

Réels

ARO
6 - Réels

Résumé du document
TBD

Table des matières

- 1. Codage binaire 2**
 - 1.1. Biais 2
- 2. Normalisés / non-normalisés 3**
 - 2.1. Normalisés 3
 - 2.2. Non-normalisés 3
- 3. Standard IEEE 754 (virgule flottante) 4**
 - 3.1. Binary8 (quarter precision) 4
 - 3.2. Binary16 (half precision) 4
 - 3.3. Binary32 (single precision) 4
 - 3.4. Binary64 (double precision) 4
 - 3.5. Binary128 (quadruple precision or Quad) 4

1. Codage binaire

Le codage créé par William Kahan en 1985 est un standard IEEE 754 qui permet de représenter les nombres réels en virgule flottante. Il est utilisé dans la plupart des ordinateurs et des langages de programmation.

1.1. Biais

Le biais est une valeur ajoutée à l'exposant pour permettre de représenter des nombres négatifs. Il est égal à $2^{k-1} - 1$ où k est le nombre de bits de l'exposant.

2. Normalisés / non-normalisés

2.1. Normalisés

Pour qu'un nombre soit considéré comme normalisé il doit forcément avoir le bit à gauche du point décimal égal à 1.

2.2. Non-normalisés

Pour qu'un nombre soit considéré comme non-normalisé il doit forcément avoir le bit à gauche du point décimal égal à 0.

3. Standard IEEE 754 (virgule flottante)

3.1. Binary8 (quarter precision)

- nombre de bits : 8
- mantisse sur 3 bits
- exposant sur 4 bits
- biais : 7 ($2^{4-1} - 1$)

3.2. Binary16 (half precision)

- nombre de bits : 16
- mantisse sur 10 bits
- exposant sur 5 bits
- biais : 15 ($2^{5-1} - 1$)

3.3. Binary32 (single precision)

- nombre de bits : 32
- mantisse sur 23 bits
- exposant sur 8 bits
- biais : 127 ($2^{8-1} - 1$)

3.4. Binary64 (double precision)

- nombre de bits : 64
- mantisse sur 52 bits
- exposant sur 11 bits
- biais : 1023 ($2^{11-1} - 1$)

3.5. Binary128 (quadruple precision or Quad)

- nombre de bits : 128
- mantisse sur 112 bits
- exposant sur 15 bits
- biais : 16383 ($2^{15-1} - 1$)