



I Représentation binaire d'un entier relatif

Nous avons vu comment représenter les entiers naturels en langage binaire. Une idée qui aurait pu être retenue est de coder le signe de l'entier sur le bit le plus à gauche, puis la valeur absolue de l'entier sur les bits restants.

Exemple : En codant sur un octet le nombre -45, on aurait obtenu 10101101 avec le 1 du bit de poids le plus fort qui code le signe '-' et 101101 qui code 45. Le problème de cette méthode est le codage de deux 0 : Le positif (00000000) et le négatif (10000000) . Il a donc fallu trouver une autre méthode.



: Les entiers sont codés sur un octet ou un multiple d'un octet : 16 bits, 32 bits...

1) Codage d'un entier relatif sur 8 bits.

Le bit de poids le plus fort (à gauche) sert à coder le signe de l'entier. Il reste donc 7 bits pour coder le nombre soit des valeurs entre -128 et 127.

Pour coder un entier positif, on met le bit de poids le plus fort à 0 et l'on détermine l'écriture en binaire du nombre

Exemple : Codage de 89 sur 8 bits 01011001.

On va représenter 89 par $256 (2^8) - 89 = 167$. Ainsi $-89 = 10100111$. On notera que pour tout entier relatif x entre 0 et -128, $256 + x \geq 128$, ce qui assure le 1 en début de codage .

2) Méthode du complément à 2

Il existe une méthode simple qui permet de trouver la représentation des entiers relatifs en binaire.

1. On écrit la valeur absolue du nombre (par exemple 45 pour -45) : 00101101
2. On inverse tous les bits : Les 1 deviennent des 0 et inversement : 11010010
3. On ajoute 1 au résultat : $11010010 + 1 = 11010011$
4. Le nombre obtenu est le codage de -45 sur 8 bits.

Vérifions la cohérence du résultat obtenu. Il semble logique que $+45 - 45 = 0$

On a donc :

00101101

+11010011

= 00000000

Il reste une retenue (sur le neuvième bit), mais l'entier étant codé sur 8 bits, on obtient bien 0.

Cette méthode de calcul vient de la formule $x' = 2^n - |x|$, ici $256 - |x|$ où $|x|$ est la valeur absolue de x .

Astuce : Pour utiliser la méthode du complément à 2 : On part de la droite, on ne change les bits que lorsque l'on rencontre le premier 1, sans toucher à celui-ci..

Exemple : $18 = 00010010$ devient 11101110 qui code -18.

3) Exercices

1) Donner les représentations en binaire sur 8 ou 16 bits suivant les besoins des entiers suivants :

-98 ; 358, -129, 0, -1, 127.

2) Quels sont les entiers que l'on peut coder sur 16 bits ?, sur 2^n bits ?

3) Ecrire un algorithme qui à partir de la représentation d'un entier positif affiche l'écriture en binaire de son opposé.