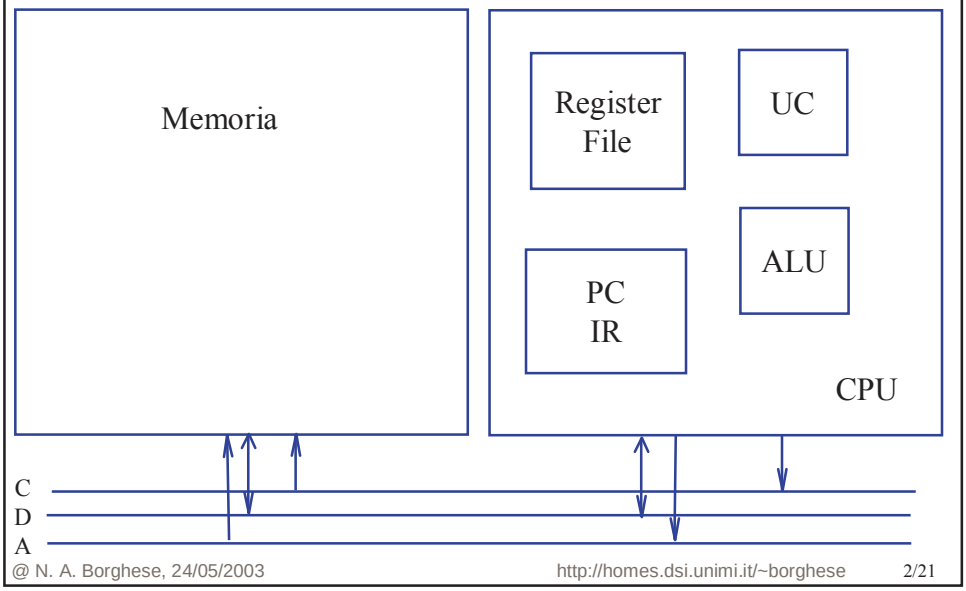
Memoria

Architettura
degli Elaboratori e delle Reti,
Turno I

Alberto Borghese
Università degli Studi di Milano
Dipartimento di Scienze dell'Informazione
email: borghese@dsi.unimi.it

Gli attori principali di un'architettura



Principio di località

I programmi riutilizzano dati e istruzioni che hanno usato di recente.

Regola pratica: un programma spende circa il **90%** del suo tempo di esecuzione per solo il **10%** del suo codice.

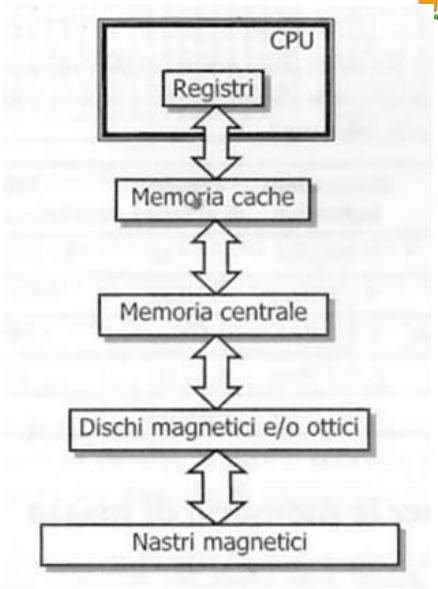
Basandosi sul passato recente del programma, è possibile predire con ragionevole accuratezza quali dati e istruzioni userà nel prossimo futuro.

Località temporale: elementi ai quali si è fatto riferimento di recente saranno utilizzati ancora nel prossimo futuro.
Località spaziale: elementi i cui indirizzi sono vicini, tendono ad essere referenziati in tempi molto ravvicinati.

Gerarchia di memorie

Nel livello superiore troviamo un sottoinsieme dei dati del livello inferiore.

Ciascun livello vede il livello inferiore.



Gerarchia di memorie - caratteristiche

Livello	Dimensioni indicative	Tempo di Accesso	Velocità di Trasferimento (Mbyte/s)
Registri	< 1 Kbyte	< 0,01 ns	400,000 (4 byte)
Cache Primaria (Pentium 4, 3Ghz) ExecTrace cache	8kbyte 12kbyte	0.16ns	192,000 (32 byte)
Cache Secondaria (Pentium 4, 3Ghz)	256-512 kbyte	0.3ns	96,000 (32 byte in parallelo)
Memoria centrale (RAM, DRAM)	< 4 Gbyte	< 3-5 ns (233Mhz / 320Mhz)	1,600 – 3,000 (DDSRAM – doppia lettura)
Bus PCI Bus PCI 64	133 Mhz 100 Mhz	8 byte 64 byte	1,060 (8 byte) 6,400 (64 byte)
Dischi	> 50 Gbyte	< 10ms	< 200 (Seagate Cheetah SCSI)
Nastri	> 100 Gbyte	> 100ms	1

Tassonomia del funzionamento

HIT Successo nel tentativo di accesso ad un dato: è presente al livello superiore della gerarchia.

MISS Fallimento del tentativo di accesso al livello superiore della gerarchia => l'indirizzo deve essere cercato al livello inferiore.

HIT_RATE Percentuale dei tentativi di accesso ai livelli superiori della gerarchia che hanno avuto successo.
HIT_RATE=Numero_successi/Numero_accessi_memoria

MISS_RATE Percentuale dei tentativi di accesso ai livelli superiori della gerarchia che sono falliti
MISS_RATE=Numero_fall./Numero_accessi_memoria

HIT_RATE + MISS_RATE = 1

Sottosistema di memoria

Porta nella cache primaria i dati richiesti mentre il binomio processore-memoria sta lavorando.

1) Controlla se un dato è in cache (Hit).

2) Porta un dato in cache dal livello inferiore (Miss).

Principio di funzionamento della cache

lw $t0, 0($s4)

X_n -> $t0

Indirizzo Cache	*Prima del caricamento di Xn*	*Dopo il caricamento Xn*
000	X_4	X_4
001	X_1	X_1
010	X_{n-2}	X_{n-2}
011		X_n
100	X_2	X_2
101	X_3	X_3
110		
111	X_{n-1}	X_{n-1}

Problemi:

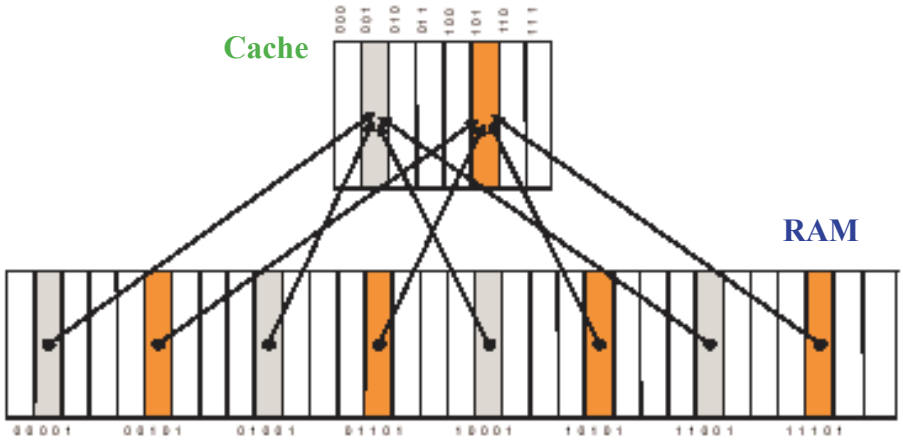
•Corrispondenza tra 0($s4) e l'indirizzo di cache.

•Come si fa a sapere se un dato è già in cache?

•Come si fa a recuperare un dato nella cache?

Corrispondenza diretta (direct mapped)

Ad ogni indirizzo di RAM corrisponde un indirizzo di cache.



Indirizzi diversi di RAM corrispondono allo stesso indirizzo di cache.

Come si può recuperare un dato dalla cache?

Mediante operazione di modulo:

Indirizzo_cache =
indirizzo_RAM (in #parole) *modulo* #parole_cache.

Esempio:

lw $t0, 65($zero) => Indirizzo in cache di 4 parole è:
65 appartiene alla 17a parola in RAM.
indirizzo_cache = 1 (17 modulo 4).

Nel caso in cui la base sia binaria:

Indirizzo_cache = least_significant_bits (tolti quelli di offset all'interno della parola)

#Indirizzi è log2(#parole_cache)

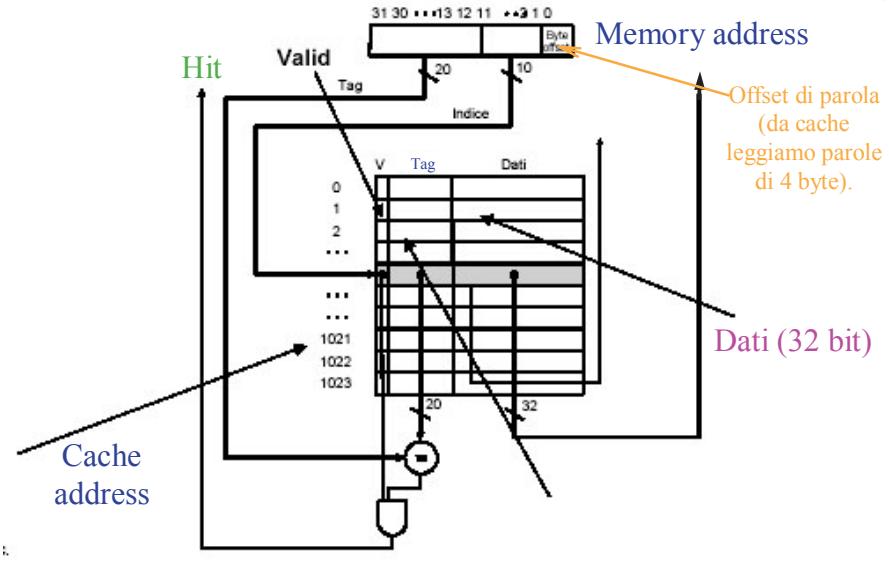
Come si può sapere se un dato è presente in cache?

Aggiungiamo a ciascuna parola della memoria, un campo (***tag***).

Il tag contiene i bit che costituiscono la parte più significativa dell'indirizzo.

Occorre anche l'informazione dato valid / not_valid: ***bit di validità***.

Circuito di lettura/scrittura cache



Esercizi

Sia data una cache a corrispondenza diretta contenente 64Kbyte di dati e avente blocchi di 1 parola. Assumendo che gli indirizzi siano di 32 bit quale è il numero totale di bit richiesto per l'implementazione della cache?

Supponendo che il MIPS abbia una cache di 512byte, indicare cosa succede nei campi della cache quando vengono eseguite le seguenti istruzioni:

lw $t1, 0x0000($t0) $t0 = 1kbyte = 1,024 byte
lw $t1, 0x0000($t0) $t0 = 0
lw $t1, 0x0202($t0) $t0 = 1kbyte = 1,024 byte
lw $t1, 0x0001($t0) $t0 = 0
lw $t1, 0x0201($t0) $t0 = 1kbyte = 1,024 byte

Gestione dei fallimenti di una cache

Hit – è quello che vorremmo ottenere, il funzionamento della CPU non viene alterato.

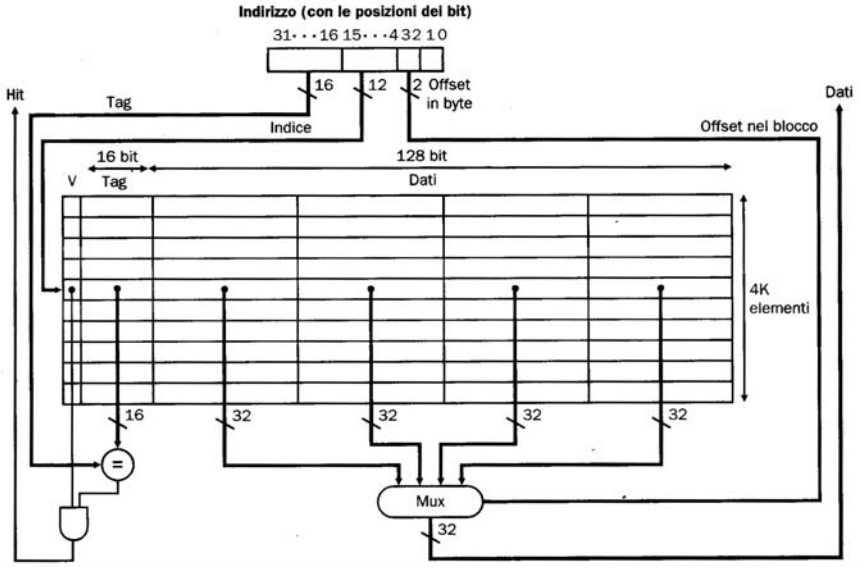
Miss – in lettura devo aspettare che il dato sia pronto in cache -> stallo.

Passi da eseguire in caso di Miss:
1) Ricaricare l'indirizzo dell'istruzione (PC-> PC-4)
2) Leggere il blocco di memoria dalla memoria principale.
3) Trasferire il blocco in cache, aggiornare i campi validita' e tag.
4) Riavviare la fase di fetch dell'istruzione.

NB Il programma non può continuare!!

In scrittura?

Cache a blocchi, funzionamento



Come migliorare le prestazioni di una cache?

Tempi di accesso:

1 ciclo di clock per inviare l'indirizzo.

15 cicli di clock per ciascuna attivazione della RAM (lettura di 1 parola).

1 ciclo di clock per trasferire una parola al livello superiore (cache).

Blocco di cache di 4 parole, blocco di memoria RAM di 1 parola

$\Downarrow$

Miss_Penalty = 1 + 15 * 4 *(parole)* + 1 * 4 *(parole)* = *65 cicli_clock*

#byte / ciclo_clock = 4(*parole*) * 4(*byte/parola*) / 65(*cicli_clock*) = 0,25(*byte / ciclo_clock*)

Diminuire la penalità di fallimento (miss_penalty).

Diminuire il tasso di fallimento (miss_rate).

Diminuzione della "miss_penalty"

Architettura standard: penalità di miss è di 65 cicli_clock.

Maggiore ampiezza della memoria:
• Organizzazione della RAM per blocchi.
• Bus più ampio (bus dati largo un blocco).
• Per blocchi di RAM di 2 parole, blocchi di cache di 4 parole:
Miss_penalty = 1 + 15 * 2 + 1 * 2 = 33 cicli_clock.
#byte / ciclo_clock = 4(*parole*) * 4(*byte/parola*) / 33(*cicli_clock*) = 0,48(*byte / ciclo_clock*)

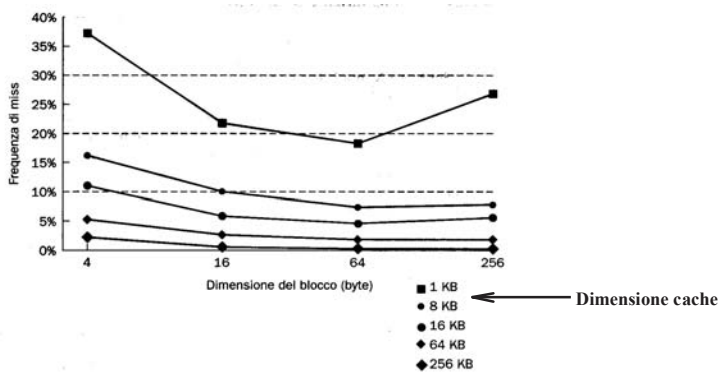
Interleaving:
•Organizzazione della RAM per banchi con accesso indipendente alla memoria.
• Bus standard (trasferimento di 1 parola alla volta).
• Per blocchi di RAM di 1 parola, blocchi di cache di 4 parole:
Miss_penalty = 1 + 15 * 1 + 1*4 = 20 cicli_clock.
#byte / ciclo_clock = 4(*parole*) * 4(*byte/parola*) / 20(*cicli_clock*) = 0,80(*byte / ciclo_clock*)

Miss rate e cache a blocchi

La parola di cache (blocco) è un multiplo della parola della macchina.

Vantaggi: per la località spaziale, diminuisco la frequenza di miss.

*Svantaggi:*per le dimensioni del blocco aumenta la penalità di miss.



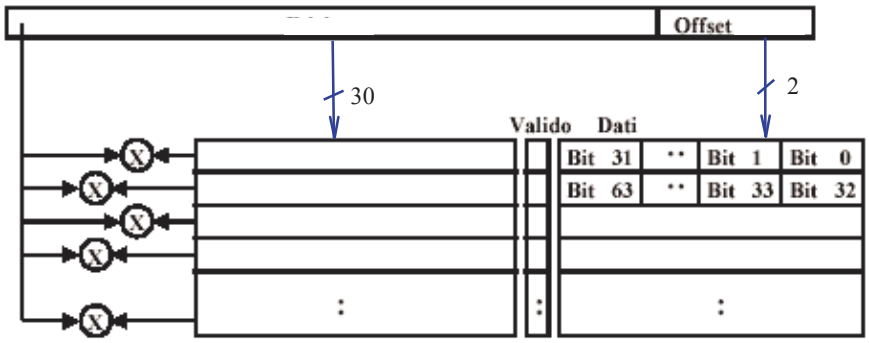
Diminuzione della "miss_rate"

• A corrispondenza diretta: un blocco di memoria può essere scritto in un'unica posizione nella cache.

• Completamente associative: un blocco di memoria può essere trasferito in un qualsiasi blocco della cache.

• n-associative: un blocco di memoria può essere trasferito in un insieme ridotto di possibili blocchi della cache (almeno 2).

Accesso alle memorie associative



Tramite comparatori individuo in quale blocco si trova il mio dato.
Il segnale di Hit si genera come AND (comparatore_output, Valido)

Dove scrivo il blocco?

Cache

Principio di località.

Copia di porzioni di RAM.

Organizzazione a blocchi.

Direct mapped o associative.