

Chapitre 3

Représentation d'un entier relatif

Exercice 10

Représenter dans un tableau à 3 colonnes, toutes les valeurs de base en hexadécimal et les valeurs correspondantes en décimal et en binaire sur 4 bits. Prévoir de rajouter plusieurs colonnes en plus.

Un entier relatif est un entier qui peut être positif, nul ou négatif : $n \in \mathbb{Z}$.
La solution des exercices est en fin de chapitre.

1 Première approche

Comment représenter un entier négatif en informatique ? En général, la première idée qui vient est d'utiliser le bit de poids fort comme bit de signe.

Exercice 11

1. Reprendre le tableau de l'exercice précédent et dans une nouvelle colonne, regarder comment on pourrait noter les entiers si le bit de gauche est le bit de signe.
2. Combien y a-t-il de zéro ?
3. Est-ce qu'avec cette convention, on retrouve le fait que $2 + (-2) = 0$?

2 Deuxième approche

Exercice 12

En réalité, on a une contrainte matérielle forte : on doit utiliser le même circuit logique (appelé additionneur) pour faire des calculs sur les entiers, qu'ils soient relatifs ou non. Donc, on va prendre le problème à l'envers.

1. On considère l'entier $1_{(10)} = 0001_{(2)}$ codé en binaire sur 4 bits. Quel nombre binaire, auquel on ajouterait $0001_{(2)}$ donnerait 0 ?
2. Dans une nouvelle colonne du tableau, noter les valeurs de 1 et -1.
3. Refaire le même raisonnement pour -2, puis les autres valeurs.

Cette méthode s'appelle le **complément à deux**.

Représentation d'un entier relatif

Un entier relatif n est représenté sur N bits par un entier naturel :

- si $n \geq 0$, alors n est représenté par la valeur n
- sinon, n est représenté par la valeur $2^N + n$

Avec cette manière de faire,

- il y a un seul 0 représenté en binaire par N bits à 0
- on utilise le même circuit électronique pour additionner des entiers positifs ou négatifs, on va voir qu'une soustraction correspond à 2 additions, on préserve les multiplications, ...
- il se trouve que le premier bit (qui n'est pas forcément le bit de poids fort) est le bit de signe, mais c'est une conséquence du choix du complément à 2 et non un choix imposé initialement

Représentation binaire d'un entier relatif : méthodologie

Si $n \geq 0$, alors on cherche sa représentation en binaire sur N bits.

Si $n < 0$,

- on cherche la représentation en binaire sur N bits de $-n$ qui est positif
- on complémente le nombre binaire trouvé à 2 : $0 \leftrightarrow 1$, un bit nul passe à 1 et vice-versa
- on ajoute 1 et on oublie l'éventuelle retenue sur le N ième bit

Exercice 13

1. Quelle est la représentation de $-1_{(10)}$ en binaire sur 8 bits ?
2. Quel est le plus grand entier positif que l'on peut représenter sur N bits en complément à 2 ?
3. Idem pour l'entier négatif le plus petit ?
4. Faire un schéma circulaire représentant les entiers relatifs sur 4 bits.
5. Refaire sur 8 bits ce qu'on a fait sur 4 bits en complément à deux.
6. Faire un schéma circulaire représentant les entiers relatifs sur 8 bits.

3 Dépassement de capacité

Quand une mémoire est encodée sur N bits (64 bits actuellement), l'éventuelle retenue sur le premier bit est perdue quoi qu'il arrive. Cela a pour conséquence des dépassements de capacité (*overflow* en anglais)

Exercice 14

On reprend l'exemple du début de ce chapitre avec le tableau et l'encodage sur 4 bits (dont 1 bit de signe) d'un entier relatif.

1. Calculer $6+3$ en binaire. A quoi correspond le résultat en complément à 2 ?
2. Calculer $(-7)+(-3)$. A quoi correspond ce résultat en complément à 2 ?
3. Comment peut-on facilement repérer un dépassement de capacité en regardant le signe des nombres ?

4 Résumé

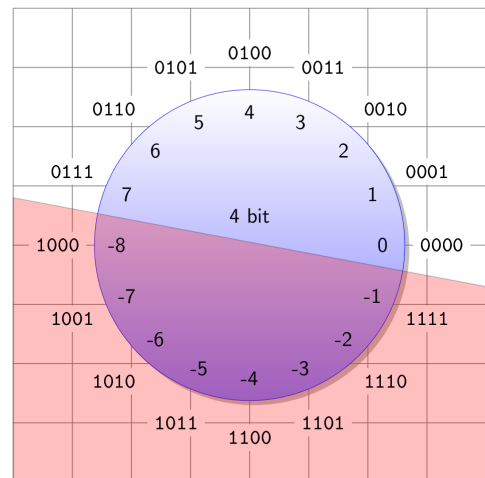
En informatique, un entier qui peut être négatif est un **entier signé**, tandis qu'un entier naturel sera qualifié de **entier non signé**.

En complément à deux, un entier signé est représenté par un entier non signé. On utilise l'indication $2cN$ en indice pour préciser qu'on a affaire à un entier signé représenté en complément à 2 sur N bits. Par exemple, $-5_{(10)}$ correspond à $1111\ 1011_{(2c8)}$.

Sachant qu'on a un entier non signé codé en complément à 2 sur N bits, comment fait-on pour retrouver l'entier signé correspondant ? On verra ça en DS !

5 Correction des exercices

hexa	décimal	binaire	1ère approche	2ème approche
0	0	0000	0	0
1	1	0001	1	1
2	2	0010	2	2
3	3	0011	3	3
4	4	0100	4	4
5	5	0101	5	5
6	6	0110	6	6
7	7	0111	7	7
8	8	1000	-0	-8
9	9	1001	-1	-7
A	10	1010	-2	-6
B	11	1011	-3	-5
C	12	1100	-4	-4
D	13	1101	-5	-3
E	14	1110	-6	-2
F	15	1111	-7	-1



1ère approche

- Dans la 1ère approche, il y a 2 zéros.
- On remarque que $2 + (-2)$ correspond à $0010 + 1010 = 1100$ correspond à -4 dans cette approche.

2ème approche

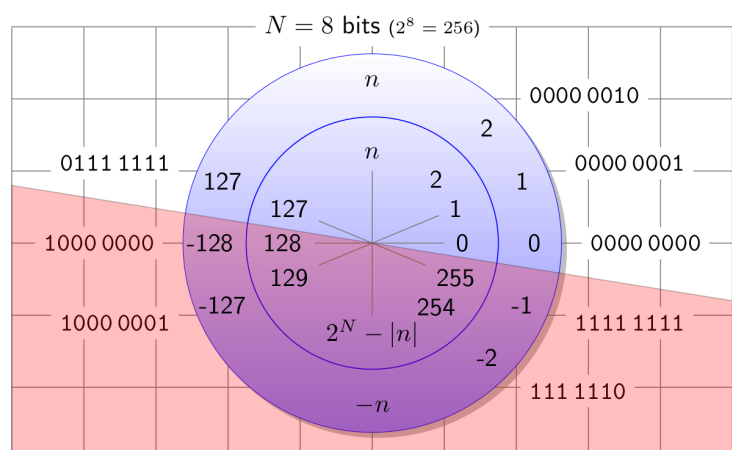
- $0001 + \text{????} = 0000 : 0001 + 1111 = (1)0000$
- on complète la colonne de la 2ème approche

Représentation de -1 en complément à 2 : on applique la méthode

- l'opposé de -1 est 1, soit en binaire sur 8 bits : 0000 0001
- on complémente à 2, soit 1111 1110
- on ajoute 1 + oubli éventuelle retenue : 1111 1111

Valeurs extrémales représentables sur N bits en complément à deux : on représente pour moitié des valeurs positives à partir de 0 et autant de valeurs négatives à partir de -1. Sur N bits, on code 2^N valeurs donc la moitié, soit 2^{N-1} valeurs à partir de 0 : donc la plus grande valeur obtenue est $2^{N-1} - 1$. Côté valeurs négatives, on a aussi 2^{N-1} valeurs à partir de -1 jusqu'à -2^{N-1} .

entier non signé	représentation binaire sur 8 bits	entier signé
0	0000 0000	0
1	0000 0001	1
2	0000 0010	2
	...	
127	0111 1111	127
128	1000 0000	-128
129	1000 0001	-127
	...	
254	1111 1110	-2
255	1111 1111	-1



Les exemples ci-dessus sont donnés sur $N = 8$ bits. Si la somme de deux entiers relatifs dépasse $127 = 2^{N-1} - 1$, il y a débordement et le résultat sera faux. De même si le résultat est inférieur à $-128 = -2^{N-1}$. Une telle situation se repère facilement : les 2 opérandes ont le même signe, mais le résultat est de signe opposé.

- Calculer $6+3$ en binaire. A quoi correspond le résultat en complément à 2 ?
 $0110 + 0011 = 1001$ soit -7 en complément à deux sur 4 bits et non 9!
- Calculer $(-7)+(-3)$. A quoi correspond ce résultat en complément à 2 ?
 $1001 + 1101 = 0110$ avec l'oubli de la retenue devant puisqu'on est sur 4 bits, ce qui fait 6 en complément à 2 et non -10.
- Comment peut-on facilement repérer un dépassement de capacité en regardant le signe des nombres ?
Le résultat est d'un signe opposé à celui des nombres qu'on manipule.