Calcolatori Elettronici
Sistema di memoria
parte seconda

Ing. Gestionale e delle Telecomunicazioni
A.A. 2009/10
Gabriele Cecchetti - Anna Lina Ruscelli

Sistema di memoria – *parte seconda*

- Sommario:
  - Gerarchia di memoria
  - Memoria cache
  - Schemi di indirizzamento
  - Riassunto e particolari
  - Prestazioni della cache
  - Memoria cache del processore 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, introducendo 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)
- Sarà trattato il sistema di memoria cache, che ha lo scopo di rendere più veloce la memoria.
- Verrà ripreso il concetto di memoria virtuale che e' di pertinenza specialmente del SO.

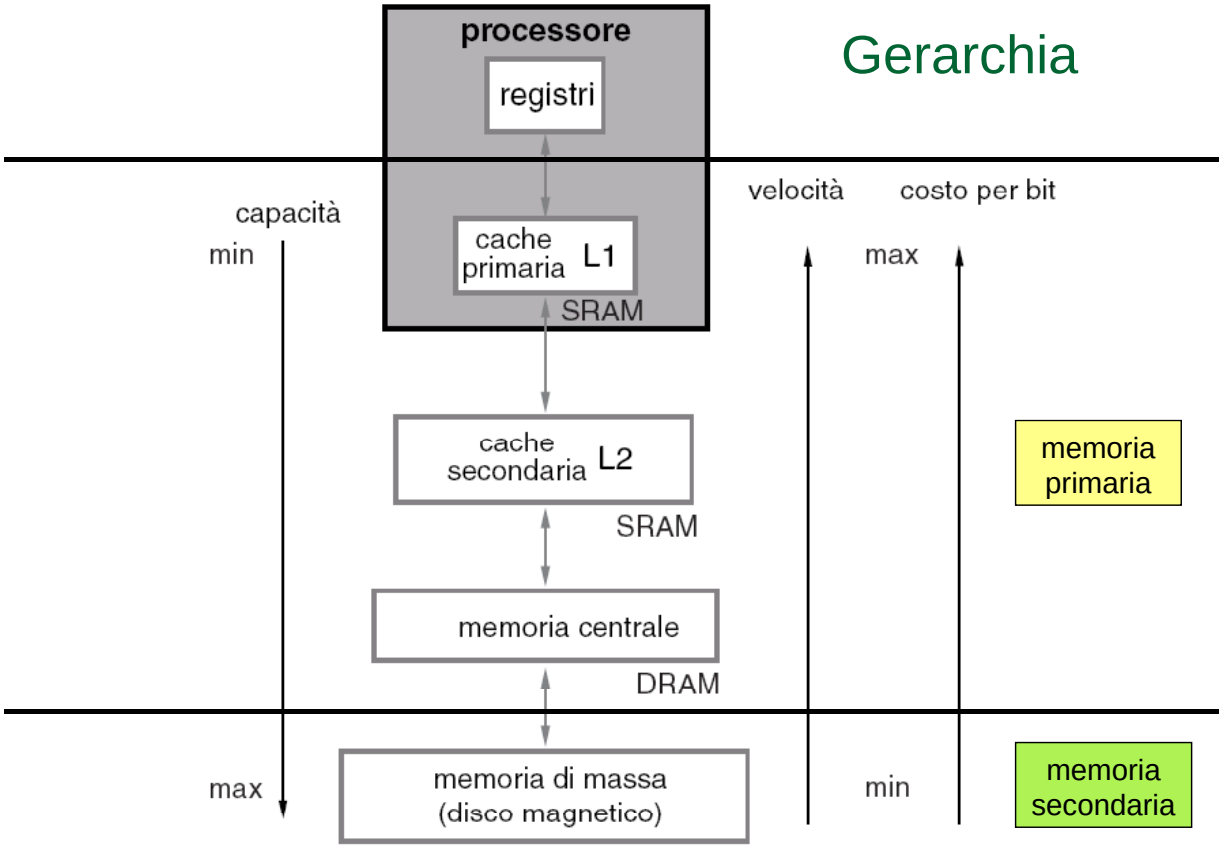
Tempo di Accesso

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

Gerarchia



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 superiori contengono solo copie dell'informazione alla base, non nuova informazione).

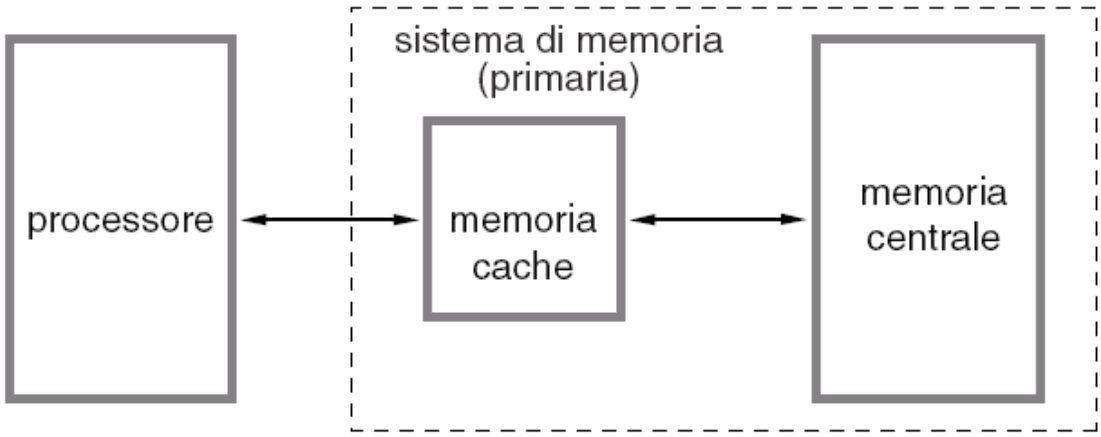
Scomposizione della Piramide

- 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 presente di tecnologie sufficientemente differenziate.

Memoria Primaria

- Motivazione 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

- Motivazione 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 (memoria 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 della cache capace di accogliere esattamente un blocco di memoria centrale
  - ad 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)

- Quando 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 dalla memoria centrale.

Funzionamento della Cache (2/2)

- Quando il processore deve leggere un dato da memoria per elaborarlo (per esempio aritmeticamente):
  - si procede come per il prelievo di istruzione
- Quando 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
  - 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

- Un 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 di passa ad 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 causa dell'esistenza di strutture dati aggregate: se un dato della struttura va in uso gli altri dati della stessa struttura, che hanno uno scopo correlato, 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 causa dell'esistenza di cicli: un ciclo ben programmato si reputa plausibile vada ripetuto almeno qualche volta, in particolare se è corto; così le istruzioni che esso contiene vengono di nuovo eseguite 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.
- Poiché spesso si presentano insieme si possono allora raggruppare in un solo principio, chiamato principio di **località spaziale-temporale**:
  - *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.*

Località e Cache (1/2)

- Quando il processore usa una parola di memoria l'unita' di controllo della cache duplica e scrive in copia nella cache il blocco che la contiene.
- L'uso della 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, cioè l'intero blocco (**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)

Località e Cache (2/2)

- Quindi 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 usato di nuovo più volte entro breve tempo.

Hit e Miss

- **Evento di hit** (successo): accesso in lettura o prelievo / scrittura di una parola (dato o istruzione) in una posizione di memoria cache:
  - **hit rate** (tasso di successo): rapporto tra numero di accessi a memoria che trovano la parola in cache e 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 alla 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 – hit rate
  - **penalità di miss** (tempo di fallimento): tempo per sostituire un blocco in cache + tempo di accesso alla parola in cache

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

SCHEMI DI INDIRIZZAMENTO

1) Indirizzamento Diretto

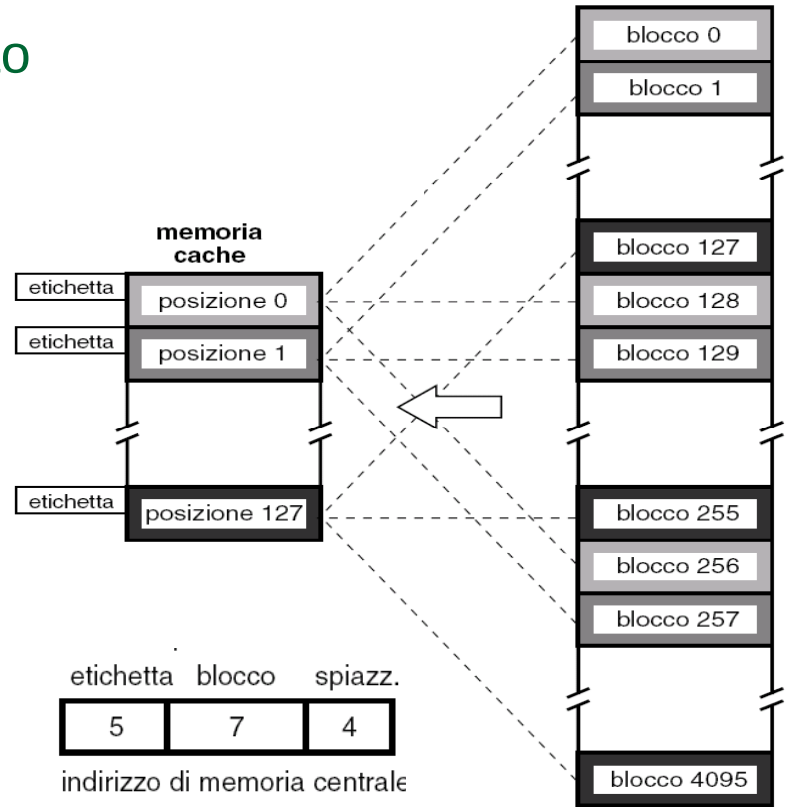
- Ogni blocco della memoria centrale è caricabile in una **sola** posizione della memoria cache.
- Si possono caricare più blocchi di memoria centrale nella stessa posizione di cache (**non in modo simultaneo**) - conflitto
- Il blocco numero *j* della memoria centrale è caricabile **solo** nella posizione numero:

  resto (div. intera j / n° posizioni della cache)
- È anche chiamato schema *direct mapping*.

Indirizzamento Diretto

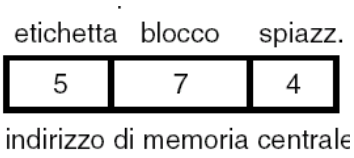
Legenda:

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 − spiazz.−blocco



Dimensionamento

- Indirizzo di memoria centrale: $n = 16$ bit
- Indirizzo di memoria cache: $m = 11$ bit
- Dimensione in parole di blocco e di 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 posizioni $= 7$ bit
- **Etichetta**:
  16 bit – 7 bit – 4 bit = 5 bit



Scomposizione di Indirizzo

n bit di indirizzo di memoria centrale

etichetta	(indice di) blocco	spiazzamento

n – blocco – spiazz.

quale blocco di memoria centrale è caricato in cache nella posizione ammissibile per quel blocco

indice del blocco in cache, dà la posizione ammissibile per il blocco

spiazzamento della parola nel blocco

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 = $\lceil \log_2$ (# posizioni)$\rceil$

etichetta = n − spiazz.−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 non valide
  - **campo etichetta**, che contiene il valore che identifica il blocco di memoria correntemente in cache, tra quelli ammissibili
  - **campo dati**, contenente copia del blocco
- L'**accesso alla cache** avviene nel modo seguente:
  - il campo indice di blocco dell'indirizzo di memoria centrale viene usato per selezionare la posizione in cache dove è caricato il blocco
  - se la posizione è valida (cfr. bit di validità), il campo etichetta dell'indirizzo viene **confrontato** con l'etichetta della posizione selezionata
  - se il confronto ha esito positivo significa che il blocco richiesto è presente in cache e il campo spiazzamento dell'indirizzo di memoria centrale viene utilizzato per accedere alla parola di interesse.

Gestione di Hit e Miss di lettura

- **Successo** (hit) nella lettura:
  - la parola è presente in cache e lì viene letta
- **Fallimento** (miss) nella lettura:
  - stallo della CPU e richiesta alla memoria centrale del blocco contenente la parola
  - copia del blocco in cache quindi 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 di scrittura

- **Successo** (hit) 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** (miss) nella scrittura:
  - stallo della CPU e richiesta alla memoria centrale del blocco contenente la parola cercata
  - copia in cache del blocco quindi 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 essere copiato in **una sola** posizione
- Negli schemi di indirizzamento successivi (di tipo associativo), la sostituzione di blocco sarà invece un problema rilevante da risolvere.

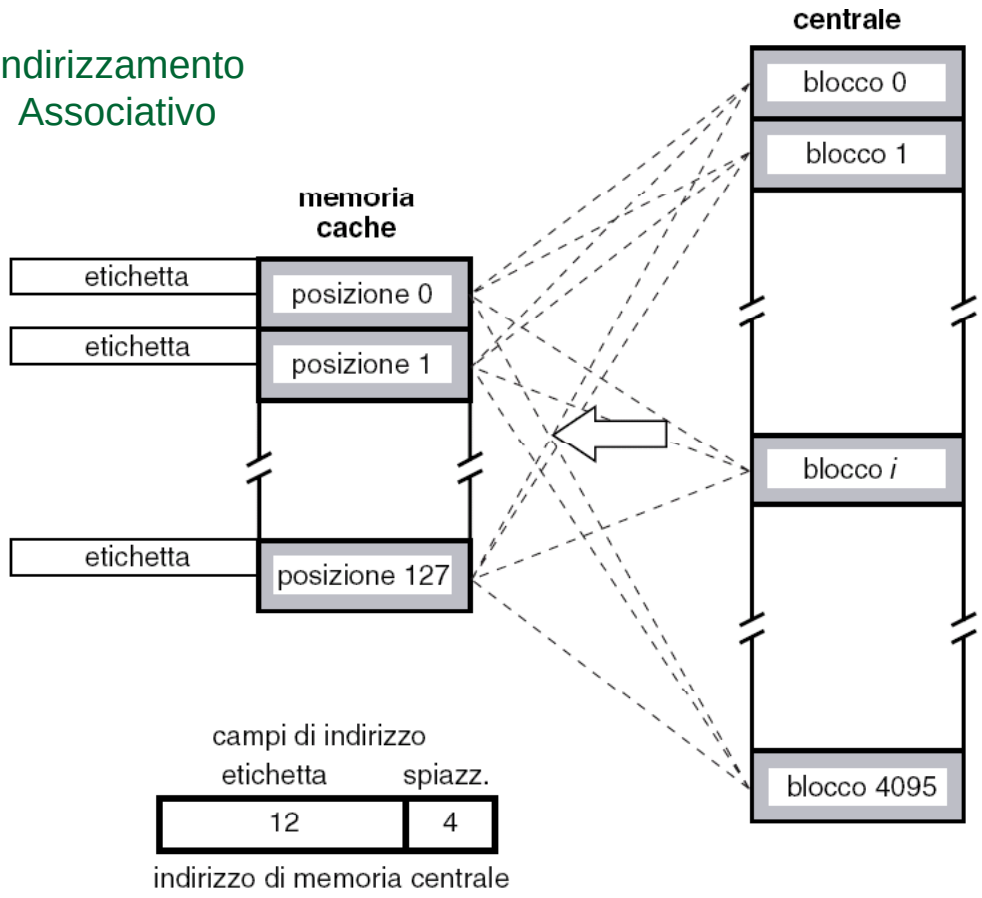
2) Indirizzamento Associativo

- 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 centrale di *n* bit, con blocchi di memoria centrale e posizioni di cache da 2^b parole:

  b bit meno significativi dell'indirizzo di memoria centrale individuano la parola all'interno del blocco di mem. centrale e della posizione di cache -> spiazzamento di blocco

  n – b 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
- Dimensione in parole di blocco e di 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 − 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 (**miss**) 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 **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).

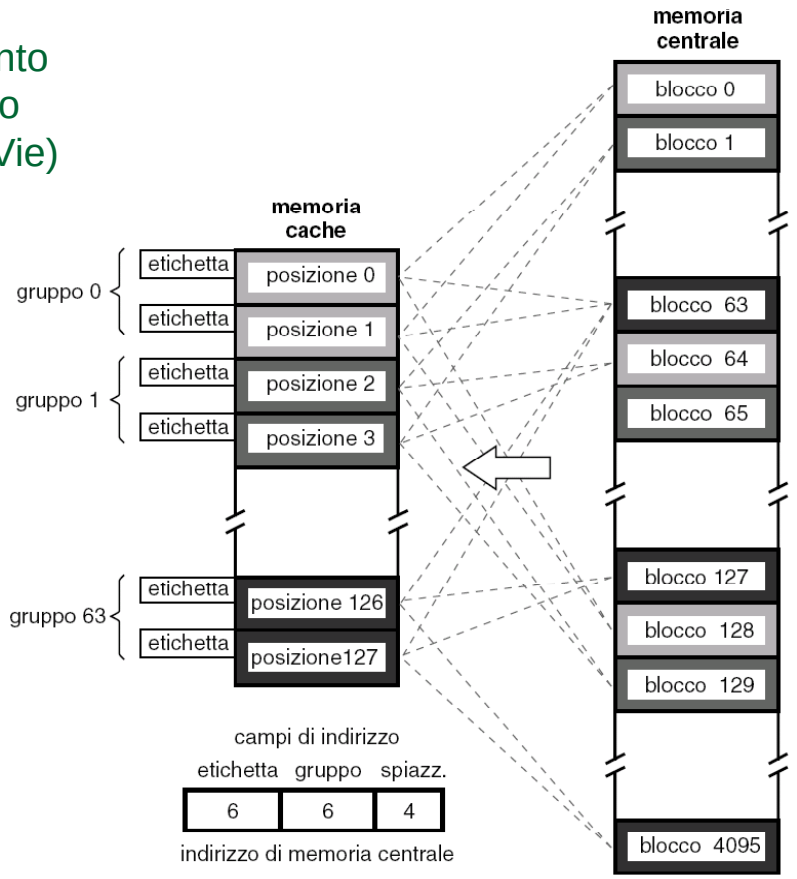
3) Indirizzamento Associativo a Gruppi

- 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 le *g* posizioni (tutte nel medesimo gruppo), si chiama "**a *g* vie**".
- È anche chiamato schema *set associative.*

Esempio - Cache Associativa a 2 Vie

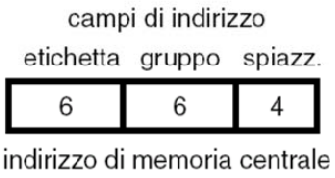
- Di seguito l'esempio di cache associativa a 2 vie.
- Cache a due vie: ogni gruppo è costituito da due posizioni (ogni gruppo è capace di accogliere fino a due blocchi di memoria centrale).
- Il campo **numero o indice di gruppo** dell'indirizzo di memoria centrale seleziona il gruppo dove si deve lavorare, ovvero le due posizioni che lo costituiscono.
- Il campo **etichetta** dell'indirizzo di memoria centrale viene confrontata in parallelo con le due etichette associate rispettivamente alle due posizioni appartenenti al gruppo.
- La posizione di cache 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
- Dimensione in posizioni del gruppo $v = 2$ posizioni per gruppo (2 vie)
- Dimensione in parole di blocco e di 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 (Cache 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 = $\lceil \log_2$ (# gruppi)$\rceil$

etichetta = n – spiazzamento – gruppo

Identificazione e Sostituzione

- L'identificazione della posizione in cache contenente il blocco di interesse procede come nel caso dello 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.

Punti essenziali
Memoria associativa
Memoria cache a due livelli

RIASSUNTO E PARTICOLARI

Riassunto - Indirizzamento

- **Indirizzamento diretto**:
  - **posizione univoca**: indirizzo di memoria centrale *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 di cache corrispondente al blocco
  - 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 (1/3)

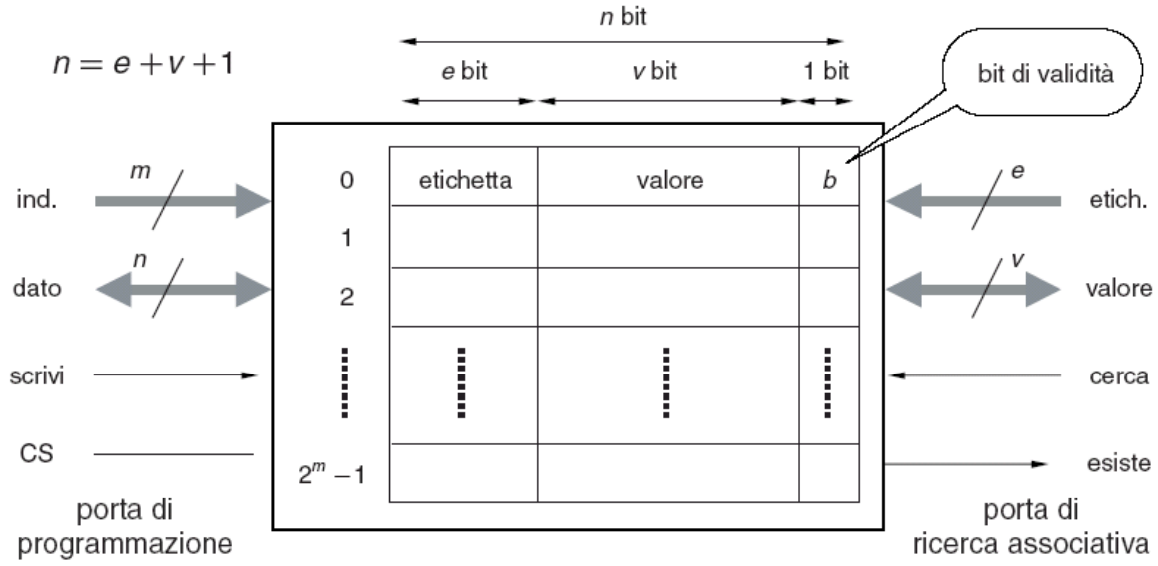
- La ricerca di una parola nella cache richiede il confronto di tutte le etichette presenti in cache (indirizzamento associativo), o delle etichette delle posizioni contenute nel gruppo (indirizzamento associativo a gruppi) o della etichetta dell'unica posizione selezionata (indirizzamento diretto) con l'etichetta del blocco di memoria centrale richiesto.

- Per le etichette viene usato un componente di memoria speciale: la memoria associativa (da non confondere con la cache associativa).

Memoria Associativa (2/3)

- La memoria associativa 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 il campo dati associato
- La memoria associativa è realizzata con tecnologie microelettroniche ad hoc.

Memoria Associativa (3/3)



Ricerca Associativa

- Si usa la porta di ricerca associativa.
- Si invia alla memoria associativa 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, l'etichetta inviata:
  - Se la trova, la memoria emette il campo "valore" (formato da *d* bit) associato all'etichetta trovata e attiva il segnale "esiste".
  - Se non la trova, la memoria disattiva il segnale "esiste".
- Si suppone che non ci siamo mai etichette duplicate.

Memoria Cache a Due Livelli

- Nel caso di memoria cache a 2 livelli la memoria primaria contiene due memorie cache, collegate **in cascata** e interposte tra processore e memoria centrale.
- Normalmente il processore opera (legge e scrive) 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 entrambe le cache hanno tempo di accesso inferiore a quello della memoria centrale e la cache di primo livello e' più veloce di quella di secondo livello -> può essere adottato se la tecnologia di cache è differenziata.

Analisi di prestazione della memoria cache

PRESTAZIONI DELLA CACHE

Prestazioni - 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.

Prestazioni - Una Sola Cache

- Se il sistema di memoria primaria comprende cache e memoria centrale, ci sono due tempi di accesso, quello di memoria centrale e quello di memoria cache.
- Il tempo di accesso esatto va allora sostituito con un tempo medio di accesso ΔT_{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 quello di memoria centrale.

Prestazioni - 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)

ΔT_{acc} tempo medio di accesso al sistema di memoria primaria con cache (per una sola parola)

NB: *m* (tasso di miss) = 1 – *h* (tasso di hit)

I tassi *h* e *m* vanno normalizzati a 1 ($0 \le h, m \le 1$)

Guadagno - Una Sola Cache

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

$H_c = 1$ ns accesso a una sola parola in cache

$H_m = 100$ ns accesso a una sola parola in memoria centrale

$M = 1\ \mu s$ tempo di penalità di miss per caricare in cache un blocco

$$\Delta T_{acc} = 0{,}95 \times 1\ ns + (1 - 0{,}95) \times 1\ \mu s =$$
$$= 0{,}95\ ns + 0{,}05 \times 1000\ ns =$$
$$= 0{,}95\ ns + 50\ ns =$$
$$= 5{,}95\ ns \approx 56\ ns$$

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

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

Prestazioni - Cache Istruzioni e Dati (1/2)

- Si può perfezionare la memoria primaria prevedendo due cache separate, una per le istruzioni ed 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 necessariamente essere 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 di dati, le quali possono presentare qualche differenza.

Prestazioni - Cache Istruzioni e Dati (2/2)

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

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

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

NB: # tot. accessi = # acc. istr. + # acc. dato

Guadagno - Cache Istruzioni e Dati (1/2)

Assunzioni

$h_{istr} = 0{,}95$	95% di eventi di hit di istruzione
$h_{dati} = 0{,}9$	90% di eventi di hit di dato
$H = 1$ ciclo di clock	tempo accesso di istruzione o dato in cache (sono identici)
10 cicli	tempo accesso di istruzione o dato in memoria centrale in un sistema senza cache (sono identici)
$M = 17$ cicli di clock	penalità di miss - per caricare in cache un blocco di istruzioni o dati

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

NB: i tempi sono misurati in numero di cicli di clock.

Guadagno - Cache Istruzioni e Dati (2/2)

$$g = \frac{\textit{tempo di accesso del sistema di memoria senza cache}}{\textit{tempo di accesso del sistema di memoria con cache}} =$$

$$= \frac{130 \times 10}{100\left(h_{istr.}H + \left(1 - h_{istr.}\right)M\right) + 30\left(h_{dati}H + \left(1 - h_{dati}\right)M\right)} =$$

$$= \frac{130 \times 10}{100\left(0,95 \times 1 + 0,05 \times 17\right) + 30\left(0,9 \times 1 + 0,1 \times 17\right)} \approx 5,04$$

Il sistema di memoria primaria con due cache separate per istruzioni e dati ha un guadagno di prestazione di circa il 500%.

Guadagno - Cache Ideale e Reale

- Supponiamo che entrambe le cache (istruzioni e dati) abbiano tasso di hit pari a 1 (non c'è mai fallimento).-> È una situazione di **cache ideale**
- Confrontiamo la cache ideale con quella reale (con i tassi di hit precedenti).

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

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

Prestazioni - Cache a Due Livelli

- La formula di prestazione della 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 (cfr l'esempio numerico sul testo).

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

- 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).

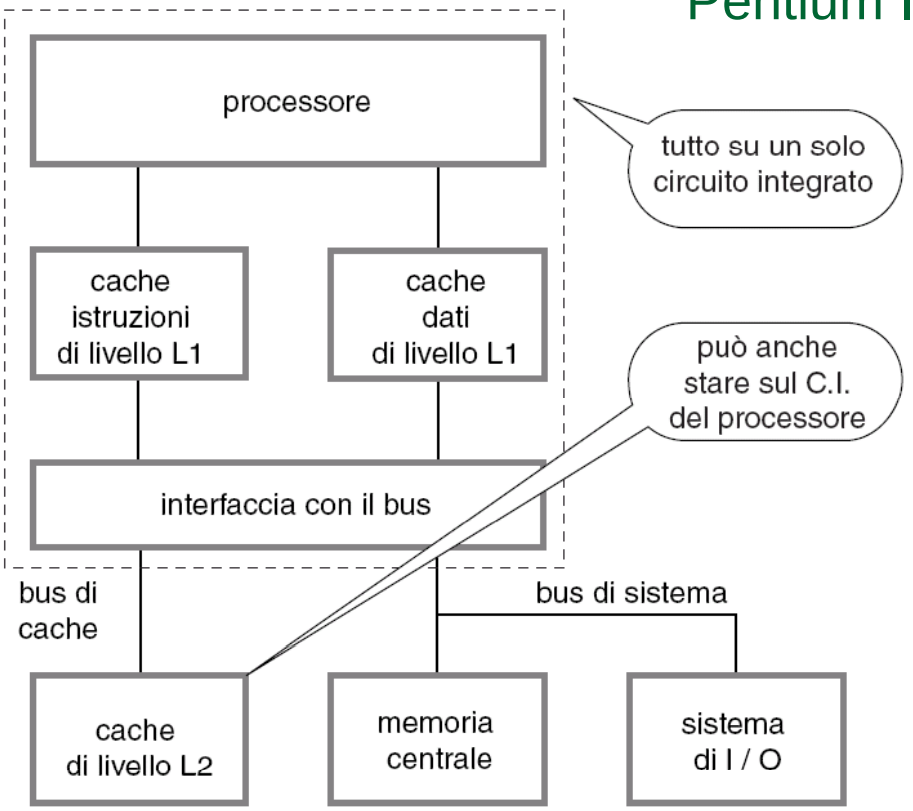
Esempio: sistema di memoria cache per processori IA-32

MEMORIA CACHE IA-32

Cache del processore Pentium IA-32

- I processori Pentium Intel a 32 bit sono dotati di un sistema di cache sofisticato, a due livelli (esistono alcune varianti a seconda del modello).
- 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.

Pentium IA-32



Struttura del sistema di memoria virtuale
Relazione con la 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** a quella della memoria primaria (memoria 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 direttamene dal S.O.

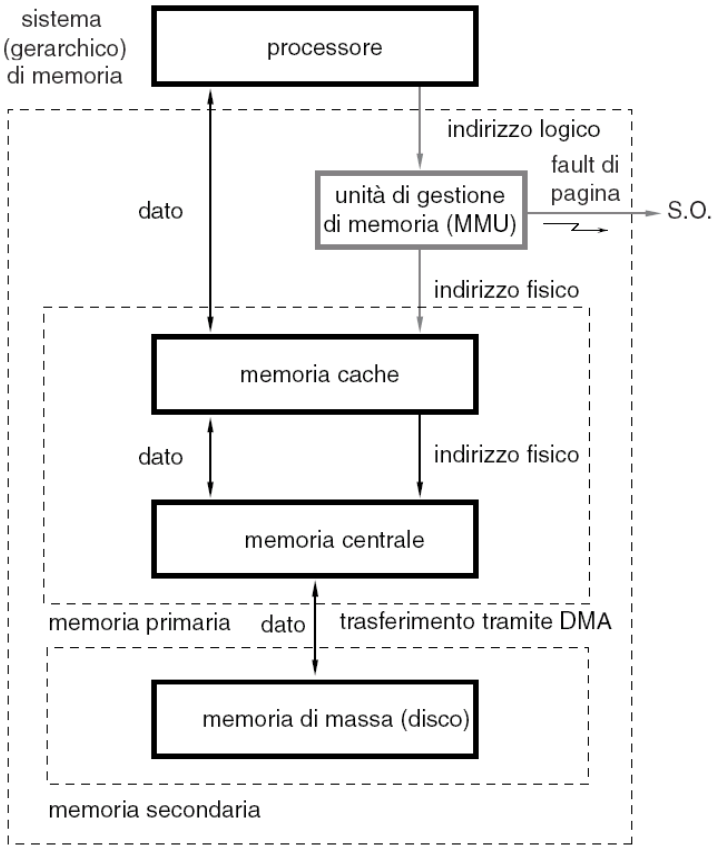
Relazione con la Cache (1/2)

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

Relazione con la Cache (2/2)

- Diversamente dalla memoria virtuale, la memoria cache **non** è gestita direttamente dal S.O., se non per configurarla inizialmente (se essa è configurabile).
- Il segnale di hit/miss deve essere 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.

MMU e Cache



Il S.O. attiva il gestore di memoria virtuale

Esso interagisce con la memoria secondaria per caricare e scaricare pagine

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** (indirizzo 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”.

Schema di traduzione dell’indirizzo con Tabella delle Pagine

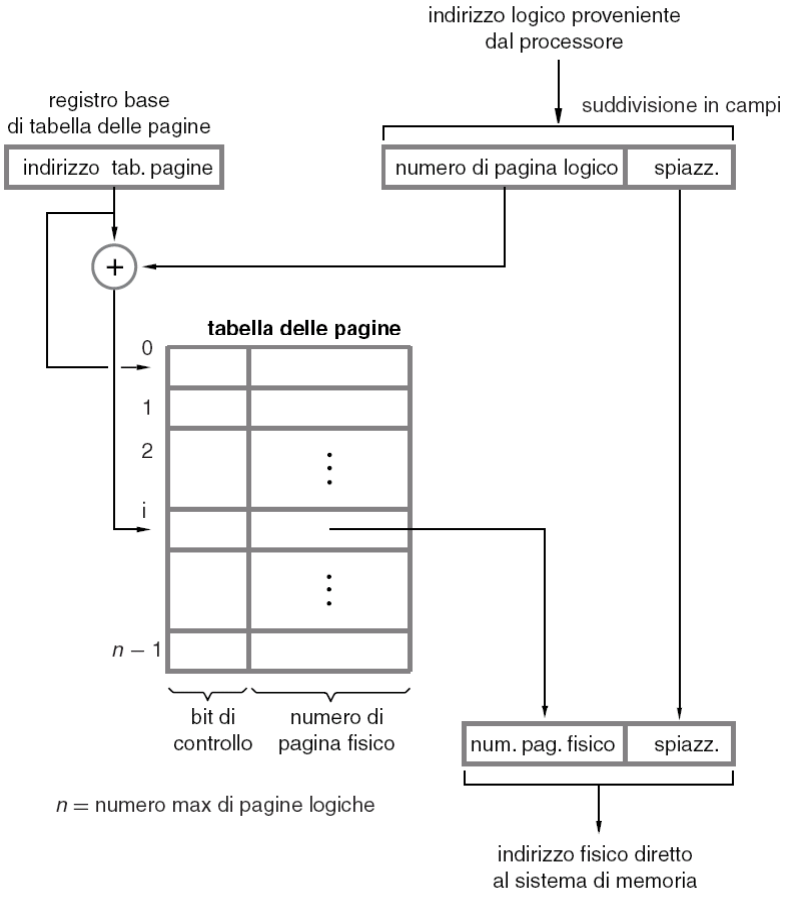
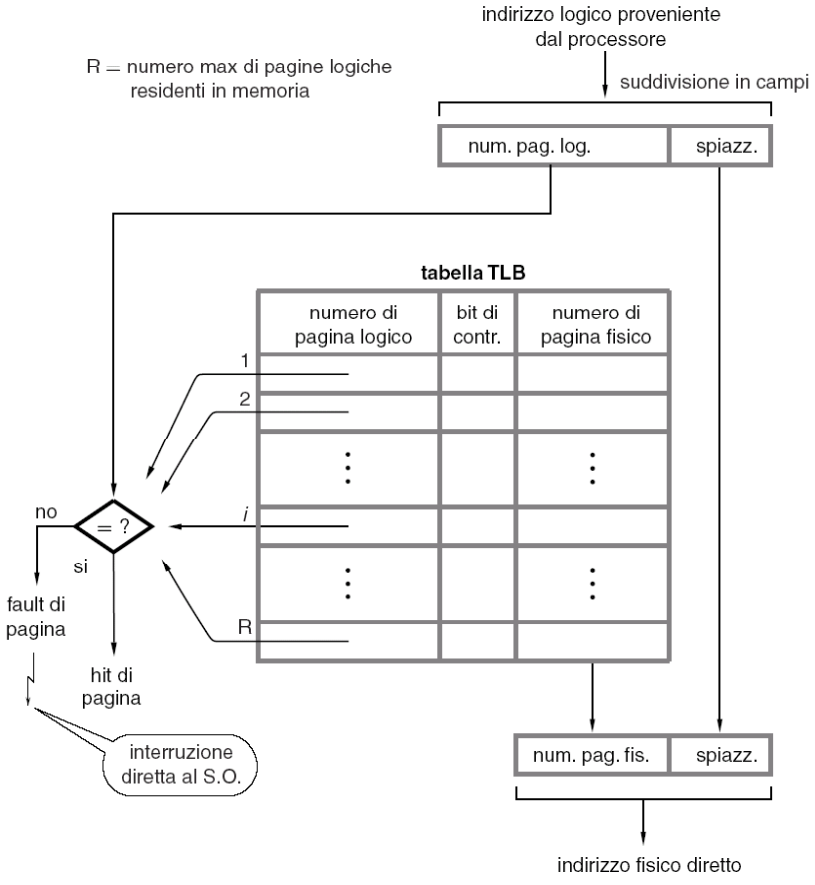


Tabella TLB

- Tra le ottimizzazioni strutturali dell'MMU c'e' l'uso di una tabella di pagine ausiliaria oganizzata come ***Translation Lookaside Buffer***, (TLB).
- 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 ed ai bit di controllo). 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.

Schema di traduzione dell'indirizzo con Tabella TLB



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

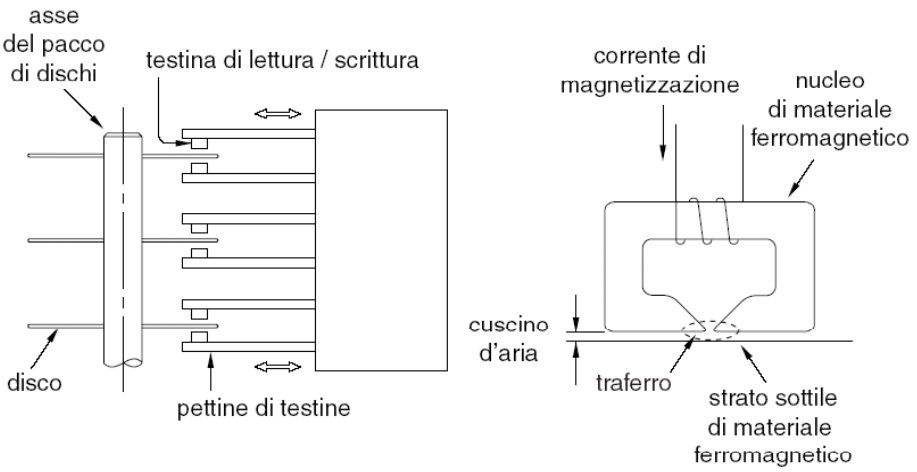
MEMORIA DI MASSA

Memoria Secondaria

- La memoria secondaria o di massa ha la funzione di mantenere **grandi volumi di informazione** (programmi e dati) per un 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 causa 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).
- La memoria secondaria ha **tempo di accesso molto superiore** 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).

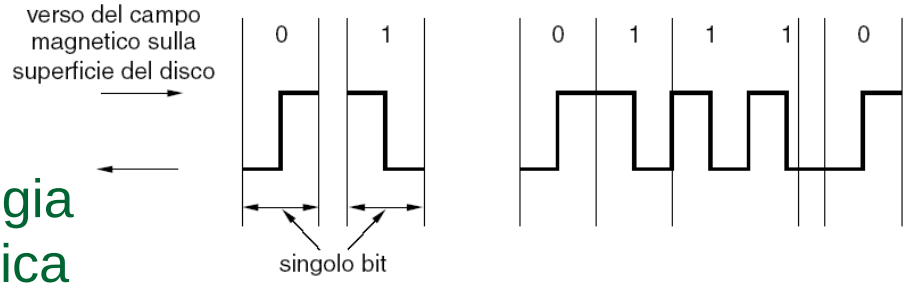
Tecnologia Magnetica

- La forma di memoria di massa più versatile e di costo abbastanza contenuto (per bit memorizzato), è basata sulla 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.
- Sono adottate varie tecniche di codifica dei bit.



(a) struttura meccanica

(b) testina di lettura / scrittura



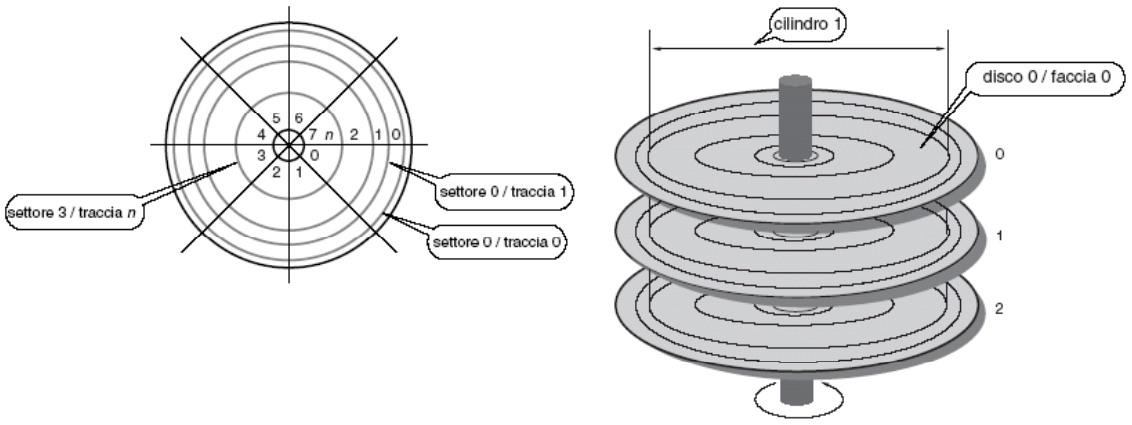
(c) codifica dei bit nella traccia

Tecnologia Magnetica

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.
- 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).

Tipiche strutture del Disco Magnetico



(a) disco a faccia singola (b) pacco di dischi

Dimensionamento

- La capacità complessiva di un disco (o di un pacco di dischi) va dà 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, a seconda della 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 molto lentamente.

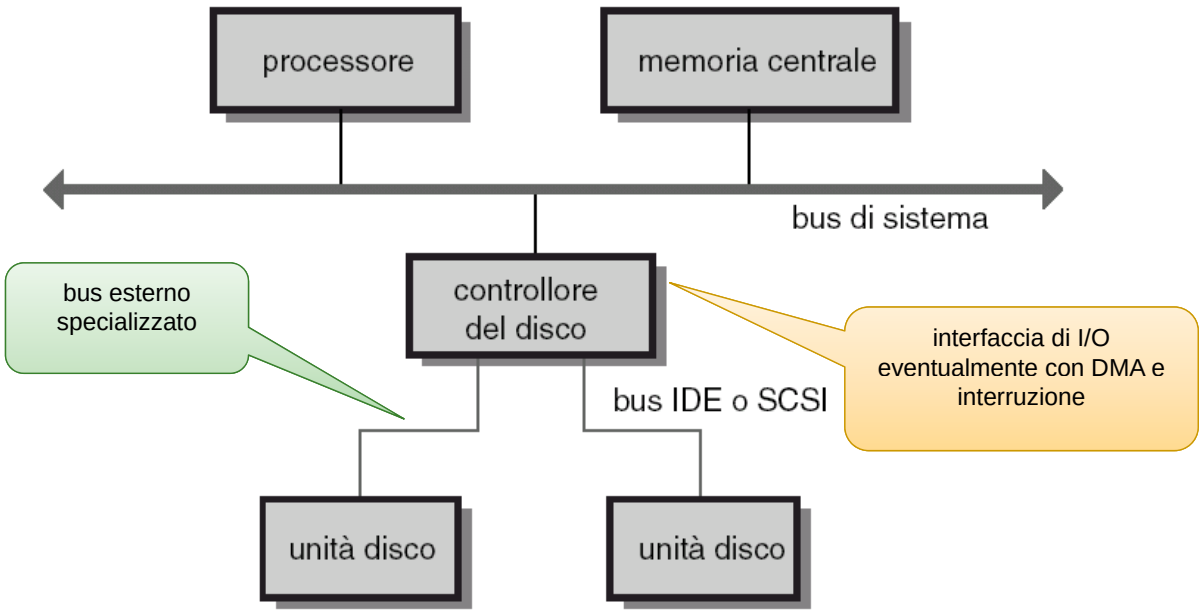
Organizzazione a Blocchi

- Qualunque sia il sistema di identificazione in uso per localizzare il 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.
- 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 dimensione del blocco in numero di settori (fattore di blocco) vengono stabilite al momento della formattazione.
- La dimensione 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 **dispositivo di controllo** (disk drive) che ne pilota la testina, aziona il motore, legge e scrive la traccia.
- L'unità disco è collegata al calcolatore tramite un'interfaccia di I/O (come ogni altra periferica).
- L'interfaccia del disco (o **controllore di disco**)
  - dal lato del calcolatore, ha la struttura di una porta di I/O di tipo più o meno standard
  - dal lato della periferica, pilota un bus esterno più o meno specializzato per i dischi (bus IDE o SCSI)
- L'interfaccia riceve comandi dal sistema operativo, li elabora e li invia al disco (cioè al circuito interno di controllo) e scambia blocchi di dati con il disco stesso.
- Non di rado i dischi veloci e capaci sono gestiti in modalità DMA (il controllore di DMA è nell'interfaccia), con interruzione finale.

Interfaccia Disco-Calcolatore



Partizione del 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 su 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 ma spesso il disco ospita un certo numero di partizioni distinte, di varia dimensione.
- Numero e disposizione delle partizioni vengono decisi al momento della formattazione e sono 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 diverso.

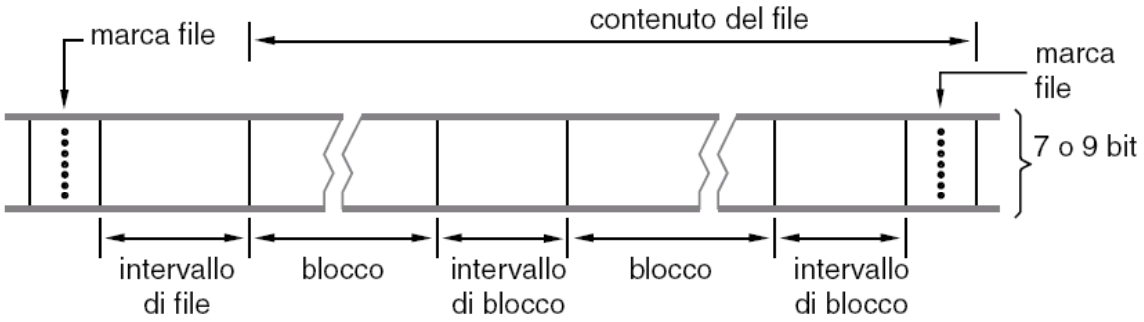
Memoria Secondaria e Virtuale

- 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 non in uso.
- In questo caso il disco o la partizione fanno parte del sistema di memoria virtuale, gestito dal 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 (MMU) e al sistema operativo.
- Si tratta di un uso diverso da quello come file system e in genere l'area di memoria secondaria per uso virtuale non è strutturata in 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 celluloide rivestito di materiale ferromagnetico e letto/scritto tramite una testina di lettura/scrittura.
- Il nastro è un dispositivo strettamente **sequenziale, organizzato a blocchi**, con tempo di accesso al blocco lungo (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 ed il supporto è più fragile.
- La capacità è grande o grandissima, il flusso di dati elevato, il costo per bit è molto basso ma il tempo di accesso lungo.
- È presumibile che compaiano nuove tecnologie di tipo ottico, con capacità estremamente elevata e costo ridotto.
- In generale sono supporti inadatti all'uso come espansione della memoria primaria a causa del tempo di accesso troppo lungo.