Lezione 2 – Memoria *cache* e politica *Tag Associative*

Architettura degli elaboratori

Modulo 5 - Principali linee di evoluzione architetturale

Unità didattica 1 - Memoria *cache* e gerarchia di memoria

Nello Scarabottolo

Università degli Studi di Milano - Ssri - CDL ONLINE

Struttura generale

Abbiamo parlato di *cache* come contenitore di una copia dei blocchi di celle di memoria di lavoro che circondano la cella richiesta dalla CPU in un dato accesso a memoria.

Il progetto della *cache* richiede la seguente struttura generale:

- memoria di lavoro (**MdL**) divisa in blocchi di dimensione predefinita (es. 16 celle o parole);
- memoria *cache* (**MC**) divisa a sua volta in blocchi di dimensione uguale a quelli di MdL;
- ogni volta che non è presente una cella in *cache*, si trasporta in MC l'intero blocco di MdL che la contiene.

Problemi da risolvere

Mapping dei blocchi da MdL a MC:

- ogni blocco di MdL in quale o quali blocchi di MC può essere copiato.

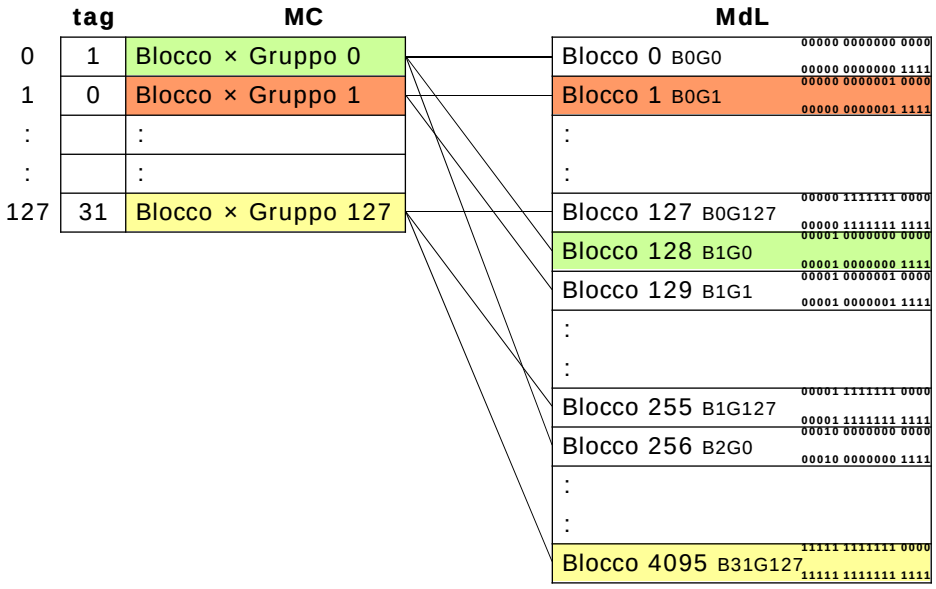
Ricerca della parola richiesta dalla CPU:

- ***HIT***: la parola richiesta si trova in *cache*;
- ***MISS***: la parola richiesta NON si trova in *cache* e deve essere prelevata da MdL.

Sostituzione del blocco di MC in caso di MISS:

- quali azioni si devono intraprendere per ottimizzare l'uso della *cache*.

Politica *Tag Associative*

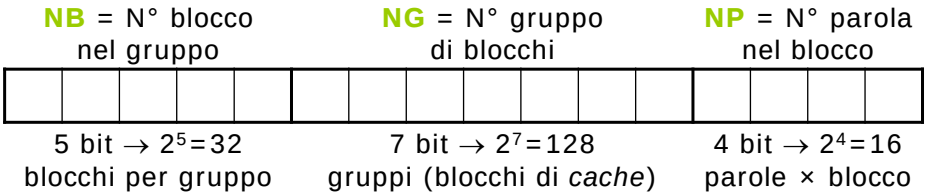


Mapping nella politica *Tag Associative*

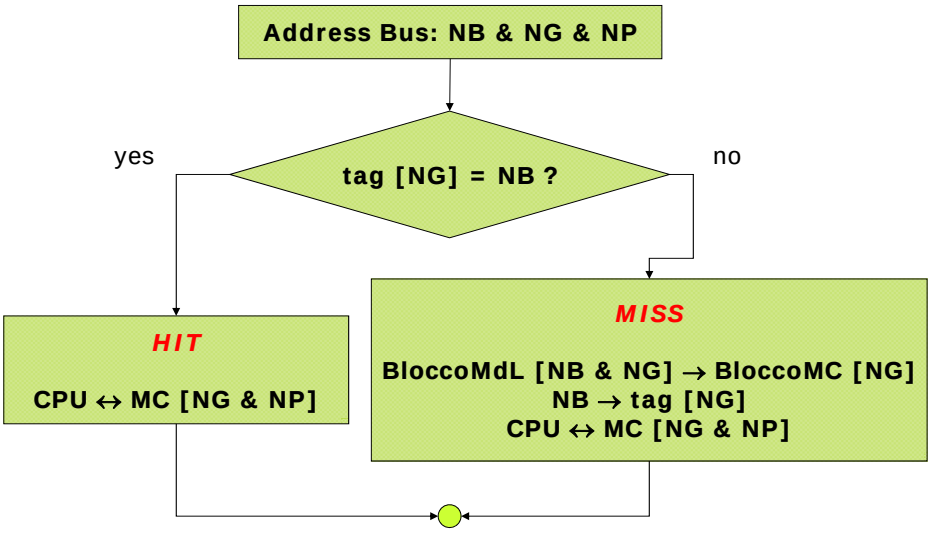
Si definisce a priori una corrispondenza univoca fra:

- gruppo di blocchi in memoria di lavoro (MdL);
- blocco di possibile destinazione in *cache* (MC).

Nell'esempio l'indirizzo generato dalla CPU (16 bit) ha la seguente struttura:



Ricerca parola in *cache Tag Associative*



Caratteristiche *Tag Associative*

☺ **Politica semplice:**

- il blocco richiesto dalla CPU può trovarsi solo in un blocco di *cache:*
  - ⇒ la scoperta di HIT/MISS è rapida e priva di problemi;
- in caso di MISS, il blocco richiesto può essere ricopiato in un'unica posizione.

☹ **Politica non ottimizzata:**

- ogni blocco di MC ottimizza **localmente** l'accessibilità ai blocchi di MdL cui è assegnato;
- lo sfruttamento dei blocchi di MC non è uniforme.

In sintesi...

Abbiamo individuato un metodo semplice di allocazione dei blocchi di MdL in MC.

La politica *Tag Associative* di allocazione dei blocchi di MdL in *cache* privilegia la semplicità realizzativa a scapito dello sfruttamento ottimale della MC.

Forse si può fare di meglio...

Chiusura

Fine della lezione