E 65 = Structure Machine