CIAO A TUTTI

Maurizio Minieri

Università degli Studi di Napoli Federico II

Sito Personale: **www.mauriziominieri.it**

E-mail: **mauminieri@gmail.com**

CODICE BINARIO

Il computer è in grado di elaborare le informazioni solo se esse sono rappresentate nel codice binario B={0,1}, costituito da 2 simboli:

- acceso (passaggio di 12 volts), rappresentabile con il numero 1;
- spento (assenza di corrente), rappresentabile con il numero 0.

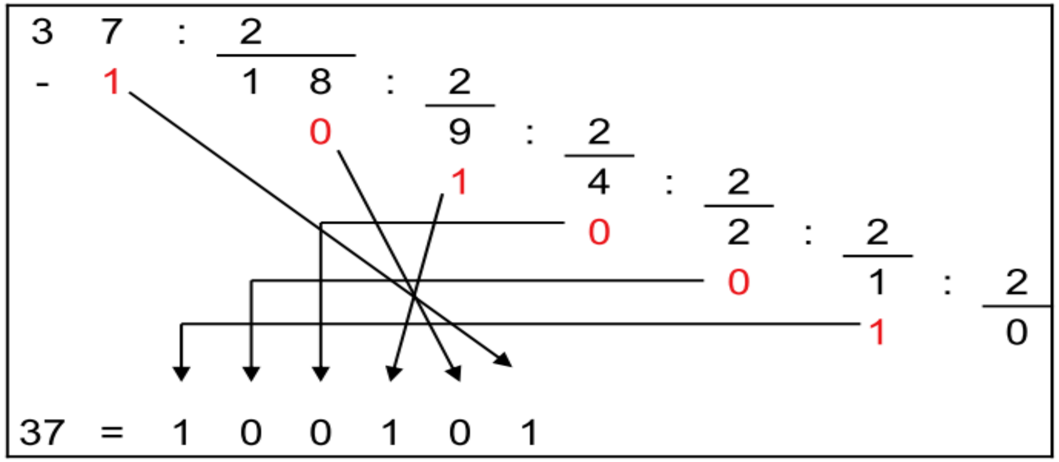
Qualunque informazione può essere rappresentata in binario:

- Numeri;
- Lettere;
- Simboli speciali;
- Immagini;
- Suoni;
- …

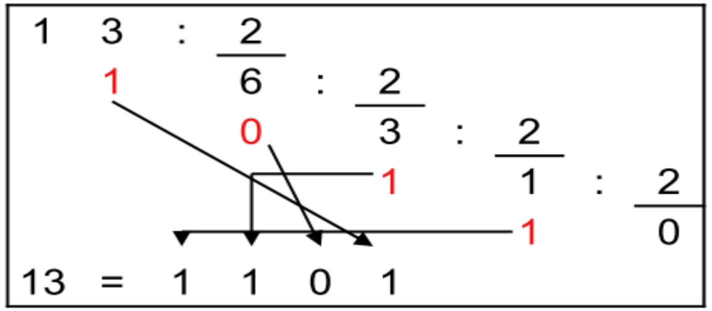
DA DECIMALE A BINARIO

INFORMAZIONE NUMERICA:
Solitamente, rappresentiamo i numeri in base decimale, cioè usando i 10 simboli D={0,1,…,9}.

Per trasformare un decimale # in binario, si procede dividendo # per 2 e prendendo i resti della divisione in ordine inverso



37 ⟺ 100101



13 ⟺ 1101

SOMMA TRA NUMERI BINARI

<u>INFORMAZIONE NUMERICA</u>: addizione

Nel sistema binario, è possibile ovviamente anche sommare due numeri.

Di seguito è mostrata la somma di 100101 (37) e 1101 (13):

riporti →		1	1		1	
	1	0	0	1	0	1
			1	1	0	1
risultato →	**1**	**1**	**0**	**0**	**1**	**0**

Verifica: 37+13 = 50. Verifichiamo che 110010 è 50 in binario.

- **<u>La somma tra i due numeri è possibile nonostante il loro numero di bit sia diverso</u>**. 1101 ha 4 bit, per adattarlo ai 6 bit lo si trasforma in 001101. Scrivere i due 0 iniziali è superfluo.
- Per verificare che il numero 110010 è veramente il 50 decimale c'è un procedimento ben preciso.

SOTTRAZIONE TRA NUMERI BINARI

- **La sottrazione funziona allo stesso modo della sottrazione decimale, basta ricordare che:**
  - 0 – 0 = 0
  - 0 – 1 = 1, scalando a sinistra
  - 1 – 0 = 1
  - 1 – 1 = 0

decimale	binaria
9 0 0 0 -	1 0 1 -
0 1 1 1	0 1 1
8 8 8 9	0 1 0

- **Esempio**: 101 – 011, cioè 5 – 3. Proviamo 1010 – 111. Oppure 1000.
- In decimale per fare 9000 – 111 partiamo da destra, lo 0 diventa 10 e facciamo 10 - 1= 9, gli 0 a sinistra scalano tutti a 9, il 9 iniziale scala a 8. Allo stesso modo funziona la sottrazione binaria, ma sarà molto più facile, perché gli 1 diventano semplicemente 0 e gli 0 diventano 1.

RAPPRESENTAZIONE DEI NUMERI

- **Generalmente risulta scomodo trattare lunghe stringhe di bit**: si fa uso di sistemi numerici che consentono di esprimere in maniera più compatta le lunghe stringhe di 0 e 1
- **Per eseguire le conversioni bisogna ragionare a seconda della base**: **un numero decimale ha base 10, binario 2, ottale 8 e esadecimale 16.**
  - **Le cifre binarie sono: 0 e 1**
  - **Ottali: 0, 1, 2, 3, 4, 5, 6, 7**
  - **Esadecimali: 0,1, ..., 9, A, B, C, D, E, F**

- Es. da binario a decimale: $(101)_2 = 1 * 2^2 + 0 * 2^1 + 1 * 2^0 = 4 + 0 + 1 = (5)_{10}$
- Considerando l'esempio di prima, 110010 quindi sarà: 32 + 16 +2 = 50
- Da binario a ottale si dividono i bit in gruppi da 3(da destra), da binario a esadecimale in gruppi da 4(da destra).
- Con lo stesso identico procedimento si possono convertire i numeri in altre basi, come $(205)_8 = (133)_{10}$, oppure $(2F)_{16} = (47)_{10}$

RAPPRESENTAZIONE DEI NUMERI - ESEMPI

- Es. da binario a ottale: $(101110)_2$ → *gruppi da 3* → $(101)_2$ & $(110)_2$ = $(56)_8$
- Es. da binario a esadecimale: $(101110)_2$ → *gruppi da 4* → $(0010)_2$ & $(1110)_2$ = $(2E)_{16}$
- Es. da ottale a binario: $(56)_8$ → *gruppi da 3* → 101 & 110 = $(101110)_2$
- Es. da esadecimale a binario: $(2E)_{16}$ → *gruppi da 4* → 0010 & 1110 = $(00101110)_2$
- Es. da decimale a ottale: $(46)_{10}$ → 46/8 = 5 con resto 6, 5/8 = 0 con resto 5 = $(56)_8$
- Es. da decimale a esadecimale: $(46)_{10}$ → 46/16 = 2 con resto 14, 2/16 = 0 con resto 2 = $(2E)_{16}$
- Es. da ottale a decimale: $(56)_8$ → $5 * 8^1 + 6 * 8^0 = 40 + 6 = (46)_{10}$
- Es. da esadecimale a decimale: $(2E)_{16}$ → $2 * 16^1 + 14 * 16^0 = 32 + 14 = (46)_{10}$

CODICE ASCII

INFORMAZIONE ALFABETICA:

I caratteri di stampa ed i caratteri speciali sono trasformati in binario attraverso il **codice ASCII**:

American **S**tandard **C**ode for **I**nformation **I**nterchange.

Ad ogni carattere corrisponde 1 byte (8 bit – 1 bit è 0 o 1). Dunque, per memorizzare "Amico" occorrono 5 byte.

Byte								Carattere
0	1	0	0	0	0	0	0	A
0	1	1	0	1	1	0	0	m
0	1	1	0	1	0	0	0	i
0	1	1	0	0	0	1	1	c
0	1	0	0	0	0	0	0	o

Per memorizzare una pagina di un libro, occorrono mediamente 2.000 caratteri, cioè 2.000 byte.

Per memorizzare un libro di 300 pagine, occorrono mediamente 600.000 byte, cioè circa 600 KB.

Byte								Carattere
0	0	1	0	0	1	0	0	%
0	0	1	0	1	0	0	0	)
0	0	1	1	1	1	1	1	@
0	1	0	0	0	0	0	0	A
0	1	1	0	1	0	0	0	ì

L'MMAGINE A COLORI

<u>ANCHE LE IMMAGINI DIVENTANO NUMERI</u>:

Il testo scritto consta di un numero finito di caratteri, ciascuno corrispondente ad un gruppo di 8 bit, ciascuno dei quali sarà 0 oppure 1.

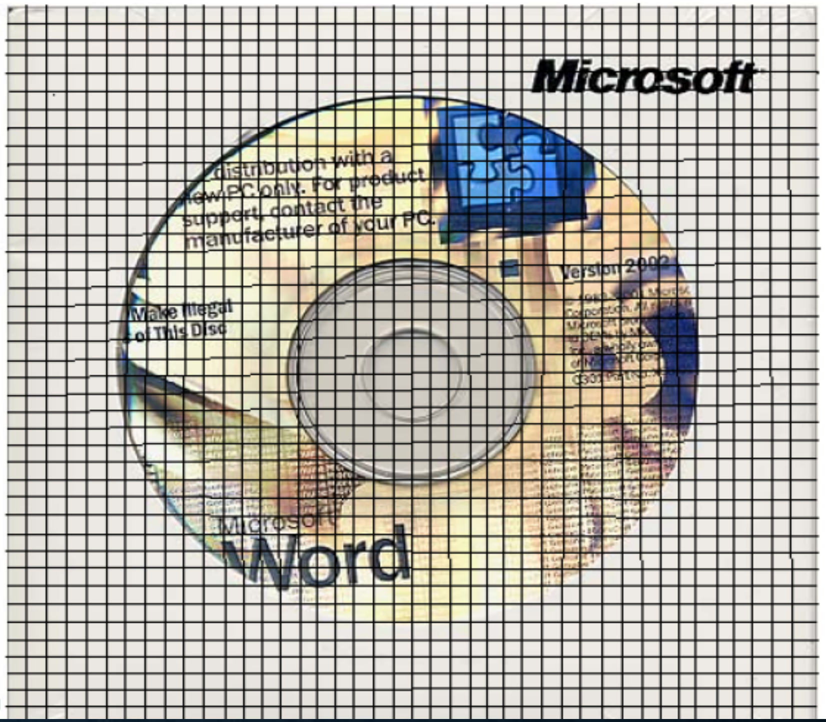
In che modo si può trasformare in una sequenza di numeri binari un'immagine a colori???

Si inserisce l'immagine in una griglia.

Ogni celletta della griglia è detta pixel.

La trasformazione avviene assegnando a ogni pixel un numero binario che codifichi il suo colore oppure la sfumatura del suo colore.

Quanto più fitta è la griglia tanto maggiore sarà la "definizione" dell'immagine.



L'IMMAGINE

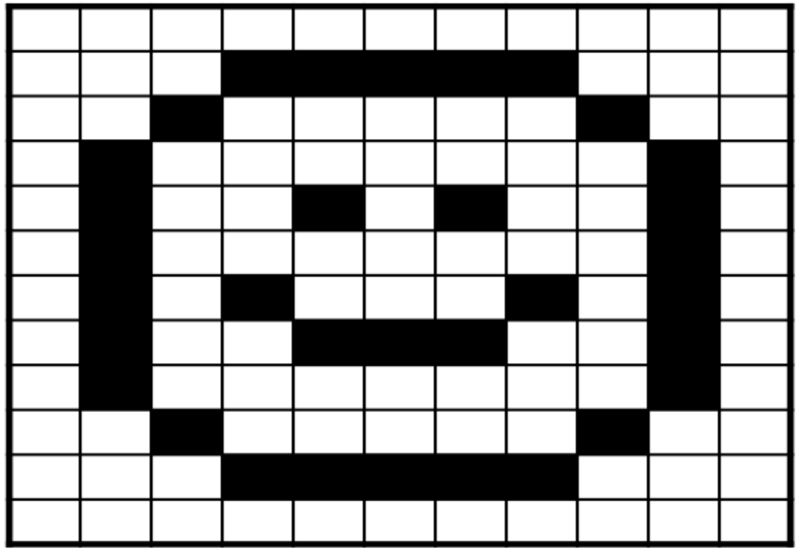
ANCHE LE IMMAGINI DIVENTANO NUMERI:

Se l'immagine è in bianco e nero, è sufficiente utilizzare un bit per ogni pixel: 0 indicherà bianco; 1 indicherà nero.

Se l'immagine è a colori, ogni pixel ha un suo colore, per cui non basta un solo bit.

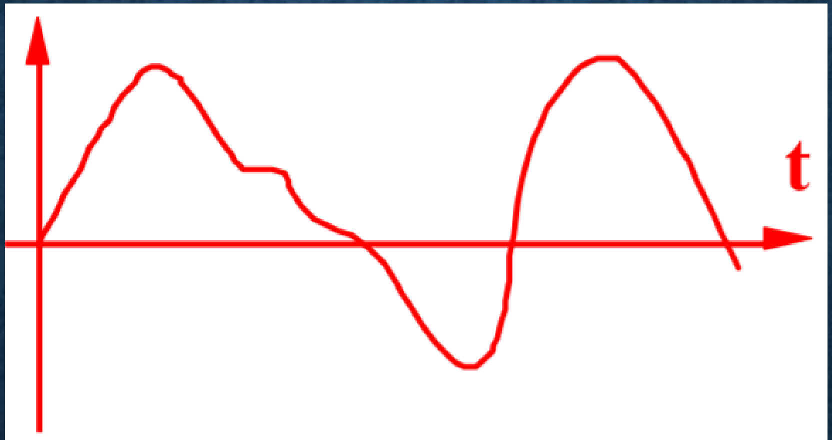
Con 8 bit si possono codificare 256 $= 2^8$ colori; con 16 bit si possono codificare 65.536 $= 2^{16}$ colori; con 24 bit si possono codificare 16 milioni di colori.

Per trasformare in digitale un film, occorre trasformare ogni sua singola immagine.



IL SUONO

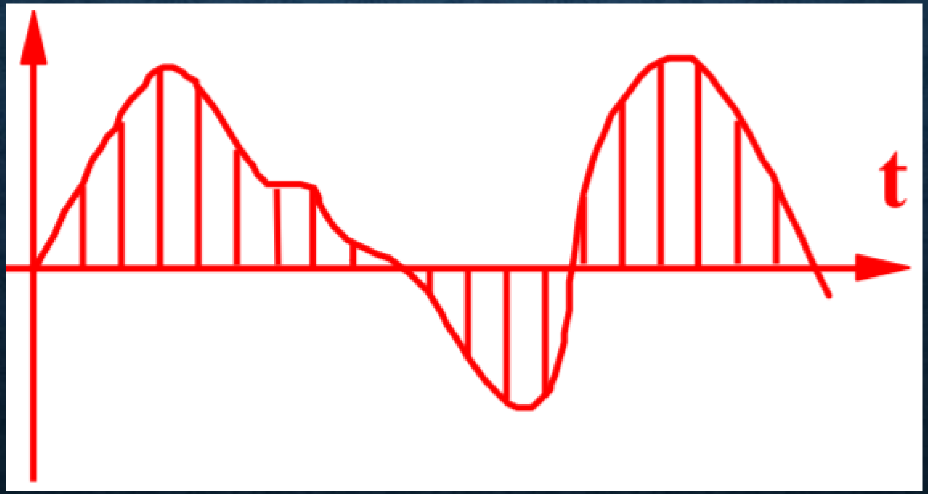
- **Dal punto di vista fisico un suono è un'alterazione della pressione dell'aria** che, quando rilevata, ad esempio dall'orecchio umano, viene trasformata in un particolare stimolo elettrico.
- **La durata, l'intensità e la frequenza della variazione nel tempo della pressione dell'aria sono le quantità fisiche che rendono un suono diverso da ogni altro**.



- **Fisicamente un suono è rappresentato come un'onda(onda sonora)** che descrive la variazione della pressione dell'aria nel tempo t. **Sull'asse delle ascisse viene rappresentato il tempo e sull'asse delle ordinate viene rappresentata la variazione di pressione**.

IL SUONO - CAMPIONAMENTO

- **La rappresentazione precedente viene detta analogica**, in quanto descrive esattamente l'analogo della quantità fisica in esame, e fornisce una descrizione continua dell'onda sonora
- **È necessario trovare un modo di per rappresentare in forma digitale(numerica) un'onda sonora**
- **Si effettuano dei campionamenti sull'onda sonora (cioè si misura il valore dell'ampiezza dell'onda a intervalli costanti di tempo), e si codificano in forma digitale le informazioni estratte da tali campionamenti**



IL SUONO – ESEMPIO DI CAMPIONAMENTO

- **Prendiamo per esempio una classica sinusoide elementare** (*Figura 1).*
- **Ipotizziamo di possedere un dispositivo che effettui in un certo lasso di tempo un certo numero di campioni del segnale**: **14 campioni per periodo della sinusoide**. Otterremo una serie di campioni come quella in *Figura 2*. **Vediamo che la sinusoide originaria è ancora intuibile, per cui sarà possibile ricostruirla e invertire il procedimento**.
- **Immaginiamo di dimezzare la frequenza del campionamento**. Otterremo una diversa serie di campioni, meno fitta della precedente (*Figura 3*). **La sinusoide è ancora intuibile ma è evidente che abbiamo perso parte dell'informazione originale**.
- **Dimezzando ancora, la situazione è critica** (*Figura 4*). Non potremo ricostruire la sinusoide originaria.

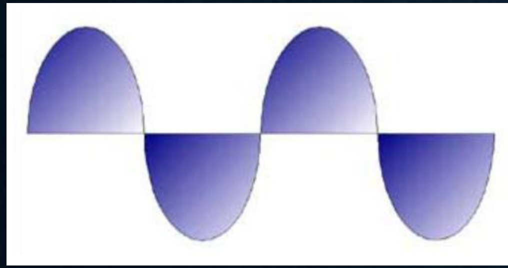


Figura 1

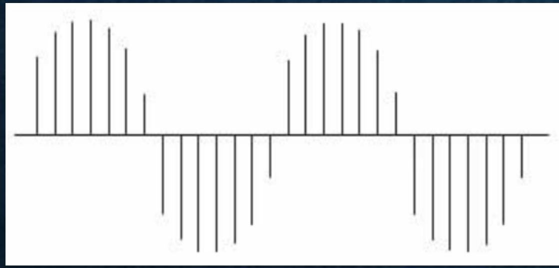


Figura 2

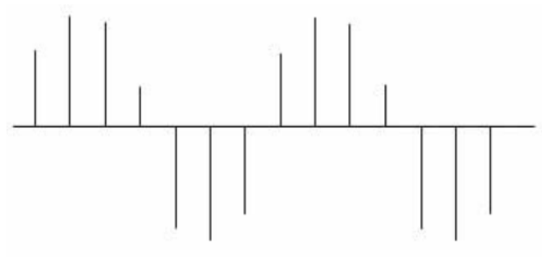


Figura 3

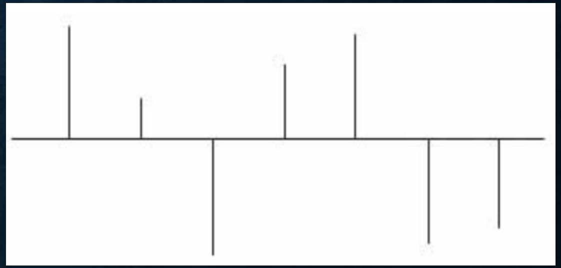


Figura 4

IL SUONO

- **Abbiamo capito che:**

1. **Maggiore è la frequenza di campionamento più accurato è la descrizione del segnale**, ovviamente un maggior numero di campioni richiede un maggior spazio in memoria.
2. **C'è un punto critico, al di sotto del quale la frequenza di campionamento non può scendere**, pena la perdita totale dell'informazione.

- **Esiste una frequenza di campionamento minima per ottenere descrizioni accurate?**

IL SUONO - CODIFICA

- **Si, il tasso di Nyquist**: **Afferma che la minima frequenza di campionamento di un segnale necessaria per evitare perdita di informazione, deve essere ALMENO il doppio della sua frequenza massima**.

- Ad esempio: se vogliamo campionare il segnale di un violino, che arrivi, ad esempio, fino a 15.000Hz, sarà necessaria una frequenza di campionamento di almeno 30.000Hz, ossia si dovranno prendere almeno 30.000 misurazioni al secondo.
- Nella pratica, **dato che il suono udibile per l'orecchio umano è compreso tra i 20 e i 20.000Hz, una frequenza di 40.000Hz è sufficiente a campionare ogni possibile suono udibile dall'uomo** (tralasciando le problematiche dei vari dispositivi).

- **Clicca su**: Frequenza del suono per vedere il video di dimostrazione della frequenza del suono udibile dall’uomo.

- **Come possiamo sfruttare questi campionamenti?**

IL SUONO – QUANTIZZAZIONE

- **Rappresenta l'ampiezza del segnale nell'istante in cui avviene il campionamento.**
- **Ipotizziamo che il calcolatore metta a disposizione 3 bit per la quantizzazione** (*Figura 1*). Questo significa che avremo a disposizione i valori binari da 000 a 111, ossia solo 8 possibili valori. Questo ci costringerà ad avere un grafico digitale piuttosto rozzo.
- **Vediamo ad esempio come cambia la rappresentazione della sinusoide, passando da una quantizzazione a 3 bit ad una a 4 bit** (*Figura 2*). E' facile osservare che un solo bit in più aumenta di molto la precisione del grafico digitale.

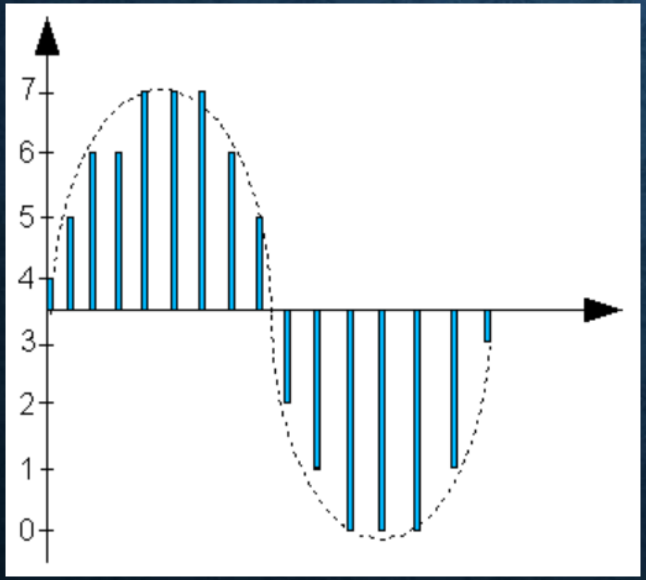


Figura 1

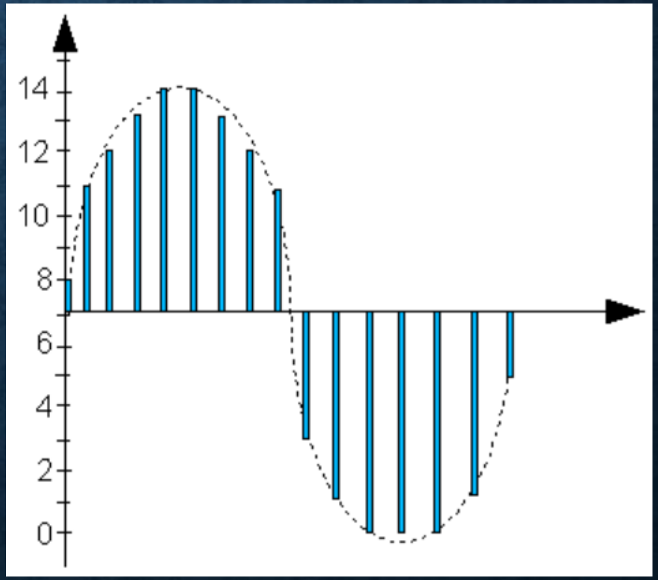


Figura 2

IL SUONO – MUSICA NEI CD

- Il tasso di campionamento dei CD, per adattarsi alla percezione umana del suono, è 44100 volte al secondo, si usano due registrazioni corrispondenti a due microfoni, e si usano 16 bit.
- 1 secondo di musica qualità CD richiede:
    - (44100 * 16 * 2)bit = 1411200bit = 176400 byte = 176 kB
- 1 minuto di musica quindi richiede:
    - 84672000 bit = 10584000 byte = 10MB