Rappresentazione di valori numerici

La rappresentazione dei valori numerici è importante per stabilire la procedura necessaria a eseguire dei calcoli. In questo capitolo si riepilogano alcuni concetti che possono essere utili per comprendere il significato di alcune forme di rappresentazione dei valori numerici nei linguaggi di programmazione.

Codifica delle singole cifre

Un valore numerico potrebbe essere espresso come una stringa di caratteri, corrispondenti alle cifre numeriche che lo rappresentano secondo la notazione in base dieci. Naturalmente, una rappresentazione del genere implica uno spreco di spazio nel sistema di memorizzazione e richiede una trasformazione prima di poter procedere all'esecuzione di calcoli numerici.

Esistono diverse forme di rappresentazioni numeriche, intese come sequenze di cifre in base dieci, che utilizzano quattro bit per ogni cifra. Il sistema più comune è noto con il nome **BCD: *Binary coded decimal***.

Alcune codifiche per la rappresentazione di cifre numeriche (in base dieci) a gruppi di quattro bit:

Cifra decimale	Codice BCD (8421)	Codice «eccesso 3»	Codice 2421	Codice 5211	Codice 631-1	Codice 732-1
0	0000_2	0011_2	0000_2	0000_2	0000_2 0011_2	0000_2
1	0001_2	0100_2	0001_2	0001_2 0010_2	0010_2	0011_2
2	0010_2	0101_2	0010_2 1000_2	0100_2 0011_2	0101_2	0010_2
3	0011_2	0110_2	0011_2 1001_2	0101_2 0110_2	0100_2	0100_2
4	0100_2	0111_2	0100_2 1010_2	0111_2	0110_2	0111_2
5	0101_2	1000_2	0101_2 1011_2	1000_2	1001_2	0110_2
6	0110_2	1001_2	0110_2 1100_2	1010_2 1001_2	1000_2	1001_2
7	0111_2	1010_2	0111_2 1101_2	1011_2 1100_2	1010_2	1000_2
8	1000_2	1011_2	1110_2	1110_2 1101_2	1101_2	1011_2
9	1001_2	1100_2	1111_2	1111_2	1100_2 1111_2	1010_2

La codifica BCD e altre sono ***codici pesati***, in quanto a ogni bit viene attribuito un peso, da sommare per determinare il valore. I pesi per la codifica BCD sono 8, 4, 2 e 1; pertanto, il codice BCD 1001_2 corrisponde a 1*8+0*4+0*2+1*1 = 9. Nella tabella riepilogativa, i codici pesati sono: BCD, 2421, 5211, 631-1 e 732-1. I nomi usati per questi codici sono costituiti dalla sequenza dei pesi stessi.

Alcuni codici pesati prevedono la rappresentazione di alcune cifre in più di un modo alternativo. Per esempio, nel codice 2421, il numero due si può ottenere sia come 1000_2, sia come 0010_2.

I codici pesati come BCD (ovvero 8421), 2421 e 5211, prevedono pesi positivi; i codici come 631-1 e 732-1, prevedono anche pesi negativi. Per esempio, con il codice 732-1, si ottiene il valore uno con il codice 0011_2, perché il secondo bit (a destra) vale come il numero due, mentre il primo bit (a destra) sottrae una unità.

Dei codici che appaiono nella tabella, il codice a eccesso tre, non è un codice pesato, in quanto corrisponde al codice BCD, a cui si aggiunge il valore tre.

È necessario sottolineare che il codice BCD, a seconda del contesto, può essere riferito anche a un codice a otto bit, dove i primi quattro, più significativi, sono posti a zero.

Esempio:

La sigla BCD è l'acronimo di Binary Code Decimal, e funziona come il sistema di numerazione binario, con la differenza che ogni cifra è rappresentata con quattro bit, ma possono essere rappresentate le cifre da zero a nove.
Rispetto al sistema binario il BCD ha quindi una minore capacità di rappresentazione (ridondanza).
Con quattro cifre si rappresentano:in binario da 0 a 2^4-1=15
in BCD da 0 a 9.
Per convertire un numero decimale in BCD è sufficiente sostituire ad ogni cifra il corrispondente numero binario a quattro bit.
Esempio:
$(1972)_{10}=(0001|1001|0111|0010)_{BCD}$
1 | 9 | 7 | 2
N.B.: quando si rappresenta il numero si omettono le barre divisorie.

Codice Gray

Si tratta di un codice contraddistinto dal fatto che, ogni numero differisce dal suo successivo e dal suo precedente per un unico bit. Esso **si ottiene aggiungendo alla singola cifra (espressa in Binario Puro) la sua successiva partendo da destra verso sinistra**, senza considerare il riporto (codice di tipo riflesso). Il codice Gray non è un codice pesato, quindi le cifre binarie non assumono valore differente a seconda della posizione occupata.

Tabella di conversione

Numero Decimale	Numero Binario Puro	Numero in Gray
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

Es. 0010 (binario puro)= 0011 (codice Gray) (Lettura dall'ultima cifra alla prima)

0+1=1
1+0=1
0+0=0
0+0=0

Questo codice non è un codice BCD, infatti, non codifica le singole cifre decimali ma è invece un codice completo in quanto esprime direttamente tutti i numeri.

Rappresentazione binaria di numeri interi senza segno

Quando si rappresentano dei valori numerici in forma binaria, senza passare per una conversione di ogni singola cifre decimale, si usa tutta la sequenza di bit per il valore. La rappresentazione di un valore intero senza segno coincide normalmente con il valore binario contenuto nella variabile. Pertanto, una variabile della dimensione di 8 bit, può rappresentare valori da zero a 2^8-1:

00000000_2 (0_{10})
00000001_2 (1_{10})
00000010_2 (2_{10})
...
11111110_2 (254_{10})
11111111_2 (255_{10})

Rappresentazione binaria di numeri interi con segno

Per rappresentare valori interi con segno (positivo o negativo), si deve riservare un bit per tale informazione. Generalmente si usa il bit più significativo, considerando lo zero come l'indice di un valore positivo e uno come indice di un valore negativo, in quanto complementato.

Complemento a due (cambio di segno)

Facilissimo: Inversione bit + 1

19 -> -19

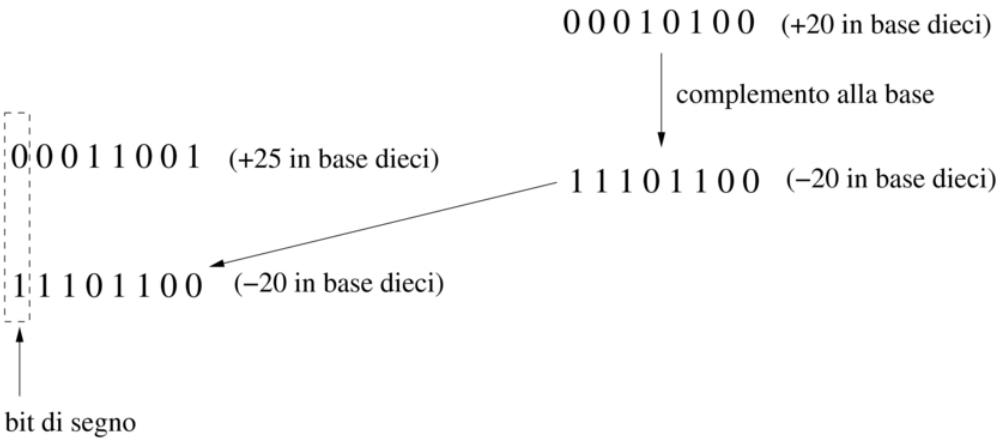
19 in complemento a 2	010011
Inverti i bit	101100
Somma 1	101101

-23 -> 23

-23 in complemento a 2	101001
Inverti i bit	010110
Somma 1	010111

Per comprendere questa cosa è necessario fare degli esempi. Viene proposta la rappresentazione di valori all'interno di variabili a soli 8 bit complessivi, dove il bit più significativo serve a rappresentare il segno. Si supponga di volere rappresentare i valori +25 e -20: il primo valore corrisponde a 00011001_2, mentre il secondo corrisponde a -00010100_2, che però va convertito nel suo complemento alla base (complemento a due), ovvero 11101100_2.

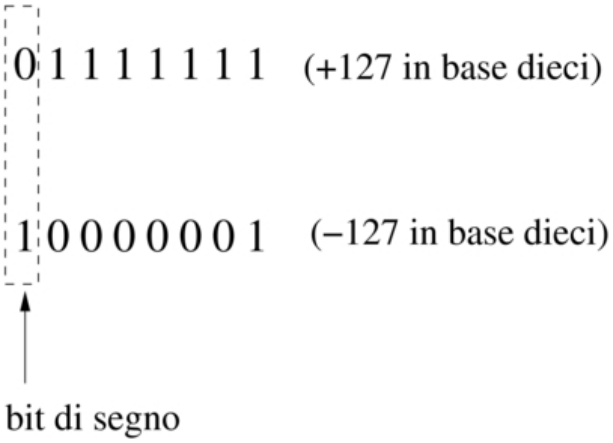
Confronto tra due valori interi con segno.



In pratica, il bit usato per indicare il segno, manifesta il fatto che un numero sia rappresentato in modo normale (positivo), oppure complementato (negativo). In questo modo, le sottrazioni si ottengono attraverso le somme.

Disponendo di una variabile per rappresentare valori interi con segno, considerato che il bit più significativo serve a rappresentare il segno stesso, si dispone di un bit in meno per indicare il valore. Pertanto, se si dispone di *n* bit, si possono rappresentare valori fino a *n*-1 bit, ovvero valori fino a $2^{(n-1)}$-1. Per i numeri negativi, il calcolo è lo stesso, anche se si considera che si fa riferimento a valori complementati: si può rappresentare fino a -($2^{(n-1)}$-1).

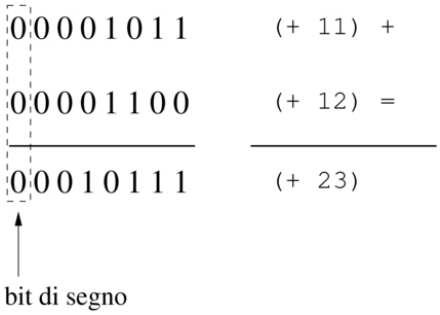
Valori massimi rappresentabili con soli otto bit; in questo caso, per il valore negativo, si evita di usare 10000000_2, dato che il valore positivo corrispondente non sarebbe rappresentabile, in quanto il complemento corrisponde sempre a 10000000_2.



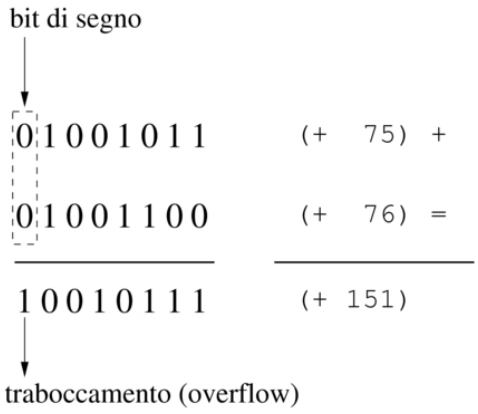
Sommatorie con i valori interi con segno

Quando si sommano dei valori interi con segno, si possono presentare circostanze diverse per il modo di considerare il riporto che si ottiene. Vengono proposti alcuni esempi che servono a dimostrare le varie situazioni, dove si deve ricordare che i valori negativi sono rappresentati come complemento alla base del valore assoluto corrispondente.

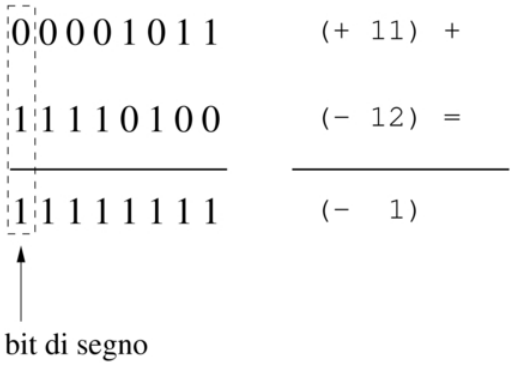
Somma di due valori positivi che non genera un riporto.



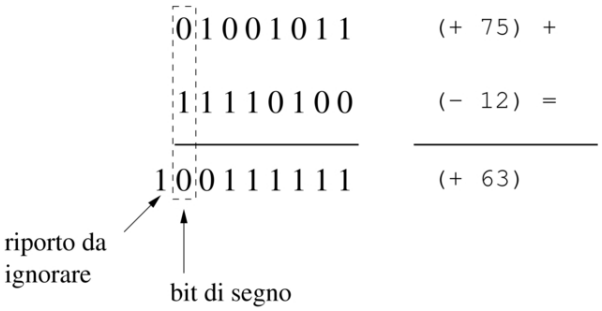
Somma di due valori positivi che genera un riporto, che però non può essere gestito: il risultato apparentemente negativo indica la presenza di un traboccamento.



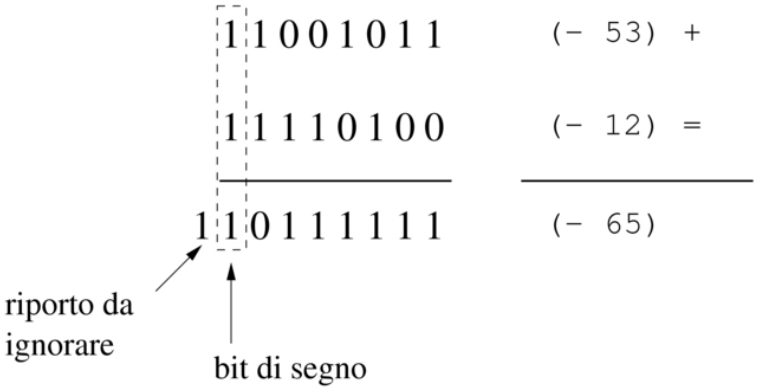
Somma di un valore positivo e di un valore negativo senza riporti.



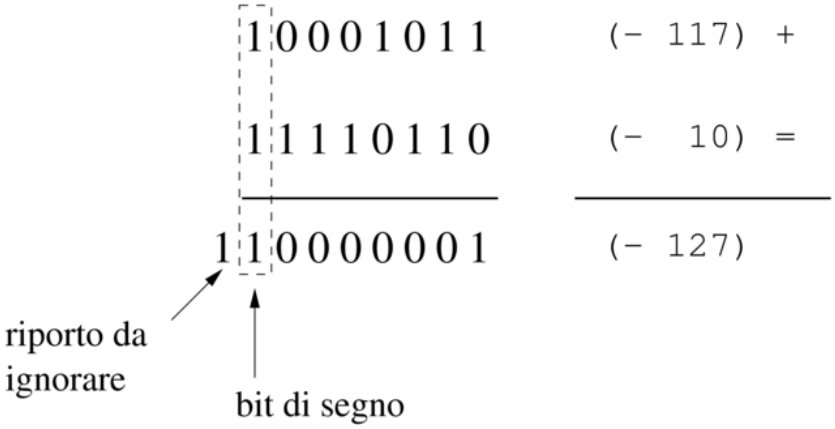
Somma di un valore positivo e di un valore negativo che produce un riporto da ignorare.



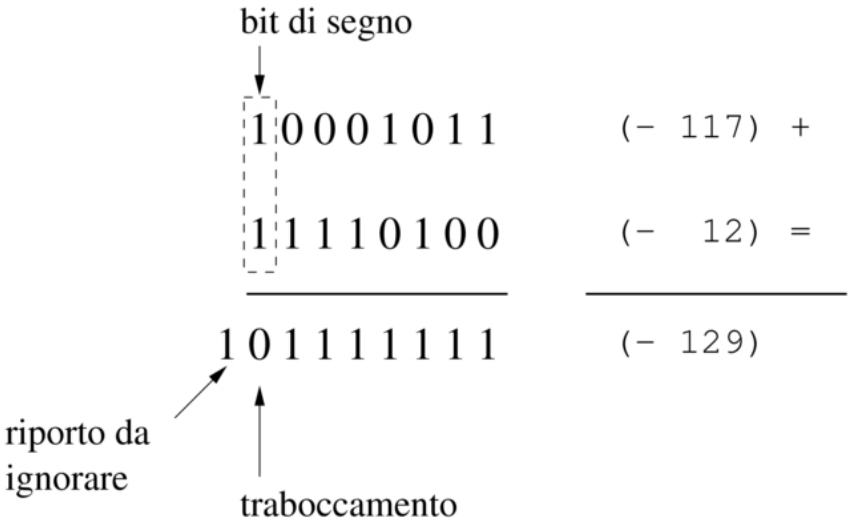
Somma di due valori negativi che produce un segno coerente e un riporto da ignorare.



Somma di due valori negativi che genera il valore massimo contenibile nella variabile (-127).



Somma di due valori negativi che genera un traboccamento, evidenziato da un risultato con un segno incoerente.



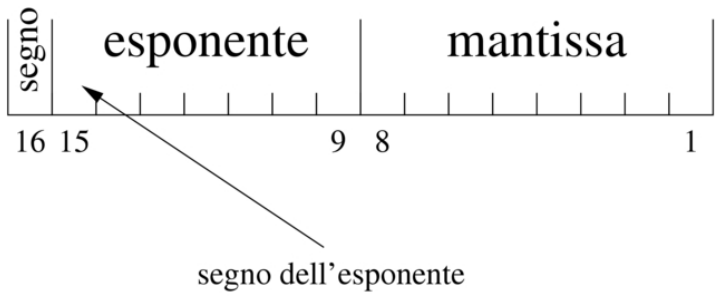
Rappresentazione binaria di numeri in virgola mobile

Una forma diffusa per rappresentare dei valori molto grandi, consiste nell'indicare un numero con dei decimali moltiplicato per un valore costante elevato a un esponente intero. Per esempio, per rappresentare il numero 123 000 000 si potrebbe scrivere $123 \cdot 10^6$, oppure anche $0{,}123 \cdot 10^9$. Lo stesso ragionamento vale anche per valori molti piccoli; per esempio 0,000 000 123 che si potrebbe esprimere come $0{,}123 \cdot 10^{-6}$.

Per usare una notazione uniforme, si può convenire di indicare il numero che appare prima della moltiplicazione per la costante elevata a una certa potenza come un valore che più si avvicina all'unità, essendo minore o al massimo uguale a uno. Pertanto, per gli esempi già mostrati, si tratterebbe sempre di $0{,}123 \cdot 10^n$.

Per rappresentare valori a ***virgola mobile*** in modo binario, si usa un sistema simile, dove i bit a disposizione della variabile vengono suddivisi in tre parti: segno, esponente (di una base prestabilita) e mantissa, come nell'esempio che appare nella figura successiva.

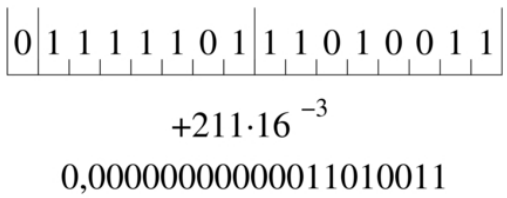
Ipotesi di una variabile a 16 bit per rappresentare dei numeri a virgola mobile.


_segno="segno" esponente="esponente" mantissa="mantissa" 16="16" 15="15" 9="9" 8="8" 1="1" segno_dell_esponente="segno dell'esponente" />

Nella figura si ipotizza la gestione di una variabile a 16 bit per la rappresentazione di valori a virgola mobile. Come si vede dallo schema, **il bit più significativo della variabile viene utilizzato per rappresentare il segno del numero**; **i sette bit successivi si usano per indicare l'esponente** e gli **otto bit finali per la mantissa**, ovvero il valore da moltiplicare per una certa costante elevata all'esponente. Come si intende intuitivamente, il primo bit dell'esponente rappresenta il segno di questo.

Quello che manca da decidere è come deve essere interpretato il numero della mantissa e qual è il valore della costante da elevare all'esponente indicato. Sempre a titolo di esempio, si conviene che il valore indicato nella mantissa esprima precisamente «0,*mantissa*» e che la costante da elevare all'esponente indicato sia 16 (ovvero 2^4), che si traduce in pratica nello spostamento della virgola di quattro cifre binarie alla volta.

Esempio di rappresentazione del numero 0,051 513 671 875 ($211 \cdot 16^{-3}$), secondo le convenzioni stabilite. Si osservi che il valore dell'esponente è negativo ed è così rappresentato come complemento alla base (due) del valore assoluto relativo.



$$+211 \cdot 16^{-3}$$

$$0{,}0000000000011010011$$