Travaux Dirigés n°2

Architecture des ordinateurs

—IR 2000—

Représentation des nombres

Dans cette séance de travaux dirigés, nous allons étudier les méthode de représentations des nombres : le complément à 2 pour les nombres négatif ainsi que la norme IEEE 754 pour les nombres à virgule flottante.

▸ **Exercice 1. (Binaire, décimal et hexadécimal).**

1. Exprimer en binaire et en hexadécimal les nombres décimaux suivants : 32625 puis $128,75$ et $5/7$.
2. convertir en décimal $\quad$ $\mathrm{DADA,C}_{16}$ $\quad$ $11011,01111_2$.

▸ **Exercice 2. (Complément à deux).**

1. Donner sur 8 bits les représentations des valeurs entières suivantes : $-32 \quad -17 \quad -128$.

▸ **Exercice 3. (Additions et soustractions et multiplications).**

Convertir 36 et 25 en base 2 et 16, les ajouter, les soustraire et les multiplier, vérifier le résultat obtenu.

▸ **Exercice 4. (Détection des dépassements de capacité).**

Dans un premier temps on suppose les nombres non signés sur 8 bits.

1. Quel est l'intervalle des nombres représentables ?
2. Que se passe-t-il si l'on ajoute 151 et 200 ?
3. Quand y a-t-il dépassement de capacité ?

On suppose maintenant les nombres écrits sur 8 bits en complément à 2.

4. Quel est l'intervalle des nombres représentables ?
5. Que se passe-t-il si l'on ajoute les nombres 100 et 120 ?
6. Peut-il y avoir dépassement de capacité si l'on ajoute un nombre positif avec un nombre négatif ?
7. Quand y a-t-il dépassement de capacité ?

▸ **Exercice 5. (Norme IEEE 754).**

Les nombres réels sont le plus souvent représenté en virgule flottante. Il existe un format standard de représentation des nombres utilisé par la plupart des processeur : la norme IEEE 754. Dans cette norme, pour les nombrea en simple précision, on utilise 1 bit pour le signe, 8 bits pour les exposants et 23 bits pour la mantisse

$$n = s \times m \times 2^e.$$

1	8	23
s	e	m

- L'exposant est biaisé à 127. C'est-à-dire que pour coder un exposant e, on code le nombre $e+127$. Les exposants représentables vont de -126 à 127 (les valeurs 00000000_2 et 11111111_2 sont réservées pour des cas exceptionnels).
- Les nombres sont normalisés, c'est-à-dire que la mantisse est entre 1 et 2.

1. Quel est l'avantage de normaliser la mantisse ?
2. Quel est le plus grand nombre représentable ainsi, le plus petit ?

Les cas exceptionnels évoqués plus haut sont les suivants :

- Si $e = 255$ et $m = 0$, le nombre représenté est + ou − l'infini. Les opérations sur l'infini sont gérées correctement.
- Si $e = 255$ et $m \neq 0$, la représentation est considéré comme n'étant pas un nombre (NaN : Not a Number).
- Si $e = 0$ et $m = 0$ on représente le nombre $+0$ ou -0 selon le signe.
- Si $e = 0$ et $m \neq 0$ on dit que le nombre est dénormalisé. C'est-à-dire que la mantisse est écrite sans le 1 implicite.

3. Quel est le plus petit nombre strictement positif représentable ainsi ?
4. Que pensez-vous de cette représentation ?
5. Donner un algorithme pour l'addition et la multiplication de nombre flottant IEEE 754.