

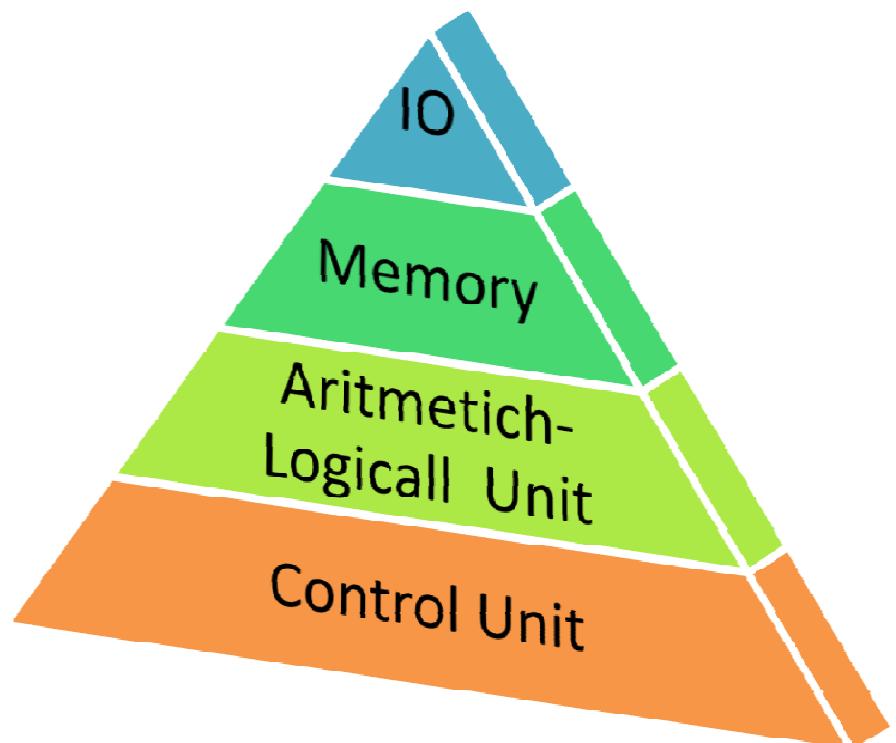
Architettura degli elaboratori

Macchina di von Neumann

Dott. Franco Liberati

ARGOMENTI DELLA LEZIONE

- Macchina di von Neumann
 - Unità di Controllo
 - Unità Logico-Aritmetica
 - Co-processore matematico
 - Memoria Centrale
 - Unità di input Output
 - Interconnessione
 - Bus e arbitratore





Architettura degli elaboratori

Macchine programmabili



ELABORATORE ELETTRONICO

Macchina di von Neumann

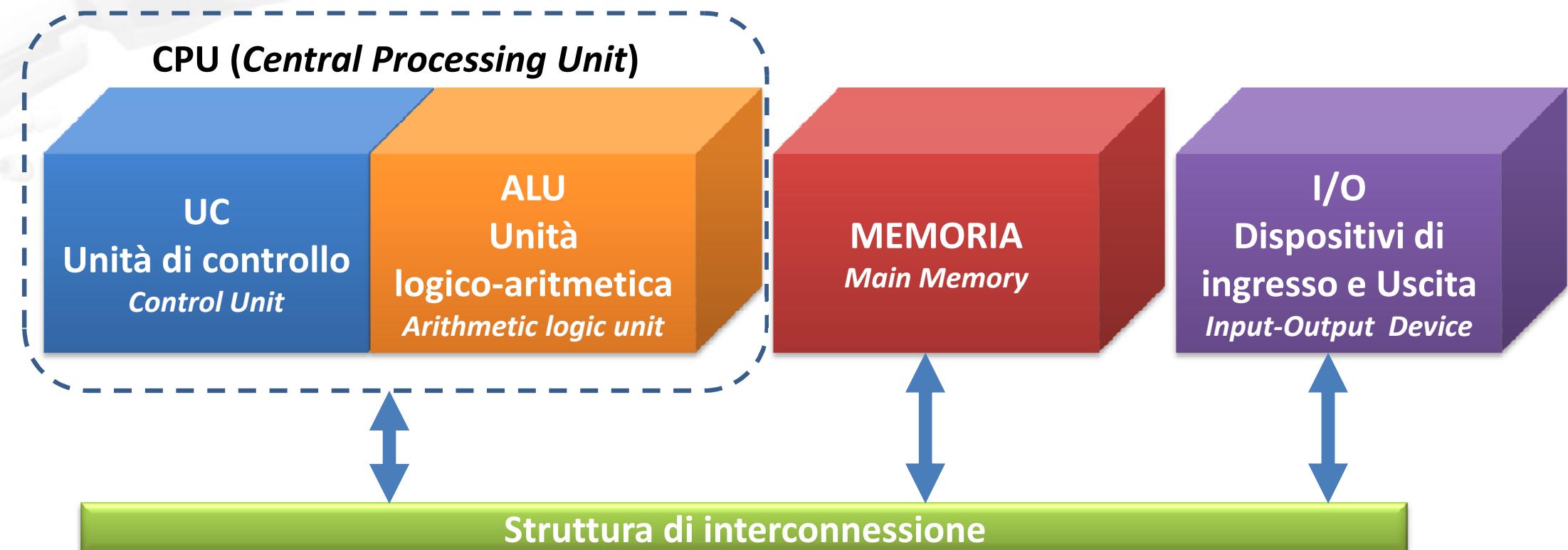
- Nel 1945 fu presentato un **modello di calcolatore ad uso generale** descritto da John von Neumann e Hermann Goldstine
- Il **modello di “Macchina di von Neumann”** (o “Macchina di Princeton”) prevedeva la definizione di quattro moduli essenziali ed una struttura di interconnessione nonché la presenza di istruzioni ed operandi siti nella stessa memoria (architettura adottata dal calcolatore EDSAC del 1949)
- Tale modello, per quanto perfezionato nei singoli componenti, è stato fino agli anni 1980 il punto di riferimento per la progettazione di un qualsiasi elaboratore elettronico e fonte di sviluppo dei moderni processori



John von Neumann

ELABORATORE ELETTRONICO

Macchina di von Neumann: schema





Architettura degli elaboratori

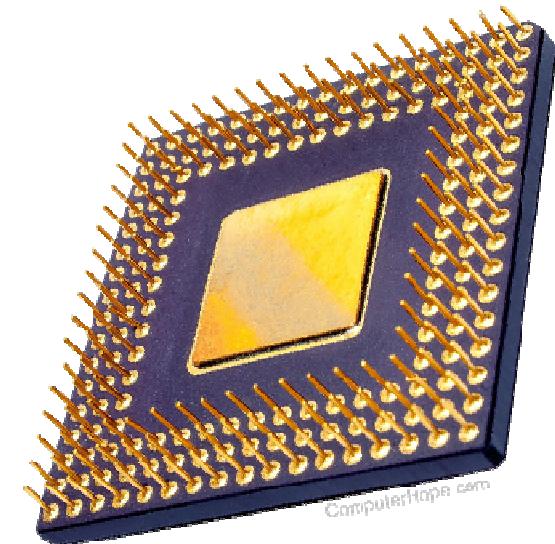
CPU



MACCHINA DI VON NEUMANN

CPU

- ❑ L'Unità di Elaborazione (o *Central Processing Unit*, CPU) è il modulo che si occupa di eseguire le istruzioni di un programma residente in Memoria
- ❑ L'Unità di Elaborazione preleva un'istruzione alla volta dalla Memoria Centrale, la riconosce e genera i comandi per eseguirla, attivando tutte le Unità interessate
- ❑ Oltre a questo monitora e coordina i comandi erogati da se stessa e i segnali provenienti dalle altre Unità (es.: un disco magnetico che richiede un trasferimento di dati nella Memoria Centrale, l'invio di dati immessi da tastiera, l'assegnazione del bus ad una periferica,...)

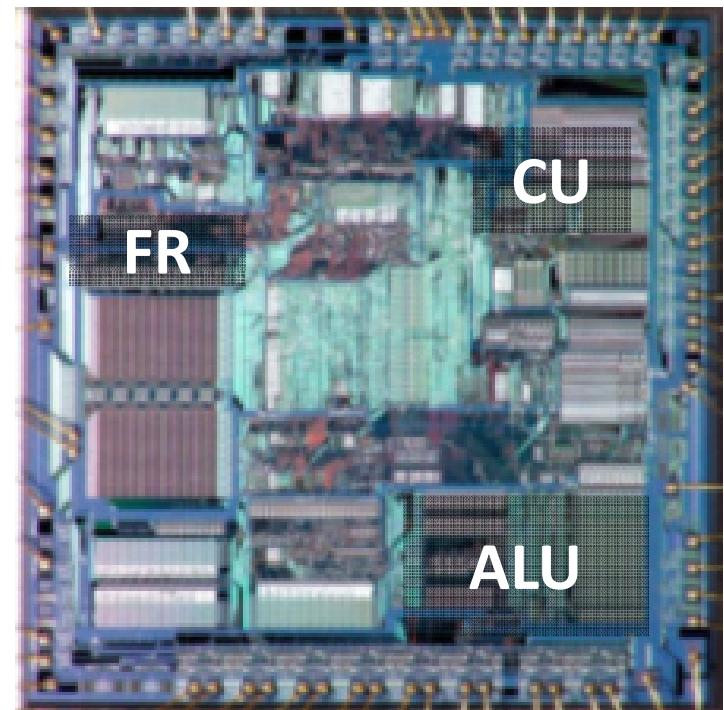




MACCHINA DI VON NEUMANN

CPU: unità di controllo e unità di calcolo

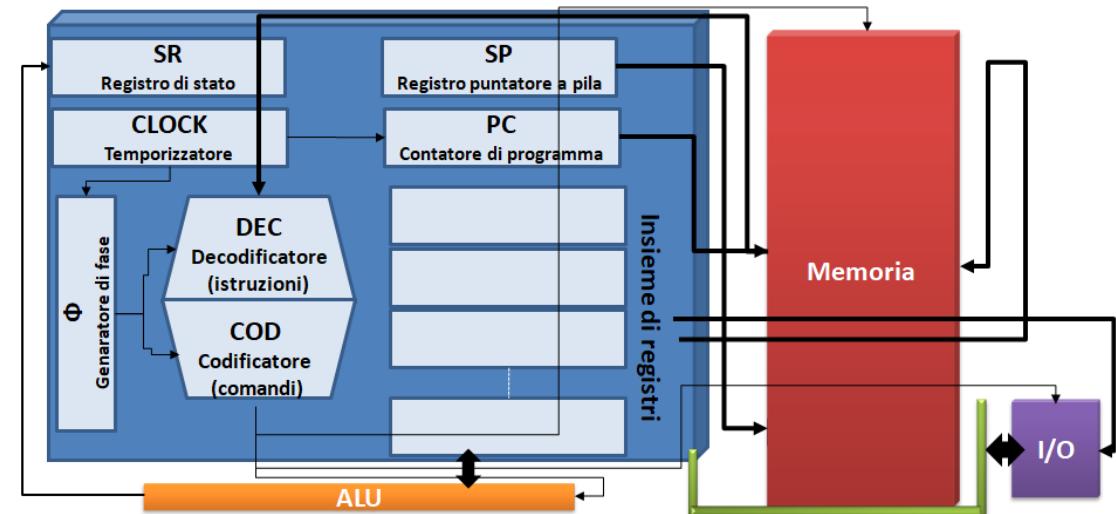
- L'Unità di Elaborazione è suddivisa in due parti:
 - l'**Unità di Controllo**, che preleva l'istruzione dalla memoria, la riconosce, genera i comandi per la sua elaborazione (o per il trasferimento verso altri moduli) e svolge il monitoraggio sul corretto funzionamento della macchina
 - l'**Unità di Calcolo**, che esegue le operazioni logiche e aritmetiche,
- Nei moderni calcolatori i circuiti che compongono l'Unità di Elaborazione (registri, reti sequenziali e combinatorie) sono realizzati con dei transitori e risiedono su una **piastrina di silicio**, il *chip*



MACCHINA DI VON NEUMANN

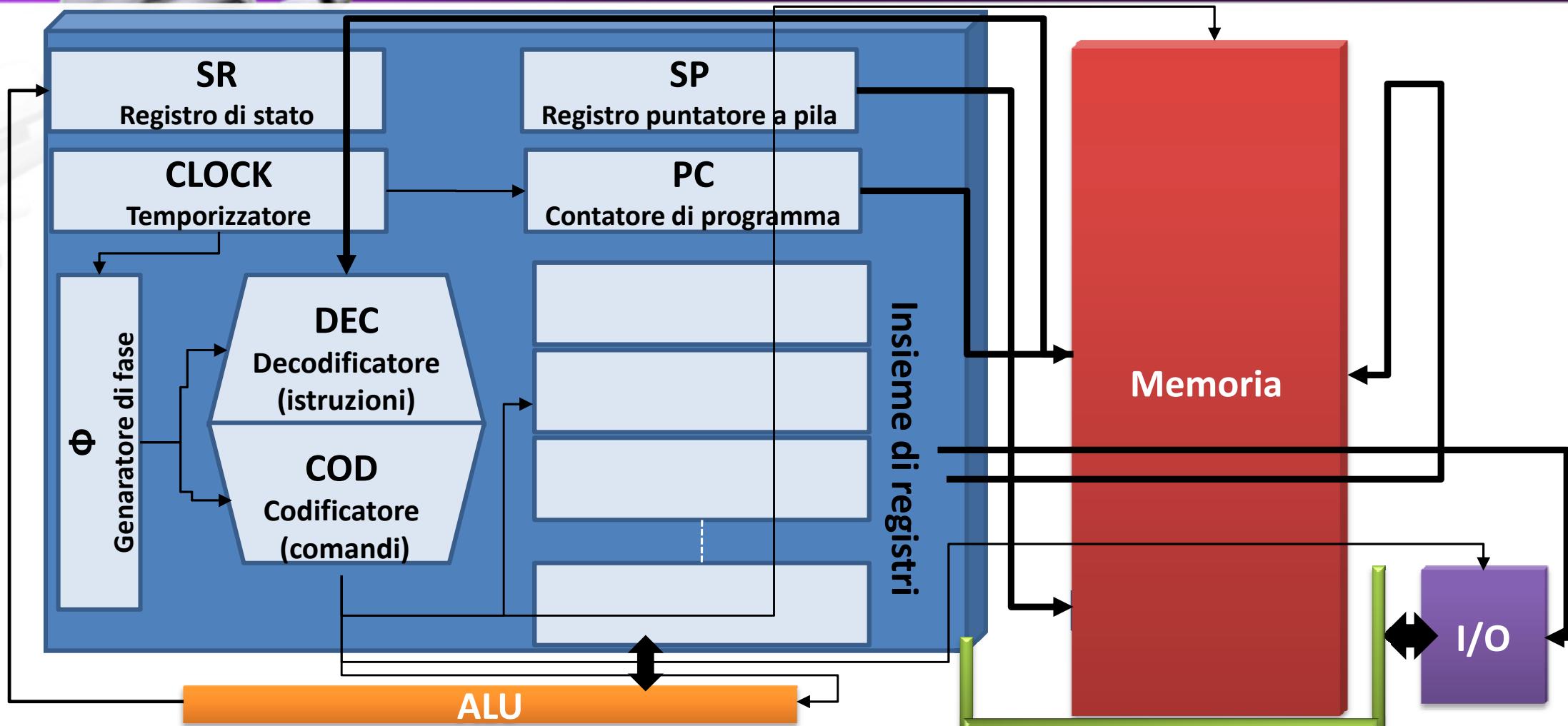
UNITÀ DI CONTROLLO

- L'Unità di Controllo (Control Unit, CU) è costituita da un **insieme di registri** e una **circuiteria** (realizzata con una rete combinatoria o una rete sequenziale) predisposta a prelevare ed elaborare istruzioni contigue presenti nella Memoria Centrale
- La circuiteria dell'Unità di Controllo deve riconoscere e generare i **comandi** atti all'esecuzione di ciascuna istruzione (attivazione della struttura di interconnessione, passaggio dei dati e degli indirizzi in memoria, calcoli aritmetici o logici,...)



MACCHINA DI VON NEUMANN

UNITÀ DI CONTROLLO

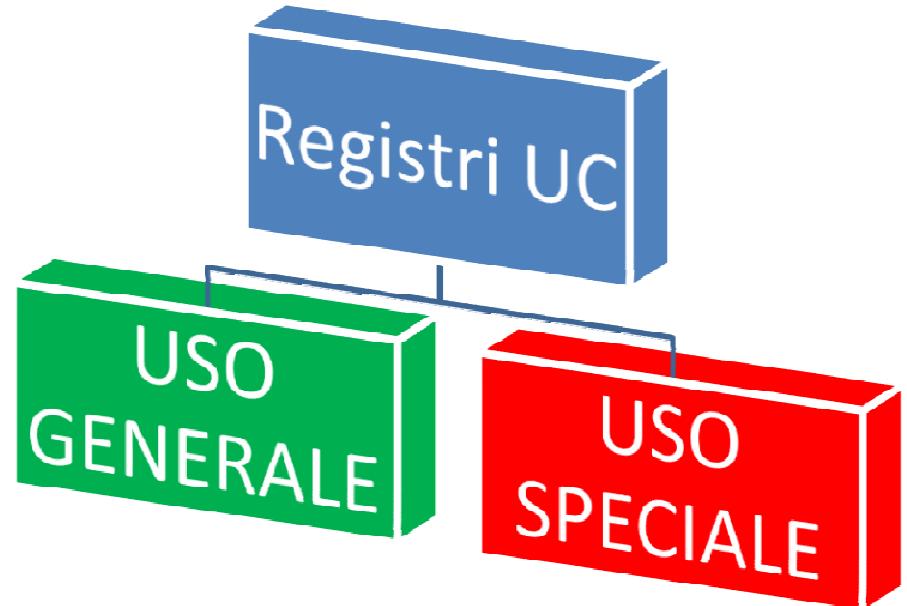




MACCHINA DI VON NEUMANN

UNITÀ DI CONTROLLO: Registri

- Nell'Unità di Controllo sono presenti dei **registri** in cui sono memorizzati dati (operandi o indirizzi)
- Le differenze principali tra i registri e le celle di memoria è l'esiguo numero dei primi (sono realizzati con componenti costosi e performanti) e la loro presenza all'interno della CU garantisce uno scambio di informazione diretto e rapido (si può ricorrere ad una interconnessione di tipo mesh)
- I registri sono classificati come registri ad **uso generale** e ad **uso speciale**

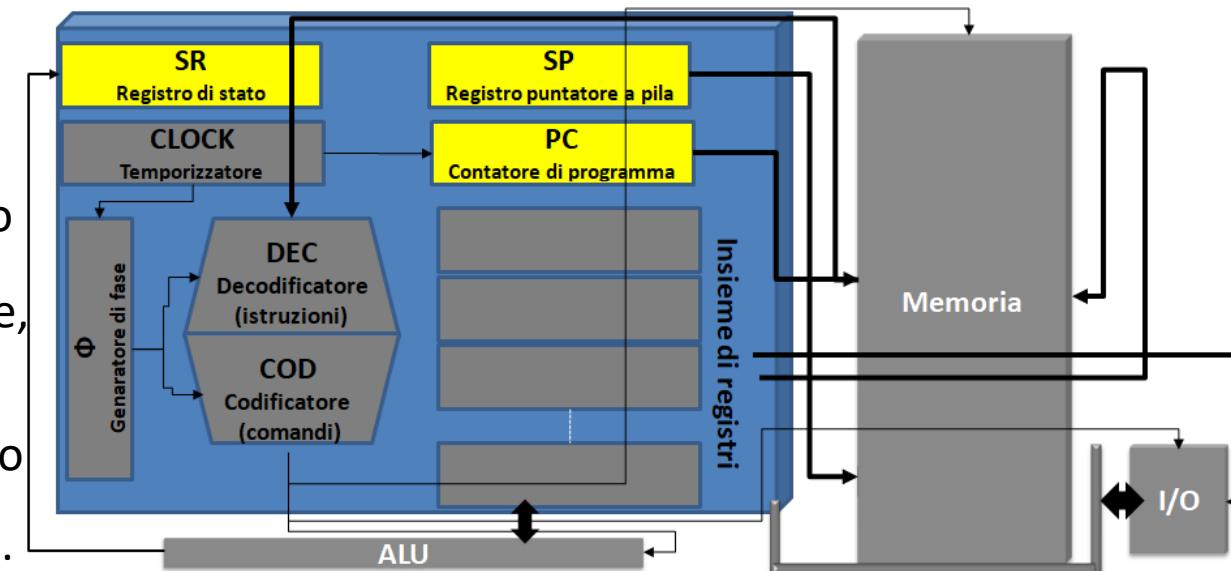


MACCHINA DI VON NEUMANN

UNITÀ DI CONTROLLO: registri ad uso speciale

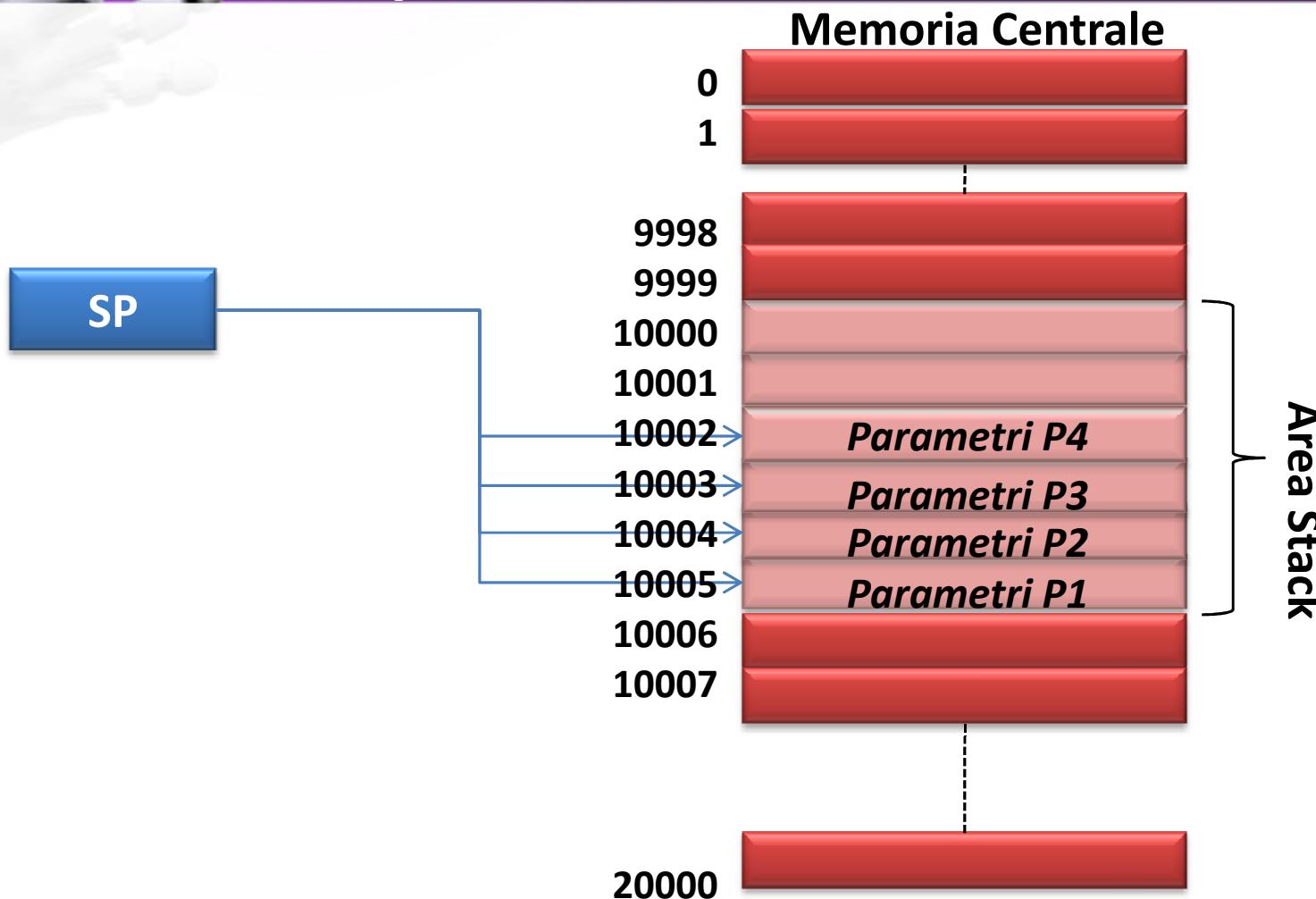
□ I registri ad uso speciale sono :

- Il **Contatore di Programma** (*program counter*, PC): è un registro contatore preselezionabile incrementato ad ogni colpo di clock, contenente l'indirizzo della locazione di Memoria dove è sita l'istruzione da elaborare
- Il **Registro di Stato** (*status register*, SR o *processor status word*, PSW): contiene informazioni sullo stato dell'elaboratore, tra cui, ad esempio, quelle relative all'ultima operazione eseguita (i *condition codes* provenienti dalla ALU) o il verificarsi di una interruzione
- Il **Puntatore alla Pila** (*stack pointer*, SP): contiene l'indirizzo della cima della pila (o canasta) cioè la zona di memoria usata per il passaggio di parametri tra sotto programmi



MACCHINA DI VON NEUMANN

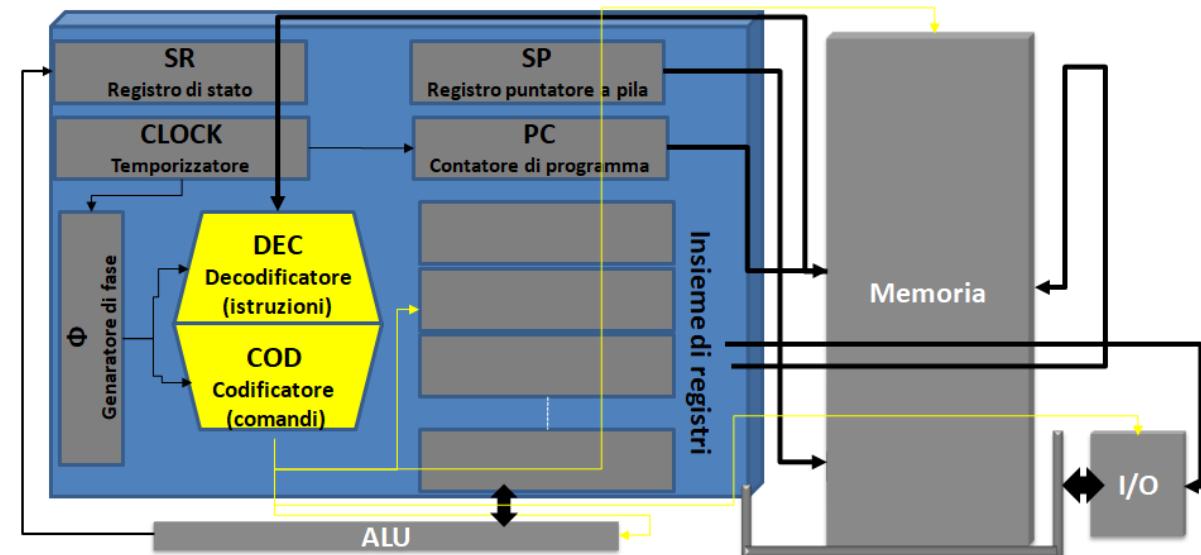
Lo stack pointer e lo stack di memoria



MACCHINA DI VON NEUMANN

UNITÀ DI CONTROLLO: transcodificatore

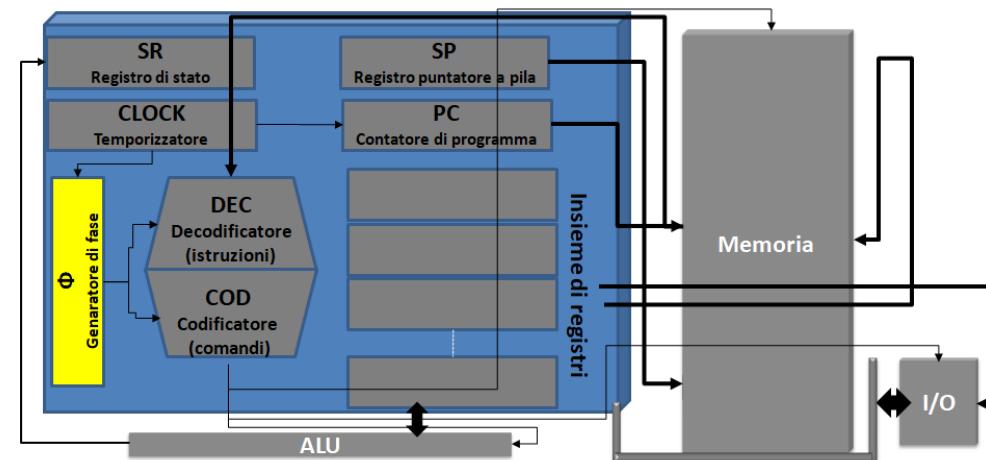
- Una volta che l'istruzione è prelevata dalla Memoria Centrale, grazie all'indirizzo presente nel Program Counter, viene trasferita nel **transcodificatore** (cioè il decodificatore delle istruzioni connesso col codificatore dei comandi) che riconosce l'istruzione e genera i comandi per eseguire l'istruzione stessa



MACCHINA DI VON NEUMANN

UNITÀ DI CONTROLLO: generatore di fase

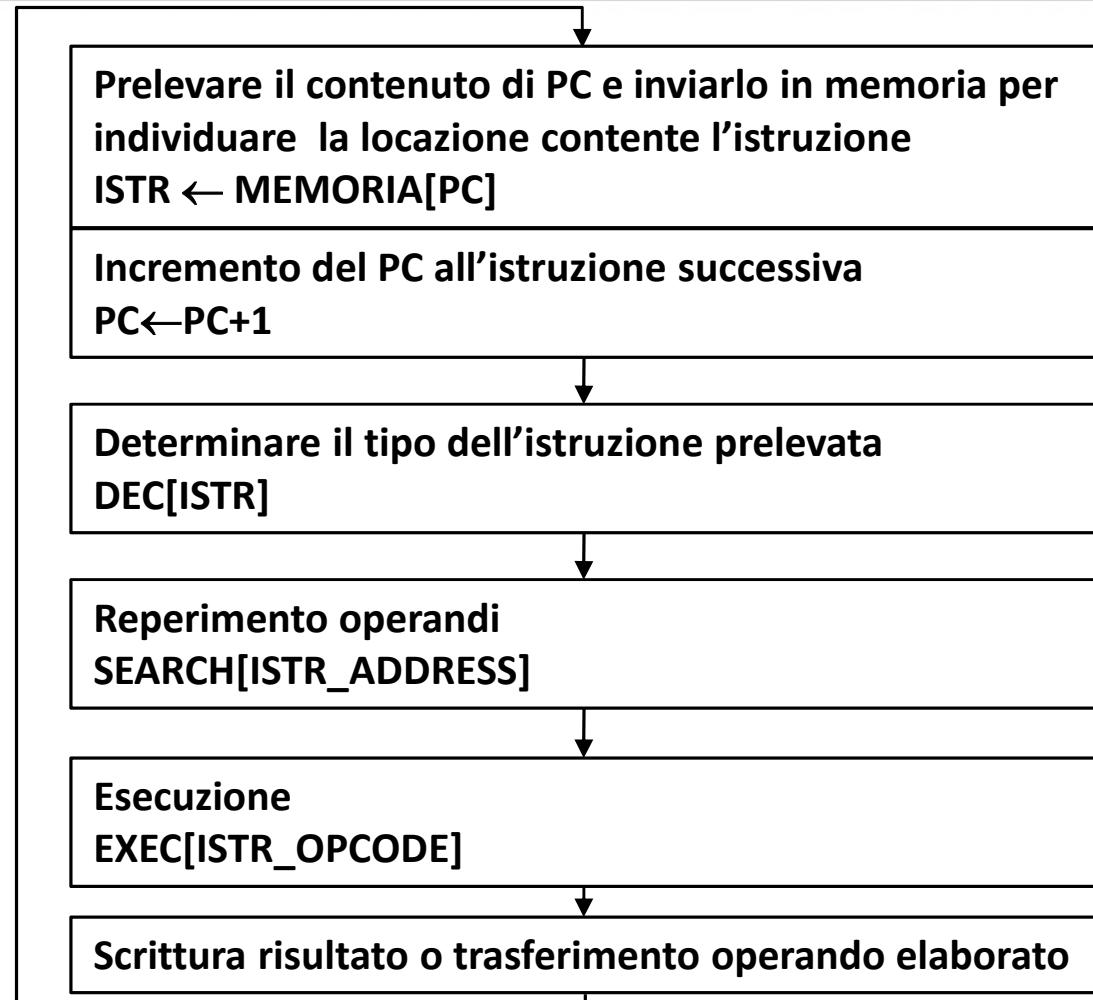
- Il **Generatore di Fase** (tipicamente realizzato con un registro contatore) scandisce le fasi delle operazioni elementari eseguite dalla CU che possono essere così schematizzate:
 - **Caricamento (FETCH)**: lettura dalla locazione di Memoria il cui indirizzo è presente nel Program Counter. In questa fase la Memoria ed il codificatore presente nella CU si connettono per consentire il trasferimento dell'istruzione. Il Program Counter è automaticamente incrementato per prelevare l'istruzione contigua
 - **Decodifica (DECODE)**: interpretazione del tipo di istruzione e del modo di riferimento degli operandi da parte del decodificatore. Non vi è alcuna connessione con componenti esterni perché il decodificatore è interno alla CU
 - **Reperimento degli operandi (MODE)**: si recuperano gli operandi che devono essere processati. Mentre le prime due fasi sono uguali per ogni istruzione, questo passo varia a seconda del modo e del luogo in cui risiedono i dati
 - **Esecuzione (EXECUTE)**: esecuzione dei comandi definiti dal codice operativo dell'istruzione. Vi è connessione con tutte le unità richieste. Anche in questo caso la fase può richiedere più cicli di clock per istruzioni complesse
 - **Spostamento (MOVE)**: il passo nel quale si ripristina il risultato nella destinazione opportuna



MACCHINA DI VON NEUMANN

UNITÀ DI CONTROLLO: prelievo dell'istruzione

- Tutti questi passi sono ripetuti continuamente a partire dall'accensione della macchina
 - Il prelievo dell'istruzione dalla memoria ha il nome di **fase di FETCH** ("andare a prendere")
 - Dopo il Fetch c'è la **decodifica** (DECODE) cioè il riconoscimento delle istruzioni
 - Segue, quando necessario, il **reperimento degli operandi** (MODE)
 - Si effettua l'**esecuzione dell'istruzione** (EXECUTE)
 - Quando necessario avviene l'**archiviazione del risultato** (MOVE) in memoria o nei registri





MACCHINA DI VON NEUMANN

UNITÀ DI CONTROLLO: Tempi di esecuzione delle istruzioni

- Una istruzione è sempre eseguita in queste quattro-cinque fasi ma comprende un **numero di cicli macchina variabile** che dipende, ad esempio, dal tipo di operazione, dal numero di accessi in memoria o dall'interazione con le unità di I/O
- Ogni ciclo macchina, infatti, è implementato mediante una successione di un piccolo numero di operazioni elementari eseguite in circuiti diversi sotto il controllo del transcodificatore
- Ogni operazione elementare occupa un ciclo di clock e pertanto la durata di una istruzione dipende dal numero di accessi alla memoria, all'esterno della CPU e dal numero di operazioni elementari richieste

Istruzione	Significato	Tempo di esecuzione
BNE	Salto condizionato	3E
LOAD <imm>	Caricamento di un valore	2E
DEX	Decremento di una unità	3E
NOP	Nessuna operazione	2E

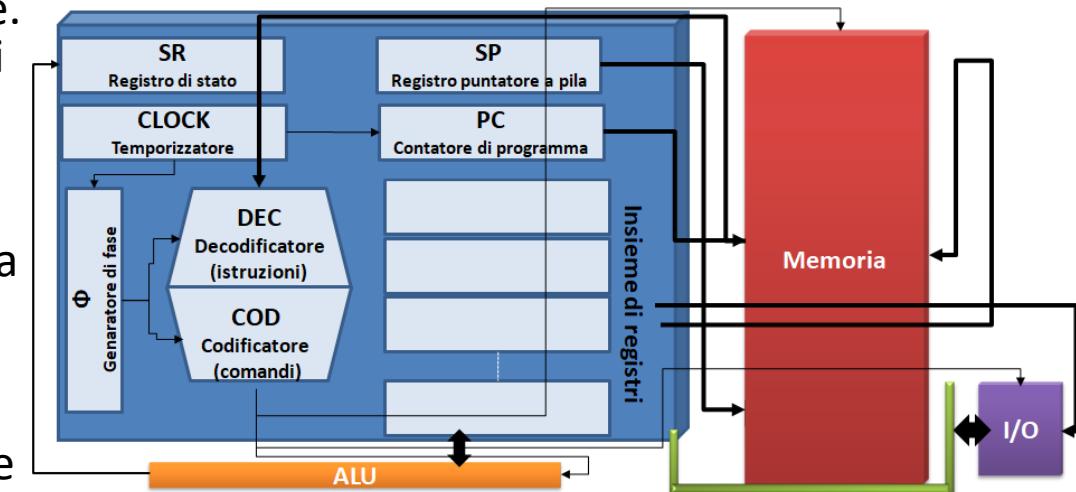
MACCHINA DI VON NEUMANN

UNITÀ DI CONTROLLO: esempio di esecuzione istruzione

□ Esempio

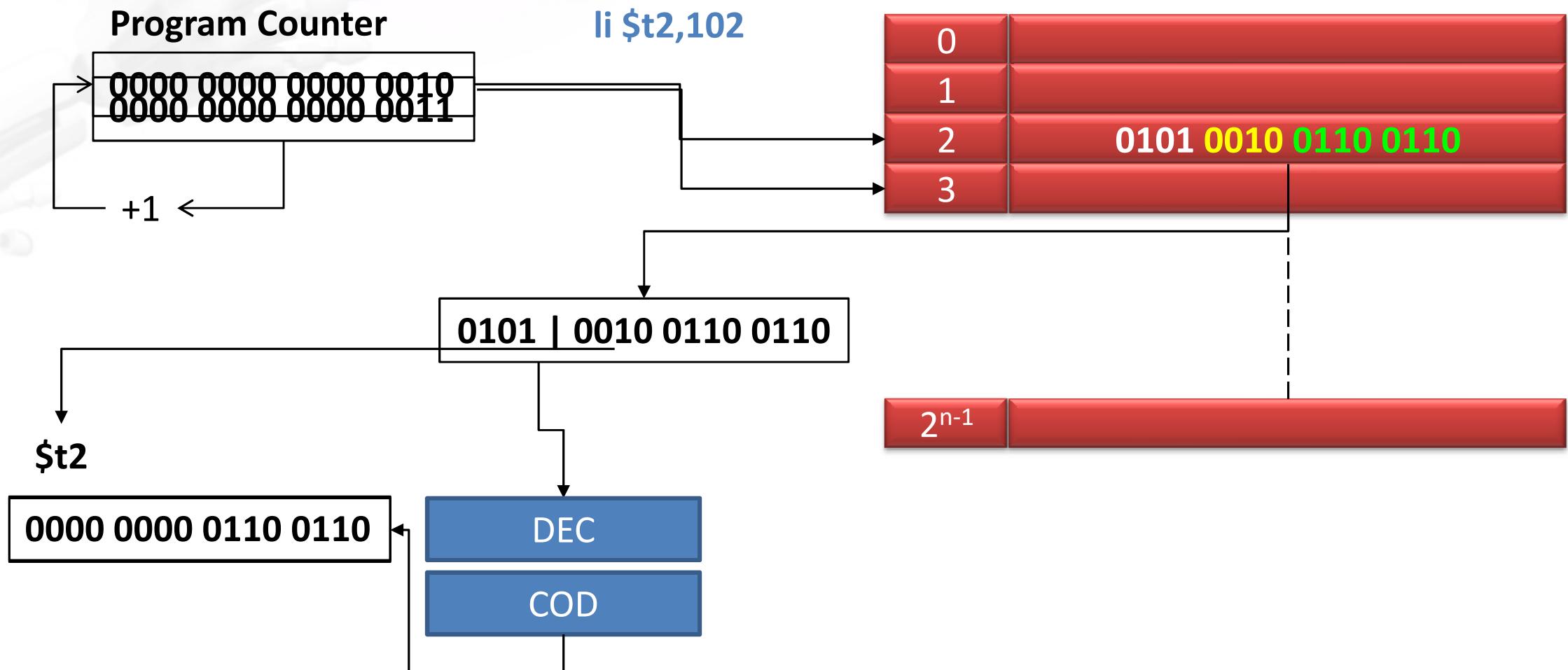
ADDM 0x300,0x100,(0x200)

- ❖ Il decodificatore riconosce l'addizione tra operandi siti in Memoria alle locazioni indicate. Il codificatore, a sua volta, genera dei comandi lungo le linee di ingresso della ALU che specificano di attivare l'addizionatore nell'ALU
- ❖ Contemporaneamente il codificatore invia dei comandi per prelevare gli operandi in memoria alla locazione 0x100 e poi alla locazione il cui indirizzo è specificato nella locazione di memoria 0x200
- ❖ Una volta reperiti gli operandi si deve generare la connessione con la ALU disconnettendo la memoria
- ❖ La ALU esegue l'operazione e aggiorna il registro di stato (il campo Condition Code)
- ❖ Si riattiva la linea con la memoria per trasferire il risultato alla locazione 0x300



MACCHINA DI VON NEUMANN

UNITÀ DI CONTROLLO: esempio di esecuzione istruzione



MACCHINA DI VON NEUMANN

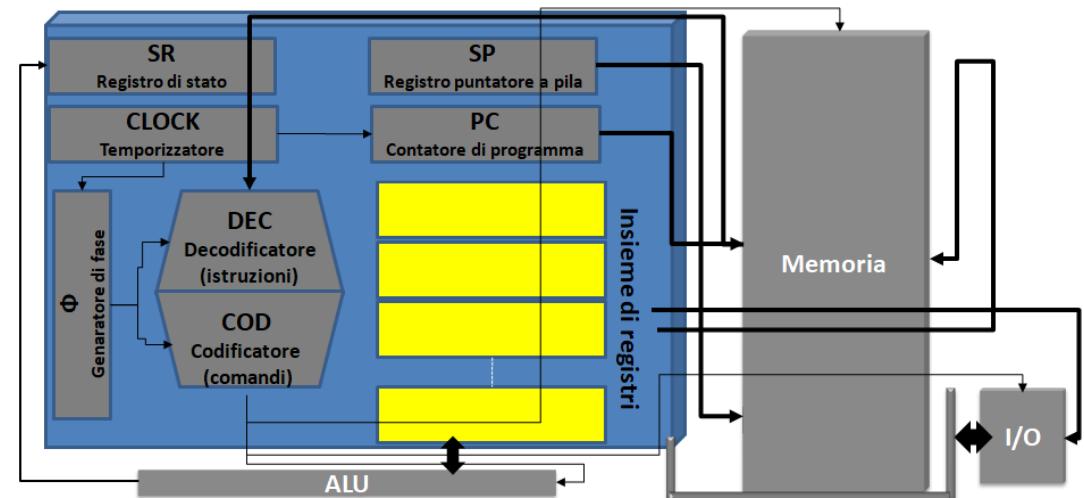
UNITÀ DI CONTROLLO: registri ad uso generale

- L'insieme dei **registri ad uso generale** è anche denominato FR (archivio di registri, *File Register*)
- Tali registri sono utilizzati, tra l'altro, per velocizzare il processo di elaborazione e per memorizzare all'interno dell'Unità di Elaborazione i risultati temporanei, gli operandi e gli indirizzi con maggiore frequenza di uso, allo scopo di diminuire il numero di accessi alla memoria esterna

Curiosità. Il processore Motorola 68000 ha 16 registri ad uso generale classificati in 8 registri dati, predisposti a contenere operandi (D_n) su cui effettuare operazioni, ed 8 registri indirizzi (A_n), in cui tipicamente risiedono indirizzi per accedere a dati in Memoria.

Il processore MIPS 100 ha oltre 30 registri alcuni dei quali riservati (\$ra, \$at).

Intel I7 ha 8 registri a 32bit e 16 a 64bit



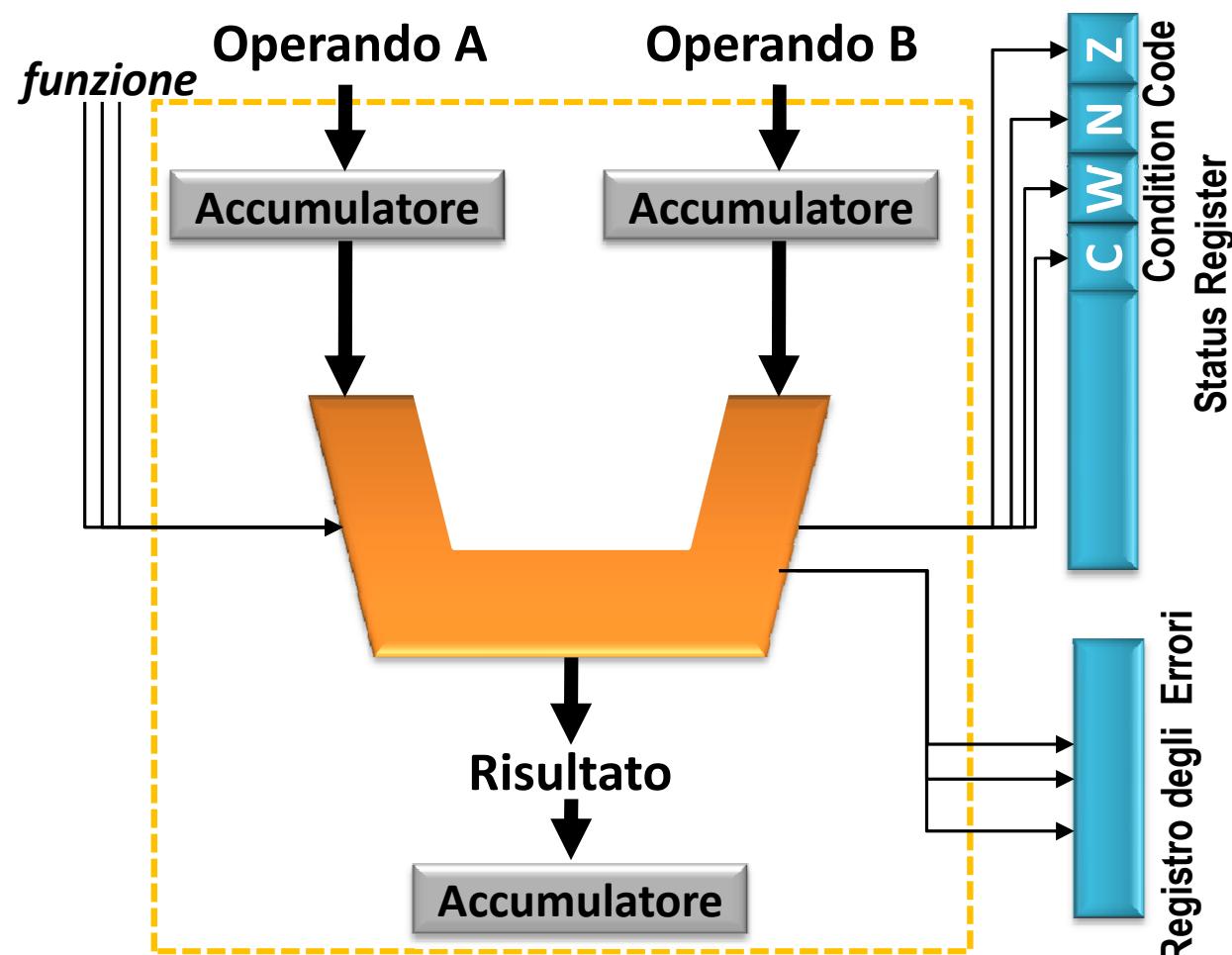
ADD R2,R1,R0

Somma l'operando sito nel registro R0 con l'operando memorizzato nel registro R1 e pone il risultato in R2

MACCHINA DI VON NEUMANN

ALU: funzionamento e accumulatori

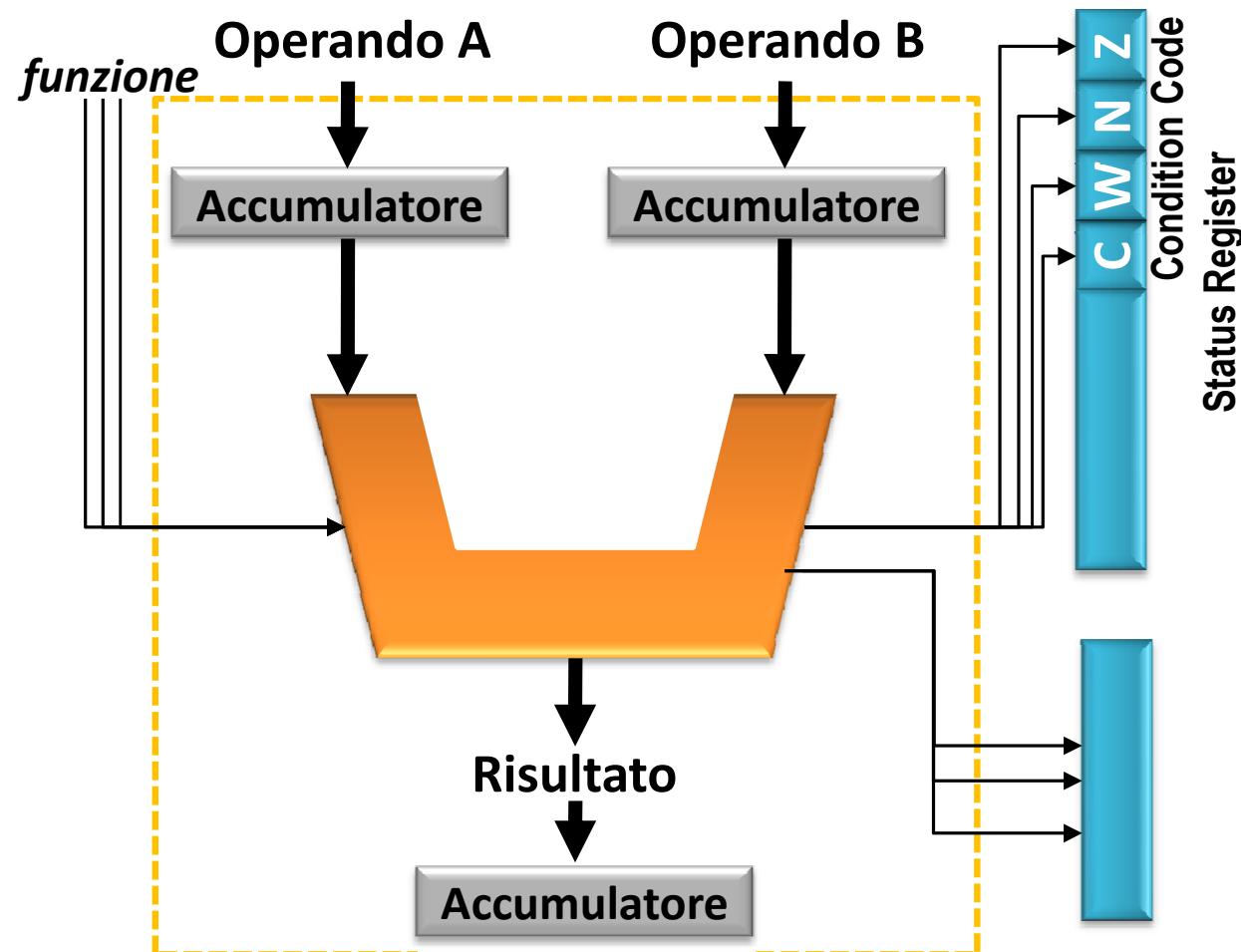
- L'unità logico-aritmetica è il componente che si occupa di effettuare le operazioni logiche ed aritmetiche
- Per il funzionamento di questa unità di solito sono impiegati un insieme di registri, gli **accumulatori**, che servono a contenere gli operandi e il risultato delle operazioni.
- Gli accumulatori sono **trasparenti al programmatore** (cioè il contenuto non può essere modificato dal programmatore mediante istruzioni) e svolgono la funzione di ospitare operandi e risultati prima e dopo l'attivazione del circuito di calcolo
- Gli accumulatori hanno un ruolo fondamentale per l'**indirizzamento implicito**: sono i registri in cui implicitamente sono inviati gli operandi nel momento in cui si ricorre ad una istruzione logico-aritmetica



MACCHINA DI VON NEUMANN

ALU: condition code

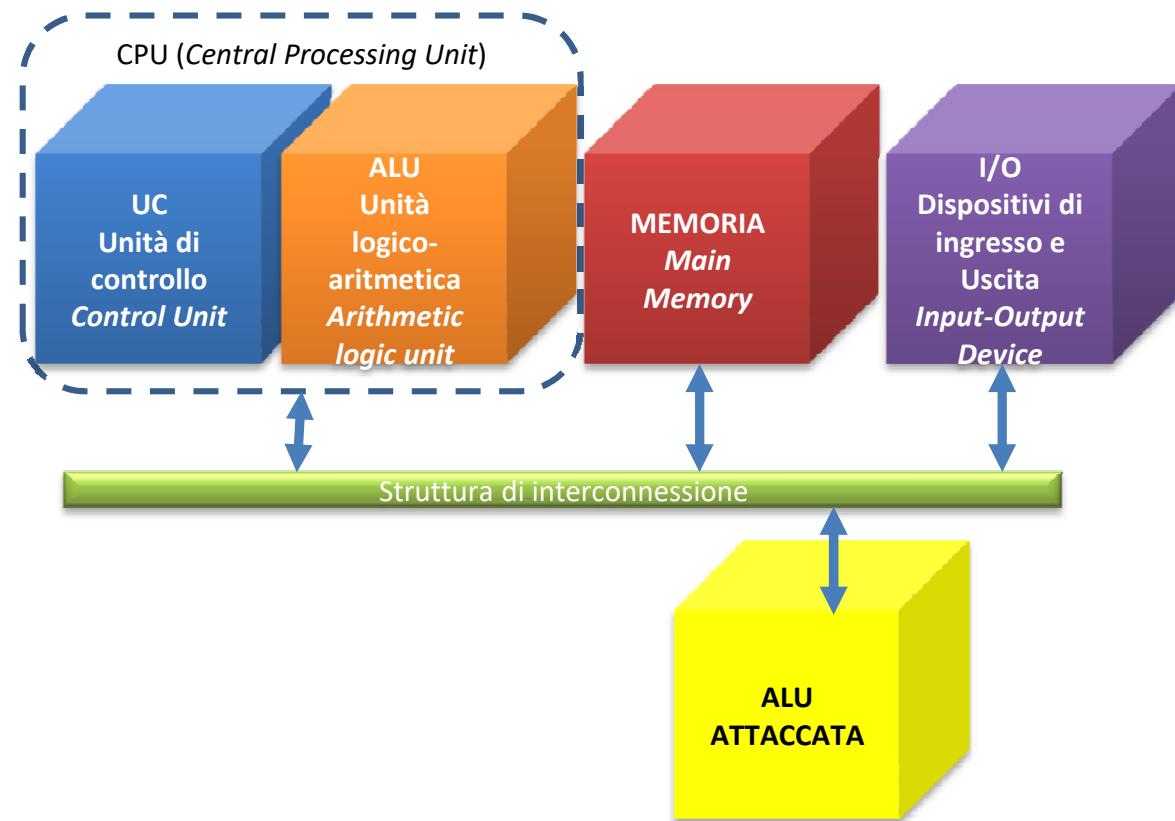
- Oltre agli accumulatori sono presenti dei fasci di **linee di ingresso** su cui viaggiano i comandi che individuano la **funzione/operazione** che deve essere attivata (le operazioni principali sono: ADD, NEG, AND, OR, COMP, TESTB, SHIFT) e un fascio di **linee di uscita** su cui è riportato il **risultato**
- Inoltre ci sono delle **linee di uscita** denominate **flags** che riportano informazioni relative all'ultima operazione eseguita e sono memorizzati tipicamente nel campo del registro di stato noto come **Condition Code**
- Dalla ALU fuoriescono anche delle linee che vanno in un registro speciale, detto **Registro degli Errori**, in cui sono riportate situazioni non risolvibili (es.: divisione per zero, radice quadrata di un numero negativo) e che grazie al quale è possibile attivare un'**interruzione interna**



MACCHINA DI VON NEUMANN

ALU: coprocessore matematico

- Col tempo si è provveduto a realizzare il **co-processore matematico**, cioè una circuiteria in grado di svolgere funzioni complesse come i calcoli in virgola mobile con un insieme di istruzioni dedicato (MULS, MULD, DIVS, DIVF, COMPS, COMPF,...) non presente in quello della macchina
- Il co-processore matematico non era incluso nel modello di Macchina di von Neumann ma era visto come una unità esterna (**ALU-Attaccata**)
- Sebbene questa strategia ormai è stata abbandonata, perché le funzionalità complesse sono incluse direttamente nella ALU-Nativa, è tuttavia una pratica utilizzata nel caso in cui si voglia aggiungere nuove ALU che fanno operazioni che l'ALU-Nativa non svolge (es.: trasformata Fourier)

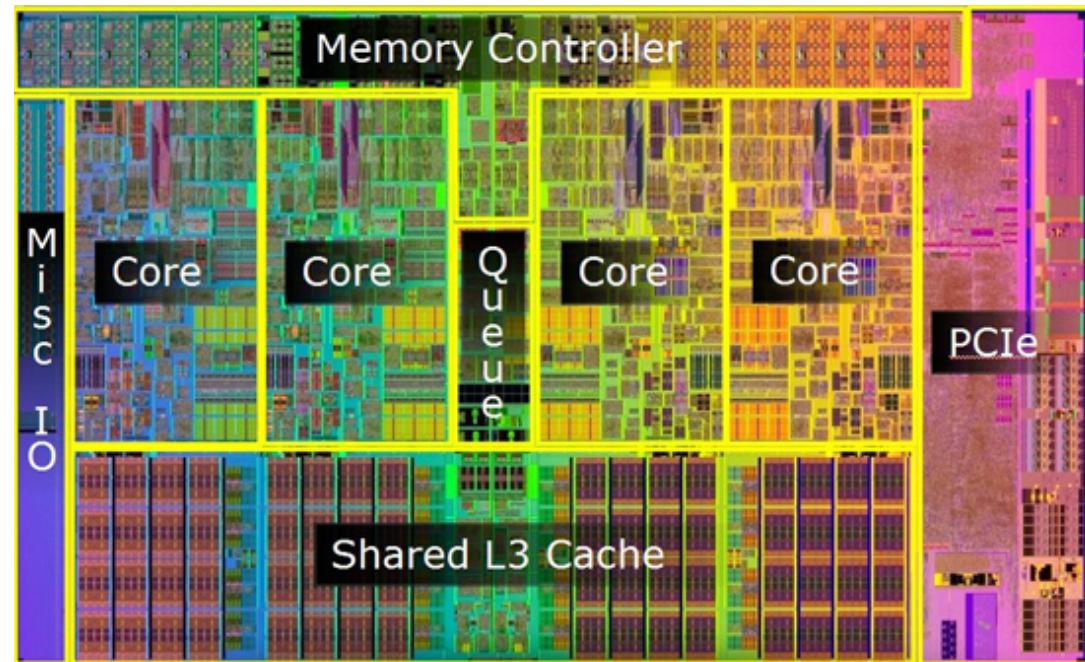




MACCHINA DI VON NEUMANN

CPU (evoluzioni)

- ❑ Nelle architetture moderne il chip ospita anche una memoria ausiliare (più piccola della Memoria Centrale, ma più rapida), nota come **memoria cache** (o memoria di basso livello)
- ❑ In alcuni modelli, il chip ospita anche altri circuiti elettronici che svolgono compiti specifici: il coprocessore matematico per i calcoli trigonometrici, circuiti dedicati per la gestione della grafica e interfacce per i dispositivi di I/O, ed altro
- ❑ Nelle architetture più recenti si supera il modello della Macchina di von Neumann integrando nel chip più Unità di Elaborazioni (**core**) o interconnettendo più processori distinti (**multiprocessor**)





Architettura degli elaboratori

Memoria

MEMORIA

Generalità

- La **Memoria Centrale** è una zona di archiviazione dove risiedono i programmi (istruzioni e indirizzi) e i risultati elaborati (operandi)
- La Memoria Centrale è di tipo **RAM (Random Access Memory)** ovvero ha un accesso ai dati di tipo casuale: si riesce a raggiungere ciascuna locazione in tempo costante





MACCHINA DI VON NEUMANN

MEMORIA

- La **Memoria Centrale** è una memoria volatile di tipo RAM costituita da tante **locazioni** ciascuna delle quali può immagazzinare una stringa binaria di lunghezza finita n pari alla parola dell'architettura
 - Le stringhe memorizzate sono: ***istruzioni, operandi o indirizzi***
 - Le locazioni sono numerate in sequenza da 0 a $2^m - 1$ (con m dimensione massima della memoria) e tale numero prende il nome di **indirizzo** della locazione
 - L'indirizzo specifica **univocamente** una locazione

INDIRIZZO	LOCAZIONE
(0) 00000000	
(1) 00000001	
(2) 00000010	
(3) 00000011	
(4) 00000100	
(5) 00000101	
(6) 00000110	
(7) 00000111	
255) 11111111	



MACCHINA DI VON NEUMANN

MEMORIA

- Per motivi progettuali una locazione di memoria è fisicamente composta da 8 celle di memoria
 - È compito di una circuiteria prossima alla Memoria prelevare, in un colpo di clock, tanti bit quanti compongono la parola della macchina (es.: se l'elaboratore ha parole lunghe 32bit, in un colpo di clock la macchina preleva 4 locazioni contigue in Memoria)

Osservazione: per architetture a 32bit: la scelta di locazioni di 8 celle facilita anche il programmatore quando vuole utilizzare i dati residenti in una locazione fisica (8bit, *byte*); in due locazioni fisiche (16bit, *half word*) o l'intera parola (32bit, *word*)

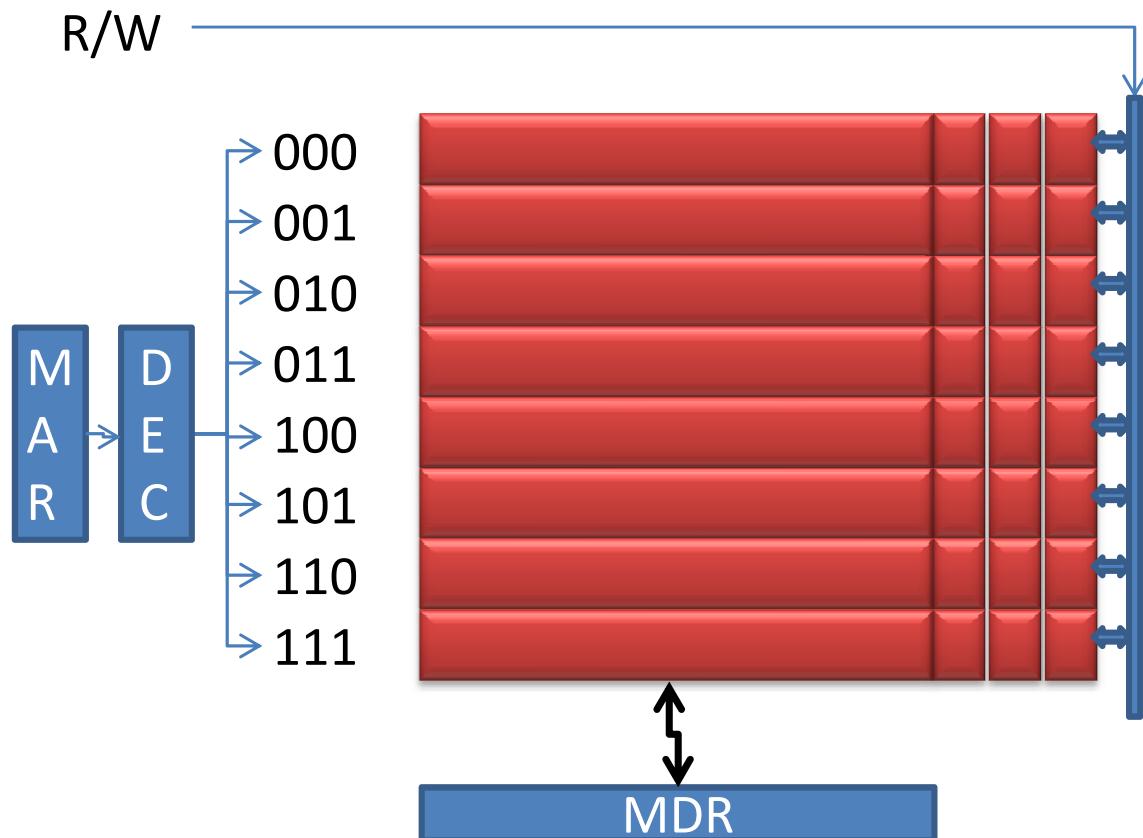
INDIRIZZO	LOCAZIONE
(0) 00000000	
(1) 00000001	
(2) 00000010	
(3) 00000011	
(4) 00000100	
(5) 00000101	
(6) 00000110	
(7) 00000111	
(255) 11111111	

MACCHINA DI VON NEUMANN

MEMORIA: Componenti e registri

- Per poter operare sulla memoria è necessario che ci siano:
 - **linee di ingresso** che specificano un indirizzo (in alcuni testi si fa riferimento al registro **MAR**, *memory address register*)
 - **linee di uscita** per poter inviare o trasferire il dato (in alcuni testi si fa riferimento al registro **MDR**, *memory data register o*)
 - **un segnale di controllo** (un comando generato dalla CU) per la lettura o la scrittura del dato

Osservazione. Oltre al segnale di controllo ce ne possono essere altri: protezione da scrittura della cella, impostazione del bit di parità,....

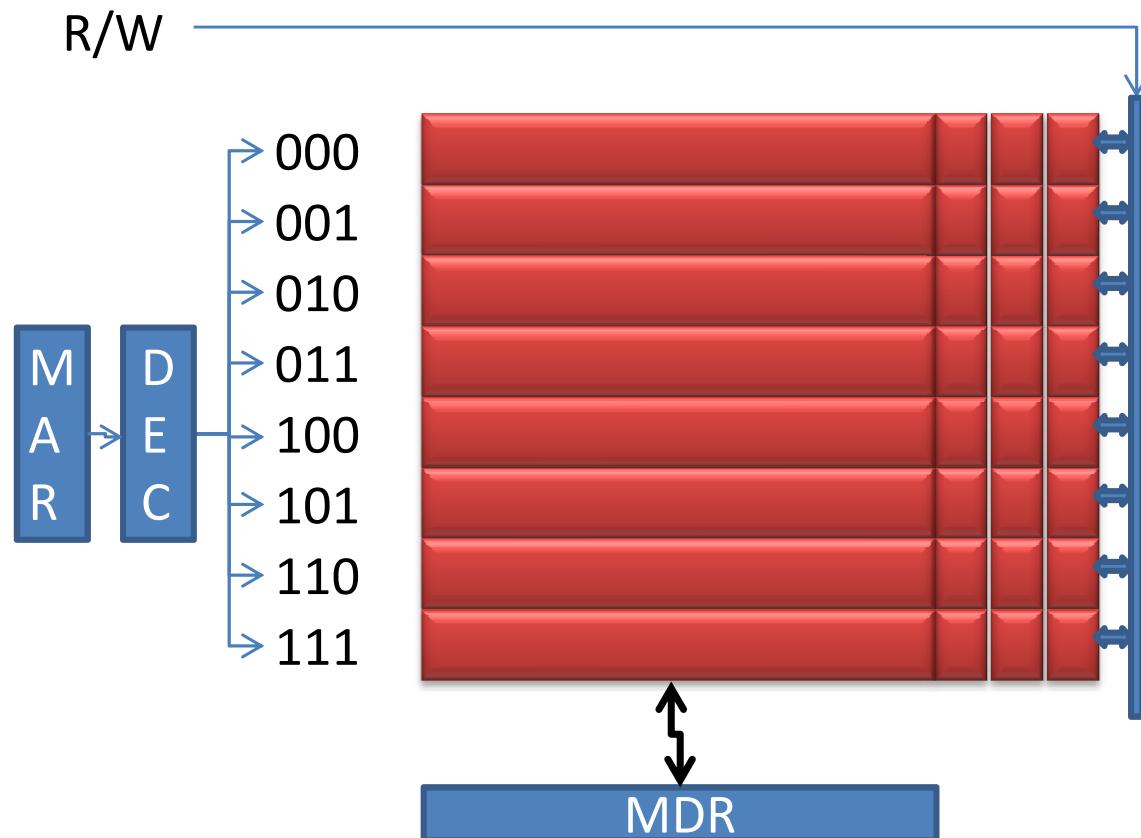




MACCHINA DI VON NEUMANN

MEMORIA:organizzazione fisica

- È prevista, pertanto, una architettura costituita da un **decodificatore** che riceve in ingresso l'indirizzo della cella di memoria alla quale si vuole accedere ed una linea che abilita questa a porre il suo contenuto in uscita dalla memoria o trascrivere in essa il dato da memorizzare



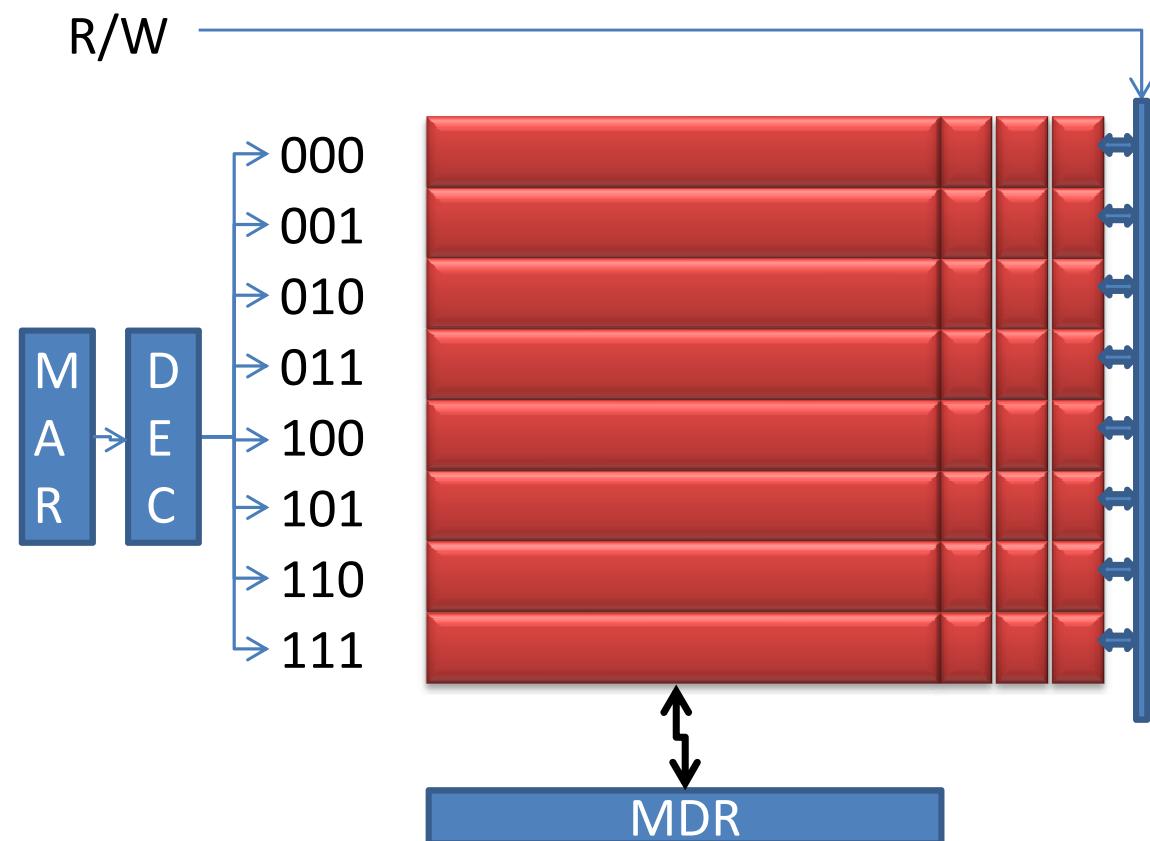
MACCHINA DI VON NEUMANN

MEMORIA: organizzazione fisica (limiti)

- Una organizzazione di questo tipo è **impraticabile** nel caso in cui la memoria abbia una grande dimensione. Con indirizzi, di lunghezza m , è possibile indirizzare 2^m celle di memoria. Pertanto se il valore m è grande (supera la decina di unità) una architettura gestita da un singolo decoder è da escludere

Osservazione. Dal 2010 si usano sistemi con parole di lunghezza 32; pertanto è possibile indirizzare $2^{32} = 4294967296$ celle di memoria. Un decoder che abbia 32 linee di ingresso e più di 4 miliardi di linee di uscita è una possibilità progettuale da escludere.

Inoltre dal 2014 i sistemi impiegano parole da 64bit

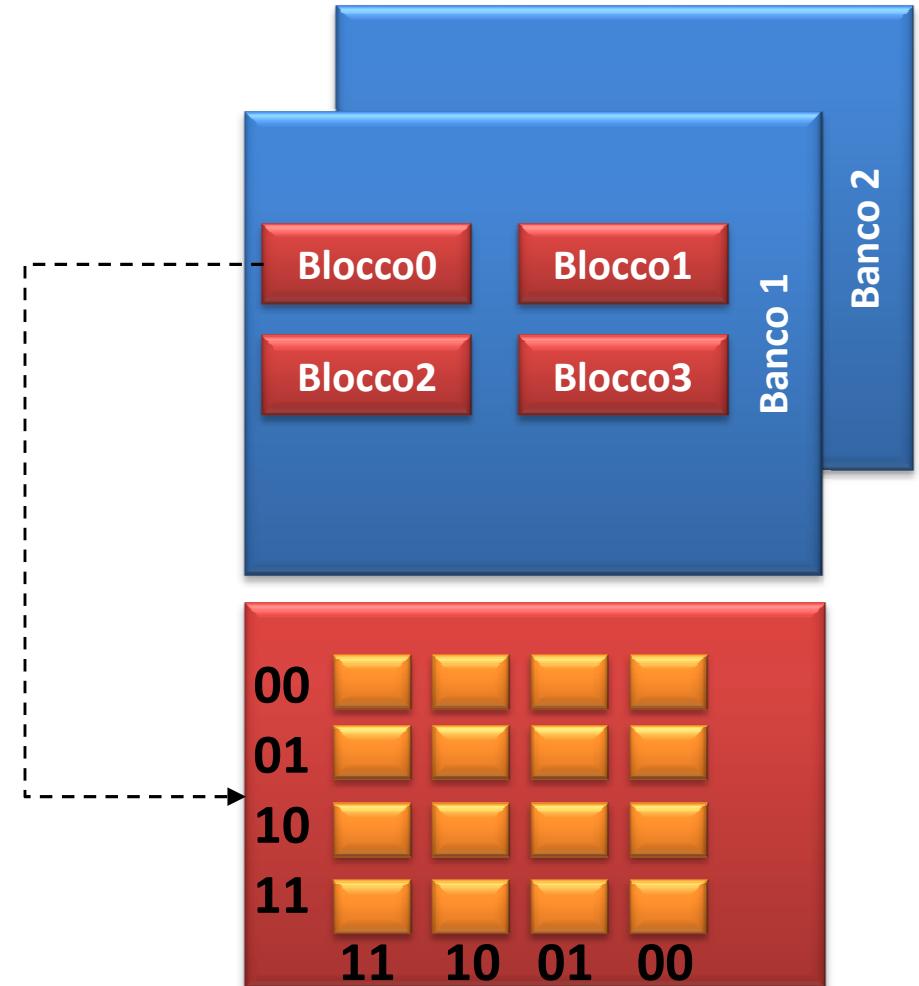




MACCHINA DI VON NEUMANN

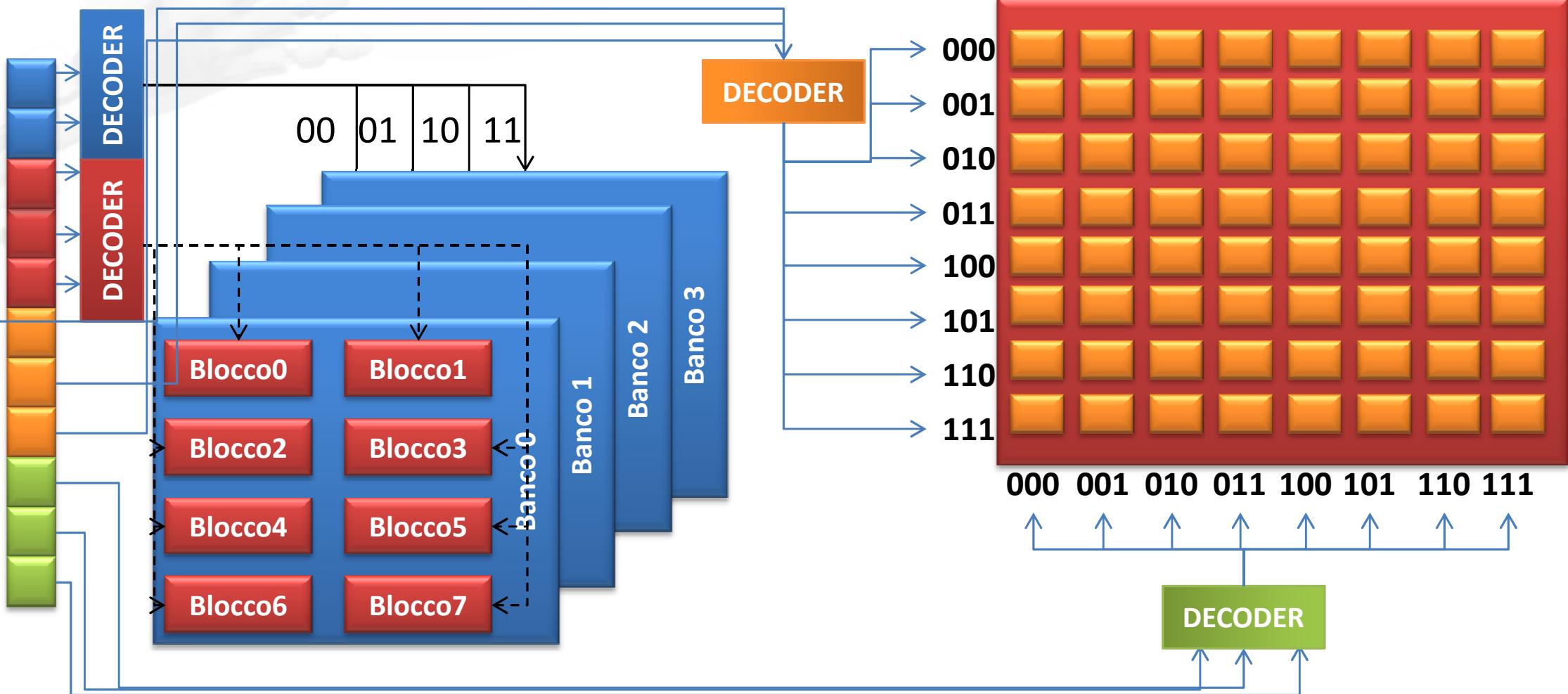
MEMORIA: suddivisione logica

- Per questo si ricorre ad una **suddivisione logica della memoria** (*multi-dimensione*)
- La memoria è partizionata logicamente in **banchi** (o piastre), **blocchi** e **locazioni**
- L'indirizzo è suddiviso in campi ognuno dei quali fa riferimento ad un partizionamento
Ad esempio nel caso si abbia una memoria costituita da 2 banchi, 4 blocchi e 16 locazioni per blocco avremo una suddivisione dell'indirizzo in tre campi : il primo campo (di un bit) per discriminare il banco; un altro campo (di due bit) per individuare il blocco e poi due campi (di 2 bit ciascuno) che individuano la posizione in cui risiede la locazione
- Per ogni campo si usa un **decoder**, minimizzando in questo modo il numero di linee di connessione
- In questo modo il risparmio in termini di linee di connessione tra decoder e celle di memoria è del 90%-99%



MACCHINA DI VON NEUMANN

MEMORIA: suddivisione logica





MACCHINA DI VON NEUMANN

MEMORIA: organizzazione dati

- I dati in memoria possono avere due tipi di **ordinamento (endian)**: la numerazione comincia a partire dall'estremo più “grande” (cioè dal byte più significativo) è chiamato *big endian*, in contrapposizione c’è il sistema *little endian*

- Intel, Digital (*little endian*)
 - Motorola, IBM, SUN (*big endian*)
 - MIPS può essere impostato in entrambe le organizzazioni

320246224	(00010011 00010110 10010001 11010000) ₂
LITTLE ENDIAN del valore 320246224	
100	11010000
101	10010001
102	00010110
103	00010011
BIG ENDIAN del valore 320246224	
100	00010011
101	00010110
102	10010001
103	11010000



MACCHINA DI VON NEUMANN

Suddivisione logica della memoria

- La Memoria Centrale presenta delle **aree riservate**, in cui risiedono delle informazioni basilari utili al funzionamento della macchina (Sistema operativo), ed altre in cui, per comodità sono riservate per operazioni particolari (stack, zona per trasferimento I/O,...)

SISTEMA OPERATIVO
(Kernel)

AREA DEI PROGRAMMI

AREA PER STRUTTURE DATI DINAMICHE (liste, grafi,...)

STACK

Moduli di I/O

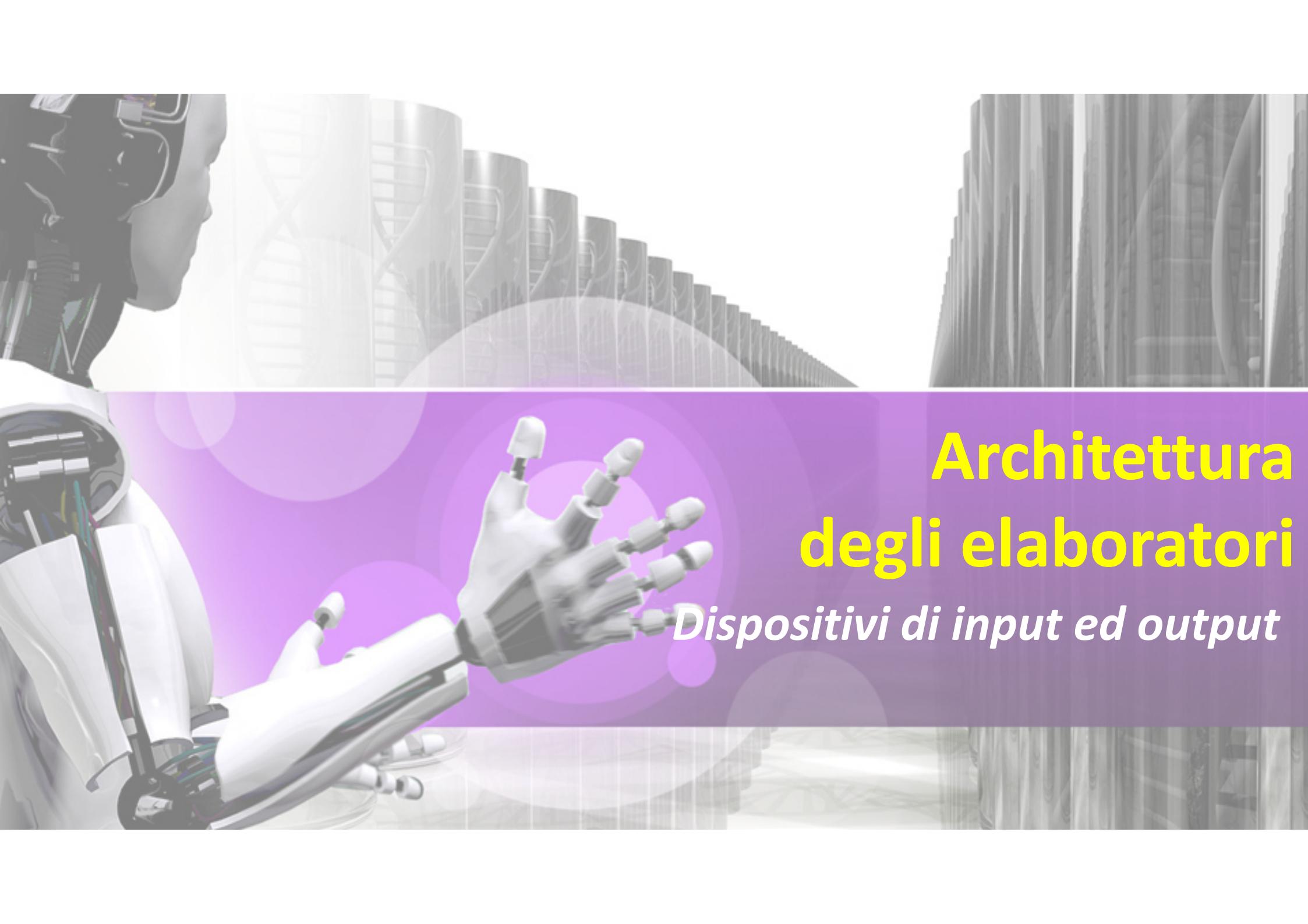
MEMORIA

Tipologie

□ Esistono diverse tipologie di memoria:

- **SRAM** (Static RAM): Memorie statiche nelle quali l'informazione è memorizzata nell'equivalente di un latch D. La SRAM è usata come **memoria cache** per il processore e quando installata sul chip non può essere modificata né sostituita (quest'ultima è una scelta progettuale frequente nelle macchine RISC e nei processori multi-core).
- **DRAM** (Dynamic RAM) Memorie dinamiche nelle quali l'informazione è memorizzata in un condensatore. Anche in presenza della tensione di alimentazione l'informazione contenuta in ogni cella è conservata per un breve periodo di tempo (dell'ordine di grandezza di 2 ms), passato il quale il contenuto deve essere ripristinato: questo avviene ciclicamente tramite un'operazione di "rinfresco" (*refresh*) che è effettuata dal sistema. La DRAM (sono i moduli di colore verde che si incastrano negli slot della scheda madre) è meno costosa da produrre, ma ha tempi di risposta più lenti di una SRAM e per questo costituisce la tecnologia che comunemente realizza la Memoria Centrale
- **SDRAM** (Synchronous DRAM) consente una maggiore flessibilità di impiego permettendo, grazie ad un apposito registro in uscita, di modificare il dato contenuto in una cella mentre si sta utilizzando il vecchio dato. Una ulteriore evoluzione della SDRAM è la DDR (Double Data Rate) che, come dice il nome, consente di operare a frequenza doppia potendo essere pilotata sia sul fronte di salita che sul fronte di discesa del clock. Il DDR è più veloce e utilizza meno energia rispetto alla tipologia SDRAM; tanto da essere stata adottata dal 2015 come Memoria Centrale negli elaboratori dedicati al grande pubblico (e non solo).





Architettura degli elaboratori

Dispositivi di input ed output



MACCHINA DI VON NEUMANN

I/O

- Le unità di **input/output** (dispositivi di I/O o periferiche) consentono di collegare il calcolatore, ed in particolare la Memoria Centrale, con il mondo esterno (persone o altri dispositivi)
- Esistono numerosi tipi di dispositivi di I/O con caratteristiche molto varie che comportano problemi relativi alla conversione tra rappresentazione interna ed esterna dell'informazione





MACCHINA DI VON NEUMANN

I/O

- Le **velocità di trasferimento** sono inferiori a quelle possibili all'interno delle altre componenti (CPU, memoria) a causa della natura tecnologica di fabbricazione (componenti meccanici,...) e quindi l'uso delle periferiche comporta problemi di sincronizzazione ed adattamento della velocità

Dispositivo	Comportamento	Dati scambiati (KB/s)
Tastiera	Input	0.01
Mouse	Input	0.02
Stampante ad aghi	Output	1
Floppy	Storage	50
Stampante laser	Output	100
Disco	Storage	10.000
LAN	Input/Output	10.000
Display	Output	30.000



MACCHINA DI VON NEUMANN

I/O: protocollo

- Nel momento in cui c'è un trasferimento dati è opportuno che il dispositivo coinvolto e il processore operino in modo coordinato mediante un insieme di regole: **il protocollo**
- Il protocollo consente l'interazione tra il dispositivo e il processore
 - il dispositivo deve essere in grado di operare qualora il processore ne richieda l'intervento
 - il processore deve eseguire le operazioni che consentono il trasferimento dei dati solo quando il dispositivo è pronto

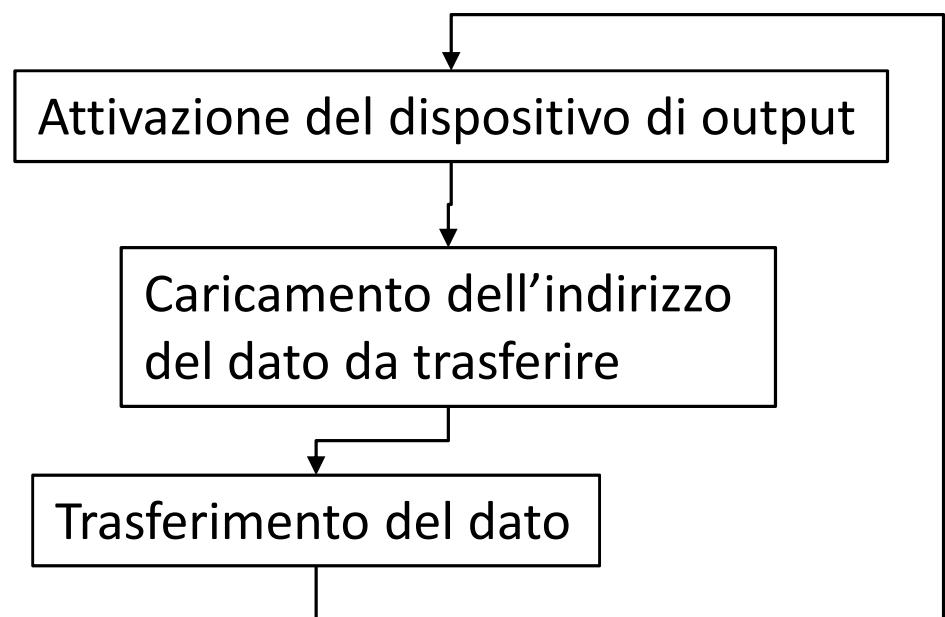




MACCHINA DI VON NEUMANN

I/O protocollo

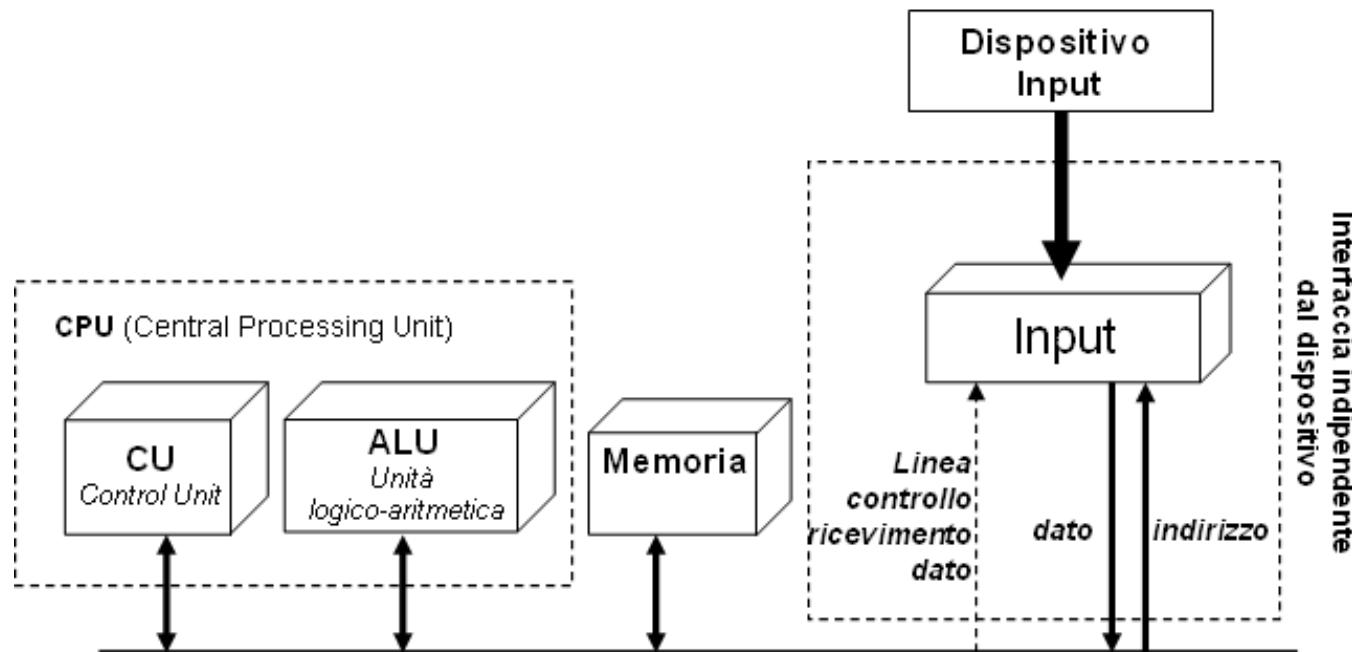
- Per interagire con il processore ogni dispositivo deve essere dotato di un **modulo di I/O** (o modulo di interfaccia I/O), cioè una rete sequenziale che colloquia con il processore inviando e ricevendo (tramite un bus di I/O) i segnali che, secondo il protocollo, controllano le operazioni di trasferimento
- Il protocollo di I/O pertanto è caratteristico dell'elaboratore, in quanto determinato dal modo di operare del processore stesso, cioè dall'insieme di istruzioni di cui il processore può disporre per i trasferimenti. Questo vuol dire che i diversi dispositivi esterni collegati allo stesso calcolatore devono rispettare tutti lo stesso protocollo di I/O, indipendentemente dalla natura delle informazioni trasferite a dalla struttura fisica del dispositivo. Solamente in seguito il dispositivo da il giusto significato al codice ricevuto



MACCHINA DI VON NEUMANN

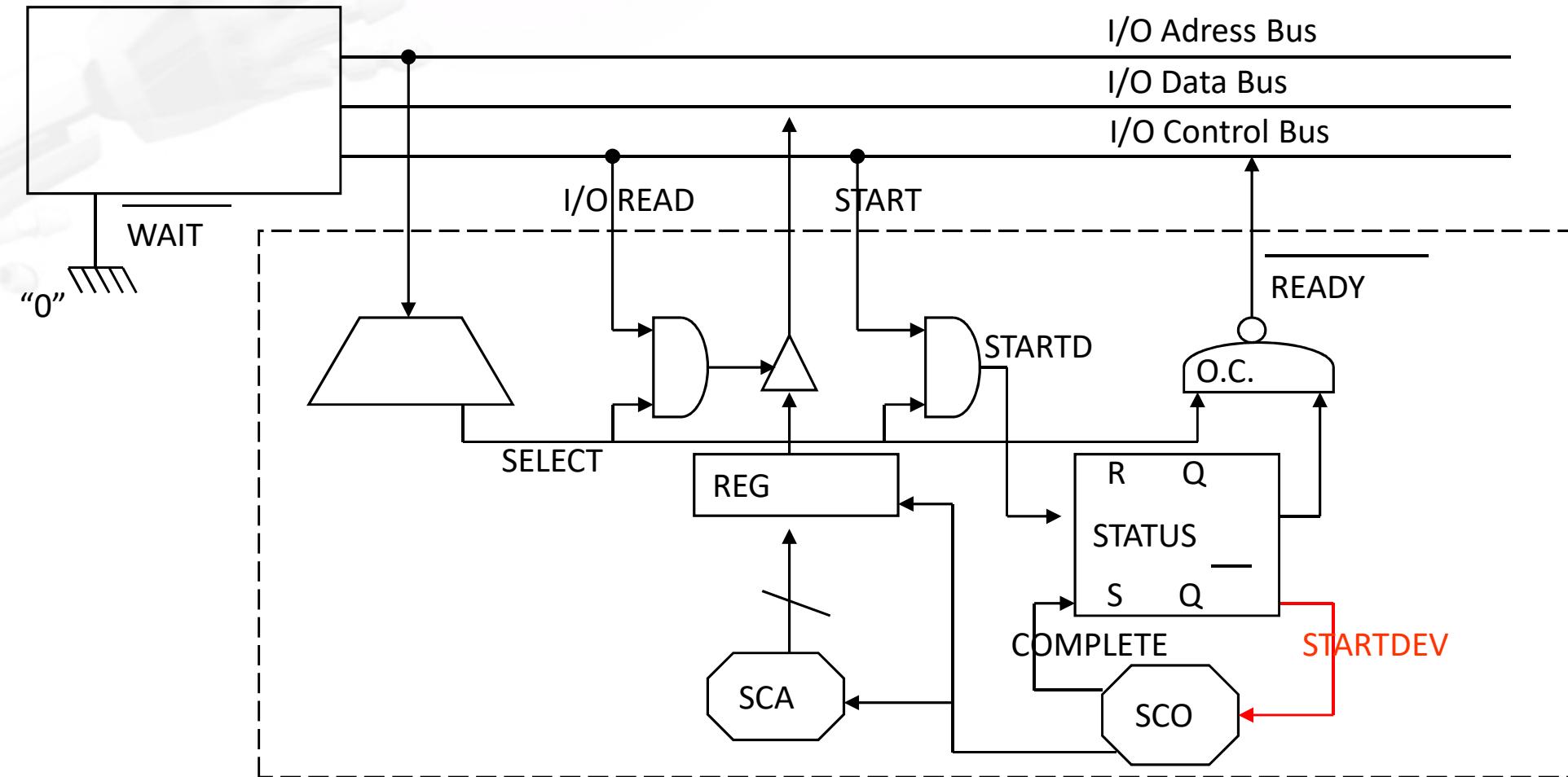
I/O: controller (INPUT)

- La sotto-rete che non dipende dal dispositivo e che è interessata nel colloquio con il processore (*controller*) di un dispositivo di input può essere così semplificata



MACCHINA DI VON NEUMANN

I/O: controller (INPUT)





MACCHINA DI VON NEUMANN

I/O: controller (protocollo INPUT)

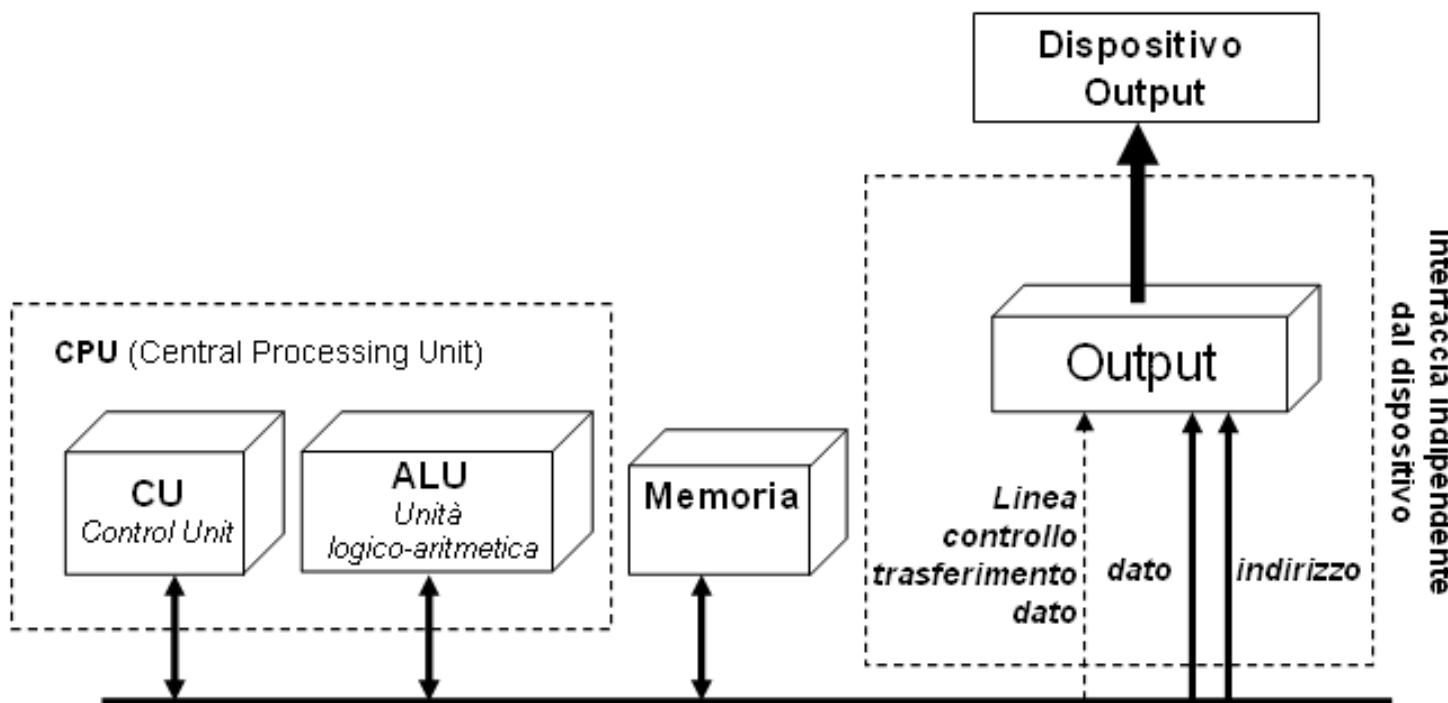
Protocollo INPUT

1. Il processore invia sull'I/O Address bus l'indirizzo del dispositivo e ne esamina lo stato tramite la linea di controllo READY
2. Se il dispositivo non è pronto il processore deve attendere (o elaborare un'altra istruzione e ripetere la richiesta di interazione col dispositivo)
3. Altrimenti il processore può prelevare il dato prodotto dalla periferica (seleziona il dispositivo tramite le linee indirizzi e invia il segnale START). START resetta il flip-flop STATUS a 1 e in tale stato rimane per tutta la durata delle operazioni di produzione del dato da parte del dispositivo
4. Quando il dato è stato prodotto ed è disponibile in REG, il dispositivo genera il segnale COMPLETE, settando STATUS (READY=0)
5. Nel frattempo il processore, in attesa del dato, esamina lo stato campionando il segnale READY
6. Se READY= 1 il processore deve attendere e tornare al punto 5.
Se READY= 0 il processore esegue una istruzione di INPUT (seleziona il dispositivo ed invia il segnale di controllo IO READ per trasferire il dato presente in REG in memoria; per questo motivo il processore deve anche indicare l'indirizzo di memoria dove ospitare il dato e il comando di scrittura)

MACCHINA DI VON NEUMANN

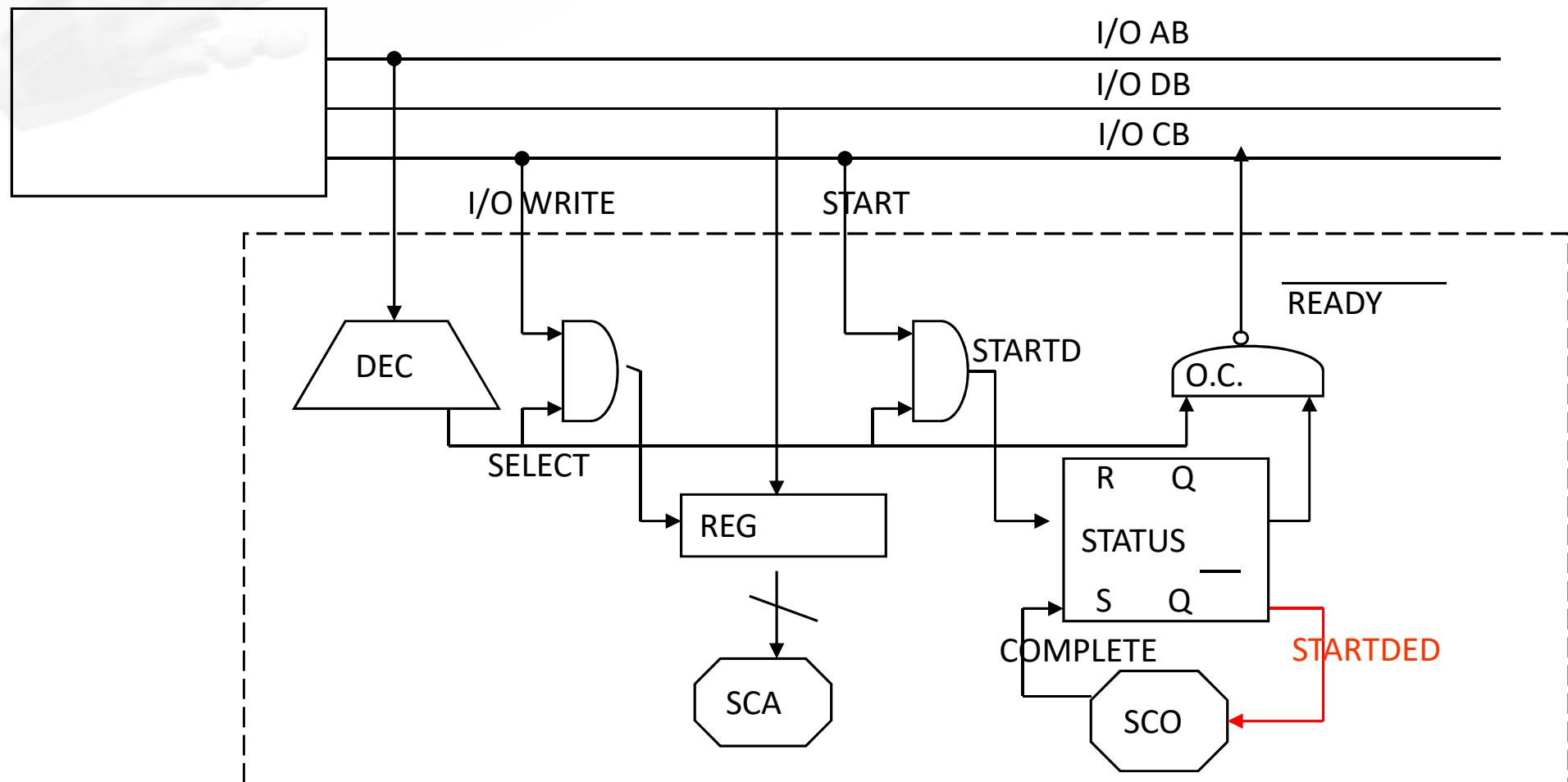
I/O: controller (OUTPUT)

- La sotto-rete che non dipende dal dispositivo e che è interessata nel colloquio con il processore (**controller**) di un dispositivo di output può essere così semplificata
Osservazione: Va evidenziato che il dispositivo fisico può essere situato anche in lontananza rispetto all'interfaccia e sfruttare canali fisici (USB) o protocolli di rete (WI-FI)



MACCHINA DI VON NEUMANN

I/O: controller (OUTPUT)





MACCHINA DI VON NEUMANN

I/O: controller (protocollo OUTPUT)

Protocollo OUTPUT

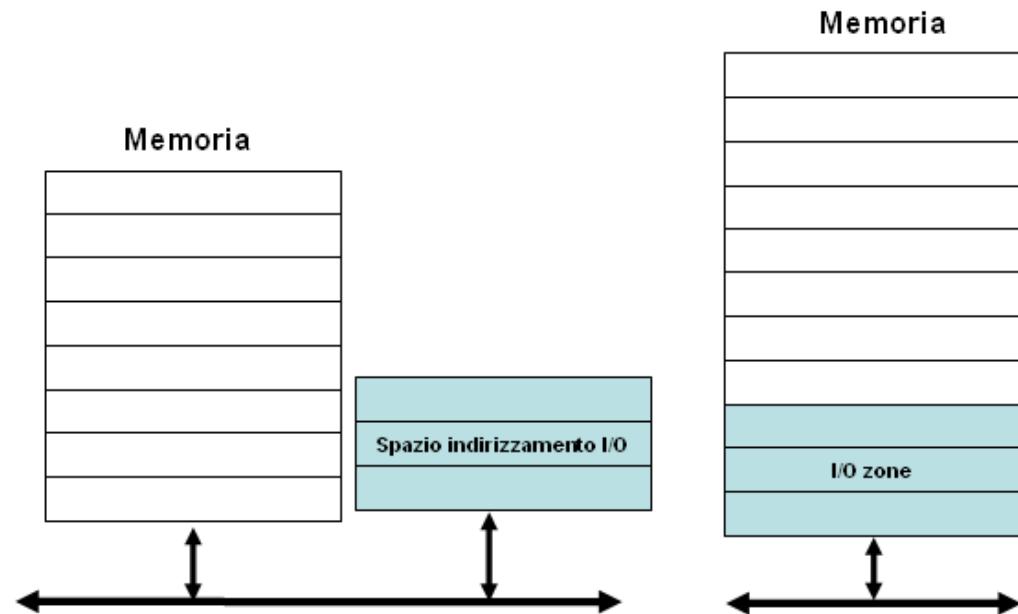
1. Il processore invia sull'I/O Address bus l'indirizzo del dispositivo e ne esamina lo stato tramite la linea di controllo READY.
2. Se il dispositivo non è pronto il processore deve attendere (o elaborare un'altra istruzione e ripetere la richiesta di interazione col dispositivo)
3. Se $\text{READY}=0$ il processore esegue una istruzione di OUTPUT e trasferisce il contenuto di un registro nel registro di interfaccia del dispositivo (mediante il segnale di controllo I/O WRITE)
4. Il processore avverte il dispositivo che gli ha trasferito un dato (seleziona il dispositivo tramite le linee indirizzi e invia il segnale START). START resetta il flip-flop STATUS e in tale stato rimane per tutta la durata delle operazioni di consumo del dato da parte del dispositivo. Quando il dato è stato letto da REG, il dispositivo genera il segnale COMPLETE, settando STATUS ($\text{READY}=0$).
5. Nel frattempo il processore, in attesa, esamina il valore memorizzato in STATUS campionando il segnale READY.
6. Se $\text{READY}=1$ il processore deve attendere e tornare al punto 5
Se $\text{READY}=0$ il processore può eseguire un'altra istruzione.



MACCHINA DI VON NEUMANN

I/O: canonico e programmato

- Per indirizzare le unità di I/O e consentire l'accesso al dato da trasferire o recuperare, l'Unità di controllo ricorre ad una delle due seguenti tecniche :
 - ❖ Riservare all'I/O uno spazio di indirizzamento indipendente: l'unità centrale utilizza specifiche istruzioni nelle quali fornisce anche l'indirizzo identificativo del dispositivo da utilizzare nell'operazione
(I/O -CANONICO)
 - ❖ Riservare una porzione dello spazio di indirizzamento in memoria centrale ai dispositivi di I/O, in modo che ogni volta che la CPU utilizza un indirizzo di questa porzione (con una tipica istruzione di trasferimento dati con la memoria e cioè senza ricorrere a specifiche istruzioni) in realtà inizializza un dispositivo di I/O
(I/O PROGRAMMATO)





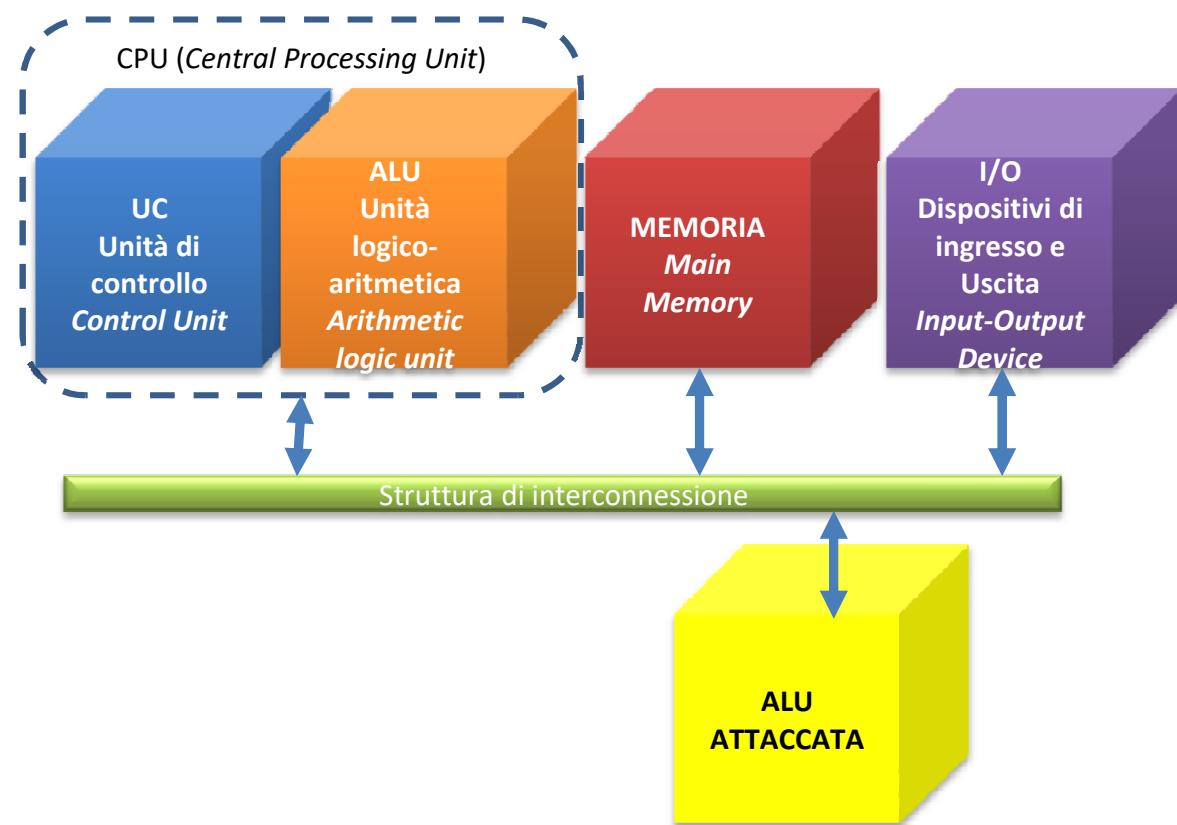
Architettura degli elaboratori

Interconnessione tra moduli

MACCHINA DI VON NEUMANN

Interconnessione BUS

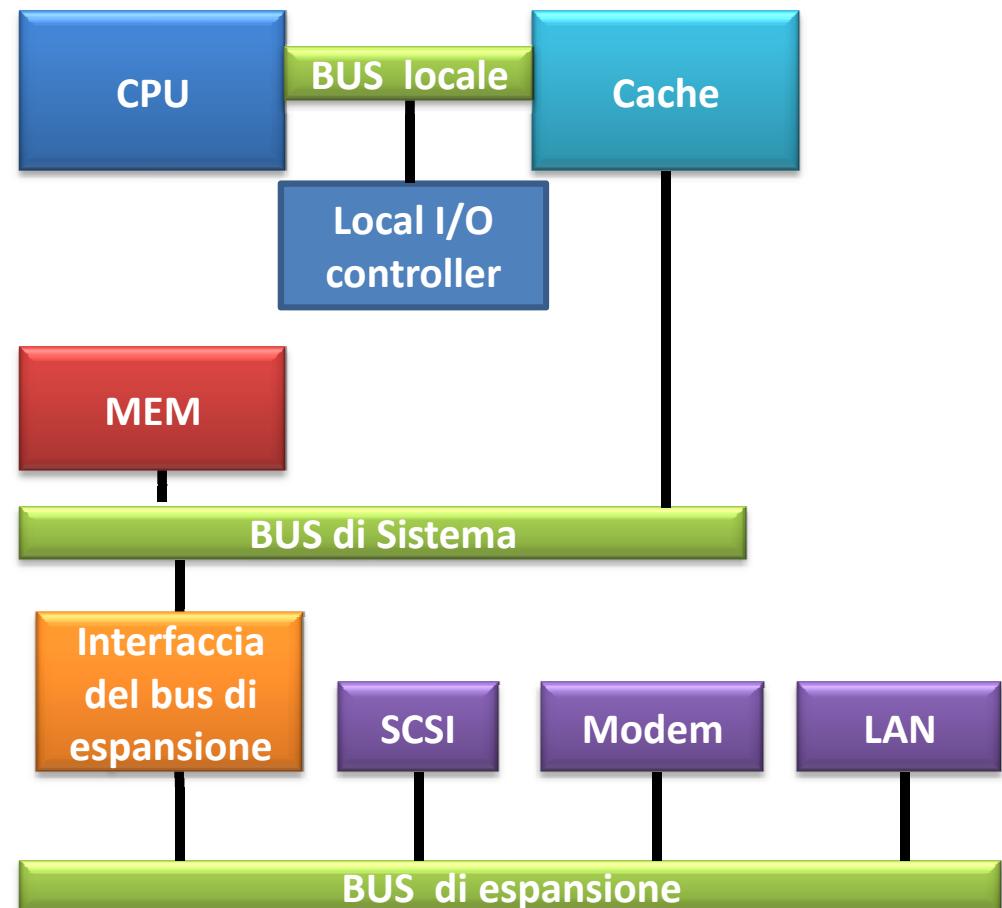
- Nelle moderne architetture le Unità sono interconnesse con il **bus** che consente il transito di operandi, dati e comandi
- Il numero di linee che costituisce il bus influenza la progettazione della macchina (costi, organizzazione topologica,...)
- I primi elaboratori avevano un unico bus chiamato anche **bus di sistema**. Esso era composto da 50 a 100 fili paralleli di rame che si inserivano nella scheda madre e i cui connettori erano distanziati a intervalli regolari per permettere l'inserimento di memorie e schede di I/O



MACCHINA DI VON NEUMANN

Interconnessione BUS

- Attualmente si usano più bus (**pluri bus**): uno specifico tra la CPU e la memoria e (almeno) un altro bus per le periferiche

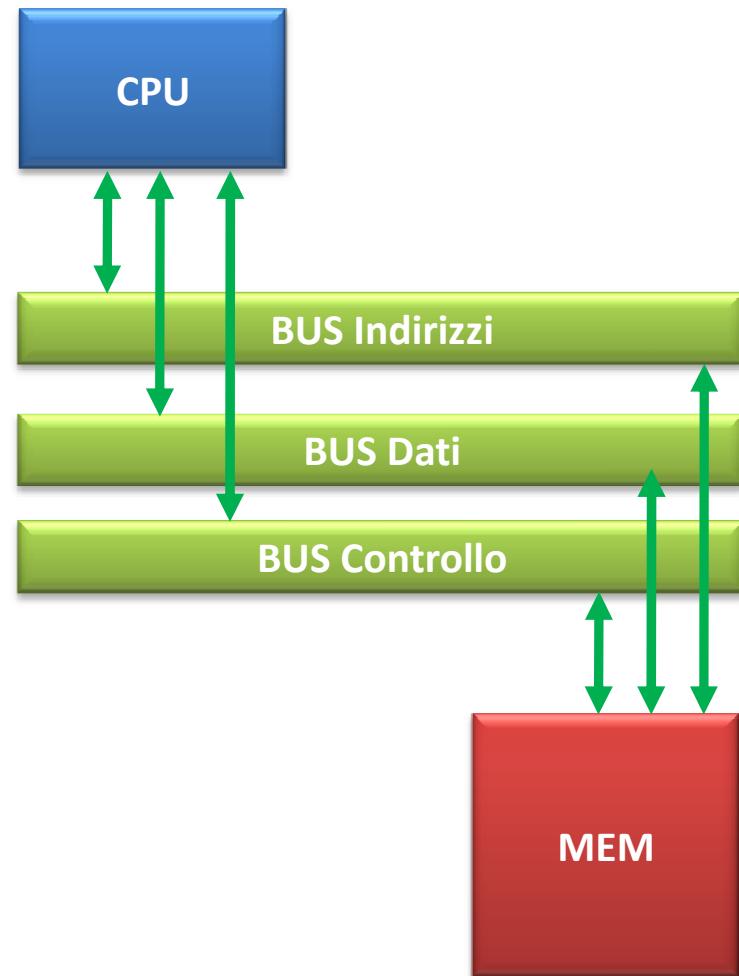




MACCHINA DI VON NEUMANN

Interconnessione BUS

- ❑ In alternativa si possono usare i **multi bus** o (bus multicanale) che consentono una netta suddivisione tra le informazioni, i segnali, da dover inviare (comandi, dati e indirizzi)
- ❑ Sono consentiti trasferimenti simultanei verso una Unità



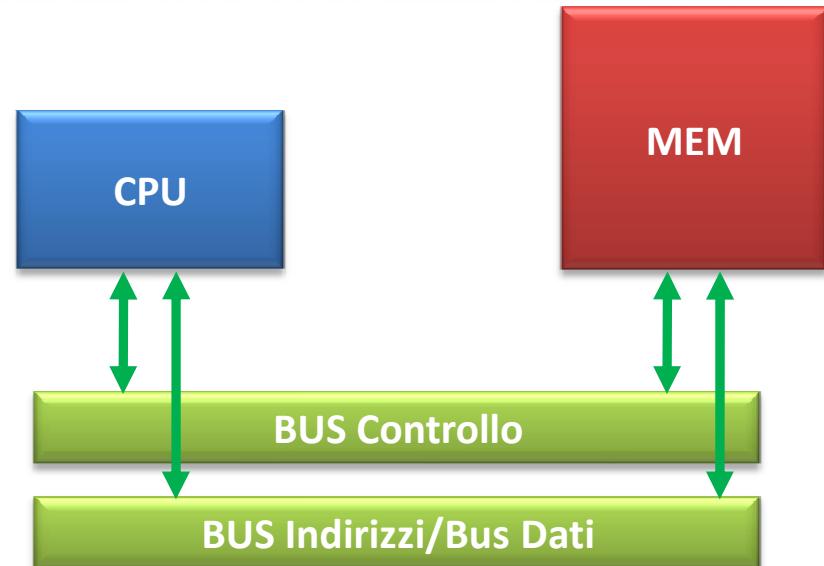


MACCHINA DI VON NEUMANN

Interconnessione BUS

- Per aggirare il problema di multi bus, che sono costosi ed ingombranti, si può usare un **bus multiplexato**

Invece di tenere separate le linee d'indirizzo e quelle dei dati, si utilizza un certo numero di linee per entrambi: all'inizio di un'operazione sul bus le linee sono utilizzate per gli indirizzi, mentre in seguito vengono impiegate per lo scambio dei dati



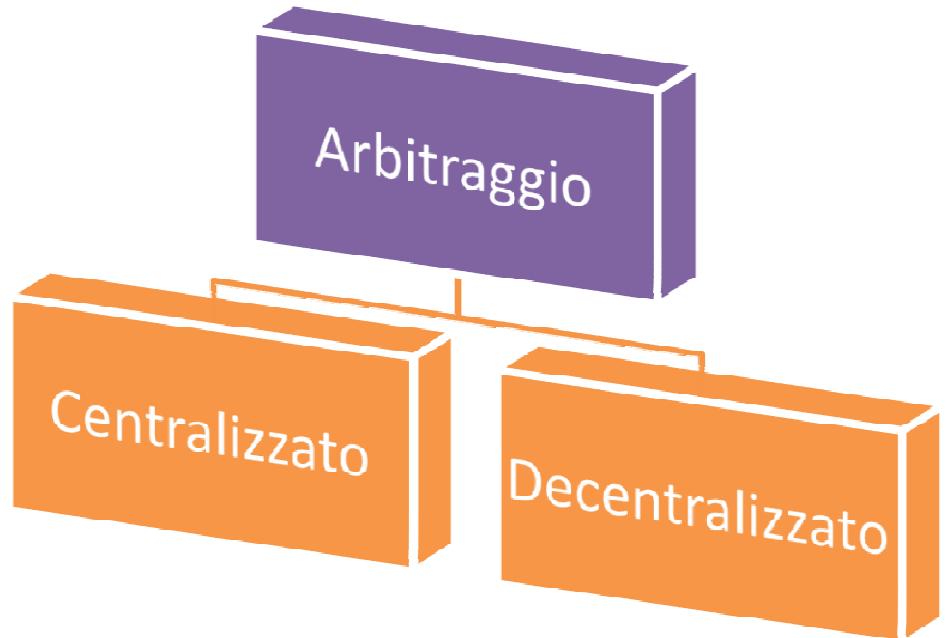
	CONTRO	PRO
Bus separati	Costoso	Rapido
Bus multiplexato	Economico	Lento



MACCHINA DI VON NEUMANN

Interconnessione BUS

- ❑ Nel caso due o più dispositivi richiedano contemporaneamente l'uso del bus si ricorre ad un arbitraggio del bus
- ❑ L'arbitraggio può essere **centralizzato** o **decentralizzato**

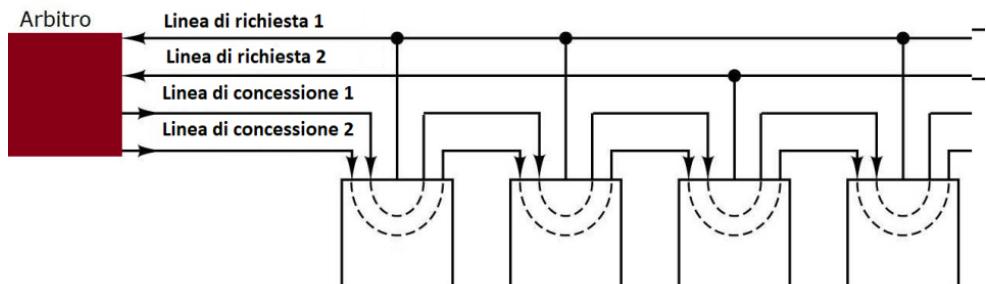
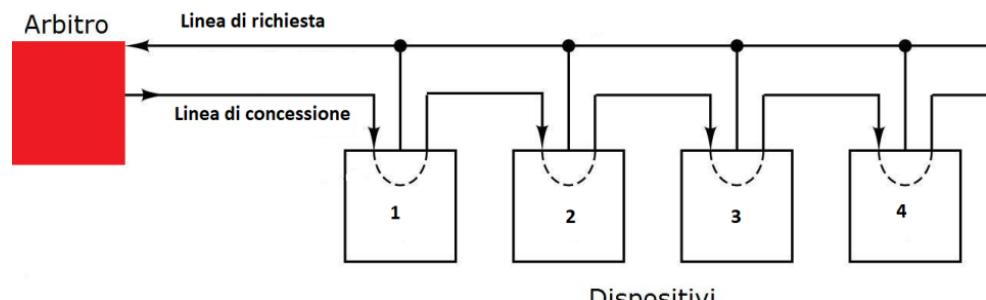




MACCHINA DI VON NEUMANN

Interconnessione BUS

- **Arbitraggio centralizzato:** un arbitro del bus (contenuto nella CPU o esterno ad esso) determina chi è il prossimo dispositivo
- L'arbitratore del bus, nel caso più semplice sfrutta la tecnica **daisy chaining**: ha un'unica linea di richiesta (non sa quanti e quali dispositivi hanno richiesto il bus, ma solo che c'è o non c'è almeno una richiesta)
- L'arbitratore abilita l'uso della linea. Il dispositivo di I/O più vicino verifica se ha richiesto l'uso del bus: se sì, si impossessa del bus e non consente di trasmettere oltre il segnale di concessione. Se invece non ha fatto richiesta, propaga la concessione sulla linea in direzione del prossimo dispositivo
- Per evitare che la scelta ricada sempre sulla distanza e non sul tipo di dispositivo si possono usare delle **linee di priorità**
- Nei sistemi in cui la memoria è collegata al bus principale, la CPU deve competere con tutti i dispositivi di I/O praticamente a ogni ciclo. Di solito la CPU ha la priorità più bassa rispetto gli I/O. I dispositivi di I/O sono obbligati ad acquisire il bus molto velocemente, pena la perdita dei dati in arrivo. I dischi che ruotano ad alte velocità, per esempio, non possono aspettare

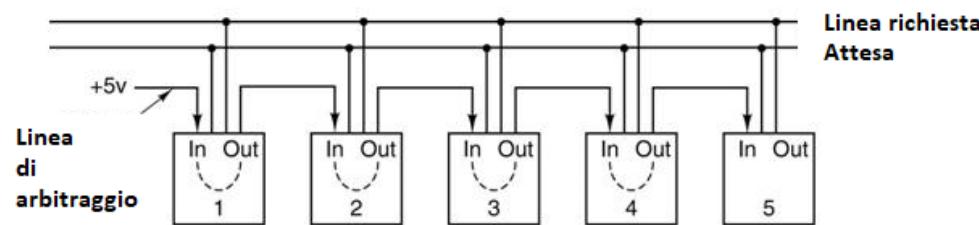


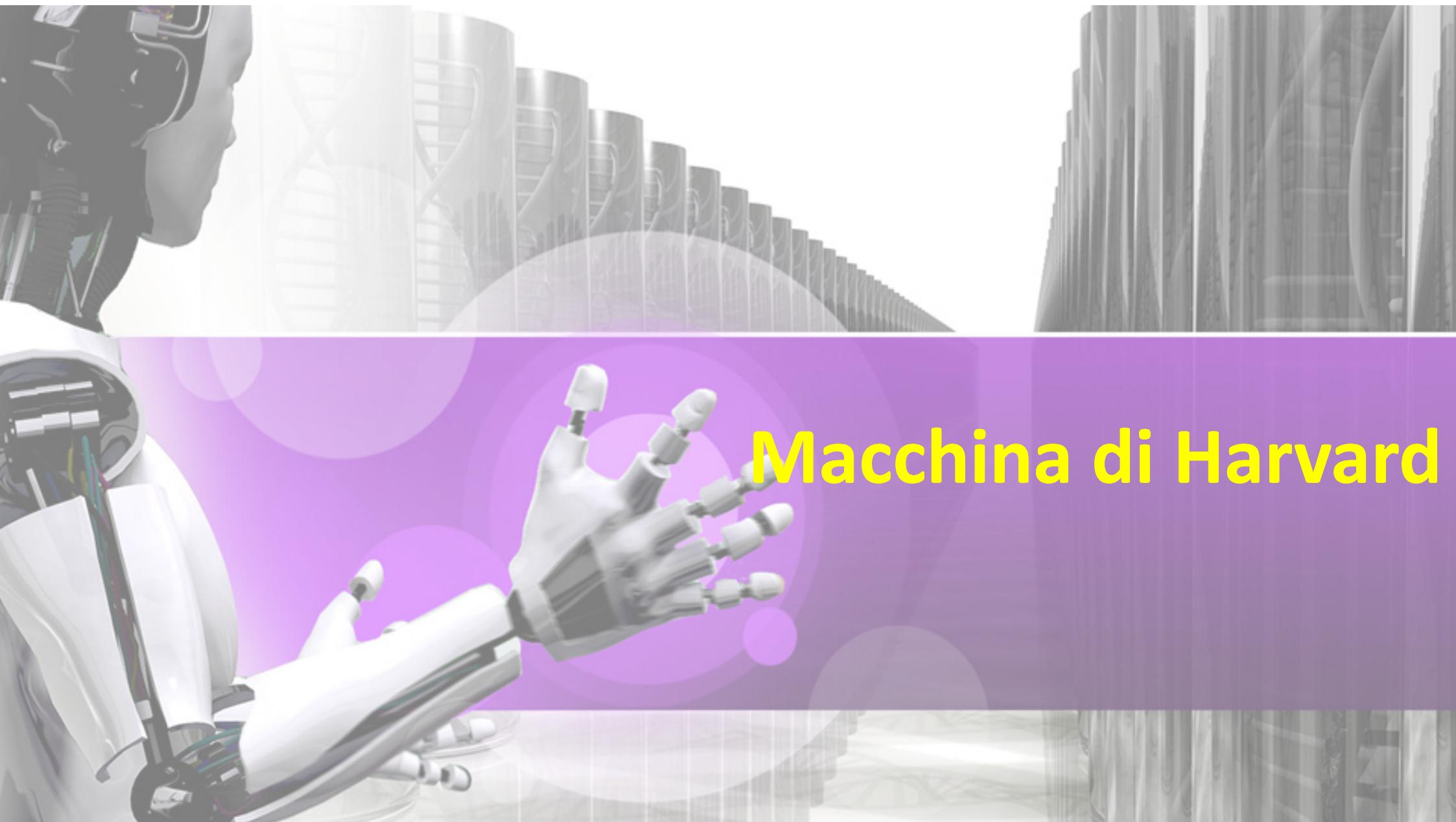


MACCHINA DI VON NEUMANN

Interconnessione BUS

- Nell'arbitraggio decentralizzato del bus si usano più linee di richiesta, ciascuna con la propria priorità
- Quando un dispositivo vuole utilizzare il bus invia un segnale lungo la linea di richiesta
- Tutti i dispositivi monitorano tutte le linee di richiesta in modo che alla fine di ciascun ciclo di analisi del bus ognuno di loro può sapere se era il richiedente con priorità più elevata e se quindi ha diritto a utilizzare il bus durante il ciclo successivo
- Rispetto al metodo centralizzato questo schema di arbitraggio richiede un maggior numero di linee ma evita il potenziale costo dell'arbitratore. Un altro limite è che il numero di dispositivi non può superare il numero delle linee di richiesta





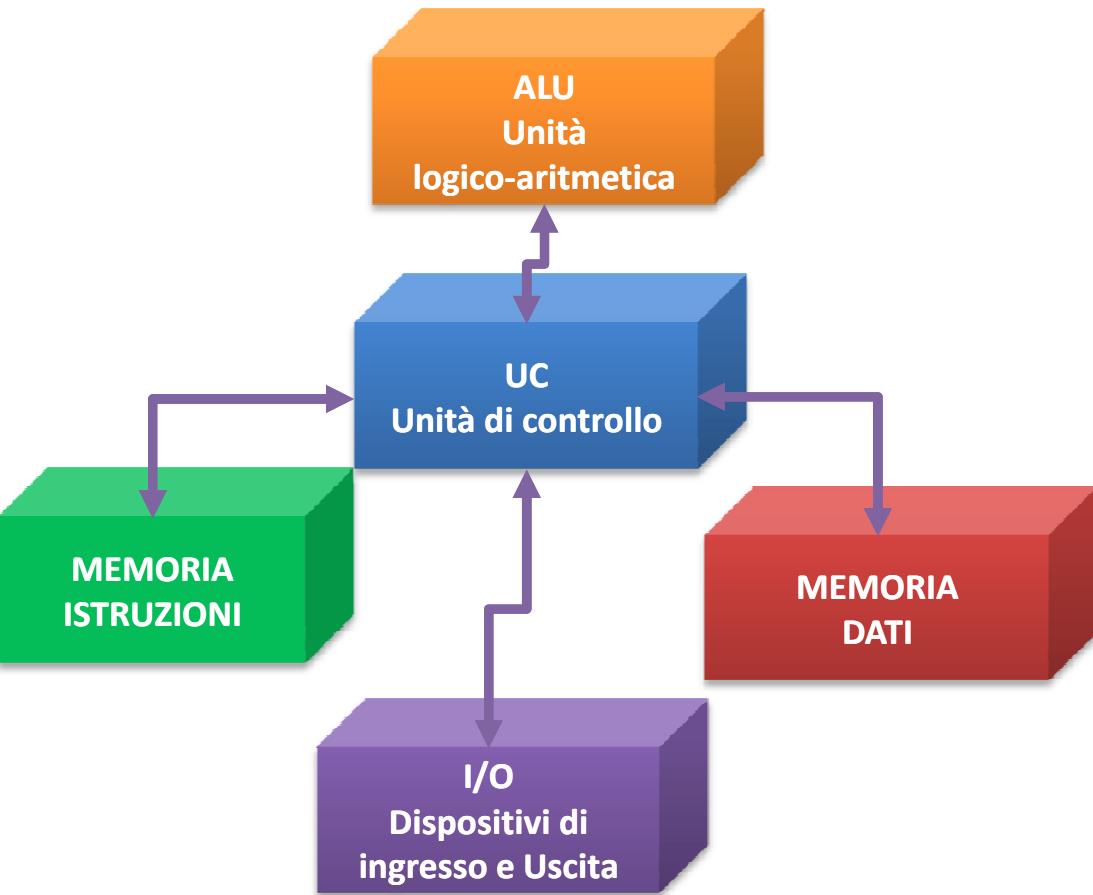
Macchina di Harvard



MACCHINA DI HARVARD

Generalità

- La prima data ufficiale in cui si realizzò un modello di calcolatore elettronico fu il 1944 con la “macchina di Harvard”, dal nome del college in cui si trovava il gruppo di lavoro che l’aveva ideata
- La macchina di Harvard era costituita da una unità di calcolo, una unità di controllo, una memoria contenente il programma (realizzato con valvole termoioniche preimpostate dai programmatori), una memoria dati (si usavano le schede perforate) ed un modulo per i dispositivi di input ed output, opportunamente collegati per consentire un flusso di comandi e di controlli che viaggiavano dalla unità di controllo alle altre e ne permettevano il funzionamento e colloquio reciproco



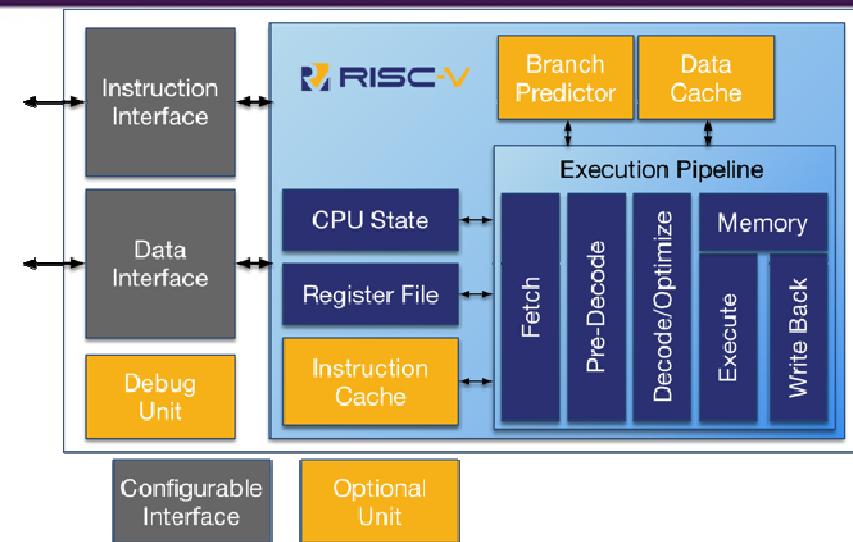
Curiosità. Le macchine signal processor impiegano una architettura Harvard per poter aumentare la velocità di esecuzione delle istruzioni attraverso l’accesso alla memoria in cui risiede il programma e a quella in cui sono presenti i dati



MACCHINA DI HARVARD

Progetto RISC I

- Lo sviluppo dei compilatori, che consentirono un'ottimizzazione del codice; la presenza di un'interfaccia comune ai moduli di I/O, che si occupò solo dei trasferimenti e demandava il resto delle operazioni alla periferiche stessa; e la presa di coscienza che l'uso di istruzioni complesse fosse saltuario e non fondamentale, portarono alla stesura del **progetto RISC1** presso l'Università di Berkeley
- Il gruppo di ricercatori accademici definirono un processore semplice, dotato di un insieme di istruzioni completo ma minimo e pochi modi di indirizzamento pur presentando prestazioni paragonabili a quelle dei più avanzati processori CISC del tempo. Il progetto dell'**architettura RISC** o elaboratore con set di istruzioni ridotto (*Reduced Instruction Set Computer*) fu realizzato grazie agli sforzi di diversi gruppi di ricerca



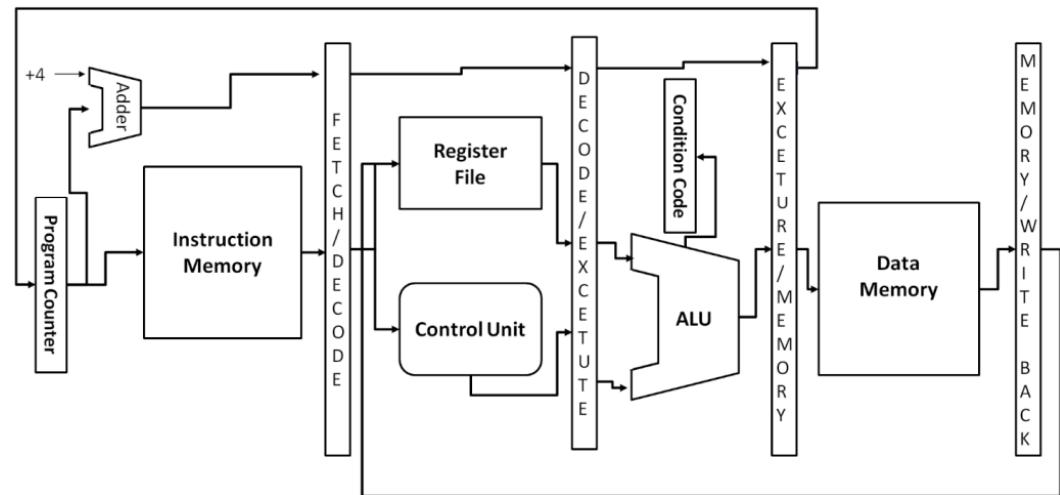
CISC	RISC
Molteplici istruzioni e modi di indirizzamento	Poche istruzioni e pochi modi di indirizzamento
Pochi registri	Molti registri
Microprogrammazione	Compilatore complesso
n-cicli macchina per eseguire una istruzione	Un ciclo macchina per eseguire una istruzione
Specializzazione delle componenti hardware	Specializzazione delle componenti software



MACCHINA DI HARVARD

Progetto RISC I

- Tra i pionieri in questo campo ci furono i ricercatori della Stanford University, tra cui David Patterson e John Hennessy. Il gruppo di scienziati, con una solida esperienza nella realizzazione dei compilatori, ideò l'architettura **Microprocessor without Interlocked Pipelined Stages** (MIPS), in cui, nella versione base, ogni istruzione poteva essere eseguita durante un solo segnale di clock. Il set d'istruzioni malgrado fosse minimo ed avesse pochi modi di indirizzamento era sufficiente a eseguire algoritmi complessi e accedere in tutte le parti della memoria.
- Il processore, in più, sfruttava la tecnica della **canalizzazione** (pipeline): le istruzioni non erano più eseguite sequenzialmente ma, rendendo indipendenti le fasi in cui si preleva, codifica ed esegue una singola istruzione, si procedeva alla loro sovrapposizione; migliorando le prestazioni complessive della macchina



Istr1

Istr2

Istr3

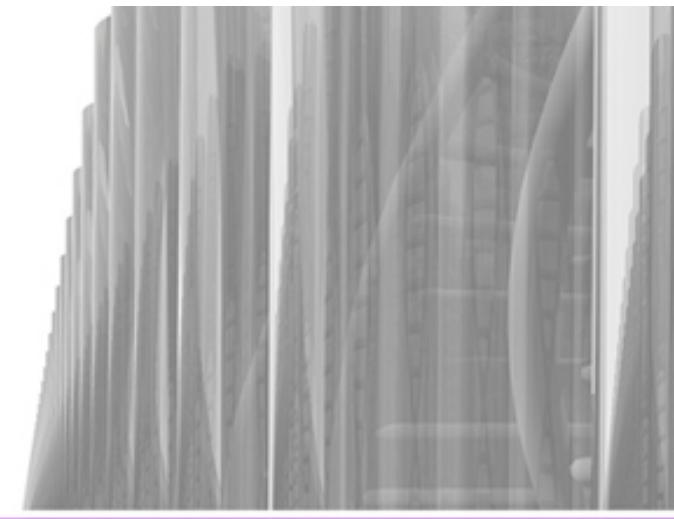
Istr4

Istr1

Istr2

Istr3

Istr1



Fine