

# Architettura dei computer

## Prerequisiti specifici

### CONOSCENZE

- ◆ Capire perché è importante conoscere i componenti fisici

### di un computer

- ◆ Comprendere le peculiarità dei vari componenti del computer

## Obiettivi specifici

### CONOSCENZE

- ◆ Conoscere la struttura logico-funzionale di un computer
- ◆ Conoscere le caratteristiche di ogni componente di un computer, le relative funzioni e il loro rapporto
- ◆ Conoscere e utilizzare correttamente i termini tecnici

### relativi ai componenti architeturali dell'“automa computer”

### ABILITÀ

- ◆ Saper utilizzare le varie periferiche di I/O
- ◆ Saper descrivere le fasi salienti dell'esecuzione di un'istruzione macchina

## 1 L'informatica

Il termine **informatica** proviene dal francese *informatique*, derivato dall'espressione *information automatique* (informazione automatica), ed è stato introdotto per indicare:

“ la **scienza** che si occupa dei processi e delle tecnologie che consentono il trattamento (creazione, raccolta, elaborazione, memorizzazione e comunicazione) automatico e razionale delle informazioni, nonché della progettazione degli strumenti che concretizzano tali funzioni (computer). ”

Elencare tutte le possibili applicazioni dell'informatica è difficile, se non addirittura impossibile, perché l'uso del computer è ormai diffuso in ogni ambito delle attività umane. Risulta infatti sempre meno facile immaginare settori in cui non sia prevedibile la sua introduzione.

È tuttavia interessante notare che, in un primo momento, i computer sono stati impiegati prevalentemente per l'esecuzione di *calcoli matematici* e per lo svolgimento di elaborazioni relativamente semplici.

Si può oggi affermare che, con il loro presente e continuo inserimento, i computer modificano l'ordinamento della società.

Da un lato, infatti, sostituiscono l'uomo nell'esecuzione dei lavori ripetitivi e manuali, eseguendo le operazioni programmate dall'uomo stesso e lasciandogli soltanto i compiti di controllo; dall'altro, i computer ampliano i confini della conoscenza e le possibilità dell'uomo, perché, grazie al loro impiego, sono realizzabili applicazioni in settori altrimenti inesplorabili.

## 2 Il computer

Prepariamoci a compiere un viaggio interessante: *l'esplorazione del computer*. Vedremo come è composto e cosa fanno i programmi una volta caricati su questa macchina misteriosa. Prima di procedere, cerchiamo di capire cos'è un computer.

“ Il **computer** o elaboratore è una macchina costituita da dispositivi di diversa natura (meccanici, elettrici, ottici ecc.) in grado di elaborare dati in modo automatico, veloce, sicuro ed efficiente. ”

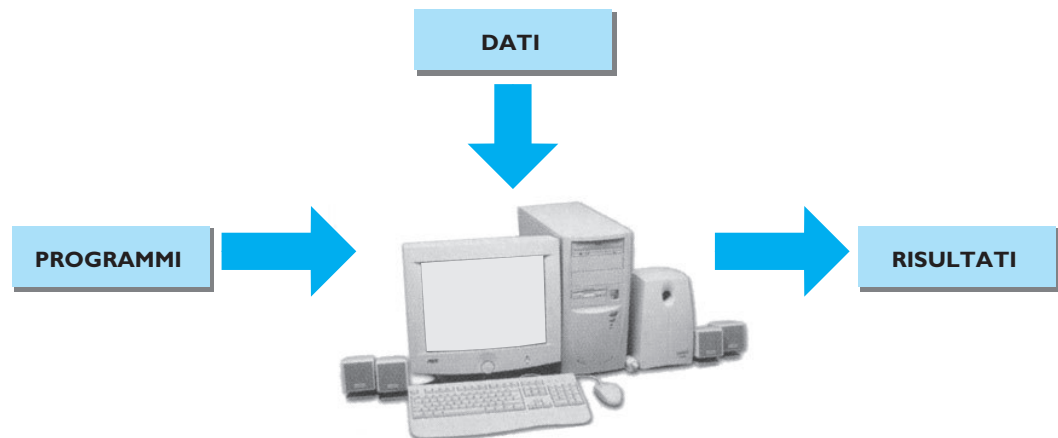
Le sue caratteristiche essenziali sono la *rapidità* e l'*affidabilità*. Chiariamo alcuni termini usati nella definizione. Abbiamo detto che il computer:

- è una **macchina**, cioè un dispositivo privo di intelligenza autonoma, come l'automobile;
- è **elettronico**, cioè il suo funzionamento si basa su componenti di tipo elettronico;
- è un **elaboratore di dati**, cioè, eseguendo le istruzioni di un programma, è capace di ricevere dati dall'esterno, operare su di essi e fornire i risultati dell'elaborazione, sollevando così l'uomo da compiti noiosi, ripetitivi e complessi.

Se vogliamo quindi che il computer *risolva un problema*, dobbiamo fornirgli i dati su cui lavorare e una sequenza ordinata d'istruzioni (cioè il programma) che elabori i dati ricevuti e fornisca i risultati (Fig. 1). Ne consegue che:

“ il **computer** è una macchina programmabile, cioè una macchina che può essere utilizzata per problemi diversi, in grado di interpretare ed eseguire una serie di ordini impartiti dall'esterno. ”

Figura 1  
Configurazione base  
per un computer



Non dimentichiamo che il *computer* è una *macchina*. L'elaborazione dei dati avviene eseguendo le istruzioni contenute nei programmi, attraverso i suoi componenti fisici. Parleremo, quindi, di **sistema di elaborazione** per sottolineare il fatto che l'elaborazione avviene grazie all'*interazione di risorse diverse*.

In questo corso di studi ci occuperemo del **personal computer** (o semplicemente **PC**) ossia di un particolare tipo di macchina, molto utilizzata in ambito lavorativo e domestico.

## 3 Hardware e software

Un computer svolge i suoi compiti utilizzando le proprie **risorse** che possono essere raggruppate in due grandi categorie: **hardware** e **software**.

- Le **risorse hardware** sono la parte fisica del computer, cioè l'insieme delle componenti meccaniche, elettriche, elettroniche, magnetiche e ottiche. Il termine inglese *hardware*, che letteralmente vuol dire "ferramenta", è composto da *hard* che significa "duro" e *ware* che significa "componente".
- Le **risorse software** sono, invece, la parte logica del computer, cioè l'insieme di tutti i programmi che ci consentono di gestire il sistema. Il termine inglese *software* è composto da *soft* che significa "soffice" e *ware* che, come sappiamo, significa "componente".

Esistono anche dispositivi che sono sia hardware che software. Questi dispositivi costituiscono il **firmware**, che non possiamo definire una vera e propria categoria, bensì qualcosa di intermedio fra hardware e software. Possiamo dire che il firmware è costituito da una serie di *istruzioni software memorizzate (cablate) nell'hardware e non modificabili dall'utente*. Esempio tipico sono le **memorie ROM**, di cui parleremo nei paragrafi successivi.

## 4 Il computer tratta dati o informazioni?

I **concetti** di **dato** e di **informazione** sono da alcuni considerati diversi, da altri sovrapponibili. Per una nostra scelta didattica, li considereremo diversi, in modo da rendere più semplice l'approccio all'argomento.

Analizziamo dunque queste differenze, cominciando con un esempio.

### ESEMPIO

Supponiamo che un insegnante, sfogliando il suo registro personale, trovi un foglio sul quale è riportato il simbolo 8. Le domande che si può porre sono le seguenti: "Che cosa volevo indicare con questa cifra? Che cosa rappresenta? Può rappresentare un voto? Oppure il numero di compiti che mi restano da correggere? O forse il numero delle interrogazioni?". Le risposte possibili sono infinite. Ma attenzione: se l'insegnante avesse trovato la cifra 8 nella griglia di valutazione dello studente Rossi, non avrebbe esitato a pensare: "Rappresenta il voto assegnato allo studente Rossi".

Da questo esempio possiamo affermare che la **cifra 8** rappresenta il **dato** e tutte le varie ipotesi formulate dall'insegnante costituiscono le possibili interpretazioni relative a quel dato. Facciamo un altro esempio.

### ESEMPIO

Supponiamo che, dopo una rapina in una gioielleria, un testimone vada dalla polizia e dica: "Ho visto un rapinatore salire su una automobile targata XXZZ". Questo dato può costituire un'informazione se non si è già trovata l'automobile utilizzata nella rapina. Consideriamo, invece, che lo stesso testimone vada dalla polizia e affermi: "XXZZ". In questo caso il poliziotto potrebbe chiedere: "Cosa vuoi? Hai qualche problema? Cosa vuoi dire con XXZZ?" (Fig. 2).

È chiaro che un dato da solo non genera informazione se non è evidente il suo significato.

“ Il **dato**, quindi, è una *conoscenza elementare* che, presa individualmente e fuori da un preciso contesto, non ha alcun valore. ”

“ L'**informazione**, invece, è il *dato elaborato*, ossia l'incremento di conoscenza che deriva dall'*interpretazione di un dato*. ”

Secondo questa distinzione, un dato diviene informazione solo quando viene esattamente interpretato. Conseguentemente, per essere ritenuto un'informazione, un dato deve essere arricchito da indicazioni ausiliarie che permettano di attribuirgli un'interpretazione. Tornando all'esempio, il numero 8 non rappresenta per l'insegnante un'informazione importante. Può diventarlo soltanto attribuendo al numero un significato, utilizzando una precisa **chiave di interpretazione**, così come accade per il poliziotto con le lettere XXZZ.

Figura 2  
Dal dato  
all'informazione



Il computer non interpreta i dati, non attribuisce loro significati precisi, come invece fa la mente umana: per una macchina elettronica sono soltanto semplici simboli! Per questo motivo possiamo affermare che:

“ il computer tratta i dati, l'uomo tratta le informazioni. ”

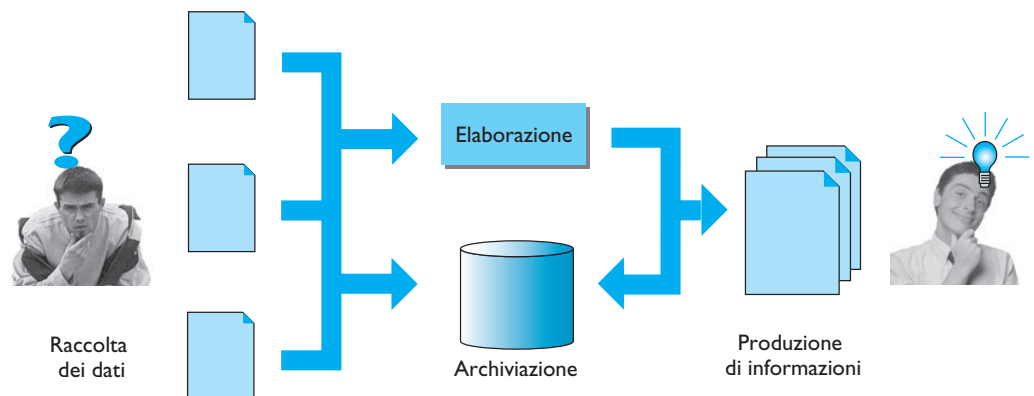
I dati si presentano alla nostra osservazione in varie forme:

- **numeri** (il peso di una persona, il voto di un compito in classe);
- **caratteri alfabetici** (il titolo di un film, il nome di un cane);
- **caratteri alfanumerici** (il numero di una targa, un codice fiscale, una parola chiave per accedere a un videogioco);
- **immagini** (una foto, un manifesto);
- **grafici** (un istogramma, una piantina topografica, un elettrocardiogramma);
- **suoni** (una sirena, l'allarme di un'auto, il suono della sveglia);
- **luci** (un semaforo, un faro);
- **gesti** (un saluto, un abbraccio, un bacio).

Numeri e caratteri rappresentano i **dati semplici** (o *primitivi*), perché sono relativamente semplici da raccogliere e analizzare; gli altri costituiscono i **dati complessi**, in quanto derivanti dalla fusione di più dati semplici. In particolare, dati complessi quali suoni, animazioni, filmati prendono il nome di **dati multimediali**: di tale tipologia ci occuperemo a lungo in questo corso per la sua importanza nell'informatica dei nostri giorni.

“ L'attività che si occupa del trattamento dei dati per trasformarli in informazioni prende il nome di **elaborazione** (Fig. 3). ”

Figura 3  
Dati e informazioni



Concentrandoci sulle nuove tecnologie informatiche e della comunicazione, è importante distinguere tra **dati analogici** e **dati digitali**.

Gli ultimi decenni hanno visto la nascita e lo sviluppo di nuovi dispositivi elettronici quali computer, computer tascabili, videogame che oggi affiancano in molte famiglie radio, televisioni e videocamere. In ognuno di questi dispositivi i dati da elaborare e trasmettere sono rappresentati fisicamente da segnali elettrici.

Figura 4  
Un orologio analogico

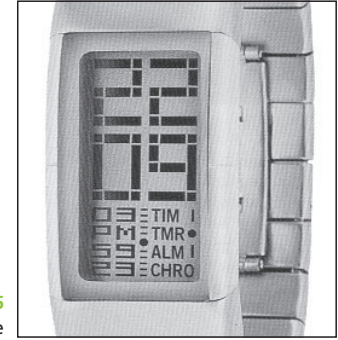


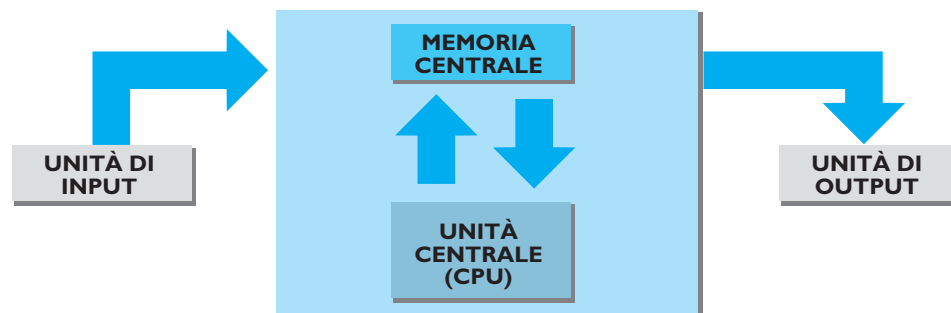
Figura 5  
Un orologio digitale

In un **sistema analogico** il segnale da elaborare è **continuo** (Fig. 4): un orologio analogico, quello con le lancette, rappresenta tutti gli istanti del tempo, senza salti. In un **sistema digitale**, i segnali da elaborare sono **discreti**, cioè non assumono tutti i valori di un intervallo, ma procedono a salti (Fig. 5): un orologio digitale rappresenta il tempo a salti di minuti o di secondi o anche di frazioni più piccole.

## 5 La macchina di von Neumann

I moderni *sistemi di elaborazione* oggi in commercio non sono tutti uguali, anzi presentano notevoli differenze in termini di velocità, prestazioni, memorie. Nonostante questo, la maggior parte dei sistemi di elaborazione presenta una caratteristica comune: il loro hardware rispecchia il modello ideato da **John von Neumann** nel 1946, che riportiamo nella **figura 6**.

Figura 6  
Lo schema di un elaboratore secondo il modello di von Neumann



Secondo tale modello, un sistema per l'elaborazione elettronica dei dati è composto da un insieme di **unità funzionali**, ognuna delle quali ha compiti ben precisi:

- le **unità di input** sono predisposte per l'immissione delle istruzioni dei programmi e dei dati su cui tali programmi dovranno operare;
- le **unità di output** permettono al sistema di comunicare con l'esterno, ossia di comunicare i risultati dell'elaborazione compiuta;
- la **memoria centrale** conserva dati e istruzioni;
- l'**unità centrale**, detta **CPU** (*Central Processing Unit*), ha il compito di eseguire le istruzioni, i calcoli aritmetici e logici e di controllare tutto il sistema.

La macchina di von Neumann era caratterizzata dalla presenza di una sola unità centrale: un computer basato su questo modello viene chiamato **elaboratore seriale**.

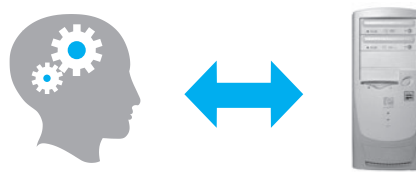
Esistono, poi, altre architetture di elaboratori (*array processor, pipeline, multiprocessor* ecc.) basate su altri modelli e utilizzate in settori specifici.

## 6 Il case e la scheda madre

Osservando la **figura 1** si nota che un computer è formato da varie parti: schermo, tastiera, mouse ecc., che sono collegati a dei dispositivi presenti all'interno di un contenitore di metallo detto **case** (o **modulo base**).

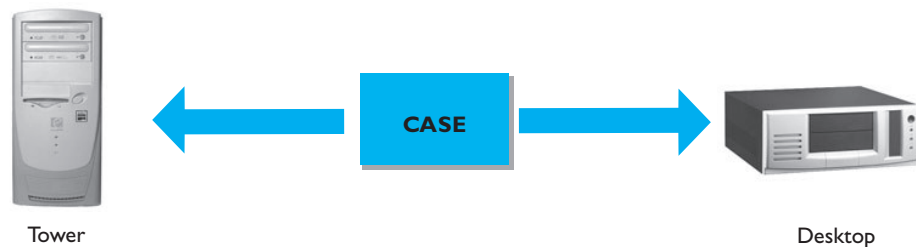
Paragonandolo al corpo umano, potremmo dire che il modulo base rappresenta la "testa" ossia la parte del corpo contenente il cervello e il "centro della memoria" (**Fig. 7**). Tale dispositivo, infatti, contiene tutte le componenti fondamentali necessarie per poter "manipolare" le informazioni.

**Figura 7**  
Modulo base di un PC



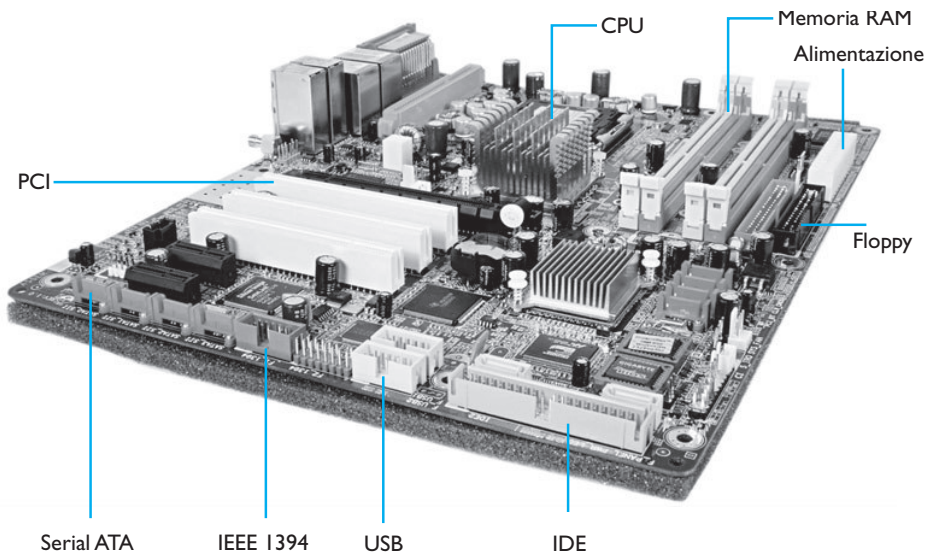
Il case assume nomi diversi in base alla conformazione (**Fig. 8**). Le versioni più utilizzate sono *Desktop* (da tavolo) e *Tower* (verticale) che, a seconda delle caratteristiche di grandezza, prende il nome di *Minitower, Miditower, Hightower* (piccolo, medio e grande).

**Figura 8**  
Tipi di case



All'interno del case trova alloggio la **scheda madre (motherboard)**, componente fondamentale del computer. La scheda madre è la componente che ha il ruolo di gestire la comunicazione fra tutte le componenti. A essa vengono collegate tutte le altre parti di un PC che grazie a essa possono dialogare fra loro (**Fig. 9**).

**Figura 9**  
La scheda madre di un computer





Con il termine **scheda** in elettronica si indica una unità fisica composta da una base (wafer) di sostegno arricchita da più componenti elettroniche (linee di alimentazione, chip, transistor ecc.)

debitamente collegate tra loro. Ogni scheda ha la possibilità di collegarsi con l'esterno (con altre schede o altri dispositivi in genere) attraverso "porte" o "interfacce di comunicazione".

## 7 Organizzazione della memoria centrale

“ La **memoria centrale** (detta anche **memoria principale** o **memoria di lavoro**) è un dispositivo elettronico in grado di contenere tutte le informazioni necessarie per l'elaborazione (cioè dati, istruzioni del programma da eseguire, risultati intermedi e finali). È quindi coinvolta continuamente durante il processo di elaborazione. ”

In memoria è possibile:

- **scrivere** i dati provenienti dalle varie componenti del computer. La scrittura è un'operazione distruttiva, ossia la scrittura di un'informazione comporta la distruzione di quella precedentemente contenuta nella stessa posizione;
- **conservare** informazioni, per averle sempre disponibili durante l'elaborazione;
- **leggere** l'informazione memorizzata, senza rischio di distruggerla. La lettura non è distruttiva, in quanto viene prelevata una copia dell'informazione letta.

All'interno del computer le informazioni non sono composte da numeri o lettere così come noi le digitiamo da tastiera, ma sono convertite in *codici digitali* realizzati con segnali elettrici. Questi codici sono rappresentabili mediante le cifre 0 e 1, proprio come gli stati assunti da un generico interruttore: **0 significa spento** (non c'è passaggio di corrente), **1 significa acceso** (c'è passaggio di corrente). Nell'elaboratore elettronico, quindi, si utilizzano **elementi bistabili** per immagazzinare dati e istruzioni. Possiamo immaginare la memoria come un dispositivo composto da migliaia di circuiti elettrici, che possono assumere solo due stati fisici a cui si associano i suddetti simboli 0 e 1. Essi sono detti *cifre binarie* o, semplicemente, **bit** (**binary digit**). La memoria centrale, quindi, è composta da **celle** (o *posizioni di memoria*) adiacenti, di uguale dimensione, che possono contenere un dato scritto in forma binaria.

Per rappresentare un qualunque carattere (cifra, lettera, simbolo ecc.), però, un singolo bit non basta; per questo si usano configurazioni di otto bit dette **byte** (**binary term**). Il byte assume particolare importanza, in quanto permette di rappresentare un carattere alfabetico, numerico o un simbolo speciale. Pertanto, una parola di dieci lettere occupa dieci byte, una di quattro ne occupa quattro e così via.

Il byte viene assunto come unità di misura della **capacità** della memoria. Per comodità si usano spesso multipli del byte. La configurazione appena più grande del singolo byte prende il nome di **parola** (o **word**) e riveste un ruolo di primaria importanza, in quanto la lunghezza della parola (cioè il numero di bit di cui è composta) definisce la **dimensione della cella standard di sistema**.

Le parole sono composte da un numero di bit *multiplo di otto* (16, 32 o 64 bit) e si parla di "sistemi a 16 bit", "sistemi a 32 bit" ecc. *Più alto è il numero di bit che compongono la parola, maggiore è, in genere, la velocità del computer.* Questo perché nell'elaboratore il trasferimento delle informazioni tra la memoria centrale e la CPU avviene *in parallelo*, ossia i bit che compongono una parola vengono trasmessi contemporaneamente. Usando parole più lunghe, diminuiscono gli accessi in memoria che la CPU deve eseguire quando necessita di un determinato numero di bit e, conseguentemente, aumenta la velocità del computer.

La seguente tabella mostra le unità di misura della memoria, multipli del byte.

BYTE	
UNITÀ DI MISURA	CORRISPONDE A
Byte	8 bit
KiloByte (KB)	$2^{10}$ byte 1024 byte
MegaByte (MB)	$2^{10}$ KB 1 048 576 byte
GigaByte (GB)	$2^{10}$ MB 1 073 741 824 byte
TeraByte (TB)	$2^{10}$ GB 1 099 511 627 776 byte
PetaByte (PB)	$2^{10}$ TB
HexaByte (HB)	$2^{10}$ PB

Le celle di memoria sono numerate progressivamente partendo da zero. Il numero che le contraddistingue, espresso naturalmente come sequenza di bit, viene detto **indirizzo** ed è molto importante: infatti a un'informazione contenuta nella memoria si può fare riferimento solo tramite l'indirizzo della cella che la contiene. Grazie all'indirizzo, quindi, è possibile stabilire, **univocamente**, su quale cella di memoria occorrerà effettuare l'**accesso**, ossia l'operazione di lettura o di scrittura. Il numero di bit utilizzati per identificare gli indirizzi definisce la dimensione dello spazio di indirizzamento, cioè il numero di celle di memoria direttamente indirizzabili.

Generalmente la memoria di un elaboratore è *indirizzabile* a livello di byte, ma quando la parola è composta da più di un byte (cioè da 16, 32 o 64 bit) viene indirizzata mediante l'indirizzo del suo *primo byte di destra o di sinistra* (in base alle convenzioni attuali).

Alla luce di questi concetti possiamo affermare che la parola è per il computer la **più piccola unità di informazione** significativa e indirizzabile.

## 8 | bus

Come *viaggiano* le informazioni all'interno della macchina? Il collegamento fisico tra le varie unità funzionali di un computer è realizzato da un insieme di linee, dette **bus**.

“ Il **bus** è costituito da una serie di collegamenti hardware (è simile a un gruppo di fili), uno per bit, su cui viaggiano tutti i dati che vengono scambiati tra l'unità centrale, la memoria e le unità di input/output. ”

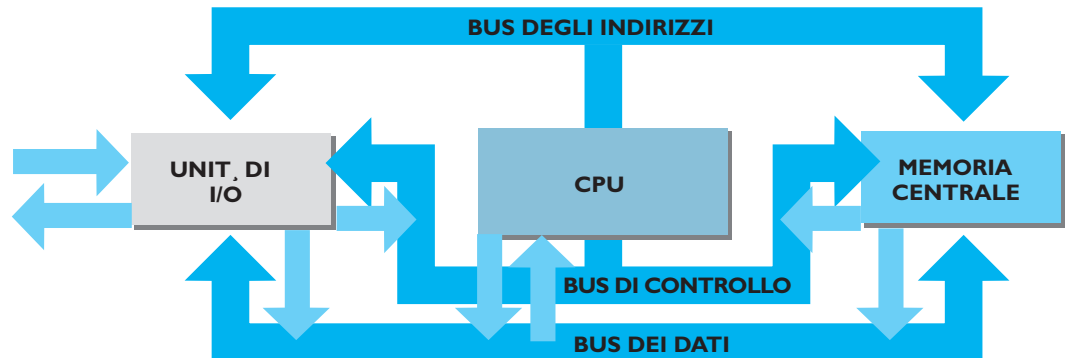
In un sistema di elaborazione si trovano tre bus principali (Fig. 10): il **bus degli indirizzi** (*address bus*), il **bus dei dati** (*data bus*) e il **bus di controllo** (*control bus*):

- il **bus degli indirizzi** trasporta l'indirizzo necessario per reperire una cella di memoria o un'unità di I/O. L'unico dispositivo abilitato a inviare informazioni su questo bus, che è **unidirezionale**, è la CPU. Il bus degli indirizzi è logicamente composto da tanti fili quanti sono i bit che compongono l'indirizzo;
- il **bus dei dati** viene utilizzato per lo scambio di informazioni tra i vari dispositivi. È **bidirezionale**, in quanto l'invio di dati non è di sola pertinenza della CPU. Inoltre è logicamente composto da tanti fili quanti sono i bit che compongono la parola utilizzata come unità di trasferimento (ad esempio 16, 32 bit);
- il **bus di controllo**, anch'esso bidirezionale, è utilizzato *per sincronizzare la trasmissione e per permettere lo scambio di segnali di controllo* tra le varie unità (segnali di lettura o scrittura, segnale di inizio o fine trasmissione, segnale di unità libera o occupata ecc.). Il suo scopo principale è, pertanto, di *coordinare e controllare il traffico* di tutte le informazioni



viaggianti sugli altri due bus. Il numero di fili componenti questo bus è variabile: più linee lo compongono, maggiore è il numero di informazioni che può trasportare.

**Figura 10**  
La struttura a bus di un elaboratore

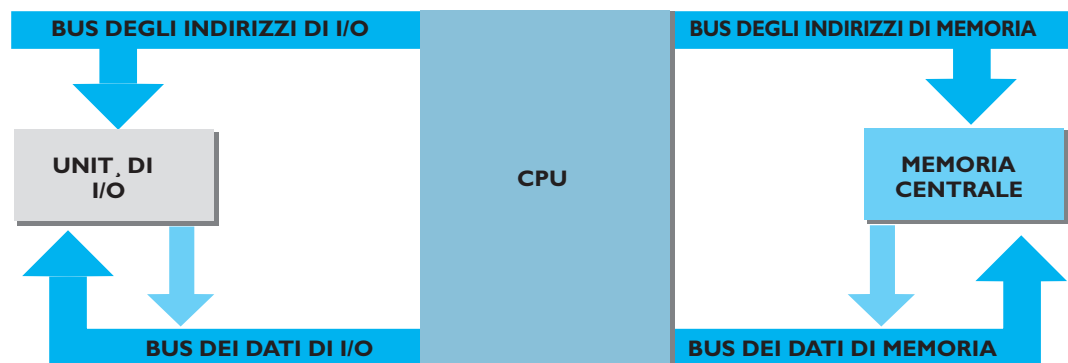


I bus necessitano di un **controllore**, ossia di un dispositivo spesso residente nello stesso microprocessore. Esso stabilisce quale dispositivo deve fare uso dei bus

evitando, così, i conflitti che potrebbero sorgere quando più unità tentano di accedere ai bus contemporaneamente.

La struttura a bus vista precedentemente e raffigurata nella **figura 10** prende il nome di **architettura unibus**. Spesso, per raggiungere velocità più elevate, le case costruttrici adottano altre architetture basate sulla presenza di *coppie distinte di bus degli indirizzi e di bus dei dati*: una coppia si dedicherà agli accessi in memoria, l'altra agli accessi ai dispositivi di I/O. Questa architettura, riportata nella **figura 11**, prende il nome di **architettura multibus** e offre la possibilità di poter colloquiare con le unità di I/O mentre stanno avvenendo accessi in memoria. I computer odierni utilizzano strutture di bus e di controller molto più sofisticati sia per gli accessi in RAM che per gli altri dispositivi di I/O.

**Figura 11**  
L'architettura multibus



## 9 Architettura della memoria centrale

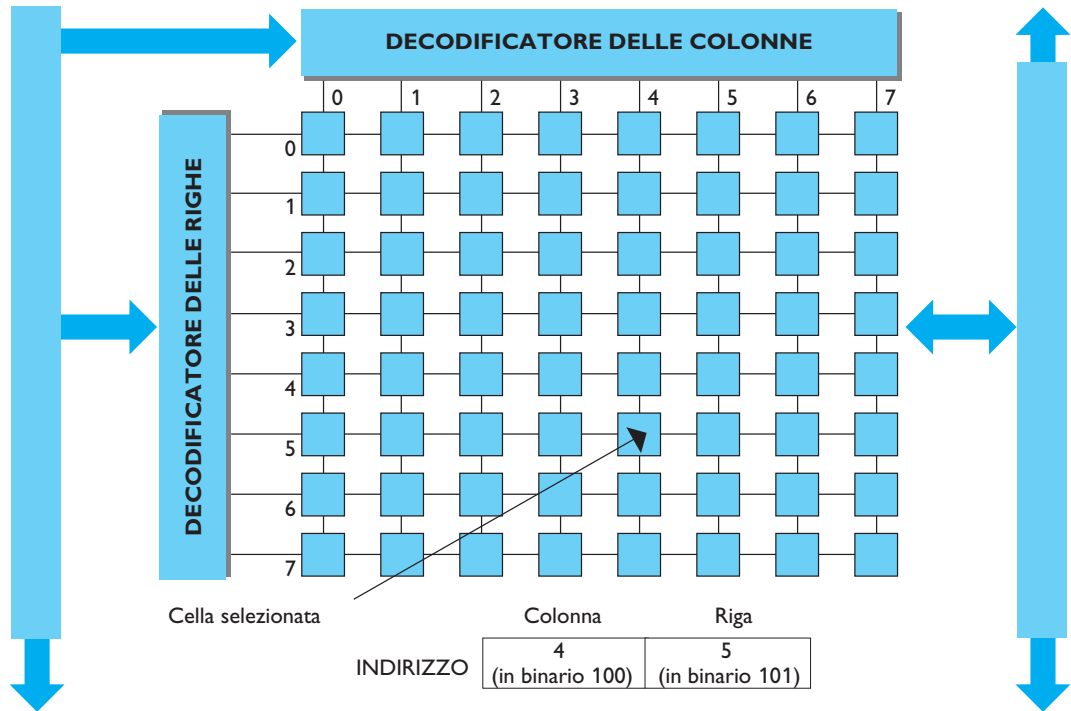
Le **celle della memoria** sono numerate progressivamente, ma non disposte in successione. Sono organizzate in una matrice quadrata, come nella **figura 12**.

A ogni intersezione tra le colonne e le righe si trovano le celle di memoria. L'**indirizzo di memoria**, quindi, è composto da due parti: la prima a destra individua la riga e la restante parte individua la colonna.

### ESEMPIO

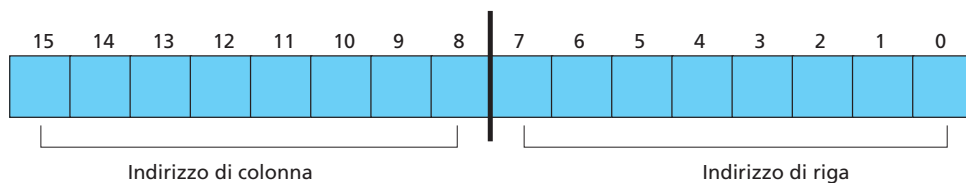
Un indirizzo di 16 bit viene così interpretato: i primi 8 bit meno significativi individuano la riga e i restanti individuano la colonna.

Figura 12  
La struttura della memoria centrale



Nella figura 13 è riportata l'interpretazione di un indirizzo di 8 bit.

Figura 13  
L'indirizzo di una cella di memoria



Altre componenti logiche della memoria centrale indispensabili per poter realizzare il dialogo con la CPU sono i **registri**.

“ I **registri** sono piccole e veloci memorie, generalmente composti da 1, 8, 16, 32 o 64 bit, che svolgono funzioni ben precise e vengono utilizzati in determinate fasi dell'elaborazione. ”

I **registri della memoria centrale**, che riportiamo nella figura 14, sono i seguenti:

- **MAR** (*Memory Address Register*). È il *registro degli indirizzi di memoria* e contiene l'indirizzo della cella di memoria in cui la CPU deve scrivere o da cui deve leggere un dato. La lunghezza di questo registro è pari alla lunghezza dell'indirizzo stesso;
- **MDR** (*Memory Data Register*). È il *registro dei dati di memoria* e contiene la parola che deve essere scritta o quella appena letta. La capacità di questo registro è, quindi, pari alla lunghezza della parola;
- **CONTR**. Questo registro contiene appositi messaggi provenienti dalla memoria stessa e dalle altre unità. Ad esempio, dopo un accesso, la memoria imposta in esso un messaggio di *pronto*, mentre le altre unità, quando richiedono l'utilizzo della memoria, caricano in esso un messaggio di *lettura* o di *scrittura*.

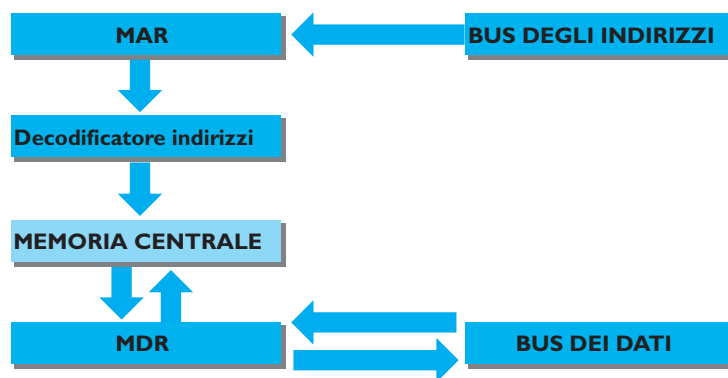
Vediamo, ora, come avvengono tecnicamente la **lettura** e la **scrittura** di un dato. Quando la CPU necessita di un dato, invia al MAR, sul bus degli indirizzi, l'indirizzo della cella di memoria che intende leggere. Contemporaneamente la CPU invia al CONTR, sul bus di controllo, il messaggio di lettura. Una volta ricevuti questi ordini, la memoria si posiziona sulla cella

riportante l'indirizzo indicato dal MAR, legge il dato e lo deposita nel registro MDR, poi imposta nel registro CONTR un messaggio di *pronto* e invia il tutto sugli appositi bus. La **lettura** è così compiuta.

La **scrittura**, invece, inizia con l'invio alla memoria, da parte della CPU, del dato che dovrà essere memorizzato, dell'indirizzo della cella che lo dovrà contenere e del segnale di scrittura. Tali informazioni saranno inviate alla memoria per mezzo degli appositi bus e depositate nei registri preposti. Terminato il trasferimento, la memoria analizzerà il registro CONTR e provvederà a registrare il dato presente nel MDR nella cella il cui indirizzo è riportato nel MAR. A registrazione avvenuta, la memoria trasmetterà, sul bus di controllo, il messaggio di *pronto*.

Questo insieme di azioni (necessarie per poter compiere un'operazione di lettura o di scrittura) prende il nome di **ciclo di memoria** e il tempo necessario per realizzarlo si chiama **tempo di accesso** che, negli attuali elaboratori, è dell'ordine dei nanosecondi (1 nanosecondo = 1 milionesimo di secondo).

Figura 14  
L'architettura della memoria centrale



La memoria centrale è generalmente costituita da tre memorie:

- la memoria **RAM** (*Random Access Memory* = memoria ad accesso casuale);
- la memoria **ROM** (*Read Only Memory* = memoria a sola lettura);
- la memoria **CACHE** o *memoria tampone*.

La **memoria RAM** è la memoria di lavoro sinora descritta e, come ben sappiamo, offre la possibilità di leggere e scrivere dati e istruzioni necessari per l'esecuzione di un programma. È

ad **accesso casuale** perché in essa ogni *informazione può essere letta o scritta* in un tempo che non dipende dalla posizione in cui l'informazione è registrata. In questo tipo di memoria il tempo di accesso è costante.

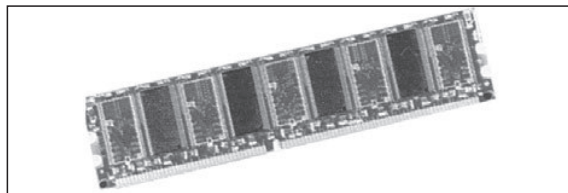
La RAM è una memoria velocissima, ma è

**volatile**, cioè perde il suo contenuto quando viene interrotta l'alimentazione elettrica, per via del tipo di materiale impiegato per la sua costruzione.

La **memoria ROM** è una memoria di sola lettura. In essa sono memorizzate **permanente**mente le istruzioni che la CPU deve leggere e interpretare per poter svolgere tutti i suoi compiti. Tale memorizzazione avviene all'atto della fabbricazione. Più semplice e meno costosa della RAM, la ROM è una memoria a sola lettura, in quanto viene programmata dalle ditte produttrici di hardware e il suo contenuto non può più essere cancellato o modificato: può solo essere letto. Per questa caratteristica è adatta a contenere programmi particolari, quali, ad esempio, quelli necessari per l'inizializzazione del sistema, ossia tutti quei programmi che vengono eseguiti al momento dell'avviamento della macchina.

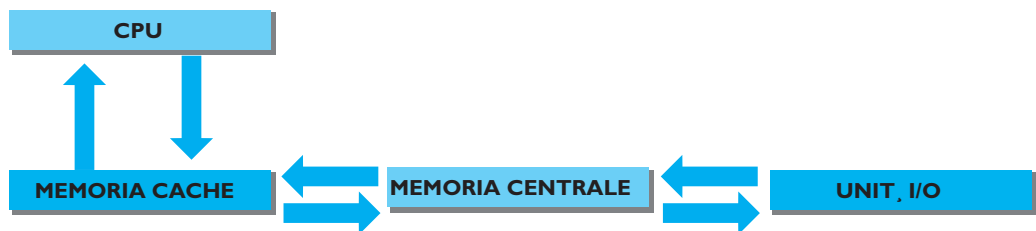
Esistono due fondamentali tipi di ROM: la ROM mascherata, della quale abbiamo parlato poc'anzi e la ROM programmabile, detta **PROM** (**P**rogrammable **R**OM), sulla quale l'utente può sempre leggere le informazioni memorizzate, ma può scrivere *una sola volta* per mezzo

Figura 15  
La memoria RAM



di apposite apparecchiature, dette *programmatori di PROM*. Essendo programmabili una sola volta, eventuali errori commessi in fase di programmazione non possono essere corretti. Per poter cambiare il programma memorizzato, è necessario utilizzare un'altra PROM. Per ovviare a questo inconveniente, sono state realizzate le **EPROM (Erasable PROM)** sulle quali, invece, l'utente grazie ai *programmatori di EPROM* può scrivere ripetutamente, sfruttando le potenzialità dell'energia elettrica (per memorizzare) e dei raggi ultravioletti (per cancellare). Si tratta, quindi, di un "ibrido" tra RAM e PROM. La **memoria CACHE** è una particolare memoria RAM caratterizzata da un'elevatissima velocità. È utilizzata come memoria intermedia tra CPU e memoria centrale, per aumentare la velocità e le prestazioni del computer (Fig. 16). Essenzialmente è una memoria di transito con accesso molto rapido e viene utilizzata per memorizzare i dati temporanei usati più frequentemente, rendendoli disponibili senza doverli prelevare ogni volta dalla memoria RAM che è decisamente meno veloce.

Figura 16  
Il computer con memoria cache



La memoria cache può essere organizzata su due livelli: la **cache di primo livello (L1)**, inclusa nel chip che ospita il processore, e la **cache di secondo livello (L2)**, incorporata sulla scheda madre. Essendo incorporata nel microprocessore, la cache L1 lavora alla sua stessa velocità (frequenza) e, pertanto, risulta essere *più veloce e più costosa della L2*. A questo punto, possiamo affermare che:

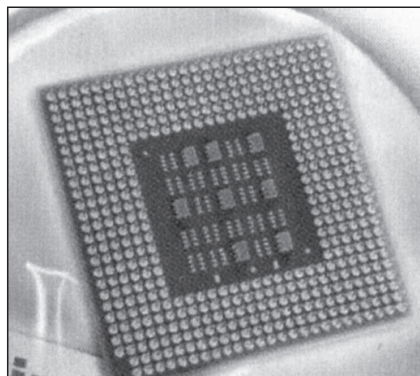
- tempo di accesso;
- capacità;
- lunghezza della parola;
- velocità;
- metodi di accesso;

sono i **parametri** fondamentali che caratterizzano le memorie.

## 10 La CPU (Central Processing Unit)

“ La **CPU (Central Processing Unit)** rappresenta il nucleo di tutto il sistema di elaborazione ed è responsabile dell'esecuzione e del controllo dei processi: per tale motivo, viene anche chiamata **processore centrale**. ”

Figura 17  
Il microprocessore



Questo importantissimo processore (ma non è l'unico) viene realizzato, proprio come la memoria centrale, su un chip di silicio. La ridottissima dimensione di questo dispositivo gli conferisce il nome di **microprocessore** o **microchip** (Fig. 17). Microprocessore e CPU non sono proprio la stessa cosa anche se, in pratica, coincidono fisicamente: con il termine **microprocessore** infatti ci riferiamo all'oggetto che si trova nel computer (e ormai in tantissimi altri dispositivi, dalle automobili alle macchine fotografiche agli impianti HI-FI), mentre con **CPU** ci riferiamo alla funzione svolta da tale oggetto.

La CPU ha il compito di:

- provvedere alla *registrazione in memoria centrale* dei dati e delle istruzioni dei programmi;
- riconoscere e interpretare le *istruzioni del programma* da eseguire;
- eseguire le *operazioni logiche e aritmetiche*;
- controllare tutte le operazioni necessarie per eseguire il *processo di elaborazione* e gestire il traffico di informazioni con l'esterno.

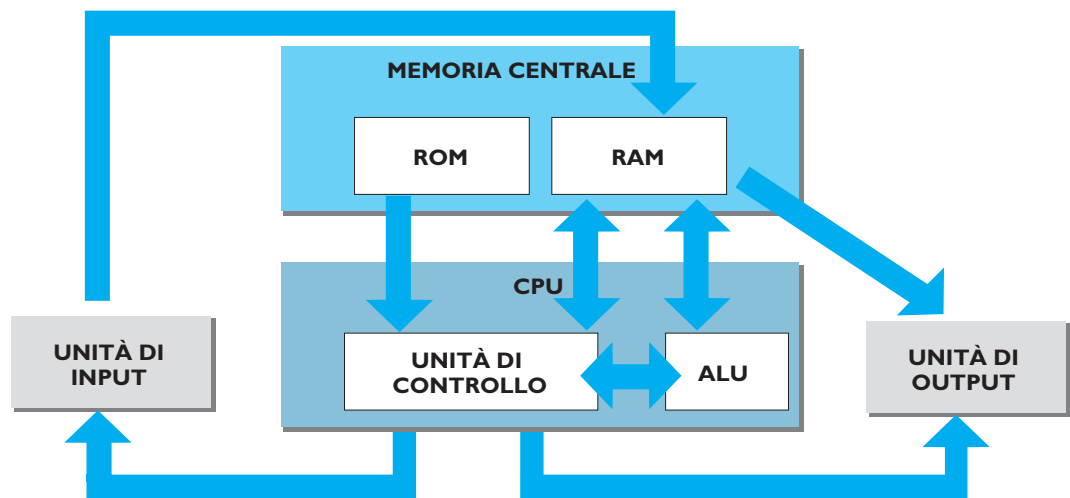
Questa unità funzionale è composta da:

- un'**unità di controllo** detta **CU** (*Control Unit*);
- un'**unità aritmetico-logica** detta, generalmente, **ALU** (*Arithmetic Logic Unit*);
- alcuni **registri**.

Questi elementi sono collegati tra loro da un **bus interno** sul quale viaggiano le informazioni.

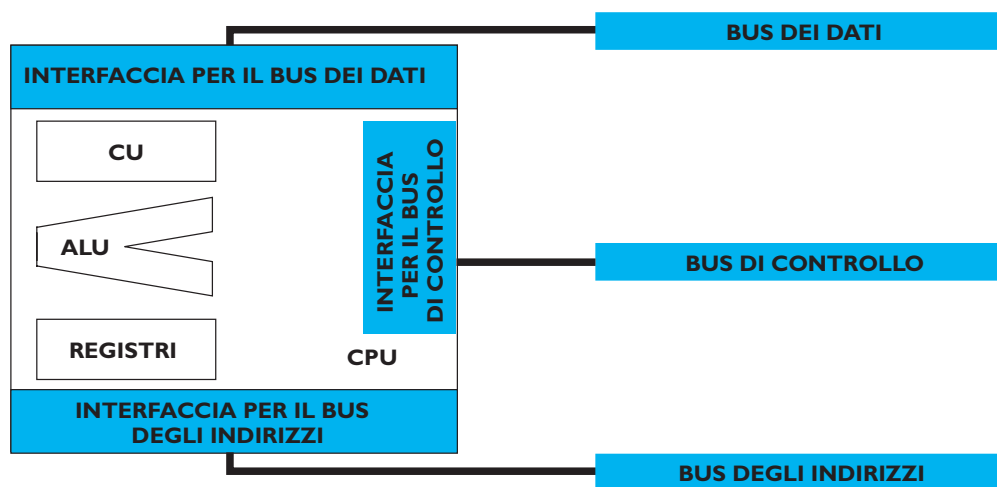
Schematizziamo più in dettaglio la macchina di von Neumann e riportiamo la struttura di una CPU (**Fig. 18**).

**Figura 18**  
La schematizzazione dettagliata della macchina di von Neumann e la struttura di una CPU



Nella **figura 19** riportiamo invece la struttura di una CPU.

**Figura 19**  
La struttura di una CPU



Nella struttura dell'unità centrale di elaborazione riportata nella **figura 19** abbiamo utilizzato il termine **interfaccia**.

“ L'**interfaccia** è qualunque elemento *hardware* o *software* che consente il collegamento fisico e/o logico tra dispositivi diversi. ”

È un'interfaccia, quindi, sia il cavo che permette di collegare il video all'elaboratore, sia il dispositivo che consente il trasferimento dei dati dalla CPU ai bus. Ma anche i linguaggi di programmazione e i prodotti software sono interfacce; consentono infatti il dialogo uomo-macchina, permettendo all'utente di sfruttare le potenzialità del computer senza doverne conoscere le caratteristiche tecniche.

### 10.1 L'unità di controllo (Control Unit: CU)

“ L'**Unità di controllo** (o di governo) ha il compito di gestire e sovrintendere al funzionamento di tutte le unità del computer e all'esecuzione di tutti i processi da compiere per eseguire i programmi residenti in memoria centrale. ”

Questa unità è composta da:

- un **registro dell'istruzione corrente** (*IR = Instruction Register*), che contiene l'istruzione che la CU deve eseguire;
- un **registro contatore di programma** (*PC = Program Counter*), che contiene l'indirizzo di memoria in cui è contenuta l'istruzione successiva a quella che la CU sta eseguendo (cioè quella il cui codice è contenuto nel registro IR);
- alcuni **registri generali**, comodi per aver sempre disponibili dati che potranno essere utili durante l'esecuzione del programma. Ad esempio: dati o risultati intermedi, che permettono così un notevole risparmio di tempo;
- il **registro Stack Pointer** (*SP*), utilizzato per gestire una zona di memoria con la tecnica della **pila** (*LIFO*). Quest'area di memoria sarà utilizzata per salvare e recuperare non solo dati, ma anche la situazione del processore dopo l'arrivo e la gestione di un **segnale di interruzione**;
- un **decodificatore**, che ha il compito di decodificare (ossia di interpretare) l'istruzione contenuta nel registro *IR* al fine di riconoscerne il tipo;
- un **temporizzatore** (*clock*), cioè un oscillatore al quarzo che genera una serie di impulsi a frequenza regolare, utilizzato per sincronizzare l'invio di tutti i segnali generati dalla CU.

Con questi dispositivi, la CU è in grado di svolgere le sue principali funzioni:

- **prelevare** le istruzioni del programma da eseguire registrate in memoria centrale;
- **interpretare**, volta per volta, le istruzioni lette;
- **eseguire** le istruzioni attivando e controllando le unità coinvolte nell'esecuzione.

### 10.2 L'Unità Aritmetico-Logica (Arithmetic Logic Unit: ALU)

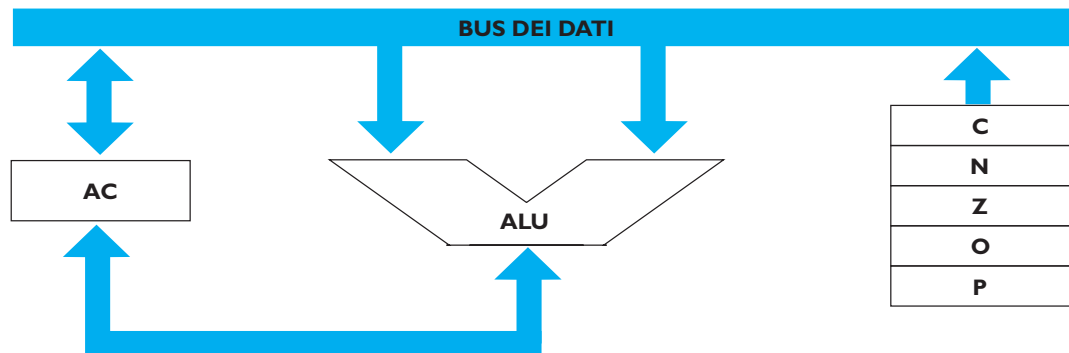
“ All'**Unità Aritmetico Logica** spetta il compito di eseguire, sotto la continua supervisione della CU, le operazioni aritmetiche e logiche sui dati provenienti dalla memoria. ”

L'**ALU** svolge il suo compito attraverso:

- una **rete logica di calcolo**, costituita da:
  - un circuito addizionatore per eseguire le operazioni aritmetiche operante con due parole in ingresso e una in uscita, codificate in formato binario;
  - alcuni circuiti per eseguire le operazioni logiche elementari (AND, OR, NOT), lo shift, la determinazione del segno di un numero e il confronto;
- uno o più **registri accumulatori** (*AC*), utilizzati per contenere temporaneamente i dati provenienti dalla memoria centrale. Tramite questi registri vengono eseguiti materialmente i calcoli. Se la macchina dispone di un solo registro accumulatore, durante l'ese-

- cuzione delle operazioni aritmetico-logiche, in esso sarà depositato un operando in entrata e, dopo l'elaborazione, sarà contenuto il risultato in uscita. Se la macchina dispone di più accumulatori, invece, essi saranno utilizzati anche per mantenere i risultati temporanei delle elaborazioni, evitando, così, ulteriori accessi in memoria centrale;
- un **registro di stato** (SR = *Status Register*) con il quale comunica alla CU informazioni sul tipo di risultato ottenuto dopo l'esecuzione dell'operazione. Questo registro è composto da un insieme di bit (*flag*), a ognuno dei quali è attribuito un significato particolare. I **bit di stato** più comuni sono i seguenti:
    - **C** (*bit di carry*), chiamato anche bit di riporto: contiene il valore binario 1 se nell'operazione eseguita dall'ALU c'è stato un riporto;
    - **N** (*bit di segno*), chiamato anche bit di negatività: contiene 1 se il risultato dell'operazione terminata è negativo;
    - **Z** (*bit di zero*): contiene 1 se il risultato dell'operazione conclusa è uguale a zero;
    - **O** (*bit di overflow*): contiene 1 se nell'operazione conclusa c'è stato un trabocco;
    - **P** (*bit di parità*): contiene 1 se il risultato dell'operazione è costituito da un numero pari di cifre 1.

**Figura 20**  
Lo schema di un'unità aritmetico-logica con un solo registro accumulatore



## APPROFONDIMENTO

### 11 Linguaggio macchina e formato delle istruzioni

Le istruzioni di un programma da eseguire, residente in memoria centrale, sono codificate in *linguaggio macchina*, unico linguaggio comprensibile dalla CPU.

“

Il **linguaggio macchina** è composto da un insieme di istruzioni macchina (esprese utilizzando il codice binario). Tali istruzioni svolgono ciascuna una funzione elementare, eseguibile direttamente dalla CPU.

”

Per poter eseguire il programma è necessario che tutte le istruzioni macchina che lo compongono siano contenute nella memoria centrale in *parole* adiacenti. Per quanto riguarda la natura funzionale delle istruzioni, queste possono essere divise in cinque gruppi, cioè l'insieme minimo di classi di istruzioni eseguibili dalla CPU:

- **istruzioni di input/output dei dati**: con queste istruzioni si opera sui dispositivi che trasferiscono i dati dalle unità di input alla memoria centrale del computer (o da quest'ultima alle unità di output). Ad esempio: "Leggi N" oppure "Stampa N";
- **istruzioni aritmetico-logiche**: con queste istruzioni si eseguono le operazioni di calcolo (come l'addizione o la divisione) e le operazioni logiche (AND, OR, NOT ecc.);
- **istruzioni di controllo e di salto**: con queste istruzioni si interviene per alterare la sequenza di un programma in fase di esecuzione, saltando senza che si verifichi alcuna condizione verso un altro punto del programma (**salto incondizionato**), oppure saltando un determinato gruppo di istruzioni in conseguenza del risultato di un test (**salto condizio-**