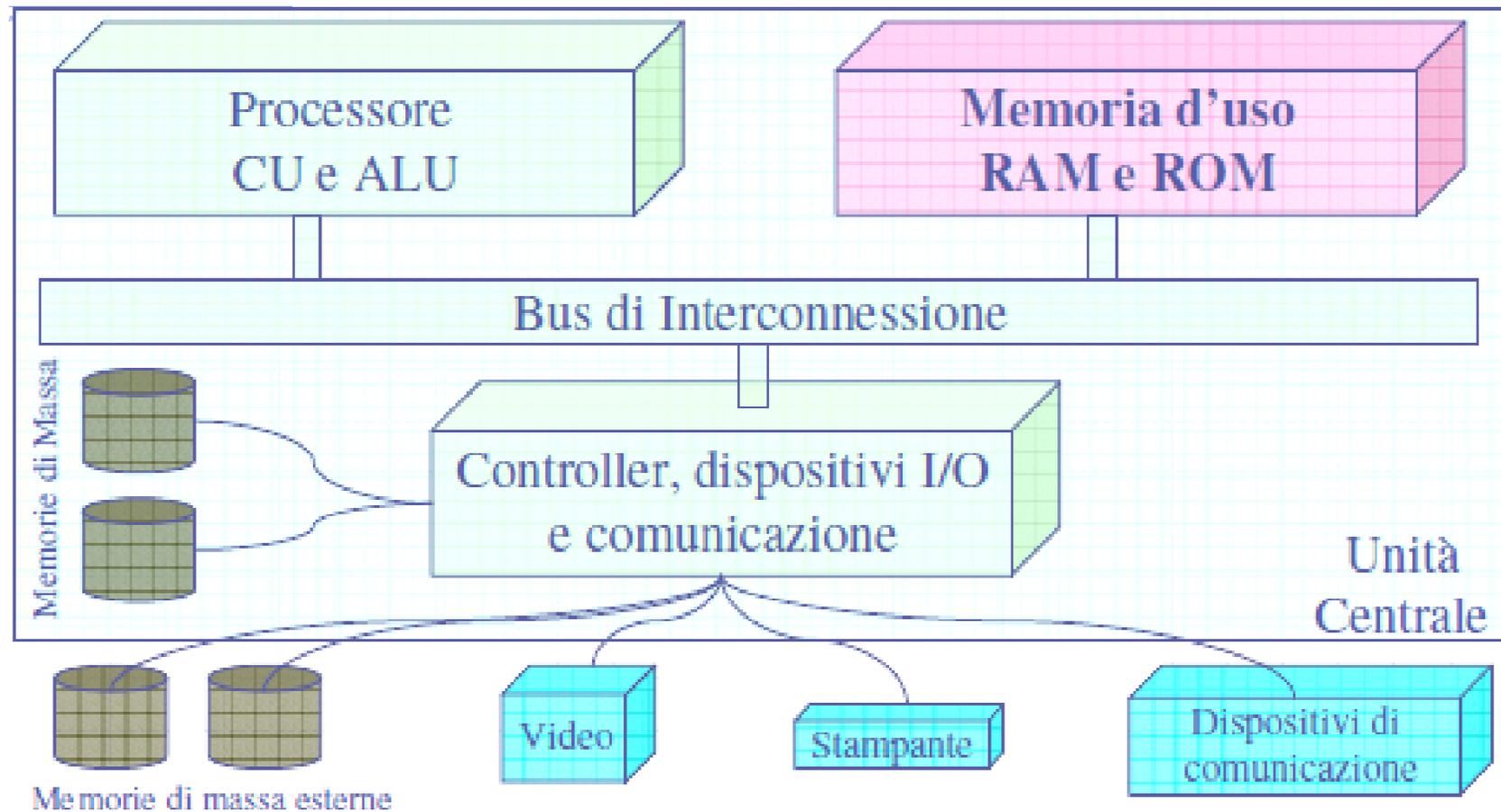


Le memorie

Alessandra Giordani

ITT Marconi

Le memorie d'uso



Fasi

- ◆ **Caricamento:**
 - La CU preleva l'istruzione successiva presente in memoria e la immagazzina al suo interno in un registro.
- ◆ **Decodifica:**
 - La CU interpreta l'istruzione e trasferisce i dati a cui questa fa riferimento dalla RAM alla ALU.
- ◆ **Esecuzione:**
 - La ALU esegue l'operazione logica o aritmetica richiesta.
- ◆ **Memorizzazione:**
 - I risultati delle operazioni svolte sono immagazzinati nella RAM oppure in un registro della ALU detto accumulatore.

Tempi dell'interazione

- ◆ **Tempo di istruzione:**
 - tempo necessario per portare a termine le fasi 1 e 2
- ◆ **Tempo di esecuzione:**
 - tempo necessario per portare a termine le fasi 3 e 4

Tipi di memoria

- ◆ Memorie RAM sono o DRAM o SRAM
 - differiscono per costo e velocità
 - la scelta dipende dall'utilizzo della memoria
- ◆ Le RAM più diffuse sono le DRAM
 - Realizzazione:
 - DRAM: Condensatori
 - SRAM: Flip-Flop

SRAM

- ◆ Static RAM:
 - Non necessita di cicli di refresh
 - Molto costosa
 - Molto veloce, qualche nanosecondo
 - Utilizzata come memoria cache

DRAM

- ◆ DRAM: Dynamic RAM
 - Dinamica perché ha bisogno di continui (ogni due ms) cicli di refresh (aggiornamento dei dati presenti in essa)
- ◆ Pregi:
 - basso costo e grandi capacità di memoria di un singolo chip
- ◆ Difetti:
 - Più lente e più difficili da costruire rispetto alle SRAM

Principali tipi di DRAM

◆ Asynchronous DRAM

- Le operazioni di accesso alla memoria sono gestite da un clock indipendente
- L'accesso non è sincronizzato con le operazioni della CPU e quando i dati sono pronti appaiono sul bus
- Lavorano bene solo a basse velocità (< 66 MHz)
- obsolete

◆ SDRAM: Synchronous DRAM

- le operazioni di accesso alla memoria sono sincronizzate con il clock della CPU
- consentono trasferimenti dati veloci ed affidabili

◆ (Il chipset della) scheda madre deve essere compatibile con il tipo di memoria

Principali tipi di SDRAM

- ◆ **DDR SDRAM: Double Data Rate SDRAM**
 - Raddoppia la velocità di trasferimento dati tra CPU e RAM effettuando due trasferimenti per ogni ciclo di clock
 - Utilizza minore potenza (utile per dispositivi portatili)
- ◆ **DRDRAM: Direct Rambus DRAM**
 - Molto veloce, ma il sistema deve essere modificato.
 - Funziona con frequenze superiori a 500MHz
 - 10 volte più veloce della DRAM standard

La situazione attuale

- ◆ SDRAM (Pentium III e AMD)
 - lavorano con bus a 100 MHz o 133 MHz
 - economiche ma quasi introvabili
- ◆ RDRAM (Pentium 4)
 - lavorano con bus a 400 MHz (due dati per ogni ciclo)
 - più veloce in assoluto ma costosa
- ◆ DDR SDRAM (AMD e Pentium 4)
 - lavorano con bus fino a 200 MHz
 - al momento soluzione più utilizzata
- ◆ DDR2 SDRAM
 - Fanno lavorare la memoria a minore frequenza di DDR e consumano minor energia
 - Soluzione del futuro

Cronologia

DRAM	primi PC
EDO DRAM	Pentium
SDRAM	Pentium MMX
SRAM	memorie cache
RDRAM	Pentium 4
DDR SDRAM	Athlon, Pentium

Moduli di memoria

- ◆ La memoria centrale viene venduta in moduli
 - Circuiti stampati contenenti chip di memoria
 - Vengono installati in appositi slot della scheda madre
- ◆ Il numero di moduli installabili dipende dalla scheda madre
- ◆ Esistono diversi tipi di moduli
- ◆ La scheda madre potrebbe
 - essere compatibile solo con alcuni tipi
 - non consentire di mischiare tipi di moduli diversi
 - accettare solo moduli di una data dimensione e con una data velocità di trasferimento

Tipi di moduli di memoria

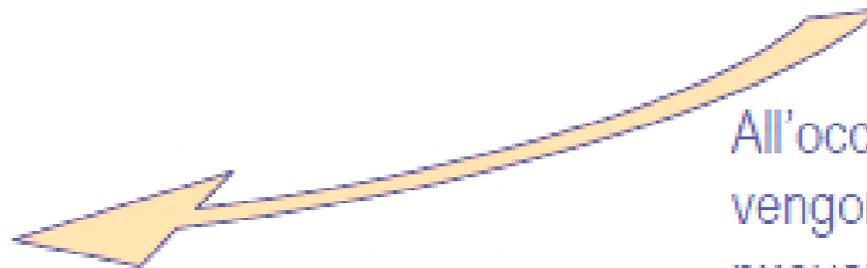
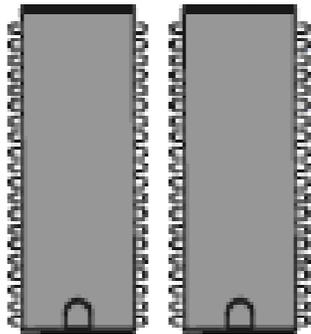
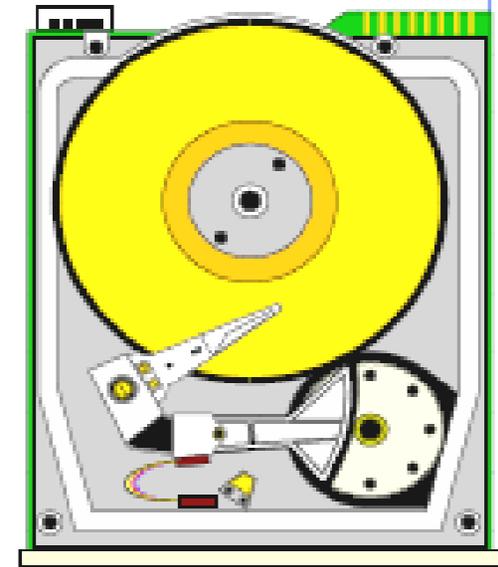
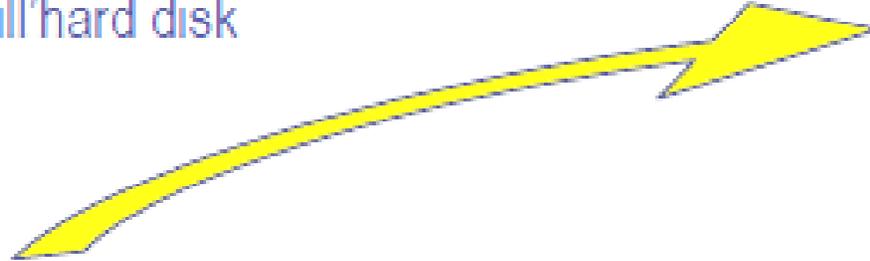
- ◆ SIMM (Single In-line Memory Module)
 - Con i nuovi processori devono essere installati a coppie
- ◆ DIMM (Dual In-line Memory Module)
 - Possono essere installati singolarmente
- ◆ RIMM
 - moduli contenenti memorie RDRAM
 - utilizzabili solo con Pentium 4
 - incompatibili con altri tipi di moduli

Memoria virtuale

- ◆ I programmi possono eccedere la quantità di memoria disponibile
 - una parte dei dati deve essere memorizzata da qualche altra parte, non accessibile direttamente dalla CPU
- ◆ La memoria virtuale consente di
 - **Tenere in memoria centrale solo i dati e le istruzioni attualmente in uso da parte del programma**
- ◆ La memoria virtuale esiste logicamente ma non fisicamente

Memoria Virtuale

I dati in eccesso vengono spostati temporaneamente sull'hard disk

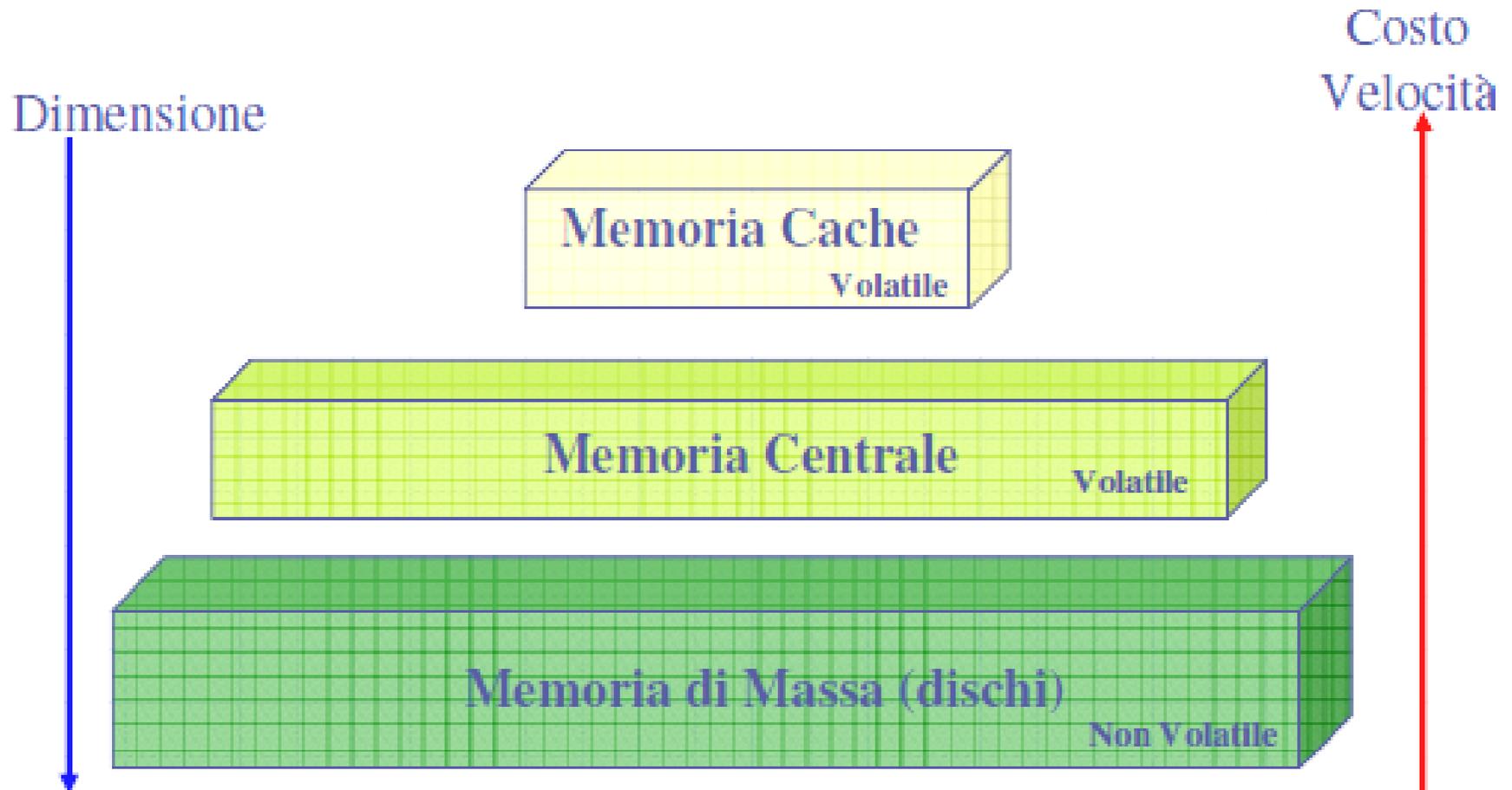


All'occorrenza i dati vengono trasferiti nuovamente nella RAM

Costo della memoria virtuale

- ◆ Ogni volta che un programma vuole accedere ad un dato fisicamente non presente in memoria il sistema operativo deve preoccuparsi di recuperarlo
 - ferma la computazione
 - cerca il dato sull'hard disk e lo copia nella memoria centrale
 - copia un altro dato dalla memoria centrale all'hard disk
- ◆ Accedere all'hard disk richiede molto più tempo che accedere alla memoria centrale
 - Il processore resta in attesa dei dati
- ◆ È inutile avere un processore velocissimo con poca memoria di lavoro

Gerarchie di memoria



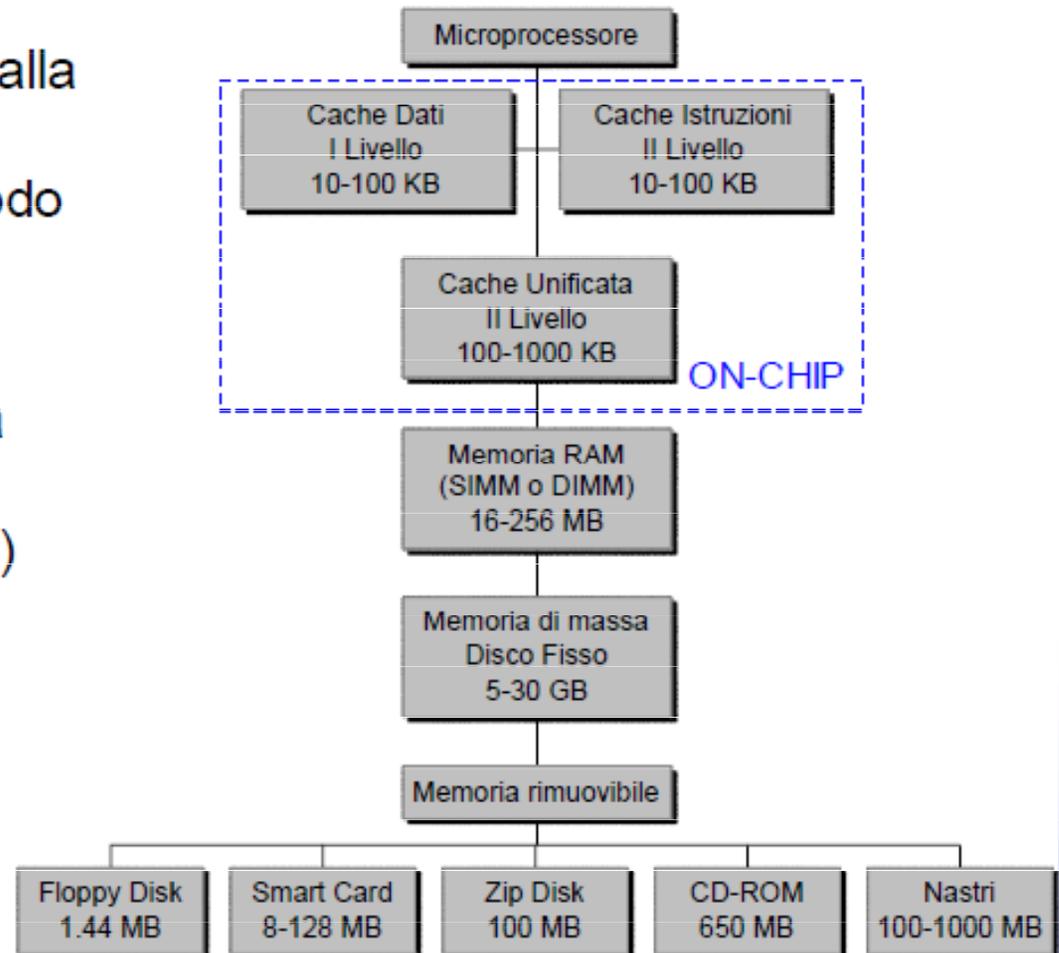
Memoria Cache

- ◆ Memoria ad accesso veloce posta tra processore e RAM.
 - Contiene istruzione e dati di immediato o prossimo utilizzo.
 - Nella cache ci sono sia il dato che informazioni sul suo indirizzo.
 - Il processore cerca prima nella cache poi nella RAM
- ◆ Esistono cache di primo, secondo, (terzo) livello.
 - Il processore esamina i livelli in sequenza e se non trova un dato al livello i lo cerca al livello $i+1$
 - Cache di primo livello (L1)
 - installata sullo stesso chip del processore
 - in genere di 32 Kb
 - Cache di secondo livello (L2)
 - installata sulla scheda madre o direttamente sul chip
 - in genere di 256 Kb
 - Processori economici hanno una cache L2 più piccola

Gerarchia di memoria

Gerarchia di memoria

- Per consentire al microprocessore di lavorare alla velocità più alta possibile, la memoria è organizzata in modo gerarchico
- I diversi livelli della gerarchia (procedendo dal microprocessore a scendere) sono caratterizzati da
 - *Velocità decrescente*
 - *Dimensione crescente*
- Forniscono l'illusione di una **memoria infinitamente grande e veloce**.

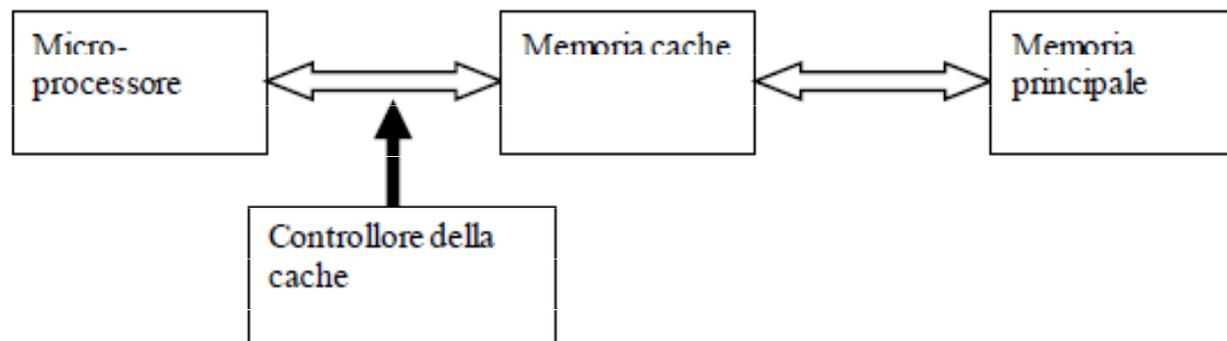


Memoria principale - Cache

- Memoria piccola e molto veloce nata per compensare la strozzatura nella comunicazione tra RAM e processore.
- Principio di località:
 - *Quando il processore utilizza un'istruzione o dato è molto probabile che usi anche quelli ad esso vicini nella memoria (località spaziale).*
 - *Quando il processore utilizza un'istruzione o dato è molto probabile che lo usi di nuovo in breve tempo (località temporale).*
- Il principio di località ispira il funzionamento della cache che contiene il blocco di dati ed il blocco di istruzioni immediatamente vicini al dato (o all'istruzione) corrente. Essi verranno conservati per un certo tempo e poi scartati se non più utilizzati.

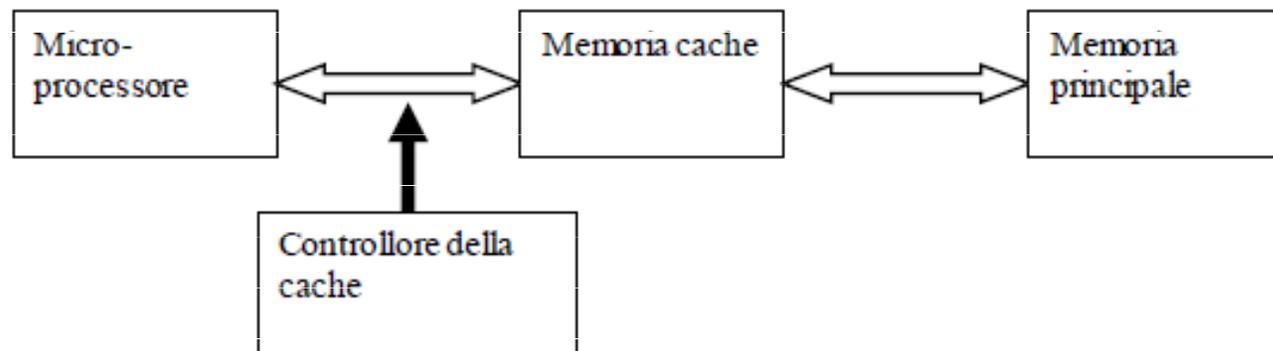
Memoria principale - Cache

- Memoria piccola e molto veloce (SRAM che è più veloce e costosa della DRAM).
- Ogni volta che una parola viene letta dalla memoria essa viene copiata nella cache ed insieme vengono copiate quelle immediatamente precedenti e seguenti secondo un opportuno schema di indirizzamento per “blocchi di parole”.
- Viene creata una lista ordinata di utilizzo dei blocchi e, quando necessario per fare spazio, il blocco in coda (non utilizzato da più tempo) viene eliminato.



Memoria principale - Cache

- Il controllo della cache viene svolto dal chipset
- La cache L1 di solito è integrata nel chip del microprocessore . La cache L2 in genere è realizzata su un chip separato (talvolta saldato sulla stessa cartuccia del microprocessore)
- Dimensione
 - *512 Kb o più (ormai almeno 1MB sui processori delle macchine di uso comune)*
- Clock
 - *Spesso lavora ad un clock maggiore della scheda madre*
 - *CPU 2 – 3 GHz,*
 - *Cache II liv. e RAM: 233 Mhz*
 - *scheda madre (bus PCI) 66 Mhz*



Memoria principale - Cache

➤ Aggiornamento della cache

- *Write-back*
- *Write-through*

➤ Write-back

- *La cache aggiorna il proprio contenuto subito ed aggiorna la RAM quando deve svuotarsi oppure quando una periferica esterna chiede di accedere al blocco di memoria RAM copiato in cache*

➤ Write-through

- *La cache aggiorna il proprio contenuto e quello in memoria ogni volta che riceve un nuovo dato*