**Parte I - Capitolo 5 - La Gerarchia Di Memoria**

Il comportamento alla base dei programmi di un calcolatore è reso possibile dal ***principio di località*** che afferma che un programma, in un certo istante di tempo, accede soltanto a una porzione relativamente piccola del suo spazio di indirizzamento. Questo principio può essere diviso in due diversi tipi: la ***località temporale*** (se si accede a una determinata locazione di memoria, è molto probabile che vi si acceda di nuovo dopo poco tempo) e la ***località spaziale*** (se si accede a una determinata locazione di memoria, è molto probabile che si acceda alle locazioni vicine a essa dopo poco tempo). Il principio di località viene sfruttato strutturando la memoria del calcolatore in forma gerarchica. La ***gerarchia di memoria*** consiste in un insieme di livelli di memoria, ciascuno caratterizzato da una diversa velocità e dimensione: a parità di capacità, le memorie più veloci hanno un costo più elevato per singolo bit di quelle più lente, perciò di solito sono più piccole. Le tecnologie maggiormente utilizzate sono: la ***DRAM*** (***Dynamic Random Access Memory***) utilizzata per la realizzazione della memoria principale, la ***SRAM*** (***Static Random Access Memory***) utilizzata per i livelli più vicini al processore come la cache, e i ***dischi magnetici*** che implementano il livello di memoria più capiente. Le SRAM sono le memorie più veloci, più piccole e più costose, mentre i dischi magnetici sono i più lenti, ma con dimensioni più elevate e un costo più basso. Una gerarchia di memoria può essere strutturata tra più livelli, ma lo scambio di dati può avvenire soltanto tra due livelli vicini. La più piccola quantità di informazione presente nella gerarchia a due livelli è detta ***blocco***. Se il processore richiede un blocco al livello superiore e la richiesta ha successo si ha una ***hit***, in caso contrario si ha una ***miss***. La frequenza di hit viene chiamata ***hit rate*** ed è la frazione degli accessi alla memoria nei quali il dato desiderato è stato trovato nel livello superiore. La frequenza delle miss è detta ***miss rate*** ed è uguale a 1-hit rate, ed è la frazione degli accessi alla memoria nei quali il dato desiderato non è stato trovato nel livello superiore. Poichè con l'organizzazione gerarchica della memoria si punta a un aumento delle prestazioni, la velocità degli accessi è importante sia in caso di hit sia in caso di miss. Il ***tempo di hit*** è il tempo di accesso al livello superiore della gerarchia di memoria, e comprende anche il tempo necessario a stabilire se il tentativo di accesso di risolve in un successo o in un fallimento. La ***penalità di miss*** è il tempo necessario a sostituire un blocco del livello superiore con un nuovo blocco, caricato dal livello inferiore della gerarchia, e a trasferire il dati contenuti in questo blocco al processore. Il tempo di hit è generalmente molto inferiore al tempo di miss, poichè il livello superiore è più piccolo e più veloce.

Con il termine ***cache*** si indica il livello di memoria gerarchica che si trova tra il processore e la memoria principale. Una cache contiene un insieme di elementi utilizzati di recente e può provocare un miss quando il processore richiede una parola non presente al suo interno. La ricerca del dato nella cache è molto semplice in quanto un dato può essere scritto in una sola posizione; infatti l'organizzazione della cache è detta ***a mappatura diretta*** (***direct mapped cache***), cioè a ogni locazione della memoria principale corrisponde in modo univoco una locazione della cache. Poichè ogni elemento della cache può contenere elementi provenienti da diverse locazioni della memoria principale, si pone il problema di riconoscere effettivamente il dato ricercato. A questo proposito vengono utilizzati un insieme di campi detti ***tag*** (***etichette***) che contengono le informazioni necessarie a verificare se una parola della cache corrisponde alla parola cercata. Solitamente un tag contiene la parte superiore dell'indirizzo della parola (i bit più significativi). È necessario sapere anche quando un blocco della cache non contiene informazioni valide. Le procedure più comuni risolvono il problema aggiungengo un bit di validità per indicare se il blocco della cache contiene elementi validi.

La gestione delle miss di una cache viene svolta dall'unità di controllo che deve riconoscere il fallimento del tentativo di accesso e provvedere a risolverlo andando a prelevare i dati dalla memoria principale o da una cache del livello inferiore. La risoluzione di una miss richiede lo stallo della pipeline: si mette in stallo il processore bloccando il contenuto dei registri temporanei e di quelli visibili al programmatore per tutto il tempo necessario a caricare i dati dalla memoria principale. Quando si verifica una miss di un'istruzione devono essere eseguiti i seguenti passi: inviare il valore originale alla memoria (program counter - 4), comandare alla memoria principale di eseguire un'operazione di lettura e attendere che la memoria termini la lettura, scrivere la parola che proviene dalla memoria nella posizione opportuna del blocco della cache, far ripartire l'esecuzione dell'istruzione dall'inizio ripetendo la fase di fetch. Per la scrittura il funzionamento è diverso. Se il dato scritto nella cache non è riportato nella memoria, allora la memoria e la cache sono incoerenti. Uno schema secondo il quale un elemento viene scritto sia in cache che nel livello inferiore della memoria, assicurando così che i dati presenti nelle due memorie siano sempre coerenti è detto ***write-through***. Un altro elemento fondamentale della scrittura è la gestione della miss. Dopo aver caricato il blocco e averlo scritto nella cache, possiamo sovrascrivere la parola del blocco che aveva causato la miss. Questo meccanismo non offre buone prestazioni, quindi si utilizza una memoria tampone detta ***buffer di scrittura*** che memorizza i dati in attesa che essi vengano scritti in memoria. Dopo aver scritto il dato nella memoria principale il corrispondente spazio nel buffer di scrittura viene liberato. Se il buffer di scrittura è pieno il processore viene messo in stallo finchè non si libera uno spazio nel buffer. Lo schema alternativo al write-through è detto ***write-back*** o ***capy-back*** e consiste nello scrivere il blocco solamente nella cache. Il blocco modificato viene salvato nel livello inferiore solo quando deve essere rimpiazzato. Questa tecnica può produrre un miglioramento delle prestazioni, ma risulta più complesso da implementare.

Passiamo ora a progettare il sistema di memoria per supportare una cache. Le miss delle cache devono essere risolte dalla memoria principale. È possibile ridurre la penalità di miss aumentando la larghezza della banda di trasferimento tra la memoria e la cache. La frequenza di clock del bus che collega il processore alla memoria influenza la penalità di miss. Quindi invece di allargare la banda dell'intero cammino che va dalla memoria al processore, i chip di memoria possono essere organizzati in banchi, in modo tale da poter leggere e scrivere più parole a ogni accesso. Ogni banco può essere lungo una sola parola in modo da non dover modificare la larghezza del bus e della cache e si possono leggere più parole memorizzate in banchi diversi, contemporaneamente. Questo schema è detto ***memoria interallacciata*** (***interleaved***) e mantiene il vantaggio di dover pagare una sola volta il costo della latenza nella memoria principale. In seguito alla diffuzione massiccia delle cache, per disporre di una larghezza maggiore dei blocchi, i produttori di DRAM hanno dotato le memorie di un accesso a raffica (burst) ai dati contenuti in posizione adiacente. La tecnologia più recente è la ***DDR*** (***Double Data Read***) in cui i dati vengono trasferiti sia sul fronte di salita che sul fronte di discesa del clock ottenendo, ottenendo così una banda di larghezza doppia. La DRAM è organizzata in banchi di memoria interallacciati. Queste tecnologie offrono il vantaggio di utilizzare dei circuiti già presenti nelle DRAM, incrementando di poco il costo del sistema e allo stesso tempo aumentando in modo significativo la larghezza di banda.

Esistono diversi modi per misurare e analizzare le prestazioni di una cache. Esaminiamo due diverse tecniche per incrementare le prestazioni. La prima tecnica è basata sulla riduzione della frequenza delle miss, ottenuta riducendo la probabilità che due differenti blocchi di memoria principale entrino in conflitto per la stessa locazione della cache. La seconda tecnica riduce la penalità ddelle miss introducendo nella gerarchia un livello aggiuntivo. Questo secondo metodo è chiamato ***cache multilivello*** (***multilevel caching***). il tempo di CPU è definito nel modo seguente:

I cicli di stallo sono dovuti principalmente alle miss della cache. Sono definibili come:

I cicli di stallo in lettura e scrittura sono definiti nel seguente modo:

Per tenere conto del fatto che il tempo di accesso ai dati influenza le prestazioni sia nel caso di miss sia nel caso di hit, i progettisti spesso utilizzano il ***tempo medio di accesso alla memoria*** (***AMAT***, ***Average Memory Access Time***) che può essere descritto come:

Tornando alla struttura della cache, oltre alla mappatura diretta, esiste uno schema nel quale un blocco della memoria principale può essere scritto in una qualsiasi locazione della cache che viene detto ***cache completamente associativa***, perchè un blocco della memoria può essere associato a un qualsiasi blocco della cache. La ricerca in una cache di questo tipo deve essere effettuata su tutti gli elementi della cache. Per rendere la ricerca più efficace, questa viene svolta in parallelo utilizzando un comparatore per ogni blocco di cache. Il costo dell'hardware aumenta significativamente, quindi lo schema completamente associativo è realizzabile solo per cache con un numero ridotto di blocchi. Uno schema intermedio tra quelli visti fin ora è detto ***cache set-associative***. In questo caso, ogni blocco può essere caricato in un numero prefissato di posizioni alternative. Ciascun blocco della memoria principale viene mappato su un'unica linea della cache, individuata nel campo indice, e il blocco può essere scritto in uno qualsiasi dei blocchi costituenti la linea. Dato che il blocco può essere scritto in qualsiasi blocco di una linea, la ricerca deve essere effettuata analizzando i campi tag di tutti i blocchi della linea. Possiamo facilmente comprendere che una cache a mappatura diretta non è altro che una cache set-associative a una sola via, così come una cache di m elementi completamente associativa è una cache set-associative a m vie. Consideriamo ora il problema di trovare un blocco all'interno di una cache set-associative. Come per la cache a mappatura diretta, viene esaminato il contenuto del campo tag in parallelo per tutti i blocchi della linea selezionata: una ricerca sequenziale renderebbe eccessivo il tempo di hit di una cache set-associative. Una volta selezionati i contenuti dei tag, bisogna compararli tra loro attraverso un multiplexer che selezionerà il dato desiderato. Per quanto riguarda la sostituzione di un blocco quando si verifica una miss, per una cache set-associative è possibile scegliere dove scrivere il blocco richiesto e quindi possiamo scegliere quale blocco sostituire. Lo schema descritto è quello più usato ed è detto ***schema utilizzato meno di recente*** (***LRU***, ***Least Recently Used***)

La memoria principale può agire come una cache della memoria di massa, che di solito è costituita da dischi magnetici. Questa tecnica è chiamata ***memoria virtuale***. Questa tecnica ha il fine di proteggere i programmi che condividono la stesa memoria l'uno dall'altro, assicurando che un programma possa solamente scrivere o leggere le porzioni di memoria principale che gli sono stati assegnati. La memoria principale conterrà solo le parti attive dei programmi.