

## 2.2

# Représentation des entiers relatifs

NSI 1ÈRE - JB DUTHOIT

### 2.2.1 Introduction

Pour traiter les nombres négatifs, l'ordinateur ne dispose pas du signe moins. Il faut donc mettre en place une convention pour représenter en binaire des nombres entiers négatifs.

Il existe plusieurs méthodes, et notamment :

- Représentation du bit signé
- le complément à 2

### 2.2.2 La représentation des entiers à l'aide du bit signé

Pour coder un entier relatif, une méthode simple consiste à utiliser le bit de poids fort pour indiquer le signe :

- 0 pour un nombre positif
- 1 pour un nombre négatif

### Exemples

$$12_{10} = \dots$$

$$-12_{10} = \dots$$

### Remarque

Il se pose deux problèmes :

- Il y a deux zéros : Sur 4 bits, 0000 et 1000
- Cette représentation induit de la complexité pour des opérations simples (addition de deux nombres de signes différents par exemple). Essayez de faire  $2 + (-2)$  pour en avoir le cœur net :-)

### 2.2.3 Le complément à deux

Comme précédemment, on convient que :

- Bit de signe à 0 pour un entier positif
- Bit de signe à 1 pour un entier négatif

Exemple sur 8 bits :

| s |   |   |   |   |   |   |   | valeur décimale | compl à $2^7$ |
|---|---|---|---|---|---|---|---|-----------------|---------------|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | -1              | 128-127       |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | -2              | 128 -126      |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | -3              | 128 -125      |
| 1 |   |   |   |   |   |   |   |                 |               |
| 1 |   |   |   |   |   |   |   | -12             | 128-116       |
| 1 |   |   |   |   |   |   |   |                 |               |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | -127            | 128-1         |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | -128            | 128-0         |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 127             |               |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 126             |               |
| 0 |   |   |   |   |   |   |   |                 |               |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 2               |               |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1               |               |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0               |               |

Le complément à 2 d'un nombre binaire s'obtient de la façon suivante :

Voici un exemple pour  $-12$  :

|                           |     |    |    |    |   |   |   |   |
|---------------------------|-----|----|----|----|---|---|---|---|
| Poids en binaire          | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |
| Valeur absolue en binaire |     |    |    |    |   |   |   |   |
| Inversion des bits        |     |    |    |    |   |   |   |   |
| On ajoute 1               |     |    |    |    |   |   |   |   |

Donc  $-12$  est représenté par 11110100 sur 8 bits, en complément à 2.

## Savoir-Faire 2.7

Savoir passer d'un entier relatif à sa représentation en binaire en complément à 2.

- ( $\triangle$  car facile!) Représenter  $+7_{10}$  sur 8 bits, en complément à 2.
- Représenter  $-8_{10}$  sur 8 bits, en complément à 2.
- Représenter  $-121_{10}$  sur 8 bits, en complément à 2.
- Représenter  $-100_{10}$  sur 8 bits, en complément à 2.
- Représenter  $-4_{10}$  sur 4 bits, en complément à 2.

## Exercice 2.4

Trouver le complément à 2 de  $-59$  avec une mémoire de 8 bits.

Faire l'addition binaire  $59 + (-59)$

## Savoir-Faire 2.8

SAVOIR PASSER D'UN ENTIER BINAIRE EN COMPLÉMENT À 2 À SA PRÉSENTATION EN BASE 10.

Trouvez la représentation décimale des entiers relatifs dont la représentation binaire sur 8 bits :

1. 0000 0000
2. 1000 0000

3. 0111 1111

4. 1001 1001

## Savoir-Faire 2.9

### SAVOIR EFFECTUER DES OPÉRATIONS AVEC LES ENTIERS RELATIFS

On considère les opérations suivantes :

1.  $49 + 25$

2.  $35 + 65$

3.  $45 - 12$

4.  $67 - 13$

5.  $29 - 84$

Pour chaque calcul :

- Effectuer le calcul directement avec les représentations décimales (trop facile :-))
- Représenter chaque opérande en binaire sur 8 bits, en complément à deux.
- Effectuer l'opération binaire
- Représenter le résultat de l'opération en base 10 et comparer avec le résultat attendu.