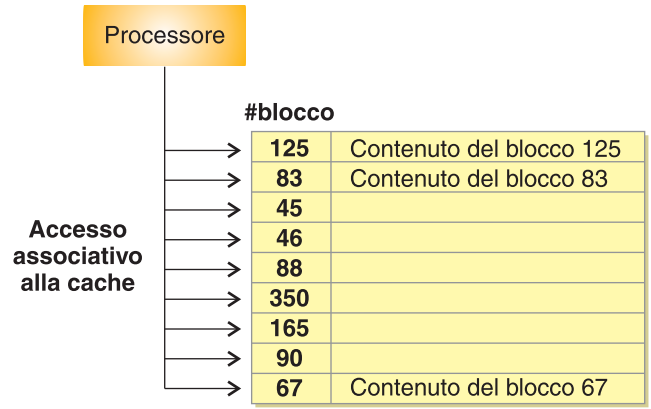
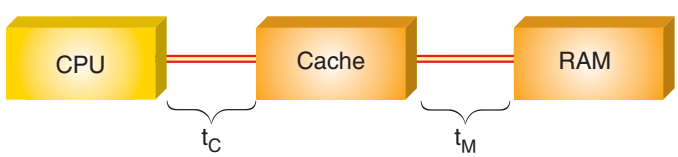
Prestazioni della cache e memorie a più livelli

In una memoria a due livelli, quando la parola di memoria cercata si trova nella *cache* il suo trasferimento avviene in tempi molto brevi, dell'ordine di qualche ns. Se invece il dato deve essere prelevato dalla memoria centrale il trasferimento è molto più lento, dell'ordine di decine di ns. Il tempo di accesso medio dipende, naturalmente, dalla percentuale di successi (**cache hit**) nel trovare i dati in *cache*. Nella pratica si ottengono percentuali di successo superiori al 95%, quindi il tempo di accesso medio è di poco superiore al tempo di accesso alla *cache*.
La ragione alla base di percentuali di successo così elevate sta nei **principi di località** visti in precedenza.
I dati sono messi nella *cache* per blocchi, per esempio in blocchi di 8 parole di memoria. Il processore per trovare una parola di memoria deve identificare il blocco di appartenenza e poi scoprire se quel blocco è nella *cache*. La prima operazione è semplice: se gli indirizzi di memoria sono di 21 bit, la suddivisione della memoria in blocchi di 8 parole porta a collocare nel medesimo blocco tutte quelle parole il cui indirizzo coincide per i 18 bit più significativi (cioè i 18 bit più a sinistra dell'indirizzo). Questi bit costituiscono, insomma, un **indirizzo di blocco**.



La cache è formata da un certo numero di **linee di cache**. La figura mostra una cache con 9 linee, in grado, quindi, di memorizzare 9 blocchi da 8 parole ciascuno. Un nuovo blocco è collocato in una qualsiasi linea di *cache* libera, se ce ne sono, sopra un'altra se la *cache* è piena. Per identificare i blocchi in ogni linea di *cache*, oltre alle 8 parole che compongono il blocco, è memorizzato anche l'indirizzo del blocco.
Il processore per trovare un blocco accede in parallelo a tutte le linee di *cache* e riesce a scoprire, con una sola lettura, se il blocco cercato è nella *cache* e, in tal caso, a prelevare la parola cercata nel blocco. Si dice che l'accesso alla *cache* avviene in **modo associativo**. Con questo termine si fa riferimento a una modalità di accesso alla memoria che permette di recuperare i dati in base al loro valore e non, come avviene in memoria centrale, in base alla posizione nella memoria.
Per meglio comprendere come funziona un sistema di memoria a due livelli, valutiamo il tempo medio di accesso di una memoria con *cache*. Indichiamo con $\mathbf{t_c}$ il tempo di accesso alla *cache* e con $\mathbf{t_M}$ il tempo necessario per trasferire un blocco di dati dalla RAM alla *cache*. Se la parola di memoria cercata è nella *cache* il trasferimento nella CPU avviene nel tempo $\mathbf{t_c}$. Quando la parola cercata non è in *cache*, essa viene prima trasferita nella *cache* e poi nel processore. In questo secondo caso il tempo di accesso diventa: $\mathbf{t_c} + \mathbf{t_M}$.



© Istituto Italiano Edizioni Atlas

Indichiamo ora con **p** la probabilità di successo (**cache hit**) e consideriamo **N** accessi alla memoria. Di questi: **N·p** avvengono accedendo direttamente alla cache, mentre **N·(1-p)** richiedono di accedere alla RAM. Considerando **N** accessi, il tempo medio di accesso T_m è dato da:

$$T_m = \frac{N \cdot p \cdot t_c + N \cdot (1 - p) \cdot (t_c + t_M)}{N}$$
$$= p \cdot t_c + (1 - p) \cdot (t_c + t_M)$$
$$= p \cdot t_c + t_c - p \cdot t_c + (1 - p) \cdot t_M$$
$$= t_c + (1 - p) \cdot t_M$$

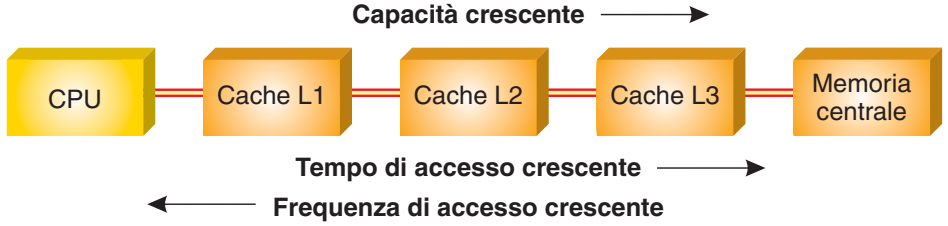
La relazione che esprime il valore di T_m può essere interpretata in questo modo: il tempo medio di accesso alla memoria è maggiore di quello alla *cache* della quantità $(1 - p) \cdot t_M$ che dipende, secondo il fattore t_M, da $(1 - p)$, cioè dalla probabilità di insuccesso nell'accedere alla *cache* (**cache miss**). Calcoliamo il tempo medio di accesso alla memoria ipotizzando una percentuale di successi del 90%. In questo caso si ha: $p = 0.9$ e $(1 - p) = 0.1$. Se la cache ha tempo di accesso di 3 ns e la memoria RAM di 30 ns, il tempo medio di accesso alla memoria con *cache* diventa:

$$T_m = t_c + 0{,}1 \cdot t_M = 3 + 0{,}1 \cdot 30 = 6 \text{ ns}$$

Il risultato dimostra che il sistema a due livelli permette di realizzare una memoria con tempi di accesso medi non troppo lontani da quelli di una memoria *cache* di capacità pari a quella della memoria centrale.
Si tenga poi presente che le percentuali di successo effettivamente osservate sono anche superiori al 95%. Con un *cache hit* di 0.95, $(1-p)=0.05$ si avrebbe:

$$T_m = t_c + 0{,}05 \cdot t_M = 3 + 0{,}05 \cdot 30 = 4{,}5 \text{ ns}$$

Dalla espressione del tempo medio di accesso: $T_m = t_c + (1 - p) \cdot t_M$ si osserva che, per ridurre T_m, si può diminuire t_c oppure aumentare **p**.
La riduzione del tempo di accesso alla *cache* è legata alla tecnologia delle memorie, mentre la seconda ipotesi richiede di costruire memorie *cache* sufficientemente grandi da intercettare una frazione sempre maggiore di accessi. Queste considerazioni hanno portato a costruire sistemi di **memorie a più livelli**.



Il processore, per usare una parola di memoria, la cerca nella *cache* di primo livello (L1). Se la parola cercata non c'è in L1, la cerca nella *cache* di secondo livello (L2) e, in caso di esito ancora negativo, nella *cache* di terzo livello (L3) e infine in memoria centrale.
Nel sistema di memorie la *cache* di livello 2 è più lenta della *cache* di livello 1 e, analogamente, la *cache* di livello 3 è più lenta della *cache* di livello 2, ma più veloce della memoria centrale.
La *cache* L1, di dimensione tipica di 64-256KB, è posta nello stesso *chip* del processore, mentre la *cache* L2, di dimensioni 512 KB-2 MB, è collocata nel package che include il *chip*; infine la *cache* L3, avente le dimensioni di alcuni MB, è collocata nella stessa scheda della CPU.

© Istituto Italiano Edizioni Atlas

Domande

1. Che cosa significa accesso associativo a una memoria?
   a) L'accesso avviene sia in memoria che nella cache.
   b) L'accesso avviene in base all'indirizzo del dato associato al contenuto del registro indirizzi di memoria.
   c) L'accesso alla memoria avviene in base al suo contenuto.
   d) L'accesso avviene sia in memoria che nel disco.
   e) Cerca dati con lo stesso valore in posizioni differenti della memoria.
   Risposta: c)

2. Perchè ci sono le memorie a più livelli?
   a) Per distribuire i dati su più memorie.
   b) Per realizzare memorie con tempi di accesso medi vicini alla più veloce e con capacità pari alla più grande.
   c) Perchè la memoria centrale è volatile.
   d) Per duplicare i dati su più memorie.
   e) Per avere una copia in memoria dei dati su disco.
   Risposta: b)

Problema

1. Si consideri una memoria con tempo di accesso di 100 ns e una cache con tempo di accesso di 8 ns. Preparare un tabella che riporta il tempo medio di accesso in funzione del valore di cache hit, facendolo variare da 0,15 a 0,95 con un passo di 0,01.

© Istituto Italiano Edizioni Atlas