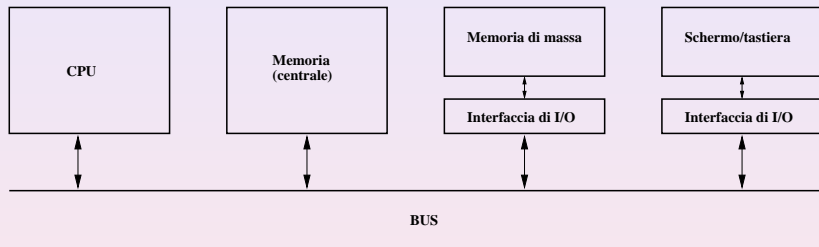


# Calcolatori elettronici: Hardware

Andrea Passerini  
passerini@disi.unitn.it

Informatica

# Macchina di Von Neumann



- La Central Processing Unit (CPU) è il “motore” principale dell’elaboratore.
- La memoria centrale contiene dati ed istruzioni ad accesso rapido.
- I dispositivi di input/output (I/O) o *periferiche* permettono di trasferire dati tra memoria centrale e/o CPU e l’ambiente circostante (e.g. schermo, tastiera, memoria di massa).
- Il bus di sistema rappresenta il canale di comunicazione tra le parti dell’elaboratore.

## Diviso in 3 parti

- **Bus dati:** usato per trasferire dati fra CPU e memoria o interfacce di I/O
  - La sua dimensione corrisponde ad una *word*, ossia al numero di bit con cui opera la CPU.
- **Bus indirizzi:** usato per identificare la posizione delle celle di memoria in cui la CPU va a scrivere/leggere
- **Bus di controllo:** dove passano i segnali di controllo che indicano il tipo di operazione da effettuare. E.g.:
  - R: indica (se alto) un'operazione di lettura (READ)
  - W: indica un'operazione di scrittura (WRITE)
  - M/IO: distingue tra operazione con la memoria (alto) o con una unità di IO (basso).

# Principi di esecuzione di calcolatore di Von Neumann

- dati ed istruzioni sono memorizzati in una memoria *unica* che permette lettura e scrittura.
- i contenuti della memoria sono indirizzati in base alla posizione, indipendentemente dal tipo di dato o istruzione contenuto.
- le istruzioni vengono eseguite in modo sequenziale

- Il linguaggio in cui sono scritti i programmi che la CPU è in grado di eseguire si chiama *linguaggio macchina*.
- Ogni istruzione (e.g. lettura o somma) è definita da un codice binario speciale (*codice operativo*).
- Ogni CPU è caratterizzata da un suo insieme di istruzioni (*instruction set*) che ne costituisce il linguaggio macchina.
- CPU diverse “parlano linguaggi” diversi!

- Un'istruzione è costituita da una stringa di bit contenenti:
  - Il codice operativo dell'istruzione
  - Gli operandi dell'istruzione
- Esempio di istruzione che effettua la somma di due operandi `src1` e `src2` contenuti nei registri `R02` e `R03` e salva il risultato nel registro `R01` (`dst`).

Struttura istruzione	codice operativo	dst	src1	src2
Linguaggio assembleatore	<code>add</code>	<code>R01</code>	<code>R02</code>	<code>R03</code>
Linguaggio macchina	000000 00000 100000	00001	00010	00011

# Linguaggio assemblatore

- Il linguaggio assemblatore (*assembly*) è il linguaggio di programmazione di più basso livello possibile (ossia più vicino al linguaggio macchina).
- Permette di programmare senza dover scrivere direttamente codice binario:
  - sostituendo i codici operativi con stringhe mnemoniche (e.g. `add` per la somma)
  - permettendo di scrivere gli indirizzi di memoria in codifica esadecimale.
- E' un linguaggio praticamente isomorfo al linguaggio macchina, per cui è anch'esso specifico per ogni CPU.
- Per poter essere eseguito dalla CPU deve essere convertito in linguaggio macchina tramite un programma *compilatore* (detto *assembler*).

- **Istruzioni aritmetico-logiche:** manipolano dati in ingresso e ne restituiscono il risultato in uscita.
- **Istruzioni di salto:** permettono di alterare l'esecuzione sequenziale di un programma. Si dividono in istruzioni di:
  - *salto incondizionato:* specificano l'indirizzo di memoria in cui si trova la prossima istruzione da eseguire
  - *salto condizionato:* specificano inoltre una condizione che deve essere verificata perché il salto abbia luogo, altrimenti l'elaborazione prosegue in sequenza
- **Istruzioni di ingresso/uscita:** servono a trasferire dati da e verso l'elaboratore, specificando l'unità periferica da usare (e.g. memoria di massa) e la posizione del dato sia nella periferica che nell'elaboratore.

# Esempi di istruzioni in linguaggio assembleatore

Istruzione	Significato	Tipo
add R01,R02,R03	$R01 \leftarrow R02 + R03$	<i>aritmetico-logica</i>
addi R01,R02,421	$R01 \leftarrow R02 + 421$	<i>aritmetico-logica</i>
lw R01,R02,421	$R01 \leftarrow M[R02 + 421]$	<i>ingresso/uscita (I)</i>
sw R01,R02,421	$M[R02 + 421] \leftarrow R01$	<i>ingresso/uscita (O)</i>
j 84210	<b>vai a 84210</b>	<i>salto (incondizionato)</i>
beq R01,R02,-421	<b>se (R01==R02) vai a PC-421</b>	<i>salto (condizionato)</i>

# Ciclo di esecuzione della CPU

- Nell'esecuzione di un programma, la CPU opera in modo ciclico, ripetendo fino alla terminazione del programma queste tre fasi:
  - 1 lettura (*fetch*): acquisizione dalla memoria della prossima istruzione da eseguire;
  - 2 decodifica (*decode*): riconoscimento dell'istruzione e identificazione delle operazioni da svolgere per completarne l'esecuzione;
  - 3 esecuzione (*execute*): effettuazione delle operazioni corrispondenti all'istruzione.

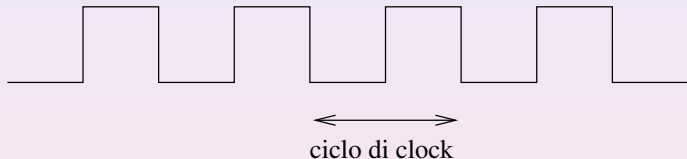
## Nota

Per poter essere eseguito dalla CPU, un programma deve risiedere in memoria centrale. Deve quindi essere preventivamente *caricato* in memoria dalla periferica che lo contiene (in genere la memoria di massa)

## Nota

- La CPU è il motore hardware del computer: esegue in maniera sequenziale istruzioni
- La CPU non ha consapevolezza del programma le cui istruzioni sta eseguendo, né del fatto che (come vedremo) possa alternare l'esecuzione di pezzi di programmi diversi
- E' il sistema operativo (software) a coordinare l'esecuzione dei diversi programmi, mandandoli in esecuzione nella CPU

- Tutte le azioni eseguite all'interno dell'elaboratore vengono sincronizzate da un oscillatore a cristalli di quarzo (*clock*), che segna la scansione temporale degli eventi:



- In un ciclo di clock viene eseguita una singola azione atomica (salvo parallelizzazioni)
- La frequenza di clock si misura in Hertz (cicli al secondo)

- **Unità di controllo:** è il sistema di coordinazione di tutto il processore.
- **Registri dati:** unità di memorizzazione temporanea ad accesso molto veloce in cui vengono mantenuti i dati da elaborare ed il risultato dell'elaborazione
- **Arithmetic-Logic Unit (ALU):** il dispositivo (o i dispositivi) in grado di elaborare i dati

- Dirige le operazioni della CPU. I suoi compiti principali sono:
  - 1 Recuperare l'istruzione successiva da eseguire
  - 2 Decodificarla
  - 3 Organizzare la sua esecuzione
- Contiene almeno tre registri fondamentali:
  - **Program counter (PC)**: contiene l'indirizzo in memoria della prossima istruzione da eseguire
  - **Instruction register (IR)**: contiene l'istruzione corrente
  - **Program status word (PSW)**: contiene dei bit speciali che riflettono il risultato dell'ultima operazione.

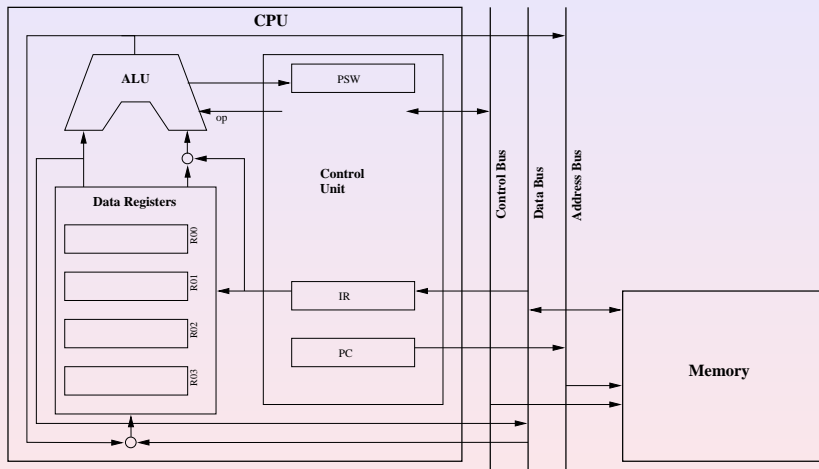
- La ALU non può in genere operare direttamente su operandi situati in celle di memoria:
  - ❶ Gli operandi vengono caricati nei registri dalla memoria tramite il bus dati
  - ❷ Gli operandi vengono trasferiti dai registri alla ALU che effettua l'operazione richiesta su di essi e salva il risultato in un registro.
  - ❸ Il risultato contenuto nel registro può essere scritto in memoria.

# Arithmetic-Logic Unit (ALU)

- Esegue le operazioni logiche (AND,OR,NOT) ed aritmetiche (somme, prodotti, confronti)
- In genere memorizza il risultato in un registro predefinito.
- Inoltre imposta i bit del Program Status Word in accordo al risultato.
- Operazioni su valori in virgola mobile e funzioni trascendenti (trigonometriche, radici quadrate, etc.) vengono in genere calcolate da un'unità apposita detta *Floating Point Unit* (FPU).

- Contiene una serie di bit (*flags*) che se impostati ad 1 indicano delle particolari situazioni nel risultato:
  - **bit di carry**: l'ultima addizione ha avuto un riporto
  - **bit di zero**: il risultato dell'ultima operazione è 0
  - **bit di segno**: segno del risultato dell'ultima operazione
  - **bit di overflow**: il risultato dell'ultima operazione è più grande del massimo intero rappresentabile.
  - etc.

# Struttura della CPU e collegamenti con la memoria



## 1 Fase di *fetch*

- L'unità di controllo manda il contenuto del PC (ossia l'indirizzo della prossima istruzione da eseguire) sul bus indirizzi e comunica alla memoria tramite il bus di controllo di effettuare una lettura
- La memoria manda sul bus dati il contenuto della cella di memoria corrispondente
- L'unità di controllo recupera l'istruzione dal bus dati e la memorizza nell'IR.
- Il PC viene incrementato di 1 per contenere l'indirizzo della istruzione successiva

2 Fase di *decodifica*: l'unità di controllo decodifica l'istruzione contenuta nell'IR e ne predispone l'esecuzione

3 Fase di *esecuzione*: l'istruzione viene eseguita.

4 Ritorna al passo 1.

- L'unità di controllo comunica al banco registri quali sono i registri contenenti gli operandi (o l'operando) e (se non è fissato) quale registro conterrà il risultato
- Il banco registri manda il contenuto dei registri specificati all'ALU
- L'unità di controllo comunica all'ALU quale operazione deve essere compiuta sugli operandi
- L'ALU esegue l'operazione mandando il risultato al banco registri ed aggiornando il PSW.
- Il banco registri memorizza il risultato nel registro di destinazione.

```
addi R01,R02,421
```

- L'unità di controllo comunica al banco registri che un operando è contenuto in R02, ed invia il secondo operando (il valore 421) direttamente alla ALU.
- Il banco registri manda il contenuto di R02 alla ALU.
- L'unità di controllo comunica alla ALU che l'operazione da effettuare è una somma.
- L'ALU esegue la somma mandando il risultato al banco registri ed aggiornando il PSW.
- Il banco registri memorizza il risultato in R01

- *Salto incondizionato*

- L'unità di controllo assegna al PC l'indirizzo contenuto nell'istruzione.
- Esempio  $j\ 208$  corrisponde a  $PC \leftarrow 208$ .

- *Salto condizionato*

- L'unità di controllo coordina l'esecuzione dell'operazione di cui si deve verificare la condizione.
- controlla (leggendo il PSW) se la condizione si è verificata
- In caso affermativo assegna al PC l'indirizzo contenuto nell'istruzione.
- Altrimenti il PC mantiene il suo contenuto (ossia l'indirizzo successivo a quello dell'istruzione corrente)

# Esempio di salto condizionato

```
beq R01, R02, -50
```

- L'unità di controllo comunica al banco registri che gli operandi sono in R01 ed R02
- Il banco registri ne manda il contenuto alla ALU.
- L'unità di controllo comunica alla ALU che l'operazione da effettuare è una sottrazione (per verificare se il risultato è zero)
- L'ALU effettua l'operazione ed imposta il PSW.
- L'unità di controllo verifica il bit di zero del PSW, e se vale 1 (il risultato è stato zero) aggiorna il PC con il valore  $PC - 50$ .

- L'unità di controllo coordina l'esecuzione dell'operazione che deve fornire l'indirizzo di memoria da cui prelevare (o dove memorizzare) il dato.
- L'indirizzo così calcolato viene messo sul bus indirizzi
- L'unità di controllo comunica al banco registri quale registro è coinvolto nell'operazione
- Il dato viene trasferito tramite il bus dati dalla memoria al registro selezionato (lettura) o viceversa (scrittura).

# Esempio di istruzione di ingresso/uscita

```
sw R01, R02, 421
```

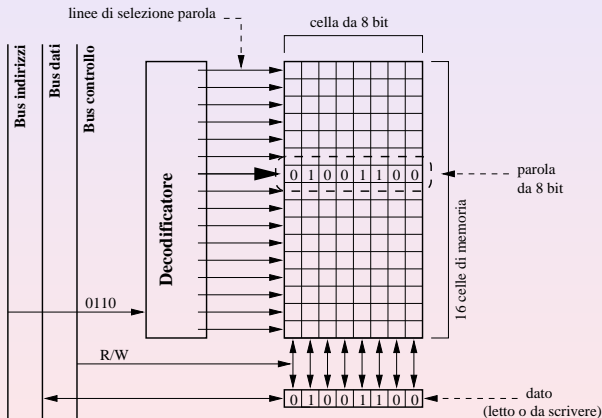
- L'unità di controllo comunica al banco registri che un operando è contenuto in R02, ed invia il secondo operando (il valore 421) direttamente alla ALU.
- Il banco registri manda il contenuto di R02 alla ALU.
- L'unità di controllo comunica alla ALU che l'operazione da effettuare è una somma (il cui risultato sarà inviato al bus indirizzi)
- L'unità di controllo comunica al banco registri che deve inviare il contenuto del registro R01 sul bus dati
- Il banco registri mette il contenuto di R01 sul bus dati
- L'unità di controllo comunica alla memoria tramite il bus di controllo di effettuare una scrittura
- La memoria legge il dato dal bus dati e lo memorizza nella cella specificata nel bus indirizzi.

- **memoria centrale:** deve contenere i programmi in esecuzione ed i relativi dati. Deve garantire un'elevata velocità per non penalizzare le prestazioni del sistema (è la memoria vista finora).
- **memoria di massa:** è destinata a contenere grandi quantità di dati in maniera stabile, ossia tali che siano disponibili anche dopo un eventuale spegnimento del calcolatore (e.g. disco fisso, dvd). E' caratterizzata da una velocità di accesso nettamente inferiore rispetto alla memoria centrale e da una capacità nettamente superiore.

- E' divisa in locazioni dette *celle*. La dimensione di una cella corrisponde alla dimensione della *parola* dell'elaboratore (16, 32, 64 bit) ed in genere alla dimensione del bus dati.
- Ogni cella ha un *indirizzo* che rappresenta la sua posizione rispetto alla prima cella di memoria (avente indirizzo 0).
- Il numero di bit dell'indirizzo definisce la dimensione massima della memoria indirizzabile (lo *spazio di indirizzamento*) e corrisponde alla dimensione del bus di indirizzi. Con  $k$  bit si possono indirizzare  $2^k$  celle diverse.
- E' realizzata con circuiti a semiconduttori (memoria elettronica) ed è *volatile*, ossia perde le informazioni in essa contenute quando non alimentata (e.g. quando il calcolatore viene spento).

# Modalità di accesso

- *Random Access Memory* (RAM): si può accedere a qualsiasi cella di memoria con un tempo costante.



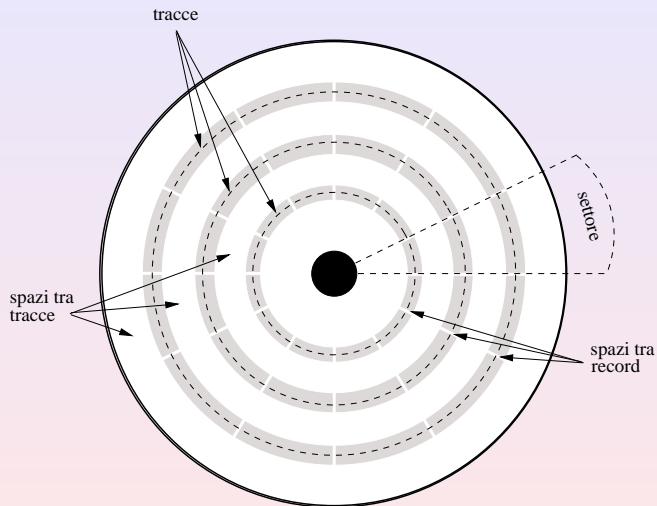
- *Read Only Memory* (ROM): memoria sempre a semiconduttori ma di tipo non volatile e di sola lettura. Utilizzata ad esempio per contenere le informazioni di inizializzazione del calcolatore (BIOS).
- Memorie *flash*: memorie elettroniche riscrivibili non volatili. Sono più lente delle memorie RAM (soprattutto in scrittura) ma sono sempre più diffuse in dispositivi portatili (macchine fotografiche digitali, telefoni cellulari, etc) e per trasportare dati (collegandosi al calcolatore tramite interfaccia USB) in sostituzione del floppy disk.
- *Memorie di massa*: sono caratterizzate da grande capacità e non volatilità dei dati. Sono principalmente realizzate tramite tecnologie magnetiche e ottiche.

- Sono supporti circolari ricoperti su entrambe le facce da un sottile strato di materiale magnetico, le cui condizioni di magnetizzazione consentono di memorizzare i dati.
- Ogni unità atomica di memorizzazione può essere in due sole condizioni di magnetizzazione, corrispondenti ai valori 0 e 1 di un bit.
- Le operazioni di lettura/scrittura sono eseguite da una testina che viene posizionata in prossimità della zona magnetizzata coinvolta nell'operazione.

# Organizzazione di un disco magnetico

- Le informazioni sono memorizzate in cerchi concentrici detti *tracce* la cui larghezza è pari alla dimensione della testina, e che contengono tutti la stessa quantità di dati (la densità di memorizzazione cresce dalla periferia al centro del disco)
- Le tracce sono suddivise in *settori* ed ogni settore ospita un blocco di dati (il *record*), in genere l'unità minima di trasferimento tra disco e memoria centrale.
- Tracce adiacenti e settori successivi sono separati da *gap* che permettono di evitare errori provocati da un posizionamento impreciso della testina.

# Organizzazione di un disco magnetico



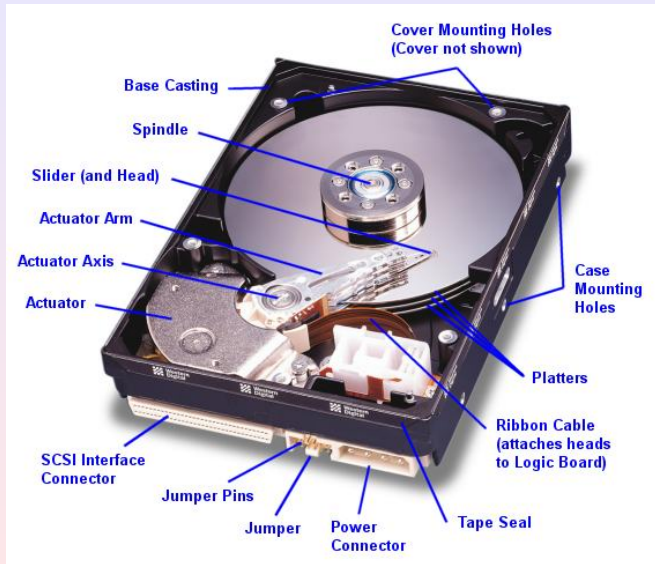
# Funzionamento di un disco magnetico

- Per poter essere utilizzato un disco deve essere *formattato*: si organizza la sua superficie in tracce e settori, identificandone le posizioni e memorizzandole in un'apposita sezione del disco stesso.
- Durante il funzionamento, il disco ruota a velocità costante.
- Per accedere ad un determinato record del disco si deve:
  - Posizionare la testina nella traccia contenente il record (*seek time*)
  - Attendere che la rotazione posizioni il settore cercato sotto la testina (*latency time*, dipende dalla velocità di rotazione del disco)
  - Attivare la lettura/scrittura dei dati

# Tipi di dischi magnetici

- L'*hard disk* (o disco fisso o disco rigido) è la tipica unità di memoria di massa dei calcolatori. Consta di un *disk drive* contenente:
  - Un insieme di dischi magnetici sovrapposti a distanza di pochi millimetri l'uno dall'altro
  - Un insieme di testine di lettura/scrittura (due per disco per le due facce) solidali ad un braccio meccanico.
  - Un motore per far ruotare i dischi e l'elettronica che gestisce le operazioni
- Il *floppy disk* è stato per anni lo strumento standard di trasferimento dati. La sua capacità (1.4MB) è da tempo inadeguata alla dimensione media dei dati ed è stato soppiantato dai dispositivi a memoria flash (chiamati penne o chiavi USB).

# Hard disk



- Costituiti da un disco di 12cm di diametro e 1mm di spessore, contenente materiale riflettente annegato in un supporto plastico trasparente.
- Un raggio laser esplora la superficie del disco e la riflessione del fascio luminoso indica il bit memorizzato nella posizione puntata.
- Sono ad accesso *sequenziale* (percorso a spirale dal centro alla periferia).
- I CD (*Compact Disc*) hanno capacità di 600MB-700MB, i DVD (*Digital Versatile Disc*) da 4.7GB a 17GB.

- Memorizzano i dati con maggiore risoluzione rispetto ai CD (dimensione minore dell'area corrispondente ad un bit e spirale più stretta)
- DVD a strato singolo (*single layer*): 4.7GB
- DVD a doppio strato (*double layer*): 8.5GB. Usano materiali semitrasparenti: il primo strato risulta trasparente al raggio laser usato per leggere il secondo strato (che contiene 3.8GB).
- DVD a doppio strato e doppia faccia (*double face double layer*): 17GB. Usano un doppio strato su entrambe le facce del disco.

# Sola lettura, registrabili, riscrivibili

- CD-ROM/DVD-ROM: distribuiti con contenuto già memorizzato, non possono essere modificati (*Read Only Memory*)
- CD-R/DVD-R/DVD+R (due standard diversi per i DVD): distribuiti privi di contenuto, possono essere registrati “bruciando” parti del supporto con un laser più potente di quello usato in lettura. Il grado di riflettività delle zone bruciate sarà diverso e permetterà di distinguere 0 da 1.
- CD-RW/DVD-RW/DVD+RW: sono riscrivibili in quanto un opportuno ciclo di riscaldamento/raffreddamento può eliminare le “bruciature” e riportare il supporto allo stato originario.

- Le periferiche di I/O comunicano con l'unità centrale tramite delle *interfacce*, schede collegate al bus di sistema mediante connettori ed alle periferiche mediante *porte*.
- La trasmissione dei dati può avvenire mediante due modalità:
  - *seriale*: viene trasmesso un bit per volta (e.g. mouse)
  - *parallela*: tutti i bit del byte vengono trasmessi contemporaneamente (e.g. prime stampanti).

- Negli ultimi anni si sono diffusi nuovi standard di connessione seriale con velocità superiori alla connessione parallela:
  - Universal Serial Bus (USB) (diventato standard per stampanti, modem, scanner..)
  - Firewire (o IEEE 1394)
  - Collegamenti *wireless* (ossia senza fili) mediante onde radio, quali il *Bluetooth*.

## Tastiera

- trasforma la pressione di ogni tasto in un segnale elettrico corrispondente al carattere digitato
- il segnale viene inviato sia all'unità centrale per l'elaborazione sia al video per la verifica di ciò che è stato digitato.

## Dispositivo di puntamento

- E' associato ad i sistemi di interfaccia grafica e permette di selezionare una posizione qualsiasi nello schermo.
- Il dispositivo più diffuso è il *mouse*, la cui funzione viene equivalentemente svolta nei calcolatori portatili dal *touch pad* (o dispositivi analoghi).

## Video

- Lo schermo è costituito da una matrice di punti (*pixel*) indirizzabili singolarmente, il cui numero indica la *risoluzione* dello schermo (tipicamente tra i 1024x768 ed i 2048x1536 pixels).
- Il numero di bit con cui si rappresenta un pixel indica la gamma di colori rappresentabile (tipicamente 8 bit per ognuno dei tre colori fondamentali rosso, verde, blu più altri 8 per trasparenze e altro, totale 32 bit)
- Una memoria RAM contiene la configurazione di tutti i pixel dello schermo, che viene convertita in opportuni segnali inviati allo schermo.

## Scheda grafica

- La velocità di visualizzazione delle immagini sullo schermo è fortemente condizionata dalla dimensione dell'informazione da trasferire (la matrice dei pixel).
- Il problema viene affrontato in genere tramite speciali *coprocessori grafici* (le *schede grafiche acceleratrici*) dotate di:
  - memoria RAM aggiuntiva rispetto alla memoria centrale
  - circuiti dedicati per gestire la scrittura dei pixel indipendentemente dalla CPU.

## Cathode Ray Tube o CRT

- Video basati su tubi a raggi catodici (stessa tecnologia della televisione)
- Pesante ed ingombrante, la profondità del tubo deve essere proporzionale al diametro dello schermo.
- Permettono di utilizzare più risoluzioni (fino alla massima consentita) senza problemi.
- Possono presentare uno *sfarfallamento* dell'immagine per frequenze di refresh non abbastanza elevate.

## Liquid Cristal Display o LCD

- Video con schermi a cristalli liquidi.
- Peso ridotto ed ingombro minimo, minor assorbimento di potenza (particolarmente adatti ai portatili)
- Hanno una risoluzione *nativa*. Altre risoluzioni possono essere selezionate ma con peggioramento delle prestazioni.
- La immagine vista cambia al cambiare dell'angolo di visuale e della distanza dallo schermo.
- Hanno una qualità ancora inferiore in termini di gamma di colori e persistenza dell'immagine.

## Stampanti

- *Stampanti ad aghi*: fanno scorrere sulla linea di stampa una testina di scrittura ad aghi, ognuno dei quali può premere sulla carta un nastro inchiostroato ottenendo un punto stampato. Molto economiche, lente e con bassa qualità di stampa.
- *Stampanti a getto d'inchiostro (ink jet)*: stampano configurazioni di punti tramite una testina di scrittura con ugelli che spruzzano sulla carta microscopiche gocce d'inchiostro. Economiche, abbastanza veloci e con buona qualità di stampa anche a colori.

## Stampanti laser

- l'immagine da stampare viene impressa su un rullo di materiale fotosensibile da un raggio laser che carica elettricamente le zone raggiunte.
- Tali zone attirano le particelle di inchiostro in polvere (*toner*), che vengono trasferite sulla carta e fissate mediante riscaldamento.
- Il rullo viene poi riportato allo stato originario tramite una forte esposizione luminosa.
- Costo più elevato, alta velocità e ottima qualità.
- Necessitano di processore e memoria RAM propri nel caso la mappa di bit dell'immagine venga realizzata dalla stampante stessa. Altrimenti la mappa deve essere realizzata dal calcolatore e trasferita alla stampante con grande spreco di banda e maggior utilizzo della CPU.