Séance 2 semaine 19 (7 janvier 8h — 10h / 8 janvier 13h30 —17h30)

Cette séance a pour but de savoir convertir un nombre à virgule de la base 10 à la base 2 et inversement.

Nous avons fait toutes les applications du cours. La correction des exercices 1 et 10 de la feuille d'exercices a également été faites (sauf pour le groupe du vendredi soir pour lequel les exercices doivent être travaillés).

Cours :
Nous avons repris la partie sur la représentation des données par un ordinateur. Tout cela concerne le binaire, avec des 0 et des 1, représentant des impulsions électriques.

1) Lisez le reste du cours disponible sur **bouillotvincent.github.io** (chapitre 6)

2) Réalisez les applications du cours et comparez-les aux résultats donnés. Trouvez-vous bien la même chose ? Utilisez les mêmes méthodes que celles du cours. Cela est plus formateur que d'utiliser des méthodes brouillonnes et mal ficelées.

3) Dans un second temps, reprenez les corrections des exercices 1 et 10 ci-dessous.

4) Ensuite, faites l'exercice 11 de la feuille d'exercices (correction ci-dessous)

Correction Exercice 1:
a) Avec la méthode du complément à 2, convertir -14 en base 2.
On code la valeur absolue : $|-14| = 14 = 00001110.$
On fait le complément à 1 : 11110001
On ajoute 1 : 11110010
Conclusion : $-14 = 11110010$

Remarquez que la représentation de ce nombre négatif en binaire avec la méthode du complément à 2 commence par un **1**. Cela ressemble vraiment à la méthode du bit de signe (où on changeait le bit de poids fort pour indiquer le signe).

b) Vérifier votre réponse en faisant la somme de 14 et de -14 en base 2.
Additionnons :

$$\begin{array}{r} 00001110 \\ +11110010 \\ \hline 100000000 \end{array}$$

Le bit de poids fort est perdu en raison de l'overflow (au-delà des 8 bits autorisés).

c) Représentez sur 8 bits l'entier 4 puis représentez, toujours sur 8 bits, l'entier -5. Additionnez ces 2 nombres (en utilisant les représentations binaires bien évidemment), vérifiez que vous obtenez bien -1.

$4 = 00000100$

$-5 = 11111011$

On additionne pour trouver : 11111111 .

Il faut maintenant vérifier que 11111111 est bien égal à -1. Pour cela, il faut utiliser la méthode de décodage (base 2 vers base 10).

On commence par soustraire 1 : 11111110

On fait le complément à 1 : 00000001

On trouve la valeur absolue en convertissant ce nombre en base 10 : 1

Donc, comme ce nombre est négatif, on trouve $11111111 = -1$

Correction Exercice 10:

a) Quel est le plus grand entier positif que l'on peut représenter sur 8 bits ?
On remarque que les nombres relatifs sont obtenus en faisant +1 de 0 jusqu'au nombre maximum autorisé :
00000000, 00000001, 00000010, 00000011… … 01111110, 01111111
Si on refait +1, on trouve 10000000. Or ce nombre est négatif car le bit de poids fort vaut 1. Donc 01111111 est le plus grand entier positif que l'on peut représenter. En base 10 : il s'agit de 127 !

b) Quel est le plus petit entier négatif que l'on peut représenter sur 8 bits ?
On peut représenter 255 entiers autres que zéro sur 8 bits. Comme le plus grand entier est 127,on obtient le plus petit en faisant 127-255 = -128.
Pour vérifier, convertissons -128 en binaire :
valeur absolue : 128 = 10000000
Complément à 1 : 01111111
+1 : 10000000
-128 est égal à 10000000, le nombre obtenu quand on avait plus +1 depuis 01111111.

c) Quelles sont les bornes inférieure et supérieure d'un entier relatif codé sur 16 bits ?
On a trouvé principalement les bornes inférieures et supérieures sur 8 bits : un entier relatif est compris entre -2^7 et $2^7 - 1$.
De la même manière, sur 16 bits, -2^{15} et $2^{15} - 1$, soit entre -32768 et 32767.

La puissance 15 peut également se comprendre ainsi : sur 16 bits, nous avons un bit pour représenter le signe et 15 bits pour représenter les nombres. Il est donc normal que nous ayons ce 2^{15}.