Représentation des entiers relatifs.

I Le binaire non signé.

Les entiers naturels sont codés sur machine en base 2 (nous savons déjà passer de la base 10 à la base 2) sur un nombre arbitraire de bits. Pour simplifier nos illustrations, nous considérerons des entiers codés sur 1 octet, c'est-à-dire sur 8 bits.

Rappel, le nombre 10 en base 2 s'écrit : 1010.
Dans la mémoire, sur 8 bits, il est codé : 00001010

Sur 8 bits, nous pouvons coder $2^8 = 256$ nombres, soit de 0 à 255.

II Le binaire signé.

Dans la représentation en binaire signé, le bit de poids fort (le plus à gauche) sert à représenter le signe :
0 pour le positif
1 pour le négatif.

Le problème est le suivant : nous connaissons par exemple le codage de 10 : 00001010
Naturellement, nous aurions envie de coder le nombre -10 par : 10001010

Si nous additionnons 10 et -10, nous devons tomber sur 0, or :

$$\begin{array}{r} 00001010 \\ +\ 10001010 \\ \hline 10010100 \end{array}$$

Vous constatez que nous ne tombons pas sur 00000000 qui est la représentation de 0

Pour comprendre comment est codé un nombre négatif relativement à un nombre positif, il faut revenir à la base : Un nombre positif additionné avec son opposé donne 0, soit 00000000

Pour prendre l'opposé d'un nombre positif sur n bits, la technique consiste donc à prendre son complément à 2^n + 1

A Le complément à 2^n.

Prendre le complément à 1 d'un nombre consiste à inverser les 0 et les 1 de sont écriture binaire.
Donc :
le complément à 1 de 0 est 1
le complément à 1 de 1 est 0

Pour 10, nous avons : 00001010
Son complément à 2^8 est : 11110101

B +1

Nous constatons qu'en additionnant un nombre et son complément à 2^8 nous obtenons : 11111111
Exemple avec 10 et son complément à 2^8 :

$$\begin{array}{r} 00001010 \\ +11110101 \\ \hline 11111111 \end{array}$$

Nous remarquons que :

$$\begin{array}{r} 11111111 \\ +\ 00000001 \\ \hline (1)00000000 \end{array}$$

Et donc en perdant la retenue, sur 8 bits nous tombons bien sur 00000000

L'opposé de 10 sur 8 bits sera donc :

$$\begin{array}{lr} & complément\ \text{à}\ 2^8\ de\ 10 : 11110101 \\ +00000001 & \underline{+00000001} \\ & 11110110 \end{array}$$

En effet :

$$\begin{array}{r} 00001010 \\ \underline{+11110110} \\ (1)00000000 \end{array}$$

Sur 8 bits, en perdant la retenue, la somme est bien égale à 00000000

Exercice : Calculer la représentation binaire de l'opposé de 3, puis de 13.

C Comment déterminer l'écriture de l'opposé d'un nombre négatif.

On lui retranche 1, puis on prend son complément à 2^n

Reprenons notre exemple de -10, dont la représentation sur 8 bits est : 11110110

$$\begin{array}{r} 11110110 \\ \underline{-00000001} \\ 11110101 \end{array}$$

Le complément à 2^8 de 11110101 est 00001010 qui est bien la représentation de 10 sur 8 bits.

Exercice : Retrouvez la représentation de 3 et 13 en partant des représentations de -3 et de -13 trouvés dans l'exercice précédent.

Nous pouvons constater que sur 8 bits, nous ne pouvons représenter les nombres que de -127 à 127
En effet, il ne reste que 7 bits pour coder le nombre derrière le signe : $2^7 = 128$, donc de 0 à 127