



[e-mail per suggerimenti](#)

1 CENNI SULLA STRUTTURA DEL CALCOLATORE

Per comprendere il funzionamento di un calcolatore elettronico, si considerino le operazioni che una persona (*calcolatore umano*) deve eseguire per risolvere un problema e fornire i risultati, conoscendo i dati iniziali.

In ordine le varie fasi si possono così sintetizzare:

- 1°- Si è in presenza di un problema da risolvere del quale si conosce con esattezza ciò che si deve conseguire.
- 2°- Si deve sviluppare un metodo di risoluzione detto algoritmo , come sequenza ordinata di istruzioni, mediante la quale possono essere risolti più problemi, ciascuno di una determinata classe (aree, volumi, soluzioni di sistemi...), in modo che, introdotti i dati iniziali come input, si possa pervenire ad un risultato finale output (uscita).
- 3°- Si introducono i dati iniziali (INPUT).
- 4°- A questo punto il calcolatore umano ha a disposizione dei fogli sui quali è riportato (o si riporta) l'algoritmo, che fornisce l'elenco della successione di istruzioni delle operazioni da eseguire. Inoltre l'operatore ha a disposizione una calcolatrice manuale (o effettua il calcolo a mente).
- 5°- Così il calcolatore umano, introdotti i dati iniziali (input), letto l'algoritmo, esegue il calcolo con la calcolatrice (o a mente) e registra i risultati parziali (risultati intermedi) su dei fogli. Alla fine il calcolatore umano effettua una relazione, dando i risultati finali (output).

In questo caso il calcolatore umano controlla il tutto: Dati - Algoritmo - Macchina Calcolatrice - Calcolo - Risultati intermedi - Risultati finali.

Nella descrizione fatta si possono distinguere le seguenti *funzioni*

Dati di ingresso

È l'operazione di introduzione dei dati e dell'algoritmo (*input*) forniti dall'esterno. Sui dati verranno poi effettuate le operazioni previste nell'algoritmo.

Uscita valori

Estrazione delle informazioni (*output*). Consiste nella trascrizione dei risultati in un supporto: fogli di carta nel caso del calcolatore manuale.

Memorizzazione

È la funzione mediante la quale si fissano su un supporto i dati iniziali, i risultati intermedi ecc.

Il calcolatore umano fissa tutto (*memorizza*) su fogli di carta.

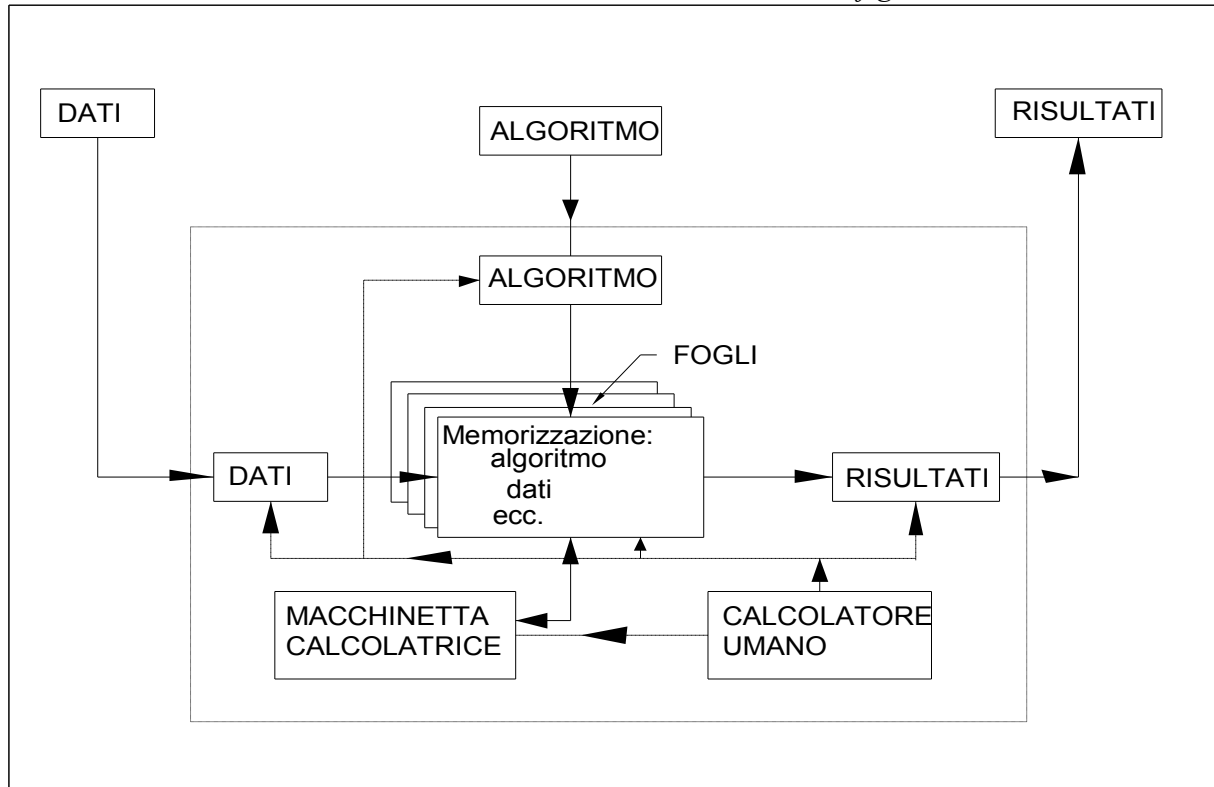
Calcolo

È la funzione mediante la quale vengono eseguite le operazioni, descritte nell'algoritmo, sui dati introdotti in ingresso, ottenendo in uscita i risultati finali (*output*).

Funzione di controllo

È la funzione che gestisce il sistema di elaborazione (è l'uomo nella elaborazione manuale). Consiste: nel leggere i dati- interpretare in modo corretto le istruzioni contenute nell' algoritmo - governare il funzionamento della calcolatrice - stabilire la sequenza - gestire i supporti (fogli di carta nell'esempio) sui quali sono memorizzati i risultati - provvedere alla estrazione dei risultati.

fig. 1.1



Sistema di elaborazione automatico

Nel sistema manuale tutte le funzioni rilevate sono gestite dall'uomo. Esse possono essere automatizzate nel calcolatore elettronico: schema fig.1.2

Si possono impostare le seguenti sostituzioni:

- | | | |
|-----------------------------|--------|--------------------------------|
| <i>Fogli di carta</i> | =====> | <i>Supporti di memoria</i> |
| <i>Calcolatrice manuale</i> | =====> | <i>Unità Aritmetica Logica</i> |
| <i>Calcolatore umano</i> | =====> | <i>Unità di Controllo</i> |
| <i>Introduzione dati</i> | =====> | <i>Unità di ingresso-Input</i> |
| <i>Estrazione dati</i> | =====> | <i>Unità di uscita-Output</i> |

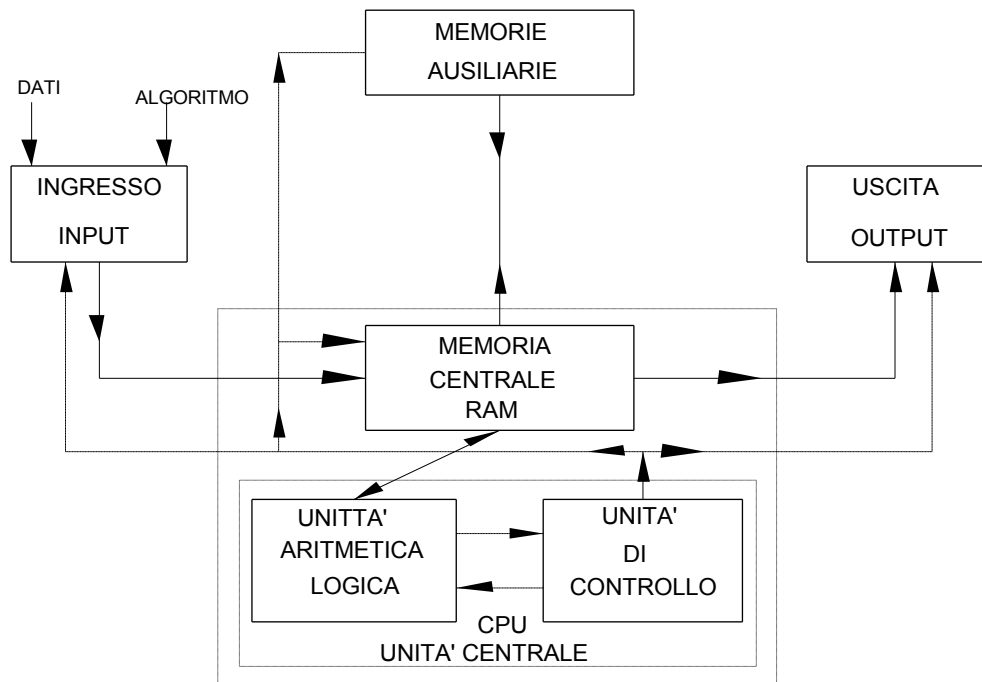


Dove CPU sta per "Central Processing Unit": Unità Centrale di Elaborazione.

Introdotti i dati e l'algoritmo nell'input, questi vengono registrati nella memoria centrale del calcolatore.

Successivamente, i dati e l'algoritmo vengono prelevati dalla memoria centrale e inviati nella *Unità Aritmetica Logica*, la quale eseguirà le operazioni previste nell'algoritmo e restituirà un Risultato che verrà registrato nella memoria centrale. Alla fine, i risultati vengono inviati all'uscita "Output".

fig. 1.2



L'UNITA' DI CONTROLLO sovrintende alla gestione del sistema

La potenza di un processore è data dal numero di operazioni che può effettuare in un secondo espressa GHz (il cloc). 1GHz corrisponde a un miliardo di operazioni al secondo.

Maggiore potenza si ottiene con processori **multi core** (nucleo elaborativo); ossia a più nuclei elaborativi capaci di elaborare i dati individualmente e eseguire, così, le operazioni più rapidamente con maggiore potenza di calcolo, senza aumentare la **frequenza di clock**, diminuendo il calore dissipato .

Vi sono così processori **dual core** da 2 GHz e più.

Sono in commercio CPU a più nuclei elaborativi di marca **INTEL** e **AMD**.

Così della marca **INTEL** si hanno diverse opzioni nelle serie **i1, i3, i6, i7**.

In **i5** si hanno processori a 2 o a 4 core.

In **i7** si hanno processori a 2, 4, 6 core.

Riguardo alla scelta tra un processore **INTEL** o **AMD**, va detto che, in generale, i chip di **AMD**, a parità di potenza, sono più economici dei processori **INTEL**. Questi però offrono una gamma di scelta con maggiori prestazioni complessive più veloci, preferibili su computer con scheda grafica più potente, Così se si vuole un computer di alte prestazioni conviene la scelta di processori **INTEL i5, i7**.

La **i7** è da scegliere, attualmente, per ottenere il massimo di prestazione per giocare videogame graficamente o per utilizzare programmi professionali che richiedono alte risorse.

I processori **AMD** sono apprezzati invece da utenti che utilizzano frequentemente il computer con abilità installando programmi e funzioni (gli smanettoni) ottenendo il massimo con le procedure di **overclock**

Memoria cache

Nelle operazioni svolte dal computer vi sono delle istruzioni che normalmente vengono più usate per il suo funzionamento. Queste istruzioni sono memorizzate in una limitata memoria detta **memoria cache**. Un'istruzione viene subito ricercata nella memoria cache, se non è in questa va cercata nella RAM

Registri

L'unità centrale CPU è dotata di appositi **registri** (non rappresentati nello schema), per la memorizzazione temporanea e veloce di dati, risultati e informazioni di controllo; vedi oltre.

Periferiche

Le unità di ingresso e di uscita possono essere più di una e di varia natura: *stampanti, video, tastiera...* Alcune di queste possono essere dei supporti magnetici, aventi funzione di *Memorie Ausiliarie* che vengono così ad aggiungersi alla Memoria Centrale. Tutte queste unità vengono denominate Unità Periferiche.

Unità di Canale

Occorre osservare che la *CPU*, per effettuare una operazione, impiega un tempo molto più piccolo di quello impiegato dalle periferiche per introdurre i dati o estrarre valori ecc.

Per poter minimizzare l'inconveniente della disparità dei tempi, per cui il CPU debba rimanere inattivo nell'attesa del disimpegno delle periferiche, sono state introdotte le *Unità di Canale*.

Le *Unità di Canale* collegano le periferiche alla *CPU* e hanno **capacità di elaborazione autonoma**. La *CPU* trasmette l'input alle Unità di canale e queste autonomamente effettuano le operazioni richieste, tenendo conto del tipo e del funzionamento della periferica collegata. Così la *CPU* può effettuare altre operazioni più complesse, mentre le Unità di Canale stanno eseguendo, ed elaborando autonomamente, i comandi precedentemente trasmessi dal *CPU* stesso.

1.1 TIPI DI MEMORIA

1.1.1 Memoria Principale

È la memoria centrale del CPU.

È di tipo RAM -*Random Acces Memory*-(*memoria ad accesso casuale*) Si occupa di memorizzare temporaneamente (virtualmente) i dati e le informazioni delle varie operazioni e attività effettuate al PC. La memorizzazione avviene casualmente con lo stesso tempo di accesso. Essa è capace di immagazzinare una quantità limitata di informazioni accessibili in tempi brevi dell'ordine dei nanosecondi.

La memoria centrale di un calcolatore è costituita da più *celle*, ciascuna delle quali ha la capacità di 1 bit : essa può assumere cioè due stati opposti, come i contatti di un relè (eccitato: 1 -diseccitato: 0).

Le celle sono poi raggruppate in *parole*.

La dimensione minima di una parola è di $2^3=8$ bit . tale dimensione viene denominata *byte*

$$1 \text{ byte}=8 \text{ bit}$$

La parola può essere composta da più di un *byte* (*più di 8 bit*). Vi sono così parole di 16, 32, 64 bit.

Come si vedrà oltre, la memoria è indirizzabile a livello di parola, in quanto ogni parola è contraddistinta da un indirizzo.

Una delle caratteristiche fondamentali di una memoria è il numero di byte che la costituiscono e che determina la capacità di memorizzazione. Come unità di detta capacità, usualmente, viene scelta la potenza 10 di 2 detta K (kbyte):

$$1 K = 2^{10} \text{ byte} = 1024 \text{ byte}$$

È evidente che maggiore è la capacità della RAM di un CPU e maggiore sarà la velocità di elaborazione dei dati e le varie operazioni che si possono svolgere al computer: entrare e navigare su Internet, utilizzare videogiochi, programmi, aprire cartelle, documenti, file ecc.

La qualità di un computer è anche espressa da una sufficiente capacità della RAM, un buon computer ha capacità da un **minimo di 2-3 GB di RAM a 4 GB di RAM** o superiore. Molti PC hanno nella scheda madre **altri slot di RAM liberi** che in futuro permettano di **espandere la memoria RAM ulteriormente**.

1.1.2 Memorie Ausiliarie

Vengono anche dette *Memorie di Massa*. Esse immagazzinano una grande quantità di informazioni. l'accesso richiesto è molto più lungo di quello della memoria principale: 10^{-3} sec .

Riassumendo, le *unità* viste si possono distinguere in

Unità con funzioni elaborative	$\left\{ \begin{array}{l} \text{Unità di Controllo} \\ \text{Unità Aritmetica Logica} \\ \text{Unità di Canale} \end{array} \right.$
Unità con funzioni di registrazione	$\left\{ \begin{array}{l} \text{Memoria principale} \\ \text{Memorie ausiliarie} \\ \text{Input - Output} \end{array} \right.$

1.1.2 Memori di massa **HARD DISK**

È un disco rigido magnetizzato utilizzato per la memorizzazione dei programmi installati nel computer, e dei file, foto, video ecc.. aggiunti dall'utente.

È denominato *hard disk driver*, o semplicemente **hard disk HDD**. Esso può essere installato sia nei **computer desktop** da scrivania sia in quelli **portatili**. In questi si sempre va più impiegando dischi allo stato solido **SSD**.

L'hard disk HDD è costituito essenzialmente da uno o più piatti, rivestiti di materiale magnetico posti in rapida rotazione. Per ogni disco vi sono due testine, una per lato, mantenute a piccolissima distanza dalle superficie dei dischi dall'aria mossa dalla rotazione

Capacità di un hard disk

La capacità di un hard disk è misurata dal numero di byte che possono essere in esso memorizzati. Si costruiscono nel tempo hard disk sempre con maggiore capacità e a minor costo; attualmente hanno valori dell'ordine di 500GB fino a 1TB o più. Se poi serve più spazio si potrà comprare anche un **Hard Disk esterno** da collegare con il cavo USB al PC e avere così altra memoria aggiuntiva

Velocità dell'hard disk rpm

Importante per la rapidità del reperimento dati è la velocità di rotazione del disco. La velocità è espressa in **rpm**, con il significato di **giri per minuto**.

Un buon hard disk, attualmente, deve avere una velocità non inferiore a **5400 rpm**. Si hanno dischi a **7200 rpm** fino a **15.000 rpm**.

Tempo di accesso

È il tempo medio necessario per il reperimento di un dato posto in un punto casuale del disco. Esso dipende dalla velocità della testina a spostarsi sulla traccia dove risiede il dato e dalla velocità di rotazione del disco. Per minimizzare il tempo di spostamento della testina sulla traccia si cerca di realizzare testine sempre più leggere in modo da diminuire la massa inerziale.

. Il tempo di accesso tipico per un disco rigido da 7200 rpm è di circa 9 (ms), per uno da 15.000 rpm è inferiore a 4 ms

Interfaccia

È il tipo di connessione interna, tramite un cavo, che collega l'**hard disk** con il **PC**.

Attualmente la maggior parte dei computer montano hard disk con **interfaccia SATA** ("Serial Advanced Technology Attachment"), che ha sostituito la vecchia **ATA**.

Vi sono interfacce SATA con diversa velocità di trasferimento dati

La *velocità di trasmissione*, espressa in bit/s e principalmente dai suoi multipli; usuale è il

$Gbit/s = 10^9 \text{ bit/s}$

Nel 2002 la prima versione dell'interfaccia SATA supportava una velocità di 1,5Gbit/s

Da WikidediaA si riportano i seguenti dati

	Nome commerciale	Prestazioni teoriche	Prestazioni effettive
SATA 1.0	Serial ATA-150	1,5 Gbps (192 MB/s)	1,2 Gbit/s (150 MB/s)
SATA 2.0	Serial ATA-300	3 Gbps (384 MB/s)	2,4 Gbit/s (300 MB/s)
SATA 3.0	Serial ATA-600	6 Gbps (768 MB/s)	4,8 Gbit/s (600 MB/s)

1.1.3 Disco allo stato solido **SSD**

È una memoria di massa allo **stato solido** basata su **semiconduttore**. È indicato con la sigla **SSD** (*solid-state drive*). Si può considerare una evoluzione dei tradizionali hard disk.

La memorizzazione avviene in modo statico, con assenza di rumore, senza l'utilizzo di organi in movimento che comportano la possibilità di rotture. La memoria SSD non ha nulla a che vedere con la forma di un disco; la sua denominazione deriva dal fatto che essa svolge le stesse funzioni di un HDD.

Il reperimento dei dati non dipende dalla velocità di movimento di organi meccanici, ottenendo così una **velocità di scrittura e lettura elevata fino a 50 volte più rapida dei tradizionali HDD**.

I vantaggi della memoria di massa SSD si possono così riassumere:

- assenza di rumorosità non essendo presenti organi in movimento
- minore possibilità di rotture
- minori consumi durante il procedimento di scrittura e lettura
- tempi di accesso e archiviazione ridotti, fino a 50 volte minori di un HDD. I tempi sono dell'ordine dei millisecondi
- maggiore resistenza agli urti

- minore produzione di calore

Si riscontrano i seguenti svantaggi:

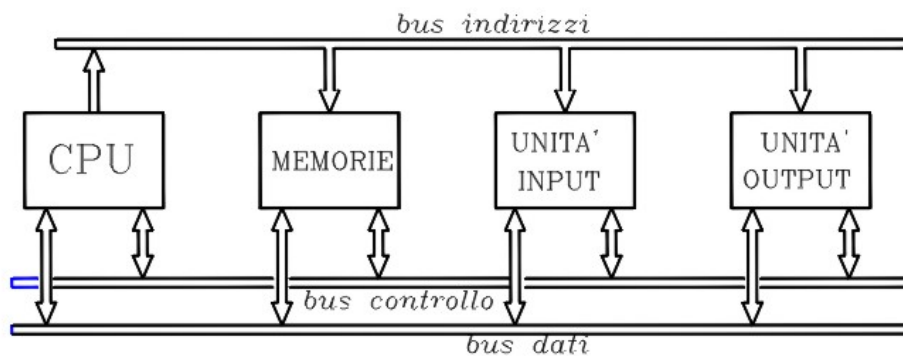
- la memoria SSD deve essere rimossa correttamente per non provocare perdita di dati
- ha maggiore prezzo che può variare da 0,50 fino a 1,00 €/GB
- si ha minore permanenza di dati quando non è alimentato; la permanenza dipende inoltre dalla temperatura.

1.2 I BUS

Il collegamento fisico tra le varie parti del calcolatore viene effettuato con un gruppo di fili, uno per ogni bit, che costituisce una struttura detta a **bus**.

Bus Gruppo di fili, uno per bit, sui quali transitano le informazioni scambiate tra il CPU, le MEMORIE e le PERIFERICHE.

fig.1.3



Bus Indirizzi È unidirezionale: dal CPU alle altre unità. È costituita di norma da almeno 16 fili; può in questo caso trasportare in parallelo 16 bit, uno per filo.

Bus Dati È bidirezionale; in esso transitano dati e istruzioni nei due sensi tra CPU e le altre unità. Il numero di bit che in esso possono transitare definiscono la grandezza della parola del sistema.

Bus Controllo È bidirezionale; gestisce il controllo tra le varie parti del sistema. Ha la capacità di 1 bit.

1.3 Memorizzazione delle informazioni

La memoria è un dispositivo nel quale è possibile introdurre informazioni, ivi conservarle, e poterle estrarre successivamente.

La memoria è costituita da un supporto, che subisce delle modifiche dello stato fisico, dipendenti dalla informazione tradotte in un determinato codice.

La memoria è costituita da più *celle* elementari, ciascuna capace di memorizzare un carattere particolare.

La codifica ormai in uso assume come caratteri due soli stadi, corrispondenti ai due simboli binari 1,0. Una cifra binaria viene detta *bit*. La *cella* è capace di memorizzare una cifra binaria: 1 *bit*.

La capacità di una memoria si riferisce al numero totale di bit che possono essere immagazzinati in essa. Il bit è una unità molto piccola; per esprimere la capacità di una memoria si adoperano i suoi multipli:

$$1 \text{ BYTE} = 8 \text{ bit}$$

Il byte con le sue combinazioni $2^8=256$ è sufficiente a rappresentare tutti i caratteri alfa numerici più altri simboli.

$$1 K = 1 KBYTE = 2^{10} = 1024 BYTE$$

Così la memoria di 64K = 64x1024 = 65536 byte

$$1 M = 1000K = 1000x1024 = 1.024.000 \text{ byte}$$

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

1.3.1 Parola

Le celle sono raggruppate in parole.

La parola può essere formata da uno o più byte: essa può essere composta da 8, 16, 32, 64 bit.

La memoria di un calcolatore è indirizzabile a livello di parole

Per ottenere ciò ogni parola è contraddistinta da un *indirizzo* che inizia da 0 e prosegue nella sequenza dei numeri naturali. Quando la parola è formata da più byte allora l'indirizzo è sul primo byte di sinistra o l'ultimo di destra.

1.3.2 Tempo di accesso

Una caratteristica della memoria è la velocità di funzionamento, che può essere espressa dal tempo che intercorre tra l'istante nel quale viene richiesto il contenuto della parola e quello nel quale esso è disponibile. Nella memoria RAM è misurato in nanosecondi, nei dispositivi di memoria di massa in millisecondi (ms)

In riferimento al tempo di accesso le memorie si possono distinguere in tre categorie:

- Memorie ad accesso uniforme*
- Memorie ad accesso diretto*
- Memorie ad accesso sequenziale*

1.3.3 Memoria ad accesso uniforme

Costituisce la memoria principale - Ogni parola, scelta a caso, ha lo stesso tempo di accesso - Tale memoria viene denominata *RAM (Random Access Memory)*.

La capacità delle RAM è misurata dal numero di byte che può memorizzare, espresso in multipli di esso.

Si hanno memorie con più moduli di 128 MB

Così sono disponibili memorie con: 128, 256, 512, 1024, ... fino a 4 GB

Il tempo di accesso è dell'ordine dei nanosecondi. La velocità di accesso è misurata in Mhz. I primi moduli erano a 440 Mhz, ora le DDR3 hanno velocità di 1200-1330 Mhz

1.3.4 Memoria ad accesso diretto

L'accesso dipende dal movimento di elementi meccanici che vanno a cercare l'informazione (con indirizzo) contenuta in una certa parte di un supporto. Tali sono le memorie ausiliarie come: *floppy disk - hard disk*

Il tempo di accesso è superiore a quello che si ha nella RAM : dipende dalla velocità degli elementi meccanici e dalla loro posizione iniziale

1.3.5 Memoria ad accesso sequenziale

L'accesso avviene secondo una determinata sequenza di esame di un supporto, come ad esempio nei nastri magnetici. In tal caso, il tempo dipende molto dalla posizione della registrazione rispetto alla posizione dell'elemento di lettura.

Le memorie del *C.P.U.* inoltre si possono distinguere in:

1.3.6 Memorie volatili (*delebili*)

Il contenuto si cancella ogni volta che viene tolta l'alimentazione al *CPU*. Appartiene a tale categoria la memoria centrale *RAM*.

1.3.7 Memorie indelebili

Il contenuto si mantiene anche quando viene tolta l'alimentazione

Altra distinzione delle memorie si ha in base alla capacità di poter in esse scrivere o solamente leggerne il contenuto

1.3.8 Memoria di scrittura e lettura

È la memoria *RAM -Random Acces Memory-(memoria ad accesso casuale)* nella quale si può scrivere e leggere (effettuare programmi). Essa , come si è detto, è volatile. È di limitata capacità, di alta velocità di accesso

1.3.9 Memoria di sola lettura

Viene denominata *ROM (Read Only Memory)*. In essa sono registrati dati, istruzioni prima di montare il calcolatore. La *ROM* è indelebile, è di sola lettura ; non è possibile variare i dati e le istruzioni in essa contenuti. Non è programmabile dall'operatore.

1.3.9 Memoria PROM

La *ROM* non è programmabile dopo la fabbricazione. Una modifica della *ROM* è la memoria *PROM (Programmable Read Only Memory)*. Essa è una memoria *ROM* programmabile, nella quale possono essere immessi i dati binari dopo la fabbricazione. Vi è nella *PROM una struttura di base* e la possibilità di aggiungere informazioni diverse secondo le esigenze. In questa memoria però commesso un errore, non vi è la possibilità di correggerlo: non si può cancellare.

1.3.12 Memoria EPROM

La memoria *EPROM (Erasable Programmable Read Only Memory)* è una memoria *PROM* di sola lettura, programmabile, con la possibilità di cancellazione. Fatto un errore, occorre però cancellare tutto il programma eseguito e ricominciare tutto daccapo.

1.3.12 Memoria EAROM

Un miglioramento della *EPROM* è la memoria *EAROM* - *Electrically Alterable Read Only Memory*. È una memoria di sola lettura, programmabile, con la possibilità di cancellare una porzione del programma.

1.4 II BIOS

Il BIOS (**Basic Input-Output System**) è un insieme di istruzioni di routine che fornisce una serie di **funzioni base** per il funzionamento del computer.

Il BIOS è scritto su memoria ROM o altra memoria non volatile. Esso procura le istruzioni base, per l'accesso all'hardware del computer e alle periferiche; ha il compito di fornire i primi comandi per la fase di avvio (**bootstrap**) con il caricamento **nella memoria primaria** del kernel del sistema operativo. In questa fase, sono eseguiti i controlli preliminari per testare la funzionalità dei componenti fondamentali quali: *RAM, tastiera, mouse, dischi, porte, ecc.*

Per potere correttamente dare inizio al funzionamento delle periferiche il BIOS legge alcuni parametri (come l'ora corrente) da una piccola RAM in tampone con una batteria.

Con la necessità di poter ottenere aggiornamenti sempre più frequenti è invalso l'uso di memorizzare il BIOS su memorie EPROM

1.5 FUNZIONE DI CONTROLLO E DI CALCOLO UNITA' DI CONTROLLO

Le funzioni di controllo e di calcolo vengono svolte nell'*Unità di Elaborazione Centrale* che comprende l'*Unità di Controllo* e l'*Unità Aritmetica Logica* (ALU) e i *Registri*.

All'*Unità di Elaborazione Centrale* sono accessibili solamente dati e istruzioni contenuti nella *memoria centrale*. Le informazioni, contenute nelle *memorie di massa*, devono essere prelevate e portate nella *memoria centrale* per poter essere, poi, elaborate dalla ALU.

Le istruzioni da eseguire vengono presentate come una sequenza di ordini elementari, in modo da essere interpretati dall'UNITÀ DI CONTROLLO.

Le istruzioni si possono distinguere in:

- *Istruzioni macchina*
- *Istruzioni simboliche*

Le seconde verranno spigate in seguito.

Le istruzioni macchina sono redatte in *linguaggio base della macchina* secondo un codice binario; Vengono registrate sequenzialmente nella memoria centrale e ne occupano una porzione a volte consistente. Tale occupazione è temporanea: le istruzioni che non servono più, vengono sostituite da altre.

L'istruzione macchina è divisa in due parti:

1°-Codice operativo È la prima parte dell'istruzione macchina che caratterizza il tipo di operazione. È contraddistinta da un codice *CO*, detto *codice operativo o di funzione*.

2°-Operandi L'altra parte dell'istruzione macchina è composta dai dati dell'operazione, detti *operandi*, ognuno dei quali è contraddistinto da un proprio indirizzo *I₁, I₂...*

L'indirizzamento della memoria principale ad accesso uniforme (*RAM*) si ottiene stabilendo una corrispondenza biunivoca tra l'insieme delle parole, che costituiscono la memoria, e l'insieme dei numeri naturali, compreso lo zero. Così, se la memoria è costituita da $2^{10}=1024$ byte l'indirizzo di ogni byte va da 0 a 1023.

Alla stessa maniera, per gli indirizzi degli operandi vi è un codice che li associa biunivocamente alle celle o alle parole della memoria, e ciò si realizza fisicamente con dei

circuiti capaci di leggere o registrare il contenuto di una parola che si trova in un determinato indirizzo.

Consideriamo per esempio l'istruzione corrispondente all'operazione di sottrazione tra due numeri. L'istruzione macchina può essere così suddivisa:

CO	I ₁	I ₂	I ₃
----	----------------	----------------	----------------

CO è il codice dell'operazione sottrazione. I₁ I₂ I₃ sono gli indirizzi, rispettivamente, del minuendo, sottraendo, differenza.

Di norma l'interpretazione e l'esecuzione di una istruzione si susseguono secondo l'ordine di registrazione nella memoria principale, a partire da un indirizzo iniziale. Vi sono però istruzioni che permettono di alterare l'ordine, effettuando dei salti che interrompono la sequenza un corso, per eseguire altre istruzioni.

1.6 MICROPROGRAMMI - MICROISTRUZIONI

Una istruzione macchina viene tradotta dall'*Unità di Controllo* in una sequenza di comandi o di istruzioni elementari.

Nei primi calcolatori ogni istruzione veniva eseguita da una rete logica sequenziale. Ciò poteva comportare, con l'aumentare delle operazioni eseguite dal calcolatore, una notevole complessità circuitale.

Per ridurre e semplificare i circuiti, si è osservato che svariate sequenze logiche corrispondono a più istruzioni macchina ripetitive; allora tali sequenze, costituenti dei microprogrammi, vengono registrate in una memoria di sola lettura ROM che possono essere richiamate all'occorrenza nella memoria centrale.

In tal modo, quando nella istruzione macchina serve un microprogramma registrato nella ROM, si associa ad esso un indirizzo che lo richiama, lo estrae dalla ROM stessa e lo trasferisce nella memoria centrale, per esser poi eseguito.

L'insieme dei microprogrammi di cui è dotato un calcolatore viene detto *firmware* del sistema e costituisce l'interfaccia tra il software e l'hardware.

Le istruzioni eseguite da un circuito logico sequenziale costituisce la così detta *Logica cablata*.

Le istruzioni eseguite con microprogrammi registrati nella ROM costituiscono una *Logica programmata*.

Una istruzione in linguaggio macchina viene detta di *Livello 1*. Questa, per essere eseguita, debbono svolgersi, in sequenza, una serie di passi elementari (in logica cablata o programmata), detti di *Livello 0*, che costituiscono le microistruzioni.

1.5.1 UNITÀ DI CONTROLLO

L'Unità di Controllo, per ogni istruzione, genera una sequenza di comandi, che si svolgono per passi successivi o, come si dice, per cicli macchina.

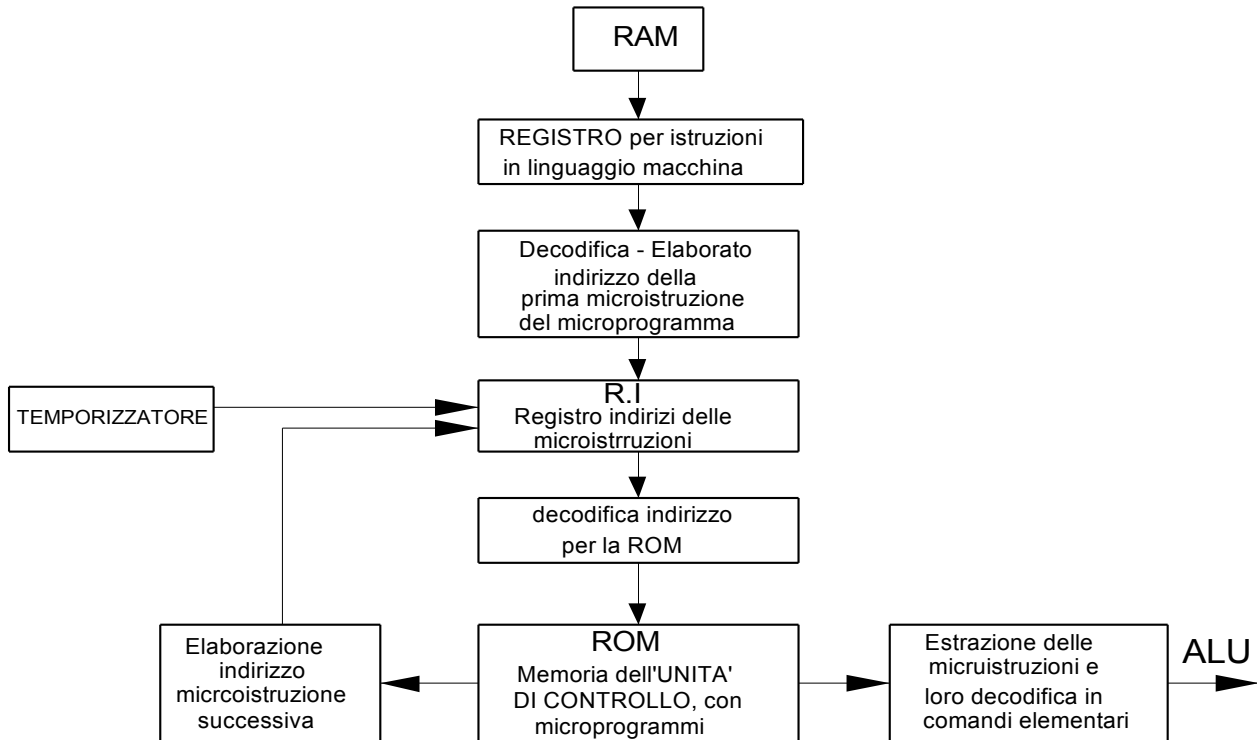
Nell'interno del CPU vi è un orologio, clock, che determina il tempo del ciclo.

Ci si riferisca allo schema a blocchi di fig.

- Dalla memoria principale viene prelevata la istruzione, corrispondente ad una certa operazione, e viene memorizzata in un registro contenente le istruzioni in linguaggio macchina.
- L'istruzione viene prelevata da detto registro e viene interpretata e decodificata, secondo un codice operativo, in modo da elaborare l'indirizzo della prima microistruzione del microprogramma corrispondente all'istruzione impartita: microprogramma che è memorizzato nella memoria di tipo ROM dell'UNITÀ DI CONTROLLO

- L'indirizzo della prima microistruzione, associata al microprogramma da richiamare, decodificato, viene memorizzato nel registro degli indirizzi R.I.

fig1.4



- Il *temporizzatore* successivamente abilita l'uscita del registro *R.I* contenente l'indirizzo della *microistruzione* richiamata, e detto indirizzo, dopo essere decodificato, selezionerà la prima microistruzione del *microprogramma* contenuto nella *ROM*.
- Dalla *ROM* viene così letta la prima microistruzione, che dopo essere decodificata in comando elementare, viene inviata all'*ALU* (Unità Aritmetica Logica).
- Contemporaneamente, in una apposita rete, viene calcolato l'indirizzo della successiva microistruzione del microprogramma e memorizzato nel registro *R.I*.
- Così il nuovo indirizzo, corrispondente alla successiva microistruzione da prelevare nella *ROM*, è pronto nel registro *R.I* per essere inviato in uscita, quando lo comanderà il temporizzatore.
- Per cicli successivi, comandati dal temporizzatore, le microistruzioni vengono compiute secondo un ordine stabilito dagli indirizzi, fino a completare il microprogramma, corrispondente all'istruzione impartita e prelevata dalla memoria principale.

1.6.2 Circuito ID

Per l'esecuzione delle microistruzioni, come si è visto, occorre l'intervento di un particolare circuito detto decodificatore "*ID*" (*Instruction Decoder*). Esso ha la funzione di decodificare le istruzioni, secondo un codice operativo, e inviare gli opportuni segnali alle parti del computer interessate al programma, affinché eseguano le operazioni comandate dalle istruzioni.

1.6.3 REGISTRI

Per effettuare le operazioni, contenute nelle istruzioni componenti un programma, occorrono delle parti circuitali denominate *registri*

I registri servono per la *memorizzazione temporanea* di dati in una certa parte del *CPU*.

Per spiegare lo scopo dei vari registri, si continui osservare lo svolgimento e l'attuazione, delle istruzioni contenute in un programma. Esso è costituito da più microprogrammi contenuti nella ROM e questi, a loro volta, sono composti da più microistruzioni.

Precedentemente si è seguita l'attuazione del primo microprogramma, dalla prima all'ultima microistruzione.

1.6.3.1 Registro RSP

Per l'esecuzione degli altri microprogrammi occorre l'impiego di un registro detto Registro di Stato del Programma "*RSP*" o *PSW* (Program Status Word).

Il registro *RSP* contiene le informazioni necessarie per poter controllare in ogni momento la situazione del programma: Controllare la situazione attuale, le operazioni che sono in esecuzione e confrontarle con le istruzioni di comando, controllare l'esecuzione delle operazioni successive ecc.

Così, prima di avviare il programma già registrato nella *RAM*, viene scritto, nella *RSP*, l'indirizzo del 1° microprogramma e quello del successivo, da effettuarsi ultimato l'esecuzione del precedente

1.6.4 Contatore del programma PC

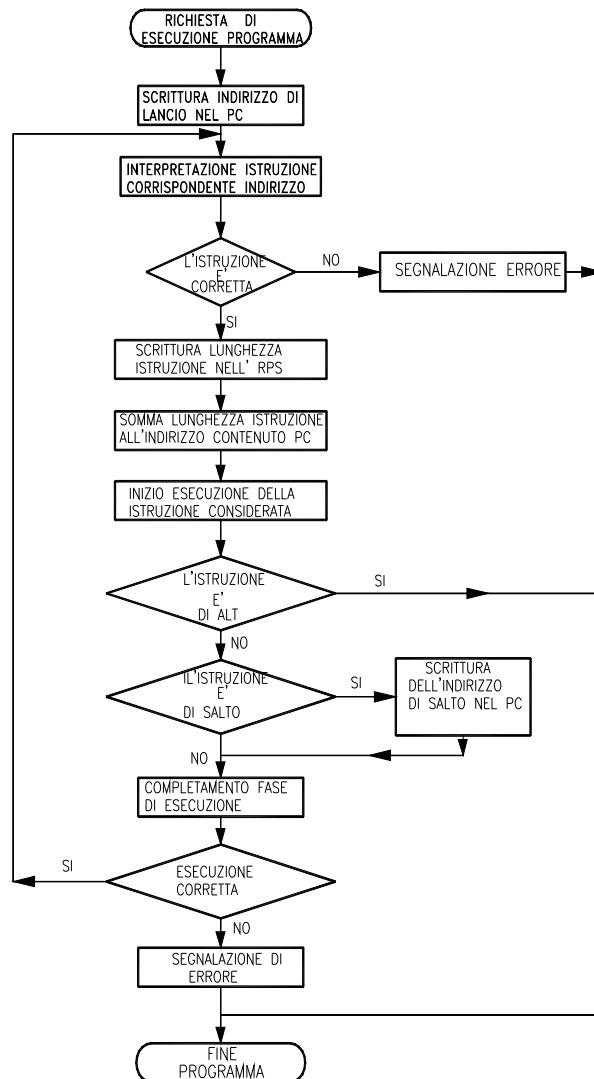
La parte dell'*RSP* ove vengono registrati gli indirizzi costituisce il *PC* (Program Counter), Contatore del Programma.

Nello schema a blocchi di fig. viene riportata la sequenza logica per l'esecuzione di un programma.

Si abbia una richiesta di esecuzione di un programma scritto nella *RAM*. Si possono distinguere i seguenti passi.

- 1 Viene scritto il primo indirizzo di lancio del programma nella zona del registro RSP contenente l'indirizzo dell'istruzione successiva.
- 2 Viene così avviata l'esecuzione del primo microprogramma, registrato nella memoria di controllo, che ha inizio con l'interpretazione dell'istruzione corrispondente all'indirizzo scritto nel *PC*
- 3 Si ha ora un condizionale. Se l'istruzione è stata interpretata in modo corretto si passa al passo successivo, altrimenti si ha una segnalazione di errore e si salta a fine programma
- 4 Se l'istruzione è stata interpretata correttamente, la sua lunghezza (*dipendente dal codice operativo*) viene scritta nell'apposita zona dell'*RSP* e viene a sommarsi all'indirizzo registrato nel *PC*, ottenendo l'indirizzo dell'istruzione successiva
- 5 Inizia l'istruzione considerata
- 6 Si ha un nuovo condizionale:- Se l'istruzione è di *ALT* allora si salta a fine programma, altrimenti si va al passo successivo.
- 7 Si ha un nuovo condizionale: -Se l'istruzione non è di *SALTO* allora si procede nel passo successivo; se invece l'istruzione è di *SALTO* allora viene elaborato nel *PC* un indirizzo che sostituisce il successivo a quello corrispondente all'istruzione in corso. In tal caso, prima dell'attuazione e completamento dell'istruzione in corso, viene effettuato un sotto-programma.
- 8 Viene eseguita l'istruzione in corso fino al suo completamento.

Fig.1.5



- 9 Se l'esecuzione dell'istruzione è corretta allora inizia un nuovo ciclo per effettuare la successiva istruzione del microprogramma corrispondente all'indirizzo elaborato nel PC; se invece l'esecuzione non è corretta, allora si ha la segnalazione di errore e si salta a fine programma.

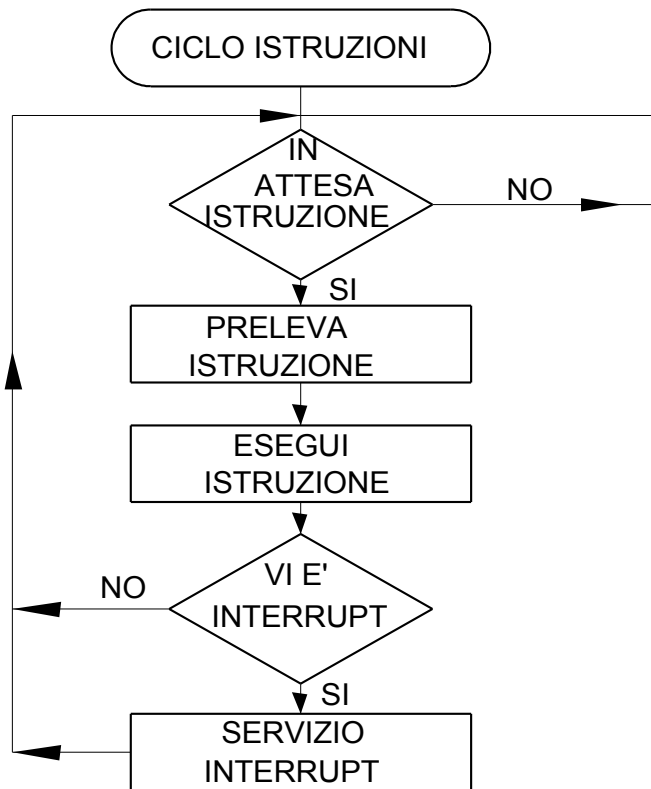
Il ciclo descritto può ripetersi con un'altissima velocità consentita dai circuiti elettronici ed è del tutto automatico.

Come si è evidenziato, in caso di errore si ha un arresto del programma : si dice che il calcolatore si trova in una situazione di *program check* (si è rilevata una situazione anomala)

1.6.5 Segnali di interrupt

L'argomento verrà ripreso nel capitolo dei *Sistemi Operativi* .

fig.1.6



Nell'esecuzione delle istruzioni contenute nei programmi, occorre spesso impiegare dei segnali che interrompano il programma principale per eseguire dei sottoprogrammi detti di servizio. Tali possono essere i segnali che provengono da periferiche, che debbono eseguire particolari operazioni, prima che il programma principale possa proseguire.

Quando arriva un *segnale di interruzione*, questo viene registrato in un apposito registro dell'unità di governo. Tale registro viene esaminato al termine di ogni istruzione in linguaggio macchina. Quando arriva un segnale di interrupt, viene eseguito un particolare programma, chiamato *servizio dell'interrupt*.

Al termine del sottoprogramma di servizio dell'interruzione riprende il programma in corso.

Vi è una *scala di priorità* nelle interruzioni, nel senso che particolari interruzioni importanti hanno una *priorità*

rispetto ad altre: per esempio il segnale di errore nel funzionamento dell'hardware è preminente rispetto a tutti gli altri.

Per il ripristino del programma in corso, dopo una interruzione, è necessario che il contenuto del *Registro Stato del Programma RSP* lo permetta, occorre cioè che:

- 1° Sia stato elaborato l'indirizzo della successiva istruzione
- 2° Siano soddisfatte le condizioni poste dall'ultima istruzione. *Tali condizioni prendono il nome di FLAG.*

Il FLAG è la condizione che l'ultima istruzione pone affinché possa proseguire il programma.

Tali condizioni possono essere di varia natura; per esempio che un risultato sia maggiore o minore di zero, che, in generale, si verifichi un certo evento...

Concludendo un sottoprogramma di servizio delle interruzioni inizia con una memorizzazione, in una zona opportuna della memoria dei registri e termina, dopo aver ripristinato in essi il precedente contenuto.

Da quanto detto si vede l'importanza che giocano i registri per lo svolgimento di un programma.

I principali registri sono:

- 1°- *Registro Contatore del Programma* (Program Counter) PC -Contiene l'indirizzo in linguaggio macchina dell'istruzione successiva a quella in esecuzione.
- 2°- *Registro dell'istruzione* - In esso viene trasferita l'istruzione da eseguire, dopo averla prelevata dalla memoria centrale
- 3°- *Registro delle Interruzioni* -In esso vengono registrate le segnalazioni delle interruzioni
- 4°- *Registro dei Flag* -In esso viene registrato l'effetto della condizione posta dall'ultima istruzione
- 5°- *Registri Indice* - Sono particolari registri che vengono utilizzati per la creazione degli indirizzi di memoria

- 6°- *Registro Puntatore dell'area Stack* -Utilizzato per poter gestire una zona di memoria da riservare alla memorizzazione temporanea di dati

1.6.6 Clock

Come si è detto, la successione delle microistruzioni viene abilitata da un orologio interno "clock", che le fa eseguire in una successione di cicli macchina. Il numero di cicli macchina vengono misurati in MHz=Mcicli/secondo.

Il *clock* è essenziale per lo svolgimento delle operazioni di hardware e di software. Esso è generato da un minuscolo oscillatore al quarzo posto all'interno del CPU con il quale si raggiungono oscillazioni al secondo che superano i 4GHz.

1.7 UNITÀ ARITMETICA LOGICA ALU

La funzione dell'ALU è quella di eseguire le operazioni fondamentali somma, sottrazione, moltiplicazione.

Alcuni calcolatori sono dotati di circuiti che possono eseguire solo somme e sottrazioni ; in tal caso le altre operazioni si ottengono con particolari sottoprogrammi.

Per velocizzare il calcolo, la tendenza è quella di realizzare le funzioni con *microprogrammi registrati* che vanno ad accrescere il *firmware* del sistema.

Le principali parti dell'ALU sono:

- 1°- Rete logica di calcolo -É costituita da circuiti elettronici elementari capaci di eseguire le operazioni logiche *AND OR NOT* e operazioni semplici come l'addizione. I circuiti vengono pilotati dall'Unità di Controllo che comanda l'operazione da eseguire; gli operandi vengono presi dalla memoria centrale *RAM*
- 2°- **Registri detti ACCUMULATORI** -Vengono utilizzati per contenere uno o più dati dell'istruzione che è in corso di esecuzione
- 3°- **Flag di controllo** - Per indicare le particolari condizioni finali.

1.8 SISTEMI OPERATIVI

Come è stato spiegato nei paragrafi precedenti , un'istruzione, impartita alla macchina per effettuare una determinata operazione, deve essere composta da un certo numero di microistruzioni, formulate in codice binario, direttamente comprensibile dalla macchina stessa. Scrivere un programma in questa forma, pur se possibile, è difficoltoso e richiede una approfondita conoscenza del sistema.

Per superare questa obbiettiva difficoltà sono stati elaborati particolari programmi, detti "*Sistemi Operativi*", da introdurre nella memoria centrale del CPU in modo che, con una istruzione impartita con parole generalmente in lingua inglese, o con un clic del mouse su un'icona su un **desktop** viene tradotta in una istruzione macchina, composta da microistruzioni in codice binario, direttamente comprensibili dalla macchina stessa. Inoltre detto programma deve gestire nel modo migliore tutte le risorse del calcolatore.

Un *sistema operativo* costituisce il *software di base*; esso è un insieme di programmi, ciascuno specializzato a svolgere una funzione specifica.

Nei primi sistemi operativi le istruzioni erano impartite solamente attraverso la tastiera: i computer non erano dotati del mouse.

Nei sistemi **MS-DOS**, **LINUX** tutte le risorse del sistema (contenuto delle memorie di massa e periferiche) venivano viste in termini di *file*, e l'interazione utente-macchina virtuale avveniva attraverso **comandi** sui file, inviati da tastiera.

Nei sistemi operativi WINDOWS lo schermo del computer compare come il piano di una scrivania ove sono appoggiate delle **icone**, queste sono dei **contenitori di risorse**. Selezionando con il mouse o con tastiera un'icona viene attivata la risorsa ad essa associata

In un sistema operativo si possono distinguere i seguenti compiti fondamentali:

- 1- *Gestione delle risorse*
- 2- *Supervisione del sistema*
- 3- *Gestione dei dati*

1.7.1 Gestione delle risorse

Nei paragrafi precedenti si sono esaminati, nelle linee generali, il funzionamento e la struttura del calcolatore. Nei moderni calcolatori, oltre al processore principale CPU, vi sono altri processori secondari *dedicati* a funzioni particolari.

Le comunicazioni tramite *canale* sono gestite da un processore, così un processore provvede alla visualizzazione dei caratteri sullo schermo, un altro processore può consentire sofisticate prestazioni di stampa...

È compito però del processore principale impartire le istruzioni ai processori dedicati, i quali lavorano in modo autonomo dalla CPU.

Occorre quindi l'utilizzo di particolari segnali, detti segnali di interruzione "*Interrupt*" per la comunicazione tra il processore principale e i secondari

Chiamiamo *PROCESSO* l'esecuzione di una sequenza di istruzioni impartite da un processore.

Nell'eseguire un processo può avvenire che esso debba essere temporaneamente interrotto per far eseguire una particolare *routine di servizio* (es. introduzione dati da periferica); terminata la routine, il processo potrà continuare. L'interruzione e il proseguimento del processo saranno comandati da segnali di *interrupt*.

Così, per esempio, un segnale di interrupt interrompe un processo in presenza di un errore

Segnali di interruzione vengono attivati per:

-Errore macchina

Dovuto al cattivo funzionamento della macchina a livello circuitale

-Errore di programma

Errore effettuato durante l'esecuzione di un programma

-Errore di indirizzamento memoria

Può dipendere dalla chiamata di indirizzi inesistenti o da indirizzi protetti

-Interruzione di tipo esterno

L'interruzione è provocata da comandi impartiti dalla consolle o da dispositivi a tempo, *timer* esterni.

-Interruzione da periferiche

Interruzioni che si hanno durante le operazioni input/output I/O

-Interruzione da programma

Interruzioni programmate per il verificarsi di particolari condizioni di relazioni (per esempio booleane)

Per la corretta gestione dei processi e tener sotto controllo tutti gli eventi che possono verificarsi vengono assegnati *livelli di priorità* ai segnali di *interrupt*; così un segnale di errore per cattivo funzionamento dell'hardware ha un livello di priorità superiore a quello di interruzione da programma.

Affinché si attivi (*venga servito*) un segnale di interrupt esso deve avere un livello di priorità maggiore di quello del processo in corso.

Così pure nell'esecuzione dei vari programmi, specialmente in ambiente di multiprogrammazione, vengono stabiliti livelli di priorità che verranno serviti secondo un ordine gerarchico.

Per la corretta gestione del sistema occorre, come si è detto, un *clock* temporizzatore del ciclo macchina; oltre a questo sono presenti temporizzatori sia di tipo hardware che di tipo software che producono segnali di interrupt. Nei moderni calcolatori vi è un timer esterno, indipendente dal CPU, che funziona anche a macchina spenta: esso garantisce una più perfetta gestione del sistema.

In ambiente di multiprogrammazione, in calcolatori più sofisticati, possono essere svolti parallelamente più processi e questi vengono controllati dal *S.O.*

1.7.2 Supervisione del sistema

Il *S.O.* deve controllare in ogni istante il corretto funzionamento di tutte le componenti, sia hardware sia software, programmi di sistema o applicativi (software utente).

Il *S.O.* controlla che non siano richieste operazioni impossibili, per errore di programma o per dati erroneamente immessi; controlla che non si introducano dati in zone di memoria riservate ad altri compiti.

In caso di errore deve emettere una segnalazione che lo individui

1.7.2.1 Registri di stato

Per avere la completa supervisione e la ininterrotta informazione dello stato e della situazione del programma in corso vi è il *Registro di Stato*; esso contiene l'informazione sullo stato dei segnali di interrupt, sulla situazione delle memorie. Tutte queste informazioni sono necessarie per il buon funzionamento e svolgimento di un programma.

Così, nel caso di interruzione di un processo, per una routine di servizio interessante una periferica, esso potrà riprendere il suo svolgimento quando, nel *Registro di Stato*, vi sarà l'informazione che tutte le periferiche interessate si ritrovino nello stato nel quale erano al momento dell'interruzione; prima cioè che l'interruzione abbia modificato lo stato di una di esse.

1.7.3 Gestione dei dati

Il Sistema Operativo sovrintende alla memorizzazione alla ricerca dei dati e programmi nelle memorie di massa. Esso si occupa della gestione dei dischi, della ricerca degli spazi liberi su di essi per la memorizzazione dei dati, per poi, successivamente, trovare i punti ove quei dati sono stati memorizzati, estrarli e portarli nella memoria centrale.

Proprio per la proprietà del *S.O.* di gestire i dischi, ad alcuni sistemi è stato dato il nome di *DOS -Disk Operating System*

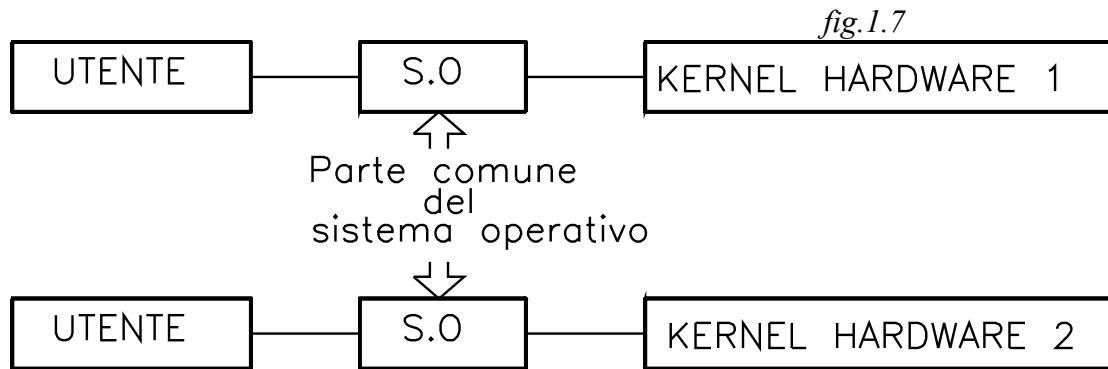
1.7.4 Struttura di un Sistema Operativo

Un sistema operativo è composto da due parti essenziali.

Kernel- È quella parte interna del *S.O.* che conosce la macchina (l'hardware) e tutta la periferia ad essa collegata. In questa parte si trovano i programmi che permettono ad altri più generali di impartire istruzioni, utilizzare il disco, scrivere a terminale video ecc.

Il *Kernel* è differente da macchina a macchina, è la parte che si interfaccia tra l'hardware e il resto del programma del *S.O.* è un programma sempre in esecuzione, gli altri sono programmi applicativi

Parte comune S.O Il restante programma, oltre il *Kernel*, che compone il *S.O.* è la parte comune che verrà caricata su ogni macchina, anche con hardware diverso; essa comunica con l'utente e il *Kernel* che conosce il tipo particolare di hardware.



Nella fig. è rappresentato il sistema operativo su due macchine con hardware diverso.

I programmi che compongono il sistema operativo, come si è detto, sono scritti in *linguaggio macchina*.

A livello di linguaggio macchina si possono distinguere due stati:

Stato sistema Vengono eseguite istruzioni privilegiate riguardanti routines del sistema.

Stato programma utente Il calcolatore è in grado di eseguire istruzioni normali, dando la precedenza a quelle di sistema.

I programmi del Sistema Operativo sono disposti su tre livelli:

- Interfaccia utente*
- File manager*
- Device driver*

1.74.1 **Interfaccia utente**

È la parte del programma che viene utilizzata dal CPU per poter *interpretare* le istruzioni, i dati immessi dall'operatore, attraverso le periferiche di input.

Le istruzioni e i dati di input passano, poi, all'elaborazione di un secondo livello di programmi detto *File Manager*.

1.7.4.2 **File Manager**

Si incarica *dell'organizzazione dei file* provenienti dall'interfaccia utente.

È un programma il quale, ricevuti i comandi, individua il dispositivo adatto alla loro corretta esecuzione e li invia al *Device drive* che si incarica di eseguirli.

1.7.4.3 **Device Drive**

È la guida dei dispositivi.

È il programma che si occupa dell'esecuzione delle operazioni di basso livello internamente all'hardware: ad esempio la memorizzazione di un byte in un punto specifico della memoria di massa ecc.

Per ogni dispositivo sia interno che esterno esiste un *Device Drive*.

1.8 UNIPROGRAMMAZIONE MULTIPROGRAMMAZIONE

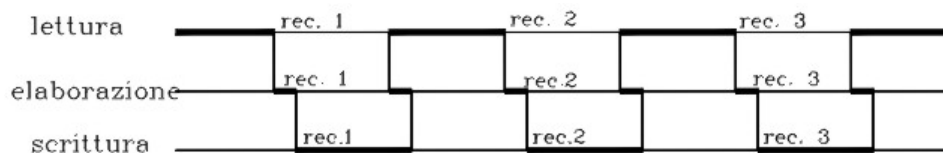
Riguardo ai programmi che possono essere elaborati, i sistemi operativi si distinguono in:

- Uniprogrammazione* Con tale sistema si può eseguire un programma alla volta. gli utenti possono lavorare uno alla volta.
- Multiprogrammazione* In memoria possono essere presenti più programmi i quali vengono eseguiti con sfalsamento dei tempi di esecuzione, in modo da sfruttare tutti i tempi di pausa.

1.8.1 ELABORAZIONE IN UNIPROGRAMMAZIONE

In tale elaborazione non viene sfruttato nel modo migliore l'alta velocità di elaborazione della *CPU*, in quanto i record che provengono o che vengono inviati alle periferiche hanno un tempo di attesa molto lungo, che potrebbe essere sfruttato in ambiente di *multiprogrammazione*.

fig.1.8



Così, dalla figura si può notare che un record impiega un tempo elevato per essere letto da console e stampato rispetto a quello impiegato per l'elaborazione. Questi tempi elevati potranno essere meglio impiegati in ambiente di multiprogrammazione.



[Clic per continuare](#)

Avanti...



[Clic per la pagina iniziale](#)

Indietro...