



ELEMENTI DI INFORMATICA

DOCENTE: FRANCESCO MARRA

INGEGNERIA CHIMICA

INGEGNERIA ELETTRICA

SCIENZE ED INGEGNERIA DEI MATERIALI

INGEGNERIA GESTIONALE DELLA LOGISTICA E DELLA PRODUZIONE

INGEGNERIA NAVALE

UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE

ARCHITETTURA DEL CALCOLATORE: MODELLO DI VON NEUMANN



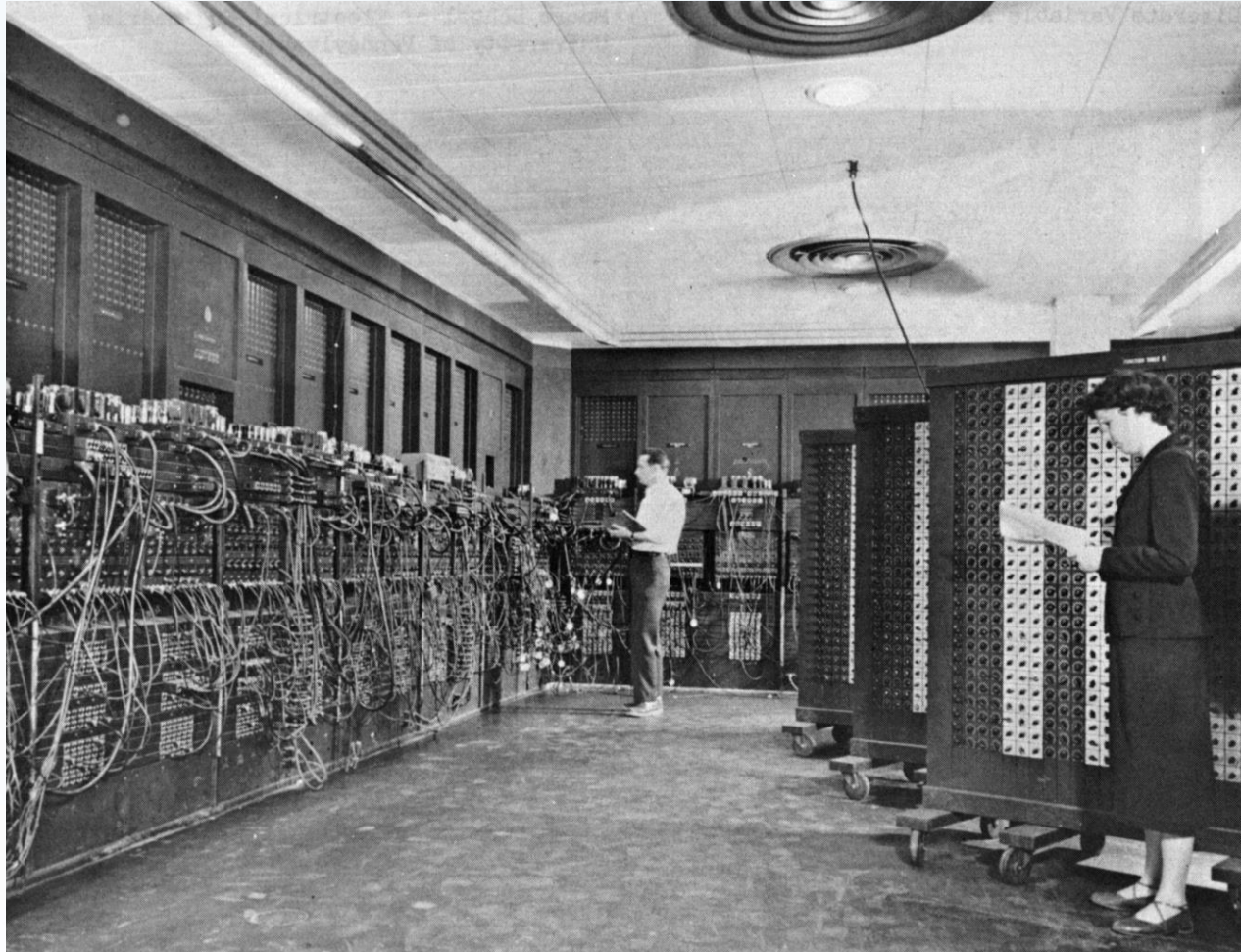
AGENDA

- Architettura del calcolatore
 - Evoluzione
 - Modello di von Neumann
 - CPU
 - Memoria
 - Dispositivi di Input e Output
 - Bus
 - Clock

CALCOLATORE

- **Macchina** che, in maniera **automatica**, ad **altissima velocità**, esegue **operazioni elementari** dettate da un **algoritmo** memorizzato
 - Non ha nessuna capacità decisionale o discrezionale, ma si limita a compiere determinate azioni secondo procedure prestabilite
 - Milioni di istruzioni per secondo (MIPS)
 - Può compiere solo un numero limitato di operazioni
 - Un algoritmo deve essere comunicato al calcolatore in un linguaggio ad esso comprensibile

EVOLUZIONE DEI CALCOLATORI



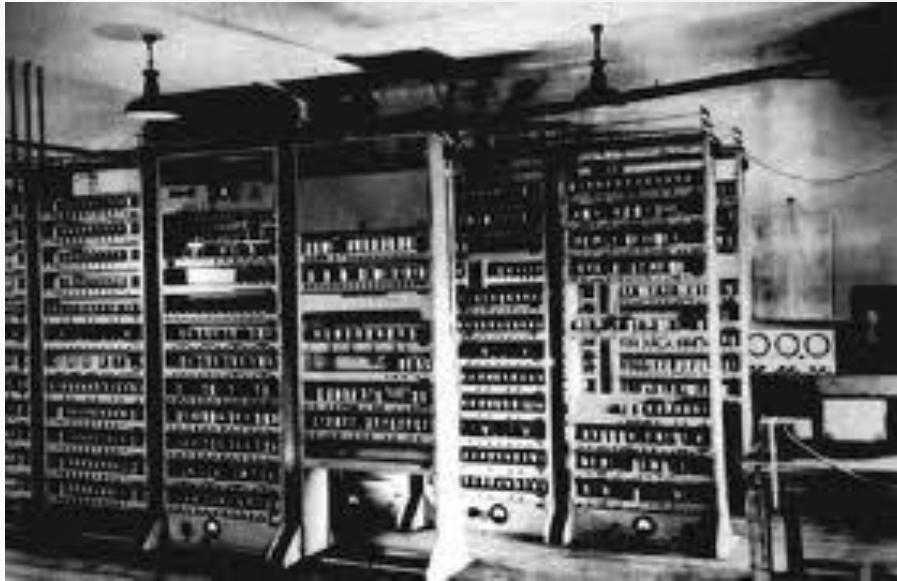
ENIAC:
Electronic Numerical
Integrator And Computer

Primo esempio:
macchina limitata
quasi priva di memoria
priva di elasticità

EVOLUZIONE DEI CALCOLATORI

- **Miglioramenti decisivi:**
 - **Intuizione di Alan Turing**
 - permettere al computer (l'hardware) di eseguire istruzioni codificate in un programma (software) inseribile e modificabile dall'esterno.
 - **Architettura proposta da John von Neumann**
 - realizzazione della macchina universale inventata da Turing
 - computer programmabile nel senso moderno del termine

EVOLUZIONE DEI CALCOLATORI



EDVAC

**(Electronic Discrete Variable
Automatic Calculator)**



***Personal
Computer***

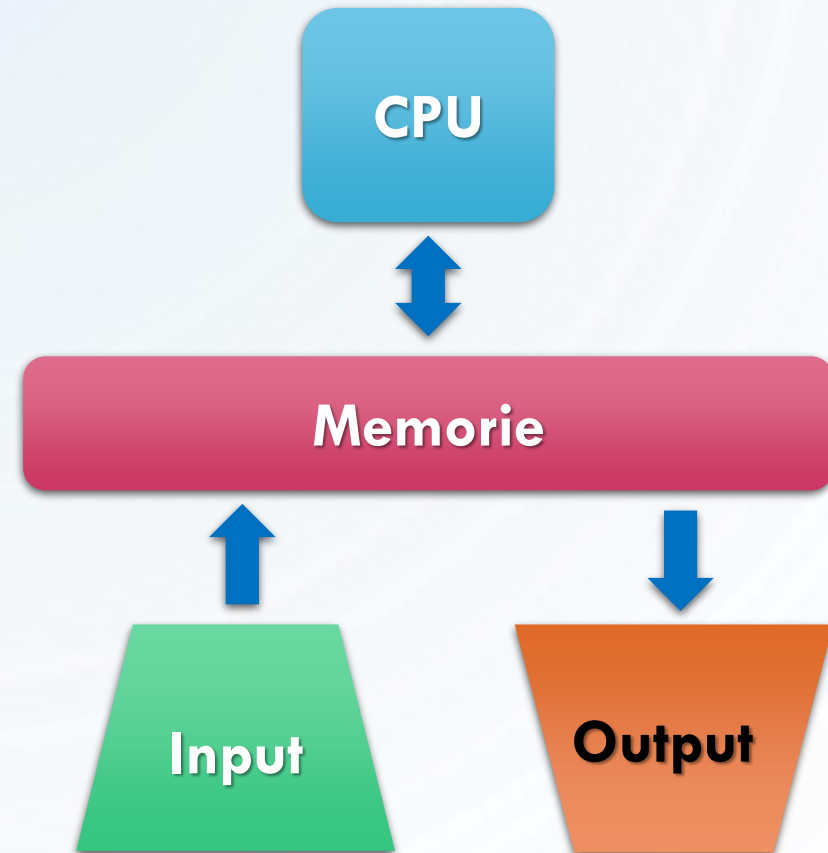


Smartphone

MODELLO DI VON NEUMANN

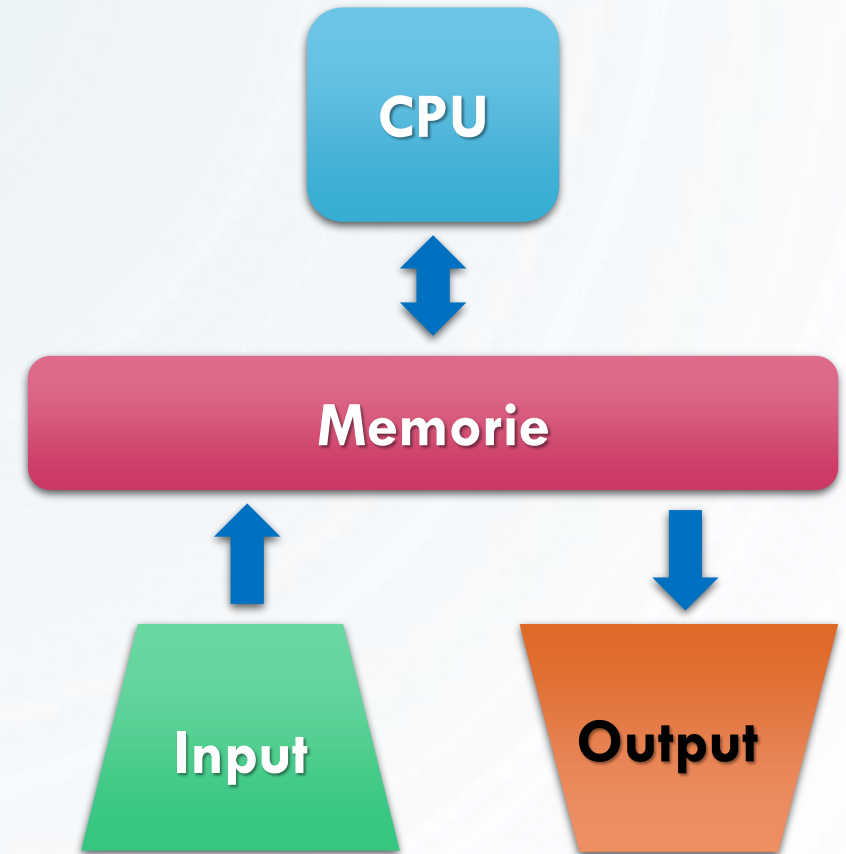


- **Schema di principio** rappresentativo dei tradizionali computer
- Prende il nome da Von Neumann, il primo ricercatore che lo propose nel 1945



UNITÀ DEL MODELLO DI VON NEUMANN

- **CPU (Central Processing Unit)**
 - Coordina l'esecuzione delle operazioni fondamentali
- **Memoria**
 - Contiene l'algoritmo con le operazioni da eseguire e i dati su cui opera
- **Unità di input**
 - Consente l'inserimento di algoritmo e dati in memoria
- **Unità di output**
 - Permette di presentare i risultati dell'attività della CPU



CARATTERISTICHE DEL MODELLO

- Si basa sul concetto di **programma memorizzato**
 - La macchina immagazzina nella propria memoria i dati su cui lavorare e le istruzioni per il suo funzionamento
- **Flessibilità** operativa
 - Macchine nate per fare calcoli possono essere impiegate nella risoluzione di problemi di natura completamente diversa, come problemi di tipo amministrativo, gestionale e produttivo
- Schema di funzionamento **semplice** nelle sue linee generali

CARATTERISTICHE DEL MODELLO

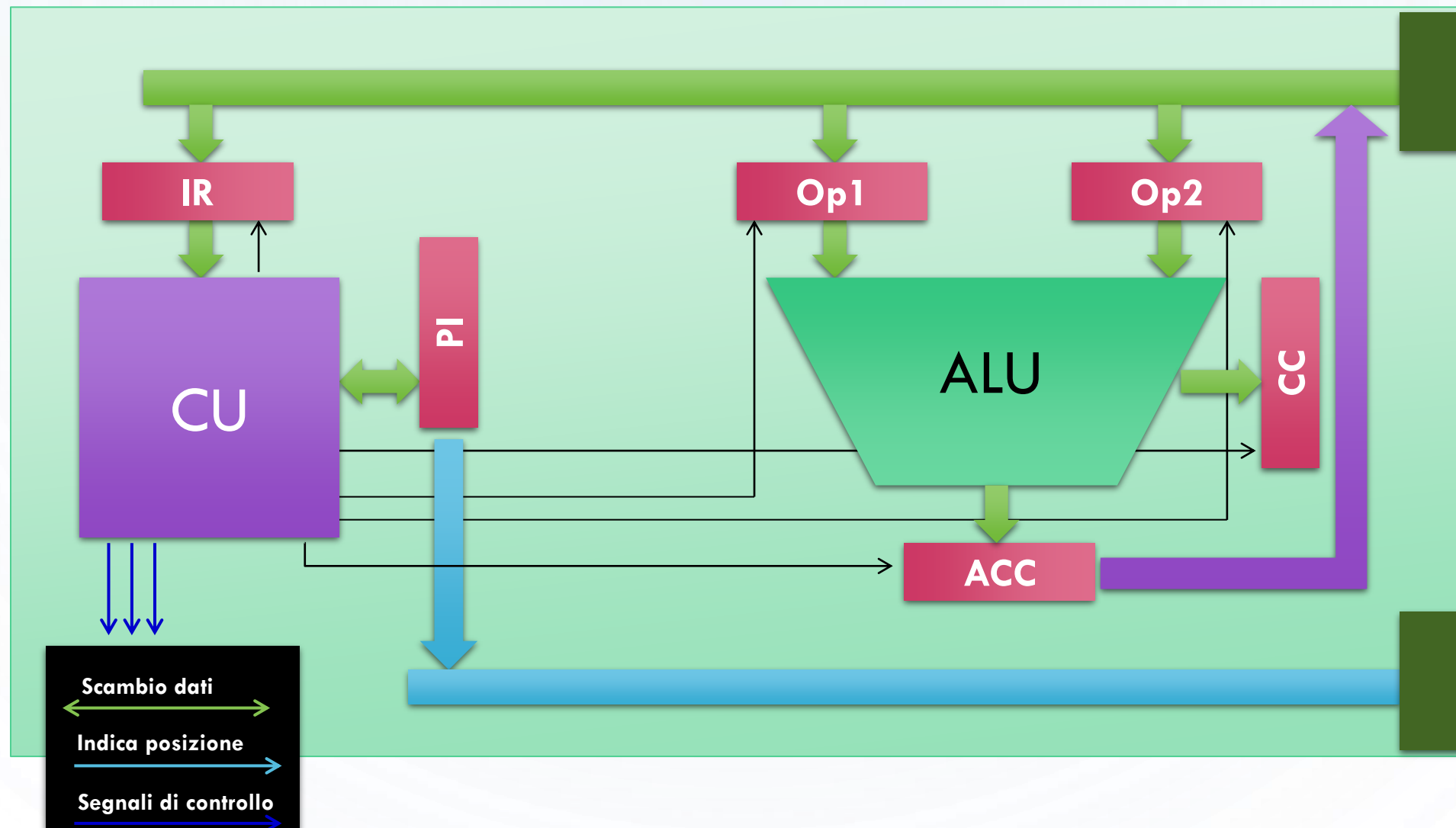
- **Velocità** nella esecuzione degli algoritmi
 - Milioni di istruzioni svolte dalla CPU in un secondo
- **Affidabilità**
 - Un computer non commette mai errori di algoritmo poiché è un esecutore obbediente dell'algoritmo stesso la cui esecuzione gli è stata affidata
- Adeguata **capacità** di memoria (numero di informazioni)
 - Varia in base al tipo di memoria usato, all'architettura della memoria stessa ed al tipo di informazione
 - Si misura in numero di byte
- Costo vantaggioso

CPU

- Contiene i dispositivi elettronici in grado di acquisire, interpretare ed eseguire il programma contenuto in memoria centrale operando la trasformazione dei dati
- È composta da tre parti fondamentali:
 - *Unità di Controllo o Control Unit (CU)*
 - *Unità Logico Aritmetica (ALU)*
 - *Registri interni*
- La CPU è anche detta *processore*
 - In senso stretto, il processore corrisponde alla CU



SCHEMA INTERNO DI UNA CPU



UNITÀ DI CONTROLLO (CU)

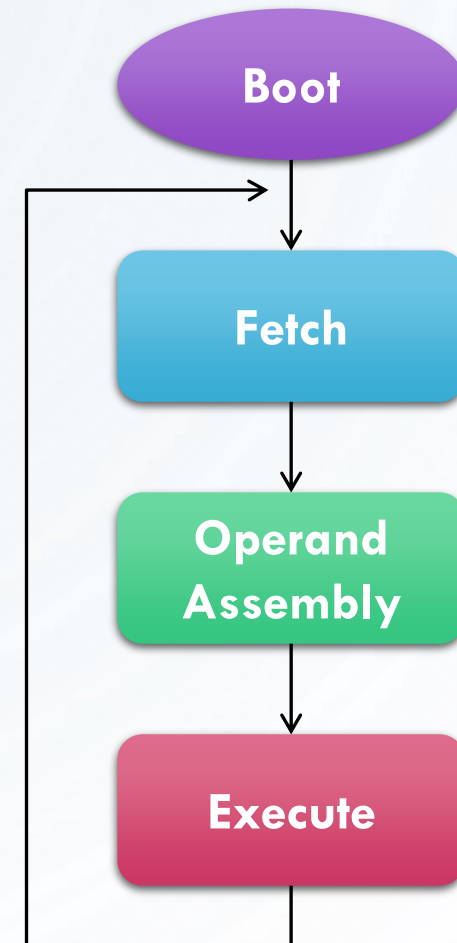
- Interpreta le singole istruzioni e attiva tutti i meccanismi necessari al loro espletamento
- Controlla in maniera ciclica che una serie di fasi vengano correttamente eseguite



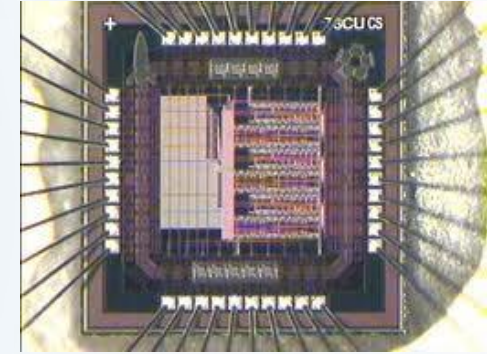
Ciclo del processore

FASI DEL CICLO DEL PROCESSORE

- **Boot**
 - Operazione iniziale atta ad informare la CU dell'indirizzo del registro di memoria che contiene la prima istruzione da eseguire
- **Fetch**
 - Un'istruzione viene prelevata dalla memoria centrale, **decodificata** e **memorizzata** in un registro interno
 - Anche detta *Instruction Fetch and Decode*
- **Operand Assembly**
 - I dati vengono prelevati dalla memoria (se servono all'istruzione)
- **Execute**
 - L'istruzione viene eseguita



UNITÀ LOGICO ARITMETICA (ALU)



- **Esegue** operazioni aritmetiche, di confronto o bitwise sui dati della memoria centrale o dei registri interni
- L'esito dei suoi calcoli viene segnalato da appositi bit in un registro chiamato *Condition Code*
- A seconda dei processori l'ALU può essere molto complessa
 - Nei sistemi attuali l'ALU viene affiancata da processori dedicati alle operazioni sui numeri in virgola mobile detti **processori matematici**

REGISTRI INTERNI

- Durante le sue elaborazioni la CU può depositare informazioni nei suoi *registri interni*
 - Sono più facilmente individuabili e hanno ***tempi di accesso inferiori*** a quelli dei registri della memoria centrale
- Il numero e tipo di tali registri varia a seconda dell'architettura della CPU

PRINCIPALI REGISTRI INTERNI

- **Instruction Register (IR)**

- Contiene l'**istruzione** prelevata dalla memoria e che la CU sta eseguendo

- **Prossima Istruzione (PI) o Program Counter (PC)**

- Ricorda alla CU la **posizione** in memoria della successiva istruzione da eseguire
- Dopo ogni prelievo di una istruzione dalla memoria, il suo valore viene aggiornato in maniera tale da puntare alla prossima istruzione

- **Accumulatore (ACC)**

- Serve come **deposito** di dati da parte dell'ALU
 - Può contenere prima di un'operazione uno degli operandi, e al termine il risultato calcolato

- **Condition Code (CC) o Status Register (SR)**

- Indica le **condizioni** che si verificano durante l'elaborazione, quali risultato nullo, NaN e overflow

MEMORIE

- Insieme di contenitori fisici di dimensioni finite e fissate, detti **registri**
 - La posizione di un registro nell'insieme si chiama **indirizzo di memoria**
 - La dimensione di un registro si misura in numero di **bit**
- La memoria centrale di un computer è organizzata come un array di stringhe di bit, dette **parole** o **word**, aventi lunghezza m ($m = \text{LUNGHEZZA DI PAROLA}$)
 - gli m bit di una parola sono accessibili dal processore (in lettura/scrittura) mediante un'unica operazione
 - ogni parola è individuata da un indirizzo, cioè un intero compreso tra 0 e $N-1$ (SPAZIO DI INDIRIZZAMENTO), con $N = 2^{\text{(numero di bit usabili per ogni indirizzo)}}$

MEMORIE

- **Logicamente** si può pensare ad un memoria come ad una tabella a due colonne

Spazio di
Indirizzamento = 2^N

N bit = 32, 64		M bit = Lunghezza della Parola, con M= 8, 16, 32..	
Indirizzo		Dato	
0010101000101010		0101101001011010	
0010101011111001		0101111101011111	
0100011101010100		1010101001011111	
1010101010100010		1010101010101010	
.....		

OPERAZIONI SUI REGISTRI DI MEMORIA

- **Lettura**

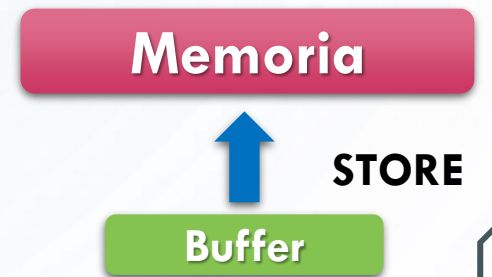
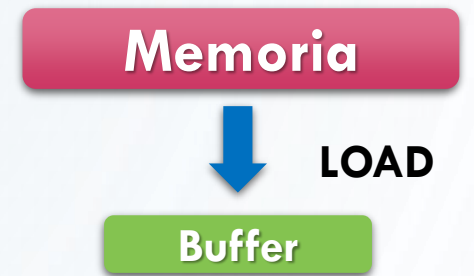
- Preleva l'informazione contenuta nel registro senza però distruggerla
- Anche detta LOAD

- **Scrittura**

- Inserisce una informazione nel registro eliminando quella precedente
- Anche detta STORE

- **Buffer**

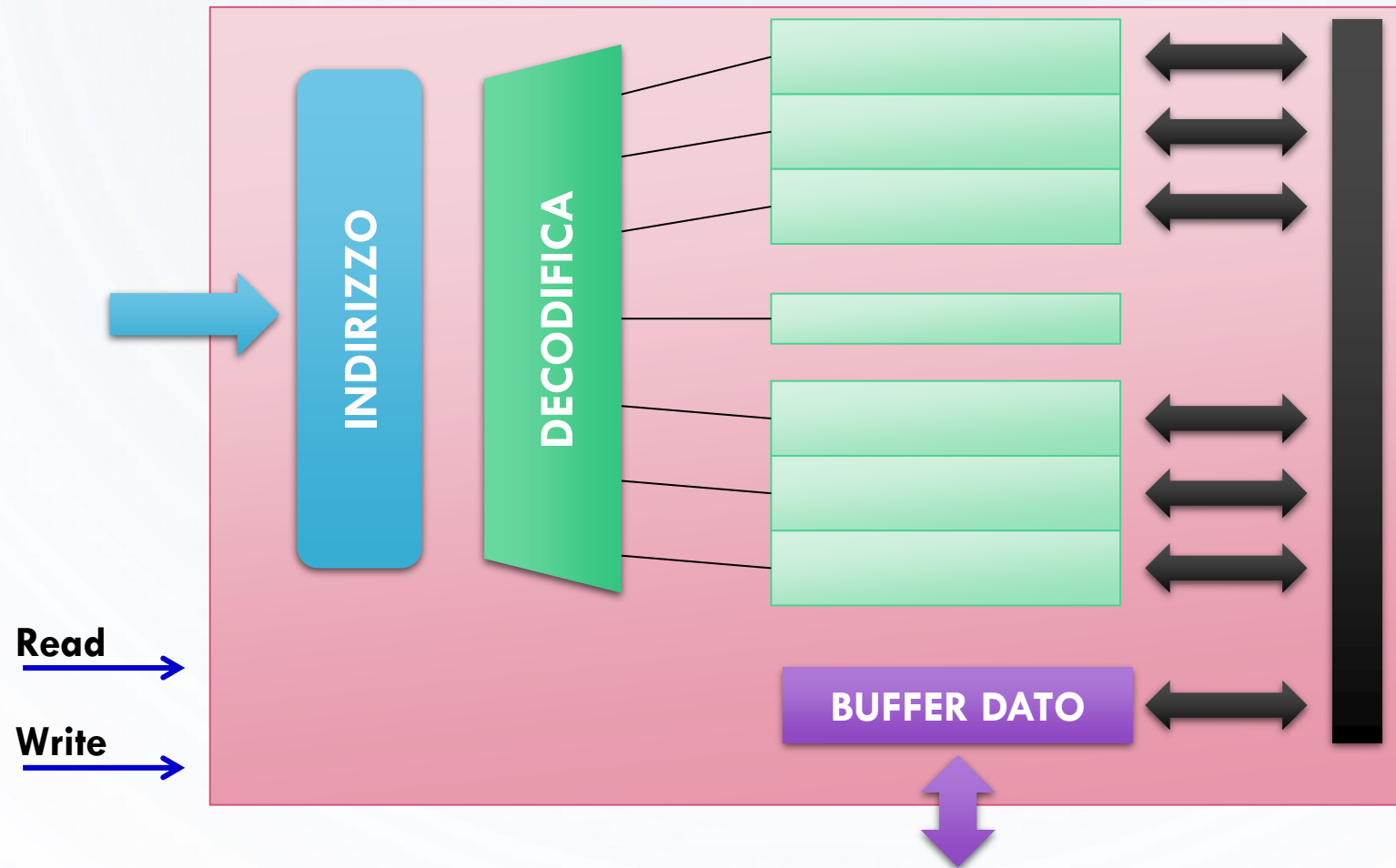
- Area di transito dei dati dalla CPU alla memoria e viceversa



FUNZIONAMENTO DELLE MEMORIE

- La CPU indica l'indirizzo del registro interessato dall'operazione
- La memoria decodifica l'indirizzo abilitando solo il registro ad esso corrispondente affinché:
 - Per una operazione STORE il dato del buffer sia copiato nel registro
 - Per una operazione LOAD il dato del registro sia copiato nel buffer
- I tempi di attuazione delle operazioni di LOAD e STORE dipendono da:
 - tecnologie usate per la costruzione delle memorie
 - modalità di accesso

LOAD E STORE: SCHEMI DI FUNZIONAMENTO



PRESTAZIONI DELLE MEMORIE

- Le prestazioni di un componente di memoria vengono misurate in termini di ***tempi di accesso***
- I tempi di attuazione delle operazioni di LOAD e STORE possono essere differenti e dipendono da:
 - Tecnologie usate per la costruzione delle memorie
 - Modalità di accesso
 - **LOAD**
 - *Tempo di accesso* = tempo che trascorre tra la selezione del registro di memoria e la disponibilità del suo contenuto nel registro di buffer
 - **STORE**
 - *Tempo di accesso* = tempo necessario alla selezione del registro e il deposito del contenuto del registro di buffer in esso

PRESTAZIONI VS COSTI

- Le memorie devono mostrare tempi di accesso adeguati alle capacità della CPU
 - Non devono introdurre ritardi quando essa trasferisce dati



Memorie con tempo di accesso bassi

vs

Costo componenti

CLASSIFICAZIONE MEMORIE

- Selezione di un registro di memoria

- **Casuale**

- Il tempo di accesso non dipende dalla posizione
 - Le memorie ad accesso casuale sono dette **RAM** (*Random Access Memory*)



- **Sequenziale**

- Il tempo di accesso dipende dalla posizione
 - Le memorie ad accesso sequenziale sono dette **SAM** (*Sequential Access Memory*)



CLASSIFICAZIONE MEMORIE

- **Memorie volatili**

- Memorie che perdono le informazioni in esse registrate quando il sistema viene spento
 - RAM e memorie elettroniche in generale

- **Memorie permanenti**

- Memorie che conservano le informazioni in esse registrate anche quando il sistema viene spento
 - Memorie di tipo magnetico, ottico e tutti i tipi di ROM

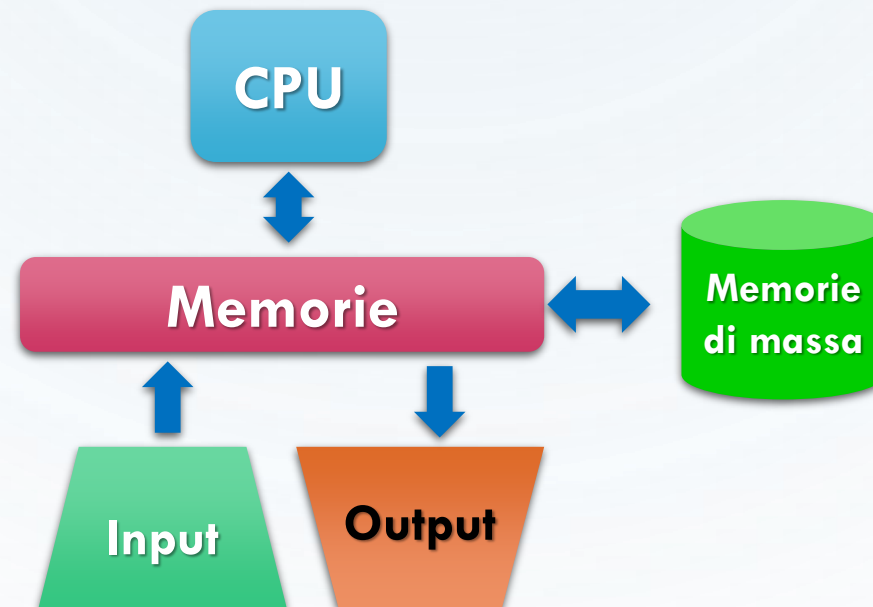
- **ROM** (*Read Only Memory*)

- Alcune memorie vengono realizzate in modo che sia possibile una sola scrittura di informazioni
 - Ad esempio, il BIOS (*Basic Input-Output System*) è un insieme di programmi che fornisce funzionalità di base per l'accesso all'hardware del computer e alle periferiche integrate sulla scheda madre, da parte del sistema operativo e degli altri programmi

MODELLO DI VON NEUMANN MODIFICATO

- **Memorie di massa**

- Memorie ausiliarie caratterizzate da una elevata capacità



MEMORIE DI MASSA



TRASFERIMENTO DATI

- **Memoria centrale ↔ CPU**

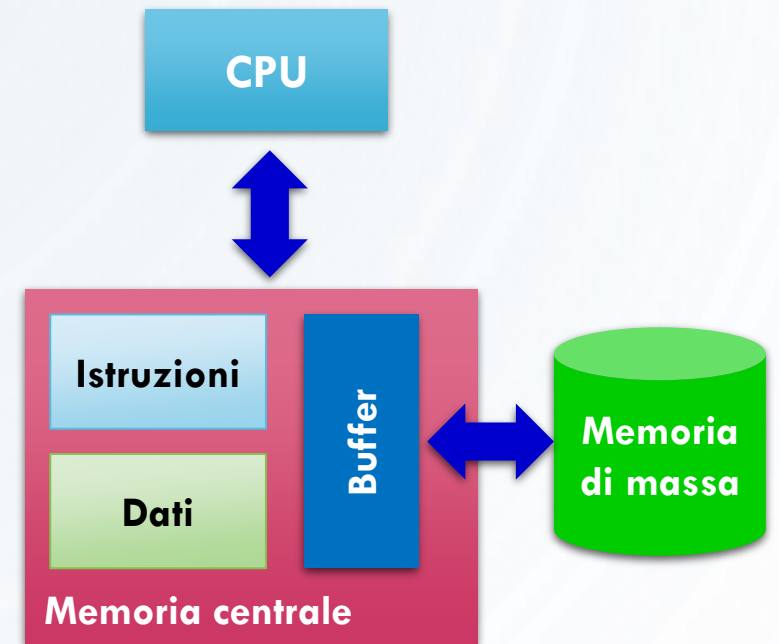
- Le informazioni contenute nella memoria centrale possono essere direttamente prelevate dalla CPU

- **Memoria di massa → Memoria centrale → CPU**

- Le informazioni contenute nella memoria di massa devono essere dapprima trasferite nella memoria centrale e successivamente elaborate

- **CPU → Memoria centrale → Memoria di massa**

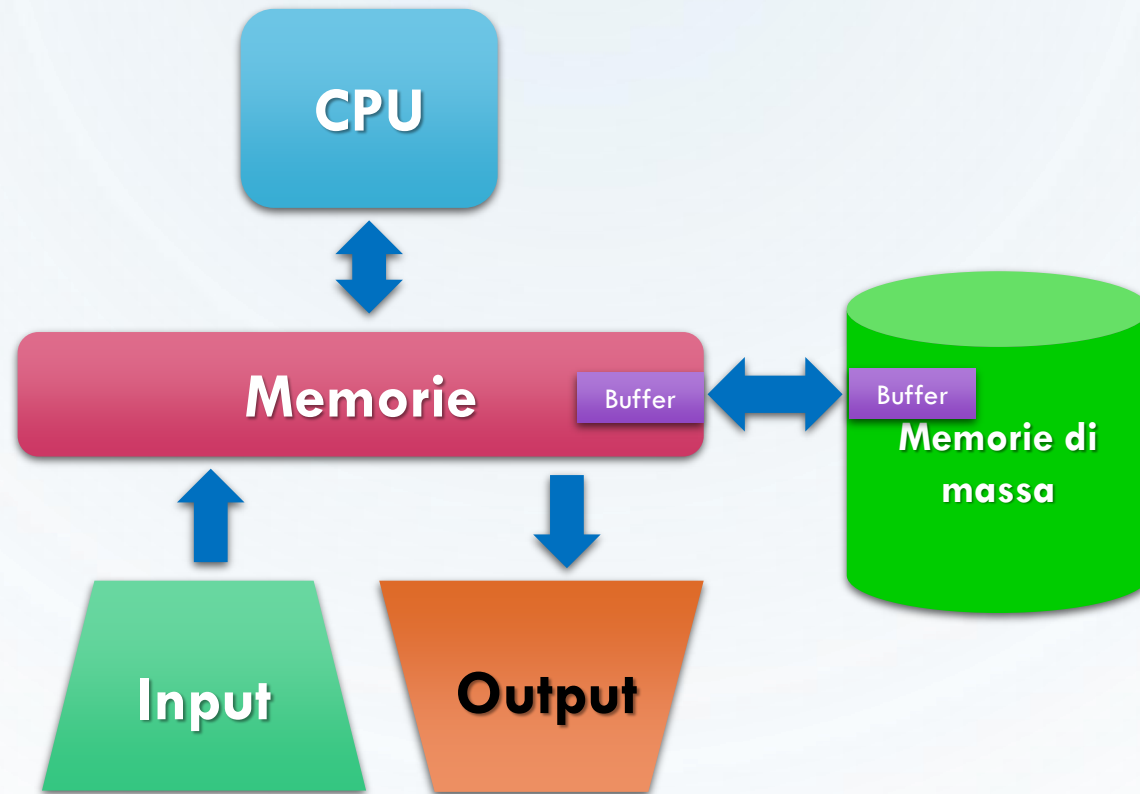
- Le informazioni prodotte dalla CPU devono essere depositate in memoria centrale per poi essere conservate nelle memorie di massa



BUFFER

- Le memorie di massa hanno **tempi di accesso maggiori** rispetto alle memorie centrali dovuti alle tecnologie impiegate per realizzarle
- Per ovviare a questa differenza di velocità si provvede facendo in modo che la memoria centrale contenga delle aree di accumulo → *buffer* interni alle memorie centrali
- **Buffer di input**
 - Ha il compito di accumulare dati in memoria ricevendoli da un dispositivo lento prima che la CPU provveda ad elaborarli
- **Buffer di output**
 - La CPU, molto più veloce, accumula i dati prodotti in un buffer di uscita prima di abilitarne il trasferimento
- Con i buffer si procede verso una *separazione* dei compiti tra i componenti del modello di von Neumann
 - Rendono possibile la cooperazione tra dispositivi caratterizzati da velocità di trattamento dati diverse tra loro

MODELLO DI VON NEUMANN MODIFICATO



DISPOSITIVI DI INPUT E OUTPUT

- **Unità di input**

- Tastiera, mouse, penna ottica, tavoletta grafica, scanner,...



- **Unità di output**

- Monitor, plotter, stampanti,...

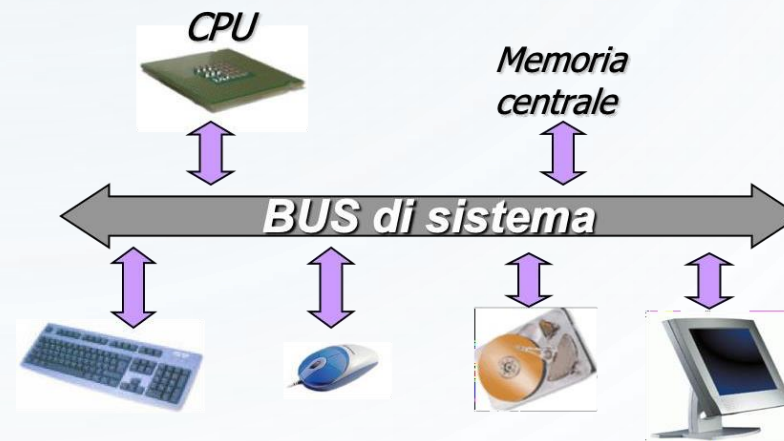


CANALE DI COMUNICAZIONE

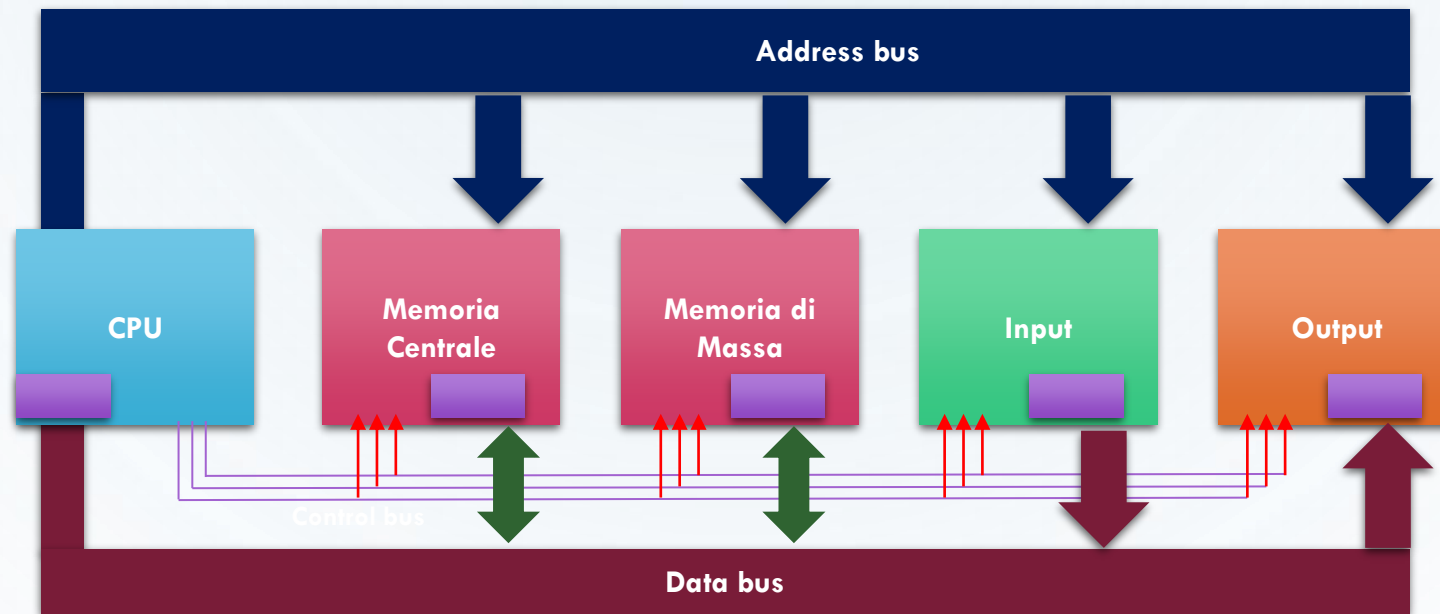
- **Obiettivo**
 - *permettere lo scambio di informazioni tra le varie componenti funzionali del calcolatore trasferimento dei dati e delle informazioni di controllo*
- **Due possibili soluzioni**
 - *collegare ciascun componente con ogni altro componente*
 - *collegare tutti i componenti a un unico canale (bus)*
- *L'utilizzo di un bus favorisce la modularità e l'espandibilità del calcolatore*

BUS

- **Canale di comunicazione** condiviso da più utilizzatori
 - Permette alla CPU di comunicare con la memoria e tutti i dispositivi di input ed output
- Collega due unità alla volta abilitandone una alla trasmissione e l'altra alla ricezione
 - Il trasferimento di informazioni avviene sotto il controllo della CPU
- Tipologie di bus
 - *Bus di Controllo (Control Bus)*
 - *Bus Dati (Data Bus)*
 - *Bus Indirizzi (Address Bus)*

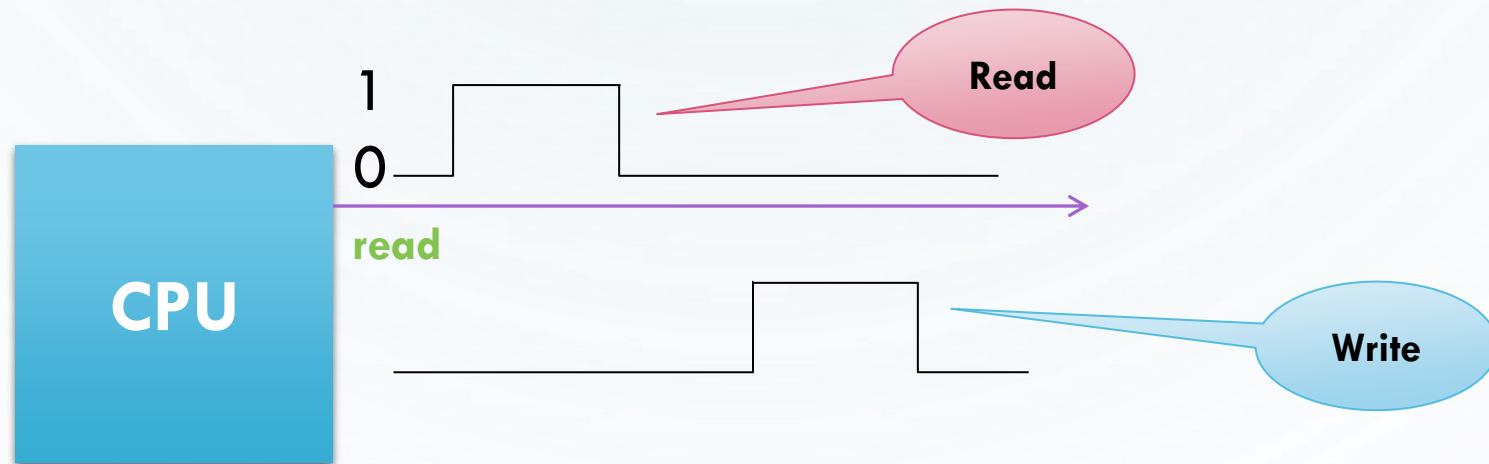


BUS



CONTROL BUS

- Serve alla CU per indicare ai dispositivi cosa devono fare
- Tipici segnali del Control Bus sono quelli di *read* e *write*
 - Usati dalla CU per indicare ai dispositivi se si tratta di un'operazione di lettura (read) dal dispositivo o di scrittura (write) sul dispositivo



DATA BUS

- Permette ai ***dati*** di fluire tra CPU e registro di memoria selezionato per operazioni di STORE e LOAD
- La CU controlla il flusso di informazioni con il mondo esterno abilitando il transito delle informazioni dalla memoria verso le risorse di output e viceversa da quelle di input
- Il funzionamento delle memorie di massa è simile a quello di un dispositivo che opera sia in input che in output

ADDRESS BUS

- Serve alla CU per comunicare l'**indirizzo** del dispositivo interessato da una operazione di lettura o scrittura
- Anche i *dispositivi di input/output* sono identificati da un indirizzo
- Tutti i componenti del sistema devono essere in grado di riconoscere sull'Address Bus il proprio indirizzo
 - Attraverso l'Address Bus la CU effettua la **selezione** del dispositivo a cui sono rivolti i comandi e i dati

A decorative graphic consisting of thin, dark blue lines and small circles, resembling a circuit board or a network diagram. The lines are of varying lengths and angles, connecting the circles in a non-linear fashion. The background is a light blue gradient.

-
- A decorative graphic consisting of stylized circuit lines and nodes. It features several vertical and diagonal lines of varying lengths, some ending in small circles, arranged in a way that suggests a network or circuit board layout. The lines are dark gray, and the circles are white with dark gray outlines.



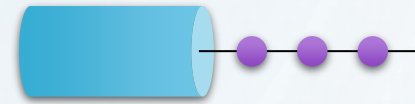
REGOLE PER LO SCAMBIO DI INFORMAZIONI

- La CPU è l'unico elemento che fornisce un **indirizzo** all'Address Bus
- Memorie e dispositivi di input ed output devono **“ascoltare”** l'Address Bus per attivarsi quando su di esso compare il proprio indirizzo identificativo
- La **memoria** si attiva quando viene riconosciuto l'indirizzo corrispondente ad uno dei registri di cui essa è composta
- Il **dispositivo** attivo deve interpretare i segnali del Control Bus per eseguire i comandi della CU
- Le memorie prelevano dati dal Data Bus o immettono dati in esso in funzione del **comando** impartito dalla CU
- I dispositivi di **input** possono solo immettere dati sul Data Bus
- I dispositivi di **output** possono solo prelevare dati dal Data Bus

LARGHEZZA DEL CANALE DI UN BUS

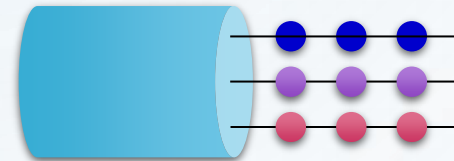
- **Bus seriale**

- Bus costituito da un solo filo
- Su di esso i bit transitano uno dietro l'altro



- **Bus parallelo**

- Bus costituito da n fili
- Su di esso transitano n bit alla volta
 - Ad es. 8, 32,...



- L'Address Bus ed il Data Bus sono paralleli e le **loro dimensioni caratterizzano i sistemi di calcolo**

IMPORTANZA DELLE DIMENSIONI DEI BUS

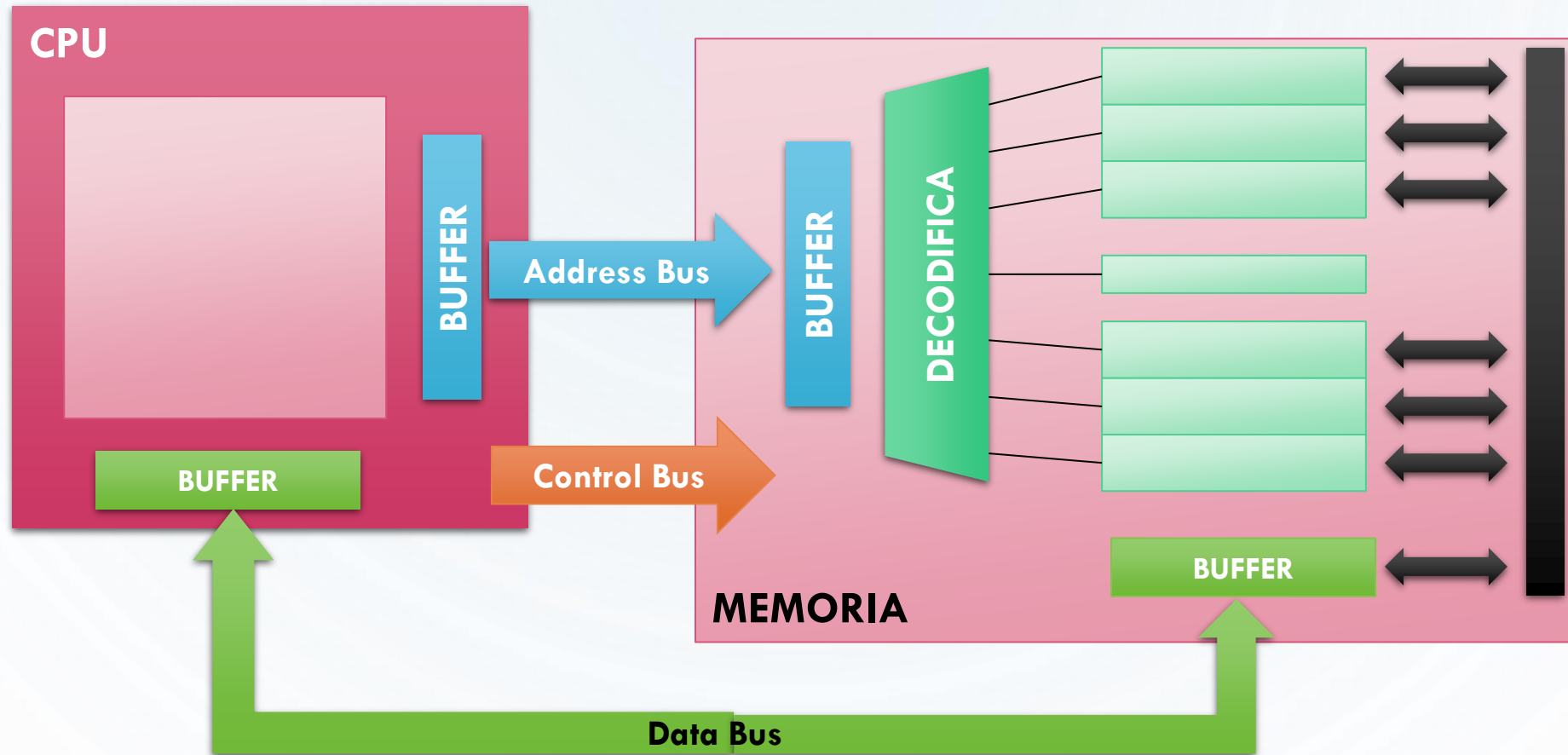
- **Capacità di indirizzamento della CPU**

- Capacità di gestire la dimensione della memoria e il numero di dispositivi di input ed output
- Corrisponde al numero di bit dell'**Address Bus**
- Con n bit un Address Bus consente di *selezionare* un registro tra 2^n

- **Velocità di scambio delle informazioni tra i dispositivi**

- Condizionata dalla dimensione del **Data Bus**
- Con m fili possono viaggiare *contemporaneamente* m bit

TRASFERIMENTO DATI TRA CPU E MEMORIA



CLOCK

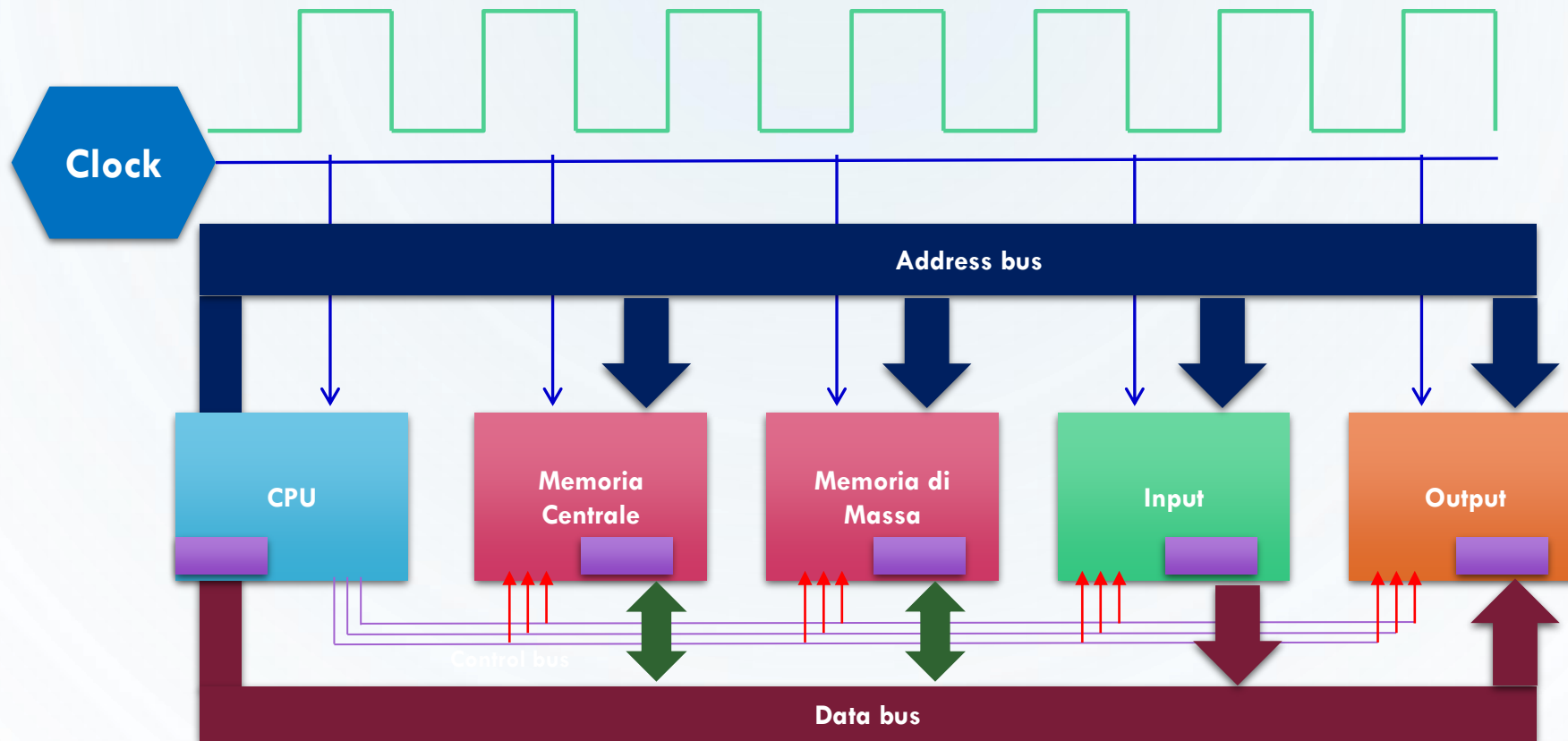
- I componenti del modello di von Neumann vengono coordinati dalla CU secondo sequenze prestabilite
- Ad ogni operazione della CU corrisponde una prefissata sequenza di attivazione dei diversi dispositivi
- **Clock**
 - Le attività di tutti i dispositivi non si svolgono casualmente ma vengono sincronizzate tra loro mediante un *orologio interno* che scandisce i ritmi di lavoro



FREQUENZA DEL CLOCK

- Il clock è un segnale periodico di **periodo fisso**
 - È un'onda quadra caratterizzata da un periodo T (detto ciclo) e da una frequenza f ($f=1/T$) misurata in Hertz (Hz)
- Esempio
 - Un clock composto da 10 cicli al secondo ha la frequenza $f = 10$ Hz e il periodo $T = 100$ ms
- Le attuali frequenze dei clock spaziano dai MHz ai GHz
 - 1 MHz corrisponde a un milione di battiti al secondo
 - 1 GHz corrisponde a un miliardo di battiti al secondo
- Il clock è un segnale che raggiunge tutti i dispositivi per fornire la cadenza temporale per l'esecuzione delle operazioni elementari

CLOCK



CLOCK E PRESTAZIONI

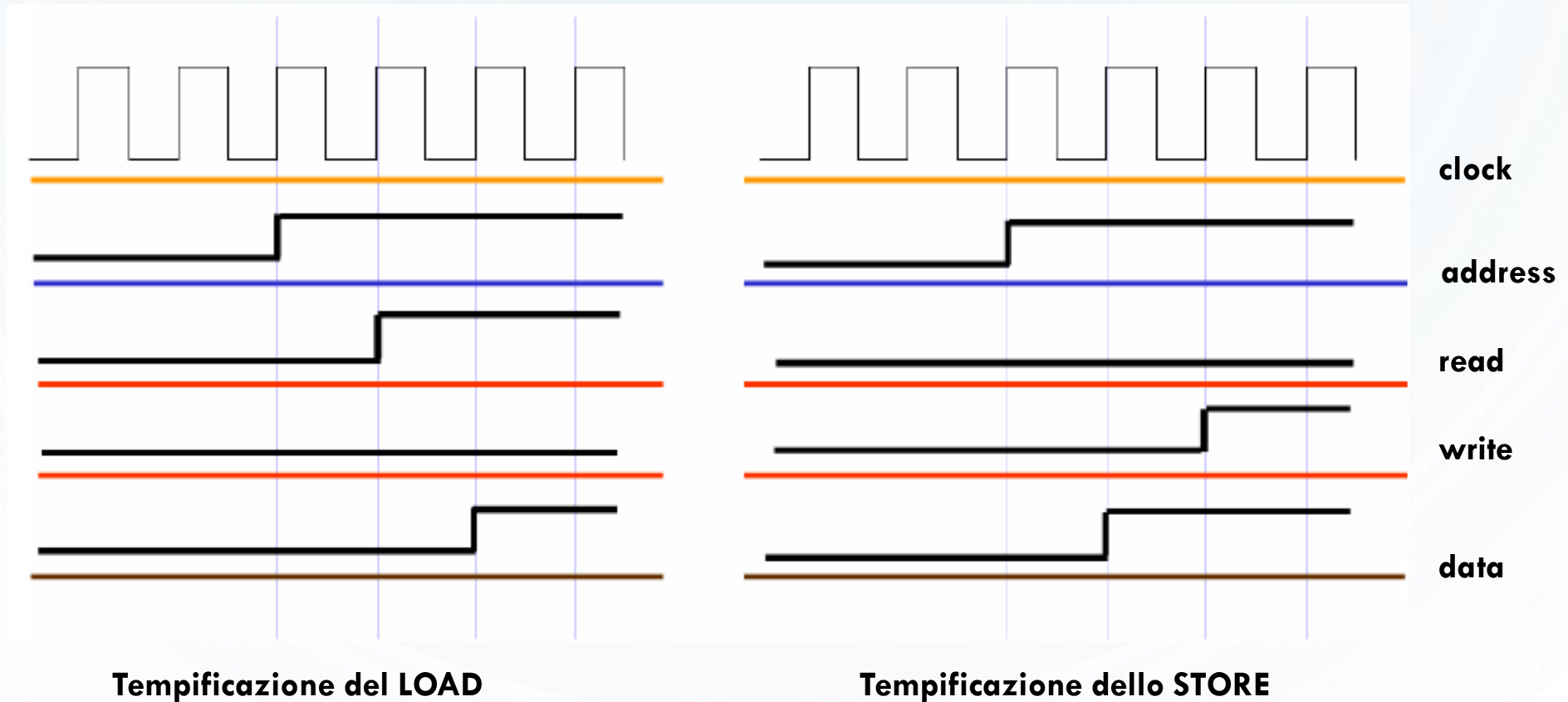
- Alla frequenza del clock è legato il **numero di operazioni** elementari che vengono eseguite nell'unità di tempo dalla CU
- Esempio
 - Se ad ogni ciclo di clock corrisponde esattamente l'esecuzione di una sola operazione
 - La frequenza del clock indica il numero di operazioni che vengono eseguite nell'unità di tempo dalla CU
 - Con un clock a 3 GHz, il processore è in grado di eseguire 3 miliardi di operazioni al secondo

CLOCK E PRESTAZIONI

- L'esecuzione di una operazione può richiedere ***più cicli di clock***
 - Per la complessità delle operazioni
 - Per la lentezza dei dispositivi collegati alla CPU
- La memoria centrale è realizzata mediante moduli che hanno prestazioni ***inferiori*** rispetto alla tecnologia utilizzata per costruire le CPU
 - Si realizzano quindi dei bus che rallentano la trasmissione di un fattore 10 rispetto al clock

ESEMPI DI TEMPIFICAZIONE

- Ipotesi semplificativa
 - In un solo ciclo di clock è possibile leggere/scrivere il dato dal Data Bus sulla memoria



DOMANDE, DUBBI, PERPLESSITÀ

