



***Hardware:  
memoria centrale***

## 1. La memoria

Una delle componenti fondamentali di un computer è la memoria: un dispositivo capace di conservare nel tempo un dato.

La memoria di un computer può essere considerata come una sequenza finita di **celle** o **locazioni di memoria** dove ad ogni cella di memoria è associato, in maniera biunivoca, un **indirizzo** ovvero un numero che identifica la cella e permette di rintracciarla.

Indirizzo	Locazione/Cella
0000	01011010
0001	10001010
0010	11010010
0011	01011010
0100	01001010
0101	11011110
....	.....

Figura 1: Esempio di celle di memoria

La **Capacità** di una memoria, indica la quantità di informazioni che può contenere. L'Unità di Misura della capacità di una memoria è il Byte.

Le moderne memorie sono molto capaci, ossia sono composte da numerosissimi BIT. Per questo motivo, per indicarne la capacità, è indispensabile usare i seguenti Multipli del Byte:

- **Kilo-Byte (KB)** = 1024 Byte = circa 1000 Byte
- **Mega-Byte (MB)** = 1024 KB = circa 1 milione di Byte
- **Giga-Byte (GB)** = 1024 MB = circa 1 miliardo di Byte
- **Tera-Byte (TB)** = 1024 GB = circa 1000 miliardi di Byte

Il **tempo di accesso** (o Velocità) di una memoria, è il tempo che intercorre fra il momento in cui la memoria riceve la richiesta di recuperare un dato precedentemente memorizzato e il momento in cui la memoria rende effettivamente disponibile il dato richiesto.

Il tempo di accesso di una memoria si misura in secondi (s) ma, vista la velocità raggiunta dagli attuali dispositivi di memoria, si preferisce utilizzare i Sottomultipli del Secondo:

- **milli-secondo (ms)** =  $1 / 1.000 \text{ s} = 10^{-3} \text{ s}$  = millesimo di secondo
- **micro-secondo ( $\mu\text{s}$ )** =  $1 / 1.000.000 = 10^{-6} \text{ s}$  = milionesimo di secondo
- **nano-secondo (ns)** =  $1 / 1.000.000.000 = 10^{-9} \text{ s}$  = miliardesimo di secondo

## 2. La gerarchia di memoria

Il sistema di memorie viene progettato e gestito in modo da ottenere:

- capacità di memorizzazione adeguata, in quanto l'elaborazione delle informazioni sui moderni sistemi di calcolo richiede grandi quantità di memoria;
- prestazioni accettabili, per non ritardare le esecuzioni dei calcoli, già in quantità considerevoli;
- costi ridotti.

Per questi motivi viene utilizzata una gerarchia che cerca di ottimizzare costi (minimi), capacità (massime) e tempi di accesso (minimi).

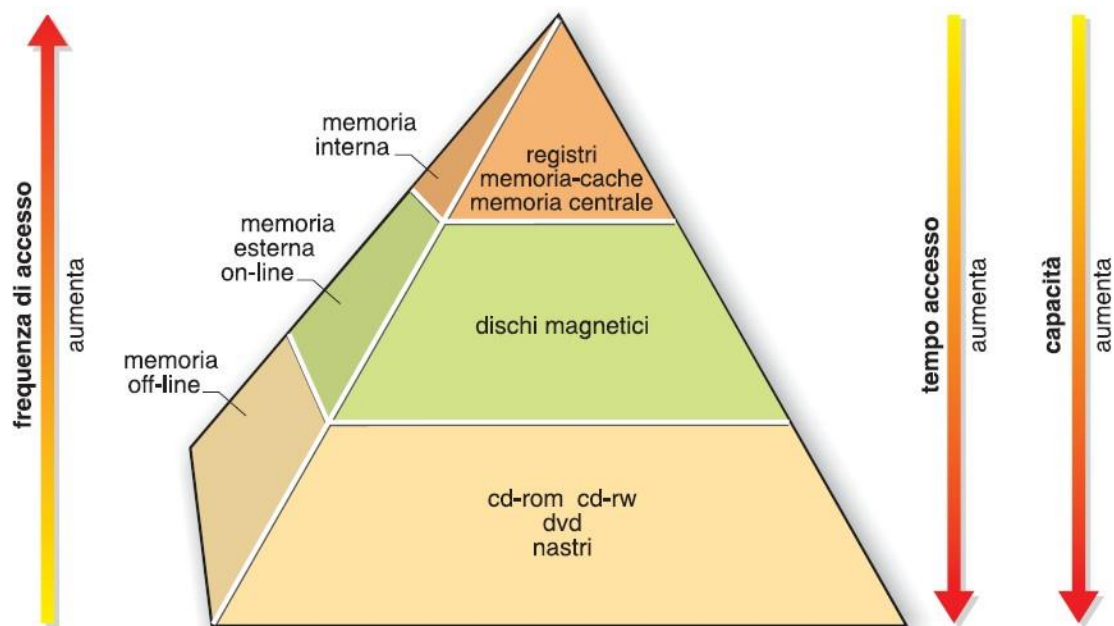


Figura 2: Gerarchia di memorie

Come si nota dalla figura 2, al livello più alto (quello più vicino alla CPU) troviamo memorie più piccole e veloci, al livello più basso memorie più lente e meno costose.

L'obiettivo di questa gerarchia è quello di dare all'utente l'illusione di poter usufruire di una memoria al tempo stesso veloce (quanto la memoria al livello più alto) e grande (quanto quella al livello più basso).

Il meccanismo di gestione della gerarchia di memoria è conosciuto con il nome di **Memoria Virtuale**. Alla base del sistema della gerarchia vi è il **principio di località**, che in informatica è un principio che dipende dall'esatta natura del programma, quindi non è da considerarsi una legge inflessibile, che non può essere cambiata, ma come una linea guida in quanto valido per la maggior parte dei casi.

Secondo tale principio, durante l'esecuzione di un'istruzione presente in memoria è molto probabile che le successive istruzioni saranno ubicate nelle vicinanze di quelle in corso.

Esistono due differenti tipi di località:

- **località temporale (nel tempo):** se la CPU sta eseguendo un'istruzione presente in memoria, vuol dire che con molta probabilità le prossime istruzioni da eseguire si troveranno fisicamente nelle vicinanze di quella in corso; vale a dire che, durante la normale esecuzione dei programmi, la CPU passa molto tempo accedendo a zone di memoria ristrette e solo occasionalmente accede a locazioni molto lontane;
- **località spaziale (nello spazio):** in genere, gran parte del codice di cui sono costituiti i programmi viene eseguito solo raramente, al verificarsi di errori o condizioni anomale: quindi, succede spesso che di tutto il codice che un programma carica in memoria ne venga realmente eseguita solo una piccola parte. La località temporale si basa su questo: nell'arco temporale di esecuzione del programma, si accede molto spesso alle stesse zone del programma. Dunque, mantenendo una copia nella memoria cache dei dati più richiesti durante l'esecuzione del programma, si sfrutta tale principio<sup>2</sup>.

### 3. La memoria centrale

La memoria centrale è utilizzata durante lo svolgimento delle operazioni per contenere sia programmi memorizzati dal costruttore, come quelli relativi al sistema operativo, sia risultati o dati da utilizzare temporaneamente durante l'esecuzione delle operazioni.

Su una memoria è possibile fare operazioni di lettura e scrittura.

In genere per far riferimento a queste operazioni in maniera generica si utilizza il termine **"accesso"**.

L'accesso alla memoria può essere:

- **sequenziale:** le celle di memoria sono disposte in sequenza e per accedere ad una qualunque di esse è necessario scorrere tutte quelle che la precedono o la seguono;
- **casuale:** quando il tempo necessario per ricercare il dato e leggerlo/scriverlo non dipende dalla posizione del dato ma è lo stesso per tutti i dati conservati in memoria;
- **diretto:** quando è possibile accedere direttamente a qualunque indirizzo di memoria con tempo di accesso variabile e dipendente dall'indirizzo di memoria a cui è avvenuto l'accesso precedente.

Esistono diverse tipologie di memoria centrale che differiscono per velocità e modalità di accesso:

- la RAM;
- la ROM;
- la cache.

### 3.1 RAM (Random Access Memory)

La **RAM** (**R**andom **A**ccess **M**emory) è una memoria ad accesso casuale, essa serve a contenere i dati e le istruzioni dei programmi in esecuzione (o almeno una parte). La sua funzione è strettamente legata al fatto che i tempi di lettura/scrittura sulla RAM sono nettamente inferiori rispetto a quelli dei supporti di memoria di massa.

Questo tipo di memoria ha la caratteristica di essere assai veloce nel memorizzare i dati, ma allo stesso tempo è **volatile**, ovvero mantiene i dati memorizzati solo fino allo spegnimento del computer.

Si distinguono generalmente due categorie di memorie RAM:

- **DRAM** (**D**ynamic **R**andom **A**ccess **M**odule)
- **SRAM** (**S**tatic **R**andom **A**ccess **M**odule)

La DRAM fu inventata e brevettata per la prima volta da Robert Dennard nel 1968 ma fu messa sul mercato da Intel nell'ottobre 1970.

Questo tipo di memoria memorizza le sue informazioni in una cella contenente un condensatore e un transistor, proprio a causa di questo design, queste celle devono essere aggiornate ad un intervallo di tempo regolare detto **ciclo di refresh** affinché la memoria mantenga i propri dati.

La velocità di memoria DRAM è caratterizzata da due fattori:

- il **transfer rate**: ovvero il numero di trasferimenti di dati che un singolo banco di memoria può gestire nell'unità di tempo;
- il **CAS Latency**: ovvero il numero di cicli di clock che intercorrono tra il momento in cui il controller della RAM chiede di accedere ai dati contenuti in una o più celle di memoria e il momento in cui i dati sono disponibili per essere trasferiti alla CPU.

La SRAM è un tipo di RAM volatile che a differenza della DRAM non necessita di refresh.

Le memorie SRAM possono essere suddivise in:

- **async SRAM** (SRAM asincrona): lavora in modo asincrono rispetto al clock della CPU, questo implica degli stati di attesa della CPU per l'accesso;
- **sync SRAM** (SRAM sincrona): lavora in sincronia con il clock della CPU, si hanno quindi dei tempi di attesa ridotti (o annullati).

## 3.2 ROM (Read Only Memory)

La memoria ROM è una memoria di sola lettura non cancellabile e non riscrivibile, che a differenza della RAM non perde le informazioni allo spegnimento del computer.

La memoria ROM contiene le informazioni necessarie all'avvio del PC. Ogni volta che si accende il computer, il processore va a controllare il tipo di hardware che è presente, e, attraverso vari test di controllo, verifica che non ci siano problemi per qualche componente.

La ROM viene scritta dall'azienda che produce il computer e non può essere modificata dall'utente.

In particolare, la ROM è utilizzata per memorizzare il firmware ovvero il BIOS che è l'insieme delle operazioni base che un PC effettua all'avvio.

L'operazione di avvio è detta **bootstrap** e si articola in tre fasi:

1. verifica dell'hardware;
2. attivazione dell'hardware;
3. avvio del software di base del computer ovvero il sistema operativo.

Nonostante la ROM sia una memoria di sola lettura, su di essa è possibile effettuare alcune operazioni di scrittura e a seconda di come queste operazioni vengono effettuate distinguiamo diverse topologie di ROM, quali:

- **PROM (Programmable ROM)**: è una memoria programmabile, quindi scrivibile, una sola volta dopo la sua costruzione;



- **EPROM** (Erasable **P**rogrammable **ROM**): una volta programmata, non può più essere modificata direttamente ma può essere però cancellata tramite raggi ultravioletti. Una volta cancellato il suo contenuto, può essere programmata una sola volta prima di un'altra eventuale cancellazione
- **EEPROM** (Electrically **E**rasable **P**rogrammable **ROM**): in questa memoria i dati possono essere modificati direttamente attraverso impulsi elettrici.

### 3.3 La memoria **CACHE**

La **cache** è una memoria temporanea utilizzata per trasferire dati tra dispositivi che lavorano a velocità diverse.

L'introduzione di queste memorie consente di intervenire efficacemente in tutte quelle situazioni in cui l'avanzamento dei processi è condizionato dai rallentamenti dovuti ai tempi diversi di elaborazione da parte delle unità del sistema.

È possibile trovare memorie cache:

- nella comunicazione tra memoria RAM e unità a disco;
- nel trasferimento dati tra memoria RAM e CPU quando la RAM non è in grado di servire tempestivamente le richieste della CPU.