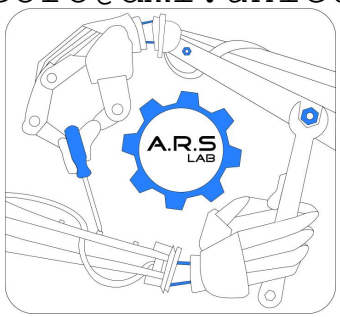
La Cache Memory

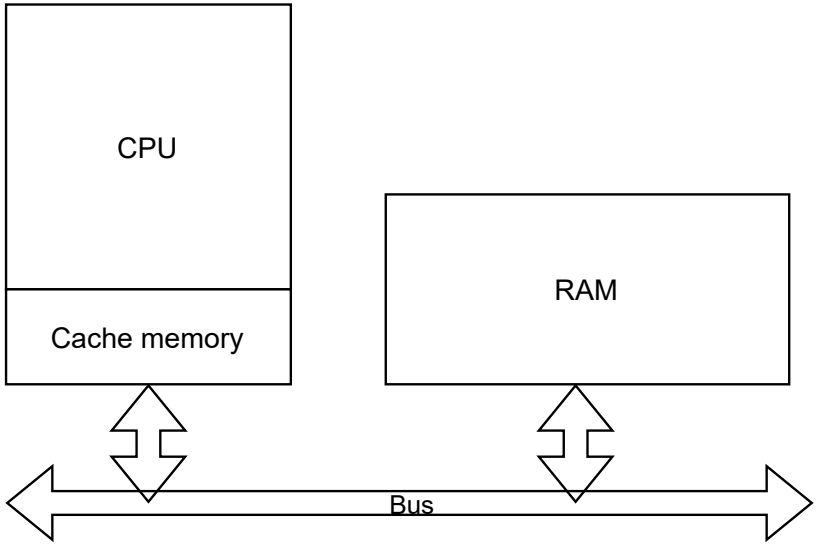
Corrado Santoro

Dipartimento di Matematica e Informatica
santoro@dmi.unict.it



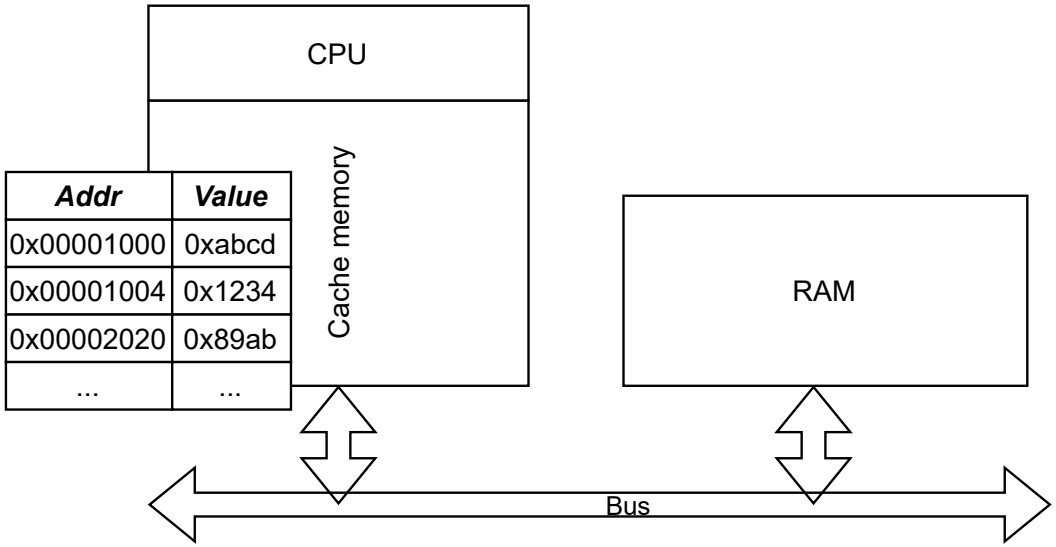
Corso di Architettura degli Elaboratori

Processore, Memoria e Cache



Cache Memory

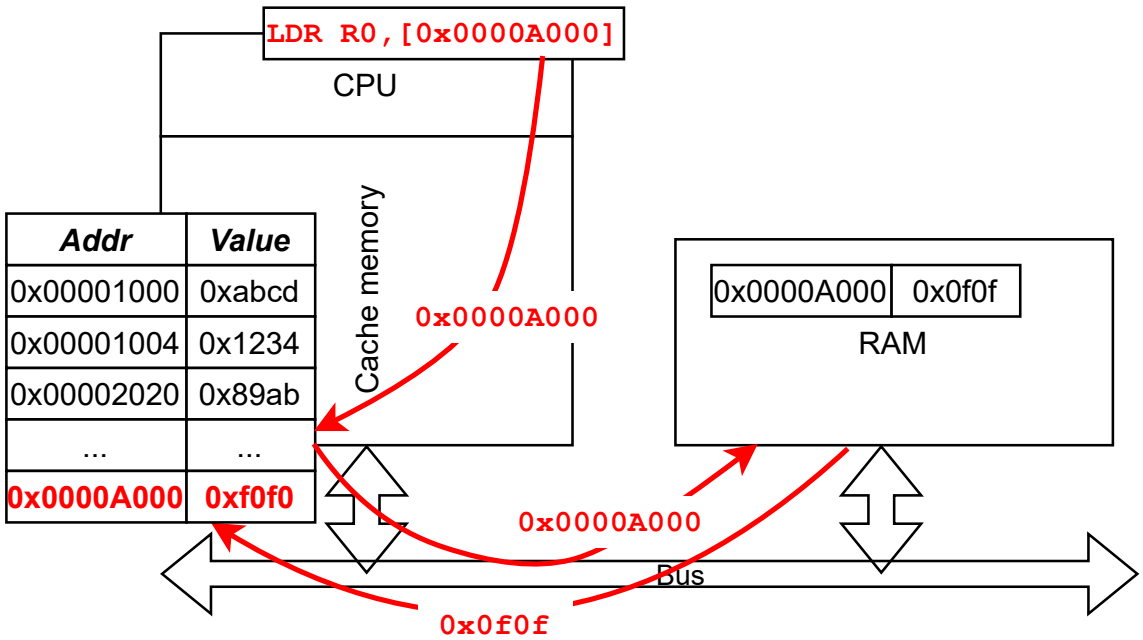
- La **Cache** (memoria tampone) è un dispositivo di memoria posto tra il processore e la memoria centrale vera e propria
- E' un dispositivo **più veloce** della memoria centrale ma con **meno capacità**
- Permette di velocizzare le operazioni di Load/Store per i dati più frequentemente utilizzati



Funzionamento della Cache

- La **Cache** memorizza coppie (*Address*, *Value*), dove *Address* si riferisce ad un indirizzo della memoria centrale e *Value* la relativa word memorizzata

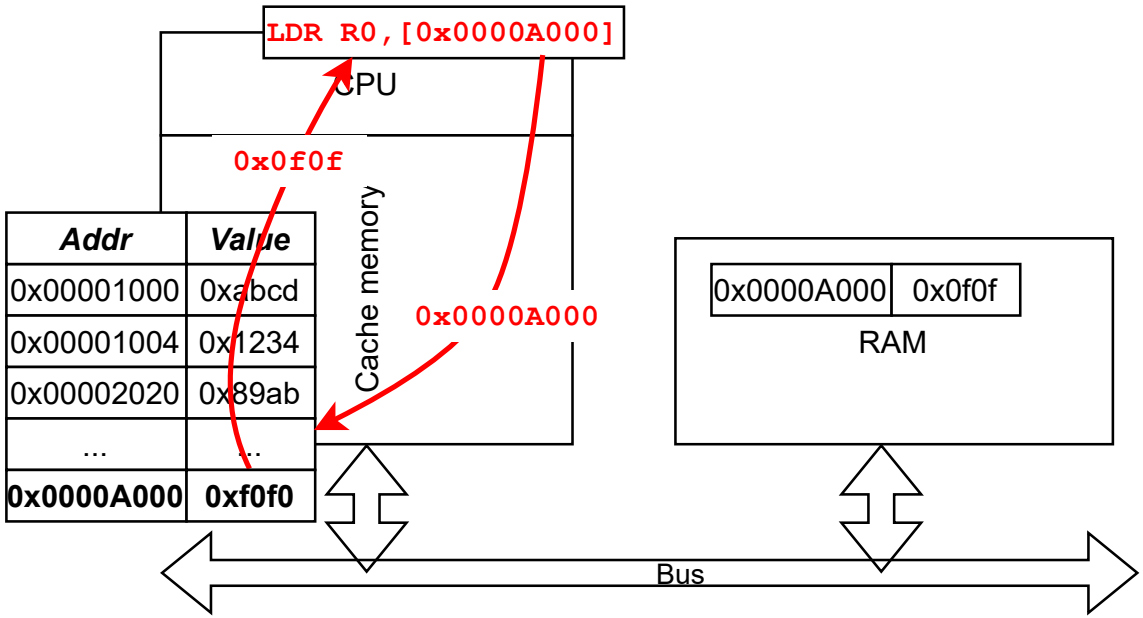
Cache Miss



Cache Miss

- Un'operazione di LDR provoca la ricerca dell'indirizzo nella cache
- Se l'indirizzo *non viene reperito* si verifica un evento di **cache miss**
- Il cache controller recupera il dato dalla memoria e lo inserisce in cache

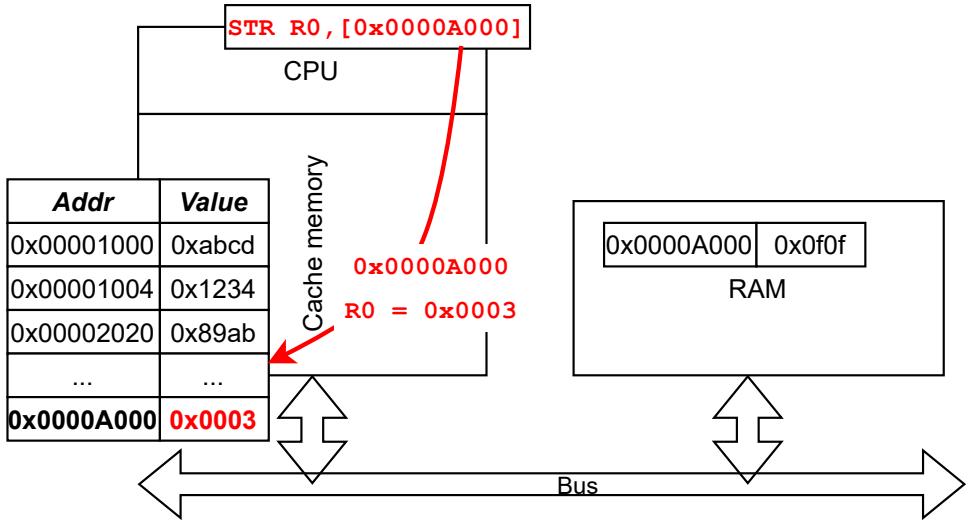
Cache Hit



Cache Hit

- Un'operazione di LDR provoca la ricerca dell'indirizzo nella cache
- Se l'indirizzo *viene reperito* si verifica un evento di **cache hit**
- Il cache controller recupera il dato dalla cache e lo restituisce alla CPU senza coinvolgere la memoria centrale e il BUS

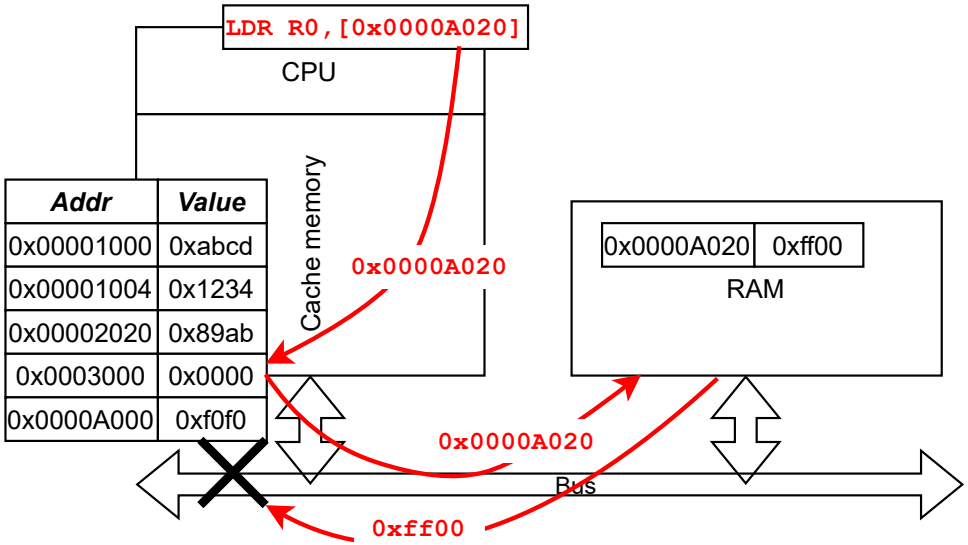
Cache Write



Cache Write

- Un'operazione di STR provoca l'aggiornamento del valore associato all'indirizzo nella cache
- Il valore in memoria centrale sarà **differente**, ma poichè le LDR fanno riferimento alla cache otterranno sempre il dato aggiornato
- La locazione di memoria aggiornata viene marcata, nella cache, come *dirty* per segnalare che essa contiene un valore differente da quello in memoria centrale

Cache Full e Write-Back



Cache Full e Write-Back

- Poichè la cache è più piccola della memoria centrale, può accadere che, a seguito di una LDR con *cache miss*, non ci sia **più spazio** nella cache
- Occorre scegliere un *entry* della cache da liberare
- Se l'entry è marcata *dirty*, occorre aggiornare il valore nella RAM (*write back*)

Principi di Funzionamento della Cache

Principio di Località dei riferimenti temporale

Data una locazione di memoria X acceduta al tempo t, la probabilità che essa venga acceduta nuovamente nell'immediato futuro è molto elevata

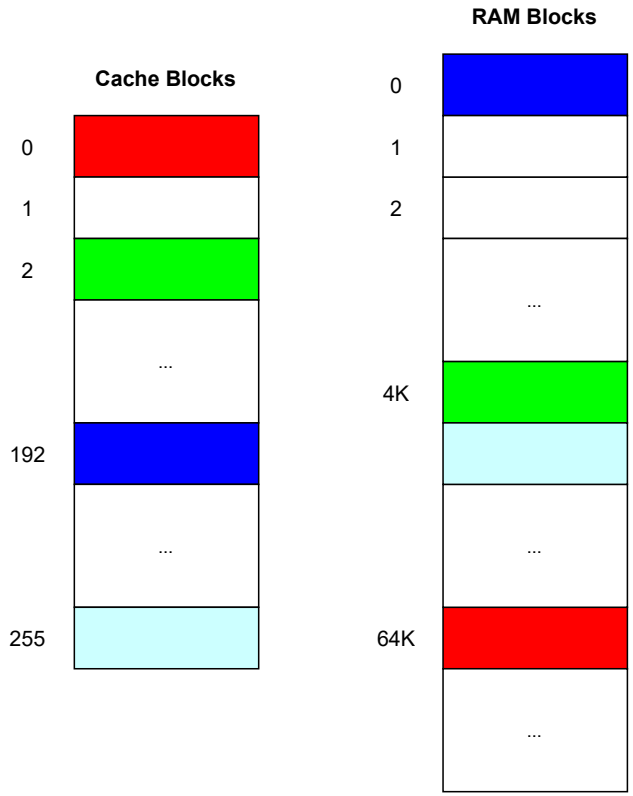
Principio di Località dei riferimenti spaziale

Data una locazione di memoria X acceduta al tempo t, la probabilità che vengano accedute *locazioni contigue* a X nell'immediato futuro è molto elevata

Organizzazione della cache

- In virtù del principio di località *spaziale*, la cache non è organizzata per locazioni ma per **blocchi** o **pagine** di dimensioni costanti e potenze del 2 (in genere 64 byte)
- L'accesso ad una locazione di memoria X (tramite LDR o STR) implica quindi il coinvolgimento dell'intero blocco che contiene X
- La cache include una **funzione di mapping** che lega il numero di un blocco della cache al numero del blocco nella RAM (e viceversa):

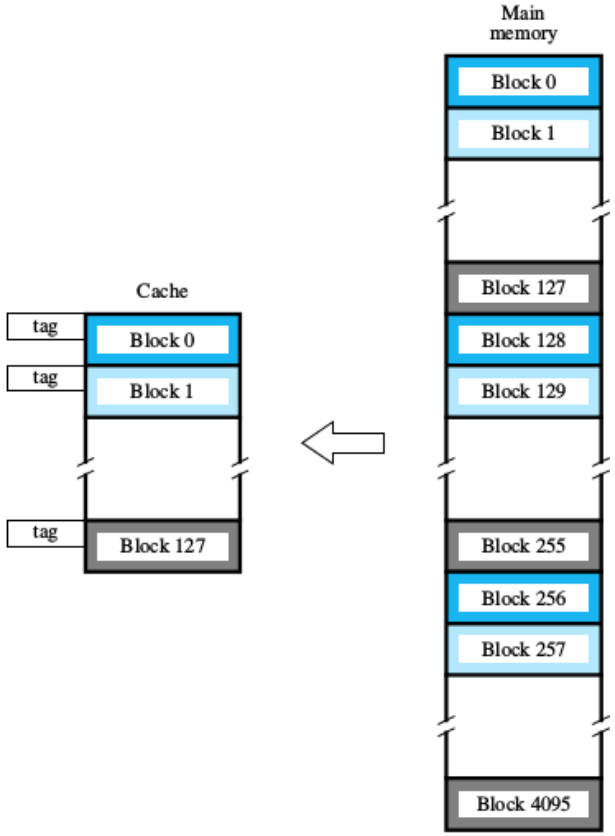
$$RAM\_Block \leftarrow INT(\frac{X}{64})$$
$$Cache\_Block \leftarrow map(RAM\_Block)$$
$$Cache\_Address \leftarrow Cache\_Block \cdot 64 + (X \mod 64)$$



Direct Mapping

- Un blocco *j* della memoria centrale è mappato sul blocco *j* mod 128 della cache
- Ogni blocco della cache possiede un **tag** (etichetta) utilizzato per ricostruire l'indirizzo del blocco in RAM
- Supponendo un indirizzo a 16 bit:

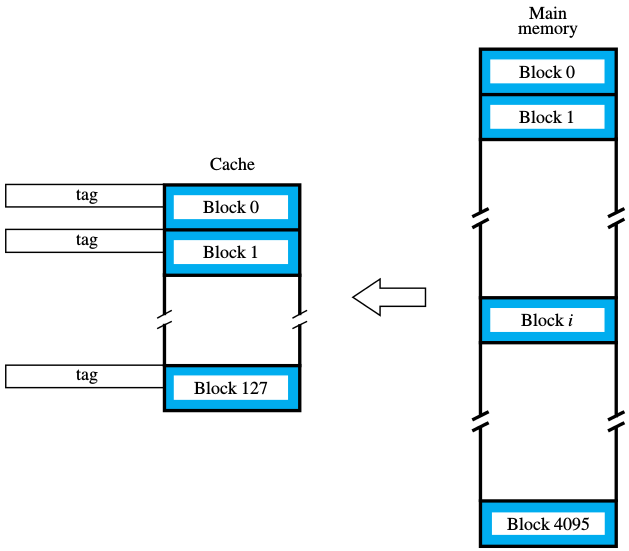
Tag	Block	Offset
3	7	6



Associative Mapping

- Un blocco *j* della memoria centrale è mappato su un blocco qualunque *k* della cache
- Ogni blocco della cache possiede un **tag** (etichetta) utilizzato per ricostruire l'indirizzo del blocco in RAM
- Supponendo un indirizzo a 16 bit:

Tag	Offset
10	6

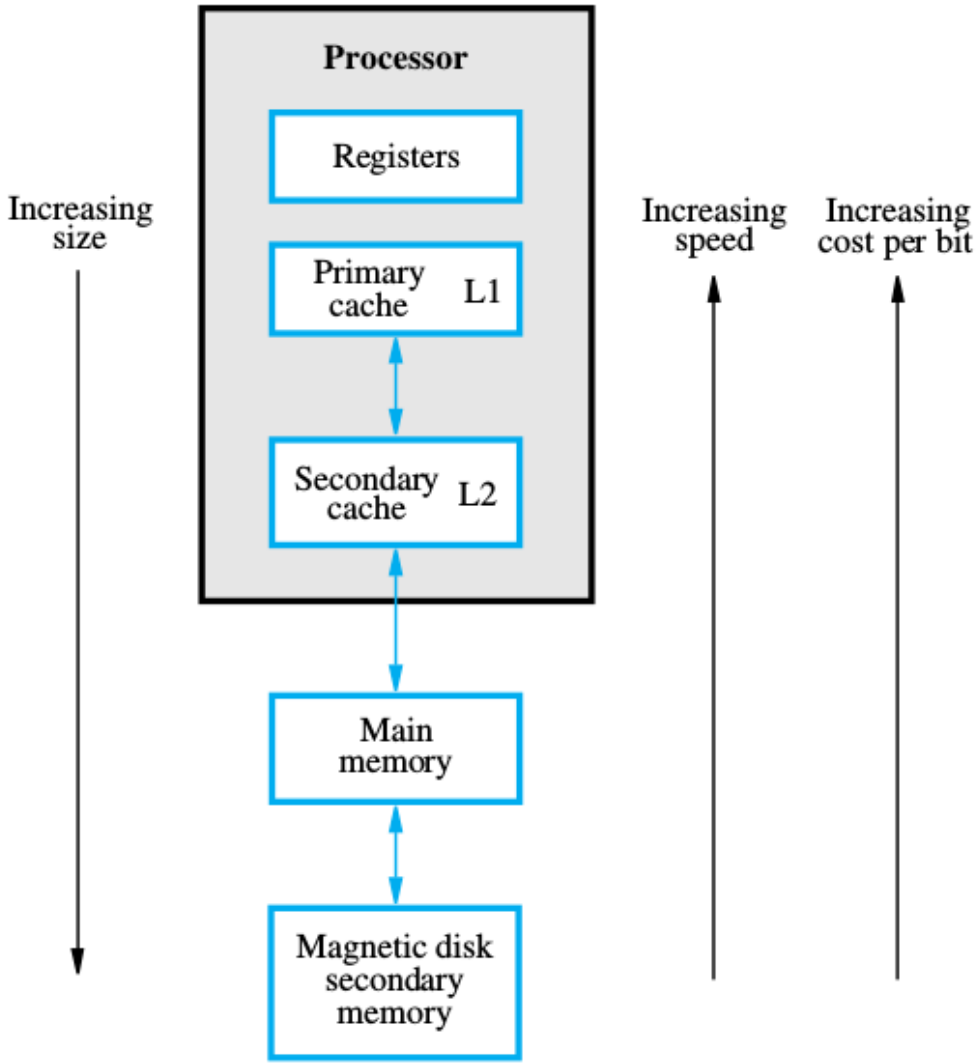


Algoritmi di Rimpiazzamento

LRU: Least Recently Used

- Ogni blocco k della cache possiede un contatore C_k che rappresenta da quanto tempo il blocco non è stato riferito
- Quando si accede al blocco k (cache hit), tutti i contatori degli altri blocchi che sono inferiori a C_k sono incrementati di 1: $\forall j \neq k : C_j < C_k, C_j \leftarrow C_j + 1$
- Quando accade un cache miss e non ci sono più blocchi liberi, si sceglie il blocco k con C_k massimo, se dirty lo si trasferisce in memoria (write-back), e si ospita il nuovo blocco in arrivo dalla memoria nel blocco k; C_k viene inizializzato a 0

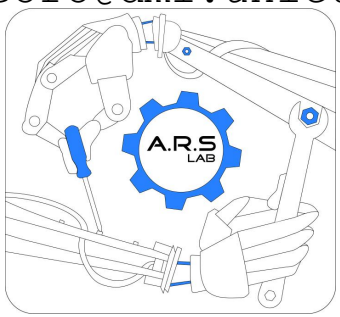
Gerarchia delle Memorie



La Cache Memory

Corrado Santoro

Dipartimento di Matematica e Informatica
santoro@dmi.unict.it



Corso di Architettura degli Elaboratori