Il y a une différence fondamentale entre un nombre et son écriture. Un nombre existe en dehors de toute écriture au sens où, par exemple, cinq est le point commun entre cinq chaises, cinq carottes, cinq étoiles. Plus précisément, un nombre entier est le cardinal d'un ensemble fini. Une écriture, en revanche, est une manière de représenter un nombre et en particulier n'est pas unique. Par exemple le nombre douze peut se représenter par « douze », « 12 », « XII », etc.

On décrit dans ce chapitre des systèmes d'écritures dits à notation positionnelle. Sauf dans la partie 4, on ne parlera que des nombres entiers positifs auxquels on fera référence par le mot « nombres ».

# 1 Le système décimal

**Définition :** Le système décimal comporte un alphabet de dix chiffres : 0,1,2,3,4,5,6,7,8 et 9 avec lequel on forme des mots (par exemple 2314). Pour un mot donné on donne alors une interprétation en terme de nombre selon les principes suivants :

— Les chiffres 0,1,2,3,4,5,6,7,8 et 9 représentent, seuls, les dix premiers nombres (en partant de zéro).

— En lisant un mot de droite à gauche, on lit des nombres d'unités (à droite), de dizaines, de centaines, etc.

Par exemple, le mot 2314 s'interprète dans le système décimal comme :

— quatre unités ;

— une dizaine (paquet de dix)

— trois centaines (paquets de dix fois dix)

— deux milliers (paquets de dix fois dix fois dix)

Ainsi, en remarquant que 10 correspond exactement à une dizaine, on peut écrire :

$$2314 = 2 \times 10^3 + 3 \times 10^2 + 1 \times 10^1 + 4$$

**Remarque :** On dispose d'un certain nombre d'algorithmes liés à cette écriture, citons-en quelques uns :

— addition et soustraction posée ;

— multiplication posée, division euclidienne ;

— comparaison.

**Propriétés :** Soit $k \in \mathbb{N}$. On a :

1. $10^k = 10...0$ (un suivi de $k$ zéros).
2. $10^k - 1 = 9...9$ ($k$ neufs).
3. Avec des mots de $k$ chiffres, on représente en base 10 tous les nombres entre 0 et $10^k - 1$, soit un total de $10^k$ nombres différents.
4. Multiplier par $10^k$ consiste à ajouter $k$ zéros à droite.

1. Décomposer les nombres 230, 3564 et 100021 en puissances de dix.
2. Écrire 99, 9999 et 99999 sous la forme $10^k - 1$.
3. Dans le système décimal, combien de nombres peut on représenter avec 5 chiffres ? De combien à combien ?

# 2 Le système binaire

## 2.1 Principe

**Définition :** Pour comprendre le système binaire, il faut avoir en tête que le principe est exactement le même que celui du système décimal, mais avec (dramatiquement) moins de chiffres. Ainsi, le système binaire est constitué d'un alphabet de deux chiffres appelés ici des bits (binary digits) : 0 et 1 avec lequel on forme des **mots binaires** (par exemple 10011).

On interprète ces mots comme des nombres selon des principes similaires (mais deux va remplacer dix) :

— Les chiffres 0 et 1 représentent respectivement les nombres zéro et un.

— En lisant un mot de droite à gauche, on lit des nombres d'unités (à droite), de paquets de deux, de paquets de quatre, etc.

Ainsi, le mot 10011 s'interprète comme :

— une unité

— un paquet de deux

— zéro paquet de quatre (deux fois deux)
— zéro paquet de huit (deux fois deux fois deux)
— un paquet de seize (deux fois deux fois deux fois deux)

Ce qu'on peut aussi écrire, en utilisant le système décimal :

$$(10011)_2 = 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1$$

**Remarque** A partir de maintenant, on dispose de plusieurs manières d'écrire les nombres et il faut donc préciser le système utilisé car sans cela, on ne sait pas faire la différence entre 11 (onze décimal) et 11 (trois binaire). On adopte donc la convention suivante pour interpréter un mot $m$ :

— si rien n'est précisé, $m$ est à interpréter en base 10 ;
— sinon, on précise la base $b$ utilisée en écrivant $(m)_b$ (on en verra d'autres).

Ainsi, 11 correspond à onze et $(11)_2$ correspond à trois.

**Remarque** On dispose d'un certain nombre d'algorithmes liés à cette écriture (les mêmes qu'en base 10) :

— addition et soustraction posée ;
— multiplication posée, division euclidienne ;
— comparaison.

**Propriétés** Soit $k \in \mathbb{N}$. On a :

1. $2^k = (10...0)_2$ (un suivi de $k$ zéros)
2. $2^k - 1 = (1...1)_2$ ($k$ uns)
3. Avec des mots de $k$ bits, on représente en base 2 tous les nombres entre 0 et $2^k - 1$, soit un total de $2^k$ nombres différents.
4. Multiplier par $2^k$ consiste à ajouter $k$ zéros à droite.

1. Décomposer les nombres $(101)_2$, $(10101)_2$ et $(1110)_2$ en puissances de deux.
2. Écrire 11111 et 111 sous la forme $2^k - 1$.
3. Dans le système binaire, combien de nombres peut on représenter avec 5 bits ? De combien à combien ?

## 2.2 De la base 2 à la base 10

**Passer de la base 10 à la base 2 :** Pour avoir l'écriture en base dix du nombre $x = (a_k...a_0)_2$ on effectue le calcul :

$$x = a_k \times 2^k + ... + a_2 \times 2^2 + a_1 \times 2^1 + a_0$$

**Exemples**

— $(100)_2 = 1 \times 2^2 + 0 \times 2^1 + 0 = 4$
— $(110)_2 = 1 \times 2^2 + 1 \times 2^1 + 0 = 6$
— $(1001)_2 = 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 = 9$

**Premières puissance de deux** :

| $k$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $2^k$ | 1 | 2 | 4 | 8 | 16 | 32 | 64 | 128 | 256 | 512 | 1024 |

**Une astuce pour aller (parfois) plus vite :** En utilisant la propriété $2^k - 1 = (1...1)_2$, on peut chercher à compléter une écriture binaire comportant beaucoup de 1 pour obtenir 1...1. Par exemple déterminons l'écriture décimale de $x = (11110110)_2$ :

$$x + (1001)_2 = (11111111)_2 = 2^8 - 1 = 255$$
$$(1001)_2 = 2^3 + 1 = 9$$
$$x = 255 - 9 = 246$$

1. Écrire $(101)_2$, $(10101)_2$ et $(1110)_2$ en base 10.
2. Écrire $(1111)_2$ et $(1111\ 1111)_2$ en base 10.
3. Écrire $(1101)_2$ et $(1110\ 0111)_2$ en base 10 en s'aidant des égalités précédentes.

## 2.3 De la base 10 à la base 2

**Compter en binaire**

- $0 = (0)_2$
- $1 = (1)_2$
- $2 = (10)_2$
- $3 = (11)_2$
- $4 = (100)_2$
- $5 = (101)_2$
- $6 = (110)_2$
- $7 = (111)_2$
- $8 = (1000)_2$
- $9 = (1001)_2$
- $10 = (1010)_2$
- $11 = (1011)_2$
- $12 = (1100)_2$
- $13 = (1101)_2$
- $14 = (1110)_2$
- $15 = (1111)_2$
- $16 = (10000)_2$
- $17 = (10001)_2$
- $18 = (10010)_2$
- $19 = (10011)_2$
- $20 = (10100)_2$
- $21 = (10101)_2$
- $22 = (10110)_2$
- $23 = (10111)_2$
- $24 = (11000)_2$
- $25 = (11001)_2$
- $26 = (11010)_2$
- $27 = (11011)_2$
- $28 = (11100)_2$
- $29 = (11101)_2$
- $30 = (11110)_2$
- $31 = (11111)_2$

**Outil (division euclidienne) :** Soient $a$ et $b$ deux entiers, $b \neq 0$. Il existe un unique couple $(q, r)$ d'entiers tels que $0 \leq r \leq b - 1$ et :

$$a = bq + r$$

Dans ce contexte, $q$ et $r$ sont appelés respectivement quotient et reste de la division euclidienne de $a$ par $b$.

**Exemples :** Effectuons quelques divisions euclidiennes à la main :

- Division de 32 par 5 : $32 = 6 \times 5 + 2$, quotient : 6, reste : 2.
- Division de 181 par 3 : $181 = 60 \times 3 + 1$, quotient : 60, reste : 1.
- Division de 160 par 12 : $160 = 13 \times 12 + 4$, quotient : 13, reste : 4.

En pratique et pour l'instant, on calculera $q$ en prenant la partie entière de $a/b$ et $r$ en écrivant $r = a - qb$.

**Passer de la base 10 à la base 2 :** Pour avoir l'écriture en base deux d'un nombre $a$ écrit en base dix on effectue une suite de divisions euclidiennes par deux en commençant avec $a$ puis avec les quotients successifs, jusqu'à trouver un quotient nul. L'écriture en base deux est alors la suite des restes obtenus, lue à l'envers. En écrivant :

$$\begin{aligned} a &= 2q_0 + r_0 \\ q_0 &= 2q_1 + r_1 \\ &... \\ q_{k-1} &= 2q_k + r_k \end{aligned}$$

avec $q_k = 0$, on peut en déduire que :

$$a = (r_k ... r_0)_2$$

**Exemple :** Écriture en base deux de 75 :

$$\begin{aligned} 75 &= 2 \times 37 + 1 \\ 37 &= 2 \times 18 + 1 \\ 18 &= 2 \times 9 + 0 \\ 9 &= 2 \times 4 + 1 \\ 4 &= 2 \times 2 + 0 \\ 2 &= 2 \times 1 + 0 \\ 1 &= 2 \times 0 + 1 \end{aligned}$$

On trouve $75 = (1001011)_2$.

**Explications de la méthode** La première division nous dit que $a$ peut se décomposer en $q_0$ paquets de deux plus $r_0$ unités. La deuxième division nous permet de décomposer les $q_0$ paquets de deux en $q_1$ paquets de quatre plus $r_1$ paquets de deux. La troisième division nous permet de décomposer les $q_1$ paquets de quatre en $q_2$ paquets de huit plus $r_2$ paquets de quatre. On continue ainsi jusqu'à ne plus pouvoir composer de paquets plus grands.

1. Écrire les nombres suivants en base 2 :
   (a) 123

   (b) 246
   (c) 492

   Que remarque-t-on ?

2. En utilisant l'égalité $(1...1)_2 = 2^k - 1$, écrire les nombres suivants en base 2 :
   (a) 127
   (b) 1023
   (c) 1022

# 3 Généralisation et cas particuliers

## 3.1 Base $b \geq 2$

Pour avoir un système de numération de base $b \geq 2$, il suffit de reprendre les principes vus plus haut avec un alphabet de $b$ chiffres représentant les nombres de 0 à $b-1$. Un mot est alors interprété comme (de droite à gauche) les nombres d'unités, paquets de $b$, paquets de $b^2$, etc.

Si $b \leq 10$, on prend par convention les chiffres arabes jusque $b-1$. Si $10 < b \leq 35$, on complète à partir de 10 par des lettres ($a = 10$, $b = 11$, $c = 12$, etc).

Ce qui a été développé en bases 2 et 10, et notamment les méthodes de passage d'une base à l'autre, peut se généraliser pour une base $b \geq 2$ quelconque.

> *Passer de la base b à la base 10 :* Pour avoir l'écriture en base dix du nombre $x = (a_k...a_0)_b$ on effectue le calcul :
>
> $$x = a_k \times b^k + ... + a_2 \times b^2 + a_1 \times b^1 + a_0$$

**Exemples :**

— $(2423)_5 = 2 \times 5^3 + 4 \times 5^2 + 2 \times 5^1 + 3 = 358$

— $(1a2b)_{12} = 1 \times 12^3 + 10 \times 12^2 + 2 \times 12^1 + 11 = 3203$

1. Décomposer les nombres $(123)_5$, $(142)_6$ et $(ab123)_{12}$ en puissances de la base correspondante.
2. En déduire leur écriture en base 10.
3. Écrire $(bbbbb)_{12}$ et $(444)_5$ sous la forme $b^k - 1$.
4. En déduire leur écriture en base 10.
5. Combien de nombres peut on représenter avec 4 chiffres en base 16 ? De combien à combien ?

> **Passer de la base 10 à la base $b$ :** Pour avoir l'écriture en base $b$ d'un nombre $a$ écrit en base dix on effectue une suite de divisions euclidiennes par $b$ en commençant avec $a$ puis avec les quotients successifs, jusqu'à trouver un quotient nul. L'écriture en base deux est alors la suite des restes obtenus, lue à l'envers. En écrivant :
>
> $$\begin{aligned} a &= bq_0 + r_0 \\ q_0 &= bq_1 + r_1 \\ &... \\ q_{k-1} &= bq_k + r_k \end{aligned}$$
>
> avec $q_k = 0$, on peut en déduire que :
>
> $$a = (r_k...r_0)_b$$

**Exemples**

— Écriture en base 5 de 131 :

$$\begin{aligned} 391 &= 5 \times 78 + 1 \\ 78 &= 5 \times 15 + 3 \\ 15 &= 5 \times 3 + 0 \\ 3 &= 5 \times 0 + 3 \end{aligned}$$

On a donc $391 = (3031)_5$.

— Écriture en base 12 de 4522 :

$$\begin{aligned} 4522 &= 12 \times 376 + 10(a) \\ 376 &= 12 \times 31 + 4 \\ 31 &= 12 \times 2 + 7 \\ 3 &= 5 \times 0 + 3 \end{aligned}$$

On a donc $4522 = (374a)_{12}$.

1. Écrire les nombres suivants dans la base demandée :
   (a) 123 en base 5
   (b) 987 en base 12
2. Calculer les premières puissances de 5 et en déduire l'écriture de 3124 en base 5.

### 3.2 Généralisation et cas particuliers : le cas de la base 16

En base 16, les chiffres sont 0, 1, ... , 9, a(10), b(11), c(12), d(13), e(14) et f(15). En base 2 on peut donc écrire :

— $(0)_{16} = (0000)_2$
— $(1)_{16} = (0001)_2$
— $(2)_{16} = (0010)_2$
— $(3)_{16} = (0011)_2$
— $(4)_{16} = (0100)_2$
— $(5)_{16} = (0101)_2$
— $(6)_{16} = (0110)_2$
— $(7)_{16} = (0111)_2$
— $(8)_{16} = (1000)_2$
— $(9)_{16} = (1001)_2$
— $(a)_{16} = (1010)_2$
— $(b)_{16} = (1011)_2$
— $(c)_{16} = (1100)_2$
— $(d)_{16} = (1101)_2$
— $(e)_{16} = (1110)_2$
— $(f)_{16} = (1111)_2$

**Bases 2 et 16 :** Chaque chiffre de la base 16 peut être encodé par un et un seul mot de quatre bits et inversement. Lorsqu'on veut passer de la base 16 à la base 2, il suffit donc de traduire chaque chiffre par les 4 bits correspondant sans changer l'ordre général. Pour aller de la base 2 à la base 16, on regroupe les bits par groupes de 4 (en partant de la gauche) et on traduit.

**Exemples**

— $(ea13)_{16} = (\boxed{1110}\boxed{1010}\boxed{0001}\boxed{0011})_2$
— $(\boxed{1101}\boxed{0010}\boxed{1100}\boxed{0011})_2 = (d2c3)_{16}$

Ce type de propriété provient du fait que 16 est une puissance de 2, elle émerge d'ailleurs entre chaque couple de bases ($b$, $b^k$). En pratique, la base 16 peut être vu comme une compactification de la base 2, divisant le nombre de chiffres nécessaires à l'écriture d'un nombre par 4.

1. Écrire les nombres $(1111\ 1010\ 0010)_2$ et $(1000\ 1011\ 1011)_2$ en base 16.
2. Écrire les nombres $(abcd)_{16}$ et $(f4a3)_{16}$ en base 2.

## 4 Représentation des entiers relatifs, méthode du complément à deux

### 4.1 Principe et explications

Mathématiquement parlant, on peut tout à fait écrire $-(1011)_2$ pour désigner l'entier relatif $-11$. Cependant, dans le but de stocker informatiquement des nombres entiers pouvant être positifs ou négatifs, il nous faut une méthode (un encodage) permettant de représenter de manière cohérente les deux.

Une première idée, un nombre de bits étant fixé pour la représentation, pourrait consister à réserver le premier bit de l'écriture au signe du nombre ($0 = +$, $1 = -$) et le reste à sa valeur absolue (sa partie non signée). Par exemple, sur 4 bits 0101 pourrait représenter $+(101)_2 = +5$ et 1101 pourrait représenter $-(101)_2 = -5$.

Malheureusement, cette méthode naïve possède deux inconvénients :

1. Elle représente 0 de deux manières différentes puisque $+0 = -0$.
2. Elle ne s'adapte pas bien aux algorithmes d'addition ou de soustraction essentiels au fonctionnement d'un ordinateur.

La méthode couramment utilisée est appelée la **méthode du complément à deux**. En utilisant $n$ bits :

1. On y représente les entiers relatifs de $-2^{n-1}$ à $2^{n-1}-1$.
2. Si l'entier $x$ à représenter est positif, alors on donne son écriture binaire (dont le premier bit sera 0).
3. Si l'entier $x$ à représenter est strictement négatif, alors on donne l'écriture binaire de $2^n + x$ (dont le premier bit sera 1).

**Exemples :** Sur 8 bits :

1. On représente les nombres de $-2^7 = -128$ à $2^7 - 1 = 127$
2. 12 est représenté par l'écriture binaire de 12, soit donc 0000 1100.
3. -5 est représenté par l'écriture binaire de $2^8 - 5 = 256 - 5 = 251$ soit donc 1111 1011.

Pour déchiffrer un nombre $x$ représenté par cette méthode par un mot binaire $m$, il suffit de faire le raisonnement dans l'autre sens. Sur $n$ bits, cela donne :

1. Si le premier bit de $m$ est un 0, c'est que $x$ est positif donc $m$ est l'écriture binaire de $x$, soit :
$$x = (m)_2$$
2. Si le premier bit de $m$ est un 1, c'est que $x$ est négatif donc $m$ est l'écriture binaire de $2^n + x$, soit $(m)_2 = 2^n + x$ et :
$$x = (m)_2 - 2^n$$

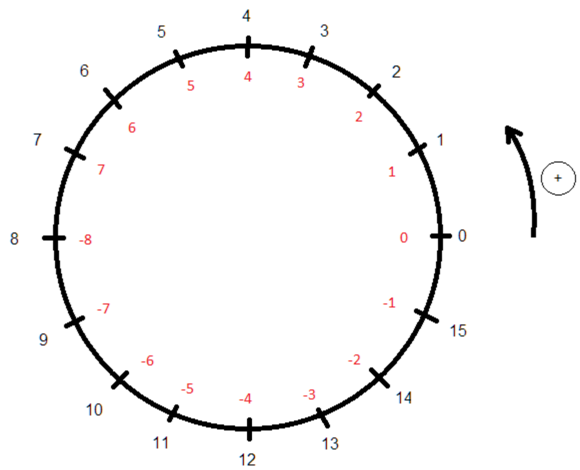
**Exemples :** Sur 8 bits :

1. 0101 1000 commence par zéro et représente donc un nombre $x$ positif, ici on a donc :
$$x = (0101\ 1000)_2 = 88$$
2. 1100 1100 commence par un et représente donc un nombre $x$ négatif, avec $(1100\ 1100)_2 = x + 2^8$ soit :
$$x = (1100\ 1100)_2 - 2^8 = 204 - 256 = -52$$

Trouver la représentation en complément à deux 8 bits de :

1. 14 et 125 ;
2. -16 et -85.

Pour comprendre la logique derrière cette méthode, il faut organiser les nombres en cercle. Voyons ce que cela donne sur 4 bits ($2^4 = 16$ nombres représentables) :




Sur le cercle on a représenté :

— les seize entiers naturels de 0 à 15 ;

— les seize entiers relatifs de -8 à +7.

| $x$ (base 10) | nombre associé (base 10) | nombre associé (base 2) |
|---|---|---|
| 0 | 0 | 0000 |
| 1 | 1 | 0001 |
| 2 | 2 | 0010 |
| 3 | 3 | 0011 |
| 4 | 4 | 0100 |
| 5 | 5 | 0101 |
| 6 | 6 | 0110 |
| 7 | 7 | 0111 |
| $-8$ | 8 | 1000 |
| $-7$ | 9 | 1001 |
| $-6$ | 10 | 1010 |
| $-5$ | 11 | 1011 |
| $-4$ | 12 | 1100 |
| $-3$ | 13 | 1101 |
| $-2$ | 14 | 1110 |
| $-1$ | 15 | 1111 |

FIGURE 1 – Méthode du complément à deux, 4bits

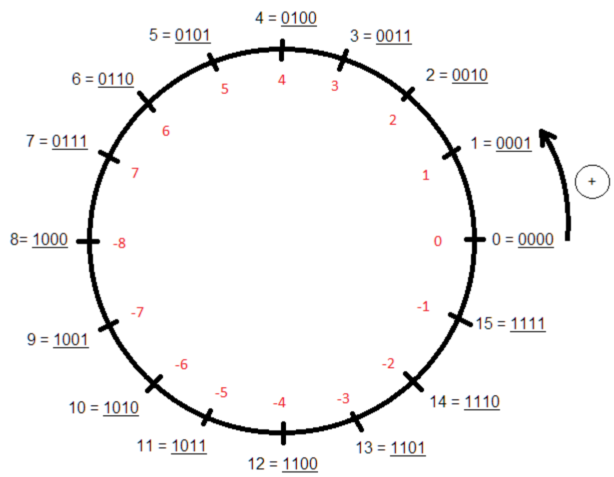
Pour se déplacer sur le cercle, on tourne dans le sens trigonométrique (anti-horaire).

Le fait d'être sur un cercle amène à associer à chaque nombre négatif un autre nombre, lui positif. Ainsi, reculer de 7 crans sur le cercle revient à y avancer de 10 crans, d'où l'association $-7 \equiv +10$. On parle ici de congruence modulo 16.

Plus généralement, reculer de $k$ cases sur le cercle, peut être vu comme y avancer de $16 + (-k)$ cases.

C'est la propriété que l'on utilise dans la méthode du complément à deux quand le nombre à représenter est négatif : on associe à $x < 0$ le nombre $2^n + x$ qui est à la même place sur un cercle à $2^n$ crans.

Reprenons le cercle, en écrivant les nombre positifs en base 2. On peut alors en déduire la représentation 4 bits des entiers de -8 à +7 par la méthode du complément à deux.




Dessiner le cercle précédent dans le cas de la représentation 3 bits et en déduire les représentations 3 bits des nombres de -4 à 3.

## 4.2 Passage à l'opposé par inversion des bits

On développe maintenant une méthode permettant de passer de la représentation binaire d'un nombre à celle de son opposé en complément à deux.

Soient donc $n > 0$ et $m$ un mot binaire $n$ bits représentant un entier en base 2 en complément à deux.

**Exemple :** $n = 8$, $m =' 0101\ 0110'$

On définit $\overline{m}$ le mot obtenu en inversant les bits de $m$.

**Exemple :** $\overline{m}$ = '1010 1001'

Si on voit $m$ et $\overline{m}$ comme des représentations d'entiers positifs, on peut les additionner et, leur bits étant inversés, on se retrouve avec un mot comportant exactement $n$ bits 1, soit l'écriture binaire de $2^n - 1$.

**Exemple :**

$$\begin{array}{cccccccccc} & 0 & 1 & 0 & 1 & & 0 & 1 & 1 & 0 \\ + & 1 & 0 & 1 & 0 & & 1 & 0 & 0 & 1 \\ \hline = & 1 & 1 & 1 & 1 & & 1 & 1 & 1 & 1 \end{array}$$

**Propriété :** Si $m$ est un mot binaire $n$ bits, alors $(m)_2 + (\overline{m})_2 = 2^n - 1$.

Propriété que l'on peut réécrire :

$$(\overline{m})_2 + 1 = 2^n - (m)_2$$

D'où la méthode suivante, en remarquant que $2^n - (m)_2$ correspond à la représentation en complément à deux de $-(m)_2$ :

> Si $x$ est un nombre entier compris entre $-2^{n-1}$ et $2^{n-1} - 1$ représenté en complément à deux sur $n$ bits par le mot binaire $m$, alors on peut représenter $-x$ en suivant les étapes suivantes :
> — inverser les bits de $m$ ;
> — incrémenter le nombre obtenu (lui ajouter 1).

Cette méthode est utile :

— lorsqu'on connaît déjà la représentation de $x$ et qu'on veut celle de $-x$ (dans un ordinateur, cela sera toujours le cas puisque tous les nombres entiers vont être stockés de la sorte) ;

— lorsque $x > 0$ est un petit nombre car il sera alors plus simple de :
  1. trouver l'écriture binaire de $x$ (petit) ;
  2. passer à l'opposé par la méthode précédente ;

  plutôt que de :
  1. calculer en décimal $2^n - x$ ;
  2. trouver l'écriture binaire de ce nombre (grand).

**Exemples :** sur 8 bits :

— Soit $x$ représenté par 1100 1011. Cherchons la représentation de $-x$ :
  1. Inversion des bits : 0011 0100 ;
  2. Incrémentation : 0011 0101.

  $-x$ est représenté par 0011 0101.

— Cherchons la représentation de $-12$ :
  1. $12 \geq 0$ est représenté par 0000 1100 (car $12 = 8 + 4$) ;
  2. Inversion des bits : 1111 0011 ;
  3. Icrémentation : 1111 0100.

  $-12$ est représenté par 1111 0100.

## 4.3 Adaptation à l'addition binaire

Nous allons voir ici en quoi la représentation par la méthode du complément à deux s'adapte très bien à l'algorithme de l'addition binaire à travers quelques exemples. On se place en représentation 8 bits.

Prenons plusieurs exemples d'additions :

1. $15 + 7 = 22$
2. $7 + (-15) = (-8)$
3. $15 + (-7) = 8$
4. $(-7) + (-15) = (-22)$

Pour vérifier si ces additions fonctionnent en complément à deux, nous avons besoin des représentations de $\pm 7$, $\pm 8$ $\pm 15$ et $\pm 22$. Les voici :

| $x$ | représentation 8 bits de $x$ |
|---|---|
| 7 | 0000 0111 |
| $-7$ | 1111 1001 |
| 8 | 0000 1000 |
| $-8$ | 1111 1000 |
| 15 | 0000 1111 |
| $-15$ | 1111 0001 |
| 22 | 0001 0110 |
| $-22$ | 1110 1010 |

Effectuons les additions 1 et 2 :

1.
$$\begin{array}{cccccccccc}
 & & & & {\scriptstyle 1} & {\scriptstyle 1} & {\scriptstyle 1} & {\scriptstyle 1} & \\
 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
+ & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\
\hline
= & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0
\end{array}$$

2.
$$\begin{array}{cccccccccc}
 & & & & & {\scriptstyle 1} & {\scriptstyle 1} & {\scriptstyle 1} & \\
 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\
+ & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\
\hline
= & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0
\end{array}$$

On remarque qu'aucun problème n'apparaît, l'addition des nombres en complément à deux (même négatifs) fonctionne ici.

Effectuons les additions 3 et 4 :

3.
$$\begin{array}{cccccccccc}
 & {\scriptstyle 1} & {\scriptstyle 1} & {\scriptstyle 1} & {\scriptstyle 1} & {\scriptstyle 1} & {\scriptstyle 1} & {\scriptstyle 1} & {\scriptstyle 1} & \\
 & & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
+ & & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\
\hline
= & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0
\end{array}$$

4.
$$\begin{array}{cccccccccc}
 & {\scriptstyle 1} & {\scriptstyle 1} & {\scriptstyle 1} & {\scriptstyle 1} & & & & {\scriptstyle 1} & \\
 & & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\
+ & & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\
\hline
= & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0
\end{array}$$

Sur ces additions, un problème apparaît : on trouve une retenue finale rendant le résultat incorrect. Cependant, puisque dans la représentation choisie on ne représente les nombres que sur 8 bits, on peut choisir d'ignorer cette dernière retenue et le résultat devient correct.

**Remarque :** Le fait d'ignorer le 9-ème bit, soit $2^8$, peut aussi se comprendre sur le cercle puisqu'ajouter $2^8$ correspond à faire un tour du cercle, ce qui n'a pas d'effet.

**Attention !** Certaines additions restent problématiques. En effet, prenons l'addition $125 + 7 = 132$.

125 se représente sur 8 bits par 0111 1101 et 7 par 0000 0111, mais 132 étant plus grand que $2^7 - 1 = 127$, il ne se représente pas sur 8 bits en complément à deux. Voyons ce qu'il se passe si on effectue tout de même l'addition :

$$\begin{array}{ccccccccc}
 & {\scriptstyle 1} & {\scriptstyle 1} & {\scriptstyle 1} & {\scriptstyle 1} & {\scriptstyle 1} & {\scriptstyle 1} & {\scriptstyle 1} & \\
 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\
+ & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\
\hline
= & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0
\end{array}$$

On trouve 1000 0100, ce qui correspond bien à l'écriture binaire de 132 mais qui, dans la méthode du complément à deux, correspond à $132 - 2^8 = -124$. Cette addition est donc fausse en complément à deux.

Si on effectue une addition de nombres représentés par la méthode du complément à deux $n$ bits :

— le résultat sera correct en complément à deux ;

à condition de s'assurer que :

— le résultat de l'addition reste compris entre $-2^{n-1}$ et $2^{n-1} - 1$ afin de toujours être représentable ;

— on ignore l'éventuelle dernière retenue qui augmenterait le nombre de bits de la représentation.