

La représentation des nombres réels

- Un nombre réel est constitué de deux parties : la partie entière et la partie fractionnelle (les deux parties sont séparées par une virgule)
- **Problème** : comment indiquer à la machine la position de la virgule ?
- Il existe deux méthodes pour représenter les nombre réel :
 - Virgule fixe : la position de la virgule est fixe
 - Virgule flottante : la position de la virgule change (dynamique)

Représentation en virgule flottante

- Chaque nombre réel peut s'écrire de la façon suivante :

$$N = \pm M * b^e$$

- M : mantisse ,
- b : la base ,
- e : l'exposant

- **Exemple** :

$$15,6 = 0,156 * 10^{+2}$$

$$-(110,101)_2 = -(0,110101)_2 * 2^{+3}$$

$$(0,00101)_2 = (0,101)_2 * 2^{-2}$$

Remarque :

on dit que la mantisse est **normalisée** si le premier chiffre après la virgule est différent de 0 et le premier chiffre avant la virgule est égale à 0.

La virgule fixe

- Dans cette représentation la partie entière est représentée sur **n** bits et la partie fractionnelle sur **p** bits , en plus un bit est utilisé pour le signe.
- Exemple : si n=3 et p=2 on va avoir les valeurs suivantes

Signe	P.E	P.F	valeur
0	000	00	+ 0,0
0	000	01	+ 0,25
0	000	10	+ 0,5
0	000	11	+ 0,75
0	001	.00	+ 1,0
.	.	.	.
.	.	.	.

Dans cette représentation les valeurs sont limitées et nous n'avons pas une grande précision

- Dans cette représentation sur **n** bits :

- La mantisse est sous la forme signe/valeur absolue
 - 1 bit pour le signe
 - et **k** bits pour la valeur.
- L'exposant (positif ou négatif) est représenté sur **p** bits .

Signe mantisse	Exposant	Mantisse normalisée
1 bit	p bits	k bits

- Pour la représentation de l'exposant on utilise :
 - Le complément à deux
 - Exposant décalé ou biaisé

Représentation de l'exposant en complément à deux

- On veut représenter les nombres $(0,015)_8$ et $-(15,01)_8$ en virgule flottante sur une machine ayant le format suivant :

Signe mantisse	Exposant en CA2	Mantisse normalisée
1 bit	4 bits	8 bits

$$(0,015)_8 = (0,000001101)_2 = 0,1101 \cdot 2^{-5}$$

Signe mantisse : positif (0)

Mantisse normalisée : 0,1101

Exposant = -5 → utiliser le complément à deux pour représenter le -5

Sur 4 bits CA2(0101)=1011

0	1 0 1 1	1 1 0 1 0 0 0 0
1 bit	4 bits	8 bits

$$-(15,01)_8 = -(001101,000001)_2 = -0,1101000001 \cdot 2^4$$

Signe mantisse : négatif (1)

Mantisse normalisée : 0,1101000001

Exposant = 4 , en complément à deux il garde la même valeur (0100)

On remarque que la mantisse est sur 10 bits (1101 0000 **01**), et sur la machine seulement 8 bits sont utilisés pour la mantisse.

Dans ce cas on va prendre les 8 premiers bits de la mantisse

1	0 1 0 0	1 1 0 1 0 0 0 0
1 bit	4 bits	8 bits

Remarque :

si la mantisse est sur k bits et si elle est représentée sur la machine sur k' bits tel que $k > k'$, alors la mantisse sera tronquée : on va prendre uniquement k' bits → **perdre dans la précision** .

L' Exposant décalé (biaisé)

- en complément à 2, l'intervalle des valeurs qu'on peut représenter sur p bits :

$$-2^{(p-1)} \leq N \leq 2^{(p-1)} - 1$$

Si on rajoute la valeur $2^{(p-1)}$ à tout les terme de cette inégalité :

$$-2^{(p-1)} + 2^{(p-1)} \leq N + 2^{(p-1)} \leq 2^{(p-1)} - 1 + 2^{(p-1)}$$

$$0 \leq N + 2^{(p-1)} \leq 2^p - 1$$

- On pose $N' = N + 2^{(p-1)}$ donc : $0 \leq N' \leq 2^p - 1$
- Dans ce cas on obtient des valeurs positives.
- La valeur 2^{p-1} s'appelle le **biais** ou le **décalage**

- Avec l'exposant biaisé on a transformé les exposants négatifs à des exposants positifs en rajoutons à l'exposant la valeur 2^{p-1} .

$$\text{Exposant Biaisé} = \text{Exposant réel} + \text{Biais}$$

Exemple

- On veut représenter les nombres $(0,015)_8$ et $-(15,01)_8$ en virgule flottante sur une machine ayant le format suivant :

Signe mantisse	Exposant décalé	Mantisse normalisée
1 bit	4 bits	11 bits

$$(0,015)_8 = (0,000001101)_2 = 0,1101 * 2^{-5}$$

Signe mantisse : positif (0)

Mantisse normalisée : 0,1101

Exposant réel = -5

Calculer le biais : $b = 2^{4-1} = 8$

Exposant Biaisé = $-5 + 8 = +3 = (0011)_2$

0	0011	1 1 0 1 0 0 0 0 0 0 0
1 bit	4 bits	11 bits

$$-(15,01)_8 = (001101,000001)_2 = 0,1101000001 * 2^4$$

Signe mantisse : négatif (1)

Mantisse normalisée : 0,1101000001

Exposant réel = + 4

Calculer le biais : $b = 2^{4-1} = 8$

Exposant Biaisé = $4 + 8 = +12 = (1100)_2$

1	1100	1 1 0 1 0 0 0 0 0 1 0
1 bit	4 bits	11 bits

Opérations arithmétiques en virgule flottante

- Soit deux nombres réels $N1$ et $N2$ tel que $N1 = M1 * b^{e1}$ et $N2 = M2 * b^{e2}$
- On veut calculer $N1 + N2$?
- Deux cas se présentent :
 - Si $e1 = e2$ alors $N3 = (M1 + M2) b^{e1}$
 - Si $e1 \neq e2$ alors élever au plus grand exposant et faire l'addition des mantisses et par la suite normalisée la mantisse du résultat.

Exemple

- Effectuer l'opération suivante : $(0,15)_8 + (1,5)_8 = (?)$:

$$(0,15)_8 = (0,001101)_2 = 0,1101 * 2^{-2}$$

$$(1,5)_8 = (001, 1 01)_2 = 0,1101 * 2^1$$

$$\begin{aligned} (0,15)_8 + (1,5)_8 &= 0,1101 * 2^{-2} + 0,1101 * 2^1 \\ &= 0,0001101 * 2^1 + 0,1101 * 2^1 \\ &= 0, 1110101 * 2^1 \end{aligned}$$

0	0001	111010
1 bit	4 bits	6 bits

Exercice

Donner la représentation des deux nombres **N1= (-0,014)₈** et **N2=(0,14)₈** sur la machine suivante :

Signe mantisse	Exposant biaisé (décalé)	Mantisse normalisée
----------------	--------------------------	---------------------

1 bit

5 bits

10 bits

- Calculer **N2-N1** ?.

$$\begin{aligned}
 N2 - N1 &= 0,14 - (-0,014) = 0,14 + 0,014 \\
 N2 - N1 &= (0,1100)_2 \cdot 2^{-2} + (0,1100)_2 \cdot 2^{-5} \\
 &= (0,1100)_2 \cdot 2^{-2} + (0,0001100)_2 \cdot 2^{-2} \\
 &= (0,1101100)_2 \cdot 2^{-2}
 \end{aligned}$$

• Si on fait les calculs avec l'exposant biaisé :

$$\begin{aligned}
 N2 - N1 &= (0,1100)_2 \cdot 2^{14} + (0,1100)_2 \cdot 2^{11} \\
 &= (0,1100)_2 \cdot 2^{14} + (0,0001100)_2 \cdot 2^{14} \\
 &= (0,1101100)_2 \cdot 2^{14}
 \end{aligned}$$

Exposant biaisé = 14

Exposant réel = Exposant biaisé – Biais

Exposant réel = 14 – 16 = -2

Donc on trouve le même résultat que la première opération.

Avant de représenter les deux nombres on doit calculer le biais (décalage)

$$B = 2^{5-1} = 2^4 = 16$$

$$N1 = (-0,014)_8 = (-0,000001100)_2 = (-0,1100)_2 \cdot 2^{-5}$$

$$\text{ExpB} = -5 + 16 = 11 = (01011)_2$$

$$N2 = (0,14)_8 = (0,001100)_2 = (0,1100)_2 \cdot 2^{-2}$$

$$\text{ExpB} = -2 + 16 = 14 = (01110)_2$$

Donc on va avoir la représentation suivante pour N1 et N2:

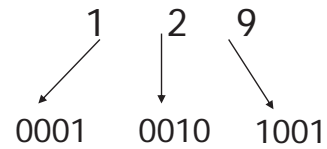
N1	1	01011	1100000000	1 010 1111 0000 0000 (AF00)₁₆
N2	0	01110	1100000000	0011 10 11 0000 0000 (3B00)₁₆

Le codage BCD (Binary Coded Decimal)

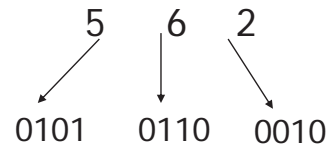
- Pour passer du décimal au binaire, il faut effectuer des divisions successives. Il existe une autre méthode simplifiée pour le passage du décimal au binaire.
- Le principe consiste à faire des éclatements sur 4 bits et de remplacer chaque chiffre décimal par sa valeur binaire correspondante.
- Les combinaisons supérieures à 9 sont interdites

Décimal	Binaire
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

Exemple



$$129 = (0001\ 0010\ 1001)_2$$



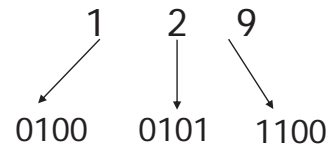
$$562 = (0101\ 0110\ 0010)_2$$

Codage des caractères

- Les caractères englobent : les lettres alphabétiques (A, a, B, B,...) , les chiffres , et les autres symboles (> , ; / :) .
- Le codage le plus utilisé est le **ASCII** (American Standard Code for Information Interchange)
- Dans ce codage chaque caractère est représenté sur **8 bits** .
- Avec 8 bits on peut avoir $2^8 = 256$ combinaisons
- Chaque combinaison représente un caractère
 - Exemple :
 - Le code 65 $(01000001)_2$ correspond au caractère **A**
 - Le code 97 (01100001) correspond au caractère **a**
 - Le code 58 (00111010) correspond au caractère **:**
- Actuellement il existe un autre code sur 16 bits , se code s'appel **UNICODE** .

Le codage EXCESS3 (BCD+3)

Décimal	BCD+3	Binaire
0	3	0011
1	4	0100
2	5	0101
3	6	0110
4	7	0111
5	8	1000
6	9	1001
7	10	1010
8	11	1011
9	12	1100



Fin de la cinquième séance