Cours Introduction en informatique, E2I1, septembre 2014

Résumé: codage des nombres.

Nombres entiers

Nombres entiers non-signés

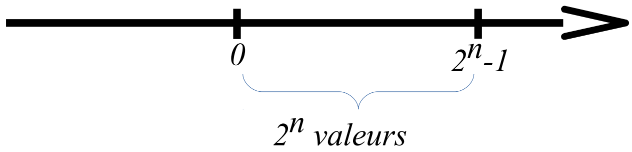
Un **mot**, un **mot binaire**: un ensemble de *n* bits utilisés pour coder une unité de données (un nombre, un caractères, etc...).
Pour le **codage** d'un nombre **entier non-signé** :

0 (zéro) est codé comme ça : $\underbrace{\boxed{0}\boxed{0}\boxed{0}\boxed{0}\ \ldots\ldots\ \boxed{0}\boxed{0}}_{n}$

La valeur max. est codée comme ça :

$\underbrace{\boxed{1}\boxed{1}\boxed{1}\boxed{1}\ \ldots\ldots\ \boxed{1}\boxed{1}}_{n}$

Cette valeur maximale vaut 2^n-1 On peut donc coder 2^n valeurs non-signés.



Nombres entiers signés.

Codage le plus utilisé : **complément à deux**

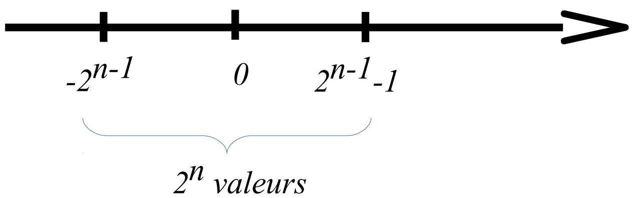
Pour coder zéro : $\underbrace{\boxed{0}\boxed{0}\boxed{0}\boxed{0}\ \ldots\ldots\ \boxed{0}\boxed{0}}_{n}$

Pour coder les nombres de 0 jusqu'à $2^{n-1}-1$: la même chose que pour les nombres non-signés, sur *n-1* bits de poids faible.
Pour coder les nombres de -2^{n-1} jusqu'à -1, on utilise un algorithme suivant:

- 1) On code d'abord la valeur absolue, comme un nombre positif
- 2) On complémente tous les bits de ce nombre
- 3) On additionne 1 à ce nombre et on rejette (tronque) la retenue éventuelle sur le bit n+1
  Exemple pour x=-5 et n=8, l'étape 1) donne 0000 0101, l'étape 2) donne 1111 1010, l'étape 3 donne 1111 1011.

Valeurs négatives « particulières » :

- -1=1111...1111 (tous les bits ont une valeurs de 1)
- Valeur minimale négative : 1000...000 (tous les bits sont à zéro sauf le premier.



Exercices.

1) Notez en format décimal les nombres suivants, encodé en format « entier signés à complément à deux » sur 8 bits:

1111 0000
0011 0011
1000 0010

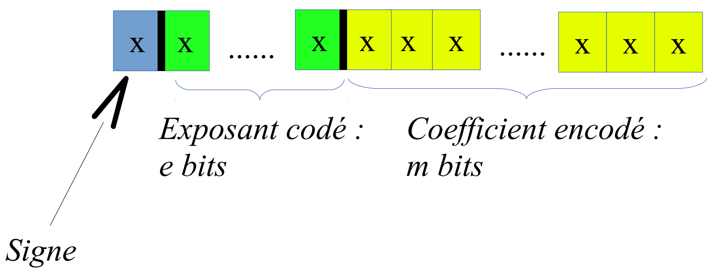
2) Encodez en format « entier signés à complément à deux », sur 8 bits, les nombres:
10
-100
-122
129

Nombres à virgule flottante: la norme IEEE 754 (source : Wikipedia, article sur IEEE 754).

Principe d'encodage.

Deux types de nombres: **normalisés** et **dénormalisés**.
Nombres normalisés : ceux qui contiennent un seul chiffre non-nul avant la virgule. La position « réelle » de la virgule est donnée par l'exposant. Exemple, pour un nombre décimal, $1.1234 \cdot 10^{-3}$. 1.1234 s'appelle « coefficient » ou « mantisse », -3 s'appelle « exposant ». Il faut encoder trois information: le coefficient, l'exposant et le signe.

Premièrement, les nombres sont considéré étant des nombres binaires, ex., $1.001101101 \cdot 2^{-1101}$
Un nombre à virgule flottante est donc encodé dans un mot découpé en 3 champs:



Ce mot encode un nombre binaire à virgule flottante, par ex., $1.001101101 \cdot 2^{-1101}$

Notez qu'il y a une différence entre la valeur « physique » inscrite dans les champs de l'exposant encodé et du coefficient encodé et les valeurs réelles correspondantes, cf. l'explication ci-dessous.

Voici l'algorithme permettant d'« interpréter » ce nombre encodé.

1. Si le bit de signe est 0, le nombre est positif, sinon, il est négatif.
2. Selon la valeur de l'exposant encodé, deux cas se présentent.
   2.1. L'exposant encodé est entre 1 et 2^e-2, c.a.d., toute la plage des valeurs possibles sauf 0 et la valeur maximale 2^e-1. Dans ce cas, c'est un nombre normalisé qui est encodé. Le premier bit du coefficient réel, celui avant la virgule, est forcement 1. Les champs du coefficient encodé représentent la valeur du coefficient réel après la virgule (la partie fractionnaire). L'exposant réel est égal à l'exposant encodé moins 2^{e-1}-1 (un décalage de l'exposant). L'exposant réel peut donc prendre des valeurs entre -2^{e-1} et 2^{e-1}-1.

2.2. L'exposant encodé est aux valeurs extrêmes. Il s'agit des cas particuliers, qui sont les suivants.

a) Exposant encodé = 2^e-1(la valeur max.) Deux cas se présentent :
   - Le champs du coefficient =0 : le nombre est plus ou moins infini selon le bit du signe
   - Le champs du coefficient est différent de zéro : un Not A Number (NAN), par ex., un résultat de division par zéro.

b) Exposant encodé = 0 (la valeur min.). Là encore, deux cas se présentent:
   - Le champs du coefficient = 0 : le nombre encodé est zéro (+0 ou -0).
   - Le champs du coefficient ≠ 0 : c'est un nombre dénormalisé, c.a.d., le chiffre avant la virgule est 0. A part cela, le calcul du nombre est exactement comme dans le cas 2.1.

Application: la norme IEEE 754

Les deux formats fixés par la norme IEEE 754 sont sur 32 bits (« simple précision ») et 64 bits (« double précision »). La répartition des bits est la suivante :

Encodage	Signe	Exposant	Mantisse
Simple précision, 32 bits	1 bit	8 bits	23 bits
Simple précision, 64 bits	1 bit	11 bits	52 bits

Exercices.

1) Décoder les nombre suivant, encodé en format simple précision :

```
1   11110000   000 0000 0000 0000 1111 1110
0   00001111   000 0000 0000 1000 0000 0000
0   00000000   000 0000 0000 0000 0000 0001
1   11111111   000 0000 0000 0000 1111 1111
```

2) Donner les valeurs maximales et minimales pour les formats de simple et double précision du standard IEEE 754.