

Corso di Laurea triennale in Ingegneria Civile in condivisione
con
Corso di Laurea triennale in
Ingegneria Gestionale

Corso di ***Elementi di Informatica***
Anno accademico 2017/18

Docente: Ing. Pino Villano
e-mail: pinogiuseppe.villano@unina.it

Il modello di esecutore

Riferimenti:

Alla scoperta dei Fondamenti dell'Informatica, capitolo 2
(paragrafi 2.1, 2.2, 2.3, 2.4, 2.5, 2.6)

Cos'è un computer

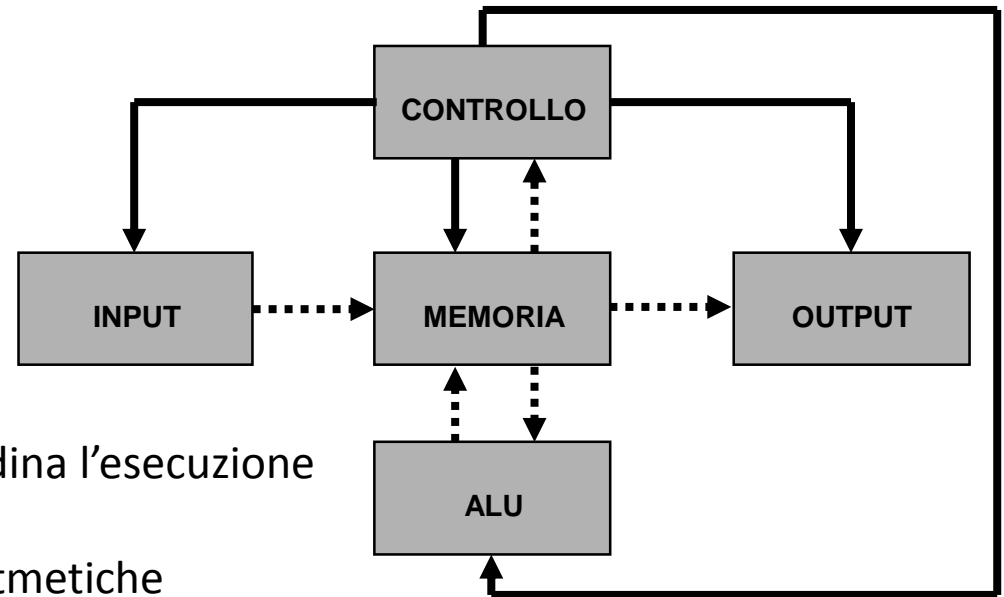
- Un computer è un apparecchio elettronico
 - strutturalmente non ha niente di diverso da un televisore, uno stereo, un telefono cellulare o una calcolatrice elettronica
- progettato per eseguire autonomamente e velocemente attività diverse
 - Come tutte le macchine, non ha nessuna capacità decisionale o discrezionale, ma si limita a compiere determinate azioni *secondo procedure prestabilite*.
- Il computer è una macchina che in maniera automatica esegue operazioni “*elementari*” ad altissima velocità.
 - L'altissima velocità di elaborazione (ad es. milioni di istruzioni per secondo) fa sì che operazioni complicate
 - espresse mediante un gran numero di operazioni semplici siano eseguite in tempi ragionevoli per l'ambiente esterno

Algoritmo, Processo, Processore e Programma

- Informalmente, un **algoritmo** è la descrizione di un lavoro da far svolgere ad un *esecutore*.
 - ... se si vuole usare un computer bisogna non solo progettare preliminarmente un algoritmo, ma anche provvedere a comunicarglielo in modo che gli risulti comprensibile.
- L'esecuzione di un algoritmo da parte di un esecutore si traduce in una successione di azioni che vengono effettuate nel tempo.
- Si definisce *processo* il lavoro svolto eseguendo l'algoritmo, e **processore** il suo esecutore.
 - Il processo non è altro che l'elenco delle azioni effettivamente svolte come si susseguono nel tempo.
 - Ogni algoritmo evoca da uno a più processi
 - a seconda delle condizioni in cui il lavoro viene svolto, si possono verificare comportamenti diversi da parte dell'esecutore
- La descrizione di un algoritmo in un linguaggio comprensibile ad un calcolatore è detto **Programma**
- Un programma è un insieme di **istruzioni**

Il modello di Von Neumann

- è uno schema di principio rappresentativo dei tradizionali computer che prende il nome da Von Neumann, il primo ricercatore che lo propose nel 1945.



- La **Central Processing Unit (CPU)** coordina l'esecuzione delle operazioni fondamentali;
- La **ALU** effettua le operazioni logico-aritmetiche
- La **memoria** contiene un programma che descrive le operazioni da eseguire e i dati su cui il programma stesso opera;
- i dispositivi di **input** e **output** sono le interfacce della CPU nei confronti del mondo esterno

Stored Procedure

- Il modello di Von Neumann si basa sul concetto di programma memorizzato:
 - la macchina immagazzina nella propria memoria i dati su cui lavorare e le istruzioni per il suo funzionamento.
- → flessibilità operativa
 - macchine nate per fare calcoli possono essere impiegate nella risoluzione di problemi di natura completamente diversa, come problemi di tipo amministrativo, gestionale e produttivo.
- Caratteristiche
 - schema di funzionamento semplice nelle sue linee generali
 - velocità e affidabilità nella esecuzione degli algoritmi;
 - ... milioni di istruzioni svolte dalla CPU in un secondo (*MIPS*: milioni di istruzioni per secondo)
 - un computer non commette mai errori di algoritmo poiché è un esecutore obbediente dell'algoritmo, la cui esecuzione gli è stata affidata
 - adeguata capacità di memoria
 - La capacità è la misura del numero di informazioni immagazzinabili nella memoria: oggi si misura in numero di byte
 - costo vantaggioso

Memorie

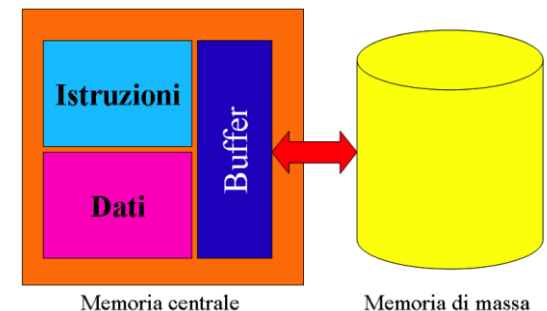
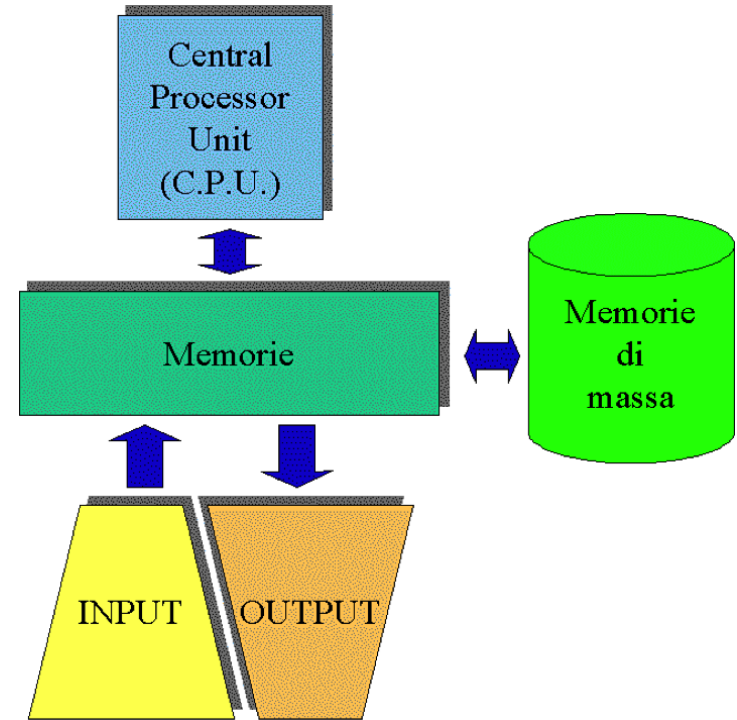
- In generale le memorie possono essere viste come un insieme di contenitori fisici, detti anche *registri*, di dimensioni finite e fissate a cui si può far riferimento mediante la posizione occupata nell'insieme detta *indirizzo* di memoria.
 - La dimensione di un registro si misura in numero di bit. Ciascun bit è memorizzato tramite un dispositivo capace di assumere due sole condizioni riconducibili allo 0 e 1 logico:
 - nelle memorie di tipo elettronico sono circuiti detti flip-flop che mostrano un valore di tensione o uguale a 5 Volt o a 0 Volt;
 - nelle memorie di tipo magnetico è una sorta di calamita polarizzata o positivamente o negativamente;
 - nelle memorie di tipo ottico è una superficie con o senza un buco in modo da riflettere diversamente il raggio laser che la colpisce.
 - I dispositivi di lettura devono essere in grado di associare allo stato del bit il valore 1 o il valore 0
- Memorie con registri di 8 bit sono dette *a byte* o *a caratteri*; con più di 8 (solitamente 16 o 32) vengono invece dette a voce (o a word).

Tassonomia delle memorie

- *Per tipo di accesso:*
 - Memoria ad accesso casuale
 - ... il tempo di accesso non dipende dalla posizione - *R.A.M.* (*Random Access Memory*);
 - Memoria ad accesso sequenziale
 - il tempo di accesso dipende dalla posizione, come avviene nei nastri magnetici.
- *Per tipi di operazioni consentite:*
 - Alcune memorie vengono realizzate in modo che sia possibile una sola scrittura di informazioni. Tali memorie vengono dette a *sola lettura* o *ROM* (da *Read Only Memory*).
 - L'uso di queste memorie è necessario quando si desidera che alcune istruzioni o dati non siano mai alterati o persi.
- *Per volatilità:*
 - Memorie volatili
 - .. perdono le informazioni in esse registrate quando il sistema viene spento perché il valore è sostenuto dall'alimentazione elettrica
 - Memorie permanenti
 - Non perdono le informazioni quando non vengono alimentate: memorie di tipo magnetico, ottico, e tutti i tipi di ROM (es. le memorie flash).

Memorie di Massa

- Le memorie di massa sono memorie ausiliarie caratterizzate da una **elevata capacità**.
 - ... Le informazioni contenute nella memoria di massa devono essere dapprima trasferite nella memoria centrale e successivamente elaborate
 - ... le informazioni prodotte dalla CPU, viceversa, devono essere depositate in memoria centrale per poi essere conservate nelle memorie di massa.
- Le memorie di massa hanno tempi di accesso maggiori dovuti alle tecnologie impiegate per realizzarle.
- Per ovviare alla differenza di velocità tra i due dispositivi si impiegano tecniche che prevedono che la memoria centrale contenga anche *aree di accumulo dei dati in transito verso tutti i dispositivi esterni (buffer)*.



Operazioni sulla memoria

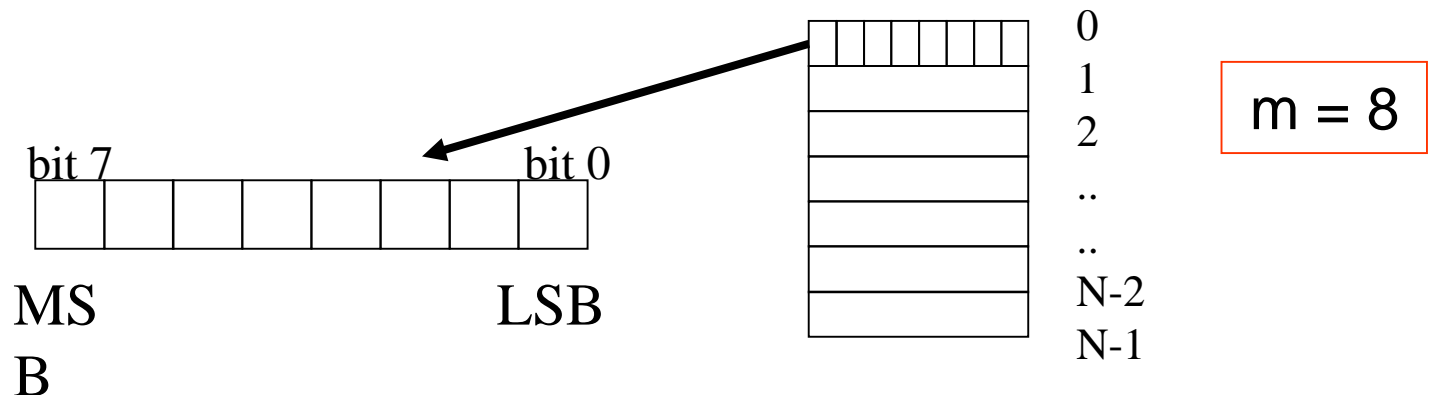
- **Load/lettura:**
 - si preleva l'informazione contenuta nel registro senza però distruggerla
- **Store/scrittura:**
 - si inserisce una informazione nel registro eliminando quella precedente.
- Per comprendere il funzionamento di un registro di memoria si può pensare ad una lavagna il cui uso può essere così esemplificato:
 - leggere informazioni a patto che vi siano state scritte;
 - la lettura non cancella quanto scritto;
 - la scrittura di nuove informazioni obbliga a cancellare quelle precedenti che pertanto vengono perse.
- La memoria è un sistema che assolve al compito di conservare il dato, depositandolo in un registro nel caso di operazione di scrittura, e di fornire il dato conservato in un registro in caso contrario di operazione

Funzionamento di una Memoria

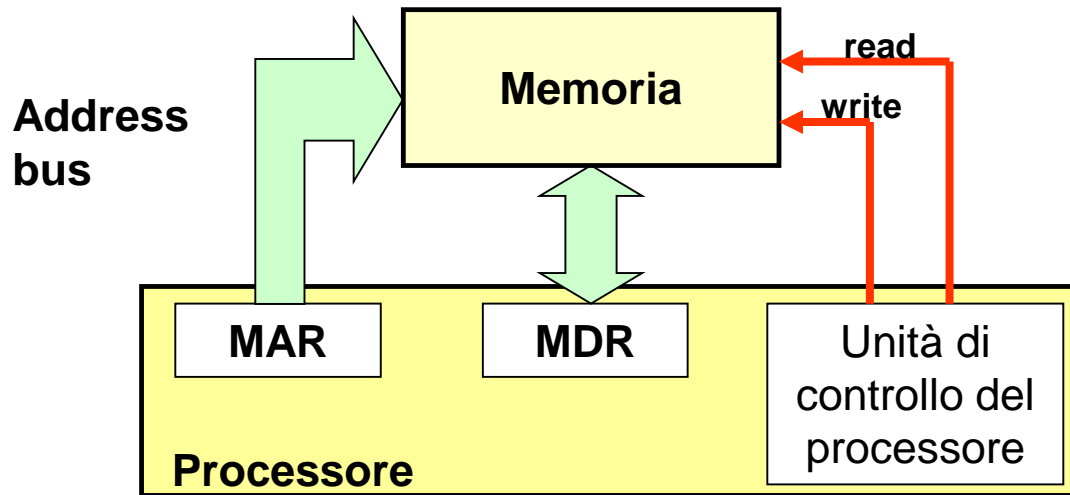
- La CPU indica preventivamente l'indirizzo del registro interessato dall'operazione;
- La memoria decodifica tale indirizzo abilitando solo il registro ad esso corrispondente affinché:
 - per uno *store* copi il dato del buffer nel registro;
 - per un *load* copi il dato del registro nel buffer.
 - dove il *buffer* può essere vista come un'area di transito dei dati dalla CPU alla memoria e viceversa.
- Le operazioni di load e store richiedono tempi di attuazione che dipendono dalle tecnologie usate per la costruzione delle memorie e dalle modalità di accesso.
 - Nel caso di load, il tempo di accesso misura il tempo che trascorre tra la selezione del registro di memoria e la disponibilità del suo contenuto nel registro di buffer.
 - Il tempo di accesso nel caso dello store misura invece il tempo necessario alla selezione del registro e il deposito del contenuto del registro di buffer in esso

La memoria centrale

- La memoria centrale di un computer è organizzata come un array di stringhe di bit di lunghezza m , dette *locazioni*
- Gli m bit di una locazione sono accessibili dal processore (in lettura/scrittura) mediante un'unica operazione (questo parametro indica il **parallelismo** della memoria)
- Ogni locazione è individuata da un *indirizzo*, cioè un intero compreso tra 0 e $N-1$, con $N = 2^c$
 - $[0, N-1]$ = SPAZIO DI INDIRIZZAMENTO (insieme delle locazioni indirizzabili)
- La memoria centrale è *ad accesso casuale* (RAM) cioè il tempo di accesso non dipende dalla posizione del dato

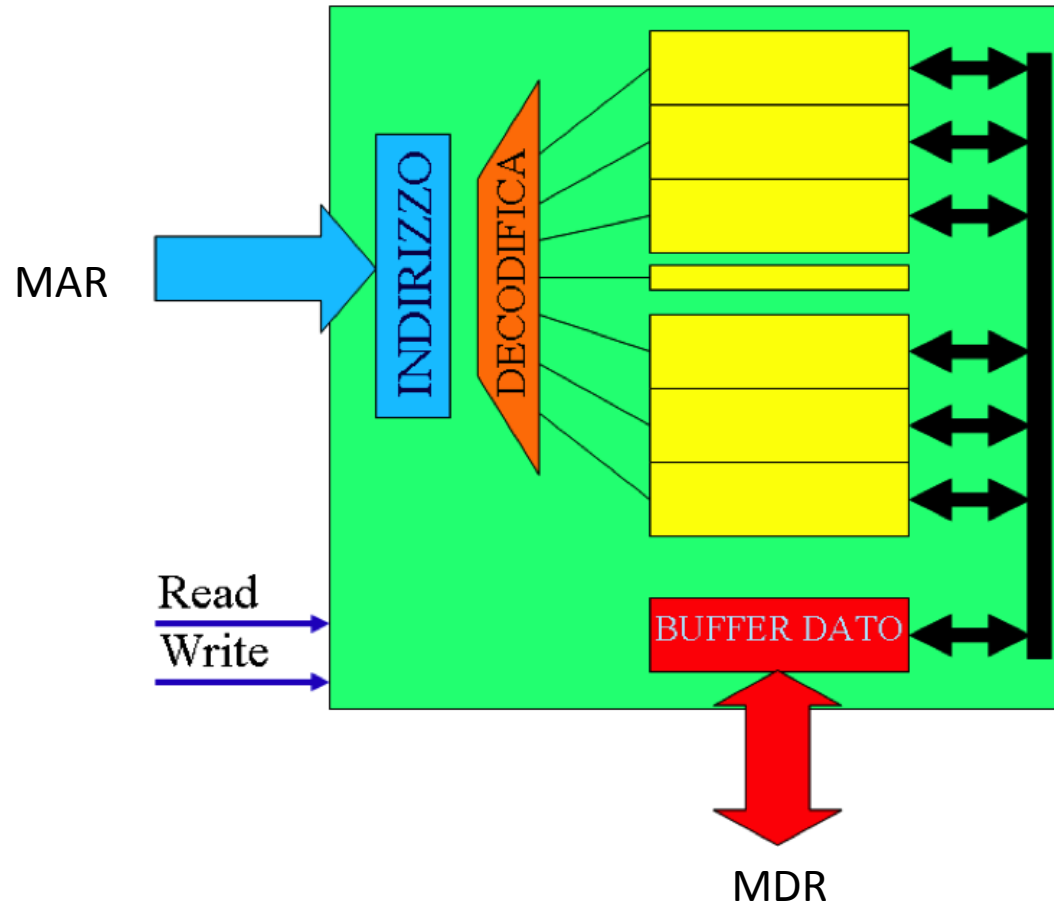


Interazione processore-memoria

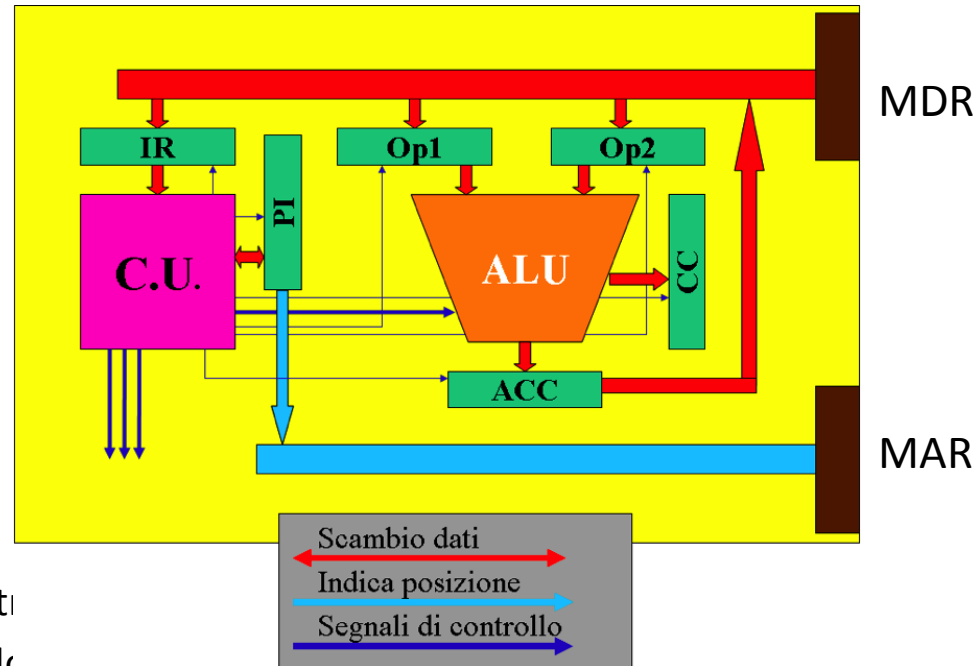


- Registro Memory Address (MAR)
 - fornisce alla memoria l'indirizzo del dato da leggere/scrivere
- Registro Memory Buffer o Memory Data (MBR/MDR)
 - contiene il dato da scrivere *verso*, o da leggere *dalla* memoria
- Segnali di controllo: *read / write*
 - ordinano alla memoria operazioni di lettura oppure di scrittura

Schema di accesso alla memoria



La C. P. U.



- ALU: unità logica aritmetica
- CU: Unità di Controllo
- Registri Interni:
 - IR: instruction register
 - PI o PC (program counter): prossima istruzione
 - MAR: memory address register
 - MDR: memory buffer register
 - ACC, Op1, Op2 registri per input/output ALU
 - CC : codici di condizione del processore
 - Registri generali per memorizzare dati temporanei

Control Unit

- È preposta
 - ... all'interpretazione delle singole istruzioni e all'attivazione di tutti i meccanismi necessari per eseguire le singole istruzioni.
- In particolare la CU ha il compito di:
 - prelevare ogni istruzione dalla memoria centrale e decodificarla,
 - prelevare i dati che servono all'istruzione,
 - eseguire l'istruzione.
 - Per esempio: se l'istruzione prelevata è di tipo aritmetico e richiede due operandi, la CU predispone dapprima il prelievo di tali operandi (eventualmente dalla memoria), attiva poi l'ALU perchè esegua l'operazione desiderata, ed infine deposita il risultato di nuovo in memoria.
- Al termine dell'esecuzione di una istruzione la CU procede al prelievo dalla memoria della successiva istruzione secondo un ordine rigidamente sequenziale
 - l'esecuzione di una istruzione può avere inizio solo se la precedente è stata portata a termine.

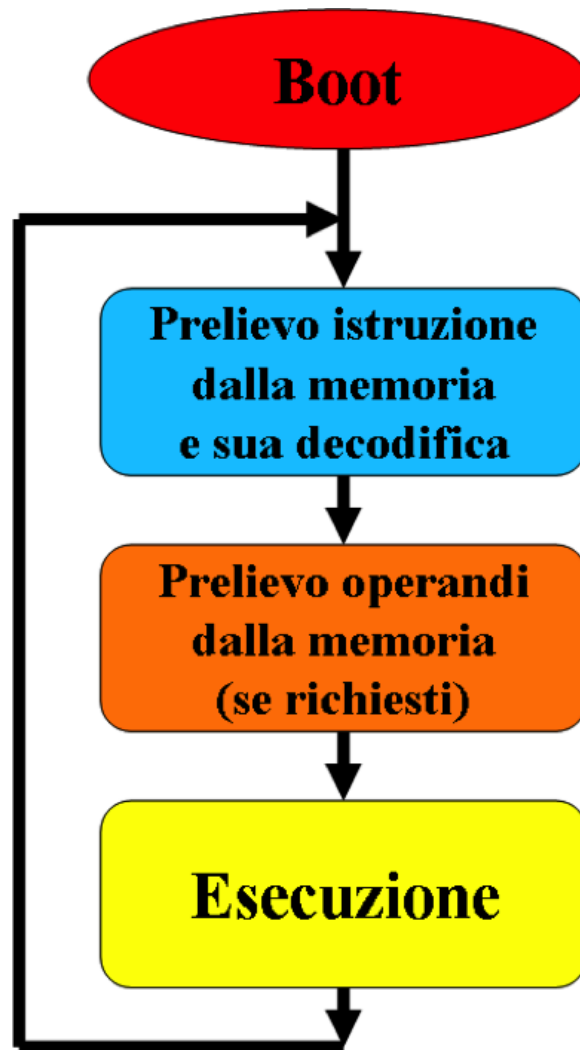
Aritmetic Logic Unit

- Esegue operazioni
 - aritmetiche, di confronto o bitwisesui dati della memoria centrale o dei registri interni.
- L'esito dei suoi calcoli viene segnalato da appositi bit in un registro chiamato *Condition Code*.
- A seconda dei processori l'ALU può essere molto complessa.
- Nei sistemi attuali l'ALU viene affiancata da processori dedicati alle operazioni sui numeri in virgola mobile detti *processori matematici*.
- Durante le sue elaborazioni la CU può depositare informazioni nei suoi registri interni in quanto sono più facilmente individuabili e hanno tempi di accesso inferiori a quelli dei registri della memoria centrale.

Registri Interni Speciali

- Il numero e tipo di tali registri varia a seconda dell'architettura della CPU.
- In molte CPU
 - *Instruction Register (IR)*
 - contiene l'istruzione prelevata dalla memoria e che l'unità di controllo sta eseguendo
 - *Prossima Istruzione (PI) o Program Counter (PC)*
 - ricorda alla CU la posizione in memoria della successiva istruzione da eseguire.
 - Nei casi in cui ogni registro di memoria contenga un'intera istruzione, e l'insieme delle istruzioni del programma sia disposto ad indirizzi consecutivi, la CU incrementa di uno il valore contenuto in PI dopo ogni prelievo di una istruzione dalla memoria
 - *Accumulatore (ACC)*
 - serve come deposito di dati da parte dell'ALU nel senso che contiene prima di un'operazione uno degli operandi, e al termine della stessa operazione il risultato calcolato.
 - In questo caso i registri *Op1* e *Op2* diventano interni all'ALU
 - *Condition Code (CC)*
 - indica le condizioni che si verificano durante l'elaborazione, quali risultato nullo, negativo e overflow

Il Ciclo del Processore



- Le tre fasi del ciclo vengono anche dette
 - *fetch*
 - *operand assembly*
 - *execute.*

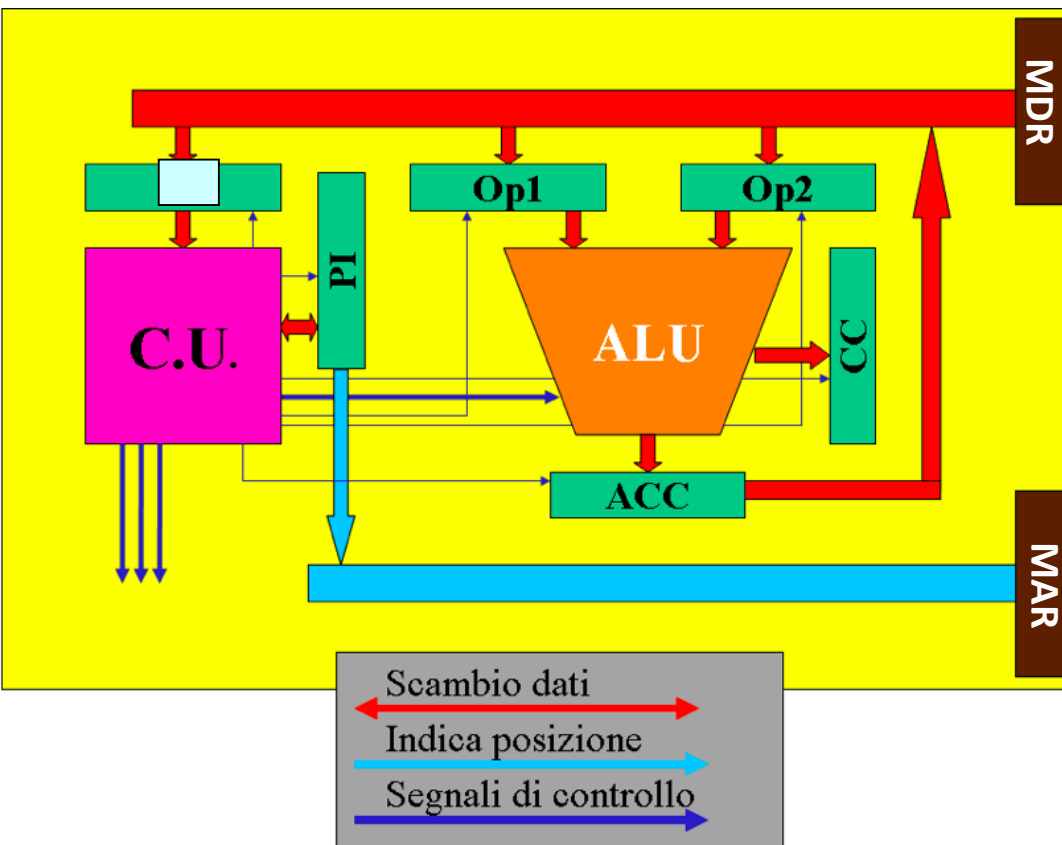
Considerazioni

- Fase di Boot
 - Perché il ciclo del processore possa avere inizio si deve predisporre in modo che il registro PI contenga l'indirizzo del registro di memoria contenente la prima istruzione da eseguire
- Una volta avviato, il ciclo del processore non termina mai e quindi ad ogni istruzione deve sempre seguirne un'altra da eseguire successivamente.
 - quando termina un'applicazione un elaboratore torna ad eseguire i programmi del sistema operativo.
 - Perché tutto ciò proceda nel rispetto del modello di Von Neumann, deve avvenire che in memoria siano sempre presenti i programmi e i dati del sistema

Ciclo del Processore: dettaglio

- La **fase fetch** inizia con il prelievo dell'istruzione dalla memoria.
 - Per farlo la UC comunica alla memoria il valore del puntatore ad istruzione presente nel registro PI.
 - La risposta della memoria viene depositata nel registro IR così da consentire alla UC di:
 - interpretare il codice operativo dell'istruzione da eseguire;
 - conoscere i puntatori ai dati di input ed output;
 - ricevere il puntatore all'istruzione da eseguire successivamente.
- La **fase fetch si conclude** con l'aggiornamento del registro PI con il valore del puntatore all'istruzione successiva presente in IR.
- La **fase operand assembly** serve alla UC per predisporre gli operandi che servono al codice operativo.
 - Per farlo la UC usa i puntatori ai dati contenuti nel registro IR.
- La **fase execute** consiste nel mettere in essere le azioni richieste con il codice operativo presente nel registro di IR.
- Nel caso vengano prodotti risultati, ne verrà effettuata la memorizzazione negli indirizzi specificati dai puntatori ai dati di output presenti nel registro IR.

Decode



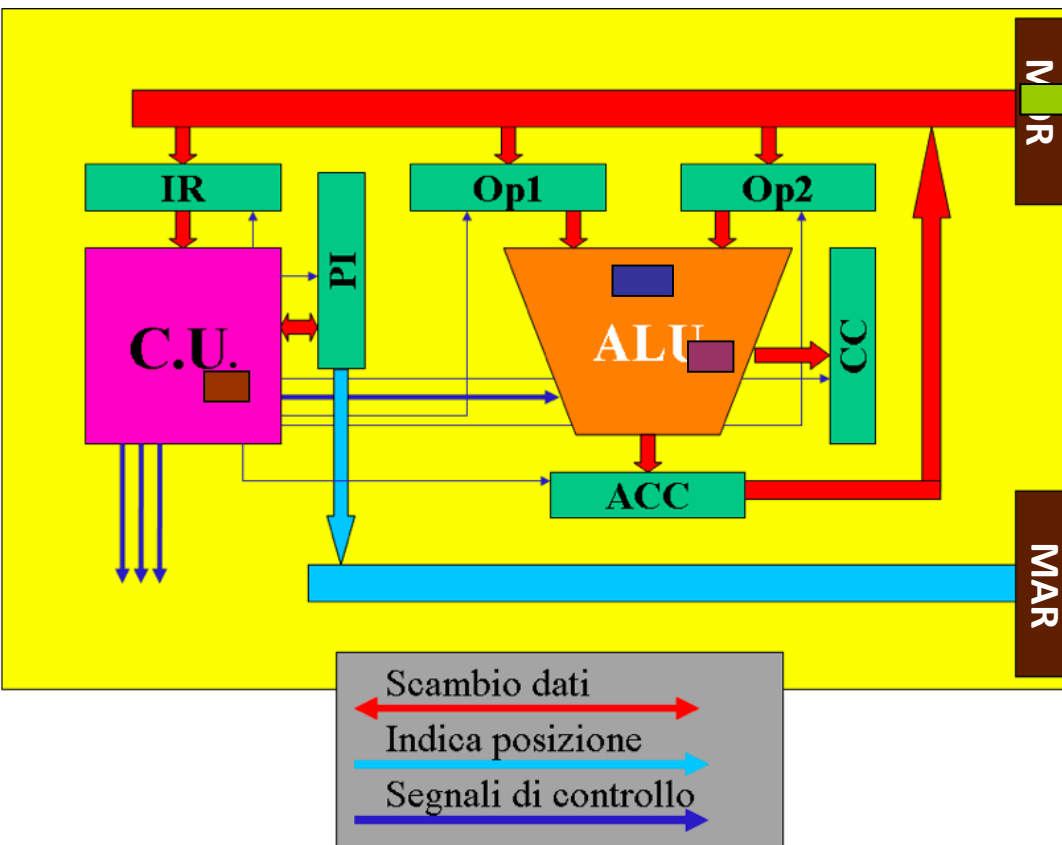
Fase di decode

1. L'istruzione viene interpretata, prelevando eventualmente dalla memoria altri dati necessari se l'istruzione completa non è contenuta nel dato prelevato in precedenza. In ogni caso, alla fine, PI punterà all'istruzione da eseguire successivamente..

Execute

Fase di execute

1. Se, per esempio, l'istruzione da eseguire è una somma, i registri verranno caricati con i numeri da sommare e la CU invia alla ALU il comando per l'esecuzione della somma. La ALU esegue l'operazione: il risultato prodotto sarà conservato in uno dei registri e verrà aggiornata il CC.



Istruzioni

- Le istruzioni sono operazioni semplici:
 - trasferimento dati da un registro ad un altro
 - da memoria a memoria,
 - da memoria a registri della CPU o viceversa,
 - da memoria a output, da input a memoria;
 - operazioni aritmetiche o logiche eseguite dall'ALU;
 - controllo di condizioni riportate dal registro CC o deducibili dal confronto di due registri;
 - Salto condizionato e incondizionato: modificano il valore del program counter;
 -

Istruzione in Linguaggio Macchina

- Concettualmente è una quadrupla:
 - $i = (C_{op}, P_{di}, P_{do}, P_{is})$
 - in cui:
 - C_{op} è il codice operativo, ossia il codice che indica alla UC della CPU l'operazione da compiere;
 - l'insieme dei C_{op} prende il nome di *repertorio di istruzioni* e dipende dalla specifica CPU;
 - P_{di} sono i puntatori ai dati che servono per svolgere l'operazione C_{op} detti anche di input;
 - ... esistono istruzioni che non hanno operandi di input;
 - P_{do} sono i puntatori ai dati prodotti dall'operazione C_{op} detti anche di output;
 - ... esistono istruzioni che non hanno operandi di output;
 - P_{is} è il puntatore all'istruzione da svolgere al termine dell'esecuzione di quella corrente (questa info non è codificata nell'istruzione, ma è contenuta nel registro Prossima Istruzione)

Gli operandi dell'istruzione: modi di Indirizzamento

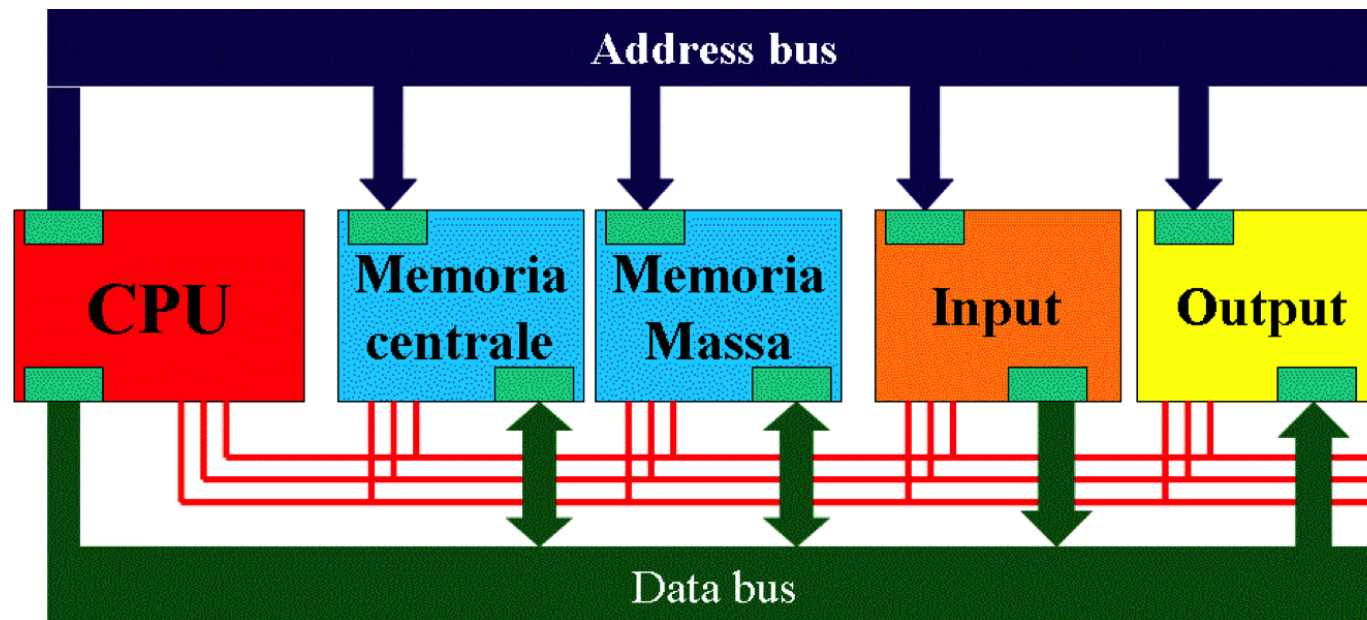
- Nella costruzione dell'indirizzo di un operando di una istruzione le tecniche di indirizzamento più diffuse sono:
 - indirizzamento **immediato** che indica che il valore è contenuto già nell'istruzione come costante (dato presente nell'istruzione stessa);
 - indirizzamento **diretto** con il quale viene riportato nell'istruzione qual è il registro interno del processore che contiene il valore o nel quale depositare il valore (dato presente in un registro interno del processore);
 - indirizzamento **indiretto** che riporta nell'istruzione qual è il registro interno del processore al cui interno è specificato l'indirizzo del registro di memoria dal quale prelevare un valore o nel quale depositare un valore (dato presente in memoria);
 - indirizzamento **assoluto** con il quale l'indirizzo del registro di memoria che contiene il valore o nel quale depositare il valore è specificato direttamente nell'istruzione come costante (dato presente in memoria).

Modi di indirizzamento: considerazioni

- Le diverse tecniche di indirizzamento vengono indicate nell'istruzione o diversificando il codice operativo o aggiungendo dei bit appositi il cui valore indica alla UC come costruire l'indirizzo.
- La specifica del modo di indirizzamento è fondamentale perché si possano eseguire le istruzioni

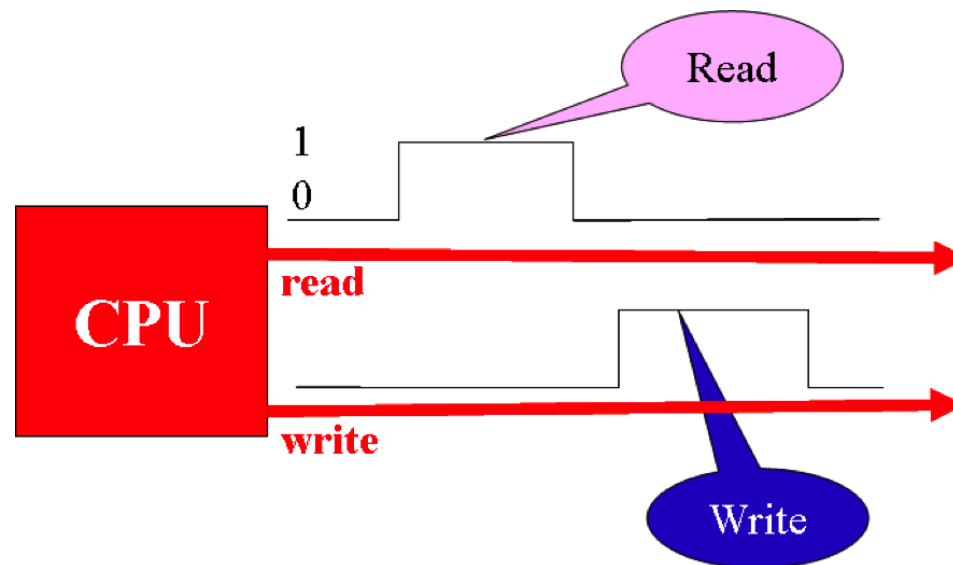
Il Bus

- E' un canale di comunicazione *condiviso* da più utilizzatori.
- E' fisicamente costituito da uno o più fili su cui possono transitare uno o più bit contemporaneamente.
- A seconda delle informazioni trasportate si hanno:
 - *bus dati (data bus)*
 - *bus indirizzi (address bus)*
 - *bus comandi o di controllo (command o control bus)*



Letture e Scrittura su Bus

- Il control bus serve alla C.U. per indicare ai dispositivi cosa essi devono fare.
 - Tipici segnali del control bus sono quelli di *read* e *write* mediante i quali la CU indica ai dispositivi se devono leggere un dato dal data bus (*read*) o scriverlo su di esso (*write*).
 - Il data bus permette ai dati di fluire da CPU a registro di memoria selezionato per operazioni di *store* e viceversa per quelle di *load*



Address Bus

- L'address bus serve alla CU per comunicare l'indirizzo del dispositivo interessato da una operazione di lettura o scrittura.
 - In questa ottica anche i dispositivi di input /output sono identificati da un indirizzo alla stessa stregua dei registri di memoria.
- Tutti i componenti del sistema (memoria, input, output, memoria di massa, etc.) devono essere dotati della capacità di riconoscere sull'address bus il proprio indirizzo.
 - In altri termini attraverso l'address bus la CU effettua la selezione del dispositivo a cui sono rivolti i comandi e i dati.

Scambio di Informazioni

- La CPU è l'unico elemento che fornisce l'indirizzo all'address bus;
- Memorie e dispositivi di input ed output devono *ascoltare l'address bus* per attivarsi quando su di esso compare il proprio indirizzo identificativo;
 - nel caso della memoria l'attivazione avviene quando viene riconosciuto l'indirizzo corrispondente ad uno dei registri di cui essa è composta;
 - il dispositivo attivo deve interpretare i segnali del control bus per eseguire i comandi della CU;
 - le memorie prelevano dati dal data bus o immettono dati in esso in funzione del comando impartito dalla CU;
- i dispositivi di input possono solo immettere dati sul data bus;
- viceversa i dispositivi di output possono solo prelevare dati dal data bus.

Bus Seriale e Parallelo

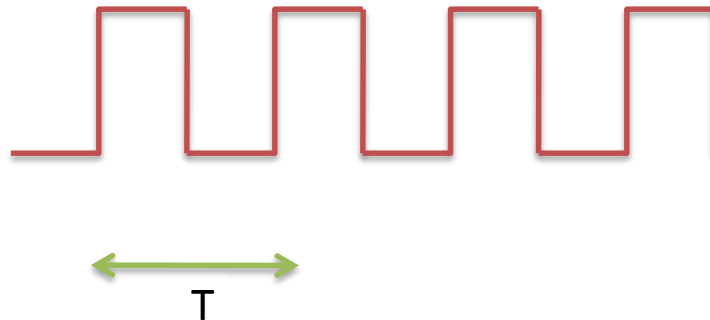
- Un bus costituito da un solo filo è chiamato bus *seriale* e su di esso i bit transitano uno dietro l'altro.
- Un bus costituito da n fili è chiamato bus *parallelo* perché su di esso transitano n bit alla volta.
 - Tipici sono i bus a 8 e 32 fili sui quali si possono trasferire rispettivamente 8 e 32 bit (4 Byte) alla volta.
- L'address e il data bus sono paralleli e le loro dimensioni caratterizzano i sistemi di calcolo.
 - Il numero di bit dell'address bus indica la *capacità di indirizzamento* della CPU: ossia la sua capacità di gestire la dimensione della memoria centrale e il numero di dispositivi di input ed output.
 - Infatti un address bus con n bit consente di selezionare un registro tra 2^n .
 - La dimensione del data bus condiziona invece la velocità di scambio delle informazioni tra i diversi dispositivi in quanto con m fili solo m bit possono viaggiare contemporaneamente.

Gestione Input Output

- tecnica *memory-mapped*
 - ... l'UC usa le stesse istruzioni utilizzate per leggere e scrivere in memoria anche per accedere ai dispositivi di I/O.
 - I dispositivi di I/O hanno quindi dei propri indirizzi che devono essere riservati e non sovrapposti a quelli usati per la memoria. I dispositivi di I/O controllano il bus indirizzi e rispondono solo quando riconoscono un indirizzo a loro assegnato
 - Il vantaggio dell'uso del memory-mapped è che, non richiedendo da una parte hardware aggiuntivo per la gestione della periferia e dall'altra un insieme di istruzioni specifiche, consente la realizzazione di CPU con una complessità inferiore, più economiche, veloci e facili da costruire.
- Tecnica *I/O-mapped*.
 - ... vengono invece usate istruzioni specifiche per l'esecuzione dell'input/output.
 - I dispositivi di I/O hanno uno spazio indirizzi separato da quello della memoria, e un segnale del control bus serve alla UC per specificare se si tratta di un accesso alla memoria o ad un dispositivo periferico.

Il clock

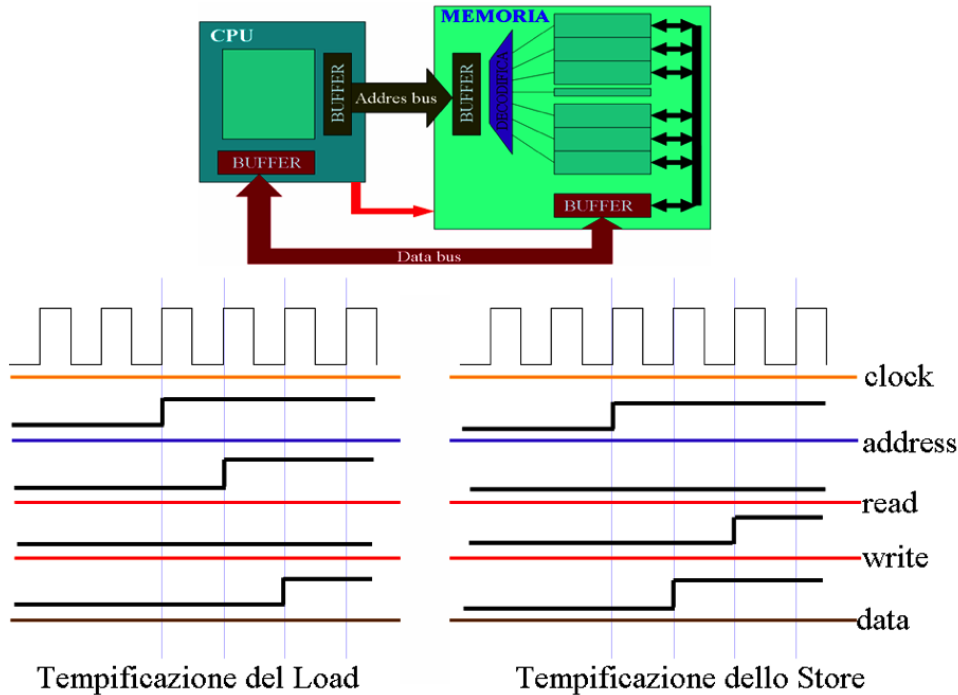
- Le attività di tutti i dispositivi vengono sincronizzate tra loro mediante un orologio interno che scandisce i ritmi di lavoro.
- Il *clock* è un segnale periodico di periodo fisso
 - un'onda quadra caratterizzata da un periodo T (detto ciclo) e da una frequenza f ($f=1/T$) misurata in Hertz (Hz).
 - Ad esempio un clock composto da 10 cicli al secondo ha la frequenza $f = 10$ Hz e il periodo $T = 100$ ms.
 - La frequenza dei clock presenti nei moderni sistemi spazia dai MHz (1 MHz corrisponde a un milione di battiti al secondo) ai GHz (1 GHz corrisponde a un miliardo di battiti al secondo).



Clock e velocità di elaborazione

- La velocità di elaborazione di una CPU dipende dalla frequenza del suo clock
 - più accelerato è il battito del clock maggiore è la velocità di esecuzione.
- Alla frequenza del clock è legato il numero di operazioni elementari che vengono eseguite nell'unità di tempo dalla CU
 - Ad esempio, se si assume che ad ogni ciclo di clock corrisponde esattamente l'esecuzione di una sola operazione, allora la frequenza del clock indica il numero di operazioni che vengono eseguite nell'unità di tempo dalla UC.
 - ... con un clock a 3 GHz si ha che il processore è in grado di eseguire 3 miliardi di operazioni al secondo.
 - ...In realtà tale ipotesi non è sempre vera in quanto l'esecuzione di una operazione può richiedere più cicli di clock sia per la complessità delle operazioni che per la lentezza dei dispositivi collegati alla CPU.

Esempio di Tempificazione



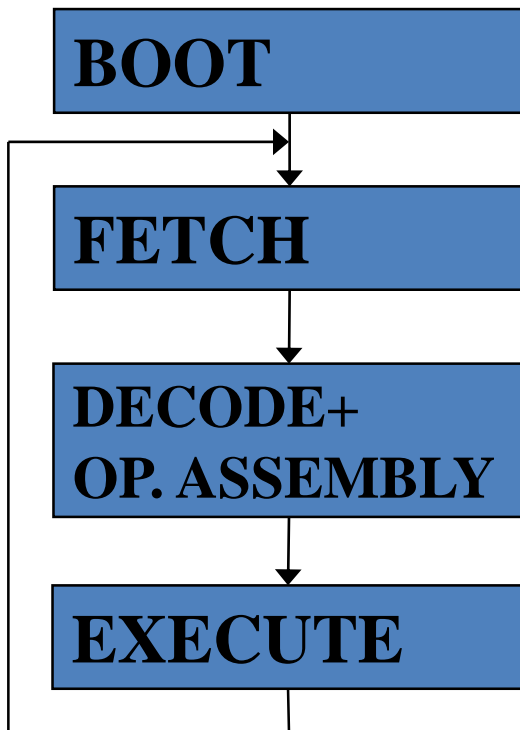
• LOAD

- con il primo battito di clock si pone l'indirizzo del registro di memoria di cui vuole leggere il contenuto sull'address bus;
- con il secondo battito si segnala alla memoria che si tratta di una operazione di read;
- con il terzo battito si prende il dato dal data bus dove la memoria ha provveduto a depositarlo.

STORE

- con il primo battito di clock si pone l'indirizzo del registro di memoria di cui vuole leggere il contenuto sull'address bus;
- con il secondo battito si deposita il dato sul data bus;
- con il terzo battito si segnala alla memoria che si tratta di una operazione di write e quindi che il dato è pronto per essere depositato nel registro selezionato.

Evoluzione del modello di Von Neumann: interruzioni

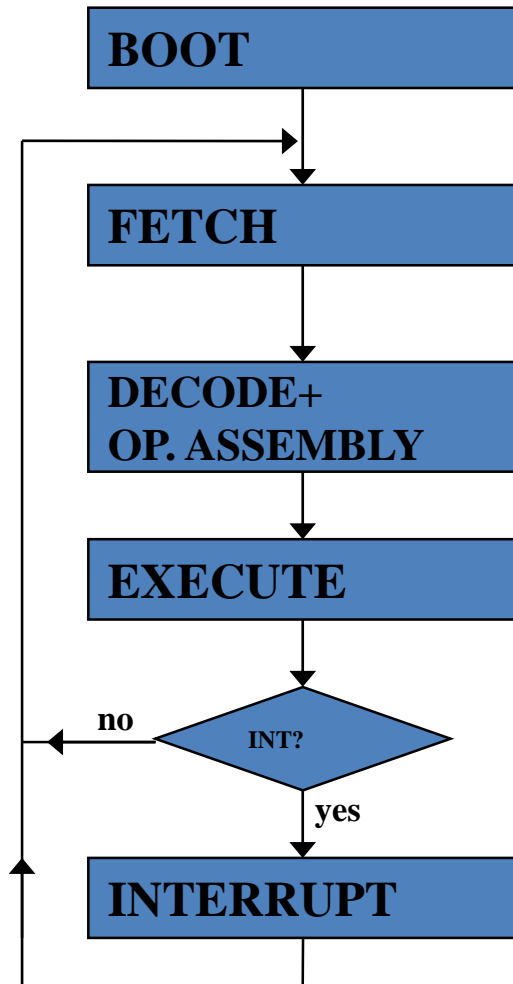


- Se il ciclo del processore fosse effettivamente quello mostrato in figura, sorgerebbero alcuni problemi, come per esempio:
 - un'applicazione “prepotente” potrebbe impadronirsi della risorsa processore senza mai lasciarla
 - non ci sarebbe modo di rimuovere forzatamente un'applicazione che entri per errore in un ciclo infinito
 - il sistema operativo, in generale, avrebbe un controllo limitato sul sistema

Evoluzione del modello di Von Neumann: interruzioni

- La soluzione comunemente adottata consiste nel prevedere che, in caso di eventi eccezionali (di solito *asincroni*), l'esecuzione del programma venga interrotta per gestire tali eventi
- Gli eventi eccezionali possono essere di vario tipo:
 - RESET del processore
 - Errori interni causati dall'esecuzione di un'istruzione (es. divisione per zero, o codice operativo errato)
 - Richieste esterne di attenzione provenienti da periferiche di input/output

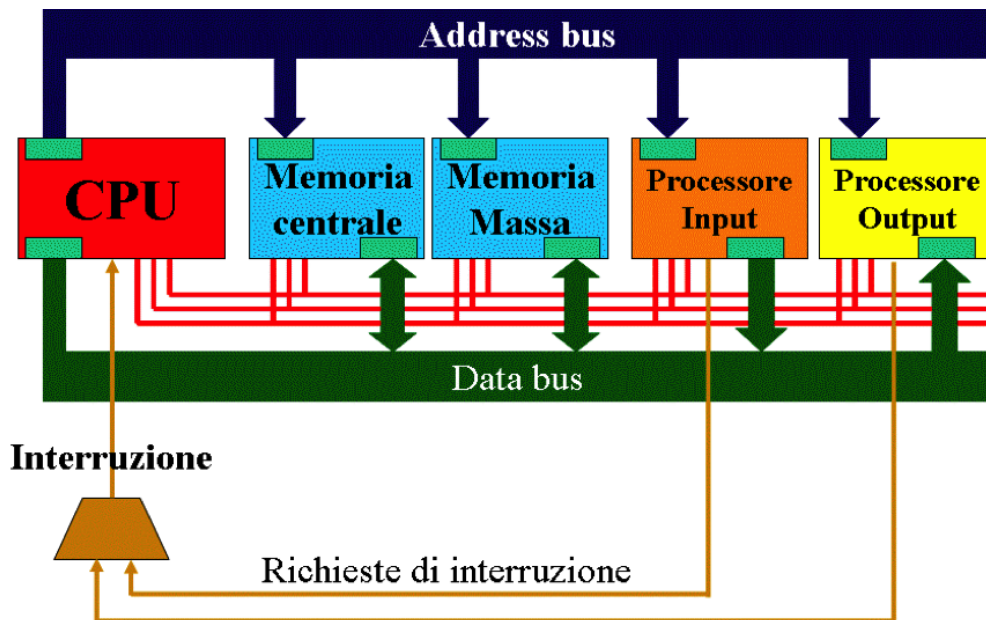
Ciclo del processore con interruzioni



- Quando si scatena un'interruzione (interna o esterna), vengono generati appositi segnali che avvisano il processore.
- Contemporaneamente, vengono settati dei bit all'interno del cosiddetto **registro di stato** (che include i CC)
 - Tali bit potrebbero semplicemente avvisare che c'è stata un'interruzione, oppure dare un'indicazione sulla relativa priorità, oppure codificare la causa dell'interruzione (se interna)
- La UC controlla il registro di stato alla fine dell'esecuzione di ogni istruzione:
 - se non sono segnalate interruzioni procede normalmente con il prelievo dell'istruzione successiva;
 - in caso contrario comincia l'esecuzione di un programma del sistema operativo, detto **ISR** (interrupt service routine), che «serve» l'interruzione;
 - nel caso si accorga della presenza di più richieste contemporanee stabilisce quale servire per prima secondo criteri di importanza o priorità di intervento.

Le interruzioni esterne

1. Il Dispositivo genera il segnale di interruzione (INT);
2. se la priorità del Dispositivo ha un valore adeguato, il Processore interrompe il programma in esecuzione;
 - Significa che il processore tiene traccia della priorità del programma che sta eseguendo in ogni istante
3. il Dispositivo viene informato che l'interrupt è stato ricevuto (ACK);
4. viene eseguita la procedura di servizio dell'interruzione (ISR);
5. si ripristina il Programma originale dal punto in cui era stato interrotto.



- In alcuni casi i dispositivi non necessitano della CPU perché utilizzano hardware dedicato (es. DMA per accesso in memoria)
 - la CPU può attivare il dispositivo e disinteressarsene, continuando a lavorare
 - Quando il dispositivo termina il suo compito, avanza una richiesta di interruzione al processore centrale e aspetta che gli venga rivolta attenzione.

Identificazione dei dispositivi (1/2)

- Se ci sono più dispositivi che possono interrompere, il processore deve essere in grado di identificare il dispositivo che ha generato l'interruzione, poiché probabilmente diverse azioni dovranno essere intraprese a seconda del particolare dispositivo
- I dispositivi hanno una linea comune attraverso la quale segnalano richieste di interruzioni (INT)
- Quando INT è alto si pone il problema di identificare da quale dispositivo è partita la richiesta, poiché potrebbero esserci più richieste contemporanee

Identificazione dei dispositivi (2/2)

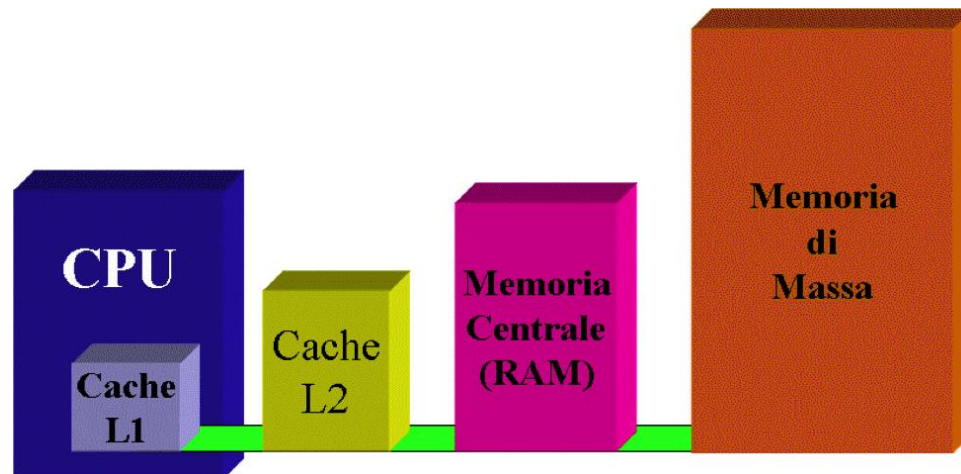
- Esistono diverse soluzioni a questo problema, e tutte impiegano un misto di hardware e di software. Tutte le soluzioni dipendono fortemente sia dall'architettura del sistema che da quella del processore
 - Polling
 - Il processore interroga il **registro di stato** di tutti i dispositivi, in cui c'è un bit che diventa alto (1) quando il dispositivo genera un'interruzione. L'ordine con cui vengono interrogati i dispositivi potrebbe dipendere dalla posizione degli stessi (daisy chain) o dalla loro priorità (stabilita a priori). Viene servito il dispositivo più prioritario.
 - Interrupt vettorizzato
 - Il processore è in grado di conoscere chi è il dispositivo interrompente perché esso è connesso ad una particolare linea fisica associata staticamente ad un valore di priorità (autovettorizzato) oppure perché esso manda il proprio identificativo in risposta al segnale di ack secondo un protocollo specifico. In questo secondo caso la priorità potrebbe essere gestita da un dispositivo apposito.

Evoluzione del modello di Von Neumann: le Cache

- Per ridurre i tempi di trasferimento dalla memoria centrale ai registri interni della CPU, viene replicata una porzione di memoria e posta tra memoria e CPU stessa. Tale memoria, molto veloce, viene chiamata *cache* e fa da buffer per il prelievo di informazioni dalla memoria centrale.
 - Con operazioni particolari, istruzioni e dati vengono trasferiti dalla memoria centrale nella cache secondo la capacità di quest'ultima (si parla di pochi mega byte nei moderni processori).
 - La UC procede nelle tre fasi del suo ciclo al prelievo di istruzioni e operandi dalla cache.
 - Quando la UC si accorge che il prelievo non può avvenire scatta un nuovo travaso dalla memoria centrale.
 - Se la cache è interna alla CPU viene detta di primo livello (L1);
 - Le cache di secondo livello (L2) sono invece esterne e solitamente un po' più lente di quelle di primo livello ma sempre più veloci della memoria centrale.
 - la cache L2 risulta 4 o 5 volte più lenta della cache L1 mentre la RAM lo è addirittura 20 o 30 volte. I due livelli possono coesistere.

Gerarchie di Memorie

- Consente di offrire ai programmi l'illusione di avere una memoria grande e veloce.
 - Nella gerarchia i livelli più prossimi alla CPU sono anche quelli più veloci, ma sono anche quelli con dimensioni più piccole visto il loro elevato costo.
 - I livelli più lontani sono quelli che mostrano una capacità massima ed anche tempi di accesso maggiori.



Classificazione dei processori

- sulla base dei seguenti parametri:
 - **parallelismo** esterno espresso come numero di bit trasferiti o prelevati in un singolo accesso in memoria (8, 16, 32, 64,...) e caratterizzanti quindi il suo data bus;
 - **capacità di indirizzamento** legata alla dimensione in bit del suo address bus
 - numero, tipo e parallelismo dei registri interni;
 - **tecniche di indirizzamento** intese come la modalità con la quale costruire **l'indirizzo logico con il quale prelevare o salvare il valore dell'operando** di una istruzione;
 - gestione delle periferiche di **input ed output**;
 - **repertorio delle istruzioni** inteso come numero e tipo di istruzioni costituenti il suo linguaggio macchina;
 - **tempi necessari all'esecuzione di alcune istruzioni fondamentali** come l'addizione da utilizzare per la valutazione del MIPS con il quale effettuare confronti sulle prestazioni.

Processori RISC e CISC

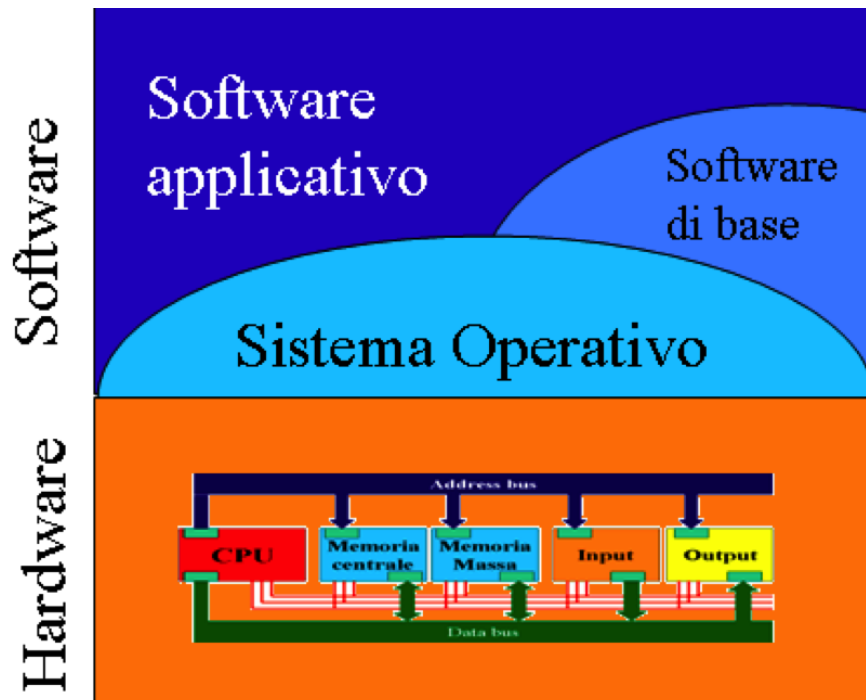
- CISC (Complex Instruction Set Computer)
 - inserimento nel linguaggio macchina istruzioni con potenza espressiva prossima a quella dei linguaggi di programmazione di alto livello
 - sono caratterizzati quindi da un ampio repertorio di istruzioni
 - sviluppo di programmi più semplice
 - ... anche se molte di esse non risultano strettamente necessarie, potendosi ottenere con l'esecuzione di sequenze di istruzioni più semplici
 - maggiore complessità costruttiva
- RISC (Reduced Instruction Set Computer).
 - repertorio costituito da un ridotto ed essenziale insieme di istruzioni al fine di ottenere processori più veloci e di costo ridotto, data la minore complessità del loro progetto.
 - L'obiettivo fondamentale dell'approccio RISC è di disporre di un insieme fondamentale di istruzioni per ridurre al minimo il numero dei cicli di macchina (clock) necessari per loro esecuzione.
 - Tutte le istruzioni RISC fondamentali hanno la stessa durata (un ciclo macchina), la stessa lunghezza e lo stesso format

Linguaggio macchina e linguaggio assembly

- Una *istruzione macchina* è una sequenza di 0 e 1 suddivisa in campi, diversi nel formato e nel significato a seconda dell'istruzione
- Nel riferirsi alle istruzioni macchina si preferisce la rappresentazione simbolica a quella numerica, costituita dal **linguaggio assembly** (o assembler)
- L'**assemblatore** è il programma che esegue la traduzione da linguaggio assembly a linguaggio macchina.
- E' il più semplice traduttore di linguaggi presente in informatica
 - fa corrispondere ai codici mnemonici del codice operativo (es. ADD, MOVE, JMP) il rispettivo codice binario;
 - converte da decimale a binario indirizzi e valori dei dati;
 - converte dati alfanumerici nella loro rappresentazione binaria.

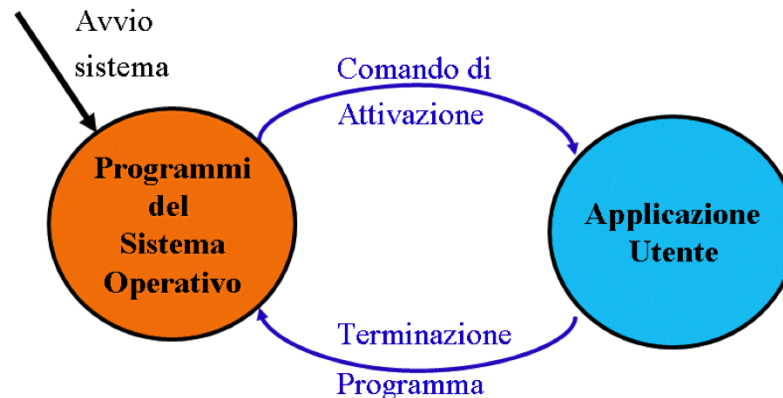
Hardware vs. Software

- programmi che servono a tutti gli utenti del sistema
 - classificati come *software di base*
 - sistemi operativi e traduttori dei linguaggi di programmazione
- programmi che risolvono problemi specifici
 - *software applicativo*.



Sistema Operativo

- Il sistema operativo è un insieme di programmi che deve garantire la gestione delle risorse hardware in modo semplice ed efficiente a tutti gli utenti del sistema
 - ... persone oppure altre applicazioni.
 - I primi calcolatori non avevano il sistema operativo.
 - In essi il programmatore doveva prevedere tutto
 - Al termine dell'esecuzione del programma il programmatore o l'operatore del sistema doveva provvedere ad un nuovo caricamento in memoria ed ad una successiva attivazione.
 - Con il sistema operativo il passaggio da una applicazione ad un'altra è svolto in automatico
 - La CPU si trova così ad eseguire i programmi del sistema operativo in alternanza con quelli applicativi.



Funzioni di un Sistema Operativo

- Il Sistema Operativo è responsabile della gestione e del controllo:
 - delle componenti hardware che costituiscono il computer (processi di Input/Output da e verso le periferiche collegate al sistema);
 - dello scheduling ed esecuzione dei programmi (processi) e dell'assegnazione delle risorse agli stessi;
 - dell'archiviazione e dell'accesso ai file su memoria di massa;
 - del trasferimento di dati dalla memoria di massa a quella principale e viceversa per l'esecuzione dei programmi;
- Infine, se è prevista interazione con l'utente, offre un'interfaccia software (grafica o testuale) per accedere alle risorse hardware (dischi, memoria, I/O in generale) del sistema