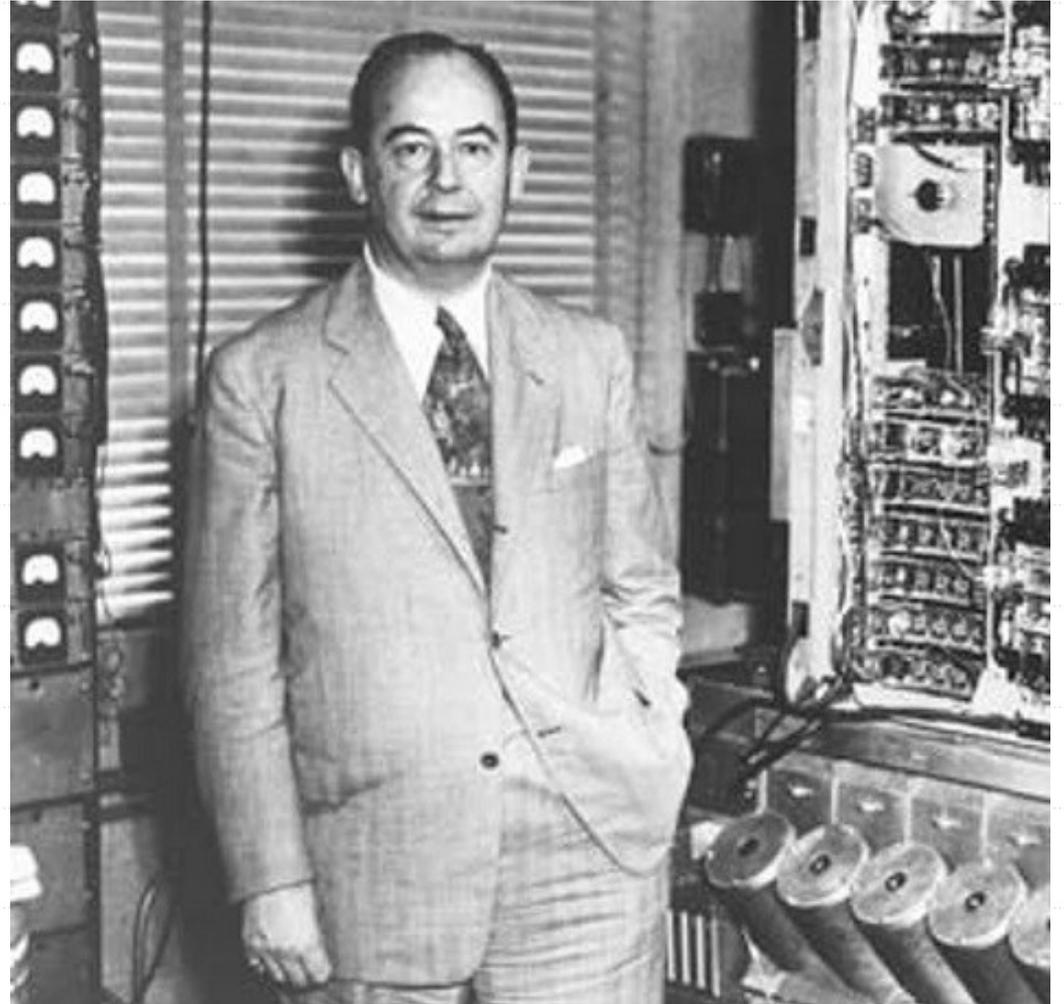


IL MODELLO DI VON NEUMANN



IL MODELLO DI VON NEUMANN

*John von Neumann nacque a **Budapest** il 28 dicembre **1903** da una famiglia di banchieri **ebrei**. Le cronache narrano che già all'età di sei anni intrattenesse gli ospiti di famiglia con la sua prodigiosa memoria, ripetendo all'istante intere pagine di elenco telefonico che gli erano state mostrate solo per pochi istanti o eseguendo rapidamente a mente divisioni con due numeri da otto cifre.*

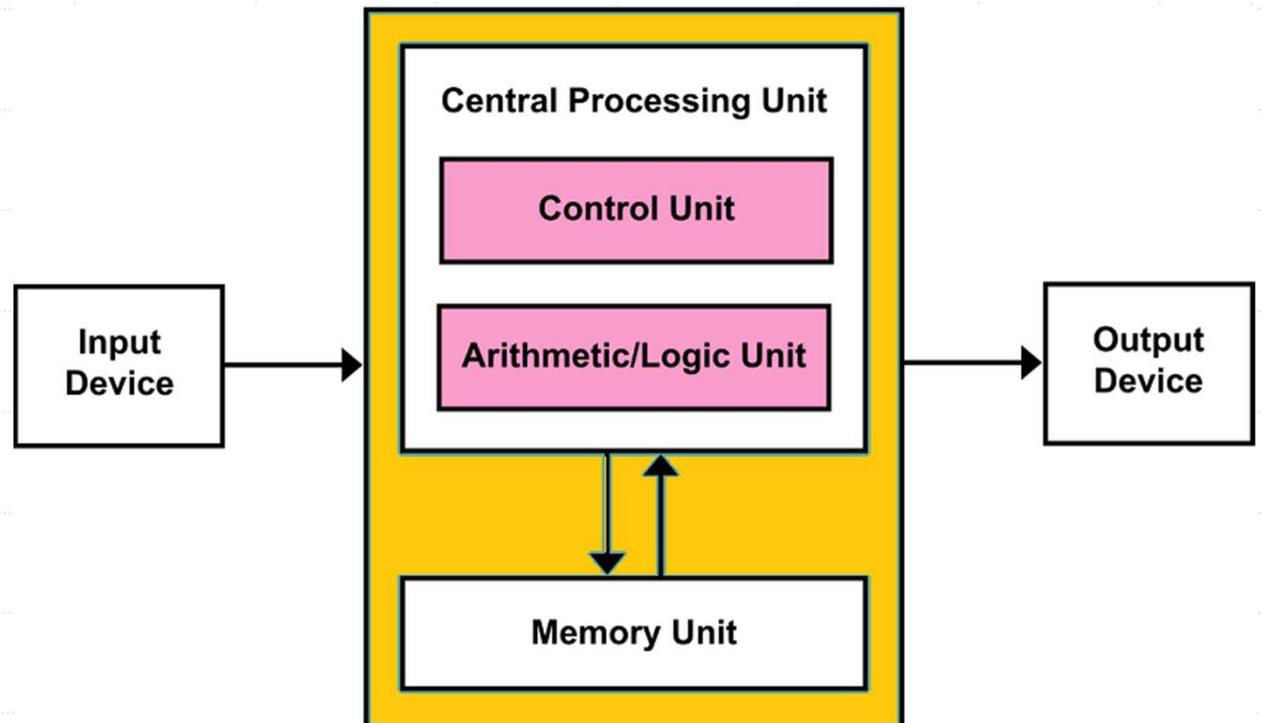


- Oggi parleremo del modello di Von Neumann, un principio fondamentale dell'architettura dei computer moderni
- Questo modello, sviluppato negli anni '40 da John von Neumann, è alla base della struttura di quasi tutti i computer che utilizziamo oggi
- Per comprendere meglio, analizzeremo il modello in dettaglio, esaminando ogni componente e come interagiscono tra loro



Componenti del Modello di Von Neumann

- Unità di Controllo
- Unità Aritmetico-logica
- Memoria
- Input/Output
- Bus



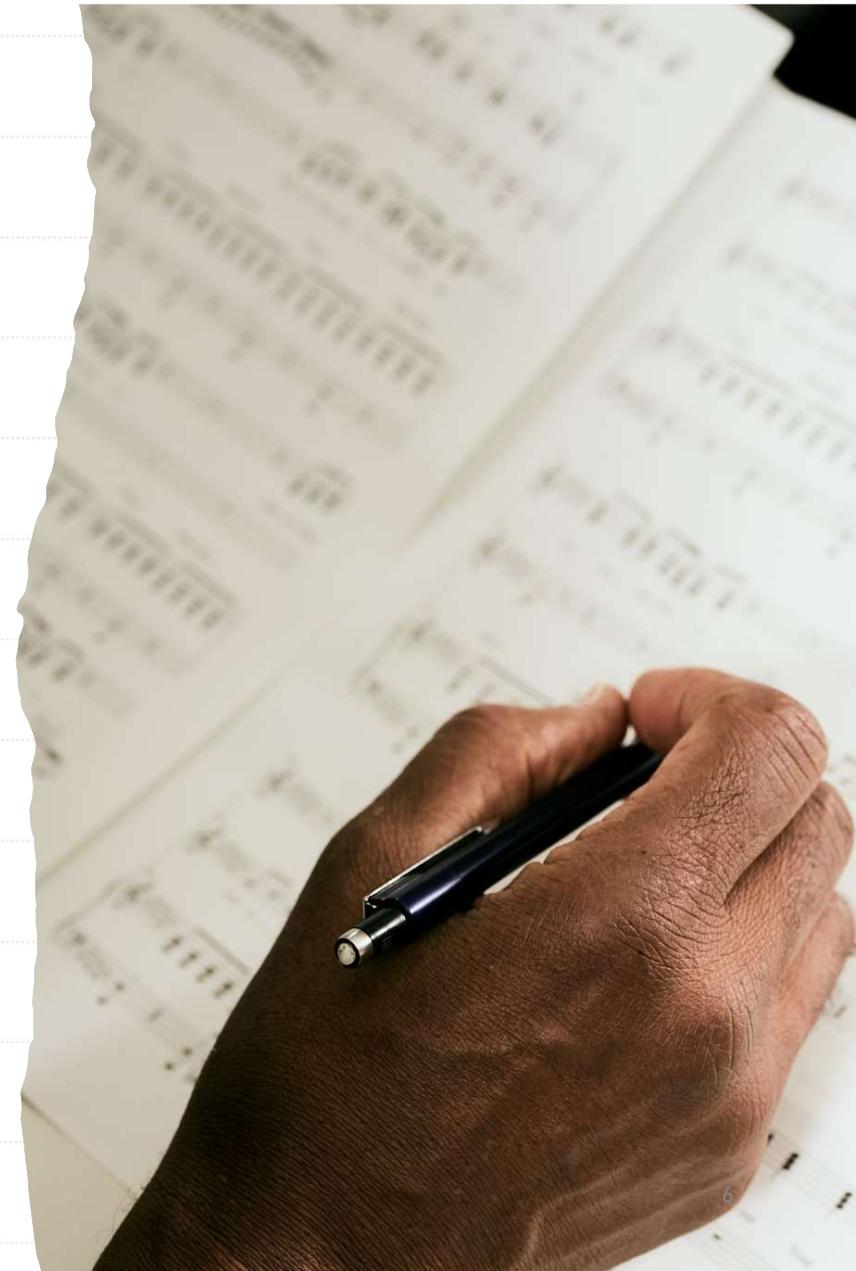
Unità di Controllo

- È responsabile dell'esecuzione dei programmi che sono memorizzati nella memoria, leggendo e interpretando ogni istruzione, e controllando l'operazione delle altre unità in modo che possano eseguire le istruzioni correttamente
- Pensiamo all'Unità di Controllo come a un direttore d'orchestra che guida i musicisti durante un concerto
- Il direttore decide quale musicista deve suonare e in quale momento, garantendo che la musica fluisca armoniosamente



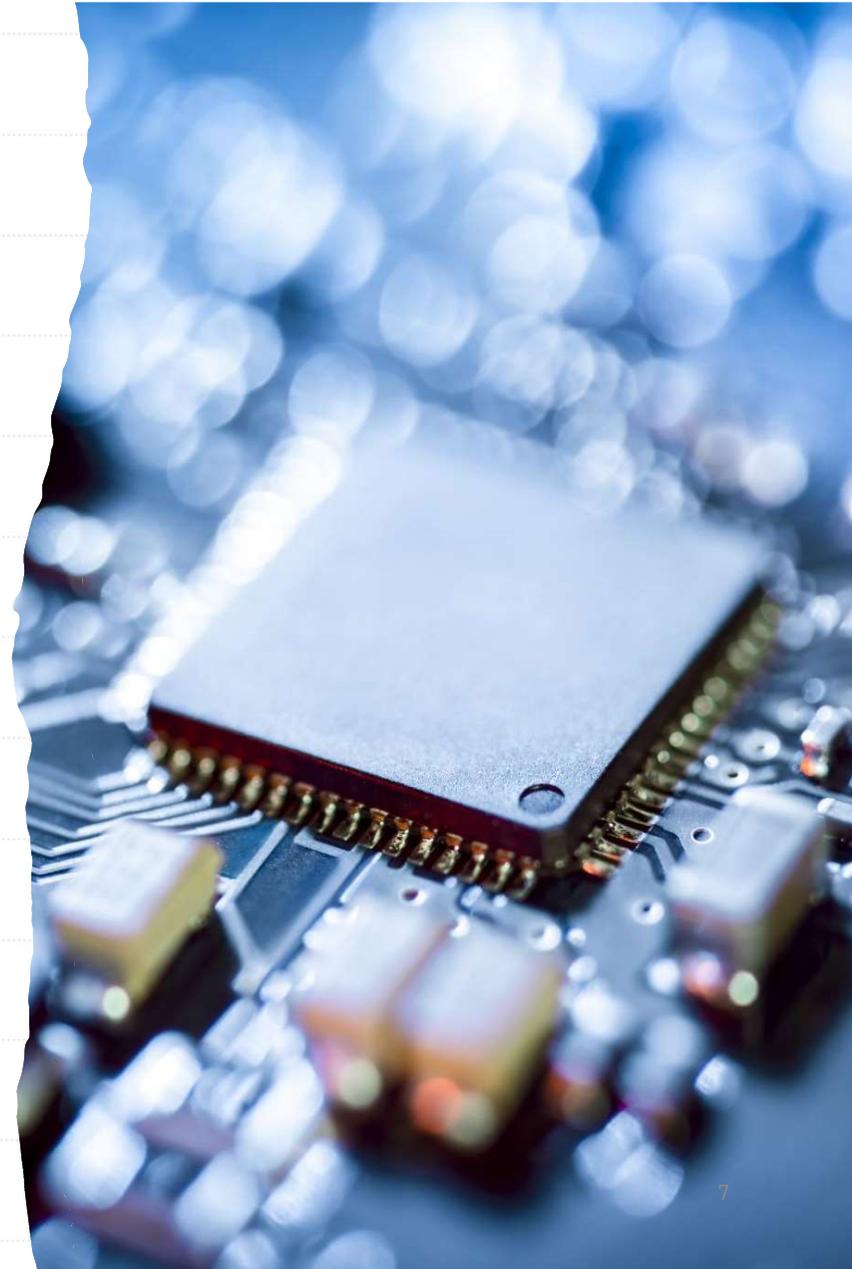
Unità Aritmetico-Logica

- Esegue tutti i calcoli e le operazioni logiche che sono necessarie durante l'esecuzione di un programma
- Se consideriamo l'Unità di Controllo come il direttore d'orchestra, l'ALU sarebbe come i musicisti che eseguono le note, suonando gli strumenti per produrre la musica
- L'ALU esegue le operazioni matematiche e logiche necessarie per tradurre le istruzioni in azioni



Memoria

- La Memoria è il componente che conserva sia i dati che le istruzioni necessarie per l'esecuzione dei programmi
- La memoria può essere di due tipi: RAM , che è volatile, e la memoria di massa , che è non volatile (**...che non fa parte!**)
- La memoria può essere paragonata alla partitura che i musicisti leggono mentre suonano



Input/Output

- Le unità di Input/Output sono i mezzi con cui il computer comunica con il mondo esterno
- L'input può provenire da varie fonti come tastiere e mouse, mentre l'output può essere visualizzato su monitor o stampanti
- Gli spettatori inviano applausi o richieste al direttore e ai musicisti, e ricevono la musica come risposta



Bus

- Il Bus è un sistema di comunicazione che trasporta dati tra le diverse parti del computer
- Esistono diversi tipi di bus, ciascuno con compiti specifici, come il bus dei dati, il bus degli indirizzi e il bus di controllo
- Il Bus può essere immaginato come i corridoi di un teatro, che permettono ai musicisti, al direttore e agli spettatori di muoversi tra le diverse aree del teatro



Il Flusso di Esecuzione



Il programma viene caricato in memoria



L'Unità di Controllo legge la prima istruzione dalla memoria



L'istruzione viene decodificata per determinare l'operazione da eseguire



L'ALU esegue l'operazione, come una somma



Il risultato viene memorizzato in memoria o inviato all'unità di output

Limitazioni e Sviluppi

IL MODELLO DI VON NEUMANN HA ALCUNE LIMITAZIONI

UNA DELLE PIÙ EVIDENTI È IL COLLO DI BOTTIGLIA DI VON NEUMANN, UN LIMITE ALLA VELOCITÀ DI ELABORAZIONE DOVUTO AL FATTO CHE LE ISTRUZIONI E I DATI CONDIVIDONO LA STESSA MEMORIA E IL BUS

IL COLLO DI BOTTIGLIA DI VON NEUMANN È UN LIMITE INTRINSECO DELL'ARCHITETTURA DI VON NEUMANN, DOVUTO AL FATTO CHE DATI E ISTRUZIONI CONDIVIDONO LA STESSA MEMORIA E LO STESSO BUS

Conclusione

- Il modello di Von Neumann ha rappresentato un punto di partenza cruciale per lo sviluppo dei sistemi informatici che usiamo ogni giorno
- La sua struttura, composta da Unità di Controllo, ALU, Memoria, Input/Output e Bus, ci permette di eseguire una vasta gamma di applicazioni e programmi, dai più semplici ai più complessi
- Attraverso esempi pratici e confronti, abbiamo visto come ogni componente del modello ha un ruolo preciso e indispensabile nell'esecuzione dei programmi, e come il flusso di esecuzione si sviluppi attraverso il ciclo di fetch-decode-execute

