

Chapitre 2 : Représentation binaire d'un entier relatif

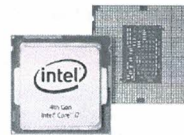
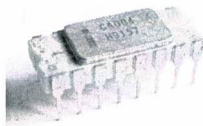
Nous avons vu précédemment comment écrire un nombre entier naturel dans une base quelconque, et plus particulièrement en base 2. Nous allons voir dans ce qui suit comment aller plus loin et incorporer en plus le signe d'un nombre, pour représenter ainsi un nombre entier relatif, principalement dans sa représentation binaire, et répondre à la question :

Comment écrire -5 en représentation binaire ?

1. Un peu d'histoire

Que ce soit des images, des vidéos, des fichiers de traitement de texte, des nombres, toutes données stockées sur un ordinateur ou transmises vers un ordinateur ne l'est que par l'intermédiaire de circuit électronique (transistor) ne pouvant se trouver que sous deux états : sous tension (on note cela l'état 1) ou hors tension (on note cela l'état 0).

En 1970, le premier microprocesseur (*Intel 4004*) commercialisé comptait environ 2 000 transistors, en 2018, le micro processeurs *Intel Core i7* (destiné au grand public) en compte 2,3 milliards !



Cette unité de stockage est appelée *bit*, qui peut prendre les valeurs conventionnelles 0 ou 1. Ainsi, toute action informatique est une suite d'opérations sur des paquets de 0 et de 1, regroupés par huit : les *octets*.

Avec des octets, on peut représenter les entiers naturels de 0 (0000 0000 en base 2) à 255 (1111 1111 en base 2). Donc 45 est représenté par (0010 1101)₂.

Les processeurs récents sont des processeurs 32 bits ou 64 bits qui disposent de 32 ou 64 bits pour stocker un nombre. Si on utilise 16 bits, soit 2 octets, on peut représenter les entiers naturels jusqu'à 65 535 (1111 1111 1111 1111 en base 2) et dans ce cas 45 est représenté par (0000 0000 0010 1101)₂.

Avec n bits, on peut représenter les nombres entre 0 et $2^n - 1$.

Un entier est en général codé sur 4 octets, soit 32 bits, ou sur 8 octets, soit 64 bits.

2. Une fausse bonne idée

Partant de la représentation des entiers naturels, il est assez naturel d'envisager de coder un entier relatif en prenant l'entier naturel associé à sa valeur absolue et en lui ajoutant un bit de signe avec la

convention :
 • entier positif à un bit de poids fort à 0
 • entier négatif à un bpf à 1

Une convention supplémentaire stipulerait 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, -2 serait codé sous la forme 1010 : un bit de signe égal à 1 suivi du codage binaire 010 de 2.

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. Ce qui est clairement regrettable.

Un deuxième inconvénient, en relation avec les circuits électroniques réalisant les opérations arithmétiques, rend même cette solution inenvisageable. En effet, un *circuit additionneur* permet l'addition de deux entiers naturels codés sur un nombre de bits donné.

En pratique, le circuit ne voit que des codes binaires. Ainsi, l'addition binaire de 0101 (soit 5 en base 10) et 1101 (-5 en base 10) donne :

$$\begin{array}{r}
 + 5 \\
 - 5 \\
 \hline
 0
 \end{array}
 \qquad
 \begin{array}{r}
 5 \\
 0101 \\
 1101 \\
 \hline
 (1)0010
 \end{array}$$

Une solution à ce problème existe mais elle a un coût : construire un circuit spécifique pour effectuer les soustractions ! Une autre solution ne nécessite que l'utilisation d'un additionneur pour réaliser à la fois les additions et les soustractions ! Il s'agit d'utiliser l'encodage dit par *complément à 2*. Dans cet encodage, le *bit* de poids fort est toujours utilisé pour représenter le signe des entiers, la représentation des nombres positifs est inchangée mais celle des nombres négatifs utilise un encodage par complément à 2.

3. Une solution : l'algorithme du complément à 2 via le complément à 1

Dans la suite, nous allons définir une méthode qui, en utilisant le *bit* de signe, nous permettra de trouver la représentation binaire associée à l'opposé d'un entier relatif.

Activité : Le complément à 1

On cherche ainsi à construire un nombre x dont la représentation binaire commence par 0 si x est positif et 1 sinon, et qui respecte le fait que $a + x = 0$.

Compléter les additions suivantes nécessaires aux calculs de $a + b + 1 = (0)$ où b est inconnu :

$\begin{array}{r} a \\ + b \\ + 1 \\ \hline a + b + 1 = 0 \end{array}$	$\begin{array}{r} 0101 \rightarrow S \\ + 1010 \\ + 1 \\ \hline (1)0000 \end{array}$ 1011	$\begin{array}{r} 0001 \\ + 1110 \\ + 1 \\ \hline (1)0000 \end{array}$	$\begin{array}{r} 0110 \\ + 1001 \\ + 1 \\ \hline (1)0000 \end{array}$
--	--	--	--

Le complément à 1 est le nombre b obtenu en intervertissant les valeurs des *bits* de a , c'est à dire que l'on change les 0 en 1 et les 1 en 0. En ajoutant 1 au complément à 1, on définit $1 + b$ qui est l'opposé du nombre a !

Activité : Représentation binaire de l'opposé via l'algorithme du complément à 2

a) Proposer enfin un algorithme qui permet de construire l'opposé d'un entier relatif quelconque a :

- Prendre a
- _____
- _____ pour obtenir le nombre $-a$

b) Écrire alors tous les opposés possibles des nombres binaires s'écrivant sur 4 bits, en donnant leur équivalent décimal.

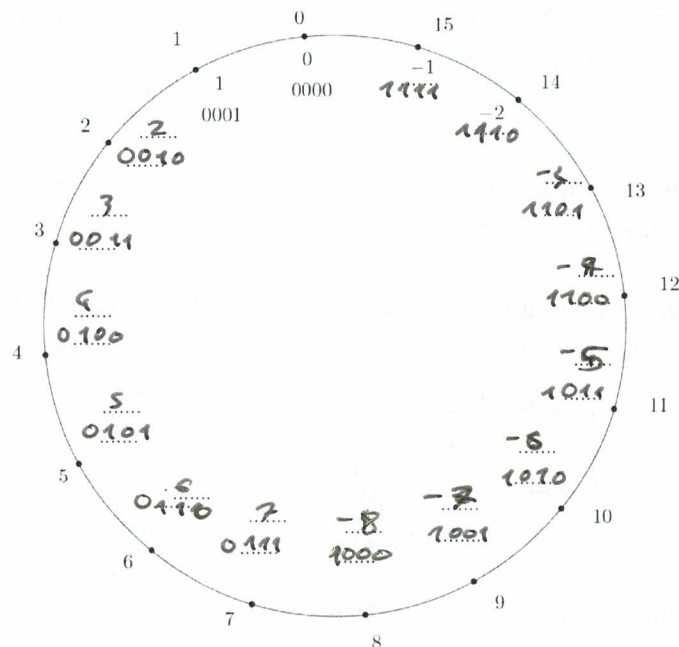
- 0 : 0000 dont l'opposé est $1111+1 = (1)0000$ qui correspond bien à 0.
- 1 : 0001 dont l'opposé est 1110+1 = 1111 qui correspond bien à -1.
- 2 : 0010 " 1101+1 = 1110
- 3 : 0011 " 1100+1 = 1101
- 4 : 0100 " 1011+1 = 1100
- 5 : 0101 " 1010+1 = 1011
- 6 : 0110 " 1001+1 = 1011
- 7 : 0111 " 1000+1 = 1001
- 8 : 1000 " X

c) Que remarque-t-on sur le bit de poids fort des nombres positifs ? Des nombres négatifs ?

0

1

d) Pour les prochaines questions, on pourra s'aider du cercle suivant à compléter.



e) De même, remplir le tableau suivant et les phrases suivantes :

Binaire	000	001	010	011	100	101	110	111
Entier signé	0	1	2	3	-4	-3	-2	-1

a. sur 3 bits,

- la valeur absolue est codée sur 2 bits
- le nombre le plus petit vaut -4 = -2^2
- le nombre le plus grand vaut 3 = 2^{2-1}

b. sur 4 bits,

- la valeur absolue est codée sur 3 bits
- le nombre le plus petit vaut -8 = -2^3
- le nombre le plus grand vaut 7 = 2^{3-1}

f) Sur 8 bits, quel sera le nombre relatif le plus petit et le plus grand qu'il sera possible d'écrire en binaire ?

-2⁷ 2⁷⁻¹

g) Sur n bits, quel sera le nombre relatif le plus petit et le plus grand qu'il sera possible d'écrire en binaire ?

-2ⁿ⁻¹ 2ⁿ⁻¹

Exercice 1 :

Coder, sur 8 bits, en complément à deux les entiers relatifs : $(-121)_{10}$ et $(-100)_{10}$.

$$\begin{array}{lcl}
 121 = 60 \times 2 + 1 & 121 = 01111001 & 100 = 50 \times 2 + 0 & 100 = 01100100 \\
 60 = 30 \times 2 + 0 & -121 = 1000111 & 50 = 25 \times 2 + 0 & -100 = 10011100 \\
 30 = 15 \times 2 + 0 & & 25 = 12 \times 2 + 1 & \\
 15 = 7 \times 2 + 1 & & 12 = 6 \times 2 + 0 & \\
 7 = 3 \times 2 + 1 & & 6 = 3 \times 2 + 0 & \\
 3 = 2 \times 1 + 1 & & 3 = 2 \times 1 + 1 & \\
 1 = 0 \times 2 + 1 & & 1 = 2 \times 0 + 1 &
 \end{array}$$

Exercice 2 :

Donner en base 10 la valeur des octets signés 1110 0111 et 1100 0001.

$$\begin{array}{l}
 11100111 \xrightarrow{-1} 11100110 \xrightarrow{\text{renverse}} 00011001 \\
 \text{-----} \\
 \text{-----} \\
 \text{-----} \\
 \text{-----} \\
 \text{-----}
 \end{array}$$

Exercice 3 :

Écrire une fonction `complement(b)` qui calcule le complément à 2 d'un nombre binaire donné sous la forme d'un tableau de taille quelconque contenant uniquement les chiffres 0 et 1. La fonction renverra un nouveau tableau, de même taille que `b`.

4. L'essentiel à retenir

- En représentation binaire, un nombre **négatif commence toujours par 1**.
- Il est ainsi possible de n'utiliser que l'addition.
- L'algorithme (dit du **complément à 2**) qui permet de trouver l'opposé d'un nombre s'écrit :
 1. On code en binaire la valeur absolue de l'entier relatif comme s'il s'agissait d'un entier naturel.
 2. Inverser les bits $0 \leftrightarrow 1$ (pour obtenir le complément à 1)
 3. Ajouter 1 au résultat pour obtenir la représentation binaire.
- Sur n bits, on code tous les entiers relatifs compris dans l'intervalle : $[-2^{n-1}; 2^{n-1} - 1]$

Quel est alors la représentation binaire du nombre -5 ? _____