

Concorsi **ENTI**
LOCALI

Capitolo 1 | CONOSCENZE SISTEMICHE DI BASE, RETI LOCALI E GEOGRAFICHE, RETE FONIA E DATI

SOMMARIO: 1. Architettura dei calcolatori elettronici. – 2. Il sistema di elaborazione. – 3. Classificazione degli Elaboratori Elettronici. – 4. Architettura interna di un elaboratore. – 5. L'unità centrale di elaborazione. – 6. Assembler. – 7. La memoria centrale. – 8. Le memorie di massa. – 9. Le tipologie di collegamento con le periferiche. – 10. Periferiche di Input e di Output. – 11. Architettura dei sistemi distribuiti. – 12. Elaborazione distribuita. – 13. Sistemi operativi per macchine MIMD. – 14. Tipi di rete. – 15. Reti LAN. – 16. Reti MAN. – 17. Reti WAN.

1. Architettura dei calcolatori elettronici

In informatica il termine *architettura* indica l'organizzazione logica dei componenti interni di un computer macchina e il modo in cui questi componenti cooperano armonicamente per eseguire operazioni più o meno complesse. Pertanto, l'architettura di un computer viene dedotta dalla conoscenza della struttura interna e dal modo in cui è possibile far cooperare le unità fisiche tramite un insieme di comandi o istruzioni, che definiscono il linguaggio macchina.

2. Il sistema di elaborazione

Il computer viene definito come un insieme di dispositivi di diversa natura: elettrici, meccanici ed ottici in grado di acquisire dall'esterno dati e programmi e di produrre in uscita il risultato dell'elaborazione. L'insieme organizzato di risorse diverse va sotto il nome di **Sistema di Elaborazione**. La struttura fisica, interna e esterna del computer prende il nome di hardware (hard = duro, ware = componente). L'insieme di tutti i programmi che permettono di utilizzare l'elaboratore si chiama software (soft = soffice).

3. Classificazione degli Elaboratori Elettronici

Gli elaboratori elettronici o computer si possono classificare in cinque grandi categorie:

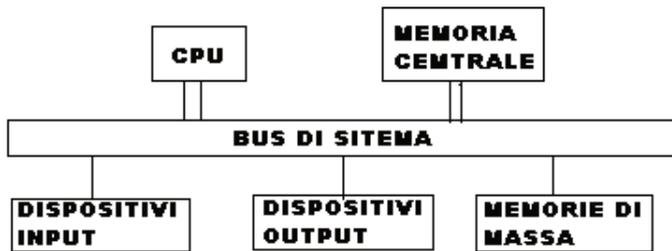
- **I supercomputer** sono i più potenti, i più veloci e i più costosi. Sono utilizzati principalmente nelle università e nei centri di ricerca.
- **I mainframe** hanno processori potenti e grande quantità di memoria RAM. Sono particolarmente utilizzati in multiutenza, ossia da più persone contemporaneamente, ciascuna delle quali utilizza un terminale collegato al mainframe. Sono

molto costosi, pertanto sono utilizzati da grosse società commerciali, banche, ministeri, aeroporti.

- **I minicomputer** sono elaboratori un po' più piccoli ma in grado di gestire grandi quantità di dati in multiutenza. Il loro costo è dell'ordine di decine di milioni e sono usati da società di medie dimensioni.
- **I personal computer** sono quelli usati per lavoro d'ufficio o in ambito domestico da un solo utente per volta. Una ulteriore distinzione può essere fatta tra computer da tavolo (desktop computer) e portatili (notebook o palmtop).
- **Terminale stupido** (senza nessuna capacità di calcolo) oppure **intelligente** (svolge localmente una certa quantità di funzioni in modo da alleggerire il computer principale).

4. Architettura interna di un elaboratore

I moderni elaboratori elettronici, dai computer di casa ai grandi mainframe derivano tutti da una macchina fondamentale chiamata di Von Neumann, che può essere schematizzata come un insieme di blocchi funzionali ed interconnessi secondo lo schema seguente:



Il blocco denominato Input serve ad immettere i dati iniziali ed il programma in Memoria Centrale, la CPU (o Central Processing Unit) è il blocco funzionale in cui vengono eseguite le elaborazioni. La Memoria è un dispositivo in grado di memorizzare una grandissima quantità di informazioni, dell'ordine delle centinaia di migliaia o dei milioni di caratteri ed il blocco Output è il dispositivo mediante il quale sono resi disponibili i dati in uscita, per esempio su schermo video.

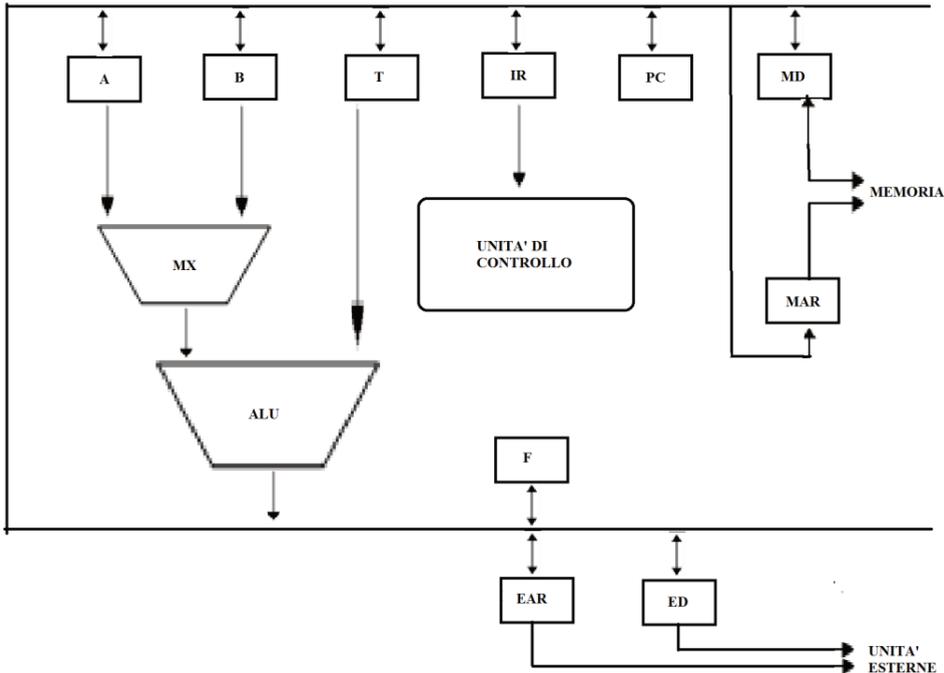
Le informazioni caricate nella memoria sono dunque dati ed istruzioni, mentre la sequenza di istruzioni caricate in memoria costituisce il programma, che è la codifica dell'algoritmo di calcolo che la macchina deve eseguire. La codifica di un algoritmo consiste nella traduzione di istruzioni logico matematiche, di cui esso è costituito, in una sequenza di istruzioni che la CPU è in grado di eseguire.

5. L'unità centrale di elaborazione

L'**unità centrale** di elaborazione, detta **CPU**, (*central processing unit*) è il '*cervello*' del computer, in quanto è responsabile dell'esecuzione dei programmi e del controllo di tutto

ciò che avviene all'interno dell'elaboratore. Esegue operazioni logiche aritmetiche e di trasferimento dei dati. Esse è realizzato su un CHIP di silicio (tecnologia di integrazione su larga scala, cioè circuito elettronico su una superficie molto ridotta).

La CPU è generalmente considerata scomposta in una unità di calcolo e una unità di controllo (CU). L'unità di calcolo è costituita da un insieme di registri e una unità logico-aritmetica, chiamata ALU (o *Arithmetic and Logic Unit*), interconnessi tra loro mediante linee di comunicazione chiamate Bus di Sistema. Lo schema di una CPU elementare è riportato dallo schema seguente.



I registri A, B, T contengono dati su cui è possibile far eseguire l'operazione elementare da parte dell'ALU. Il registro IR contiene l'informazione relativa alla operazione da eseguire ed è perciò chiamato registro dell'istruzione corrente. Il registro PC, chiamato contatore di programma, fornisce l'indirizzo di memoria in cui è contenuta l'istruzione successiva del programma, pertanto, in ogni istante si ha che il registro IR contiene l'istruzione in corso di esecuzione, mentre il registro PC punta alla locazione di memoria in cui è contenuta l'istruzione successiva a quella in esecuzione. Il registro MD serve a scambiare dati tra la CPU e la memoria, mentre il registro ED è utilizzato per scambiare dati tra la CPU e le unità esterne di Input e di Output. Il registro MAR è caricato di volta in volta con l'indirizzo della locazione di memoria che in un certo istante la CPU vuole leggere o scrivere, mentre il registro EAR è caricato con il numero di porta di I/O con cui scambiare dati.

Il funzionamento di questa CPU è abbastanza semplice. Sono prima caricate le istruzioni del programma da eseguire su indirizzi di memoria consecutivi: Si suppone per

semplicità che ogni istruzione è contenuta in una sola locazione di memoria anche se in molte macchine reali vi sono istruzioni che occupano una, due o più locazioni di memoria a seconda della complessità dell'istruzione stessa. Inizialmente il registro PC è caricato con l'indirizzo della prima istruzione del programma. Si può ancora supporre che il registro PC contenga il valore 0 e che il programma sia in memoria a partire dalla prima locazione di memoria, di indirizzo 0. L'unità di controllo esegue la sequenza di operazioni che riguardano il prelievo della istruzione (*fase fetch*), la decodifica e l'esecuzione della istruzione (*fase di esecuzione*). I registri e la memoria contengono informazioni di tipo binario (stringhe di bit) che rappresentano istruzioni, indirizzi o dati. Una informazione contenuta nel registro PC rappresenta un indirizzo, in IR una istruzione, in A, B, T un dato su cui è possibile effettuare delle operazioni o delle modifiche interne per mezzo della ALU.

La sequenza di operazioni eseguite costituisce il cosiddetto *ciclo di istruzione* di una CPU. Esso è a sua volta suddiviso in uno o più cicli macchina. Il *ciclo macchina* è definito come sequenza di operazioni elementari che l'unità di controllo esegue ogni volta che accede alla memoria o ad unità esterne di I/O. La CPU esegue continuamente il ciclo di operazioni fino a che non carica in IR ed esegue una particolare istruzione di *stop* o *di halt*. Per descrivere nei vari passi la fase di esecuzione della generica istruzione, occorre specificare il tipo di istruzioni che la macchina è in grado di eseguire. Le istruzioni possono essere raggruppate in classi in base al tipo di operazione che eseguono:

- istruzioni per lo spostamento dati;
- istruzioni aritmetiche su numeri interi;
- istruzioni di tipo logico;
- istruzioni di rotazione e shift;
- istruzioni di controllo programma;
- istruzioni di controllo macchina.

L'insieme precedente è il numero indispensabile di istruzioni che una CPU deve eseguire per essere efficiente. Le istruzioni, come detto in precedenza, sono codificate con una stringa di bit ed interpretate come tali una volta caricate nel registro IR.

Il lavoro della CPU viene scandito dagli impulsi di un orologio chiamato **Clock**, che compie milioni di battiti al secondo. La velocità di funzionamento si misura in Hz (hertz), infatti 1 Hz corrisponde ad 1 ciclo di Clock al secondo, mentre se passiamo ai multipli risulta che 1 MHz (megahertz) corrisponde ad 1 milione di battiti al secondo e 1GHz (gigahertz) risulta 1 miliardo di battiti al secondo.

La memoria di lavoro o memoria centrale della CPU è strutturata come successione di celle di lunghezza standard pari a otto bit (1Byte). Le dimensioni massime che la memoria di un calcolatore può raggiungere dipendono in maniera diretta dalla lunghezza in bit dei registri della macchina. Il registro MAR ed il registro PC definiscono lo "*spazio di indirizzamento*" o capacità massima di un calcolatore. Si definisce invece grado di parallelismo, il numero massimo il numero di byte che possono essere letti o scritti in un ciclo macchina, per esempio una macchina con parallelismo uno e registri interni di 16 bit richiede due cicli macchina per caricare o scaricare un dato dalla memoria o che già si trova in memoria.

Nelle moderne CPU il numero di registri interni della macchina è di gran lunga maggiore, questi registri aggiuntivi hanno funzione generale se possono contenere dati o

indirizzi e si chiamano “registri generali”, oppure hanno funzione particolare se contengono una informazione specifica, nel qual caso si chiamano registri speciali. I vantaggi derivanti dall’impiego di registri generali e speciali sono molteplici; i più evidenti sono relativi ai modi più flessibili di indirizzamento degli operandi e alla possibilità di mantenere nella CPU i dati di accesso più frequente.

I registri generali di una macchina a registri possono essere usati come:

- *accumulatori*: il dato da elaborare è in un registro;
- *puntatori*: il dato è in memoria e il suo indirizzo in un registro;
- *registri indice*: il dato è in memoria e il registro contribuisce a definire il suo indirizzo.

I registri speciali sono:

- il contatore di programma (PC);
- gli “stack pointers” o puntatori di pila (SP);
- il registro di stato della CPU (PSW).

In questo elenco non sono considerati il registro IR e i registri di interfaccia con la memoria e le unità esterne in quanto sono trasparenti al programmatore. Il modo in cui sono utilizzati i registri generali e i registri speciali riguarda in particolare i modi di indirizzamento della memoria e le varianti che possono assumere le istruzioni di ogni classe.

6. Assembler

Le istruzioni macchina ed i relativi programmi scritti in linguaggio macchina usano normalmente una rappresentazione simbolica che utilizza codici formate da parole di più facile interpretazione rispetto alla rappresentazione binaria, questa rappresentazione simbolica è chiamata “*linguaggio assembler*” ed il programma che traduce da questo linguaggio a linguaggio macchina si chiama invece “*assemblatore*”.

Nel linguaggio macchina le istruzioni sono stringhe di bit, mentre nel *linguaggio assembler* si utilizzano codici simbolici e formati speciali per definire il modo di indirizzamento degli operandi. Gli assembler sono speciali programmi che servono a tradurre un programma scritto in assembler in programma codificato in linguaggio macchina; nella maggioranza dei casi il programma tradotto, detto programma oggetto, necessita di un ulteriore programma, detto caricatore (*loader*) per essere caricato nella memoria centrale del calcolatore per essere eseguito.

La traduzione da linguaggio assembler a linguaggio macchina può essere fatta anche su una macchina diversa da quella su cui il programma deve essere eseguito nel qual caso si parla di *cross assemblaggio* ed il programma di traduzione è chiamato *cross assembler*. Un linguaggio *assembler*, a differenza dei linguaggi ad alto livello (Pascal, C, Java, V Basic), è strettamente vincolato alla macchina per il quale è stato progettato, pertanto ogni macchina, o famiglia di macchine, possiederà il proprio linguaggio assembler. Inoltre, per la stessa macchina sono possibili più di un linguaggio assembler e questo è particolarmente vero per il microprocessore per il quale esiste l’assembler definito dal costruttore del chip ma possono esserci altri assembler prodotti da altre ditte (*Software Houses*). Tutti i linguaggi assembler sono strutturati a linea ed ogni linea è formata da quattro campi, nel modo seguente:

```
<ETICHETTA> <OPCODE> <OPERANDI> <COMMENTI>
```

I campi ETICHETTA e COMMENTI sono facoltativi, gli altri due sono obbligatori. Il campo OPERANDI può essere formato da nessuno o più operandi a seconda del codice operativo. L'etichetta è una stringa di caratteri che rappresenta in modo simbolico un indirizzo di memoria, tranne in un caso particolare che sarà illustrato in seguito. Le regole per scrivere una linea in assembler sono diverse a seconda che il particolare assembler richieda un formato fisso o variabile. Nel primo caso si ha che i campi sono separati da almeno uno spazio e l'etichetta, se presente, deve iniziare nella prima colonna. Se manca, la linea deve iniziare con almeno uno spazio e gli operandi multipli sono separati da virgola o altro carattere di interpunzione. Inoltre, non sono ammessi spazi ed una linea che inizia con * è una linea di solo commento.

Negli assembler con linea a formato libero valgono le seguenti regole l'etichetta consiste di una stringa di caratteri seguita dal carattere di interpunzione (due punti), ed i campi OPCODE e OPERANDI devono essere separati da almeno uno spazio, fra gli altri campi lo spazio è inutile, oltretutto il campo commento inizia con un carattere speciale, generalmente (punto e virgola). La stringa di caratteri nel campo etichetta definisce un indirizzo simbolico della linea assembler a cui si riferisce; per esempio, la linea: START MOV PPO, A sta ad indicare che l'istruzione MOV PIPPO, A si trova all'indirizzo START.

Un programma scritto in assembler contiene sia istruzioni macchina espresse in modo simbolico, sia pseudo-istruzioni o direttive all'assemblatore che servono per indicare a quest'ultimo le modalità di assemblaggio. L'etichetta molte volte invece di rappresentare un indirizzo indica generalmente un dato; per esempio, la pseudoistruzione NMAX EQU 100 sta ad indicare che la costante NMAX è stata fissata pari a 100.

Una espressione può trovarsi sia nelle direttive assembler sia nel campo operandi di una istruzione vera e propria e può essere un valore costante, una etichetta definita altrove oppure costanti e/o etichette in espressioni aritmetiche.

“Gli assembler più semplici ammettono solo gli operatori * e – in una espressione; quelli più sofisticati accettano anche gli operatori * e +. Va tuttavia notato che l'espressione aritmetica è valutata dall'assemblatore senza regole di precedenza tra gli operatori per cui l'espressione $3 + \alpha * \beta$ (α e β sono etichette) è valutata come se fosse scritta $(3 + \alpha) * \beta$.

7. La memoria centrale

Abbiamo detto che la memoria è un dispositivo indispensabile al funzionamento della CPU e deve contenere i dati e le informazioni del programma da eseguire. Nei moderni calcolatori, dal pc al mainframe, la quantità di informazioni che deve avere in memoria è estremamente alta, che non può essere contenuta nella sola memoria centrale per due motivi fondamentali, in primis la memoria centrale, realizzata a semiconduttori, è una memoria cosiddetta volatile perché il suo contenuto viene completamente perduto non appena manca l'alimentazione, e poi la memoria centrale è un organo piuttosto costoso per cui non conviene costruire calcolatori con memoria centrale troppo grande ameno che non siano richieste determinate prestazioni. L'architettura della macchina prevede quindi una gerarchia di memorie che nelle soluzioni attuali è costituita da memoria centrale, memoria di massa e memorie esterne.