

Corso di Architettura dei Sistemi a Microprocessore

Codifica Binaria



Luigi Coppolino

Rappresentazione dell'Informazione

- Tutta l'**informazione** in un calcolatore è rappresentata tramite numeri
- Questi numeri sono espressi come **sequenze di 0 e 1**
- I calcolatori **elaborano l'informazione** effettuando **operazioni su numeri**

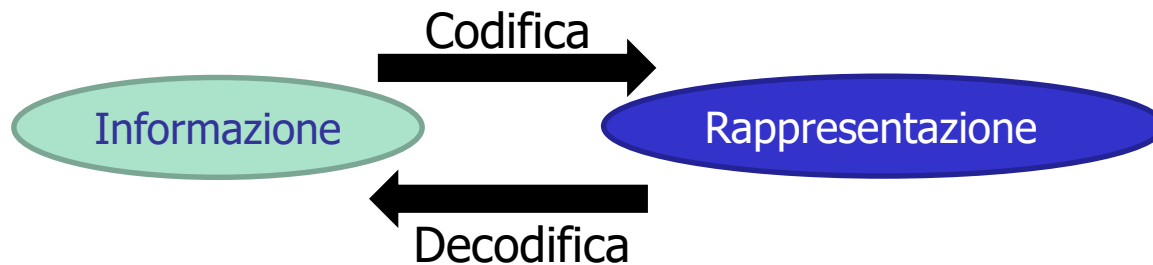
- I sistemi di comunicazione **scambiano informazioni spostando sequenze di numeri**

Problema

- Abbiamo informazioni (numeri, testi, immagini, suoni. . .) che vogliamo rappresentare (e poter elaborare) in un calcolatore.
- Vincolo: per motivi tecnologici un calcolatore lavora solo con i valori 0 e 1

Codifica e Decodifica

- Il processo che permette di ottenere la rappresentazione delle informazioni



- Il processo inverso è invece la decodifica
- La codifica è una convenzione!
- E' il modo in cui associamo un'informazione ad una sua rappresentazione, ad esempio quella binaria.

Rappresentazione dell'Informazione

- I computer sono sistemi digitali perché lavorano con digit (0 e 1)
- **Bit:** valore 0 (tensione bassa), valore 1 (tensione alta)
- **Byte:** sequenza di 8 bit. Esempio: 01110010
- L'informazione viene creata ed acceduta in unità di informazione dette **word**. Una word è creata come un multiplo di byte.
- Lunghezze di una word: 8 (byte), 16, 32, 64, 128 bit.
- Un word a k bit può contenere 2^k valori (da 0 a 2^{k-1}).

Numeri Naturali

- Numerazione araba: dieci cifre (0..9), **notazione posizionale**: il valore di una cifra dipende dalla sua posizione.
- La cifra più a sinistra ha un valore maggiore
- Ad esempio: 312, le tre cifre hanno valore diverso a seconda della posizione che occupano.
- La numerazione romana è **non posizionale** ma additiva. Ad esempio: MCCCX.

Numeri Naturali

- Sistemi di numerazione posizionale in base **p**:

$$N_p = a_n \times p^n + a_{n-1} \times p^{n-1} + \dots + a_1 \times p^1 + a_0 = \sum_{i=0}^n a_i \times p^i$$

- Se **p=10** si avrà:

$$745 = 7 \times 10^2 + 4 \times 10^1 + 5 \times 10^0$$

- Se **p=4** si avrà:

$$213 = 2 \times 4^2 + 1 \times 4^1 + 3 \times 4^0$$

Numeri Binari, Ottali, Esadecimali

➤ Sistema **Binario**

- $p=2$
- Alfabeto: $\{0,1\}$

$$\begin{aligned}10101101 &= (1x2^7 + 0x2^6 + 1x2^5 + 0x2^4 + 1x2^3 + 1x2^2 + 0x2^1 + 1x2^0) \\ &= (128 + 32 + 8 + 4 + 1) = 173_{10}\end{aligned}$$

$$\begin{aligned}01101010 &= (0x2^7 + 1x2^6 + 1x2^5 + 0x2^4 + 1x2^3 + 0x2^2 + 1x2^1 + 0x2^0) \\ &= (64 + 32 + 8 + 2) = 106_{10}\end{aligned}$$

$$\begin{aligned}00101111 &= (0x2^7 + 0x2^6 + 1x2^5 + 0x2^4 + 1x2^3 + 1x2^2 + 1x2^1 + 1x2^0) \\ &= (32 + 8 + 4 + 2 + 1) = 47_{10}\end{aligned}$$

Numeri Binari, Ottali, Esadecimali

➤ Sistema **Ottale**

- $p=8$
- Alfabeto: $\{0,1,2,3,4,5,6,7\}$

$$254_8 = 2 \times 8^2 + 5 \times 8^1 + 4 \times 8^0 = 128 + 40 + 4 = 172_{10}$$

➤ Sistema **Esadecimale**

- $p=16$
- Alfabeto= $\{0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F\}$

$$B7F_{16} = 11 \times 16^2 + 7 \times 16^1 + 15 \times 16^0 = 2943_{10}$$

Numeri Binari, Ottali, Esadecimali

decimale	binario	ottale	esadecimale
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10

Conversione da base 2 a base 10

- Si divide il numero per due e si prende il resto
- L'elenco dei resti si legge dall'alto verso il basso

$$\begin{array}{rcll} 173 : 2 = 86 & \rightarrow & 1 \\ 86 : 2 = 43 & \rightarrow & 0 \\ 43 : 2 = 21 & \rightarrow & 1 \\ 21 : 2 = 10 & \rightarrow & 1 \\ 10 : 2 = 5 & \rightarrow & 0 \\ 5 : 2 = 2 & \rightarrow & 1 \\ 2 : 2 = 1 & \rightarrow & 0 \\ 1 : 2 = 0 & \rightarrow & 1 \end{array} \rightarrow 10101101$$

Conversione da base 2 a base 8 e 16

- La conversione da base 2 a base 8:
 - Si divide il numero in **triple** e ogni tripla si traduce nella cifra ottale (01.101.100 → 154)
- La conversione da base 2 a base 16:
 - Si divide il numero in **quadruple** e ogni quadrupla si traduce nella cifra esadecimale (0011.1011 → 3B).

Esercizio

- Convertire in base 10 e in base 16 i seguenti numeri:
 - 1101010
 - 1110001
 - 1010101110
- Convertire in base 2 i seguenti numeri in base 10:
 - 139
 - 255
 - 180