**GERARCHIA DELLE MEMORIA**

**CACHE**

**Che cos’è il caching e come funziona?**

La memoria cache si basa sul principio di località, cerca di trovare i dati desiderati nei livelli più alti della gerarchia delle memorie, fornisce i meccanismi necessari per garantire che, quando un dato non è presente nel livello superiore, trovi e utilizzi i dati corretti presi dai livelli inferiori della gerarchia.

**L’unità minima di una cache? Dimensione di un blocco della cache?**

Blocco o linea è l’unità minima di informazione che può essere presente o assente in una cache.

**Cos’è una cache direct mapped e com’è fatta? Qual è la regola per trovare un blocco in una cache direct?**

È un tipo di implementazione di una cache in cui a ogni locazione della memoria principale corrisponde in modo univoco una locazione di memoria della cache.

Oltre ai dati richiesti, la cache deve permettere la memorizzazione di informazioni aggiuntive (tags)

I tag contengono le informazioni necessarie a verificare se una parola della cache corrisponde o meno alla parola cercata.

Per associare univocamente una parola ad una posizione in una cache direct mapped è sufficiente memorizzare, assieme al dato, anche la parte più significativa dell’indirizzo, quella che non corrisponde ai bit utilizzati come indice nella cache.

Per distinguere le posizioni della cache contenenti dati validi (es.all’avvio del processore la cache è vuota ed il suo contenuto è privo di significativo), il metodo più comune è aggiungere un bit di validità.

(Indirizzo del blocco) modulo (numero di blocchi nella cache)

**Cos’è una cache full associative?**

Cache full associative (completamente associativa): ogni blocco può essere collocato in qualsiasi locazione della cache

Vantaggio: massima utilizzazione della cache

Svantaggio: per ricercare un blocco nella cache è necessario cercarlo in tutte le linee della cache. La ricerca sequenziale è troppo lenta.

Per velocizzare la ricerca è necessario effettuarla in parallelo, associando un comparatore ogni blocco della cache.

**Perché le cache full associative non si usano mai?**

Perché per ricercare un blocco nella cache è necessario cercarlo in tutte le linee della cache. La ricerca sequenziale è troppo lenta.

**Cos’è una cache set associative?**

Sono delle cache che adottano schemi intermedi tra la mappatura diretta e quella completamente associativa.

Una cache set associative a n vie è costituita da un certo numero linee, ciascuna costituita da n blocchi.

Ogni blocco della memoria principale viene mappato (può essere caricato in un numero prefissato di posizioni alternative) su un'unica linea della cache e il blocco di dati può essere scritto in uno dei qualsiasi blocchi che costituiscono la linea.

**MODALITà DI SCIRTTURA**

**Differenza tra write-through e write-back?**

Le operazioni di scrittura devono essere gestite in maniera più complessa rispetto alle letture. Se le scritture alterassero solo lo stato della cache e non venissero propagate fino alla memoria principale si creerebbero situazioni di incoerenza.

Abbiamo due modalità di scrittura

Write-through: uno schema secondo il quale un elemento viene scritto sia in cache che in memoria principale assicurando così che i dati presenti nelle due memorie siano sempre coerenti tra di loro.

La scrittura sia in cache che in memoria principale richiede molto tempo.

Una possibile soluzione a questo problema consiste nell’utilizzare una memoria tampone, chiamata buffer di scrittura.

Il buffer di scrittura memorizza i dati in attesa che essi vengono scritti in memoria. Una volta completata la scrittura del dato in memoria il relativo spazio nel buffer viene liberato.

Il limite che ha questa soluzione è che il processore può generare richieste di scritture ad un tasso maggiore di quello con cui la memoria porta a termine le scritture quindi potremo trovarci in una situazione in cui il processore genera richieste di scrittura e il buffer è pieno, in questo caso il processore viene messo in stallo finché non si libera spazio.

Write back: scrivere solo quando necessario.

I blocchi sono effettivamente scritti in memoria principale solo quando è necessario rimpiazzarli nella cache.

Per ricordarmi che i dati scritti in memoria e quelli in cache sono inconsistenti, viene utilizzato un ulteriore bit chiamato dirty, quando i due valori sono inconsistenti alzo il bit a dirty e quando vado a sovrascrivere tutto quello che è dirty va scritto in memoria.

Svantaggio: maggiore complessità implementativa rispetto al write-through.

Vantaggio: minore numero di accessi alla memoria principale e quindi maggiore efficienza. Infatti, se una variabile viene modificata più volte mentre è in cache si accede alla memoria una sola volta (quando esce dalla cache)

**CACHE MISS**

**I tipi di cache miss?**

**Miss obbligate** sono le miss della cache causate dal primo accesso a un blocco che non era mai stato caricato in cache.

**Miss di capacità** sono le miss che si verificano perché la cache anche se fosse completamente associativa, non potrebbe contenere tutti i blocchi necessari a soddisfare le richieste.

**Miss di conflitto** indicano le miss che avvengono in una chace set associative o a mappatura diretta quando più blocchi competono per la stessa linea.

La miss di conflitto possono essere eliminate utilizzando una cache completamente associativa della stessa dimensione.

**Aggiunte Johnny**

**Località temporale:** quando si fa riferimento a un elemento di memoria, c’è la tendenza a far riferimento allo stesso elemento entro breve (ad esempio: riutilizzo di istruzioni e dati contenuti nei cicli). (Vale sia per i dati che per le istruzioni)

**Località spaziale:** quando si fa riferimento a un elemento di memoria, c’è la tendenza a far riferimento entro breve tempo ad altri elementi che hanno indirizzo vicino a quello dell’elemento ’elemento corrente.

Si ha **hit** (successo nell’accesso) quando i dati richiesti dal livello superiore (ad es. il processore) compaiono in qualche blocco nel livello inferiore.

Si ha una **miss** (fallimento nell’accesso) se il dato non è presente nel livello immediatamente inferiore ed occorre accedere al livello più distante.

**Tempo di hit:** tempo necessario a prelevare il dato dal livello più vicino, comprendendo il tempo necessario a determinare se l’accesso è un hit o una miss.

**Tempo di miss:** tempo necessario a sostituire un dato nel livello più vicino con il blocco corrispondente nel livello inferiore, più il tempo necessario per consegnare il dato al livello richiedente (ad es. il processore).

**Gerarchia delle memorie:** una struttura che utilizza più livelli di memoria, all’aumentare della distanza dal processore, crescono sia la dimensione sia il tempo di accesso alla memoria, quindi Una gerarchia di memoria è composta da più livelli, ma i dati vengono di volta in volta copiati solo tra due livelli adiacenti.

**Ricerca di un blocco in una cache set associativa**, il contenuto del campo indice serve a selezionare la linea che può contenere l’elemento cercato e viene esaminato il campo tag di tutti i blocchi appartenenti alla linea selezionata. (Il contenuto del campo tag viene esaminato in parallelo).

**Ricerca di un blocco in cache full associativa,** dato che il dato può essere scritto in qualsiasi blocco di una linea, la ricerca deve essere effettuata analizzando i campi tag di tutti i blocchi della linea.