



Le memorie Cache a mappatura diretta

Prof. Alberto Borghese
Dipartimento di Scienze dell'Informazione
alberto.borghese@unimi.it

Università degli Studi di Milano

Riferimento Patterson: 5.2, 5.3



Sommario

Circuito di lettura / scrittura di una cache a mappatura diretta

Esempi

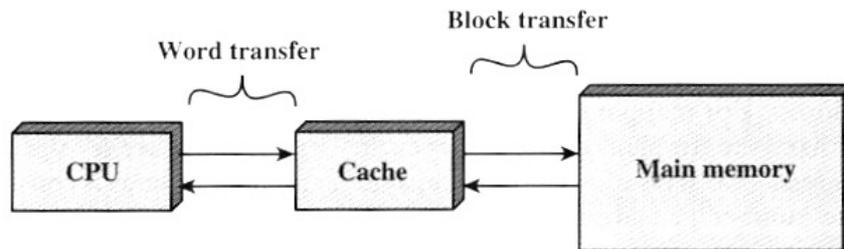


Principio di funzionamento di una cache



Scopo: fornire alla CPU una velocità di trasferimento pari a quella della memoria più veloce con una capacità pari a quella della memoria più grande.

Una cache “disaccoppia” i dati utilizzati dal processore da quelli memorizzati nella Memoria Principale.



Word transfer: Data transfer or Instruction transfer. In MIPS = 1 parola.

La cache contiene una copia di parte del contenuto della memoria principale. Di che cosa?



Sottosistema di memoria

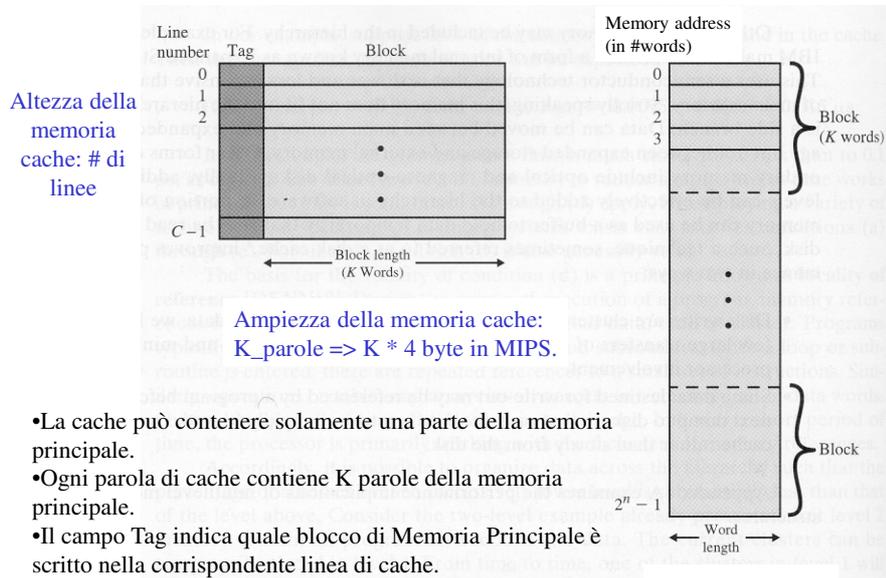


Porta nella cache primaria i dati richiesti mentre il binomio processore-memoria sta lavorando.

- 1) Controlla se una parola è in cache (Hit).
- 2) Porta una parola (e quelle vicine) in cache, prelevandole dal livello inferiore (Miss):

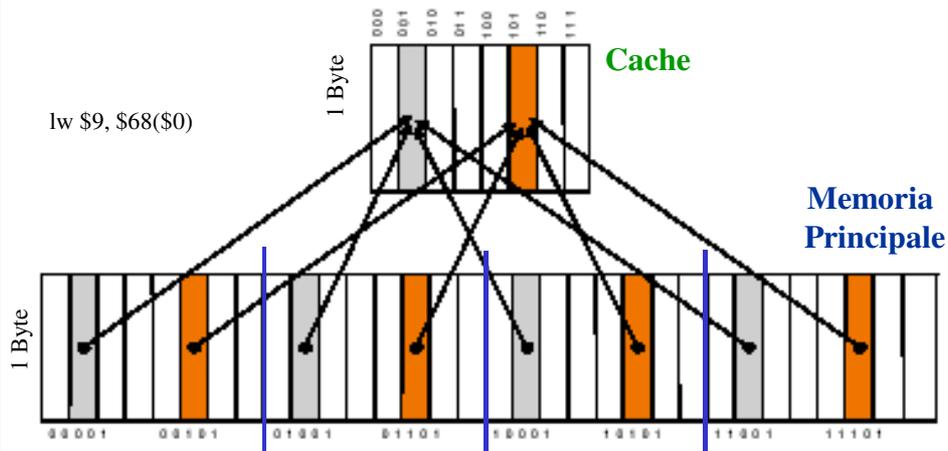


Mappatura diretta di una cache



Corrispondenza diretta (direct mapped)

Ad ogni indirizzo di Memoria Principale corrisponde un indirizzo di cache.



Indirizzi diversi di Memoria Principale corrispondono allo stesso indirizzo di cache.
Quali indirizzi della memoria principale si considerano?

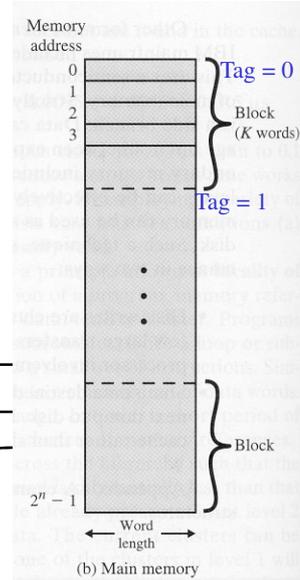
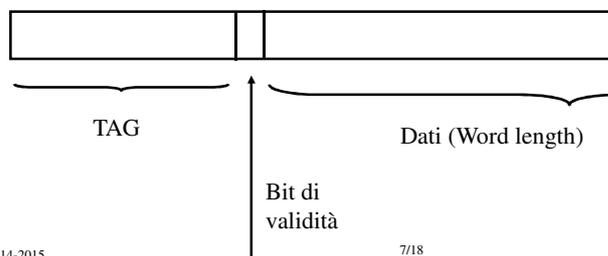


Come leggere / scrivere su cache

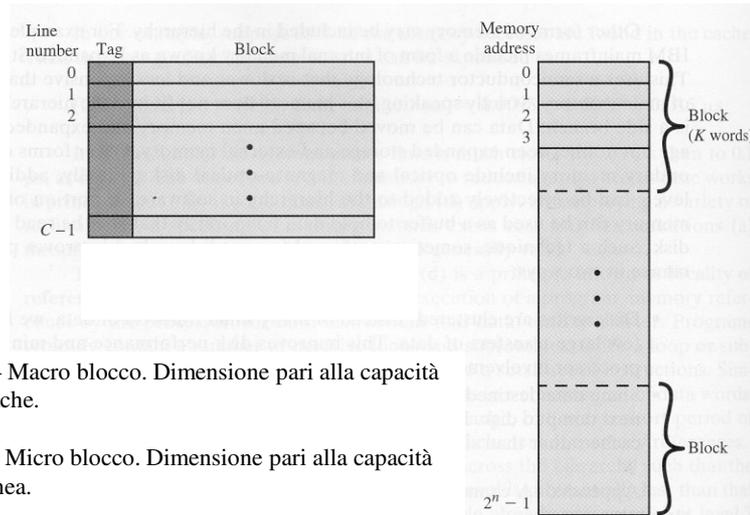


- Individuare la linea della cache dalla quale leggere / scrivere (operazione analoga all'indirizzamento del register file).
- Confrontare il campo tag con il blocco di Memoria Principale in cui risiede il dato.
- Controllare il bit di validità.
- Leggere (scrivere) il dato.

Per blocchi più ampi di una parola, occorre individuare una parola tra le k presenti nella linea di cache.



Osservazioni



Block – Macro blocco. Dimensione pari alla capacità della cache.

Linea – Micro blocco. Dimensione pari alla capacità della linea.

La linea contiene Byte in posizione contigua in memoria principale.



Esempio di parsing dell'indirizzo



0000 0000 0000 00(00) \Rightarrow 0000 0000 0 111 11 (11) 128 indirizzi diversi (32 parole di 4 byte)

La cache con linee di 4 parole (ampiezza) ed altezza di 8 linee:
Il blocco di dati contenuto in ogni linea di cache è di dimensioni: $n = 4 * 4 \text{ byte} = 16 \text{ byte}$.
La capacità della cache è di $8 * 16 \text{ byte} = 128 \text{ byte}$.

`lw $t0, 196($zero)`

0...0 00000001 100 01 (00)

$196 / [4 * 4 * 8] = 1$ (2° blocco di RAM, tag = 1) con resto $R_1 = 196 - 1 * 128 = 68$.
Il resto, R_1 , rappresenta l'offset in byte all'interno della cache.

$68 / [4 * 4] = 4$ (5ª linea della cache) con resto $R_2 = 68 - 4 * 16 = 4$.
Il resto, R_2 , rappresenta l'offset in byte all'interno della linea di cache.

$4 / 4 = 1$ (2ª parola della cache) con resto $R_3 = 4 - 1 * 4 = 0$.
Il resto, R_3 , rappresenta l'offset in byte all'interno della parola.



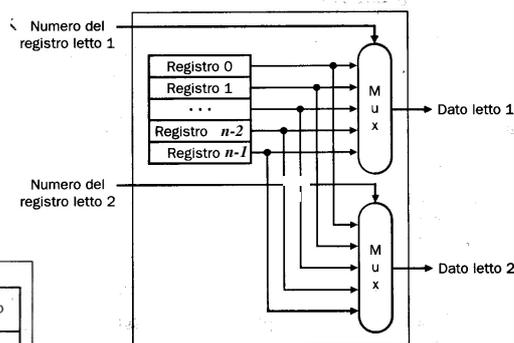
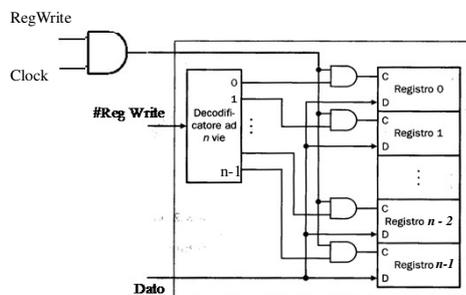
Register file



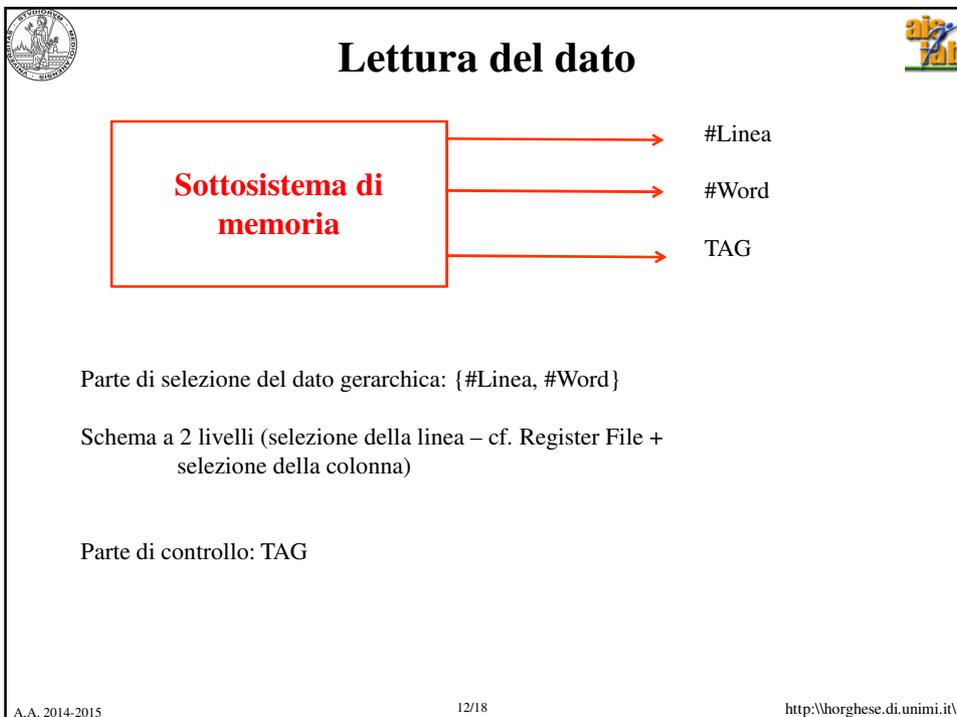
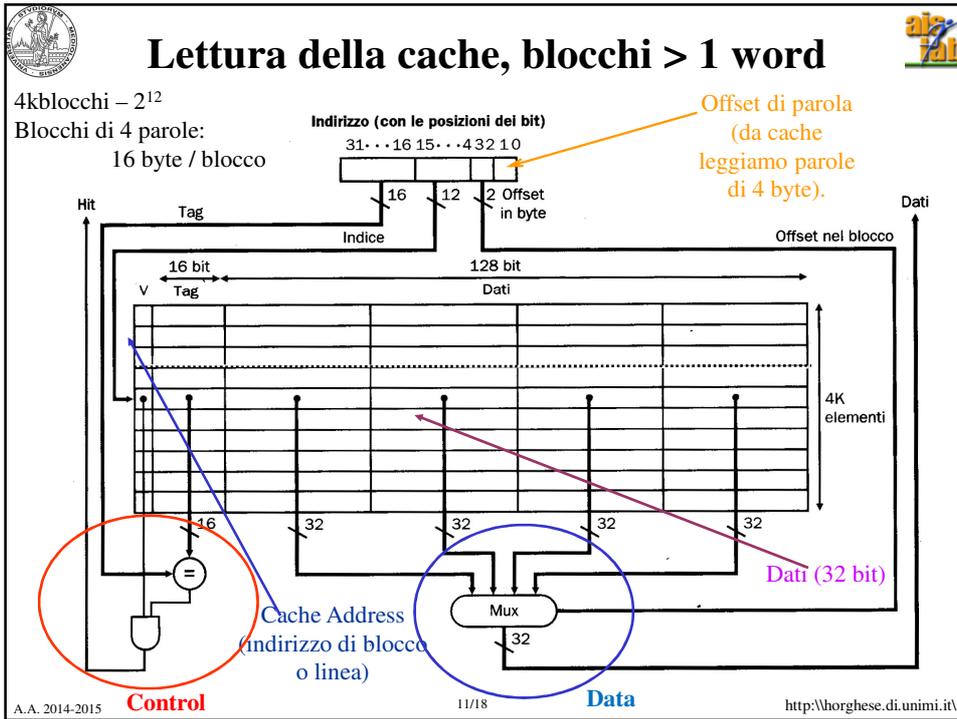
Il tempo di lettura dipende dal cammino critico dei Mux.

Il tempo di scrittura dipende dal cammino critico del Decoder.

Numero_registro = selettore.



Scrittura cache = scrittura nel register file





Scrittura del dato



Parte di selezione del dato gerarchica mediante decoder: {#Linea, #Word}

Parte di controllo: TAG.

Cosa succede se il TAG presente è diverso dal TAG dell'istruzione di *sw* ed il bit di validità è uguale a 1?



Sommario



Circuito di lettura / scrittura di una cache a mappatura diretta

Esempi



Esempio di operazioni su cache



All'inizio il bit di data_valid = False. Suppongo cache di 8 linee di 1 byte.
Non importa quello che è contenuto. Cambio il contenuto di \$t1.

			TAG	#L	Valid
• lb \$t0, 0(\$t1)	\$t1 = 22 (10 110)	MISS	10	6	0->1
• lb \$t0, 0(\$t1)	\$t1 = 26 (11 010)	MISS	10	4	0->1
• lb \$t0, 0(\$t1)	\$t1 = 22 (10 110)	HIT	10	6	1->1
• lb \$t0, 0(\$t1)	\$t1 = 26 (11 010)	MISS	11	4	1->1
• lb \$t0, 0(\$t1)	\$t1 = 16 (10 000)	MISS	10	0	0->1
• lb \$t0, 0(\$t1)	\$t1 = 10 (00 010)	MISS	00	2	0->1
• lb \$t0, 0(\$t1)	\$t1 = 7 (00 111)	MISS	00	7	0->1
• lb \$t0, 0(\$t1)	\$t1 = 16 (10 000)	HIT	10	0	1->1
• lb \$t0, 0(\$t1)	\$t1 = 18 (10 010)	MISS	10	2	1->1



Esempio di operazioni su cache



All'inizio il bit di data_valid = False. Suppongo cache di 8 linee di 2 byte.
Non importa quello che è contenuto. Cambio il contenuto di \$t1.

			TAG	#L	#W
• lb \$t0, 0(\$t1)	\$t1 = 22 (1 011 0)	MISS	1	3	0
• lb \$t0, 0(\$t1)	\$t1 = 26 (1 101 0)	MISS	1	5	0
• lb \$t0, 0(\$t1)	\$t1 = 22 (1 011 0)	HIT	1	3	0
• lb \$t0, 0(\$t1)	\$t1 = 27 (1 101 0)	HIT	1	5	1
• lb \$t0, 0(\$t1)	\$t1 = 16 (1 000 0)	MISS	1	0	0
• lb \$t0, 0(\$t1)	\$t1 = 2 (0 001 0)	MISS	0	1	0
• lb \$t0, 0(\$t1)	\$t1 = 7 (0 011 1)	MISS	0	3	1
• sb \$t0, 0(\$t1)	\$t1 = 8 (0 100 0)	n/a	0	4	0
• lb \$t0, 0(\$t1)	\$t1 = 16 (1 000 0)	HIT	1	0	0
• lb \$t0, 0(\$t1)	\$t1 = 18 (1 001 0)	MISS	1	1	0



Esercizi



Sia data una cache a corrispondenza diretta contenente 64Kbyte di dati e avente blocchi di 1 parola. Assumendo che gli indirizzi siano di 32 bit quale è il numero totale di bit richiesto per l'implementazione della cache?

Supponendo che il MIPS abbia una cache di 512byte, indicare cosa succede nei campi della cache quando vengono eseguite le seguenti istruzioni:

lw \$t1, 0x0000(\$t0) \$t0 = 1kbyte = 1,024 byte

lw \$t1, 0x0000(\$t0) \$t0 = 0

lw \$t1, 0x0202(\$t0) \$t0 = 1kbyte = 1,024 byte

lw \$t1, 0x0001(\$t0) \$t0 = 0

lw \$t1, 0x0201(\$t0) \$t0 = 1kbyte = 1,024 byte

Costruire la porta di scrittura della Cache di primo livello.



Sommario



Circuito di lettura / scrittura di una cache a mappatura diretta

Esempi