ISN LVH

CODAGE DIGITAL DES NOMBRES

1) Représentation des entiers naturels

a) Rappel 1 : la numération décimale :

La numération décimale utilise 10 chiffres : 0, 1, 2, 3, 4, 5, 6, 7, 8 et 9.

- Complétez : L'écriture du nombre $(275)_{10}$ signifie $2 \times 10^2 + 7 \times 10^1 + 5 \times 10^0$. On dit que 275 est la représentation en base 10 ou plus communément l'écriture décimale.
- Écrivez une égalité semblable pour les nombres
  - $1\,234 = 1 \times 10^3 + 2 \times 10^2 + 3 \times 10^1 + 4 \times 10^0$
  - $208 = ...2 \times 10^2 + 0 \times 10^1 + 8 \times 10^0$

b) Rappel 2 : la numération binaire :

La numération binaire utilise les deux chiffres 0 et 1.
L'écriture du nombre $(1110)_2$ signifie $1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 8 + 4 + 2 + 0 = 14$.
1110 est la représentation en base 2 de 14.

- Le bit associé à 2^0 est appelé le bit de **poids faible**.
- Le bit associé à la plus grande puissance de 2 est appelé le bit de **poids fort**.

2) Représentation des entiers relatifs

L'espace mémoire étant fini, dans tout ce chapitre, nous allons travailler sur **un octet** pour comprendre plus facilement les contraintes et les solutions retenues.
Sur un octet, on peut coder $2^8 = 256$ objets différents.

a) Première solution

- On réserve le bit de poids fort pour représenter le signe : 0 pour positif et 1 pour négatif.
  - Le plus petit entier relatif est alors $(1111\ 1111)_2$ soit -127
  - Le plus grand entier relatif est alors $(0111\ 1111)_2$ soit $+127$
- On code donc 255 nombres et non 256.
- Complétez : En effet 0 est codé 2 fois +0 et –0.

Cette première raison et la gestion des additions à différencier suivant les signes font que ce système n'est pas utilisé en pratique.

b) Le complément à deux

- On code les positifs de 0 à 127 comme les entiers naturels et on code les négatifs en les décalant de 256 donc sur la plage des naturels de 128 à 255. Cela donne en décimal :

entiers naturels	0	1	2	…	127	128	129	130	…	254	255
complément à 2	0	1	2	…	127	-128	-127	-126	…	-2	-1

- 1110 0001 en complément à 2 code –31 en décimal.
- 12 en décimal, se code 0000 1100 en complément à deux.
- –43 en décimal, se code 1101 0101 en complément à deux.
- Effectuez à la main les additions en binaire :

$$\begin{array}{r} 1\ 1\ 0\ 1\ 0\ 1\ 0\ 1 \\ +\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 0 \\ \hline 1\ 1\ 1\ 0\ 0\ 0\ 0\ 1 \end{array} \qquad \begin{array}{r} 0\ 1\ 1\ 1\ 0\ 0\ 1\ 0 \\ +\ 1\ 1\ 1\ 1\ 0\ 1\ 1\ 1 \\ \hline 0\ 1\ 1\ 0\ 1\ 0\ 0\ 1 \end{array}$$

On travaille sur un octet, ainsi la dernière retenue peut être perdue.

Complétez :

- Or en complément à deux,
  - 0111 0010 représente 114_{10} ; 1111 0111 représente -9_{10} et 0110 1001 représente 105_{10}

- 1101 0101 représente -43_{10} ; 0000 1100 représente 12_{10} et 1110 0001 représente -31_{10}
- donc les additions précédentes sont correctes.

Le complément à deux permet donc de gérer les additions sans traiter les différents cas de signes.

- Lorsque les entiers sont codés sur 64 bits, quels sont les extremums possibles ?
  Le plus petit entier naturel codé en complément à 2 sur 64 bits est -2^{32} soit $-4\ 294\ 967\ 296$.
  Le plus grand entier naturel codé en complément à 2 sur 64 bits est $+2^{32}-1$, soit $4\ 294\ 967\ 295$.

3) Représentation des réels

a) Rappel : l'écriture décimale

Complétez : 24,153 signifie $2 \times 10^{2} + 4 \times 10^{1} + 1 \times 10^{0} + 5 \times 10^{-1} + 3 \times 10^{-2}$

b) L'écriture bicimale

De la même façon, sans tenir compte de la représentation du signe, quel nombre b est représenté par l'écriture bicimale suivante 1010,0101 ?

$$b = 1 \times 2^{3} + 0 \times 2^{2} + 1 \times 2^{1} + 0 \times 2^{0} + 0 \times 2^{-1} + 1 \times 2^{-2} + 0 \times 2^{-3} + 1 \times 2^{-4}$$

$$b = 8 + 0 + 2 + 0 + 0 + \frac{1}{4} + 0 + \frac{1}{16} = \frac{165}{16} \text{ ou bien}$$

$$b = 10 + 0{,}25 + 0{,}0625 = 10{,}3125$$

c) Rappel : l'écriture scientifique

La notation scientifique consiste à écrire les réels sous la forme $\pm\ a \times 10^{p}$ avec $1 \leqslant a < 10$ et $p \in \mathbb{Z}$.

Par exemple $-125{,}037$ s'écrit $-1{,}25037 \times 10^{2}$.

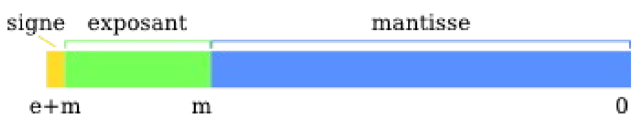
d) Les flottants

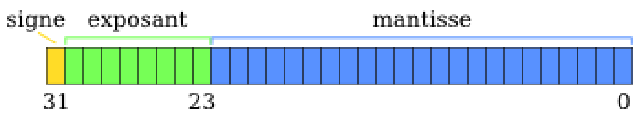
On utilise l'écriture scientifique adaptée à la base deux en réservant un nombre fixé de bits pour les informations à coder.

Les flottants sont les nombres de la forme $\pm\ a \times 2^{p}$ avec $1 \leqslant a < 2$ et $p \in \mathbb{Z}$ où :

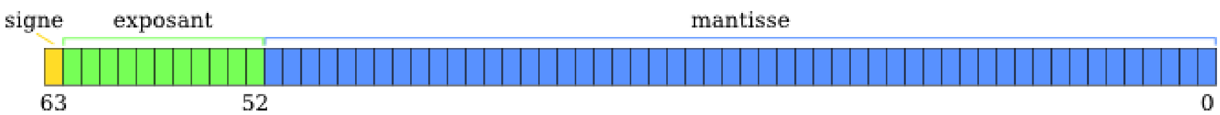
- le signe est codé par le bit de poids fort (de gauche) où 0 signifie positif comme pour les entiers,
- l'exposant n'est pas codé en complément à deux car cela complique inutilement la comparaison de 2 flottants. L'exposant est décalé d'un nombre bien choisi sur les e bits suivants
- la mantisse a est codée sur les m derniers octets de poids faibles. Comme $1 \leqslant a < 2$, la partie entière de a est toujours 1, on ne code donc en mémoire que la partie « bicimale »

Exemples : Répartitions des informations sur 32 bits





Répartitions des informations sur 64.



- Trouvez une raison simple qui explique pourquoi on ne peut pas représenter tous les réels.

Il y a une infinité de réels et $2^{64}-1$ (0 est codé 2 fois +0 et −0) flottants différents.

- Codons le nombre décimal –118,625 sur 32 bits.
  - Premièrement, nous avons besoin du signe, de l'exposant et de la partie fractionnaire. Ce nombre est négatif, le signe est donc « 1 ».
  - Puis nous écrivons la valeur absolue en binaire. Nous obtenons 1110110,101.
  - Trouvons l'équivalent de l'écriture scientifique : $1110110{,}101_2 = 1{,}110110101_2 \times 2^6$. Pour la mantisse, on omet le 1 avant la virgule. Il est toujours présent, inutile de le coder. La partie bicimale (à droite de la virgule) est complétée vers la droite avec des 0 pour obtenir 23 bits.
  - La mantisse est donc 110 0100 0000 0000 0000.
  - L'exposant est 6. Nous le décalons et le convertissons en binaire. Pour le format 32 bits, le décalage est $2^{8-1}-1 = 127$. Donc $6_{10} + 127_{10} = 133_{10}$(dec) $= 1000\ 0101_2$.
  - Finalement, $-118{,}625_{10}$ s'écrit 1100 0010 1110 1101 0100 0000 0000 0000 en flottant.

- Trouvez l'écriture de 0,25 sous forme de flottant 32 bits.

0,25 est positif, le bit de poids fort est 0.

$0{,}25 = 2^{-2} = 0{,}01_2$. La mantisse est $1{,}0_2$, mais on ne code pas l'unité donc il n'y a que des 0.
L'exposant de l'écriture « scientifique » est –2 qui décalé de 127 donne 125_{10} =0111 1101 ainsi
0,25 est codé 00111110 10000000 00000000 00000000

- Trouvez l'écriture de 0,75 sous forme de flottant 32 bits.

0,75 est positif, le bit de poids fort est 0.

$0{,}75 = 2^{-1} + 2^{-2} = 0{,}11_2$. La mantisse est $1{,}1_2$, mais on ne code pas l'unité donc il n'y a qu'un seul 1 complété par des 0. L'exposant de l'écriture « scientifique » est –1 qui décalé de 127 donne 126_{10} =0111 1110
ainsi 0,75 est codé 00111111 01000000 00000000 00000000

- Quelle est l'écriture décimale du flottant 11000001 00100101 00000000 00000000 ?

Le nombre est négatif. L'exposant de l'écriture « scientifique » est $1000\ 00010_2 = 130_{10}$ qui décalé de 127 donne 3.

La mantisse reconstituée est $1{,}0100101_2$. Le nombre est $-1010{,}0101_2$. $= -(10+0{,}25+0{,}0625)_{10} = -10{,}3125$

4) Questionnaire de fin

- Qu'as-tu fait pendant ces 2 heures ? (2 lignes maximum)

J'ai manipulé plusieurs représentations binaires des nombres entiers et décimaux.

- Liste ce que tu as appris pendant ces 2 heures ?