

Microprocesseurs

Résumés et méthodologies

Nicolas Le Guerroué

3 octobre 2020

Contact pour l'information

Document réalisé en Latex par Nicolas Le Guerroué
dans le cadre du cours de Microprocesseurs

Telephone : 06.20.88.75.12

E-mail : nicolasleguerroue@gmail.com

Version du 3 octobre 2020

Permission vous est donnée de copier, distribuer et/ou modifier ce document sous quelque forme et de quelque manière que ce soit.

Table des matières

1	Conversion des nombres	3
1.1	Coder un nombre dans la norme IEEE754	3
1.1.1	Signe du nombre	3
1.1.2	La mantisse	3
1.1.3	Normalisation de la mantisse	4
1.1.4	Gestion de l'exposant	5
1.1.5	Mise en place des composantes de la trame	5
1.1.6	Conversion en hexadécimal	5
1.1.7	Cas des valeurs négatives	6

Section 1

Conversion des nombres

- ✔ Lessive
- ✔ Lessive

1.1 Coder un nombre dans la norme IEEE754


A travers un exemple, nous allons détailler la conversion de deux nombre dans la norme IEEE754.

Ces deux nombres seront 75.375 et -75.375 . Nous allons voir que la technique pour coder les nombres positifs ou négatifs est identique.

1.1.1 Signe du nombre

Le premier bit de la trame IEEE754 correspond au signe du nombre à coder.

- 0 si le nombre est **positif**
- 1 si le nombre est **négatif**

Dans notre cas (75.375), le premier bit vaut 

1.1.2 La mantisse

La mantisse correspond à la valeur significative du nombre.

Nous allons normaliser la partie entière de la mantisse afin d'avoir un nombre de la forme $n = 1.M \cdot 2^E$

avec

- E l'exposant
- M la partie décimale de la mantisse.

La partie entière

On convertit la **valeur absolue** de la partie entière en binaire. Pour cela, on peut utiliser la technique des puissances de 2.

Dans 75, il y a $1 \cdot 2^6, 1 \cdot 2^3, 1 \cdot 2^1$ et $1 \cdot 2^0$

On a donc $(75)_{10} =$ BIN 1001011

La partie décimale

On effectue la technique des multiplications par deux.

- On prend notre partie décimale que l'on multiplie par deux.

$$0.375 \cdot 2 = \boxed{0}.75$$

- Ensuite, on prend la partie entière du résultat que l'on met de côté.

La partie décimale est mise à la ligne et est de nouveau multipliée par deux.

$$0.75 \cdot 2 = \boxed{1}.5$$

On exécute cet algorithme tant que le résultat est différent de 1.

$$0.5 \cdot 2 = \boxed{1}$$

Ici, le résultat vaut 1, la conversion est terminée.

Les parties entières obtenues sont classées par pondérations décroissantes, c'est à dire que la première partie entière représente le bit de poids fort de la partie décimale.

Ainsi, $(0.375)_{10} =$ BIN 011

1.1.3 Normalisation de la mantisse

Le nombre 75.375 codé de façon **non normalisée** vaut BIN 1001011,011

Nous allons normaliser ce nombre.

Pour cela, on décale la virgule de n emplacement(s) vers la gauche afin d'obtenir un seul '1' à gauche de la virgule.

Pour un décalage de n emplacement(s), on multiplie le nombre binaire par 2^n

Retenir l'exposant \boxed{n} pour la suite.

BIN 1001011,011 normalisée vaut donc $\boxed{1},001011011 \cdot 2^6$

1.1.4 Gestion de l'exposant

Biais de l'exposant

Pour un format simple précision 32 bits, il convient d'ajouter à l'exposant la valeur $127(2^7 - 1)$

Notre exposant normalisé vaut donc $6 + 127 = (133)_{10} = \text{BIN } 10000101$

1.1.5 Mise en place des composantes de la trame

Nous avons nos trois éléments : **le signe, la partie décimale de la mantisse et l'exposant normalisés**

On place d'abord le bit de signe puis l'exposant sur 8 bit et enfin la partie décimale de la mantisse sur 23 bits.

Notre partie décimale de la mantisse vaut donc BIN 001011011. Pour mettre sur 23 bit (protocole), il suffit de compléter avec autant de 0 nécessaires en bit de poids faibles.

La partie décimale de la mantisse sur 23 bits vaut donc BIN 00101101100000000000000

Le nombre binaire vaut donc :

$N_{IEEE} = \text{01000010100101101100000000000000}$

Avec

Le signe

L'exposant

La partie décimale de la mantisse

1.1.6 Conversion en hexadécimal

On regroupe les bits par paquets de 4 et on convertit les paquets en hexadécimal.

$N_{IEEE} = \text{0100 0010 1001 0110 1100 0000 0000 0000}$

On obtient au final :

$(+75.375)_{10} = \text{HEX } 0x4296C000$

1.1.7 Cas des valeurs négatives

Pour les nombres négatifs, seuls le bit de signe change dans la trame sur 32 bits.

$N_{IEEE} =$ 1100 0010 1001 0110 1100 0000 0000 0000

D'où :

$(-75.375)_{10} =$ HEX 0xC296C000