

---

# Calcolatori Elettronici

## Sistema di memoria

### *parte seconda*

---

Ing. Gestionale e delle Telecomunicazioni  
A.A. 2008/09  
Gabriele Cecchetti

---

## Sistema di memoria – *parte seconda*

### ■ Sommario:

- Gerarchia di memoria
- Memoria cache
- Schemi di indirizzamento
- Riassunto e particolari
- Prestazioni di cache
- Memoria cache di M68000 e IA-32
- Memoria Virtuale
- Memoria di Massa

### ■ Riferimenti

- C. Hamacher, “Introduzione all’architettura del Calcolatore”, cap. 9, sez. 9.4 e seguenti.

---

Sistema di memoria gerarchico  
Ruolo della memoria cache

# GERARCHIA DI MEMORIA

---

## Concetto di Gerarchia

- Si possono migliorare le prestazioni (la velocità) del calcolatore tramite il sistema di memoria, con il concetto di gerarchia di memoria:
    - diversi livelli di memoria, con tecnologie differenti
    - in modo da ottenere un buon compromesso tra costo / capacità / prestazione (cioè rapidità di accesso)
  - Qui l'attenzione è sul sistema di memoria cache, che ha lo scopo di rendere rapida la memoria.
  - Verso la fine verrà ripresa la memoria virtuale (di pertinenza specialmente del S.O.).
-

---

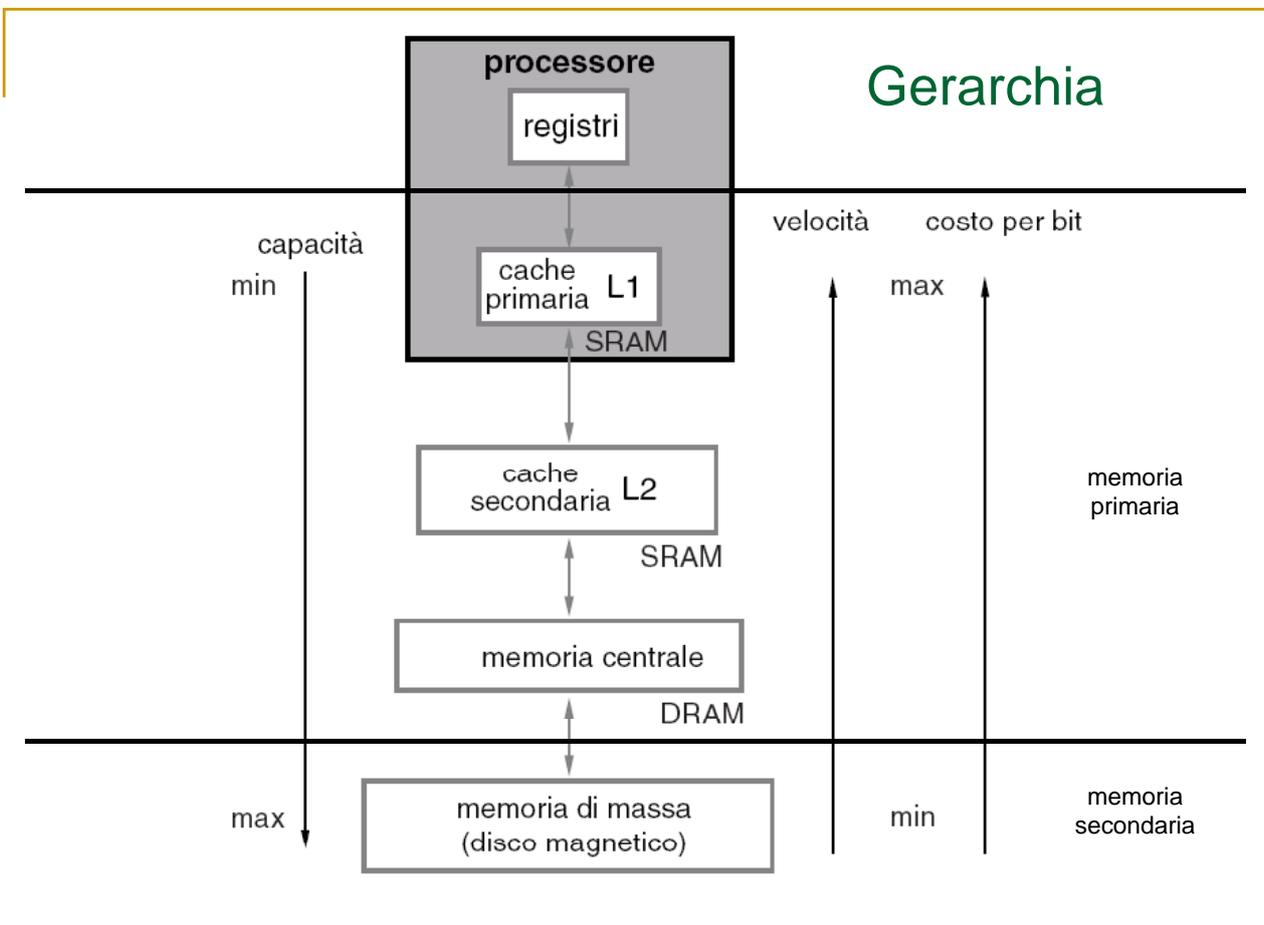
## Tempo di Accesso

- Il tempo necessario per accedere a una parola di memoria (per leggerla o scriverla) si chiama tempo di accesso (alla parola).
  - Esso è un parametro caratteristico della tecnologia di memoria.
  - Talvolta si distingue tra tempo di accesso in lettura e in scrittura (di solito maggiore del primo).
  - La velocità della memoria è maggiore se il tempo di accesso è breve, in generale (tempo di accesso lungo  $\Rightarrow$  memoria lenta).
  - Di solito il tempo di accesso cresce all'aumento della capacità della memoria, pertanto la memoria di grande capacità è lenta.
  - In memorie non accessibili per singola parola, ma solo per blocco di parole (per esempio tutti i tipi di disco, magnetico, ottico, ecc), il tempo di accesso è relativo a lettura o scrittura del blocco.
- 

---

## Gerarchia di Memoria

- **Struttura complessiva della gerarchia:**
    - banco registri di processore:
      - tempo di accesso a una parola brevissimo ( $< 1$  ns)
      - capacità limitatissima (8 - 1024 bit)
    - sistema memoria cache (primo e secondo livello):
      - tempo di accesso a una parola breve ( $\approx 1$  ns)
      - capacità limitata (ordine di grandezza: da Kbyte a Mbyte)
    - sistema di memoria centrale:
      - tempo di accesso a una parola medio (10 - 100 ns)
      - capacità media (ordine di grandezza: da Mbyte - a Gbyte)
    - sistema di memoria di massa:
      - tempo di accesso a un blocco di parole lungo (1 - 10 ms)
      - capacità grande (ordine di grandezza: Gbyte - Terabyte)
-



## Piramide di Memoria

- La gerarchia di memoria ricorda visivamente una piramide stratificata (piramide a gradoni).
- Salendo di strato in strato nella piramide la capacità della memoria diminuisce e la velocità aumenta (e il costo per bit).
- Ogni strato della piramide contiene in copia un sottinsieme (di blocchi o parole) del contenuto dello strato sottostante.
- Idealmente, la velocità e la capacità visibili dell'intera piramide sono quelle dei soli strati in cima e in fondo, rispettivamente.
- Nota bene: la capacità complessiva dell'intera piramide è solo quella dello strato alla base, non la somma delle capacità di tutti gli strati (infatti gli strati sopra contengono solo copie dell'informazione alla base, non nuova informazione).

---

## Scomposizione della Piramide

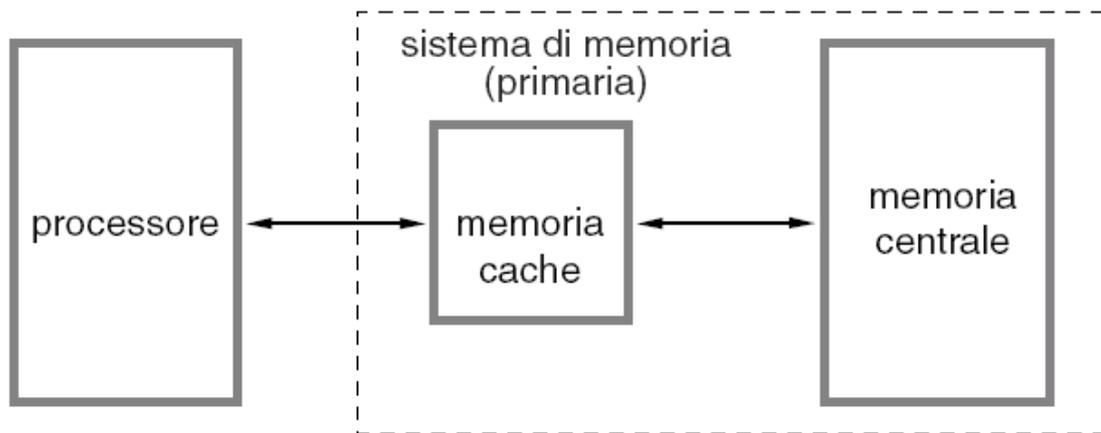
- La gerarchia (o piramide) di memoria si scompone nei tre sottosistemi tecnologici seguenti:
    - registri interni del processore
      - dati e indirizzi dove lavorano correntemente le istruzioni
      - la più veloce tecnologia elettronica disponibile, poco capace
    - memoria primaria
      - programmi in esecuzione e dati in elaborazione corrente
      - serie di tecnologie, tutte elettroniche, veloci, capaci, volatili
    - memoria secondaria
      - programmi e dati in deposito permanente o di uso differito
      - serie di tecnologie, magnetiche od ottiche, lente, molto capaci, persistenti (ma di basso costo per bit memorizzato)
  - La scomposizione è legata alla disponibilità corrente di tecnologie sufficientemente differenziate.
- 

---

## Memoria Primaria

- Motivo e struttura della memoria primaria:
    - il tasso di crescita della velocità del processore è costantemente superiore a quello della memoria
    - la memoria primaria è costituita dall'insieme di memoria centrale e memoria cache:
      - la memoria cache è molto veloce, ma è di capacità ridotta rispetto alla memoria centrale
  - Obiettivo della memoria primaria:
    - fornire al processore (e in definitiva al processo) dati alla velocità con cui esso è in grado di elaborarli
    - fornire al processo una memoria grande e veloce
-

## Memoria Primaria



con memoria primaria si intende quella parte del sistema di memoria realizzata con tecnologie microelettroniche e circuiti integrati: memoria cache e centrale

la memoria secondaria o di massa è realizzata con tecnologie magnetiche

## Memoria Secondaria

- **Motivo e struttura della memoria secondaria:**
  - la capacità della memoria primaria, in particolare la memoria centrale, è limitata dal suo costo, essendo tale forma di memoria interamente elettronica; inoltre essa è volatile (non persistente)
  - la memoria secondaria o di massa è costituita essenzialmente da dischi (magnetici od ottici) e nastri: essa è molto capace e di costo ridotto, ma assai più lenta della memoria centrale
- **Obiettivo della memoria secondaria:**
  - fornire al processo una memoria di capacità (apparentemente) superiore a quella della sola memoria primaria (mem. virtuale)
  - realizzare una forma di memoria persistente (non volatile) dove tenere grandi volumi di dati e programmi (file system)

---

Organizzazione della memoria cache  
Concetti di base  
Principi di località  
Problemi fondamentali

# MEMORIA CACHE

---

## Organizzazione della Cache

- Memoria centrale e cache sono organizzate in blocchi di parole o di byte, di dimensioni uguali. Il blocco di cache si chiama posizione.
  - La posizione di cache può contenere una copia di un blocco di memoria centrale, oppure può essere libera:
    - un blocco di memoria centrale viene caricato in una posizione di cache, vale a dire una regione di cache capace di accogliere esattamente un blocco di memoria centrale
    - a ogni posizione di cache è associato un bit di validità, che indica se la posizione sia occupata da una copia di blocco o se sia libera
  - Il sistema di gestione della memoria cache è in grado di:
    - copiare (caricare) blocchi dalla memoria centrale alla memoria cache
    - ricopiare (scaricare) blocchi dalla memoria cache alla memoria centrale
  - Entrambe le operazioni vengono svolte da un'unità funzionale apposita, l'unità di controllo della cache.
  - Il processore accede sempre prima alla memoria cache.
-

---

## Funzionamento della Cache

(1/2)

- Il processore deve prelevare un'istruzione macchina per eseguirla:
    - se il blocco che contiene l'istruzione si trova già in cache, l'istruzione viene prelevata ed eseguita
    - se l'istruzione non si trova in cache:
      - il processore sospende l'esecuzione
      - il blocco contenente l'istruzione viene caricato dalla memoria centrale in una posizione libera della memoria cache
      - il processore preleva l'istruzione dalla cache e prosegue l'esecuzione
  - Senza cache, il processore preleva l'istruzione sempre e solo da memoria centrale.
- 

---

## Funzionamento della Cache

(2/2)

- Il processore deve leggere un dato da memoria, per elaborarlo (per esempio aritmeticamente):
    - si procede come per il prelievo di istruzione
  - Il processore deve scrivere un dato in memoria (un dato elaborato e più o meno definitivo):
    - si procede in modo simile alla lettura di dato, e si scrive il valore in cache
    - ma si pone il problema della coerenza tra memoria cache e memoria centrale:
      - scrittura immediata anche in memoria centrale, o differita
  - Senza cache, il processore legge o scrive il dato sempre e solo da o in memoria centrale.
-

---

## Principi di Località

- Il programma generico soddisfa, in senso statistico, due importanti principi di località:
    - località spaziale
    - località temporale
  - Il termine "località" si riferisce alla logica con cui:
    - dal dato corrente si passa a un nuovo dato
    - dall'istruzione macchina corrente si passa a quella successiva (sequenziamento)
  - Il sequenziamento è "locale" nel senso che l'istruzione successiva è "vicina" a quella corrente (appartiene a un "intorno").
  - Similmente per il dato ... (vedi di seguito).
- 

---

## Località Spaziale

- Se il dato di indirizzo  $i$  viene letto o scritto, allora prima o poi (ma non di necessità entro breve tempo) è molto plausibile che i dati agli indirizzi vicini a  $i$  ( $i \pm 1$ ,  $i \pm 2$ , ...) vengano anch'essi letti o scritti (ma non di necessità tutti insieme o a breve distanza di tempo l'uno dall'altro).
  - Il principio vale a motivo dell'esistenza di strutture dati aggregate: se un dato della struttura va in uso, gli altri dati della stessa struttura devono pur avere uno scopo correlato e pertanto andranno anch'essi in uso.
-

---

## Località Temporale

- Se l'istruzione macchina di indirizzo  $i$  entra in esecuzione, allora è plausibile che entro breve tempo l'istruzione macchina di indirizzo  $i$  torni di nuovo in esecuzione (e che la cosa si ripeta anche più volte, a distanza di tempo ravvicinata).
  - Il principio vale a motivo dell'esistenza di cicli: se il ciclo è programmato con ragione, è perché si reputa plausibile vada ripetuto almeno qualche volta, in particolare se è corto, e così le istruzioni che esso contiene vengono rieseguite due o più volte a breve distanza di tempo.
- 

---

## Località Spaziale-Temporale

- I due principi hanno carattere sperimentale (non analitico, cioè non sono teoremi dimostrabili matematicamente), ma sono ben verificati.
  - Essi sono logicamente indipendenti, cioè non si può dedurre l'uno dell'altro.
  - Ma spesso si presentano insieme e si possono allora raggruppare in uno solo, più espressivo:
    - se un piccolo blocco di parole consecutive (dati o istruzioni) entra in uso, allora entro breve tempo lo stesso blocco di parole tornerà in uso, anche più volte
  - Si chiama località spaziale-temporale.
-

---

## Località e Cache

- L'uso di memoria cache nel sistema di memoria del calcolatore, con lo scopo di aumentarne la prestazione, si fonda appunto sui due principi di località:
    - località spaziale: vengono trasferiti in cache più parole di quante non ne siano strettamente richieste (posizione o linea di cache)
    - località temporale: viene sfruttata nella scelta del blocco da sostituire quando si verifica un evento di fallimento (o miss) (per esempio: sostituire il blocco cui si è fatto accesso meno di recente, algoritmo LRU)
  - Più in sintesi: il blocco di memoria centrale (cioè la posizione di cache) è appunto quello cui si riferisce il principio di località spaziale-temporale.
  - Caricare tale blocco in cache conviene, perché se entra in uso verrà riusato più volte entro breve tempo.
- 

---

## Hit e Miss

- Evento di hit (successo): accesso in lettura o prelievo / scrittura a parola (dato o istruzione) in una posizione di memoria cache:
    - hit rate (tasso di successo): numero di accessi a memoria che trovano la parola in cache rispetto al numero totale di accessi
    - hit time (tempo di successo): tempo per accedere al dato in cache (= tempo per determinare successo della richiesta + tempo di accesso a cache)
  - Evento di miss (fallimento): accesso fallito in cache, la parola (dato o istruzione) va recuperata nella memoria centrale:
    - miss rate (tasso di fallimento) =  $1 - \text{hit rate}$
    - penalità di miss (tempo di fallimento): tempo per sostituire un blocco in cache + tempo di accesso a parola in cache
-

---

## Quattro Domande

- Metodo di indirizzamento:
    - come scegliere una posizione di memoria cache dove caricare un nuovo blocco di memoria centrale
  - Metodo di identificazione:
    - come localizzare all'interno della memoria cache la posizione dove c'è un blocco di memoria centrale cui si vuole accedere
  - Metodo di scrittura (coerenza):
    - come comportarsi quando si deve modificare (per esempio scriverci dentro) un blocco caricato in una posizione di cache: va scritta anche la memoria centrale o no, e se sì, quando ?
  - Metodo di sostituzione:
    - come si individuano i blocchi da espellere dalla cache quando occorre liberare posizioni al fine di caricarvi nuovi blocchi
- 

---

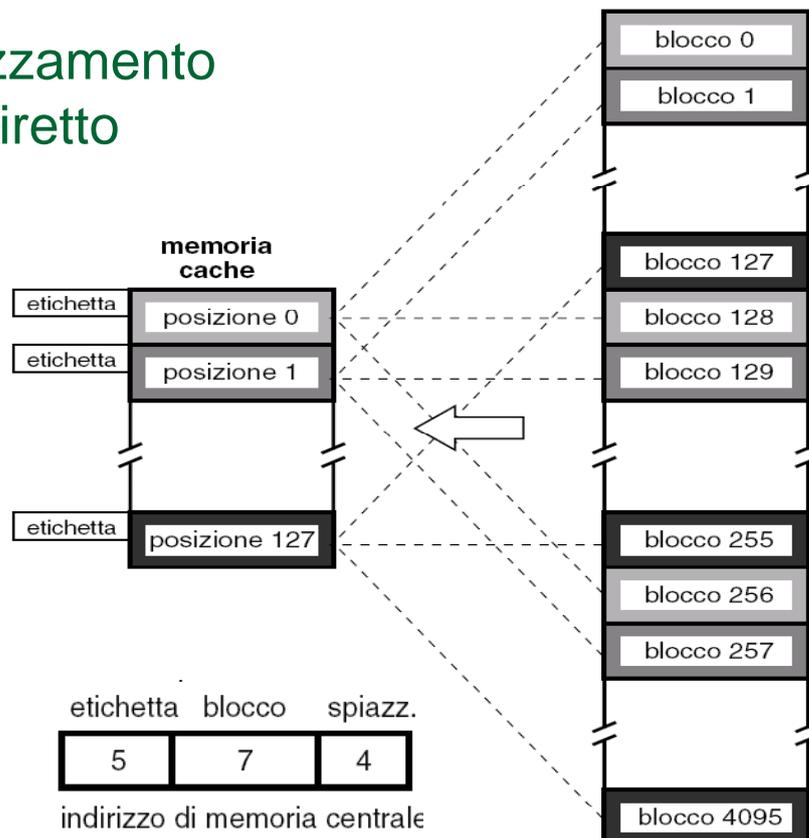
Memoria cache a indirizzamento diretto  
Memoria cache completamente associativa  
Memoria cache associativa a gruppi

## SCHEMI DI INDIRIZZAMENTO

## Indirizzamento Diretto

- Ogni blocco della memoria centrale è caricabile in una sola posizione della memoria cache.
- Si possono caricare più blocchi di memoria centrale nella stessa posizione di cache (non in modo simultaneo) - conflitto
- Il blocco numero  $j$  della memoria centrale è caricabile solo nella posizione numero:  
*resto div. intera di  $j / n^\circ$  posizioni della cache*
- È anche chiamato schema "direct mapped".

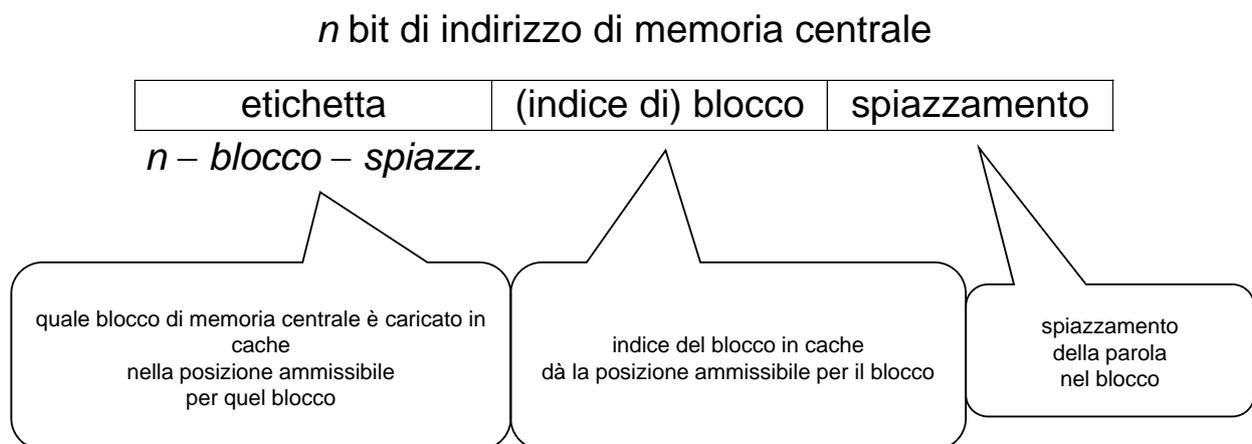
## Indirizzamento Diretto



## Dimensionamento

- Indirizzo di memoria centrale:  $n = 16$  bit
- Indirizzo di memoria cache:  $m = 11$  bit
- Taglia in parole di blocco e posizione (sono uguali):  $b = 16$  parole
- Numero di blocchi di memoria centrale:  
 $2^{16}$  parole / 16 parole =  $2^{12} = 4096$  blocchi
- Numero di posizioni di memoria cache:  
 $2^{11}$  parole / 16 parole =  $2^7 = 128$  posizioni
- Spiazzamento di blocco:  
 $\log_2 16$  parole = 4 bit
- Numero (o indice) di blocco:  
 $\log_2 128$  parole = 7 bit
- Etichetta:  
 $16$  bit – 7 bit – 4 bit = 5 bit

## Scomposizione di Indirizzo



la scomposizione è effettuata dall'unità di controllo della cache  
essa si serve dei campi di indirizzo per gestire la cache

## Formulario Essenziale

### Legenda:

$n$  = # bit di ind. di mem. centrale

$m$  = # bit di ind. di mem. cache

$b$  = dim. in parole del blocco

# blocchi di memoria =  $\lceil 2^n / b \rceil$

# posizioni di cache =  $\lceil 2^m / b \rceil$

spiazzamento =  $\lceil \log_2 b \rceil$

blocco =  $\lceil \log_2 (\# \text{ posizioni}) \rceil$

etichetta =  $n - \text{spiazz.} - \text{blocco}$

## Identificazione

- Ogni posizione deve memorizzare l'etichetta di blocco.
- Ogni posizione di cache comprende:
  - bit di validità, che indica se la posizione contenga o meno un blocco valido; all'inizio tutte le posizioni sono invalide
  - campo etichetta, contiene il valore che identifica il blocco di memoria correntemente in cache, tra quelli ammissibili
  - campo dati, contenente copia del blocco
- L'accesso a cache avviene nel modo seguente:
  - il campo indice di blocco dell'indirizzo viene usato per selezionare la posizione in cache dove è caricato il blocco
  - se la posizione è valida, il campo etichetta dell'indirizzo viene confrontato con l'etichetta della posizione selezionata
  - se il confronto ha esito positivo significa che il blocco è presente in cache, e il campo spiazzamento dell'indirizzo viene utilizzato per accedere alla parola interessata

---

## Gestione di Hit e Miss

- Successo nella lettura:
    - la parola è presente in cache e lì viene letta
  - Fallimento nella lettura:
    - stallo della CPU e richiesta alla memoria centrale del blocco contenente la parola, poi
    - copia del blocco in cache e al termine ripetizione dell'operazione di lettura della parola in cache (talvolta è chiamata lettura differita o *read-back*)
  - È possibile (ma difficile) leggere la parola "al volo", non appena essa arriva alla cache, senza aspettare che il blocco sia arrivato tutto (lettura immediata o *load-through* o *early-restart*).
- 

---

## Gestione di Hit e Miss

- Successo nella scrittura:
    - scrittura della parola sia in cache sia in memoria centrale (scrittura immediata o *write-through*)
    - scrittura della parola solo nella cache (scrittura differita o *write-back*), la copia in memoria centrale avviene in un secondo momento
  - Fallimento nella scrittura:
    - stallo della CPU e richiesta alla memoria centrale del blocco contenente la parola cercata, poi
    - copia in cache del blocco e al termine ripetizione dell'operazione di scrittura della parola in cache
-

---

## Sostituzione di Blocco

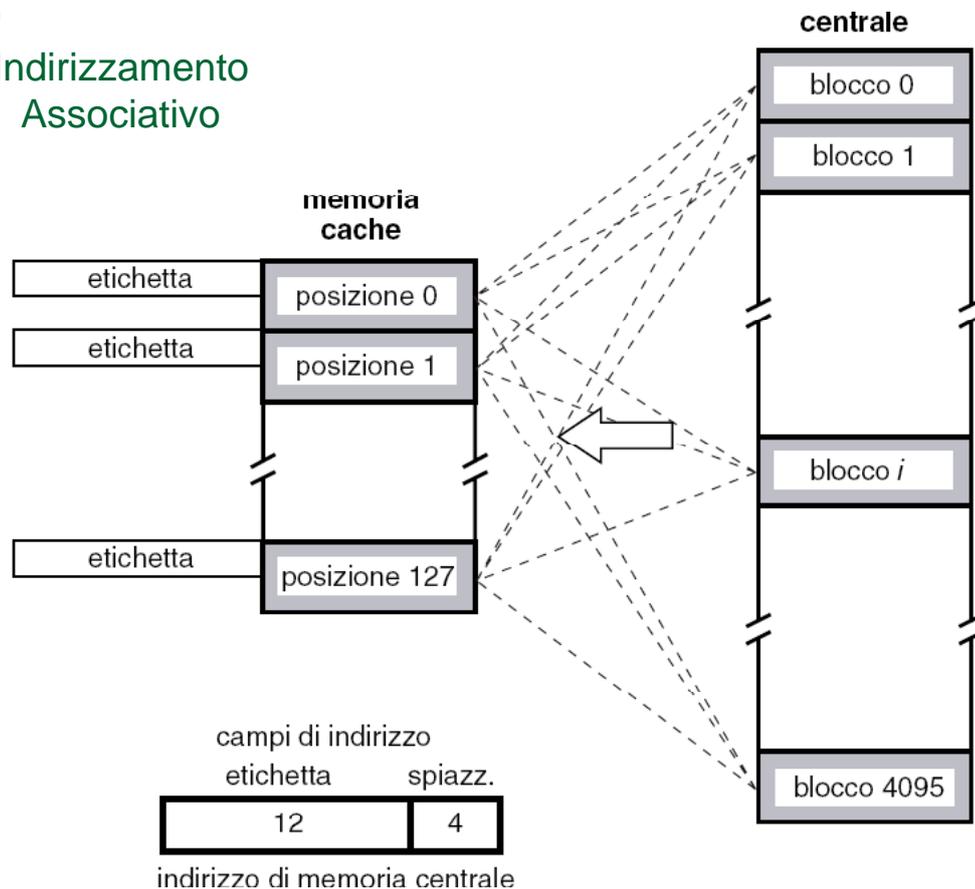
- Come si sostituiscono blocchi correntemente in cache per liberare posizioni dove caricare nuovi blocchi di memoria centrale ?
  - È un problema risolto in modo automatico dallo schema di indirizzamento diretto, giacché non è possibile fare alcuna scelta:
    - il blocco può andare una sola posizione
  - Negli schemi di indirizzamento successivi (di tipo associativo), la sostituzione di blocco sarà invece un problema rilevante, da risolvere.
- 

---

## Indirizzamento Associativo

- Si può caricare un blocco di memoria centrale in una posizione qualunque della cache (pochi conflitti).
  - Non esiste una relazione definita tra indice di blocco e posizione in cache.
  - Pertanto non esiste neppure alcun problema di corrispondenza tra blocchi di memoria centrale e posizioni di cache dove caricarli.
  - Struttura dell'indirizzo di memoria di  $n$  bit, con blocchi di memoria centrale e posizioni di cache da  $2^m$  parole:
    - $m$  bit meno significativi dell'indirizzo individuano la parola all'interno del blocco e della posizione di cache
    - $n - m$  bit più significativi: etichetta (lunga)
  - È anche chiamato schema "*fully associative*".
-

## Indirizzamento Associativo



## Dimensionamento

- Indirizzo di memoria centrale:  $n = 16$  bit
- Indirizzo di memoria cache:  $m = 11$  bit
- Taglia in parole di blocco e posizione (sono uguali):  $b = 16$  parole
- Numero di blocchi di memoria centrale:  
 $2^{16}$  parole / 16 parole =  $2^{12} = 4096$  blocchi
- Numero di posizioni di memoria cache:  
 $2^{11}$  parole / 16 parole =  $2^7 = 128$  posizioni
- Spiazzamento di blocco:  
 $\log_2 16$  parole = 4 bit
- Etichetta:  
16 bit – 4 bit = 12 bit
- Non esiste indice di blocco (0 bit !)

---

## Formulario Essenziale

### Legenda:

$n, m, b$  = come per cache a indirizz. di tipo diretto

# blocchi di memoria =  $\lceil 2^n / b \rceil$  e # posizioni =  $\lceil 2^m / b \rceil$

spiazzamento =  $\lceil \log_2 b \rceil$

etichetta =  $n - \text{spiazzamento}$

---

---

## Identificazione

- La ricerca di una parola nella cache richiede il confronto di tutte le etichette presenti in cache con l'etichetta del blocco di memoria centrale richiesto.
  - Per aumentare le prestazioni la ricerca avviene in parallelo tramite una memoria associativa (componente integrato di costo elevato, vedi di seguito).
  - In caso di fallimento della ricerca è necessario caricare il blocco dalla memoria centrale in cache.
  - Se la cache è piena è necessario sostituire il contenuto di una posizione occupata, prima liberandola.
  - Sostituzione del contenuto della posizione:
    - scelta casuale della posizione da sostituire
    - scelta della posizione utilizzata meno di recente (LRU)
-

---

## Algoritmo LRU

- Sostituisce il contenuto della posizione di cache di cache usata meno di recente (least recently used, LRU).
  - La posizione usata meno di recente è quella cui non si accede da più tempo.
  - Occorre un sistema di annotazione per tenere traccia dell'istante di tempo di ultimo uso delle posizioni di cache.
  - Il sistema di annotazione non deve essere necessariamente preciso, ma può essere più o meno approssimato (vedi esempi di seguito).
- 

---

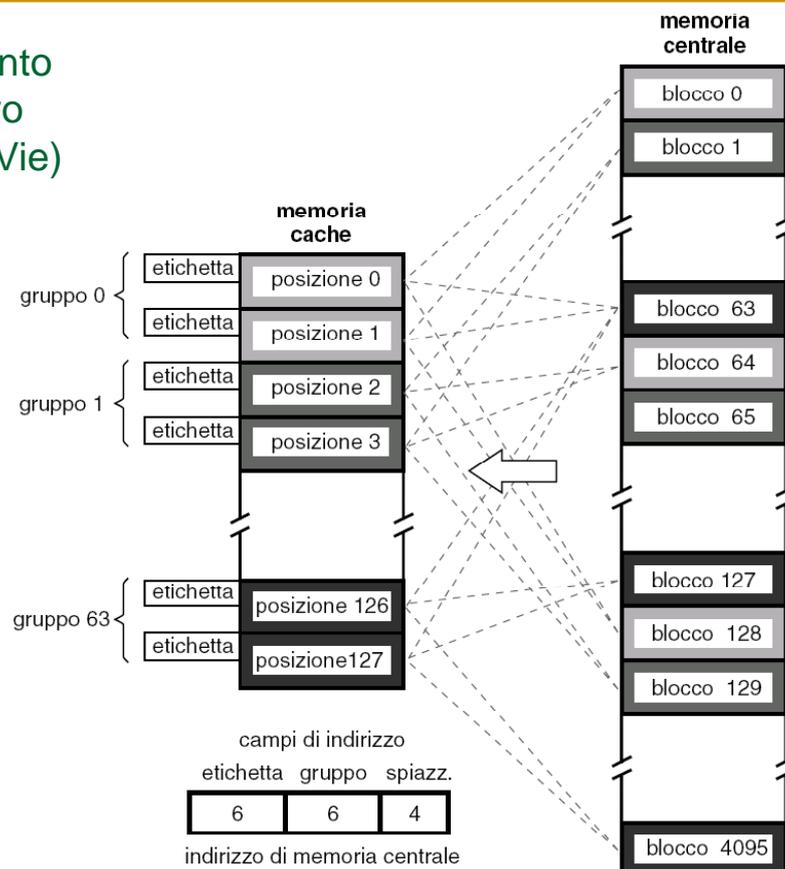
## Indirizzamento Associativo a Gruppi

- Le posizioni della cache sono ripartite in gruppi (insiemi).
  - Si indirizza la cache per gruppo (non per posizione).
  - Ogni gruppo contiene esattamente  $g$  posizioni di cache.
  - Si può caricare un blocco della memoria centrale in un solo gruppo (prefissato) di posizioni, ma in una qualsiasi delle  $g$  posizioni appartenenti al gruppo (cioè le posizioni appartenenti allo stesso gruppo sono equivalenti).
  - Una cache associativa con gruppi da  $g$  posizioni, ovvero una cache dove si possa caricare un blocco di memoria centrale a scelta in una tra  $g$  posizioni (tutte nel medesimo gruppo), si dice "a  $g$  vie".
  - È anche chiamato schema "set associative".
-

## Cache Associativa a 2 Vie

- Di seguito si fa l'esempio di cache associativa a 2 vie.
- Cache a due vie: ogni gruppo è costituito da due posizioni (ogni gruppo è capace di accogliere fino a due blocchi di memoria centrale).
- Il campo numero o indice di gruppo dell'indirizzo (di memoria centrale) seleziona il gruppo dove si deve lavorare, ovvero le due posizioni che lo costituiscono.
- Il campo etichetta dell'indirizzo (di memoria centrale) viene confrontata in parallelo con le due etichette associate alle due posizioni appartenenti al gruppo.
- La posizione contenente il blocco di memoria centrale cercato viene selezionata in base all'esito dei confronti.

### Indirizzamento Associativo a Gruppi (2 Vie)



## Dimensionamento

- Indirizzi di memoria centrale e cache:  $n = 16$  bit e  $m = 11$  bit
- Taglia in posizioni del gruppo  $v = 2$  posizioni per gruppo (2 vie)
- Taglia in parole di blocco e posizione (sono uguali):  $b = 16$  parole
- Numero di blocchi di memoria centrale:  
 $2^{16}$  parole / 16 parole =  $2^{12} = 4096$  blocchi
- Numero di posizioni di memoria cache:  
 $2^{11}$  parole / 16 parole =  $2^7 = 128$  posizioni
- Numero di gruppi nella cache:  
 $128$  posizioni / 2 posizioni per gruppo = 64 gruppi
- Spiazzamento di blocco:  
 $\log_2 16$  parole = 4 bit
- Numero (o indice) di gruppo:  
 $\log_2 64$  gruppi = 6 bit
- Etichetta:  
 $16$  bit – 6 bit – 4 bit = 6 bit

## Formulario Essenziale (a $v$ Vie)

### Legenda:

$n, m, b$  = come per cache a indirizz. di tipo diretto

$v$  = dim. in posizioni del gruppo = # vie della cache

# blocchi =  $\lceil 2^n / b \rceil$  e # posizioni =  $\lceil 2^m / b \rceil$

# gruppi = # posizioni /  $v$

spiazzamento =  $\lceil \log_2 b \rceil$  e gruppo =  $\lceil \log_2 (\# \text{ gruppi}) \rceil$

etichetta =  $n$  – spiazzamento – gruppo

---

## Identificazione e Sostituzione

- L'identificazione della posizione in cache contenente il blocco dove lavorare procede come nel caso di schema associativo.
  - Tuttavia la ricerca della posizione va ristretta al solo gruppo di appartenenza, non all'intera memoria cache.
  - In caso di sostituzione, per individuare la posizione di cache da liberare bisogna ricorrere all'algoritmo LRU, ma di nuovo restringendone l'applicazione al gruppo.
- 

---

Riassunto dei punti essenziali  
Memoria associativa  
Memoria cache a due livelli

## RIASSUNTO E PARTICOLARI

---

## Riassunto - Indirizzamento

- Indirizzamento diretto:
    - posizione univoca: indirizzo di memoria modulo numero delle posizioni in cache
  - Indirizzamento completamente associativo:
    - posizione qualunque all'interno della cache
  - Indirizzamento associativo a gruppi:
    - identificazione univoca del gruppo
    - posizione qualsiasi all'interno del gruppo prefissato
- 

---

## Riassunto - Identificazione

- Indirizzamento diretto:
    - identifica la posizione dove lavorare
    - verifica il bit di validità e l'etichetta della posizione
  - Indirizzamento completamente associativo:
    - confronta le etichette di tutte le posizioni di cache che hanno bit di validità attivo
  - Indirizzamento associativo a gruppi:
    - identifica il gruppo dove lavorare
    - confronta le etichette di tutte le posizioni del gruppo identificato, che hanno bit di validità attivo
-

---

## Riassunto - Sostituzione

- Cache a indirizzamento diretto:
    - posizione definita univocamente dall'indirizzo
  - Cache associativa e associativa a gruppi:
    - posizione scelta tramite algoritmo casuale
    - scelta tramite algoritmo LRU (*Least Recently Used*)
  - Algoritmo LRU:
    - cache a 2 vie:
      - c'è un bit di uso per ogni blocco del gruppo
      - quando un blocco è utilizzato il suo bit è messo a 0, l'altro a 1
    - cache a 4 vie:
      - c'è un contatore di uso per ogni blocco del gruppo
      - quando un blocco è utilizzato il suo contatore è messo a 0, gli altri incrementati di 1
- 

---

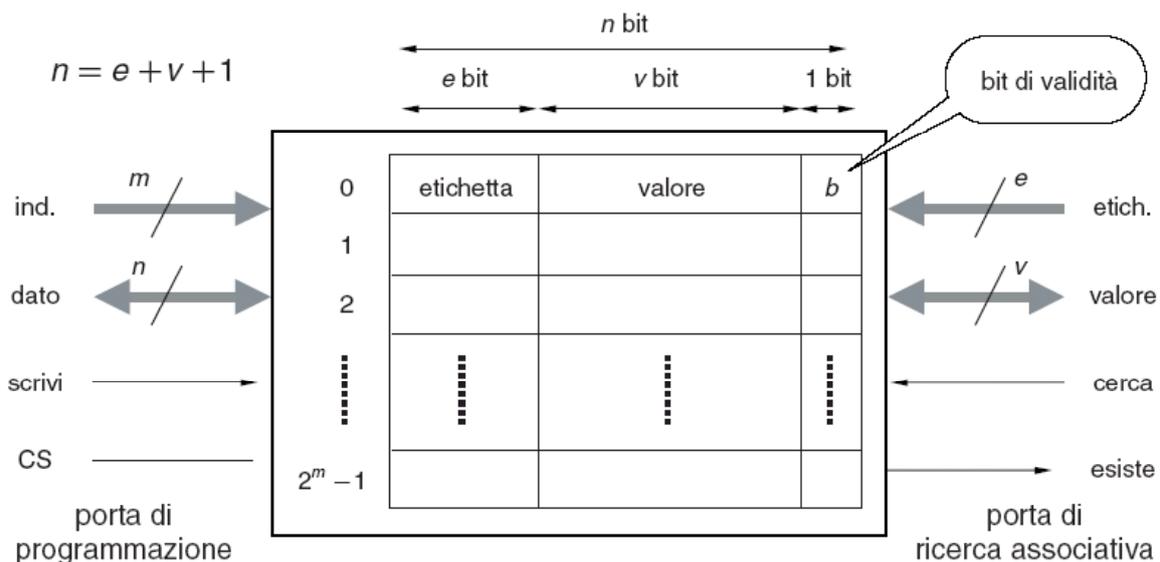
## Riassunto - Coerenza

- Write-through o scrittura immediata:
    - l'informazione viene scritta subito sia nella posizione di livello superiore (cache) sia nel blocco di livello inferiore (memoria centrale)
  - Write-back o scrittura differita:
    - l'informazione viene scritta subito solo nella posizione di livello superiore (cache)
    - il blocco di livello inferiore (memoria centrale) verrà aggiornato solo quando sarà avvenuta la sostituzione del blocco a livello superiore (cache)
-

# Memoria Associativa

- Per le etichette di blocco in cache, occorre un componente speciale: la memoria associativa (da non confondere con la cache associativa).
- Essa è dotata di due porte di accesso:
  - porta di programmazione, che funziona come una normale porta di accesso a memoria RAM
  - porta di ricerca associativa: riceve l'etichetta e in parallelo la ricerca nella memoria; se la trova emette o acquisisce il campo dati associato
- La memoria associativa è realizzata con tecnologie microelettroniche ad hoc.

## Memoria Associativa



---

## Ricerca Associativa

- Si usa la porta di ricerca associativa.
  - Si manda alla memoria l'etichetta da ricercare (formata da  $e$  bit) e si attiva il comando "cerca".
  - La memoria ricerca, in tutte le righe valide e in parallelo, se l'etichetta inviata sia presente o no.
  - Se sì, la memoria emette il campo "valore" (formato da  $d$  bit) associato all'etichetta trovata e attiva il segnale "esiste".
  - Se no, la memoria disattiva il segnale "esiste".
  - Si suppone che non ci siano mai etichette duplicate (comunque si potrebbe rimediare).
- 

---

## Memoria Cache a Due Livelli

- La memoria primaria contiene due cache, collegate in cascata e interposte tra processore e memoria centrale.
  - Normalmente il processore opera (legge e scrive) solo nella cache di primo livello (la più veloce delle due).
  - Se c'è miss, passa alla cache di secondo livello.
  - Se c'è ancora miss, passa alla memoria centrale.
  - Va gestito il caricamento di blocchi in cache di primo e secondo livello (più complicato che con un solo livello).
  - Il sistema a 2 livelli ha senso se ambo le cache hanno tempo di accesso inferiore a quello di memoria centrale.
  - Inoltre la cache di primo livello deve essere abbastanza più veloce di quella di secondo livello.
  - Si può fare se la tecnologia di cache è differenziata.
-

---

Analisi di prestazione di memoria cache

# PRESTAZIONE DI CACHE

---

## Prestazione - Senza Cache

- La memoria centrale è caratterizzata da un tempo di accesso (esatto):

$H$  tempo (esatto) per leggere o scrivere una sola parola

- Il tempo totale  $T$  di accesso a  $n$  parole di memoria vale:

$T = n \times H$  tempo totale per leggere o scrivere  $n$  parole

- Talvolta il tempo di accesso in lettura differisce da quello in scrittura (generalizza come ovvio).
-

---

## Prestazione - Una Sola Cache

- Se il sistema di memoria primaria comprende cache e memoria centrale, ci sono due tempi di accesso: di memoria centrale e di cache.
  - Il tempo di accesso esatto va allora sostituito con un tempo medio di accesso:  $\Delta T_{acc}$ .
  - Il tempo medio di accesso tiene conto dell'uso combinato di cache e memoria centrale.
  - Esso ha valore intermedio tra il tempo di accesso di cache e di memoria centrale.
- 

---

## Prestazione - Una Sola Cache

$$\Delta T_{accesso} = hH + (1 - h) M$$

$h$  tasso di hit o di successo

$H$  tempo di accesso in cache (per una sola parola)

$M$  penalità di miss o di fallimento  
(tempo di caricamento di un blocco in cache)

$\Delta T_{acc}$  tempo medio di accesso al sistema di memoria primaria  
con cache (per una sola parola)

ricorda che  $m$  (tasso di miss) =  $1 - h$  (tasso di hit)

i tassi  $h$  e  $m$  vanno normalizzati a 1 ( $0 \leq h, m \leq 1$ )

---

---

## Guadagno - Una Sola Cache

$h = 0,95$       95 % di eventi di hit

$H_c = 1 \text{ ns}$       un nanosecondo - accesso a una sola parola in cache

$H_m = 100 \text{ ns}$       100 nanosecondi - acc. a una sola par. in mem. centr.

$M = 1 \text{ }\mu\text{s}$       un microsecondo - per caricare in cache un blocco

$$\Delta T_{\text{acc}} = 0,95 \times 1 \text{ ns} + (1 - 0,95) \times 1 \text{ }\mu\text{s} =$$

$$= 0,95 \text{ ns} + 0,05 \times 1000 \text{ ns} =$$

$$= 0,95 \text{ ns} + 50 \text{ ns} =$$

$$= 5,95 \text{ ns} \approx 56 \text{ ns}$$

$$\text{guadagno} = H_m / \Delta T_{\text{acc}} \approx 100 \text{ ns} / 56 \text{ ns} \approx 1,78$$

Con la cache il sistema di memoria accelera di circa il 78%.

---

---

## Prestazione - Cache Istruzioni e Dato

- Si può perfezionare la memoria primaria, prevedendo due cache separate, una per le istruzioni e una per i dati.
  - La cache istruzioni funziona in lettura (salvo quando si carica in cache un blocco di istruzioni).
  - La cache dati funziona sia in lettura sia in scrittura.
  - Le due cache non devono essere di necessità identiche, né per capacità, né per schema di indirizzamento, né per modo di gestione dell'evento di miss.
  - Avere due cache separate permette di sfruttare meglio le proprietà di località di istruzioni e dati, le quali possono presentare qualche differenza.
-

---

## Prestazione - Cache Istr. e Dato

$$\Delta T_{\text{acc istr}} = H_{\text{acc istr}} h_{\text{istr}} + m_{\text{istr}} M_{\text{penalità istr}}$$

$$\Delta T_{\text{acc dato}} = H_{\text{acc dato}} h_{\text{dato}} + m_{\text{dato}} M_{\text{penalità dato}}$$

$$T_{\text{totale}} = \# \text{ acc. istr.} \times \Delta T_{\text{acc istr}} + \# \text{ acc. dato} \times \Delta T_{\text{acc dato}}$$

ricorda che:  $\# \text{ tot. accessi} = \# \text{ acc. istr.} + \# \text{ acc. dato}$

---

---

## Guadagno - Cache Istr. e Dato

$h_{\text{istr}} = 0,95$                       95% di eventi di hit di istruzione

$h_{\text{dati}} = 0,95$                       90% di eventi di hit di dato

$H = 1$  ciclo tempo accesso di istr. o dato in cache                      - sono  
identici

10 cicli                      tempo accesso di istr. o dato in mem.  
centr. - sono identici

$M = 17$  cicli penalità di miss - per caricare in cache  
un blocco di istr. o dati

numero di accessi a memoria: 130 (100 istruzioni e 30 dati)

Nota bene: i tempi sono misurati in numero di cicli di bus.

---

## Guadagno - Cache Istr. e Dato

$g = \frac{\text{tempo tot. acc. senza cache}}{\text{tempo tot. acc. con cache}}$

$$\begin{aligned} g &= \frac{\text{tempo di accesso del sistema di memoria senza cache}}{\text{tempo di accesso del sistema di memoria con cache}} = \\ &= \frac{130 \times 10}{100 (h_{istr.} H + (1 - h_{istr.}) M) + 30 (h_{dati} H + (1 - h_{dati}) M)} = \\ &= \frac{130 \times 10}{100 (0,95 \times 1 + 0,05 \times 17) + 30 (0,9 \times 1 + 0,1 \times 17)} \approx 5,04 \end{aligned}$$

Il sistema di memoria primaria con due cache separate istruzioni e dati guadagna in prestazione di circa il 500%.

## Guadagno - Cache Ideale e Reale

- Si può rifare il calcolo precedente, supponendo che entrambe le cache (istruzioni e dati) abbiano tasso di evento di hit pari a 1 (non c'è mai fallimento).
- È una situazione di cache ideale. Si confronta la cache ideale con quella reale (con i tassi di hit di prima).

$$\begin{aligned} g &= \frac{\text{tempo di accesso del sistema di memoria con cache reale}}{\text{tempo di accesso del sistema di memoria con cache ideale}} = \\ &= \frac{100 (0,95 \times 1 + 0,05 \times 17) + 30 (0,9 \times 1 + 0,1 \times 17)}{130 \times 1} \approx 1,98 \end{aligned}$$

- Il sistema di memoria primaria con due cache ideali guadagna solo circa il 98% rispetto alle cache reali.

## Prestazione - Cache a Due Livelli

- La formula di prestazione di cache è facilmente estensibile al caso di cache a due livelli.
- Bisogna conoscere i tassi di hit dei vari livelli di cache, e i relativi tempi di accesso per una parola, nonché la penalità di fallimento (per caricare un blocco in cache).

$$\Delta T_{\text{accesso}} = h_1 H_1 + (1 - h_1) h_2 H_2 + (1 - h_1) (1 - h_2) M$$

- $h_1$  – tasso di hit nella memoria cache di primo livello L1.
  - $h_2$  – tasso di hit nella memoria cache di secondo livello L2.
  - $H_1$  – tempo per avere accesso alla memoria cache di primo livello L1.
  - $H_2$  – tempo per avere accesso alla memoria cache di secondo livello L2.
  - $M$  – tempo per avere accesso alla memoria centrale (penalità di miss).
- Si veda l'esempio numerico sul testo.

Esempi di sistema di memoria cache  
per processori M68000 e IA-32

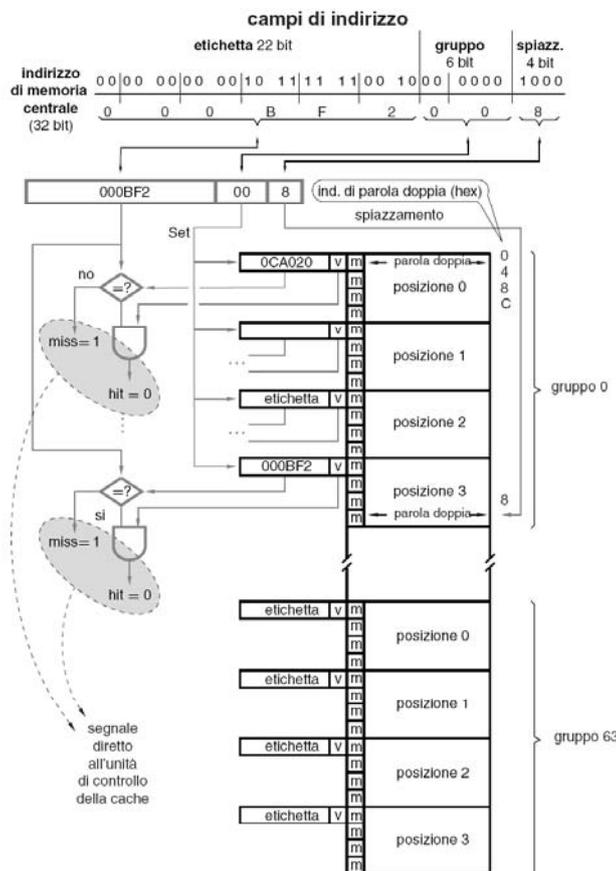
## MEMORIA CACHE M68000 E IA-32

# Cache di M68040

- Il processore Motorola 68040 (versione potenziata di M68000) dispone di una cache istruzioni e dati unificata.
- È una memoria cache associativa a gruppi, a 4 vie (con 4 posizioni per gruppo), e con 64 gruppi in totale.
- La posizione di cache (= blocco di memoria centrale) è costituita da 4 parole da 32 bit (dunque 16 byte in totale).
- La capacità totale della cache è pari a 64 gruppi × 4 posizioni × 16 byte = 4 K byte, mentre la capacità max di memoria centrale è di 4 G byte (bus indirizzi da 32 bit).
- Tale memoria cache è ospitata sullo stesso componente integrato (chip) del processore.

esempio di analisi di indirizzo  
 spiaz. = 4 bit  
 gruppo = 6 bit  
 etichetta = 22 bit

circuiti di rilevamento di hit/miss  
 (unità di controllo)

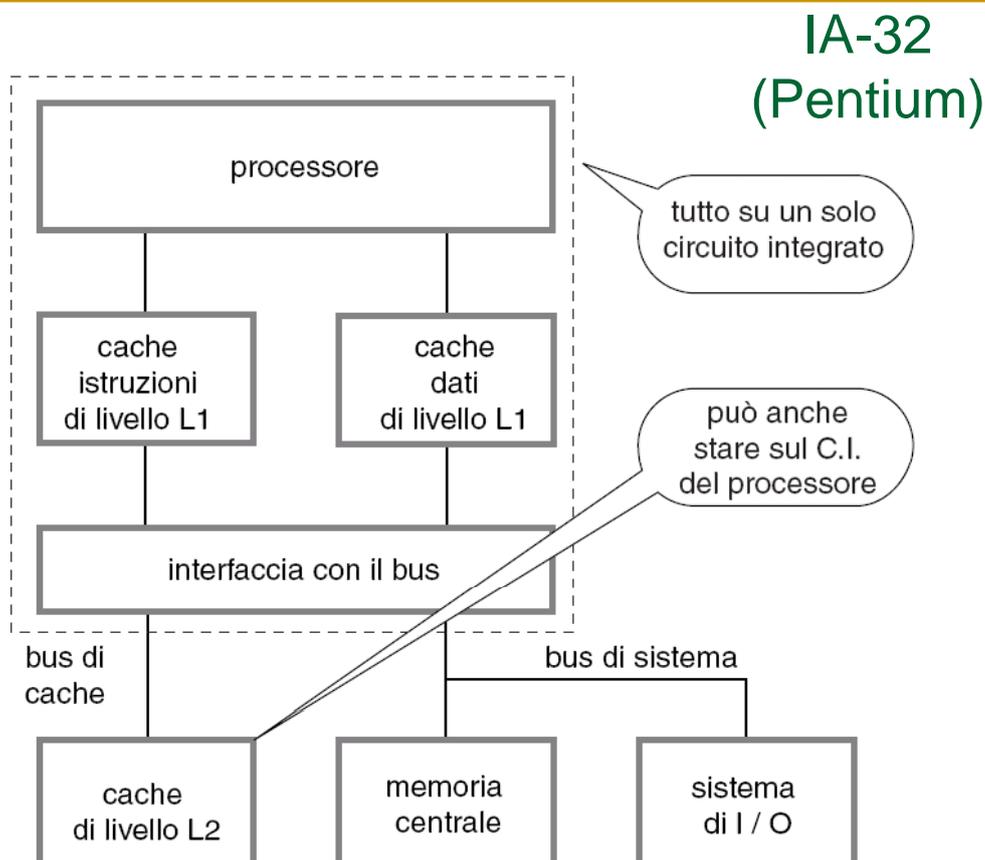


ogni posizione ha 4 bit di modifica

**M68040**

## Cache di IA-32 (Pentium)

- I processori Intel a 32 bit, di serie Pentium, sono dotati di un sistema di cache sofisticato, a due livelli (ci sono alcune varianti secondo il modello di Pentium).
- La cache di primo livello è divisa in cache dati e cache istruzioni, entrambe ospitate sullo stesso componente integrato del processore, per rendere veloce l'accesso.
- La cache di secondo livello è una sola e di solito è esterna al processore, ma in alcune versioni di Pentium è ospitata nello stesso involucro del processore (tuttavia non è integrata sulla stessa piastrina di silicio).
- L'intero sistema di memoria cache è piuttosto articolato e di grande dimensione. Si veda il testo per i particolari.



---

Struttura del sistema di memoria virtuale  
Relazione con memoria cache  
Tabella delle pagine e TLB  
Unità di gestione della memoria (MMU)

## MEMORIA VIRTUALE

---

### Memoria Virtuale

- La memoria virtuale costituisce l'insieme di funzioni incaricate di mostrare al processore un sistema di memoria di capacità superiore alla quantità di memoria primaria (fisica) effettivamente disponibile.
  - Essa si basa sull'idea di scaricare parte dei processi (codice eseguibile e dati) in memoria secondaria (disco magnetico) e così di liberare spazio in memoria primaria.
  - La sua funzione principale è quella di tradurre l'indirizzo virtuale (lungo) generato dal processore nell'indirizzo fisico (corto) destinato al sistema di memoria primaria.
  - La funzione di traduzione si basa in modo essenziale sul meccanismo di suddivisione della memoria in pagine.
  - La memoria virtuale è gestita direttamente dal S.O.
-

---

## Relazione con Cache

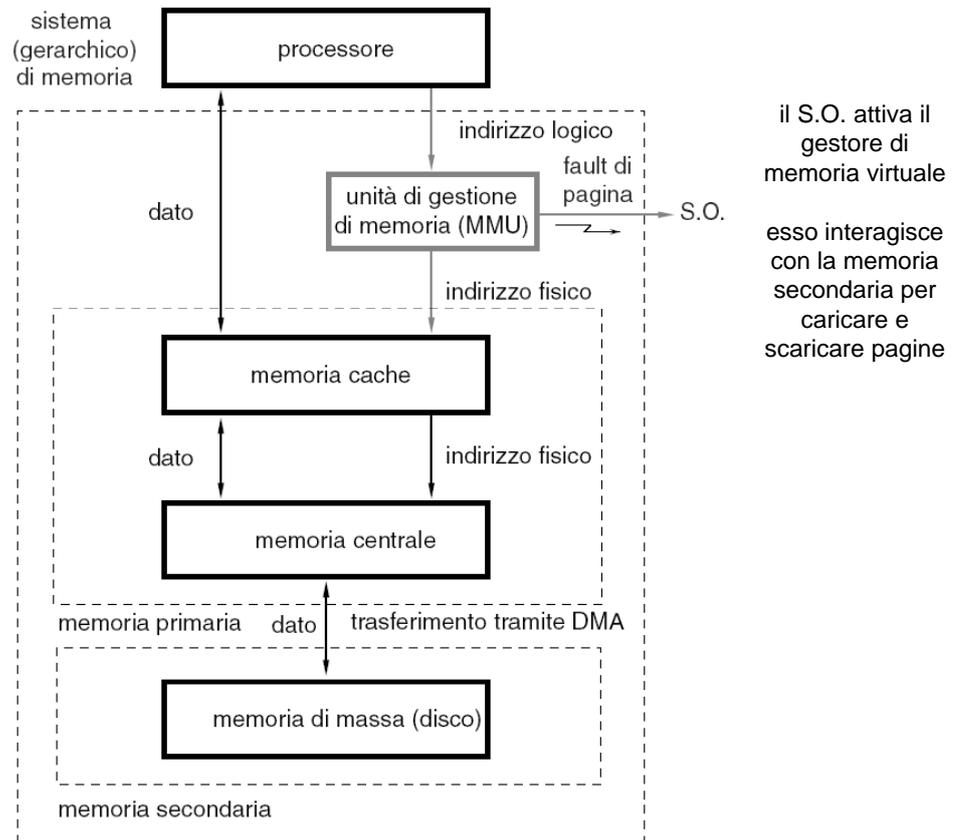
- La paginazione è realizzata dall'unità di gestione di memoria virtuale (*memory management unit* o MMU).
  - La MMU è interposta tra processore (o in generale il MASTER del sistema) e sistema di memoria primaria.
  - La MMU traduce l'indirizzo logico generato dal processore in indirizzo fisico destinato al sistema di memoria primaria (memoria centrale e memoria cache).
  - La MMU genera il segnale di "page fault" (di fatto è un'interruzione diretta al S.O.) quando occorre cambiare numero e dislocazione delle pagine di memoria.
  - Pertanto il processore interagisce con la memoria virtuale, ma la memoria primaria riceve direttamente l'indirizzo fisico ed è indipendente dalla memoria virtuale.
- 

---

## Relazione con Cache

- La memoria cache non è gestita direttamente dal S.O., come avviene per la memoria virtuale.
  - Il segnale di hit/miss va servito alla massima velocità possibile e in modo trasparente rispetto al processore.
  - Non è possibile gestire tale segnale come se fosse un'interruzione, con intervento di una routine SW.
  - Pertanto tale segnale viene interpretato direttamente dall'unità di controllo della cache, che è interamente HW.
  - Il S.O. non ha ruolo nella gestione della cache, se non per configurarla inizialmente (se essa è configurabile).
-

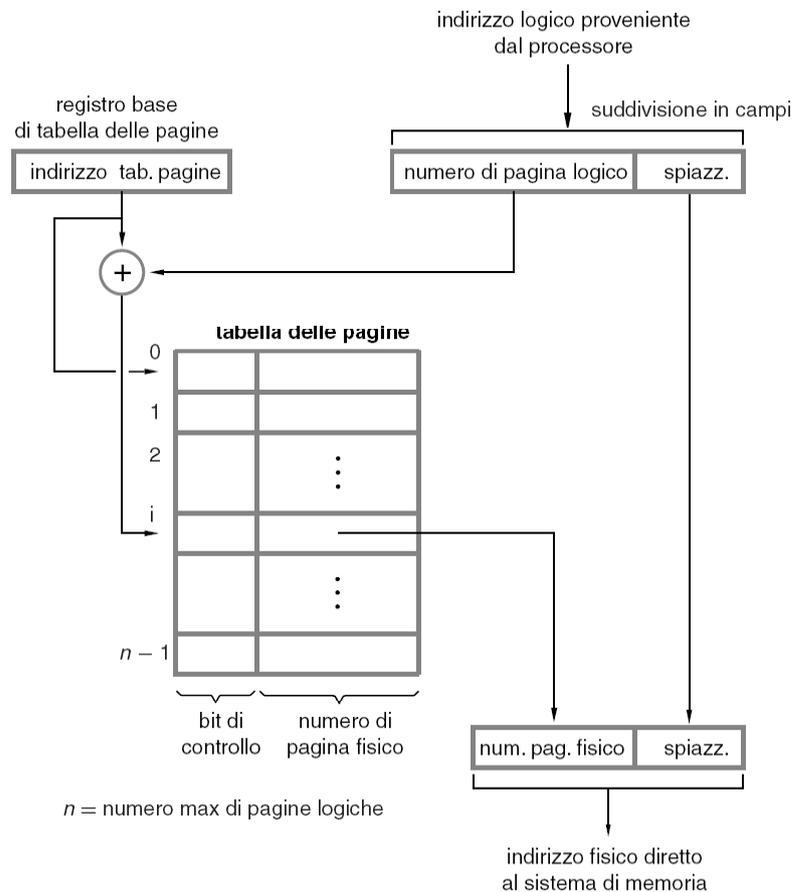
## MMU e Cache



## Tabella delle Pagine

- La MMU contiene la tabella delle pagine corrente, la quale in ogni riga contiene gli elementi seguenti:
  - il numero di pagina fisico (ind. iniziale della pagina in memoria)
  - bit di controllo della pagina fisica (lettura, modifica, validità, ecc.)
- L'indice di riga della tabella è il numero di pagina logico ( $n$  pagine logiche in totale).
- La figura seguente mostra la funzione di traduzione, nelle linee essenziali.
- Se la tabella delle pagine è molto grande, solo la parte attiva corrente sta nella MMU, il resto è in memoria primaria e viene caricato in MMU quando serve.
- Circuiti di controllo associati generano il segnale di "page fault" quando si verifica manco di pagina.

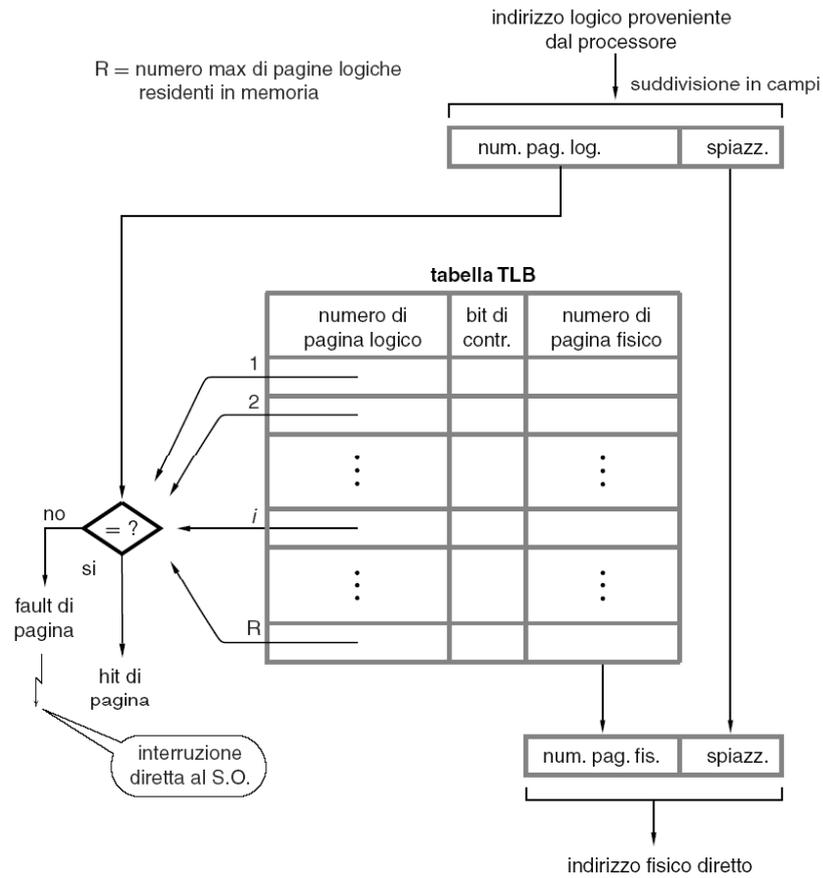
## Tabella delle Pagine



## Tabella TLB

- La MMU è un dispositivo complesso è importante, e pertanto può incorporare varie ottimizzazioni strutturali.
- Tra le tante, l'uso di una tabella di pagine organizzata come TLB (translation lookaside buffer).
- La tabella TLB elenca solo le pagine in uso da parte del processo correntemente in esecuzione.
- Ogni riga della tabella TLB contiene anche il numero di pagina logico (oltre a quello fisico). La pagina logica viene ricercata nella tabella TLB in modo associativo.
- La tabella TLB è essenzialmente una memoria di tipo associativo (come già visto per le etichette di cache).
- Per i particolari si rimanda al testo e alla teoria dei S.O.

## Tabella TLB



Tecnologia di disco e nastro magnetico  
Relazione con la gerarchia di memoria e con la memoria virtuale

## MEMORIA DI MASSA

---

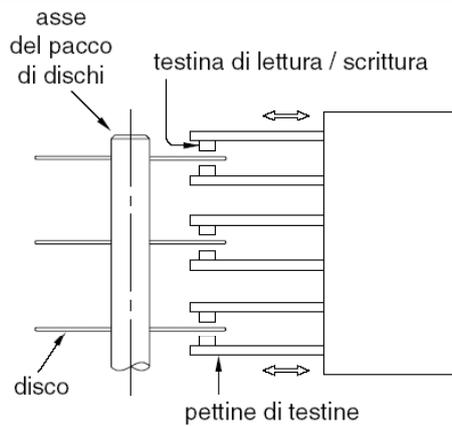
## Memoria Secondaria

- La memoria secondaria o di massa ha la funzione di mantenere grandi volumi di informazione (programmi e dati) a tempo indefinito.
  - Essa fornisce il supporto fisico per memorizzare il file system, cioè il complesso di programmi e dati persistenti installati sul calcolatore.
  - Essa fornisce anche il supporto fisico per la memoria virtuale, cioè per il complesso di funzioni capaci di realizzare uno spazio logico di memoria primaria avente capacità superiore alla memoria centrale elettronica fisica installata nel calcolatore.
  - A motivo della capacità elevata, la memoria secondaria è strutturata a blocchi (a differenza di quella primaria), dove il blocco è l'unità minima di informazione accessibile (in lettura o scrittura).
  - Per lo stesso motivo, la memoria secondaria ha tempo di accesso molto inferiore a quello caratteristico della memoria primaria, ma una volta iniziato il trasferimento, esso può procedere con flusso di dati (*throughput*) molto elevato (almeno in alcune tecnologie).
- 

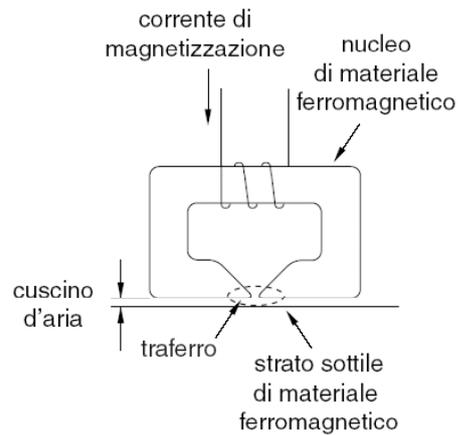
---

## Tecnologia Magnetica

- La forma di memoria di massa più versatile e di costo abbastanza contenuto (per bit memorizzato), è la tecnologia di disco magnetico.
  - Essa si basa sull'idea di registrare un blocco (= una sequenza di byte o bit) come successione di stati di magnetizzazione di una traccia (lineare) su una superficie rivestita di materiale ferromagnetico.
  - Le operazioni di lettura e scrittura (del blocco) si basano sul fenomeno di induzione elettromagnetica, muovendo una testina di lettura e scrittura lungo la traccia.
  - La tecnica di codifica dei bit nella traccia ha varianti.
-



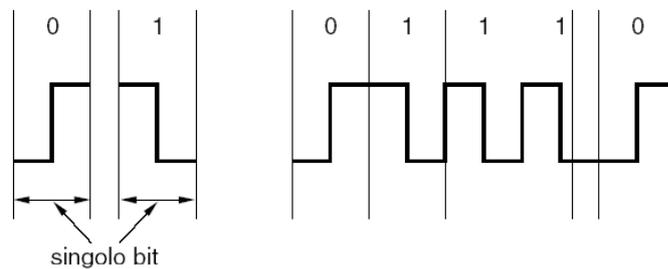
(a) struttura meccanica



(b) testina di lettura / scrittura

## Tecnologia Magnetica

verso del campo magnetico sulla superficie del disco

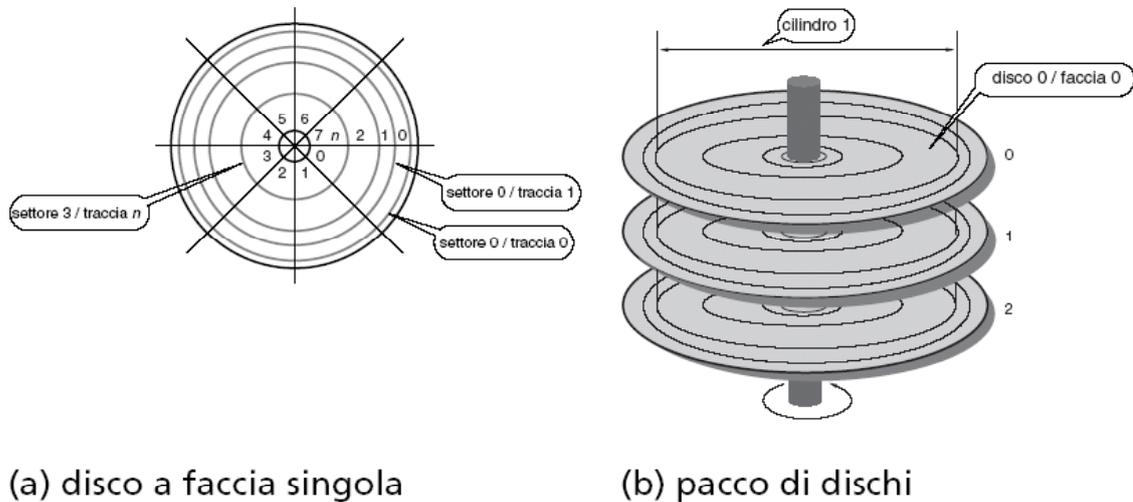


(c) codifica dei bit nella traccia

## Disco Magnetico

- Il disco magnetico a faccia singola contiene un insieme di tracce circolari concentriche registrate sulla faccia.
- Ogni traccia è numerata univocamente ed è suddivisa in settori angolari, anch'essi numerati (in ciascuna traccia).
- Un pacco di dischi (coassiali e solidali) è costituito da diversi piatti e facce, anch'esse numerate.
- L'insieme di numero di faccia, traccia e settore identifica univocamente il settore nell'intero pacco di dischi.
- Talvolta l'insieme di tracce aventi tutte la medesima distanza dal centro del pacco è chiamato cilindro. I settori sono numerati univocamente in ciascun cilindro.
- L'insieme di numero di cilindro e settore forma un sistema di identificazione alternativo (ma equivalente).

## Struttura del Disco Magnetico



tipiche strutture di disco magnetico

## Dimensionamento

- La capacità complessiva di un disco (o di un pacco di dischi) va da pochi Mbyte a circa  $10^2$ - $10^3$  Gbyte.
- Il numero di tracce varia da poche decine alle centinaia o migliaia per faccia.
- Il numero di settori varia da poche decine o centinaia per traccia.
- La velocità di rotazione varia da pochi giri / s a  $10^4$  giri / s od oltre.
- Il tempo di accesso (a un blocco) si misura tipicamente in millisecondi: 1 – 10 ms.
- Il flusso dati raggiunge il valore delle decine o centinaia di Mbyte / s.
- Il disco magnetico è estremamente versatile e ha parametri selezionabili in un intervallo molto ampio, secondo la tecnologia.
- Il tempo di accesso è il parametro più rigido, perché di necessità è limitato dalla natura meccanica del disco.
- La capacità e la velocità del disco magnetico aumentano a tasso costante (la capacità raddoppia ogni circa due anni), mentre il tempo di accesso diminuisce solo molto lentamente.

---

## Organizzazione a Blocchi

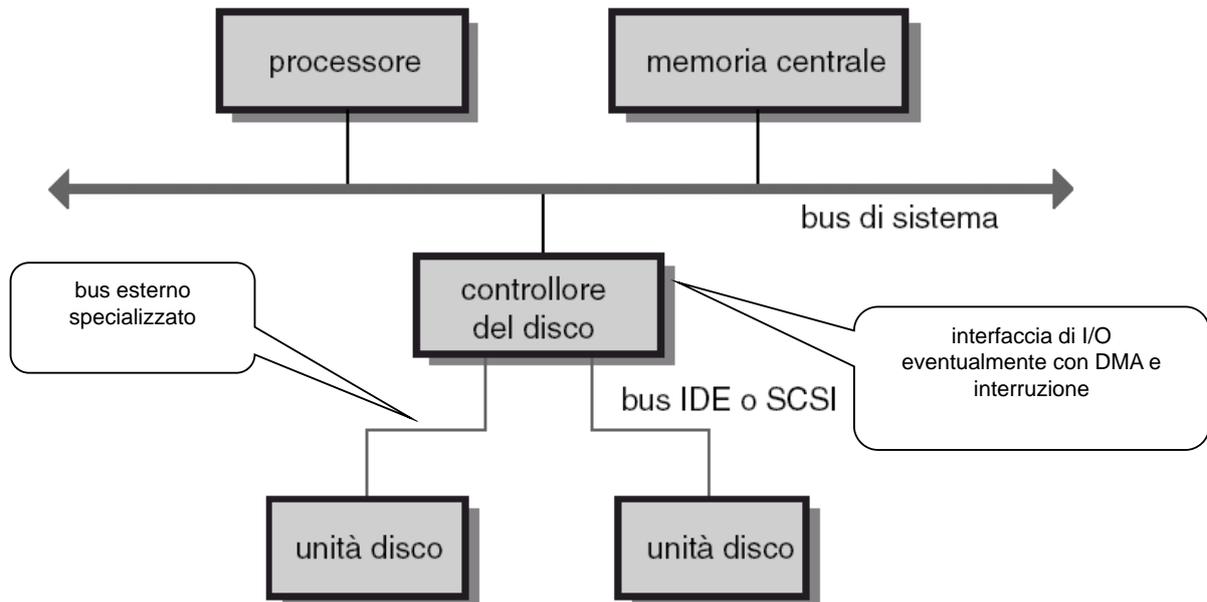
- Qualunque sia il sistema di identificazione in uso per localizzare settore, il disco è visto logicamente dal sistema operativo come se fosse una successione di blocchi numerati linearmente, la quale si estende sull'intero disco.
  - Le operazioni di disco consistono essenzialmente nel leggere o scrivere un blocco, dandone il numero.
  - L'interfaccia del disco stesso, o il sistema operativo, si incarica di convertire il numero logico del blocco nell'indirizzo fisico (faccia / traccia / settore).
  - Il blocco è costituito al minimo da un settore, ma più spesso da un numero intero e prefissato di settori, contigui sulla traccia del disco.
  - La strutturazione del disco in blocchi e la taglia del blocco in settori (fattore di blocco) vengono stabilite al momento della formattazione.
  - La taglia del blocco è assai variabile, comunque si va da qualche Kbyte alle decine di Kbyte (dipende molto dal sistema operativo).
- 

---

## Interfaccia Disco-Calcolatore

- Si può considerare il disco come una periferica con funzioni di memoria persistente, di tipo piuttosto sofisticato.
  - L'unità disco ha a bordo un circuito elettrico, più o meno complesso, che ne pilota la testina, aziona il motore, legge e scrive la traccia.
  - L'unità disco è comunque collegata al calcolatore tramite un'interfaccia di I/O (come ogni altra periferica).
  - L'interfaccia del disco può essere complicata, ma ha due lati:
    - dal lato di calcolatore, quello di una porta di I/O di tipo più o meno standard
    - dal lato di periferica, pilota un bus esterno più o meno specializzato per i dischi (bus IDE o SCSI)
  - L'interfaccia (o controllore di disco) riceve comandi dal sistema operativo, li elabora e invia al disco (cioè al circuito interno di controllo), e scambia blocchi di dati con il disco.
  - Non di rado i dischi veloci e capaci sono gestiti in modo di DMA (il controllore di DMA è nell'interfaccia), con interruzione finale.
-

## Interfaccia Disco-Calcolatore



## Partizione di Disco

- Il disco può essere suddiviso in partizioni, specialmente se è di tipo rigido e ha capacità elevata.
- La partizione è un insieme di blocchi numerati linearmente, a partire dal blocco numero 0.
- La partizione costituisce, in un certo senso, un disco (logico) autonomo e indipendente dagli altri.
- Per effettuare un'operazione di disco, occorre specificare la partizione dove lavorare e, all'interno di questa, il numero del blocco da leggere o scrivere.
- In casi semplici il disco ospita una sola partizione, che coincide con il disco stesso.
- Piuttosto spesso però il disco ospita un certo numero di partizioni distinte, di varia dimensione.
- Numero e disposizione delle partizioni vengono decisi al momento della formattazione e stanno registrati in un blocco speciale del disco, destinato a questo uso particolare.

---

## Memoria Secondaria e File

- L'intero disco o una sua partizione, visti come insieme di blocchi numerati linearmente, viene strutturato come "file system" da parte del sistema operativo.
  - Il file system è una struttura dati complessa e articolata, il cui scopo è di organizzare logicamente il disco o la partizione come insieme (ordinato e gerarchico) di file.
  - Il file è il contenitore elementare di informazione persistente: programma o dati.
  - Quando il disco o la partizione sono usati come file system, non hanno legame specifico con la memoria primaria, avendo rispetto a questa scopo ben diverso.
- 

---

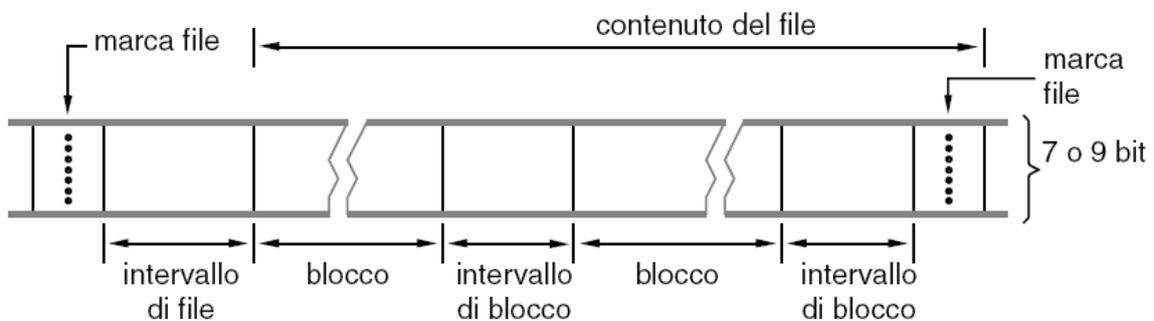
## Memoria Secondaria e Virtuale

- L'intero disco o, più spesso, una sua partizione, possono essere usati come area di memoria secondaria associata alla gerarchia di memoria (piramide di memoria), di cui formano lo strato inferiore.
  - Tale area di memoria è naturalmente strutturata a blocchi e pertanto viene usata come area di scarico per le pagine di memoria fuori uso.
  - In tale funzione, il disco o la partizione fanno parte del sistema di memoria virtuale, gestito da parte del sistema operativo.
  - Lo scopo è di realizzare una memoria primaria logica di dimensione maggiore della memoria centrale elettronica installata nel sistema.
  - La gestione del disco o della partizione di memoria virtuale è affidata all'unità di gestione di memoria e al sistema operativo.
  - Si tratta di un uso ben diverso da quello come file system, e in genere l'area di memoria secondaria per uso virtuale non è affatto strutturata a file, ma ha una sua organizzazione specifica (può però anche essere modellata come file speciale nel file system globale).
-

# Nastro Magnetico

- Il nastro magnetico mantiene informazione secondo lo stesso principio fisico del disco (induzione elettromagnetica), ma il supporto è un nastro lineare avvolgibile di celluloido rivestito di materiale ferromagnetico, e letto/scritto tramite una testina di lettura/scrittura.
- Il nastro è un dispositivo inerentemente sequenziale, organizzato a blocchi, con tempo di accesso lungo al blocco (secondi o minuti), ma di capacità molto grande e di lunga durata (decenni, se ben conservato in luogo riparato).
- Viene usato come archivio di file a lungo termine. I file sono registrati sequenzialmente, senza organizzazione gerarchica.
- Non è possibile usare il nastro magnetico come memoria secondaria annessa alla gerarchia di memoria in funzione di espansione della memoria primaria.

## Nastro Magnetico



tipica organizzazione a blocchi del nastro magnetico

---

## Altri Tipi di Disco

- Ci sono oggi numerosi altri tipi di supporto di memoria di massa, principalmente i dischi ottici: CD, DVD, ecc, con un lungo elenco di varianti tecnologiche.
  - L'informazione (blocco di byte) viene letta e scritta in modo ottico (tramite tecnologia laser), ed è espressa in vari modi fisici.
  - Sono supporti di memoria adatti principalmente per volumi di dati persistenti (dati multimediali, audio e video, ecc).
  - La durata è relativamente lunga, sebbene non come quella del nastro magnetico, e il supporto stesso è inerentemente più fragile.
  - La capacità è grande o grandissima, il flusso di dati elevato, ma il tempo di accesso lungo; tuttavia il costo per bit è molto basso.
  - È presumibile che compaiano nuove tecnologie di questo tipo, con capacità estremamente elevata e costo ridotto.
  - In generale sono supporti inadatti per uso come espansione della memoria primaria, a causa del tempo di accesso troppo lungo.
-