



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 ?

.....

.....

.....

.....

.....

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



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

.....

.....

.....

.....

.....

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

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

.....

.....

.....

.....

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** : discussions, et premier bilan.

.....

.....

.....

.....

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).

.....

.....

.....



Remarque

| **Partie collaborative** : discussions, problèmes et critique.

.....

.....

.....

Activité 3

Codage des nombres relatifs : le complément à 2

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

.....

.....

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

.....

.....

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

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

.....

.....

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

.....

.....

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

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

.....

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

-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7
														0110	0111



Remarque
 | **Partie collaborative** : discussions et bilan sur le bit de poids fort.

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 des entiers suivants (en binaire sur 8 bits) :

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

.....

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

.....

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

.....

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 ?

.....
.....
.....
.....
.....
.....

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

.....
.....
.....
.....

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

.....
.....
.....
.....
.....
.....

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** ↪