Représentation des données : types et valeurs de base

NSI - 1ère
COURS

Contenu : - Représentation binaire d'un entier relatif
- Représentation approximative des nombres réels : notion de nombre flottant

Capacités attendues : - Utiliser le complément à 2
- Calculer sur quelques exemples la représentation de nombres réels : 0.1, 0.25 ou 1/3

1. Nombre entier naturel

Il s'agit d'un nombre qui appartient au sens mathématique du terme à l'ensemble des entiers naturels N.

Sur 8 bits on pourra coder de 0 à 255 soit ______________ à ______________ soit encore ____________

Sur 16 bits on pourra coder de 0 à 65 535 soit __________________________
à __________________________

soit encore _________________

Sur 32 bits on pourra coder de 0 à 4 294 967 295 ($2^{32} - 1$)

Soit ____
à ____

soit encore ____________ à ____________

2. Opérations binaires

Il s'agit d'effectuer des opérations élémentaires dans des bases autres que 10. Pour ce faire, il suffit de **prendre conscience des mécanismes** que nous utilisons inconsciemment en base 10.

Addition.

Exemples d'additions :

$$\begin{array}{r} 1011 \\ +\ \ 111 \\ \hline (10010)_2 \end{array}$$

La table est très simple en binaire :

_________ ; ________________ ; _________ avec une retenue. Dans une autre base elle sera légèrement plus complexe.

Soustraction.

En binaire la table de soustraction est :

_________ ; _________ ; _________ ; _________ et l'on retire une unité de la puissance supérieure.

Exemples de soustraction :

$$\begin{array}{r} 111011 \\ -\ 000101 \\ \hline (110110)_2 \end{array}$$

Effectuer, à titre d'exercice, les multiplications suivantes.

On vérifiera en décimal....

$(11101 \cdot 101)_2$

Base 2
11101
x 101
11101
1110100
= 10010001
Soit $(\mathbf{145})_{10}$

$(1001\ 1101 \cdot 1011\ 0101)_2$

3. Nombre entier relatif, la représentation signée

Un nombre entier relatif est un nombre qui appartient au sens mathématique du terme à l'ensemble des entiers relatifs Z. Il s'écrit sous la forme du signe suivi d'un nombre entier naturel.

<u>Comment code t-on les entiers relatifs ?</u>
Le bit de poids le plus fort (MSB) sera le bit de signe.
MSB = 0 : le nombre est positif
MSB = 1 : le nombre est négatif

Ainsi sur 8 bits, il ne reste que 7 bits pour la "valeur absolue".
Donc, on pourra coder de -128 à + 127 soit de -2^7 à $+2^7-1$
Pour les nombres entiers positifs pas de changement. Exemple : $68_{(10)}$ = ________________

<u>Mais comment code t-on alors les entiers relatifs négatifs ?</u>

Exemple : soit à coder le nombre -74
On convertit $74_{(10)}$ en binaire : 74 = 64 + 8 + 2 donc $74_{(10)}$ = ________________
On effectue une inversion bit à bit 0 → 1 et 1 → 0 ________________
On ajoute 1 ________________
Ainsi -74 = 1011 0110

Vérification : Si l'on fait la somme $0100\ 1010_{(2)}$
+
$1011\ 0110_{(2)}$
On obtient :
____

Soit ____________
Car le _ part dans les oubliettes !!!!!!!
Sur 16 bits, le principe est le même. Le bit de poids le plus fort (MSB) est le bit de signe.
Il reste 15 bits pour "la valeur absolue".
Donc, on pourra coder de -2^{15} à $+2^{15}-1$ soit de -32 768 à + 32 767.

Sur 32 bits, le bit de poids le plus fort (MSB) est le bit de signe.
Il reste 31 bits pour "la valeur absolue".
Donc, on pourra coder de -2^{31} à $+2^{31}-1$ soit de -2 147 483 648 à + 2 147 483 647.

Mais tous les nombres ne sont pas des entiers relatifs …

4. Nombre réel, notation scientifique normalisée

Notation scientifique normalisée

L'écriture d'un nombre en notation scientifique normalisée est composée **d'un signe**, **d'un nombre décimal dont la partie entière est comprise entre 1 inclus et 10 exclus** et **d'un multiplicateur exprimé en puissances de 10**.
Cela permet ainsi de représenter de très grands nombres mais aussi de très petits nombres.
Exemples :
Soit le nombre 243,76, son écriture devient **+ 2,4376 . 10²** ($+2{,}4376 \cdot 10^2$)
Soit le nombre – 0,00341, son écriture devient **– 3,41 . 10⁻³** ($-3{,}41 \cdot 10^{-3}$)

Vocabulaire :
Le nombre décimal se nomme **<u>la mantisse (notation M)</u>**
La puissance signée du multiplicateur se nomme **<u>l'exposant (notation E)</u>**

Dans les exemples précédents :
Pour 243, 76 on a M = 2,4376 et E =+2
Pour – 0,00341 on a M = 3,41 et E =-3

Remarque : la précision diminue si E augmente.
Ainsi pour le nombre π, n = $0{,}3145 \times 10^1$ est plus précis que n = $0{,}0314 \times 10^2$

Codage des nombres réels en binaire, codage en virgule flottante

Il existe le codage en simple précision (sur 32 bits) en double précision (sur 64 bits) et en précision étendue (sur 80 bits). On se propose de voir uniquement le codage en simple précision.

Bit de rang	31	30	29	28	27	26	25	24	23	22	21	……	1	0
	S					2^3	2^2	2^1	2^0					
	Signe	Exposant E sur 8 bits								Mantisse M sur 23 bits				

Ce codage fait référence à la norme P754 de L'IEEE (Institute of Electrical and Electronics Engineers).

On peut rappeler préalablement quelques puissances négatives de 2

2^{-1} = ½ = 0,5 2^{-2} = ¼ = 0,25 2^{-3} = 1/8 = 0,125

Exemple : Codons le nombre décimal −118,625 en utilisant la norme IEEE 754.

C'est un nombre négatif, le bit de poids fort est donc 1

On écrit ensuite la valeur absolue du nombre en binaire. Ainsi 118,625 devient 1110110,101 $_{(2)}$

Explication : 118,625 = 64 + 32 + 16 + 4 + 2 + 0,5 + 0,125
$= 1x2^6 + 1x2^5 +1x2^4 +0x2^3 +1x2^2 +1x2^1 +0x2^0 +1x2^{-1} + 0x2^{-2} +1x2^{-3}$

Ensuite, on décale la virgule vers la gauche, de façon à ne laisser qu'un 1 sur sa gauche : 1110110,101 $_{(2)}$ = 1,110110101 $_{(2)}$ × 2^6.

C'est un nombre flottant normalisé : la mantisse est la partie à droite de la virgule, remplie de 0 vers la droite pour obtenir 23 bits. Cela donne 110 1101 0100 0000 0000 0000 (on omet le 1 avant la virgule, qui est implicite).

L'exposant est égal à 6, et nous devons le convertir en binaire et le décaler. Pour le format 32-bit IEEE 754, le décalage est de 127. Donc 6 + 127 = 133 $_{(10)}$ = 1000 0101 $_{(2)}$.

On a donc −118,625 $_{(10)}$ = 1100 0010 1 110 1101 0100 0000 0000 0000 $_{(2)}$ = C2ED4000 $_{(16)}$

5. Exercices

1) Exprimer dans le système décimal les valeurs suivantes signées sur 8 ou 16 bits.

% 0001 1111 % 1010 1100 % 1000 1010 1101 0000

----

2) Exprimer dans le système hexadécimal le codage IEEE 754 simple précision du nombre 521,625

3) Représenter en notation “module + signe” puis “complément à 2” les nombres suivants :

- 12
- -12
- 534
- -534

4) Que représentent en notation “complément à deux” les nombres suivants :

- 1011 1101
- 0011 1101

5) Effectuer les opérations suivantes :

- $(1\ 1101)_2 + (101\ 0111)_2$
- $(100\ 1101)_2 - (111)_2$

6) Représenter en binaire :

- 0,125
- 41,3