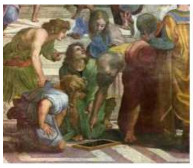

Math93.com

TD 1 - NSI Première
Représentation binaire d'un entier relatif

Activité 1 : Opération sur les nombres en binaire

Exercice 1.

1. **Représentation d'entiers naturels.**
Un ordinateur manipule des nombres binaires par groupe de 8 bits = un octet. On dispose de 8 bits, 16 bits, 32 bits, combien d'entiers naturels peut-on représenter ?

Corrigé

- Sur 8 bits on peut représenter $2^8 = 256$ entiers ;
- Sur 16 bits on peut représenter $2^{16} = 65536$ entiers ;
- Sur 32 bits on peut représenter $2^{32} = 4\,294\,967\,296$ entiers ;

2. **Addition sur 8 bits.**

2. a. Additionner sur 8 bits les nombres suivants et commenter le résultat obtenu :

$$0101\ 0001_2 \ \text{ et } \ 0111\ 0111_2$$

Aide

$$0_2 + 0_2 = 0_2 \ \text{ et } \ 1_2 + 0_2 = 1_2 \ \text{ et } \ 1_2 + 1_2 = 10_2$$

Corrigé

$$0101\ 0001_2 + 0111\ 0111_2 = 1100\ 1000_2 = 200_{10}$$

Et pour vérifier on a bien :

$$0101\ 0001_2 = 81_{10} \qquad 0111\ 0111_2 = 119_{10}$$

2. b. Faire de même avec les nombres suivants <u>sur 8 bits</u>, quel problème se pose ?

$$0101\ 0001_2 \ \text{ et } \ 1111\ 0111_2$$

Corrigé

On a :

$$0101\ 0001_2 + 1111\ 0111_2 = 1\ 0100\ 1000_2$$

donc sur 8 bits il y a dépassement et l'ordinateur ne va conserver que les 8 premiers bits soit

$$0100\ 1000$$

Vérification avec Python :

```
>>> 0b01010001
81
>>> 0b11110111
247
>>> 0b11110111+0b01010001
328
```

3. La négation sur n bits (ou complément à 1).

Définition 1 (Négation ou complément à 1)

Si x est un nombre binaire écrit en n bits, sa négation (ou complément à 1) $NON(x)$ est obtenue en transformant les 1 en 0 et les 0 en 1.
Exemple : $NON(0100\ 1001) = 1011\ 0110$

Calculer la somme d'un nombre écrit en base 2 et de son complément à 1 sur n bits sur quelques exemples. Que peut-on conjecturer ?

Remarque

Partie collaborative : discutions, et premier bilan.

Corrigé

On a toujours sur n bits :

$$x + NON(x) = \underbrace{1111\ \cdots\ 1111}_{n\ digits\ \ 1}$$

On retrouve donc le plus grand nombre entier codé sur n bits soit $2^n - 1$.

$$\left(\underbrace{1111\ \cdots\ 1111}_{n\ digits\ \ 1}\right)_2 = (2^n - 1)_{10}$$

Activité 2
Codage des nombres relatifs : une première méthode

Sur $n = 8$ bits, on a :

$$0000\ 1000_2 = 8_{10}$$

Proposer une méthode pour représenter (-8) en base 2 sur 8 bits, en n'utilisant que des 0 et des 1 sur 8 bits (pas de signe $-$ possible).

Corrigé

De nombreuses méthodes sont possibles.

Activité 3
Codage des nombres relatifs : le complément à 2

1. Donner la définition de l'opposé d'un nombre x ?

   Corrigé

   L'opposé d'un nombre x est le nombre qui ajouté à x donne 0.

2. En déduire l'opposé de 1_2 sur 8 bits.

   Corrigé

   L'opposé de 1_2 sur 8 bits est le nombre qui ajouté à 0000 0001 donne 0000 0000.

3. On utilisant le résultat conjecturé de la question 3 de l'exercice 1, que dire de l'écriture sur n bits de :

$$x + NON(x) + 1$$

   Corrigé

   $x + NON(x) + 1$ va donner sur 8 bits 0000 0000
   puisque $x + NON(x)$ va donner sur 8 bits 1111 1111.

4. On en déduit la méthode permettent d'obtenir l'opposé d'un entier en binaire.

   Corrigé

   $x + NON(x) + 1$ va donner sur 8 bits 0000 0000 donc l'opposé de x sur 8 bits est $NON(x) + 1$.
   c'est le complément à 1 + 1 que l'on nomme complément à 2.

Exercice 2. Un exemple si $n = 4$ bits.

1. Combien d'entiers positifs et négatifs peut-on représenter sur $n = 4$ bits ?

Corrigé

Sur 4 bits on peut représenter 2^4 entiers relatifs.

2. Compléter le tableau suivants et observez le lien entre **le bit de poids fort** (le premier à gauche) et le signe du nombre :

Corrigé

-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7
1000	1001	1010	1011	1100	1101	1110	1111	0000	0001	0010	0011	0100	0101	0110	0111

Exercice 3. Un exemple si $n = 8$ bits.

1. Sur l'ordinateur, utilisez la calculatrice en mode « *programmer* » et vérifier quelques résultats précédents.

2. Un exemple si $n = 8$ bits.

Après avoir donné les écritures en binaire sur 8 bits, donnez les opposés (ou **compléments à 2**) des entiers suivants (en binaire sur 8 bits) :

$$a = 1\ ;\ b = 5\ ;\ c = 10\ ;\ d = 16\ ;\ e = 32\ ;\ f = 300$$

Corrigé

- $a = 1_{10} \Rightarrow$ 0000 0001 et son complément à 2 est : 1111 1111.
- $b = 5_{10} \Rightarrow$ 0000 0101 et son complément à 2 est : 1111 1011.
- $c = 10_{10} \Rightarrow$ 0000 1010 et son complément à 2 est : 1111 0110.
- $d = 16_{10} \Rightarrow$ 0001 0000 et son complément à 2 est : 1111 0000.
- $e = 32_{10} \Rightarrow$ 0010 0000 et son complément à 2 est : 1110 0000.
- $f = 300_{10}$ ne peut pas être représenté sur 8 bits.

3. Combien d'entiers positifs et négatifs peut-on représenter sur $n = 8$ bits ?

Corrigé

Le nombre d'entiers positifs et négatifs que l'on peut représenter sur $n = 8$ bits est $2^8 = 256$.

Donner le plus petit et le plus grand en écriture décimale et binaire.

Corrigé

On peut représenter les entiers de $-128 = -2^7$ à $+127 = 2^7 - 1$.

Activité 4 : Les plus grands et plus petits entiers relatifs à coder sur n bits

1. Combien d'entiers positifs et négatifs peut-on représenter sur $n = 16$ bits, $n = 32$ bits ?

Corrigé

On peut représenter $2^{16} = 65\,536$ entiers sur 16 bits.
On peut représenter $2^{32} = 4\,294\,967\,296$ entiers sur 32 bits.

Donner le plus petit et le plus grand en écriture décimale et binaire.

Corrigé

On peut représenter les entiers de $-32\,768 = -2^{15}$ à $+32\,767 = 2^{15} - 1$ sur 16 bits.
On peut représenter les entiers de -2^{31} à $2^{31} - 1$ sur 32 bits.

2. Généralisation : reprendre la question précédente sur n bits ?

Corrigé

On peut représenter les entiers de -2^{n-1} à $2^{n-1} - 1$.

Compléments (facultatif)

1. Quel est le plus grand nombre relatif positif utilisé par une machine en 64 bits ?

2. Écrire un algorithme (en français) pour obtenir l'opposé d'un nombre binaire en complément à 2.

3. Écrire un algorithme (en français) qui demande un nombre n entier différent de 0 de bits, et un nombre relatif x (en base 10) et le convertit en binaire sur n bits. Il faut tenir compte des dépassements de capacité.

4. Écrire des algorithmes, en français et en Python permettant de passer d'un entier relatif à son écriture binaire sur n bits, et réciproquement.

Fin du devoir