# Attention !

Les transparents sont des supports de prise de note :

- Ils ne contiennent pas la totalité du cours
- ils sont faits pour être complétés par ce qui est dit en cours, en TD et en TP.

Si vous n'avez pas assisté au cours je vous conseille fortement de récupérer auprès d'un de vos collègues ce qui pourrait manquer sur ces transparents.

Connaitre par cœur le contenu des transparents n'est pas une garantie de réussite à l'examen.

# Systèmes d'exploitation et Architecture I

## Représentation des nombres (2) - virgule flottante

Marie Duflot-Kremer

Licence 2 - Université Paris Est Créteil

# Ce qui nous manque

Représentations précédentes pas adaptées pour représenter :

- les très grands nombres
- les nombres ayant une partie fractionnaire (réels)

## Notation compacte

$976000000000 = 9,76 \times 10^{11}$
$0,0000000128 = 1,28 \times 10^{-8}$

- On va faire la même chose en binaire

# Binaire → décimal

Similaire aux entiers :

- additionner des puissances de 2
- elles peuvent être négatives
- à gauche de la virgule, puissances positives
- à droite de la virgule, puissances négatives

**Exemple :**

$$\begin{aligned} 1,0101_2 &= 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} + 0 \times 2^{-3} + 1 \times 2^{-4} \\ &= 1,3125 \end{aligned}$$

# Binaire $\rightarrow$ décimal - 2$^{\text{ème}}$ méthode

Pour la partie entière on sait faire (cf. cours entiers).
Pour la partie fractionnaire :

- on part du bit le plus à droite et de la valeur 0
- tant qu'il reste des chiffres à droite de la virgule :
  - si le chiffre considéré est 1 on ajoute 1 à notre valeur,
  - on divise le résultat par 2.

### Exemple

0,1101 En décimal : $0,8125_{10}$

- 0+1 = 1 puis 1/2=0,5
- 0,5/2 = 0,25
- 0,25+1=1,25 puis 1,25/2 = 0,625
- 0,625+1 = 1,625 puis 1,625/2 = 0,8125

# Décimal $\rightarrow$ binaire

Partie entière :

- comme d'habitude
- divisions successives par 2 et recopie du reste

Partie fractionnaire :

- on multiplie par 2,
- on recopie la partie entière puis on la soustrait,
- on recommence tant que la partie fractionnaire n'est pas nulle.

# Exemple

Prenons pour valeur 6,4375 en décimal.

Partie entière : 6

En binaire : 110,0111

- 6 /2 $\rightarrow$ quotient 3 reste 0
- 3 /2 $\rightarrow$ quotient 1 reste 1
- 1 /2 $\rightarrow$ quotient 0 reste 1

Partie fractionnaire : 0,4375

- on multiplie par 2 $\rightarrow$ 0,875 $<$1 on pose 0 reste 0,875
- 0,875 $\times$ 2 $\rightarrow$ 1,75 on pose 1 reste 0,75
- 0,75 $\times$ 2 $\rightarrow$ 1,5 on pose 1 reste 0,5
- 0,5 $\times$ 2 $\rightarrow$ 1 on pose 1 reste 0

# Représentation finie ?

Représentation décimale finie $\stackrel{?}{\Rightarrow}$ représentation binaire finie.
Réponse : non

**Preuve ?**

$0,6_{10} = 0,10011001100..._2$

Représentation binaire finie $\stackrel{?}{\Rightarrow}$ représentation décimale finie.
Réponse : oui

**Preuve ?**

Diviser par deux = diviser par 10 puis multiplier par 5
Toute puissance de deux (même négative) a une représentation finie.

# Norme IEEE 754

- Standard pour la représentation des nombres en virgule flottante
- Nombre codé sur 32 bits (ou 64 pour le format double)

## La norme

- 1 bit de signe $S$, 0 si positif, 1 si négatif,
- 8 bits pour l'exposant biaisé $E$,
- 23 bits pour la mantisse $M$, qui sert à représenter la partie fractionnaire.

Notation : $SEM$ pour le nombre :

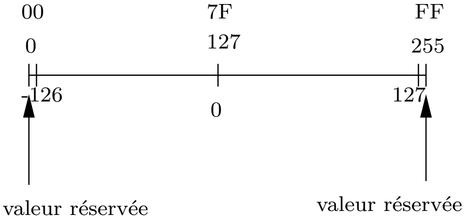
$$A = (-1)^{S} \times 1, M \times 2^{E-127}$$

# Calculer l'exposant E'

- On veut un seul chiffre significatif avant la virgule
  - 101,101 $\rightarrow$ 1,01101 $\times 2^2$, exposant E'=2
  - 1,001 $\rightarrow$ exposant E'=0, rien à changer
  - 0,1001 $\rightarrow$ 1,001 $\times 2^{-1}$, exposant E'=-1
  - 0,001 $\rightarrow$ 1 $\times 2^{-3}$, exposant E'=-3
- Le chiffre 0, tout seul, n'est pas significatif.
- Et pour 0, comment calcule-t-on l'exposant ?
  - On ne peut pas. Besoin de le coder autrement.

# Exposant biaisé

- On représente un exposant E de type entier non signé (=positif)
- On ne prend pas E=E' mais E=E'+$127_{10}$
- valeurs exprimables en exposant biaisé sur 8 bits ?




- véritable exposant $E'$ : on prend la valeur de $E - 127_{10}$

# Mantisse

Elle contient les chiffres significatifs

- on normalise la mantisse : on garde un seul chiffre ($\neq 0$) avant la virgule,
- ce chiffre est forcément un 1... donc pas besoin de l'écrire,
- on dit alors que la mantisse a un bit caché.

Ex : $1101 = 1,101 \times 2^3$

La mantisse en IEE 754 sera donc : $M = 1010000...0$

# Exemple (1)

Reprenons notre exemple : A$=110,0111_2$

1. le signe : 0 (positif)
2. On normalise : A$=1,100111 \times 2^2$
3. la mantisse : $M =$1001110...0
4. l'exposant : $E' =2$ d'où $E =129_{10} = 10000001_2$
5. le résultat : 0 10000001 10011100..0
6. et en hexadécimal : $40CE0000$

# Exemple (2)

$-0,1640625_{10}$

1. le signe : 1 (négatif)
2. en binaire : $0,0010101$
3. On normalise : $1,0101 \times 2^{-3}$
4. la mantisse : $M =$01010...0
5. l'exposant : $E' =-3$ d'où $E =124_{10} = 01111100_2$
6. le résultat : 1 01111100 010100..0
7. et en hexadécimal : $BE280000$

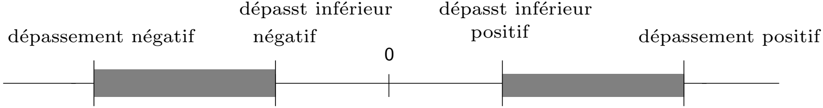
# Valeurs particulières

Avec cette norme on ne peut pas tout représenter :

- $+\infty$ et $-\infty$ exposant 1...1, mantisse 0...0
- $+0$ et $-0$, exposant et mantisse à 0...0
- indéterminées ($0/0$, $0 \times \infty$, $\infty - \infty$,...) exposant à 1...1, mantisse non nulle
- nombres dénormalisés : exposant 0...0. Il signifie $E' = -126$. Le bit caché vaut 0. Cela permet de représenter des nombres plus petits en valeur absolue que le plus petit nombre normalisé.

# Valeurs normailsées représentables

- On ne peut pas tout représenter
- on doit souvent faire des approximations
- ... mais tout de même une palette assez large



# Addition et soustraction

Plus compliquée que sur les entiers du fait du décalage.
Se fait en 4 étapes :

1. Recherche de zéros : si X ou Y vaut 0, le résultat est immédiat. Si c'est une soustraction, on inverse le bit de signe de Y.
2. Alignement des mantisses :
   - On refait apparaître le premier bit à 1
   - si les exposants sont différents, on décale la mantisse du nombre ayant le plus petit exposant vers la droite en incrémentant l'exposant.
3. Addition/soustraction des mantisses : il faut faire attention au signe, et gérer le cas particulier où l'on obtient 0.
4. Normalisation : si besoin, on décale la mantisse en ajustant l'exposant.

# Addition et soustraction - un exemple

On veut additionner C0A00000 et C1000000

- En binaire :
  - 1 10000001 010 0000 0000 0000 0000 0000 soit $-\ 1.01\ \times 2^2$
  - et 1 10000010 10000000000000000000000 soit $-\ 1.1\ \times 2^3$
- Alignement des mantisses :$1.01\ \times 2^2 = 0.101\ \times 2^3$
- Addition des mantisses : $0.101 + 1.1 = 10.001$
- Renormalisation : $10.001\ \times 2^3 = 1.0001\ \times 2^4$
- nouvelle mantisse : 0001000...0
- nouvel exposant : $4 + 127 = 131_{10} \rightarrow 10000011$
- Résultat : $1\ 10000011\ 0001000...0_2 = C1880000_{16}$
- Tout ça pour faire : $5 + 12$ ! ! !

# Erreurs d'arrondis

- Mantisse finie $\rightarrow$ erreurs d'arrondi ! !
- l'addition n'est plus associative
  - A+(B+C) plus toujours égal à (A+B)+C
- $1 + (2^{-24} + 2^{-24}) = 1 + 2^{-23}$
- mais $(1 + 2^{-24}) + 2^{-24} = 1$

## Multiplication et division

Pas de problème de décalage

1. Recherche de zéro. Si oui :
   - soit résultat $=0$ (multiplication, dividende)
   - soit une erreur (diviseur)
2. somme/soustraction des exposants

Attention : soustraire ou additionner le biais qui a été compté 2 (multiplication) ou 0 fois (division)

3. Si on a un dépassement on le signale et on s'arête
4. produit/division des mantisses puis on arrondit le résultat (mantisse de 23 bits)