

## 2. UN ESECUTORE CONCRETO: IL COMPUTER

Il **computer** è una macchina costituita da dispositivi di diversa natura (meccanica, elettrica, ottica, etc.) in grado di elaborare dati in modo automatico, veloce, sicuro ed efficiente.

Il **computer** è:

- una **macchina** ossia un dispositivo senza intelligenza autonoma (dispositivo “stupido”);
- un dispositivo **elettronico** ossia il suo funzionamento si basa su componenti di tipo elettronico”);
- un **elaboratore di dati** ossia eseguendo le istruzioni di un programma è capace di rilevare dati dall'esterno (dati di output) , operare su di essi e fornire dati all'esterno (ossia i dati di output o risultati dell'elaborazione)

Quindi il computer come abbiamo visto è una **macchina programmabile** ossia una macchina che può essere utilizzata per *problemi diversi*, in grado di interpretare ed eseguire una serie di ordini impartiti dall'esterno (*algoritmo*) che ne descrivono rigorosamente il *processo risolutivo*.

Noi qui ci occuperemo essenzialmente di **personal computer** ossia brevemente **pc**.

### HARDWARE & SOFTWARE

Il computer ha molti dispositivi indispensabili per potere svolgere i suoi compiti che vengono divisi in due grandi categorie:

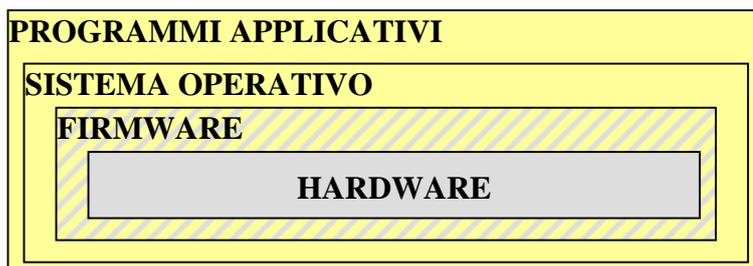
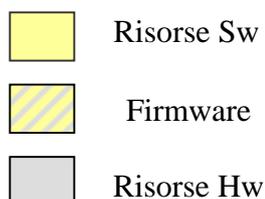
1) **le risorse hardware o l'hardware** che costituiscono essenzialmente la **parte “fisica”** del computer ossia le parti meccaniche, elettriche, magnetiche ed ottiche (dall'inglese *hardware* = lett. circuiteria);

2) **le risorse software o il software** che costituiscono essenzialmente la **parte “logica”** del computer ossia tutti i programmi che ci permettono di utilizzare il sistema nel suo complesso (dall'inglese *software* = lett. circuiteria). Questi programmi si dividono in:

- **programmi di sistema o sistema operativo:** che sono tutti quei programmi che gestiscono le risorse del sistema e ne permettono l'utilizzo;
- **programmi applicativi:** che sono tutti quei programmi che risolvono determinate categorie di problemi che l'utente può avere, ma che non gestiscono direttamente alcuna risorsa del sistema.

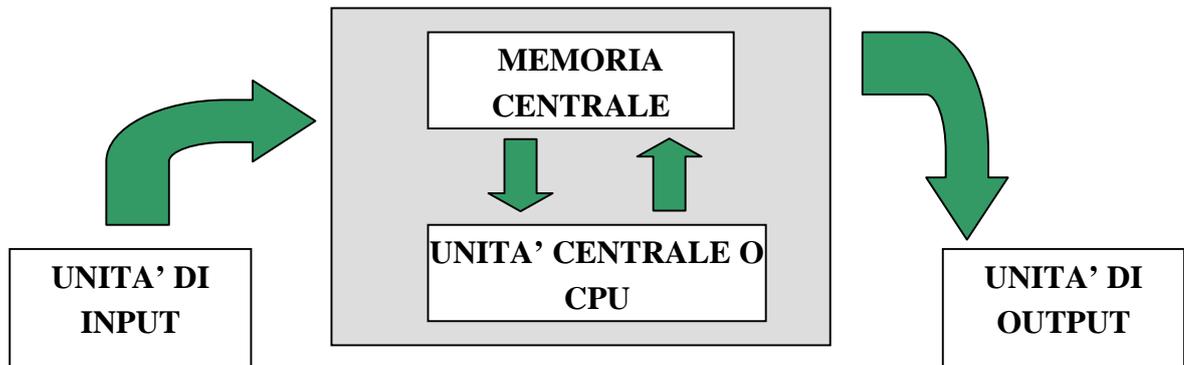
### N.B.

Esistono dei dispositivi che sono sia hardware sia software. Tali dispositivi prendono il nome di **firmware** il che non deve far pensare ad una categoria a parte, bensì a qualcosa di intermedio tra i due. Infatti il firmware è costituito da una *serie di istruzioni software* memorizzate *nell'hardware* e non modificabili da parte dell'utente (Esempio tipico: le memorie di tipo ROM che descriveremo più avanti).



## IL MODELLO DI VON-NEUMANN (1° SCHEMATIZZAZIONE)

La maggior parte dei sistemi di elaborazione, indipendentemente dalla velocità, dalle prestazioni, dalle capacità di memorizzazione o di elaborazione, presentano tutti una stessa caratteristica comune: **il loro hardware** rispecchia il **modello** ideato nel lontano 1946 da **John Von Neumann**



MODELLO DI VON NEUMANN  
Prima schematizzazione

Secondo tale modello un sistema per l'elaborazione elettronica dei dati è composto **da un insieme di unità funzionali** (ossia si pensa alle funzionalità che quell'unità deve svolgere e non a come le realizzerà), ognuna di quali ha dei compiti ben precisi:

- **le unità di input:** sono predisposte per l'immissione (dall'esterno) delle istruzioni dei programmi e dei dati (di input) su cui tali programmi andranno ad operare;
- **le unità di output:** permettono al sistema di comunicare con l'esterno i risultati dell'elaborazione;
- **la memoria centrale:** conserva dati ed istruzioni introdotti con le unità di input, ma anche dati di lavoro nonché i risultati dell'elaborazione;
- **l'unità centrale o CPU (Central Processing Unit):** ha il compito di eseguire le istruzioni dei programmi immessi sui dati relativi, di eseguire i calcoli logici ed aritmetici, nonché controllare il buon funzionamento di tutto il sistema

Il modello di Von Neumann è caratterizzato dal fatto di ipotizzare la presenza di un'unica unità centrale: un elaboratore basato su tale modello si chiama **elaboratore seriale**.

Una macchina invece basata sulla presenza di più unità centrali prende il nome di **elaboratore parallelo**.

## MEMORIA CENTRALE: ORGANIZZAZIONE

DEFINIZIONE: La **memoria centrale** (detta anche **memoria principale** o **memoria di lavoro**) è un dispositivo elettronico in grado di contenere tutte le informazioni necessarie per l'elaborazione di un programma (ossia dati, istruzioni, risultati intermedi e finali). E' quindi coinvolta continuamente durante il processo di elaborazione

Le **operazioni** possibili in memoria centrale sono:

- **scrittura** delle informazioni (dati ed istruzioni) provenienti dalle varie componenti del computer (tale operazione è *distruttiva* in quanto comporta la distruzione dell'informazione precedentemente contenuta nella stessa posizione);
- **conservazione** delle informazioni (dati ed istruzioni) per averle sempre a disposizione durante tutta l'elaborazione;
- **lettura** delle informazioni (dati ed istruzioni) memorizzate senza rischio di distruzione o corruzione delle stesse in quanto viene prelevata una copia dell'informazione letta.

All'interno del computer le informazioni (dati ed istruzioni) non sono composte da numeri o lettere così come noi le inseriamo da tastiera, ma vengono convertite in **codici binari** ossia sequenze di **0** ed **1** realizzate attraverso **dispositivi elettronici bistabili detti flip-flop** ossia associando le cifre binari ai due generici stati che tale dispositivo può assumere (**0** sta per **spento** ossia non c'è passaggio di energia elettrica o magnetica mentre **1** sta per **acceso** ossia c'è il passaggio di energia elettrica o magnetica)

Possiamo immaginare la memoria centrale come un dispositivo elettronico composto da migliaia di minuscoli circuiti elettrici che possono assumere solo due stati fisici – **acceso o spento** – cui si associano i ben noti simboli binari – **1 e 0**.

Questi minuscoli circuiti elettrici possono essere visti come tante **celle** o **posizioni di memoria adiacenti** tutte di uguale dimensione che sono in grado di immagazzinare un dato o una istruzione in forma binaria. (N.B. la cella di memoria non immagazzina un solo bit ma in genere una ben definita configurazione di bit)

Abbiamo già visto che un singolo **bit** non può bastare per rappresentare un qualsiasi carattere e che anzi ne servono più di uno ossia **occorrono più configurazioni di bit** per poter provvedere alla codifica in binario di qualunque informazione (dati ed istruzioni).

Il **BYTE** ossia **BInary Term pari ad 8 bit**, insieme con i suoi multipli, viene assunto come **unità di misura** della **capacità** della memoria centrale, mentre con il termine **parola o word** si intende il numero di bit (in genere prestabilito e multiplo di 8) che formano una singola cella.

Una parola è composta in genere da 8, 16 o 32 bit ossia 1, 2 o 3 byte e più è alto il numero di bit che costituiscono la parola più è veloce il computer che la utilizza. Cio dipende dal fatto che durante l'elaborazione il trasferimento dei dati tra la memoria centrale e la CPU avviene **in parallelo** (più bit contemporaneamente e non uno dopo l'altro) e **per parola**.

Molto spesso la lunghezza della parola coincide con il **grado di parallelismo della memoria** ossia con il numero dei bit che la CPU può leggere o scrivere contemporaneamente.

Byte		Parola o Word	
Unità di misura	corrisponde a	Unità di misura	corrisponde a
Byte	8 bit	Word	Dipende dal tipo di memoria (multipla del byte)
<b>KiloByte</b> (simbolo <b>KB</b> )	$2^{10}$ byte ossia <b>1024 Byte</b>	<b>KiloWord</b> (simbolo <b>KW</b> )	$2^{10}$ Word ossia <b>1024 Word</b>
<b>MegaByte</b> (simbolo <b>MB</b> )	$2^{10}$ KiloByte ossia <b>1024 KiloByte</b> ossia <b>1048576 Byte</b>	<b>MegaWord</b> (simbolo <b>MW</b> )	$2^{10}$ KiloWord ossia <b>1024 Kilo Word</b> ossia <b>1048576 Word</b>
<b>GigaByte</b> (simbolo <b>GB</b> )	$2^{10}$ MegaByte ossia <b>1024 MegaByte</b> ossia <b>1048576 KiloByte</b> ossia <b>1073741824 Byte</b>	<b>GigaWord</b> (simbolo <b>GW</b> )	$2^{10}$ MegaWord ossia <b>1024 MegaWord</b> ossia <b>1048576 KiloWord</b> ossia <b>1073741824 Word</b>
<b>TeraByte</b> (simbolo <b>TB</b> )	$2^{10}$ GigaByte ossia <b>1024 GigaByte</b> ossia <b>1048576 MegaByte</b> ossia <b>1073741824 KiloByte</b> ossia <b>1099511627776 Byte</b>	<b>TeraWord</b> (simbolo <b>TW</b> )	$2^{10}$ GigaWord ossia <b>1024 GigaWord</b> ossia <b>1048576 MegaWord</b> ossia <b>1073741824 KiloWord</b> ossia <b>1099511627776 Word</b>

Esempio:

Una memoria centrale **da 64 KB** (molto piccola) è formata da:

- **65536** celle o posizioni numerate da **0** a **65535** se è costituita **da parole di 8 bit**;
- **32768** celle o posizioni numerate da **0** a **32767** se è costituita **da parole di 16 bit**.

Una memoria centrale **da 64 KW** costituita da **parole di 16 bit** è formata da:

- **65536** parole numerate da **0** a **65535**

**N.B.** Essa corrisponde evidentemente ad una memoria di capacità di **128 KB**

Una memoria centrale **da 64 KW** costituita da **parole di 8 bit** è formata da:

- **65536** parole numerate da **0** a **65535**

**N.B.** Essa è perfettamente equivalente ad una memoria di capacità di **64 KB**

Infatti:

$$64 \text{ KB} = 64 * 2^{10} \text{ Byte} = 64 * 1024 \text{ Byte} = 65536 \text{ Byte} = 524288 \text{ bit}$$

Ora se la parola è di **8 bit**, siccome **8 bit = 1 Byte** allora **64 KB = 65536 parole**

Ora se la parola è di **16 bit**, siccome **8 bit = 1 Byte** allora **64 KB = 65536/2 parole** ossia **32768 parole**

Invece

$$64 \text{ KW} = 64 * 2^{10} \text{ Word} = 64 * 1024 \text{ Word} \left\{ \begin{array}{l} \text{Se Word} = 16 \text{ bit allora } 1048576 \text{ bit ossia } 128 \text{ KB} \\ \text{Se Word} = 8 \text{ bit allora } 524288 \text{ bit ossia } 64 \text{ KB} \end{array} \right.$$

## I BUS

Le linee che realizzano il collegamento *fisico* tra le varie unità funzionali di un computer si dicono **bus**.

DEFINIZIONE: Il **bus** è costituito da una serie di collegamenti hardware (simile ad un gruppo di fili), **uno per ciascun bit** da trasportare, su cui viaggiano tutte le informazioni (dati ed istruzioni) che vengono scambiate tra l'unità centrale o CPU, la memoria centrale e le unità di input output.

In un sistema di elaborazione esistono **tre bus principali**:

### 1) BUS degli indirizzi o *address bus*:

E' il bus che *trasporta l'indirizzo* necessario per reperire una cella di memoria o una unità di input/output.

E' un bus **unidirezionale** e l'unico dispositivo abilitato ad inviare informazioni su questo bus è **l'unità di controllo o CPU**.

Questo bus è logicamente **composto da tanti fili quanti sono i bit che compongono o rappresentano l'indirizzo** di una cella

### 2) BUS dei dati o *data bus*

E' il bus che viene utilizzato per *lo scambio di informazioni* tra tutti i vari dispositivi (unità di input/output/ memoria centrale, unità di controllo o CPU).

E' un bus **bidirezionale** e tutti i dispositivi che costituiscono il sistema di elaborazione sono abilitati ad inviare informazioni su questo bus.

Questo bus è logicamente **composto da tanti fili quanti sono i bit che compongono la parola o word** utilizzata come unità di trasferimento (16 bit, 32 bit, etc.)

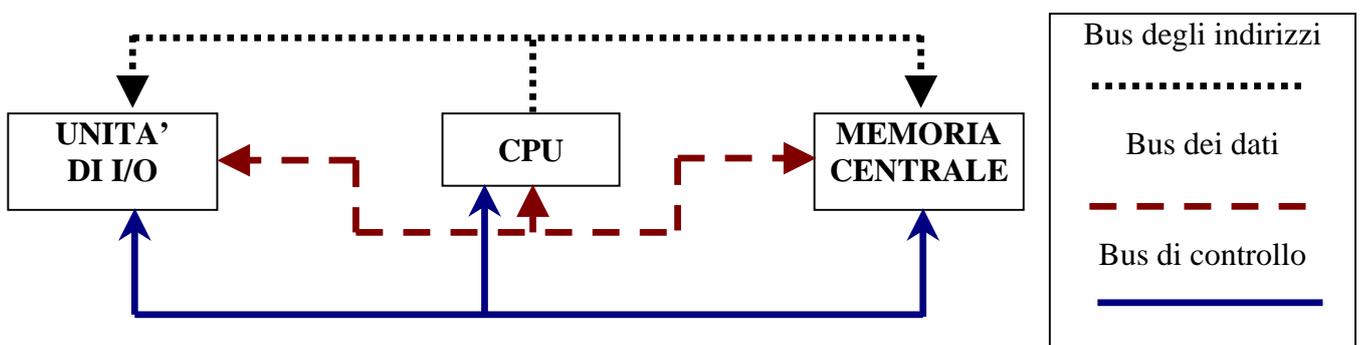
### 3) BUS di controllo o *control bus*

E' il bus che viene utilizzato per *la sincronizzazione della trasmissione e per permettere lo scambio di segnali di controllo* tra le varie unità (segnale di lettura/scrittura, segnale di inizio/fine trasmissioni, segnale di unità libera o occupata, etc.).

E' un bus **bidirezionale** ed ha come scopo principale quello di coordinare e controllare il traffico di tutte le informazioni che viaggiano sugli altri 2 bus.

Questo bus può essere **composto da un numero variabile di fili**: più fili lo compongono, più sono le informazioni che può trasportare.

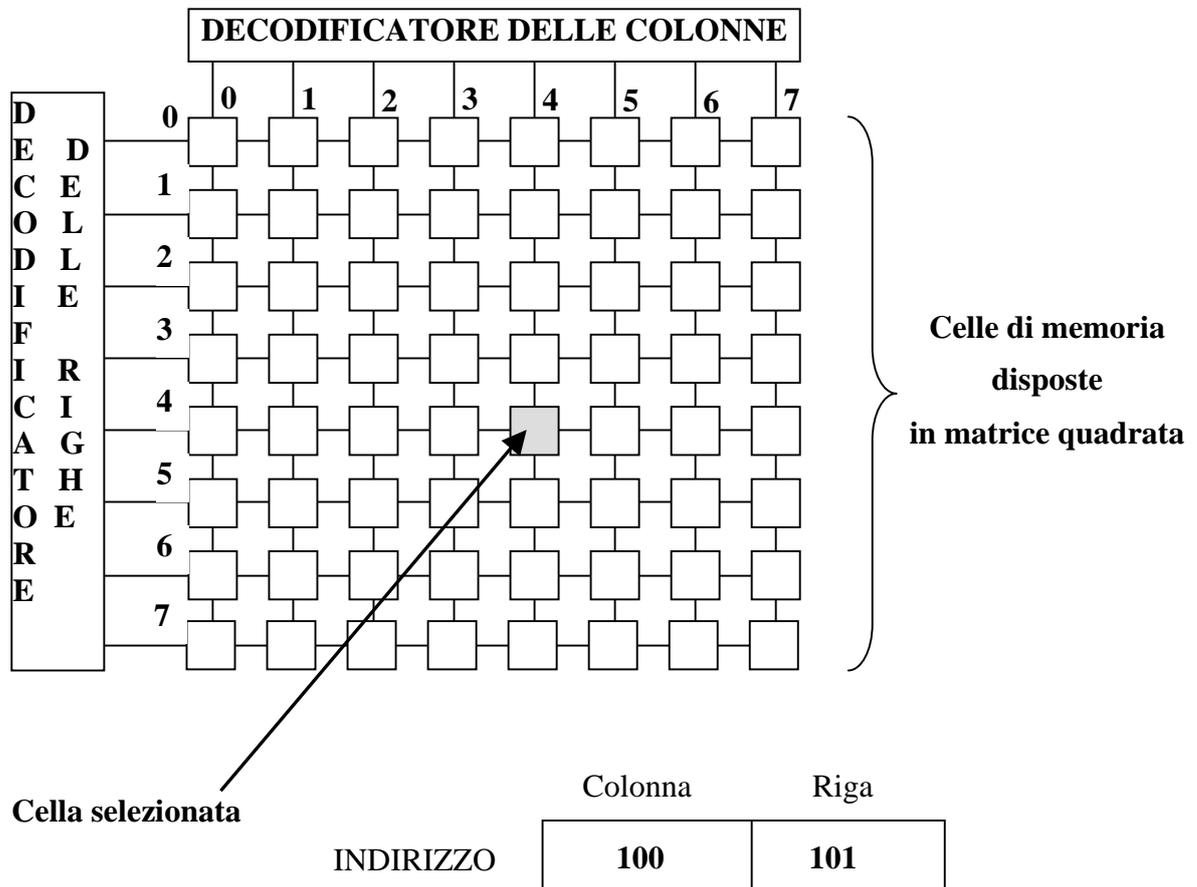
I tre tipi di bus necessitano di un **controllore** ossia di un dispositivo spesso inserito nello stesso microprocessore in grado di stabilire quale dispositivo deve usare il bus evitando i conflitti che potrebbero sorgere dai tentativi contemporanei di accesso al bus da parte di unità diverse.



## MEMORIA CENTRALE: ARCHITETTURA

**A) Le celle della memoria centrale**

Le **celle della memoria** (la cui dimensione è stabilita dalla lunghezza della **parola o word** utilizzata) sono numerate progressivamente a partire da 0 ma non sono disposte in successione bensì sono organizzate secondo la struttura di una **matrice quadrata**



Ad ogni intersezione tra le colonne e le righe si trovano **le celle di memoria**: l'indirizzo di una cella è quindi composto da due parti:

- a) **la prima parte a destra** della stringa binaria (i bit meno significativi) individua la **riga**
- b) **la restante parte a sinistra** della stringa binaria (i bit più significativi) individua la **colonna**

Esempio:

Se un indirizzo è espresso con 16 bit i primi 8 bit meno significativi (ossia i primi 8 a partire da destra) individuano la riga mentre i restanti 8 bit più significativi (ossia i primi 8 a partire da sinistra).

**N.B.** La *lunghezza in bit* con la quale è espresso l'*indirizzo* di una cella memoria non c'entra assolutamente niente con la *lunghezza in bit della cella di memoria* stessa ossia con la lunghezza della **parola o word** utilizzata per comunicare le informazioni.

## **B) I registri della memoria centrale**

All'interno della memoria centrale vi sono delle piccole componenti logiche indispensabili per poter dialogare con l'unità centrale o CPU detti **registri**.

**DEFINIZIONE:** I **registri** sono piccole e veloci memorie (generalmente composte da 1, 8, 16 o 32 bit) che svolgono funzioni ben precise e vengono utilizzati in determinate fasi dell'elaborazione.

I più importanti registri della memoria centrale sono:

- **registro MAR o Memory Address Register**

E' il *registro degli indirizzi di memoria* e contiene l'indirizzo della cella di memoria in cui l'unità centrale o CPU deve leggere o scrivere.

La lunghezza di questo registro è uguale a quella necessaria a rappresentare l'indirizzo stesso.

- **registro MDR o Memory Data Register**

E' il *registro dei dati di memoria* e contiene la parola o word che deve essere scritta o quella appena letta.

La lunghezza di questo registro è uguale a quella della parola o word utilizzata.

- **registro CONTR**

E' il registro che contiene appositi messaggi provenienti dalla memoria stessa e dalle altre unità utilizzati per coordinare e sincronizzare le comunicazioni tra la memoria e le altre varie unità.

## **C) L'operazione di lettura o scrittura di dati in memoria centrale**

Ecco come tecnicamente avvengono le operazioni di lettura e scrittura di un dato in memoria centrale:

- **lettura in memoria di un dato:** quando l'unità centrale o CPU vuole leggere un dato essa:

1. invia al registro **MAR**, attraverso il bus degli indirizzi, l'indirizzo della cella di memoria da leggere;
2. contemporaneamente invia al registro **CONTR** attraverso il bus di controllo il segnale di "leggi";
3. la memoria, visto il segnale di controllo, si posiziona sulla cella indicata dal registro **MAR** e legge il dato in essa contenuta posizionandolo nel registro **MDR**;
4. dopo avere prelevato il dato, la memoria invia al registro **CONTR** un messaggio di "pronto";
5. la memoria, invia all'unità centrale o CPU tramite gli appositi bus, il contenuto dei registri **MDR** e **CONTR**.

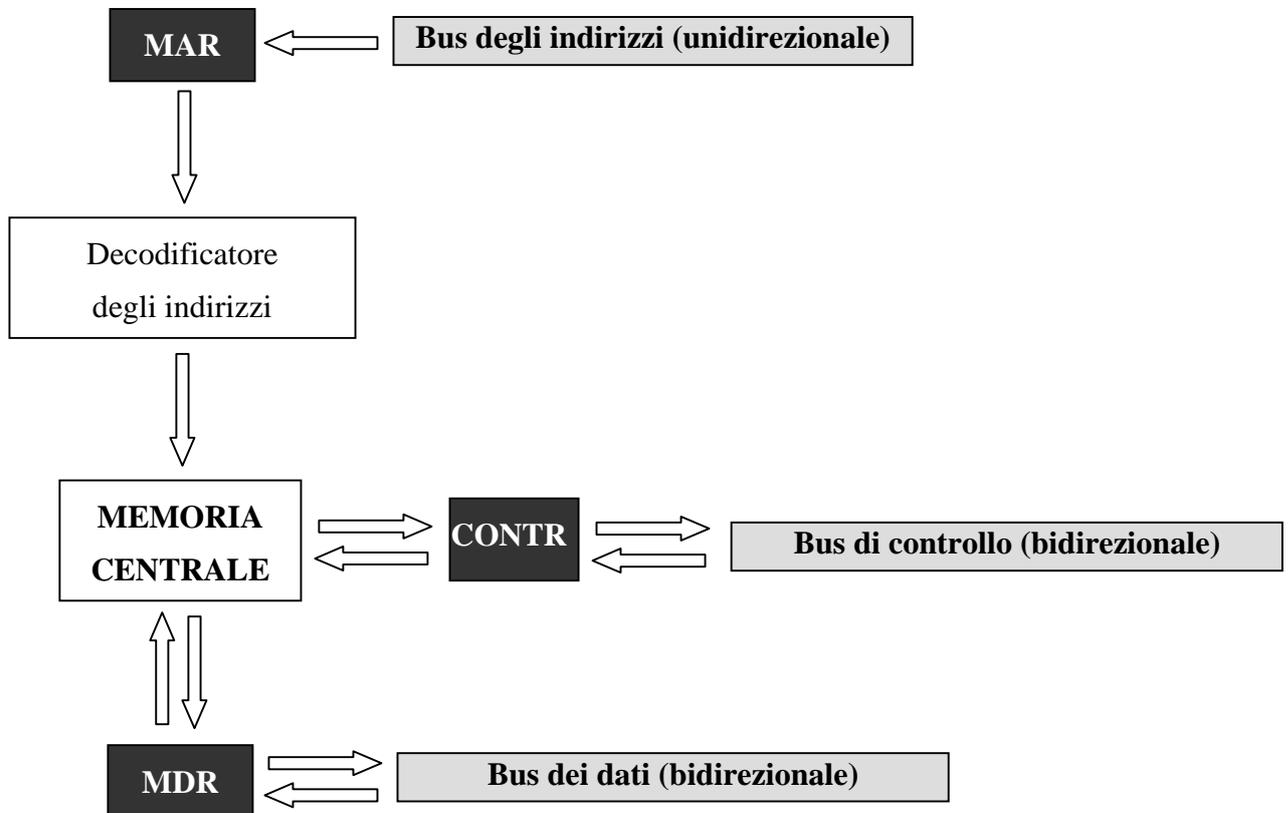
L'operazione di lettura di un dato in memoria è così compiuta.

- **scrittura in memoria di un dato:** quando l'unità centrale o CPU vuole scrivere un dato essa:

1. invia al registro **MAR**, attraverso il bus degli indirizzi, l'indirizzo della cella di memoria dove scrivere il dato;
2. contemporaneamente invia al registro **CONTR** attraverso il bus di controllo il segnale di "scrivi";
3. invia al registro **MDR**, attraverso il bus dei dati, il valore del dato da scrivere;
4. la memoria, visto il segnale di controllo, si posiziona sulla cella indicata dal registro **MAR** e scrive il dato contenuta nel registro **MDR**;
5. dopo avere scritto il dato, la memoria invia al registro **CONTR** un messaggio di "pronto";

L'operazione di scrittura di un dato in memoria è così compiuta.

Le azioni necessarie a compiere una operazione di lettura o di scrittura prendono il nome di **ciclo di memoria** ed il tempo necessario per realizzarle si chiama **tempo di accesso** (attualmente è nell'ordine dei nanosecondi con  $1 \text{ nanosec} = 1 \text{ miliardesimo di secondo}$ ).



#### **D) Componenti della memoria centrale**

La memoria centrale è generalmente costituita da 3 tipologie di memoria:

**1) la memoria RAM (o Random Access Memory):** è la *memoria di lavoro* finora descritta che offre la possibilità di leggere o scrivere informazioni (dati ed istruzioni) necessarie all'esecuzione di un programma.

Essa è detta **ad accesso casuale** ossia è possibile leggere o scrivere l'informazione in modo indipendente dalla relativa posizione di registrazione.

E' una memoria velocissima ma **volatile** ossia impossibilitata a mantenere il suo contenuto informativo quando la corrente viene interrotta.

Proprio per questo motivo sono state introdotte le **memorie di massa** (che vedremo in seguito) ossia quei supporti capaci di potere memorizzare in modo permanente le informazioni, una volta inserite.

**2) la memoria ROM (o Read Only Memory):** è una memoria di sola lettura nella quale sono memorizzate in modo permanente le istruzioni che l'unità di controllo o CPU deve eseguire per potere essere in grado di svolgere i suoi compiti.

Tale operazione di memorizzazione avviene all'atto della costruzione del computer ad opera dei costruttori di hardware stesso.

Contiene programmi importantissimi come quelli necessari all'avviamento della macchina (BIOS e caricatore del sistema operativo).

Esistono tre tipi fondamentali di **ROM**:

- la **ROM** mascherata ossia quella di cui abbiamo parlato finora;
- la **PROM** o ROM programmabile (Programmable ROM), sulla quale l'utente è sempre in grado di leggere le informazioni memorizzate dal costruttore hardware, ma che permettono all'utente **per una volta sola**, attraverso apparecchiature particolari (dette programmatori di PROM), la registrazione di informazioni;
- la **EPROM** sulla quale l'utente è sempre in grado di leggere le informazioni memorizzate dal costruttore hardware, ma che permettono all'utente **senza alcun limite**, attraverso apparecchiature particolari (dette programmatori di EPROM), la registrazione di informazioni;

**3) la memoria CACHE:** è una particolare memoria RAM caratterizzata da una elevatissima velocità di trasmissione, che viene usata come intermedia tra unità centrale o CPU e la memoria RAM centrale per aumentare la velocità di prestazioni del computer.

In questo tipo di memoria vengono conservate temporaneamente tutte quelle informazioni maggiormente richieste in modo da poter essere subito rese disponibili, trasformando molti accessi alla memoria centrale in accessi velocissimi alla memoria cache.

Questo tipo di memoria viene organizzata su due livelli.

- **la memoria cache di 1° livello:** inclusa nel chip del processore e che quindi lavora alla sua stessa frequenza;
- **la memoria cache di 2° livello:** spesso incorporata nella scheda madre

### E) I parametri fondamentali di una memoria

I parametri fondamentali che caratterizzano le memorie sono:

- **tempo di accesso;**
- **capacità;**
- **lunghezza della parola o word** (ossia dimensione della minima cella indirizzabile);
- **velocità di trasmissione;**
- **metodi di accesso.**

### F) Struttura fisica di una memoria centrale

A livello hardware la memoria centrale è una memoria realizzata con uno o più **chip** (ossia piccole schegge di silicio della superficie di 1 cm<sup>2</sup>), contenenti un numero enorme di **circuiti integrati** inseriti in un materiale plastico e provvisti di piccolissimi piedini detti **pin** che permettono l'alimentazione ed il trasporto dei segnali elettrici agli altri componenti del sistema.



## LA CPU

**DEFINIZIONE:** L'**unità centrale o CPU** (Central Processing Unit) rappresenta il nucleo di tutto il sistema di elaborazione ed è responsabile dell'esecuzione e del controllo del processo esecutivo. Per questo motivo viene anche chiamata **processore centrale**.

**N.B.**

La ridottissima dimensione del dispositivo "fisico" utilizzato gli conferisce il nome di **microprocessore** o **microchip**, ma occorre tenere ben in mente che microprocessore e CPU non sono la stessa cosa anche se fisicamente coincidono.

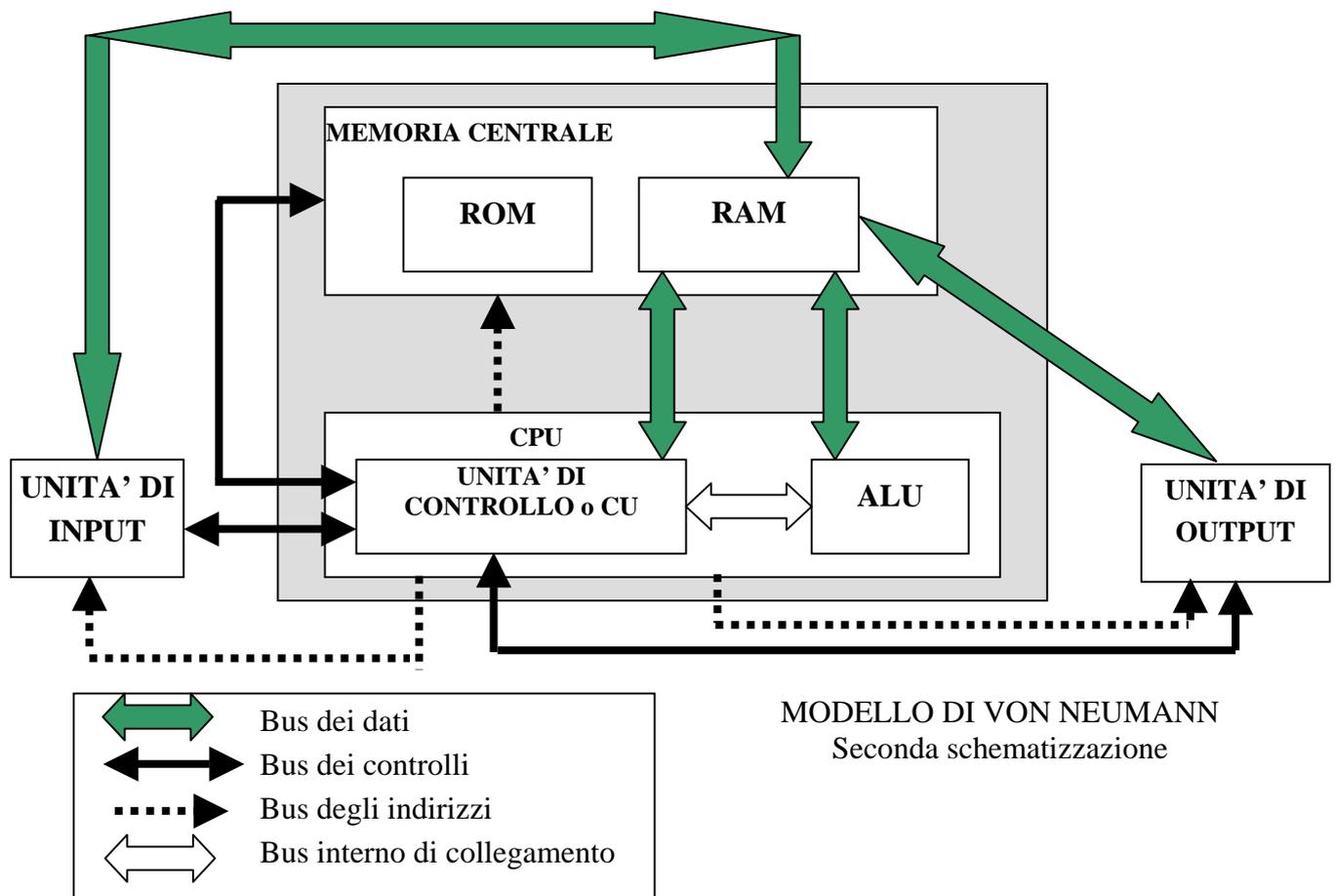
Con il termine **microprocessore** ci riferiamo all'oggetto fisico che si trova nel computer, mentre con il termine **CPU** ci riferiamo alla funzione svolta da tale oggetto.

La **CPU** ha il compito di:

- provvedere alla *registrazione in memoria centrale* dei dati e delle istruzioni dei programmi;
- riconoscere ed interpretare *le istruzioni del programma* da eseguire;
- eseguire *le operazioni logico-aritmetiche*;
- *controllare tutte le operazioni* necessarie per eseguire il processo di elaborazione e *gestire* il traffico di informazioni da e con l'esterno.

La CPU è composta da:

1. una **unità aritmetico-logica** detta generalmente **ALU** (Arithmetic Logic Unit);
2. una **unità di controllo** detta **CU** (Control Unit);
3. alcuni **registri**.



## LA CPU: L'UNITÀ ARITMETICO-LOGICA O ALU

**DEFINIZIONE: L'Unità Aritmetico-Logica o ALU** (Arithmetic Logic Unit) è una sottounità della CPU a cui spetta il compito di eseguire, sotto la continua supervisione della CU, le operazioni aritmetiche e logiche sui dati provenienti dalla memoria.

I componenti principali dell'**ALU** sono:

- una **rete logica di calcolo** formata da:
  - a) un *circuito addizionale* per eseguire le operazioni aritmetiche che utilizza *due parole o word in ingresso* ed *una parola o word in uscita* tutte codificate in binario;
  - b) alcuni circuiti per eseguire le operazioni logiche elementari corrispondenti ai connettivi logici (AND, NOT ed OR), lo shift, la determinazione del segno di un numero ed il confronto.
- uno o più **registri accumulatori (AC ossia Accumulate Register)**:  
utilizzati per contenere temporaneamente i dati provenienti dalla memoria centrale ed eseguire materialmente i calcoli.
- un **registro di stato (SR ossia Status Register)**:  
grazie al quale la ALU comunica alla CU informazioni sul tipo di risultato ottenuto dopo l'esecuzione delle operazioni. Questo registro è composto da un insieme di **bit** detti **flag** ad ognuno dei quali è attribuito un significato particolare.. Tra i **bit o flag di stato** più importanti ricordiamo:
  - a) **C** detto **bit di CARRY o di RIPORTO**: contiene il valore binario di 1 se nell'operazione eseguita dalla ALU c'è stato un riporto, 0 altrimenti;
  - b) **N** detto **bit di SEGNO o di NEGATIVITA'**: contiene il valore binario di 1 se il risultato dell'operazione eseguita dalla ALU è negativa, 0 altrimenti;
  - c) **Z** detto **bit di ZERO**: contiene il valore binario di 1 se il risultato dell'operazione eseguita dalla ALU è zero, 0 altrimenti;
  - d) **O** detto **bit di OVERFLOW**: contiene il valore binario di 1 se nell'operazione eseguita dalla ALU c'è stato un *trabocco*, 0 altrimenti;
  - e) **P** detto **bit di PARITA'**: contiene il valore binario di 1 se il risultato dell'operazione eseguita dalla ALU è costituita da un numero pari di cifre 1, 0 altrimenti

## LA CPU: L'UNITÀ DI CONTROLLO O CU

**DEFINIZIONE: L'Unità di Controllo (o di Governo) o CU** (Control Unit) ha il compito di gestire e sovrintendere al funzionamento di tutte le unità del computer ed all'esecuzione di tutti i *processi* che devono essere compiuti per eseguire un *programma* residente in memoria centrale.

I componenti principali della **CU** sono:

- un **registro dell'istruzione corrente o IR (Instruction Register)**:  
che *contiene l'istruzione* che la CU deve eseguire;
- un **registro contatore di programma o PC (Program Counter)**:  
che *contiene l'indirizzo* di memoria dove è contenuta l'istruzione successiva a quella che la CU sta eseguendo (ossia quella il cui codice è contenuta nel registro IR);

- **alcuni registri generali:**  
che contengono quei dati che possono essere utili durante l'esecuzione del programma (ad esempio dati o risultati intermedi);
- **il registro Stack Pointer o SP:**  
che viene utilizzato per gestire una zona di memoria con la tecnica della pila (ossia la **LIFO Last In First Out**) che permette di recuperare dati ma anche situazione del processore in caso di arrivo e ripristino di un **segnale di interruzione**;
- **un decodificatore:**  
che ha il compito di interpretare l'istruzione contenuta nel registro IR al fine di riconoscerne il tipo;
- **un temporizzatore o clock:**  
ossia un oscillatore al quarzo che genera una serie di impulsi a frequenza regolare, utilizzato per sincronizzare l'invio di tutti i segnali elettrici inviati dalla CU e quindi in grado di attivare le varie unità o le varie operazioni.

Grazie a tutti questi dispositivi "fisici" la CU è in grado di svolgere le sue principali funzioni:

- a) **prelevare** le istruzioni del programma (.EXE) da eseguire registrate in memoria centrale;
- b) **interpretare**, volta per volta, le istruzioni lette;
- c) **eseguire** le istruzioni attivando e controllando le unità coinvolte nella esecuzione.

#### LA CPU: LINGUAGGIO MACCHINA E FORMATO DELLE ISTRUZIONI

Le istruzioni di un programma da eseguire (detto anche **eseguibile** o **.EXE**), residente in memoria centrale, sono codificate in **linguaggio macchina** ossia nell'unico linguaggio comprensibile dalla CPU (che è l'*esecutore* del programma).

Il **linguaggio macchina** è composto da un insieme di istruzioni macchina espresse usando il codice binario. Tali istruzioni svolgono una serie di funzioni molto elementari eseguibili direttamente dalla CPU.

**Dal punto di vista funzionale** le istruzioni del linguaggio macchina possono essere suddivise in **cinque gruppi**:

**1° GRUPPO) istruzioni di input/output:** con queste istruzioni si opera sui dispositivi che trasferiscono dati dalle unità di input alla memoria centrale (ad esempio *Leggi (n)*) o dalla memoria centrale alle unità di output (ad esempio *Scrivi (m)*);

**2° GRUPPO) istruzioni aritmetico-logiche:** con queste istruzioni si eseguono operazioni di calcolo (quali addizione, sottrazione, etc.) ed operazioni logiche (AND, OR, NOT, etc.);

**3° GRUPPO) istruzioni di controllo e salto:** con queste istruzioni si interviene per modificare lo svolgimento di un programma in fase di esecuzione in due modi:

- saltando in modo incondizionato (senza il verificarsi di nessuna condizione) verso un altro punto del programma;
- saltando in modo condizionato come conseguenza del risultato di un test un determinato gruppo di istruzioni.

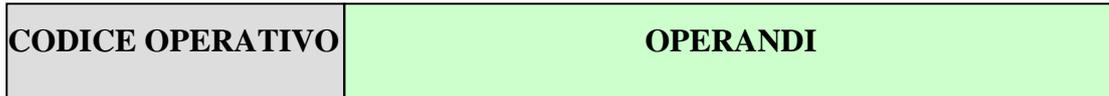
**4° GRUPPO) istruzioni per lo spostamento dei dati:** con queste istruzioni si spostano i dati all'interno della memoria centrale (ad esempio l'istruzione di assegnazione);

**5° GRUPPO) istruzioni di riordino o di controllo macchina:** con queste istruzioni non si opera direttamente per l'esecuzione di un programma, ma per ripristinare le condizioni del computer (ad esempio riavvolgi il nastro n°3)

Per quanto riguarda la **lunghezza di una istruzione macchina** essa può variare da computer a computer, in quanto ognuno ha il proprio linguaggio macchina. In generale ogni istruzione macchina può avere una lunghezza pari ad un multiplo della *parola o word di memoria*.

Per quanto riguarda la **forma di una istruzione macchina** essa può essere suddivisa in **due campi**:

- un campo **codice operativo** che identifica il *tipo di operazione* da eseguire;
- un campo **operandi** che contiene i *dati* o gli *indirizzi dei dati* sui quali sarà eseguita l'operazione descritta dal codice operativo oppure un riferimento a registri. Tale campo può essere composto da più campi *ma non supera mai il numero di tre*.



**N.B.** Molto spesso gli operandi sono uno (operazioni unarie ) oppure due (operazioni binarie) oppure nessuno. In tal caso allora in assenza di operandi l'istruzione sarà composta dal solo codice operativo.

Nel caso ci fossero presenti tre operandi, il terzo indica normalmente dove va posto il risultato dell'operazione eseguita sui primi due.

*Esempio: Supponiamo che l'istruzione da eseguire scritta in pseudolinguaggio sia.*

$$c \leftarrow a + b$$

*Naturalmente i dati contenuti nelle variabili  $a$  e  $b$  sono contenuti nella memoria centrale in apposite celle individuate da appositi indirizzi. Supponiamo che l'indirizzo della cella ove è contenuta la variabile  $a$  sia **1010** mentre quello contenente la variabile  $b$  sia **1011**.*

*Un esempio di istruzione scritta in linguaggio macchina che realizzi tale istruzione potrebbe essere la seguente:*

0101	1010	1011	1101
------	------	------	------

*dove abbiamo immaginato che il codice operativo dell'operazione somma sia **0101** ed il risultato venga riposto nella variabile  $c$  contenuta all'indirizzo **1101**.*

*L'istruzione macchina potrebbe essere così riassunta: “**effettua l'operazione con codice operativo 0101 (somma) del dato contenuto nella cella con indirizzo 1010 con il dato contenuto nella cella con indirizzo 1011 e poni il risultato ottenuto nella cella con indirizzo 1101**”.*

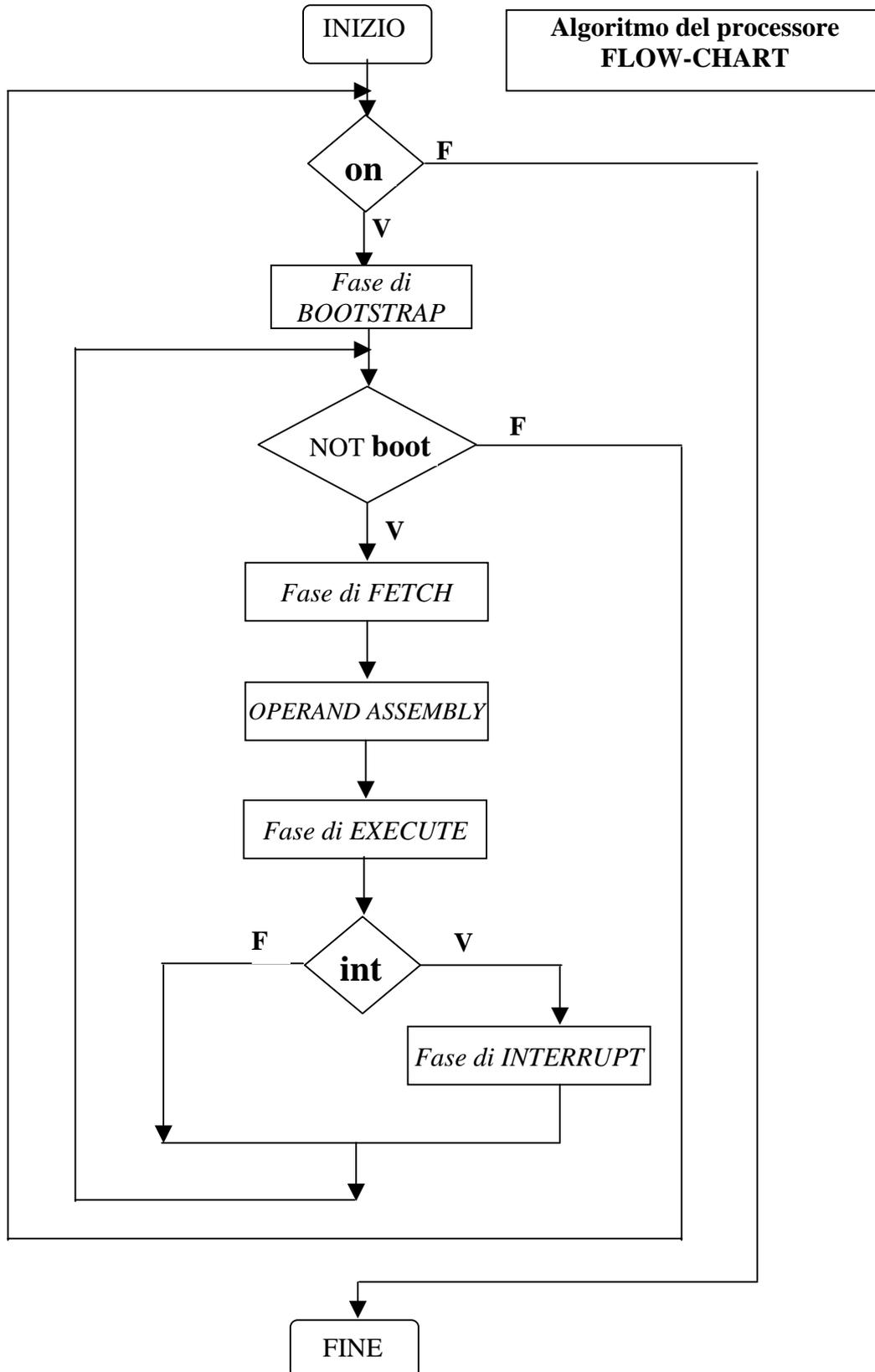
Appare evidente che per eseguire una singola istruzione macchina, la CU è costretta ad eseguire un insieme di passi elementari che devono corrispondere ad istruzioni elementari direttamente eseguibili dai componenti hardware del computer.

In passato l'esecuzione di una singola istruzione macchina veniva svolta dalla CU per mezzo di una **rete di circuiti logici** in grado di effettuare e controllare tutte le operazioni elementari necessarie alla sua esecuzione (**elaboratori a logica cablata**).

Adesso le istruzioni macchina non sono più le istruzioni di livello più basso. Ne esistono altre di livello più basso dette **microistruzioni** che vengono registrate all'interno della ROM che messe insieme in **microprogrammi** appositamente predisposti realizzano quanto previsto da ciascuna istruzione macchina. Tali microprogrammi memorizzati permanentemente nella memoria ROM costituiscono il firmware della macchina (**elaboratori a logica programmata**).

### LA CPU: L'ESECUZIONE DELLE ISTRUZIONI

L'elaborazione delle istruzioni di un programma contenuto nella memoria centrale consiste nella esecuzione da parte della CU di una precisa successione di passi elementari detto **algoritmo del processore** che qui ora analizzeremo in dettaglio.



**Algoritmo del processore  
PSEUDOCODIFICA**
**ALGORITMO** Processore

BOOL on, boot, int

**INIZIO**
**MENTRE** on **ESEGUI**
*Fase di Bootstrap*
**MENTRE** NOT boot **ESEGUI**
*Fase di FETCH*
*OPERAND ASSEMBLY*
*Fase di EXECUTE*
**SE** int

**ALLORA**
*Fase di INTERRUPT*
**FINE SE**
**FINE MENTRE**
**FINE MENTRE**
**FINE**
**LEGENDA**

**on**: variabile booleana indicante che il processore è acceso  
ossia è alimentato dalla corrente elettrica;

**boot**: variabile booleana che rappresenta una condizione  
che può essere creata dall'operatore di macchina pigiando  
un tasto o una sequenza di tasti (ad esempio  
CTRL+ALT+CANC) o scegliendo un'opzione a menù del  
sistema operativo attraverso la quale si richiede una  
“inizializzazione” del sistema (bootstrap);

**int**: variabile booleana che indica l'attivazione di un  
segnale di interruzione da gestire;

**Fase di BOOTSTRAP, Fase di FETCH, OPERAND  
ASSEMBLY, Fase di EXECUTE e Fase di  
INTERRUPT** rappresentano le varie fasi del processore  
che saranno spiegate e particolarizzate in seguito

 1) Fase di BOOTSTRAP

La fase di *bootstrap* è detta anche di *inizializzazione* o di *start* si svolge nel seguente modo:

- a) all'accensione della macchina viene caricato “in hardware” - ossia utilizzando prefissati indirizzi e locazioni di memoria senza utilizzare il normale ciclo *fetch-execute* - nell'indirizzo **PC** della **CU**, l'indirizzo della prima istruzione del programma contenuto nella **ROM**;
- b) quindi viene eseguito da CU nella sua totalità il programma contenuto nella ROM che ha come obiettivo quello di *caricare* nella RAM il programma *loader* (caricatore) del sistema operativo nonché di eseguire anche il **BIOS** della macchina;
- c) una volta completata l'esecuzione del programma contenuto nella ROM viene *eseguito* da CU il programma *loader* del S.O. presente in memoria centrale che provvederà al trasferimento nella RAM dei programmi fondamentali del sistema operativo;
- d) quando viene eseguito un “programma utente”, il sistema operativo già presente in memoria centrale provvederà a caricare nell'indirizzo PC della CU la sua prima istruzione da eseguire e proseguirà utilizzando il normale ciclo *fetch-execute* (VEDI DESCRIZIONE DETTAGLIATA SUCCESSIVA)

Quindi tale fase verifica sempre le seguenti due condizioni:

- il programma che si desidera eseguire è in memoria centrale o RAM;
- il registro PC di CU contiene l'indirizzo dell'istruzione da eseguire.

## 2) Fase di FETCH

La fase di *fetch* è detta anche “**prelevamento dell’istruzione**” o semplicemente “**fase istruzione**” ha il compito di prelevare l’istruzione dalla memoria centrale o RAM trasferendola all’interno del processore (nel registro PC di CU) e di predisporre quest’ultimo all’esecuzione dell’istruzione successiva.

## 3) OPERAND ASSEMBLY

La fase di *operand assembly* è detta anche “**preparazione degli operandi**” ha l’obiettivo di rendere disponibili nei registri interni della CPU gli operandi dell’istruzione correntemente caricata.

## 4) Fase di EXECUTE

La fase di *execute* è detta anche “**esecuzione dell’istruzione**” ha il compito di effettuare la vera e propria esecuzione dell’istruzione correntemente caricata.

Se l’istruzione corrente è:

- **di input**: la CU comanda alle unità di input preposte di introdurre i dati in memoria centrale;
- **di output**: la CU comanda alle unità di output preposte di inviare all’esterno i dati registrati in memoria centrale;
- **di elaborazione**: la CU comanda il prelevamento dei dati dalla memoria centrale ed il successivo invio alla ALU alla quale assegna i compiti per la corretta elaborazione. Poi provvederà a registrare i risultati ottenuti nella memoria centrale.

## 5) Fase di INTERRUPT

Un **segnale di interruzione** o più semplicemente una interruzione è un segnale che provoca la sospensione del programma in corso ed il salto, realizzato **via hardware**, ad un indirizzo prefissato di memoria dove è registrato un particolare sottoprogramma detto **Sottoprogramma Di Interruzione o ISR** (Interrupt Service Routine).

Grazie all’ISR si “**serve**” l’**interruzione** ossia si effettuano tutte le azioni elaborative richieste dall’interruzione stessa. Il programma “*interrotto*” dovrà poter essere ripreso, nel caso in cui tale ripresa è compatibile con l’interruzione stessa, dopo che tali azioni sono state eseguite.

E’ importante sottolineare come una causa di interruzione non provochi di per se una interruzione (ossia  $\text{int} \leftarrow \text{TRUE}$ ) ma soltanto l’invio di una **richiesta di interruzione** al sistema operativo. Affinchè questa provochi realmente una interruzione occorre che sia di fatto **abilitata**.

Il sistema operativo è dunque in grado di disciplinare **via software** la strategia delle interruzioni nelle varie fasi dell’evoluzione di un programma, esaminando di volta in volta la causa dell’interruzione.

La causa di un interruzione è in generale un evento *esterno* al programma in corso ed *asincrono* con esso ossia non è in generale prevedibile a priori l’istante in cui essa si verificherà.

Tipiche cause di interruzione sono:

- a) interruzione per richiesta di attenzione: è generata da una richiesta periferica di collegarsi al sistema centrale (può essere generata dall’operatore di console o da terminali distanti);
- b) interruzione di ingresso-uscita: è generata da una unità periferica tipicamente al termine di una operazione di I/O al fine di sincronizzarsi con l’unità centrale;
- c) interruzione per errori nel programma: è generata dall’hardware della stessa CPU a causa di errori di programmazione incompatibili con il funzionamento della macchina (errore di overflow, indirizzo di memoria illecito in quanto non appartenente alla memoria del sistema oppure violante la protezione di memoria, etc.);

- d) interruzione per guasti del sistema: mancanza o insufficienza di energia elettrica, temperatura troppo alta o troppo bassa, etc.

### **DETTAGLIO dell'esecuzione di un programma "utente" – CICLO DI ISTRUZIONE**

- 1) l'esecuzione di un programma, caricato interamente nella memoria centrale (o RAM) da parte del sistema operativo, inizia **sempre** con l'assegnazione al registro **PC** dell'unità CU dell'indirizzo della sua prima istruzione;
- 2) l'indirizzo di memoria contenuto nel registro PC viene trasferito nel registro **MAR** della memoria centrale attraverso **il bus degli indirizzi**;
- 3) la memoria centrale preleva l'istruzione il cui indirizzo è indicato nel registro MAR e la carica nel registro **MDR**;
- 4) il contenuto del registro MDR ossia l'istruzione del programma in linguaggio macchina viene inviato dalla memoria centrale tramite **il bus dei dati** alla CU e giunto a destinazione viene depositato nel registro **IR**;
- 5) il **decodificatore** delle istruzioni presente in CU provvede ad interpretare il **codice operativo** dell'istruzione presente in IR per capire il tipo di operazione da effettuare ed identificare il microprogramma necessario per eseguirla;
- 6) il **decodificatore** incrementa il registro PC in modo da avere a disposizione l'indirizzo della successiva istruzione da eseguire (Le istruzioni di un programma caricato in memoria centrale occupano celle contigue. L'indirizzo di ciascuna istruzione viene ricavata dall'indirizzo della precedente istruzione più la lunghezza della **parola o word** amessa);
- 7) la CU attiva il microprogramma preposto ad eseguire l'operazione richiesta dall'istruzione. Se sono previsti calcoli aritmetici verrà attivata anche l'ALU che provvederà a prelevare dalla memoria centrale gli operandi ed a depositarli negli appositi registri accumulatori;
- 8) terminata l'esecuzione i risultati presenti nei registri accumulatori vengono trasferiti attraverso il bus dei dati in memoria centrale;
- 9) l'esecuzione del programma prosegue dalla nuova istruzione "puntata" dal registro PC

Questo procedimento illustrato si chiama **CICLO DI ISTRUZIONE** ed è costituito da più *microoperazioni*. L'esecuzione delle *microoperazioni* avviene per passi elementari detti **CICLI DI MACCHINA** (o **CICLI DI CLOCK**) che rappresentano una serie di operazioni elementari (microistruzioni) che la CU deve eseguire ogni volta che deve accedere in memoria centrale o alle unità di I/O.

Ogni singolo ciclo di macchina deve avvenire in modo regolare e non disordinato. Per questo tutti i segnali inviati dalla CU vengono impartiti in modo sincrono con un orologio interno alla macchina detto **clock**. Ad ogni colpo di clock viene inviato un segnale e quindi viene compiuta una microistruzione.

Il clock è naturalmente un orologio molto veloce con **frequenza** nell'ordine dei **Mega/Giga Hertz** ossia **milioni/miliardi di oscillazioni al secondo**.

*Esempio: un PC con un processore che lavora ad 800 MHz (MegaHertz) è in grado di effettuare 800 milioni di istruzioni elementari al secondo.*

Questa frequenza è un importantissimo **indice della velocità** del computer. Mediamente però per poter effettuare alcuni tipi di operazioni sono necessari più colpi di clock e per questo motivo che la velocità del computer viene misurata in **MIPS** ossia **Millions of Instructions Per Second** = milioni di istruzioni al secondo.

Ovviamente la velocità di un processore espressa in MIPS è sempre inferiore a quella espressa in MHz.

## LE PERIFERICHE

Secondo il modello funzionale di Von Neumann, un elaboratore è composto oltre dalla memoria centrale e dalla CPU, dalle **unità di input/output** ossia **unità di I/O**.

Queste unità vengono spesso chiamate **periferiche** in quanto sono dei *dispositivi* (in inglese *device*) esterni all'elaboratore.

Il loro compito è quello di fungere da collegamento tra il computer (inteso come memoria e CPU) ed il mondo esterno (l'utente).

Un computer è composto generalmente da più periferiche di diverso tipo:

1. alcune si occupano dell'introduzione delle informazioni (**periferiche di input**);
2. alcune permettono la visualizzazione dei risultati delle elaborazioni (**periferiche di output**);
3. altre permettono di memorizzare permanentemente dati e programmi allo scopo di poterli riutilizzare (periferiche di memorizzazione di massa o **memorie di massa**).

Ogni periferica sia essa di input o di output necessita di un'apposita **interfaccia** per gestire il *colloquio* ossia lo *scambio dati* con la CPU.

L'interfaccia di una periferica è costituita da:

- a) dispositivi hardware: **interfacce di I/O e controller**
- b) dispositivi software: **driver di periferica**

**Le interfacce di I/O**: sono anche dette **schede**.

Sono una serie di circuiti specifici per ogni dispositivo periferico che permettono di instaurare un *dialogo* ossia uno *scambio dati* tra la periferica e la CPU.

*Esempio: la scheda audio e la scheda modem/fax.*

Il collegamento delle periferiche avviene tramite cavi connettori che si connettono con apposite prese dette **porte**.

Esse vengono alloggiare in appositi connettori (fessure della scheda madre) detti **slot**.

Dal punto di vista della trasmissione dati le porte si distinguono in **seriali** e **parallele**.

Nelle **porte seriali** i dati viaggiano in serie ossia un bit dopo l'altro (modem, mouse, tastiera,...).

Un particolare tipo di porta seriale è la **porta MIDI** (**M**usical **I**nstrumental **D**igital **I**nterface) che permette di collegare al computer vari tipi di strumenti musicali.

Nelle **porte parallele** i dati viaggiano in gruppo ossia un gruppo bit alla volta (stampante, plotter,...). Un particolare tipo di porta parallela è la **porta SCSI** (**S**mall **C**omputer **S**ystem **I**nterface) che consente di collegare più dispositivi insieme (ad esempio dischi rigidi)

Ovviamente i cavi connettori sono fisicamente diversi a seconda del tipo di porta che devono connettere rendendo di fatto impossibile connettere un cavo parallelo ad una porta seriale e viceversa.

La **porta PCMCIA** (**P**ersonal **C**omputer **M**emory **C**ard **I**nternational **A**ssociation) può permettere la connessione in genere ai notebook ed ai palmari di **schede PCMCIA** che contengono modem, hard disk, schede di rete. Queste schede sono speciali adattatori del formato di una carta di credito specificatamente studiato per le connessioni per i pc portatili realizzate secondo uno standard condiviso da oltre 400 società produttrici di hardware e software.

Le **porte USB** (**U**niversal **S**erial **B**us) oramai molto comuni (e di conseguenza le periferiche che utilizzano questo tipo di connessione per interfacciarsi all'elaboratore) sono state introdotte per sostituire le porte seriali più lente e supportano il *collegamento a cascata* di numerose periferiche. Consentono di evitare l'utilizzo di slot interni al pc e sono facilmente utilizzabili anche da utenti meno esperti.

Una caratteristica interaccia di I/O che merita un discorso a parte è la **scheda video**.

La scheda video è un vero e proprio processore dedicato alla gestione del video con una propria RAM dedicata in grado di poter coadiuvare la CPU dell'elaboratore durante le fasi impegnative della elaborazione delle immagini, alleggerendone il tempo di impiego.

Sono caratterizzate da una **classe** (VGA, SVGA, etc.) e la loro qualità dipende dalla quantità di **pixel** (**picture element**) che riescono a visualizzare all'interno del video e dalla velocità di elaborazione delle immagini (maggiori dettagli quando si parlerà dei monitor)

Un qualsiasi modulo di interfaccia di I/O contiene una particolare memoria detta **buffer** o *memoria di transito* sulla quale transitano i dati in ingresso oppure in uscita. L'utilizzo del buffer è quello di rendere **asincrona** la comunicazione tra CPU e periferiche, svincolando la velocissima CPU dall'attendere il completamento di operazioni da parte di dispositivi molto più lenti.

Infatti se la CPU dovesse trasferire i dati ad una generica periferica di I/O ed aspettare che quest'ultima termini l'operazione, aumenterebbero a dismisura i tempi morti con conseguente calo delle prestazioni.

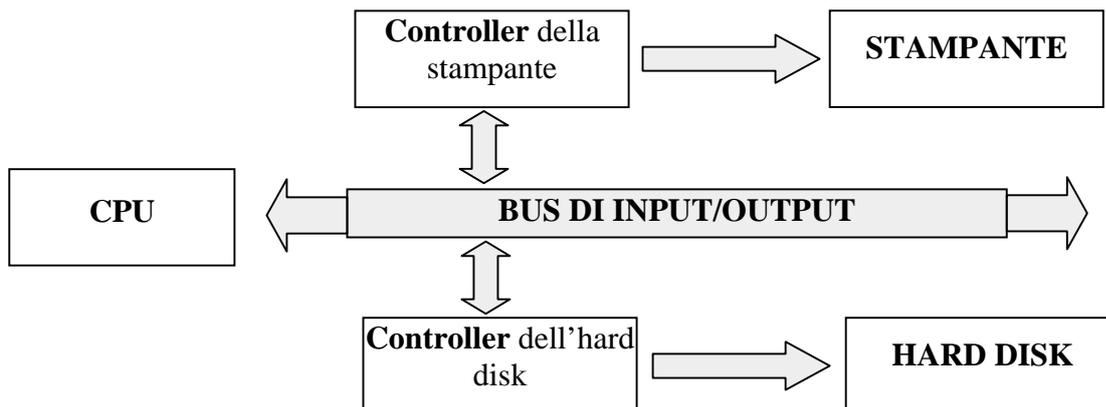
Invece, grazie all'utilizzo del buffer, la CPU invia i dati ad una periferica memorizzandoli nel buffer per una generica operazione di output oppure li preleva dal buffer per una generica operazione di input, allorché la periferica segnala la loro disponibilità svincolandosi dai tempi "fisiologici" propri della periferica e continuando il suo lavoro in attesa di una nuova richiesta.

Infine un qualsiasi modulo di interfaccia di I/O contiene una propria porta di I/O che ne permette il collegamento fisico con il sistema dei bus.

**Quindi in definitiva l'interfaccia ha il compito di collegare il sistema dei bus presente nell'elaboratore al buffer presente nella periferica per poter realizzare il trasferimento dei dati.**

**Il controller** o **controllore** della periferica ha il compito di eseguire tutte le funzioni di controllo del dispositivo: guida i movimenti meccanici del dispositivo. Funge inoltre da interprete tra CPU e periferica in modo da tradurre i messaggi scambiati..

**Il driver di periferica** o **driver** è un particolare programma che gestisce i segnali di comunicazione tra CPU e periferica.



## LE UNITA' (O PERIFERICHE) DI INPUT

**A) La tastiera ( o keyboard):** è il più tradizionale e diffuso dispositivo per l'immissione dei dati. E' simile ad una macchina da scrivere con molti più tasti e diversamente distribuiti.

*Cosa succede quando premiamo un tasto?*

Il carattere viene subito codificato in binario e l'interfaccia provvede ad accumularlo nel *buffer per l'input* da dove una volta riempita saranno inviati all'unità centrale. La tastiera è inoltre provvista di un proprio buffer di tastiera generalmente piccola (una decina di caratteri) nella quale vengono inseriti i caratteri battuti troppo rapidamente o comunque prima di poter essere accettati dall'elaboratore.

**B) Il puntatore ( o mouse):** è formato da una scatoletta con 2 o 3 oppure 5 tasti nella quale è inserita una sfera (mouse tradizionale). Fa parte di una particolare tipologia di dispositivi detta **dispositivi di puntamento**.

*Cosa succede quando muoviamo il mouse?*

Il movimento su di un piano fa ruotare la sfera permettendo così di muovere un *puntatore* (in genere a forma di freccia) sullo schermo nel punto stabilito dall'utente, il quale potrà così scegliere tra le varie attività possibili. La forma generale di freccia viene sostituita dalla forma di **cursore** nelle applicazioni che prevedono l'immissione di testo da parte dell'utente.

Sempre più comuni sono i **mouse ad infrarossi o ottici** (detti anche **optical wireless**) senza cavo di collegamento al computer e senza il problema della pulizia del meccanismo di movimento.

**C) La track-ball:** è analoga al mouse essendo di fatto un mouse al rovescio. E' costituita da una scatoletta fissa e da una sfera mobile e rotante che permette di indirizzare il puntatore.

**D) Lo scanner:** è un dispositivo simile ad una fotocopiatrice in grado di leggere testi scritti oppure disegni, nonché immagini di qualsiasi tipo digitalizzandoli ossia trasformandoli in segni comprensibili dal computer. Una volta digitalizzate, le immagini prodotte possono essere modificate utilizzando software specifici.

**F) La penna ottica:** è un dispositivo che sfrutta un laser (fascio di luce) per leggere codici come quelli a barre. Viene usata ovunque dai centri commerciali alle farmacie.

**G) La tavoletta grafica:** è un dispositivo formato da un piano di lavoro e da una penna che permette di trasportare tutto ciò che viene disegnata su di essa direttamente sul video.

**H) Lo schermo tattile o touch-screen:** è un dispositivo usato spesso nei posti pubblici (gazebo informazioni e postazioni interattive). Sulla superficie dello schermo c'è una fitta rete di raggi infrarossi e quando l'utente tocca un punto del video interrompe un raggio permettendo di individuare esattamente il punto dello schermo toccato e quindi la funzionalità attivata.

**I) Sistemi di riconoscimento vocali:** sono prodotti in grado di "capire" alcune parole pronunciate da un "utente tipo" ed opportunamente memorizzate in un dizionario e conseguentemente mettere in atto la funzionalità richiesta.

**L) Macchine fotografiche digitali:** sono prodotti in grado memorizzare fotografie e filmati che possono essere successivamente modificati utilizzando programmi specifici.

## LE UNITA' (O PERIFERICHE) DI OUTPUT

**A) Il monitor (o video):** è il principale dispositivo di output. Simile ad un normale televisore, il monitor tradizionale è costituito da un **tubo a raggi catodici** (detto **video CRT** o **Catode Ray Tube**) e da uno **schermo trasparente** trattato con fosforo che è *elettroluminescente*.

*Come vengono visualizzate le immagini?*

Opportunamente guidato, il tubo colpisce lo schermo trattato con degli elettroni, illuminando i punti colpiti. I punti colpiti si chiamano **pixel** (ossia **picture element**) e costituiscono una caratteristica fondamentale del monitor in quanto definiscono la loro **risoluzione** ossia il massimo numero di pixel visualizzabili sullo schermo.

Un qualsiasi oggetto visualizzabile su di uno schermo è ottenuto attraverso una determinata disposizione di pixel: quanti più pixel ci sono (ossia quanto più il monitor permette una risoluzione alta) tanto più nitida sarà l'immagine proposta.

*Esempio: nella risoluzione "classica" di un video 800 x 600 pixel, sul monitor ci sono 800 colonne verticali ciascuna delle quali costituita da 600 pixel.*

La distanza tra i pixel visualizzati sullo schermo è detta **dot pitch** ed è espressa in *frazioni di millimetro*. Varia a seconda della qualità del video. Più piccola è tale distanza tanto più nitida sarà l'immagine proposta.

*Esempio: per un monitor da 15" un dot pitch di 0,28 millimetri è ritenuto buono ossia la distanza tra un pixel ed un altro, indipendentemente dalla risoluzione video, è sempre pari a 0,28 millimetri.*

Quanto più è sofisticata l'immagine maggiore sarà la quantità di memoria necessaria per elaborarla ed il tempo di elaborazione. Per evitare quindi rallentamenti dovuti all'eccessiva occupazione di memoria con il conseguente calo delle prestazioni del sistema, sulla **scheda video** vengono aggiunte sia memoria (la cosiddetta **memoria video**) sia un processore dedicato (il cosiddetto **chip** della scheda video), in grado di gestire e visualizzare le operazioni che coinvolgono le immagini.

Per quanto riguarda i monitor tradizionali CRT, l'architettura con la quale sono costruiti non permette di vedere un'intera immagine contemporaneamente su tutta la superficie dello schermo, bensì essa viene disegnata dall'alto verso il basso e da sinistra verso destra e viene costantemente aggiornata anche se apparentemente ferma. La frequenza con la quale ciascuna immagine viene ridisegnata prende il nome di **frequenza di refresh** che dipende dalla risoluzione video utilizzata. Quanto più è alta la frequenza di refresh tanto più è nitida e stabile l'immagine proposta.

*Esempio: per un monitor CRT un video con risoluzione 640x480 ha un refresh di 120 Hz, che passa a 90 Hz per una risoluzione di 800x600 e che infine scende a 60 Hz con una risoluzione 1280x1024.*

**I monitor a cristalli liquidi o LCD** (ossia **Liquid Crystal Display**): analoghi ai monitor dei computer portatili oppure agli schermi televisivi di ultima generazione (quelli cosiddetti **a matrice attiva**), non hanno il tubo catodico dei monitor CRT. Essi non emanano radiazioni (non utilizzando elettroni) ed occupano un minor spazio.

**B) La stampante ( o printer):** insieme al video costituisce l'altro principale dispositivo di output per eccellenza. Il suo compito è quello di provvedere alla stampa su carta a fogli singoli oppure a modulo continuo. Forniscono in questo modo una **hardcopy** ossia una copia permanente dei dati.

Le attuali stampanti sono di piccole dimensioni, veloci ed a prezzi relativamente modici.

La qualità delle stampe prodotte dipende dalla **risoluzione di stampa** o **DPI** ossia **Dot Per Inch** (ossia **punti per pollice**)

*Esempio: una stampante che ha la risoluzione di 600x600 dpi, è in grado di stampare su ogni pollice quadrato (ossia circa 6,45 centimetri quadr – 2,54 cm x 2,54 cm) fino a 360.000 (pari a 600x600) punti.*

*Come si classificano le stampanti?*

Le stampanti possono essere classificate in base a diversi criteri:

a) dal punto di vista della **modalità di stampa**: si classificano in:

(-) **stampanti seriali**: sono stampanti che stampano un carattere per volta fino al completamento della riga. La velocità di queste stampanti si misura in **CPS** ossia **caratteri per secondo**;

(-) **stampanti parallele**: sono stampanti che stampano una riga per volta. Sono più veloci delle stampanti seriali e la loro velocità si misura in **LPM** ossia **linee per minuto**;

(-) **stampanti a pagina**: sono stampanti che stampano in blocco l'intera pagina come accade per le fotocopiatrici. La velocità di queste stampanti si misura in **LPM** ossia **linee per minuto** oppure in **PPM** ossia **pagine per minuto**;

b) dal punto di vista della **tecnica di stampa**:

(-) **stampanti ad impatto**: sono stampanti che imprimono il carattere sulla carta in modo meccanico ossia attraverso la pressione del **mezzo di scrittura** sul nastro inchiostro. Tale mezzo di scrittura può essere formato da:

(-) un dispositivo contenente tutti i caratteri stampabili (come una normale macchina da scrivere) ed in tal caso si parla di **stampa a carattere pieno o a tratto continuo**;

(-) un dispositivo contenente un determinato numero di aghi che costruiscono il carattere da stampare attraverso l'avvicinamento di più ed in tal caso si parla di **stampa a matrice di punti**;

(-) **stampanti a non impatto**: sono stampanti che **non** imprimono il carattere sulla carta in modo meccanico ma che utilizzano tecnologie più sofisticate, avanzate e silenziose a carattere *chimico, magnetico, elettrico ed ottico*. Le principali stampanti che sfruttano queste tecnologie sono:

(-) **stampanti ink-jet**: dette anche **a getto di inchiostro**. Con questo tipo di stampante, la stampa viene prodotta proiettando l'inchiostro sulla carta per mezzo di un microscopico ugello che spruzza l'inchiostro in piccolissime quantità in modo di formare un carattere a *matrice di punti*. Sono stampanti ad alta capacità grafica e consentono stampe a colori con ottime gradazioni;

(-) **stampanti bubble-jet**: dette anche **a bolle di inchiostro**. E' una particolare stampante ink-jet che sfrutta il riscaldamento dell'inchiostro nell'ugello fino al punto di ebollizione in modo da formare una bolla. L'esplosione della bolla spinge l'inchiostro nell'ugello proiettandolo sulla carta e formando il carattere;

(-) **stampanti laser**: sono le stampanti più potenti e performanti attualmente esistenti e si basano sulla tecnologia ottica. Il loro funzionamento si basa sulla rotazione di un cilindro rivestito di materiale conduttore. Il cilindro viene colpito durante la stampa da un raggio laser deviato da uno specchio. I punti illuminati trattengono l'inchiostro secco (toner) che dopo una fase di compressione e riscaldamento verrà proiettato sulla carta formando il carattere.

Concludiamo il discorso sulle stampanti con un ultimo concetto: il **font**.

Tutti i caratteri stampabili, ognuno con la propria forma ed il proprio stile, vengono raggruppati in famiglie chiamate **font**. Ogni font viene gestito sia dal video che dalla stampante in modo da ottenere sulla carta l'esatta immagine del carattere visualizzato (**WYSIWYG** ossia **What You See Is What You Get**). Molto spesso ciò non accade a causa della diversa risoluzione tra stampante e video (le stampanti hanno risoluzioni maggiori del video).

## LE MEMORIE DI MASSA

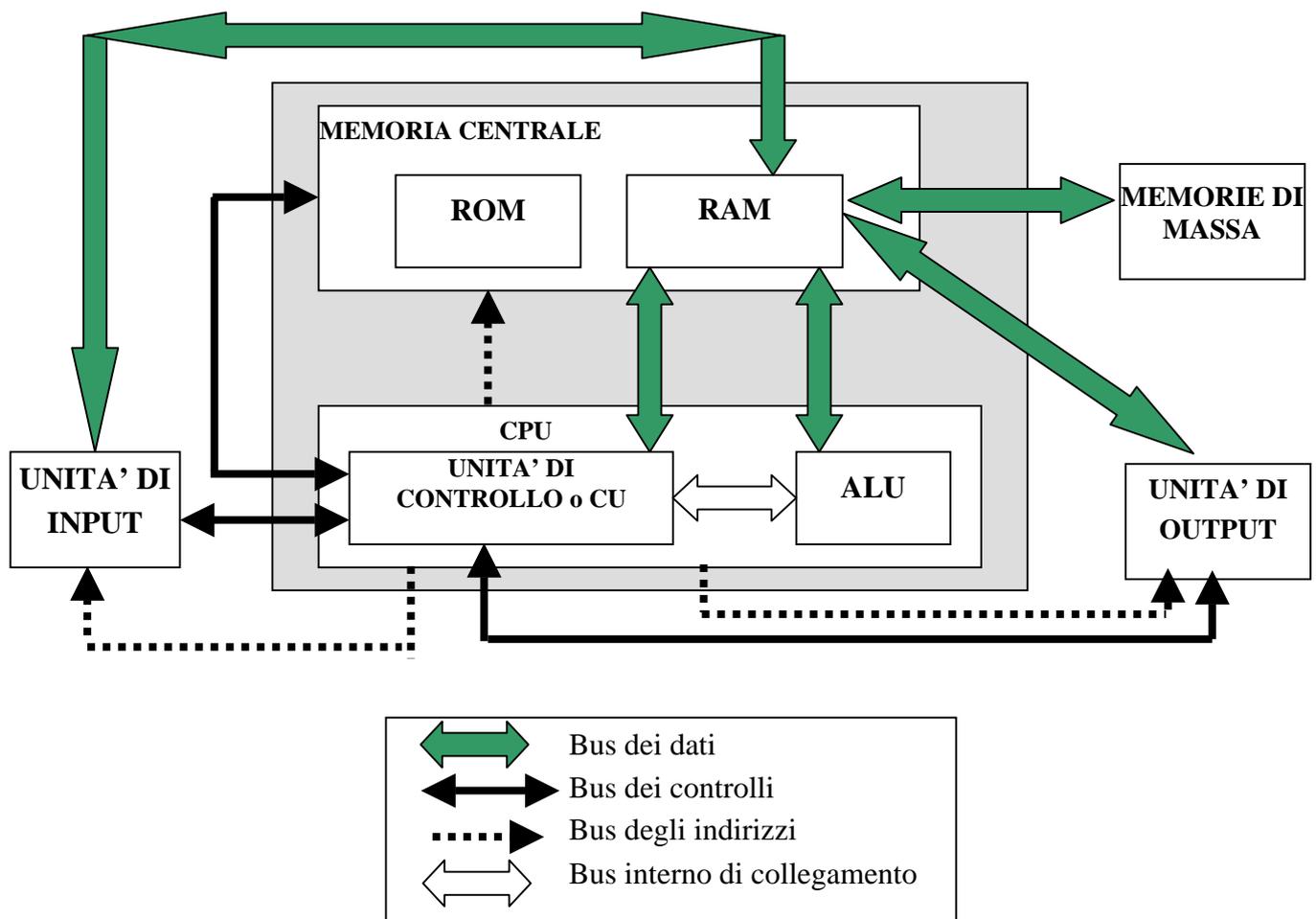
Le **memorie di massa** o **memorie ausiliarie** sono quelle memorie sulle quali è possibile esclusivamente conservare dati e programmi, senza poterli direttamente elaborare.

Quindi per potere elaborare i dati o eseguire i programmi registrati su di una memoria di massa, occorre innanzitutto trasferire dati e programmi nella memoria centrale o memoria di lavoro che è l'unica memoria sulla quale la CPU può operare.

Queste memorie presentano le seguenti caratteristiche generali:

- (-) **conservano** permanentemente i dati;
- (-) possono **contenere** grandissime quantità di dati;
- (-) sono memorie **trasportabili**;
- (-) sono memorie **meno costose** della memoria centrale;
- (-) sono memorie **più lente** della memoria centrale.

### MODELLO FUNZIONALE DI VON NEUMANN: Schematizzazione completa



I dispositivi di memoria di massa più comuni sono:

- a) **le memorie magnetiche;**
- b) **le memorie ottiche.**

**a) Le memorie magnetiche:** in generale sono costituite da un supporto piano ricoperto di materiale ferromagnetico sul quale è possibile memorizzare le informazioni magnetizzando apposite **areole** (l'equivalente delle celle di memoria). Il fenomeno di *magnetizzazione* viene utilizzato in quanto ben si presta alla memorizzazione di segnali digitali avendo due soli stati possibili in base alla direzione del campo magnetico e potendovi associare le due cifre binarie **0** ed **1**.

La registrazione avviene mediante una *testina* sul quale è avvolto a spirale un *filo conduttore*. A seconda del *verso* con cui passa la corrente la spirale genera un campo magnetico, effettuando una memorizzazione permanente.

Il passaggio delle areole sotto la testina avviene mediante un movimento meccanico a velocità costante del supporto di memorizzazione. La stessa *testina* vista in precedenza si occuperà anche di leggere i dati memorizzati (**testina di lettura/scrittura**).

Tutte le memorie di massa magnetiche hanno in comune i seguenti parametri:

- 1- **velocità di movimento** del supporto di memorizzazione misurata il *pollici/secondo*;
- 2- **densità di memorizzazione** del supporto di memorizzazione misurata il *bit/pollice*;
- 3- **velocità di trasferimento** del supporto di memorizzazione misurata il *bit/secondo* data dal prodotto dei due parametri precedenti.

I principali supporti magnetici di memorizzazione sono i **NASTRI** ed i **DISCHI**.

Il **DISCO** è un supporto di memorizzazione di massa costituito da un piatto accuratamente levigato le cui facce sono ricoperte da sostanza magnetica. L'unità di memorizzazione invece è composta da un dispositivo dotato di testine all'interno del quale il disco ruota velocemente attorno al proprio asse. Tale dispositivo è chiamato *drive* e comunica con la CPU attraverso un proprio controller.

Le areole si trovano su apposite piste circolari concentriche dette **tracce** su cui i dati vengono memorizzati sequenzialmente. Ogni traccia è suddivisa in un numero fisso di **settori**, separati da zone neutre dette **gap**.

**N.B. Procedendo dall'esterno verso l'interno i settori diventano fisicamente sempre più piccoli pur mantenendo inalterata la quantità di informazioni memorizzabili modificandosi esclusivamente la densità di memorizzazione.**

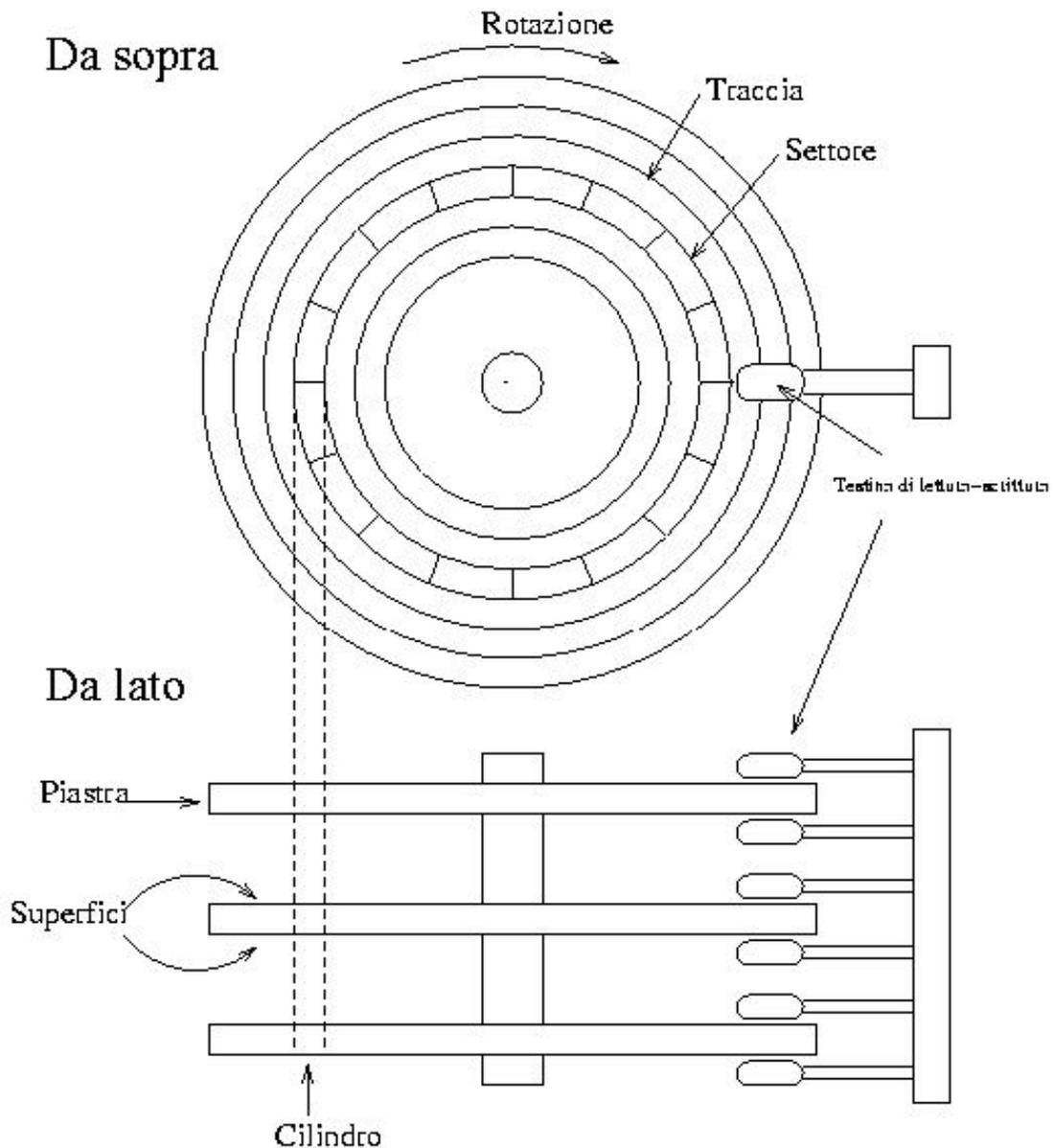
I dischi magnetici si classificano in:

- a1) Hard Disk;
- a2) Floppy Disk

**A1) L'Hard Disk o HD** è costituito da una pila di dischi di alluminio (disk-pack) disposti uno sopra l'altro ad un'opportuna di distanza che ruotano attorno ad uno stesso asse verticale con velocità uniforme (dell'ordine di 7200 giri al minuto).

Sono rinchiusi all'interno di un contenitore ermetico nel quale è presente un braccio a forma di pettine sulle cui estremità si trovano le testine di lettura/scrittura.

## Schema di un hard disk.



Se ogni braccio contiene **una** testina di lettura/scrittura **per** ciascuna faccia del disco parleremo di **HD a testine mobili**;

Se ogni braccio contiene **tante** testine di lettura/scrittura **quante** sono le tracce di ciascuna faccia del disco parleremo di **HD a testine fisse**;

All'interno di un disk-pack le informazioni possono essere memorizzate:

- **per tracce:** le informazioni seguono l'ordine delle tracce della singola faccia del disco. Solo quando le tracce di una faccia sono tutte occupate si passa all'altra;
- **per cilindri:** un cilindro è costituito da tutte le tracce di un disk-pack equidistanti dall'asse. La memorizzazione per cilindri consiste nel memorizzare dati su tutte le tracce dei dischi del disk-pack che hanno lo stesso numero d'ordine.

Il tempo necessario per poter accedere ad un settore *ossia l'unità di memoria minima che può essere letta o scritta* viene chiamato **tempo di accesso**, si misura in millisecondi (msec) ed è dato dalla somma di:ù

- a) **tempo di posizionamento o tempo di seek** ossia il tempo necessario affinché la testina si posizioni sulla traccia contenente il settore interessato. Esso dipende anche dall'ampiezza dello spostamento (aumenta man mano che ci avviciniamo alle tracce più interne) ed è nullo per i disk-pack a testine fisse;
- b) **tempo di latenza** ossia il tempo di attesa necessario affinché il settore interessato passi sotto la testina di lettura/scrittura. Esso dipende dalla velocità di rotazione dei dischi.

I dischi magnetici sono supporti di memorizzazione di massa detti **ad accesso diretto o casuale** in quanto è possibile posizionarsi direttamente su di una singola informazione prescindendo dalla sua posizione fisica. E' sufficiente conoscere **cilindro, la traccia ed il settore**. In realtà in questo tipo di supporto il tempo di accesso ad una informazione è quasi indipendente dalla posizione in cui si trova (**memoria a d accesso semicasuale**) in quanto è possibile accedere direttamente alla traccia di interesse, attendendo di accedere al settore dove è contenuta l'informazione in modo sequenziale.

Per migliorare le prestazioni del sistema (per accelerare l'accesso ai dati) la CPU sfrutta la cosiddetta **cache di disco** ossia un area dell'hard disk all'interno della quale vengono memorizzati i dati che essa sta per utilizzare. Quando viene letto un settore, i dati contenuti nei settori adiacenti vengono trasferiti nella cache del disco evitando di dover rimettere in funzione il disco fisso per reperire dati che dovrebbero essere reperiti in un secondo momento.

Il disco fisso viene utilizzato dalla CPU anche per simulare la RAM ossia la CPU sfrutta parte del disco fisso come memoria centrale quando devono essere tenuti aperti più programmi contemporaneamente o quando si manipolano grosse quantità di dati che la RAM fa sola non riesce a gestire. Quest'area del disco fisso prende il nome di **memoria virtuale**.

**A2) Il Floppy Disk o floppy** è costituito da un singolo piatto di materiale plastico dove le informazioni sono generalmente organizzate su 40 o 80 tracce per faccia suddivise in 9 settori. Sono molto economici, mobili, ma hanno una capacità di gran lunga inferiore agli hard disk. I discendenti più evoluti dei floppy possono essere considerati le **cartucce ZIP** (da 100 Mb ed oltre) e **JAZ** (da 1 GB) oppure le **pen drive** prodotte dalla IOMEGA o da altri costruttori hw.

**b) Le memorie ottiche:** sono dispositivi fisici che utilizzano le tecnologie ottiche per memorizzare le informazioni. Tali dispositivi detti in genere dischi ottici sono costituiti da un piatto di materiale plastico sui quali sia la lettura sia la scrittura avviene utilizzando un raggio laser. Attualmente esistono 4 tipi fondamentali di CD:

b1) **CD-ROM (Compact Disk - Read Only Memory):** sono dischi ottici a sola lettura e come tali non riscrivibili. Possono contenere grandi quantità di informazioni fino a 2 Gb (di solito 700 Mb) e vengono utilizzati per contenere materiale multimediale in genere;

b2) **CD-WORM** ossia **CD-R (Compact Disk – Write Once Read Many):** di capienza analoga ai precedenti sono dischi ottici a sui quali è possibile effettuare l'operazione di scrittura nelle parti vuote una volta sola utilizzando un particolare dispositivo detto **masterizzatore** e dai quali è possibile leggere quante volte si vuole. Possono contenere grandi quantità di informazioni e spesso vengono utilizzate per conservare copie di back-up delle informazioni importanti.

b3) **CD-WMRA** ossia **CD-RW (Compact Disk – Write Many Read Always)**

b4) **DVD (Digital Versatile Disk)** sono utilizzati per le correnti applicazioni multimediali (film etc.) ed hanno una capacità oscillante tra i 4,7 ed i 17 Gb.

**N.B.**

**Dal punto di vista l'organizzazione delle informazioni essi differiscono dai supporti magnetici in quanto la memorizzazione non avviene per tracce concentriche bensì su di un'unica traccia a forma di spirale come in un vecchio LP.**

## LA CLASSIFICAZIONE DEGLI ELABORATORI

Gli elaboratori vengono classificati in **quattro categorie** fondamentali a seconda delle funzioni, delle prestazioni, delle dimensioni, del costo e della velocità di elaborazione.

**1) Supercomputer:** i supercomputer sono i più potenti, costosi e veloci computer disponibili attualmente (fino a *1000 miliardi di operazioni al secondo*). Vengono utilizzati in genere nelle università e nei centri di ricerca per valutare e provare teorie scientifiche e per risolvere problemi che necessitano di un numero enorme di calcoli;

**2) Mainframe:** sono elaboratori con potenti processori e grandi quantità di RAM in grado di gestire enormi quantità di dati grazie anche a capienti memorie di massa. Sono particolarmente usati in *multiutenza* ossia permettono a più persone di accedere alle sue risorse attraverso appositi terminali (tastiera + monitor) condividendo così gli stessi dati. Sono molto costosi e per questo vengono utilizzati da grandi organizzazioni (Banche, assicurazioni, ospedali, etc.);

**3) Minicomputer:** sono elaboratori più piccoli dei mainframe ma anch'essi in grado di gestire grosse moli di dati in multiutenza. Sono utilizzati da organizzazioni di medio-piccole dimensioni;

**4) Microcomputer o Personal Computer o PC:** sono gli elaboratori in genere usati in casa o in ufficio. In genere con questo termine intendiamo macchine usate in ambito lavorativo o domestico usate da un solo utente per volta.