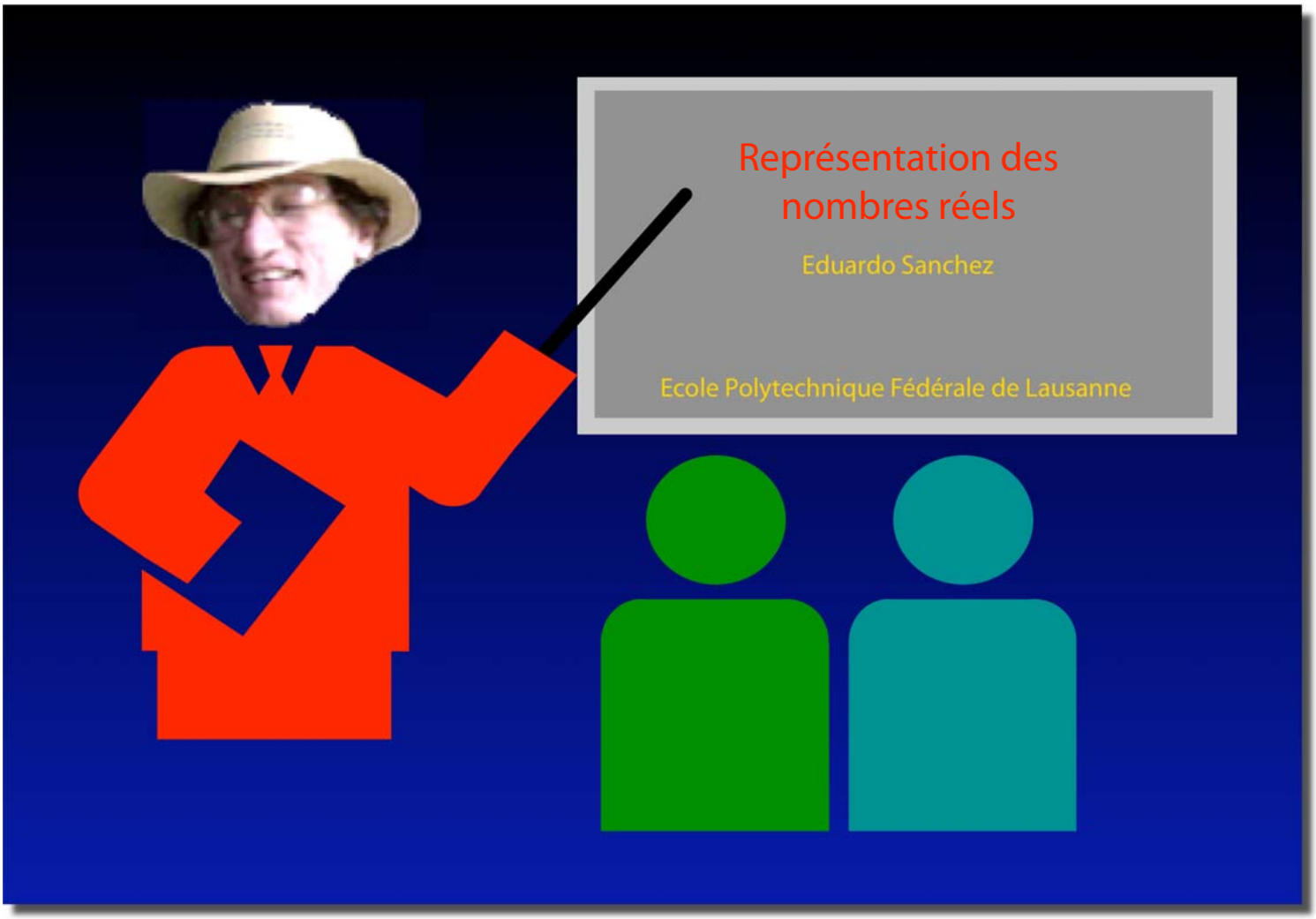


Représentation des nombres réels

- Un nombre réel est représenté en décimal sous la forme:
  $d_m d_{m-1} \ldots d_1 d_0 . d_{-1} d_{-2} \ldots d_{-n}$
  où la valeur du nombre est:

$$d = \sum_{i=-n}^{m} 10^i \times d_i$$

- Par exemple, 12.34_{10} représente le nombre:
  $1x10^1+2x10^0+3x10^{-1}+4x10^{-2} = 12\ 34/100$
- En conséquence, en décimal on ne peut représenter exactement que des nombres fractionnaires de la forme $X/10^k$

- En binaire, nous avons:

  $b_m b_{m-1} \ldots b_1 b_0 . b_{-1} b_{-2} \ldots b_{-n}$

  où la valeur du nombre est:

$$b = \sum_{i=-n}^{m} 2^i \times b_i$$

- Par exemple, 101.11_2 représente le nombre:

  $1x2^2+0x2^1+1x2^0+1x2^{-1}+1x2^{-2} = 5\ 3/4$

- En conséquence, en binaire on ne peut représenter exactement que des nombres fractionnaires de la forme $X/2^k$
- Exemples:
  - $1/3 = 0.0101010101[01]_2$
  - $1/10 = 0001100110011[0011]_2$

Eduardo Sanchez
Ecole Polytechnique Fédérale de Lausanne

- Passage à binaire d'un nombre réel en base 10:

```
0.375 = ?

        0.375x2 = 0.75
        0.75x2 = 1.5
        0.5x2 = 1.0

0.375 = 0.011
```


Eduardo Sanchez
Ecole Polytechnique Fédérale de Lausanne

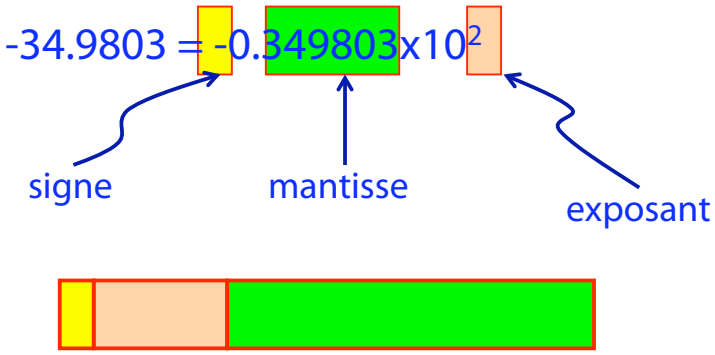
```
0.3 = ?

      0.3x2 = 0.6
      0.6x2 = 1.2
      0.2x2 = 0.4
      0.4x2 = 0.8
      0.8x2 = 1.6
      0.6x2 = 1.2
        ...

0.3 = 0.01001[1001]
```


Eduardo Sanchez
Ecole Polytechnique Fédérale de Lausanne

Codage binaire des nombres réels



- Le codage sur un nombre n de bits, fixe, implique un nombre fini de valeurs:
  - les calculs sont nécessairement arrondis
  - il y a des erreurs d'arrondi et de précision
- On ne peut plus faire les opérations de façon transparente

Eduardo Sanchez
Ecole Polytechnique Fédérale de Lausanne

- Un même nombre réel peut être écrit de différentes façons. Par exemple:

  $0.110x2^5 = 110x2^2 = 0.0110x2^6$

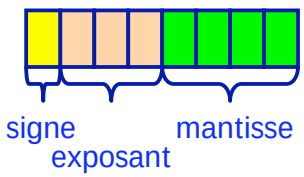
- Pour éviter des représentations différentes du même nombre, la mantisse est normalisée. Dans la convention la plus courante, un nombre binaire normalisé différent de zéro a la forme:

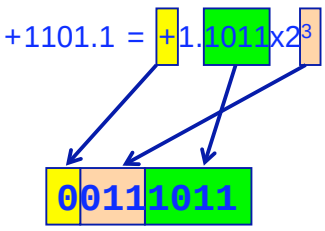
  $\pm 1.bbb...bx2^{\pm e}$

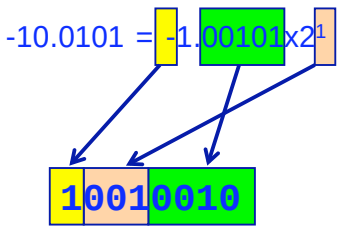
- Comme, sous cette forme, le bit le plus signifiant est toujours égal à 1, il n'est pas nécessaire de le coder: il est implicite

Eduardo Sanchez
Ecole Polytechnique Fédérale de Lausanne

- Exemple:
  supposons la représentation suivante:







Eduardo Sanchez
Ecole Polytechnique Fédérale de Lausanne

- Problème: sous cette forme, il est impossible de coder le nombre zéro
- Solution: le nombre zéro est représenté avec tous les bits à 0. Avec la représentation de l'exemple précédent, nous avons: 0.0 = 0 000 0000
- Par extension, tous les nombres avec exposant égal à 0 sont dits non normalisés: le bit à gauche du point décimal est égal à 0 et non pas à 1, comme c'est le cas pour les autres nombres, normalisés
- Nous reviendrons plus tard sur les nombres non normalisés

Eduardo Sanchez
Ecole Polytechnique Fédérale de Lausanne

- Problème: comment représenter le nombre 1.0?
- Problème: comment représenter les exposants négatifs?
- Comme un exposant est un nombre entier signé, une solution serait de le représenter en complément à 2. Ce n'est toutefois pas la solution choisie…
- En général, l'exposant est représenté de façon biaisée: une constante, le biais, doit être soustrait de la valeur dans le champ pour obtenir la vraie valeur de l'exposant:

  champ exposant = exposant + biais
- Typiquement, la valeur du biais est $2^{k-1}-1$, où k est le nombre de bits du champ de l'exposant
- Toutefois, les deux valeurs extrêmes du champ exposant sont réservées pour des cas particuliers:
  - 00…00: pour les nombres non normalisés ($0 \leq X < 1$)
  - 11..11: pour infini (positif et négatif) et NaN (*not a number*)

Eduardo Sanchez
Ecole Polytechnique Fédérale de Lausanne

- Exemple:
  si k=4, la valeur du biais sera 7, l'exposant pourra avoir une valeur entre -2^3+1 et 2^3, et la valeur représentée dans le champ exposant sera exposant+biais

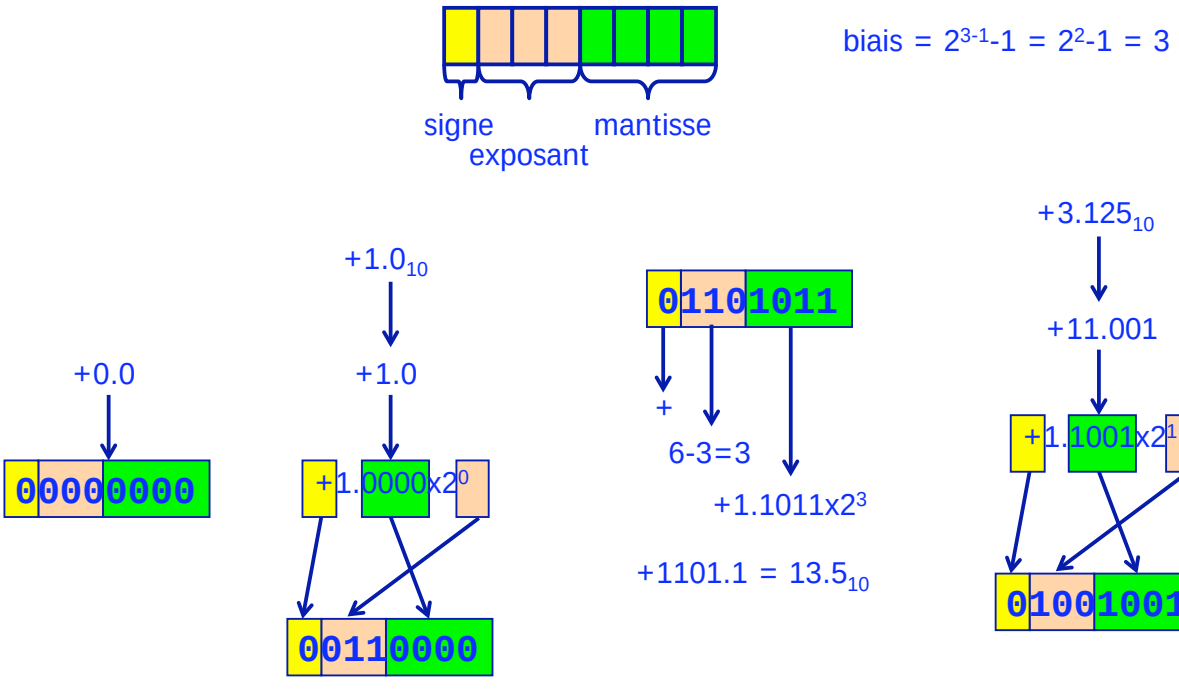
champ exposant	exposant	champ exposant	exposant
0000	non normalisé	1000	1
0001	-6	1001	2
0010	-5	1010	3
0011	-4	1011	4
0100	-3	1100	5
0101	-2	1101	6
0110	-1	1110	7
0111	0	1111	infini

où les valeurs -7 et 8 de l'exposant sont réservées pour les cas spéciaux (nombres non normalisés et infini, respectivement)

- Cette représentation facilite la comparaison entre deux nombres et permet la représentation des nombres 0.0 et 1.0

Eduardo Sanchez
Ecole Polytechnique Fédérale de Lausanne

- Exemple:
  supposons la représentation suivante:



Eduardo Sanchez
Ecole Polytechnique Fédérale de Lausanne

Nombres non normalisés

- Avec la représentation normalisée des nombres réels, le bit à gauche du point décimal est toujours égal à 1, implicitement. Il est donc impossible de représenter la valeur 0.0 de façon normalisée
- Pour résoudre ce problème, certains nombres sont non normalisés: dans ces cas, le bit à gauche du point décimal est égal à 0
- Les nombres non normalisés sont ceux dont le champ de l'exposant est égal à 0

Eduardo Sanchez
Ecole Polytechnique Fédérale de Lausanne

- Il existe donc deux représentations pour 0.0:
  - +0.0 = 0 00…00
  - -0.0 = 1 00…00
- Les nombres non normalisés différents de 0.0 sont utilisés pour représenter de très petites valeurs, proches de 0.0
- Normalement, la valeur de l'exposant d'un nombre non normalisé devrait être -biais. Toutefois, pour assurer une meilleure transition entre les nombres normalisés et les non normalisés, l'exposant est calculé dans ces cas comme 1-biais

Eduardo Sanchez
Ecole Polytechnique Fédérale de Lausanne

Standard IEEE 754

- Précision simple:
  - nombre de bits: 32
  - mantisse sur 23 bits
  - exposant sur 8 bits (biais = $2^{8-1} - 1 = 127$)
- La valeur d'un nombre est donnée par l'expression:

$$(-1)^s \times (1 + \sum_{i=1}^{23} m_i 2^{-i}) \times 2^{e - E_{\max}}$$

- Exemple:

A = 1 10000010 00110000000000000000000

s = 1 (négatif)

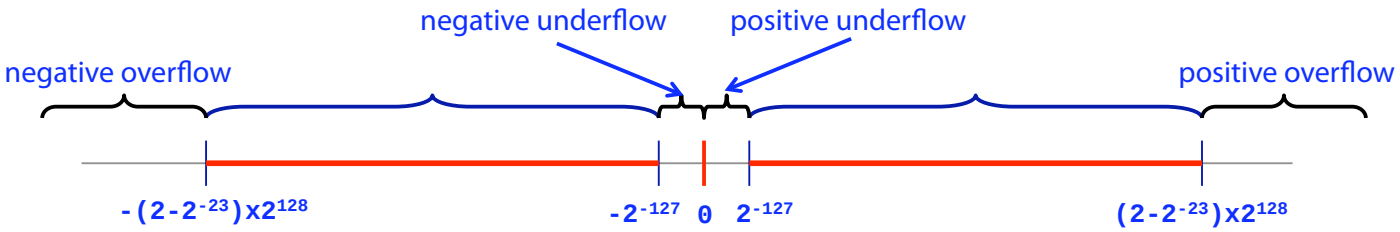
e = (2^7 + 2^1) - 127 = 130 - 127 = 3

m = 2^{-3} + 2^{-4} = 0.125 + 0.0625 = 0.1875

A = -1.1875x2^3 = -9.5

Eduardo Sanchez
Ecole Polytechnique Fédérale de Lausanne

- Le rang des nombres couverts en précision simple est:



- Il y a toujours un compromis entre le rang et la précision: si on augmente le nombre de bits de l'exposant, on étend le rang de nombres couverts. Mais, comme il y a toujours un nombre fixe de nombres que l'on peut représenter, la précision diminue. La seule façon d'augmenter le rang et la précision est d'augmenter le nombre total de bits du format

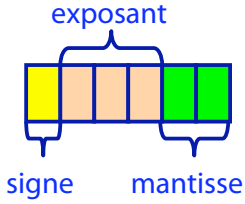
Eduardo Sanchez
Ecole Polytechnique Fédérale de Lausanne

- Précision double:
  - nombre de bits: 64
  - mantisse sur 52 bits
  - exposant sur 11 bits (biais = $2^{11-1} - 1 = 1023$)
- Dans les deux précisions il y a quelques valeurs particulières:
  - deux valeurs pour 0: les deux signes, avec exposant=mantisse=0
  - infini positif: signe positif, exposant=1...1, mantisse=0
  - infini négatif: signe négatif, exposant=1...1, mantisse=0
  - NaN (*not a number*): signe indifférent, exposant=1...1, mantisse≠0

Eduardo Sanchez
Ecole Polytechnique Fédérale de Lausanne

Exemple

- Supposons la représentation suivante:



- Le biais est égal à 3
- Il y a un total de 64 nombres représentés:

Eduardo Sanchez
Ecole Polytechnique Fédérale de Lausanne

0 000 00 = +0.0
0 000 01 = $+0.01 \times 2^{-2} = +0.0001$
0 000 10 = $+0.10 \times 2^{-2} = +0.001$
0 000 11 = $+0.11 \times 2^{-2} = +0.0011$
0 001 00 = $+1.00 \times 2^{-2} = +0.01$
0 001 01 = $+1.01 \times 2^{-2} = +0.0101$
0 001 10 = $+1.10 \times 2^{-2} = +0.011$
0 001 11 = $+1.11 \times 2^{-2} = +0.0111$
0 010 00 = $+1.00 \times 2^{-1} = +0.1$
0 010 01 = $+1.01 \times 2^{-1} = +0.101$
0 010 10 = $+1.10 \times 2^{-1} = +0.11$
0 010 11 = $+1.11 \times 2^{-1} = +0.111$
0 011 00 = $+1.00 \times 2^{0} = +1.0$
0 011 01 = $+1.01 \times 2^{0} = +1.01$
0 011 10 = $+1.10 \times 2^{0} = +1.1$
0 011 11 = $+1.11 \times 2^{0} = +1.11$

0 100 00 = $+1.00 \times 2^{1} = +10.0$
0 100 01 = $+1.01 \times 2^{1} = +10.1$
0 100 10 = $+1.10 \times 2^{1} = +11.0$
0 100 11 = $+1.11 \times 2^{1} = +11.1$
0 101 00 = $+1.00 \times 2^{2} = +100.0$
0 101 01 = $+1.01 \times 2^{2} = +101.0$
0 101 10 = $+1.10 \times 2^{2} = +110.0$
0 101 11 = $+1.11 \times 2^{2} = +111.0$
0 110 00 = $+1.00 \times 2^{3} = +1000.0$
0 110 01 = $+1.01 \times 2^{3} = +1010.0$
0 110 10 = $+1.10 \times 2^{3} = +1100.0$
0 110 11 = $+1.11 \times 2^{3} = +1110.0$
0 111 00 = +∞
0 111 01 = NaN
0 111 10 = NaN
0 111 11 = NaN

non normalisé

valeurs spéciales

Eduardo Sanchez
Ecole Polytechnique Fédérale de Lausanne

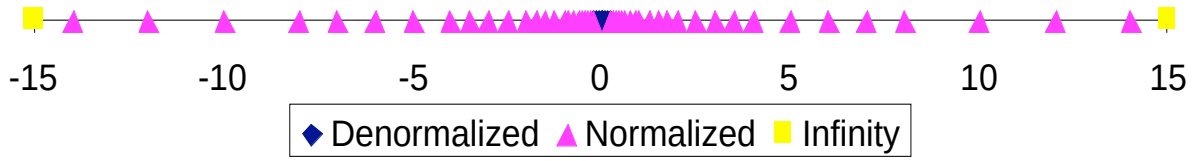
1 000 00 = -0.0
1 000 01 = -0.01x2-2 = -0.0001
1 000 10 = -0.10x2-2 = -0.001
1 000 11 = -0.11x2-2 = -0.0011
1 001 00 = $-1.00 \times 2^{-2} = -0.01$
1 001 01 = $-1.01 \times 2^{-2} = -0.0101$
1 001 10 = $-1.10 \times 2^{-2} = -0.011$
1 001 11 = $-1.11 \times 2^{-2} = -0.0111$
1 010 00 = $-1.00 \times 2^{-1} = -0.1$
1 010 01 = $-1.01 \times 2^{-1} = -0.101$
1 010 10 = $-1.10 \times 2^{-1} = -0.11$
1 010 11 = $-1.11 \times 2^{-1} = -0.111$
1 011 00 = $-1.00 \times 2^{0} = -1.0$
1 011 01 = $-1.01 \times 2^{0} = -1.01$
1 011 10 = $-1.10 \times 2^{0} = -1.1$
1 011 11 = $-1.11 \times 2^{0} = -1.11$

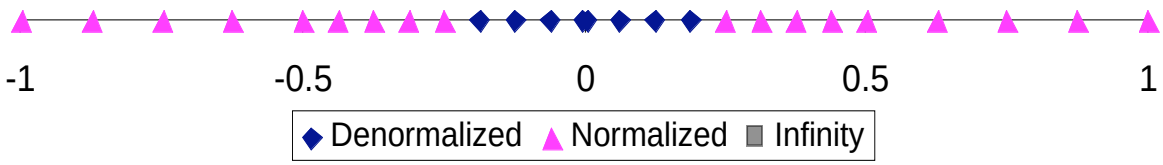
1 100 00 = $-1.00 \times 2^{1} = -10.0$
1 100 01 = $-1.01 \times 2^{1} = -10.1$
1 100 10 = $-1.10 \times 2^{1} = -11.0$
1 100 11 = $-1.11 \times 2^{1} = -11.1$
1 101 00 = $-1.00 \times 2^{2} = -100.0$
1 101 01 = $-1.01 \times 2^{2} = -101.0$
1 101 10 = $-1.10 \times 2^{2} = -110.0$
1 101 11 = $-1.11 \times 2^{2} = -111.0$
1 110 00 = $-1.00 \times 2^{3} = -1000.0$
1 110 01 = $-1.01 \times 2^{3} = -1010.0$
1 110 10 = $-1.10 \times 2^{3} = -1100.0$
1 110 11 = $-1.11 \times 2^{3} = -1110.0$
1 111 00 = -∞
1 111 01 = NaN
1 111 10 = NaN
1 111 11 = NaN

non normalisé

valeurs spéciales

Eduardo Sanchez
Ecole Polytechnique Fédérale de Lausanne





Eduardo Sanchez
Ecole Polytechnique Fédérale de Lausanne