IEEE 754

Lo **standard IEEE per il calcolo in virgola mobile** (**IEEE 754**) (ufficialmente: **IEEE Standard for Binary Floating-Point Arithmetic (ANSI/IEEE Std 754-1985)** o anche **IEC 60559:1989, Binary floating-point arithmetic for microprocessor systems**) è lo standard più diffuso nel campo del calcolo automatico. Questo standard definisce il formato per la rappresentazione dei numeri in virgola mobile (compreso ±0 e i numeri denormalizzati; gli infiniti e i NaN, "*not a number*"), ed un set di operazioni effettuabili su questi. Specifica inoltre quattro metodi di arrotondamento e ne descrive cinque eccezioni.

Esistono in questo standard quattro formati per i numeri in virgola mobile: a *precisione singola* (32 bit), *precisione doppia* (64 bit), *precisione singola estesa* (≥ 43 bit), raramente usato, e *precisione doppia estesa* (≥ 79 bit), supportata solitamente con 80 bit. La precisione singola è il minimo richiesto dallo standard, gli altri sono optional.

Struttura di un numero in virgola mobile

Un numero in virgola mobile, secondo lo standard IEEE è rappresentato su parole di 32, 64 o 80 bit divisi in tre parti:

- un bit di **segno** *s*;
- un campo di **esponente** *e*;
- un campo di **mantissa** *m*

in questo ordine. I bit di una parola di *n* bit sono indicizzati in modo decrescente con numeri interi da 0 a *n*-1. In un numero in questo standard, l'importanza del bit decresce col suo indice.

Numeri a precisione singola (32 bit)

Di seguito è rappresentato un numero in una parola di 32 bit:

```
 1     8               23               lunghezza in bit
+-+--------+-----------------------+
|S|  Esp.  |  Mantissa             |
+-+--------+-----------------------+
31 30      22                      0    indice dei bit
```

Il valore del numero rappresentato è calcolabile come:

$$(-1)^s \times 2^E \times M$$

Il campo *s* specifica il segno del numero: 0 per i numeri positivi, 1 per i numeri negativi. Il campo *e* contiene l'esponente del numero in forma intera. Essendo costituito da 8 bit, permette di rappresentare 256 valori. Ai valori 0 e 255 vengono riservati per funzioni speciali (descritte in seguito); gli altri permettono di rappresentare 254 valori per i numeri in forma *normale*, compresi tra -126 e 127, dato che questo campo deve poter rappresentare sia numeri enormi che minimi; tuttavia, adoperando il metodo usato per la rappresentazione del segno dei numeri interi, si creerebbero problemi per il confronto tra numeri. Per risolvere questo problema, il campo è rappresentato in eccesso *k* detto bias, per cui:

$e = E + k$

e reciprocamente

$E = e - k$

In questo standard, per i numeri a precisione singola, il *bias* è uguale a 127. In questo modo, i valori dell'esponente compresi tra -126 e 127 assumono invece, nella scrittura del byte, i valori compresi tra 1 e 254, eliminando la necessità di un bit riservato al segno. In fase di decodifica del numero, il *bias* viene nuovamente sottratto per

recuperare il valore originale.

I valori assunti dall'esponente *e* e dalla mantissa *m* determinano l'appartenenza del numero ad una di queste categorie:

- zeri;
- numeri in forma normale;
- numeri in forma denormalizzata;
- infiniti;
- Nan (*not a number*).

L'esponente distingue i numeri in modo primario, la mantissa in modo secondario.

Categoria	Esp.	Mantissa
Zeri	0	0
Numeri denormalizzati	0	non zero
Numeri normalizzati	1-254	qualunque
Infiniti	255	0
Nan (not a number)	255	non zero

Il campo *m* è una stringa di bit che rappresenta la sequenza di cifre dopo la virgola. Tutte le mantisse sono normalizzate in modo che il numero prima della virgola sia 1, per cui per un dato *m* il valore matematico corrispondente è

$M=1,m$

In pratica, la mantissa è costituita dal numero binario **1**, seguito dalla virgola e dalla parte intera del numero rappresentato, in forma binaria; la mantissa risulta così artificialmente compresa tra 1 e 2. Quando un numero è normalizzato, come risulta dal suo esponente, il primo bit della mantissa, pari a 1, viene omesso per convenienza: viene quindi chiamato **bit nascosto**, o **bit implicito**.

Con questo sistema di rappresentazione, si hanno due zeri (+0 e −0) e due infiniti(+∞ e −∞) a seconda del valore del primo bit; e che i numeri subnormali possono avere un segno e una mantissa, utili però solo per l'analisi.

Questo sistema di rappresentazione permette di avere una precisione relativa *x* quasi costante per tutti i valori rappresentabili. Infatti

$$2^{-\text{lunghezza}(m-1)} \leq x \leq 2^{\text{lunghezza}(m)}$$

Facciamo un semplice esempio: codifichiamo il numero −118.625 nel sistema IEEE 754.

Dobbiamo determinarne il segno, l'esponente e la mantissa.

Poiché è un numero negativo, il primo bit è "1".

Poi scriviamo il numero in forma binaria: 1110110.101.

Successivamente spostiamo la virgola verso sinistra, lasciando solo un 1 alla sua sinistra: 1110110,101 = 1,110110101·2^6

La mantissa è la parte a destra della virgola, riempita con zeri a destra fino a riempire i 23 bit: 11011010100000000000000.

L'esponente è pari a 6, ma dobbiamo convertirlo in forma binaria e adattarlo allo standard. Per la precisione singola, dobbiamo aggiungere 127. Quindi 6 + 127 = 133. In forma binaria: 10000101.

Assemblando il tutto:

```
 1     8               23
+-+--------+-----------------------+
```

```
|S|  Exp    |  Fraction             |
|1|10000101|11011010100000000000000|
+-+--------+-----------------------+
31 30      22                      0
```

Precisione doppia (64 bit)

La precisione doppia è molto simile alla singola, in dimensioni maggiori:

```
 1     11                                  52
+-+-----------+-------------------------------------------------------+
|S|  Exp      |  mantissa                                             |
+-+-----------+-------------------------------------------------------+
63 62        51                                                      0
```

I NaN e gli infiniti sono rappresentati con esponenti formati da una serie di 1 (pari a 2047).

Per i numeri normalizzati il bias è pari a 1023 (quindi e = E − 1023). Per i numeri denormalizzati l'esponente è −1022 (il minimo esponente per un numero normalizzato). Come prima, sia gli infiniti che gli zeri possono essere rappresentati con entrambi i segni.

La precisione decimale è di circa 16 cifre decimali.

Precisione quadrupla (128 bit)

La precisione quadrupla è molto simile alla doppia, in dimensioni maggiori:

```
 1     15
112
+-+---------------+-------------------------------------------------------------------------------
|S|  Exp          |  mantissa

|
+-+---------------+-------------------------------------------------------------------------------
```

Il numero di bits dedicati all'esponente sale da 11 a 15 e quelli per la mantissa a 112.

La precisione totale sale a circa 34 cifre decimali.

Revisione dello standard

Lo standard è attualmente sotto revisione (IEEE 754r).

Voci correlate

- Bias (distorsione)
- Virgola mobile

Fonti e autori delle voci

IEEE 754 *Fonte*: http://it.wikipedia.org/w/index.php?oldid=27906215 *Autori*:: Beta16, Biopresto, Brunocip, Engineer123, Fabexplosive, Hellis, Jaqen, La Corona, Laurentius, Otrebla86, Pegasovagante, Ronnie28, Sbazzone, Slim8shady9, Square87, Stiffmaister, TierrayLibertad, 26 Modifiche anonime

Licenza