

INFORMATICA DI BASE

- 6 crediti -

Docente: Michele Piana

Email: michele.piana@univr.it

URL: <http://www.di.univr.it/~piana>

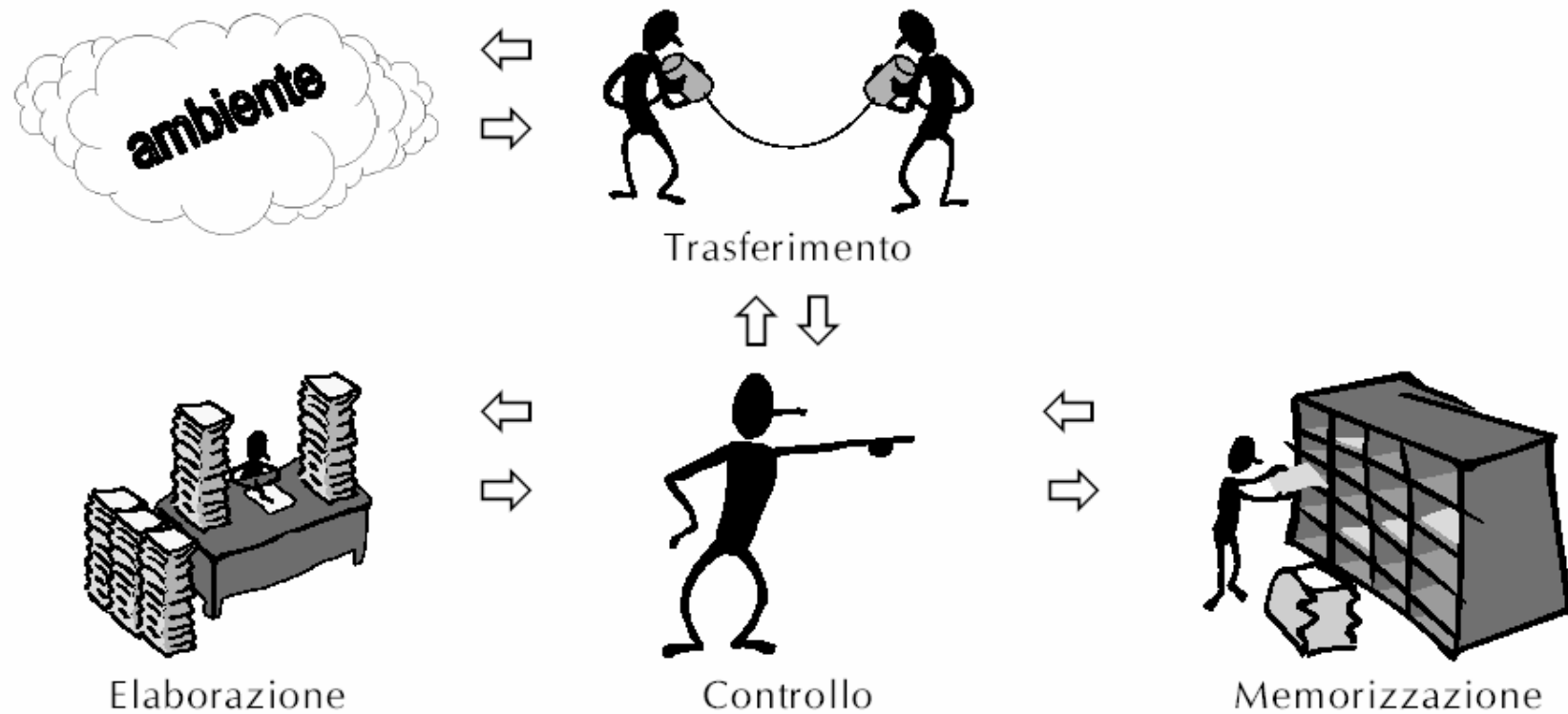
Ricevimento: Lunedì ore 14:00 – 15:00

Studio: 2.05 (Ufficio del Preside)

CONTENUTO DELLA LEZIONE

- **Funzioni di un calcolatore**
- **Architettura**
- **La CPU**
- **La Memoria**

Funzionalità di un calcolatore



Funzioni del Calcolatore - 1

Elaborazione dei dati: il calcolatore elabora i dati provenienti dall'ambiente esterno per fornire all'ambiente stesso i risultati ottenuti.

Tipo di applicazione ↔ Architettura

Le architetture attuali sono generali

Funzioni del Calcolatore - 2

Memorizzazione dei dati:

- Di breve periodo per salvare i dati di elaborazione intermedi**
- Di lungo periodo per immagazzinare i dati (archivi elettronici)**

Funzioni del Calcolatore - 3

Trasferimento dei dati:

- Un calcolatore deve essere in grado di comunicare con l'esterno.

→ Operazioni di *Input/Output*

- Si usano le *periferiche*

Es: collegamento in rete

Funzioni del Calcolatore - 4

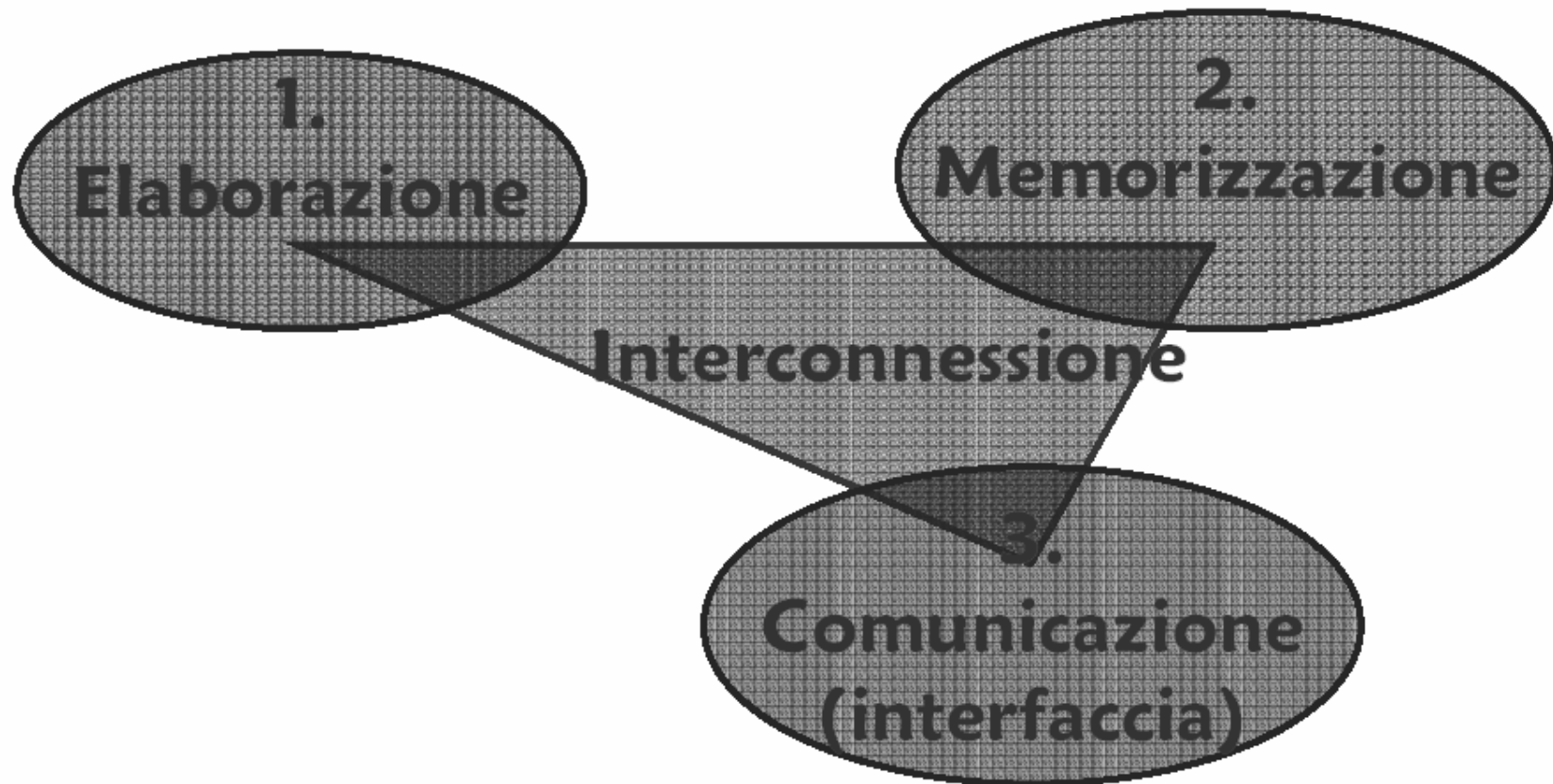
Controllo:

- **Controllo da parte dell'utente**
- **Controllo interno tramite *unità di controllo***

Caratteristiche dell'architettura

- **Flessibilità**
 - adatta a svolgere diverse tipologie di compiti
- **Modularità**
 - ogni componente ha una funzione specifica
- **Scalabilità**
 - ogni componente può essere sostituito con uno equivalente
- **Standardizzazione**
 - componenti facilmente sostituibili in caso di malfunzionamento
- **Riduzione dei costi**
 - grazie alla produzione su larga scala
- **Semplicità**
 - di installazione ed esercizio del sistema

Il calcolatore: modello concettuale



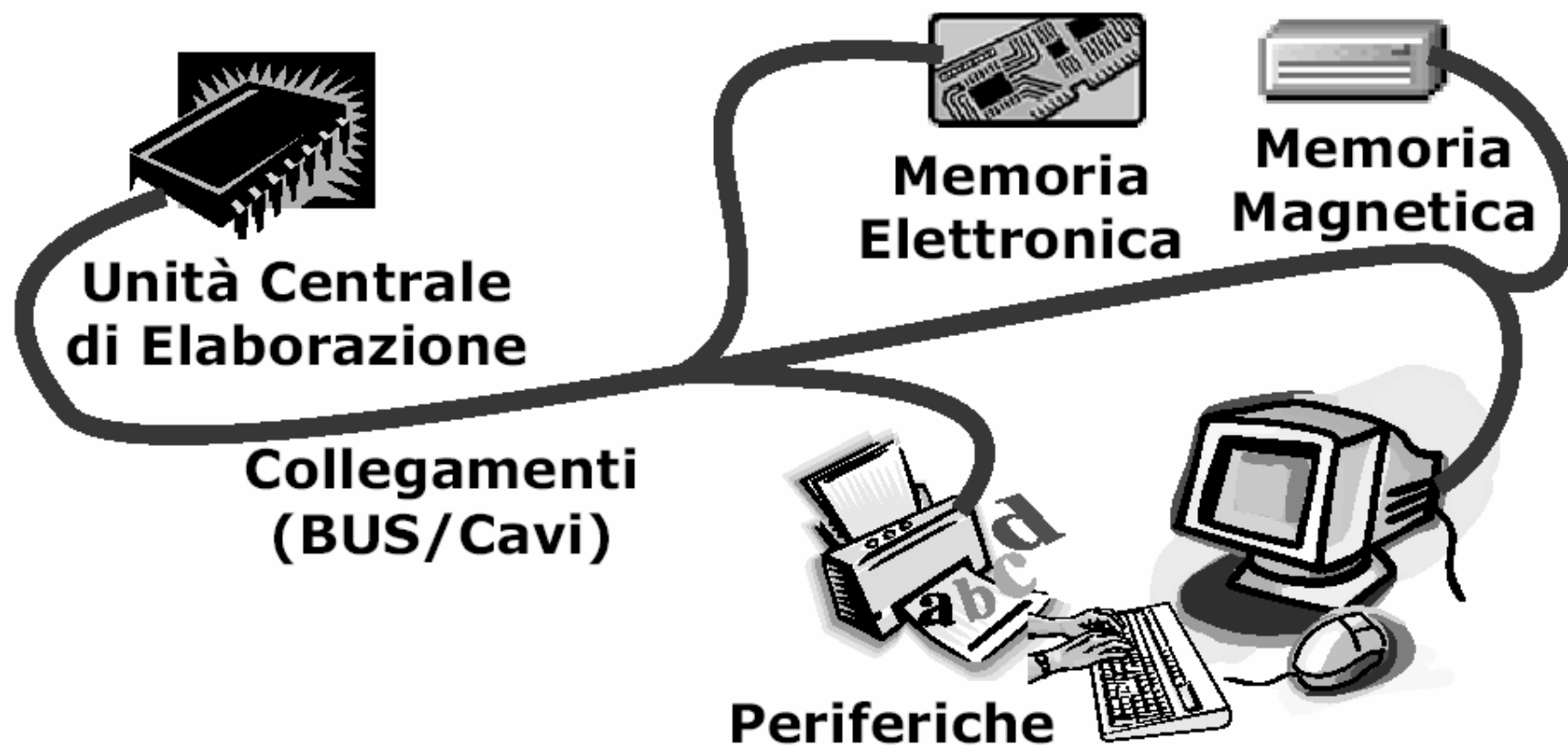
Funzioni e architetture

- **Per ognuna delle funzioni di base svolte dal calcolatore si possono prendere in esame relativi componenti fisici (HW)**

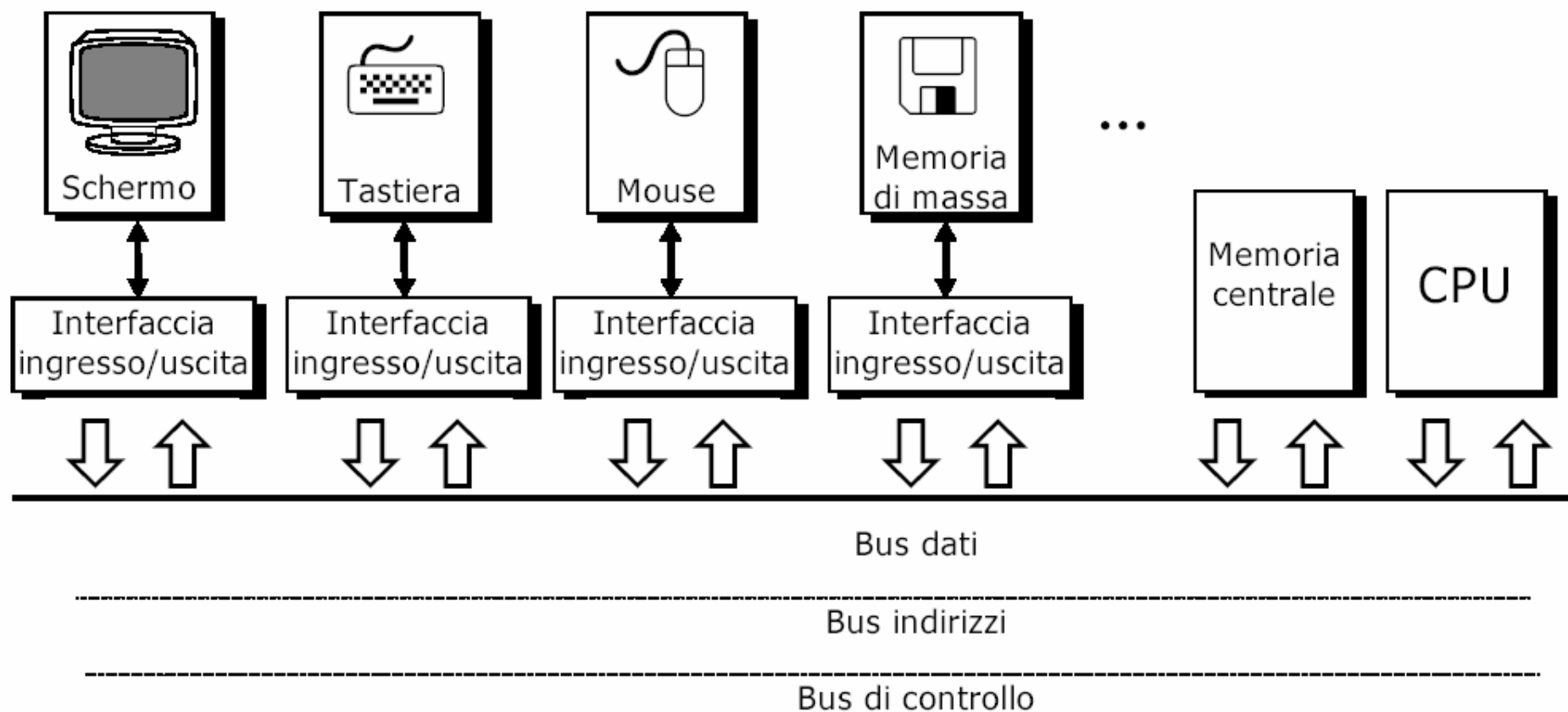


Macchina di Von Neumann

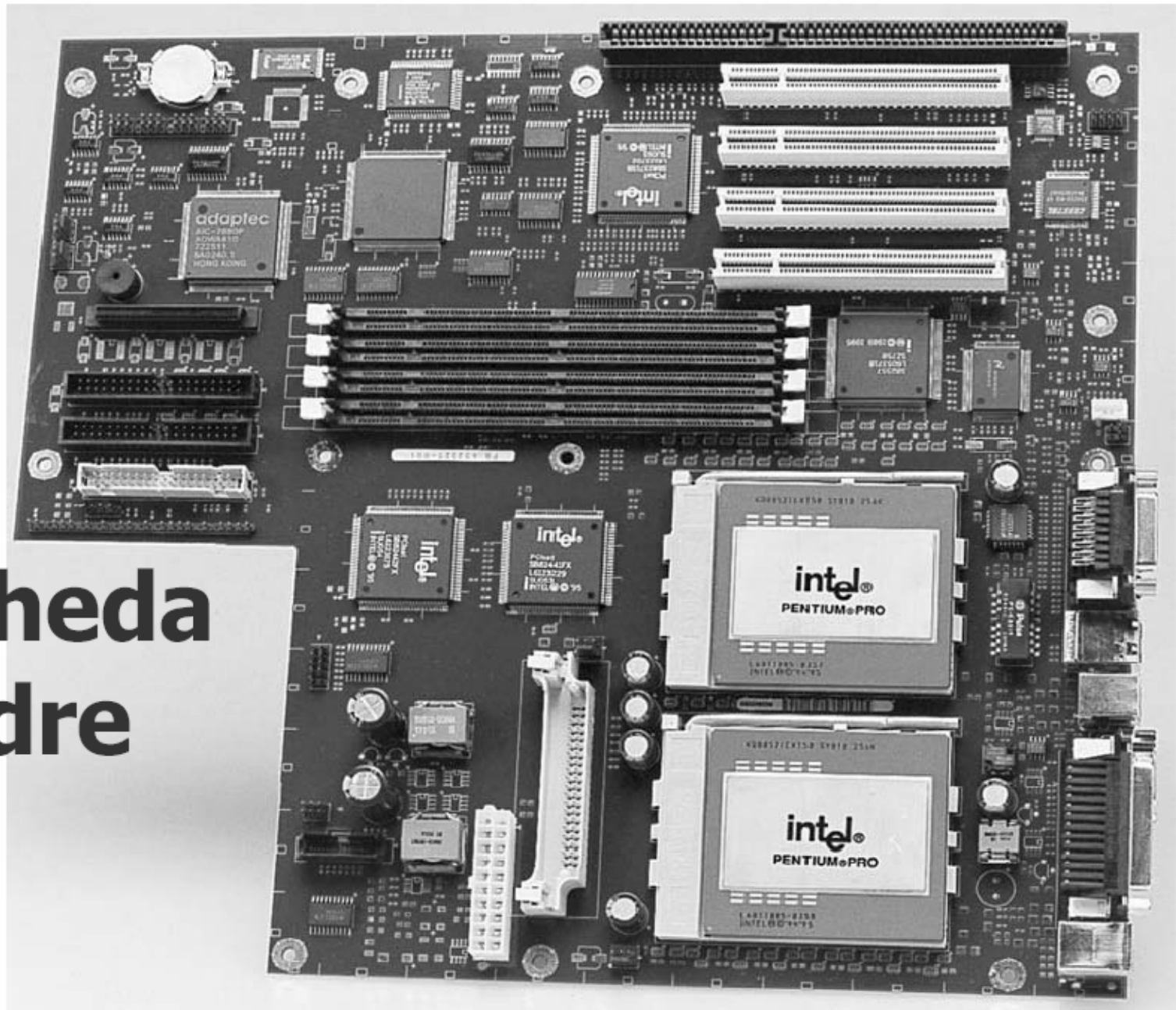
Il calcolatore: modello architetturale



Lo schema di riferimento



La scheda madre



Collegamento tra unità: BUS

- Il bus consente il trasferimento tra le unità
- La CPU controlla in maniera centralizzata la gestione dell'intero sistema (*master*)
 - sincronismo

Caratteristiche del collegamento a BUS

- Semplicità
 - un'unica linea di connessione → costi ridotti di produzione
- Estendibilità
 - aggiunta di nuovi dispositivi molto semplice
- Standardizzabilità
 - regole per la comunicazione da parte di dispositivi diversi
- Lentezza
 - utilizzo in mutua esclusione del bus
- Limitata capacità
 - al crescere del numero di dispositivi collegati
- Sovraccarico del processore (CPU)
 - perchè funge da *master* sul controllo del bus

BUS: componenti

- **Bus dati**: usato per trasferire i dati
- **Bus indirizzi**: identifica la posizione delle celle di memoria in cui la CPU scrive o legge
- **Bus di controllo**: coordinamento del sistema (connessioni, direzione di scambio, selezione unità coinvolte)

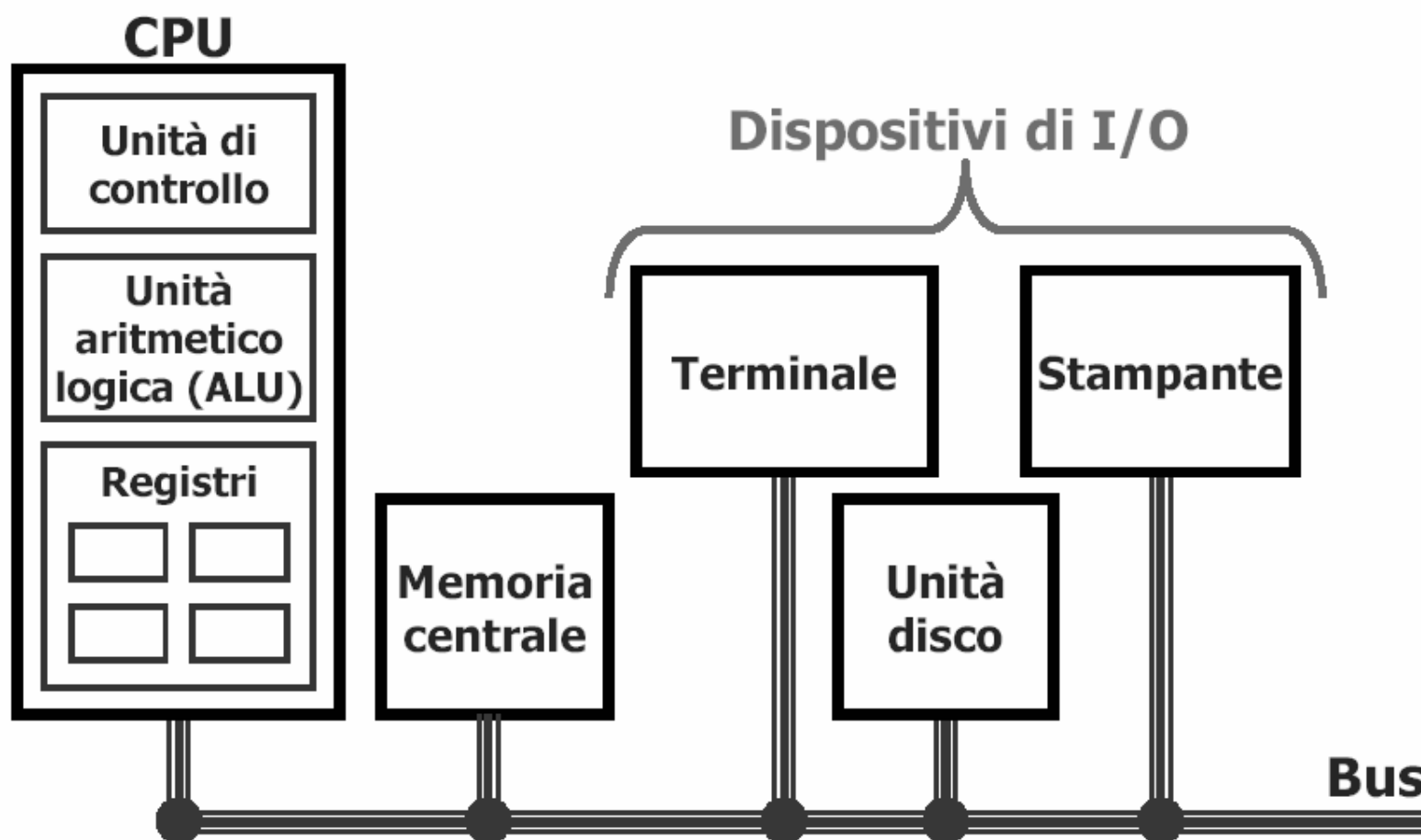
Unità centrale di elaborazione CPU

CPU e linguaggio macchina

Linguaggio macchina o *assembler*

- Istruzioni rudimentali (pochi operandi, poche istruzioni di base disponibili, etc.)**
- La programmazione in assembler richiede una conoscenza approfondita della architettura specifica su cui si sta lavorando.**
- Codifica binaria (identificazione dell'istruzione, indirizzi operandi)**

Organizzazione tipica di un calcolatore "bus oriented"



Elementi di una CPU

➤ **Unità di controllo**

- legge le istruzioni dalla memoria e ne determina il tipo.

➤ **Unità aritmetico-logica**

- esegue le operazioni necessarie per eseguire le istruzioni.

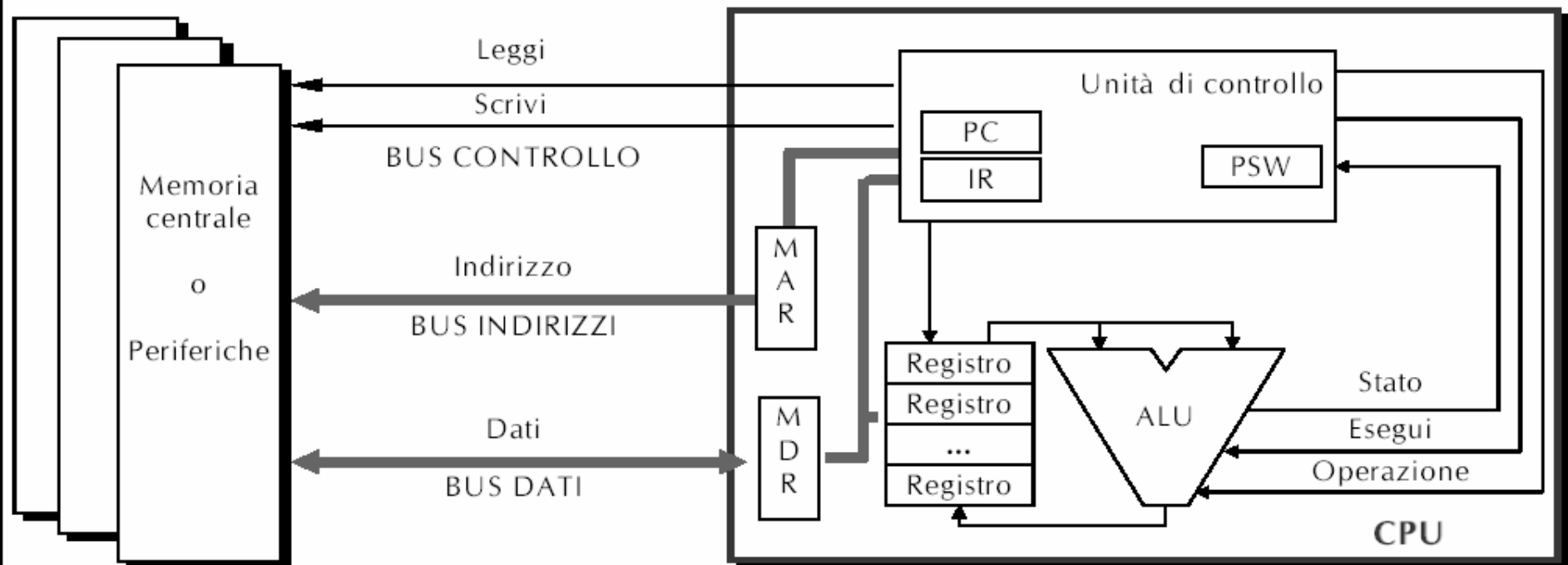
➤ **Registri**

- **memoria ad alta velocità** usata per risultati temporanei e informazioni di controllo;
- il **valore massimo** memorizzabile in un registro è determinato dalle **dimensioni** del registro;
- esistono registri di uso generico e registri specifici:
 - **Program Counter (PC)** – qual è l'istruzione successiva;
 - **Instruction Register (IR)** – istruzione in corso d'esecuzione;
 - ...

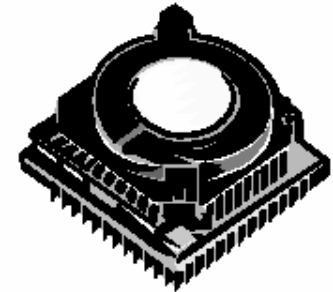
REGISTRI

- PC (Program Counter): indica la prossima istruzione da eseguire specificando la sua posizione (indirizzo) in memoria.
- IR (Instruction Register): contiene una copia dell'istruzione da eseguire.
- MAR (Memory Address Register): contiene l'indirizzo di memoria ove estrarre o trascrivere un dato ed è direttamente collegato al bus indirizzi.
- MDR (Memory Data Register): contiene una copia del dato da trascrivere o estrarre dalla memoria ed è direttamente collegato al bus dati.
- PSW (Processor Status Word): contiene informazioni sulle condizioni verificatesi in seguito all'esecuzione.

La struttura della CPU



CPU



- In grado di eseguire solo istruzioni codificate in **linguaggio macchina**
- Ciclo Fetch – Decode - Execute
 1. Prendi l'istruzione corrente dalla memoria e mettila nel registro istruzioni (IR) (***fetch***)
 2. Incrementa il Program Counter (PC) in modo che contenga l'indirizzo dell'istruzione successiva
 3. Determina il tipo di istruzione da eseguire (***decode***)
 4. Se l'istruzione necessita di un dato in memoria determina dove si trova e caricalo in un registro della CPU
 5. Esegui l'istruzione (***execute***)
 6. Torna al punto 1 e opera sull'istruzione successiva

ESECUZIONE

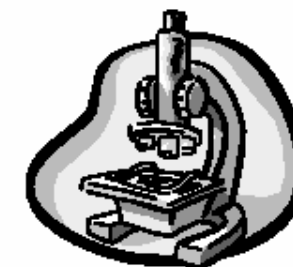
Le operazioni di esecuzione dipendono dall'istruzione

- 1. Trasferimento dati tra CPU e memoria centrale**
- 2. Trasferimento dati tra CPU e interfacce di I/O**
- 3. Elaborazione dei dati: la CPU esegue operazioni aritmetico-logiche**
- 4. Controllo del flusso (Es: modifica della sequenza di esecuzione)**

CLOCK e FREQUENZA

- **L'unità di controllo comanda il funzionamento della CPU tramite l'emissione di segnali.**
- **Occorre temporizzare gli eventi**
- **Uso di un orologio (clock) che fornisce una cadenza temporale a cui tutte le attività elementari sono sincronizzate**
- **La frequenza del clock determina dunque il numero di attività elementari per unità di tempo (hertz o MegaHertz – MHz)**

Evoluzione delle CPU



CPU	Anno	Frequenza (MHz)	Dimensione registri / bus dati	Numero di transistor
8086	1978	4.77 — 12	8 / 16	29 000
80286	1982	8 — 16	16 / 16	134 000
80386	1986	16 — 33	32 / 32	275 000
80386 SX	1988	16 — 33	32 / 16	275 000
80486	1989	33 — 50	32 / 32	1 200 000
Pentium	1993	60 — 200	32 / 64	3 100 000
Pentium II	1997	233 — 400	32 / 64	7 500 000
Pentium III	1999	450 — 1133	32 / 64	24 000 000
Pentium 4	2000	1600 — 2000	32 / 64	42 000 000

La memoria

- **Supporto alla CPU:** deve fornire alla CPU dati e istruzioni il più rapidamente possibile
- **Archivio:** deve consentire di archiviare dati e programmi garantendone la conservazione e la reperibilità anche dopo elevati periodi di tempo
- Diverse esigenze:
 - **velocità** per il supporto alla CPU
 - **non volatilità** ed **elevate dimensioni** per l'archivio
- Diverse tecnologie
 - **elettronica:** veloce, ma costosa e volatile
 - **magnetica** e **ottica:** non volatile ed economica, ma molto lenta

Criteri di caratterizzazione di una memoria

- Velocità
 - tempo di accesso (quanto passa tra una richiesta e la relativa risposta)
 - velocità di trasferimento (quanti byte al secondo si possono trasferire)
- Volatilità
 - cosa succede quando la memoria non è alimentata?
 - per quanto tempo i dati vi rimangono immagazzinati?
- Capacità
 - quanti byte può contenere? qual è la dimensione massima?
- Costo (per bit)
- Modalità di accesso
 - diretta (o casuale): il tempo di accesso è indipendente dalla posizione
 - sequenziale: il tempo di accesso dipende dalla posizione
 - mista: combinazione dei due casi precedenti
 - associativa: indicato il dato, la memoria risponde indicando l'eventuale posizione che il dato occupa in memoria.

La memoria centrale (R.A.M.)

- Mantiene al proprio interno i dati e le istruzioni dei programmi in esecuzione
- Memoria ad accesso "casuale"
- Tecnologia elettronica
 - **veloce** ma **volatile** e **costosa**
- Due "eccezioni"
 - **R.O.M.:** *elettronica ma permanente e di sola lettura*
 - **Flash:** *elettronica ma permanente e riscrivibile*

Indirizzi di memoria

- I bit nelle memorie sono raggruppati in **celle**:
 - tutte le celle sono formate dallo **stesso numero di bit**;
 - una cella composta da **k bit**, è in grado di contenere una qualunque tra **2^k combinazioni** diverse di bit.
- Ogni cella ha un **indirizzo**:
 - serve come accesso all'informazione;
 - in una memoria con **N celle** gli indirizzi vanno da **0** a **N-1**.
- **La cella è l'unità indirizzabile più piccola.**
In quasi tutti i calcolatori è di **8 bit** (un **byte**).
- I byte vengono raggruppati in **parole** (che oggi sono di **32/64 bit**), su cui la CPU esegue le operazioni.

Memoria vs. CPU

- Le **CPU** sono sempre state **più veloci** delle **memorie**
 - l'aumento di integrazione ha consentito di realizzare **CPU** pipeline e super scalari, molto efficienti e **veloci**;
 - nelle memorie è aumentata la **capacità** più che la **velocità**.
- L'**accesso** alla memoria passa **attraverso** il **bus**
 - la **frequenza** di funzionamento **del bus** è molto **più bassa** di quella della CPU;
 - il **bus** può essere **impegnato** ad effettuare trasferimenti controllati **da dispositivi** di I/O "**autonomi**" (e.g. DMA).
- **È difficile riordinare le istruzioni** in modo da poter sfruttare i tempi di attesa della memoria.
- È possibile fare **memorie molto veloci** se stanno nel chip della CPU, ma **sono piccole e costose**.

Il principio di località

➤ Località spaziale:

quando si accede all'indirizzo A, è molto probabile che gli accessi successivi richiedano **celle vicine ad A**.

- le istruzioni del codice vengono in genere lette da locazioni consecutive della memoria;
- gli accessi ad array o a strutture dati sono "vicini".

➤ Località temporale:

quando si accede all'indirizzo A, è molto probabile negli accessi successivi si richieda **di nuovo** la cella **A**.

- cicli di istruzioni accedono ripetutamente alle stesse locazioni di memoria;
- istruzioni vicine tendono ad utilizzare le stesse variabili.

Come si sfrutta la località

- Diversi approcci a seconda del tipo di località:
 - località **temporale**: i dati prelevati dalla MGL vengono conservati nella MPV **il più a lungo possibile**;
 - località **spaziale**: quando si copia un dato dalla MGL alla MPV, si copiano anche i dati vicini (**cache line** o **blocco**).
- La frequenza di successo (**hit ratio – h**) cresce fino a **superare il 99%**:
 - in effetti **h** dipende da due caratteristiche contrastanti:
 - **la dimensione dei blocchi**
un blocco grande sfrutta meglio la località spaziale;
 - **quanti sono i blocchi in memoria**
se c'è spazio per tanti blocchi un dato resta in memoria più a lungo e può sfruttare più a lungo la località temporale;
 - c'è anche il problema del **costo** della cache!

Una gerarchia di memoria

Ottenuta per “generalizzazione” dell’applicazione del principio di **località** e tipicamente costituita da

1. **registri** contenuti nella CPU (qualche KB)
2. **cache** (da circa 32KB a circa 1024KB)
3. **memoria principale** (da circa 64MB a qualche GB)
4. **dischi fissi** (da qualche GB a qualche TB)
5. **nastri magnetici e dischi ottici** (da qualche GB a qualche TB per ogni supporto)

Man mano che ci si sposta verso il basso nella gerarchia aumenta il valore dei parametri fondamentali:

- **aumenta il tempo di accesso;**
- **aumenta la capacità** di memorizzazione;
- **ma diminuisce il costo per bit.**

RIFERIMENTI AL LIBRO

- Architettura e funzioni di un calcolatore:
Cap. 5 (introduzione)
- CPU
Paragrafi 5.1, 5.2, 5.2.1, 5.2.2, 5.2.3
- Memoria
Paragrafi 5.3, 5.3.1