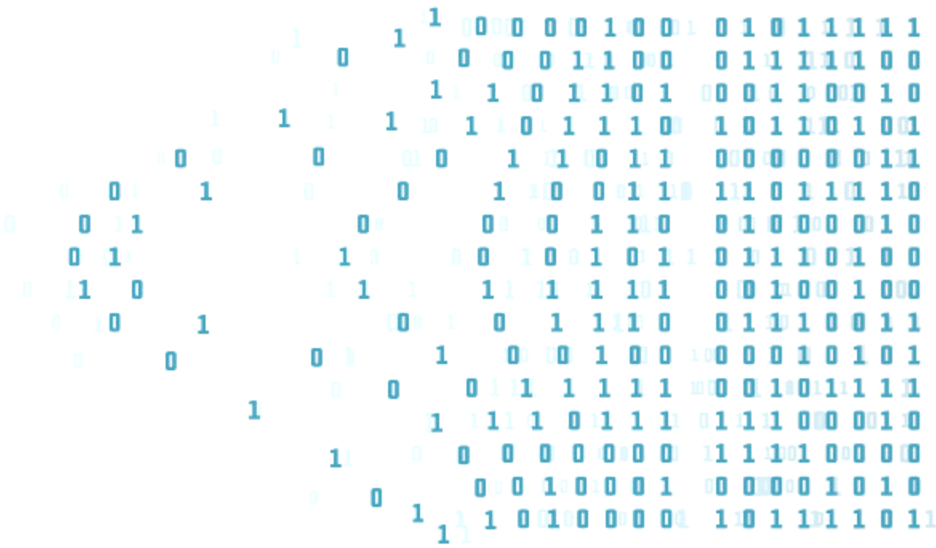
REPRESENTATION DES DONNEES

NSI | 02



REPRESENTATION DES NOMBRES ENTIERS RELATIFS

[Marion SZPIEG]

Savoir comment sont représentés les entiers relatifs en base 2 et être capable de les manipuler

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 1011 : un bit de signe égal à 1 suivi du codage binaire 011 de 3.

Avec cette représentation, sur N bits il serait ainsi possible de coder pratiquement tous les entiers compris entre -2^{N-1} et $2^{N-1} - 1$.

Mais si ce codage est simple à mettre en œuvre, il présente un premier inconvénient : le zéro admet deux codages : 00...0 et 10...0.

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

$$\begin{array}{rcl} & 0101 & \rightarrow 5 \\ + & 1011 & \rightarrow -3 \\ \hline & 0000 & \rightarrow 0 \end{array}$$

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	$(15)_{10} = (00001111)_2$
On inverse chaque bit de cette écriture (les 0 deviennent des 1 et vice versa)	On inverse : 11110000
On ajoute 1 (en base 2) en tenant compte des éventuelles retenues	$(11110000)_2 + (00000001)_2 = (11110001)_2$

Donc : $-15 = (11110001)_{c2}$ Vérifions qu'on trouve toujours 15-15=0 :	$\begin{array}{rcccccccc} & 1 & 1 & 1 & 1 & 1 & 1 & 1 & \\ & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ + & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ \hline & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$	*Remarque : il reste une retenue qu'on aimerait rajouter sur un 9^e^ bit, mais comme on est sur un octet, on l'ignore, et on retombe bien sur 0.*

Remarques : • le bit de poids fort représente le signe. On retrouve l'idée du bit de signe.

• l'étendue : sur un octet on code les entiers compris entre -128 et 127. Plus généralement, sur n bits, on peut représenter les entiers entre -2^{n-1} et $2^{n-1}-1$

• la soustraction se traduit par l'addition du complément à 2 : 10 – 3 = 10 + (-3)