Architettura degli Elaboratori - Rappresentazione dell'informazione

Andrea Malvezzi

04 Ottobre, 2024

Contents

1 Codifiche e sistemi di numerazione

I calcolatori elaborano tipi diversi di informazioni, ma alla base di ognuna di queste categorie ci sono i bit della memoria, ovvero dei valori binari.
Per convertire i dati dalla loro ”forma originale” a un qualcosa di comprensibile dal calcolatore (ovvero sequenze di zero e di uno), occorre una codifica.

1.1 Sistemi di numerazione posizionali

Una codifica ha una **base** b, a cui corrisponde un quantitativo di caratteri con cui codificare i valori da convertire in un certo formato.
Ad esempio, la codifica binaria ha base 2, in quanto utilizza solamente lo 0 e l'1.
Altre codifiche importanti sono quella ottale (base 8) ed esadecimale (base 16, dal 9 in poi si usano le lettere in ordine alfabetico).

1.2 Leggere un numero in una certa base

Avendo un numero codificato in base b, allora tale numero corrisponderà a:

$$\sum_{i=0}^{k} d_i \cdot b^i \tag{1}$$

1.2.1 Esempio di diversi tipi di codifica

Prendiamo come esempio il numero 2001 in base decimale (quella con cui siamo abituati a contare nella matematica ”standard”).
Ora convertiamolo in binario, ottale e esadecimale:

$$2001_{10} = 11111010001_2$$

In quanto il primo uno corrisponde a $1 * 2^{10}$, il secondo a $1 * 2^9$, e così via, fino all'ultimo che corrisponde a $1 * 2^0$, quindi a 1.

$$2001_{10} = 3721_8$$

$$2001_{10} = 7D1_{16}$$

Tutte queste 3 codifiche sono **equivalenti**, in quanto rappresentano tutte lo stesso numero.

1.3 Metodi alternativi per effettuare conversioni

Per convertire un numero in binario da decimale, basta dividerlo per 2 e segnare i resti:

$$2999 : 2 = 1499, \text{ con resto pari a 1.}$$
$$1499 : 2 = 749, \text{ con resto pari a 1.}$$
$$\text{etc} \ldots$$
$$5 : 2 = 2, \text{ con resto pari a 1}$$
$$2 : 2 = 1, \text{ con resto pari a 0}$$

Mentre per fare la conversione inversa, si può optare per un metodo più alternativo, ovvero quello delle **moltiplicazioni successive**:

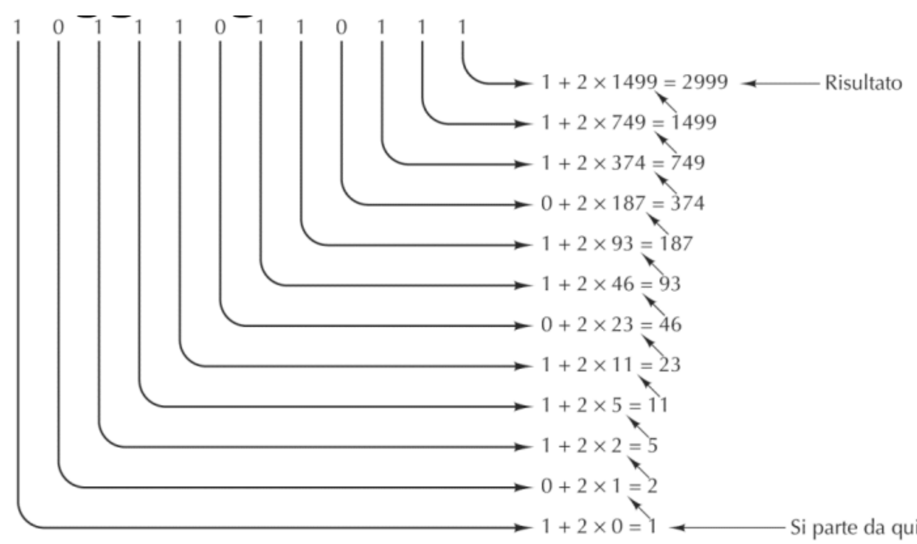

Figure 1: Partendo da sx e con un moltiplicatore pari a 0, per ogni cifra si somma la cifra in questione al doppio del moltiplicatore, per poi assegnare al moltiplicatore valore pari a quello appena ottenuto.

2 Numeri binari negativi

Per codificare i numeri binari negativi si possono usare diverse tecniche, tra cui:

2.1 Modulo e segno

Si usa il bit più significativo (a sx) come segno, dove 0 indica il "+" e l'1 indica il "-".

2.1.1 Esempio

Usando 8 bit:

- $6_{10} = \colorbox{yellow}{0}\,0000110_2$, dove il bit evidenziato indica il segno "+";
- $-6_{10} = \colorbox{yellow}{1}\,0000110_2$, dove il bit evidenziato indica il segno "-".

2.2 Complemento a 1

Qualora il numero da rappresentare fosse negativo, si complementa il modulo della codifica binaria di tale numero.

2.2.1 Esempio

- $6_{10} = \colorbox{yellow}{0}\,0000110_2$;
- $-6_{10} = \colorbox{yellow}{1}\,1111001_2$, dove semplicemente si invertono i numeri dopo al bit di segno del 6.

2.3 Complemento a 2

Come il complemento ad 1, ma se il numero è negativo aggiunge anche un 1 in seguito all'operazione di complemento.

2.3.1 Esempio

- $6_{10} = \colorbox{yellow}{0}\,0000110_2$;

- $-6_{10} = \boxed{1}\,11110\,\boxed{10}_2$, dove guardando il secondo esempio del completamento a 1, è facile notare la differenza tra le due (le cifre evidenziate in verde).