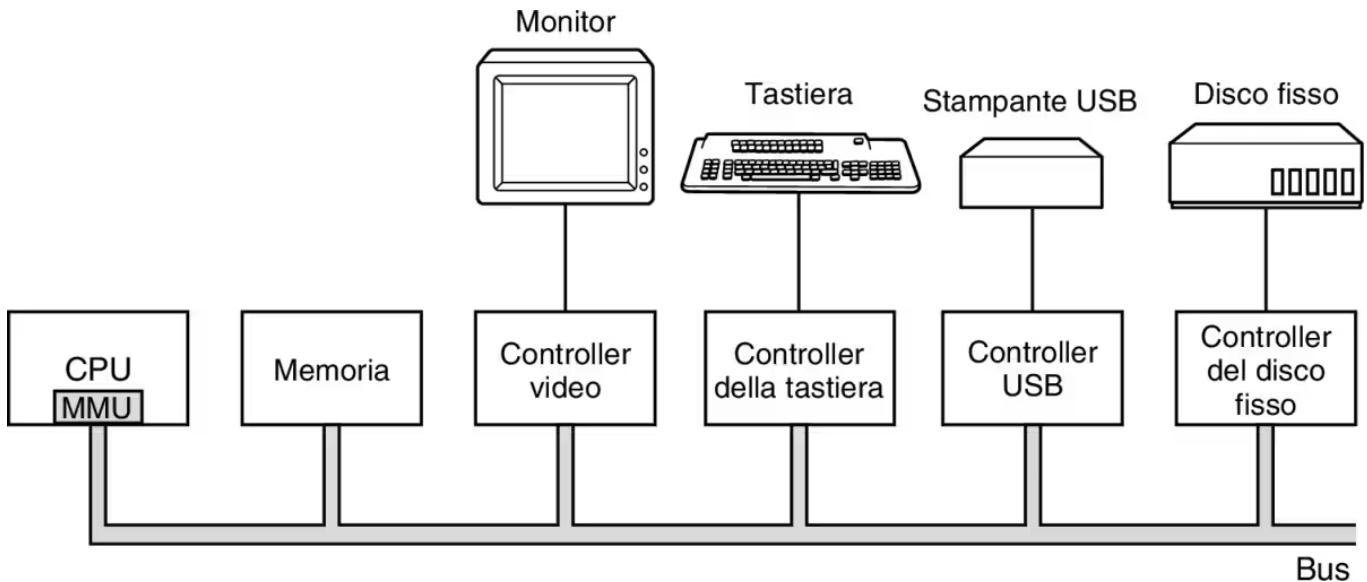


# Uno sguardo all'hardware

Abbiamo capito che il SO è molto legato all'hardware della macchina su cui gira, d'altronde il SO non è nient'altro che un'astrazione delle componenti di un computer.

A questo punto guardiamo molto velocemente le componenti principali:



## Processore

È il cervello del computer esegue le istruzioni dalla memoria sulla base del ciclo:

- **Fetch** (Preleva)
- **Decode** (Decodifica)
- **Execute** (Esegui)

Ogni programma è eseguito grazie a questo ciclo.

Possono eseguire un insieme specifico di istruzioni e sono composti da dei registri che permettono di memorizzare dati importanti e i risultati delle operazioni.

Alcuni registri speciali:

- **Program Counter**: Indica l'istruzione successiva in memoria.
- **Stack Pointer**: Punta alla cima dello stack in memoria.
- **Program Status Word**: Contiene informazioni sullo stato del programma.

✍ **Stack:** contiene i frame di procedure con parametri e variabili locali.

Il SO gestisce il **multiplexing temporale della CPU** durante il quale salva e ripristina i registri, in modo da eseguire programmi in maniera efficiente.

Una progettazione avanzata della CPU prevede la **pipeline**.

✍ **Pipeline:** E' un tipo di architettura hardware utilizzata nel progetto dei microprocessori per incrementarne la produttività, ovvero la quantità di istruzioni eseguite nell'unità di tempo.

Quando abbiamo di più processori si parla di: **Multithreading** nei quali abbiamo all'interno della CPU lo stato di due thread (sottoparti di un processo) anche se non ci sta una vera e propria esecuzione in parallelo.

Alcuni vantaggi dei multiprocessori:

- **Throughput**
- **Economia di scala**
- **Affidabilità**

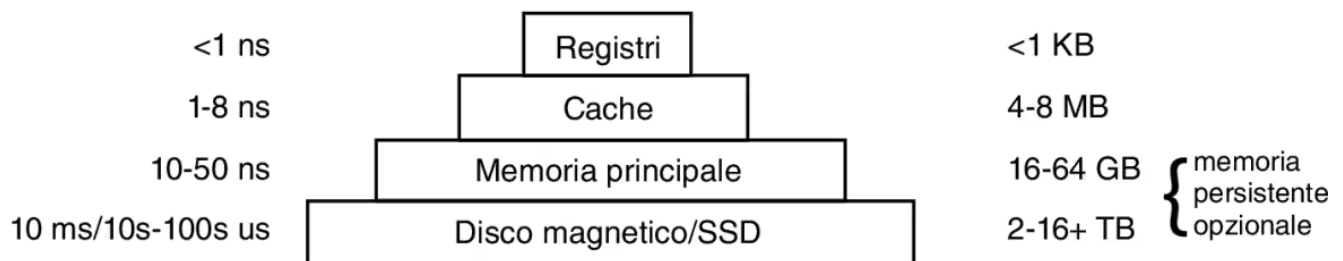
Un impatto fondamentale su il multithreading lo hanno le **Cache**.

## Memorie

Le memorie può essere visualizzato tramite una piramide:

Tempo d'accesso tipico

Capacità tipica



Nella parte **alta** abbiamo le memorie più piccole e veloci, nella parte **bassa** abbiamo le memorie più lente, ma grandi.

Ultimamente la piramide sta subendo dei cambiamenti con la nascita delle SSD.

LE SSD non sono veri e propri dischi magnetici perché non ci sta un disco.

# Dispositivi di I/O

Sono composti da due componenti

- **Controller:** Per semplificare l'utilizzo da parte del SO.
- **Dispositivo in sè:** Interfaccia elementare, ma complicata da usare.

Ogni controller ha bisogno di un *Driver* per il SO.

I driver interagiscono con il controller attraverso le porte di I/O, quindi tramite:

- Istruzioni tipo IN/OUT
- Mappatura in memoria

Esecuzione dell'I/O:

1. Il processo esegue la chiamata di sistema;
2. Il kernel effettua una chiamata al driver;
3. Il driver avvia l'I/O;

Poi il driver interroga il dispositivo per vedere se ha finito o chiede al dispositivo di generare un interrupt quando ha finito (Fa il compito di un arbitro).

 L'arbitro gestisce le precedenze di utilizzo dei dispositivi hardware all'interno di un computer.

Esiste un hardware speciale che semplifica il tutto, il **DMA**.

## DMA

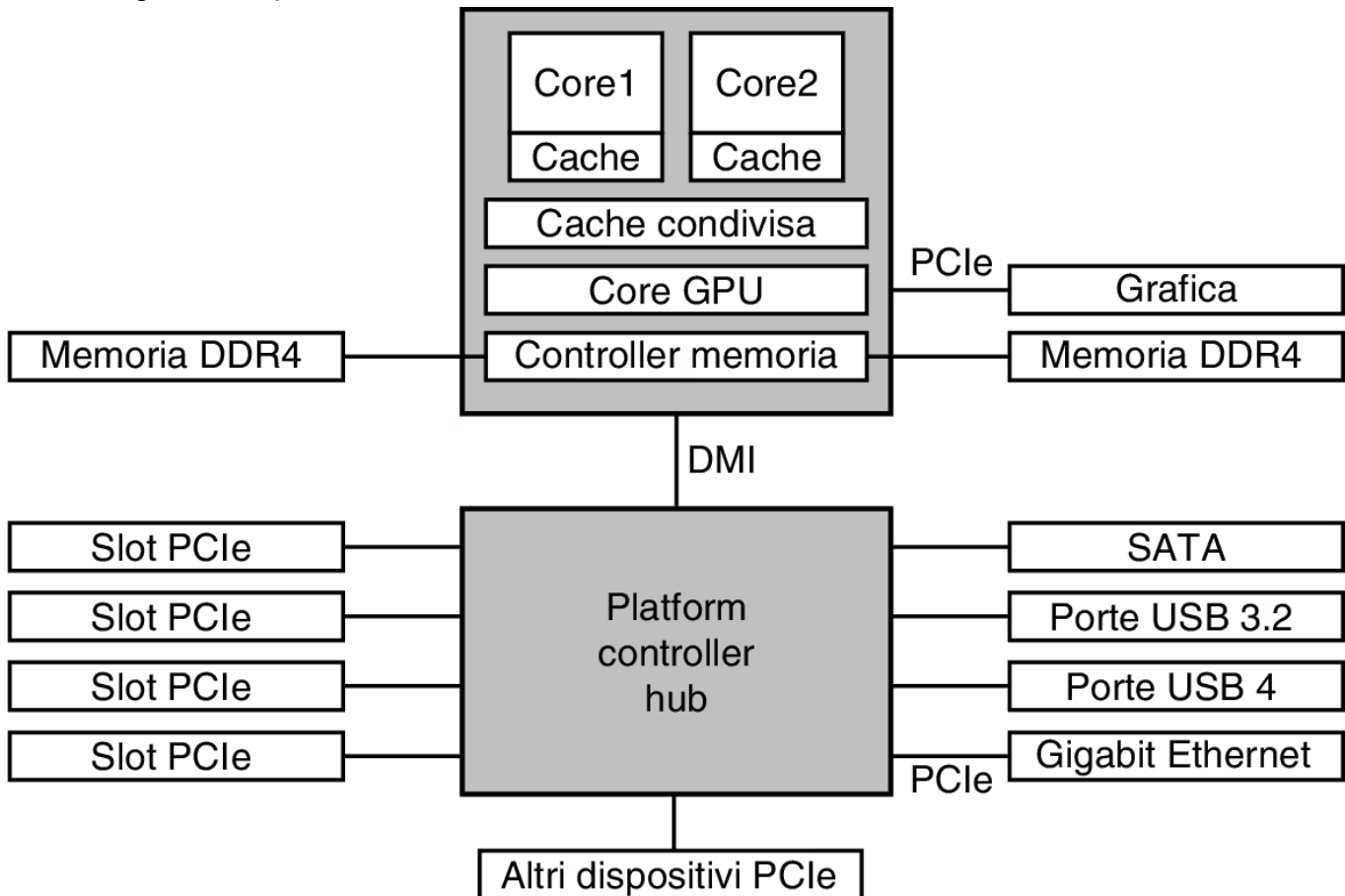
Acronimo di **Direct Access Memory**, consente alle componenti di accedere alla memoria senza dover passare per la CPU, ciò riduce il carico di lavoro della CPU sulle operazioni di I/O, permettendosi di concentrarsi su compiti critici.

## Buses

L'evoluzione dei computer ha portato all'aggiunta di diversi BUS che gestiscono il traffico dei dati.

La CPU comunica con la memoria attraverso un bus veloce DDR4 con una periferica grafica esterna sul bus PCIe.

Con tutti gli altri dispositivi comunica **attraverso un hub** su un bus DMI.



✍ Il bus principale è il **PCIe** che è il più veloce grazie ad un'architettura a connessione punto a punto dedicate, più veloce a bus condivisi.

I Dispositivi legacy possono essere collegati a un processore hub separato.

## Avvio del sistema (BOOT)

Sulla scheda madre ci sta una memoria flash contenente il firmware (**BIOS**, **tipo specifico di firmware**).

Dopo aver premuto il tasto di accensione:

1. La CPU **esegue il BIOS**;
2. Il BIOS **inizializza RAM e altre risorse**;
3. Il BIOS esegue la **scansione dei bus PCI/PCIe** e inizializza i dispositivi.
4. Imposta il **firmware runtime** per i servizi critici che il sistema usa dopo l'avvio.

Successivamente **cerca la posizione della tabella delle partizioni sul secondo settore dei dispositivi di avvio che contiene le posizioni di altre partizioni.**

Il BIOS è in grado di leggere semplici file system e avvia il primo programma di **bootloader** che può caricare altri programmi **bootloading**.

Alla fine viene caricato il SO.