représentation d'entiers relatifs

NSI | 01 bis

1. Une fausse bonne idée

Nous avons vu dans le chapitre précédent comment représenter en binaire des nombres entiers positifs. En s'inspirant de cette représentation, on pourrait naturellement avoir envie de coder un entier relatif en prenant l'entier naturel associé sa valeur absolue et en lui ajoutant un bit de signe avec la convention :

- si l'entier est positif ou nul, le bit de signe est 0
- si l'entier est négatif, le bit de signe est 1

De plus, on pourrait choisir que ce bit de signe est le bit de plus fort poids dans le codage de l'entier.

Ainsi, un entier relatif serait codé par un bit de signe suivi du codage de sa valeur absolue.

Par exemple, sur 4 bits, -3 serait codé sous la forme …….. : un bit de signe égal à..... suivi du codage binaire..... de 3.

Avec cette représentation, sur N bits il serait ainsi possible de coder pratiquement tous les entiers compris entre …….

Mais si ce codage est simple à mettre en œuvre, il présente un premier inconvénient : le zéro admet deux codages : …….. et ……...

Un deuxième inconvénient, est que les calculs arithmétiques deviennent faux. Par exemple, posons l'addition $(0101)_2 + (1011)_2$:

$$\begin{array}{rr} & 0101 \rightarrow \\ + & 1011 \rightarrow \\ \hline & \rightarrow \end{array}$$

Cette représentation a donc été vite abandonnée.

2. La méthode du complément à deux

Avant de représenter un entier relatif avec cette méthode, il est nécessaire de définir le nombre de bits qui seront utilisés pour cette représentation (souvent 8, 16 , 32 ou 64 bits)

Voici la méthode du complément à deux pour coder les nombres entiers négatifs (pour les positifs, on garde la représentation binaire du chapitre 1), avec un exemple en parallèle :

Méthode	Exemple : écriture de -15 sur un octet
On écrit la valeur absolue du nombre négatif en binaire	
On inverse chaque bit de cette écriture (les 0 deviennent des 1 et vice versa)	
On ajoute 1 (en base 2) en tenant compte des éventuelles retenues	

Donc : -15 = $(\ldots\ldots\ldots\ldots)_{c2}$ Vérifions qu'on trouve toujours 15-15=0 :	*Remarque :* + _____

Remarques :