

Numeri reali

Sistema in virgola mobile (floating point)

- Un numero reale X può essere espresso nella forma

$$X = f \times 10^E$$

- dove, f è detta **mantissa**, E **esponente**

$$X = 0,314 \times 10^1$$

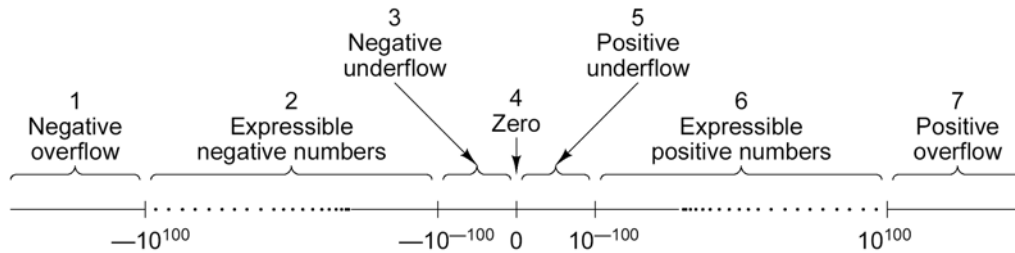
- Fissato X, la virgola si sposta a sinistra (destra) se E aumenta (diminuisce)

$$- \quad X = 3,14 \times 10^0 = 0,314 \times 10^1 = 0,0314 \times 10^2$$

- Per evitare ambiguità si dovrà definire una forma normalizzata...

Caratteristiche

- Ammettiamo di disporre di $p = 3$ cifre decimali per esprimere f e $k=2$ per E . Inoltre usiamo i segni $+$ e $-$
 - $|f| \in [0.1, 0.101, \dots \dots 0.999]$;
 - $|E| \in [0..99]$



Calcolatori Elettronici, Beraldi, aa 04/05

IEEE Floating Point 754

Nozioni di base

- Standard IEEE 754
 - Introdotto nel 1985 come standard per aritmetica in virgola mobile per consentire portabilità dei programmi
 - Praticamente supportato da tutte le CPU
- Necessità per calcoli numerici
 - Standard per rounding, overflow, underflow
- Formato base
 - Precisione singola (32 bit)
 - Precisione doppia (64 bit)
- Formato esteso
 - Precisione singola estesa (≥ 44 bit)
 - Precisione doppia estesa (≥ 80 bit)

IEEE Floating Point 754

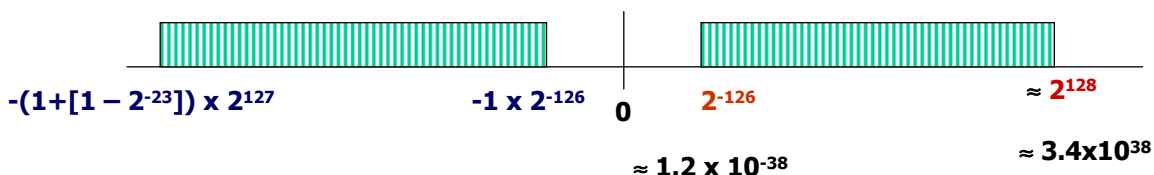
Formati

- Esistono due forme: forma normalizzata e forma denormalizzata
- Forma normalizzata: $(-1)^s \times 1.F \times 2^{\text{EXP}-\text{Bias}}$
 - s è il bit di segno
 - Il primo bit della mantissa è sempre 1, e non è rappresentato (bit implicito)
 - $F=\{0,\dots,111\dots1\}$
 - Il valore $1.F$ è detto anche **significando**
 - L'esponente E è dato in forma polarizzata, ossia si memorizza $\text{EXP}=E+\text{Bias}$ (Bias è la costante di polarizzazione)
- Forma denormalizzata: $(-1)^s \times F \times 2^{\text{Emin}}$
- Un numero è quindi espresso mediante la tripla di valori $\langle S, F, \text{EXP} \rangle$
 - Un valore di $\text{EXP}=0$ indica che il numero è in forma denormalizzata

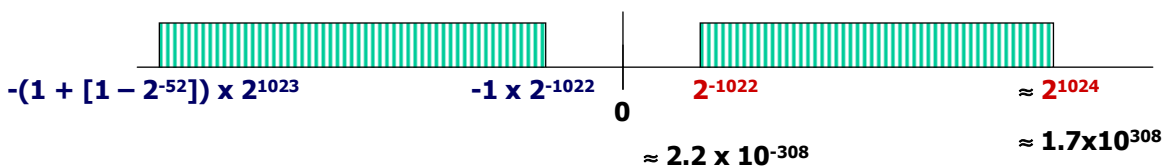
Calcolatori Elettronici, Beraldi, aa 04/05

Range dei valori (normalizzati)

Singola precisione 32 bit: 1 Segno, 8 Esponente, $p=23$ per F



Doppia precisione 64 bit: 1 Segno, 11 Esponente, $p=52$ per F



Caratteristiche

- L'insieme S dei valori rappresentabili ha le seguenti proprietà
 - E' simmetrico rispetto allo 0
 - Esiste un valore minimo, $x_{\min}$ diverso da 0
 - Il tentativo di esprimere un valore $|X| > 0$, ma minore di $x_{\min}$ produce **underflow**
 - Esiste un valore massimo, $x_{\max}$
 - Il tentativo di esprimere un valore $|X| > x_{\max}$ produce **overflow**
- Perdita della proprietà del continuo
 - $\forall x, y \in \mathbb{R} \rightarrow (x+y)/2 \in \mathbb{R}$
 - Ciò non è vero in S (approssimazione)

Calcolatori Elettronici, Beraldi, aa 04/05

Approssimazione

Errore assoluto

- In generale è impossibile rappresentare con esattezza tutti i valori reali disponendo di un numero finito di cifre ed impiegando il sistema posizionale
 - Numeri irrazionali (π)
 - Periodici ($1/3 = 0,33..$)
- Indichiamo con X il valore esatto e con x il valore approssimato
 - $X = 1/3 \rightarrow x = 0,33$
- Si definisce **errore assoluto**, EA, la differenza fra x ed X
- $EA = X - x$ ($|EA| = |X - x|$)
- E' una prima misura della bontà dell'approssimazione

Approssimazione

Errore relativo

- Si definisce **errore relativo** il rapporto
- $ER = (X-x)/x = EA/x$, con $x \neq 0$.
 - Si impiega x al denominatore perché di solito si dispone del valore approssimato
 - $X = x(1+ER)$
- Perché è utile introdurre tale misura?
- Un errore di 1 cm su una misurazione di 10 m è diverso che un errore di 1 cm su una misura di 10000 Km !
- E' legato alla nozione intuitiva di **precisione**

Calcolatori Elettronici, Beraldi, aa 04/05

Errore relativo e precisione

- Calcoliamo l'errore assoluto massimo
 - Consideriamo due valori consecutivi (assumiamo esponente invariato)
 - $x_1 = 1.F \times 2^E$
 - $x_2 = (1.F + 2^{-p}) \times 2^E$
 - Fissato E , l'errore massimo si commette quando X cade a metà fra x_1 ed x_2 .
 - $\max\{EA\} = (x_2 - x_1)/2 = (0,5 \times 2^{-p}) \times 2^E$
- Il valore minimo di x vale
 - $\min\{x\} = 1.0 \times 2^E$
- L'errore relativo vale al più
 - $ER_{\max} = (0,5 \times 2^{-p}) \times 2^E / 2^E = 2^{-1} \times 2^{-p} = 2^{-p-1}$
- L'errore relativo e quindi la precisione della rappresentazione è stabilita dal **numero di cifre della mantissa**

Alcuni parametri

Elemento	Singola Precisione	Doppia Precisione
Numero bit di segno	1	1
Numero bit esponente	8	11
Numero bit della frazione F	23	52
Numero di bit, totale	32	64
Rappr. Esponente	Eccesso 127	Eccesso 1023
Intervallo esponente	-126 ... +127	-1022 ... +1023

Calcolatori Elettronici, Beraldi, aa 04/05

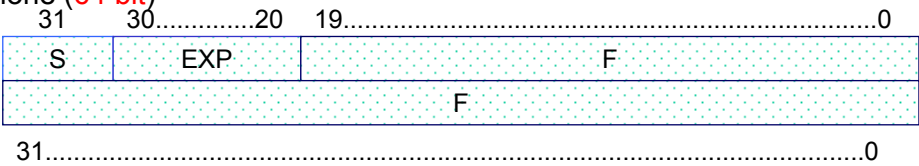
Formati

Singola Precisione (32 bit)



Lunghezza campi: 1,8,23

Doppia Precisione (64 bit)



Lunghezza campi: 1,11,52

Configurazioni

Normalizzato	$\pm$	$0 < \text{Exp} < \text{Max}$	Qualunque sequenza di bit	$(1.F) \times 2^{\text{EXP}-\text{Bias}}$
Denormalizzato	$\pm$	0	Qualunque sequenza di bit non zero	$F \times 2^{\text{Emin}}$
Zero	$\pm$	0	0	± 0
Infinito	$\pm$	1 1 1...1	0	$\pm \infty$
Non un numero	$\pm$	1 1 1...1	Qualunque sequenza di bit non zero	NaN

↖ Bit di segno

Calcolatori Elettronici, Beraldi, aa 04/05

Esempio

- Rappresentare in singola precisione il valore $X=28,125$
- Parte Intera
 - $28 = 16+8+4 \rightarrow 11100$
- Parte Frazionaria
 - $0,125 = 1/8 \rightarrow 0,001$
- $X=11100,001$
- Trasformiamolo in formato IEEE 754 SP
 - Segno = 0
 - $X = 1,1100001 \times 2^4$
 - $F = 1100001$
 - $\text{EXP}=127+4 = 131 = 128+2+1 \rightarrow 10000011$
 - $X = 0 \ 10000011 \ 1100001 \ 0000000000000000 = 0x41E10000$

Esempio

- Calcolare il valore massimo, $X_{\max}$, esprimibile in SP
 - EXP=254
 - il valore 255=(11111111)₂ è riservato
 - F=(111...1)₂, (23 bit)
 - $X_{\max} = (1.11...1)_2 \times 2^{254-127} = (2-2^{-23}) \times 2^{127} \approx 2^{128} \approx 3.4 \times 10^{38}$
- Calcolare il valore minimo $X_{\min}$ che può essere espresso in SP
 - EXP=1 (il valore 0 è riservato)
 - F=0...0
 - $X_{\min} = (1.00...00)_2 \times 2^{1-127} = 2^{-126} \approx 1.2 \times 10^{-38}$

Calcolatori Elettronici, Beraldi, aa 04/05

Esempio

La stringa esadecimale 0x56700000 e' un numero in formato IEEE.754.
Calcolarne il valore.

5 → 0101

6 → 0110

7 → 0111

0 101 0110 0 111 0000 0000 0000 0000 000

Bit segno = 0 → numero positivo

Esponente = bit (30:23)-127=(10101100)₂ - 127 = 172 - 127 = 45

Mantissa = 1.bit(22:0) = (1.111)₂ = 1.875

Risultato = + 1.875 x 2⁴⁵

Esempio

La stringa esadecimale 0x80880000 e' un numero in formato IEEE.754.
Calcolarne il valore.

Segno = bit (31) = 1 = -

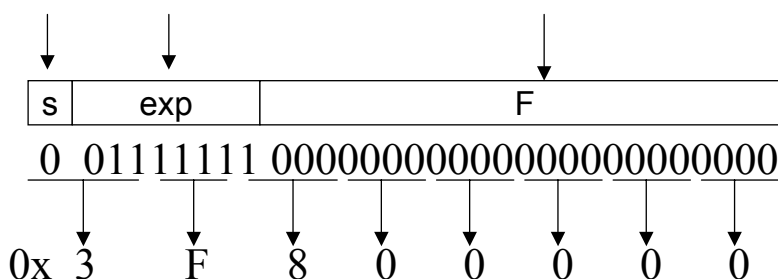
Esponente= bit(30:23) -127 = 00000001₂ - 127 = 1 - 127 = -126

Mantissa = 1. bit(22:0) = 1.0001₂

Risultato = -1.0625×2^{-126} (= $-1.24896..... \times 10^{-38}$)

Esempio

- Rappresentare il valore 1 in singola precisione IEEE 745
 - 23 bit mantissa (parte frazionaria F),
 - 1 segno (S)
 - 8 esponente (EXP)
 - Forma normalizzata: $1.0 \cdot 2^0 \Rightarrow s=1, F=0, EXP-127=0$ (EXP=127)
 - $s=0, EXP=127=(0111111)_2, F=0 = (000..0)_2,$



Vedere la rappresentazione in C

```
#include <stdio.h>
main (){

    float f = 1;
    unsigned char* cp = (unsigned char*) &f;
    for (int i=sizeof(f)-1;i>=0;i--)
        printf("%X ",cp[i]); /*stampa: 3F 80 0 0*/

}
```