

ARCHITETTURA DEL CALCOLATORE (libro pag. 3,4)

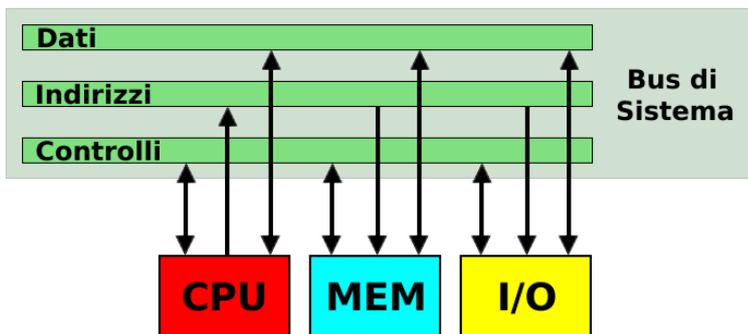
Il **computer** (elaboratore, calcolatore) è una macchina programmata in grado di svolgere le seguenti attività:

1. Input (acquisizione dati)
2. Storage (memorizzazione)
3. Processing (elaborazione)
4. Output (trasmissione dati)

Ogni computer è composto da due parti:

1. Hardware (componenti fisici)
2. Software (programmi)

L'**hardware** di un PC può essere schematizzato secondo il modello di Von Neumann.

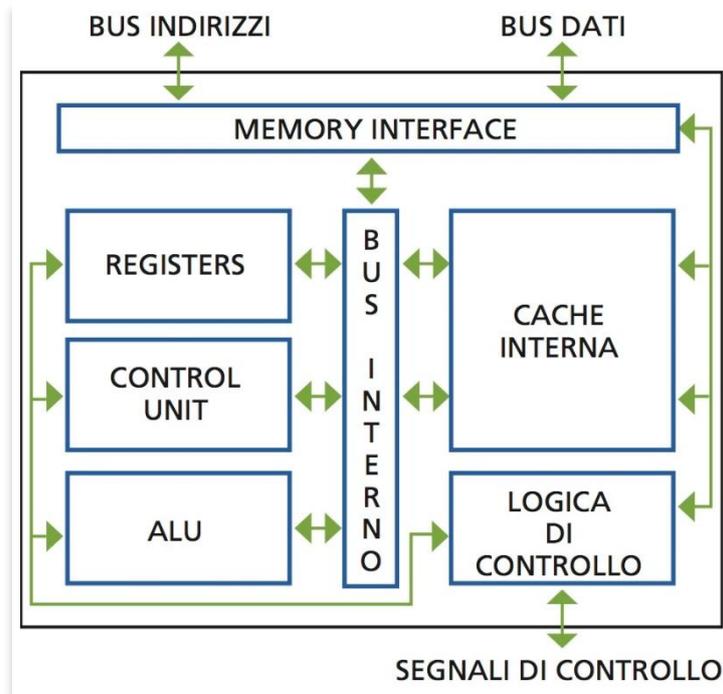


CPU (processore), memoria centrale (RAM) e interfaccia di input/output.

Tutti i componenti principali sono montati sulla **scheda madre** (motherboard).

La **CPU** (Central Processing Unit) è l'elemento dedicato all'esecuzione dei programmi. E' composta da 3 elementi:

1. **Control Unit (CU)**: Unità di controllo che decodifica le istruzioni e controlla il funzionamento del sistema.
2. **Arithmetic Logic Unit (ALU)**: Unità aritmetico logica che esegue le operazioni matematiche.
3. **Registers**: Registri di sistema, ossia piccole memorie ad alta velocità che contengono temporaneamente i dati e gli indirizzi necessari alle operazioni.



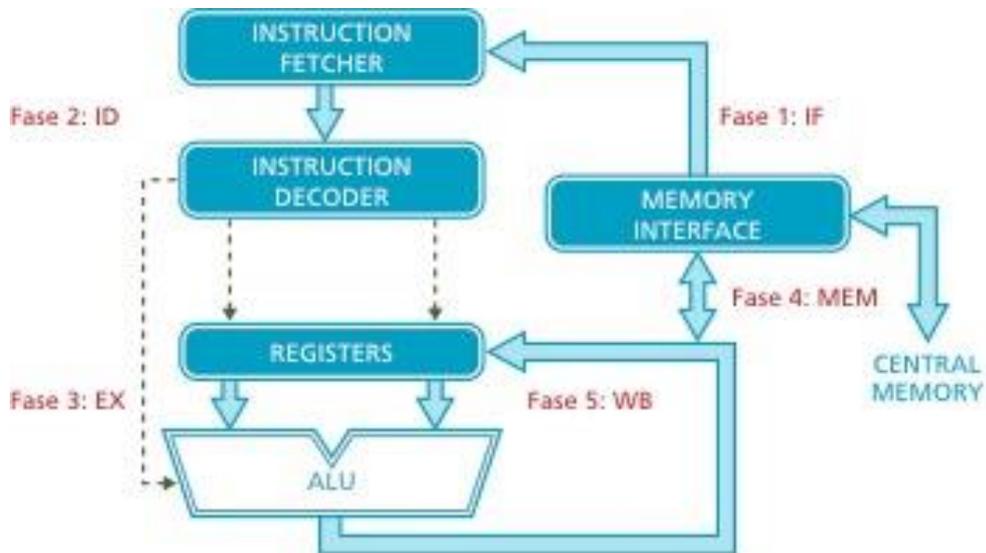
La CPU lavora ciclicamente su 5 fasi:

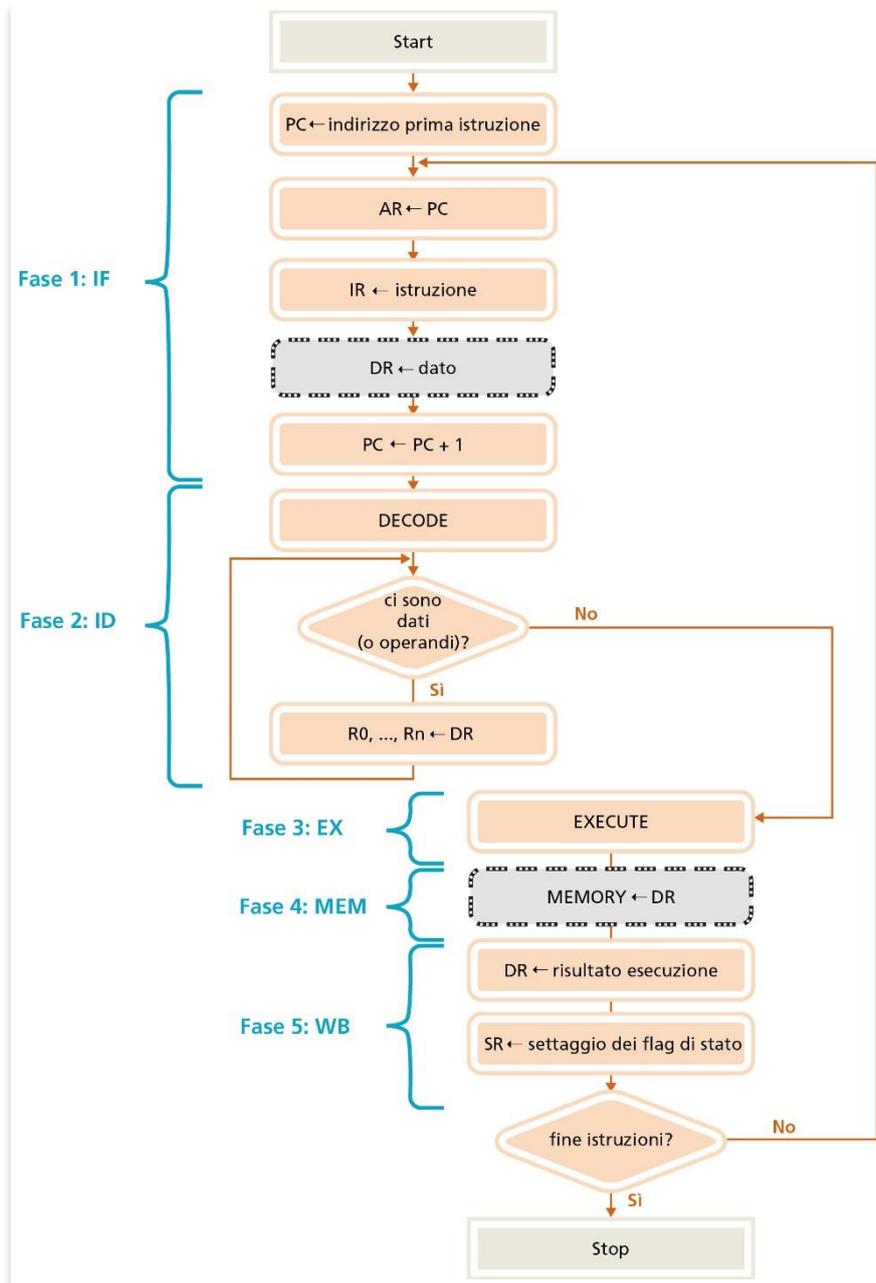
1. **Fase 1: Fetch** (Instruction fetch - IF)
 - a. La CU legge dal **Program Counter (PC)** l'indirizzo della prima istruzione e lo trascrive nell'**Address Register (AR)**.
 - b. La CU copia il contenuto di AR sul **BUS indirizzi** richiedendo una **lettura in memoria**.
 - c. La memoria invia l'istruzione sul **BUS istruzioni** e la CU la trascrive nel **Data Register (DR)** e successivamente nell'**Instruction Register (IR)**.
 - d. Il **Program Counter (PC)** viene incrementato al fine di contenere l'indirizzo della successiva istruzione da eseguire. L'indirizzo sarà quello subito successivo a meno di istruzioni di salto.
2. **Fase 2: Decode** (Instruction decode - ID)
 - a. La CU prende l'istruzione e la **divide** nelle sue due parti: **operazione** (opcode) e **operandi** (1, 2 o 3 indirizzi di dato a seconda delle architetture).
 - b. L'operazione viene **decodificata** ed in base ad essa si **caricano opportunamente i registri di lavoro** con gli operandi richiedendone la **lettura in memoria**.
 - c. La CU attiva le microcircuiterie necessarie a svolgere l'operazione.
3. **Fase 3: Execute** (EX)
 - a. **L'operazione viene eseguita**. Questa parte è differente per ciascuna operazione. Per i calcoli matematici si attiva l'**ALU**. Per le altre operazioni si attivano gli specifici componenti (scheda video, scheda audio, periferiche, ecc).
4. **Fase 4: Memory** (MEM)

- a. In caso l'istruzione preveda il **coinvolgimento delle memorie** effettua tali operazioni (salvataggio, spostamento dati, ecc), altrimenti non fa nulla.

5. Fase 5: Write Back (WB)

- a. Scrive gli eventuali risultati nei **registri dati (DR)**.
b. Imposta lo stato del processore nello **Status Register (SR)**.





Ogni operazione è sincronizzata dal **clock** che un segnale ad onda quadra. Ad ogni fronte di salita (impulso di clock) i componenti eseguono un'operazione.

Le caratteristiche principali che differenziano i processori sono:

1. **Velocità di elaborazione** = frequenza di clock = numero di impulsi al secondo. 3 GHz = 3 miliardi di impulsi al secondo.
2. **Parallelismo** = numero di cifre binarie (bit) di ciascun calcolo (4 - 64 bit) = precisione e velocità dei calcoli.
3. **Quantità di memoria cache** = memoria veloce interna al processore (alcuni kB - MB), suddivisa in 3 livelli (L1, L2, L3) a capacità crescente ma velocità decrescente.
4. **Numero di processori (core)** = numero di CPU che lavorano in parallelo (max 12).
5. **Set di istruzioni** = numero di istruzioni possibili (CISC - Complex Instruction Set Controller; RISC - Reduced). Istruzioni complesse necessitano di UC e firmware più complessi e lenti.
6. **Potenza assorbita.**

Gerarchia delle memorie

Capacità
Tempo di accesso

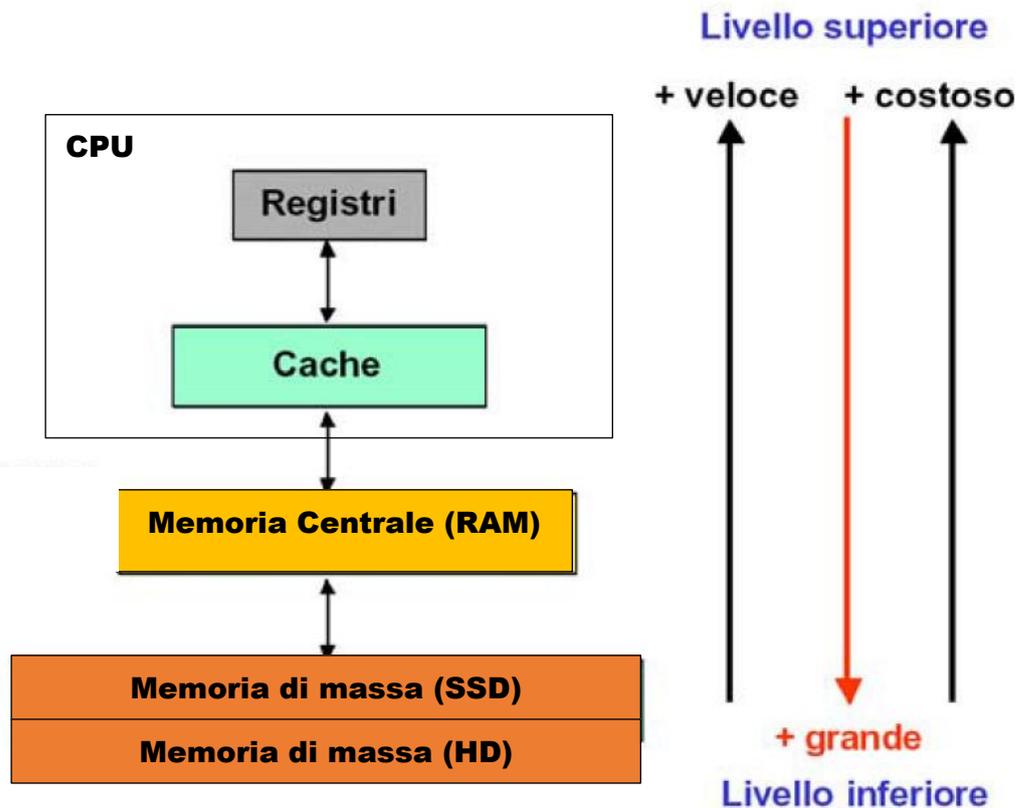
Registri CPU
< 1 kB
0.2 - 0.5 ns

Cache (S-RAM)
(Static RAM)
< 10 MB
5 - 10 ns

Mem. centr. (D-RAM)
(Dynamic RAM)
< 50 GB
20 - 70 ns

Solid State Drive
>500GB 100-300 ns

Hard Disk
>1TB 4-10ms



La **CACHE** è una memoria interna al microprocessore, gestita dalla CPU (non dal sistema operativo). E' di tipo **SRAM** (Static Random Access Memory): le celle di memoria sono costituite da FLIP-FLOP sono sempre alimentati. E' composta da 3 livelli (L1, L2, L3) a capacità crescente ma velocità decrescente. Nei sistemi multicore i primi due livelli sono privati mentre il terzo è condiviso dai core. Vi sono memorizzati temporaneamente i dati della RAM che si presume possano essere utilizzati a breve dalla CPU. Quando la CPU chiede un indirizzo di memoria RAM, il cache controller prima controlla che sia già nella cache, se c'è (**cache hit**) il trasferimento è molto veloce altrimenti (**cache miss**) si dovrà interrogare la RAM.

Cache	Tempo di accesso	Dimensioni
L1	1÷4 cicli di clock	16÷64 KB
L2	8÷15 cicli di clock	256÷1.024 KB
L3	25÷50 cicli di clock	2÷10 MB

La **RAM** (memoria centrale) è una memoria volatile che contiene i dati che servono alla CPU per l'elaborazione dei programmi. E' di tipo **DRAM** (Dynamic Random Access Memory): le celle di memoria sono costituite da condensatori non sempre alimentati ma con continuo refresh dei dati presenti. Le celle sono organizzate in una tabella e l'indirizzo di ciascun dato è composto dall'indirizzo di riga e di colonna espressi in esadecimale. La capacità di memoria è sempre una potenza di 2. 1kB = 2¹⁰ B = 1024 B (kibibyte) kiB.

DRAM DDR (Double Data Rate): RAM che permette un flusso di dati doppio per ogni impulso di clock. Oggi DDR3,4,5,ecc.

ROM (Read Only Memory): memoria sovrascrivibile solo con una procedura particolare (e potenzialmente rischiosa!) che contiene le informazioni fondamentali di una macchina senza le quali non potrebbe funzionare. Esempi di ROM:

- **BIOS** (Basic Input-Output System), piccola memoria con programma di avvio del computer residente sulla scheda madre.
- **UEFI** (Unified Extensible Firmware Interface) evoluzione del BIOS.
- **Firmware**, memoria che contiene il programma di funzionamento di apparecchi dedicati (es. modem, elettrodomestici, navigatore, sistema di allarme, ecc).

Altre memorie: Flash, CD, DVD, Blue ray, ecc

Memoria virtuale: se la RAM non è sufficiente l'hard disk può essere usato come una RAM virtuale. I tempi di accesso sono lunghissimi ma almeno i programmi non si bloccano per mancanza di memoria.

BUS DI SISTEMA (pag. 7-9)

BUS deriva dal latino "omnibus" (tutti) e indica un sistema di comunicazione a cui sono collegati vari dispositivi. Si contrappone alla comunicazione punto-punto.

Il bus di sistema permette il collegamento dati tra CPU, RAM e interfacce I/O (scheda video, HD, scheda audio, scheda di rete, Bios, porte di comunicazione, ecc). E' costituito da un centinaio di fili stampati direttamente sulla scheda madre e si suddivide in:

1. BUS dati: trasporta i dati di input e di output (32 o 64 bit bidirezionale);
2. BUS indirizzi: trasporta gli indirizzi di memoria da leggere (32 o 64 bit monodirezionale da CPU a RAM);
3. BUS di controllo: trasporta i segnali di stato ed i comandi (variabile).

Ottimizzazioni

1. Direct Memory Access (DMA)

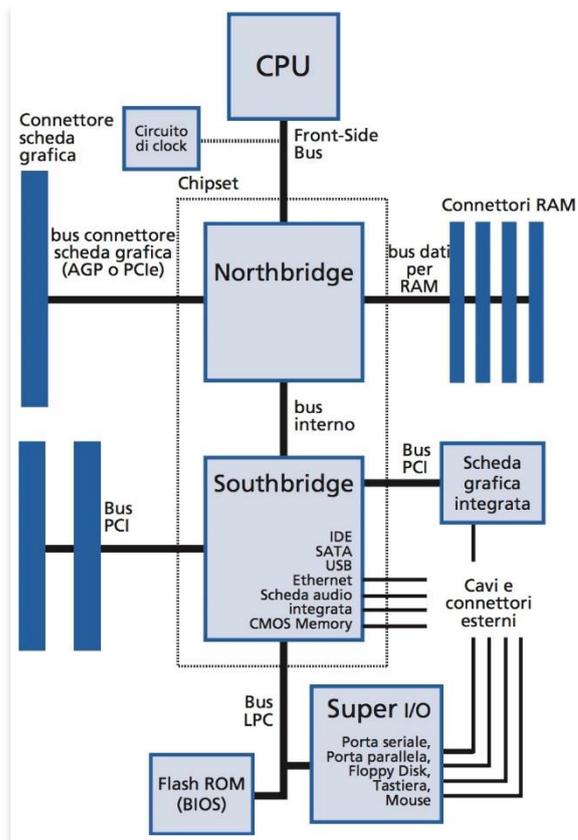
Circuito che permette di trasferire dati tra due memorie (es. RAM e HD) senza usare la CPU.

2. Suddivisione del BUS

Il BUS di sistema è stato diviso in diversi BUS dedicati a dispositivi con differenti velocità. Tra i vari BUS c'è un circuito chiamato **chipset** che gestisce le comunicazioni. Il chipset si compone di due parti:

- a. **Northbridge** (Memory Controller Hub) collega il BUS della CPU (Front Side Bus FSB) con quello della RAM e della scheda grafica.
- b. **Southbridge** (I/O Controller Hub) collega i BUS di tutte le altre interfacce I/O (HD, scheda audio, scheda di rete, Bios, porte di comunicazione, ecc).

Negli Intel i3-5-7 il Northbridge è integrato direttamente dentro la CPU.



PERIFERICHE E STANDARD DI COMUNICAZIONE (pag. 22-31)

1. Input: mouse, tastiera, scanner, ecc.
2. Output: scheda video e monitor, stampante.
3. Input/Output: scheda audio, modem, scheda di rete.