

# Cours : Représentation des nombres en machine

## 1. Introduction

Les ordinateurs ne comprennent que deux symboles : 0 et 1. Ces deux états correspondent à la présence ou l'absence de courant électrique. Mais comment représenter tous les nombres (entiers, réels, négatifs, très grands ou très petits) avec uniquement des 0 et des 1 ? La réponse se trouve dans différentes méthodes de codage normalisées (ex. norme IEEE 754).

## 2. Les entiers positifs

Les entiers positifs sont représentés en binaire. Avec  $n$  bits, on peut coder les entiers de 0 à  $2^n - 1$ .  
Exemple : avec 3 bits, on représente 0 à 7 ( $000_2$  à  $111_2$ ). Avec 8 bits, de 0 à 255. Avec 16 bits, de 0 à 65 535.

## 3. La virgule fixe

On réserve une partie des bits pour la partie entière et une partie pour la fraction. Exemple : sur 8 bits, 4 pour l'entier et 4 pour la fraction :  $1101.0101_2 = 13,3125_{10}$ . La partie fractionnaire s'obtient en utilisant les puissances négatives de 2 ( $1/2$ ,  $1/4$ ,  $1/8$ ...).

## 4. La virgule flottante

Inspirée de la notation scientifique. Exemple :  $13,25_{10} = 1101,01_2 = 1,10101 \times 2^3$ . On stocke séparément la mantisse et l'exposant. Exemple simplifié sur 8 bits : 3 bits pour l'exposant, 5 bits pour la mantisse.

## 5. Les entiers négatifs

Deux méthodes principales :

- **Bit de signe** : le bit de gauche indique si le nombre est positif (0) ou négatif (1).
- **Complément à 2** : méthode standard utilisée aujourd'hui. On inverse les bits puis on ajoute 1 pour obtenir la valeur négative.

## 6. La norme IEEE 754

Norme internationale pour la représentation des nombres flottants. Un nombre est représenté par trois parties :

- 1 bit pour le signe
- Plusieurs bits pour l'exposant biaisé
- Plusieurs bits pour la mantisse

Exemple (simple précision, 32 bits) : 1 bit de signe, 8 bits d'exposant (biais 127), 23 bits de mantisse.