Rappresentazione di numeri reali

Argomento:

- Rappresentazione a virgola fissa
- Rappresentazione a virgola mobile

Materiale di studio:

- Capitolo 10: Sezione 10.4

Fixed-point

- Sia R un numero reale

$$\boldsymbol{R = (I.F)_2}$$

- I: parte intera; F: parte frazionaria (in base 2)

- Nella rappresentazione fixed-point si usano k_I bit per parte intera, k_F bit per parte frazionaria
  - $k = k_I + k_F$ bit totali
  - Il numero di bit riservati a I e a F non dipende da R: il punto decimale è fisso.

Fixed-point

- Viene rappresentato in complemento a 2 il numero intero $\mathrm{R}' = \mathrm{R} \cdot 2^{k_F}$ (approssimazione per troncamento o arrotondamento) usando N bits.

- Esempio: $\mathrm{R} = 5.3 = (101.0\overline{1001})_2$
- La rappresentazione fixed-point con $k_I = 4$ e $k_F = 5$ è:

010101001

Encoding: da numero reale fixed-point

Un numero reale q è rappresentato dalla stringa di bit $R_{Fix,k,k_F}(q) = (x_{k-1}, \dots x_0)$ con

$$(x_{k-1} \dots\ x_0) = R_{C2,k}(int\big(q\ \cdot 2^{k_F}\big))$$

La funzione $int()$ trasforma un numero reale in un numero intero tramite troncamento o arrotondamento.

Decoding: da fixed-point a numero reale

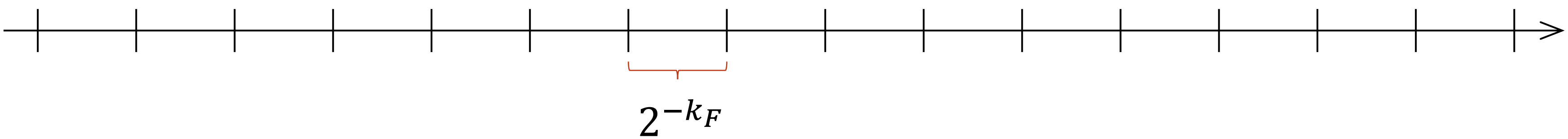
La stringa di bit $x = (x_{k-1}, \dots x_0)$ rappresenta il numero reale:

$$R_{Fix,k,k_F}^{-1}(x) = \frac{-x_{k-1}2^{k-1} + \sum_{i=0}^{k-1} x_i \cdot 2^i}{2^{k_F}}$$

Approssimazione con fixed point

Errore assoluto err$_a$ di approssimazione con k_F bit frazionari:

- Definito come: $err_a = |\, R' - R|$
- Con arrotondamento: $err_a \leq 2^{-(k_F+1)}$
- Con troncamento: $err_a \leq 2^{-k_F}$



Errore relativo err$_r$:

- Definito come $err_r = \dfrac{err_a}{|R|} \leq \dfrac{1}{|R|2^{k_F}}$
- L'errore relativo diminuisce con magnitudine di R

Floating-point

- Rappresentazione a virgola mobile: Floating-Point (standard IEEE 754)

- R: numero frazionale diverso da 0
- Esprimiamo R tramite un prodotto:

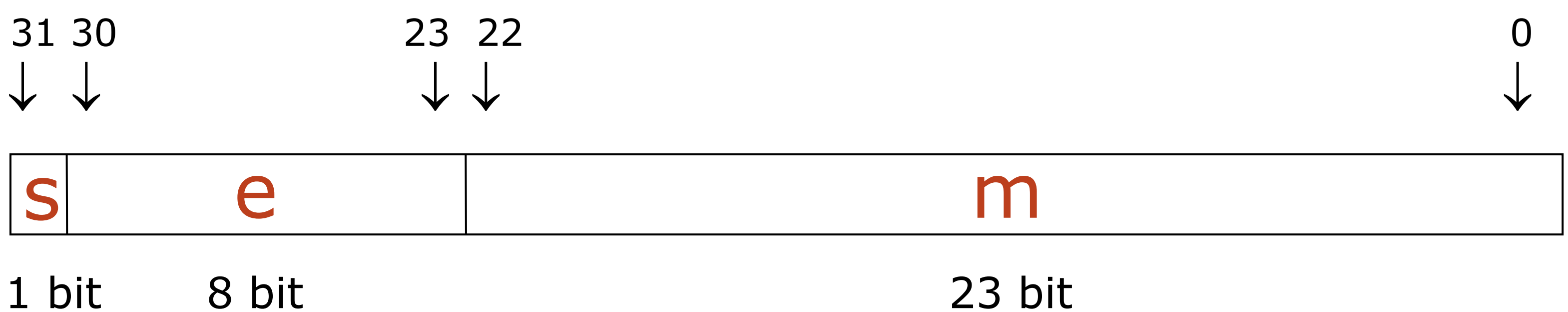
$$R = M \cdot 2^E = (1.\overline{M})_2 \cdot 2^E$$

- $\mathrm{M} = (1.\overline{M})_2$: mantissa normalizzata di R
- E: caratteristica di R

Encoding: da numero reale a floating-point

- Un numero reale q è rappresentato dalla stringa di 32 bit $R_{Float}(q) = (x_{31}, \dots x_0)$ con

- x_{31}: 1 bit di segno s
- $x_{30} \dots x_{23}$: 8 bit e per rappresentare la caratteristica E
- $x_{22} \dots x_0$: 23 bit m per rappresentare la mantissa normalizzata M



Encoding della mantissa M

- La mantissa normalizzata M viene rappresentata per ampiezza e segno
  - Bit s per il segno
  - I 23 bit m codificano solo la parte frazionaria $\bar{M}$: la parte intera di M non viene rappresentata perché sempre a 1

- Se $\bar{M}$ eccede 23 bit
  - Lo standard IEEE-754 prevede arrotondamento al valore più vicino
  - Per semplicità, noi tronchiamo il numero a 23 bit

Encoding della caratteristica

- La caratteristica E viene rappresentata in eccesso 127 su 8 bit e

- $e = R_{E,127,8}(E) = R_{N,8}(E + 127)$

Esempio di floating-point

$$R = M \cdot 2^E$$

Es: $0.3_{10} = 0.0\overline{1001}_2 = 1.\overline{0011}_2 \cdot 2^{-2}$

$M = +1.\overline{0011}$ $s = 0$ $m = \overline{0011}$

$E = -2 = e-127$ $e = 125_{10} = 01111101_2$

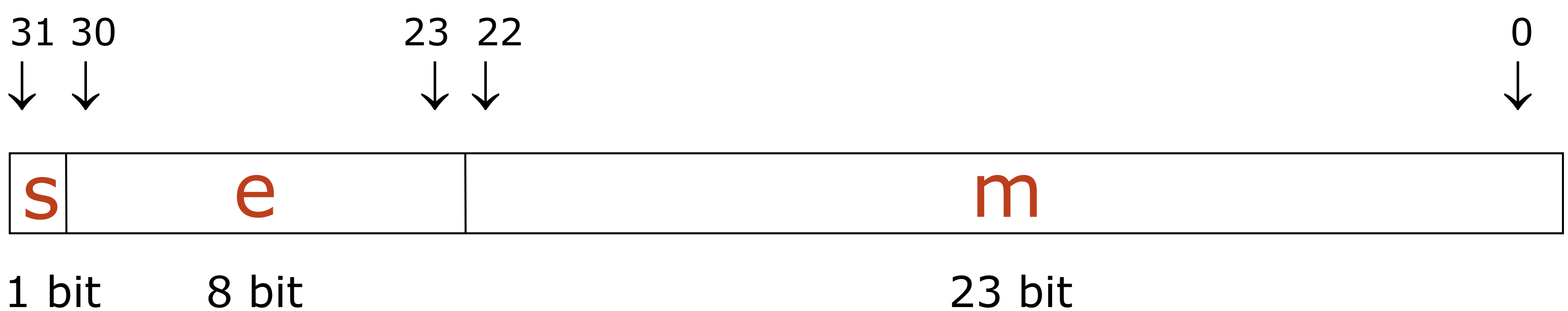
31	30 … 23	22 … 0
0	01111101	00110011001100110011001
1	8 bit	23 bit

In esadecimale: **3E999999**

Decoding: da floating-point a numero reale

La stringa di bit $x = (x_{31}, \dots x_0)$ rappresenta il numero reale:

$$R_{Float}^{-1}(x) = (1 - 2s) \cdot (1.m)_2 \cdot 2^{(e)_2 - 127}.$$



Casi speciali

I valori estremi dell'esponente sono riservati per situazioni particolari:

- $e = 0$
  - Se $m = 0$: il numero 0
  - Se $m \neq 0$: numero non normalizzato moltiplicato per 2^{-126}

$$R_{Float}^{-1}\,(x) = (1 - 2s)\;0.m\;\cdot 2^{-126}$$

- $e = 255$
  - Se $m = 0$: il numero ∞
  - Se $m \neq 0$: numero non valido NaN

Intervallo dei numeri normalizzati rappresentabili

Mantissa: $M = 1.m$

$$1.00\ldots0_2 \leq |M| \leq 1.11\ldots1_2 \rightarrow 1 \leq |M| \leq 2 - 2^{-23}$$

Esponente: $E = e - 127$ (il valore $e = 0\ o\ 255$ non sono usati)

$$0 < e < 2^8 - 1 \Rightarrow -126 \leq E \leq 127$$

Estremi dei floating point

Il numero $P \neq 0$ con modulo più piccolo:

- Con mantissa normalizzata

$$|P| = 1 \cdot 2^{-126} \approx 1.1810 \cdot 10^{-38}$$

- Con mantissa non normalizzata

$$|P| = 0.0 \ldots 01 \cdot 2^{-126} = 2^{-23} \cdot 2^{-126} = 2^{-149} \approx 1.410 \cdot 10^{-45}$$

Il numero $\mathrm{G} \neq \infty$ con modulo più grande:

$$|G| = (2 - 2^{-23}) \cdot 2^{127} \approx 3.410 \cdot 10^{38}$$

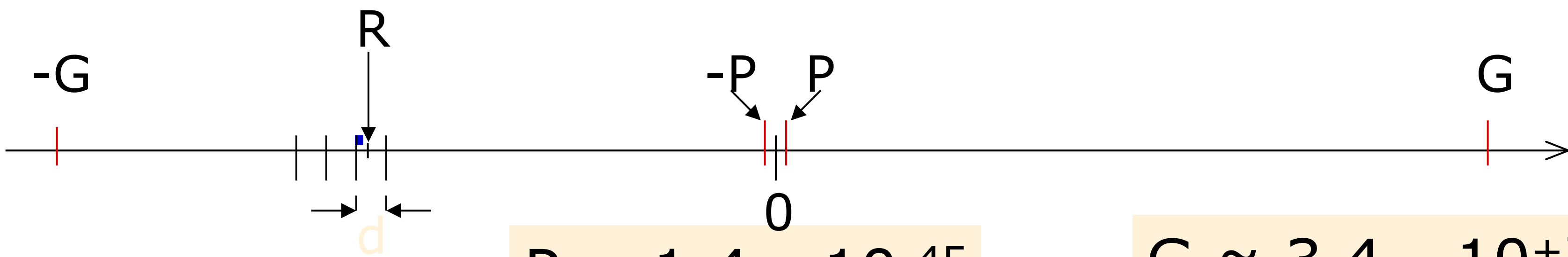
Estremi dei floating point

Esempi di grandezze fisiche “estreme” i cui valori sono “gestibili” con questa rappresentazione f.p.:

- distanza terra - quasar: $\approx 10^{27}$ m
- dimensione quark: $\approx 10^{-18}$ m

Errore di approssimazione

Intervallo dei numeri rappresentabili sull'asse reale:



$P \approx 1.4_{10} \cdot 10^{-45}$

$G \approx 3.4_{10} \cdot 10^{+38}$

Degli infiniti numeri reali compresi tra -G e G sono rappresentabili solo 2^{32} ($\approx 4 \cdot 10^{9}$) numeri razionali

Errore assoluto

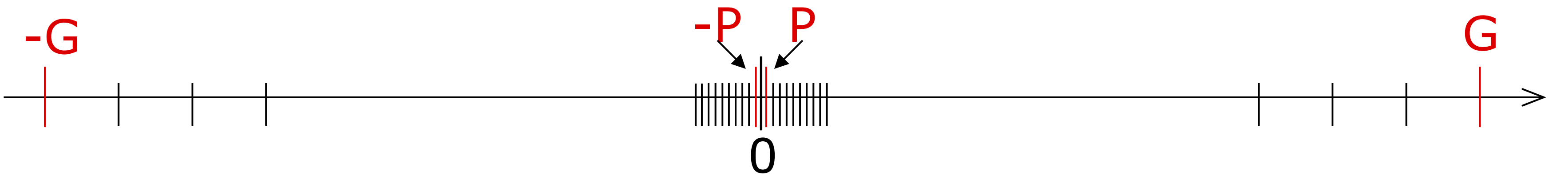
- Un numero reale R è rappresentato dal più vicino di questi numeri razionali

- La distanza d tra due numeri rappresentabili consecutivi è:

$$d = 2^{-23} \cdot 2^{E} = 2^{E-23}$$

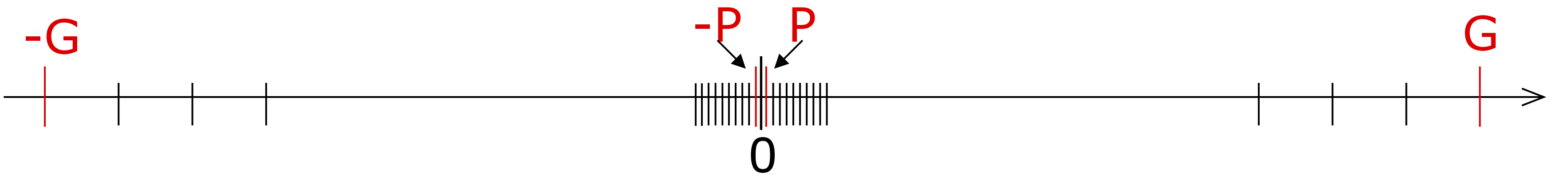
- L'errore assoluto di approssimazione è
  - $err_a \leq d/2 = 2^{E-24}$ se si usa l'arrotondamento al numero più vicino
  - $err_a \leq d = 2^{E-23}$ se si usa il troncamento

Errore assoluto



- L'errore assoluto di approssimazione err_a è funzione di E:
    - $err_a \leq 2^{E-24}$
    - L'errore è piccolo vicino allo 0: $err_a \sim 2^{-126-24} = 2^{-150}$
    - L'errore è grande vicino a $\pm$ G: $err_a \sim 2^{+127-24} = 2^{+103}$

Errore relativo



- Ciò che più interessa è l'errore relativo **err_r**

$$err_r = \frac{err_a}{|R|} = \frac{2^{E-24}}{|M|2^E} = \frac{2^{-24}}{|M|} \leq 2^{-24} \qquad (1 \leq |M| < 2)$$

(Per numeri normalizzati)

Double precision floating point

PRECISIONE DOPPIA: 64 bit

$$R = M\ 2^{E}$$

63	62 … 52	51 … 0
s	e	m
1 bit	11 bit	52 bit

$$M = (1-2s) \cdot 1.m$$

$$E = e - 1023$$

Double precision floating point (2)

- Sono rappresentabili solo 2^{64} = 16 E (EXA = 10^{18}) degli infiniti numeri reali compresi tra -G e G.
- Valore più piccolo: P ≈ $2.2_{10} \cdot 10^{-308}$
- Valore più grande: G ≈ $1.8_{10} \cdot 10^{+308}$
- Errore relativo $err_r \approx 10^{-17}$
- Precisione ≈ 16 cifre decimali significative.

- Esempio di grandezza fisica estrema gestibile con questa rappresentazione f.p.:
    - numero di particelle subatomiche nell'universo: ≈ 10^{80}

Pentium FDIV bug

- Problema con divisione in floating point

$$\frac{4{,}195{,}835}{3{,}145{,}727} = 1.333820449136241002$$

$$\frac{4{,}195{,}835}{3{,}145{,}727} = 1.333739068902037589$$

- 4,195,835 = 0x4005FB e 3,145,727 = 0x2FFFFF: il '5' in 0x4005 causava un errore nella logica FPU
- Scoperto nel 1994 da Thomas Nicely durante il calcolo di una costante (ma Intel già lo sapeva)
- Intel annuncia "a pre-tax charge of $475 million against earnings” che è stata associata al costo della sostituzione dei processori.