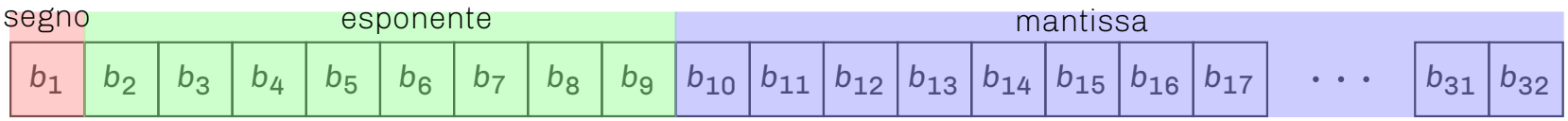
Algoritmi Numerici (Parte I)

[Lezione 5] Formato float, mantissa denormalizzata ed errore di rappresentazione

Alessandro Antonucci
alessandro.antonucci@supsi.ch

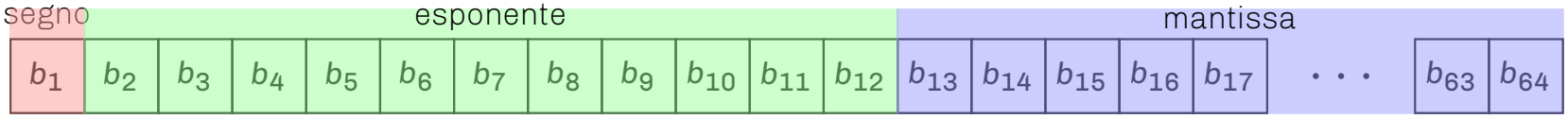
Il formato `float` a 32 bit (ripasso)



- IF $b_1 = 0$ THEN segno = +1

  ELSE segno = -1
- esponente = $\texttt{horner}(b_2 b_3 \ldots b_9)$ - 127
- mantissa = $1.[b_{10} b_{11} \ldots b_{32}]_2$
- numero = segno $\cdot$ mantissa $\cdot$ $2^{\text{esponente}}$
- RETURN numero

Il formato `double` a 64 bit



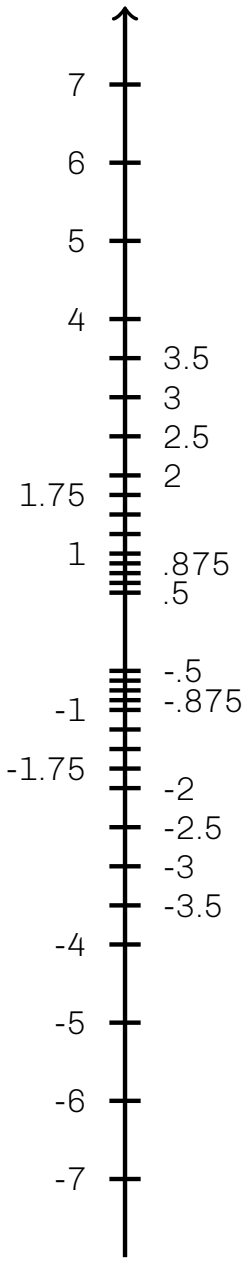
- 1/11/52 bits per segno/esponente/mantissa (anziché 1/8/23)
- Esponente - 1023 ($= \mathbf{2^{11-1} - 1}$ anziché $\mathbf{127} = 2^{8-1} - 1$)

- IF $b_1 = \mathbf{0}$ THEN segno = +1 ELSE segno = -1
- esponente = $\mathtt{horner}(b_2 b_3 \ldots b_{12})$ - 1023
- mantissa = $\mathbf{1}.[b_{13} b_{14} \ldots b_{64}]_2$

Un formato float-like a 5 bit

1/2/2 bit per segno/espo/mantissa, espo - $2^{2-1} - 1 = 1$

$$0|00|00 \rightarrow +1.00_2 \cdot 2^{-1} = 0.5$$
$$0|00|01 \rightarrow +1.01_2 \cdot 2^{-1} = 0.625$$
$$0|00|10 \rightarrow +1.10_2 \cdot 2^{-1} = 0.75$$
$$0|00|11 \rightarrow +1.11_2 \cdot 2^{-1} = 0.875$$
$$0|01|00 \rightarrow +1.00_2 \cdot 2^{0} = 1$$
$$0|01|01 \rightarrow +1.01_2 \cdot 2^{0} = 1.25$$
$$0|01|10 \rightarrow +1.10_2 \cdot 2^{0} = 1.5$$
$$0|01|11 \rightarrow +1.11_2 \cdot 2^{0} = 1.75$$
$$0|10|00 \rightarrow +1.00_2 \cdot 2^{1} = 2$$
$$0|10|01 \rightarrow +1.01_2 \cdot 2^{1} = 2.5$$
$$0|10|10 \rightarrow +1.10_2 \cdot 2^{1} = 3$$
$$0|10|11 \rightarrow +1.11_2 \cdot 2^{1} = 3.5$$
$$0|11|00 \rightarrow +1.00_2 \cdot 2^{2} = 4$$
$$0|11|01 \rightarrow +1.01_2 \cdot 2^{2} = 5$$
$$0|11|10 \rightarrow +1.10_2 \cdot 2^{2} = 6$$
$$0|11|11 \rightarrow +1.11_2 \cdot 2^{2} = 7$$



I formati float-like

Vantaggio

Numeri macchina vicini fra loro vicino allo zero
più distanti per numeri grandi (in valore assoluto)

Errore di rappresentazione cresce in termini assoluti,
costante in termini relativi

Svantaggio

numero macchina più piccolo in valore assoluto $\neq 0$

$0|00000000|00\ldots00 = +1.0 \cdot 2^{-127} = \frac{1}{2^{127}} \neq 0$

$1|00000000|00\ldots00 = -1.0 \cdot 2^{-127} = -\frac{1}{2^{127}}$

Mantissa denormalizzata

Eccezione al formato `float` per avere **0** numero macchina

- IF $(b_2, \ldots, b_9) = (\mathbf{000000000})$ THEN
- mantissa = $\mathbf{0}.[b_{10}b_{11}\ldots b_{32}]_2$ (denormalizzata)
- esponente = -126 (e non 0-127=-127)

Così $\mathbf{0}|\mathbf{00000000}|\mathbf{0}\ldots\mathbf{0} = +\mathbf{0.0}$ e
$\mathbf{1}|\mathbf{00000000}|\mathbf{0}\ldots\mathbf{0} = -\mathbf{0.0}$

- Zero macchina ($\mathbf{0}_m$) più piccolo numero macchina (non zero)
- Mantissa denormalizzata produce $\mathbf{0}_m$ molto più piccolo
- Stessa cosa per **`double`**, con -1022
- Se $(b_2, \ldots, b_9) = (\mathbf{111111111})$ eccezioni per $\pm\infty$ e ***NaN***

Stima dell'errore di rappresentazione

- Rappresentazione float $x_m = \pm 1.[b_{10}, b_{11}, \ldots, b_{32}]_2 \cdot 2^p$
- "Vero" numero $x = \pm 1.[b_{10}b_{11}\ldots b_{32}b_{33}b_{34}\ldots]_2 \cdot 2^p$
- Se il formato float facesse troncamento:

  $\epsilon_a = |x - x_m| = |0.[00\ldots 0 b_{33}b_{34}\ldots]_2| \cdot 2^p$
- $\epsilon_a = |x - x_m| = |0.[b_{33}b_{34}\ldots]_2| 2^{p-23} < 2^{p-23}$
- $\epsilon_r = |\frac{x_m - x}{x}| = \frac{\epsilon_a}{|x|} < \frac{2^{p-24}}{|x|} < 2^{-23}$ perché $x \geq 1.0000 \cdot 2^p$
- Il formato float arrotonda, le stime per gli errori sono esattamente la metà!

Stima dell'errore di rappresentazione (ii)

- Se p è l'esponente del numero da rappresentare (in base 2 e con mantissa normalizzata)
- Se s è il numero di bit a disposizione per la mantissa

con TRONCAMENTO	con ARROTONDAMENTO
$\epsilon_a < 2^{p-s}$	$\epsilon_a < 2^{p-s-1}$
$\epsilon_r < 2^{-s}$	$\epsilon_r < 2^{-s-1}$

Es. `float` $\epsilon_r < 2^{-24} \simeq 6 \cdot 10^{-8}$, 7/8 cifre corrette in decimale

Es. `double` $\epsilon_r < 2^{-53} \simeq 1.1 \cdot 10^{-16}$, 15/16 cifre corrette

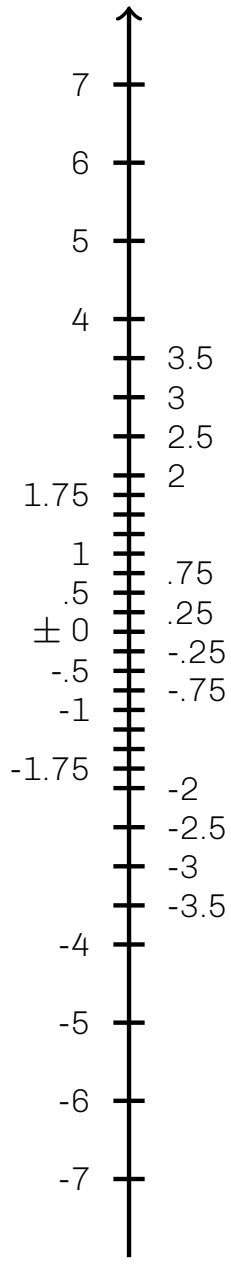
Esercizio

Analizza come si modifica il formato a 5 bit introdotto precedentemente quando si considera anche l'eccezione per la mantissa denormalizzata

Soluzione

1/2/2 bit per segno/espo/mantissa, espo - $2^{2-1} - 1 = 1$

$$0|00|00 \rightarrow +0.00_2 \cdot 2^0 = 0.0$$
$$0|00|01 \rightarrow +0.01_2 \cdot 2^0 = 0.25$$
$$0|00|10 \rightarrow +0.10_2 \cdot 2^0 = 0.5$$
$$0|00|11 \rightarrow +0.11_2 \cdot 2^0 = 0.75$$
$$0|01|00 \rightarrow +1.00_2 \cdot 2^0 = 1$$
$$0|01|01 \rightarrow +1.01_2 \cdot 2^0 = 1.25$$
$$0|01|10 \rightarrow +1.10_2 \cdot 2^0 = 1.5$$
$$0|01|11 \rightarrow +1.11_2 \cdot 2^0 = 1.75$$
$$0|10|00 \rightarrow +1.00_2 \cdot 2^1 = 2$$
$$0|10|01 \rightarrow +1.01_2 \cdot 2^1 = 2.5$$
$$0|10|10 \rightarrow +1.10_2 \cdot 2^1 = 3$$
$$0|10|11 \rightarrow +1.11_2 \cdot 2^1 = 3.5$$
$$0|11|00 \rightarrow +1.00_2 \cdot 2^2 = 4$$
$$0|11|01 \rightarrow +1.01_2 \cdot 2^2 = 5$$
$$0|11|10 \rightarrow +1.10_2 \cdot 2^2 = 6$$
$$0|11|11 \rightarrow +1.11_2 \cdot 2^2 = 7$$



Esercizio 1

- Scrivere la sequenza che codifica il numero x=50.02 secondo le regole del formato `float`
- Se il numero non è un numero macchina calcolare l'errore assoluto verificando che sia inferiore al valore di stima pessimistica
- Ripetere la stessa analisi per l'errore relativo

Esercizio 1 (i)

Converto separatamente il 50 ed il .02 in base 2

50 mod 2 = 0
25 mod 2 = 1
12 mod 2 = 0
6 mod 2 = 0
3 mod 2 = 1
1 mod 2 = 1

$$\mathbf{50 = 110010_2}$$

int(.02 x 2) = 0
int(.04. x 2) = 0
int(.08. x 2) = 0
int(.16 x 2) = 0
int(.32 x 2) = 0
int(.64 x 2) = 1
int(.28 x 2) = 0
int(.56 x 2) = 1
int(.12 x 2) = 0
int(.24 x 2) = 0
int(.48 x 2) = 0
int(.96 x 2) = 1
int(.92 x 2) = 1
int(.84 x 2) = 1
int(.68 x 2) = 1
int(.36 x 2) = 0
int(.72 x 2) = 1
int(.44 x 2) = 0
int(.88 x 2) = 1
int(.76 x 2) = 1
int(.52 x 2) = 1
int(.04 x 2) =

$$\mathbf{0.02 = 0.\overline{00000101000111101011}_2}$$

Esercizio 1 (ii)

$50.02 = \mathbf{110010.0\overline{0000101000111101011 1}_2}$

In notazione scientifica

$+\mathbf{1.10010\overline{0000101000111101011 1} \cdot 2^5}$

Esponente 5 come qualcosa meno 127, il valore è 132

132 come un numero naturale in base 2 ad 8 bit

132 mod 2 = 0
66 mod 2 = 0
33 mod 2 = 1
16 mod 2 = 0
8 mod 2 = 0
4 mod 2 = 0
2 mod 2 = 0
1 mod 2 = 1

132 = 1000|0100

Esercizio 1 (iii)

in grigio i bit della mantissa dopo il 23-esimo

$$+\mathbf{1.10010000010100011110101}\overline{\mathbf{111}} \cdot \mathbf{2}^{\mathrm{Horner}(\mathbf{1000}|\mathbf{0100})-\mathbf{127}}$$

Con troncamento, approssimo per difetto:

$$+\mathbf{1.10010000010100011110101} \cdot \mathbf{2}^{\mathrm{Horner}(\mathbf{1000}|\mathbf{0100})-\mathbf{127}}$$

Ma il formato float fa arrotondamento,
approssimo per eccesso (primo bit fuori uno):

$$\boldsymbol{x} = +\mathbf{1.10010000010100011110110} \cdot \mathbf{2}^{\mathrm{Horner}(\mathbf{1000}|\mathbf{0100})-\mathbf{127}}$$

ovvero

$$0|10000100|10010000010100011110110 \rightarrow 4248147B$$

Esercizio 1 (iv)

Per valutare l'errore assoluto leggo il numero macchina:

0|10000100|10010000001010001111011

$+\mathbf{1.10010000001010001111011} \cdot \mathbf{2^5} =$

110010.00000101000111101 1

A sx della virgola 50

A dx (in esadecimale) .0000|0101|0001|1110|1100=.051EC

Esercizio 1 (v)

C/16 + E = 12/16+14 = 14.75
14.75/16+1 = 1.921785
1.921785/16+5 = 5.1201171875
5.1201171875/16+0 = 0.32000732421875
0.32000732421875/16 = 0.020000457763671875

Il numero macchina e' quindi 50.020000457763671875

(leggermente piu' grande di 50.02 in seguito
all'approssimazione per eccesso)

L'errore "esatto" è quindi
$\mathbf{0.000000457763671875 = 4.577636718751 0^{-7}}$

minore del valore di stima pessimistico (errore assoluto per arrotondamento con 23 bit di mantissa)
$\mathbf{2^{p-s-1} = 2^{5-23-1} = 2^{-19} = \frac{1}{524288} \simeq 1.9073 \cdot 10^{-6}}$

Esercizio 1 (vi)

Per l'errore relativo

Valore "esatto"
$\epsilon_r = 0.00000045776367 1875/50.02 \simeq$
$9.151612792383045 \cdot 10^{-9}$

Valore pessimistico $2^{-24} \simeq 5.960464477539063 \cdot 10^{-8}$

La formula è rispettata

Esercizio 2

Scrivere il numero $-0.5 \cdot 2^{-128}$ secondo le regole del formato float.

$-0.5 \cdot 2^{-128} = -.1_2 2^{-128} = -1.0 2^{-129}$

L'esponente -129 non si può esprimere come $k - 127$

Devo lavorare con mantissa denormalizzata
$-1.0 \cdot 2^{-3} \cdot 2^{-126} = -0.001 \cdot 2^{-126}$

sequenza bit $1|00000000|0010\ldots0 \rightarrow 80010000$

Esercizio 3

Scrivere il numero $2^{-128} + 2^{-150}$ secondo le regole del formato float.

Raccolgo 2^{-128}:

$$2^{-128}(1 + 2^{-22}) = +1.\overbrace{00\ldots00}^{21 zeri}1 \cdot 2^{-128}$$

Esponente -128: troppo piccolo per mantissa normalizzata!

Uso mantissa denormalizzata, esponente $= -126$, virgola mantissa si sposta di due posizioni a sx per compensare:

$$+0.01\overbrace{00\ldots00}^{21 zeri}1 \cdot 2^{-126}$$

24 bit dopo virgola, approssimo a: $+0.\overbrace{0100\ldots01}^{23 bits} \cdot 2^{-126}$

Sequenza 32 bits: 0|00000000|01000000000000000000001