

# Programma del corso

---

- *Introduzione*
  - **Rappresentazione delle Informazioni**
  - *Calcolo proposizionale*
  - *Architettura del calcolatore*
  - *Reti di calcolatori*
-

# Codifica dell'informazione

---

- Il calcolatore memorizza ed elabora vari tipi di informazioni
    - Numeri, testi, immagini, suoni
  - Occorre rappresentare tale informazione in formato facilmente manipolabile dall'elaboratore
-

# Rappresentazione delle informazioni

---



## Idea di fondo

- usare presenza/assenza di carica elettrica
- usare passaggio/non passaggio di corrente

Usiamo cioè una rappresentazione binaria (a due valori) dell'informazione

L'unità minimale di rappresentazione è il **BIT** (**BI**nary digi**T** – cifra digitale): **0** o **1**

---

# Informazioni complesse

---

Con 1 bit rappresentiamo solo 2 diverse informazioni:

**si/no - on/off - 0/1**

Mettendo insieme più bit possiamo rappresentare più informazioni:

**00 / 01 / 10 / 11**

**Informazioni complesse si memorizzano come sequenze di bit**

---

# Informazioni complesse

---

- Per codificare i nomi delle 4 stagioni bastano 2 bit
  
  - Ad esempio:
    - **0 0** per rappresentare **Inverno**
    - **0 1** per rappresentare **Primavera**
    - **1 0** per rappresentare **Estate**
    - **1 1** per rappresentare **Autunno**
  
  - Quanti bit per codificare i nomi dei giorni della settimana?
-

# Informazioni complesse

---

In generale, con **N** bit, ognuno dei quali può assumere **2** valori, possiamo rappresentare  **$2^N$**  informazioni diverse (**tutte le possibili combinazioni di 0 e 1 su N posizioni**)

viceversa

Per rappresentare **M** informazioni dobbiamo usare **N** bit, in modo che  **$2^N \geq M$**

---

# Esempio

---

Per rappresentare **57** informazioni diverse:

**5** bit non bastano, poiché  $2^5 = 32 < 57$

**6** bit invece bastano:  $2^6 = 64 > 57$

Cioè un gruppo di 6 bit può assumere

64 configurazioni diverse:

000000 / 000001 / 000010 ... / 111110 / 111111

# Il Byte

---

□ Una sequenza di **8 bit** viene chiamata **Byte**

■ 0 0 0 0 0 0 0 0

■ 0 0 0 0 0 0 0 1

■ .....

**byte = 8 bit =  $2^8$  = 256 informazioni diverse**

Usato come unità di misura per indicare

■ le dimensioni della memoria

■ la velocità di trasmissione

Usando sequenze di byte (e quindi di bit) si possono rappresentare caratteri, numeri, immagini, suoni.

---

# Altre unità di misura

---

- KiloByte (**KB**), MegaByte (**MB**), GigaByte (**GB**)
  - Sistema Internazionale (SI):
    - 1 KB = 1000 x 1 byte =  $10^3$  byte
    - 1 MB = 1000 x 1 KB =  $10^6$  byte
    - 1 GB = 1000 x 1 MB =  $10^9$  byte
  - In informatica si adotta anche un altro sistema di misure – IEC - International Electrotechnical Commission:
    - 1 KB = 1024 x 1 byte =  $2^{10} \sim 10^3$  byte
    - 1 MB = 1024 x 1 KB =  $2^{20} \sim 10^6$  byte
    - 1 GB = 1024 x 1 MB =  $2^{30} \sim 10^9$  byte
  - IEC è usato per la capacità della memoria di un elaboratore
  - SI è usato per le capacità degli hard disk
-

# Il sistema decimale

---

- 10 cifre di base: 0, 1, 2, ..., 9
  - **Notazione posizionale:** la posizione di una cifra in un numero indica il suo **peso** in potenze di **10**. I pesi sono:
    - unità =  $10^0 = 1$  (posiz. 0)
    - decine =  $10^1 = 10$  (posiz. 1)
    - centinaia =  $10^2 = 100$  (posiz. 2)
    - migliaia =  $10^3 = 1.000$  (posiz. 3)
    - $10^4 = 10.000$ ;  $10^5 = 100.000$ ;  $10^6 = 1.000.000$ ; ...
-

# Esempio: numero rappresentato in notazione decimale

---

Il **numerale** 2304 in notazione decimale (o in base 10) rappresenta la quantità:

$$2304 = 2 * 10^3 + 3 * 10^2 + 0 * 10^1 + 4 * 10^0 =$$

$$2000 + 300 + 0 + 4 = 2304 \text{ (**numero**)}$$

**Nota:** numero e numerale qui coincidono, perché il sistema decimale è quello adottato come sistema di riferimento.

---

# Il sistema binario

---

- 2 Cifre di base: 0 e 1.
  - **Notazione posizionale:** la posizione di una cifra in un numero binario indica il suo **peso** in potenze di **2**. I pesi sono:
    - $2^0 = 1$  (posiz. 0)
    - $2^1 = 2$  (posiz. 1)
    - $2^2 = 4$  (posiz. 2)
    - $2^3=8; 2^4=16; 2^5=32; 2^6=64; 2^7=128;$   
 $2^8=256; 2^9=512; 2^{10} = 1024; 2^{11}=2048,$   
 $2^{12}=4096; \dots$
-

# Esempio di numero rappresentato in notazione binaria

---

Il **numerale** 10100101 in notazione binaria (o in base 2) rappresenta la quantità:

10100101

$$1*2^7 + 0*2^6 + 1*2^5 + 0*2^4 + 0*2^3 + 1*2^2 + 0*2^1 + 1*2^0 =$$

$$128 + 0 + 32 + 0 + 0 + 4 + 0 + 1 =$$

165 (**numero**)

---

# Il numero più grande rappresentato con **N** cifre

---

- Sist. Decimale = 99...99 =  $10^N - 1$
  - Sist. Binario = 11..11 =  $2^N - 1$
  - **Esempio:** 11111111 (8 bit binari) =  $2^8 - 1$   
= 255. Per rappresentare il n. 256 ci vuole  
un bit in più: 100000000 =  $1 * 2^8 = 256$ .
-

# Quindi...

---

Fissate quante cifre (bit) sono usate per rappresentare i numeri, si fissa anche il numero più grande che si può rappresentare:

- con 16 bit:  $2^{16} - 1 = 65.535$

- con 32 bit:  $2^{32} - 1 = 4.294.967.295$

- con 64 bit:  $2^{64} - 1 = \text{circa } 1,84 * 10^{19}$

---

# Conversione da base 2 a base 10

---

Basta moltiplicare ogni bit per il suo peso e sommare il tutto:

**Esempio:**

10100

$$1*2^4 + 0*2^3 + 1*2^2 + 0*2^1 + 0*2^0 =$$

$$16 + 4 = 20$$

La conversione e' una **somma di potenze**

(N.B. se il numero binario termina per 1 e' dispari, altrimenti e' pari).

---

# Conversione da base 10 a base 2

---

- Dividere il numero per 2 ripetutamente finché il risultato non è 0
- Scrivere i resti in ordine inverso.

**Esempio:** conversione del numero **12**

Divisioni:  $12/2=6$ ;  $6/2=3$ ;  $3/2=1$ ;  $1/2=0$

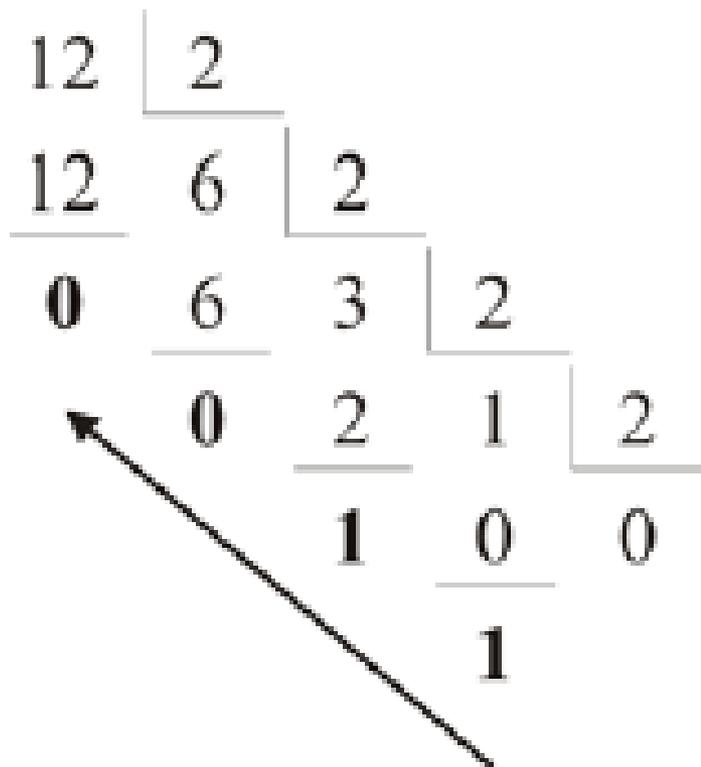
Resti:           0           0           1           1

$$12_{10} = 1100_2$$

---

# Conversione da base 10 a base 2

---



# Esistono anche altre basi di numerazione

---

## □ CODICE OTTALE

- cifre: 0, 1, 2, 3, 4, 5, 6, 7
- 10 (ottale) = 8 (decimale)

## □ CODICE ESADECIMALE

- "cifre": 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F
  - 10 (esadecimale) = 16 (decimale);
  - $B = 11$ ;  $2B = 2 * 16^1 + B * 16^0 = 32 + 11 = 43$
-