

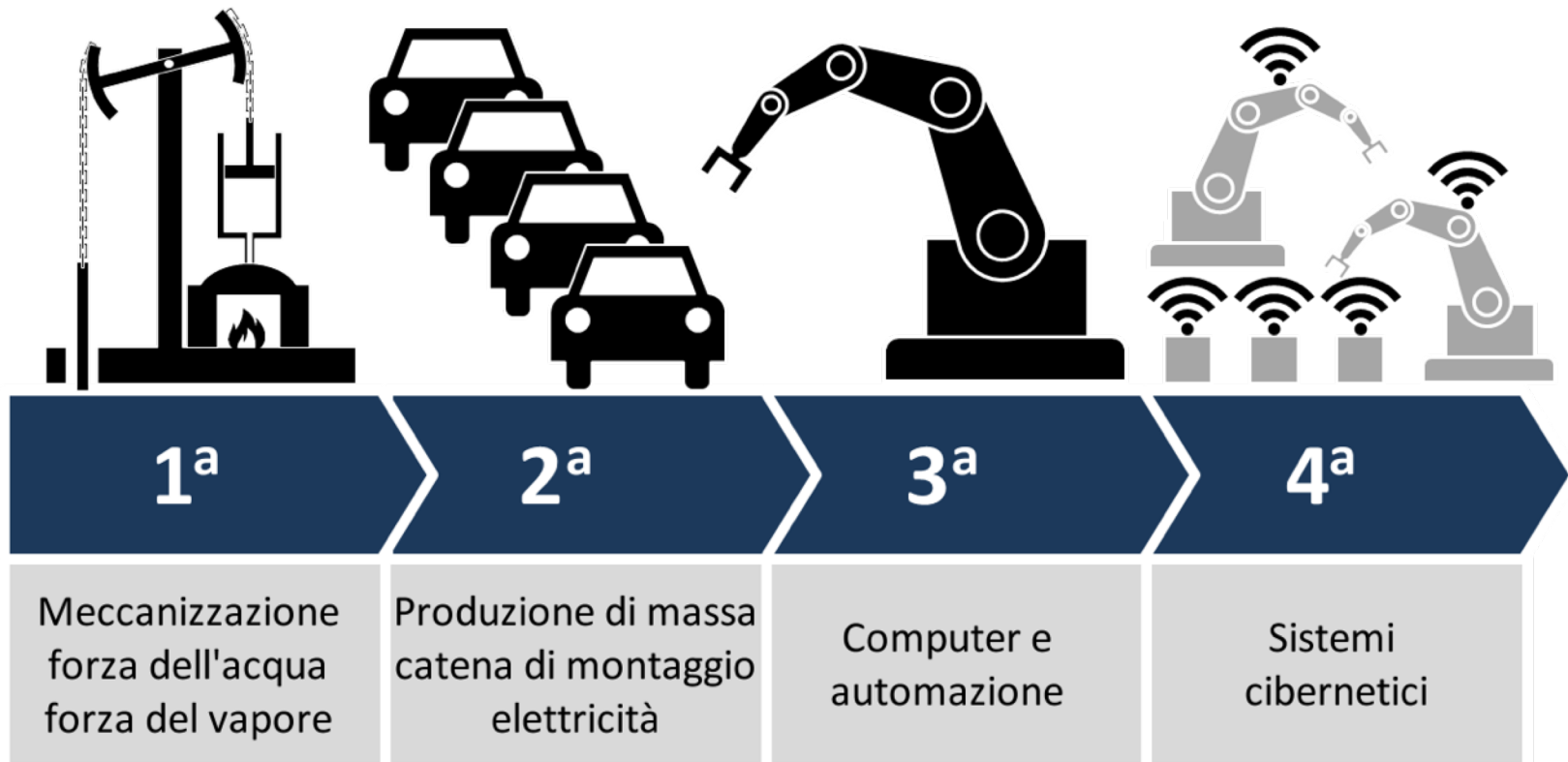
Perché informatica?

- L'informatica è ovunque
- L'informatica ha moltissime branche, fornisce molte professioni e opportunità
- Non solo i professionisti, anche gli utenti è bene che conoscano principi di informatica di base
- Il futuro probabilmente è informatico

Storia dell'uomo

- Nei rari periodi in cui l'uomo non è stato impegnato ad uccidere o invadere, ha sempre cercato di migliorare la propria vita
- Ha inizialmente migliorato il mondo liberandosi della fatica fisica (lavoro)
- Quindi producendo prodotti in massa (industria)
- Ora però nasce un problema.....

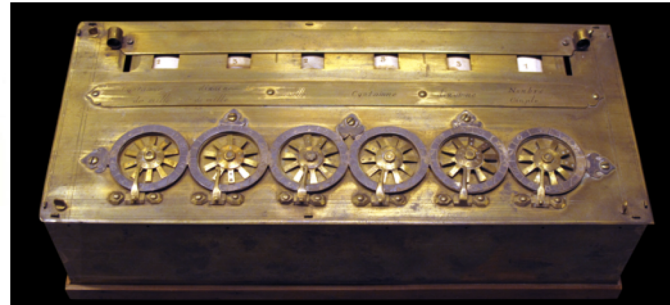
Le rivoluzioni industriali



Timeline 1



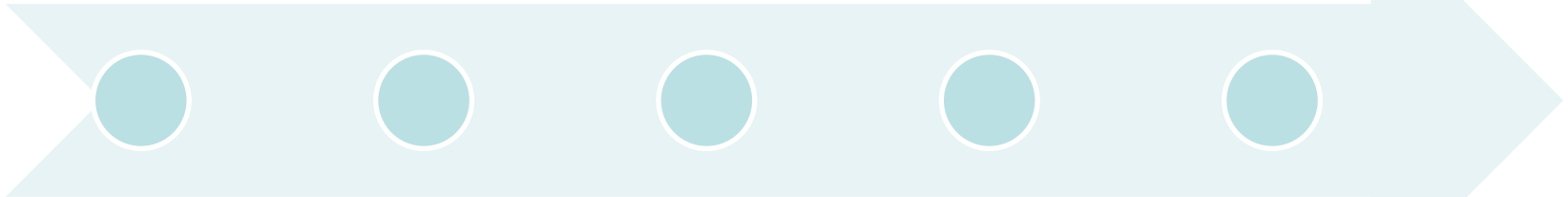
100 AC
Macchina
di
Antikythera



1642
Pascalina



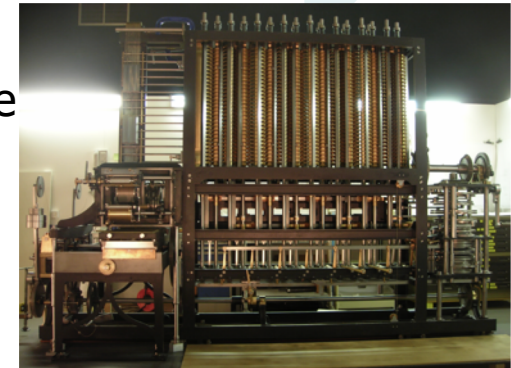
1889
Hollerith
brevetta la
scheda
perforata



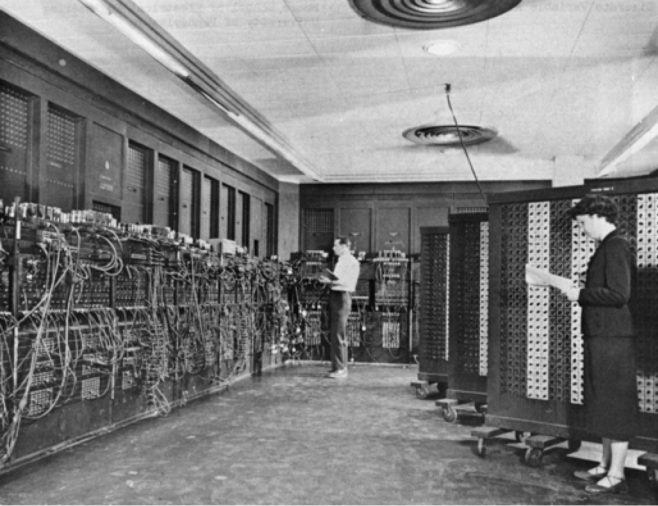
Medioevo
Abaco



1833
Macchina
di Babbage



Timeline 2



1944
Colossus,
ENIAC

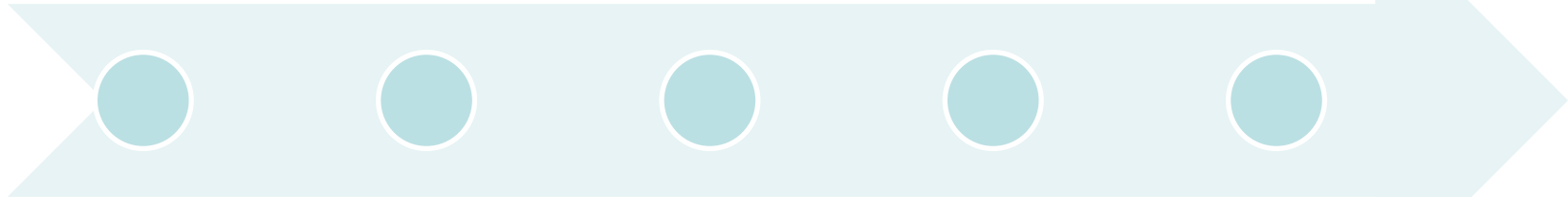


1964
Perottina

MICRO
SOFT

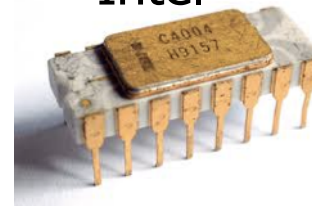


1975-76
Microsoft
e Apple



1951
UNIVAC

1968
Intel



Timeline 3



1981
IBM PC



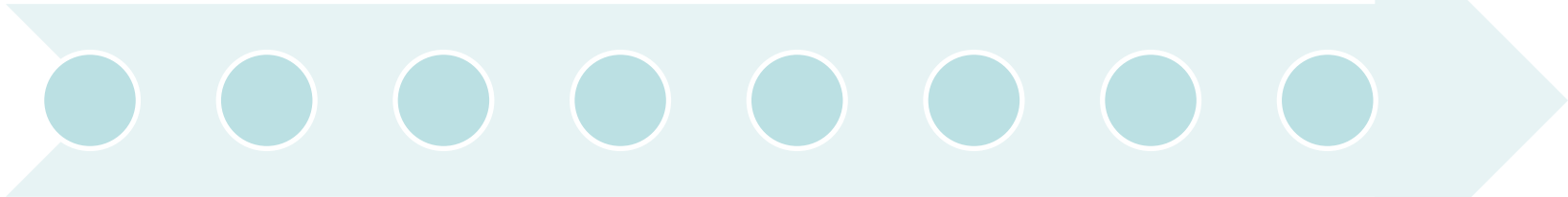
1985
Windows



1991 Linux



1997
Deep Blue



Timeline 4



2000
USB Flash
Drive



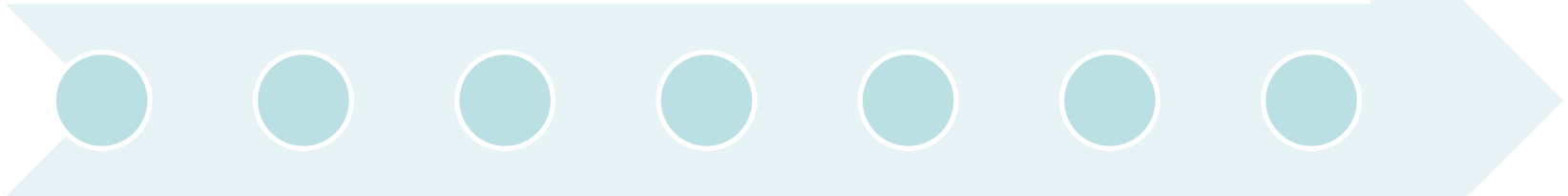
2006
AWS



2008
Facebook



2010
iPhone



2005
Google



2007
Scratch



2009
Minecraft,
BitCoin



L'informatica si nasconde anche...

- Televisori
- Sistemi di sorveglianza con riconoscimento
- Digital signage (nei negozi per coinvolgere le persone)
- Semafori intelligenti
- Google Home, Amazon Echo, domotica, Internet of Things
- Elettrodomestici: dal robot aspirapolvere al frigorifero
- Playstation
- Sistemi bancari, POS
- Giocattoli e giochi in scatola con sensori
- Sistemi di comunicazione e social
- Agricoltura (monitoraggio irrigazione)

Cosa faremo

- Cosa è l'informatica: concetti di base
- Architettura dei computer
- Dati e informazioni, codifica
- Il sistema operativo
- Programmi di produttività
- Le reti, il cloud
- Sicurezza, crittografia, blockchain, i rischi
- Programmazione?

Concetti di Base

Tecnologie Informatiche
Istituti Tecnici - Classi Prime

Panoramica

- Che cos'è l'informatica?
- Dati ed informazioni
- Tipi di dato
- Rappresentazione di numeri
- Rappresentazione di testi
- Rappresentazione di immagini
- Rappresentazione di suoni
- Rappresentazione di video

Che cos'è l'Informatica?

Informatica /in·for·mà·ti·ca/ *sostantivo femminile*

La scienza che si occupa dell'ordinamento, del trattamento e della trasmissione delle informazioni per mezzo dell'elaborazione elettronica, la quale rende possibile gestire e organizzare le ingenti masse di dati prodotte dal moderno sviluppo sociale, scientifico e tecnologico.

Origine: Dal fr. *informatique*, comp. di *informat(ion)* 'informazione' e *(automat)ique* 'automatica', voce coniata nel 1962 •1968.

Si può dire che l'informatica è la disciplina che studia l'elaborazione automatica delle informazioni

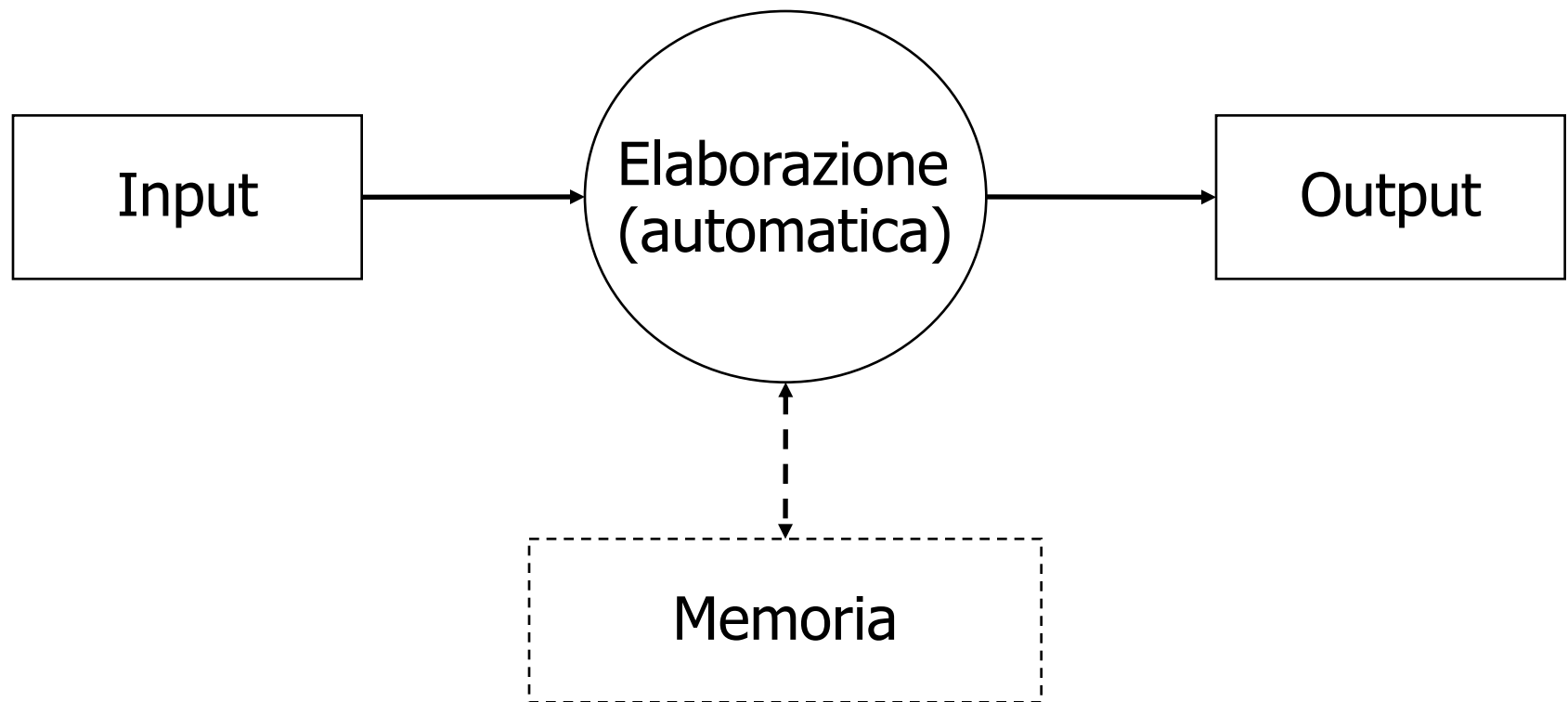
Alcuni esempi

Un foglio di carta contiene informazioni, ma non le elabora automaticamente! Uno stereo suona ma non elabora.

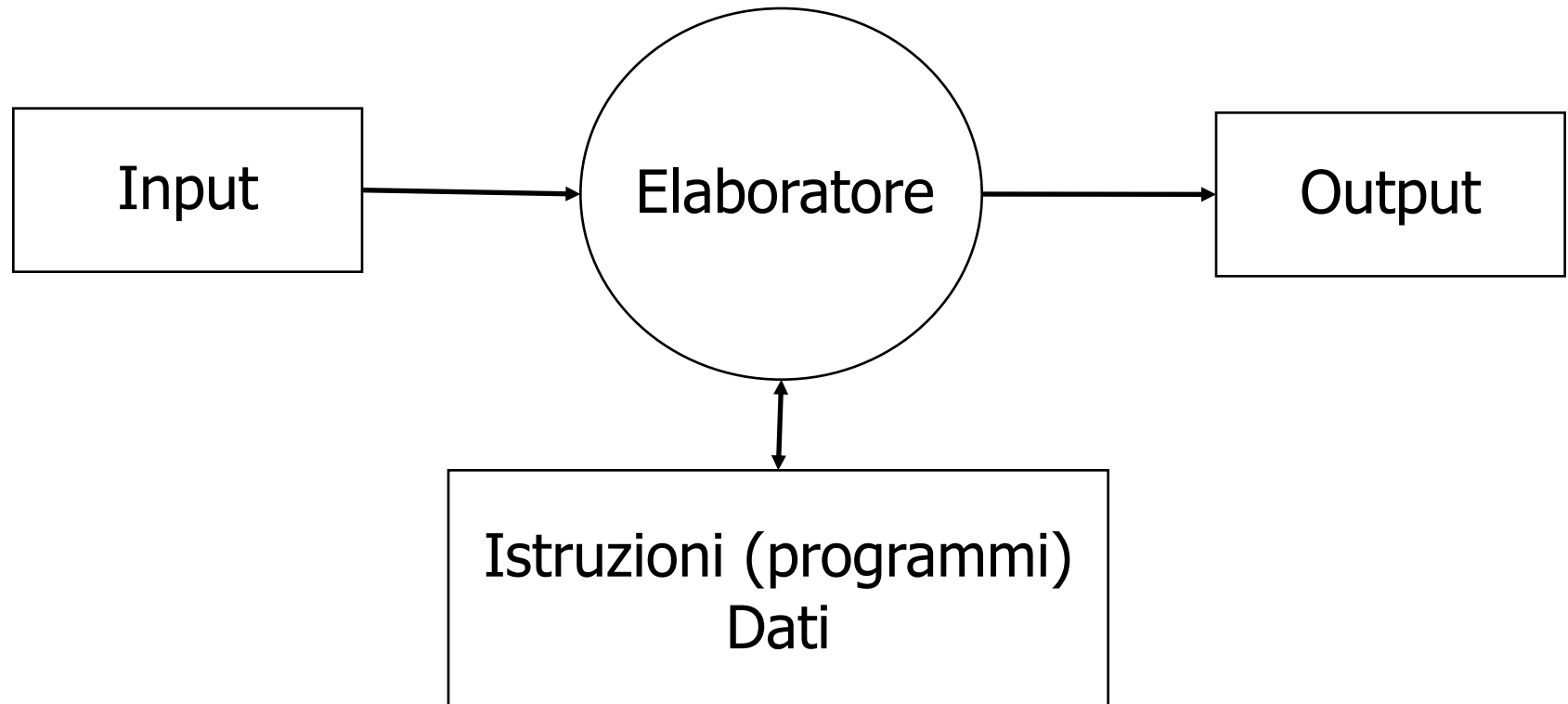
Strumenti che gestiscono l'informazione in maniera automatica:

- Calcolatrice tascabile
- Registratore di cassa (meccanico, elettromeccanico, elettronico, con lettore di codici a barre)
- Navigatore satellitare (es.: TomTom)
- Pilota automatico (aereo, nave, in futuro auto)
- Personal computer e sue varianti (notebook, tablet, smartphone)
- Supercomputer meteorologici

Schema generale



Elaboratore programmabile (Computer)



Dati ed Informazioni - 1

- Un dato è un simbolo con cui viene rappresentata la realtà

Esempio: il numero **20010911**

- È un dato che non significa nulla se non se ne conosce il significato (semantica)

20010911 (aaaammgg) => **11 Settembre 2001**

informazione = dato + significato

- In altre parole: il computer elabora **dati** da cui noi, interpretandoli, traiamo **informazione**

Dati ed Informazioni - 2

Dato:

Rossi Marco Milano 111295 Genova Roma 13 4 0101234567

Informazione:

Cognome: Rossi

Nome: Marco

Nato a: Milano

Il: 11 Dicembre 1995

Residente a: Genova

Via: Roma 13/4

Tel: 0101234567

Dettagli insignificanti?

Il diavolo si nasconde nei dettagli ...

The **Mars Climate Orbiter** was intended to enter orbit at an altitude of 140–150 km above Mars. However, a navigation error caused the spacecraft to reach as low as 57 km. **The spacecraft was destroyed** by atmospheric stresses and friction at this low altitude. The navigation error arose because a NASA subcontractor (Lockheed Martin) **used Imperial units (pound–seconds) instead of the metric units (newton–seconds)** as specified by NASA.

On June 4, 1996 an unmanned **Ariane 5 rocket launched by the European Space Agency exploded** just forty seconds after its lift-off from Kourou, French Guiana. The rocket was on its first voyage, after a **decade of development costing \$7 billion**. The destroyed rocket and its cargo were valued at **\$500 million**. (...) Specifically a **64 bit floating point number** relating to the horizontal velocity of the rocket with respect to the platform **was converted to a 16 bit signed integer**. The number was larger than 32,767, the largest integer storeable in a 16 bit signed integer, and thus the conversion failed.

On February 25, 1991, during the Gulf War, an **American Patriot Missile** battery in Dharan, Saudi Arabia, **failed to track and intercept an incoming Iraqi Scud missile**. The Scud struck an American Army barracks, **killing 28 soldiers and injuring around 100 other people**. (...) Because of the way the Patriot computer performs its calculations and the fact that its registers are only 24 bits long, the conversion of time from an integer to a real number cannot be any more precise than 24 bits. This conversion results in a **loss of precision causing a less accurate time calculation**.

The **British destroyer H.M.S. Sheffield** was sunk in the Falkland Islands war. According to one report, the ship's radar warning systems were programmed to **identify the Exocet missile as "friendly"** because the British arsenal includes the Exocet's homing device and allowed the missile to reach its target, namely the Sheffield.

Tipi di dato

Un moderno calcolatore è in grado di manipolare vari tipi di dato:

- Numeri
- Testo
- Immagini
- Suoni
- Video (sequenza di immagini più suono)
- Documenti multimediali (combinazione dei tipi precedenti)
- Ambienti VR (realtà virtuale)

Come vengono rappresentati al suo interno?

Prima del digitale - Numeri



Abaco cinese - 200 a.c.



Calcolatrice meccanica - 1974

Prima del digitale - Testi



Invicta (15 kg) - 1927-29



Lettera 32 (5,5 kg) - 1963

Prima del digitale - Immagini



Fotokor (URSS) - 1930



Nikon F3 - 1980

Prima del digitale - Suoni

K7 PHILIPS
i portatili che registrano
come "professional"



EL 3302 fa tutto con un tasto solo

Incisione, riascolto, avvolgimento e riavvolgimento rapidi: tutto con un tasto solo. EL 3302 ha il cuore fedele dei professionali Philips e fornisce registrazioni e riproduzioni di alto livello. Alimentazione a pile, strumento indicatore del livello di registrazione, microfono con telecomando, prese per cuffia e altoparlante supplementare. Un portatile della gamma K7 Philips.

PHILIPS



Prima del digitale - Video



BBC VERA - 1952

Prima del digitale - Video



NAGRA VPR-5 - 1983



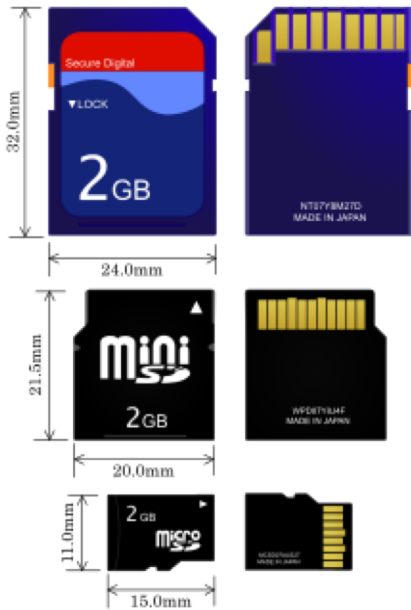
SONY BMC-500P - 1985

Supporti analogici



[Museo Nazionale Scienza e Tecnologia Leonardo da Vinci - Milano](#)

Supporti digitali



Analogico e digitale

- Un vinile al microscopio

<https://www.youtube.com/watch?v=GuCdsyCWmt8>

- Un CD al microscopio

<https://www.youtube.com/watch?v=RZUxemOE07Q&t=890s>

Rappresentazione dei **dati**

- I calcolatori riconoscono solo due stati, indicati tipicamente con 0/1 (anche on/off, vero/falso).
- Ogni singola cifra binaria (**binary digit**) si chiama **bit** e può assumere solo due valori:

0 = OFF (spento, falso)

1 = ON (acceso, vero)

QUALSIASI TIPO DI DATO VIENE
RAPPRESENTATO DA UNA SEQUENZA DI BIT
(anche molto lunga ...)

Alcuni esempi

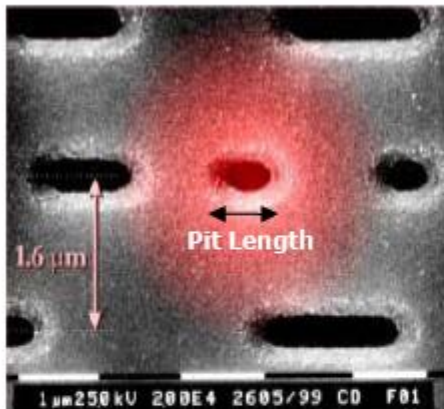
- Un singolo numero (es.: il nostro anno di nascita)
- Un breve testo (es.: un SMS)
- Un lungo testo (es.: un libro, un e-book)
- Un'immagine a colori di qualche milione di punti (Mpixel)
- Un brano musicale (p.es. in formato mp3)
- Un breve video (p.es. tratto da YouTube)
- Un intero film, magari in HD, con audio multicanale

TUTTI QUESTI DATI SONO RAPPRESENTATI DA SEQUENZE DI BIT (più o meno lunghe)

Solo 0 ed 1: 010001000100010010101010001000100001000100010010101010101 ...

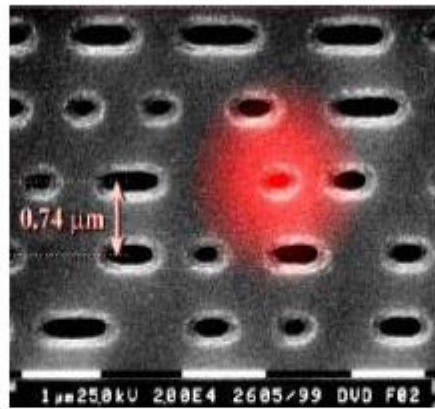
Supporti ottici (CD/DVD/BD)

CD 0.7GB



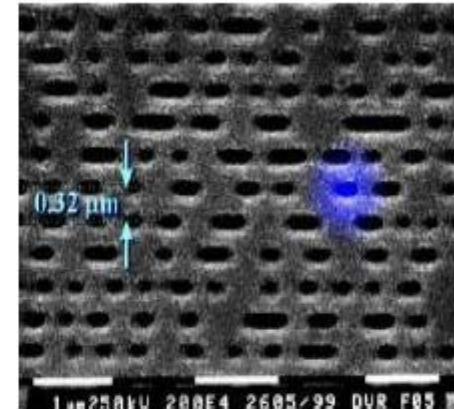
Track Pitch: 1,6 μm
Minimum Pit Length: 0,8 μm
Storage Density: 0,41Gb/in²

DVD 4.7GB



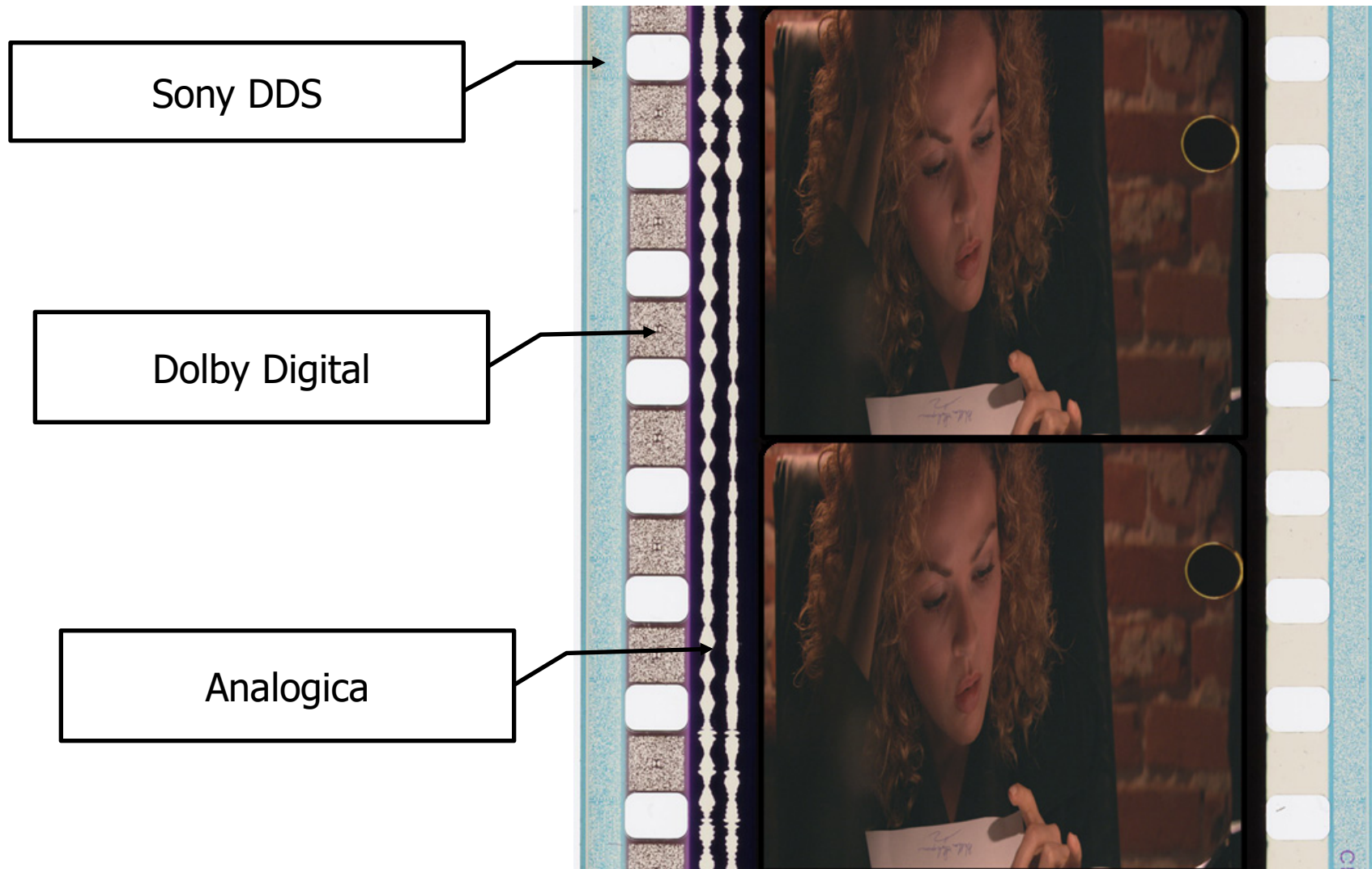
Track Pitch: 0,74μm
Minimum Pit Length: 0,4μm
Storage Density: 2,77Gb/in²

Blu-ray Disc 25GB

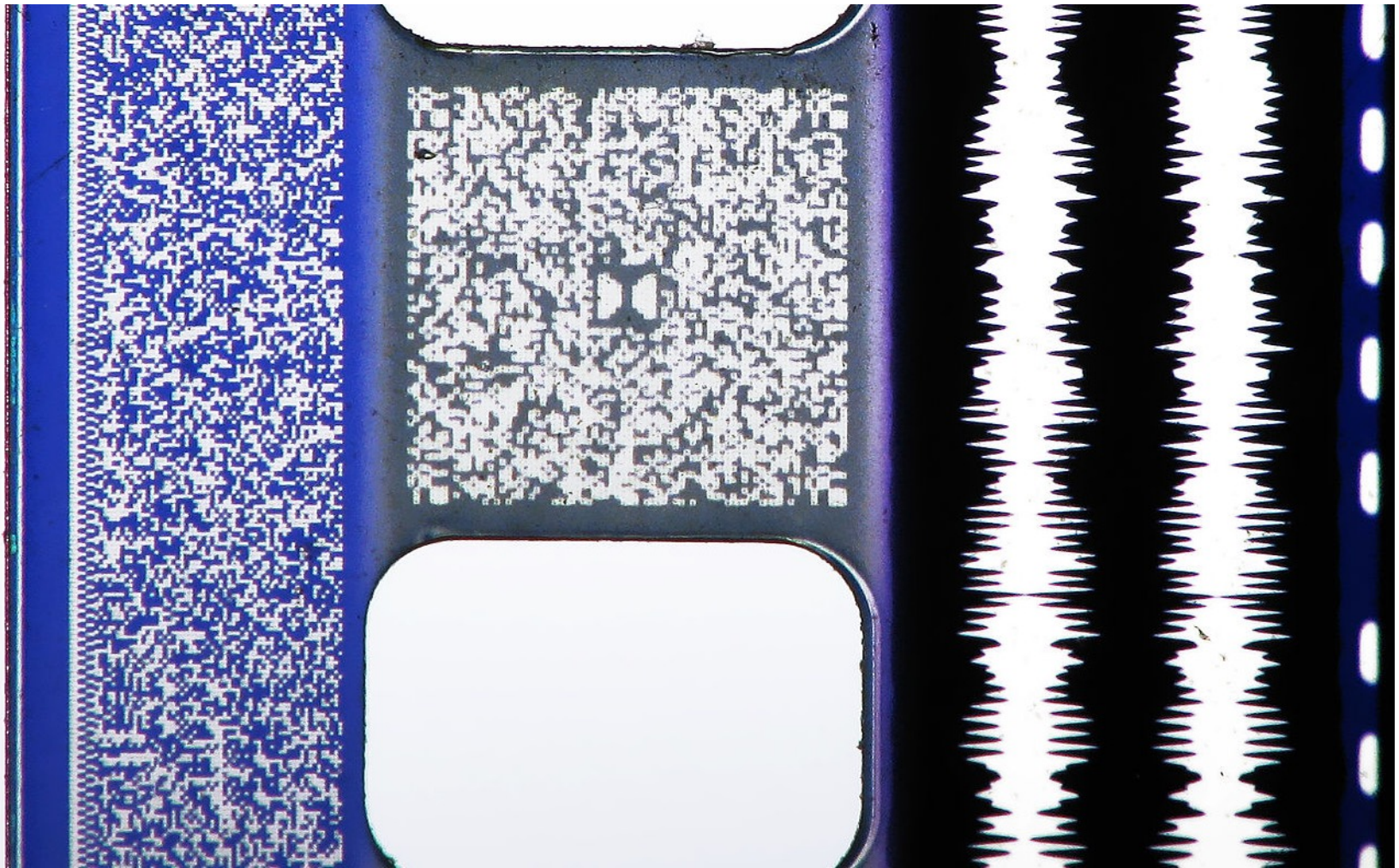


Track Pitch: 0,32μm
Minimum Pit Length: 0,15μm
Storage Density: 14,73Gb/in²

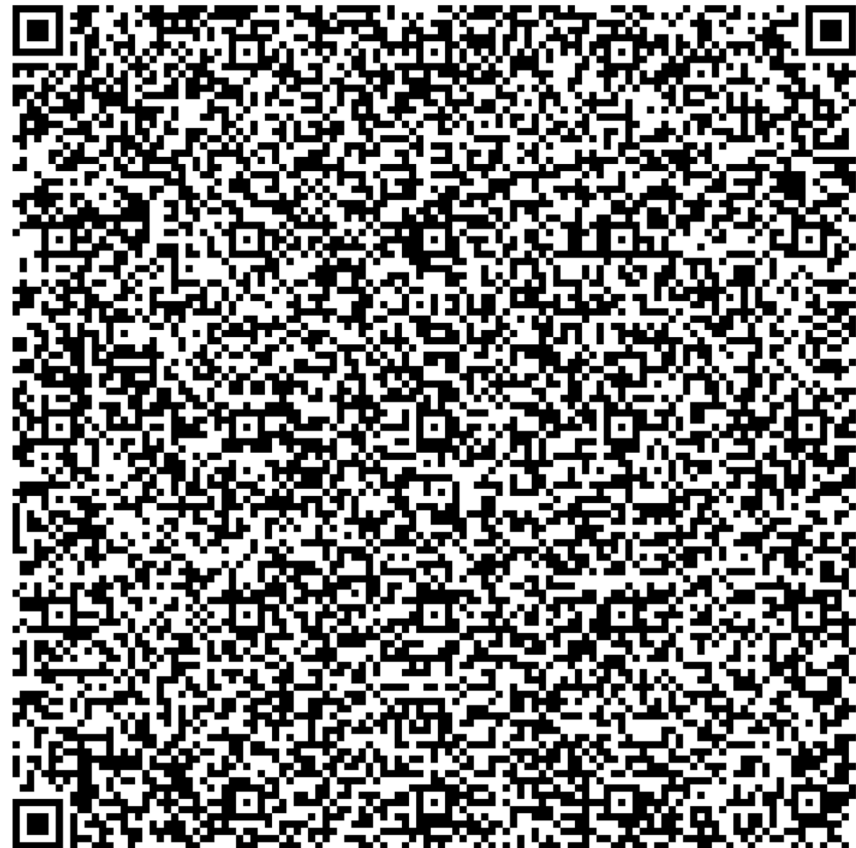
Colonna sonora cinematografica



Colonna sonora cinematografica



QR-Code



N.B.: QR è un acronimo che significa "Quick Response"

Numerazione decimale

Come siamo abituati a rappresentare i numeri?

10 simboli: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

| | | | |
|----------|----------|----------|----------|
| 1 | 4 | 9 | 2 |
| x1000 | x100 | x10 | x1 |
| 10^3 | 10^2 | 10^1 | 10^0 |

Il "peso" di ogni cifra è dato dalle potenze di 10

Si chiama **notazione posizionale** (in base 10)

(N.B.: In numeri romani si scriverebbe MCDXCII ...)

Quante cifre ci vogliono?

In **decimale**, ad es. con **2 cifre** si rappresentano tutti i numeri da 0 fino a 99 (**100** combinazioni)

Con 3 cifre: 1'000

Con 4 cifre: 10'000

Con 5 cifre: 100'000



In generale, in **base 10**: **N cifre** → **10^N** diversi valori

Numerazione ottale (Base 8)

| Base 10 | Base 8 |
|-----------|-----------|
| 0 | 0 |
| 1 | 1 |
| 2 | 2 |
| 3 | 3 |
| ... | ... |
| 7 | 7 |
| 8 | 10 |
| 9 | 11 |
| 10 | 12 |
| ... | ... |
| 15 | 17 |
| 16 | 20 |

Ho a disposizione solo 8 simboli, da 0 fino a 7 (l'8 e il 9 non esistono proprio!)

I "pesi" variano con le potenze di 8:

$$\begin{array}{cccc} 8^3 & 8^2 & 8^1 & 8^0 \\ (512) & (64) & (8) & (1) \end{array}$$

Numerazione binaria (Base 2)

| Base 10 | Base 8 | Base 2 |
|-----------|-----------|--------------|
| 0 | 0 | 0 |
| 1 | 1 | 1 |
| 2 | 2 | 10 |
| 3 | 3 | 11 |
| ... | ... | ... |
| 7 | 7 | 111 |
| 8 | 10 | 1000 |
| 9 | 11 | 1001 |
| 10 | 12 | 1010 |
| ... | ... | ... |
| 15 | 17 | 1111 |
| 16 | 20 | 10000 |

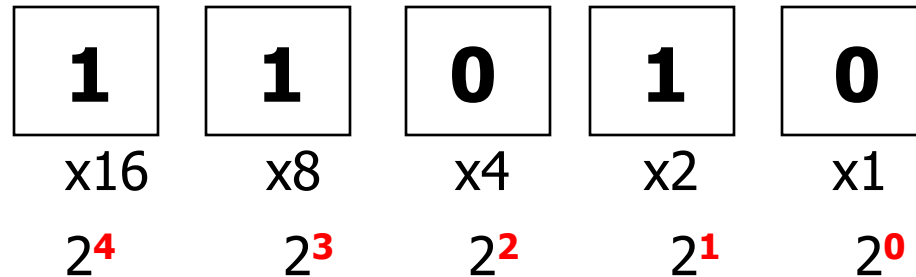
Ho a disposizione solo 2 simboli: 0 ed 1

I "pesi" variano con le potenze di 2:

$$\begin{array}{cccc} 2^3 & 2^2 & 2^1 & 2^0 \\ (8) & (4) & (2) & (1) \end{array}$$

Numerazione binaria

Esempio di numero binario



Il "peso" di ogni cifra è dato dalle **potenze di 2**
È sempre notazione posizionale, ma in **base 2**

Conversione binario → decimale

0 1 0 1 0 0 0 1

128 64 32 16 8 4 2 1

0 + 64 + 0 + 16 + 0 + 0 + 0 + 1

$$64 + 16 + 1 = 81$$

Conversione decimale \rightarrow binario

Come convertire 81_{10} in binario?

$$81 : 2 = 40 \text{ resto } \mathbf{1}$$

$$40 : 2 = 20 \text{ resto } \mathbf{0}$$

$$20 : 2 = 10 \text{ resto } \mathbf{0}$$

$$10 : 2 = 5 \text{ resto } \mathbf{0}$$

$$5 : 2 = 2 \text{ resto } \mathbf{1}$$

$$2 : 2 = 1 \text{ resto } \mathbf{0}$$

$$1 : 2 = 0 \text{ resto } \mathbf{1}$$



Risultato: $81_{10} = \mathbf{1010001}_2$

Conversione decimale → binario

Procedimento:

- 1) Divido per 2 e annoto il resto
- 2) Ripeto (1) fino ad arrivare a ZERO.
- 3) Leggo i resti **in ordine inverso**

$$81 : 2 = 40 \text{ resto } \mathbf{1}$$

$$40 : 2 = 20 \text{ resto } \mathbf{0}$$

$$20 : 2 = 10 \text{ resto } \mathbf{0}$$

$$10 : 2 = 5 \text{ resto } \mathbf{0}$$

$$5 : 2 = 2 \text{ resto } \mathbf{1}$$

$$2 : 2 = 1 \text{ resto } \mathbf{0}$$

$$1 : 2 = 0 \text{ resto } \mathbf{1}$$



Quante cifre ci vogliono?

In **binario**, con 1 cifra (bit) si possono rappresentare solo due numeri: 0 ed 1

Con 2 cifre: 00, 01, 10, 11 (4 combinazioni)

Con 3 cifre: da 000 a 111 (8 combinazioni)

In generale, in **base 2**:

N cifre (bit) → 2^N diversi valori

Riepilogo decimale/binario

Decimale:

Base: **10**

10 simboli: 0, 1, 2, 3, ... ,9

Pesi delle cifre: **10^0 10^1 10^2 10^3 ...**

Con N cifre rappresento **10^N** diversi valori

Binario:

Base: **2**

2 simboli: 0, 1

Pesi delle cifre: **2^0 2^1 2^2 2^3 ...**

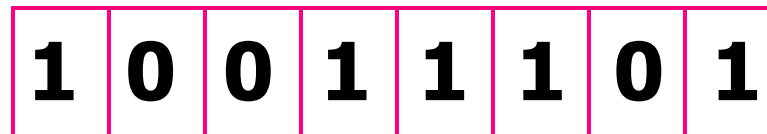
Con N cifre (bit) rappresento **2^N** diversi valori

[Potenza di due \(Wikipedia\)](#)

Il Byte

Con due sole cifre a disposizione, i numeri diventano velocemente lunghi → si usano **gruppi di 8 bit** detti

BYTE



Un Byte può assumere quindi **2^8** (256) valori diversi: da 00000000_2 (0_{10}) fino a " 11111111_2 " (255_{10})

(N.B.: In inglese "bite" significa boccone, morso)

Byte come unità di misura

Il byte è anche **l'unità di misura** della capacità di memoria.

| | |
|--------------------|-----------------|
| Byte | 8 bit |
| Word | 16 bit (2 byte) |
| Double Word | 32 bit (4 byte) |
| Quad Word | 64 bit (8 byte) |

| | | |
|----------------------|-------------|------------------|
| kiloByte (kB) | 1' 024 byte | (2^{10} byte) |
| MegaByte (MB) | 1' 024 kB | (2^{20} byte) |
| GigaByte (GB) | 1' 024 MB | (2^{30} byte) |
| TeraByte (TB) | 1' 024 GB | (2^{40} byte) |

Capacità tipiche

Capacità tipiche di alcuni tipi di memoria:

RAM

Alcuni GB

USB key

Da decine di GB fino a 1 TB

SD Card

Da decine di GB fino a 512 GB

FLOPPY DISK

Da 180 kB a 1.44 MB

HARD DISK

Centinaia di GB - Alcuni TB

CD

650 MB (74' audio) - 700 MB (80')

DVD

4.7 - 17 GB (SD video)

BD

25 - 50 GB (HD video)

(in sviluppo fino a 200 GB per UHD video)

Se i Byte fossero ...

- **grammi**, un TeraByte equivarrebbe a un milione di tonnellate!
- **millimetri**, un TeraByte equivarrebbe a un milione di km!
- **secondi**, un TeraByte equivarrebbe a 31'688 anni!

| 10^0 | 10^3 | 10^6 | 10^9 | 10^{12} |
|-------------|-----------|-----------|-----------|--------------|
| Byte | kB | MB | GB | TB |
| g | kg | t | 1'000 t | 1'000'000 t |
| mm | m | km | 1'000 km | 1'000'000 km |
| s | ~17 min | ~278 ore | ~32 anni | ~32'000 anni |

L'influenza del Binario

- Numero di colori: 256 ($=2^8$)
- Indirizzi IP: 32 bit (4 byte) rappresentati come ad es.: 192.8.234.125
- Dimensione delle memorie (128 MB, 256 MB, 512 MB, 1GB, 2GB, ecc)
- Dimensione blocchi hard disk (512 byte, 1024 byte, 2048 byte)
- Dimensione canali di comunicazione (es: Bus) (8 fili, 16 fili, 32 fili, 64 fili)

Vecchio proverbio inglese ...

*"There are **10** types of people in the world: those who understand binary and those who don't"*

Da Binario a Esadecimale

- Base 16 invece di Base 2
- 16 simboli invece di 2
- 0, 1, 2, 3, 4, ..., 9, A, B, C, D, E, F
- Ogni cifra esadecimale occupa 4 bit
- 4 bit = 1 nibble
- 2 nibble = 1 byte
- Permette di rappresentare grandi numeri con meno cifre
- Utilizzato principalmente nella programmazione

Da Binario a Esadecimale

The binary, decimal, and hexadecimal conversion table illustrates the three numbering systems.

| <u>Binary</u> | <u>Decimal</u> | <u>Hexadecimal</u> |
|---------------|----------------|--------------------|
| 0 0 0 0 | 0 | 0 |
| 0 0 0 1 | 1 | 1 |
| 0 0 1 0 | 2 | 2 |
| 0 0 1 1 | 3 | 3 |
| 0 1 0 0 | 4 | 4 |
| 0 1 0 1 | 5 | 5 |
| 0 1 1 0 | 6 | 6 |
| 0 1 1 1 | 7 | 7 |
| 1 0 0 0 | 8 | 8 |
| 1 0 0 1 | 9 | 9 |
| 1 0 1 0 | 10 | A |
| 1 0 1 1 | 11 | B |
| 1 1 0 0 | 12 | C |
| 1 1 0 1 | 13 | D |
| 1 1 1 0 | 14 | E |
| 1 1 1 1 | 15 | F |

Rappresentazione di **testi**

- Ogni **carattere** occupa **1 byte**
- Max $2^8=256$ diversi caratteri
- Tabella ASCII (si pronuncia "ASKI"): ad ogni carattere corrisponde un codice numerico:
 - 0100 0001 (65_{10}) = A
 - 0100 0010 (66_{10}) = B
- Maiuscole e minuscole hanno valori distinti
- Contiene anche cifre, simboli e caratteri speciali

Tabella ASCII (da 0 a 127)

| ASCII value | Character | Control character | ASCII value | Character | ASCII value | Character | ASCII value | Character |
|-------------|-------------------|-------------------|-------------|-----------|-------------|-----------|-------------|-----------|
| 000 | (null) | NUL | 032 | (space) | 064 | @ | 096 | |
| 001 | ☺ | SOH | 033 | ! | 065 | A | 097 | a |
| 002 | ☻ | STX | 034 | " | 066 | B | 098 | b |
| 003 | ♥ | ETX | 035 | # | 067 | C | 099 | c |
| 004 | ♦ | EOT | 036 | \$ | 068 | D | 100 | d |
| 005 | ♣ | ENQ | 037 | % | 069 | E | 101 | e |
| 006 | ♠ | ACK | 038 | & | 070 | F | 102 | f |
| 007 | (beep) | BEL | 039 | ' | 071 | G | 103 | g |
| 008 | ■ | BS | 040 | (| 072 | H | 104 | h |
| 009 | (tab) | HT | 041 |) | 073 | I | 105 | i |
| 010 | (line feed) | LF | 042 | * | 074 | J | 106 | j |
| 011 | (home) | VT | 043 | + | 075 | K | 107 | k |
| 012 | (form feed) | FF | 044 | , | 076 | L | 108 | l |
| 013 | (carriage return) | CR | 045 | - | 077 | M | 109 | m |
| 014 | ♪ | SO | 046 | . | 078 | N | 110 | n |
| 015 | ☼ | SI | 047 | / | 079 | O | 111 | o |
| 016 | ▴ | DLE | 048 | 0 | 080 | P | 112 | p |
| 017 | ▾ | DC1 | 049 | 1 | 081 | Q | 113 | q |
| 018 | ↕ | DC2 | 050 | 2 | 082 | R | 114 | r |
| 019 | !! | DC3 | 051 | 3 | 083 | S | 115 | s |
| 020 | π | DC4 | 052 | 4 | 084 | T | 116 | t |
| 021 | § | NAK | 053 | 5 | 085 | U | 117 | u |
| 022 | ▬ | SYN | 054 | 6 | 086 | V | 118 | v |
| 023 | ↕ | ETB | 055 | 7 | 087 | W | 119 | w |
| 024 | ↑ | CAN | 056 | 8 | 088 | X | 120 | x |
| 025 | ↓ | EM | 057 | 9 | 089 | Y | 121 | y |
| 026 | → | SUB | 058 | : | 090 | Z | 122 | z |
| 027 | ← | ESC | 059 | ; | 091 | [| 123 | { |
| 028 | (cursor right) | FS | 060 | < | 092 | \ | 124 | |
| 029 | (cursor left) | GS | 061 | = | 093 |] | 125 | } |
| 030 | (cursor up) | RS | 062 | > | 094 | ^ | 126 | ~ |
| 031 | (cursor down) | US | 063 | ? | 095 | - | 127 | ☑ |

Copyright 1998, JimPrice.Com Copyright 1982, Loading Edge Computer Products, Inc.

Testi: un esempio

Consideriamo un file di testo (oppure un SMS, od un messaggio di WhatsApp) che contenga solo l'esclamazione "BUON GIORNO!"

Se esaminiamo il suo contenuto byte a byte, troviamo che il file conterrà la sequenza di numeri:

| | | | | | | | | | | | |
|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| 01000010 | 01010101 | 01001111 | 01001110 | 00100000 | ecc ... | | | | | | |
| 66 | 85 | 79 | 78 | 32 | 71 | 73 | 79 | 82 | 78 | 79 | 33 |
| B | U | O | N | | G | I | O | R | N | O | ! |

Ogni numero (sia binario, sia decimale) corrisponde al codice ASCII del carattere sottostante (verificare!).

N.B.: Anche lo spazio è un carattere (32)

E questi?

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 良 | 農 | 業 | 商 | 議 | 選 | 員 | 記 | 運 | 轉 | 者 | 事 |
| 味 | 試 | 次 | 難 | 形 | 適 | 當 | 同 | 違 | 正 | 惡 | 点 |
| 念 | 殘 | 落 | 格 | 果 | 受 | 說 | 接 | 面 | 驗 | 合 | 指 |
| 約 | 決 | 旅 | 消 | 流 | 深 | 案 | 投 | 洗 | 打 | 扌 | 折 |
| 備 | 準 | 到 | 癸 | 線 | 泊 | 特 | 絡 | 連 | 急 | 談 | 相 |
| 注 | 押 | 意 | 故 | 路 | 信 | 局 | 機 | 関 | 割 | 交 | 引 |
| 用 | 器 | 願 | 知 | 求 | 台 | 具 | 取 | 自 | 窓 | 由 | 營 |
| 期 | 產 | 個 | 倆 | 品 | 資 | 銀 | 誌 | 雜 | 辭 | 服 | 紙 |
| 心 | 報 | 告 | 々 | | | | | | | | |

Unicode

- Il codice ASCII non è in grado di rappresentare tutte le lingue del pianeta.
- 256 simboli non bastano per lingue come il cinese, il giapponese e tante altre lingue orientali, basate su migliaia di ideogrammi e non su un alfabeto di poche decine di simboli.
- Per rappresentare i testi di queste lingue si utilizza il sistema [Unicode](#) in cui ogni simbolo è rappresentato da 1, 2 o 4 byte (UTF-7, UTF-8, UTF-16, UTF-32)

Rappresentazione di **immagini**

- Un'immagine digitale è costituita da una fitta griglia di punti, detti **pixel** (= **picture element**)
- Ad esempio, in una TV digitale **Full-HD** lo schermo è composto da 1920 x 1080 pixel per un totale di 2'073'600 pixel (**~2 Mpixel**)
- Ogni pixel può assumere un colore diverso, dando origine ad una infinità di immagini possibili

Risoluzione

- Si dice **risoluzione** di un'immagine il numero di pixel (in orizzontale e verticale) da cui essa è costituita.
- Maggiore è la risoluzione di un'immagine e maggiore è la quantità di dettagli che possiamo individuare.
- Anche gli schermi (monitor) sono caratterizzati dalla loro risoluzione. Es.:
 - VGA 640 x 480 ~0.3 Mp
 - XGA 1024 x 768 ~0.8 Mp
 - HD 1280 x 720 ~0.9 Mp
 - Full HD 1920 x 1080 (FHD, 2K) ~2 Mp
 - Ultra HD 3840 x 2160 (UHD, 4K) ~8 Mp
 - Ultra HD 7680 x 4320 (UHD, 8K) ~33 MpEcc ...

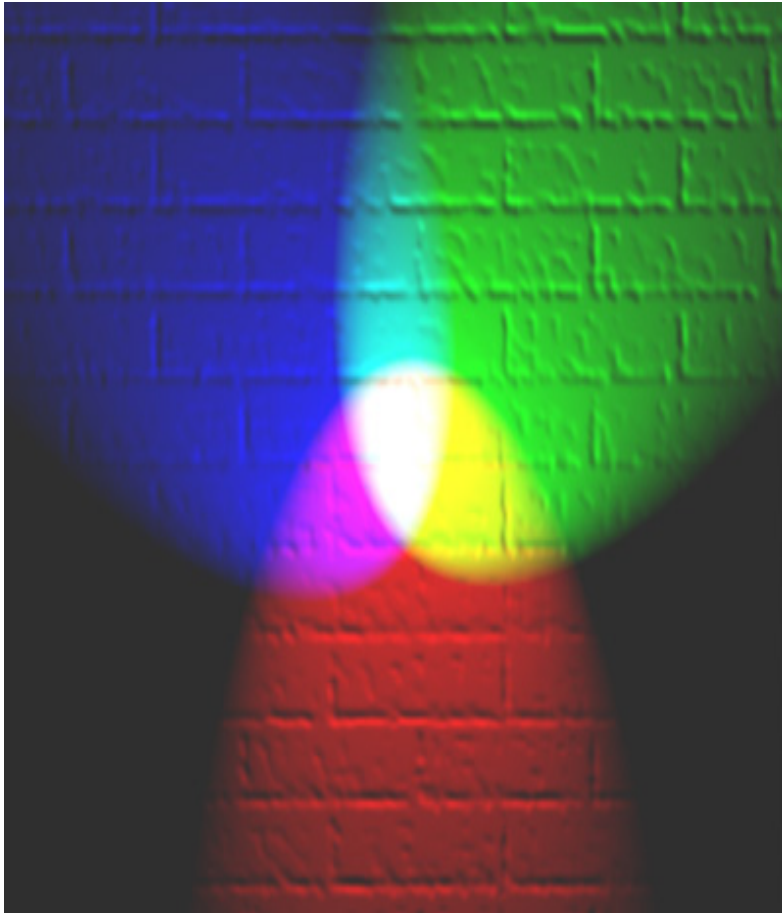
Densità dei pixel

- La **densità** dei pixel si misura in pixel per pollice, in inglese ppi (pixel per inch)
- Due schermi con **uguale risoluzione** ma **diversa dimensione** avranno quindi diversi valori di ppi.
- Alcuni esempi:
 - TV Full HD da 40": 55 ppi
 - TV Full HD da 20": 110 ppi
 - TV Ultra HD da 55": 80 ppi
 - Smartphone con schermo FHD da 5": 440 ppi

Densità dei pixel

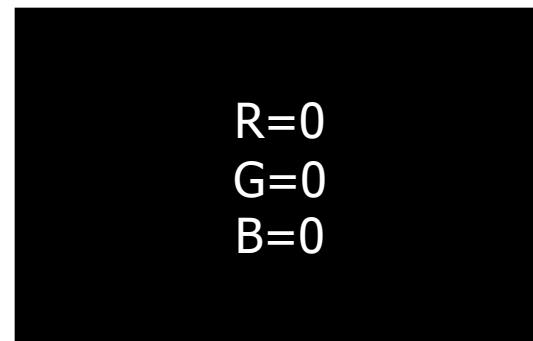
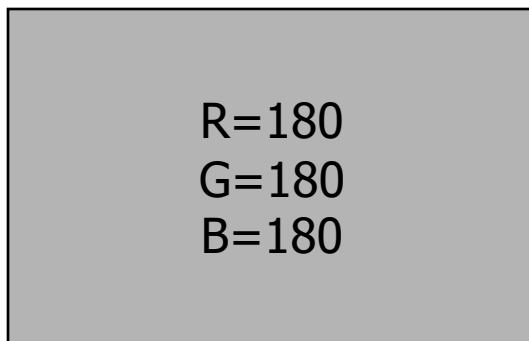
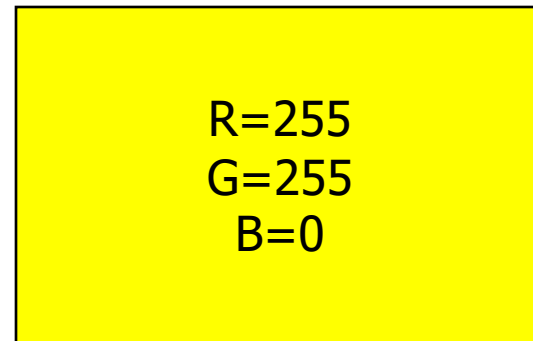
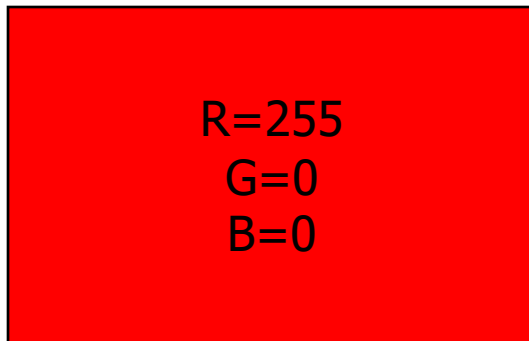
- La **densità** dei pixel si deve specificare anche ogni volta che si digitalizza un'immagine con uno scanner o che la si deve stampare.
- In questo caso, la densità si misura in **punti per pollice**, in inglese **dpi** (dots per inch)
- I valori vanno da **75 dpi** per scansioni molto grossolane fino a **2400** o **4800 dpi** per scansioni estremamente dettagliate.
- Ovviamente, al crescere del valore scelto, aumenta anche la **dimensione** del file generato ed il **tempo** necessario per la scansione.

Red Green Blue - RGB



- Ogni colore si può ottenere miscelando i colori primari **Rosso**, **Verde** e **Blu**.
- Un dato colore è quindi individuato da tre numeri che indicano rispettivamente la quantità di **Rosso**, di **Verde** e di **Blu** che lo compongono.
- Tipicamente ognuna di queste tre componenti può variare da un min di 0 ad un max di 255.

RGB - Esempi



Quanti colori sono possibili?

- Tipicamente, un colore è rappresentato da 3 gruppi di 8 bit (3 byte) per un totale di 24 bit.

01010101

11001100

00011101

- Usando 24 bit in questo modo, si possono ottenere 2^{24} (circa **16 milioni**) di colori diversi.
- In questo caso, si dice che l'immagine ha una **profondità di colore** di 24 bit
- L'unità di misura della profondità di colore è detta **bpp** (bit per pixel) ed in alcuni casi può spingersi fino a 48 bpp ed oltre

Memoria video

Si consideri uno schermo da

$$\mathbf{1920 \times 1080 = 2'073'600 \text{ pixel}}$$

Se il colore di ogni pixel è rappresentato da

3 Byte (24 bit)

Un'immagine che riempia esattamente lo schermo occuperà

$$3 \times 2'073'600 = 6'220'800 \text{ byte, ossia}$$

~ 6 MB

Sintesi additiva e sottrattiva

- La sintesi RGB è utilizzata nei **display** ed in **proiezione** ed è denominata **sintesi additiva** in quanto le singole componenti luminose si **sommano** per formare il colore risultante.
- In **stampa**, invece, viene utilizzata la **sintesi sottrattiva YMCK** (Yellow, Magenta, Cyan, Black) in quanto i pigmenti presenti nell' inchiostro **sottraggono** parte delle componenti della luce bianca che illumina l' immagine lasciando solo il colore desiderato.

Sintesi additiva e sottrattiva



sintesi additiva



sintesi sottrattiva

Compressione di immagine

Si consideri un apparecchio fotografico (od uno smartphone) con risoluzione $4000 \times 3000 = \mathbf{12 \text{ Mp}}$

Se il colore di ogni pixel è rappresentato da **3 Byte** (ossia con profondità di colore 24 bit/pixel), ogni immagine occuperà

$$3\text{B/p} \times 12\text{Mp} = \mathbf{36\text{MB}}$$

Poiché però le immagini sono memorizzate in **formato compresso JPEG**, il file occuperà solamente circa **3.6 MB**, ossia circa 1/10 (il 10%) dell'originale.

Ciò significa che il 90% restante delle informazioni è stato eliminato!

Compressione di immagine

- Non si può eliminare il 90% dei dati di un'immagine senza conseguenze!
- I file in formato JPEG hanno la **stessa risoluzione** e **stessa profondità di colore** dell'immagine originale, ma **qualità inferiore**
- In altre parole, la compressione comporta una **perdita di informazione** irrecuperabile (ma entro certi limiti trascurabile, in quanto elimina dettagli che il sistema visivo umano considera secondari)
- Per questo motivo, i formati compressi come il JPEG sono detti "**lossy**"

N.B.: Sono possibili anche compressioni (meno drastiche) senza perdita di informazione, dette per questo motivo "**lossless**"

Compressione di immagine



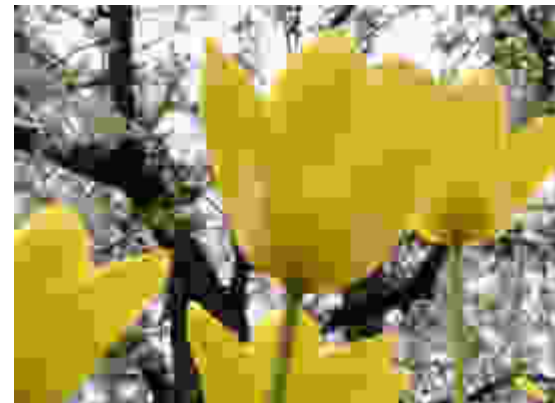
JPEG qualità 100% - 87,7 KB



JPEG qualità 90% - 30,2 KB



JPEG qualità 50% - 6,7 KB



JPEG qualità 10% - 3,2 KB

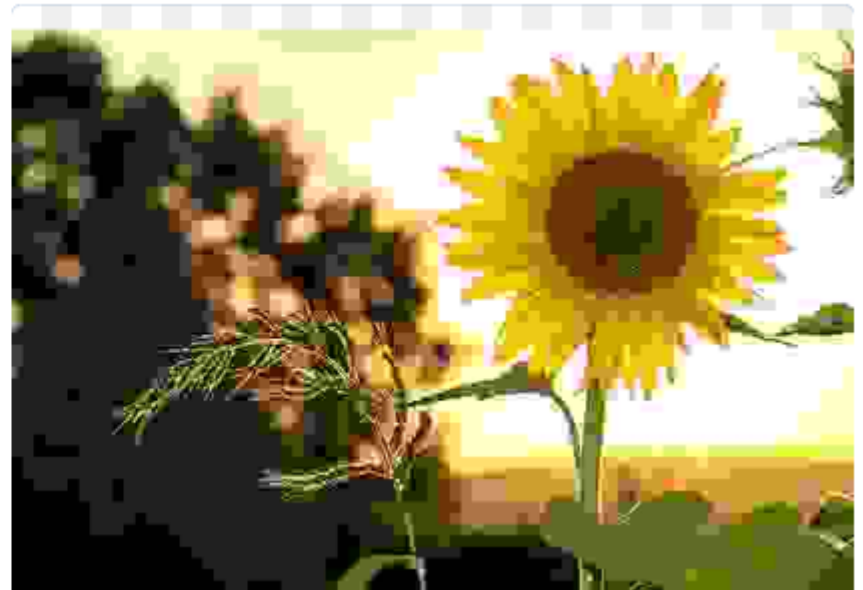
Compressione di immagine

original jpg



Qualità 100%

new quality jpg

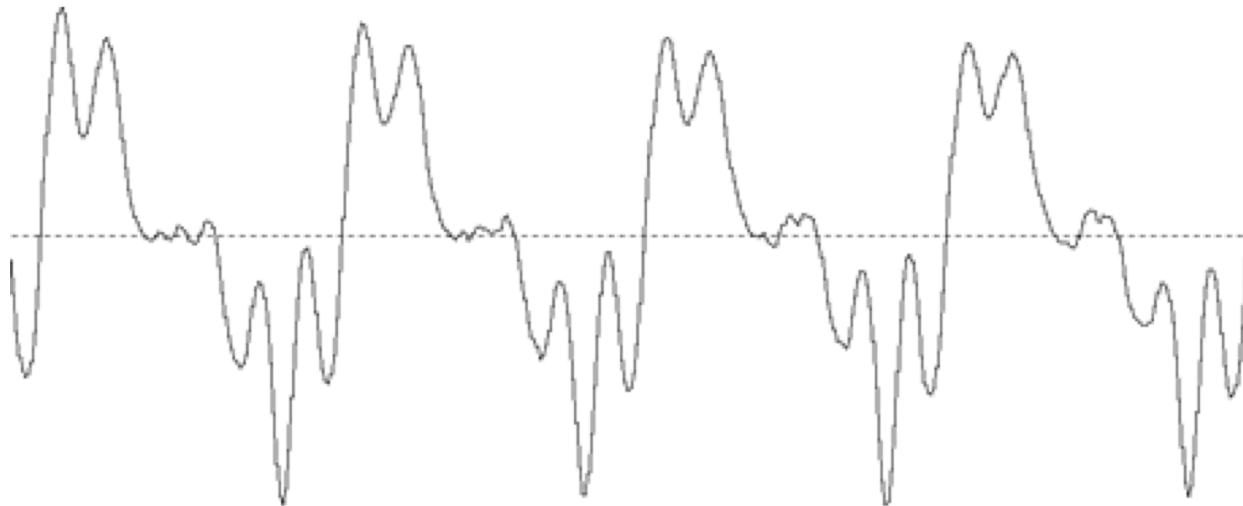


Qualità 0%

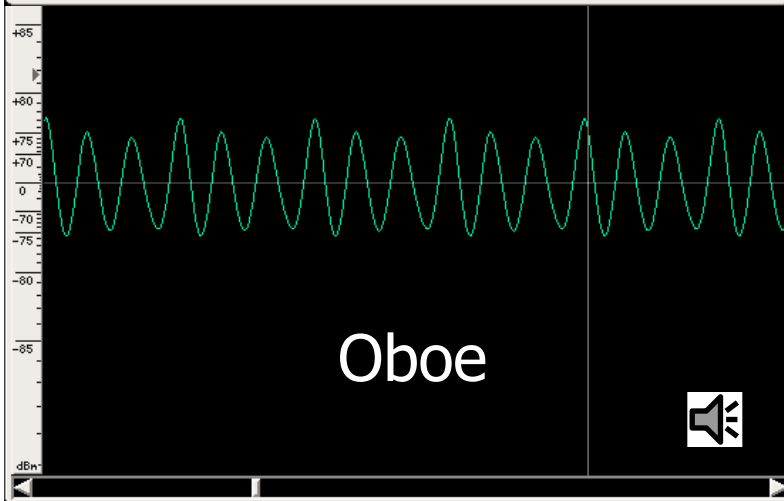
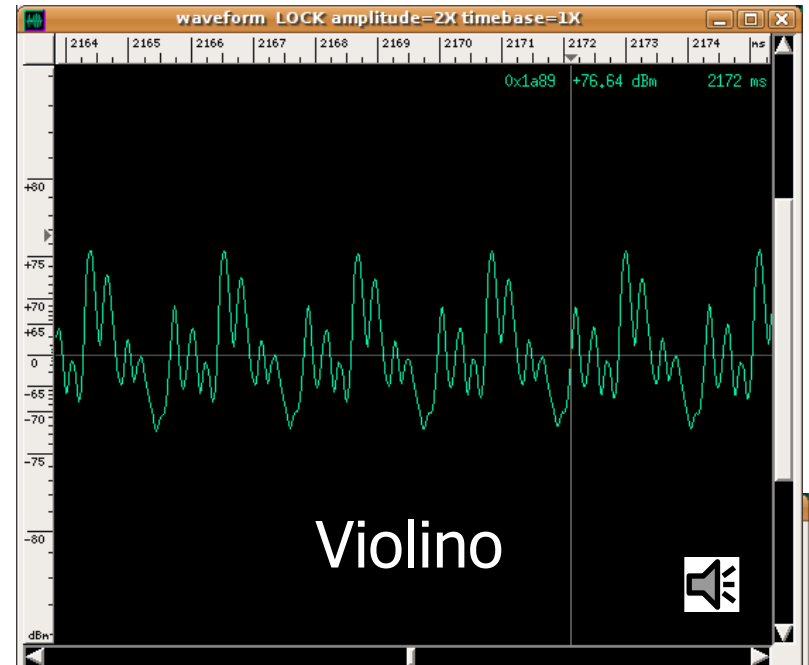
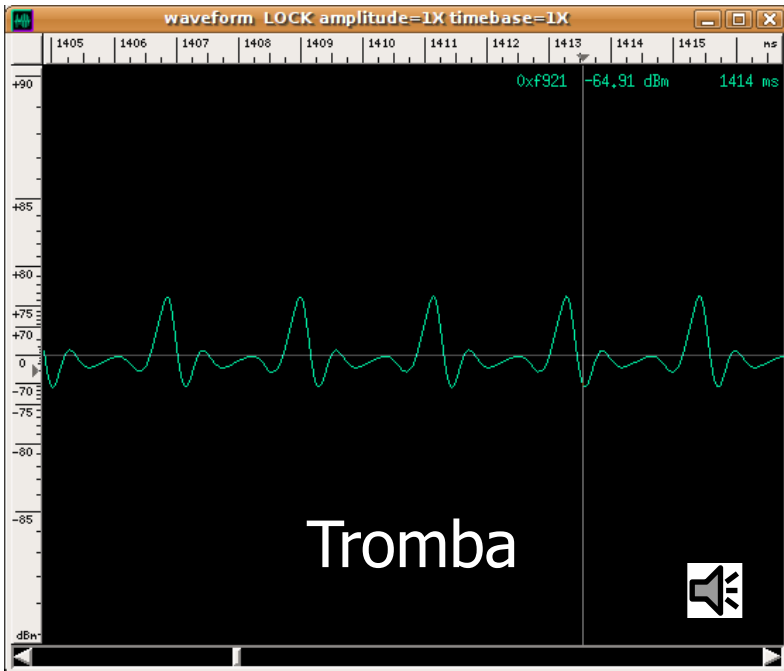
Rappresentazione di **suoni**

Il suono è una vibrazione che si propaga in un mezzo elastico (p.es. l'aria)

Lo si può rappresentare come un'onda che descrive l'andamento della pressione sonora nel tempo

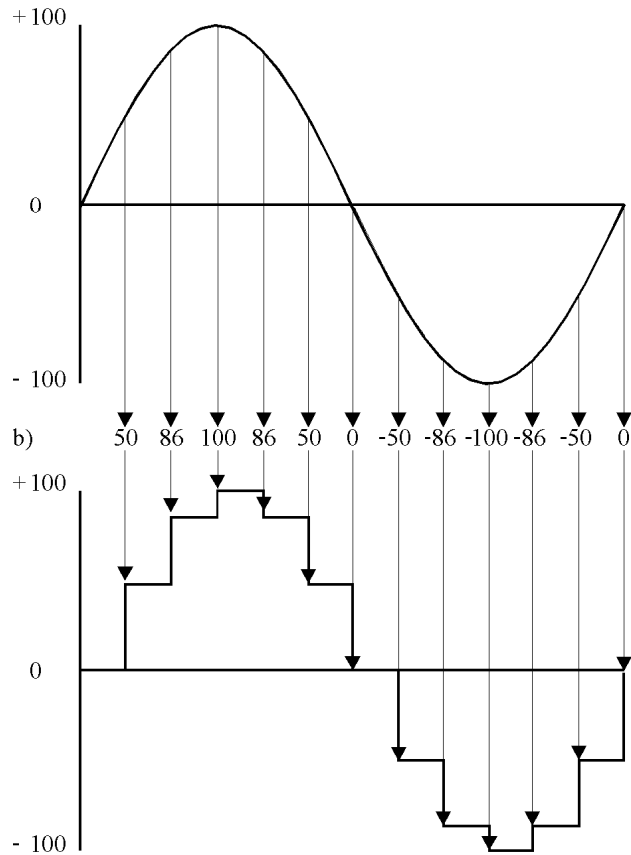


Esempi di forme d'onda



Campionamento del suono

a) Theoretical waveshape:

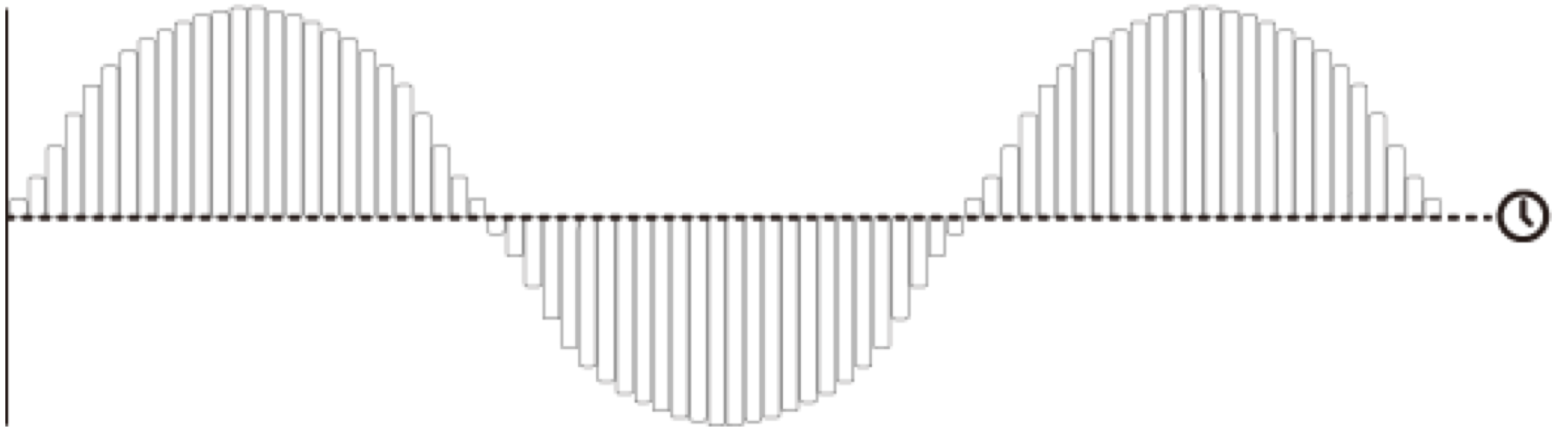


La digitalizzazione del suono avviene come segue:

- L'onda sonora viene **campionata** ad intervalli regolari, ogni campione è un numero
- Il valore di ogni campione viene memorizzato
- La sequenza dei valori registrati permette di ricostruire la forma d'onda e quindi il suono originario

Frequenza di campionamento

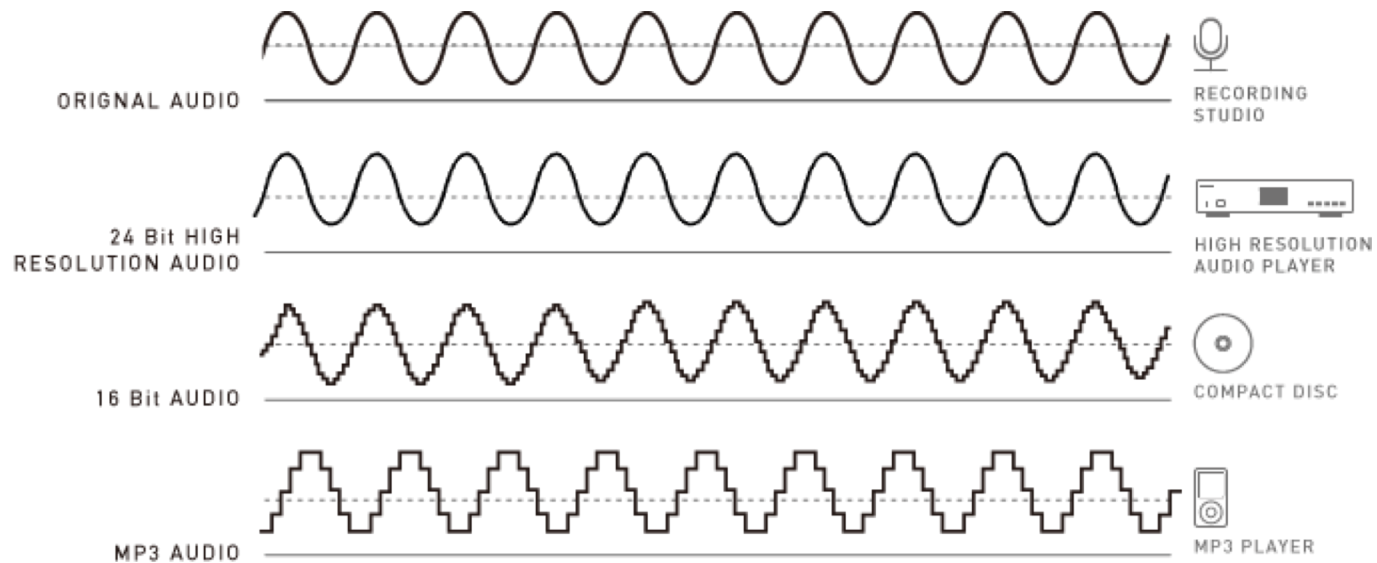
LA "FREQUENZA DI CAMPIONAMENTO" È IL NUMERO DI CAMPIONAMENTI PER SECONDO



Definita come il **numero di campionamenti per secondo**, si misura in Hertz (Hz)

Maggiore è la frequenza di campionamento, più la digitalizzazione produrrà un suono fedele all'originale

Profondità (in bit)



Definita come il **numero di bit per campione**, si misura in bit.

Maggiore è la profondità, più la digitalizzazione produrrà un suono fedele all'originale

Esempio - Il CD Audio

- Il suono registrato su un CD Audio è campionato a 44.1 kHz, cioè 44'100 volte al secondo
- Ogni campione è grande 16 bit (2 Byte)
- Due valori: Un campione per il canale sinistro ed uno per il destro (stereofonia)
- Ogni secondo di musica occupa quindi ...

$$2 \times 2 \times 44'100 = 176'400 \text{ Byte} \quad (\sim 170 \text{ kB/s})$$

$$(\sim 1.4 \text{ Mb/s})$$

- Un CD audio da 74 minuti quindi contiene ...

$$74 \times 60 \times 170 = 737'000 \text{ kB} \quad (\sim 720 \text{ MB})$$

HRA High Resolution Audio

Utilizzato in campo professionale (p.es. negli studi di registrazione).

Fornisce una qualità audio superiore al CD.

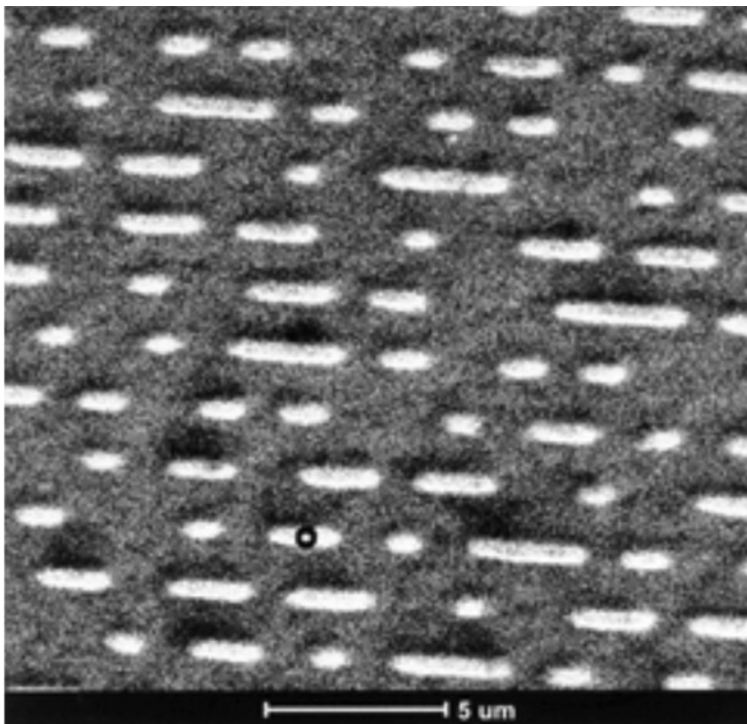
Caratterizzato da:

- Frequenze di campionamento > 44.1 kHz
(Valori tipici 96 kHz, 192 kHz)
- Profondità del singolo campione > 16 bit
(Valori tipici 24 bit, 48 bit)

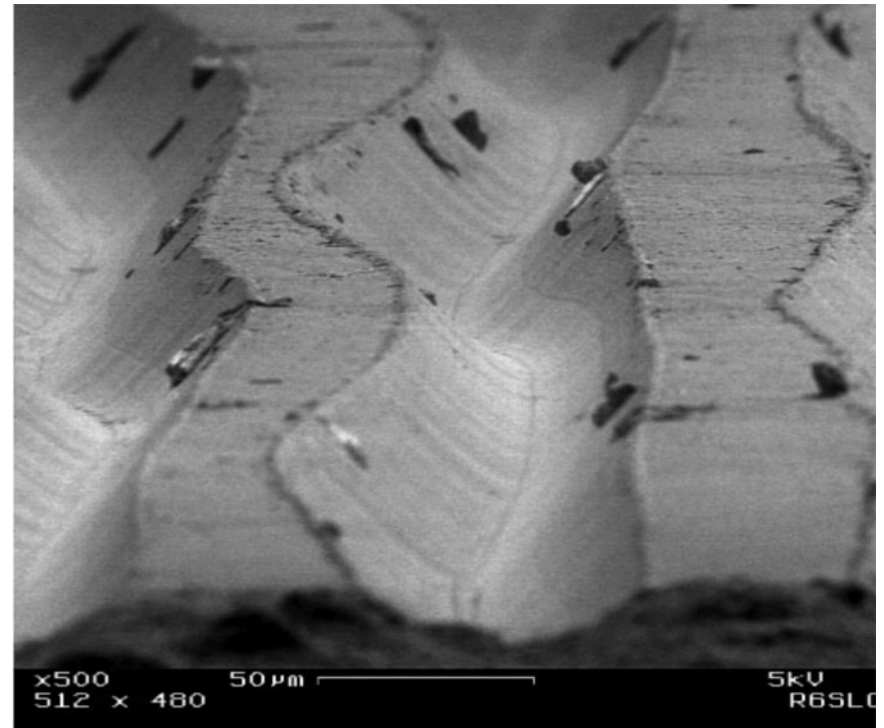
[Audio ad alta risoluzione \(Technics\)](#)

CD ed LP a confronto

Immagini riprese al microscopio elettronico



Superficie di un CD



Superficie di un LP

(10 volte meno ingrandito!)

Rappresentazione di **video**

Un video è costituito da:

- Una serie di **immagini** (tipicamente 25 al secondo)
- Una **colonna sonora**, costituita da 1 (mono), 2 (stereo) o più (5.1 surround ecc) canali audio

I formati video permettono di riprodurre **immagini** e **suono** mantenendoli in sincronia

La **risoluzione** dei video è quella già definita per le immagini, riferita ad un singolo fotogramma

Il numero di **fotogrammi al secondo** è rappresentato dal parametro **fps** (frames per second)

Esempi di risoluzioni video

- TV tradizionale **SD:** 720 x 576
- TV alta definizione **HD:** 1280 x 720 (circa 0.9 Mp)
- TV Full-HD **FHD:** 1920 x 1080 ("2K", circa 2 Mp)
- TV Ultra-HD1 **UHD-1:** 3840 x 2160 ("4K", circa 8 Mp)
- TV Ultra-HD2 **UHD-2:** 7680 x 4320 ("8K", circa 33 Mp)

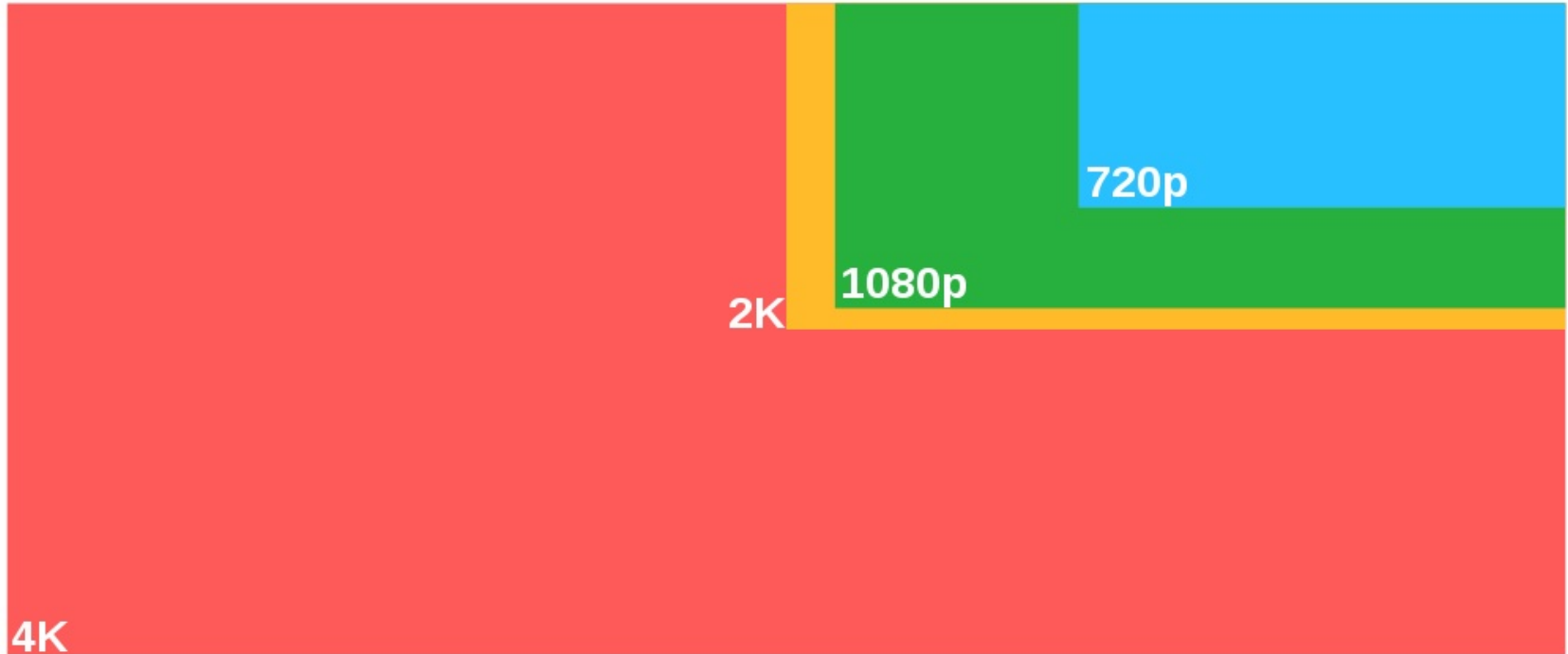
- **Cinema:**
 - 2K:** 2048 x 1080
 - 4K:** 4096 x 2160

- **Youtube:** il numero (240, 360, 480, 720, ecc) che identifica la risoluzione rappresenta il numero di righe dell'immagine (p.es.: 1080 = Full HD)

Esempi di risoluzioni video

Common Digital Cinema Formats

relative pixel dimension comparison at 2.39:1 aspect ratio
(1080p and 720p formats letterboxed)



approximately 1/8 of actual pixel dimensions

Frame rate più comuni

- **Cinema** (analogico) **24** fps
- **PAL TV** (analogico) **25** fps
- **NTSC TV** (analogico) **30** fps
- **HDTV** (digitale) **24, 25, 30, 50, 60, 72, 120** fps

Anche i comuni smartphone e videocamere digitali possono effettuare riprese a valori di qualche centinaio di fps per realizzare effetti di rallentamento (slow-motion)

Inoltre, vi sono in commercio speciali [cineprese digitali ad alta velocità](#) che possono effettuare riprese ad altissimi valori di fps (decine di migliaia!) per realizzare effetti di rallentamento ([slow-motion](#))

Compressione video (1)

Consideriamo un video Full HD come quello che visualizziamo su un qualsiasi canale HD del digitale terrestre. Esso ha le seguenti caratteristiche:

- Risoluzione: 1920 x 1080 pixel (circa **2 Mp**)
- Profondità di colore: 24 bit/pixel (**3B/p**)
- Frame rate: **25 fps**

Questo determinerebbe un flusso dati di

$$\mathbf{2\ Mp \times 3B \times 25\ fps = 150\ MB/s \text{ (ossia } \mathbf{1.2\ Gb/s})}$$

In realtà, un canale televisivo Full HD ha un bit rate tipico di circa **20 Mb/s** ossia circa 1/60 di quanto abbiamo appena calcolato!

Com'è possibile?

Compressione video (2)

Per evitare di dover memorizzare/trasmettere enormi quantità di dati, i formati video prevedono sofisticati meccanismi di **compressione**.

In estrema sintesi, i meccanismi di compressione video combinano due tecniche:

- **Intraframe:** ogni fotogramma viene compresso individualmente (come una qualsiasi immagine fissa)
- **Interframe:** vengono memorizzate solo le differenze fra fotogrammi adiacenti (spesso minime)

Queste due tecniche, combinate fra loro, permettono di raggiungere fattori di compressione tipici dell'ordine di 30-50:1 (mpeg-2) o addirittura 60-100:1 (mpeg-4)

Esempi di formati video

- **.avi** (**AVI**, Audio Video Interleaved, Microsoft)
- **.mov** (**QTFF**, Quick Time File Format, Apple)
- **.mpg, .mp2, .mp4** (**MPEG**, Moving Pictures Expert Group)
- **.mkv** (**Matroska**, Open and free standard)
- **.mts, m2ts** (**AVCHD**, Advanced Video Codec High Definition, Sony, Panasonic e altri)

N.B.: Tutti questi formati sono incompatibili fra di loro e quindi richiedono programmi specifici per essere creati, modificati, riprodotti

Fine del modulo

