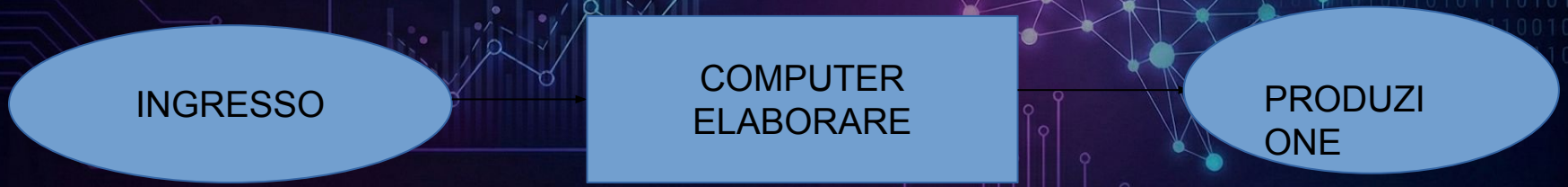
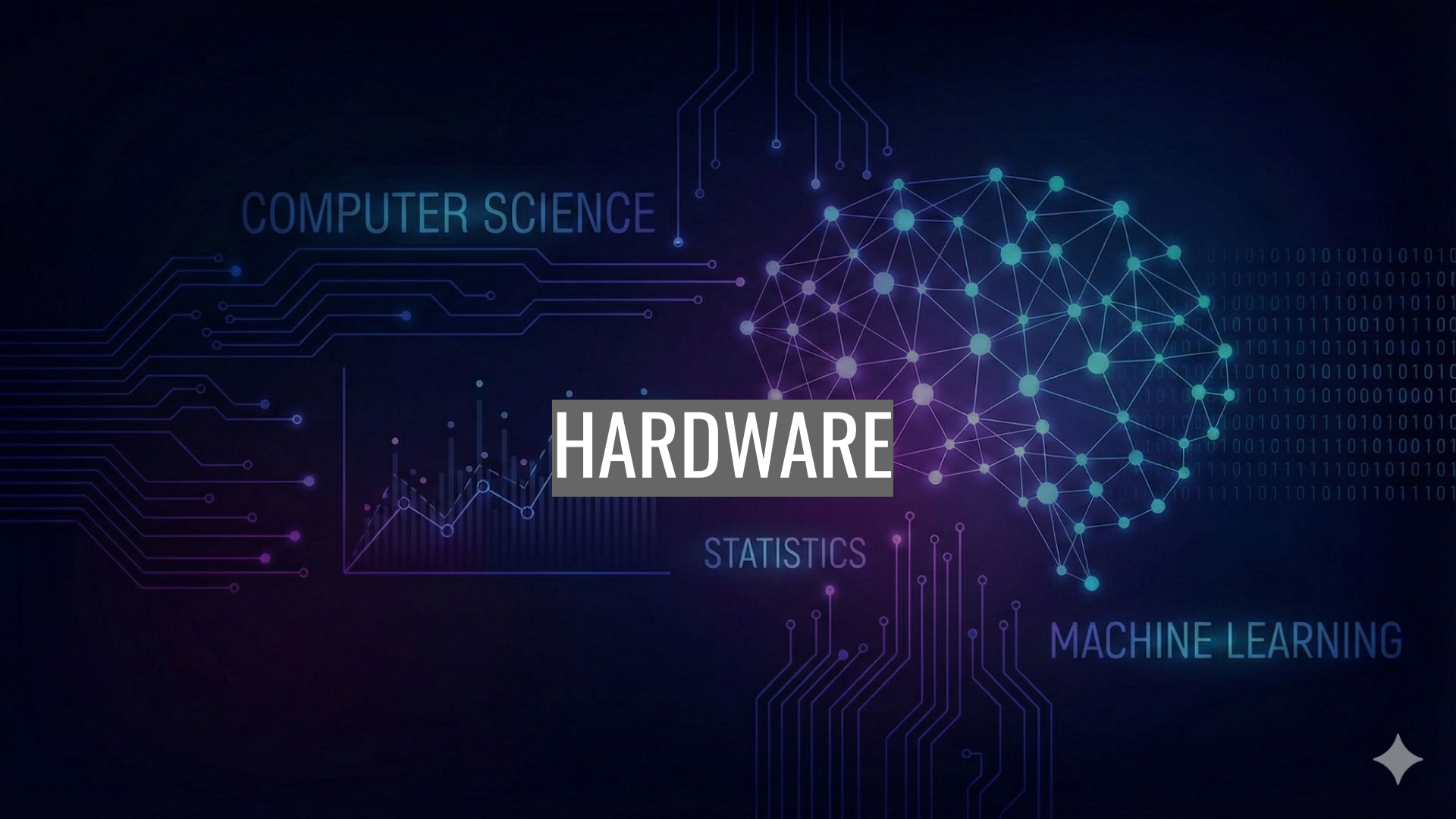
Cos'è un computer e come è fatto

Prof. Loriano Storchi
loriano@storchi.org
https://www.storchi.org/

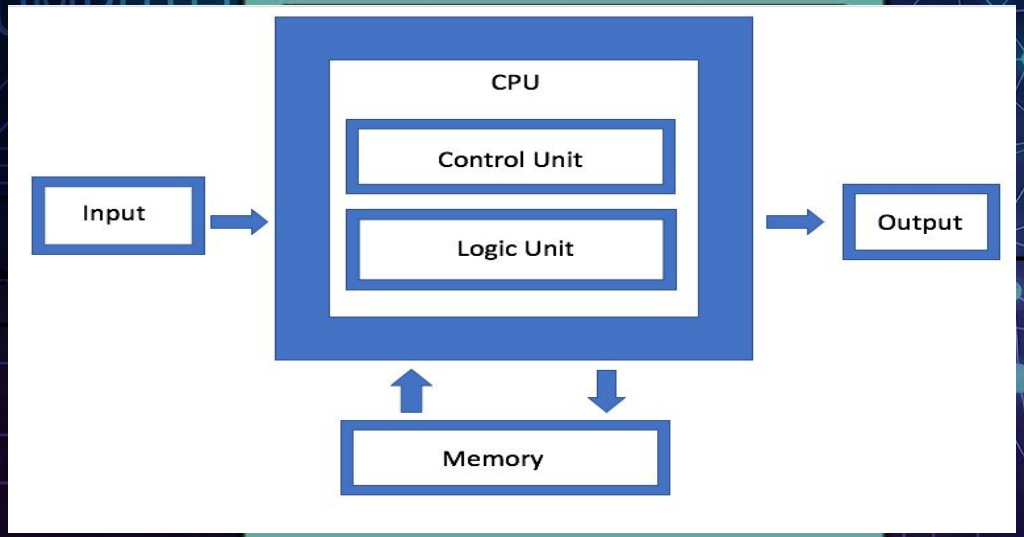
Informatica

Possiamo utilizzare diverse definizioni, come ad esempio: l'informatica è la scienza della rappresentazione e dell'elaborazione delle informazioni, parafrasando lo studio degli algoritmi che descrivono e trasformano le informazioni.




COMPUTER SCIENCE
HARDWARE
STATISTICS
MACHINE LEARNING

Macchina di Von Neumann (Zuse)



BUS DI SISTEMA, connessione (architettura Harvard che separa la memoria dati dalla memoria di programma)

Macchina di Von Neumann (Zuse)

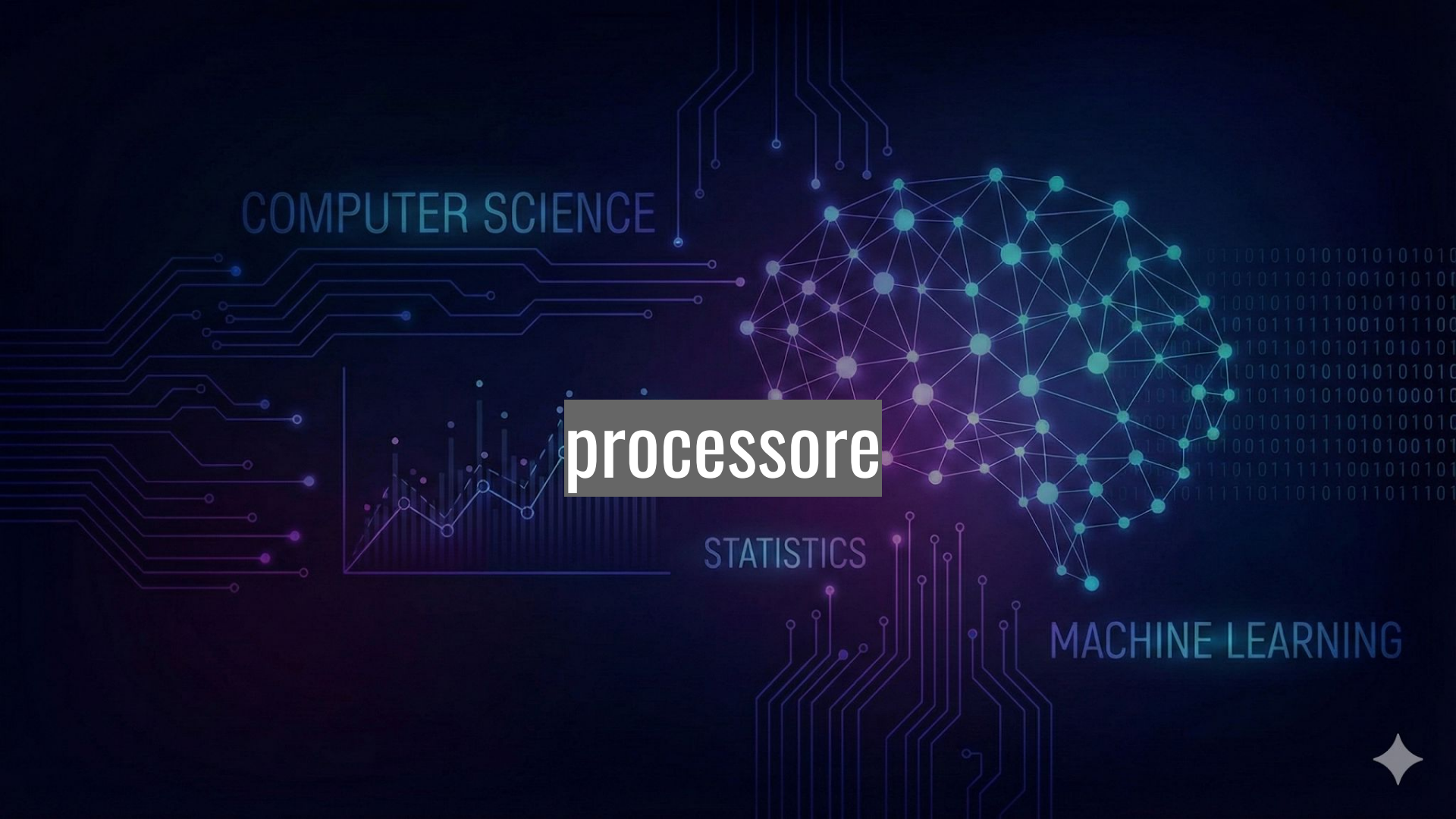
L'architettura descritta da Von Neumann (Zuse) è quindi composta da:

- La CPU è quindi un'unità di elaborazione centrale
- Un dispositivo di memoria utilizzato per memorizzare dati che possono poi essere identificati tramite il loro indirizzo.
- I vari dispositivi di input/output (I/O) utilizzati per interagire con l'utente esterno o altri sistemi
- Una linea di interconnessione che collega i vari sottosistemi, il bus

Macchina di Von Neumann (Zuse)

Un computer costruito in questo modo è una macchina estremamente flessibile. L'hardware fornisce le funzionalità di base, mentre il software specializza la macchina per eseguire compiti specifici.

Di seguito presenteremo i vari componenti di base della calcolatrice da una prospettiva "moderna".


COMPUTER SCIENCE
processore
STATISTICS
MACHINE LEARNING

processore

La CPU (Unità Centrale di Elaborazione) coordina e gestisce tutti i vari dispositivi hardware per acquisire, interpretare ed eseguire le istruzioni del programma.

Oggi è costituito da un singolo chip e, come qualsiasi altro chip, **comunica con il mondo esterno tramite pin** . Utilizzando questi pin, riceve segnali e invia segnali elettrici (informazioni costituite da sequenze di bit).

CPU - CU

L'unità di controllo (CU) è un componente fondamentale della CPU che dirige il funzionamento del processore.

- Pensatela come il direttore d'orchestra. Non suona gli strumenti (l'ALU si occupa dei calcoli), ma dice a tutti gli altri componenti quando suonare e cosa suonare.
- Gestisce il flusso di dati tra la CPU, la memoria e i dispositivi di input/output.
- Senza la CU, la potente ALU e i registri veloci rimarrebbero inattivi e silenziosi.

CPU - ALU

L'ALU (Unità Aritmetico-Logica) esegue operazioni logiche e aritmetiche.

- Input: Operandi (A e B) e un Opcode (codice operativo).
- Output: Risultato (Y) e indicatori di stato.
- Ruolo: Funge da "calcolatrice" della CPU. Senza di essa, il computer non può elaborare i dati, ma solo trasferirli.

Nei diagrammi architettonici viene spesso rappresentato con un simbolo a forma di "V".

CPU - ALU

Operazioni principali

ARITHMETIC

These operations handle numerical calculations. The complexity depends on the ALU design.

- **ADD / SUB:** Basic integer addition and subtraction.
- **INC / DEC:** Increment or decrement by 1.
- **MUL / DIV:** Often handled by specialized hardware in modern CPUs, but basic ALUs may use repeated addition/subtraction.

LOGIC

These operations treat data as individual bits rather than numbers, essential for decision making.

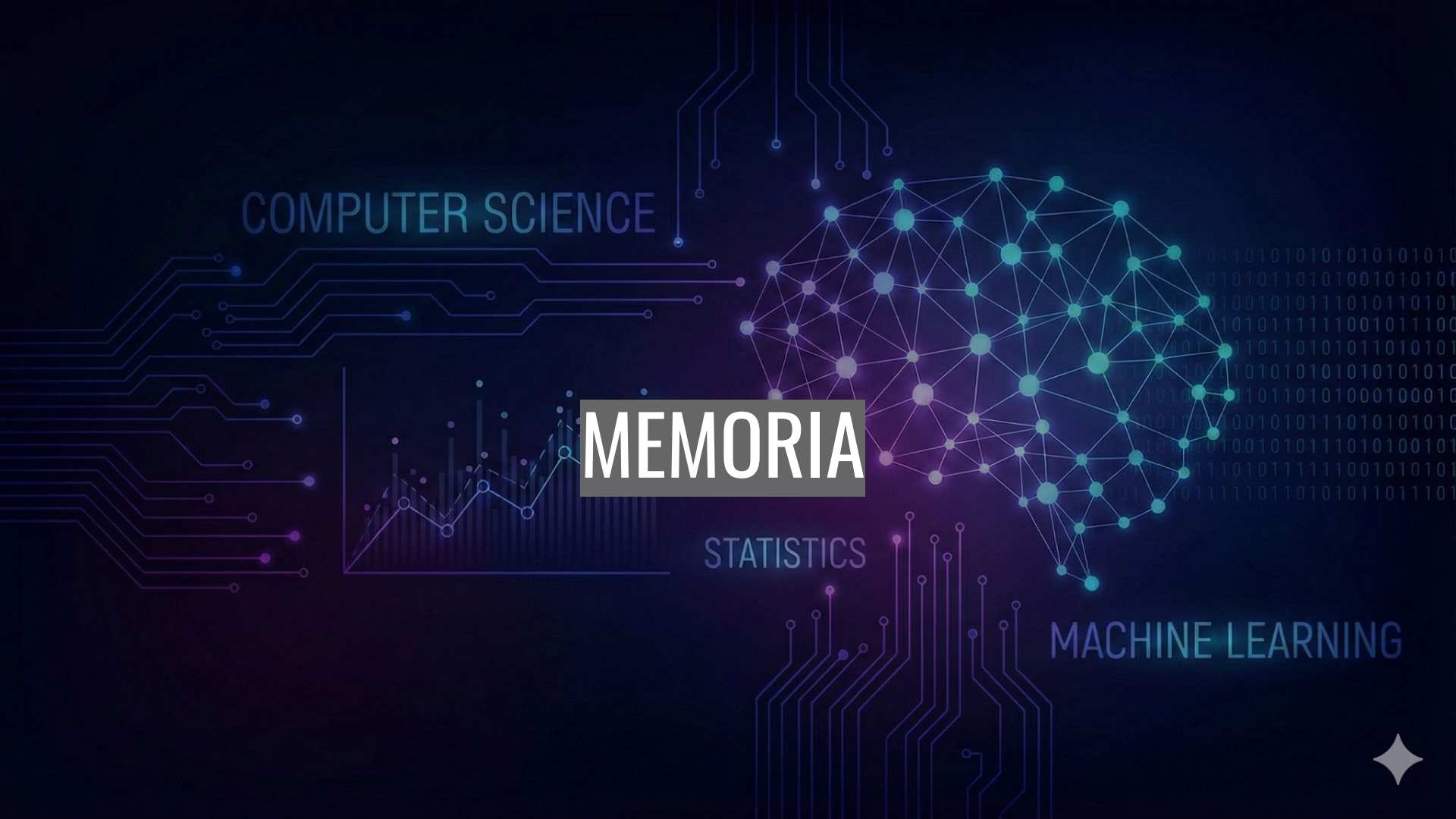
- **AND:** Used for masking (clearing specific bits).
- **OR:** Used for setting specific bits.
- **XOR:** Used for toggling bits or comparing values.
- **NOT:** Inverts all bits (1s Complement).

CPU - Registri

- **Registri** , in pratica memoria interna della CPU che consente l'accesso ai dati in modo molto più rapido
- Registri: Tutte le CPU hanno sempre almeno due registri:
  - **IP (Instruction Pointer o Program Counter PC)** che contiene il puntatore alla prossima istruzione da eseguire.
  - **Registro dei flag** . Questo registro è essenzialmente una serie di bit che rappresentano un particolare stato della CPU; ad **esempio, il flag di overflow** viene impostato a 1 nel caso in cui il risultato dell'operazione appena eseguita sia troppo grande per il campo del risultato.

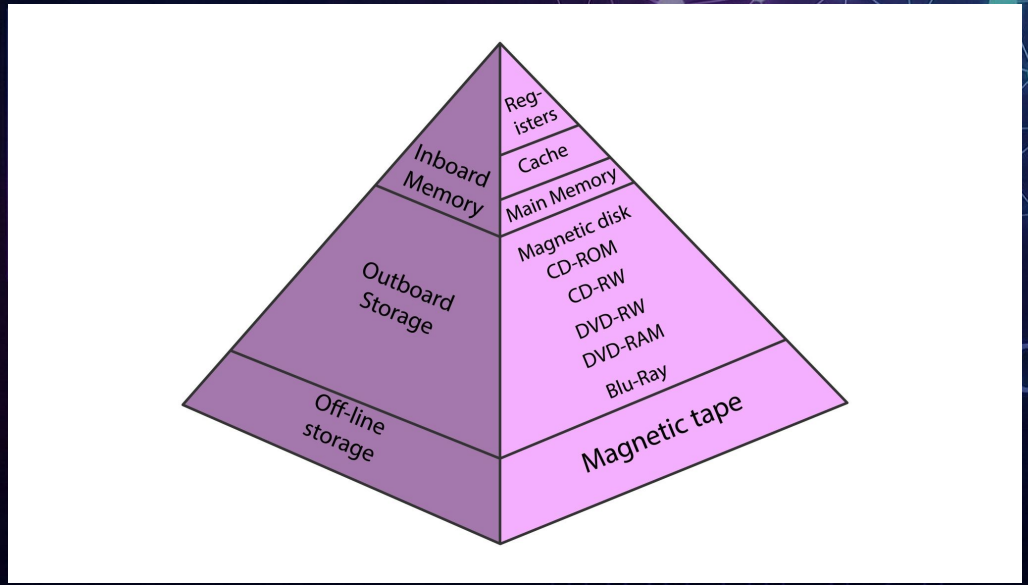
CPU - OROLOGIO

- **CLOCK** : indica gli intervalli di tempo in cui operano i dispositivi interni della CPU. Determina la loro **velocità, espressa come numero di intervalli per unità di tempo .**
- **Lo stato della CPU cambia ogni volta che viene inviato un impulso.** Pertanto, il tempo di esecuzione di una determinata operazione si misura in numero di cicli di clock.
- Una parte importante della CPU è costituita dalla serie di "circuiti" che servono a propagare questo impulso tra tutti i componenti della CPU.
- Nessuna CPU può funzionare più velocemente del tempo impiegato dal segnale di clock per percorrere il tragitto più lungo di questo "circuito" di distribuzione del segnale, ovvero **il percorso critico.**


COMPUTER SCIENCE
MEMORIA
STATISTICS
MACHINE LEARNING

gerarchia della memoria

La gerarchia di memoria nei processori attuali è composta da diversi livelli, ognuno dei quali è caratterizzato da velocità di accesso ai dati inversamente proporzionali alle sue dimensioni: più grandi sono queste aree, più tempo occorre per recuperare i dati in esse contenuti.



Registri (Le "mani" della CPU)

- Posizione: All'interno della CPU stessa.
- Velocità: La più veloce in assoluto (accesso immediato).
- Capacità: Estremamente ridotta (contiene solo pochi dati, come i numeri specifici che vengono aggiunti al momento).
- Ruolo: Di solito non vengono considerate "memoria" nel senso di archiviazione; sono le parti operative effettive del processore.

Cache

Cache: Generalmente, vengono installati da uno a tre livelli di memoria cache . La memoria cache ha tempi di accesso e dimensioni differenti, a seconda che sia integrata nel chip o esternamente, e in base alla tecnologia utilizzata per costruire le celle di memoria. (Eseguire il comando `cat /proc/cpuinfo` per visualizzare la dimensione della cache)

L'utilizzo e i vantaggi di una cache si basano sul principio di località, ovvero un programma tende a riutilizzare i dati e le istruzioni utilizzati di recente. Di conseguenza, esiste una regola empirica secondo cui il 90% del tempo totale viene impiegato per eseguire il 10% delle istruzioni. Più precisamente, l'utilizzo di una cache è vantaggioso quando un codice sfrutta sia la località spaziale che quella temporale dei dati.

Cache

Località temporale

"Se l'hai usato di recente, probabilmente lo userai di nuovo presto."

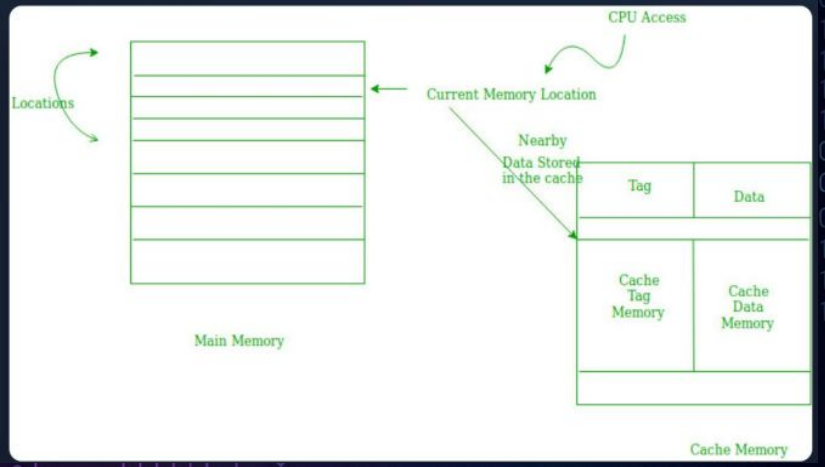
Impatto: La cache mantiene i dati a cui si è acceduto di recente (variabili, istruzioni) pronti per un rapido riutilizzo.

Esempio: una variabile contatore all'interno di un ciclo for.

Località spaziale

"Se usi questo indirizzo, probabilmente userai anche quelli vicini."

Impatto: Durante il recupero dei dati, la cache carica l'intero "blocco" o "riga" di memoria circostante.



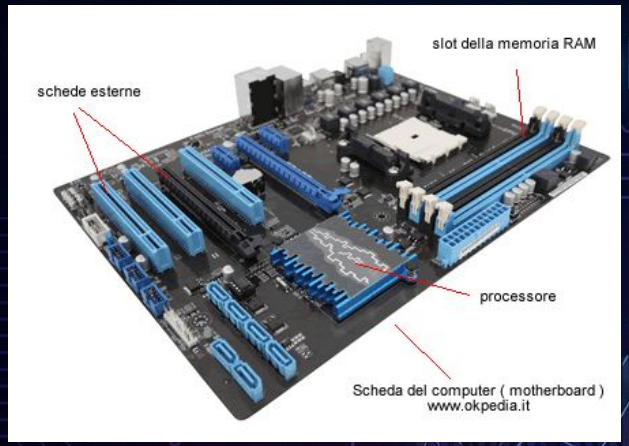
Memoria centrale

- **Si tratta della memoria che si interfaccia direttamente con la CPU** , ad esempio, è collegata direttamente alla scheda madre del computer. Ciò consente un flusso continuo di dati da e verso la CPU.
- Questo tipo di memoria è caratterizzato da una velocità di accesso ai dati estremamente elevata.
- La memoria principale (chiamata anche ***memoria primaria***) può essere di sola lettura come **la ROM** o di lettura/scrittura come **la RAM**

Memoria centrale - RAM

- **La RAM (Random Access Memory)** è suddivisa in celle di memoria, e **a ciascuna cella viene assegnato un indirizzo** utilizzato per leggere e scrivere i dati in essa contenuti.
- La RAM **contiene anche le istruzioni (codici operativi)** che verranno eseguite e i dati su cui queste istruzioni agiranno.
- Caratteristiche:
  - **Volatile** : il contenuto viene perso quando il computer viene spento.
  - **Veloce** : circa 100 cicli di clock. Veloce, quindi costoso.
  - **Dimensioni ridotte** rispetto alla memoria di massa, dell'ordine di pochi GiB

Memoria centrale - RAM

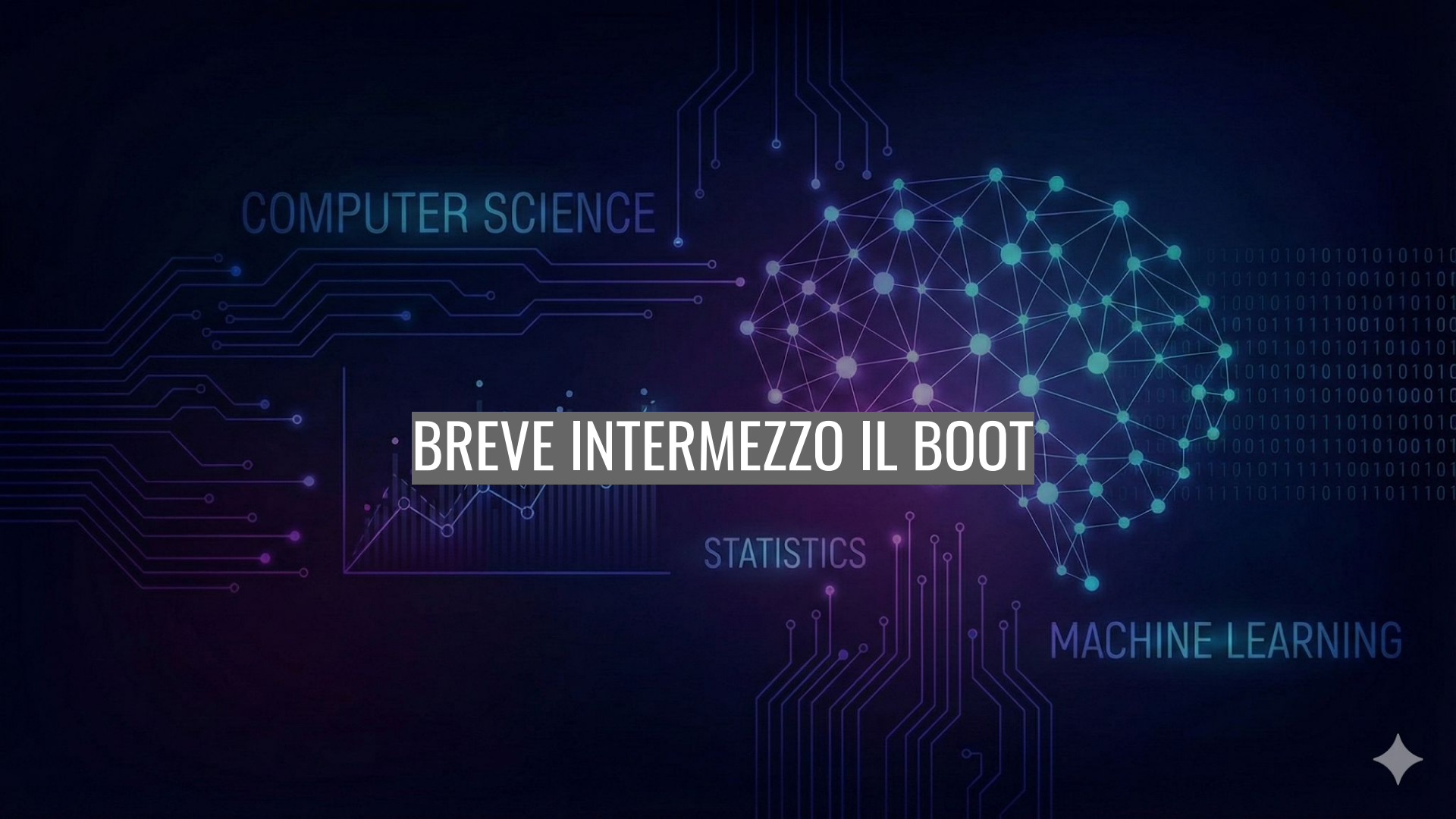


Esistono diverse tecnologie RAM, la più comunemente utilizzata al giorno d'oggi è **la DRAM (Dynamic RAM).**
In pratica, ogni bit viene memorizzato in un condensatore (**il bit 1 o 0 dipende dalla carica del condensatore**). Ogni condensatore deve essere ricaricato periodicamente, altrimenti si verificherebbe una perdita di carica e quindi di informazioni. Esistono quindi diverse varianti tecnologicamente differenti di DRAM.

La SRAM (Static RAM) è una tecnologia di memoria che non richiede un aggiornamento continuo, ma può conservare le informazioni per periodi di tempo molto lunghi. Offre **un basso consumo energetico e tempi di accesso brevi, ma è costosa da produrre. Viene generalmente utilizzata come memoria cache.**

Memoria centrale - ROM

- **ROM (Read Only Memory)** : memoria di sola lettura con una velocità di accesso superiore rispetto alla memoria di massa.
- **Consente di memorizzare i dati in modo permanente** . Questo tipo di memoria viene utilizzato per memorizzare i dati e il codice necessari per la procedura di avvio del computer e altri programmi/procedure (**firmware**).
- **BIOS (Basic Input/Output System)** , nei sistemi moderni sostituito da **UEFI (Unified Extensible Firmware Interface).**
- **EPROM, acronimo di Erasable Programmable Read Only Memory** (memoria di sola lettura programmabile e cancellabile). Può essere cancellata e riscritta un numero di volte generalmente limitato.



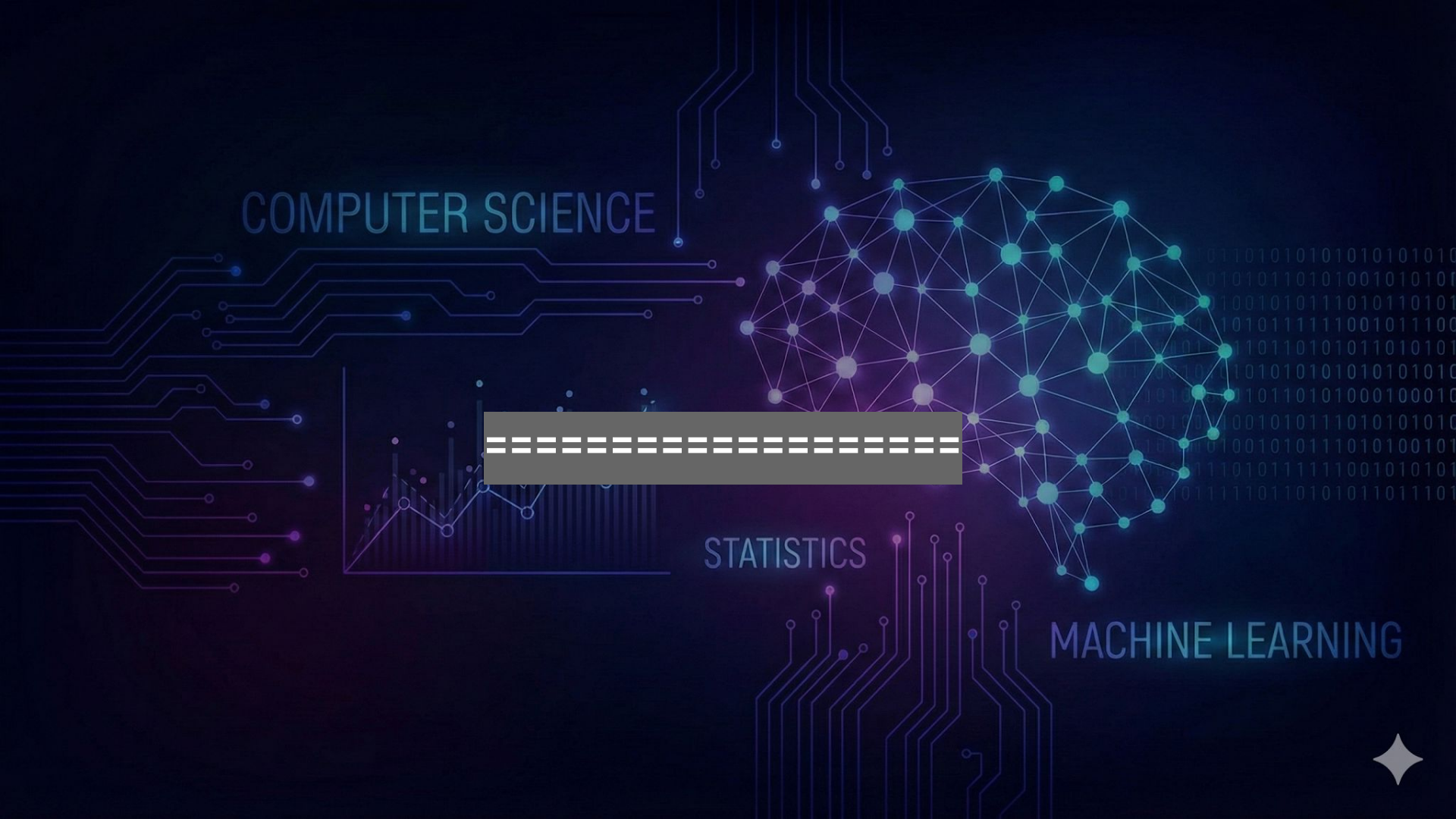
BREVE INTERMEZZO IL BOOT

Processo di avvio

- **Accensione** : Ovviamente la prima fase del processo è l'accensione, che di solito viene avviata dall'utente.
  - **"Wake on LAN"** , che semplifica l'accensione tramite rete, senza richiedere la presenza fisica o l'intervento dell'utente.
  - Non appena la CPU viene alimentata, esegue il codice contenuto nella **ROM** (memoria di sola lettura) presente sulla scheda madre.

Processo di avvio

- **POST** : Il sistema esegue quindi una procedura chiamata POST (**Power-On Self Test**), che garantisce che tutto l'hardware sia operativo e pronto all'uso. Ciò include il controllo **della memoria e dei dischi rigidi e...** Una volta completato il POST, il sistema cerca il primo dispositivo nell'elenco dell'ordine di avvio.
- A questo punto, dopo **il caricamento del BIOS o dell'UEFI**, il sistema cerca un dispositivo attivo nell'elenco dei dispositivi di avvio. Quando trova un dispositivo disponibile, il **BIOS/UEFI** fornisce informazioni sulle comunicazioni di base con le periferiche e sulle comunicazioni con la scheda madre stessa.


COMPUTER SCIENCE
==========
STATISTICS
MACHINE LEARNING

Archiviazione di massa – Archiviazione secondaria

- **A differenza della memoria principale, non è direttamente accessibile dalla CPU, ma la CPU comunica con un controller (bus I/O).**
- I dati vengono trasferiti dalla memoria secondaria a un'area della memoria principale e letti direttamente da lì dalla CPU.
- Fisicamente, questi dispositivi sono collegati alla scheda madre tramite un cavo ad alta velocità o tramite cavi connessi a interfacce esterne.

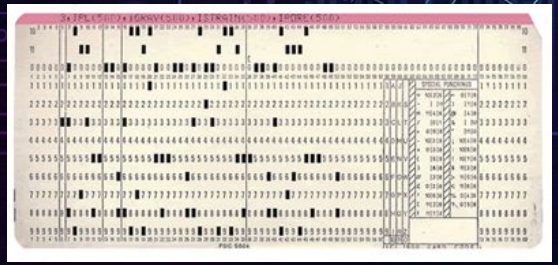
Possono essere realizzati con tecnologia magnetica, ottica o a stato solido.

Archiviazione di massa

- Composto da dischi rigidi, CD, DVD, unità a stato solido, nastri
  - **Non volatile** , quindi i dati rimangono memorizzati anche dopo il riavvio del computer
  - **Lento rispetto alla RAM** (> 10000 cicli), ovviamente molto variabile a seconda del tipo di supporto
  - **In media più economico** , o almeno più economico della DRAM o della SRAM.
  - **Di grandi dimensioni** (oggi centinaia di GiB o qualche TiB)
  - **Possono essere di sola lettura o di lettura/scrittura** (si pensi agli hard disk e ai DVD-ROM).

Archiviazione di massa: gli alberi

- Le schede perforate rappresentano la prima memoria secondaria nella storia dei computer. I dati vengono memorizzati su schede di cartone e registrati perforandole, per poi essere letti dal computer.
- Nastri perforati simili alle schede perforate per tipo



Archiviazione di massa – Nastro magnetico

- In questo caso i dati vengono memorizzati su un nastro magnetico. Questi nastri erano storicamente utilizzati nei mainframe (computer di grandi dimensioni in grado di eseguire elaborazioni molto rapide e memorizzare grandi quantità di dati) negli anni '70 e '80, ma **sono ancora oggi impiegati per il backup dei dati.**

Archiviazione di massa – Disco rigido

- **Composto da diverse piastre magnetiche** posizionate una sopra l'altra e rotanti
- I dati vengono memorizzati su entrambi i lati dei singoli dischi e sono organizzati in **tracce e settori.**
- Le **testine** vengono utilizzate per leggere e scrivere dati.
- La velocità di rotazione dei piatti (**rpm**) è uno dei fattori determinanti per la velocità di lettura e scrittura.
- **Le unità a stato solido SSD offrono** latenze inferiori e sono meno soggette a guasti e urti meccanici.

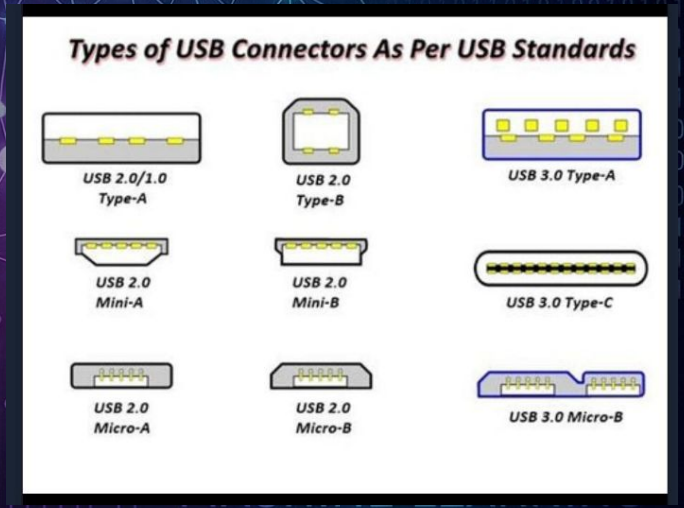
Archiviazione di massa – Rimovibile

- Può essere rimosso e separato dal computer.
- **Nei CD e nei DVD** , sia di sola lettura che di lettura/scrittura, i dati vengono memorizzati e letti utilizzando un raggio laser (l'idea di base è la semplice riflessione o l'assenza di riflessione per identificare i bit come 1 o 0).
- **chiavetta USB**
- **I Floppy disk** non sono più utilizzati

USB

GENERATION	PORT COLOR	MAX SPEED	MAX POWER
USB 2.0 High Speed	● Black / White	480 Mbps	2.5 W (500mA)
USB 3.0 SuperSpeed	● Blue	5 Gbps	4.5 W (900mA)
USB 3.1 / Type-C SuperSpeed+ / PD	● Teal / Any	10 Gbps+	100 W (Power Delivery)

Backwards Compatibility: USB 3.0 ports (Blue) can accept USB 2.0 devices, but speeds will drop to 480 Mbps.



UNITÀ DI MISURA

Unità di misura dell'informazione

- **BIT** = è l'unità di misura dell'informazione (dall'inglese "binary digit"), definita come **la quantità minima di informazione necessaria per distinguere tra due eventi ugualmente probabili** . (Wikipedia)
- **BYTE** = 8 BIT (storicamente, i caratteri erano rappresentati da 8 BIT, motivo per cui **1 Byte rimane l'unità di memoria indirizzabile minima ancora oggi**)
- “KiloByte” meglio “kibibyte” KiB = 2^{10} Byte = 1024 Byte
- “MegaByte” meglio “mebibyte” MiB = 1024 * 1024 Byte
- “GigaByte” meglio “gibibyte” GiB = 1024 * 1024 * 1024 Byte
- “Terabyte” meglio “tebibyte” TiB = 1024 * 1024 * 1024 * 1024 Byte

Prestazioni del computer

- Chiaramente, aumentare le prestazioni di un computer significa ridurre il tempo necessario per eseguire un'operazione.
- T_{clock} è il periodo di clock della macchina (**incremento di frequenza**)
- CPI_i è invece il numero di "colpi" di clock necessari per eseguire la data istruzione i (**riduzione della complessità per la singola istruzione**)
- N_i è infine il numero di istruzioni di tipo i (ad esempio somme, salti, ecc.).

$$T_{esecuzione} = T_{clock} \sum_{i=0}^{n} N_i CPI_i$$

MIPS

MIPS è l'acronimo di **Mega Instructions Per Second (Mega Istruzioni al Secondo)** e indica il numero di istruzioni generiche che una CPU esegue in un secondo. Si tratta di un'unità di misura più generale, utilizzata per misurare le prestazioni di un computer, rispetto ai FLOPS, che vedremo a breve.

$$\text{MIPS} = \frac{f_{\text{clock}}}{\text{CPI} \times 10^6}$$

Questo tipo di misurazione non tiene conto, ad esempio, delle ottimizzazioni dovute alla presenza della cache e delle percentuali delle diverse istruzioni all'interno di programmi reali, e non solo.

FLOP

FLOPS è l'acronimo di **Floating Point Operations Per Second (Operazioni** in virgola mobile al secondo) e indica il numero di operazioni in virgola mobile eseguite da una CPU in un secondo. È un'unità di misura utilizzata per valutare le prestazioni di un computer, in particolare nel calcolo scientifico.

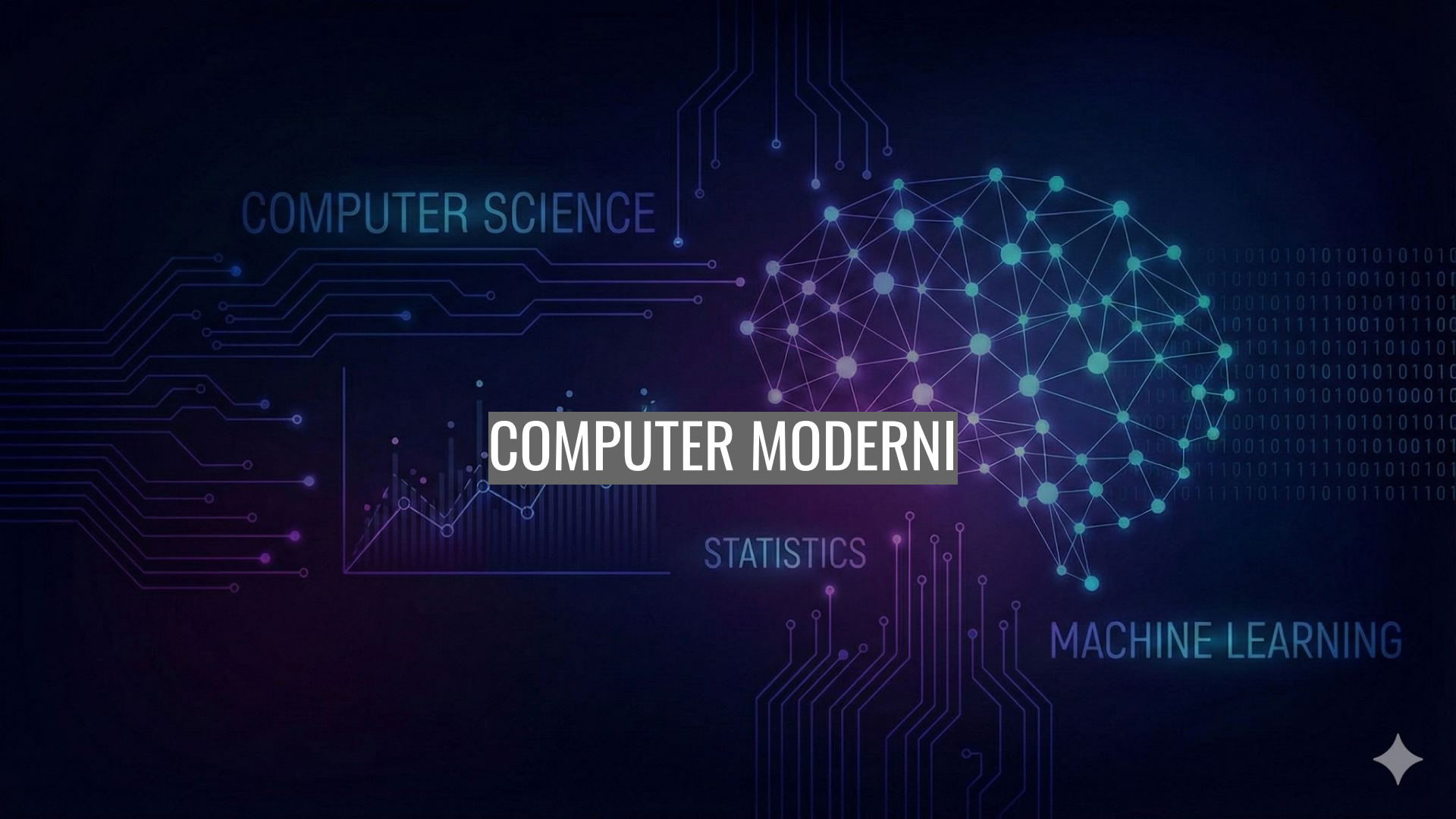
Ad esempio, nel caso di un classico prodotto di matrici, vengono eseguite $2*N^3$ operazioni , quindi posso valutare i FLOPS esattamente misurando il tempo necessario per eseguire questa moltiplicazione e ottenere:

$$[flops] = 2*N^3 / tempo$$

SPEC

La Standard Performance Evaluation Corporation è un'organizzazione senza scopo di lucro che produce e mantiene un insieme standardizzato di benchmark per computer (ovvero, un insieme di programmi di test rappresentativi di applicazioni informatiche reali).

Esistono diversi set di specifiche tecniche (SPEC) che sono specifici, ad esempio, per diversi usi previsti del computer.


COMPUTER SCIENCE
COMPUTER MODERNI
STATISTICS
MACHINE LEARNING

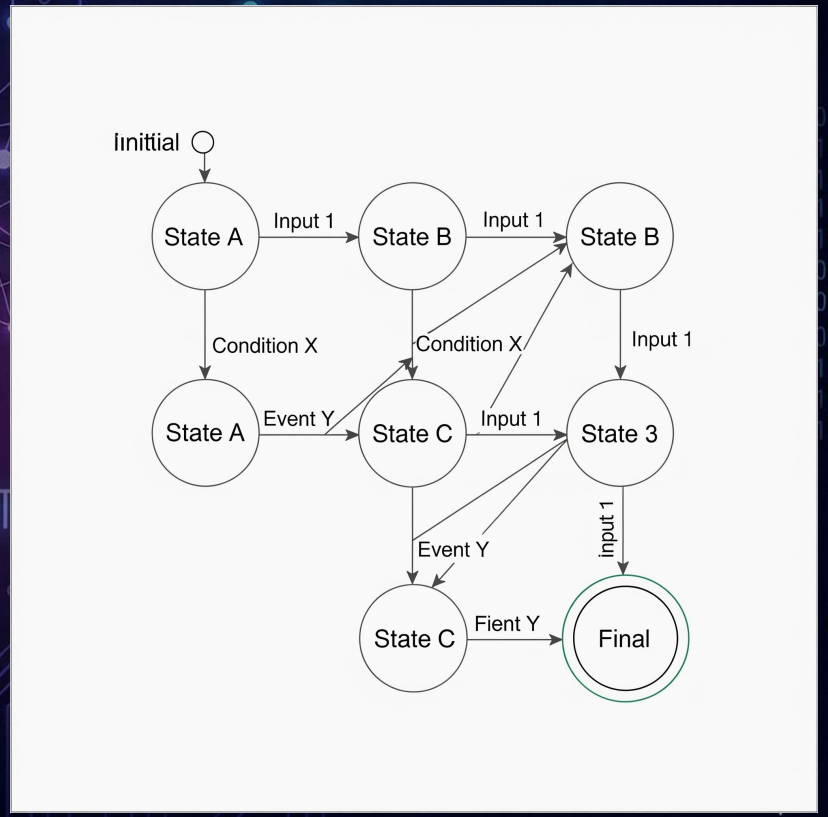
FLOP

Prefix	Scale	Notation	Typical Device
GigaFLOPS	Billion	10^{9}	Standard Laptop
TeraFLOPS	Trillion	10^{12}	PS5 / Gaming PC
PetaFLOPS	Quadrillion	10^{15}	Supercomputers (2010s)
ExaFLOPS	Quintillion	10^{18}	Frontier (Current)
ZettaFLOPS	Sextillion	10^{21}	Future AI Clusters

Macchina a stati finiti

Lo specialista della rigidità

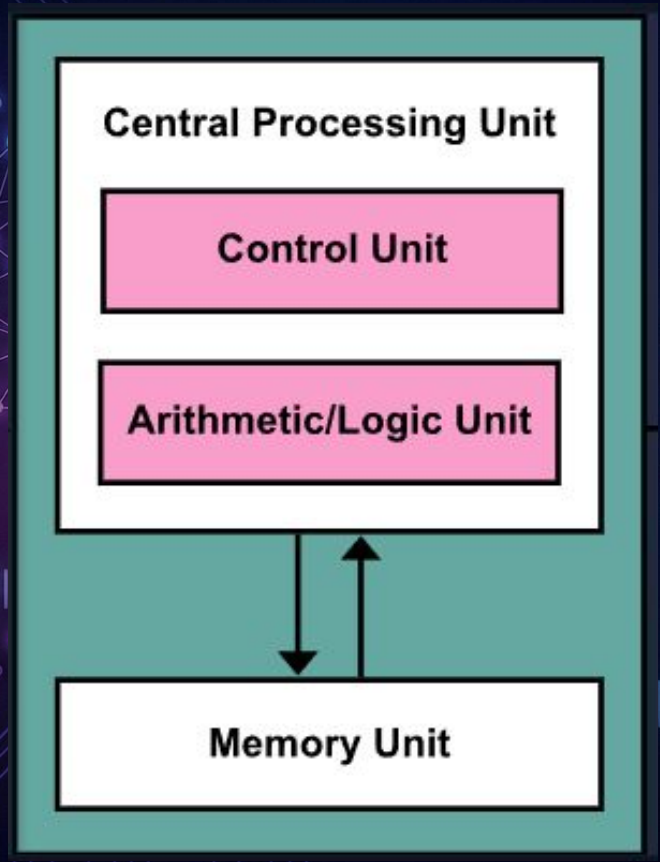
- Un FSM è un modello computazionale che può trovarsi in un solo stato tra un numero finito di stati in un dato momento.
- Logica: Le transizioni tra gli stati sono innescate da input specifici (condizioni).
- Archiviazione: Non memorizza un "programma" in memoria. La logica è solitamente cablata o fissa nella struttura del circuito.
- **Esempio: un distributore automatico. Attende le monete (Input), passa allo stato "Credito" (Stato) ed eroga una bibita (Azione). Non può "imparare" a giocare a scacchi.**



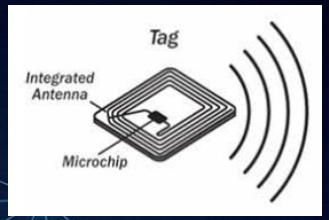
Macchina di Von Neumann

Il generalista flessibile

- L'architettura di Von Neumann descrive un computer in cui le istruzioni del programma e i dati sono memorizzati nella stessa memoria.
- Programma memorizzato: la macchina può modificare il proprio comportamento caricando un programma software diverso nella memoria.
- Ciclo: Utilizza un'unità di elaborazione centrale (CPU) per prelevare, decodificare ed eseguire istruzioni in modo continuo.
- **Flessibilità: lo stesso hardware può calcolare le tasse, far girare i videogiochi o navigare sul web.**



RFID



- **L'identificazione a radiofrequenza RFID** costa pochi centesimi e può raggiungere **alcuni MIPS**
  - La maggior parte dei tag RFID standard sono **macchine a stati finiti** (**FSM**), non macchine di Von Neumann. Tuttavia, i tag RFID "intelligenti" più avanzati (come quelli presenti nei passaporti o nelle carte di credito) possono essere considerati veri e propri computer che seguono l'architettura di **Von Neumann** (o di Harvard).
- Utilizzano campi elettromagnetici per identificare e tracciare automaticamente gli oggetti tramite un ID o altre informazioni.
- **Passivi** se utilizzano l'energia emessa dal lettore RFID. Anche alimentati a batteria.
- **Alimentati** a batteria, inviano periodicamente un segnale

Sistemi embedded e smartphone e tablet

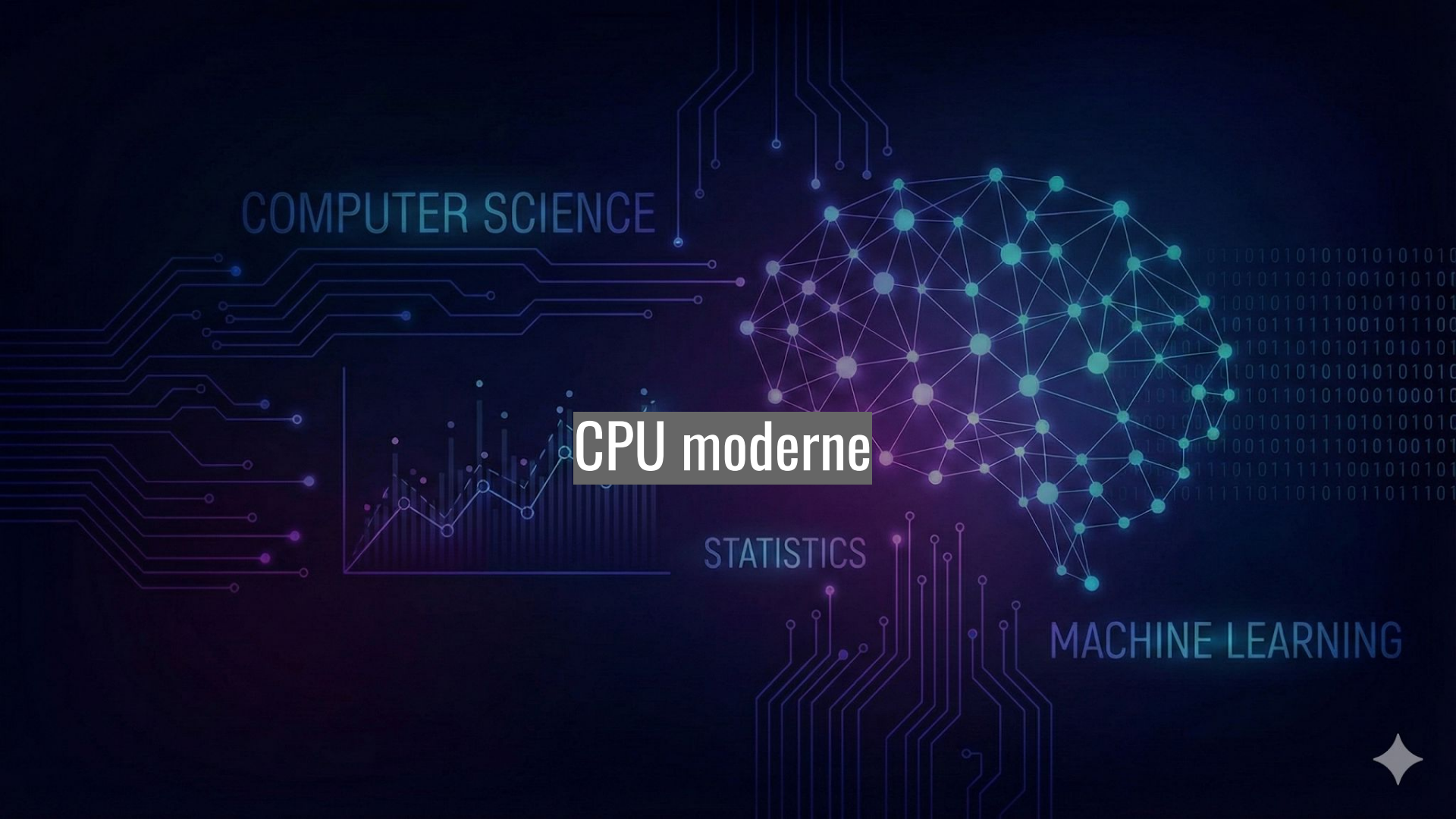
- **i sistemi embedded** molto diffusi sono le calcolatrici progettate per orologi, automobili, elettrodomestici, dispositivi medici e lettori audio/video (Android Box). Queste calcolatrici costano poche decine di euro e offrono prestazioni dell'ordine di **alcune centinaia di MIPS** (**spesso dotate di sistema operativo Linux**).
- **Gli smartphone e i tablet,** d'altro canto, sono sistemi con una potenza di calcolo significativamente superiore. **Gli smartphone di fascia alta moderni (come iPhone 15 Pro o Galaxy S24) dispongono di motori neurali e GPU che raggiungono prestazioni nell'ordine dei Teraflops, non solo dei GFLOPS.**

Console per videogiochi, PC e workstation

- **Le console di gioco** sono sistemi con prestazioni complessive che possono raggiungere circa **10-12 TFLOPS (10.000-12.000 GFLOPS), considerando anche le GPU.**
- **I PC, considerando desktop e laptop,** coprono una vasta gamma di possibilità, a partire da poche centinaia di euro fino a qualche migliaio, con prestazioni che **vanno quindi da centinaia a 80.000 GFLOPS considerando la GPU.**
- Esistono **server e workstation** progettati per **il calcolo ad alte prestazioni** e la centralizzazione dei servizi. **Con 20.000 euro oggi, ci si può aspettare oltre 100 TFLOPS per grafica/intelligenza artificiale (workstation), ma forse solo 10-20 TFLOPS per simulazioni scientifiche ad alta precisione (server).**

HPC

- Per aumentare le prestazioni delle risorse di calcolo in generale è necessario collegare tra loro numerosi computer **interconnessi tramite reti ad alte prestazioni (calcolo parallelo).**
- Ad esempio, **cluster di workstation**
- **Supercomputer Nel 2024, i supercomputer più potenti (come Frontier) hanno superato la barriera dell'Exaflop (1000 PFLOPS)**
- Ovviamente, i costi di produzione e di manutenzione aumentano allo stesso modo (anche solo in termini di potenza assorbita).


COMPUTER SCIENCE
CPU moderne
STATISTICS
MACHINE LEARNING

CISC vs RISC

La velocità di esecuzione di una singola istruzione è uno dei fattori determinanti delle prestazioni della CPU. Due prospettive diverse:

- **CISC (Complex Instruction Set),** in questo caso l'idea di base è che il set di istruzioni di base di una CPU debba essere il più ricco possibile, anche se **ogni singola istruzione richiede effettivamente più cicli di clock per essere eseguita.**
- In questo caso, nei processori **RISC (Reduced Instruction Set) ogni istruzione viene eseguita in un singolo ciclo di clock** . Ovviamente, saranno necessarie più istruzioni RISC per eseguire la stessa singola istruzione di un processore CISC.
- L'architettura CISC era quella predominante sul mercato negli anni '70 e '80. Oggi si osserva una tendenza a favore delle CPU di tipo RISC.

Migliorare le prestazioni

L'incremento delle prestazioni di una CPU rappresenta sempre un compromesso tra costi e consumi; ad esempio, si possono adottare alcune strategie di base:

- **Ridurre il numero di cicli** necessari per eseguire una singola istruzione (esempio banale di moltiplicazione).
- **Aumentare la frequenza** (di clock) entro limiti fisici evidenti (**ad esempio la velocità della luce**).
- **Il parallelismo, ad esempio, consiste nell'eseguire più operazioni in parallelo** (contemporaneamente).
  - **A livello di istruzioni, più istruzioni** vengono eseguite in parallelo dalla stessa CPU, ad esempio utilizzando **pipeline o processori superscalari.**
  - **Parallelismo a livello di core, ovvero più core per CPU**

Migliora le prestazioni: aumenta la frequenza di clock.

Fino al 2000, l'aumento delle prestazioni delle CPU ha coinciso in gran parte con l'aumento della frequenza di clock. **Abbiamo raggiunto circa 4 GHz.** Abbiamo raggiunto i limiti fisici (**1 GHz e quindi in un nanosecondo la distanza che un impulso elettrico può percorrere, immaginando che viaggi alla velocità della luce nel vuoto, è di circa 33 cm**).

- Le alte frequenze creano **rumore** e **aumentano il calore da dissipare**
- Ritardi nella propagazione del segnale,
- **bus-skew** segnali che viaggiano su linee diverse viaggiano con tempi diversi

Miglioramento delle prestazioni: aumento della frequenza di clock e miniaturizzazione.

Il limite della velocità della luce (ritardo del segnale): questa è la ragione fondamentale. L'elettricità viaggia molto velocemente, ma non istantaneamente. All'interno di fili di rame o oro, i segnali viaggiano a circa 15-20 centimetri al nanosecondo (più lentamente della luce nel vuoto).

- Lo scenario: immaginate una CPU che funziona a 4 GHz.
  - Temporizzazione: Ha un tempo di ciclo di 0,25 nanosecondi.
  - La distanza: in 0,25 nanosecondi, un segnale può percorrere fisicamente solo dai 4 ai 5 centimetri.
  - Se una CPU fosse di grandi dimensioni (ad esempio, delle dimensioni di un piatto da portata), il ciclo di clock terminerebbe prima ancora che il segnale elettrico possa viaggiare da un lato all'altro del chip. I dati non arriverebbero in tempo per il calcolo successivo.

Per aumentare la frequenza (ridurre il tempo), è necessario ridurre la distanza. Bisogna avvicinare i componenti in modo che il segnale arrivi istantaneamente.

Miglioramento delle prestazioni: aumento della frequenza di clock e miniaturizzazione.

Il transistor del "problema del secchio" si comporta come un minuscolo condensatore : immaginatevelo come un secchio d'acqua.

- Per commutare un transistor da "0" a "1" (da spento ad acceso), è necessario riempire quel contenitore con elettroni.
- Per tornare a "0", devi svuotarlo.
- Il problema: transistor di grandi dimensioni = contenitori di grandi dimensioni. Ci vuole più tempo per riempirli. Se si tenta di commutarli troppo velocemente (alta frequenza), non si riempiono né si svuotano mai completamente e i dati risultano corrotti.
- Transistor piccoli = Secchielli piccoli. Puoi riempire e svuotare un ditale molto più velocemente di una vasca da bagno.

La miniaturizzazione riduce le dimensioni della "capacità di gate", consentendo al transistor di cambiare stato miliardi di volte al secondo senza ritardi.

Miglioramento delle prestazioni: aumento della frequenza di clock e miniaturizzazione.

Potenza e calore (densità di potenza)

La formula per la potenza consumata da una CPU è approssimativamente la seguente:

$$P = C * V^2 * f$$

(Potenza = Capacità × Tensione al quadrato × Frequenza)

Se si aumenta la frequenza (f), la potenza (P) aumenta vertiginosamente, generando un calore enorme.

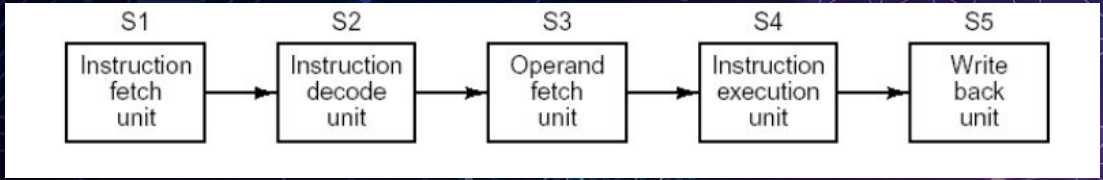
Per evitare che la CPU si surriscaldi, è necessario ridurre le altre variabili.

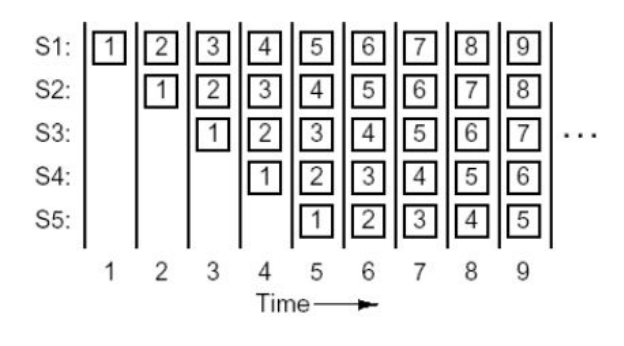
- **La miniaturizzazione riduce la capacità (C).**
- **La miniaturizzazione consente di ridurre la tensione (V).**

Riducendo le dimensioni delle particelle, si richiede meno energia per muovere gli elettroni, compensando così il calore generato dall'aumento della velocità.

Miglioramento delle prestazioni – Pipeline

Ogni singola istruzione è suddivisa in più fasi e ciascuna fase è gestita da una parte dedicata della CPU (hardware):

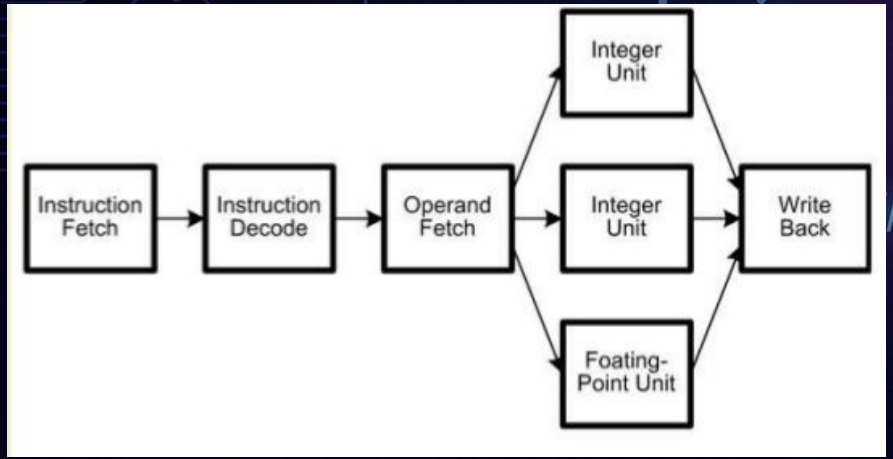




Chiaramente, **le operazioni devono essere indipendenti** . In questo caso, dopo un tempo iniziale (**latenza**) per caricare la pipeline, verranno eseguite n operazioni in parallelo. È possibile utilizzare anche più pipeline, e quindi più istruzioni vengono lette ed eseguite contemporaneamente.

Migliorare le prestazioni – Superscalari

Istruzioni diverse elaborano i loro operandi simultaneamente su unità hardware diverse; **in pratica, ci sono diverse unità funzionali dello stesso tipo, ad esempio ci sono diverse ALU.**



Unità di esecuzione multiple: a differenza di un processore scalare, che ha una singola pipeline e può (al massimo) completare un'istruzione per ciclo, un processore superscalare ha più unità di esecuzione (ad esempio, due ALU per numeri interi, due unità a virgola mobile e un'unità di caricamento/memorizzazione).

Miglioramento delle prestazioni – Previsione dei salti

La pipeline funziona particolarmente bene con le istruzioni sequenziali, ma una delle strutture di base della programmazione sono le istruzioni di diramazione, come **le decisioni IF...THEN...ELSE** :

```
if (a == b)
  printf("i due numeri sono uguali");
else
  printf("Sono diversi \n");
```

Miglioramento delle prestazioni – Previsione dei salti

Le CPU moderne possono (pre-caricamento) provare a indovinare se il programma salterà :

- **Previsione statica** : utilizziamo criteri che si basano su presupposti di "buon senso", ad esempio presupponiamo che tutti i salti vengano eseguiti
- **Dinamico** : in pratica la CPU mantiene una tabella basata sulle **statistiche di esecuzione**

Miglioramento delle prestazioni – Esecuzioni fuori ordine e speculative

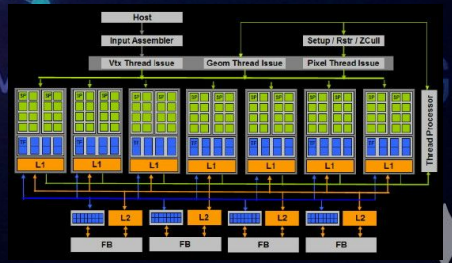
La progettazione di una CPU è notevolmente più semplice se tutte le istruzioni vengono eseguite in sequenza, ma, come abbiamo visto, non possiamo fare questa ipotesi a priori. Esistono delle dipendenze. Ad esempio, l'esecuzione di una determinata istruzione richiede che sia noto il risultato dell'istruzione precedente.

- **Esecuzione fuori ordine** : le CPU moderne possono **saltare temporaneamente alcune istruzioni per aumentare le prestazioni, mettendole in attesa** per eseguire altre istruzioni che non introducono dipendenze.
- **Esecuzione speculativa** : eseguire parti di codice particolarmente complesse prima di essere certi che siano effettivamente necessarie.

CPU - Considerazioni finali

CPU moderne:

- **Multi-core** : in pratica, nello stesso chip sono presenti più processori indipendenti, **ognuno con la propria memoria cache, ad esempio, interconnessi tra loro** . Ciò consente di aumentare le prestazioni "teoriche" senza incrementare la frequenza (anche con **l'Hyper-threading**).
- **le GPU** (processori grafici) sono sempre più utilizzate per accelerare i calcoli.



CPU - Considerazioni finali

Sistemi embedded, IoT e CPU mobili (incluso Raspberry Pi):

- ARM si riferisce a un'importante famiglia di architetture di processori RISC (a 32 e 64 bit) sviluppate da Arm Holdings (un'azienda britannica). La particolarità di Arm è che non produce direttamente i chip, ma progetta l'architettura e la concede in licenza ad altre aziende. Queste vengono poi prodotte come System-on-Chip (SoC) da produttori come Apple, Qualcomm, Samsung, NVIDIA, STMicroelectronics e Broadcom (utilizzata nel Raspberry Pi).
- Queste CPU RISC sono progettate per ottimizzare l'equilibrio tra alte prestazioni ed efficienza energetica. Sebbene inizialmente dominata nei dispositivi mobili e embedded, l'architettura si è ora diffusa anche nei laptop e nei supercomputer.
    - Confronto: un moderno ed efficiente core ARM (ad esempio, un Cortex-A55 o simili, presenti in smartphone e dispositivi IoT) consuma in genere tra 100 mW e 500 mW sotto carico normale. Al contrario, una CPU x86 desktop ad alte prestazioni (come un moderno Intel Core i9 o un AMD Ryzen 9) può consumare oltre 200 W al massimo delle prestazioni.

Perché RISC consuma meno energia di CISC

La penalità di decodifica (il principale colpevole). Questo è il fattore più importante in assoluto.

- **CISC (ad esempio, Intel x86): Le istruzioni sono come frasi complesse** . Hanno lunghezze variabili (alcune sono lunghe 1 byte, altre 15 byte) e possono eseguire molte operazioni contemporaneamente (ad esempio, "Vai alla memoria, prendi un numero, aggiungilo a questo registro e salvalo di nuovo").
  - Il **costo: la CPU necessita di un circuito enorme e ad alto consumo energetico, chiamato decodificatore, solo per capire dove finisce un'istruzione e inizia la successiva** . È necessaria una notevole quantità di energia elettrica per "tradurre" queste istruzioni complesse in segnali comprensibili al chip.
- **RISC (ad esempio, ARM): Le istruzioni sono come semplici** comandi. Hanno una lunghezza fissa (solitamente 32 bit).
  - Il vantaggio: **l'hardware sa esattamente quanto è grande ogni istruzione** . Il decodificatore è minuscolo, semplice ed essenzialmente "cablato". Consuma pochissima energia perché non deve eseguire analisi complesse sul flusso di codice.

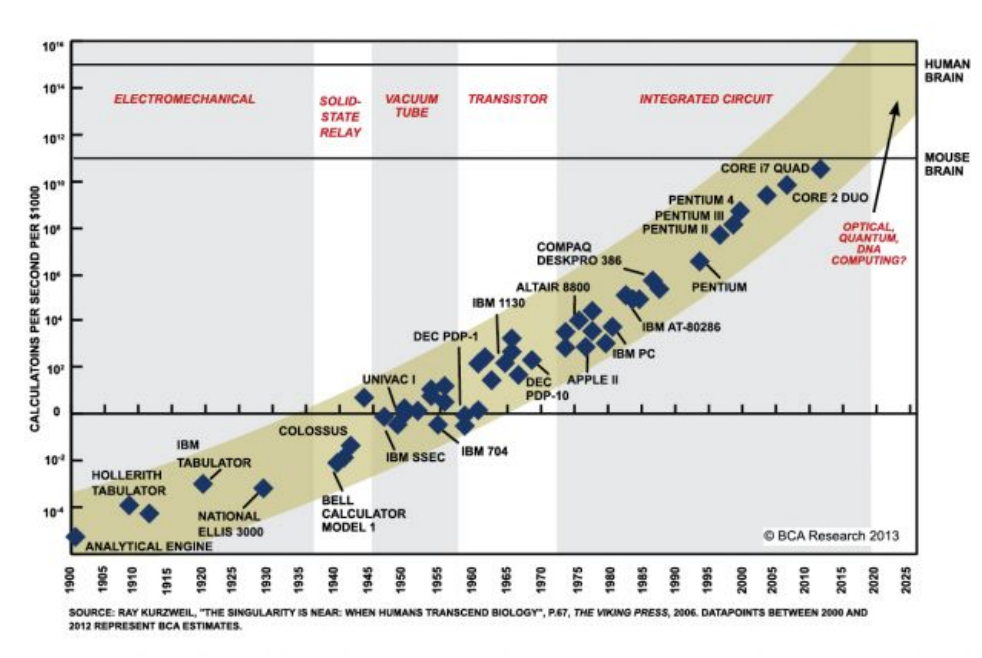
Perché RISC consuma meno energia di CISC

Oggi, il confine è un po' sfumato.

- **I moderni processori CISC (come Intel Core i9) in realtà barano: utilizzano un traduttore hardware per convertire internamente le istruzioni CISC in "micro-operazioni" di tipo RISC. Tuttavia, anche questa fase di conversione consuma costantemente energia.**
- **I moderni processori RISC (come Apple M3 o Snapdragon) stanno diventando sempre più complessi per ottenere prestazioni migliori, ma continuano a beneficiare dell'efficienza fondamentale del loro set di istruzioni, che consente loro di funzionare a temperature più basse e di consumare meno batteria rispetto alle loro controparti CISC.**

TOP500 E LA LEGGE DI MOORE

Legge di Moore



La prima legge di Moore affermava inizialmente che la complessità dei microcircuiti (ad esempio, il numero di transistor per chip) raddoppia ogni 18 mesi. Sebbene valida per decenni, la legge ha rallentato il periodo di raddoppio (ora è di 2,5-3 anni). Gli attuali miglioramenti prestazionali privilegiano architetture specializzate (come gli acceleratori per l'intelligenza artificiale), l'impilamento 3D (chiplet) e la progettazione efficiente rispetto alla sola miniaturizzazione dei transistor.

Legge di Moore

È ancora valida oggi? Negli anni 2020, la Legge di Moore sta rallentando o, secondo alcuni (come l'amministratore delegato di Nvidia), è "morta".

- Limiti fisici: ci stiamo avvicinando alle dimensioni degli atomi. I transistor ora si misurano in nanometri (ad esempio, 3 nm, 2 nm). A questa scala, l'effetto tunnel quantistico e la dissipazione del calore diventano problemi enormi.
- Limiti economici (seconda legge di Moore): sebbene sia ancora possibile integrare un numero maggiore di transistor, il costo di costruzione degli stabilimenti necessari sta raddoppiando. Uno stabilimento moderno costa oltre 20 miliardi di dollari, il che rende più difficile mantenere la promessa di "più economico e più veloce" della legge originale.

TOP500

- Differenza tra **prestazione sostenuta e prestazione di picco**

- Per valutare oggettivamente le prestazioni di un computer è necessario un test di riferimento, un **benchmark standard, ad esempio Linpack.**

- TOP500 http://www.top500.org/ , classifica dei 500 computer più potenti al mondo

Classifica dei 500 migliori di novembre 2025

GLOBAL TOP 10 SUPERCOMPUTERS

Ranking based on HPL Performance (Rmax)

Source: Top500.org (Novembe

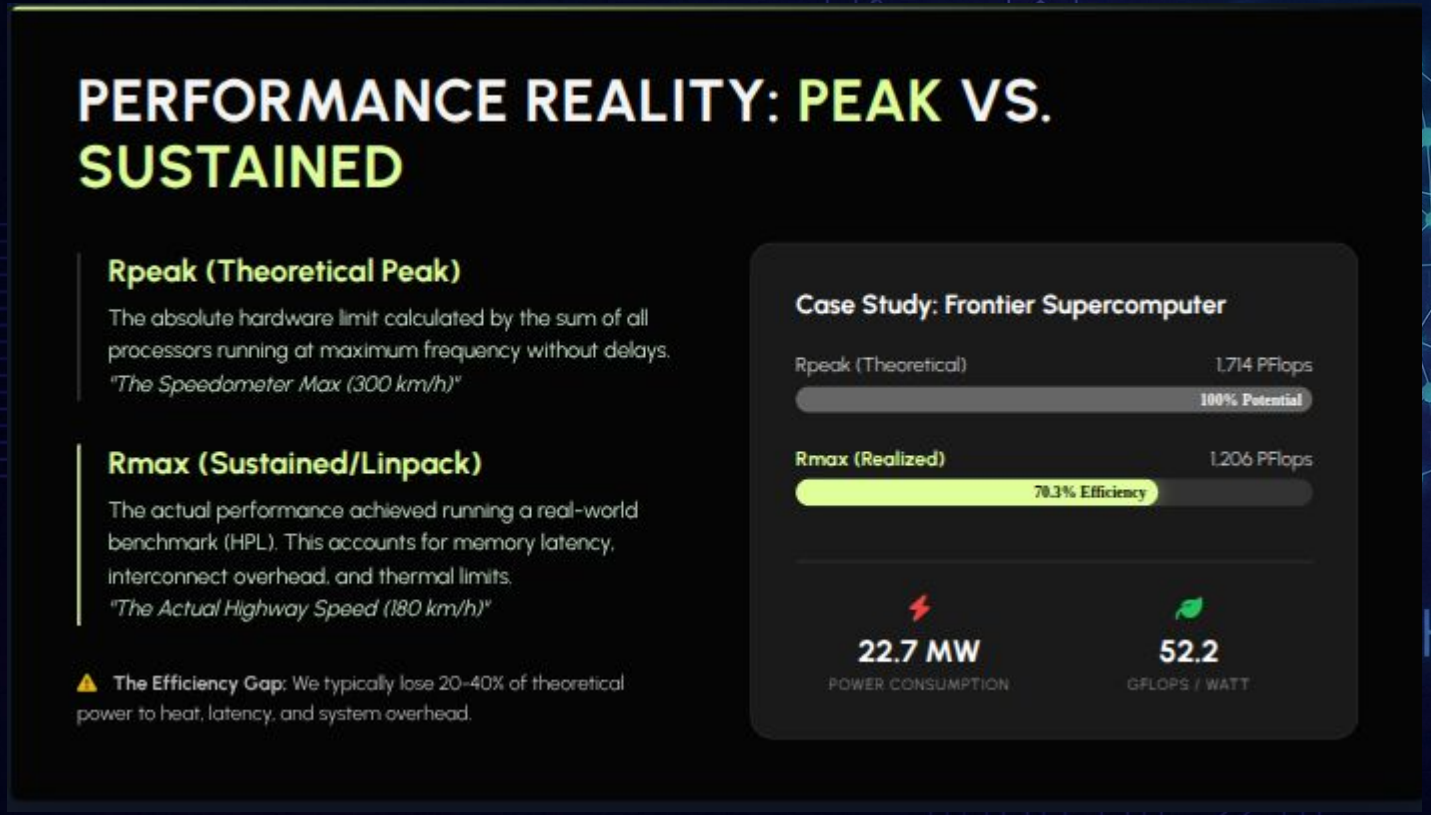
#	SYSTEM	CORES	RMAX (PFLOP/S)	POWER (KW)
1	El Capitan NEW #1 *USA (LLNL)*	11.3M	**1,809.00**	29,685
2	Frontier *USA (ORNL)*	9.1M	**1,353.00**	24,607
3	Aurora *USA (ANL)*	9.3M	**1,012.00**	38,698
4	JUPITER Booster NEW *Germany (EuroHPC)*	4.8M	**1,000.00**	15,794
5	Eagle *USA (Microsoft Azure)*	2.1M	561.20	N/A
6	HPC6 NEW *Italy (Eni S.p.A)*	3.1M	477.90	8,461
7	Fugaku *Japan (RIKEN)*	7.6M	442.01	29,899
8	Alps *Switzerland (CSCS)*	2.1M	434.90	7,124
9	LUMI *Finland (EuroHPC)*	2.8M	379.70	7,107
10	Leonardo *Italy (EuroHPC)*	1.8M	241.20	7,494

Grafico storico della classifica Top500

Si stima che la **potenza di calcolo della mente umana si aggiri intorno** a qualche decina **di petaflops o più.**

Tale stima (poche decine di petaflop) è considerata corretta ma prudente, basandosi su modelli teorici più datati, sebbene molti neuroscienziati e ingegneri informatici moderni ritengano ora che il numero sia significativamente più alto, probabilmente nell'ordine degli **exaflop (1.000 petaflop).**

Grafico storico della classifica Top500



Nelle applicazioni reali il divario è molto più ampio

HINE LEARNING

Grafico storico della classifica Top500

Da dove proviene la stima "Petaflop"?

La stima di 10^{15} -10^{16} operazioni al secondo (1-50 Petaflops) è stata resa popolare da futuristi come Hans Moravec e Ray Kurzweil tra la fine degli anni '90 e l'inizio degli anni 2000.

- **Hanno stimato che il cervello abbia circa 10^{14} sinapsi (connessioni). Se ogni sinapsi elabora circa 10 segnali al secondo, si ottengono 10^{15} operazioni (1 Petaflop).**
- **Se il cervello funziona in modo semplice (come un interruttore binario), "poche decine di petaflop" è corretto.**

Grafico storico della classifica Top500

La visione moderna: la stima "Exaflop"

Ricerche recenti suggeriscono che il cervello sia molto più complesso. Una sinapsi non è un semplice interruttore on/off; è una complessa macchina molecolare che elabora le informazioni in modo non lineare.

- Se è necessario simulare le interazioni chimiche all'interno delle sinapsi e dei dendriti per replicare la funzione del cervello, il fabbisogno computazionale sale a 1 Exaflop (10^{18}) o addirittura a 1 Zettaflop (10^{21}).
- Secondo questo criterio, la stima di "poche decine di petaflop" è troppo bassa di un fattore compreso tra 100 e 1.000.

Grafico storico della classifica Top500

Feature	Human Brain	El Capitan (Nov 2025 #1 Supercomputer)
Speed (Est.)	~1 Exaflop (Variable)	1.8 Exaflops (Sustained)
Parallelism	Massive (Billions of threads)	High (Millions of cores)
Clock Speed	Slow (~200 Hz / 0.0002 GHz)	Fast (1.8 GHz - 3.0 GHz)
Power Usage	**~20 Watts** (A dim lightbulb)	**~30,000,000 Watts** (A small city)
Cooling	Blood flow	Massive liquid cooling plant

SEMPRE A PROPOSITO DI ARCHITETTURA

FPGA

- **Il Field Programmable Gate Array** (FPGA) è un circuito integrato le cui funzioni sono programmabili tramite software. Gli FPGA sono dispositivi a semiconduttore basati su una matrice di blocchi logici configurabili (CLB) collegati tramite interconnessioni programmabili.
- Le FPGA possono essere riprogrammate in base ai requisiti applicativi o funzionali desiderati dopo la produzione. Questa caratteristica distingue le FPGA dai circuiti integrati specifici per applicazioni (ASIC), che sono realizzati su misura per compiti di progettazione specifici. Sebbene siano disponibili FPGA programmabili una sola volta (OTP), i tipi predominanti sono basati su SRAM, che possono essere riprogrammati man mano che il progetto si evolve.
- **ASIC (circuito integrato specifico per applicazione)**