

Architettura di **von Neumann**

(Budapest, 28 dicembre 1903 – Washington, 8 febbraio 1957)

Uno dei primi modelli logico-funzionali di un elaboratore fu proposto nel 1946 dal matematico Ungherese **Johann von Neumann**.

Ancora oggi i calcolatori seguono questa organizzazione (o struttura) di base.

Lo schema si basa su sei componenti fondamentali:

- 
1. **A.L.U.** (Arithmetic Logic Unit)
 2. **Unità di controllo**
 3. **Memoria centrale o principale**
 4. **Unità di INPUT**
 5. **Unità di OUTPUT**
 6. **BUS**
- C.P.U**
(Central Processing Unit)

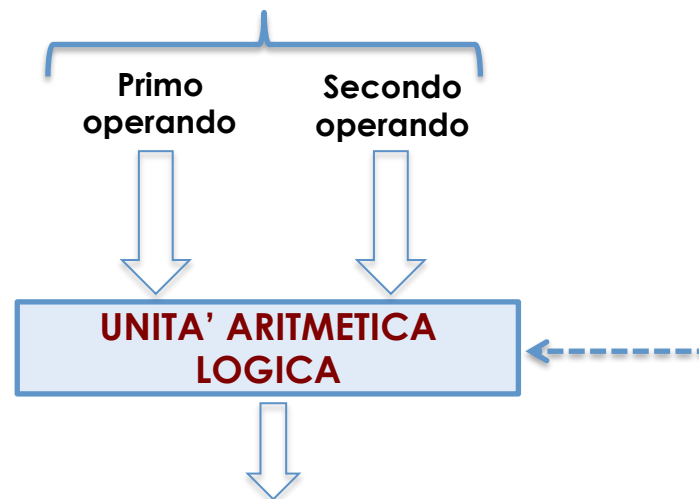
L'**Unità Aritmetica Logica** ha la funzione di eseguire.

- ❑ Semplici operazioni aritmetiche
- ❑ Operazioni logiche e di confronto

Le sue **Interazioni** con altri componenti riguardano

- ❑ **Memoria centrale** da cui legge i dati da elaborare e dove scrive i risultati
- ❑ **Unità di controllo** che guida lo svolgimento di tutti i calcoli

Dati letti dalla **MEMORIA CENTRALE**



Informazioni provenienti dall'**UNITA' DI CONTROLLO** per selezionare il tipo di operazione da eseguire

Risultati scritti nella **MEMORIA CENTRALE**

La **memoria centrale** o **principale** ha la funzione di contenere memorizzati in forma binaria:

- ❑ le istruzioni dei programmi eseguite nell'unità di controllo con il supporto per i calcoli dell'Unità Aritmetica Logica;
- ❑ i dati da elaborare;
- ❑ risultati intermedi e quelli finali.

Una memoria può essere considerata astrattamente come una sequenza finita di **celle** (*unità indirizzabile più piccola*) in cui ogni cella contiene una sequenza finita di bit (0,1). Normalmente i bit sono gestiti a gruppi di otto, detti **byte**.

Si definisce **parola** (word) una sequenza di byte. Le lunghezze di parola più comuni sono: 1, 2, 4, 8 byte

01110101	01000101	11100100	01010101
----------	----------	----------	----------

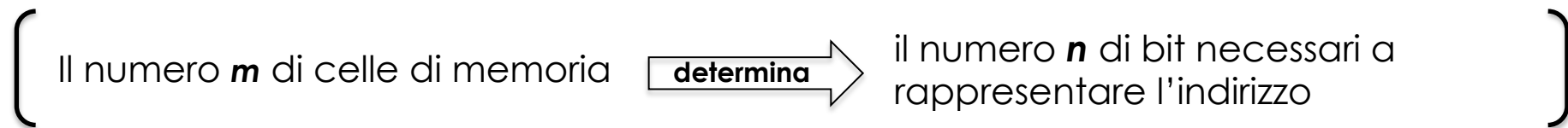
parola a 32 bit = 4 Byte

01110101	01000101	11100100	01010101	01110101	01000101	11100100	11010100
----------	----------	----------	----------	----------	----------	----------	----------

parola a 64 bit = 8 Byte

Ogni cella oltre ad avere un proprio contenuto ha un proprio **indirizzo** che serve per identificare in modo univoco la cella e di conseguenza l'informazione in essa contenuta

Anche **l'indirizzo** è rappresentato in forma binaria



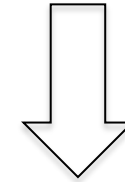
Infatti, se è noto il numero m di celle, il numero n di bit necessari a rappresentare l'**indirizzo** è possibile ricavarlo dalla formula $2^n = m$

VICEVERSA



In una memoria composta da $m = 32$ celle

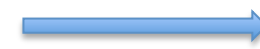
01110101		11100100	



$$2^n = m \implies 2^n = 32 \implies n = 5$$

1 1 1 0 0 1 0 0

Indirizzo



0 0 0 1 0

0 1 1 1 0 1 0 1

Indirizzo



0 0 0 0 0

La memoria centrale è detta anche **memoria ad accesso diretto** o **RAM** in quanto l'accesso alla singola cella può avvenire direttamente per mezzo del suo indirizzo fisico.

La traduzione letterale dall'inglese RAM (Random Access Memory) Memoria ad Accesso Casuale può essere fuorviante, in quanto un accesso "casuale" alla memoria sarebbe del tutto inutile.

La memoria RAM appartiene alla categorie di **Memorie Volatili**, ovvero memorie che per preservare il loro contenuto devono essere alimentate con energia elettrica continuamente..

Un'interruzione dell'alimentazione comporta la perdita del contenuto della RAM

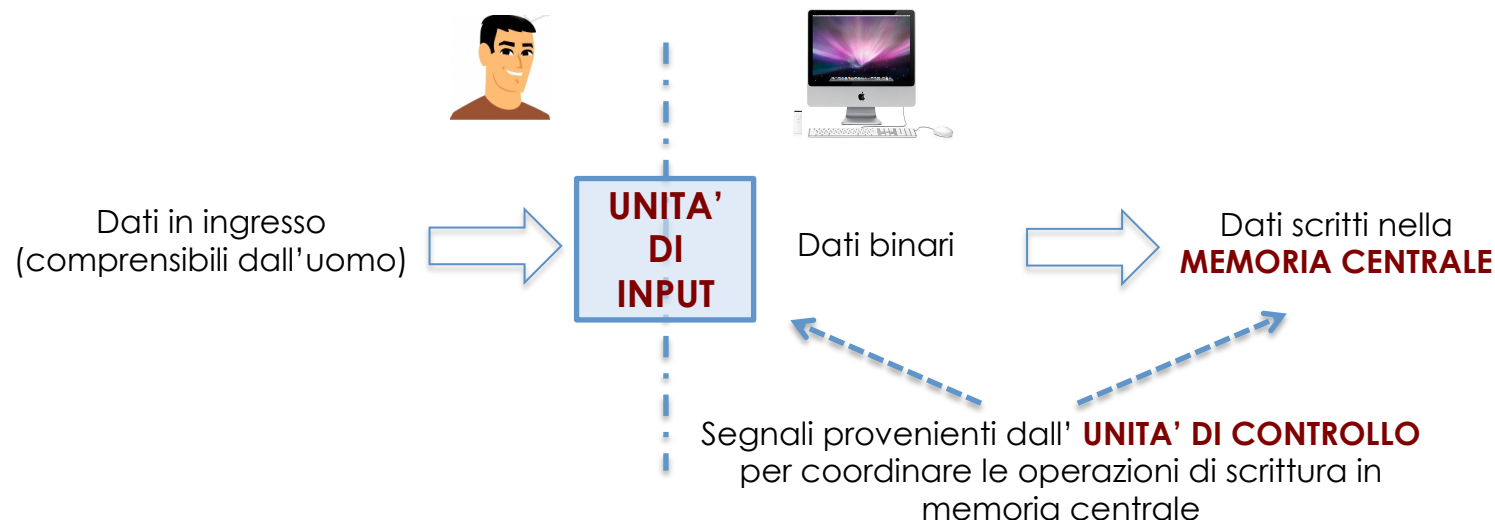
Una **unità di input** ha la funzione di trasferire i dati e le istruzioni dei programmi introdotti da un operatore umano all'interno della memoria principale dell'elaboratore. La sua funzione principale:

❑ è quella di codificare i dati da una forma comprensibile all'uomo, in quella binaria.

Le sue **Interazioni** con altri componenti riguardano

❑ **Memoria centrale**, in cui si scrivono i dati in ingresso codificati in binario

❑ **Unità di controllo** che ha la funzione di coordinare le operazioni di scrittura in memoria.

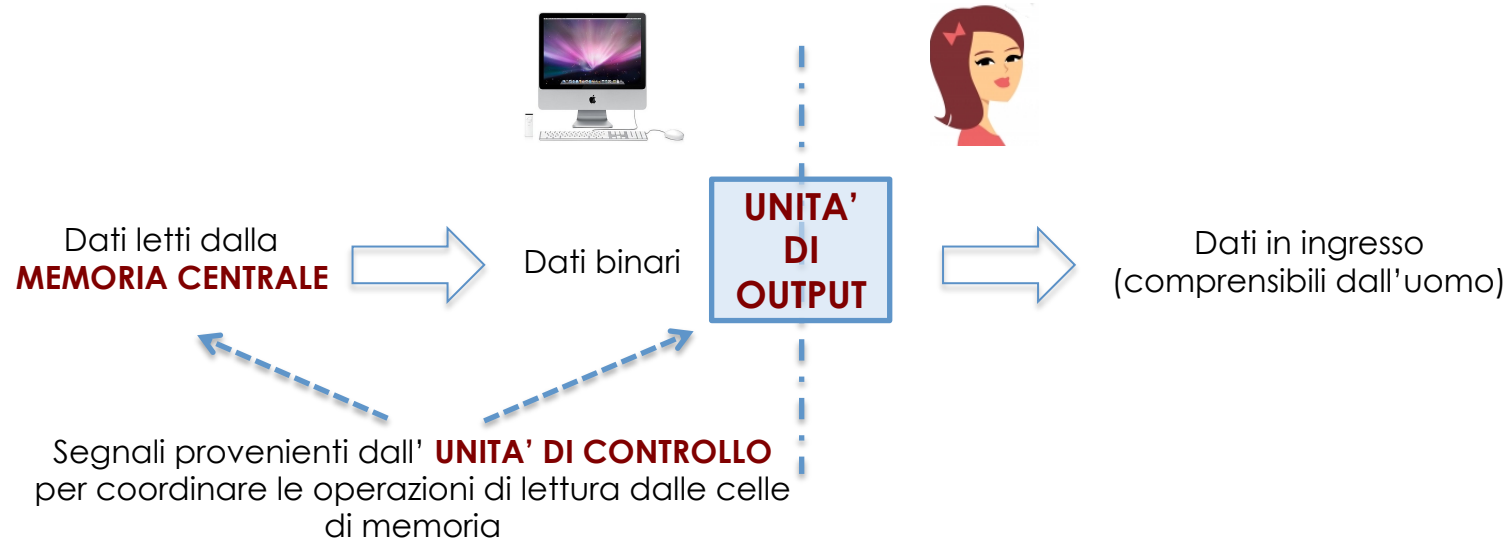


Una **unità di output** svolge la funzione opposta a quella di ingresso. La sua funzione principale è:

- ❑ quella di decodificare i dati dal formato binario in simboli comprensibili all'uomo.

Le sue **Interazioni** con altri componenti riguardano

- ❑ **Memoria centrale**, dalla quale si leggono i dati da presentare all'esterno del calcolatore
- ❑ **Unità di controllo** che ha la funzione di coordinare le operazioni di lettura dalle celle in memoria.



L' **unità di controllo** svolge le seguenti funzioni:

- ❑ Sorvegliare e coordinare tutte le attività delle unità presenti nel sistema calcolatore, mediante opportune informazioni binarie di controllo;

Per effettuare una singola operazione , l'unità di controllo deve:

- ① **Leggere** le celle di memoria che contengono i dati (accede tramite indirizzo);
- ② **Leggere** la cella di memoria che contiene l'istruzione (vi accede tramite indirizzo) e **interpretare** (decodificare) l'istruzione per comprendere quale operazione deve svolgere
- ③ **Far eseguire** l'operazione mediante le altre unità del calcolatore
- ④ **Scrivere** il valore del risultato nella cella di memoria

Se si tratta dell'istruzione per l'addizione di due dati posti in memoria centrale, l'unità di controllo deve nell'ordine:

- ① **Leggere** i due dati nelle celle di memoria;
- ② **Interpretare** (decodificare) l'istruzione di somma
- ③ **Trasferire** i dati letti nell'unità aritmetica logica
- ④ **Attendere** che l'ALU abbia terminato l'operazione degli addendi
- ⑤ **Trasferire** il risultato dell'operazione in una nuova cella di memoria

Durante questa fase, l'unità di controllo emetterà le informazioni necessarie per coordinare la memoria principale e l'ALU

Tutti i componenti di un elaboratore hanno la necessità di scambiarsi informazioni. Tale scambio avviene tramite il **Bus di sistema** e rappresenta uno degli aspetti più critici di un elaboratore.

Nel modello di Von Neumann si distinguono tre bus distinti:

- ❑ il **bus dei dati**: bidirezionale, attraverso il quale i dati viaggiano da e verso la CPU



- ❑ il **bus di indirizzi**: in cui i dati viaggiano solo dalla CPU verso gli altri componenti



- ❑ il **bus di segnali di controllo**: anche in questo caso i dati viaggiano solo dalla CPU verso gli altri componenti



- ❑ il **bus dei dati**: E' il BUS sul quale viaggiano le informazioni. E' usufruibile da tutti i componenti del sistema, sia in scrittura che in lettura. E' BIDIREZIONALE
- ❑ il **bus di indirizzi**: E' il bus attraverso il quale la CPU decide in quale indirizzo andare a scrivere o leggere le informazioni; sia le celle di memoria (RAM) sia le periferiche di I/O (Input/Output) sono infatti divise in zone, ognuna delle quali ha un dato indirizzo. Dopo aver comunicato l'indirizzo tramite il BUS, la scrittura o lettura avviene normalmente tramite il BUS dati. Naturalmente questo BUS è fruibile in scrittura solo dalla CPU ed in lettura da gli altri componenti, in quanto tramite questo BUS viene dato solo l'indirizzo di cella, che è deciso dalla CPU. E' MONODIREZIONALE
- ❑ il **bus di segnali di controllo**: Il bus controlli è un insieme di collegamenti il cui scopo è coordinare le attività del sistema; tramite esso, la CPU può decidere quale componente deve scrivere sul bus dati in un determinato momento, quale deve leggere l'indirizzo sul bus indirizzi, quali celle di memoria devono scrivere e quali invece leggere, etc. Infatti la memoria e tutti gli altri componenti comunicano con la CPU attraverso un unico bus condiviso; questo significa che senza un controllo da parte della CPU si verrebbero a creare dei conflitti e delle collisioni.-

Per eseguire un programma, l'unità di controllo conosce di volta in volta l'indirizzo della cella che contiene l'istruzione in corso di esecuzione.

Allo stato attuale, l'unità di controllo e quella aritmetica logica (CPU) sono realizzate in un unico componente elettronico digitale, denominato **microprocessore** o **processore**

Attualmente, un singolo microprocessore è un componente elettronico digitale che occupa un'area di pochi centimetri quadrati



Avendo esaminato separatamente le due unità che compongono la CPU (**unità di controllo e unità aritmetica logica**) possiamo affermare che il ciclo della CPU si divide in due fasi, ognuna delle quali comporta più passi:

❑ **Fase di istruzione** o **fetch** dove la CPU

① **Legge** dalla memoria l'istruzione da eseguire

② **Decodifica** o **interpreta** l'istruzione per comprendere quale operazione deve svolgere

❑ **Fase di esecuzione** dove la CPU

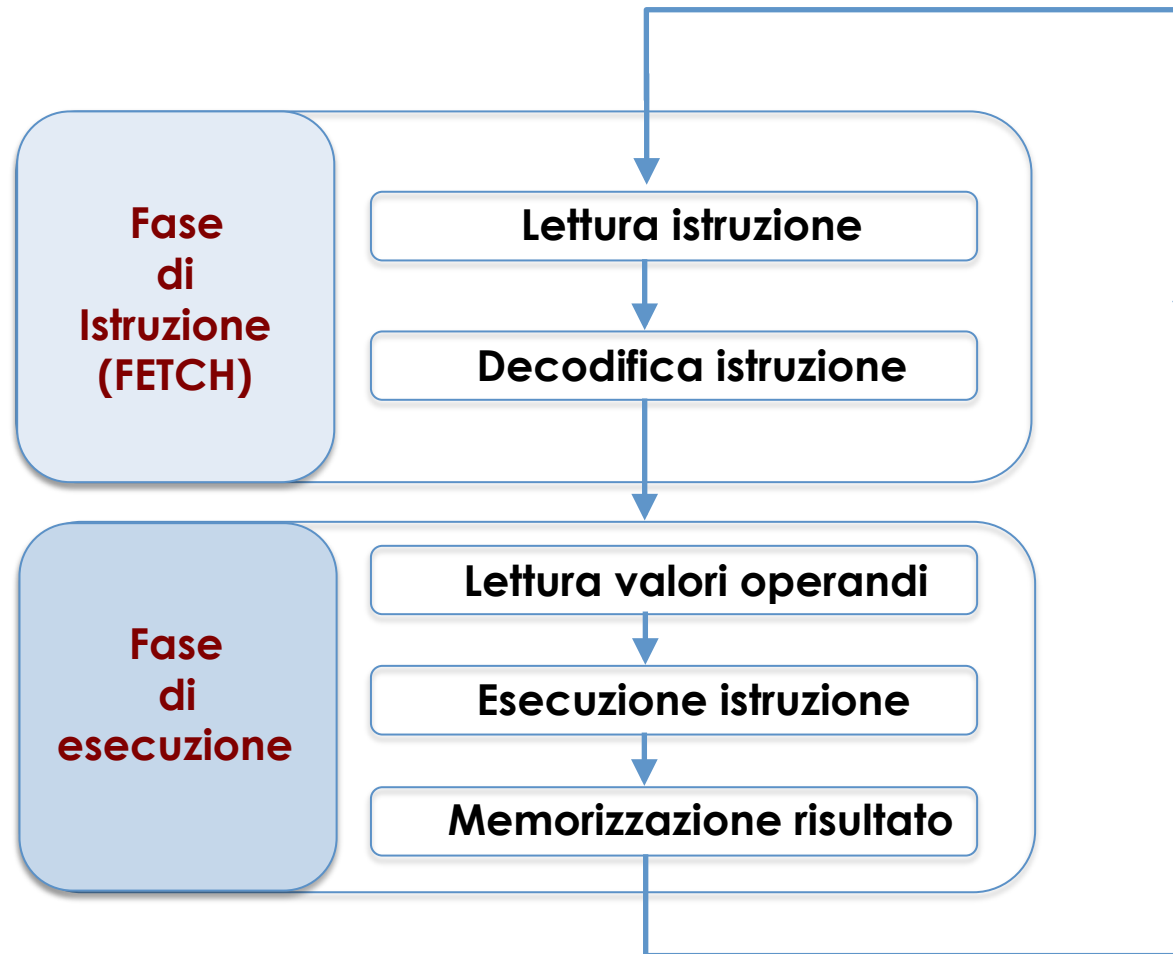
① **Legge** dalla memoria gli eventuali operandi dell'istruzione da eseguire

② **Esegue** l'istruzione;

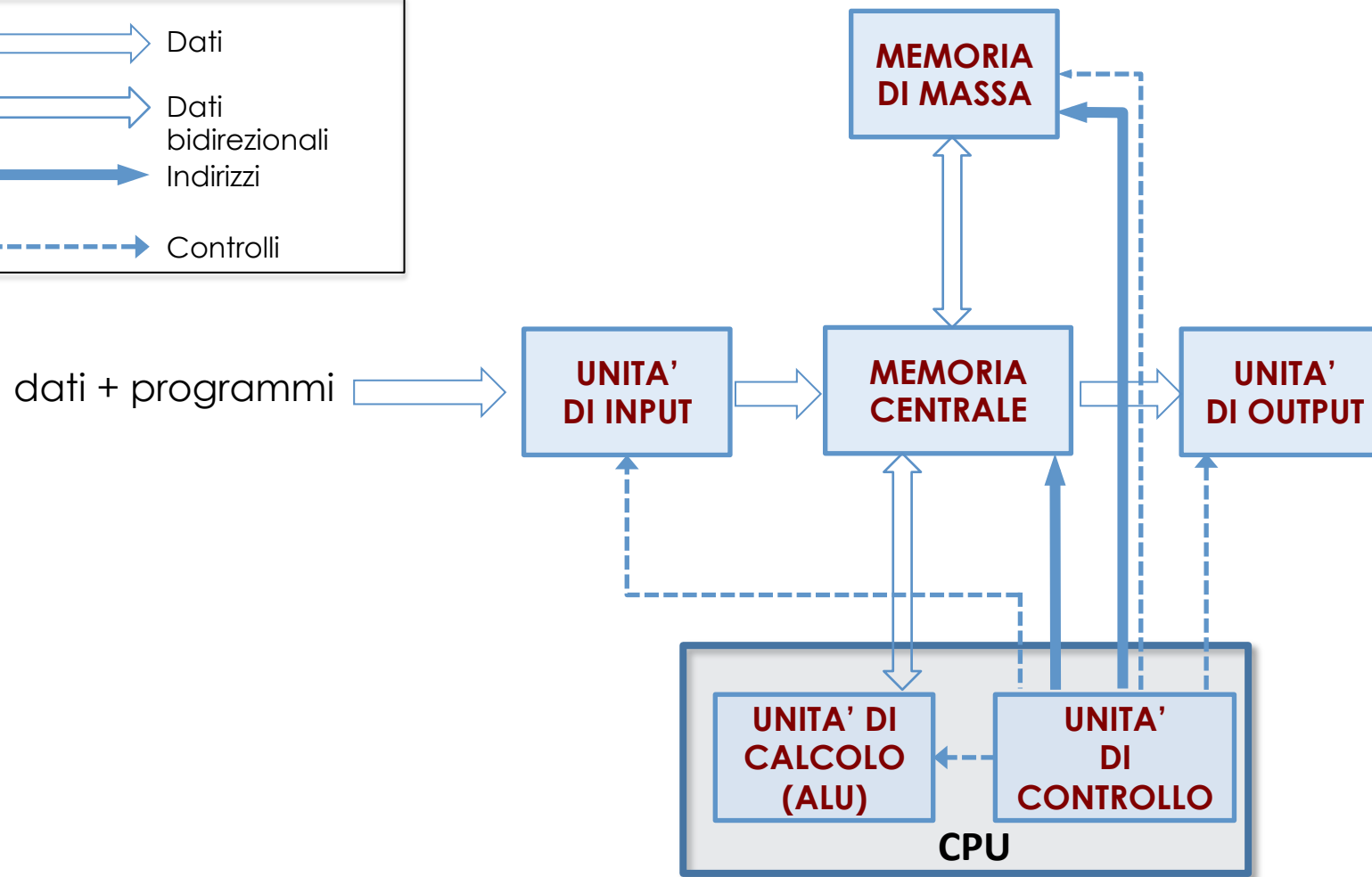
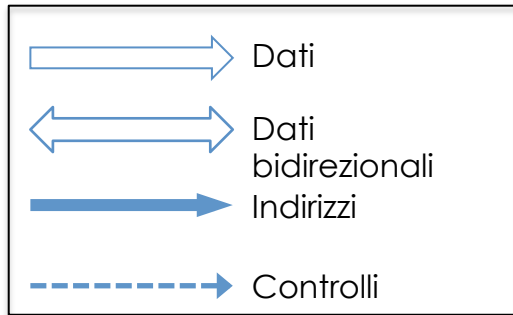
③ **memorizza** l'eventuale risultato

Al termine della fase di esecuzione la CPU riprende il suo ciclo a partire dalla successiva istruzione da eseguire

Il modello proposto da von Neumann è quindi un **ciclo strettamente sequenziale** nell'esecuzione delle istruzioni.



TIPO D'INFORMAZIONE SCAMBIATA



La **memoria ROM** (*Read Only Memory*), ovvero memoria a sola lettura, è una tipologia di memoria non volatile in cui i dati sono memorizzati nella sua fase di costruzione e non possono essere modificati successivamente.

Nella **memoria ROM**, sono residenti i programmi fondamentali per l'avvio del computer. Non si può intervenire per scrivere, memorizzare o cancellare i dati. Non è di grandi dimensioni.

Il contenuto della ROM prende il nome di **firmware** perché è un software con la caratteristica, tipica dell'hardware, di non essere modificabile nel corso del funzionamento normale.

La memoria ROM, è contenuta nella scheda madre



La **memoria secondaria** o di **massa** ha la funzione di memorizzare in modo permanente programmi e dati rappresentati in forma binaria

A tal fine si utilizzano componenti non solo di tipo elettronico, ma anche magnetico oppure ottico.

Una memoria secondaria ha un modello simile a quello di una memoria centrale, costituito da un insieme di celle, individuabili mediante un indirizzo unico

In genere le **operazioni** disponibili in una memoria sono:

- Scrittura di un dato in una locazione di memoria;
- Lettura del contenuto di una cella

Le sue **Interazioni** con altri componenti riguardano

- ❑ **Memoria centrale** nella quale si scrivono dati e/o programmi; viceversa dalla memoria centrale si può scrivere in quella secondaria per archiviare, ad esempio i risultati di un elaborazione.

- ❑ **Unità di controllo** che guida tutti i trasferimenti tra la memoria centrale e quella secondaria, impostando i segnali di scrittura/lettura e gli indirizzi delle celle di entrambe memorie

A differenza della memoria centrale (RAM) la memoria di massa è una **memoria non volatile**, ovvero è in grado di mantenere l'informazione anche quando non viene alimentata







	Memoria di massa	Memoria centrale (RAM)
Conservazione del dato	Memoria non volatile	Memoria volatile
Velocità di accesso al contenuto	Minore	Maggiore
Capacità di immagazzinamento dati	Maggiore	Minore




1 Byte = 8 bit



Simbolo	Byte	Fattore moltiplicativo	Valore approssimativo
1 KB (kilobyte)	1.024	2^{10}	mille
1 MB (megabyte)	1.048.576	2^{20}	un milione
1 GB (gigabyte)	1.073.741.824	2^{30}	un miliardo
1 TB (terabyte)	1.099.511.628.000	2^{40}	un trilione
1 PB (petabyte)	1.125.899.906.842.624	2^{50}	un quadrilione
1 EB (exabyte)	1.152.921.504.606.846.976	2^{60}	un quintilione
1 ZB (zettabyte)	1.180.591.620.717.411.303.424	2^{70}	un sestilione
1 YB (yottabyte)	1.208.925.819.614.629.174.706.176	2^{80}	un settilione

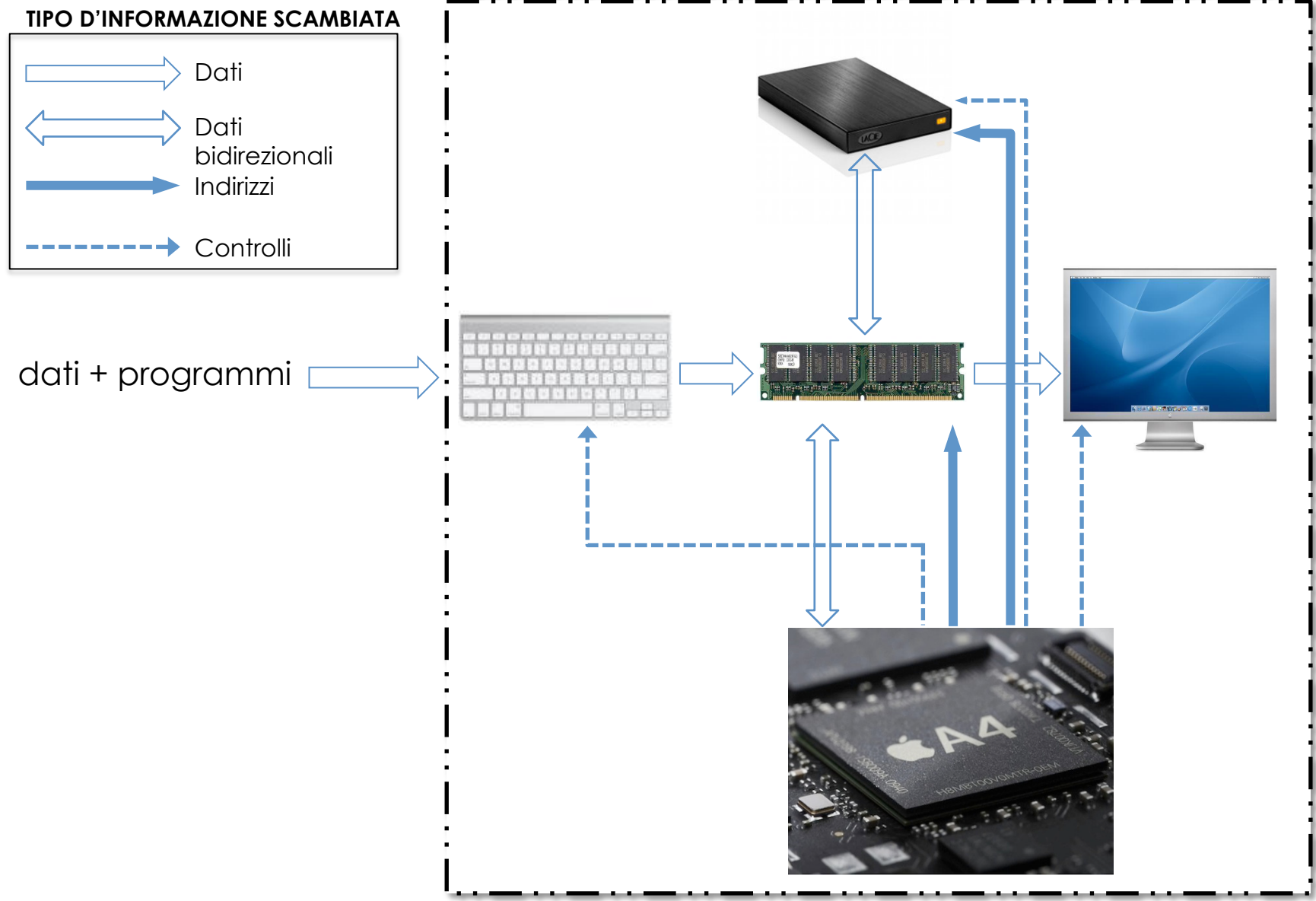
Esempio

Quando diciamo che una memoria è in grado di immagazzinare 10 GB di dati, significa che può memorizzare **circa** 10 miliardi di singoli byte.

Nome unità	Tipo	Descrizione	Immagine
Tastiera	Unità di INPUT	Quando l'operatore preme un tasto provoca l'invio nella memoria centrale di un dato binario che rappresenta il simbolo (cifra, lettera o carattere)	
Mouse	Unità di INPUT	Unità di puntamento impiegata in molti programmi per agevolare l'immissione dei dati.	
Unità video o monitor	Unità di OUTPUT	L'unità permette di visualizzare sia i dati in ingresso, digitati dall'operatore con la tastiera, sia i risultati dell'elaborazione. Questa unità decodifica i dati dal binario in un formato comprensibile all'uomo	
Stampante	Unità di OUTPUT	Permette di presentare i risultati delle elaborazioni stampandoli su fogli di carta.	
Scanner	Unità di INPUT	Permette di trasformare una qualsiasi immagine in una sequenza di dati codificati in binario che vengono archiviati nella memoria centrale, pronti per essere elaborati da applicativi.	
Hard Disk	Unità di INPUT/OUTPUT (memoria di massa)	Consente di archiviare in modo permanente, enormi quanti di dati e di programmi. Quest'ultimi possono poi essere richiamati in memoria centrale dalla CPU per essere eseguiti.	

Nome unità	Tipo	Descrizione	Immagine
Unità per floppy disk	Unità di INPUT/OUTPUT (memoria di massa)	Le sue funzioni sono del tutto analoghe all'hard disk, con le differenze in capacità (decisamente inferiori all'HD) e portabilità rispetto all'HD	
Unità ottica per CD/DVD	Unità di INPUT/OUTPUT (memoria di massa)	Unità ottica in grado di memorizzare grandi quantità di dati binari. Anche un cd/DVD può essere rimosso per essere usato da più elaboratori.	
Scheda di rete	Unità di INPUT/OUTPUT (interfaccia di periferica)	Permette di inviare (e ricevere) dati binari da un'altra scheda di rete installata su un altro computer. Due o più computer dotati di schede di rete e cavi possono comunicare realizzando una rete locale.	
USB Flash Drive	Unità di INPUT/OUTPUT (memoria di massa)	è una memoria di massa portatile di dimensioni molto contenute (qualche centimetro in lunghezza e intorno al centimetro in larghezza) che si collega al computer mediante la comune porta USB.	
Modem	Unità di INPUT/OUTPUT	Svolge funzioni analoghe alla scheda di rete, ma in questo caso i dati vengono comunicati a un altro modem anche a grandi distanze. Più computer dotati di modem realizzano una rete geografica.	
Microprocessore	CPU	Rappresenta il "cervello" del sistema di elaborazione. Allo stato attuale, può essere realizzato con un unico componente elettronico denominato microprocessore . Comunica con le altre unità scambiando informazioni.	

Nome unità	Tipo	Descrizione	Immagine
RAM	Memoria centrale	La memoria centrale degli elaboratori ha la funzione di contenere (in forma binaria) i dati e le informazioni per i processi di elaborazione. E' realizzata con componenti elettronici in grado di memorizzare i dati mediante segnali elettrici digitali.	
ROM	Memoria contenente il firmware	La memoria ROM è una memoria non volatile all'interno della quale sono memorizzati (dal costruttore) i dati necessari all'avvio del calcolatore. Tutto il contenuto della memoria è detto più propriamente firmware . La memoria ROM, è contenuta nella scheda madre	



Un **programma** è una specifica serie di operazioni che l'elaboratore deve svolgere

Un **linguaggio di programmazione** è un linguaggio artificiale che permette di esprimere un programma in una forma che sia comprensibile sia al programmatore (in genere umano), sia all'esecutore (in genere un elaboratore)

Il linguaggio è detto **artificiale** perché è stato inventato apposta per comunicare istruzioni al computer

Si definisce **Algoritmo** una sequenza di passi che portano alla realizzazione di un compito

Le caratteristiche che un algoritmo deve possedere sono:

- ❑ **non ambiguità** : i passi devono essere univocamente interpretabili dall'esecutore
- ❑ **eseguibilità**: l'esecutore deve essere in grado di eseguire ogni passo con le risorse di cui dispone
- ❑ **essere finito**: l'esecuzione dell'algoritmo deve terminare in un tempo finito

Un algoritmo può essere espresso in un **linguaggio naturale**, ossia sviluppato con l'evoluzione del genere umano, o tramite qualche altro **linguaggio artificiale**

Un **algoritmo** viene implementato mediante un programma

Esempio di algoritmo

1. Abbiamo gli ingredienti (pollo, olio, mandorle, cipolla, sale, pepe, etc.);
2. Seguiamo la *ricetta* (70 gr di mandorle, 300 gr di cipolla, 30 gr di sale, etc);
3. Serviamo il piatto a tavola.

Esempio di non algoritmo

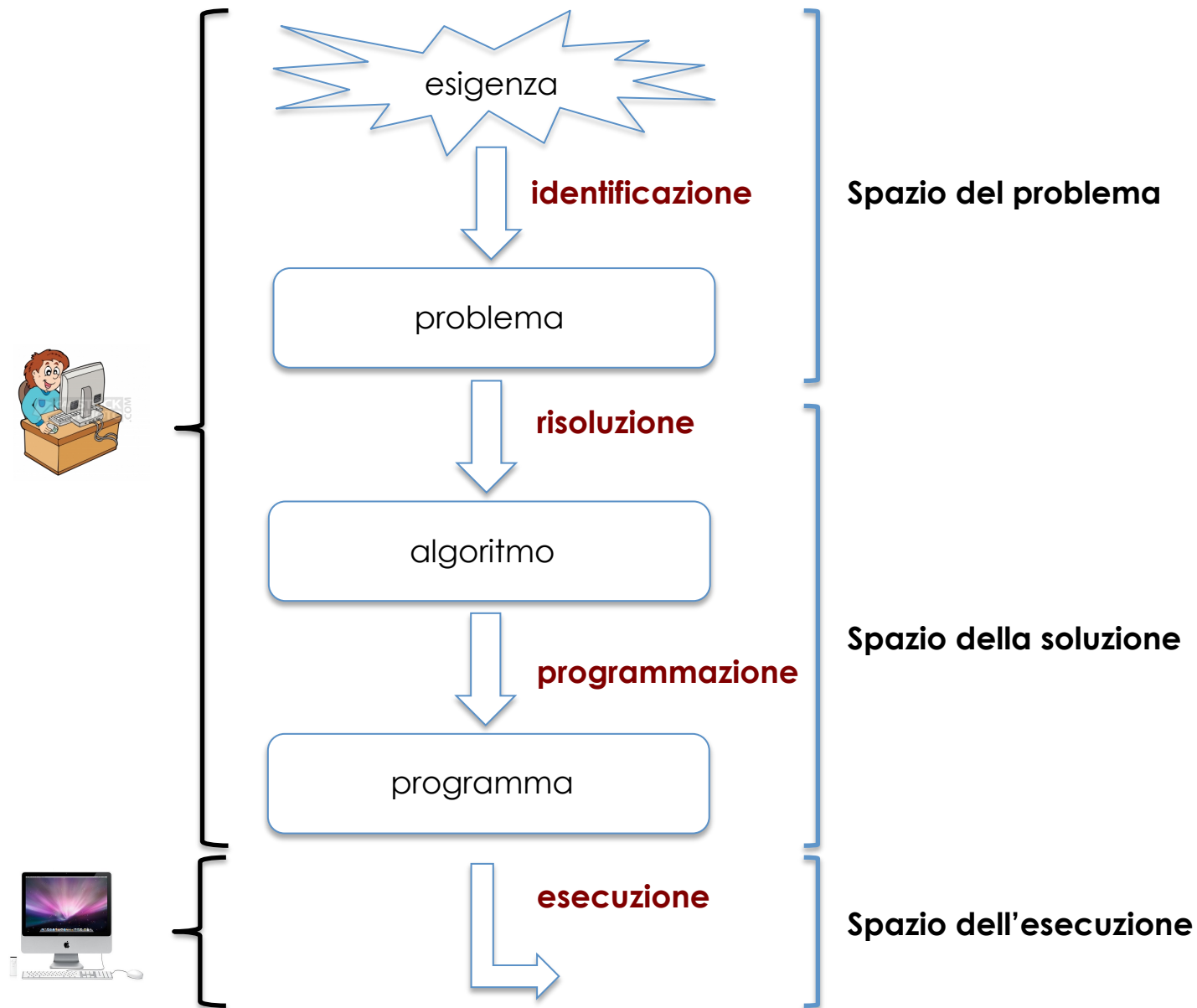
1. Abbiamo gli ingredienti (pollo, olio, mandorle, cipolla, sale, pepe, etc.);
2. Seguiamo la *ricetta* (una bel quantitativo di mandorle, 300 gr di cipolla, sale quanto basta, etc); ;
3. Serviamo il piatto a tavola.

Esempio di non algoritmo

1. Si consideri un numero N ;
2. Scrivere N ;
3. Scrivere il numero successivo;
4. Ripetere il passo precedente.

Per risolvere un problema, disponendo di un calcolatore, si può pensare di procedere attraverso quattro passi:

- 1) **identificazione:** il primo passo è quello di far emergere esplicitamente quale sia la questione da risolvere e cosa ci si attenda come soluzione
- 2) **risoluzione:** una volta che il problema sia stato bene definito, si può iniziare la ricerca di un algoritmo risolvente. Questo compito si basa su conoscenze specifiche del dominio cui il problema appartiene: matematica, fisica, economia, medicina, amministrazione di un'azienda
- 3) **programmazione:** Definito l'algoritmo risolvente, si tratta di tradurlo in un programma.
- 4) **esecuzione:** Dopo che il programma è stato completato, il calcolatore può quindi eseguirlo tutte le volte che risulti necessario.
Da notare che questa è l'unica fase di pertinenza del calcolatore.



Un elaboratore funziona grazie all'interazione tra i programmi e l'insieme di tutte le sue unità fisiche.

Si definisce **hardware (HW)** l'insieme di tutti i componenti fisici che costituiscono un sistema di elaborazione

Il termine inglese hardware significa “*ferraglia*” o “*ferramenta*”, per sottolineare come un calcolatore, senza un programma in esecuzione, è un insieme di componenti inutili

Mentre con il termine **software (SW)** s'intendono i programmi eseguibili dalla macchina

Soft-ware significa “*articoli soffici*” sta ad indicare la parte logica e non fisica dell'elaboratore

Un elaboratore dunque può essere considerato come l'insieme di due "elementi" (hardware e software) che interagiscono con l'obiettivo di elaborare dati in ingresso per fornire risultati e quindi informazioni

