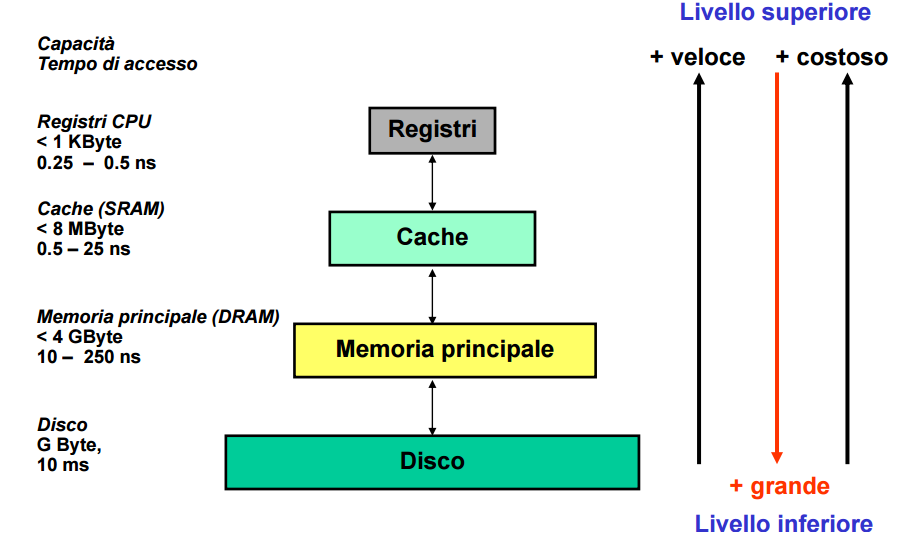
Memoria cache

**Obiettivo:** migliorare le prestazioni di un calcolatore attraverso il sistema di memoria in modo da:

• fornire agli utenti l’illusione di poter usufruire di una memoria contemporaneamente grande e veloce

• fornire al processore i dati alla velocità con cui è in grado di elaborarli (molto maggiore della velocità delle memorie)

La soluzione consiste nell’organizzare una gerarchia di memorie, ciascuno di diversa velocità e dimensione, realizzati con tecnologie diverse in modo da ottenere un buon compromesso costo/prestazioni. L’obiettivo è quello di fornire all’utente una quantità di memoria pari a quella disponibile nella tecnologia più economica, consentendo allo stesso tempo una velocità di accesso (circa) pari a quella garantita dalla tecnologia più veloce.



**Concetti base**

• Una gerarchia di memoria è composta da più livelli, ma i dati vengono di volta in volta copiati solo tra due livelli adiacenti.

• Consideriamo i due livelli: cache e memoria principale. La cache (livello superiore) è più piccola, veloce e costosa rispetto alla memoria principale (livello inferiore). La minima quantità di informazione che può essere presente o assente nella cache è il blocco o cache line. Per sfruttare la località spaziale è necessario che la dimensione del blocco della cache sia un multiplo della dimensione della **parola** di memoria (che noi considereremo per semplicità corrispondente ad un byte)

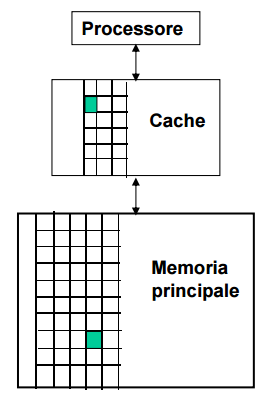
Ad esempio, un blocco da 128 bit può essere costituito da 4 parole da 32 bit

• Il numero di blocchi presenti in cache è dato da: Num. blocchi in cache = Dim Cache / Dim. Blocco

Ad esempio, una cache da 64 K Byte e blocco da 128 bit (16 Byte) contiene 4 K blocchi (212 blocchi)

Successo nell'accesso alla cache

Se il dato richiesto dal processore compare in uno dei blocchi presenti nel livello superiore, si dice che la richiesta ha successo (hit).



**Fallimento nell’accesso alla cache**

Se il dato non si trova nel livello superiore, si dice che la richiesta fallisce (miss) ⇒ per trovare il blocco che contiene i dati richiesti, bisogna accedere al livello inferiore della gerarchia. In caso di fallimento nell’accesso ad un dato:

• stallo della CPU;

• richiesta del blocco contenente il dato cercato alla memoria;

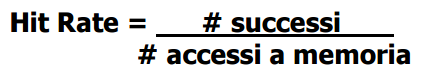
• copia in cache;

• ripetizione dell’accesso in cache;

**Gerarchia di memoria: definizioni**

Hit (successo): dati presenti in un blocco del livello superiore.

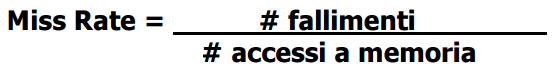
• Hit Rate (frequenza dei successi): numero di accessi a memoria che trovano il dato nel livello superiore rispetto al numero totale di accessi a memoria



• Hit Time (tempo di successo): tempo per accedere al dato nel livello superiore della gerarchia, che comprende anche il tempo necessario per stabilire se il tentativo di accesso si risolve in un successo o in un fallimento.

Miss (fallimento): i dati devono essere recuperati dal livello inferiore.

• Miss Rate (frequenza di fallimento) : numero di accessi a memoria che non trovano il dato nel livello superiore rispetto al numero totale di accessi a memoria



* Hit Rate + Miss Rate = 1
* Miss Penalty (penalizzazione): tempo necessario a sostituire un blocco nel livello superiore.
* Miss Time (tempo di fallimento ) :
* Miss Time = Hit Time + Miss Penalty
* Notare che: Hit Time << Miss Penalty

Tempo medio di accesso alla memoria

Tm = Hit Rate \* Hit Time + Miss Rate \* Miss Time

Dove:

Miss Time = Hit Time + Miss Penalty

Hit Rate + Miss Rate = 1

⇒ Tm = Hit Time + Miss Rate \* Miss Penalty

Struttura di una cache

Ogni posizione (entry) della cache include:

1. Valid bit che indica se questa posizione contiene o meno dati validi. Quando il calcolatore viene acceso tutte le posizioni della cache sono segnalate come NON valide

2. Campo etichetta (tag) che contiene il valore che identifica univocamente l’indirizzo di memoria corrispondente ai dati memorizzati

3. Campo dati che contiene una copia dei dati (blocco o cache line )



**Problema del piazzamento di un blocco**

• Problema: dato un indirizzo di un blocco nella memoria Problema principale, determinare la sua posizione nella cache.

• Occorre stabilire una corrispondenza tra l’indirizzo in memoria del blocco e l’indirizzo nella cache.

• Questa corrispondenza dipende dall’architettura della cache:

• Cache a indirizzamento diretto

• Cache completamente associativa

• Cache set-associativa a n vie

**Cache a indirizzamento diretto (direct mapped)**

• Numero di blocchi = Dim. Cache / Dim. Blocco

• Ogni locazione di memoria corrisponde a una e una sola locazione della cache:

(Ind. Bl.)cache = (Ind. Blocco)mem modulo (Num. blocchi in cache)

Una memoria cache ha la struttura di una normale memoria: il blocco al suo interno è direttamente accessibile tramite indirizzamento

Esempio:

Indirizzo di memoria: N = 32 bit

• Cache a indirizzamento diretto da 64 K Byte e blocco (line) da 128 bit ossia 16 Byte

• Numero di blocchi = Dim. Cache / Dim. Blocco = 64 K Byte / 16 Byte = 4 K blocchi

• Struttura dell’indirizzo di memoria:

• Spiazzamento del byte: B = 4 bit //individua il byte all’interno di una line

• Indice: M = 12 bit //individua la specifica line all’interno della cache

• Etichetta (Tag) : 16 bit //completa l’indirizzo dei dati memorizzati nella line…