8 – Nombres entiers relatifs

1. Le problème

On veut coder les ***nombres entiers relatifs***, c'est-à-dire les nombres 0, 1, 2 etc mais aussi -1, -2, -3 etc.
Contraintes :

- on ne pourra pas coder tous les nombres : il en existe une infinité et les ressources d'un ordinateur sont limitées ;
- il faudrait que les algorithmes soient simples ou semblables à ceux déjà vus pour les entiers naturels, en particulier :
  - convertir du décimal vers le binaire et vice-versa ;
  - reconnaître si un nombre est positif ou négatif ;
  - calculer une addition.

La méthode de codage que nous allons voir s'appelle le ***complément à deux***. À chaque fois, il faut préciser sur combien de bits on code les nombres entiers.

2. Quels nombres ?

Sur n bits, on peut coder 2^n nombres différents. Avec la méthode du complément à deux, on réserve :

- la moitié des codages aux plus petits nombres entiers naturels ;
- l'autre moitié aux nombres strictement négatifs les plus proches de 0.

Exercice 1.

1. Avec cette méthode, combien de nombres entiers peut-on coder sur 2 bits ? Lesquels ? ....
....
2. Et sur 3 bits ? ....
3. Et sur n bits ? (Où n est un nombre entier naturel quelconque.) ....

3. Codage des entiers positifs

En complément à deux, les entiers positifs sont codés avec la méthode usuelle des entiers naturels.
Si besoin, on complète l'écriture en ajoutant des zéro à gauche pour avoir le nombre de bits voulus.

Exercice 2.

1. Écrivez, avec la méthode du complément à deux sur 2 bits, tous les nombres entiers positifs possibles.

   ....

2. Écrivez, avec la méthode du complément à deux sur 3 bits, tous les nombres entiers positifs possibles.

   ....

3. Écrivez, avec la méthode du complément à deux sur 4 bits, tous les nombres entiers positifs possibles.

   ....

   ....

4. Codage des entiers négatifs : méthode visuelle

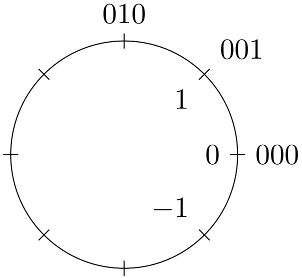
Pour coder les entiers négatifs de façon visuelle avec le complément à deux sur n bits :

1. On place 2^n points régulièrement espacés sur un cercle.
2. On numérote ces points avec le codage binaire sur n bits, dans l'ordre : 0...0 puis 0...01, etc, en tournant.
3. On numérote la moitié des points, en partant de 0...0 et en tournant dans le même sens, mais cette fois en décimal : $(0...0)_{\text{bin}}$ correspond à $(0)_{\text{déc}}$.
4. On numérote l'autre moité des points avec les négatifs : -1, -2, etc, mais cette fois-ci après 0 et en tournant dans l'autre sens.

La correspondance décimal/binaire obtenue est celle du complément à deux sur n bits.

Exercice 3.

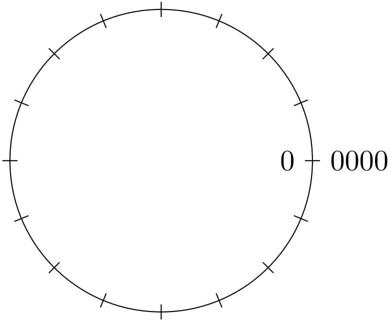
Avec la méthode donnée précédemment, complétez le cercle, puis le tableau ci-dessous, pour le complément à deux sur 3 bits (donc avec $2^3 = 8$ points).



Binaire	000	001	010	011	100	101	110	111
Décimal								

Exercice 4.

Avec la méthode donnée précédemment, complétez le cercle, puis le tableau ci-dessous, pour le complément à deux sur 4 bits.



Bin.	0000	0001	0010	0011	0100	0101	0110	0111
Déc.								

Bin.	1000	1001	1010	1011	1100	1101	1110	1111
Déc.								

5. Coder un nombre négatif : méthode 1

Pour coder un nombre entier négatif en complément à deux :

1. on code son opposé comme d'habitude en complétant si besoin avec des zéro à gauche ;
2. on inverse tous les chiffres : les 0 deviennent 1 et les 1 deviennent 0 ;
3. on ajoute 1.

Exemple 1.

Codons -62 en complément à deux sur un octet :

1. on code 62 : 00111110 ;
2. on inverse : 11000001 ;
3. on ajoute 1 : 11000010.

Exercice 5.

Coder, en complément à deux sur un octet, les nombres -8, -23, -1, -207 et -66.

6. Coder un nombre négatif : méthode 2

Un autre méthode pour coder un nombre entier négatif en complément à deux est :

1. on code son opposé comme d'habitude en complétant si besoin avec des zéro à gauche ;
2. en allant de droite à gauche :
   a. on recopie les chiffres jusqu'au premier 1 rencontré inclus ;
   b. on inverse tous les autres chiffres.

Exemple 2.

Codons -62 en complément à deux sur un octet :

On code 62 :	0	0	1	1	1	1	1	0
	!	!	!	!	!	!	$\downarrow$	$\downarrow$
Résultat :	1	1	0	0	0	0	1	0

Exercice 6.

Vérifiez cette méthode avec les nombres de l'exercice précédent.

7. Reconnaître le signe d'un nombre, trouver son opposé

Le ***bit de poids fort***, dans une écriture binaire, est celui qui est situé le plus à gauche.
Avec la méthode du complément à deux, il suffit de regarder le bit de poids fort d'un nombre pour connaître son signe :

- si le bit de poids fort est 0, le nombre est positif ou nul ;
- sinon, le nombre est négatif.

Exemple 3.

Le nombre codé par 10010100 en complément à deux sur un octet est négatif.

Exercice 7.

1. Calculer l'opposé du nombre codé par 10010100 en complément à deux sur un octet grâce à la méthode 2 ci-dessus.
2. Calculer l'écriture décimale de cet opposé.
3. En déduire l'écriture décimale du nombre donné au départ.

8. Addition

Exercice 8.

Compléter les tables d'addition suivantes, pour des nombres codés sur 2 bits par le complément à deux. Commencez par calculer les additions en écriture décimale (table de gauche), puis codez les résultats en binaire (table de droite).

TABLE – Écriture décimale

$+$	0	1	-2	-1
0				
1				
-2				
-1				

TABLE – Écriture binaire

$+$	00	01	10	11
00				
01				
10				
11				

Calculer avec l'algorithme usuel : $01 + 01$, puis $01 + 10$, puis $10 + 10$ et $11 + 11$. Comparer avec les résultats des tables ci-dessus. Que remarque-t-on ?

Pour ajouter deux nombres écrits en complément à deux sur n bits :

1. on les additionne comme les nombres naturels ;
2. si un bit supplémentaire apparaît, on l'ignore ;
3. on vérifie la cohérence des signes :
   3.1 si les deux nombres sont positifs, leur somme est positive ;
   3.2 si les deux nombres sont négatifs, leur somme est négative.

Si il n'y a pas cohérence des signes, l'addition n'est pas possible : il y a ***dépassement***. Cette cohérence est vérifiée avec les bits de poids forts : $0 + 0$ doit donner 0 et $1 + 1$ doit donner 1.

Exercice 9.

Codez les nombres suivants en complément à deux sur un octet : 17, 45, 86, -66 et -107.
Calculez les sommes suivantes : $17 + 45$, $45 + 86$, $45 + (-66)$, $-66 + (-107)$, d'abord en écriture décimale, puis en utilisant les écritures binaires des nombres à additionner.