

ALMA MATER STUDIORUM  
UNIVERSITÀ DEGLI STUDI DI BOLOGNA

---

Facoltà di Ingegneria

Corso di Laurea in INGEGNERIA INFORMATICA

Progetto di CALCOLATORI ELETTRONICI M

# **Progetto di una memoria cache per il processore DLX**

Componenti Gruppo:

Andrea Grandi

Filippo Malaguti

Massimiliano Mattetti

Gabriele Morlini

Thomas Ricci

---

Anno Accademico 2009/2010



# Indice

<b>Introduzione</b>	<b>5</b>
1 Obiettivi del progetto . . . . .	6
<b>1 Caratteristiche della memoria cache</b>	<b>7</b>
1 Politica di rimpiazzamento . . . . .	8
2 Struttura e interfacce . . . . .	9



# Introduzione

La necessità di introdurre cache per un processore deriva dal noto problema del collo di bottiglia rappresentato dall'accesso a dispositivi di memoria. Un processore durante il suo funzionamento tende ad accedere in scrittura o a reperire dati in lettura provenienti dalla memoria a valle e tale operazione richiede tipicamente diversi cicli di clock che costringono il processore (più veloce della memoria) ad attendere il dato. Ciò comporta l'introduzione di cicli di wait che ovviamente causano un peggioramento delle performance del processore, il quale attende che la memoria gli presenti il dato richiesto segnalato dal segnale di ready.

Per superare tale problema si utilizzano pertanto delle memorie cache, vicine al processore, di piccole dimensioni e molto veloci (tanto che possono avere tempi d'accesso simili a quelli dei registri interni al processore) da cui vengono reperiti i dati necessari all'esecuzione, consentendo in caso di HIT, ovvero nel caso in cui il dato si trovi in cache, di recuperarlo quasi senza ritardo.

Le cache si posizionano nella gerarchia delle memorie (insieme ai registri) tra i livelli più prossimi al processore e ciò comporta da un lato la rapidità nell'accesso e dall'altro le dimensioni limitate che fanno sì che una cache contenga un subset delle linee di memoria del dispositivo a valle (memoria o un livello superiore di cache se presente).

Pertanto quando si accede a una cache possono capitare due casi:

1. il dato si trova nella cache (HIT);
2. il dato non è presente e deve essere recuperato da un dispositivo a valle (MISS).

Ovviamente in caso di MISS si deve pagare un costo temporale per il reperimento del dato assente, detto miss penalty, dato dalla somma del tempo

d'accesso al dispositivo a valle e dal tempo di trasferimento della linea col dato cercato.

Ciononostante, è dimostrato che l'hit rate e quindi l'efficienza delle cache è tipicamente molto alta (oltre il 95%) grazie alla validità del Principio di Località spaziale e temporale, per il quale un programma in esecuzione tende ad eseguire temporalmente istruzioni eseguite di recente e ad accedere a dati acceduti di recente. Quindi sulla base di tali considerazioni, l'uso di cache contenenti le linee di memoria più recentemente accedute (working set) consente di migliorare notevolmente il tempo di reperimento dei dati necessari all'esecuzione, evitando quindi i ritardi che si avrebbero per ogni accesso diretto in memoria.

Abbiamo scelto questo progetto per approfondire le tematiche e le problematiche legate alla progettazione di un componente cache da affiancare al processore DLX visto a lezione.

## 1 Obiettivi del progetto

L'attività di progetto svolta si prefigge i seguenti obiettivi

1. **Realizzazione memoria cache:** progetto di un component VHDL che realizza il funzionamento di una memoria cache generica.
2. **Testbench del component:** progetto di una suite di test per il component.
3. **Integrazione con DLX:** modifica del progetto DLX per consentire l'integrazione del component realizzato con il processore

# Capitolo 1

## Caratteristiche della memoria cache

Si è scelto di progettare una cache di tipo set-associative, la cui schematizzazione è mostrata in Fig. 1.1. Questa tipologia di cache rappresenta un buon compromesso tra flessibilità e costo in termini di silicio.

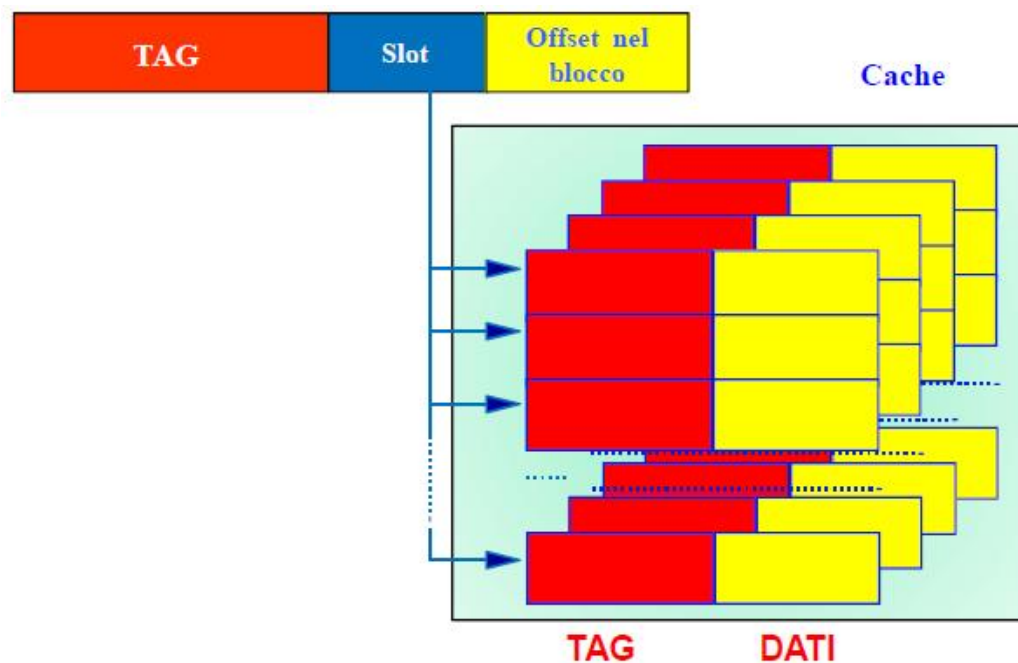


Figura 1.1: Schematizzazione di una cache set-associative

L'indirizzo di partenza del blocco è diviso in TAG (parte alta), INDEX e

OFFSET (parte bassa). Il TAG consente di identificare univocamente una linea all'interno di un sottoinsieme di linee, detto SET. L'INDEX individua immediatamente il SET all'interno del quale è possibile recuperare la linea corrente tramite il confronto del TAG. In questo modo si limita il numero di confronti tra TAG accettando il fatto che ogni linea possa appartenere ad un singolo set. La parte meno significativa dell'indirizzo rappresenta l'OFFSET che consente di individuare il dato all'interno di una linea.

Per garantire maggiore flessibilità si è scelto di parametrizzare alcune delle caratteristiche statiche della cache, quali ad esempio:

- la dimensione dei blocchi
- il numero di vie
- il numero di linee

## 1 Politica di rimpiazzamento

Nel caso in cui si debba caricare una nuova linea e tutte le vie siano occupate è necessario determinare quale linea rimpiazzare. Un buon algoritmo di rimpiazzamento dovrebbe cercare di individuare la linea vittima che meno probabilmente verrà riutilizzata in seguito.

Il criterio scelto per effettuare il rimpiazzamento è basato su contatori, che implementa una politica LRU (Least Recently Used). Tale politica è tipicamente implementata poichè statisticamente si verifica principio di località. È quindi presente un contatore per ogni via di ogni set tramite il quale si tiene traccia di quanto recentemente si è acceduti a ciascuna linea: un valore basso del contatore indica un accesso recente mentre un valore alto indica un accesso *vetusto*. Evidentemente la linea candidata al rimpiazzamento risulta essere quella alla quale è associato il contatore di valore più elevato.

Nel caso di HIT su una linea, sono incrementati i valori di contatori più basso rispetto al valore di quello della linea HIT mentre quest'ultimo viene resettato. Nel caso di MISS si procede con un rimpiazzamento e poi si agisce come nel caso di HIT sulla nuova linea. Infine, in caso di invalidazione di una linea, si porta al valore massimo il contatore della linea invalidata e si decrementano di 1 tutti i contatori con valore più elevato di quello della linea invalidata.



## 2 Struttura e interfacce

La memoria cache si interfaccia con i dispositivi esterni attraverso 3 tipi di interfacce, come mostrato in Fig. 1.2.

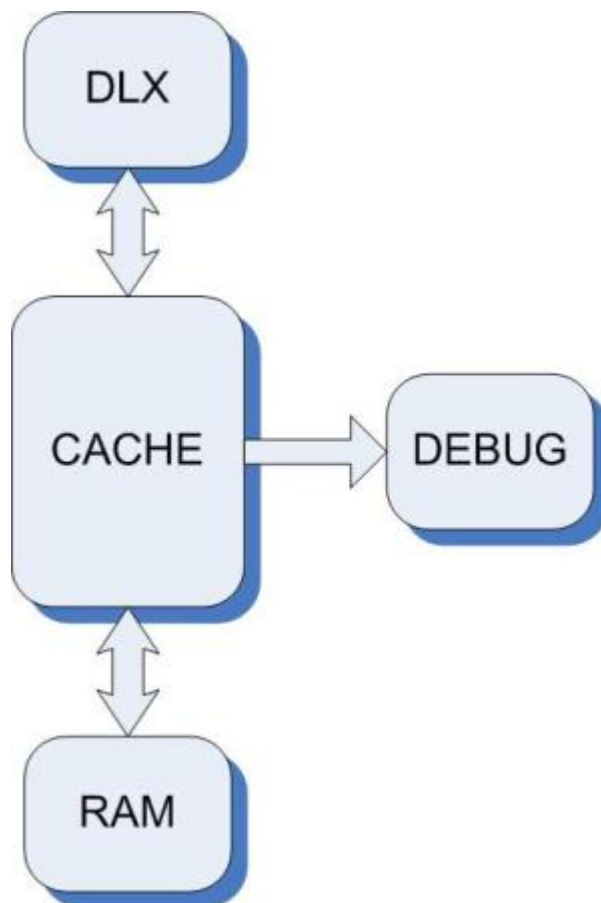


Figura 1.2: Interfacce della memoria cache

L'interfaccia verso il microprocessore, mostrata in Fig. 1.3, consente a quest'ultimo di accedere ai dati memorizzati all'interno della cache.

In particolare sono presenti i seguenti segnali:

- **Address[31-2]**: indirizzi a 32 bit emessi dal microprocessore
- **Data[32-0]**: bus dati con parallelismo 32bit
- **Write**: segnale per il comando di scrittura in cache
- **Read**: segnale per il comando di lettura da cache
- **Ready**: segnale che indica il termine dell'operazione di lettura/scrittura corrente

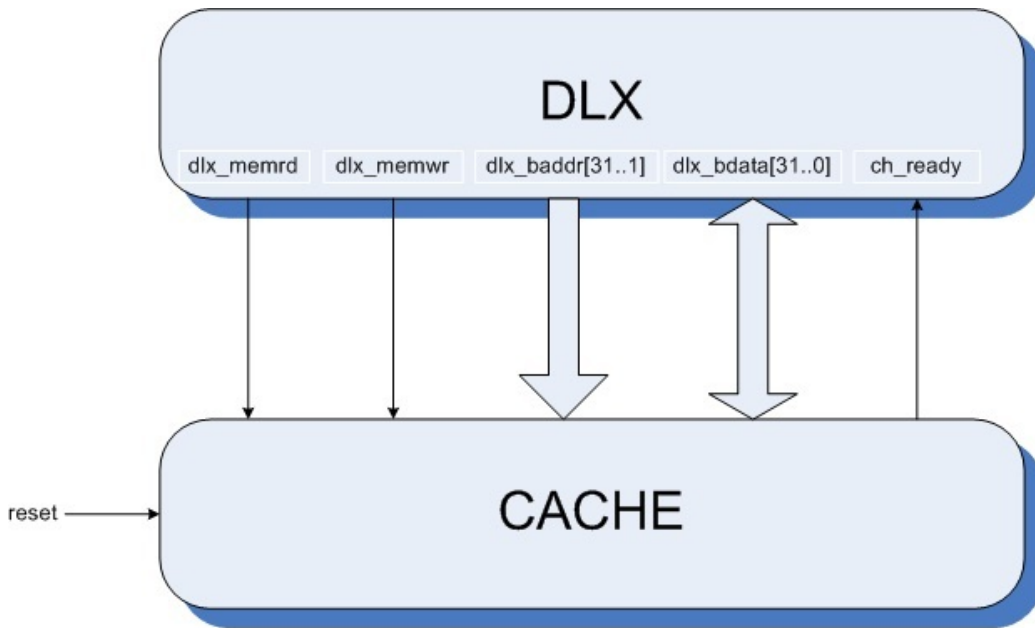


Figura 1.3: Interfaccia della memoria cache verso il processore DLX

Per quanto riguarda gli scambi di dati tra processore e memoria cache, si ipotizza che siano sempre lette e scritte parole di lunghezza fissa a 32 bit. L'interfaccia verso la RAM, mostrata in Fig. 1.4, consente alla cache di recuperare i blocchi dal livello sottostante.

In particolare sono presenti i seguenti segnali:

- **Address[31-2]**: indirizzi a 32 bit emessi dalla cache
- **Data[32-0]**: bus dati con parallelismo 32bit
- **Write**: segnale per il comando di scrittura in RAM
- **Read**: segnale per il comando di lettura dalla RAM
- **Ready**: segnale che indica il termine dell'operazione di lettura/scrittura corrente

Si noti che la cache non è a conoscenza del componente posto al livello superiore. Vista la simmetria delle due interfacce è quindi possibile sostituire la RAM con un ulteriore livello di cache, inserendo quindi più livelli di cache all'interno del processore.

È presente infine una terza interfaccia verso l'esterno, utilizzata per monitorare lo stato interno della cache e poter quindi eseguire il debug. I segnali disponibili verranno definiti nel seguito.

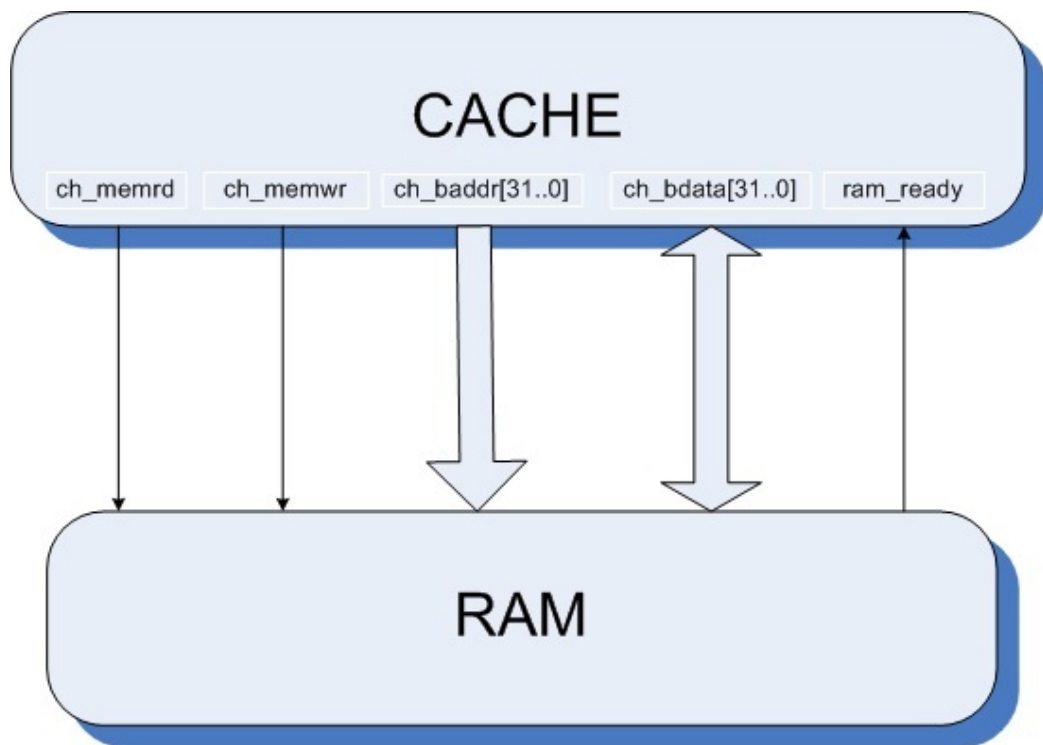


Figura 1.4: Interfaccia della memoria cache verso la RAM

Si ipotizza che la memoria cache progettata non sia impiegata in sistemi multimaster. Per tale motivo non si considereranno problematiche inerenti alla presenza di un controllore di memoria come ad esempio l'invalidazione delle linee e lo snooping dei dati presenti in cache..