

Entiers négatifs – Complément à 2



En python, un entier est codé sur 4 octets.



L'octet **[1110 1001]** n'a **a priori** aucune signification. Cela pourrait être :

- le nombre 233
- le caractère é
- le nombre -23

Problématique 1

On souhaite pouvoir représenter un nombre entier positif ou négatif.

Le binaire naturel ne suffit pas car en binaire naturel, par exemple, 11101001 vaut 233 et rien d'autre.

Il faut comprendre que dans les mémoires de la machine, si on dispose de 8 bits par exemple, on ne peut pas écrire -1111, il n'y a pas de place pour un -, on ne peut écrire que des 1 et des 0.

Une idée qui ne fonctionne pas

Une première idée est d'utiliser le premier bit du mot comme bit de signe.

Cette idée n'est pas satisfaisante. En effet, il y a de nombreux cas où on n'a pas besoin de nombres négatifs. Si on sait que l'on n'aura besoin que d'entiers positifs, il est dommage de gâcher un bit pour indiquer le signe. On voudrait donc pouvoir disposer de plusieurs modes :

- Un mode d'entiers positifs dans lequel il n'y a que des entiers positifs, donc pas de nécessité de représenter un signe.
- Un mode d'entiers signés dans lequel les entiers peuvent être positifs ou négatifs et qui nécessite donc un codage du signe.

Dans les deux cas, on peut avoir à faire une addition. Dans la machine, une addition est réalisée par un circuit électronique. On aimerait pouvoir utiliser le même circuit quelque soit le mode choisit.

C'est pour cela que l'on adopte le CA2 bien qu'il soit d'aspect plus compliqué.

Le complément à 2 CA2

- Un nombre commençant par 0 est positif et en binaire naturel.
- Un nombre commençant par 1 est négatif, **mais pas en binaire naturel!**
- pour changer le signe : inverse puis +1.

Exemples :

- 45 est positif et $45 = 101101_b$, donc on code $[0010\ 1101]_{CA2}$
- -45 est négatif et $45 = 101101_b$
 $45 = [0010\ 1101]_{CA2} \xrightarrow{inv} [1101\ 0010] \xrightarrow{+1} -45 = [1101\ 0011]_{CA2}$
- $[0010\ 1000]_{CA2}$ commence par 0 $\Rightarrow$ représente le nombre positif $0010\ 1000_b = 40$
- $[1010\ 1000]_{CA2}$ commence par 1 $\Rightarrow$ représente un nombre négatif.
 $[1010\ 1000]_{CA2} \xrightarrow{inv} [0101\ 0111] \xrightarrow{+1} [0101\ 1000]_{CA2} = 88.$
 Donc $[1010\ 1000]_{CA2}$ représente -88



Le CA2 est fondé sur le fait que les nombres occupent une place limitée en mémoire. Sur 8 bits, on ne peut aller que de $[0000\ 0000]$ à $[1111\ 1111]$. Si on va plus loin, on fait un **tour de compteur**.

Par exemple $[1111\ 1111] \xrightarrow{+1} 1\ [0000\ 0000]$. Le 1 à gauche est perdu, on a fait un tour de compteur. Tout se passe donc comme si $[1111\ 1111] = -1$.