1 Problématique

Un système *64 bits* peut représenter 2^{64} entiers. Pourtant le code ci-après donne la valeur maximale de l'entier représentable :

```python
import sys
sys.maxsize
```

Cette valeur correspond à $2^{63} - 1$. Un des 64 bits ne semble pas utilisé.

Comment sont représentés les entiers négatifs en mémoire ?

2 Rappel sur les additions

Considérons une machine *8 bits*. Elle peut stocker des nombres allant jusqu'à

$$2^7 - 1 = 127_{10} = 01111111_2$$

. Dans ces conditions le nombre 25 serait représenté en mémoire comme ci-après :

$$25_{10} = 00011001_2$$

Activité 1 :

1. Additionner $25 + 12$.
2. Encoder ces deux nombres en base 2.
3. Effectuer l'addition binaire de ces nombres.
4. Convertir le résultat en base 10. Le résultat correspond-il à la première question ?

3 Une représentation naïve des entiers négatifs

3.1 Bit de poids fort

Le bit le plus à gauche de la représentation n'est pour l'instant pas utilisé. C'est le **bit de poids fort**. Une première idée serait d'utiliser ce bit comme marqueur de signe :

— 0 pour un entier positif,

— 1 pour un entier négatif.

Ainsi l'entier -25 serait encodé :

$$-25_{10} = 10011001_2$$

3.2 Inconvénients de la représentation

3.2.1 Le zéro

Dans un système *8 bits* le zéro est représenté par 00000000_2. Cependant 10000000_2 se traduit par -0. Il y a donc deux représentations pour zéro.

Christophe Viroulaud

3.2.2 Erreur d'addition

Activité 2 :

1. Effectuer les mêmes étapes de l'activité 1, pour l'addition $-25+12$
2. L'addition est-elle correcte ?

4 Le complément à 2 puissance *n*

4.1 Définition

Cette représentation ne change rien pour les entiers positifs. Ainsi sur 8 bits :

0	1	1	1	1	1	1	1	=	127
0	...							=	...
0	0	0	0	0	0	1	0	=	2
0	0	0	0	0	0	0	1	=	1
0	0	0	0	0	0	0	0	=	0

Par contre la valeur $2^n - |x|$ représente l'entier négatif x. Ainsi sur 8 bits, -1 s'écrit

$$2^8 - 1 = 256 - 1 = 255_{10} = 11111111_2$$

1	1	1	1	1	1	1	1	=	-1	$2^8 - \|-1\| = 255$
1	1	1	1	1	1	1	0	=	-2	$2^8 - \|-2\| = 254$
1	...							=	...	
1	0	0	0	0	0	0	1	=	-127	$2^8 - \|-127\| = 129$
1	0	0	0	0	0	0	0	=	-128	$2^8 - \|-128\| = 128$
0	1	1	1	1	1	1	1	=	127	
0	...							=	...	
0	0	0	0	0	0	1	0	=	2	
0	0	0	0	0	0	0	1	=	1	
0	0	0	0	0	0	0	0	=	0	

4.2 Calculer le complément à 2

Voici le protocole pour une première méthode. Pour coder (-20) :

— Prendre le nombre positif 20 : 00010100
— Inverser les bits : 11101011
— Ajouter 1 : 11101100
— -20 : 11101100

Un deuxième protocole de garder tous les chiffres depuis la droite jusqu'au premier 1 (compris) puis d'inverser tous les suivants.

— Prendre le nombre positif 20 : 00010100
— Garder la partie à droite telle quelle : 00010$\underline{100}$
— Inverser la partie de gauche après le premier un : $\underline{11101}$100
— -20 : 11101100

Christophe Viroulaud