

Informatica A e B

Parte I

Codifica delle informazioni

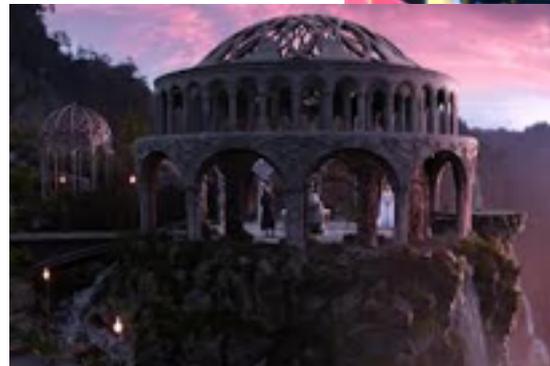
Antonio Lieto

slides su:

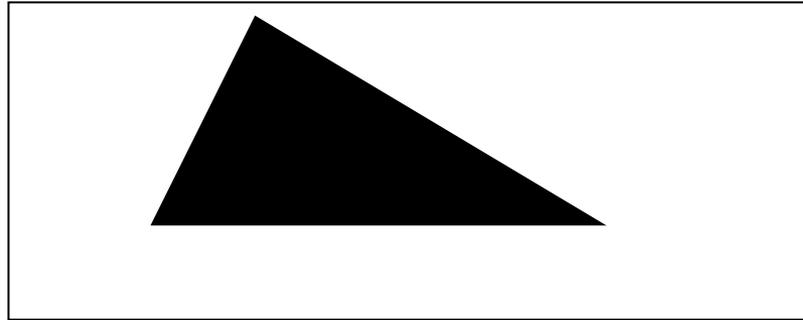
<http://www.antoniolieto.net>

Codifica delle immagini

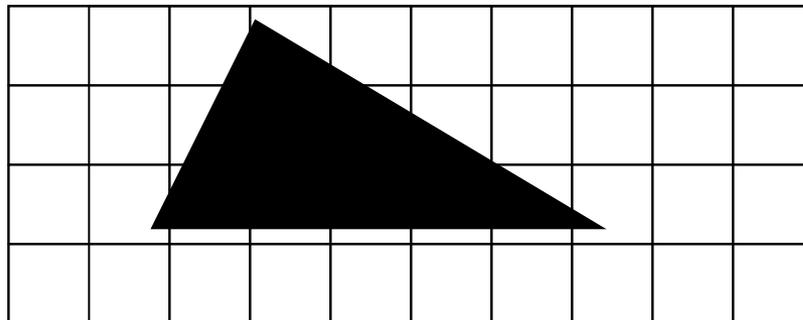
- Sistemi di supporto alla progettazione CAD/CAM
- Campo medico (per esempio, i sistemi di TAC o di ecografia)
- Telecamere e videoregistratori professionali
- Effetti speciali in film, televisione
- Telefoni cellulari
- Ecc.



Codifica delle immagini



Suddividiamo l'immagine mediante una griglia formata da righe orizzontali e verticali a distanza costante.

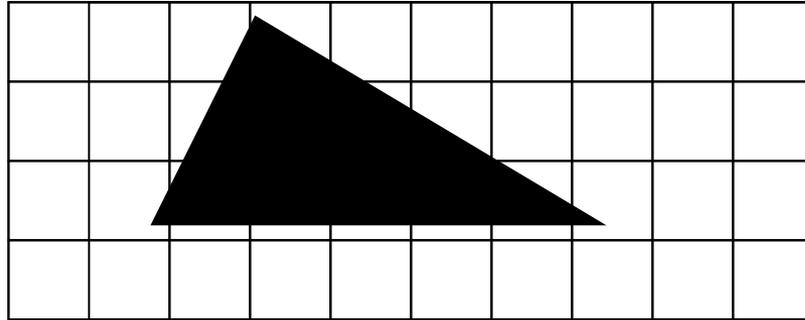


Risoluzione
10 X 4

Codifica delle immagini

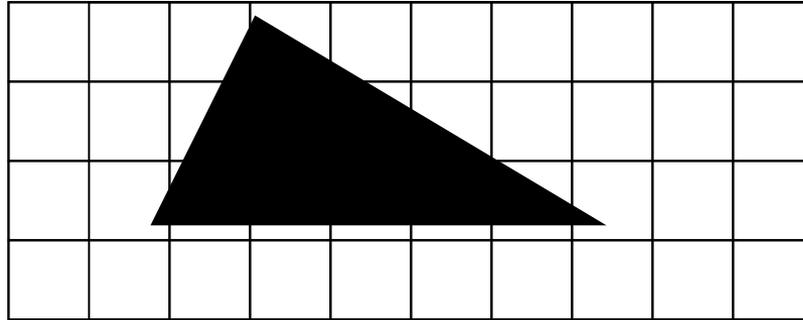
- Ogni quadratino derivante da tale suddivisione prende il nome di **pixel** (**picture element**) e può essere codificato in binario secondo la seguente convenzione:
 - Il simbolo “0” viene utilizzato per la codifica di un pixel corrispondente ad un quadratino in cui il bianco è predominante.
 - Il simbolo “1” viene utilizzato per la codifica di un pixel corrispondente ad un quadratino in cui il nero è predominante.

Codifica delle immagini



???

Codifica delle immagini



0	0	0	1	0	0	0	0	0	0
0	0	1	1	1	0	0	0	0	0
0	0	1	1	1	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0

Codifica delle immagini

0	0	0	1	0	0	0	0	0	0
0	0	1	1	1	0	0	0	0	0
0	0	1	1	1	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0



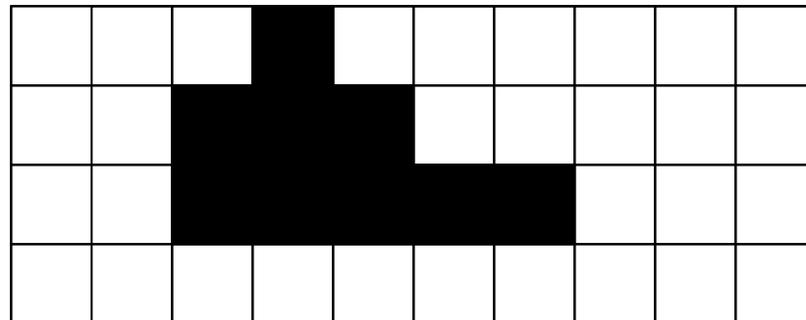
Poiché una sequenza di bit è lineare, è necessario definire convenzioni per ordinare la griglia dei pixel in una sequenza. Assumiamo che i pixel siano ordinati dal basso verso l'alto e da sinistra verso destra

0000000000 0011111000 0011100000 0001000000

Codifica delle immagini

Non sempre il contorno della figura coincide con le linee della griglia. Quella che si ottiene nella codifica è un'approssimazione della figura originaria

Se riconvertiamo la sequenza di stringhe
0000000000 0011111000 0011100000 0001000000
in immagine otteniamo:

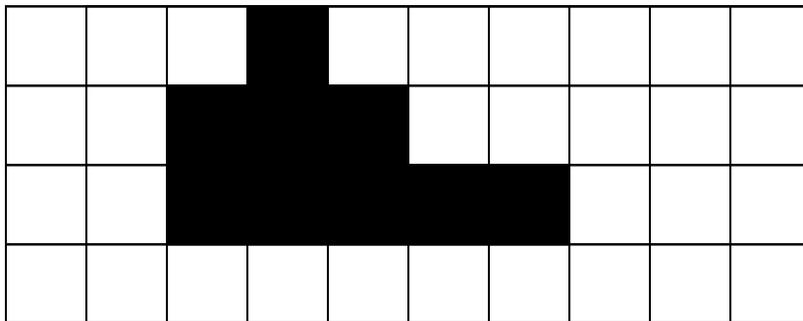


Riconversione

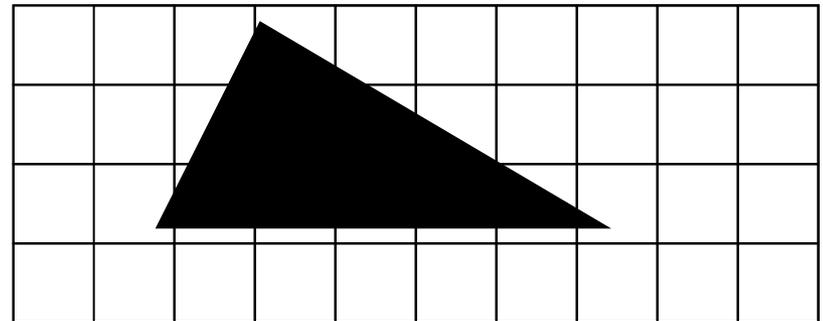
Codifica delle immagini

Non sempre il contorno della figura coincide con le linee della griglia. Quella che si ottiene nella codifica è un'approssimazione della figura originaria

Se riconvertiamo la sequenza di stringhe
0000000000 0011111000 0011100000 0001000000
in immagine otteniamo:



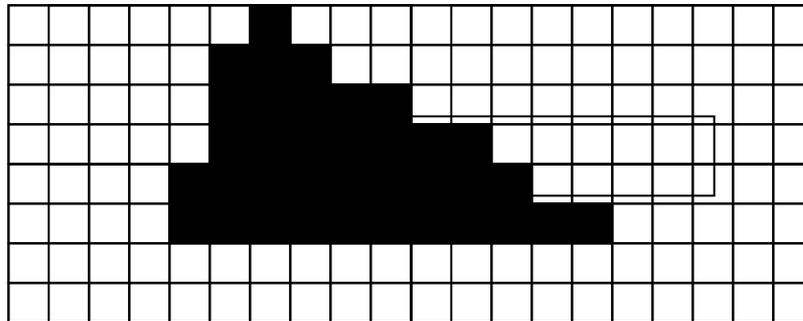
Riconversione



Originale

Codifica delle immagini

La rappresentazione sarà più fedele
all'aumentare
del numero di pixel, ossia al diminuire delle
dimensioni dei quadratini della griglia in cui è
suddivisa l'immagine.



Risoluzione
20 x 8

Codifica delle immagini

Quindi: le immagini sono rappresentate con un certo livello di approssimazione, o meglio, di risoluzione, ossia il numero di pixel usati per riprodurre l'immagine:

- 640 x 480 pixel; 800 x 600 pixel
- 1024 x 768 pixel; 1280 x 1024 pixel
- 1400 x 1280 pixel

Codifica delle immagini

- Assegnando un bit ad ogni pixel è possibile codificare solo immagini in bianco e nero.
- Per codificare le immagini con **diversi livelli di grigio** oppure **a colori** si usa la stessa tecnica: per ogni pixel viene assegnata una *sequenza* di bit.

Codifica delle immagini (grigio e colore)

- Per memorizzare un pixel non è più sufficiente un solo bit (di spazio di memoria).
 - Per esempio, se utilizziamo quattro bit possiamo rappresentare $2^4 = 16$ livelli di grigio o 16 colori diversi.
 - Mentre con otto bit ne possiamo distinguere $2^8 = 256$, ecc.

Attenzione : fare sempre attenzione nel calcolo
- a bit e byte 256 colori = 1 byte per pixel

- 1 pixel a 2 colori 1 bit
- 1 pixel a 256 colori 1 byte (1 * 8 bit)
- 1 pixel a 65535 colori 2 byte (2 * 8 bit)
- 1 pixel a 16 Mil. di colori 3 byte (3 * 8 bit)

Codifica delle immagini

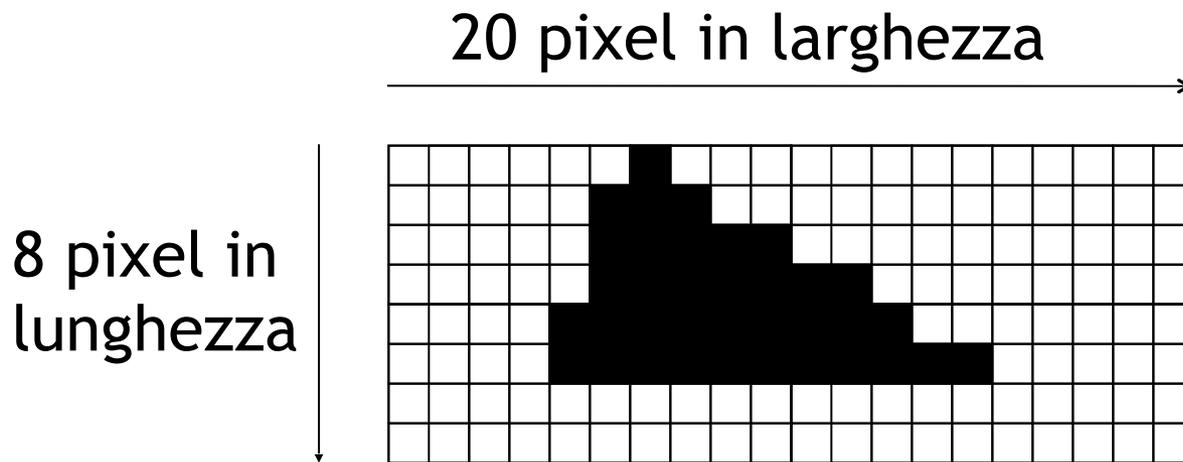
- Complessivamente per essere riprodotta (per esempio, sullo schermo), la rappresentazione binaria di un'immagine deve anche avere informazioni su:
 - Numero di righe dell'immagine.
 - Numero di colonne dell'immagine.
 - Numero di colori usati.

Qualità delle immagini digitali

- Due parametri di qualità: **risoluzione** e **profondità del colore**
 - La **risoluzione** indica la precisione con cui viene effettuata la suddivisione di un'immagine in pixel.
 - La risoluzione si misura dunque in **pixel**.
 - La **profondità del colore** è il numero di bit o byte utilizzati per rappresentare ciascun pixel.
 - La profondità del colore si misura quindi in **bit o byte**.
 - Indica (ma non è uguale a) il numero di colori diversi che possono essere rappresentati.
- La **dimensione** dell'immagine è il numero di **bit o byte** che servono per memorizzarla.
 - **Dimensione** = **risoluzione** x **profondità del colore**.
 - **Maggiore la qualità, maggiore la dimensione!**

Dimensione di un'immagine

- Esempio 1: sto usando **160 pixel** (20x8).

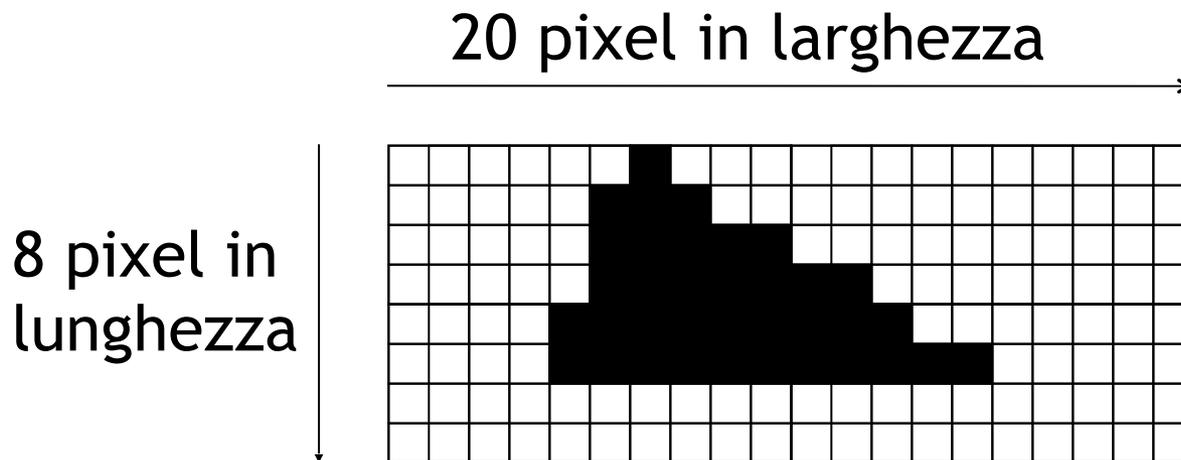


Sto inoltre usando 2 colori = **1 bit** per pixel.

- Quanti bit occupa l'immagine?
- Quanti byte?

Dimensione di un'immagine

- Esempio 1: sto usando **160 pixel** (20x8).



Sto inoltre usando 2 colori = **1 bit** per pixel.

- Quanti bit occupa l'immagine?
- Quanti byte?

L'immagine occupa dunque $160 \times 1 = \mathbf{160 \text{ bit}}$.
In byte: $160 \text{ bit} = 160/8 \text{ byte} = \mathbf{20 \text{ byte}}$.

Dimensione di un'immagine

- **Esempio 2:** 640 pixel in larghezza, 480 pixel in lunghezza, 256 colori.
- Per distinguere 256 colori quanti bit sono necessari?
- Quale è la risoluzione totale?
- Quanti byte occupa l'immagine? Quanti bit?

Dimensione di un'immagine

- **Esempio 2:** 640 pixel in larghezza, 480 pixel in lunghezza, 256 colori.
- Per distinguere 256 colori quanti colori sono necessari?
- Sono necessari **otto bit** per la codifica di ciascun pixel.
- **Quale è la risoluzione totale?**
- **$640 \times 480 = 307200$**
- **Quanti byte occupa l'immagine? Quanti bit?**
- **$307.200 \times 8 = 2.457.600$ bit; 307.200 byte (circa 307,2 KB)**

Esercizio 3

- Abbiamo un'immagine 640 X 480 pixel con 65.535 colori

Quanti bit occorrono per 65.535 colori?

- 1) Calcolate la risoluzione totale (base x altezza)
- 2) Moltiplicate la risoluzione totale x il numero di bit
- 3) Trasformate in byte

Soluzione Esercizio 3

- Abbiamo un'immagine 640 X 480 pixel con 65.535 colori

Quanti bit occorrono per 65.535 colori?

16

1) Calcolate la risoluzione totale (base x altezza):

307.200

2) Moltiplicate la risoluzione totale x il numero di bit

$307.200 * 16 = 4.915.200$

3) Trasformate in byte: **614.400 byte** (614,4 KB)

Qualità e dimensione



318x234 pixel
3 B per pixel (16M colori)
 $318 \times 234 \times 3 = 223236 \text{ B} \approx 218 \text{ KB}$



80x59 pixel
3 B per pixel (16M colori)
 $80 \times 59 \times 3 = 14160 \text{ B} \approx 14 \text{ KB}$



48x35 pixel
3 B per pixel (16M colori)
 $48 \times 35 \times 3 = 5040 \text{ B} \approx 5 \text{ KB}$



318x234 pixel
3 B per pixel (16M colori)
 $318 \times 234 \times 3 = 223236 \text{ B} \approx 218 \text{ KB}$



318x234 pixel
4 bit per pixel (16 colori)
 $318 \times 234 \times 4 = 297648 \text{ bit} \approx 36 \text{ KB}$



318x234 pixel
3 bit per pixel (8 colori)
 $318 \times 234 \times 3 = 223236 \text{ bit} \approx 27 \text{ KB}$

Compressione delle immagini

- Le immagini codificate pixel per pixel sono dette immagini in grafica **bitmap** (mappa di bit) o **raster**.
 - Le immagini bitmap occupano parecchio spazio.
- Esistono delle **tecniche di compressione** che permettono di ridurre le dimensioni.
 - Sono essenzialmente modi “più intelligenti” di memorizzare le immagini, invece che elencare semplicemente i pixel che le compongono.
 - A volte comportano una riduzione della qualità dell’immagine (che normalmente risulta impercettibile per l’occhio umano).
- Esempi di formati diffusi per le immagini:
 - **GIF, PNG e JPEG.**

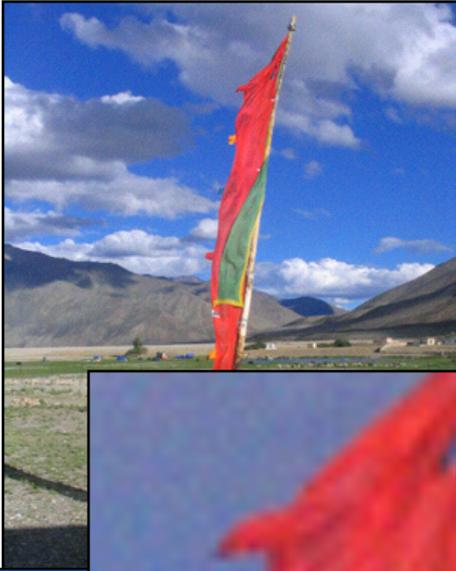
Compressione delle immagini

- Distinzione tra compressione **lossless** e compressione **lossy**.
 - **Lossless**: compressione senza perdita di informazioni.
 - Dalla versione compressa, si può ricostruire perfettamente la versione non-compressa.
 - Per esempio, GIF, PNG (ma dipende dal numero di colori).
 - **Lossy**: compressione con perdita di informazioni
 - Dalla versione compressa, non è possibile recuperare la versione originale.
 - Per esempio, JPEG.

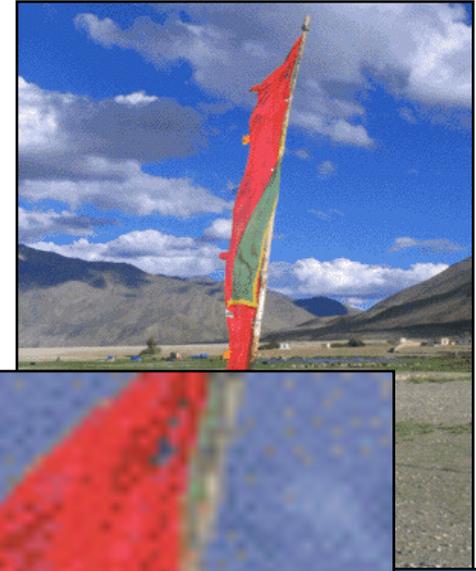
Compressione GIF

- **GIF = Graphic Interchange Format**
 - Applicabile quando i colori sono “pochi” (max 8 bit = 256 colori).
 - Vantaggi:
 - Se l’immagine ha pochi colori non viene “deteriorata” (compressione lossless).
 - Svantaggi:
 - Se l’immagine ha molti colori (es. una foto) bisogna ridurli, perdendo in qualità.
- Pertanto questo formato è preferibile per immagini dalle linee nette e con poche sfumature.

Compressione GIF



352 KB



47 KB



Compressione JPEG

- **JPEG = Joint Photographic Expert Group**
 - Studiata appositamente per le fotografie.
 - Si basa sul principio di rinunciare ad una parte dell'informazione presente nell'immagine (compressione lossy) quando quell'informazione non verrebbe comunque percepita dall'occhio umano.
 - Ovviamente se si esagera con la riduzione dell'informazione la perdita di qualità diventa percepibile.
 - Le idee su cui è basata (in breve):
 - L'occhio umano è più sensibile alle **variazioni di luminosità che di tonalità**: meglio dunque sacrificare le seconde.
 - Tali variazioni di luminosità sono ben percepite su aree ampie, **ma non su aree piccole** (un singolo pixel molto più luminoso in mezzo ad altri più scuri non viene notato).
- Il formato JPEG è adatto ad **immagini di tipo fotografico**, dove la perdita di qualità si nota di meno, o in generale ad immagini dove un limite sul numero di colori produrrebbe differenze troppo significative.

Compressione JPEG

352 KB



38 KB



23 KB



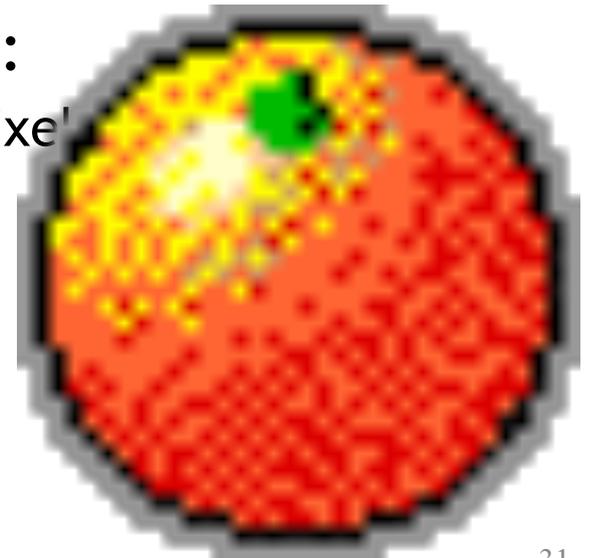
PNG

PNG (Portable Network Graphics)

- ◆ Consente di esportare in 3 formati:
8-24-32 bit
- ◆ Migliore resa del JPEG
- ◆ Ora è pienamente supportate dai
browser

Grafica bitmap vs. grafica vettoriale

- Un **oggetto bitmap/raster** è memorizzato semplicemente come una griglia di pixel a ciascuno dei quali è associato un colore.
 - Una volta disegnata una linea, essa non è più una “linea” ma solo un insieme di pixel sullo schermo.
 - La qualità dipende da tanti fattori:
 - Numero di bit utilizzati per ciascun pixel
 - Caratteristiche dell’immagine.
 - Algoritmo di compressione usato.
 - Fattore di ingrandimento sul video.
- L’ingrandimento fa perdere risoluzione



Grafica bitmap vs. grafica vettoriale

- Un **oggetto vettoriale** è costituito da una sequenza di segmenti e un insieme di attributi.
 - Segmenti:
 - Possono essere dritti o curvi, e possono essere uniti a formare una linea spezzata, aperta o chiusa.
 - Vengono memorizzati registrando le coordinate delle estremità di ciascun segmento.
 - Attributi:
 - Ad esempio il colore e lo spessore della linea, il riempimento se si tratta di una figura chiusa (ad es. un rettangolo), etc.
 - Gli attributi sono memorizzati **separatamente** dalle coordinate: è possibile modificare le coordinate lasciando inalterati gli attributi, oppure modificare gli attributi senza cambiare la forma dell'oggetto.
 - La qualità dell'immagine è indipendente dal fattore di ingrandimento.

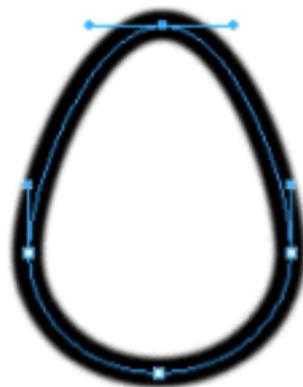
Oggetto Vettoriale

- sequenza di segmenti dritti o curvi, uniti a formare una linea aperta o chiusa

- unità di memorizzazione
base: segmento

- come viene memorizzata l'informazione sui segmenti?
coordinate delle estremità
+ attributi (colore, spessore del tratto...)

- memorizzazione separata
-> manipolazione basata sul tipo di memorizzazione:
modifico le coordinate senza cambiare gli attributi o viceversa

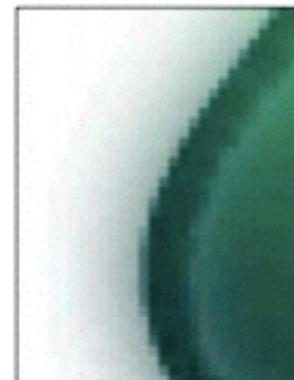


Oggetto Bitmap

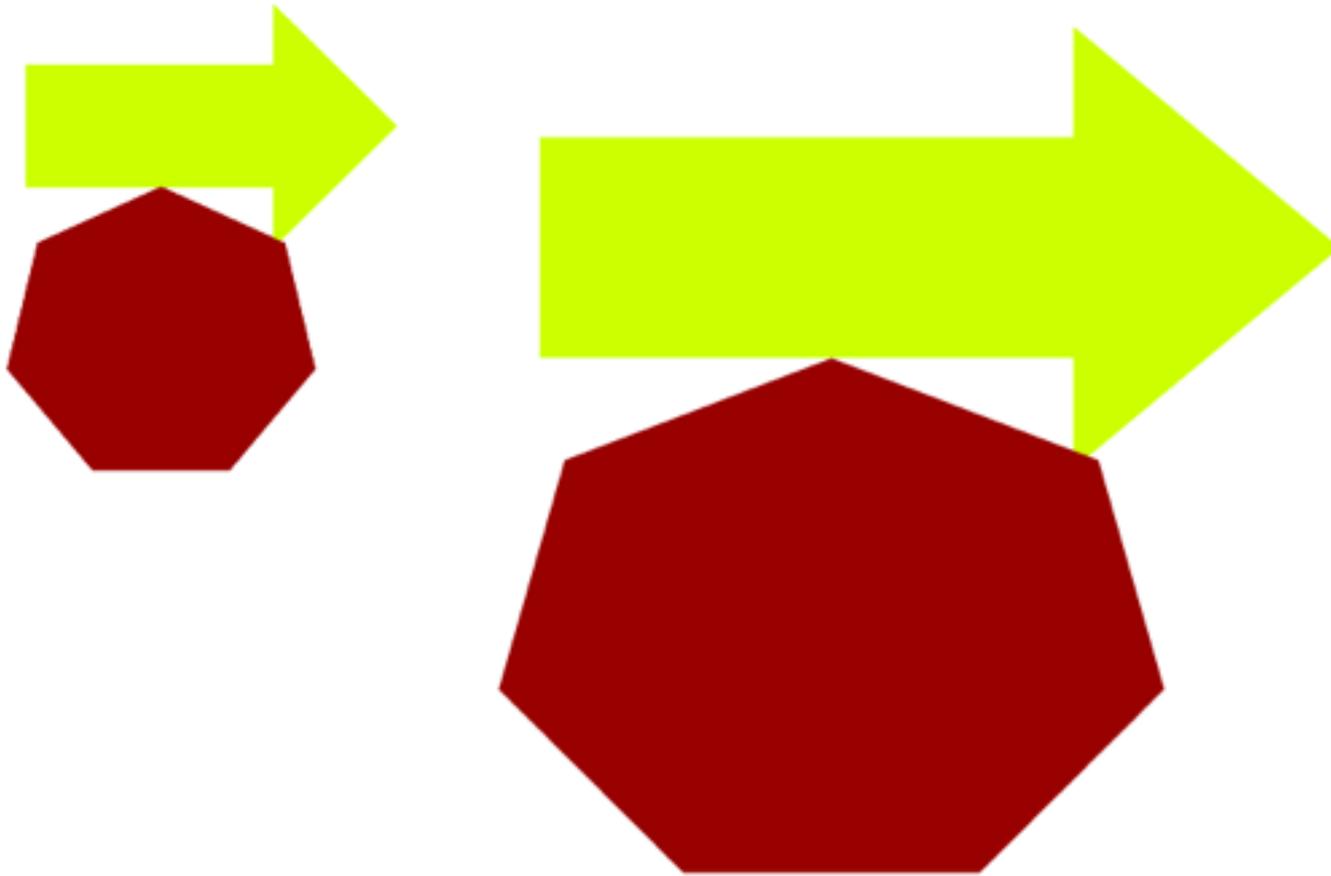
- griglia di pixel
- unità di memorizzazione
base: pixel
- come viene memorizzata l'informazione?

Per ogni pixel viene memorizzata l'informazione sul colore

- manipolazione basata sul tipo di memorizzazione, ossia ossia sul concetto di pixel:
coloro diversamente i pixel;
tipica manipolazione: applicazione di filtri



Grafica bitmap vs. grafica vettoriale



L'ingrandimento non fa perdere la risoluzione

Le immagini vettoriali

Le immagini vengono costruite a partire dalla descrizione degli elementi che le compongono mediante un linguaggio testuale

- Ad es, SVG (Standard Vector Graphics)

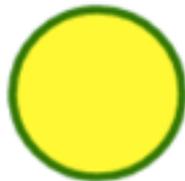
```
<!DOCTYPE html>
<html>
<body>

<h1>My first SVG</h1>

<svg width="100" height="100">
  <circle cx="50" cy="50" r="40" stroke="green" stroke-width="4" fill="yellow" />
  Sorry, your browser does not support inline SVG.
</svg>

</body>
</html>
```

My first SVG



Codifica di immagini in movimento

- Un filmato è una **sequenza di immagini statiche** (dette *fotogrammi* o *frame*).
- Per codificare un filmato si **digitalizzano i suoi fotogrammi**.
- Tanto maggiore è il numero di fotogrammi tanto migliore apparirà la qualità del movimento.
- I filmati in digitale possono essere molto pesanti.

Compressione di immagini in movimento

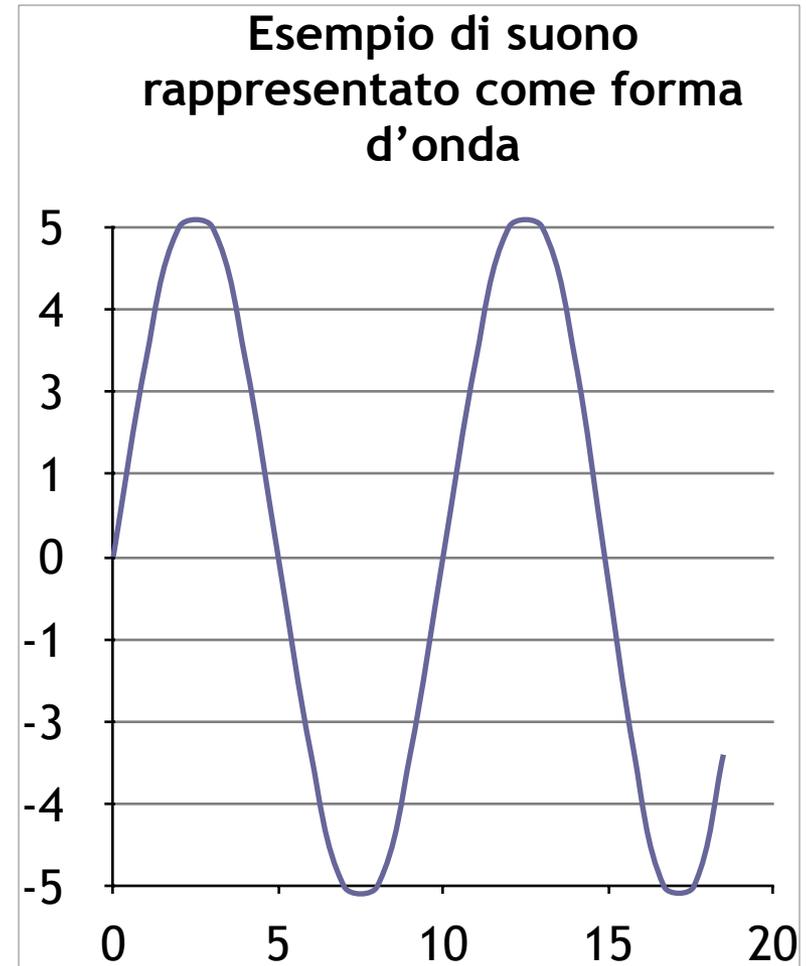
- Si possono comprimere le immagini con le tecniche viste prima.
- Si memorizza il primo fotogramma e nei successivi si memorizzano le differenze rispetto a quello iniziale.
- Dopo un certo numero di fotogrammi si memorizza un nuovo fotogramma in modo completo.
- Esempi di formati per il video: **AVI**, **MOV**.
- Compressione: **MPEG** (Moving Picture Expert Group), differenza tra fotogrammi.
- **CODEC (Compress or DECompress)**: software in grado di codificare e decodificare un flusso di dati, come una sequenza video.

Codifica dei suoni

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

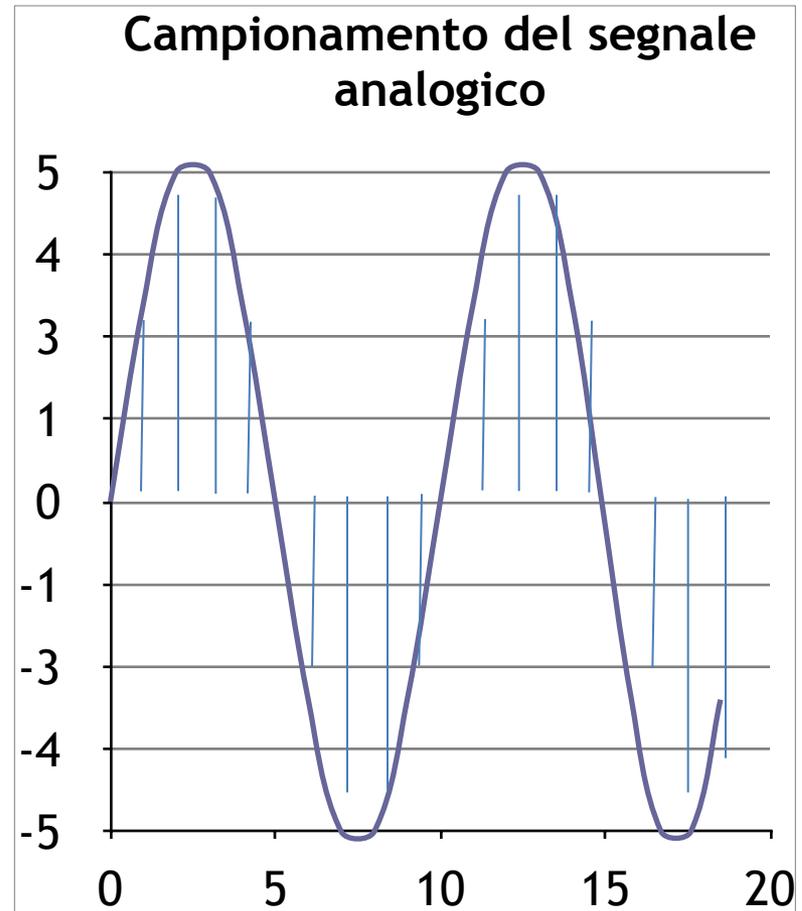
Codifica dei suoni

- Un suono può essere rappresentato mediante un'onda che descrive la variazione della pressione dell'aria nel tempo -> **onda sonora**
- Sull'asse delle ascisse viene rappresentato il **tempo** e sull'asse delle ordinate viene rappresentata la **variazione di pressione** corrispondente al suono stesso.
- Un esempio di **rappresentazione analogica** : l'onda è continua; i valori dell'onda corrispondono a numeri reali.



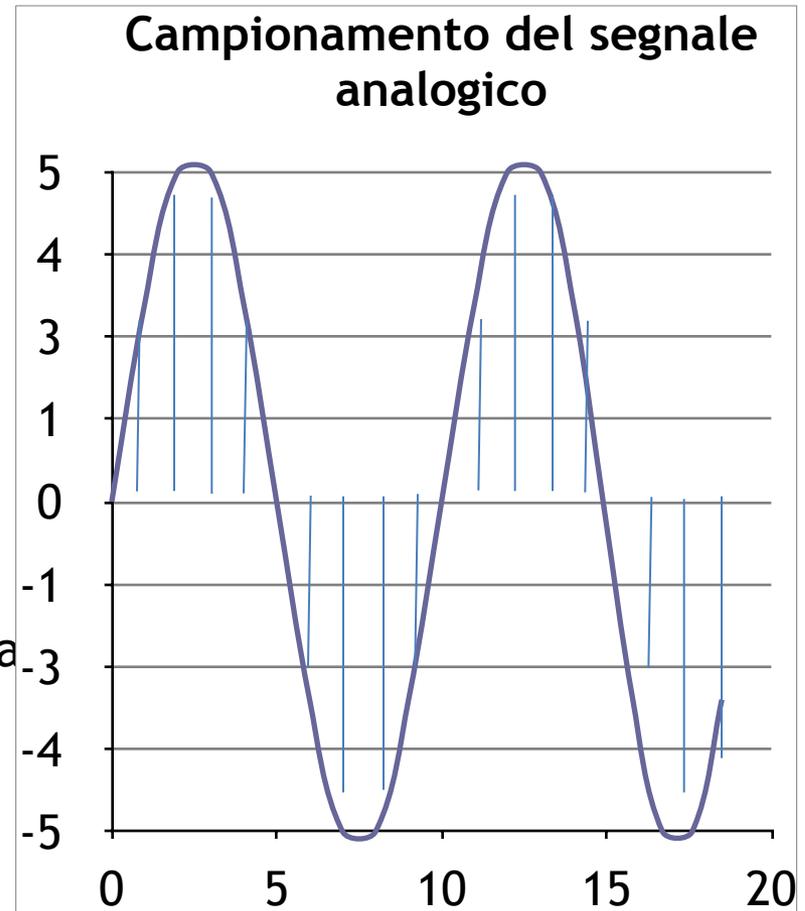
Codifica dei suoni

- Come nel caso delle immagini avevamo bisogno di una tecnica di **discretizzazione**, analogamente nel caso dei suoni abbiamo bisogno di una tecnica che ci permetta di passare dalla rappresentazione fisica continua dell'onda sonora a una rappresentazione digitale discreta
- Intuitivamente: non memorizziamo l'informazione completa sull'onda -> come nel caso delle immagini si **deve perdere in qualità dell'informazione per avere una rappresentazione digitale**



Codifica dei suoni

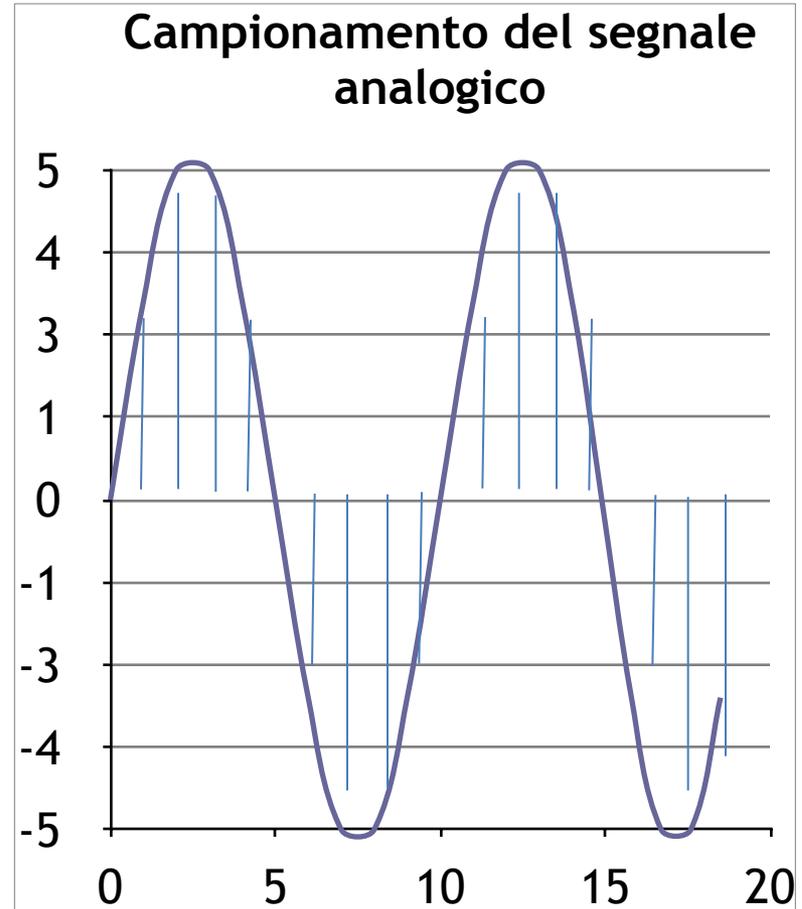
- La conversione di un segnale continuo in una successione di numeri viene eseguita con due successive operazioni elementari:
- 1. **Campionamenti sull'onda:** si preleva una successione di campioni a intervalli costanti di tempo (in altre parole: si misura il valore dell'onda a intervalli di tempo costante)
- 2. **Quantizzazione e codifica:** ogni campione viene quantizzato ossia convertito in un numero; la sequenza dei valori numerici ottenuta dai campioni può essere facilmente codificata formato digitale->



Associo una configurazione di bit a ogni campione

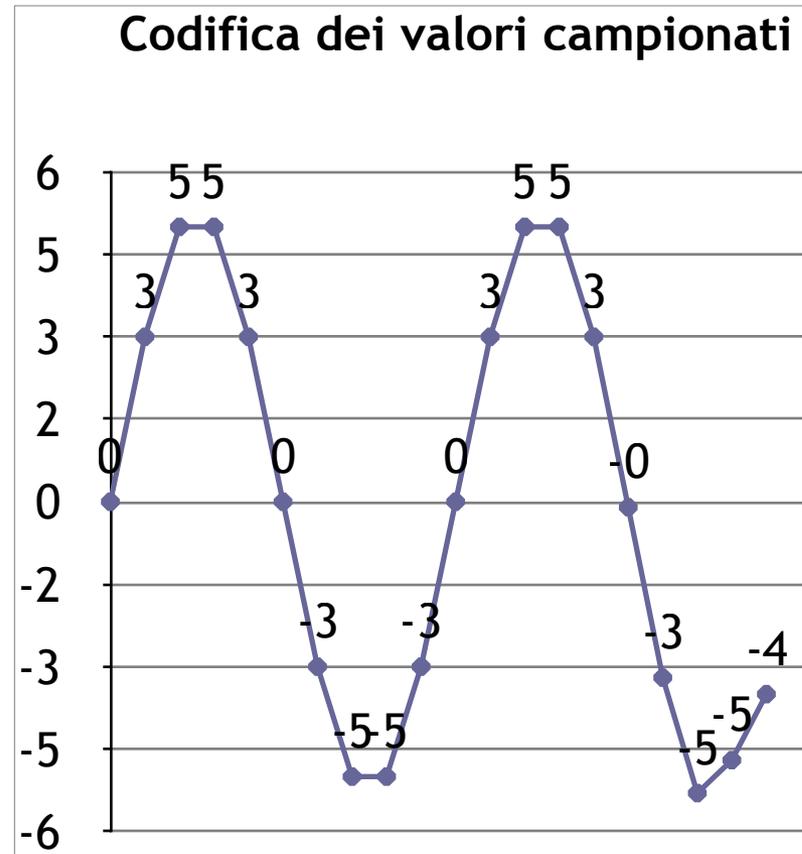
Codifica dei suoni

- Il segnale analogico deve essere convertito in rappresentazione digitale (**digitalizzato**).
 - Quanto più frequentemente il valore di intensità dell'onda viene campionato, tanto più precisa sarà la sua rappresentazione.
 - Il numero di campioni raccolti per ogni secondo definisce la **frequenza di campionamento**. Il numero di campioni raccolti per ogni secondo si misura in Hertz (Hz)
- Es. CD Musicale.
- 44000 campionamenti al secondo,
- Standard: 16 bit per campione



Codifica dei suoni

- La sequenza dei valori numerici ottenuti dai campioni può essere facilmente codificata con sequenze di bit (la **codifica dei numeri** è il prossimo argomento di questa lezione).
- La rappresentazione è tanto più precisa quanto maggiore è il numero di bit utilizzati per codificare l'informazione estratta in fase di campionamento.



Codifica dei suoni

- Codifiche standard
 - MP3 , WAV (MS-Windows), AIFF (Audio Interchange File Format, Apple), MIDI.
- **MP3**
 - Variante MPEG per suoni.
 - Lossy.
 - Grande diffusione, molto efficiente (fattore di compressione circa 5:1 - 10:1, circa 1-2 MB ogni minuto).
- **MIDI**: codifica le note e gli strumenti che devono eseguirle
 - Efficiente, ma solo musica, non voce.

Osservazione

- Per calcolare lo spazio occupato da un file di testo, da un'immagine, da un file audio, la tecnica è sempre la stessa:
- Si trova lo 'spazio' (bit) occupato dalla codifica di ogni unità elementare costituente il file (un carattere per il testo, un pixel per l'immagine, un campione per il file audio)
- Si trova il numero di unità elementari che costituiscono il file (il numero di caratteri per il testo, il numero di pixel per l'immagine -sfruttando la risoluzione, il numero di campioni per il file audio - sfruttando la frequenza di campionamento)
- Si moltiplicano queste due quantità

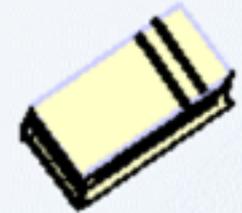
Cosa memorizzo in 1MB?

8 milioni di bit
(~1MB)

Un milione di caratteri
70 caratteri per riga
40 righe per pagina
400 pagine circa



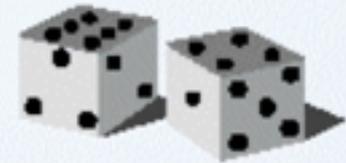
Libro giallo



1.000 x 1.000 pixel
256 livelli di grigio



Immagine in bianco
e nero ad alta risoluzione



125 sec. di voce o 5.6 sec.
di musica ad alta fedeltà



1/30 di secondo di filmato
in bianco e nero
ad alta risoluzione



Codifica dei numeri

- Il codice ASCII consente di codificare le cifre decimali da “0” a “9” fornendo in questo modo una rappresentazione dei numeri.
- Per esempio: il numero 324 potrebbe essere rappresentato dalla sequenza di byte:

00110011 00110010 00110100
3 2 4

- Quale è il problema di questa rappresentazione ?

Codifica dei numeri

00110011 00110010 00110100
3 2 4

- Questa rappresentazione non è efficiente e soprattutto *non è adatta per eseguire le operazioni aritmetiche sui numeri* (sono solo simboli allineati).
- Sono stati pertanto studiati **codici alternativi per rappresentare i numeri in modo efficiente** ed eseguire le usuali **operazioni aritmetiche**.

Codifica dei numeri

(il sistema decimale)

- La rappresentazione dei numeri con il sistema decimale può essere utilizzata come spunto per definire un metodo di codifica dei numeri all'interno degli elaboratori.
- **Notazione posizionale**: ogni cifra del numero assume un certo valore in funzione della sua **posizione**
 - Esempio: la sequenza di cifre 324 viene notazione compatta, sta per:
 - 3 centinaia + 2 decine + 4 unità
 - $324 = 3 \times 100 + 2 \times 10 + 4 \times 1$
 - $324 = 3 \times 10^2 + 2 \times 10^1 + 4 \times 10^0$ (notazione esplicita)

Codifica dei numeri

- La numerazione decimale quindi utilizza **una notazione posizionale basata sul numero 10**.
- La notazione posizionale può essere utilizzata in qualunque altro sistema di numerazione.
- Per ogni sistema di numerazione si usa un numero di cifre uguale alla base.
 - **Base:** il numero di cifre disponibile nel sistema.
 - In base 10 (il sistema di numerazione *decimale*), i numeri vengono codificati utilizzando le dieci cifre 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.
 - In base 2, (il sistema di numerazione *binario*) i numeri vengono codificati utilizzando le due cifre 0 e 1.

Riepilogo

Codifica in termini binari di diversi input:

- Testo (ASCII, UTF-8...)
- Immagini (Pixel, Risoluzione, Profondità)
- Filmati (riduzione al problema “immagini statiche”)
- Suoni (passaggio da onda sonora analogica a digitale tramite campionamento e quantizzazione)
- Numeri (da binario a decimale)

- Il sistema di numerazione binario (base 2) utilizza una notazione posizionale basata su 2 cifre (0 e 1) e sulle potenze di 2
- Il numero 1001 può essere rappresentato esplicitamente come
$$1001_2 = 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = ?$$

$$1001_2 = 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 9_{10}$$

Conversione base 2 \rightarrow base 10

- Esempio: la sequenza “1011” denota il numero

$$1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 11 \text{ (in base 10)}$$

- Esempio: la sequenza “10011” denota il numero

$$1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 19 \text{ (in base 10)}$$

- Per evitare ambiguità si usa la notazione

$$1011_2 = 11_{10}, \quad 10011_2 = 19_{10}$$

Conversione base 10 \rightarrow base 2

Per convertire un numero in base 2 si devono trovare i resti delle divisioni successive del numero per la base 2

Esempio: 157_{10}

157 / 2 resto 1
78 / 2 resto 0
39 / 2 resto 1
19 / 2 resto 1
9 / 2 resto 1
4 / 2 resto 0
2 / 2 resto 0
1 / 2 resto 1 |

10011101

Conversione base 10 \rightarrow base 2

- Consideriamo ad esempio il numerale 13 in base 10 e cerchiamo la rappresentazione corrispondente in base 2.

Conversione base 10 \rightarrow base 2

13/2	=	6	resto	1	
6/2	=	3	resto	0	
3/2	=	1	resto	1	
1/2	=	0	resto	1	

13 espresso in base 2
1101

Conversione base 10 \rightarrow base 2

- Consideriamo ad esempio il numerale 42 in base 10 e cerchiamo la rappresentazione corrispondente in base 2.

Conversione base 10 \rightarrow base 2

42/2	=	21	resto	0
21/2	=	10	resto	1
10/2	=	5	resto	0
5/2	=	2	resto	1
2/2	=	1	resto	0
1/2	=	0	resto	1



42 espresso in base 2
101010

Esercizi: conversione base 2 -> 10

- 111011
- 101111101
- 110000
- 110010101
- 101010101010
- 101000000001

Suggerimenti:

2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
256	128	64	32	16	8	4	2	1

Ancora: $2^9 = 512$. $2^{10} = 1024$. $2^{11} = 2048$

Soluzioni

- $111011 =$
- $101111101 = 381$
- $11000048 =$
- $110010101 = 405$
- $101010101010 = 2730$
- $101000000001 = 2561$

Esercizi: conversione base 10-> 2

128

1234

5654

1233

234

511

Soluzioni

128	10000000
1234	10011010010
5654	1011000010110
1233	10011010001
234	11101010
511	111111111

Es. Operazioni

- Addizione Binaria

$0 + 0$	$=$	0	con riporto	0
$0 + 1$	$=$	1	con riporto	0
$1 + 0$	$=$	1	con riporto	0
$1 + 1$	$=$	0	con riporto	1

Addizione binaria

$\begin{array}{r} 1 + \\ 1 = \\ \hline 10 \end{array}$	$\begin{array}{r} 101 + \\ 11 = \\ \hline 1000 \end{array}$	$\begin{array}{r} 10110101 + \\ 1000110 = \\ \hline 11111011 \end{array}$
--	---	---

$0 + 0 = 0$	con riporto	0
$0 + 1 = 1$	con riporto	0
$1 + 0 = 1$	con riporto	0
$1 + 1 = 0$	con riporto	1

Addizione binaria

$$\begin{array}{r} 10111001 + (185) \\ 10000100 = (132) \\ \hline 10011101 \quad (317) \end{array}$$

Notazione esadecimale

- Vale la pena menzionare un altro tipo di notazione: la notazione **esadecimale**, ossia in base 16.
 - Il principio è lo stesso della notazione decimale (base 10) e binaria (base 2).
 - Si usano 16 cifre: 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F.

– Il numero **B7FC** può essere rappresentato esplicitamente come

$$(11) \times 16^3 + 7 \times 16^2 + (15) \times 16^1 + (12) \times 16^0 = 47100$$

A	B	C	D	E	F
↓	↓	↓	↓	↓	↓
10	11	12	13	14	15

Notazione esadecimale

- Vale la pena menzionare un altro tipo di notazione: la notazione **esadecimale**, ossia in base 16.
 - Il principio è lo stesso della notazione decimale (base 10) e binaria (base 2).
 - Si usano 16 cifre: 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F.
- **Esempi:**
 - $A3_{16} = 10 \times 16^1 + 3 \times 16^0 = 160 + 3 = 163_{10}$
 - $101_{16} = 1 \times 16^2 + 0 \times 16^1 + 1 \times 16^0 = 256 + 0 + 1 = 257_{10}$
 - $FAB_{16} = 15 \times 16^2 + 10 \times 16^1 + 11 \times 16^0 = 3840 + 160 + 11 = 4011_{10}$
- Usata spesso in ambito informatico perché:
 - **Più compatta** della notazione binaria (e anche di quella decimale!)

A93E ₁₆ (43326 ₁₀)	
$A_{16} = 10_{10} = 1010_2$	$9_{16} = 9_{10} = 1001_2$
$3_{16} = 3_{10} = 0011_2$	$E_{16} = 14_{10} = 1110_2$
1010100100111110 ₂	

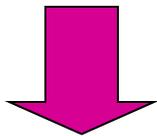
Specifica dei Colori nel Web

- Nelle pagine Web i colori vengono codificati utilizzando il sistema esadecimale.
- Perché? La rappresentazione è più concisa, esprimo tutto con due caratteri alfanumerici per **colore primario** (vedi slide seguente)

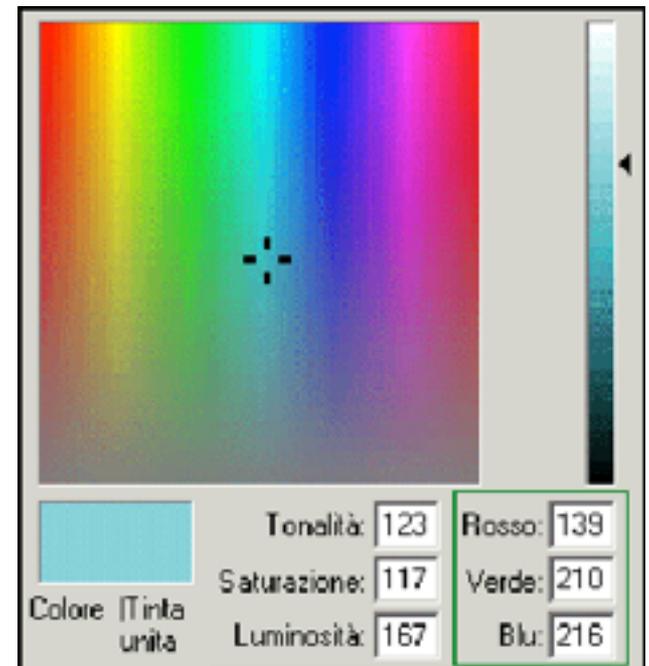
$$139_{10} = 8B_{16}$$

$$210_{10} = D2_{16}$$

$$216_{10} = D8_{16}$$



Codifica del colore:
8B D2 D8



L'uso del colore

- Il colore può essere generato componendo 3 colori: red, green, blue (codifica RGB).
 - secondo questo modello qualsiasi colore può essere rappresentato componendo Rosso Verde e Blu (colori primari)
 - idea base: invece che di partire da tanti colori e di rappresentarne diverse sfumature, possiamo rappresentare molte sfumature/gradazioni a partire dai 3 colori primari



Tutti i possibili colori sono ottenuti dalla combinazione di diverse gradazioni dei 3 colori primari

Specifica dei Colori nel Web

- In particolare quando si vuole specificare un colore è sempre possibile specificarlo specificando la sua codifica esadecimale RGB.

```
1
2 <html>
3 <head>
4 <title>Sfondo azzurro</title>
5 </head>
6 <body bgcolor="#8BD2D8">
7 </body>
8 </html>
9
```

Tonalità:	123	Rosso:	139
Saturazione:	117	Verde:	210
Luminosità:	167	Blu:	216

Ognuno dei 16.777.216 colori dello spettro RGB e' associato a un codice esadecimale che ci permette di identificarlo

Specifica dei Colori nel Web

- Alcune combinazioni note e molto usate

#00 00 00 = nero

#FF FF FF = bianco

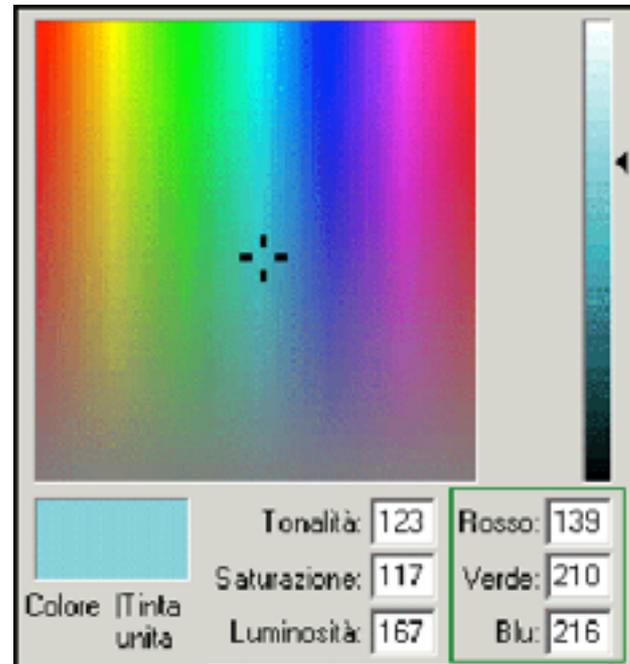
#FF 00 00 = rosso puro

#00 FF 00 = verde puro

#00 00 FF = blu puro

#FF FF 00 = giallo

#FF 00 FF = viola



N.B.

- Non vediamo (interi con segno, overflow, numeri a virgola mobile)

Informatica A e B

Parte II

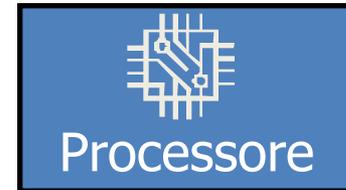
Architettura degli Elaboratori

(Come è fatto un Computer?)

Antonio Lieto

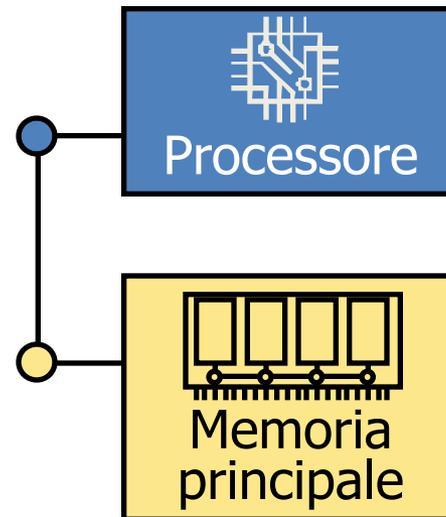
<http://www.antonioleto.net>

Uno sguardo d'insieme



- L'attività principale di un computer è **fare calcoli...**
 - Tale elaborazione è effettuata dal **processore**.
 - In **prima approssimazione** possiamo vedere il processore come **una calcolatrice molto sofisticata**.
 - Poi scopriremo che ha anche altre funzioni...

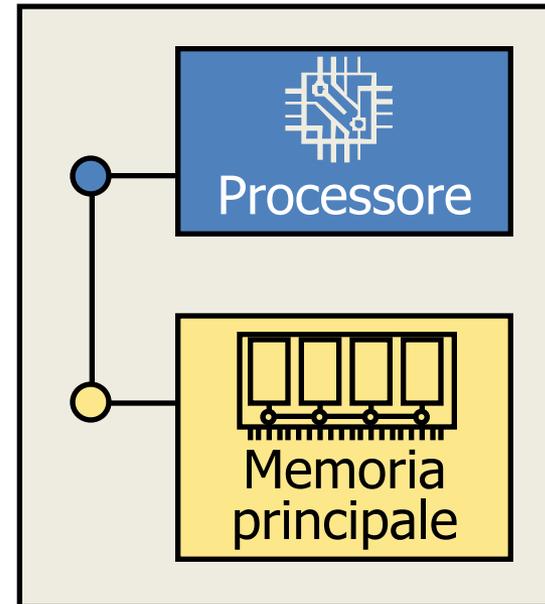
Uno sguardo d'insieme



- Per poter effettuare i calcoli autonomamente, il processore deve conoscere:
 - I **dati** su cui operare.
 - Le operazioni da eseguire → il “**programma**”.
- Durante il calcolo queste informazioni si trovano nella **memoria principale**.

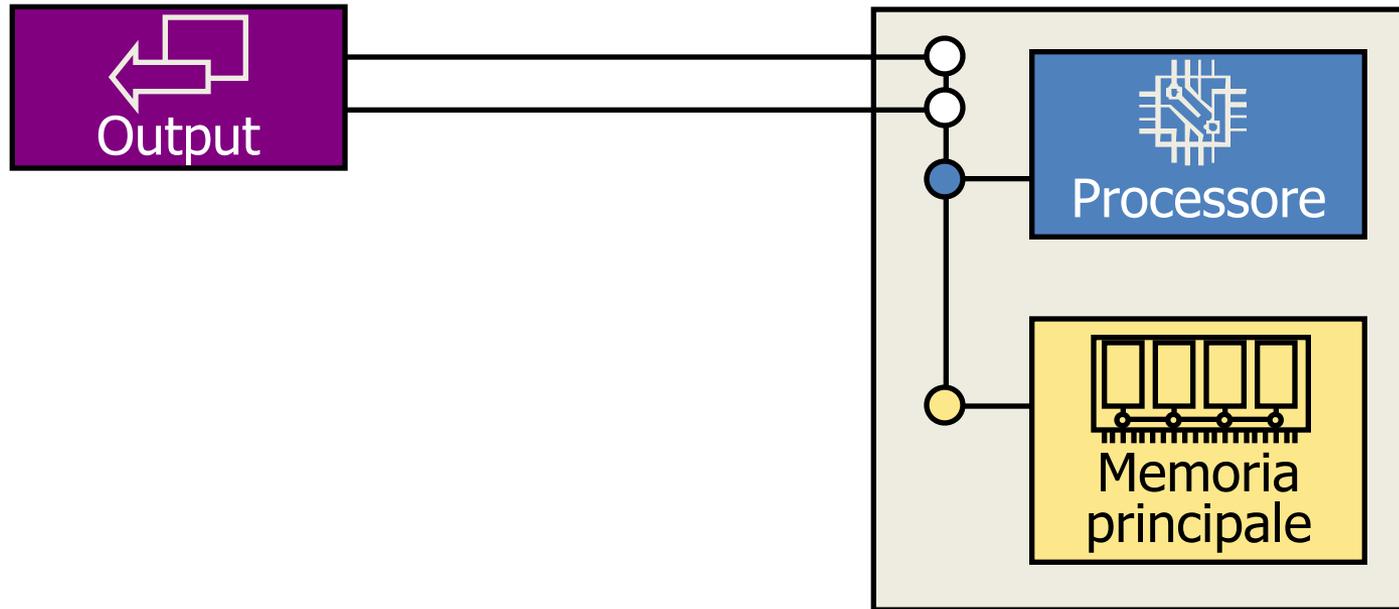
Uno sguardo d'insieme

Unità centrale



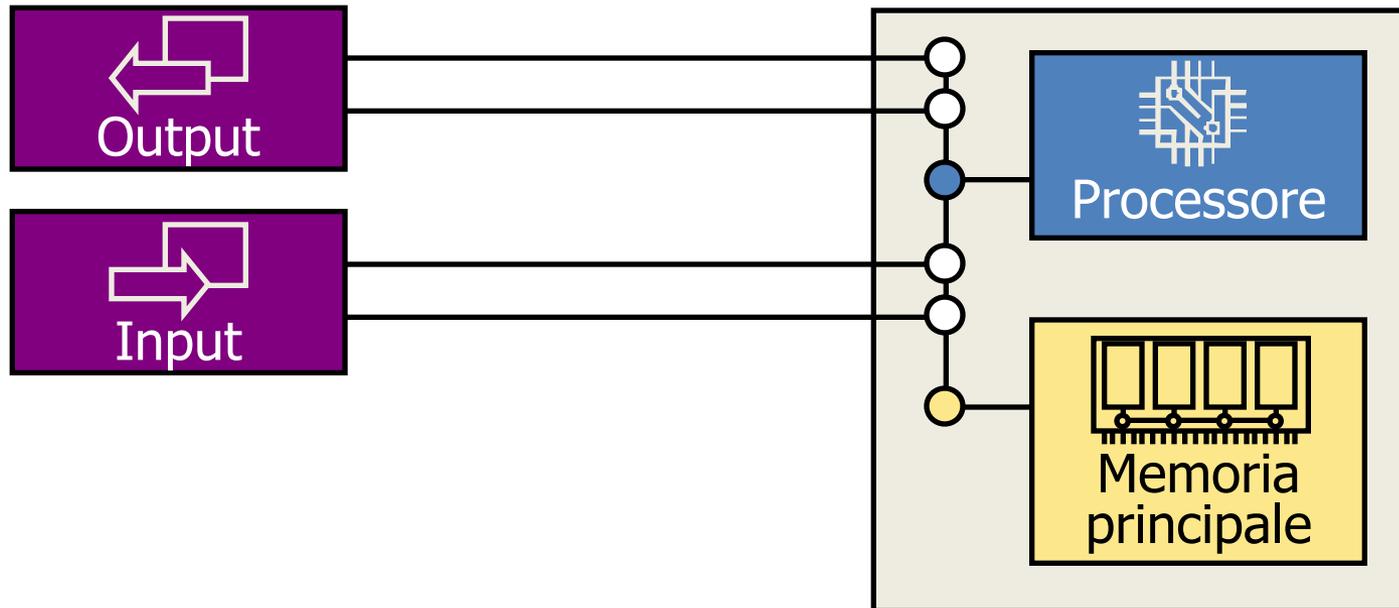
- Queste due componenti, collegate fra loro per poter trasferire le informazioni da una all'altra, sono il “cuore” del computer (l'**unità centrale**).
 - Questa architettura di base è ciò che **definisce** un computer.
 - Da Von Neumann & Co. (1945) a oggi non è cambiata l'idea, ma solo la tecnologia che la implementa.
 - In realtà a grandi linee questa era già l'idea di Babbage (XIX secolo)!

Uno sguardo d'insieme



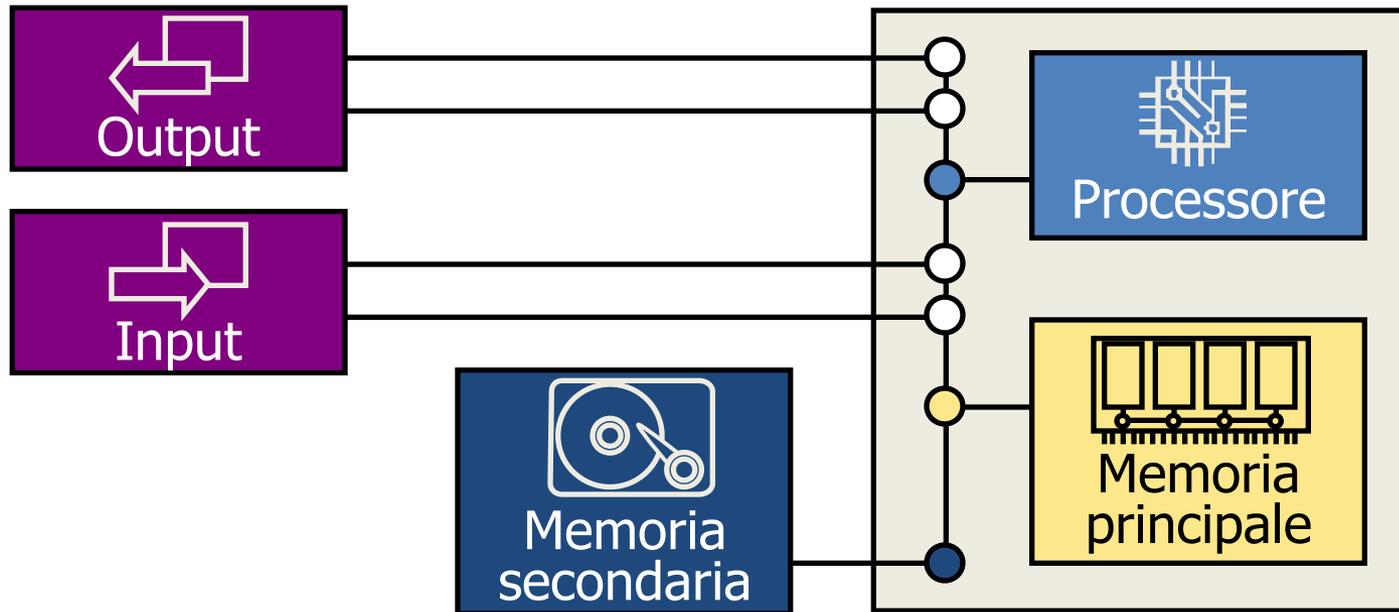
- Ovviamente sarà necessario mostrare **all'esterno** i risultati dell'elaborazione.
 - **Dispositivi di output** (letteralmente: emettere).
 - Almeno un dispositivo di output è obbligatorio: **il video!**
 - Altri tipi di output: **scritto (stampante), audio...**

Uno sguardo d'insieme



- Ma come vengono introdotti i **dati** e i **programmi** nella memoria principale?
- Vengono introdotti **dall'utente**.
 - **Dispositivi di input** (letteralmente: immettere).
 - Tipici dispositivi di input: **tastiera, mouse**.
 - Altri tipi di input: **scanner, microfono, tavoletta grafica...**

Uno sguardo d'insieme



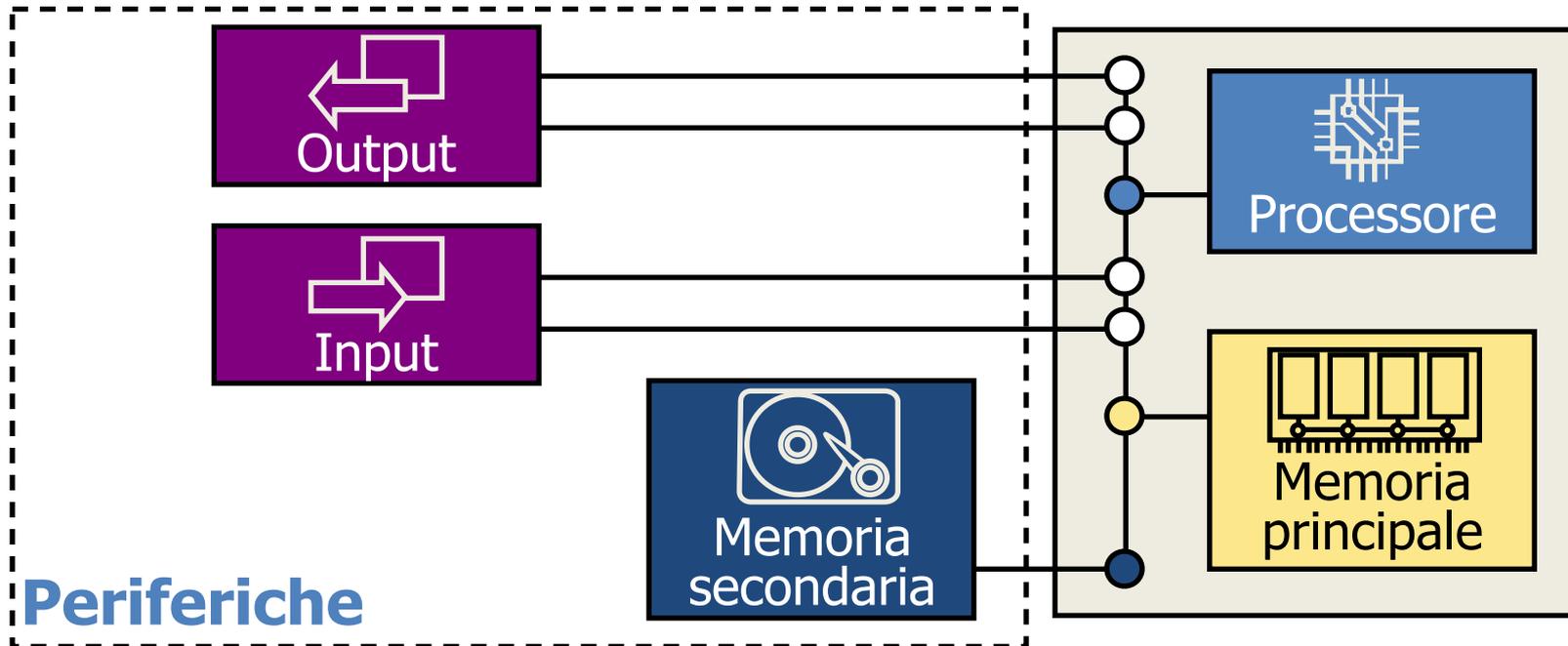
- Problema: la memoria principale si **cancella** quando si spegne il computer.
- Per non costringere gli utenti a reinserire ogni volta dati e programmi, si aggiunge una **memoria secondaria**.
 - La memoria secondaria è **permanente**.
 - Tipiche memorie secondarie: **hard disk, cd, dvd, USB memory stick...**

Principio di funzionamento

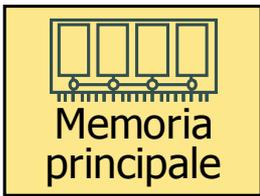
I programmi e dati risiedono in memoria secondaria.

- Per essere eseguiti (i programmi) e usati (i dati) vengono copiati nella memoria primaria.
- La CPU (Central Processing Unit, vedere slides successive) è in grado di eseguire le istruzioni che compongono i Programmi.

Uno sguardo d'insieme



- I dispositivi che non sono parte del cuore del computer sono detti **periferiche**.
 - Sebbene ciascuna funzioni in modo diverso, sono simili nel modo di **interagire** con la parte centrale.
 - Alcune periferiche sono “dentro” la scatola (*case*) che racchiude il computer, altre sono “fuori”. Questo non ne cambia il funzionamento.



La memoria principale

- La memoria principale (RAM):
 - Insieme al processore costituisce il cuore del computer
 - Immagazzina i programmi che istruiscono il processore sull'elaborazione da svolgere, e i dati da elaborare.
- Programmi e dati sono in formato digitale.
 - Sono quindi rappresentati come sequenze di bit.
- Quindi la memoria principale deve permettere di:
 - **Registrare** sequenze di bit.
 - **Recuperare** una sequenza di bit precedentemente registrata.

RAM: hardware



- Per memorizzare un bit usa due diverse cariche elettriche: una carica per lo 0 e una carica per l'1.
 - Memoria **volatile** → quando si toglie la corrente si perdono le informazioni.
- La memoria è suddivisa in **celle** da 8 bit (1 byte) ciascuna.
 - Le **celle** sono l'unità minima di lettura e scrittura.
 - Sono numerate consecutivamente 0,1,2,....
 - Il numero associato ad una cella è chiamato **indirizzo** della cella.

Dimensioni della RAM

- La **dimensione** della RAM esprime il numero di celle disponibili.
 - Si misura quindi in (Mega/Giga) Byte.
- Su cosa **influisce** la dimensione della RAM?
 - Sulle dimensioni dei **programmi** e dei **dati** che si possono caricare contemporaneamente in memoria.
 - Un computer con **poca RAM** è tipicamente **lento** perché deve continuamente liberare memoria per fare posto.
 - Se è **troppo poca** può diventare talmente lento da essere inutilizzabile.
 - D'altra parte, **troppa RAM** è **inutile** (resta vuota).
- I PC attuali hanno RAM intorno a 4GB.
 - Computer che devono sopportare elaborazioni pesanti possono averne molta di più.

Spazio di indirizzamento

- Il processore può:
 - **Scrivere** una sequenza di 8 bit in una cella di memoria principale.
 - **Leggere** una sequenza di 8 bit da una cella di memoria principale.
- In entrambi i casi deve specificare l'indirizzo della cella.
 - L'indirizzo è un **numero** → **sequenza di bit**.
 - Per ragioni di rapidità il processore deve poterlo comunicare in “un colpo solo” alla memoria → l'indirizzo deve avere una **lunghezza prefissata**.
 - Limitando tale lunghezza, limito anche il numero di indirizzi che posso esprimere.
 - Es.: indirizzo 32 bit → 2^{32} celle = 4 GB.
 - Il numero di celle indirizzabili è detto **spazio di indirizzamento**.
 - **Gli attuali PC usano 32 bit (max 4 GB di memoria) o 64 bit per gli indirizzi.**

Tempo di accesso

- La **dimensione della RAM** è un parametro **importante**.
- Altro parametro importante → **tempo di accesso**.
 - Il tempo che richiede leggere o scrivere in una cella.
 - Caratteristica importante della RAM: tale tempo **non dipende** dalla posizione della cella.
 - Accedere alla cella 2 richiede lo stesso tempo che accedere alla cella 43826798.
 - Questa caratteristica si chiama **accesso casuale** (da una traduzione dell'inglese **random access**).
 - Infatti: RAM = **Random Access Memory**.
 - C'è da dire che difficilmente all'acquisto di un PC si può scegliere il tempo di accesso della RAM!
 - Tempi di accesso negli attuali PC: **10-60 nanosecondi** .

Acquistare la RAM

- Normalmente un PC si acquista già dotato di RAM.
- È tuttavia possibile **espandere la RAM** .
 - Aggiungendo una scheda di memoria se c'è posto.
 - Solitamente i PC hanno posto per 2/4 schede.
 - Sostituendo una delle schede presenti con una più grande.
- Per fare ciò bisogna fare attenzione alla **compatibilità**:
 - Esistono vari modelli di RAM con diversi parametri fisici che potrebbero non essere compatibili con quelli del proprio PC.
- Attualmente i costi sono più o meno...
 - Schede da 2 GB → 15/50 €
 - Schede da 4 GB → 20/60 €

Memoria centrale (Principale)

bit 1 cifra binaria memorizza 0 oppure 1

byte 8 bit memorizza un carattere

Kilobyte (KB) 1024 byte circa mezza pagina di testo

Megabyte (MB) 1024 KB un libro di 200 pagine

Gigabyte (GB) 1024 MB alcuni volumi

Terabyte (TB) 1024 GB una biblioteca

Petabyte (PB) 1024 TB molte biblioteche

RAM: ricapitolando

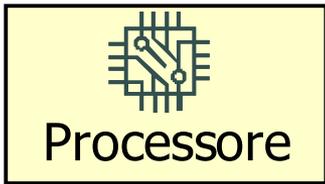
- Memorizza programmi e dati.
- La sua **dimensione** influisce sulle prestazioni del computer.
 - Se il PC è lento non è però detto che sia colpa della RAM.
 - Anzi, se la RAM ha già una dimensione ragionevole aumentarla potrebbe non servire a nulla.
- Per gli attuali PC una dimensione tipica è **2GB, 4GB o 8GB.**
 - Dipende anche dal tipo di programmi che si vogliono usare sul PC!
 - Tipicamente **richiedono molta RAM** l'elaborazione **multimediale** (nell'ordine immagini, audio, filmati e animazioni) e i **videogiochi**.

L'unità centrale: altri tipi di memoria

- Memoria di sola lettura (ROM):
 - Non può essere modificata.
 - A differenza della RAM *non* è volatile.
 - Veloce quasi come la RAM.
 - Contiene le informazioni di inizializzazione usate ogni volta che si accende l'elaboratore (**bootstrap + bios**).

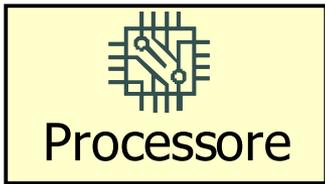
Famiglie di memoria

- SRAM (RAM statiche):
 - Costi elevati.
 - Tempi di accesso ridotti.
 - Normalmente usate per la memoria cache.
- DRAM (RAM dinamiche):
 - Tempi di accesso più della SRAM (40-60 nanosecondi).
 - Normalmente usate per la memoria principale.
 - SDRAM: DRAM sincrono, un clock (che ha una frequenza misurata in MHz) sincronizza l'accesso alla memoria.
 - DDR: Double Data Rate, raddoppia la velocità di trasferimento di dati tra la memoria e il processore.
- VRAM (RAM video): usate per la gestione delle immagini.



Il processore

- Il processore (CPU - Central Processing Unit):
 - Insieme alla RAM costituisce il “cuore” del computer.
 - Esegue le **istruzioni** contenute nei **programmi** memorizzati in RAM.
 - I programmi: sequenze di istruzioni per l’elaborazione delle informazione
 - Definiscono quale debba essere il comportamento del processore.
 - I computer eseguono *deterministicamente* istruzioni.
 - Tali istruzioni per essere comprese dalla CPU devono essere scritte in **linguaggio macchina**.
- Il processore lavora per **cicli di esecuzione**.
 - Ad ogni ciclo:
 - **Recupera dalla RAM la prossima istruzione da eseguire.**
 - **Esegue l’istruzione.**



Il processore

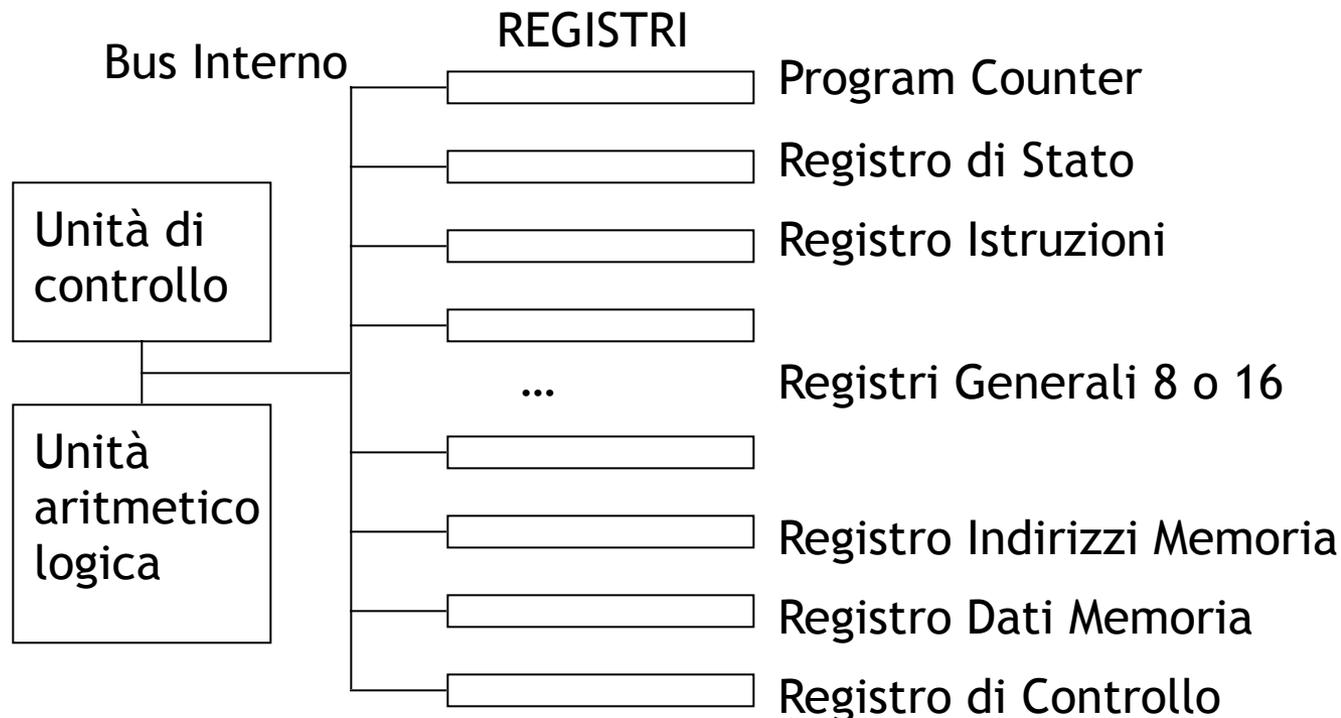
Quindi il processore

Esegue i programmi in linguaggio macchina!

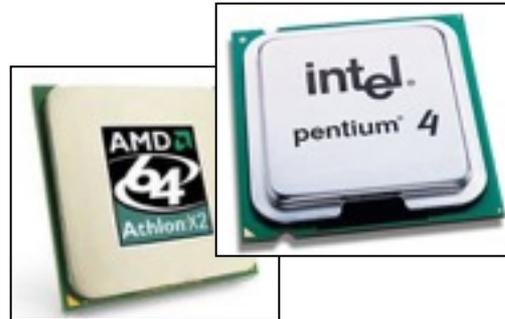
I programmi sono sequenze di istruzioni espresse in qualche linguaggio di programmazione (es. Java, C, LISP etc.)

CPU: hardware

- La CPU non è un unico componente ma è costituita da componenti diversi che svolgono compiti diversi



CPU: hardware



- Gli elementi hardware che compongono il processore:
 - **Unità di controllo** → coordina le attività del processore.
 - **Registri** → piccole unità di memoria (2-8 byte) molto veloci su cui è possibile eseguire direttamente i calcoli.
 - **Registri speciali** adibiti ad informazioni particolari. Distinguiamo in particolare:
 - **Program counter** → contiene l'indirizzo in memoria della prossima istruzione da eseguire.
 - **Registro istruzione** → contiene l'istruzione da eseguire attualmente.
 - **Registro di stato** → contiene informazioni sullo stato di esecuzione e su eventuali errori.
 - **Registri generici che contengono i dati su cui le istruzioni devono essere eseguite.**
 - **Unità logico-aritmetica (ALU)** → esegue calcoli aritmetici e test logici (Es. somma, confronto, etc).

CPU: funzionamento

- L'unità di controllo scandisce i cicli **di esecuzione** della CPU grazie al clock (orologio).
- Ad ogni ciclo:
 1. **Recupera** dalla memoria l'istruzione che si trova all'indirizzo segnalato dal **Program Counter**.
 - L'istruzione viene trasferita dalla memoria nel **Registro Istruzione**.
 2. **Incrementa** di 1 il valore del **Program Counter**.
 - Al prossimo ciclo si leggerà l'istruzione successiva.
 3. Legge il **Registro Istruzione** e ne **interpreta** il contenuto per decidere il da farsi.
 - Che tipi di istruzioni?

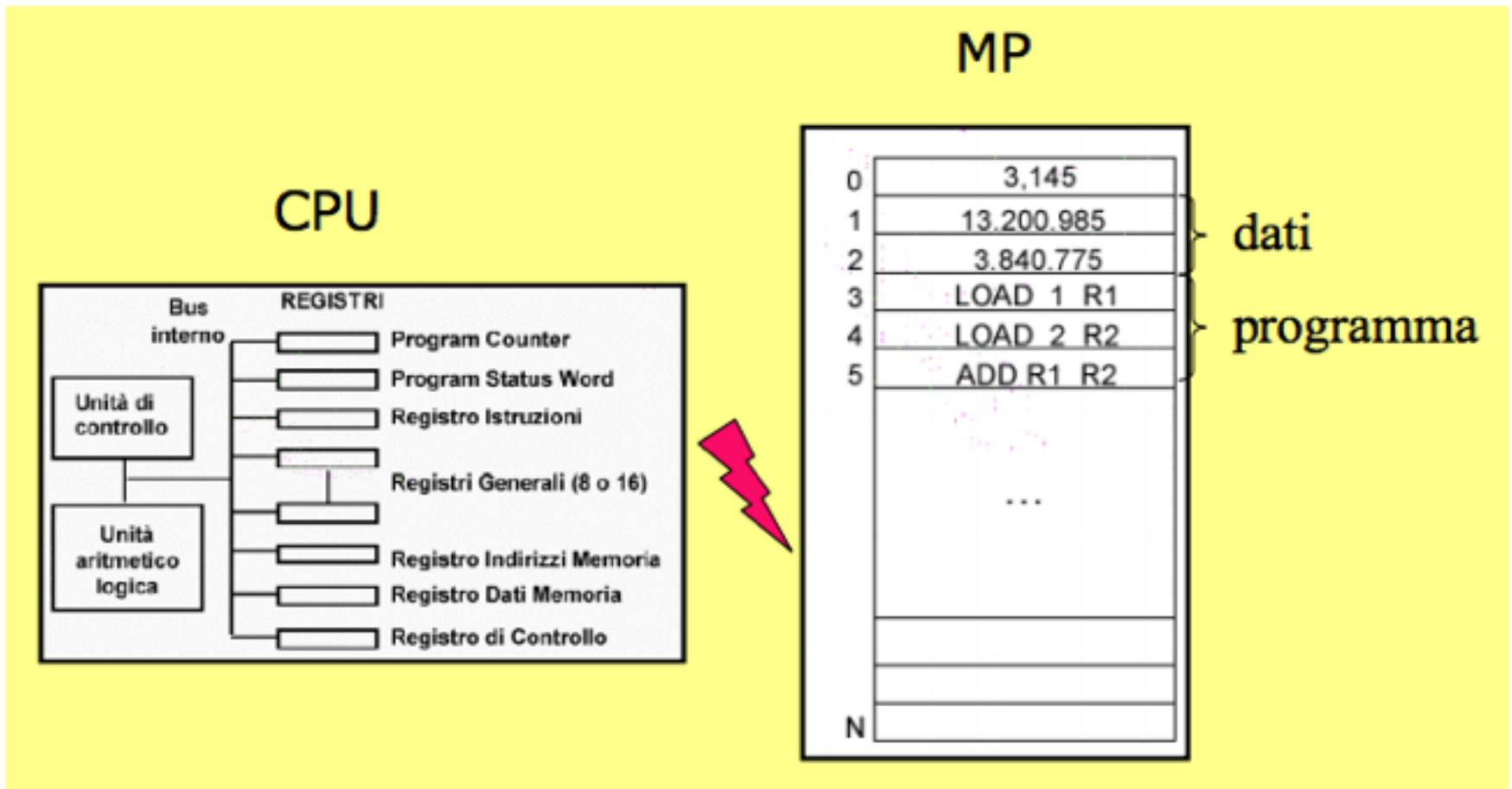
CPU funzionamento

- Che tipi di istruzioni esegue un processore? 4 tipi di istruzioni:
 - a. **Lettura** di un dato dalla memoria.
 - b. **Scrittura** di un dato in memoria.
 - c. **Calcolo** aritmetico logico su dati precedentemente prelevati.
 - d. **Salto condizionale** in un punto diverso del programma.

CPU funzionamento

- Che tipi di istruzioni esegue un processore? Alcuni esempi:
 - **leggi** la parola in RAM all'indirizzo X e mettila nel registro R5
 - **scrivi** il contenuto del registro R1 nella parola di memoria all'indirizzo y
 - **somma** il contenuto dei registri R3 e R4 mettendo il risultato nel registro R8
 - **inverti** i bit del registro R6 mettendo il risultato in R2

CPU funzionamento



Salto condizionale

- Per quanto abbiamo visto sino ad ora:
 - La CPU esegue le istruzioni nell'ordine esatto in cui sono registrate in memoria.
 - Infatti il Program Counter viene incrementato di 1 ad ogni ciclo di esecuzione.
- L'istruzione di salto condizionale permette di modificare questo ordine.
- Il salto condizionale:
 - Scrive nel Program Counter un indirizzo di memoria diverso da quello programmato, permettendo di saltare ad un'istruzione diversa (salto...).
 - Antepone a tale operazione una verifica su un dato (es.: il contenuto del registro 5 è maggiore di 2?) e la esegue solo se il test ha risultato positivo (...condizionale).

Effetti del salto condizionale

- L'istruzione di **salto condizionale** è ciò che rende i computer così potenti.
- In particolare permette di:
 1. Scrivere istruzioni diverse per trattare circostanze diverse.

11: ...

12: Carica dall'indirizzo di memoria 250 al registro A7

13: Dividi per due il contenuto di A7 e metti il **resto** della divisione in A8

14: Se il contenuto di A8 è 0 vai all'**istruzione 16**

15: Se il contenuto di A8 è 1 vai all'**istruzione 18**

16: moltiplica il contenuto di A7 per 2 e metti il risultato in A9

17: Vai all'**istruzione 19**

18: Moltiplica il contenuto di A7 per 3 e metti il risultato in A9

19: ...

Cosa contiene A9?

Effetti del salto condizionale

- L'istruzione di **salto condizionale** è ciò che rende i computer così potenti.
- In particolare permette di:
 1. Scrivere istruzioni diverse per trattare circostanze diverse.

11: ...

12: Carica dall'indirizzo di memoria 250 al registro A7

13: Dividi per due il contenuto di A7 e metti il **resto** della divisione in A8

14: Se il contenuto di A8 è 0 vai all'**istruzione 16**

15: Se il contenuto di A8 è 1 vai all'**istruzione 18**

16: moltiplica il contenuto di A7 per 2 e metti il risultato in A9

17: Vai all'**istruzione 19**

18: Moltiplica il contenuto di A7 per 3 e metti il risultato in A9

19: ...

Cosa contiene A9?

...ora A9 contiene $2N$ se N è pari e $3N$ se N è dispari (dove N è il numero all'indirizzo 250)

Effetti del salto condizionale

- L'istruzione di **salto condizionale** è ciò che rende i computer così potenti.
- In particolare permette di:
 1. Scrivere istruzioni diverse per trattare circostanze diverse.
 2. Ripetere più volte una stessa sequenza di istruzioni
 3. Ripetere una sequenza di istruzioni fino a che un certo requisito è soddisfatto
- Quest'ultima possibilità è **determinante** per le capacità di un computer.

11: ...

12: Carica dall'indirizzo di memoria 250 al registro A7

13: Metti nel registro A6 il numero 1

14: Moltiplica per 2 il contenuto di A6 e rimetti il **risultato** della moltiplicazione in A6

15: Togli 1 dal contenuto di A7

16: Se il contenuto di A7 è maggiore di 0 vai all'istruzione 14

17: ...

Cosa contiene A6?

Linguaggio Macchina

- L'insieme delle istruzioni che una CPU esegue viene svolto in Linguaggio Macchina.
 - Processori di produttori diversi hanno linguaggi macchina diversi.
 - le istruzioni valide per un PC possono non esserlo per un altro!
 - Astrazione fornita dai linguaggi di programmazione ad alto livello
- Si è comunque abbastanza imposto come standard il linguaggio macchina dei PC IBM.
 - Processori IBM compatibili (Intel, AMD).
- I processori **Motorola**, montati dai computer Apple, avevano un linguaggio diverso.
 - Ora anche Apple monta gli Intel.
- Altri tipi di processori esistono per computer non “personal”.

Frequenza della CPU

- Parametro fondamentale: frequenza della CPU.
- Misura il **numero di cicli di esecuzione** che una CPU è in grado di eseguire in un secondo.
 - Si misura in **Hertz** (1 Hz = 1 ciclo al secondo).
 - Meglio: in **Mega/Giga Hertz** (MHz/GHz):
300, 400 MHz sono circa 300/400 milioni di cicli (istruzioni) al secondo;
1 GHz corrisponde circa a un miliardo di istruzioni al secondo
- Ovviamente la frequenza determina la **velocità a cui vengono eseguite le istruzioni**.
- In generale processore a **maggiore frequenza** → PC più veloce.
 - I processori per PC attuali hanno una frequenza di **3-4 GHz** (miliardi di di istruzioni macchina al secondo).
 - Alcuni processori sopperiscono ad una frequenza inferiore (2 GHz) con una architettura più avanzata.

La memoria cache

- Una componente accessoria delle CPU.
- Non è indispensabile ma ormai c'è su tutte le CPU.
- La **memoria cache**:
 - È una **via di mezzo** fra la memoria principale e i registri.
 - La memoria principale è **poco costosa** ma **troppo lenta**.
 - I registri sono **molto veloci** ma **troppo costosi**, e quindi non se ne possono mettere troppi.
 - Usata come **“deposito intermedio”** per dati che vengono usati molto di frequente ma devono essere tolti dai registri per ragioni di spazio.
 - “Cache” deriva dal francese per **“nascondere”**.
- Una memoria cache di dimensioni maggiori **velocizza il PC → parametro importante**.
 - Cache attuali PC: circa 2-6MB, 8MB nei più avanzati.

Il bus

- Si chiamano **bus** i gruppi di cavi che trasferiscono i dati da una parte all'altra.
 - Dalla memoria alla CPU e viceversa.
 - All'interno della CPU fra le varie componenti.
- Perché si chiamano bus?
 - Perché portano in giro i bit!
- Un cavetto per ciascun bit: trasporta la carica elettrica appropriata (bassa: 0, alta: 1).
- La **dimensione del bus** stabilisce quanti bit si possono trasferire in un colpo solo.
 - Influisce sulla velocità del PC!

La scheda madre

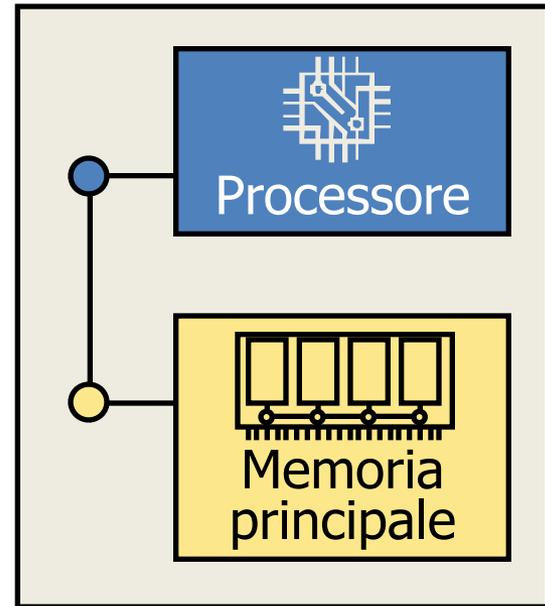
- La scheda madre (motherboard) è la base su cui sono montati CPU e RAM, collegati fra loro dai bus.



Acquistare una CPU

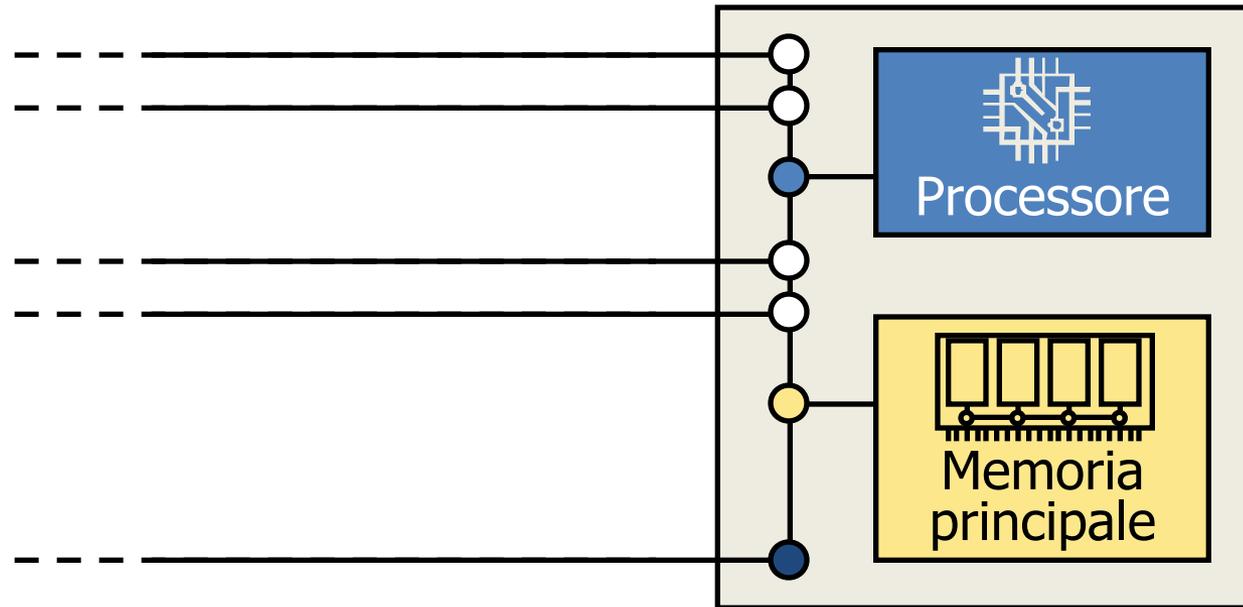
- Quando si acquista un PC ha già una CPU al suo interno!
- Parametri della CPU da verificare:
 - **Frequenza** → influisce direttamente sulla velocità, ma in alcuni casi la tecnologia può sopperire ad una frequenza inferiore.
 - **Dimensione memoria cache** → influisce sulla velocità.
 - **Dimensione della word** (e quindi dei bus) → influisce sulla velocità e sulle capacità di calcolo.
- La CPU si può sostituire, ma è meglio affidarsi ad un esperto.
 - Solitamente se una CPU è troppo vecchia conviene cambiare l'intero PC (eventualmente a esclusione del video).

Periferiche



- Abbiamo sinora visto il cuore del computer:
 - La **scheda madre** su cui sono montati **RAM** e **CPU** (unità centrale).

Periferiche



- Abbiamo sinora visto il cuore del computer:
 - La **scheda madre** su cui sono montati **RAM** e **CPU** (unità centrale).
- Passiamo ora ad analizzare le **periferiche**.
 - Analizziamo innanzitutto gli elementi comuni a tutte le periferiche.
 - Come si **collegano** al cuore del computer? Come **comunicano** con il processore?

Collegamento: le porte

- **Collegamento fisico** → tramite le **porte** (anche dette **interfacce**).
 - Le porte sono le “prese” a cui si collegano i dispositivi.
 - Sono attaccate alla scheda madre.
 - Alcune sono **esterne** (visibili al di fuori del case), alcune sono **interne**.
 - **Periferiche esterne** → si possono attaccare/staccare, maneggiabili dall’utente.
 - **Periferiche interne** → sono pensate per non essere scollegate; si possono però sostituire con relativa facilità.
 - Diversi tipi di porte; si differenziano per:
 - **Quantità di informazioni** che lasciano passare (numero di bit).
 - **Velocità di trasferimento** delle informazioni.
 - Alcune porte permettono anche di **alimentare** i dispositivi.

Tipi di porte

- Ciascun tipo di porta ha una presa riconoscibile a cui corrisponde uno specifico connettore.



- Il cavo porta avanti e indietro un certo numero di bit.
- Possiamo distinguere due famiglie di porte:
 - Porte **parallele** → i bit che arrivano dalla periferica vengono trasferiti contemporaneamente al PC (e/o vice versa).
 - Porte **seriali** → i bit che arrivano dalla periferica vengono trasferiti uno per volta al PC (e/o vice versa).

Porte in uso sul PC

- Principali porte in uso sul PC:
 - **USB (Universal Serial Bus).**
 - Porta parallela.
 - Porta seriale RS-232.
 - Porta PS-2.
- Meno diffusa:
 - Porta SCSI.
- Sui portatili:
 - Porte PCMCIA.
- Porte senza fili:
 - Raggi infrarossi / Bluetooth.
- Porte per connessione in rete
 - Modem, Ethernet, Wi-Fi,...

Stanno venendo
rimpiazzate tutte dalla
USB!

La porta USB è **veloce**,
permette di **alimentare** il
dispositivo, ed è **plug & play**.

Plug & play → il PC riesce a
riconoscere da solo che la
periferica è attaccata, e permette
di attaccarla/staccarla mentre è
acceso.

Comunicazione

- Le periferiche:
 - Operano in modo asincrono rispetto al processore.
 - **Input** → Il processore non è in grado di prevedere e di controllare il momento in cui un dato di input sarà a disposizione.
 - **Output** → Il processore non può prevedere il momento in cui un dispositivo in output avrà terminato di produrre i dati in uscita.

Si dice che sono schiave del processore tecnicamente, si parla di gestione **master-slave**: è il processore che deve coordinare le attività di tutti i dispositivi

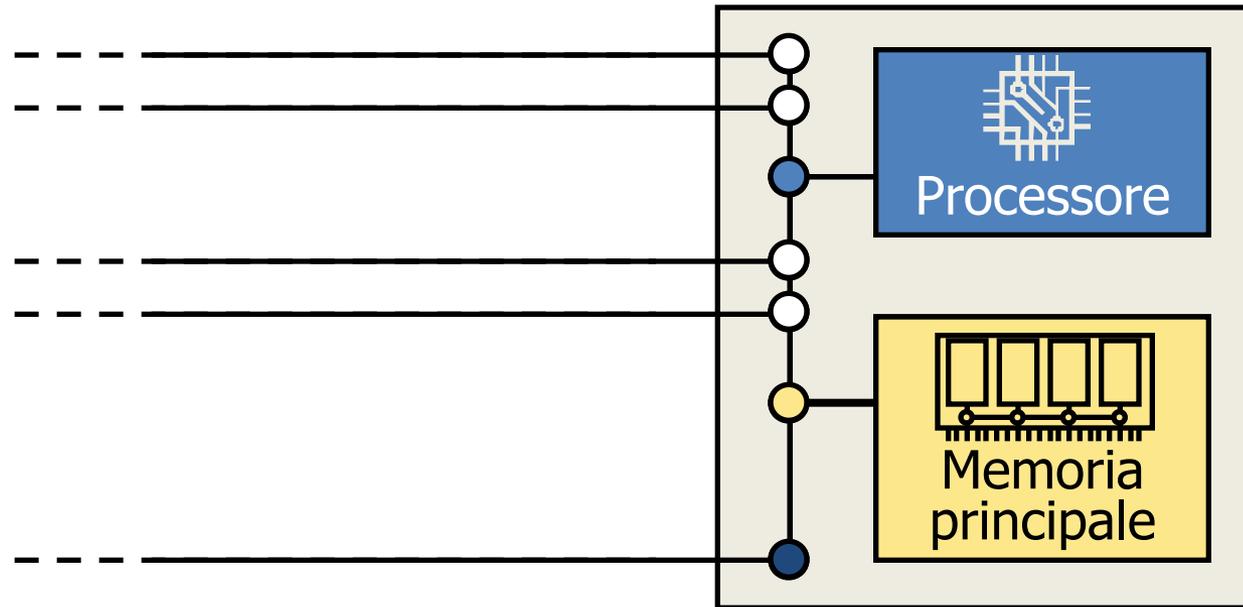
Trasferimento delle informazioni

- Come fanno le periferiche a **scambiare informazioni** con il processore?
 - A **ciascuna porta sono associati degli spazi di memoria** sulla scheda madre.
 - Essi sono contraddistinti da **indirizzi particolari**.
 - **Comunicazione da periferica a processore**:
 - Tramite la connessione, la periferica scrive le informazioni in questo spazio di memoria.
 - Avverte il processore, **tramite un apposito segnale**, che il dato è disponibile.
 - **Comunicazione da processore a periferica**:
 - Il processore scrive le informazioni nello spazio di memoria.
 - Tramite un segnale attiva la connessione via cavo che trasferisce le informazioni alla periferica.

Interrupt

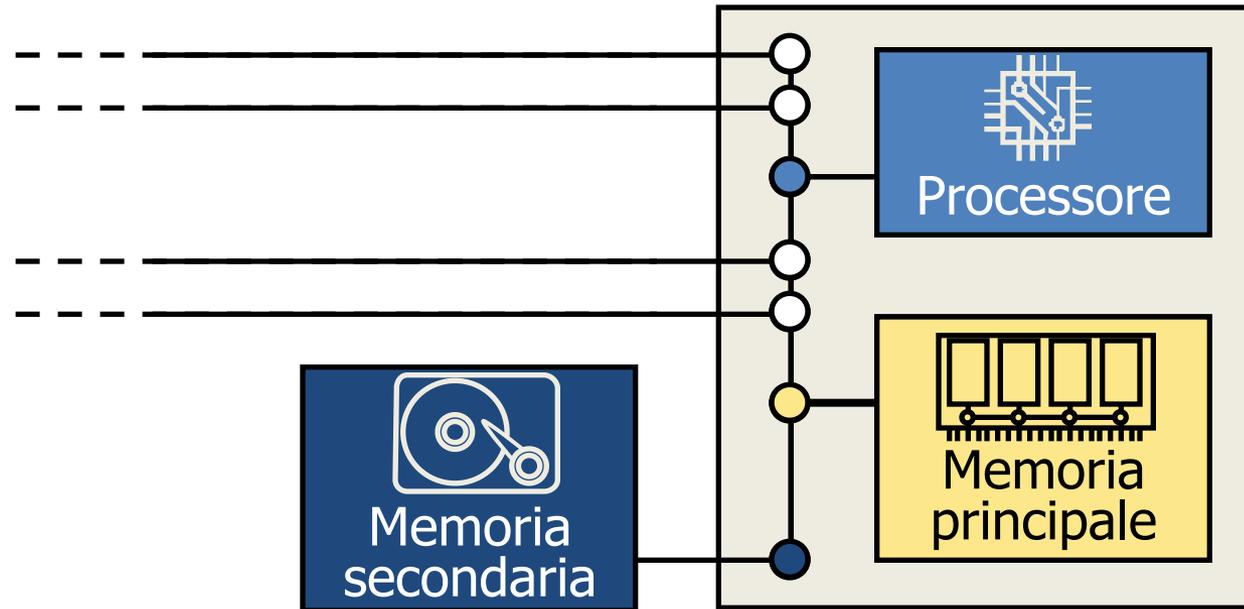
- Si chiama **interrupt** il segnale che le periferiche inviano al processore quando hanno trasferito un dato che vogliono venga considerato.
 - **Interrompe** l'attività che il processore stava eseguendo.
 - Lo “costringe” a eseguire un programma speciale che gestisce la periferica.
 - Questo programma si chiama **driver**.
 - Deve essere presente nel computer (installato) perché la periferica possa funzionare.
 - Una volta terminata l'esecuzione del driver, il processore torna a quello che stava facendo prima.

Limiti della RAM



- La RAM è **volatile**, ossia si **cancela** quando si **spegne** il computer.
- È inoltre **relativamente piccola** (in quanto **costosa**):
 - Riesce a contenere i programmi in esecuzione in un dato momento e i relativi dati.
 - **Non riesce a contenere tutti i programmi che l'utente desidera poter eseguire in momenti diversi, ne' i tutti i suoi dati.**

La memoria secondaria



- Si aggiunge quindi una **memoria secondaria**.
- Rispetto alla RAM, la memoria secondaria è:
 - **Permanente**: evita di dover reinserire ogni volta dati e programmi.
 - **Meno costosa, maggiori dimensioni**.
 - **Più lenta**: non è un problema, perché i programmi da eseguire e i dati da elaborare vengono di volta in volta trasferiti in RAM.

Memorie secondarie più diffuse

- **HARD DISK** (disco fisso):
 - È la memoria secondaria per eccellenza.
 - “Magazzino” sempre disponibile per programmi e dati.
 - Memoria di tipo magnetico.
 - Permette di leggere e scrivere dati liberamente.



Acquistare un hard disk

- All'acquisto del PC è solitamente possibile scegliere la **dimensione** dell'hard disk.
 - La dimensione influisce sulla **quantità di dati e programmi che si possono immagazzinare**.
- Attualmente i PC hanno tipicamente circa **500 GB di hard disk**.
 - Hard disk più grossi servono solo per esigenze particolari.
 - Es.: memorizzazione di grandi quantità di dati o di **dati molto “grandi”, tipo filmati, immagini di grandi dimensioni**, etc.
- Con il passare del tempo anche i **programmi** tendono a crescere.
 - Di conseguenza in futuro serviranno probabilmente hard disk più grandi.

Interfaccia fra HD e CPU

- Collegamento e comunicazione seguono i principi visti per le **periferiche**.
 - Gli hard disk sono a tutti gli effetti della periferiche!
- Gli hard disk sono solitamente **interni** al PC.
 - Si collegano tramite una porta interna (**interfaccia Serial ATA**) non visibile all'utente.
- Esistono anche **hard disk esterni** venduti separatamente.
 - Usati per aumentare la memoria di massa del proprio PC.
 - Solitamente collegati tramite **porta USB**.

Memorie secondarie più diffuse

- CD e DVD (dischi ottici):
 - Si spostano con facilità fra computer diversi.
 - Serve un dispositivo (**lettore**) per leggerli.
 - La scrittura è un po' problematica (richiede una apparecchiatura apposita → **masterizzatore**).
 - Usati per:
 - Archiviare stabilmente informazioni.
 - Distribuire dati e/o programmi che servono a più PC.
 - Memoria di tipo **ottico**.



Altre memorie secondarie

- **Nastri magnetici**
 - Usati per archiviare grandi quantità di informazioni non più in uso.
- **Chiavi USB**
 - Funzionano come piccoli Hard-Disk portatili.
 - La tecnologia è però diversa.
- **Compact Flash**
 - Schede di memoria usate principalmente per fotocamere digitali.
 - Ne esistono diversi tipi.
 - Per essere usate con il PC richiedono un lettore/scrittore.

Tipologie di accesso ai dati

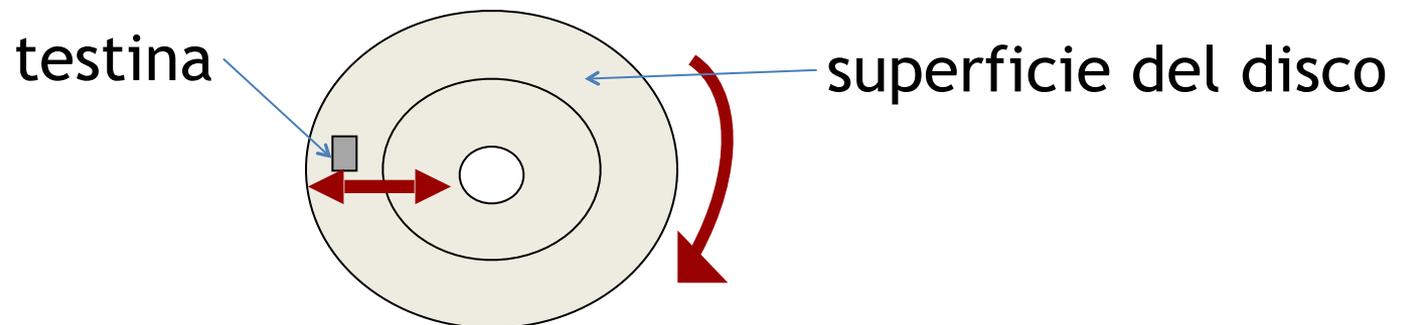
- Accesso → termine generico per lettura/scrittura.
- Accesso sequenziale vs. accesso diretto o uniforme.
 - Accesso **sequenziale** → per accedere alle informazioni in un certo punto devo scorrere tutte quelle precedenti.
 - Il tempo di accesso ad un dato dipende dalla sua posizione.
 - Per molti versi sconsigliato ma più semplice da gestire.
 - Esempi: nastri magnetici, ma anche CD e DVD.
 - Accesso **diretto** → posso accedere direttamente alle informazioni in un punto qualunque.
 - Il tempo di accesso è **uniforme** per tutti i dati (come nel caso della RAM).
 - Esempi: i dischi magnetici.

Indirizzamento in Memoria Secondaria

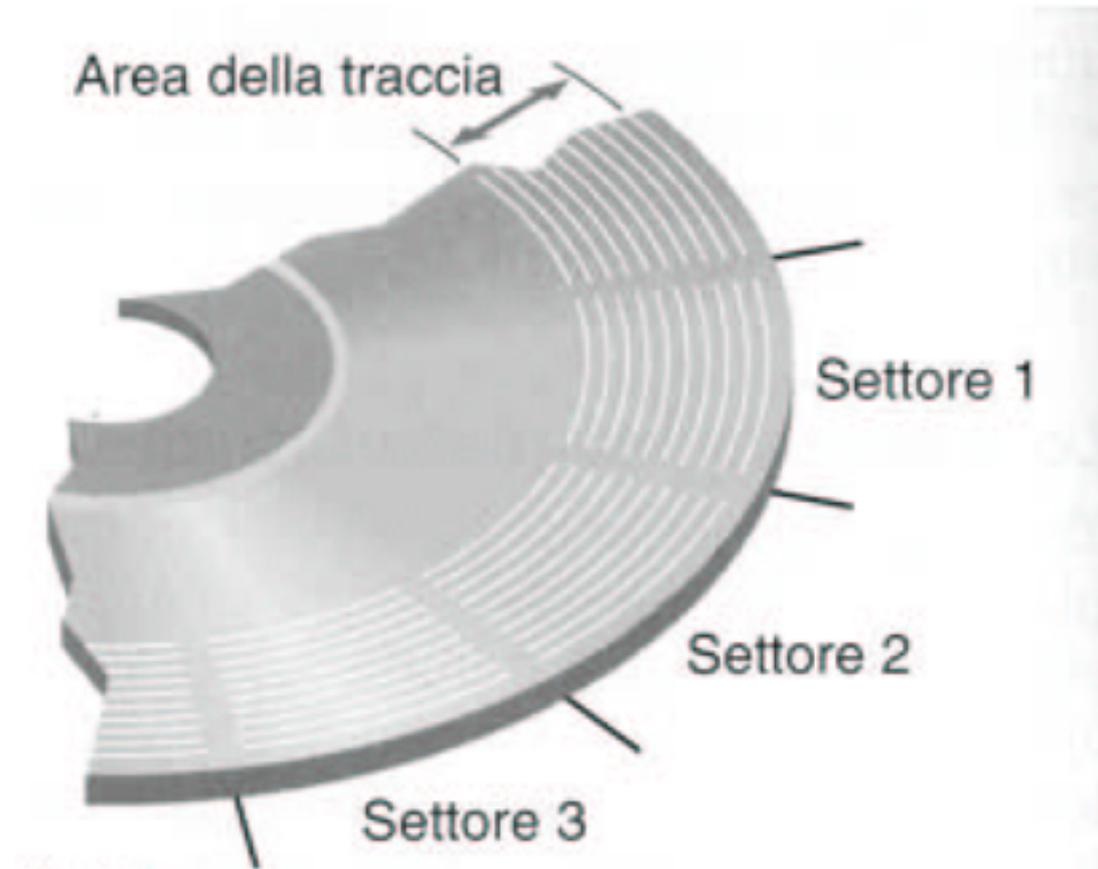
- Come per la RAM, per accedere alla memoria **secondaria** si deve specificare un **indirizzo**.
 - Unità indirizzabili in RAM → **celle**.
 - Unità indirizzabili in memoria secondaria → **blocchi**.
- I blocchi sono **molto più grandi** delle celle.
 - **Cella: 1B / Blocco: 1 KB**.
 - **Motivo 1:** la memoria secondaria è tendenzialmente **più grossa** della RAM.
 - Se le unità indirizzabili sono molto piccole, vuol dire che ce ne sono tantissime.
 - Lo spazio di indirizzamento è enorme.
 - Per scrivere un indirizzo servono tantissimi bit!
 - **Motivo 2:** la lettura/scrittura è **più lenta** che in RAM, **meglio trasferire più dati (dalla memoria secondaria alla RAM) in una operazione unica**.

Funzionamento dei dischi magnetici (1)

- Supporti di plastica, vinile o metallo, su cui è depositato **del materiale magnetizzabile**.
 - La polarizzazione può essere di due tipi: **positiva (bit = 1)** e **negativa (bit = 0)**.
 - La magnetizzazione è **permanente** fino a quando non viene modificata per effetto di un agente esterno.
 - L'apparecchiatura di scrittura è in grado di **cambiare** la polarizzazione di ciascuna unità.
- Meccanismo di **lettura/scrittura**:
 - Il disco viene mantenuto ad una **velocità di rotazione costante**.
 - Quando si deve leggere/scrivere, una **testina** viene posizionata sul punto desiderato.
 - Il disco gira; la testina sposta solo in senso radiale.



Organizzazione dati su disco magnetico



Funzionamento Dischi Magnetici

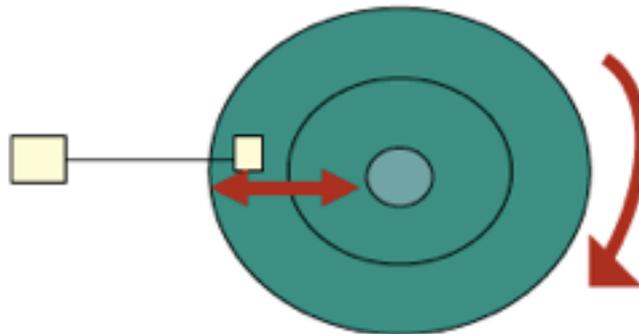
I dischi sono suddivisi in:

- Ⓡ **tracce** concentriche
- Ⓡ **settori** ossia “fette” di disco.

- Ⓡ I settori suddividono ogni traccia in porzioni dette **blocchi**
- ó Per indirizzare un blocco è sufficiente specificare la sua traccia e il suo settore (es.: sett. 2, traccia 5)

- ó **Formattazione**: operazione con cui la superficie di un disco viene suddivisa in tracce e settori.

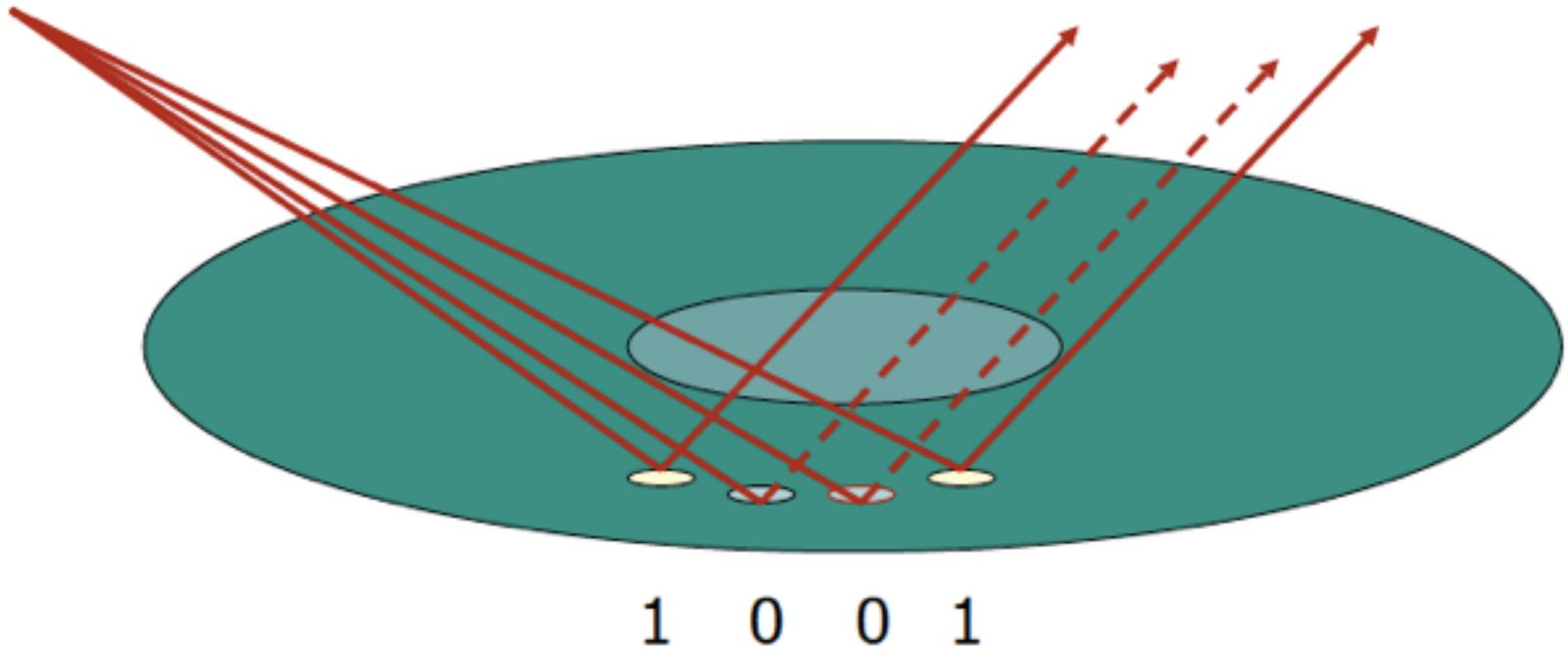
-
- * I dischi magnetici consentono l'accesso **diretto**
 - ◆ È possibile posizionare direttamente la testina su un qualunque blocco (noto il numero della traccia e il numero del settore)
 - ◆ Per effettuare un'operazione di lettura/scrittura la testina deve "raggiungere" il blocco desiderato
 - ◆ Il disco gira; la testina sposta solo in senso radiale



Funzionamento dei dischi magnetici (3)

- **Accesso diretto ai dati.**
 - Data una traccia e un settore, la testina si posiziona direttamente sul blocco corrispondente.
- **Tempo di accesso dipende da:**
 - **Seek time:** tempo necessario alla testina per raggiungere la traccia desiderata.
 - **Latency time:** tempo che si deve aspettare perché il settore desiderato passi sotto alla testina.
 - Dipende dalla **velocità di rotazione** del disco.
 - **Tempo di trasferimento:** tempo necessario a trasferire i bit del blocco desiderato dal disco alla RAM o viceversa.
 - $\text{Tempo di trasferimento} = \text{seek time} + \text{latency time} + \text{tempo di lettura/scrittura}$.
- Nei dischi attuali il tempo di accesso è dell'ordine dei **millisecondi**.

Memoria ottica



Si usa il raggio laser e si sfrutta la riflessione della luce

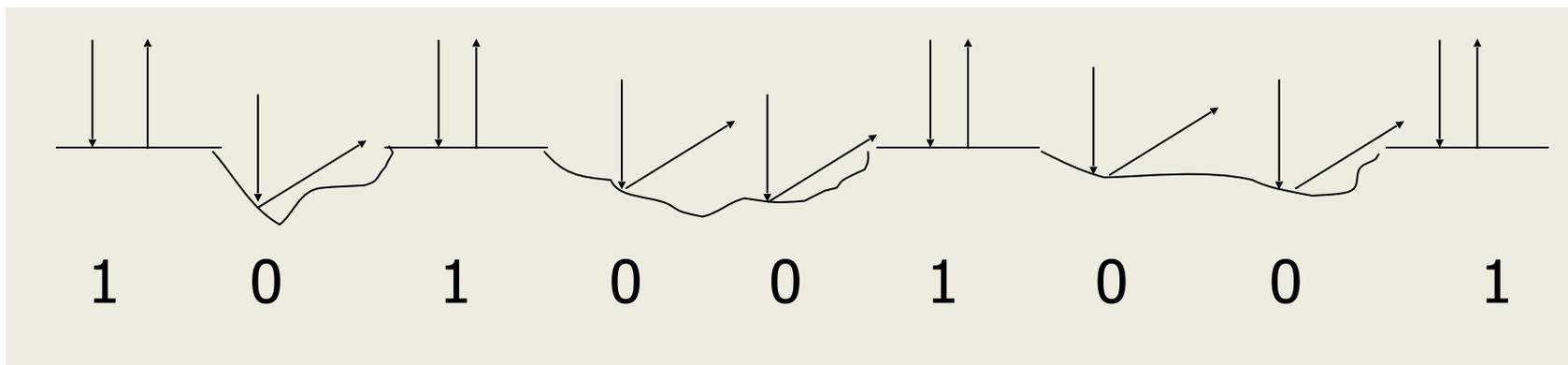
Ⓜ sul supporto ci sono dei piccoli forellini che formano zone trasparenti (lasciano passare la luce) e zone opache (riflettono la luce)

Ⓜ Si dirige un raggio laser sul supporto e si vede se viene riflesso: no riflessione = 0; riflessione = 1

ó I bit vengono memorizzati in un'unica traccia a spirale

Funzionamento dei dischi ottici (1)

- Si usa il **raggio laser** e si sfrutta la **riflessione** della luce.
 - Sul supporto ci sono dei piccoli forellini che formano zone **trasparenti** (lasciano passare la luce) e zone **opache** (riflettono la luce).
 - Si dirige un raggio laser sul supporto e si vede se viene riflesso: **no riflessione =0**; **riflessione = 1**



- I bit vengono memorizzati in un'unica **traccia a spirale**.
 - L'accesso è quindi **sequenziale**.

Funzionamento dei dischi ottici (2)

- **Scrivere** su un disco ottico è una operazione complessa.
- Ancora più complesso è **cancellarlo!**
- **Differenziazione di dispositivi:**
 - Il **lettore** permette solo di leggere su un disco ottico.
 - Il **masterizzatore** permette anche di **scriverci sopra** (eventualmente cancellando scritte precedenti).
- **Differenziazione di supporti:**
 - CD/DVD-**R** → possono essere scritti solo una volta.
 - CD/DVD-**RW** → possono essere scritti (e cancellati) più volte.
 - Ma non all'infinito, circa un migliaio, poi si deteriorano.

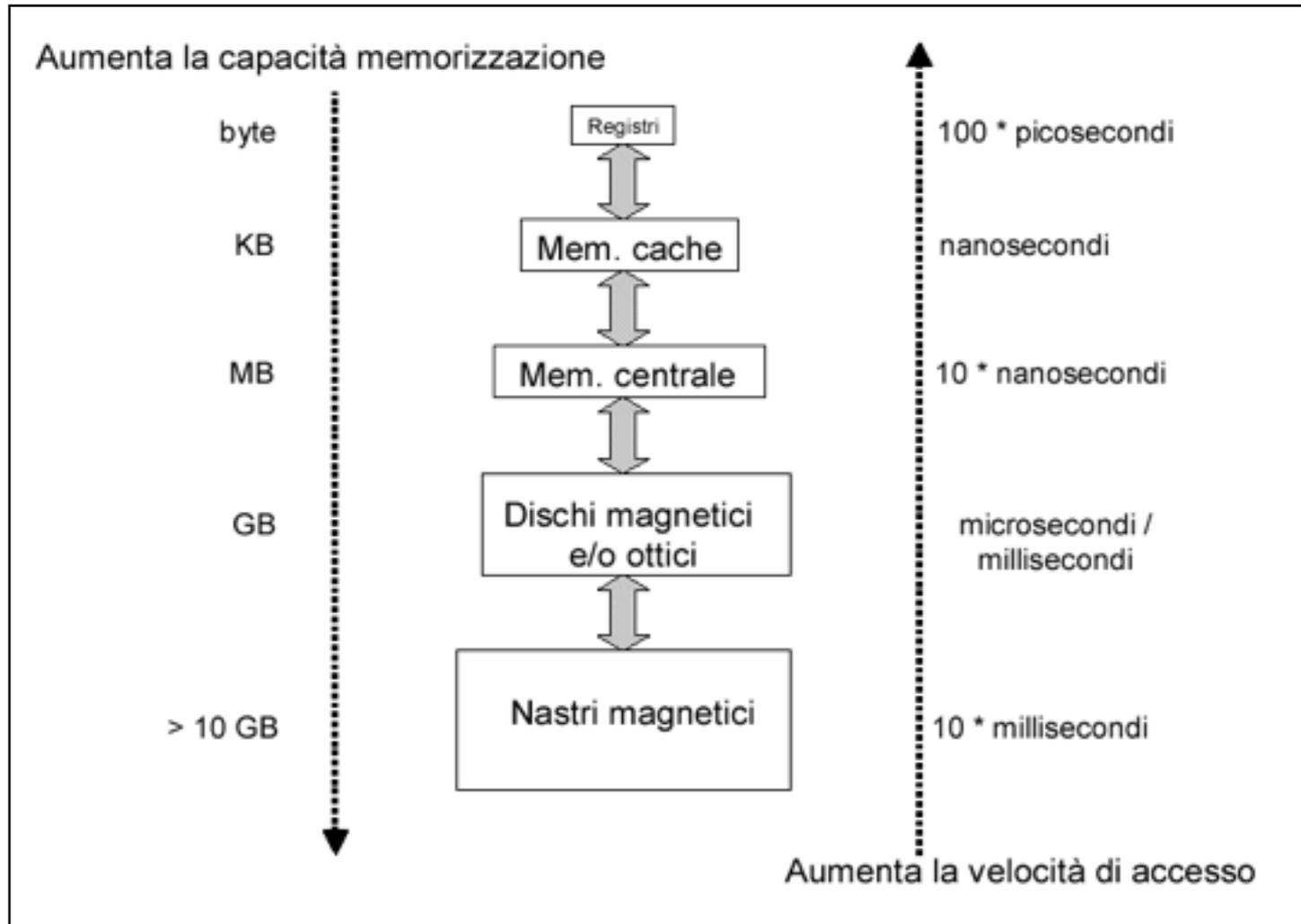
Tipi di dischi ottici

- Due famiglie principali:
 - **CD (Compact Disc)**
 - Capacità: 650-800 MB
 - **DVD (Digital Versatile Disc)**
 - Capacità: da 4,7 a 17 GB (normalmente circa 8,7 GB)
 - **Blu-Ray:**
 - Capacità: da 25 a 128 GB (normalmente circa 50 GB)
 - Il futuro? Esempio: **l'HVD (Holographic Versatile Disc)** può contenere TeraByte (migliaia di GB...) di dati.
- **Attenzione:** se cambia il supporto cambiano anche il lettore e il masterizzatore
 - Attualmente i lettore/masterizzatori per DVD leggono e masterizzano anche i CD.
 - Non vale il viceversa.
 - **All'acquisto** del PC si deve verificare quale tipo di dispositivo è incluso (lettore o masterizzatore, CD o DVD).
 - Le **velocità** di lettura/scrittura sono ormai tutte ragionevoli
 - Espresse con la dicitura S/R/L (es. 48X/24X/48X): S = scrittura, R = riscrittura, L = lettura

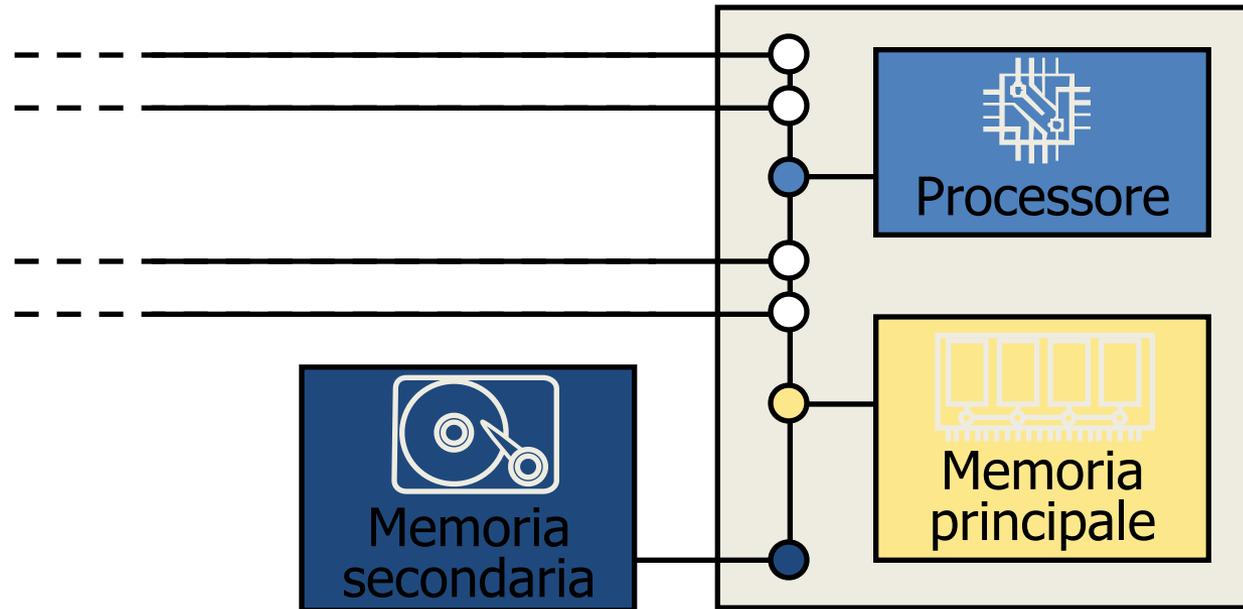
Interfaccia fra lettore/masterizzatore e CPU

- Anche i lettori/masterizzatori sono periferiche.
- Quelli interni al PC usano lo stesso tipo di interfaccia usato per gli hard disk.
- Anche in questo caso esistono lettori/masterizzatori esterni venduti separatamente.
 - Solitamente collegati tramite porta USB.
- Caso particolare: i computer portatili.
 - Molti portatili hanno il lettore/masterizzatore esterno perché il case è troppo piccolo per contenerlo.
 - In questi casi è presente una porta *ad hoc* per collegare questa periferica.

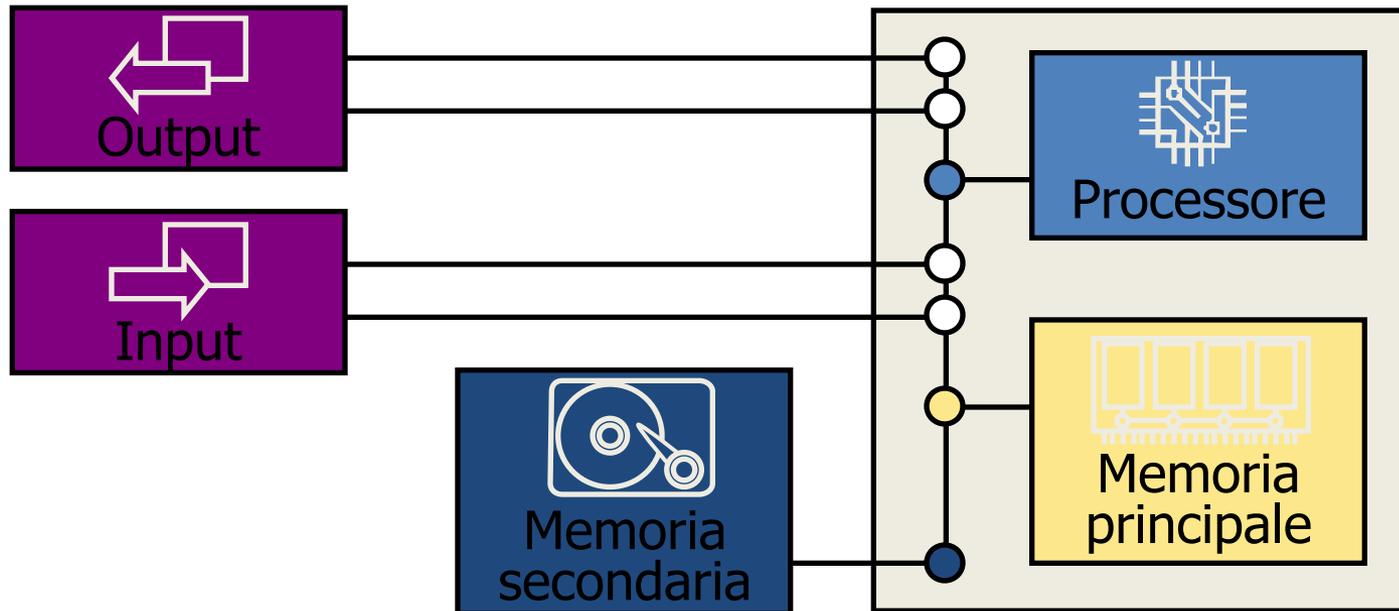
Gerarchia di memorie



Input/Output (I/O)



Uno sguardo d'insieme



- Periferiche di **output**:
 - mostrano i **risultati** delle elaborazioni
 - es.: video, stampante
- Periferiche di **input**:
 - permettono all'utente di **inserire dati** (e programmi)
 - es.: tastiera, mouse

Output: il monitor

- Il monitor **mostra un'immagine** che viene elaborata dal computer.
- L'immagine viene **aggiornata** spesso → appare animata.
- Parametri importanti nella scelta di un monitor:
 - **Risoluzione massima** dell'immagine mostrata.
 - **Profondità del colore** massima dell'immagine mostrata.
 - **Frequenza di aggiornamento** dell'immagine.
 - **Dimensione fisica** del monitor.
 - **Tipologia** di monitor.

Hardware del monitor

- La superficie del monitor è suddivisa in **pixel**.
- Monitor a **cristalli liquidi** e **matrice attiva** (LCD/TFT - Liquid Crystal Display/Thin Film Transistor).
 - Schermo **piatto**.
 - Cristalli particolari che diventano trasparenti se caricati elettricamente.
 - L'immagine è **fissa** sullo schermo, quando deve essere aggiornata cambiano solo i pixel diversi.
 - Meno problematico per l'occhio rispetto ai monitor a tubo catodico (il tipo di monitor usato ampiamente prima dello sviluppo dei monitor LCD).

Dimensione

- La **dimensione** del monitor:
 - Si misura in pollici (1 pollice = 2,54 cm).
 - Si esprime come la **lunghezza della diagonale** del monitor.
 - Perché si assume che la proporzione fra larghezza e altezza sia fissa (16:9, cioè 16 “unità” in larghezza e 9 “unità” in altezza).
 - Quando non lo è bisogna specificarla.
 - Dimensioni più comuni: **15”** - **17”** (portatile), più grande per i computer desktop.

Risoluzione e profondità

- La CPU scrive l'immagine da visualizzare in una **memoria speciale associata al video**.
 - Video-RAM (VRAM).
 - L'immagine è **codificata** come visto nella parte I del corso.
 - Il monitor visualizza l'immagine contenuta nella VRAM.
- **Risoluzione e profondità** dell'immagine sullo schermo dipendono:
 - Dalle capacità fisiche del video (dimensioni e possibilità di visualizzare più colori).
 - Dalla dimensione della memoria video VRAM.
- È possibile **variare** i parametri di risoluzione e profondità (entro i limiti massimi!).
 - Per i monitor **LCD**:
 - Esiste una controparte fisica per il pixel, e quindi esiste una **risoluzione "preferita" del monitor**.
 - Altre risoluzioni (più basse) si vedono male (sfocate).

Output: la stampante

- Principali **tecnologie** di stampa:
 - Ad **aghi** → economiche ma solo in bianco e nero.
 - Piccoli aghi colpiscono la carta imprimendovi sopra dei punti di inchiostro.
 - A **getto di inchiostro** → le più diffuse per uso personale.
 - L'inchiostro viene “sparato” sulla carta da fori piccolissimi.
 - È possibile avere inchiostri di vari colori (tricromia, quadricromia, esacromia).
 - **Laser** → più costose, convenienti per grandi quantità di stampe.
 - Un rullo viene caricato elettricamente da un laser.
 - Viene fatto passare nel toner, una polvere finissima colorata, che rimane solo nei punti caricati.
 - La carta viene passata sul rullo per imprimere l'immagine.

Qualità di una stampante

- **Velocità di stampa:**
 - Importante se si devono stampare grandi quantità di documenti.
 - Stampanti laser → velocità maggiore.
- **Risoluzione di stampa:**
 - **Precisione** della stampante nella resa dell'immagine.
 - Concetto simile alla risoluzione di memorizzazione.
 - Non è la stessa cosa → un'immagine a bassa risoluzione può essere stampata da una stampante ad alta risoluzione e viceversa!
 - Si misura in **DPI (Dot Per Inch = punti per pollice)**.
 - 600 DPI è una buona risoluzione.
 - Stampanti di alta qualità arrivano ai 1200-1440 DPI.
 - Stampanti fotografiche professionali anche oltre.

Output: scheda audio

- È una periferica interna.
- Ha il compito di trasformare informazioni digitali in suoni.
- Esternamente mostra solo il jack per inserire le cuffie o gli altoparlanti.
 - Solitamente l'audio non è amplificato per cui conviene che gli altoparlanti lo siano.
 - Può essere dotata di altoparlanti, solitamente di scarsa qualità.
 - Alcuni monitor contengono anche altoparlanti per cui è possibile collegarli anche all'uscita della scheda audio.

Input: la tastiera

- Quando l'utente preme o lascia andare un **tasto un codice binario corrispondente all'evento viene ricevuto dal processore.**
- Questa informazione viene elaborata e trasformata:
 - Nel **carattere** corrispondente se si tratta di un tasto-carattere (anche in base allo **stato** corrente).
 - In un **segnale di stato** se si tratta di un tasto con funzione particolare (es. SHIFT, CTRL, frecce, etc.).
- Molte tastiere moderne contengono tasti associati direttamente a funzioni di alto livello.
- Tastiere diverse; principali distinzioni:
 - Ordine delle lettere (Italia/USA → **QWERTY: nome che deriva dalla disposizione dei primi sei caratteri alfabetici**).
 - Numero di tasti (numero tipico: 108).

Input: il mouse

- Manda diversi tipi di segnali al processore:
 - **Pressione** di un tasto.
 - **Rilascio** di un tasto.
 - **Spostamento** rispetto alla posizione precedente.
- Questa informazione viene elaborata e tradotta:
 - In un corrispondente **spostamento della freccina** sullo schermo per gli eventi di **spostamento**.
 - In un **segnale di stato** in caso di **pressione e/o rilascio**.
 - Alcune combinazioni di eventi acquisiscono significati particolari:
 - Ad es. **pressione+rilascio+pressione+rilascio**, se eseguito velocemente, corrisponde ad un “**doppio click**”.
 - La trasformazione dalla combinazione iniziale all’evento composto viene operata dal software.

Dispositivi di **puntamento**

- **Mouse meccanici/ottici:**
 - Il classico mouse a forma di ‘topo’.
 - Il principio è lo stesso, cambia la tecnologia di registrazione dello spostamento.
 - Mouse ottico più preciso e meno deteriorabile.
- **Trackball:**
 - Invece che spostare il mouse si fa ruotare una pallina girevole.
- **Touchpad:**
 - Per comunicare lo spostamento si muove il dito su una superficie sensibile.
- **Pointing stick (o TrackPoint):**
 - Simile ad un minuscolo joystick.

Classi di elaboratori

- ó Personal computer
- ó Portatili
 - ® Notebook e NetBook
- ó Palmari
 - ® Tablet
 - ® Smartphone
- ó Workstation —> postazioni di lavoro potenti
- ó Mainframe —> gestiscono molti utenti in contemporanea
- ó Supercalcolatori —> per elaborazioni scientifiche
- ó