

Lezione 2: Aritmetica in virgola mobile

Elementi di Architettura e Sistemi Operativi
Docente: Tiziano Villa

Corso di Laurea in Bioinformatica

Gennaio 2021

- Aritmetica in virgola mobile.

Fonte:

Patt & Patel: “*Introduction to Computing Systems: From Bits and Gates to C and Beyond*”. Ch. 2 section 2.7.2.

Limitazioni delle rappresentazioni intere

- La maggior parte dei numeri non sono interi !
- Ma anche con gl'interi ci sono due altre considerazioni:
 - ▶ dominio o intervallo di rappresentazione: la grandezza dei numeri che possiamo rappresentare e' determinata da quante cifre binarie si possono usare; es., il numero piu' grande che possiamo rappresentare con 32 cifre binarie e' di circa +/- 2 miliardi, troppo piccolo per molte applicazioni
 - ▶ Precisione: l'esattezza con cui si puo rappresentare un numero; es. un numero con 32 cifre binarie ci offre una precisione di 31 cifre binarie (cioe' 9 o 10 cifre decimali)
- Serve un altro tipo di rappresentazione

Rappresentazione dei numeri non-interi

- Nel sistema decimale si rappresentano i numeri non-interi aggiungendo un altro simbolo (la virgola decimale) per ottenere una notazione a virgola fissa: es.

$$3.456,78 = 3 \times 10^3 + 4 \times 10^2 + 5 \times 10^1 + 6 \times 10^0 + 7 \times 10^{-1} + 8 \times 10^{-2}$$

- Similmente nel sistema binario si può introdurre una “virgola binaria” per separare le potenze del 2 positive da quelle negative. Se si allineano le virgole binarie, l’addizione e sottrazione in complemento a due funzionano ancora; es.

$$00101000,101 + 40,625$$

$$11111110,110 - 1,25$$

$$00100111,011 \quad 39,375$$

- C’è un problema con i numeri molto piccoli o molto grandi. Es. $6,023 \times 10^{23}$ richiede 79 cifre binarie; $6,626 \times 10^{-34}$ richiede > 110 cifre binarie

Rappresentazione dei numeri non-interi

- Nella pratica scientifica si usa la virgola mobile (“floating point” in inglese) che permette di rappresentare numeri molto grandi o molto piccoli, con poca o tanta precisione come necessario. Es., unita' di carica elettrica: $e = 1,602\,176\,462 \times 10^{-19}$ Coulomb, volume dell'universo $= 1 \times 10^{85} \text{ cm}^3$
- Si riproduce la notazione decimale in virgola mobile per creare una notazione binaria “ibrida” in virgola mobile:
 - ▶ Si usa una “virgola binaria” per separare le parti intere da quelle frazionarie: es.,
 $00011001,110 = 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} = 25,75$
 - ▶ poi si sposta la virgola binaria per ottenere una rappresentazione *normalizzata* per assicurare che un numero non-nullo abbia una sola rappresentazione: es., 00011001,110 diventa $1,1001110 \times 2^4$ con esponente = 4 e parte significativa = 1,1001110
- Si codifica la rappresentazione precedente (vedi IEEE 754)

Rappresentazione in virgola mobile IEEE 754

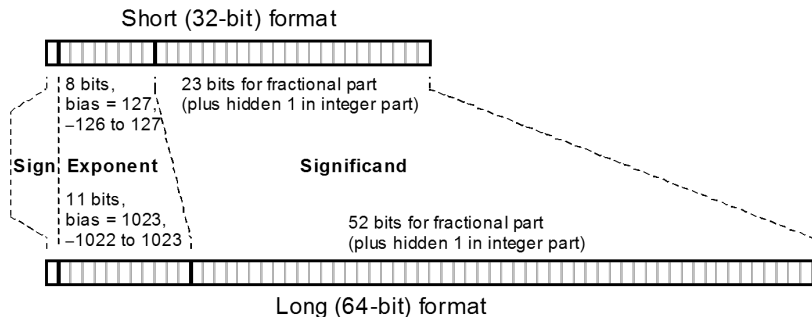
- Sviluppata da IEEE nel 1985 per proporre una rappresentazione comune e risolvere i problemi di portabilità del codice scientifico, e oggi adottata universalmente
- Nota come Standard IEEE 754, cioè ufficialmente come IEEE Standard for Binary Floating-Point Arithmetic (ANSI/IEEE Std 754-1985); o anche come IEC 60559 (IEC 60559:1989, Binary floating-point arithmetic for microprocessor systems)
- Ci sono due rappresentazioni
 - ▶ Precisione singola (32 cifre binarie)
 - ▶ Precisione doppia (64 cifre binarie)
- Rappresentazione con modulo e segno

Formato in virgola mobile IEEE 754

- Una rappresentazione in virgola mobile secondo il formato IEEE consiste di
 - ▶ Una cifra binaria di segno
 - ▶ Un esponente (" $\times 2$ alla che cosa ?")
 - ▶ Frazione (anche significando o parte significativa - in inglese "Significand", a volte impropriamente detto Mantissa)
- Si suppone che la frazione sia nella forma 1.xxxxx chiamata espressione normalizzata (cioè, è implicito che la prima cifra binaria a sinistra della virgola sia 1). Ad es., una frazione = 0 s'interpreta come 1,0, e una frazione = 1111 s'interpreta come 1,1111
- Per gestire sia esponenti positivi che negativi, invece di usare il complemento a due e allo scopo di semplificare i circuiti aritmetici si rappresenta l'esponente traslato ("biased exponent", in italiano detto anche esponente polarizzato), cioè l'esponente effettivo incrementato dello spiazzamento ("bias")

Formato in virgola mobile IEEE 754

- *Precisione singola*: 8 cifre binarie per l'esponente, 23 cifre binarie per la frazione
- *Precisione doppia*: 11 cifre binarie per l'esponente, 52 cifre binarie per la frazione



Virgola mobile IEEE 754 in precisione singola

- IEEE 754 single precision



Sign Biased exponent

Normalized Mantissa (implicit 24th bit = 1)

$$(-1)^s \times F \times 2^{E-127}$$

Exponent	Mantissa	Object Represented
0	0	0
0	non-zero	denormalized
1-254	anything	FP number
255	0	pm infinity
255	non-zero	NaN

Numeri rappresentabili in precisione singola

- Gli esponenti 00000000 e 11111111 sono riservati
- Numero minimo
 - ▶ Esponente: 00000001 \implies esponente effettivo = $1 - 127 = -126$
 - ▶ Frazione: 000...00 \implies frazione effettiva = 1,0
 - ▶ $\pm 1,0 \times 2^{-126} \approx \pm 1,2 \times 10^{-38}$
- Numero massimo
 - ▶ Esponente: 11111110 \implies esponente effettivo = $254 - 127 = +127$
 - ▶ Frazione: 111...11 \implies frazione effettiva $\approx 2,0$
 - ▶ $\pm 2,0 \times 2^{+127} \approx \pm 3,4 \times 10^{+38}$

Numeri rappresentabili in precisione doppia

- Gli esponenti 0000...00 e 1111...11 sono riservati
- Numero minimo
 - ▶ Esponente: 00000000001 \Rightarrow esponente effettivo = $1 - 1023 = -1022$
 - ▶ Frazione: 000...00 \Rightarrow frazione effettiva = 1,0
 - ▶ $\pm 1,0 \times 2^{-1022} \approx \pm 2,2 \times 10^{-308}$
- Numero massimo
 - ▶ Esponente: 11111111110 \Rightarrow esponente effettivo = $2046 - 1023 = +1023$
 - ▶ Frazione: 111...11 \Rightarrow frazione effettiva $\approx 2,0$
 - ▶ $\pm 2,0 \times 2^{+1023} \approx \pm 1,8 \times 10^{+308}$

Come rappresentare un dato numero in virgola mobile

- Si esprime il numero in formato binario
- Si normalizza il numero
- Si somma lo spiazzamento 127 all'esponente effettivo
- La frazione sono le 23 cifre binarie della precisione dopo aver rimosso l'1 a sinistra della virgola
- Dal segno del numero si determina l'esponente S di $(-)^S$
- Si consideri il numero $-(6 + 5/8)$
 - ▶ Numero in formato binario: $-(6 + 5/8)_{10} = -110,101_2$
 - ▶ Numero normalizzato: $-(6 + 5/8)_{10} = -1,10101 \times 2^2$
 - ▶ Esponente: $2 + 127 = 129_{10} = 10000001_2$
 - ▶ Frazione: 10101000000000000000000
 - ▶ Rappresentazione in virgola mobile (segno esponente frazione):
1 10000001 10101000000000000000000

Esempi di conversione in virgola mobile

- $0,0_{10}$: 0 00000000 000000000000000000000000
- $1,0_{10}$ ($1,0_2 = 1,0 \times 2^0$): 0 01111111 000000000000000000000000
- $0,5_{10}$ ($0,1_2 = 1,0 \times 2^{-1}$):
0 01111110 000000000000000000000000
- $0,75_{10}$ ($0,11_2 = 1,1 \times 2^{-1}$):
0 01111110 100000000000000000000000
- $3,0_{10}$ ($11_2 = 1,1 \times 2^1$): 0 10000000 100000000000000000000000
- $-0,375_{10}$ ($-0,011_2 = -1,1 \times 2^{-2}$):
1 01111101 100000000000000000000000
- $-20,0_{10}$ ($-10100_2 = -1,01 \times 2^4$):
1 10000011 010000000000000000000000

Altri esempi di conversione in virgola mobile

- $-0,75_{10}$ in precisione singola ($-0,11_2 = -1,1 \times 2^{-1}$):
1 01111110 100000000000000000000000
- $-0,75_{10}$ in precisione doppia ($-0,11_2 = -1,1 \times 2^{-1}$):
1 01111111110 100000000000000000000000...0 (11 cifre binarie per l'esponente e 52 cifre binarie per la frazione)

Esempi di conversione da virgola mobile

- Dato il numero in precisione singola

1 10000001 010000000000000000000000

il numero decimale corrispondente e':

$$(-1)^1 \times 1,01_2 \times 2^{129-127} = -1 \times 1,25_{10} \times 2^2 = -5$$

- Dato il numero in precisione singola

0 01111011 000000000000000000000000

il numero decimale corrispondente e':

$$(-1)^0 \times 1_2 \times 2^{123-127} = 1 \times 1_{10} \times 2^{-4} = 2^{-4} = 1/16$$

- Dato il numero in precisione singola

0 10000011 001010000000000000000000

il numero decimale corrispondente e':

$$(-1)^0 \times 1,00101_2 \times 2^{131-127} = 1 \times 1,00101_2 \times 2^4 = 18,5$$

Altri esempi di conversione da virgola mobile

- Dato il numero in precisione singola

1 10000010 001010000000000000000000

il numero decimale corrispondente e':

$$\begin{aligned} (-1)^1 \times 1,00101_2 \times 2^{130-127} &= -1 \times 1,00101_2 \times 2^3 = \\ (-1) \times 1001,01_2 &= -9,25 \end{aligned}$$

- Dato il numero in precisione singola

0 11111110 111111111111111111111111

il numero decimale corrispondente e':

$$\begin{aligned} (-1)^0 \times 1,111111111111111111111111_2 \times 2^{254-127} &\approx \\ 1 \times 2_{10} \times 2^{127} &= 2^{128} \text{ (si noti che la parte frazionaria e' } \approx 2) \end{aligned}$$

Numeri speciali

Single precision		Double precision		Object represented
exponent	significand	exponent	significand	
0	0	0	0	0
0	Nonzero	0	Nonzero	\pm denormalized number
1-254	Anything	1-2046	anything	\pm floating-point number
255	0	2047	0	$\pm \infty$
255	nonzero	2047	nonzero	NaN

Numeri speciali

- Se l'esponente rappresentato e' nullo, l'esponente effettivo e' -126 e la parte decimale s'interpreta come $0, \dots$ seguito dalle 23 cifre della frazione (invece che $1, \dots$ seguito dalle 23 cifre), permettendo di rappresentare numeri molto piccoli (*denormalizzati*)

- Dato il numero in precisione singola
0 00000000 000010000000000000000000
il numero decimale corrispondente *denormalizzato* (non c'e' l'1 nascosto prima della virgola) e':
$$(-1)^0 \times 0,00001_2 \times 2^{-126} = 1 \times 2^{-5} \times 2^{-126} = 2^{-131}$$

- Dato il numero in precisione singola
1 00000000 000000000000000000000001
il numero decimale corrispondente *denormalizzato* e':
$$(-1)^1 \times 0,0000000000000000000000001_2 \times 2^{-126} = -1 \times 2^{-23} \times 2^{-126} = -2^{-149}$$

- $\pm\infty$: esponente = $255_{10} = 11111111_2$ e frazione = 0
- NaN (“Not a Number”): esponente = $255_{10} = 11111111_2$ e frazione $\neq 0$; indica un valore indefinito che puo’ risultare da operazioni non ben definite come la radice quadrata di un negativo e la divisione per zero (un’operazione in cui un operando e’ NaN produce come risultato NaN)

Numeri denormalizzati

- Il valore più piccolo normalizzato rappresentabile è $\pm 2^{1-127} \times 1,00...00 = 2^{-126}$ che lascia scoperto il seguente intervallo di numeri molto piccoli non rappresentabili
 $-2^{-126} < ...?... < 0 < ...?... < +2^{-126}$
- I numeri denormalizzati aumentano l'intervallo di rappresentabilità dei numeri molto piccoli a
 $\pm 2^{-126} \times 0,00...01 = 2^{-126} \times 2^{-23} = \pm 2^{-149}$
che lascia scoperto un intervallo minore $-2^{-126} < ... < -2^{-149} < ...?... < 0 < ...?... < +2^{-149} < ... < +2^{-126}$

Confronto tra virgola fissa e virgola mobile

- A parità del numero di cifre disponibili
 - ▶ con la rappresentazione intera o in virgola fissa, i numeri rappresentati sono distribuiti uniformemente nel campo di rappresentabilità
 - ▶ con la rappresentazione in virgola mobile, i numeri rappresentati sono distribuiti non uniformemente nel campo di rappresentabilità: sono più fitti vicino allo 0 e più radi per valori assoluti grandi, cioè la precisione è concentrata dove serve di più
 - ★ La separazione tra $1,0 \dots 00 \times 2^{99}$ e $1,0 \dots 01 \times 2^{99}$ è di $2^{-23+99} = 2^{76}$
 - ★ La separazione tra $1,0 \dots 00 \times 2^0$ e $1,0 \dots 01 \times 2^0$ è di $2^{-23+0} = 2^{-23}$, dove $2^{76} \gg \gg 2^{-23}$

Confronto tra virgola fissa e virgola mobile

● Virgola fissa

- ▶ quanto più piccolo è il numero da rappresentare tanto maggiore è l'errore relativo che si commette nella rappresentazione;
es., dato 0,001 e il successivo 0,002, l'errore percentuale è
$$((0,002 - 0,001)/0,001) \times 100\% = 100\%$$
- ▶ quanto più grande è il numero da rappresentare tanto minore è l'errore relativo che si commette nella rappresentazione;
es., dato 100,001 e il successivo 100,002, l'errore percentuale è
$$((100,002 - 100,001)/100,001) \times 100\% \approx 0,001\%$$

● Virgola mobile

- ▶ l'errore relativo è costante = $2^{-\#cifre_frazione}$ e l'errore assoluto aumenta all'aumentare del numero da rappresentare; es.,
dato $0,128e^{-100}$ e il successivo $0,129e^{-100}$, l'errore percentuale è
$$((0,129e^{-100} - 0,128e^{-100})/0,128e^{-100}) \times 100\% = 0,78125\%;$$

dato $0,128e^{+100}$ e il successivo $0,129e^{+100}$, l'errore percentuale è
$$((0,129e^{+100} - 0,128e^{+100})/0,128e^{+100}) \times 100\% = 0,78125\%$$

[nota: $1/128 = 0,0078125 = 0,78125\%$]

Conclusioni

- L'insieme dei numeri in virgola mobile non coincide con quello dei reali, perché non è denso, è limitato tra un numero reale massimo ed uno minimo rappresentabili, e infine i numeri non sono distribuiti uniformemente
- L'aritmetica dei numeri in virgola mobile è diversa da quella dei numeri reali e l'ordine di esecuzione delle operazioni può variarne il risultato
 - ▶ L'aritmetica in virgola mobile non è associativa:
 $(x + y) + z \neq x + (y + z)$ e $(x \cdot y) \cdot z \neq x \cdot (y \cdot z)$
 - ▶ L'aritmetica in virgola mobile non è distributiva:
 $x \cdot (y + z) \neq (x \cdot y) + (x \cdot z)$
 - ▶ Esistono l'elemento neutro della moltiplicazione, l'elemento neutro dell'addizione e l'opposto, ma non sono unici
- Cause di errori: arrotondamento (es. numeri non rappresentabili come 0 , 1_{10} e $2/3_{10}$), trabocco, sottocapacità, operazioni impossibili, etc.