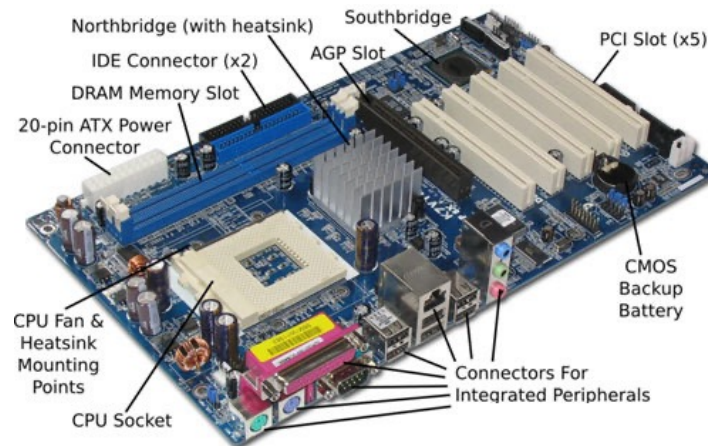




UNIVERSITÀ  
DI PARMA

## architettura degli elaboratori



# evoluzione degli strumenti di calcolo

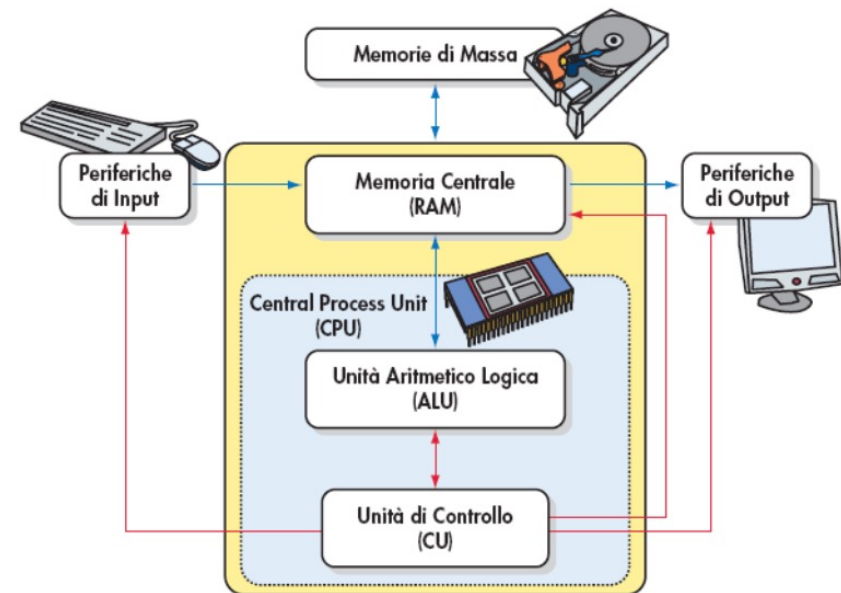
---

- antichi strumenti di ausilio al calcolo
  - abaco (ca. 2500 a.C.)
    - ausilio per addizioni, sottrazioni, moltiplicazioni e divisioni
  - bastoncini di Nepero (1617)
    - moltiplicazioni e divisioni tramite somme e sottrazioni di numeri incisi su bastoncini
  - regolo calcolatore (1632)
    - basato sui logaritmi e (usato fino agli anni '70 per calcoli ingegneristici)
- macchine meccaniche
  - pascalina (1642)
    - addizionatrice meccanica a ruote dentate: prima calcolatrice
  - macchina di Leibniz (1673)
    - simile alla pascalina, permette tutte le quattro operazioni
  - macchine a schede perforate (1801–1890)
    - telaio di Jacquard → programmazione tramite schede
    - macchina di Hollerith - schede per l'elaborazione del censimento USA (1890) → fonda la IBM.
  - macchine di Babbage (1834–1871)
    - Difference Engine
    - Analytical Engine progetto di computer programmabile - Ada Lovelace realizza il primo “programma”

- macchine elettromeccaniche (anni '1930–'1940)
  - Zuse Z1–Z3, Mark I (relè)
- macchine elettroniche (anni '1940-'1950)
  - ENIAC (1946): primo computer elettronico general purpose (valvole termoioniche)
  - EDVAC (1951): introduce l'architettura a programma memorizzato (von Neumann)
- computer moderni
  - transistor (1947) più piccoli, affidabili, e meno energivori
  - circuiti integrati (anni '60) (miniaturizzazione dei componenti)
- microprocessore (1971)
  - Intel 4004: primo processore su chip singolo
- personal computer (anni '70–'80)
  - Altair 8800, Apple I/II, IBM PC.

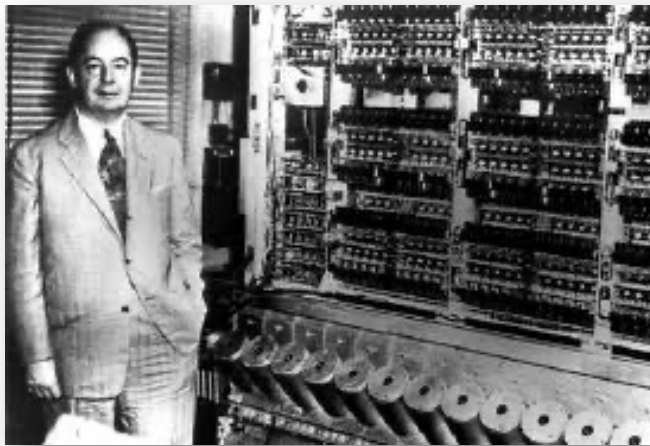
# architettura

- la macchina di *von Neumann*
- *memorie*
- *Input / Output*
- *canali di comunicazione*
- *CPU*
  - *registri*
- *istruzioni*



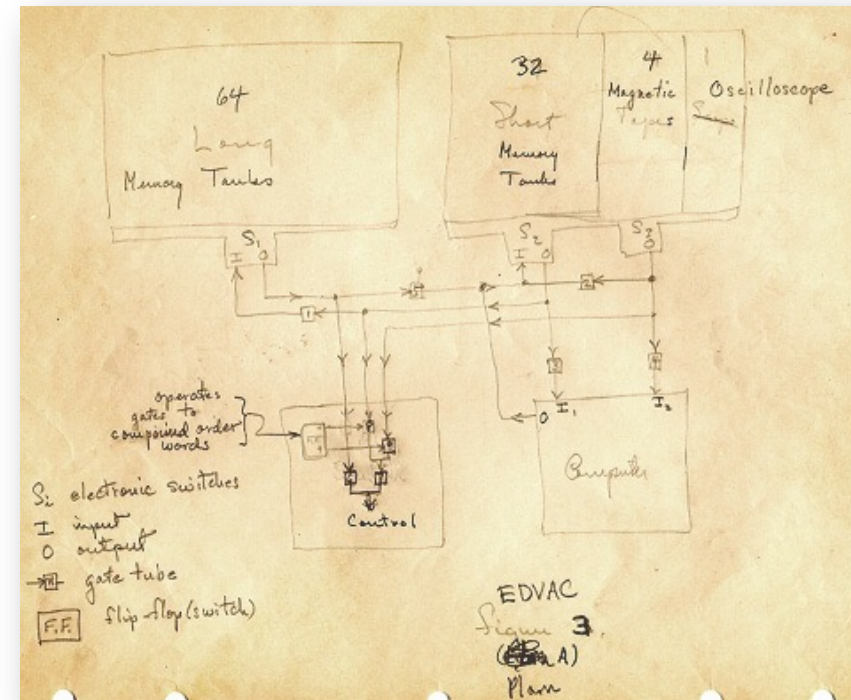
## struttura dell'elaboratore

- **modello costruttivo** riconducibile allo scienziato austriaco John Von Neumann (*architettura di Von Neumann*) anni 1940 / 1950
  - *First Draft of a Report on the EDVAC*
- **microprocessore** - tecnico italiano Federico Faggin (microprocessore Intel 4004) 1971



## EDVAC

- rispetto ai computer precedenti l'EDVAC apporta due importanti **cambiamenti concettuali**:
  - l'adozione della **matematica binaria** al posto di quella decimale, in modo da rendere il sistema più efficiente
  - invece di ricablare tutta la macchina per cambiare il programma come avveniva negli altri computer nell'EDVAC i **programmi** sono immagazzinati **in memoria**



## EDVAC (1949)

### Electronic Discrete Variable Automatic Computer

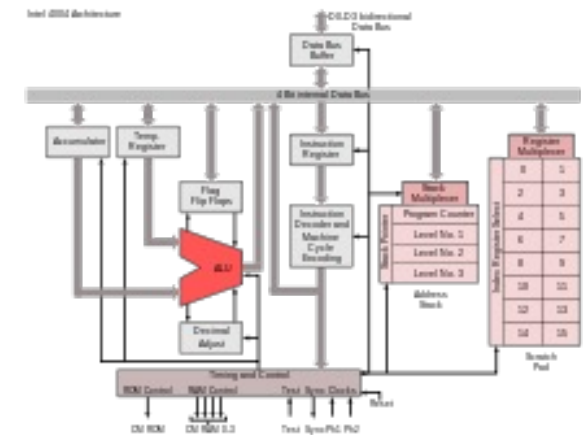
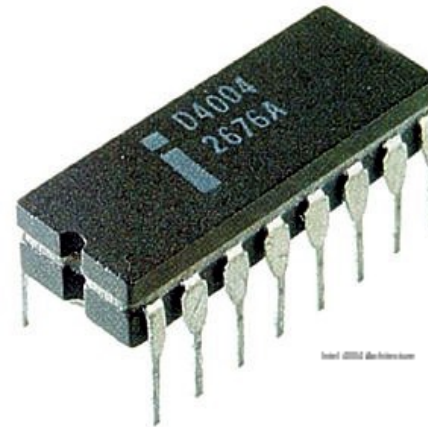
- addizione in 864 microsecondi
- moltiplicazione in 2900 microsecondi
- più di 6000 valvole termoioniche, da 12000 diodi
- consumo 56KW di potenza elettrica
- 45.5 m<sup>2</sup> di spazio
- peso 7850 chilogrammi
- personale necessario al funzionamento del sistema: 30 persone per ogni turno di 8 ore





## Faggin – Intel 4004

- l'Intel **4004** è ritenuto il primo microprocessore nella storia dell'informatica
- prodotto e commercializzato dalla Intel nel **1971**
- 4004 è un microprocessore monolitico costituito da un **unico circuito integrato**
- **Federico Faggin** è stato capo progetto dell'Intel 4004 e responsabile dello sviluppo dei microprocessori 8008, 4040 e 8080 e delle relative architetture





## EDVAC – PC moderno

Caratteristica	EDVAC (anni '40-'50)	PC Moderno (fascia media)
Tecnologia	Valvole termoioniche	Circuiti integrati - transistor
Velocità di clock (indicativa)	(funzionamento asincrono e seriale, max ~1 MHz teorico)	Tipicamente 3 GHz - 5 GHz
addizioni al secondo	Circa 1.160 operazioni al secondo	Miliardi (Giga) di operazioni al secondo
moltiplicazioni al secondo	Circa 340 operazioni al secondo	Miliardi (Giga) di operazioni al secondo
Prestazioni in FLOPS/MIPS	Pochi MIPS (inferiore a 1 MIPS)	Migliaia di GFLOPS o TeraFLOPS
Architettura	Architettura di von Neumann (primo computer a programma memorizzato)	Architettura di von Neumann con pipeline, multi-core, cache e ottimizzazioni avanzate

## istruzioni al secondo

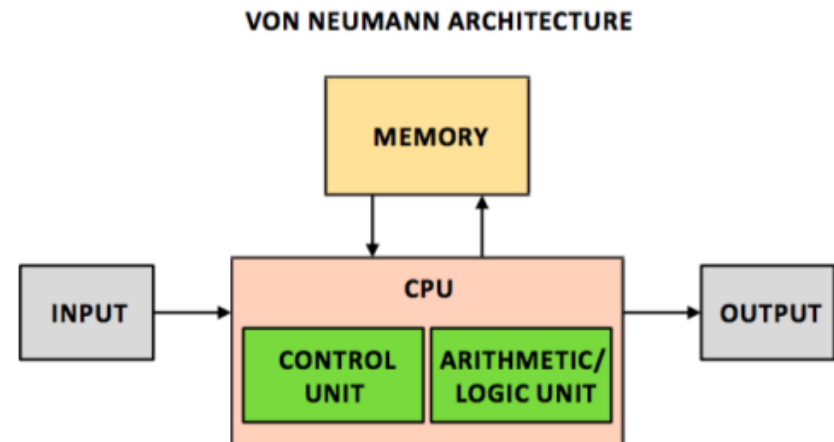
- **FLOPS** (**F**loating point **O**perations **P**er **S**econd) indica il numero di operazioni in virgola mobile eseguite in un secondo dalla CPU
- 1961 il supercomputer IBM 7030 Stretch supera la soglia del megaFLOPS.
- 1984 il supercomputer M-13 supera la soglia di un gigaFLOPS.
- 1997 il supercomputer ASCI Red supera la soglia di un teraFLOPS.
- 2008 il supercomputer IBM Roadrunner supera la soglia di un petaFLOPS
- 2012 il supercomputer XK7 della Cray raggiunge il record di 20 petaFLOPS
- 2016 il cinese Sunway TaihuLight con potenza di calcolo di 93 petaFLOPS
- Nel 2021 Aurora supercomputer presso un laboratorio in Illinois sarà il primo a superare la soglia degli exaflops

yotta FLOPS	$10^{24}$
zetta FLOPS	$10^{21}$
exa FLOPS	$10^{18}$
peta FLOPS	$10^{15}$
tera FLOPS	$10^{12}$
giga FLOPS	$10^9$
mega FLOPS	$10^6$
kilo FLOPS	$10^3$
FLOPS	1

2010 the fastest six-core PC processor reaches 109 gigaFLOPS (Intel Core i7 980 XE)

# la macchina di von Neumann

- l'architettura del computer di *von Neumann* è formata da due componenti interconnesse:
  - **CPU** (*Central Processing Unit*)
    - unità aritmetico/logica
    - unità di controllo
  - **RAM** (*Random Access Memory*)
    - sequenza di locazioni identificate da indirizzi consecutivi
    - contiene programmi e dati
- l'interfacciamento con l'esterno avviene attraverso
  - le *periferiche* di **input**
  - le *periferiche* di **output**



## unità di elaborazione - processore

---

- processore (**CPU** Central Processing Unit)
- le **istruzioni** di un programma corrispondono ad operazioni elementari di elaborazione
  - operazioni **aritmetiche**
  - operazioni **relazionali** (confronto tra dati)
  - operazioni su caratteri e **valori di verità**
- l'elaboratore è in grado di svolgere **poche** tipologie di **operazioni** elementari ma in modo **molto efficiente**
- può eseguire **centinaia di milioni** di istruzioni al secondo

## unità di controllo

---

- **CU** (control unit)
- funzione di **coordinamento** tra le varie parti del calcolatore
- è un componente dell'unità centrale di elaborazione
- ogni componente dal calcolatore esegue solo le azioni che gli vengono richieste dall'unità di controllo
- il controllo consiste nel **coordinamento** dell'esecuzione temporale delle operazioni (clock)
  - sia **internamente** all'unità di elaborazione sia da parte degli **altri** elementi funzionali

## unità di memorizzazione

---

- il calcolatore memorizza
  - i ***dati***
  - i ***programmi*** per l'elaborazione dei dati
- l'unità di memoria fornisce due sole ***operazioni***
  - memorizzazione di un valore (***scrittura***)
  - accesso al valore memorizzato (***lettura***)

## unità per il trasferimento di informazioni

---

- 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



unità di  
**memorizzazione**

Multipli del byte					
Prefissi SI			Prefissi binari		
Nome	Simbolo	Multiplo	Nome	Simbolo	Multiplo
kilobyte	kB	$10^3$	kibibyte	KiB	$2^{10}$
megabyte	MB	$10^6$	mebibyte	MiB	$2^{20}$
gigabyte	GB	$10^9$	gibibyte	GiB	$2^{30}$
terabyte	TB	$10^{12}$	tebibyte	TiB	$2^{40}$
petabyte	PB	$10^{15}$	pebibyte	PiB	$2^{50}$
exabyte	EB	$10^{18}$	exbibyte	EiB	$2^{60}$
zettabyte	ZB	$10^{21}$	zebibyte	ZiB	$2^{70}$
yottabyte	YB	$10^{24}$	yobibyte	YiB	$2^{80}$

# tipologie di memorie

---

- memoria principale **RAM** (*Random Access Memory*)
  - **volatile** (perde il suo contenuto quando si spegne il calcolatore)
  - memorizza dati e programmi
- **ROM** (*Read Only Memory*)
  - **persistente** (mantiene il suo contenuto quando si spegne il calcolatore)
  - contenuto è fisso e immutabile
  - memorizza programmi di sistema
- **Cache**
  - memoria di appoggio del processore
  - molto **veloce** (accesso estremamente rapido)
  - dimensioni limitate

## tipi di memoria

Tipo di memoria	Velocità di accesso (indicativa)	Volatilità	Funzione principale	Note
<b>Registri CPU</b>	<b>~0.2–1 ns</b>	Volatile	Memorizzano dati e istruzioni usati <i>nell'immediato</i> dalla CPU	I più veloci, pochissimi in numero
<b>Cache CPU (L1/L2/L3)</b>	<b>0.5–10 ns</b>	Volatile	Memoria rapidissima vicina alla CPU per dati usati spesso	L1 più veloce, L3 più lenta ma più grande
<b>RAM (DDR4/DDR5)</b>	<b>10–100 ns</b>	Volatile	Memoria di lavoro del sistema	Molto più lenta di cache e registri
<b>ROM</b>	<b>Microsecondi–millisecondi</b>	Non volatile	Memorizza firmware e dati permanenti	Non progettata per velocità

# memorie secondarie

- **memorie di massa - persistenti**
- supporti:
  - magnetici, ottici, a stato solido
  - nastri, hard disk, CD, DVD, SSD ...
- **grandi quantità** di informazioni
- accesso **molto meno rapido** della memoria centrale
  - i sistemi magnetici e ottici richiedono un movimento meccanico
  - i sistemi elettronici (come la memoria principale) non hanno movimenti meccanici



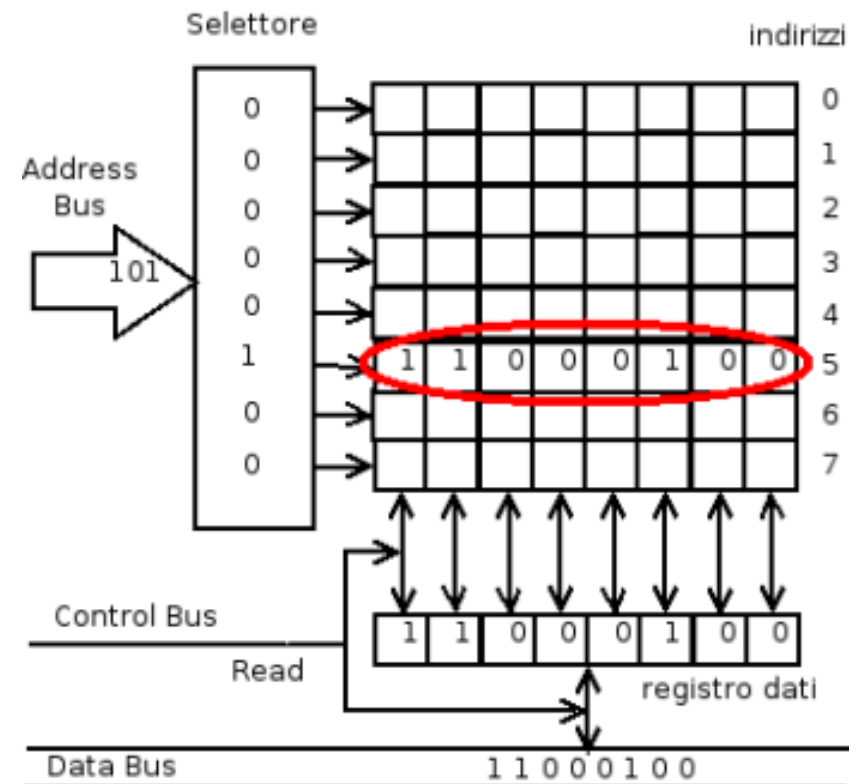
## memoria centrale - RAM

---

- contiene le istruzioni e i dati
- è un contenitore di celle
  - le celle sono *numerate* in *sequenza*
  - il numero di ogni cella costituisce il suo *indirizzo*
  - il numero totale di celle è definito *spazio degli indirizzi* (spazio di indirizzamento)
  - l'ampiezza dello spazio di indirizzamento fisico è determinato dall'ampiezza del *bus indirizzi*
  - specificando l'indirizzo di una cella, la CPU è in grado di *leggere* e/o *modificare* il valore del byte memorizzato in quella cella
- è *volatile*
  - perde il suo contenuto tutte le volte che la macchina viene spenta
- **Random Access Memory (RAM)** ogni cella è indirizzabile direttamente

# memoria

- la **memoria** è costituita da un insieme di dispositivi in grado di **conservare** in stati binari (**bit**) l'unità di **informazione**
- la dimensione standard dell'unità di informazione è 8 bit (1 **byte**)
- la memoria è suddivisa in **celle** (**locazioni di memoria**) che memorizzano un byte di informazione
- ad ogni cella è associato un **indirizzo univoco**



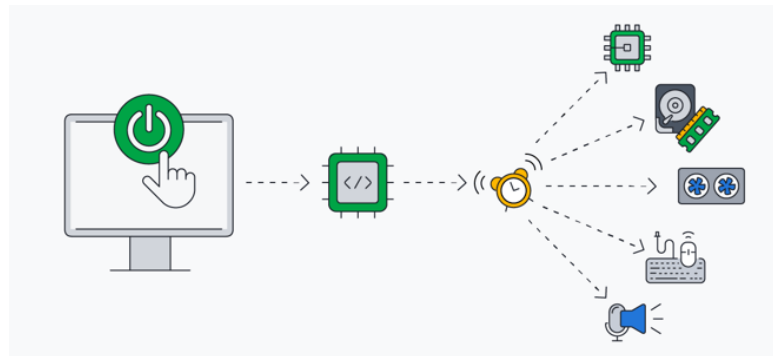
# BIOS

- nel sistema è presente una speciale area di memoria che non perde i valori dopo lo spegnimento (***non volatile***)
- il sistema nella fase di ***avviamento (bootstrap)***, deve immettere sul Bus le ***istruzioni iniziali*** per
  - ***configurare*** i dispositivi di base (*es. video e tastiera*)
    - fase di ***POST*** (Power On Self Test)
  - ***caricare*** i programmi del ***sistema operativo*** da una memoria secondaria
- quest'area è ***riservata*** all'interno dello spazio di indirizzamento ed è denominata, genericamente, ***BIOS (Basic Input/Output System)***



# ROM

- la memoria che contiene il BIOS sono realizzate in **ROM**
  - Read Only Memory
  - tecnologia che consente alle celle di mantenere il contenuto anche in assenza di alimentazione
- il codice e i programmi contenuti in maniera non volatile nella memoria centrale sono detti **Firmware**

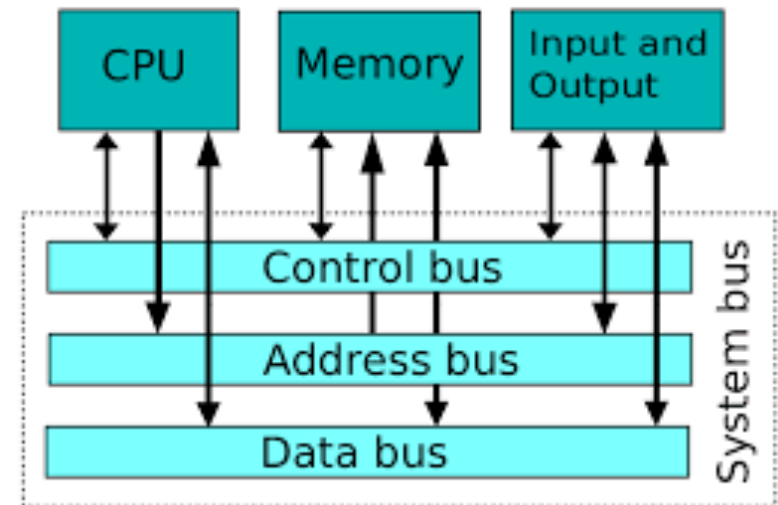


trasferimento informazioni

# BUS

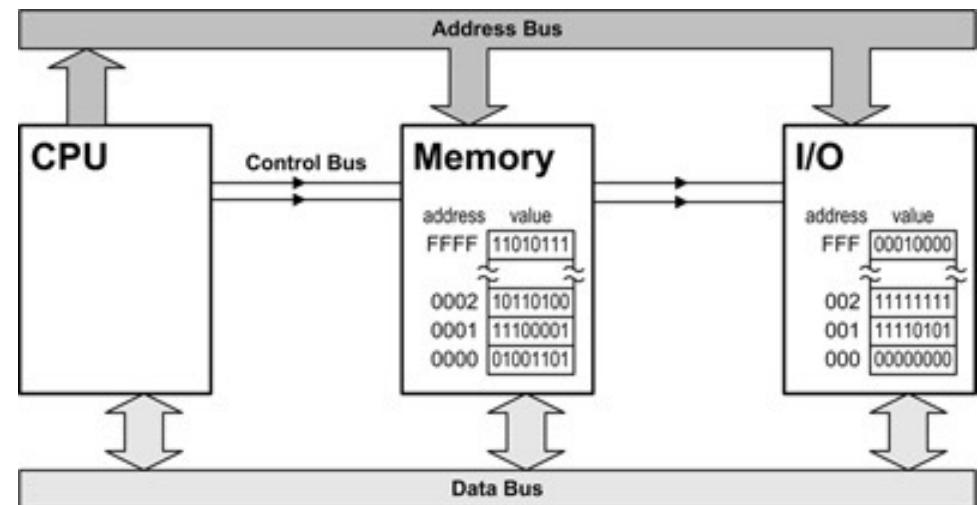
# bus

- il **bus** è l'unità di **interconnessione** tra i moduli del modello di Von Neumann
- **logicamente** è un fascio ordinato di **linee**, ognuna delle quali può assumere un valore binario
- **fisicamente** è di un insieme di **fili** paralleli (*piste di rame nella realtà*) che permettono il passaggio dei segnali elettrici fra i vari componenti



## data bus – address bus - control bus

- sul **data bus** (*bus dati bidirezionale*) viaggiano i dati che si scambiano i vari componenti
  - dati dalla memoria centrale verso la CPU per essere elaborati
  - dati dalla CPU verso la memoria per essere conservati
- sull'**address bus** (*bus indirizzi monodirezionale*) viaggiano gli indirizzi di memoria cui si vuole accedere
- sul **control bus** (*bus di controllo bidirezionale*) viaggiano i segnali di sincronizzazione fra i vari dispositivi, necessari per la comunicazione



## dimensioni dei bus

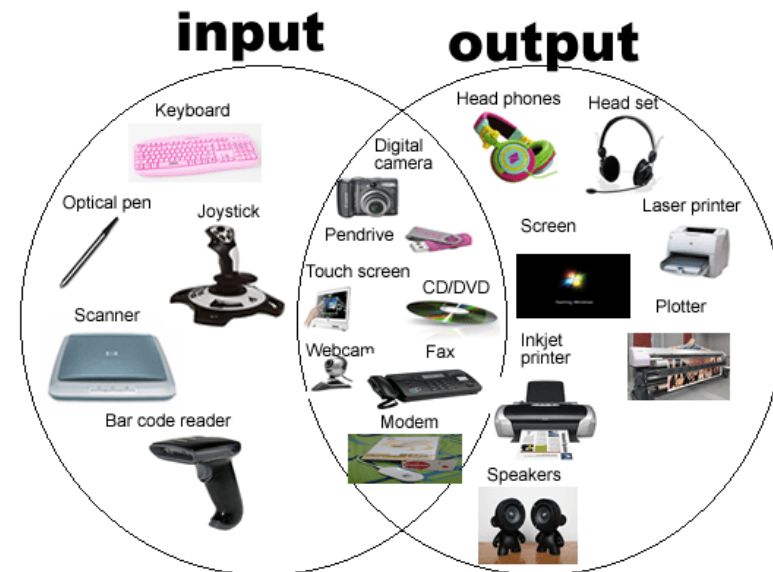
---

- la dimensione dell'**address bus** determina il massimo numero di celle di memoria *indirizzabili*
  - nella quasi totalità dei processori moderni ogni indirizzo identifica un byte (byte-addressable)
- la dimensione del **data bus** indica quanti bit possono essere trasferiti contemporaneamente tra CPU, memoria e periferiche
  - maggiore dimensione del data bus → più dati trasferiti → sistema più veloce
  - processori moderni (Intel Core, AMD Ryzen)
    - data bus esterno: 64 bit
    - data bus interno (registri, ALU, cache): (128, 256, 512 bit)

unità di  
**input / output**

# Input / Output

- la sezione di Input/Output (I/O) di un calcolatore è dedicata alla **comunicazione** con l'**esterno** per **l'acquisizione** (input) o la **comunicazione** (output) di dati
- la sezione di I/O è dotata di uno spazio di indirizzamento (**spazio degli indirizzi di I/O**)
- ogni dispositivo periferico possiede un proprio **range di indirizzi** di I/O riservato (*indirizzi di I/O, detti anche registri di I/O o porte di I/O*)

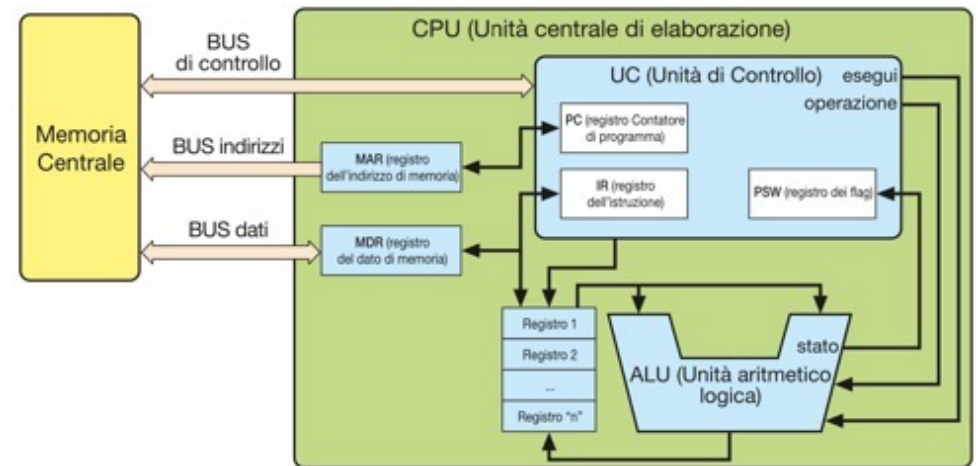




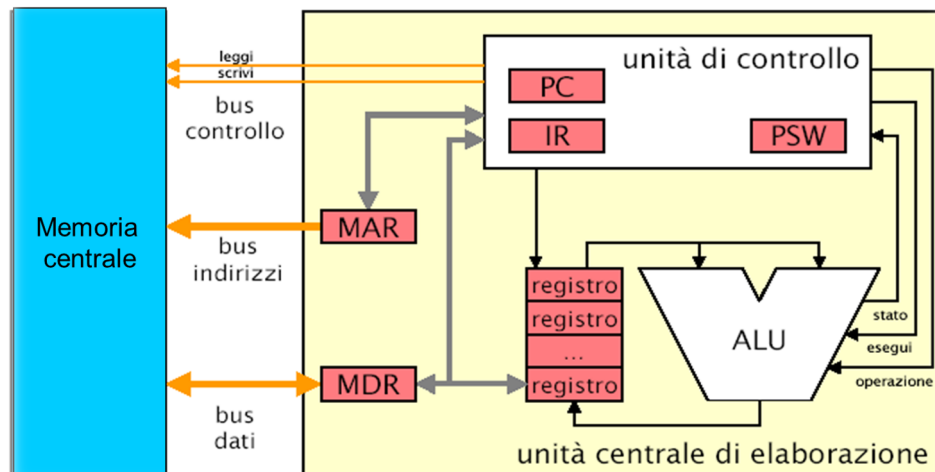
unità di  
**elaborazione e controllo**

# CPU

- La CPU è composta da
  - **unità di controllo** (*CU, Control Unit*)
    - sovrintende al funzionamento della macchina
    - controlla la sequenza delle istruzioni da eseguire
  - **unità aritmetico-logica** (*ALU, Arithmetic-Logic Unit*)
    - svolge le operazioni specificate dalle istruzioni



## unità centrale di elaborazione



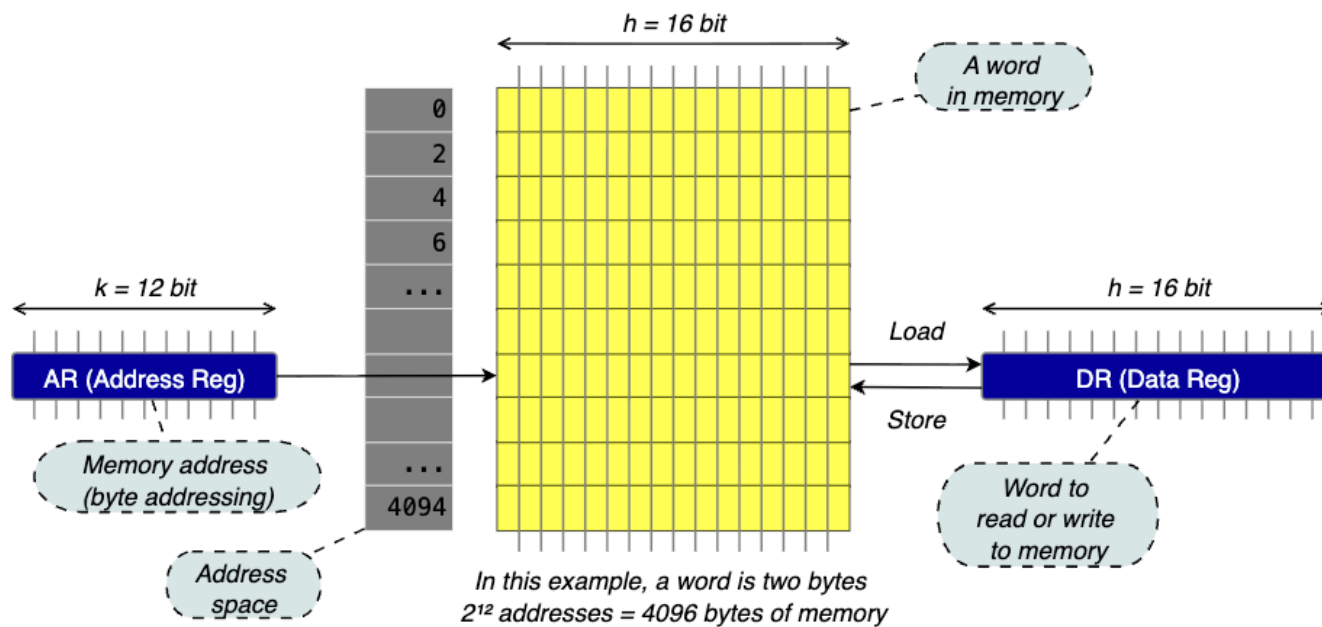
- l'unità di controllo imposta i valori sul *bus*
- i *registri* contengono i dati e i risultati delle istruzioni che saranno eseguita dall'*alu*
- l'*alu* esegue le istruzioni
  - all'interno dell'unità si trovano microprogrammi cablati direttamente in hardware, scritti in microcodice con relative microistruzioni

# registri

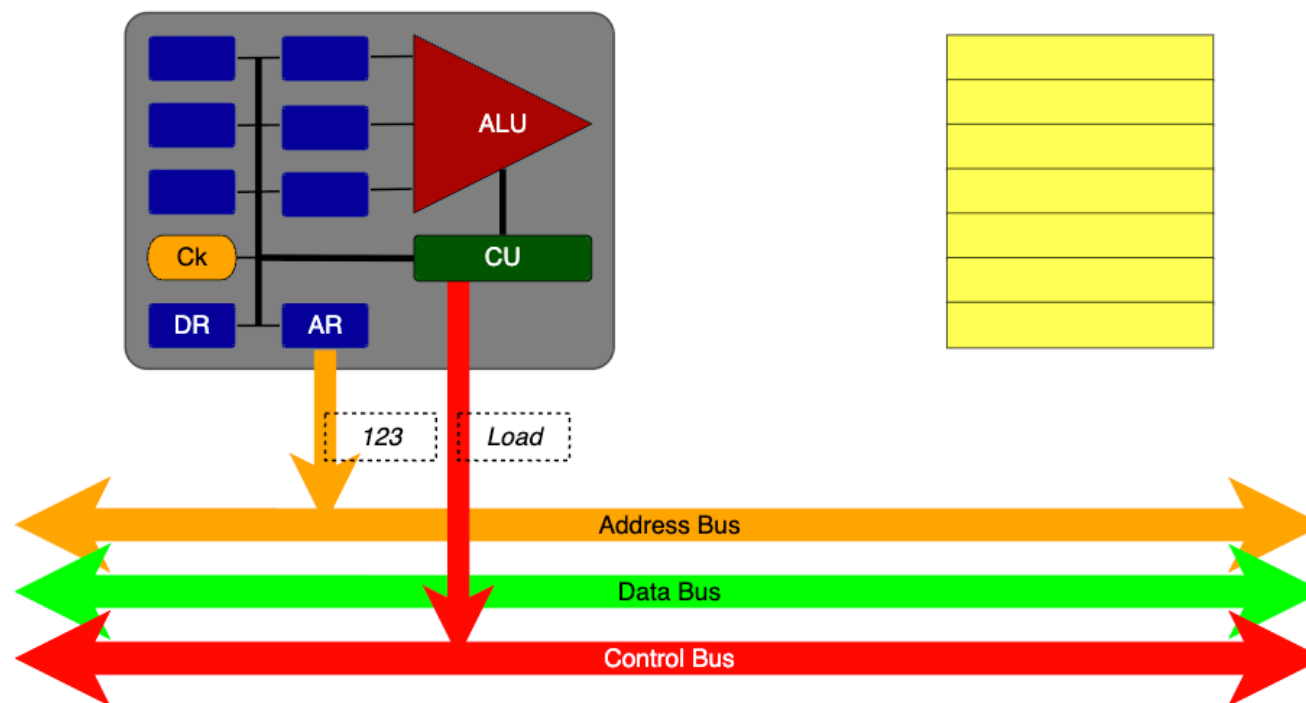
---

- circuiti di memoria usati per conservare i dati su cui svolgere operazioni
- per svolgere operazioni un dato della memoria principale va trasferito su un registro
- numero limitato di registri 8-128
- registri general purpose per operazioni
- registri speciali:
  - program counter (**PC**): indirizzo di memoria prossima istruzione da eseguire
  - instruction register (**IR**): istruzione da eseguire
  - flag / status register (**SR**): bit per segnalare condizioni di stato, es. overflow
  - address register (**AR**): indirizzo locazione memoria per

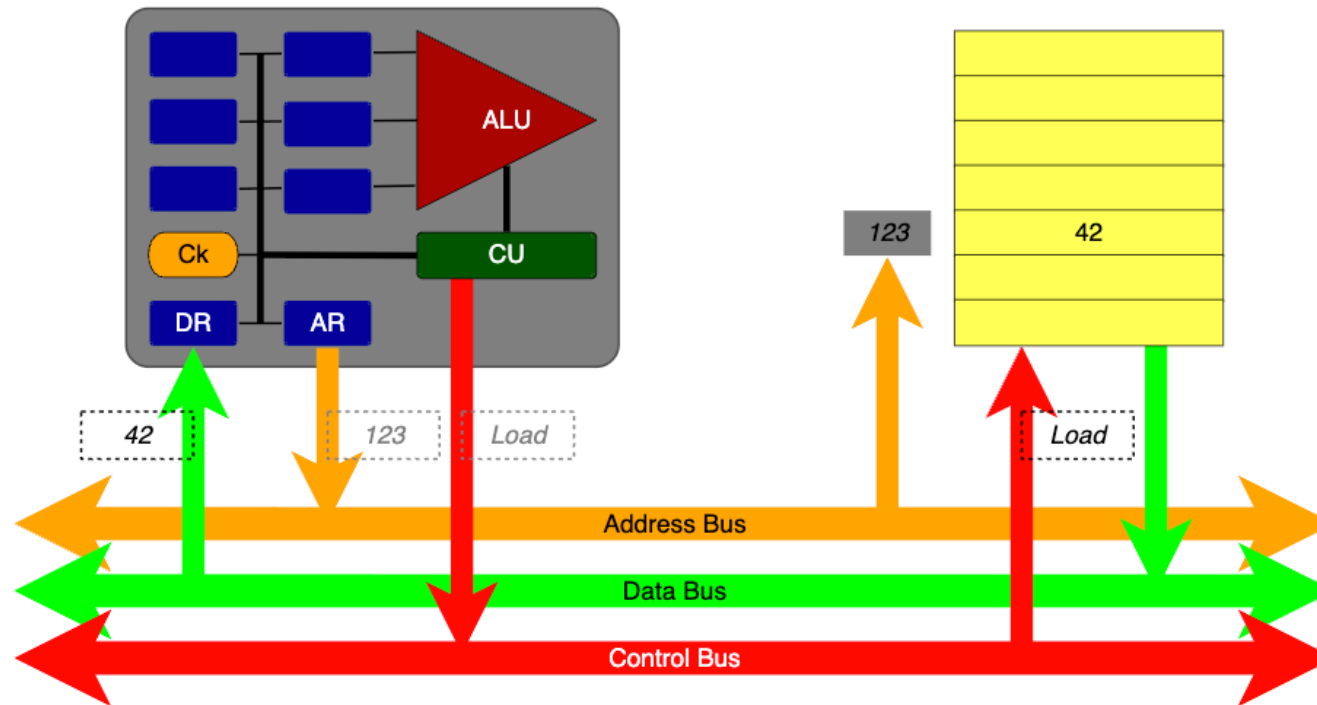
## memoria – lettura e scrittura



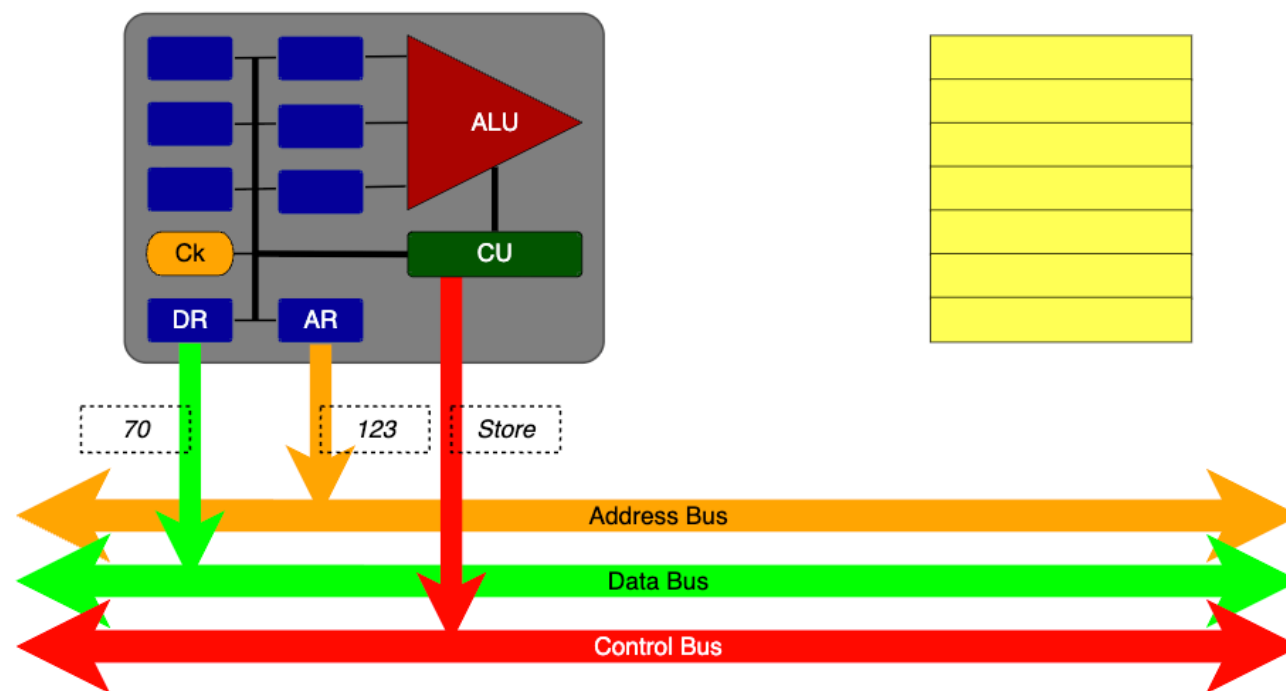
## lettura (1)



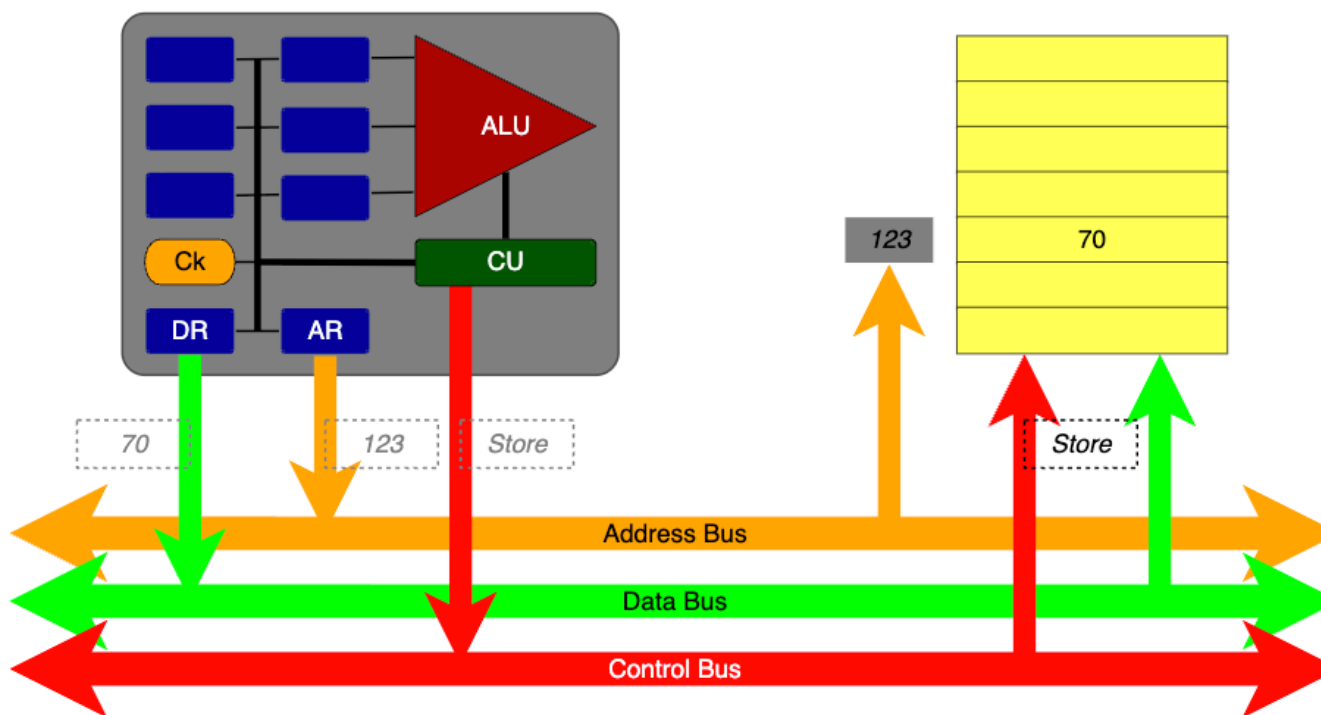
## lettura (2)







## scrittura (2)



## set di istruzioni

---

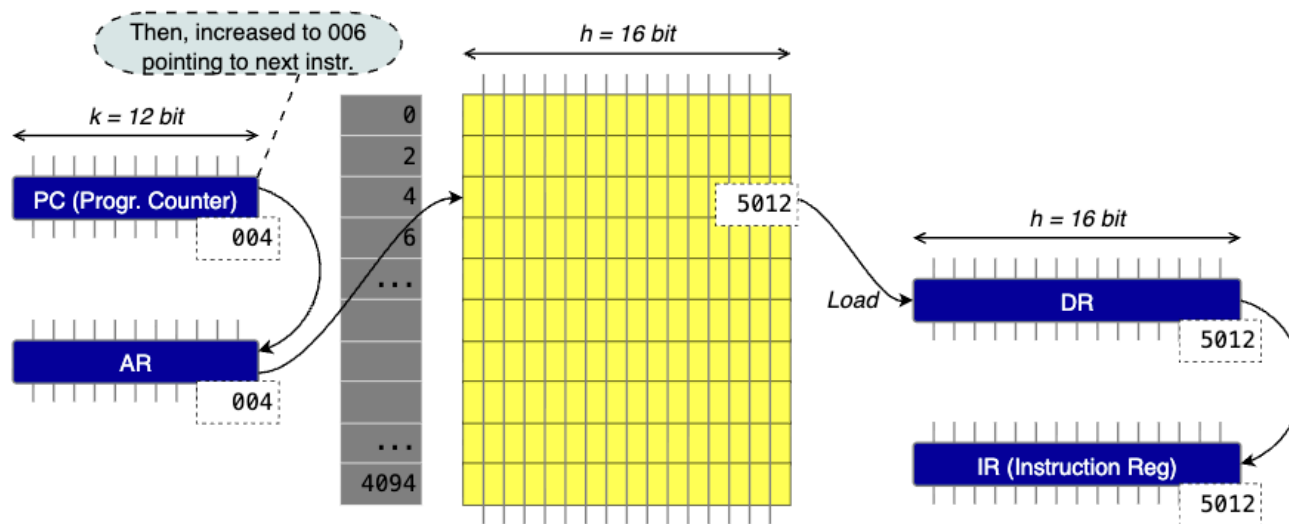
- ogni processore viene progettato con un **set di istruzioni** specifico denominato **ISA** (*Instruction Set Architecture* o *Instruction Set*)
- ogni **istruzione** è implementata da un **microprogramma**
- ogni istruzione dell'ISA è contraddistinta da un **numero** specifico, denominato **Operation Code** (*Op. Code*)
- ogni istruzione necessita di un numero preciso e definito di **parametri** che, assieme all'Op.Code, determinano la **lunghezza dell'istruzione** (espressa in byte)
- un registro speciale del processore, **Program Counter** (*PC*), si **incrementa** della lunghezza dell'istruzione appena eseguita

# ciclo di esecuzione di una istruzione

---

- ***fetch***
  - l'unità di controllo pone sul bus degli indirizzi il valore del Program Counter e legge dalla memoria il codice dell'istruzione da eseguire
- ***decode***
  - l'unità di controllo decodifica l'istruzione e legge i parametri (*Operand Fetch*) che vengono memorizzati nei registri
- ***execute***
  - viene avviato il microprogramma relativo all'Op.Code
  - la frequenza in base alla quale vengono eseguiti i microprogrammi è regolata dal ***clock*** di CPU (*frequenza del microprocessore*)
- ***store***
  - al termine della fase di execute gli eventuali risultati, posti nei registri, vengono scritti sul bus dall'UC, o verso la memoria, o verso l'I/O

# fetch



# CISC

- ***complex instruction set computer (CISC)*** indica un'architettura per microprocessori formata da un set di istruzioni contenente istruzioni in grado di eseguire operazioni complesse
  - lettura di un dato in memoria, la sua modifica e il suo salvataggio direttamente in memoria tramite una singola istruzione
- ogni singola istruzione ha un *data path* a più cicli
  - il *data path* è il percorso dei dati all'interno del processore, attraverso l'attuale istruzione, e i suoi cicli sono scanditi dal clock della CPU
- le architetture CISC possiedono un ***set di istruzioni molto ampio*** e istruzioni a ***lunghezza variabile***

# RISC

- le architetture **RISC** (*Reduced Instruction Set Computer*) seguono una filosofia di progettazione basate su di un set di istruzioni in grado di eseguire operazioni **semplici**
- il set di istruzioni di una architettura RISC è **limitato**
  - le istruzioni hanno lunghezza costante
  - un numero fisso di operandi fisso
  - la fase di decode è breve
  - ogni istruzione è eseguita direttamente in hardware con pochi cicli di clock
- gli attuali processori seguono una filosofia **ibrida** in cui istruzioni CISC vengono in qualche modo decodificate ed eseguite in termini istruzioni RISC

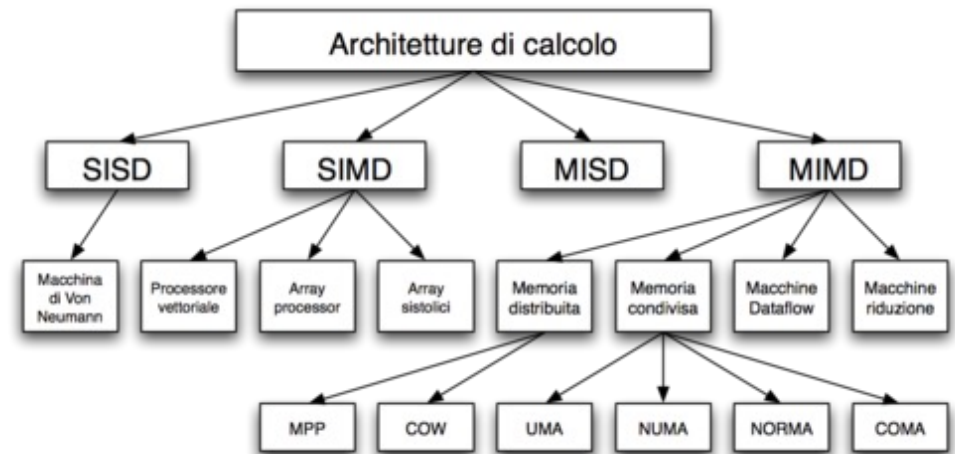
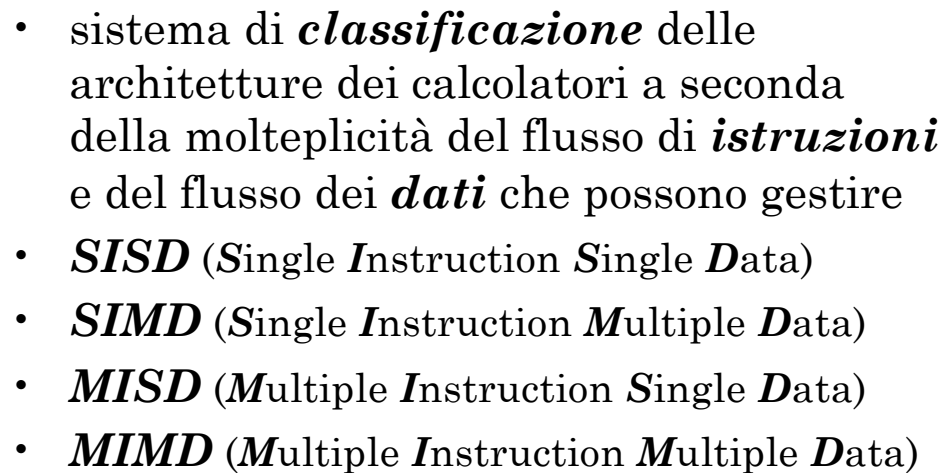
## aumentare il parallelismo in esecuzione

---

- caricare nel processore ***più istruzioni*** oltre a quella richiesta
  - ***coda di prefetch*** (buffer interno in cui il processore memorizza una serie di byte consecutivi a quelli appena letti dalla memoria)
- con un solo accesso alla memoria, si ha a disposizione una serie di valori che possono essere elaborati (istruzioni e/o operandi) senza dover accedere nuovamente al bus e alla memoria
- l'evoluzione della coda di *prefetch* porta al sistema ***pipeline***
  - sfrutta il concetto di catena di montaggio: possibilità di avviare la successiva istruzione subito dopo che la precedente è stata inserita nel data path
    - es. se la prima istruzione si trova in fase di decode, la successiva può essere posta in stato di fetch



## tassonomia di Flynn



## architettura PC attuale

- chip Northbridge (ora integrato nella CPU)
  - gestisce le comunicazioni veloci tra CPU, RAM e GPU (Graphics Processing Unit)
  - si chiama “Northbridge” perché sulla mappa della scheda madre era posizionato verso nord della CPU.
  - velocità ~ 50 GB/s
- chip Southbridge
  - gestisce le comunicazioni più lente e periferiche secondarie
  - Hard disk (SATA, IDE) - USB, audio, rete, BIOS
  - velocità ~ 500 MB/s

