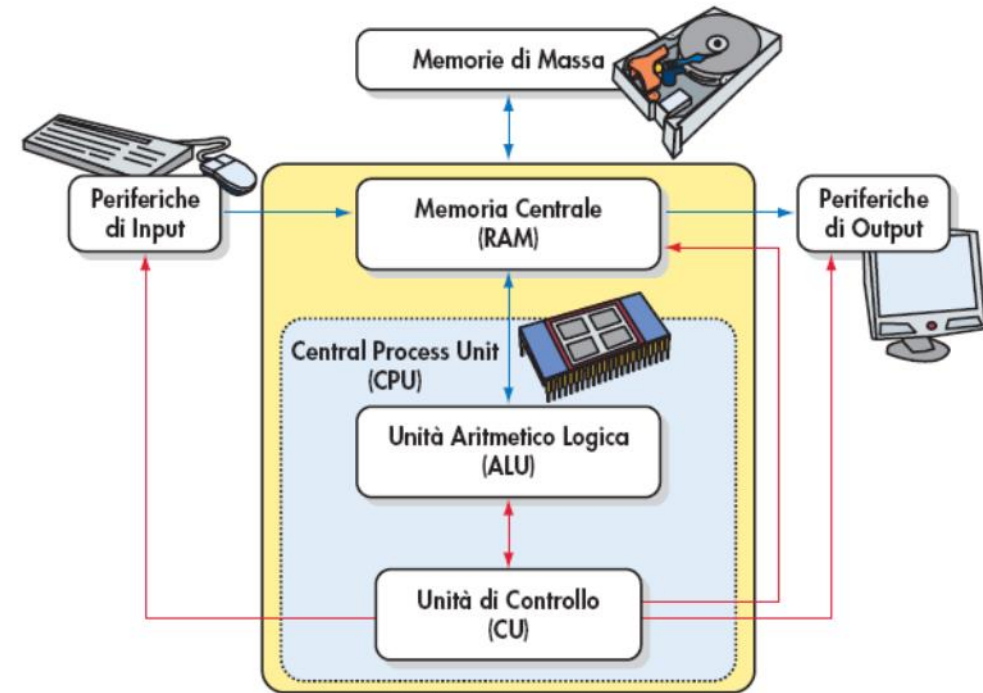




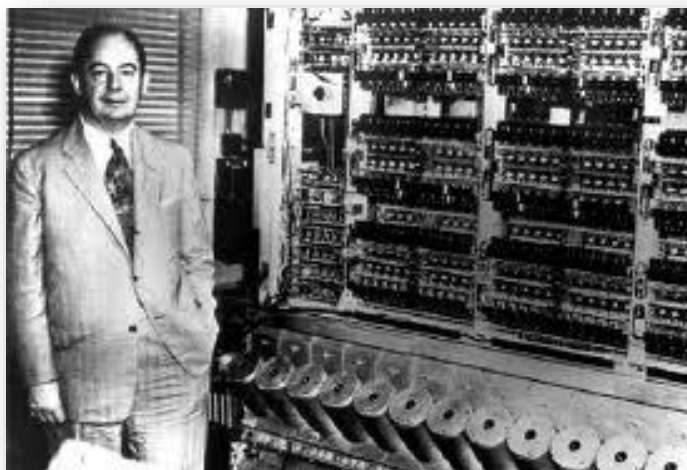
**UNIVERSITÀ
DI PARMA**

architettura degli elaboratori
Alberto Ferrari

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



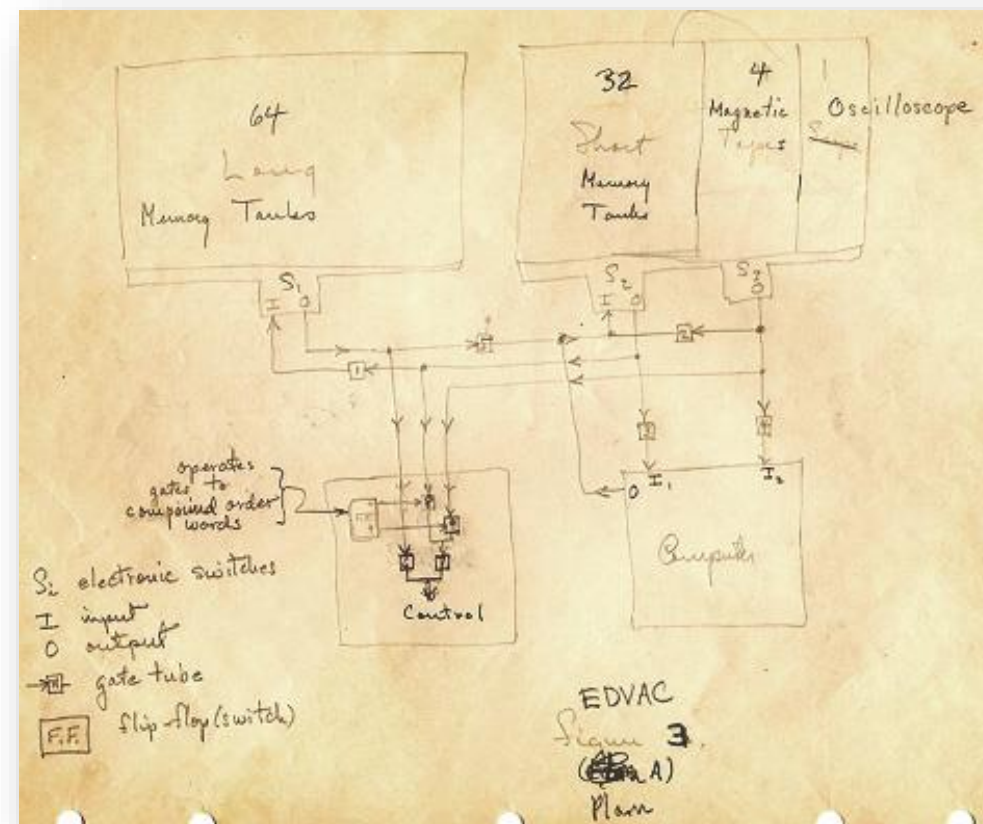
- la **struttura** dei calcolatori elettronici assume la forma attuale in base ad almeno due **svolte tecnologiche** fondamentali:
 - un **modello costruttivo** riconducibile allo scienziato austriaco John Von Neumann (**architettura di Von Neumann**) anni 1940 / 1950
 - *First Draft of a Report on the EDVAC*
 - l'invenzione del **microprocessore** da parte del tecnico italiano Federico Faggin (microprocessore Intel 4004) 1971



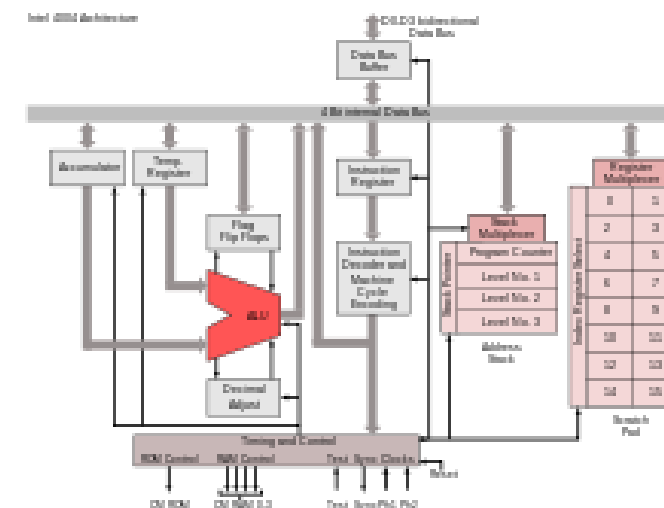
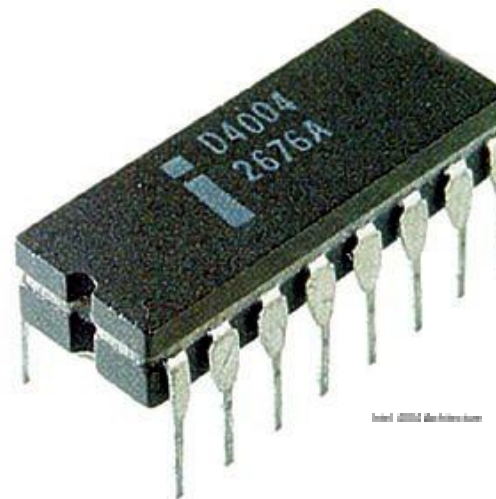
- addizione in 864 microsecondi
- moltiplicazione in 2900 microsecondi
- più di 6000 valvole termoioniche, da 12000 diodi
- consumo 56KW di potenza elettrica
- 45.5 m² di spazio
- peso 7850 chilogrammi
- personale necessario al funzionamento del sistema: 30 persone per ogni turno di 8 ore



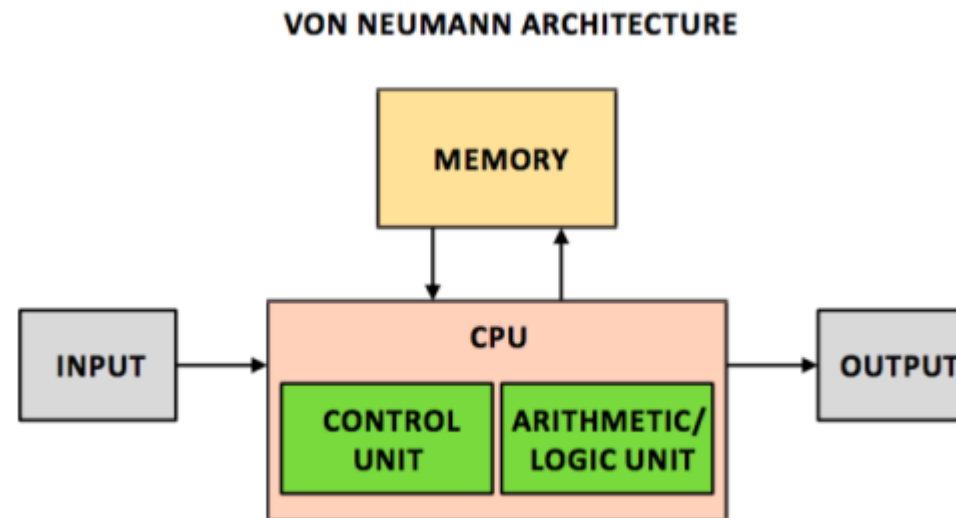
- rispetto ai computer precedenti l'EDVAC apporta due importanti **cambiamenti concettuali**:
 - l'adozione della **matematica binaria** al posto di quella decimale, in modo da rendere il sistema più efficiente
 - invece di ricablare tutta la macchina per cambiare il programma come avveniva negli altri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



- 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



- l'architettura del computer di *von Neumann* è formata da due componenti interconnesse:
 - *CPU (Central Processing Unit)*
 - unità centrale di elaborazione
 - *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*



- il funzionamento di un calcolatore è descrivibile in termini di poche componenti (*macro-unità*) funzionali
- ogni macro-unità è *specializzata* nello svolgimento di una tipologia omogenea di funzioni
- eccezione: l'*unità centrale* di elaborazione, che svolge sia funzioni sia di *elaborazione* che di *controllo*

- 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*)

- 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

- **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)

- il **coordinamento** tra le varie parti del calcolatore è svolto dall'**unità di controllo**
- 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
 - sia **internamente** all'unità di elaborazione sia da parte degli **altri** elementi funzionali

- 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 misura binarie

Unità	Simbolo	Valore	Totale byte	Note
kibibyte	KiB	1024 byte	$2^{10} = 1\,024$	La "K" del prefisso è maiuscola
mebibyte	MiB	1024 KiB	$2^{20} = 1\,048\,576$	$2^{20} = (2^{10})^2 = 1024^2$
gibibyte	GiB	1024 MiB	$2^{30} = 1\,073\,741\,824$	$2^{30} = (2^{10})^3 = 1024^3$
tebibyte	TiB	1024 GiB	$2^{40} = 1\,099\,511\,627\,776$	$2^{40} = (2^{10})^4 = 1024^4$
pebibyte	PiB	1024 TiB	$2^{50} = 1\,125\,899\,906\,842\,624$	$2^{50} = (2^{10})^5 = 1024^5$
exbibyte	EiB	1024 PiB	$2^{60} = 1\,152\,921\,504\,606\,846\,976$	$2^{60} = (2^{10})^6 = 1024^6$
zebibyte	ZiB	1024 EiB	$2^{70} = 1\,180\,591\,620\,717\,411\,303\,424$	$2^{70} = (2^{10})^7 = 1024^7$
yobibyte	YiB	1024 ZiB	$2^{80} = 1\,208\,925\,819\,614\,629\,174\,706\,176$	$2^{80} = (2^{10})^8 = 1024^8$

unità di

memorizzazione

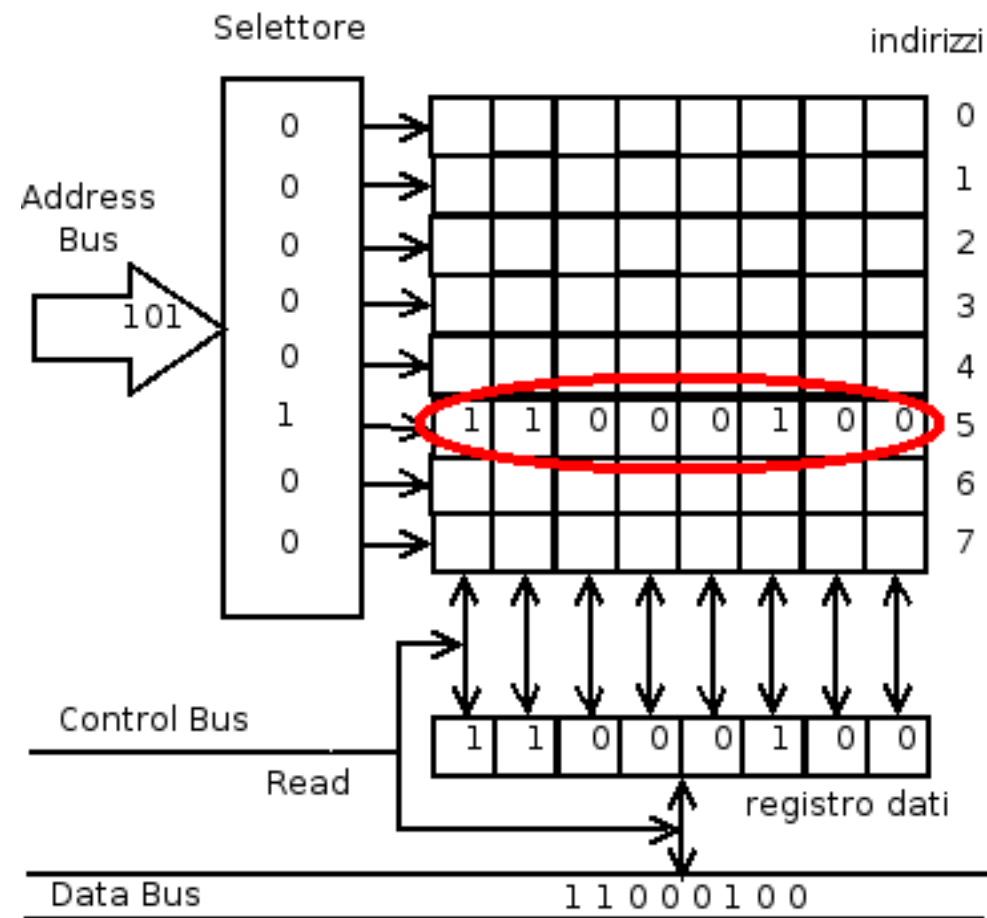
- memoria principale **RAM** (*Random Access Memory*)
 - **volatile** (perde il suo contenuto quando si spegne il calcolatore)
 - memorizza dati e programmi
- **ROM** (*Read Only Memory*)
 - **persistente** (mantiene il suo contenuto quando si spegne il calcolatore)
 - contenuto è fisso e immutabile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

- *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 *millisecondi* (10^{-3} sec) contro *nanosecondi* (10^{-9} sec) - rapporto 10^6
 - i sistemi magnetici e ottici richiedono un movimento meccanico
 - i sistemi elettronici (come la memoria principale) non hanno movimenti meccanici



- 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

- 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**



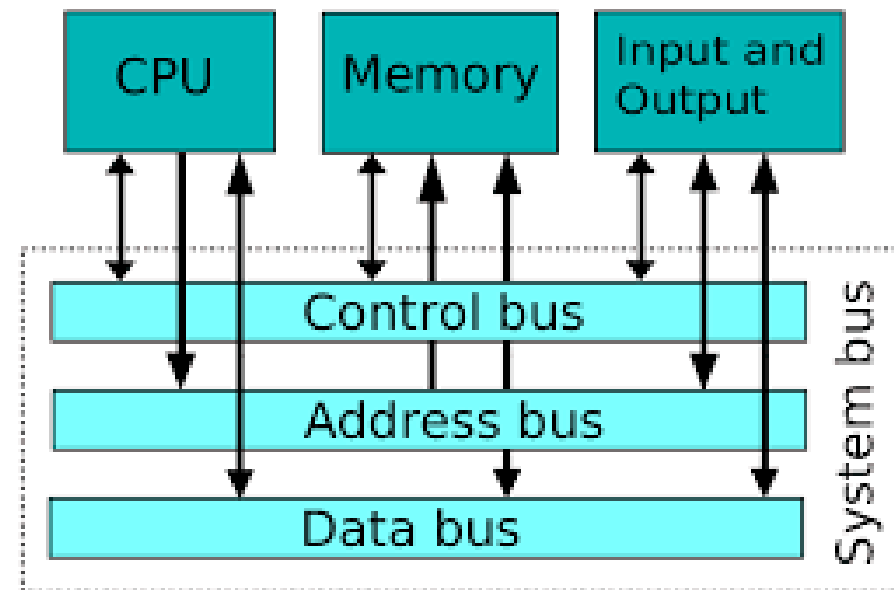
- nel sistema è presente una speciale area di memoria che non perde i valori dopo lo spegnimento (*non volatile*)
- il sistema nella fase di *avviamento (bootstrap)*, deve immettere sul Bus le *istruzioni iniziali* per
 - *configurare* i dispositivi di base (*es. video e tastiera*)
 - fase di *POST* (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)*

- 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***

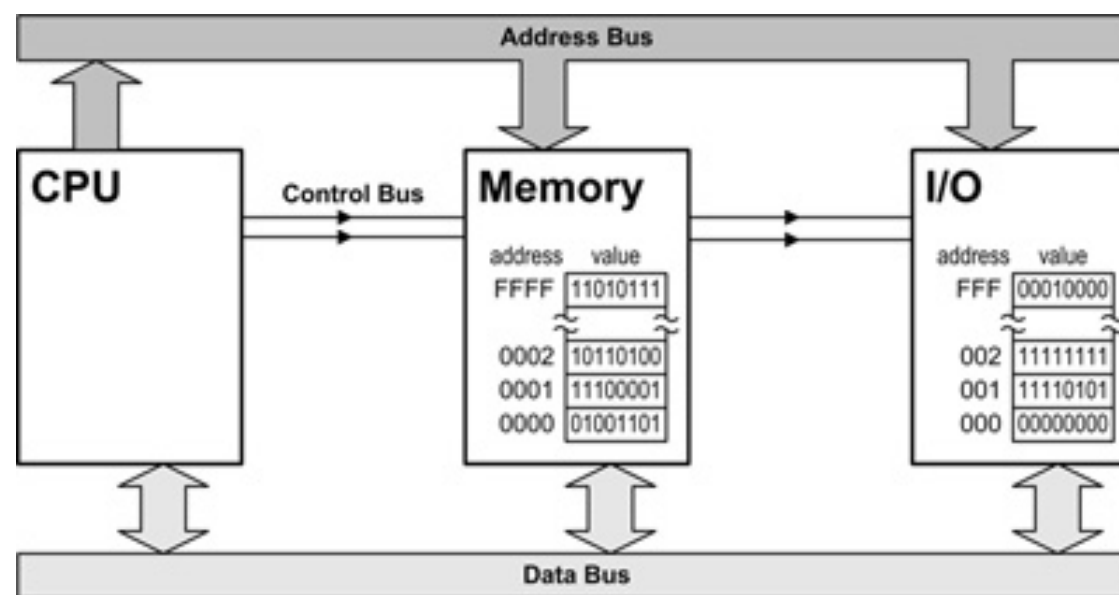
trasferimento informazioni

BUS

- 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



- 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



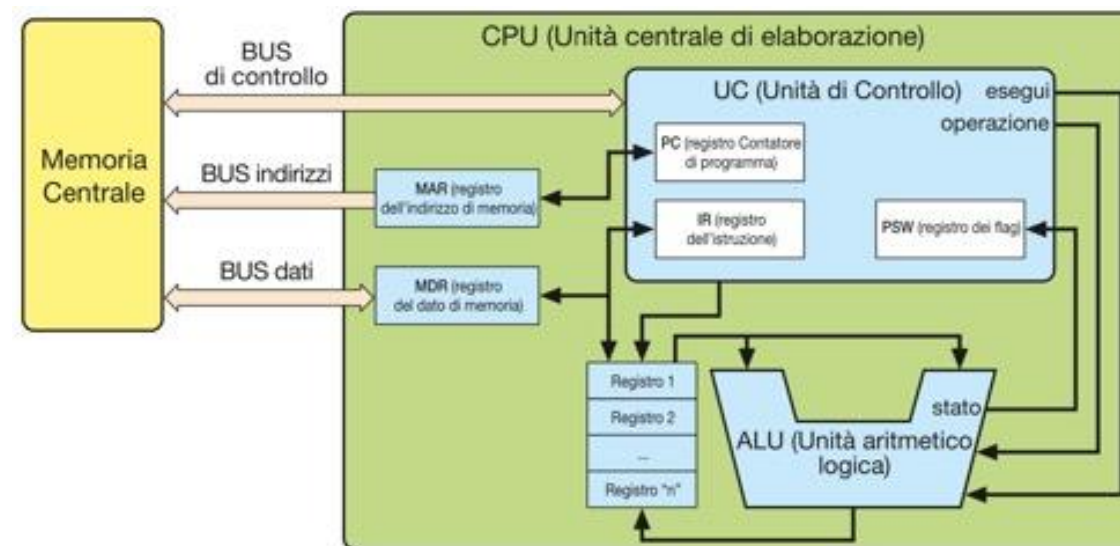
unità di
input / output

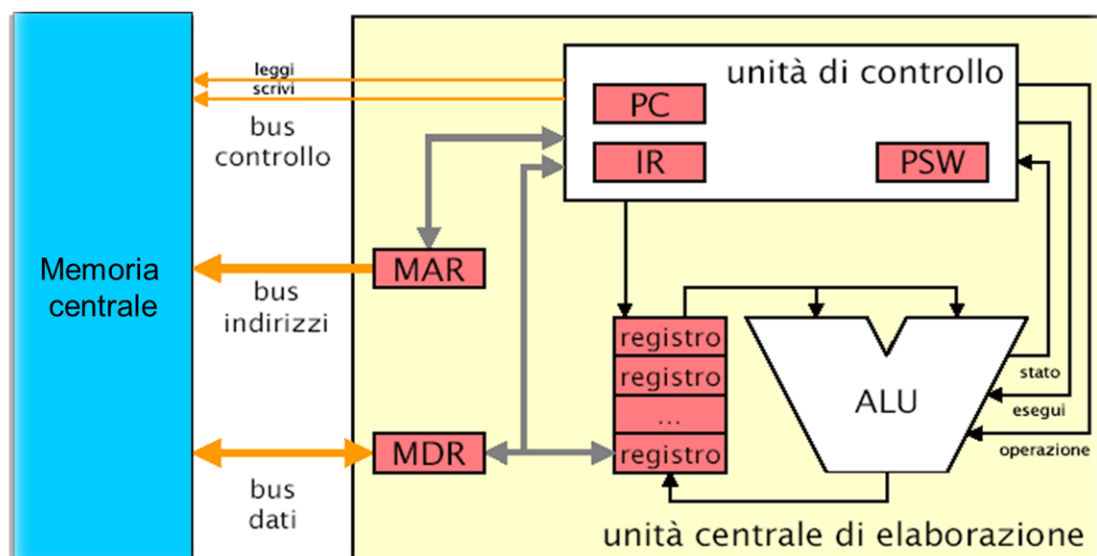
- la sezione di Input/Output (I/O) di un calcolatore è dedicata alla **comunicazione** con l'**esterno** per **l'acquisizione** (input) o la **comunicazione** (output) di dati
- concettualmente la sezione di I/O è ancora rappresentabile come un **contenitore**, del tutto analogo alla memoria, anche se dotato di uno spazio di indirizzamento (**spazio degli indirizzi di I/O**) molto ridotto
- 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

- un **processore** è un singolo **circuito integrato** in grado di effettuare operazioni **decisionali**, di **calcolo** o di **elaborazione** dell'informazione
- il microprocessore principale di un computer viene chiamato **CPU** (Central Processor Unit)
- il processore può essere visto come suddiviso in tre unità funzionali:
 - l'unità di **controllo** (**CU control unit**)
 - l'area dei **registri**
 - l'unità aritmetico logica (**ALU arithmetic-logic unit**)

- 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





- 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

- Instruction Register (IR)
- Program Counter (PC)
- Memory Address Register (MAR)
- data registers
- status register
- ...

5-2 Computer Registers

Register symbol	Number of bits	Register name	Register Function-----
DR	16	Data register	Holds memory operands
AR	12	Address register	Holds address for memory
AC	16	Accumulator	Processor register
IR	16	Instruction register	Holds instruction code
PC	12	Program counter	Holds address of instruction
TR	16	Temporary register	Holds temporary data
INPR	8	Input register	Holds input character
OUTR	8	Output register	Holds output character

- 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

- **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

- *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*

- 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

- 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

- sistema di *classificazione* delle architetture dei calcolatori a seconda della molteplicità del flusso di *istruzioni* e del flusso dei *dati* che possono gestire
- **SISD** (Single *I*nstruction Single *D*ata)
- **SIMD** (Single *I*nstruction *M*ultiple *D*ata)
- **MISD** (*M*ultiple *I*nstruction Single *D*ata)
- **MIMD** (*M*ultiple *I*nstruction *M*ultiple *D*ata)

