

# **Informatica**

**Prof.Ing. Francesco Motta**

# L'INFORMATICA

- **Esistono varie definizioni:**
- “Scienza dei calcolatori elettronici (*Computer Science*)”
- “Scienza dell’informazione”
- **Definizione proposta**
- “Scienza della rappresentazione e dell’elaborazione dell’informazione.”

# L'informatica comprende

---

- metodi per la rappresentazione delle informazioni
- metodi per la rappresentazione di procedimenti risolutivi
- linguaggi di programmazione
- architettura dei calcolatori
- sistemi operativi
- calcolo numerico
- etc.

# ELABORATORE ELETTRONICO (COMPUTER)

---

- **Strumento** per la rappresentazione e l'elaborazione delle informazioni



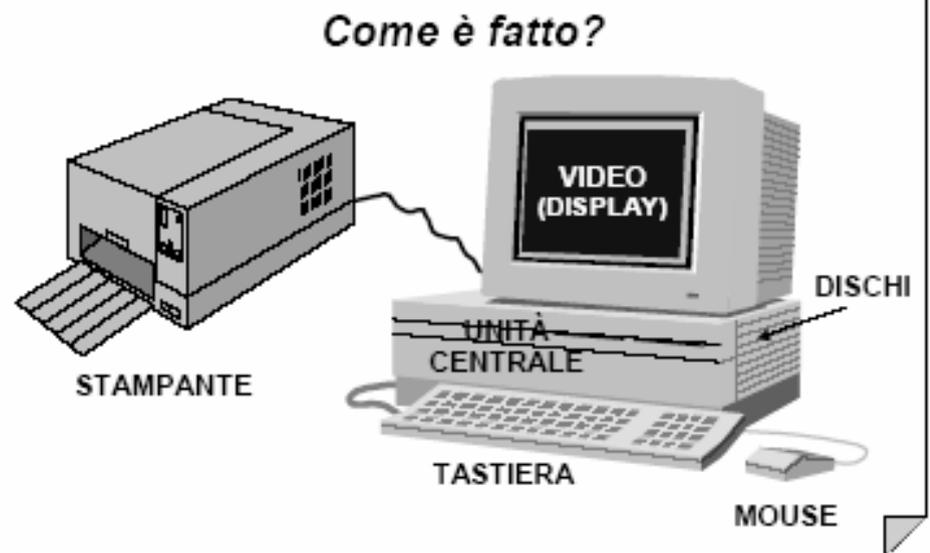
## Componenti principali

- Unità centrale
- Video (“monitor”)
- Tastiera e Mouse
- Lettore CD
- Dischi fissi (“hard disk”)
- Dischetti (“floppy”)

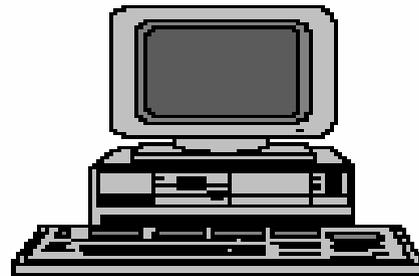
## Componenti accessori

- Stampante
- Modem
- Scanner

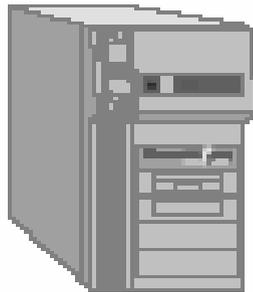
## HARDWARE



*Sono tutti uguali?*



"Desktop"



"Tower"

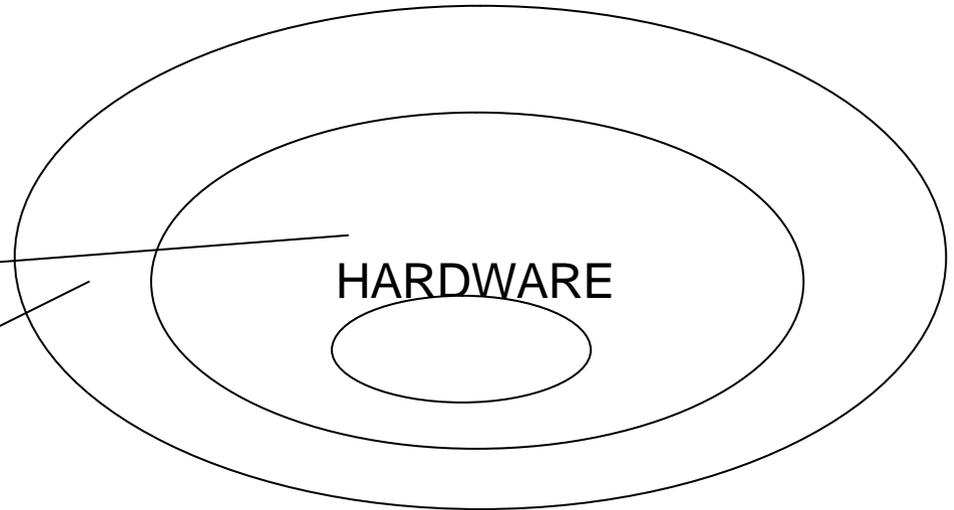


Portatile  
(Notebook)

# SOFTWARE

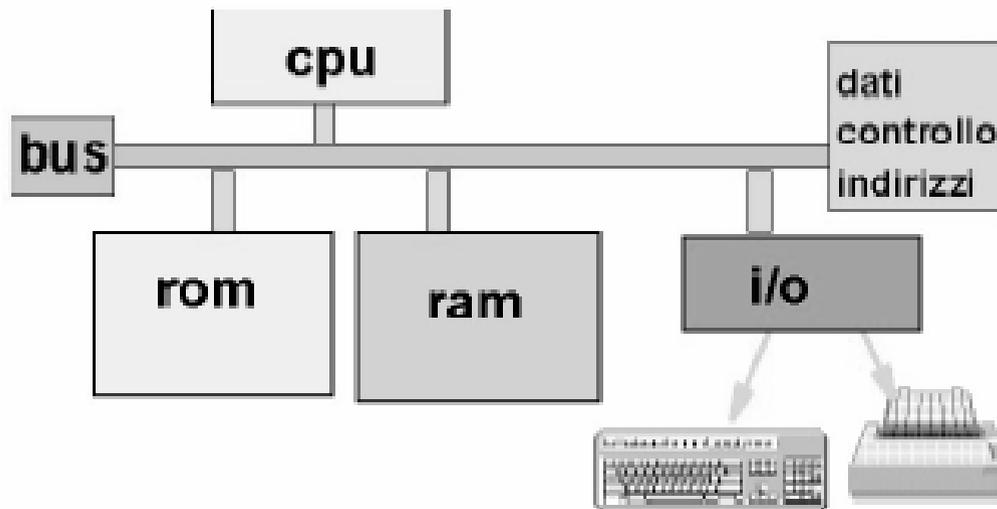
---

- **Software: programmi che**
- vengono eseguiti dal sistema.
- **Distinzione fra:**
- *Software di base* (es. Sistema Operativo)
- *Software applicativo*



# ARCHITETTURA DI UN ELABORATORE

---



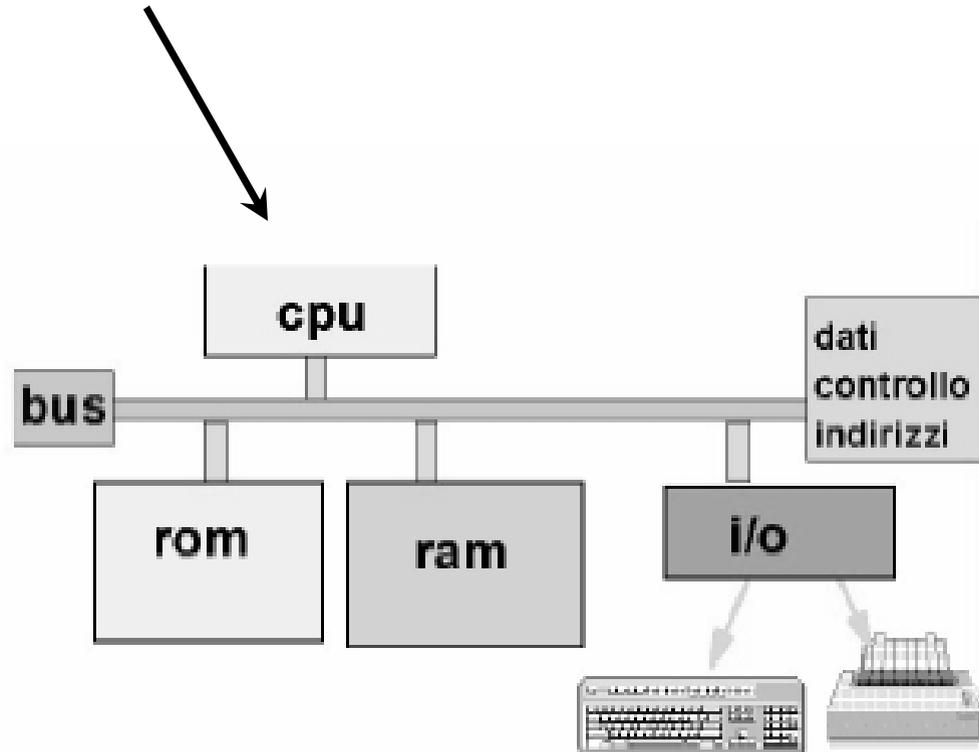
Ispirata al modello della **Macchina di Von Neumann** (Princeton, Institute for Advanced Study, anni '40).

# HARDWARE

---

- **CPU (Central Processing Unit), o Processore**

**CPU:** Svolge le elaborazioni e il trasferimento dei dati, cioè *esegue i programmi*



# CPU & MEMORIA

---



- ALU (Arithmetic & Logic Unit)
- Unità di Controllo
- Registri

# UNITÀ DI ELABORAZIONE (CPU)

---

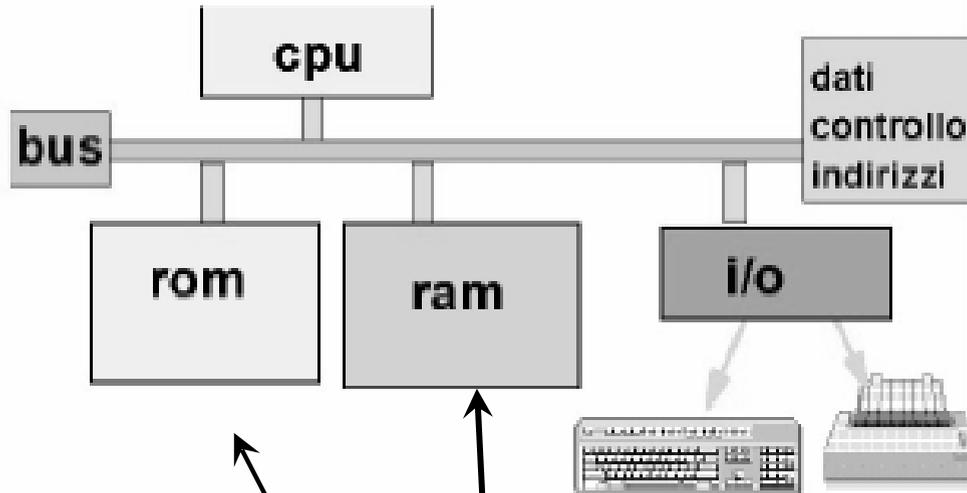
## **ALU (Arithmetic / Logic Unit)**

Esegue le operazioni aritmetiche e logiche elementari

**Unità di Controllo** (*Control Unit*): controlla e coordina l'attività della CPU. (In particolare, controlla il trasferimento dei dati tra memoria e registri e la decodifica e l'esecuzione delle istruzioni)

I **registri** sono *locazioni* usate per memorizzare dati, istruzioni, o indirizzi ***all'interno della CPU***. L'accesso ai registri è *molto veloce*.

# HARDWARE



## RAM & ROM

- Dimensioni relativamente limitate
- Accesso molto rapido

**RAM (*Random Access Memory*), e  
ROM (*Read Only Memory*)**

**Insieme formano la *Memoria centrale***

# HARDWARE

---

## ATTENZIONE

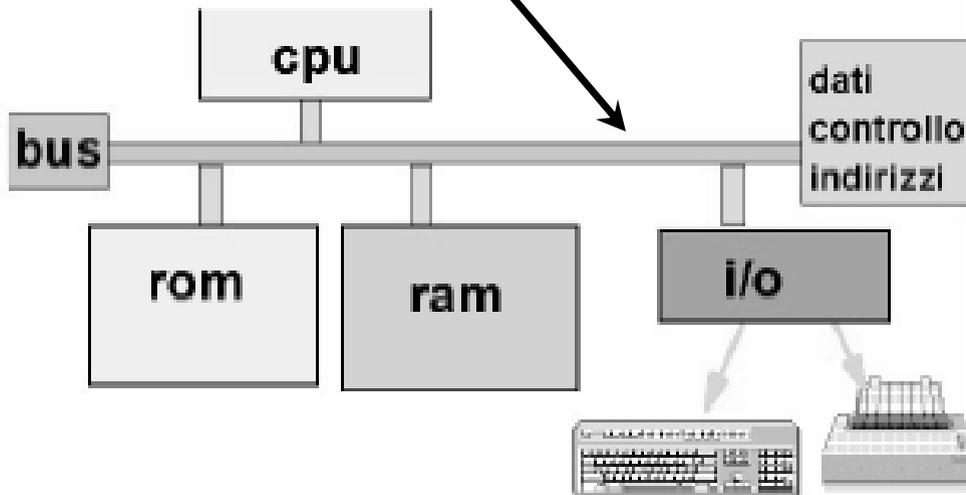
- **RAM è volatile** (perde il suo contenuto quando si spegne il calcolatore)
- usata per memorizzare dati e programmi
- **ROM è persistente** (mantiene il suo contenuto quando si spegne il calcolatore) ma il suo ***contenuto è fisso e immutabile***
- usata per memorizzare programmi di sistema

# HARDWARE

---

## BUS DI SISTEMA

È una “linea di comunicazione”  
che collega tutti gli elementi  
funzionali.



# UNITÀ DI ELABORAZIONE (CPU)

---

## **ALU (Arithmetic / Logic Unit)**

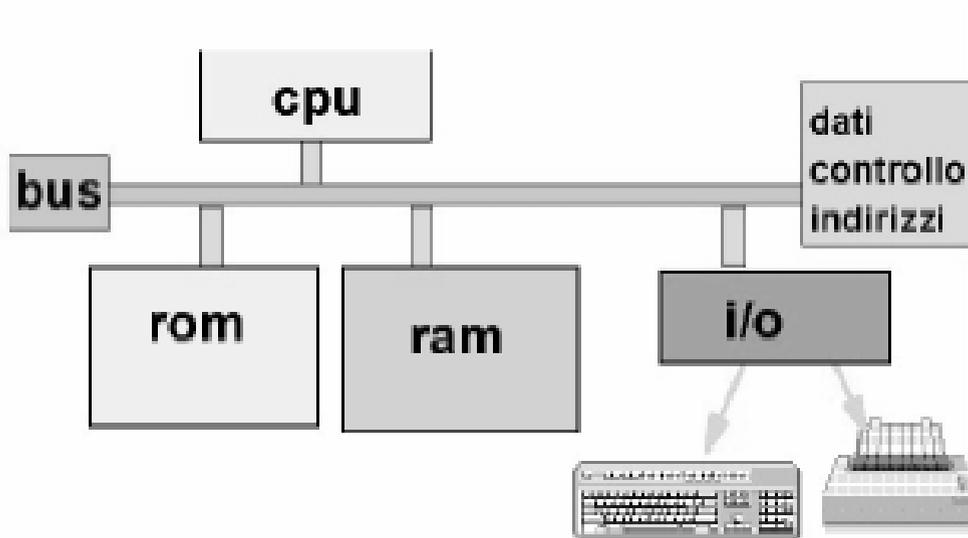
Esegue le operazioni aritmetiche e logiche elementari

**Unità di Controllo** (*Control Unit*): controlla e coordina l'attività della CPU. (In particolare, controlla il trasferimento dei dati tra memoria e registri e la decodifica e l'esecuzione delle istruzioni)

I registri sono *locazioni* usate per memorizzare dati, istruzioni, o indirizzi *all'interno della CPU*. L'accesso ai registri è *molto veloce*.

# HARDWARE

---



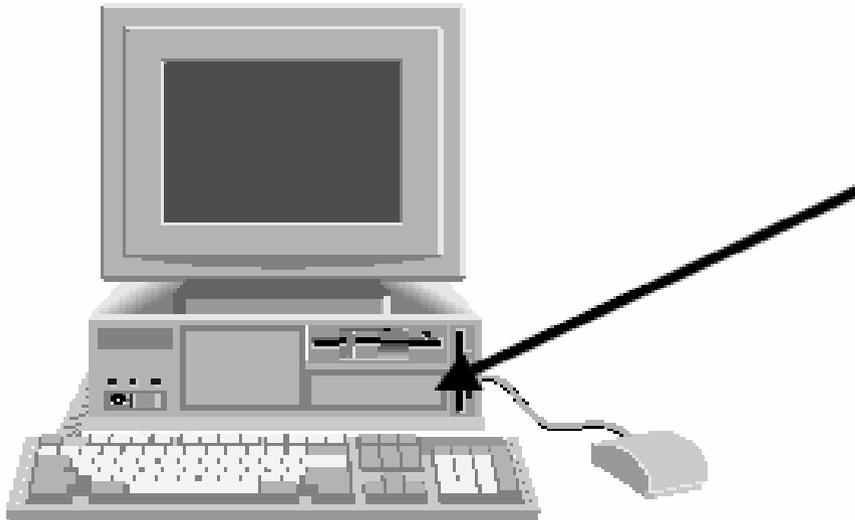
## **UNITÀ DI INGRESSO / USCITA (I/O)**

- Tastiera e Mouse
- Video e Stampante
- Scanner
- Modem
- **Dispositivi di memoria Massa**
- **Ecc...**

**Sono usate per far comunicare il calcolatore con l'esterno (in particolare con l'utente)**

# HARDWARE

---



## MEMORIA DI MASSA

- Dischi
- CD
- Nastri
- ...

- memorizza **grandi quantità** di informazioni
- **persistente** (le informazioni non si perdono spegnendo la macchina)
- **accesso molto meno rapido** della memoria centrale (millisecondi contro nanosecondi / differenza  $10^6$ )

# TECNOLOGIA DIGITALE

---

CPU, memoria centrale e dispositivi sono realizzati con **tecnologia elettronica digitale**.

Dati ed operazioni vengono codificati a partire da due valori distinti di grandezze elettriche:

- tensione alta ( $V_H$ , 5V)
- tensione bassa ( $V_L$ , 0V)

A tali valori vengono convenzionalmente associate le due cifre binarie **0 e 1**:

- logica positiva:  $1 \leftrightarrow V_H$ ,  $0 \leftrightarrow V_L$
- logica negativa:  $0 \leftrightarrow V_H$ ,  $1 \leftrightarrow V_L$

# TECNOLOGIA DIGITALE

---

Dati ed operazioni vengono codificati tramite sequenze di cifre binarie 0 e 1 (bit da Binary digit)

01000110101 ....

• Per memorizzare informazioni più complesse si considerano *collezioni di bit*:

**BYTE** (collezione di 8 bit) e suoi multipli:

- Kbyte (1.024 Byte)
- Mbyte (1.048.576 Byte)
- Gbyte (1.073.741.824 Byte)

# TECNOLOGIA DIGITALE

---

- La CPU è in grado di operare soltanto in aritmetica binaria, effettuando operazioni *elementari* :
  - somma e differenza
  - scorrimento (shift)
  - ...

Lavorando direttamente sull'hardware, l'utente è forzato a esprimere i propri comandi al livello della macchina, tramite *sequenze di bit*.

# LA MEMORIA DI MASSA

---

**Scopo:** memorizzare *grandi masse* di dati in modo *persistente*

(I dati memorizzati su questo tipo di memoria sopravvivono all'esecuzione dei programmi)

**Caratteristiche:**

- *tempo di accesso*
- *capacità*

Tempo di accesso

- disco fisso: ~10 ms
- floppy: ~100 ms

Capacità

- disco fisso: >10 GB
- floppy: 1.4 MB

# DISPOSITIVI di memoria di massa

---

## DUE CLASSI FONDAMENTALI:

- ad *accesso sequenziale* (ad esempio, **NASTRI**):  
per recuperare un dato è necessario accedere  
prima a tutti quelli che lo precedono sul dispositivo
- ad *accesso diretto* (**DISCHI**):  
si può recuperare direttamente un qualunque dato  
memorizzato

# DISPOSITIVI MAGNETICI

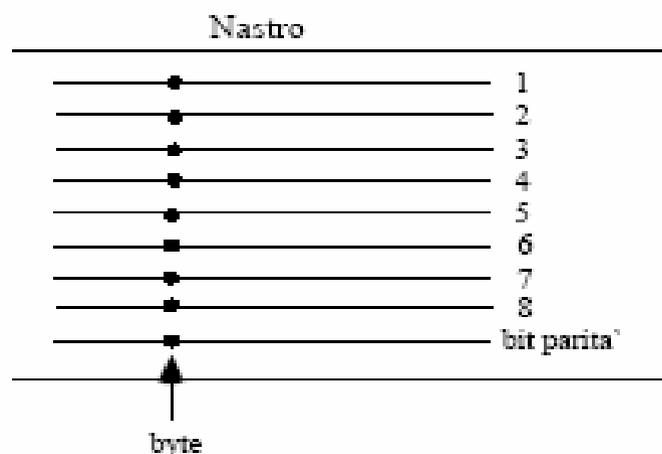
---

- L'area del dispositivo è suddivisa in **micro-zone**
- Ogni micro-zona memorizza una **informazione elementare** sotto forma di ***stato di magnetizzazione***:
  - area magnetizzata / area non magnetizzata
- Ai due possibili stati di magnetizzazione vengono **associate le due cifre binarie 0 e 1**
- Quindi, **ogni micro-zona memorizza 1 bit**

# NASTRI MAGNETICI

Nastri di materiale magnetizzabile arrotolati su supporti circolari, o in cassette.

Sul nastro sono tracciate delle **piste orizzontali parallele** (di solito 9, di cui 8 corrispondono ad un **byte** e la nona è il bit di **parità**).

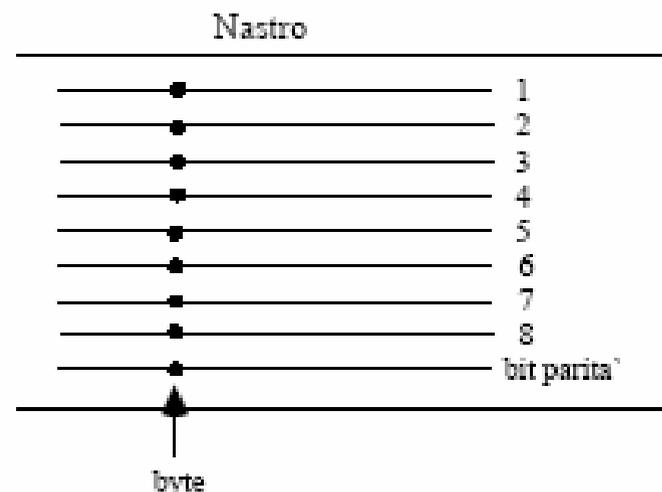


*Bit di parità': rende pari il numero di 1 contenuti nelle piste orizzontali. Serve per il controllo di eventuali errori di memorizzazione.*

# NASTRI MAGNETICI (segue)

---

I dati sul nastro sono organizzati in zone contigue dette *record*, separate da zone prive di informazione (*inter-record gap*).



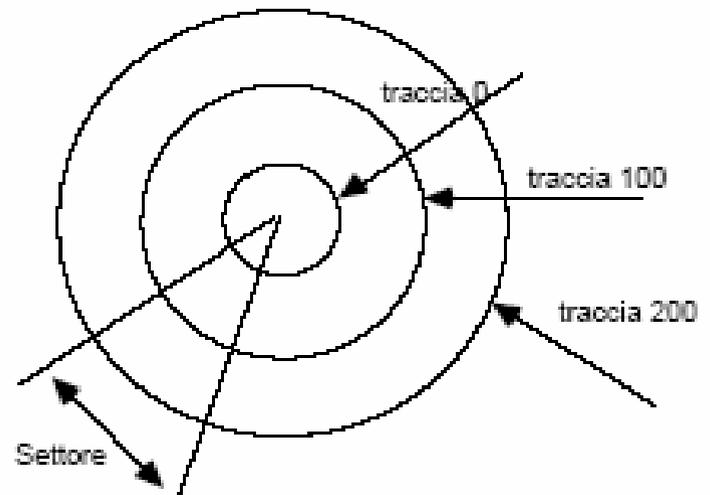
- Tutte le elaborazioni sono **sequenziali**:  
le operazioni su uno specifico record sono *lente*
- Oggi servono solo per mantenere copie di riserva (*backup*) dei dati

# DISCHI MAGNETICI

---

Un disco consiste in un certo numero di *piatti* con **due superfici** che ruotano attorno ad un perno centrale.

Ogni superficie dispone di una propria **testina di lettura / scrittura**.



Le superfici sono organizzate in *cerchi concentrici* (tracce) e in *spicchi* di ugual grandezza (settori).  
Le tracce equidistanti dal centro formano un **cilindro**.

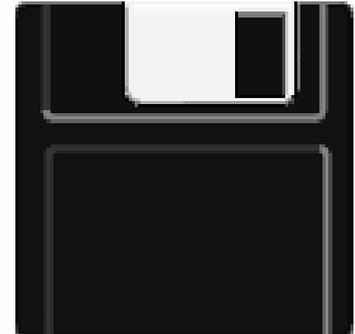
# DISCHETTI (FLOPPY)

---

Sono dischi magnetici di *piccola capacità*, portatili, usati per trasferire informazioni tra computer diversi.

Sono costituiti da un **unico disco** con due superfici.

Sopravvivono solo quelli da 3.5" di diametro (1.4 MB)



**IMPORTANTE:** per poter essere usati, i dischetti devono prima essere *suddivisi in tracce e settori* dal Sistema Operativo → **FORMATTAZIONE**

Operazione lunga, noiosa → **dischetti già formattati**

# DISPOSITIVI OTTICI

---

## 1984, CD-ROM (Compact-Disk Read-Only Memory)

- Capacità: > 600 MB
- Costo: < \$1
- Velocità di trasferimento:
  - originariamente 150 KB / s ( "1X" )
  - oggi 24, 32, 40 volte tanto...
- Parenti stretti dei CD audio (CD-DA, 1982)
- Accesso diretto ai settori (capacità 2.048 KB)

## 1984, WORM (Write Once Read Many)

- Sono dischi ottici scrivibili (una sola volta)

# DISPOSITIVI OTTICI - Il presente

---

## **1997, DVD (Digital Video Disk)**

- Evoluzione del CD-ROM
- Capacità fino a 17 GB
- Velocità di trasferimento molto elevata

Adatto per film e opere pesantemente multimediali.

# CAPACITÀ DELLE MEMORIE

---

<b>Tipo di memoria</b>	<b>Capacità</b>
Memoria centrale	16-512 Mbyte
Dischi magnetici	5 GByte -50 Gbyte
Dischi floppy	1.4 Mbyte
Nastri (bobina)	20-400 Mbyte
Nastri (cassetta)	200-5000 Mbyte
Dischi ottici	650 Mbyte - 17 GByte

# PERSONAL COMPUTER

---

## PC (ex "IBM-COMPATIBILI")

Usano processori della famiglia *Intel 80x86*:

- 8086
- 80286
- ...
- Pentium
- Pentium MMX
- Pentium II
- Pentium III
- Pentium IV
- ...



**Le prestazioni dipendono da:**

- frequenza dell'orologio di sistema (*clock*)
- dimensione della RAM
- velocità/parallelismo delle linee dati/comandi (bus)

# ALTRI SISTEMI DI CALCOLO

---

## **Workstation**

sistemi con capacità di supportare più attività contemporanee, spesso dedicati a più utenti. Prestazioni normalmente superiori a quello di un tipico Personal Computer.

## **Mini-calcolatori**

Macchine capaci di servire decine di utenti contemporaneamente, collegati tramite terminali

## **Super-calcolatori**

Hanno molti processori, grandi memorie di massa e servono tipicamente centinaia o migliaia di terminali

# IL SOFTWARE

---

## Software:

insieme (complesso) di programmi.

**Organizzazione a strati**, ciascuno con funzionalità di livello più alto rispetto a quelli sottostanti

Concetto di  
***MACCHINA VIRTUALE***



# IL FIRMWARE

---

**Firmware:**

il confine fra hardware e software.

**È uno strato di *micro-programmi*, scritti dai costruttori, che agiscono direttamente al di sopra dello strato hardware**

Sono memorizzati su una speciale *memoria centrale permanente* (ROM, EPROM, ...)

# IL SISTEMA OPERATIVO

---

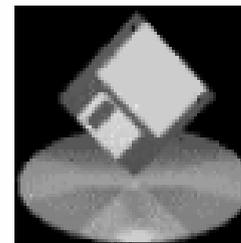
Strato di programmi che opera *al di sopra di hardware e firmware* e **gestisce l'elaboratore**.

Solitamente, è venduto insieme all'elaboratore.

**Spesso si può scegliere tra *diversi sistemi operativi*** per lo stesso elaboratore, con diverse caratteristiche.

## Esempi:

- Windows 95 / 98 / ME
- Windows NT / 2000
- Linux
- ...



# **FUNZIONI DEL SISTEMA OPERATIVO**

---

Le funzioni messe a disposizione dal S.O. dipendono dalla complessità del sistema di elaborazione:

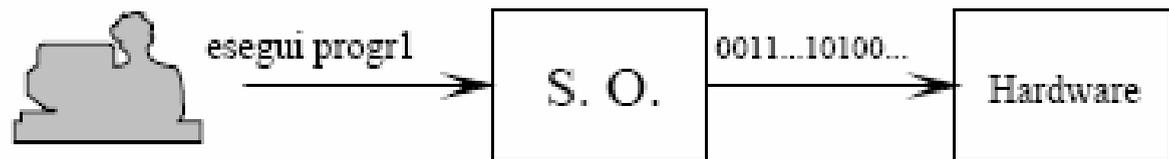
- gestione delle risorse disponibili
  - gestione della memoria centrale
  - organizzazione e gestione della memoria di massa
  - gestione di un sistema multi-utente
- interpretazione ed esecuzione di comandi elementari

**Un utente “vede” l’elaboratore solo tramite il Sistema Operativo**

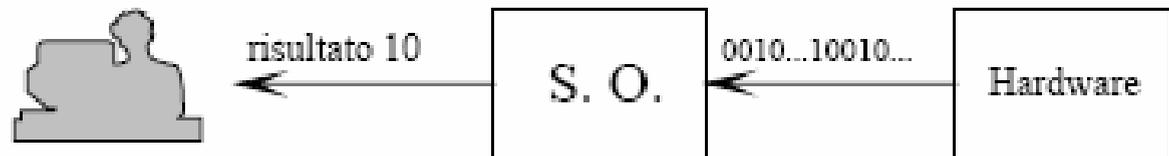
**→ il S.O. realizza una “macchina virtuale”**

# RUOLO DEL SISTEMA OPERATIVO

Il S.O. traduce le richieste dell'utente in opportune sequenze di istruzioni, a loro volta trasformate in valori e impulsi elettrici per la macchina fisica.



e viceversa:



# **RUOLO DEL SISTEMA OPERATIVO**

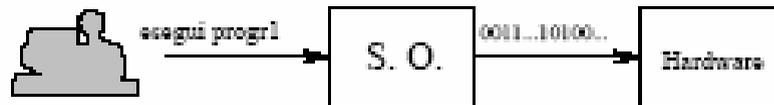
---

**Qualsiasi operazione di accesso a risorse della macchina implicitamente richiesta dal comando di utente viene esplicitata dal S.O.**

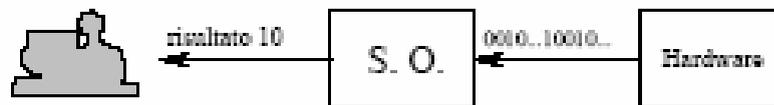
## **Esempi:**

- **accesso a memoria centrale**
- **accesso ai dischi**
- **I/O verso video, tastiera, ...**

# ESEMPIO



e viceversa:



Utente:

"esegui progr1"

Sistema Operativo:

- input da tastiera
- ricerca codice di "progr1" su disco
- carica in memoria centrale codice e dati
- <elaborazione>

Utente:

"10"

Sistema Operativo:

- output su video

# CLASSIFICAZIONE dei S.O.

---

In base al numero di utenti:

- **Mono-utente (*mono-user*):** un solo utente alla volta può utilizzare il sistema
- **Multi-utente (*multi-user*):** più utenti possono interagire contemporaneamente con la macchina.

Nel caso di più utenti contemporanei, il **Sistema Operativo** deve fornire a ciascuno l'astrazione di *un sistema "dedicato"*.

# CLASSIFICAZIONE dei S.O.

---

In base al numero di programmi in esecuzione:

- **Mono-programmato (*mono-task*):** si può eseguire *un solo programma* per volta
- **Multi-programmato (*multi-task*):** il S.O. è in grado di portare avanti contemporaneamente l'esecuzione di più programmi (pur usando una sola CPU).

Nel caso di multi-programmazione il S.O. deve gestire la suddivisione del tempo della CPU fra i vari programmi.

# CLASSIFICAZIONE dei S.O.

---

## Esempi:

- **MS-DOS:** monoutente, monoprogrammato
- **Windows95/98:** monoutente, multiprogrammato
- **OS/2:** monoutente, multiprogrammato
- **Windows NT:** multiutente, multiprogrammato
- **UNIX (linux):** multiutente, multiprogrammato

# PROGRAMMI APPLICATIVI

---

**Risolvono problemi specifici degli utenti:**

- *word processor*: elaborazione di testi (*Es. MSWord*)
- *fogli elettronici*: gestione di tabelle, calcoli e grafici (*Es. Excel*)
- *database*: gestione di archivi (*Es. Access*)
- *suite* (integrati): collezione di applicativi capaci di funzionare in modo integrato come un'applicazione unica. (*Es. Office*)

- Sono scritti in linguaggi di programmazione di alto livello
- Risentono in misura ridotta delle caratteristiche della architettura dell'ambiente sottostante (*portabilità*)

# AMBIENTI DI PROGRAMMAZIONE

---

È l'insieme dei programmi che consentono la scrittura, la verifica e l'esecuzione di nuovi programmi (*fasi di sviluppo*).

## Sviluppo di un programma

- Affinché un programma scritto in un qualsiasi linguaggio di programmazione sia comprensibile (e quindi eseguibile) da un calcolatore, occorre **tradurlo** dal linguaggio originario al linguaggio della macchina.
- Questa operazione viene normalmente svolta da speciali programmi, detti **traduttori**.

# TRADUZIONE DI UN PROGRAMMA

---

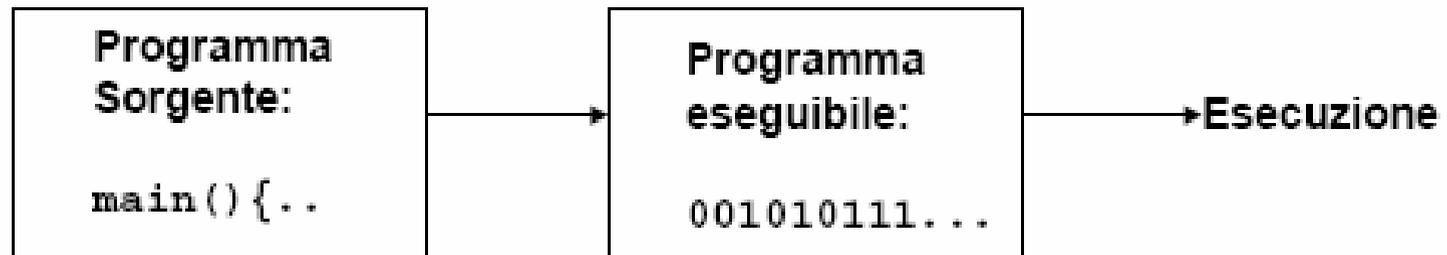
PROGRAMMA	TRADUZIONE
<code>main()</code>	
<code>{ int A;</code>	<code>00100101</code>
<code>...</code>	
<code>A=A+1;</code>	<code>11001..</code>
<code>if....</code>	<code>1011100..</code>

## Il traduttore converte

- *il testo* di un programma scritto in un particolare linguaggio di programmazione (**sorgenti**)
- nella corrispondente *rappresentazione in linguaggio macchina* (programma **eseguibile**).

# SVILUPPO DI PROGRAMMI

---

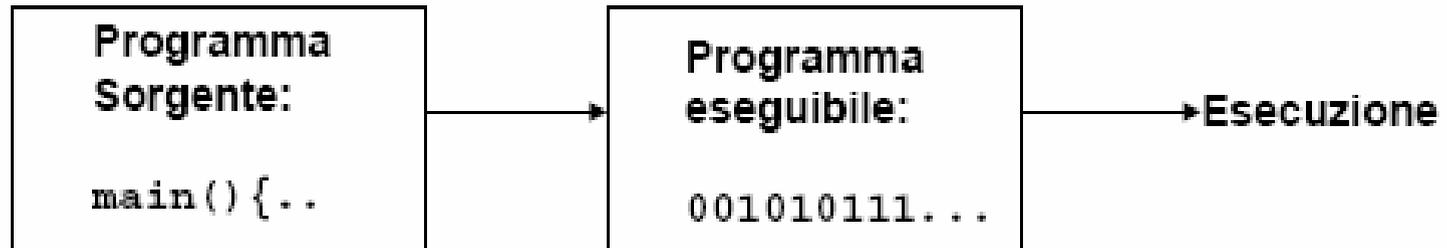


## Due categorie di traduttori:

- i *Compilatori* traducono l'intero programma (senza eseguirlo!) e producono in uscita il programma convertito in linguaggio macchina
- gli *Interpreti* traducono ed eseguono immediatamente ogni singola istruzione del *programma sorgente*.

# SVILUPPO DI PROGRAMMI (segue)

---

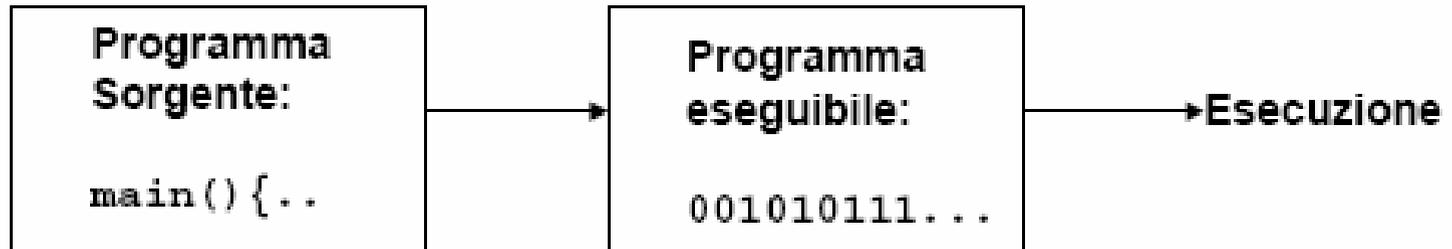


**Quindi:**

- nel caso del compilatore, lo schema precedente viene percorso *una volta sola* prima dell'esecuzione
- nel caso dell'interprete, lo schema viene invece attraversato *tante volte quante sono le istruzioni* che compongono il programma.

# SVILUPPO DI PROGRAMMI (segue)

---



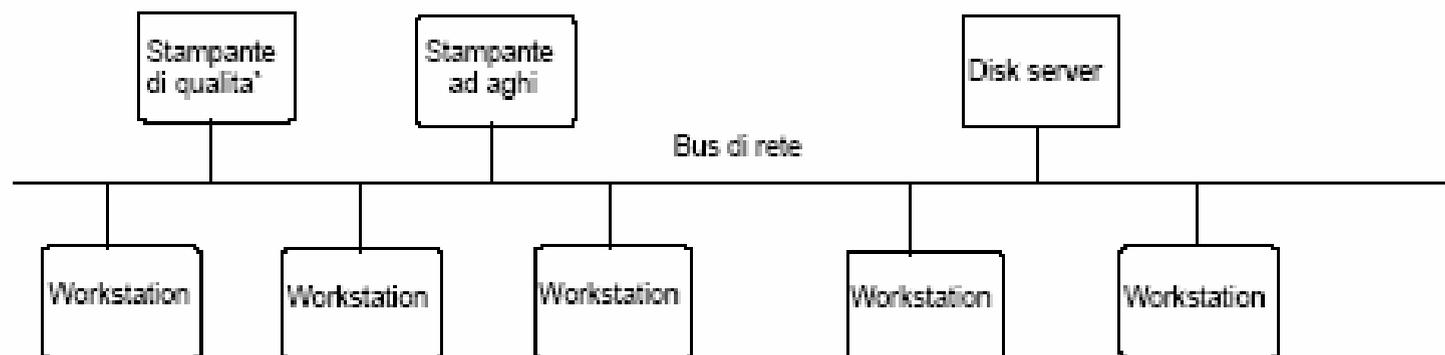
L'esecuzione di un programma *compilato* è più veloce dell'esecuzione di un programma *interpretato*

# RETI DI CALCOLATORI

---

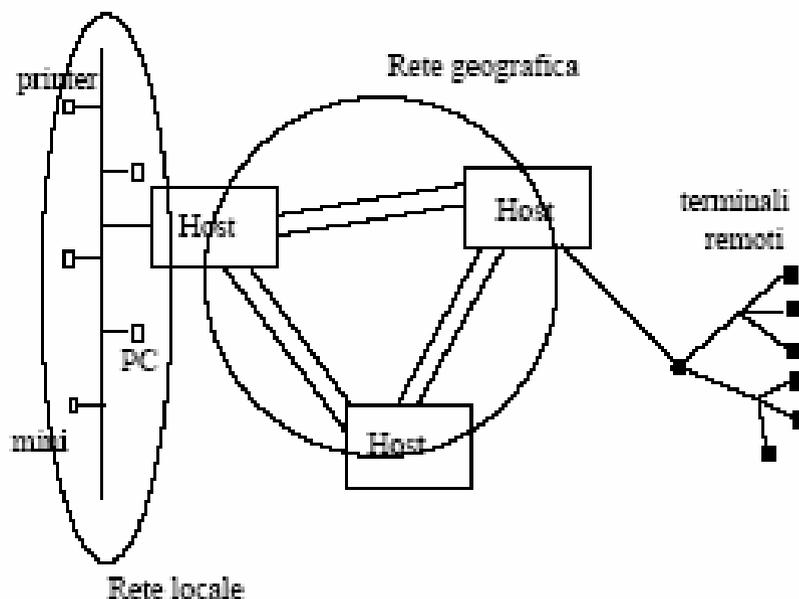
- **Reti Locali:**  
connettono elaboratori *fisicamente vicini*  
(nello stesso ufficio o stabilimento).

## LAN (Local Area Network)



# RETI DI CALCOLATORI (segue)

- **Reti geografiche:** collegano elaboratori medio-grandi situati anche a *grande distanza*.



**WAN (Wide Area Network)**

# INTERNET: la rete delle reti

---

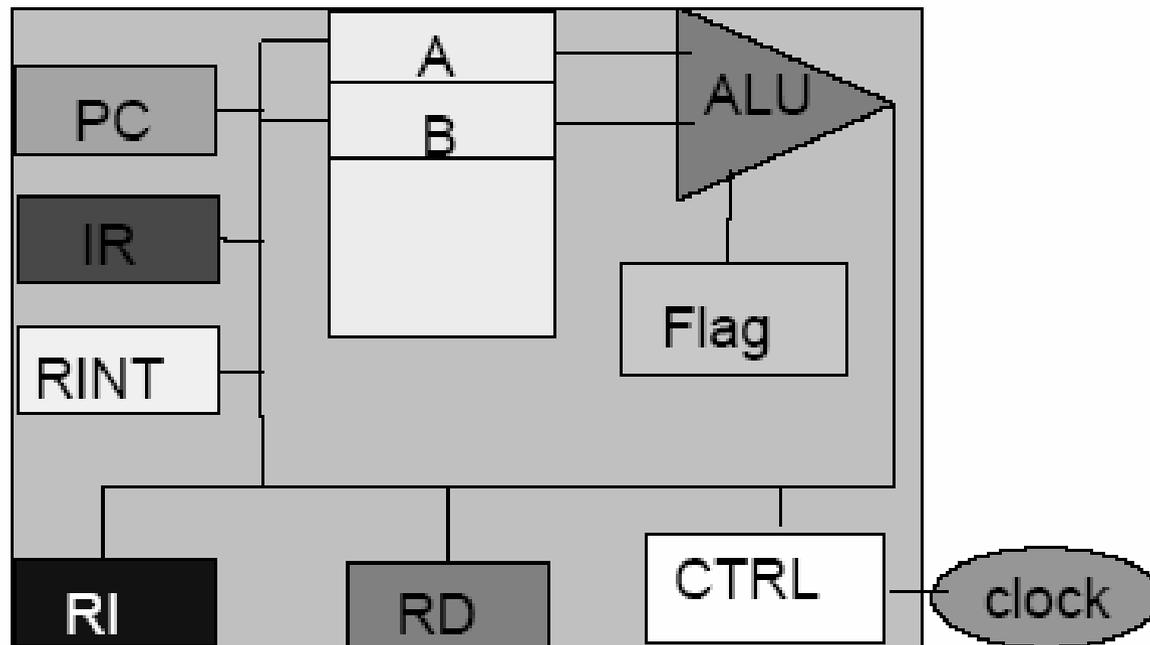
- **Internet:** la rete risultante dalla interconnessione mondiale di tutte le reti.
- Milioni di elaboratori (“**siti**”) collegati a **ragnatela**



World-Wide Web (WWW)

# UNITÀ DI ELABORAZIONE (CPU)

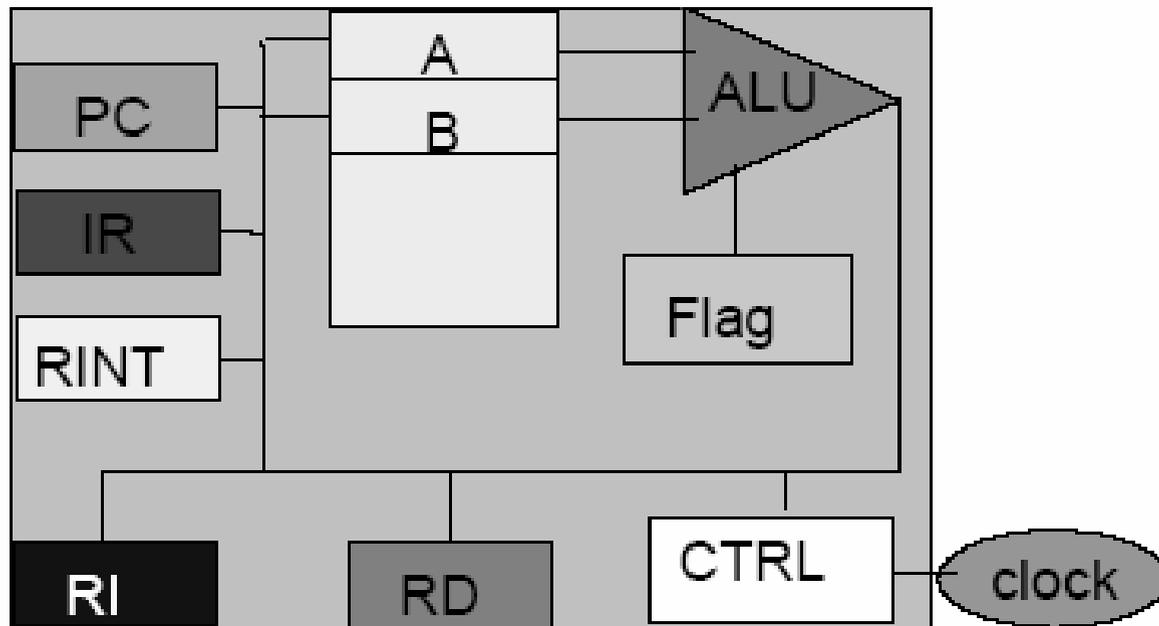
---



**Il clock dà la base dei tempi necessaria per mantenere il sincronismo fra le operazioni**

# UNITÀ DI ELABORAZIONE (CPU)

---



I registri (qui **A**, **B**, **PC**, **Flag**,...) sono *locazioni* usate per memorizzare dati, istruzioni, o indirizzi *all'interno della CPU*. L'accesso ai registri è *molto veloce*.

# UNITÀ DI ELABORAZIONE (CPU)

---



**La memoria centrale è una collezione di celle *numerate*, che possono contenere **DATI** e **ISTRUZIONI**.**

**Le istruzioni sono disposte in memoria in *celle di indirizzo crescente*.**

# UNITÀ DI ELABORAZIONE (CPU)



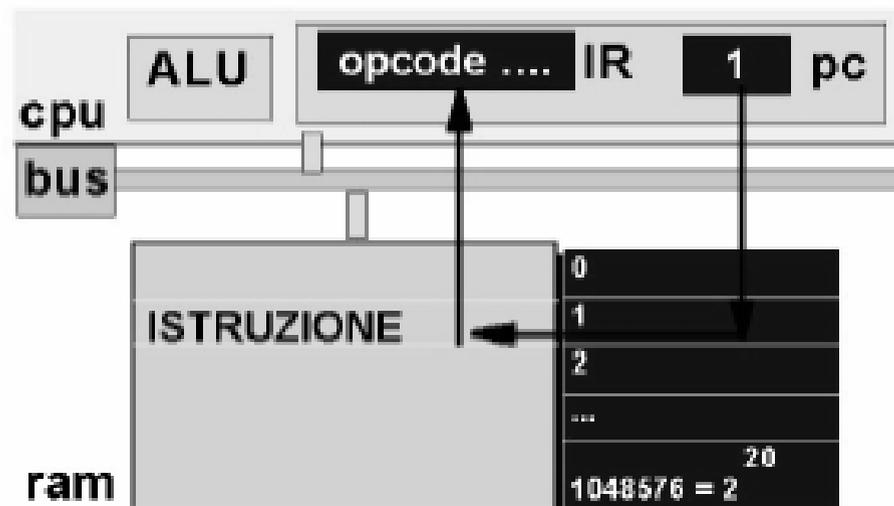
L'unità di controllo fa funzionare l'elaboratore.

Da quando viene acceso a quando è spento, essa esegue in continuazione il ciclo di *prelievo / decodifica / esecuzione* (fetch / decode / execute) .

# IL CICLO fetch / decode / execute

## FETCH

- si accede alla prossima istruzione (cella il cui indirizzo è contenuto nel registro **PC**) ...
- ... e la si porta dalla memoria centrale al *Registro Istruzioni (IR)*



# IL CICLO fetch / decode / execute

---

## DECODE

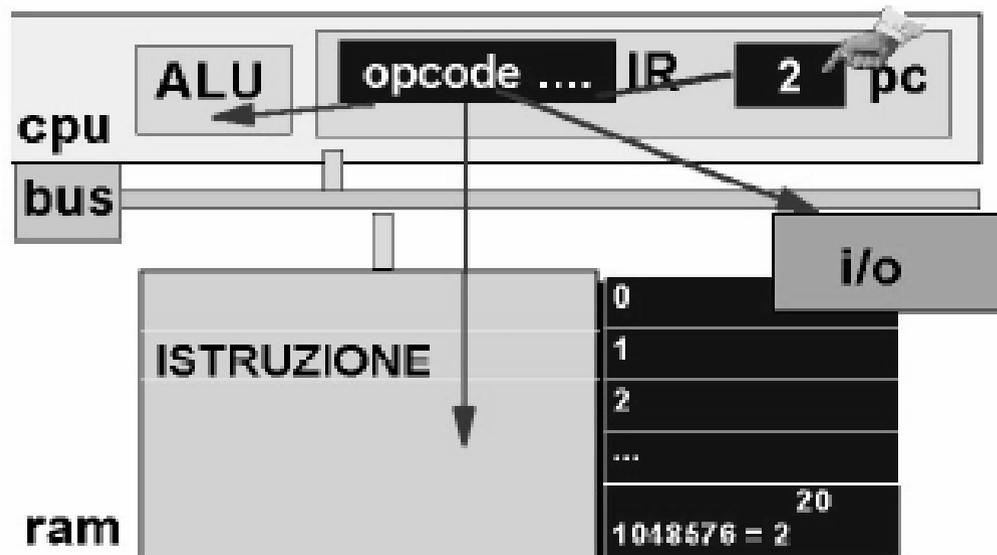
- si decodifica il tipo dell'istruzione in base al suo *OpCode* (codice operativo)



# IL CICLO fetch / decode / execute

## EXECUTE

- si individuano i dati usati dall'istruzione
- si trasferiscono tali dati nei registri opportuni
- si esegue l'istruzione.



# IL CICLO fetch / decode / execute

---

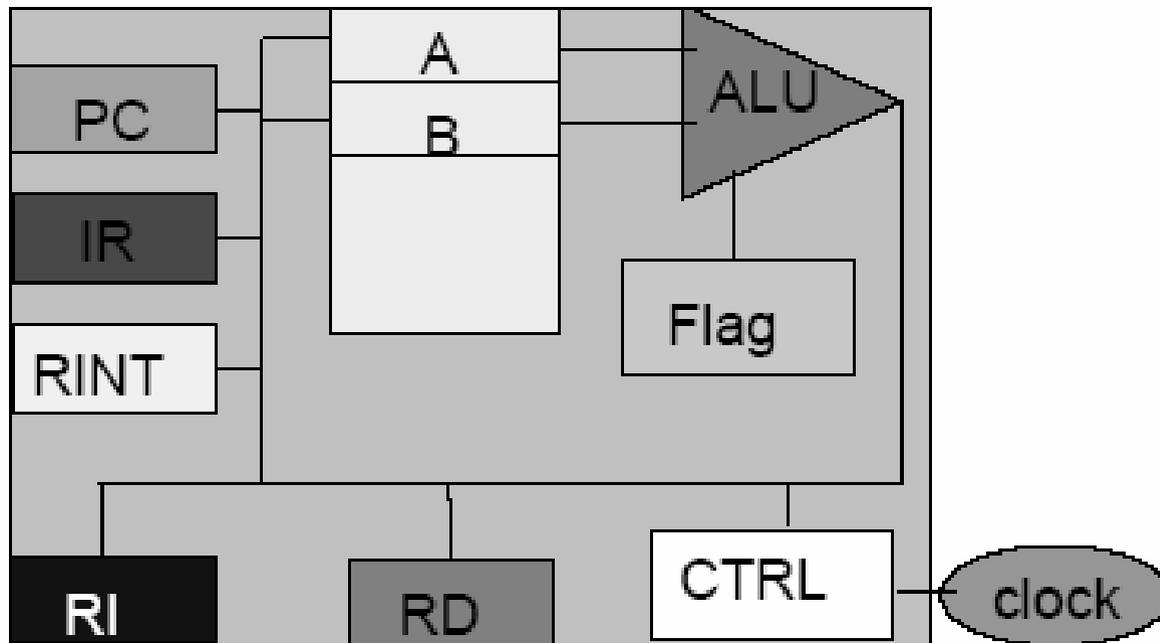
## ATTENZIONE

Istruzioni particolari possono *alterare il prelievo delle istruzioni da celle consecutive*:

- istruzioni di salto
- istruzioni di chiamata a sotto-programmi
- istruzioni di interruzione

# I REGISTRI

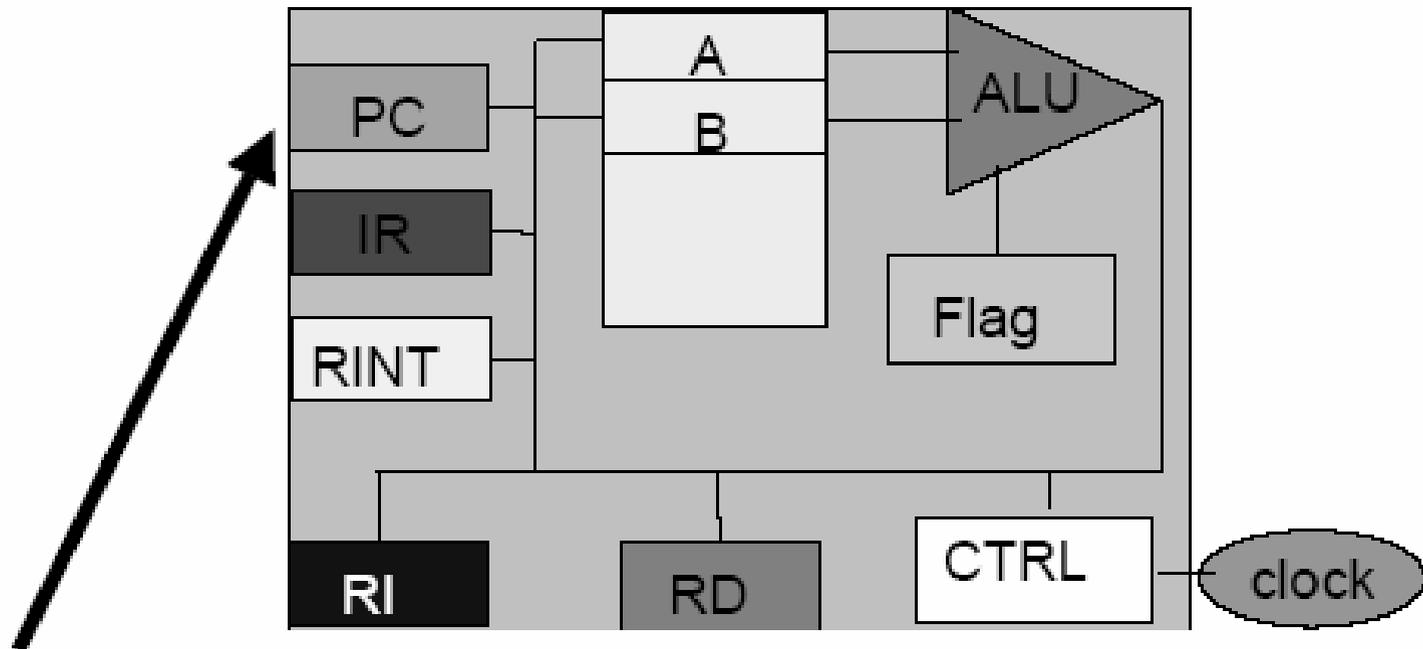
---



I registri sono *locazioni* di memoria *interne* a CPU, e come tali *molto veloci*.

# I REGISTRI

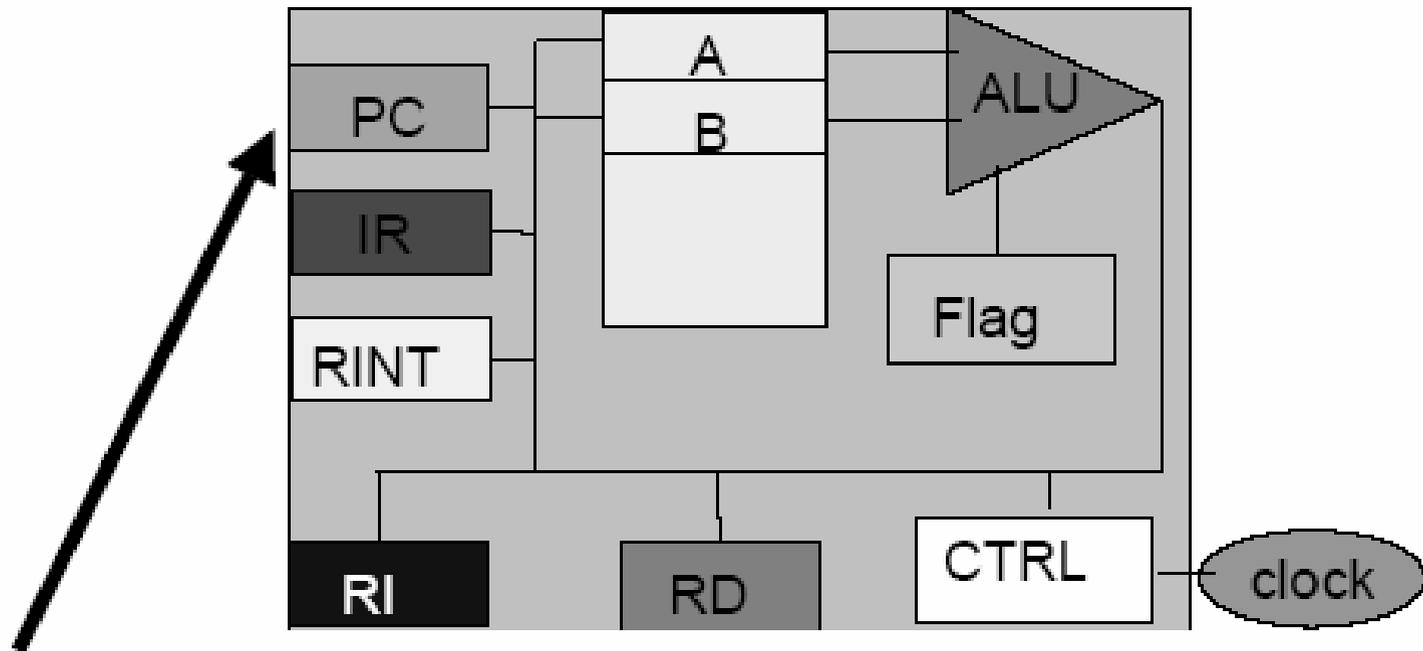
---



## **Program Counter (PC)**

Indica l'indirizzo della cella di memoria che contiene la prossima istruzione da eseguire

# I REGISTRI

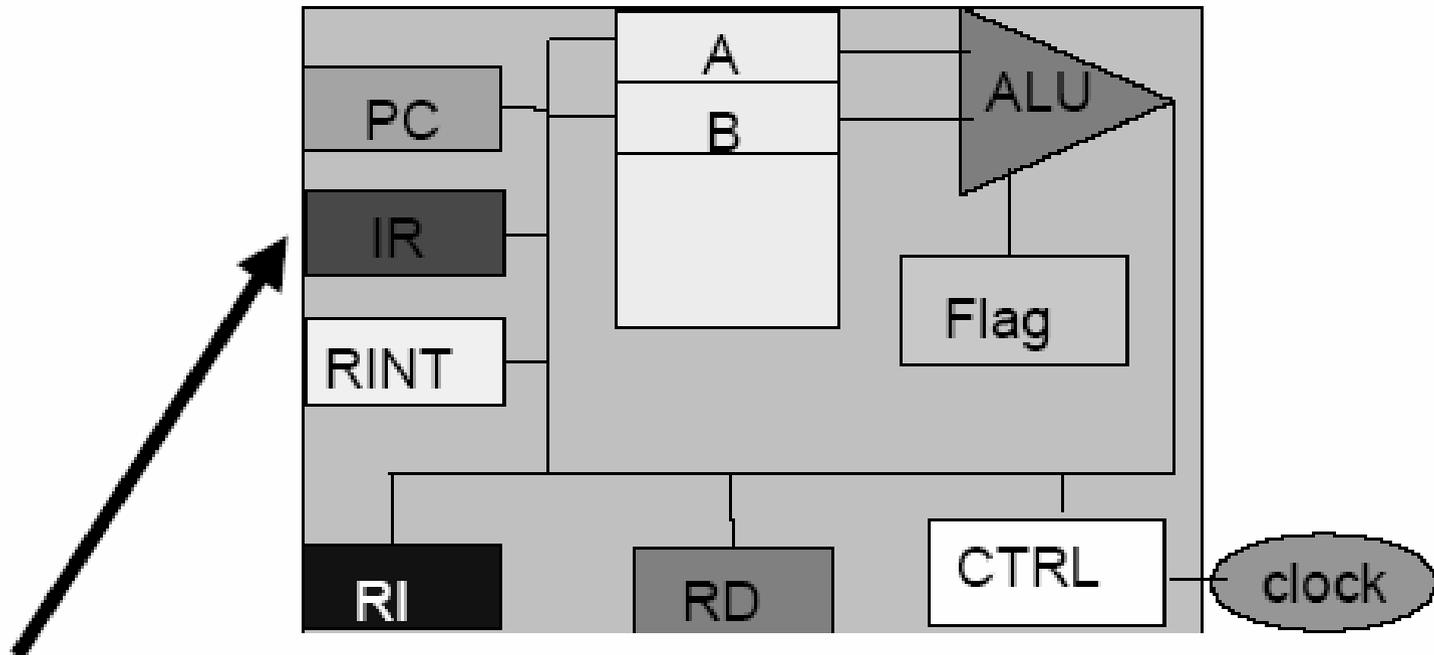


## Program Counter (PC)

Indica l'indirizzo della cella di memoria che contiene la prossima istruzione da eseguire

# I REGISTRI

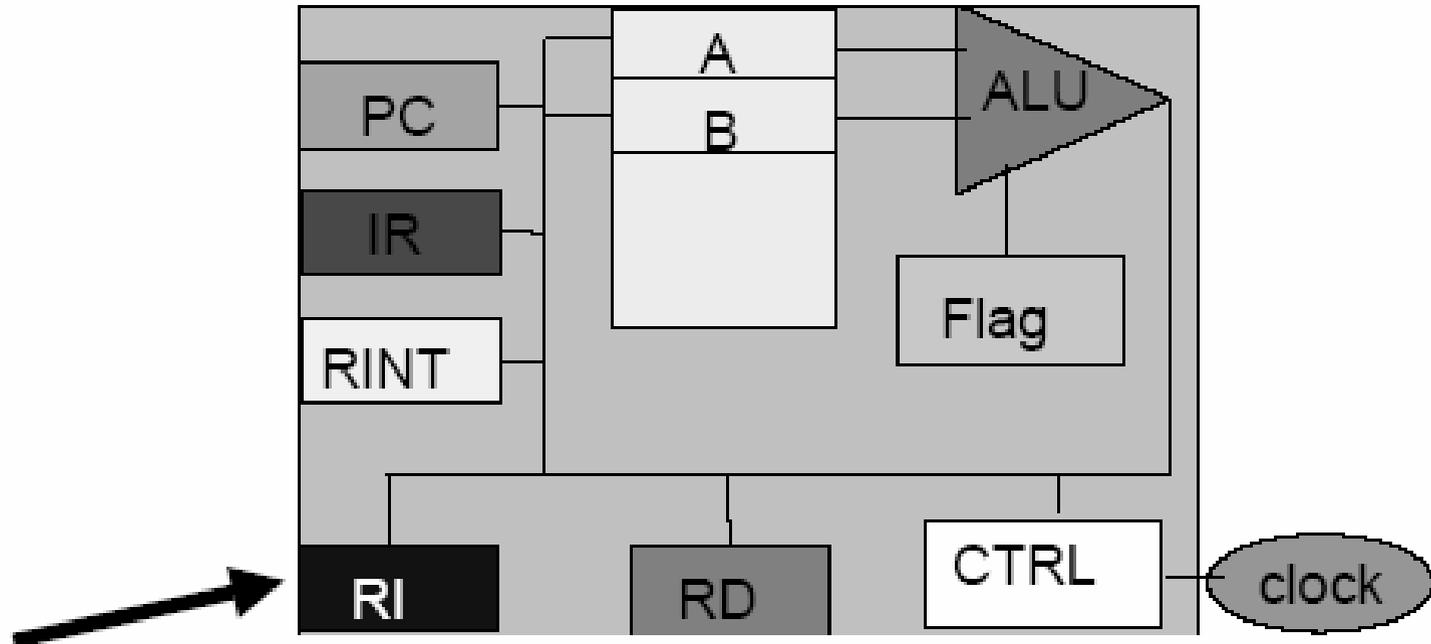
---



## **Instruction Register (IR)**

Contiene l'istruzione da eseguire.

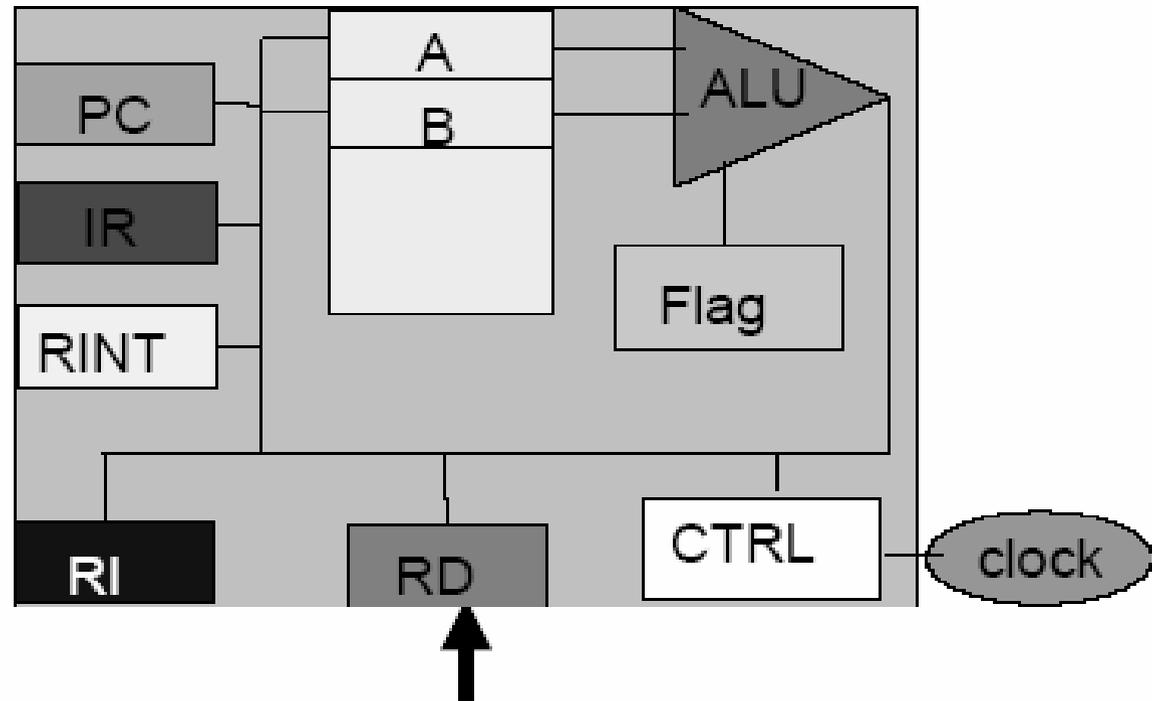
# I REGISTRI



## Registro Indirizzi (RI)

Contiene l'indirizzo della cella di memoria da selezionare per il trasferimento di un dato con la CPU

# I REGISTRI

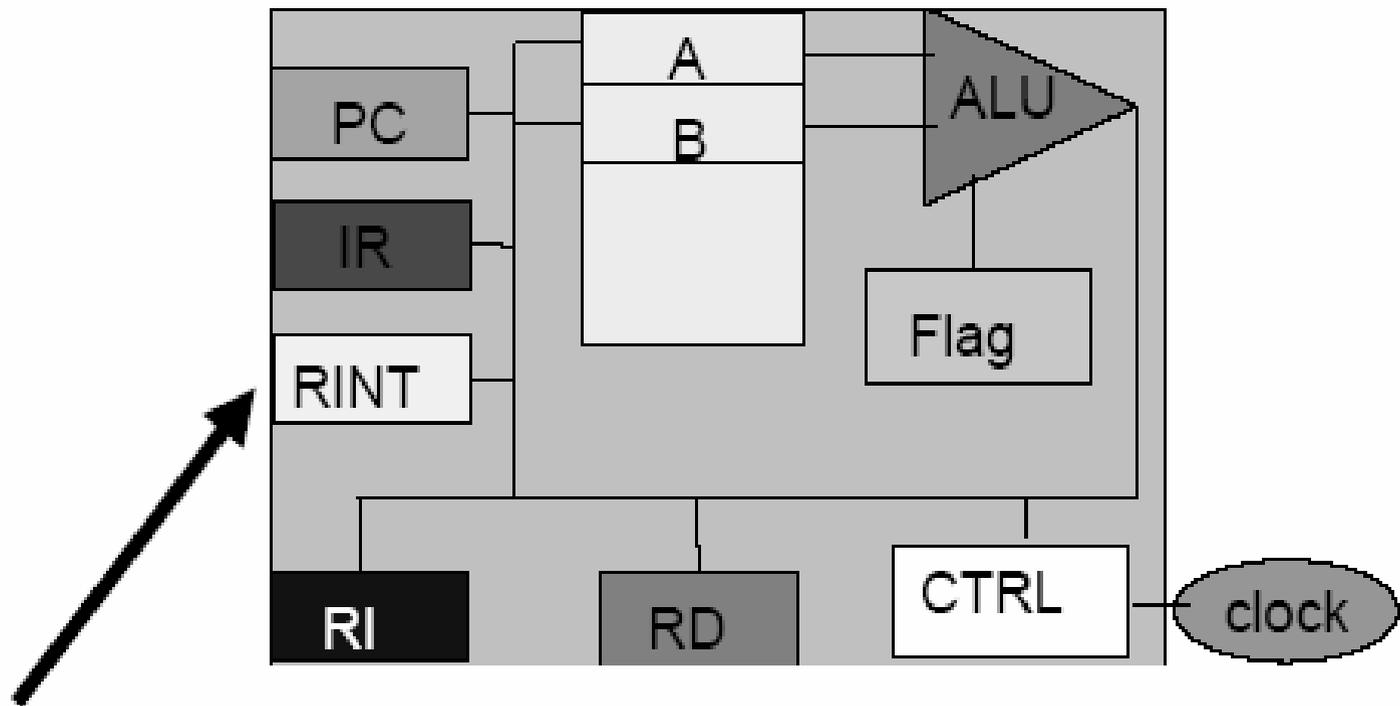


## **Registro Dati (RD) o *Accumulatore***

Contiene il dato appena trasferito dalla memoria oppure da trasferire nella memoria

# I REGISTRI

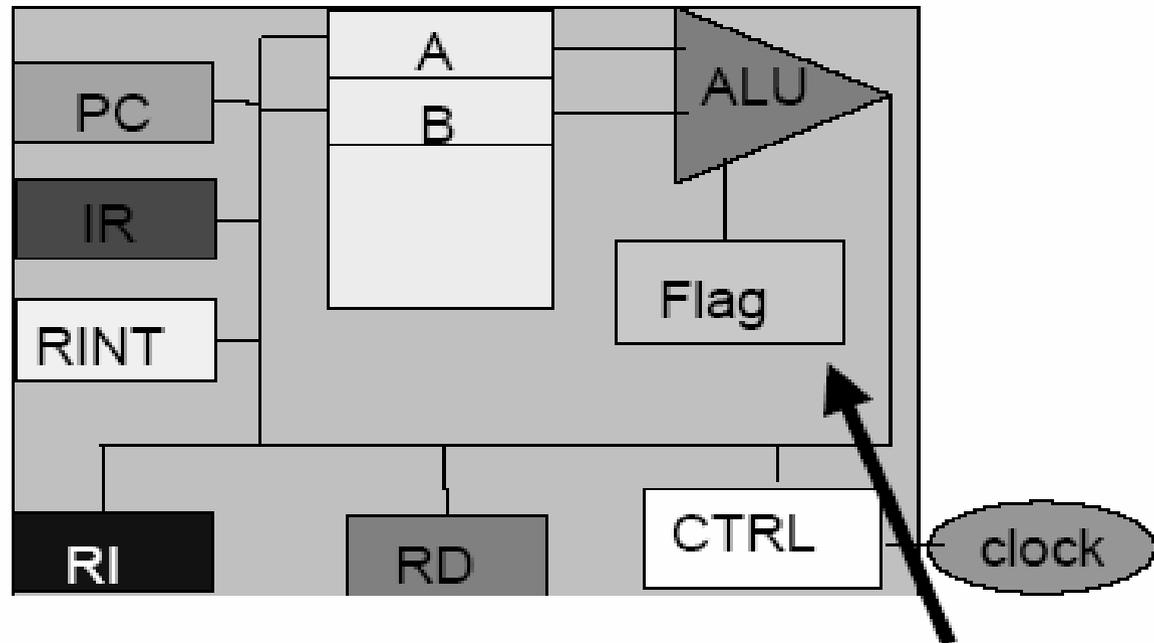
---



## **Registro Interruzioni (RINT)**

Serve per scopi particolari (non discussi)

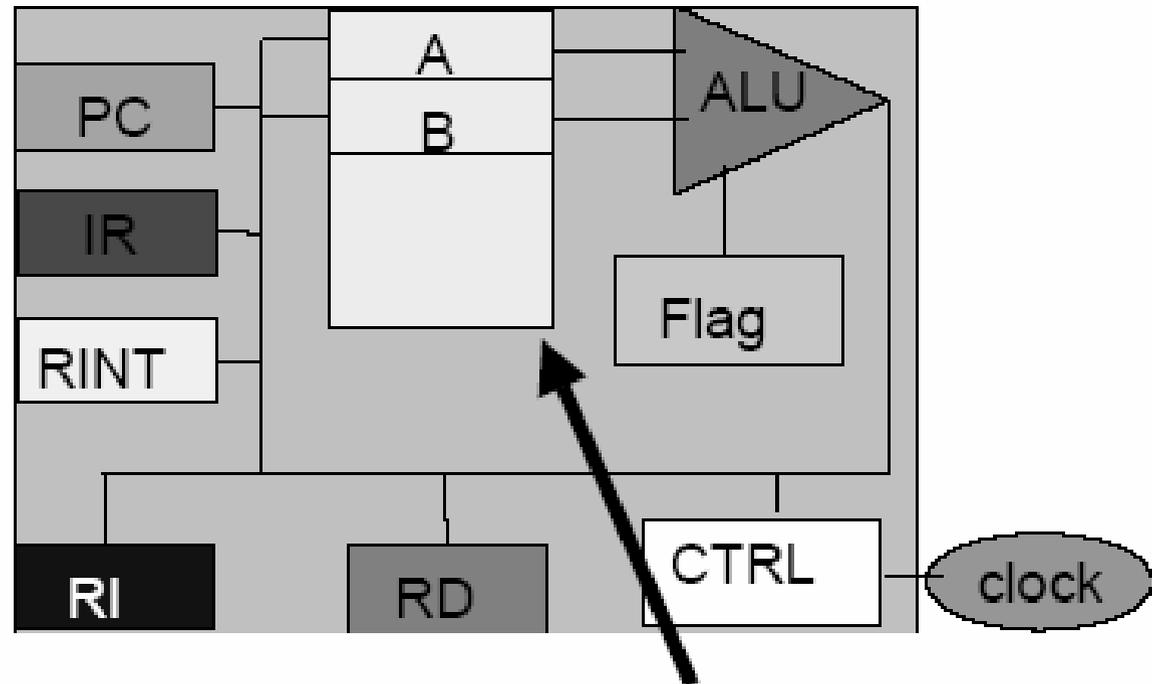
# I REGISTRI



## Registro dei Flag (Flag)

Ogni flag indica la presenza/assenza di una proprietà nell'ultimo risultato generato dalla ALU. Altri bit riassumono lo stato del processore.

# I REGISTRI



## **Registri di uso generale (A,B,C,...)**

Sono usati per contenere sia dati (in particolare, operandi di operazioni aritmetico/logiche) sia indirizzi.

# MULTITASKING

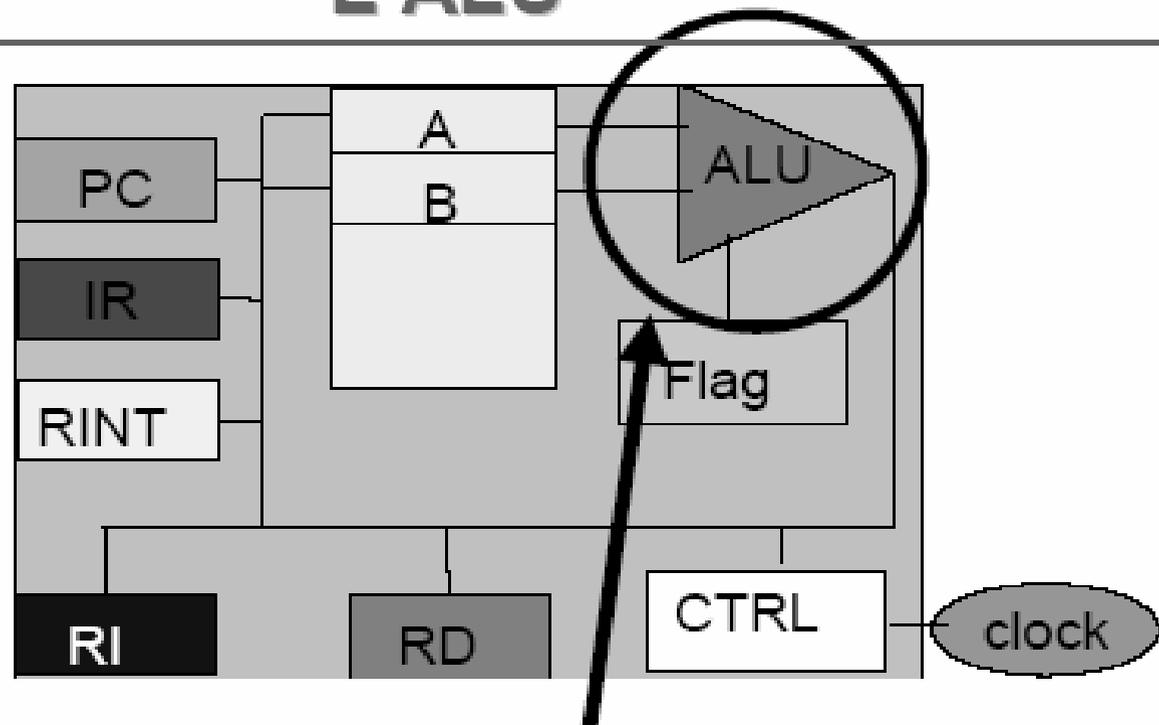
---

Poiché i registri compendiano *tutto lo stato dell'elaborazione* di un certo processo,

- salvando in memoria il contenuto di tutti i registri è possibile *accantonare un processo* per passare a svolgerne un altro
- ripristinando dalla memoria il contenuto di tutti i registri precedentemente salvati è possibile *ripristinare lo stato di un processo accantonato*, riprendendone l'esecuzione *come se nulla fosse accaduto*.

Questa possibilità è ciò che consente a un sistema operativo di eseguire *più compiti* "allo stesso tempo"

# L'ALU



Esegue operazioni aritmetiche, logiche e confronti sui dati della memoria centrale o dei registri.  
Può essere semplice oppure (più spesso) molto complessa e sofisticata.

# L'ALU (segue)

---

## ESEMPIO SEMPLICE:

ALU in grado di eseguire **somma**, **sottrazione**, **prodotto**, **divisione** con due operandi contenuti nei registri A e B.

1. I due operandi vengono caricati nei registri A e B;
2. La ALU viene attivata da un comando inviato dall'unità di controllo che specifica il tipo di operazione;
3. Nel registro A viene caricato il risultato dell'operazione eseguita dalla ALU;
4. Il registro FLAG riporta sui suoi bit indicazioni sul risultato dell'operazione (riporto, segno, etc.).



Alterazione di due bit nel registro **Flag**:  
**carry** (riporto) e **sign** (segno)

# LA MEMORIA CENTRALE

---

## INDIRIZZAMENTO

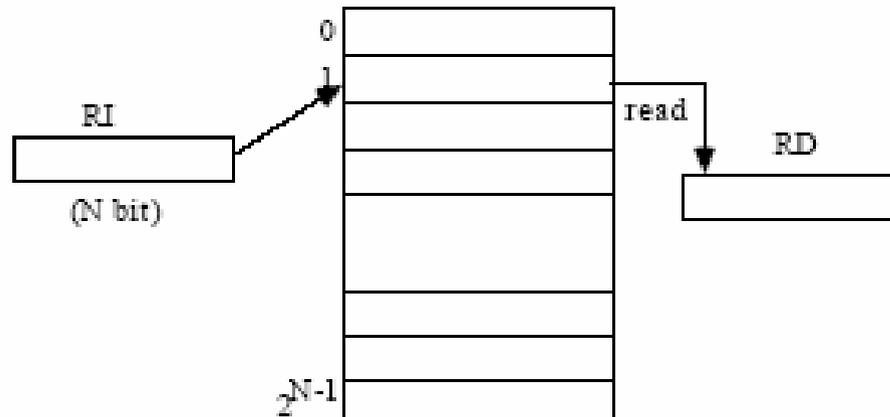
- E' l'attività con cui l'elaboratore seleziona una particolare cella di memoria
- Per farlo, l'elaboratore pone l'indirizzo della cella desiderata nel Registro Indirizzi (RI).
  - se il RI è lungo  $N$  bit, si possono indirizzare  $2^N$  celle di memoria (numerate da 0 a  $2^N-1$ ) solitamente di 1 byte
  - esempio:  $N=10 \Rightarrow 1024$  celle.
- Oggi, RI è lungo tipicamente 32 bit  
→ **SPAZIO INDIRIZZABILE di  $2^{32}$  byte = 4 Gb**

# LA MEMORIA CENTRALE (2)

---

## OPERAZIONI

- **Lettura (*Read*):** il contenuto della cella di memoria indirizzata dal Registro Indirizzi è copiato nel Registro Dati.

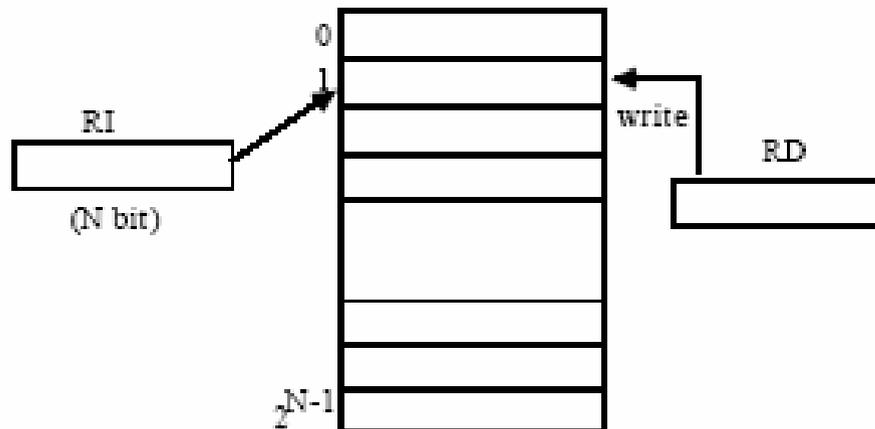


# LA MEMORIA CENTRALE (3)

---

## OPERAZIONI

- **Scrittura (*Write*):** il contenuto del Registro Dati è copiato nella cella di memoria indirizzata dal Registro Indirizzi.



# DISPOSITIVI DI MEMORIA

---

## DISPOSITIVI FISICI

- **RAM:** Random Access Memory (ad accesso casuale): su di essa si possono svolgere operazioni sia di lettura che di scrittura
- **ROM:** Read Only Memory (a sola lettura): non volatili e non scrivibili dall'utente (che la ordina con un certo contenuto); in esse sono contenuti i dati e programmi per inizializzare il sistema
- **PROM:** Programmable ROM. Si possono scrivere soltanto una volta, mediante particolari apparecchi (detti programmatori di PROM).

# DISPOSITIVI DI MEMORIA (segue)

---

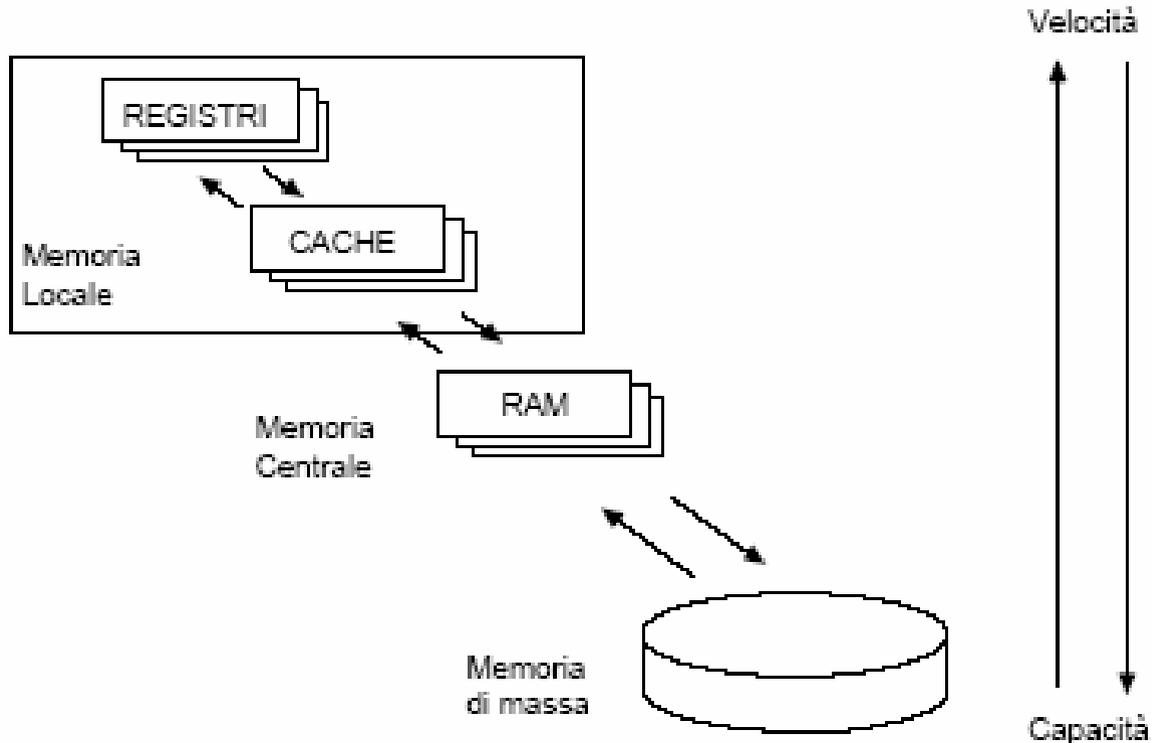
## DISPOSITIVI FISICI (segue)

- **EPROM**: Erasable-Programmable ROM (si cancellano sottoponendole a raggi ultravioletti).
- **EEPROM**: Electrically-Erasable-PROM (si cancellano elettricamente).

Il *Firmware* è costituito da software memorizzato su ROM, EPROM, etc. (codice microprogrammato).

# GERARCHIA DELLE MEMORIE

---



# MEMORIE CACHE

---

## **PROBLEMA:**

Sebbene la RAM sia veloce, non è **abbastanza** veloce da “star dietro” ai moderni processori.

## **CONSEGUENZA:**

il processore *perde tempo* ad aspettare l'arrivo dei dati dalla RAM.

## MEMORIE CACHE (2)

---

### SOLUZIONE:

Inserire tra processore e RAM una *memoria particolarmente veloce* dove tenere i dati usati più spesso (*memoria cache*)

In questo modo,

- ◆ la prima volta che il microprocessore carica dei dati dalla memoria centrale, tali dati vengono caricati anche sulla cache
- ◆ le volte successive, i dati possono essere letti dalla cache (*veloce*) invece che dalla memoria centrale (più lenta)

## MEMORIE CACHE (3)

---

### **DUBBIO:**

Ma se abbiamo memorie così veloci,  
*perché non le usiamo per costruire  
tutta la RAM?*

**Semplice...**

perché costano molto!!

**OGGI**, la cache è spesso già presente dentro al processore (cache di I° livello), e altra può essere aggiunta all'esterno del processore (cache di II° livello)

# BUS DI SISTEMA

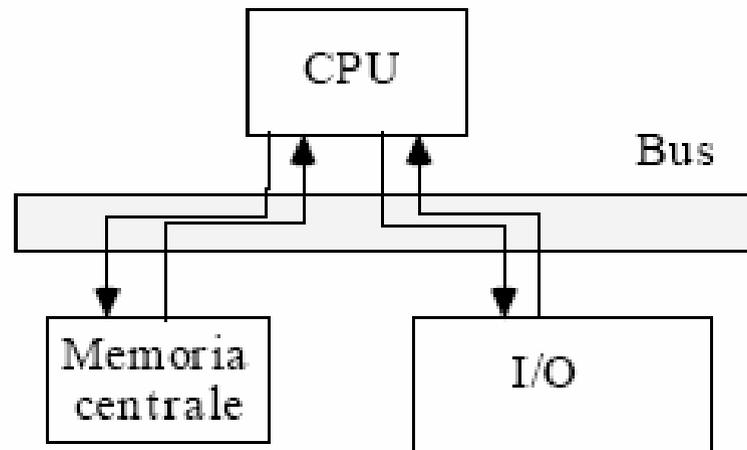
---



**Il Bus di Sistema** interconnette la CPU, la memorie e le interfacce verso dispositivi periferici (I/O, memoria di massa, etc.)

## BUS DI SISTEMA (2)

---



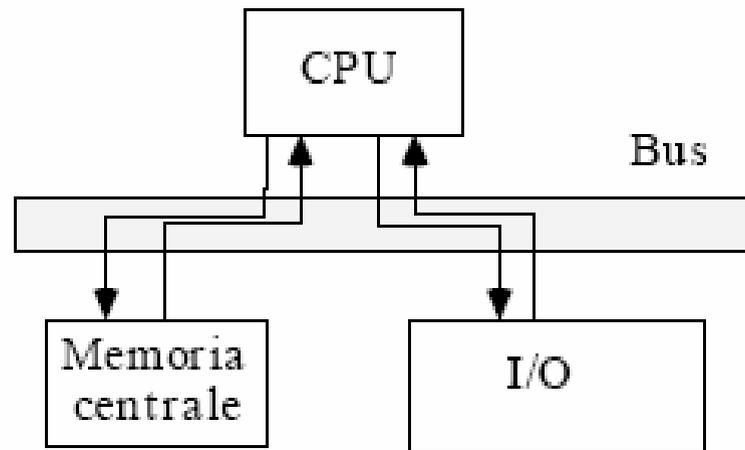
Il Bus collega *due unità funzionali alla volta*:

- una trasmette...
- ... e l'altra riceve.

Il trasferimento dei dati avviene o *sotto il controllo della CPU*.

## BUS DI SISTEMA (3)

---



Il Bus è in realtà un insieme di linee diverse:

- bus dati (*data bus*)
- bus indirizzi (*address bus*)
- bus comandi (*command bus*)

## BUS DI SISTEMA (4)

---

### BUS DATI

- bidirezionale
- serve per trasmettere dati *dalla memoria o viceversa.*

### BUS INDIRIZZI

- unidirezionale
- serve per trasmettere *il contenuto del registro indirizzi alla memoria*  
(si seleziona una specifica cella su cui viene eseguita o un'operazione di lettura o una operazione di scrittura)

## BUS DI SISTEMA (5)

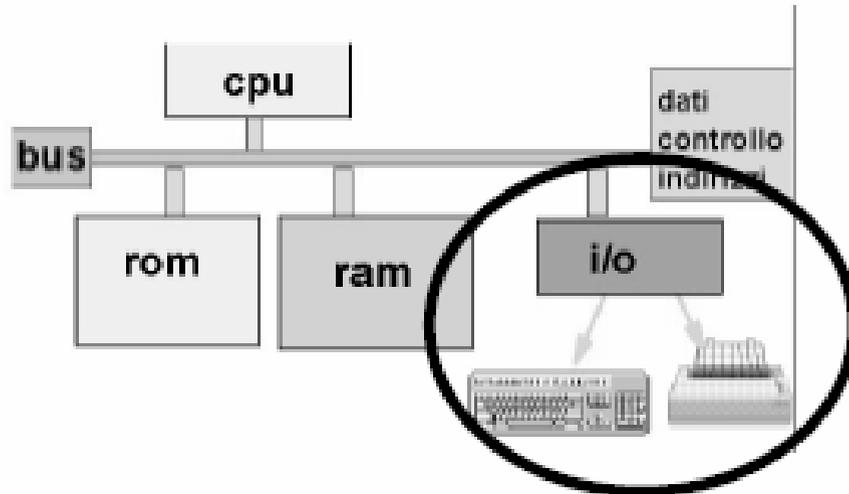
---

### BUS COMANDI

- **bidirezionale**
- tipicamente usato per *inviare comandi verso la memoria* (es: lettura o scrittura) o *verso una periferica* (es. stampa verso la stampante → interfaccia)
- può essere usato in alcuni casi per *inviare comandi verso il processore*

# INTERFACCE DI I/O

---



**Le interfacce sono molto diverse tra loro, e dipendono dal tipo di unità periferica da connettere.**

**Una interfaccia è un dispositivo che consente all'elaboratore di comunicare con una periferica (tastiere, mouse, dischi, terminali, stampanti, ...).**

# OLTRE la macchina di Von Neumann

---

- **Problema:** nella Macchina di Von Neumann le operazioni sono *strettamente sequenziali*.
- Altre soluzioni introducono forme di *parallelismo*
  - **processori dedicati** (*coprocessori*) al calcolo numerico, alla gestione della grafica, all'I/O.
  - **esecuzione in parallelo** delle varie fasi di un'istruzione: mentre se ne esegue una, si acquisiscono e decodificano le istruzioni successive (*pipeline*)
  - **architetture completamente diverse:** sistemi multi-processore, macchine dataflow, LISP machine, reti neurali.









