$\sqrt{f(x)}$

MATHEMATIQUES APPLIQUEES

Représentation des nombres négatifs en binaire

11

1- PREAMBULE

Les fiches précédentes les bases numériques considéraient que les nombres convertis en base 2 étaient uniquement des entiers positifs. Parfois il est nécessaire de travailler avec des nombres binaires négatifs.

Nous allons voir dans la suite 3 solutions possibles pour les représenter.

2- REPRESENTATION "SIGNE-MAGNITUDE"

Définition : ***Le bit de poids fort indique le signe (0 si positif, 1 si négatif) et les bits restants la valeur absolue du nombre.***

Exemple : on code sur 3 bits un nombre binaire qui peut être positif ou négatif.

Nombre en base 10	Nombre binaire codé sur 3 bits signe-magnitude
+ 3	0 1 1
+ 2	0 1 0
+ 1	0 0 1
+ 0 ou - 0	0 0 0 ou 1 0 0
- 1	1 0 1
- 2	1 1 0
- 3	1 1 1

signe magnitude signe magnitude

Remarques :

Avec n bits, on peut représenter des entiers signés entre: $-(2^{n-1}-1)$ et $+(2^{n-1}-1)$.

Il est important de connaitre le nombre de bits utilisés pour représenter le nombre binaire.

Quel que soit le nombre de bits utilisés pour le codage, le chiffre 0 systématique a deux représentations possibles.

Cette méthode de codage est **peu utilisée** car elle ne permet pas aux machines d'effectuer les opérations binaires (+, -, * ,/).

3- REPRESENTATION "CODE COMPLEMENT A 1"

Définition : ***Le "complément à un" d'un nombre binaire est la valeur obtenue en inversant tous les bits de ce nombre (en permutant les 0 par des 1 et inversement).***

Exemple : on code sur 3 bits un nombre binaire qui peut être positif ou négatif.

Nombre en base 10	Nombre binaire codé sur 3 bits complément à 1
+ 3	0 1 1
+ 2	0 1 0
+ 1	0 0 1
+ 0 ou - 0	0 0 0 ou 1 1 1
- 1	1 1 0
- 2	1 0 1
- 3	1 0 0

Remarques :

Avec n bits, on peut représenter des entiers signés entre: $-(2^{n-1}-1)$ et $+(2^{n-1}-1)$.

Il est important de connaitre le nombre de bits utilisés pour représenter le nombre binaire.

Quel que soit le nombre de bits utilisés pour le codage, le chiffre 0 systématique a deux représentations possibles.

Cette méthode de codage est **peu utilisée** car elle ne permet pas aux machines d'effectuer toutes les opérations binaires (+, -, * ,/).

4- REPRESENTATION "CODE COMPLEMENT A 2"

<u>Définition :</u> ***Le "complément à deux" d'un nombre binaire est la valeur obtenue en inversant tous les bits de ce nombre, puis en ajoutant 1.***

<u>Méthode appliquée sur un exemple :</u> on code sur 6 bits le nombre -20.

1/ On code 20 en binaire $20_{(10)} \rightarrow$ 0 1 0 1 0 0$_{(2)}$

retenue : 1 1

2/ On inverse bit à bit 1 0 1 0 1 1

3/ On ajoute 1 + 0 0 0 0 0 1

On obtient $-20_{(10)} \leftarrow$ 1 0 1 1 0 0$_{(\text{complément à 2 sur 6 bits})}$

<u>Autre méthode plus rapide :</u>

1/ On repère le premier bit à 1 du nombre binaire en partant de la droite $20_{(10)} \rightarrow$ 0 1 0 1 0 0$_{(2)}$

2/ on réécrit les bits à droite avec le premier 1 compris tel quel

3/ On inverse les bits restants $-20_{(10)} \leftarrow$ 1 0 1 1 0 0$_{(\text{complément à 2 puissance 3})}$

premier 1

on garde les bits tel quel

On inverse les bits

<u>Remarques :</u>

- Avec n bits, on peut représenter des entiers signés entre: - (2^{n-1}) et + (2^{n-1}-1).
- Le nombre 0 à 1 seule représentation possible.

Il est important de connaitre le nombre de bits utilisés pour représenter le nombre binaire.

Cette méthode de codage est **souvent utilisée** car elle permet les opérations binaires (+, -, * ,/).

<u>Représentation circulaire :</u>

On peut représenter sur un cercle sectorisé l'ensemble des nombres enroulés.

Lorsque le bit de poids fort passe à 1, le signe du nombre décimal change, mais pas sa valeur décimale.