

Gerarchie di memoria Cache

Architettura degli elaboratori

Esercitazione

Esercizio 1a

- In una cache direct-mapped con 32 blocchi (da 1 word), quali indirizzi di parole di memoria sono mappati nel blocco 13_{10} ?

Esercizio 1a – soluzione

- Poichè la cache è composta da 32 blocchi, considerando la rappresentazione binaria dell'indirizzo di memoria, sono mappati nel blocco 13_{10} della cache, tutti gli indirizzi di memoria la cui rappresentazione binaria ha gli ultimi $\log_2(32) = 5$ bit uguali a 01101 (codifica binaria di 13_{10} su 5 bit)

Esercizio 1b

- L'indirizzo della parola di memoria 301_{10} fa parte degli indirizzi mappati nel blocco 13_{10} di una cache direct-mapped di 32 blocchi (da 1 word) e di una cache direct-mapped di 64 blocchi (di una word)?

Esercizio 1b – soluzione

- Poichè $301 = 256 + 0 + 0 + 32 + 0 + 8 + 4 + 0 + 1$

la rappresentazione binaria di 301_{10} è $(100101101)_2$

Tale indirizzo fa quindi parte degli indirizzi mappati nel blocco 13_{10} in una cache direct-mapped a 32 blocchi (avendo gli ultimi 5 bit uguali a 01101)

NON fa invece parte degli indirizzi mappati nel blocco 13_{10} in una cache direct-mapped a 64 blocchi (non avendo gli ultimi 6 bit uguali a 001101)

Esercizio 1c

- Qual è l'indirizzo della word contenuta nel blocco con indice IND e con etichetta TAG, se la cache è a x blocchi da 1 word?
- Ad esempio, se la cache ha 8 blocchi
 - quali sono indice e etichetta dell'indirizzo 19_{10} ?
 - a quale indirizzo corrisponde il blocco con TAG=11 e IND=011?

Esercizio 1c – soluzione

- Sappiamo che, se una cache a mappaggio diretto ha x blocchi (da 1 word), un blocco con indice IND e etichetta TAG, contiene i dati della locazione di memoria con indirizzo (decimale) $x * \text{TAG}_{10} + \text{IND}_{10}$ oppure, in binario, pari alla concatenazione di $(\text{TAG}_{10})_2 (\text{IND}_{10})_2$
- Se abbiamo 8 blocchi ($8=2^3$ quindi IND è di 3 bit)
 - l'indirizzo $19_{10} = (10011)_2$ sarà mappato nel blocco con indice 011_2 e avrà un'etichetta pari a 10_2 . Infatti, se $\text{IND} = 011_2 = 3_{10}$ e $\text{TAG}_{10} = 10_2 = 2_{10}$, indirizzo $(8 * 2 + 3)_{10} = 10011_2$
 - Il blocco con TAG=**11** e IND=**011**, corrisponderà all'indirizzo $\text{11011}_2 = 16+8+2+1 = 8*3+3$

Esercizio 2

- Consideriamo una cache direct-mapped a 8 blocchi da 1 word, inizialmente vuota
- Data la sequenza di accessi alle word con indirizzo 22, 26, 22, 26, 16, 3, 16, 18, 16
- Qual è la sequenza di hit/miss?
- Come cambia il contenuto della cache ad ogni miss?

- $8 = 2^3$ e blocchi da una word
 - 3 bit low order dell'indirizzo, corrispondono al numero del blocco (INDEX)
 - restanti bit alti dell'indirizzo, corrispondono al TAG (etichetta) del blocco

Esercizio 2 – soluzione (1/10)

22, 26, 22, 26, 16, 3, 16, 18, 16

- Riferimento a word con indirizzo 22 (10110) → blocco con indice 110 → miss

Index	V	Tag	Data
000	N		
001	N		
010	N		
011	N		
100	N		
101	N		
110	N		
111	N		

a. The initial state of the cache after power-on

miss blocco 110



Index	V	Tag	Data
000	N		
001	N		
010	N		
011	N		
100	N		
101	N		
110	Y	10 _{two}	Memory (10110 _{two})
111	N		

b. After handling a miss of address (10110_{two})

Esercizio 2 – soluzione (2/10)

22, 26, 22, 26, 16, 3, 16, 18, 16

- Riferimento a word con indirizzo 26 (11010) → blocco con indice 010 → miss

Index	V	Tag	Data
000	N		
001	N		
010	N		
011	N		
100	N		
101	N		
110	Y	10_{two}	Memory (10110_{two})
111	N		

b. After handling a miss of address (10110_{two})

miss blocco 010



Index	V	Tag	Data
000	N		
001	N		
010	Y	11_{two}	Memory (11010_{two})
011	N		
100	N		
101	N		
110	Y	10_{two}	Memory (10110_{two})
111	N		

c. After handling a miss of address (11010_{two})

Esercizio 2 – soluzione (3/10)

22, 26, 22, 26, 16, 3, 16, 18, 16

- Riferimento a word con indirizzo 22 (10110) → blocco con indice 110 → hit

Index	V	Tag	Data
000	N		
001	N		
010	Y	11_{two}	Memory (11010_{two})
011	N		
100	N		
101	N		
110	Y	10_{two}	Memory (10110_{two})
111	N		

c. After handling a miss of address (11010_{two})

Esercizio 2 – soluzione (4/10)

22, 26, 22, 26, 16, 3, 16, 18, 16

- Riferimento a word con indirizzo 26 (11010) → blocco con indice 010 → hit

Index	V	Tag	Data
000	N		
001	N		
010	Y	11_{two}	Memory (11010_{two})
011	N		
100	N		
101	N		
110	Y	10_{two}	Memory (10110_{two})
111	N		

c. After handling a miss of address (11010_{two})

Esercizio 2 – soluzione (5/10)

22, 26, 22, 26, 16, 3, 16, 18, 16

- Riferimento a word con indirizzo 16 (10000_{two}) → miss

Index	V	Tag	Data
000	N		
001	N		
010	Y	11_{two}	Memory (11010_{two})
011	N		
100	N		
101	N		
110	Y	10_{two}	Memory (10110_{two})
111	N		

c. After handling a miss of address (11010_{two})



Index	V	Tag	Data
000	Y	10_{two}	Memory (10000_{two})
001	N		
010	Y	11_{two}	Memory (11010_{two})
011	N		
100	N		
101	N		
110	Y	10_{two}	Memory (10110_{two})
111	N		

d. After handling a miss of address (10000_{two})

Esercizio 2 – soluzione (6/10)

22, 26, 22, 26, 16, 3, 16, 18, 16

- Riferimento a word con indirizzo 3 → miss

Index	V	Tag	Data
000	Y	10_{two}	Memory (10000_{two})
001	N		
010	Y	11_{two}	Memory (11010_{two})
011	N		
100	N		
101	N		
110	Y	10_{two}	Memory (10110_{two})
111	N		

d. After handling a miss of address (10000_{two})



Index	V	Tag	Data
000	Y	10_{two}	Memory (10000_{two})
001	N		
010	Y	11_{two}	Memory (11010_{two})
011	Y	00_{two}	Memory (00011_{two})
100	N		
101	N		
110	Y	10_{two}	Memory (10110_{two})
111	N		

e. After handling a miss of address (00011_{two})

Esercizio 2 – soluzione (7/10)

22, 26, 22, 26, 16, 3, 16, 18, 16

- Riferimento a word con indirizzo 16 (10000) → hit

Index	V	Tag	Data
000	Y	10_{two}	Memory (10000_{two})
001	N		
010	Y	11_{two}	Memory (11010_{two})
011	Y	00_{two}	Memory (00011_{two})
100	N		
101	N		
110	Y	10_{two}	Memory (10110_{two})
111	N		

e. After handling a miss of address (00011_{two})

Esercizio 2 – soluzione (8/10)

22, 26, 22, 26, 16, 3, 16, 18, 16

- Riferimento a word con indirizzo 18 (10010_2) → miss

Index	V	Tag	Data
000	Y	10_{two}	Memory (10000_{two})
001	N		
010	Y	11_{two}	Memory (11010_{two})
011	N		
100	N		
101	N		
110	Y	10_{two}	Memory (10110_{two})
111	N		

d. After handling a miss of address (10000_{two})



Index	V	Tag	Data
000	Y	10_{two}	Memory (10000_{two})
001	N		
010	Y	10_{two}	Memory (10010_{two})
011	Y	00_{two}	Memory (00011_{two})
100	N		
101	N		
110	Y	10_{two}	Memory (10110_{two})
111	N		

f. After handling a miss of address (10010_{two})

Esercizio 2 – soluzione (9/10)

22, 26, 22, 26, 16, 3, 16, 18, 16

- Riferimento a word con indirizzo 16 (10000) → hit

Index	V	Tag	Data
000	Y	10_{two}	Memory (10000_{two})
001	N		
010	Y	10_{two}	Memory (10010_{two})
011	Y	00_{two}	Memory (00011_{two})
100	N		
101	N		
110	Y	10_{two}	Memory (10110_{two})
111	N		

f. After handling a miss of address (10010_{two})

Esercizio 2 – soluzione (10/10)

Decimal address of reference	Binary address of reference	Hit or miss in cache	Assigned cache block (where found or placed)
22	10110_{bin}	miss (5.6b)	$(10110_{\text{two}} \bmod 8) = 110_{\text{two}}$
26	11010_{bin}	miss (5.6c)	$(11010_{\text{two}} \bmod 8) = 010_{\text{two}}$
22	10110_{bin}	hit	$(10110_{\text{two}} \bmod 8) = 110_{\text{two}}$
26	11010_{bin}	hit	$(11010_{\text{two}} \bmod 8) = 010_{\text{two}}$
16	10000_{bin}	miss (5.6d)	$(10000_{\text{two}} \bmod 8) = 000_{\text{two}}$
3	00011_{bin}	miss (5.6e)	$(00011_{\text{two}} \bmod 8) = 011_{\text{two}}$
16	10000_{bin}	hit	$(10000_{\text{two}} \bmod 8) = 000_{\text{two}}$
18	10010_{bin}	miss (5.6f)	$(10010_{\text{two}} \bmod 8) = 010_{\text{two}}$
16	10000_{bin}	hit	$(10000_{\text{two}} \bmod 8) = 000_{\text{two}}$

Esercizio 3

- In una cache direct-mapped in cui ogni blocco contiene **due word**, qual è il valore del TAG (etichetta) per la parola di memoria con indirizzo 301_{10} ?

Esercizio 3 – soluzione

- TAG = parte alta dell'indirizzo (sequenza di bit non usati nell'indice del blocco)
- Per conoscere il TAG serve quindi conoscere il numero di bit usati per l'indice (quest'ultimo ricavabile dal numero di blocchi della cache, oppure dalla dimensione totale della cache e della dimensione di ogni blocco)
- Se ad esempio sono usati per l'indice 4 bit (informazione non disponibile in questo esercizio)

$$301 = 256 + 0 + 0 + 32 + 0 + 8 + 4 + 0 + 1 = (1001\ 0110\ 1)_2$$

il valore del TAG è 1001, l'indice è 0110 e il bit meno significativo è usato per individuare la parola

Esercizio 4a

- Sia data una cache direct-mapped con **blocchi di 4 parole** e una dimensione totale di 32 parole
- Assumendo **indirizzi** (di parola) a **16 bit**, determinare il numero di bit riservati all'etichetta, all'indice di blocco e alla parola

Esercizio 4a – Soluzione

Cache

0	blocco di 4 parole
1	
2	
3	
4	
5	
6	
7	

Memoria

0	blocco di 4 parole
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
...	

Se la cache contiene 32 parole in totale e ha blocchi da 4 word, il numero dei blocchi è 8

dei 16 bit dell'indirizzo, 3 sono usati per individuare l'indice (numero del blocco di cache) e 2 per individuare la parola (tra le 4 appartenenti al blocco) → TAG è di 11 bit

	Etichetta	Blocco Parola	
Indirizzo	11	3	2



Esercizio 4b

- Sia data una cache direct-mapped con **blocchi di 4 parole** e una dimensione totale di **32 parole** e la seguente sequenza di **indirizzi di parola** a cui si intende accedere:

1, 4, 8, 5, 33, 66, 32, 56, 9, 11, 4, 43, 88, 6, 32

- Determinare il numero di miss e hit alla cache, assumendo che la cache sia inizialmente vuota

Esercizio 4b – Soluzione

- Gli indirizzi a cui dobbiamo accedere sono espressi come indirizzi di parola
- Poichè abbiamo nella cache blocchi di 4 parole, trasformiamo gli indirizzi di parola in indirizzi di blocco

$$\text{indirizzo di blocco} = \left\lfloor \frac{\text{indirizzo di parola}}{\text{ampiezza di un blocco}} \right\rfloor = \left\lfloor \frac{\text{indirizzo di parola}}{4} \right\rfloor$$

- Quindi, la sequenza di indirizzi di blocco a cui si vuole accedere è:

*parte intera del
risultato della
divisione per 4*

indirizzo di parola	1, 4, 8, 5, 33, 66, 32, 56, 9, 11, 4, 43, 88, 6, 32
	↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓
indirizzo di blocco	0, 1, 2, 1, 8, 16, 8, 14, 2, 2, 1, 10, 22, 1, 8

Esercizio 4b – Soluzione

Indirizzi di blocco 0, 1, 2, 1, 8, 16, 8, 14, 2, 2, 1, 10, 22, 1, 8

Blocco della cache 0, 1, 2, 1, 0, 0, 0, 6, 2, 2, 1, 2, 6, 1, 0

blocco della cache = (numero blocco memoria) modulo (numero blocchi in cache)

Blocco	1	2	3	4	5	6	7	8
0	mem[0]	mem[0]	mem[0]	mem[0]	mem[8]	mem[16]	mem[8]	mem[8]
1		mem[1]	mem[1]	mem[1]	mem[1]	mem[1]	mem[1]	mem[1]
2			mem[2]	mem[2]	mem[2]	mem[2]	mem[2]	mem[2]
3								
4								
5								
6								mem[14]
7								

miss miss miss hit miss miss miss miss

Esercizio 4b – Soluzione

Indirizzi di blocco 0, 1, 2, 1, 8, 16, 8, 14, 2, 2, 1, 10, 22, 1, 8

blocco della cache = (numero blocco memoria) modulo (numero blocchi in cache)

Blocco	9	10	11	12	13	14	15
0	mem[8]	mem[8]	mem[8]	mem[8]	mem[8]	mem[8]	mem[8]
1	mem[1]	mem[1]	mem[1]	mem[1]	mem[1]	mem[1]	mem[1]
2	mem[2]	mem[2]	mem[2]	mem[10]	mem[10]	mem[10]	mem[2]
3							
4							
5							
6	mem[14]	mem[14]	mem[14]	mem[14]	mem[22]	mem[22]	mem[22]
7							
	hit	hit	hit	miss	miss	hit	hit

Risultato totale degli accessi alla cache = 9 miss e 6 hit

Esercizio 5

- Si supponga di avere una cache a mappaggio diretto con 4 blocchi da 2 word
- Se la cache parte vuota e si esegue 10 volte un loop che accede alle word che si trovano agli indirizzi 0, 1, 2, 3, 4, quanti miss ci saranno ?

Esercizio 5 – soluzione

- Possiamo fare la seguente considerazione:
 - al primo accesso si avrà un miss e le word che si trovano agli indirizzi 0 e 1 saranno caricate nel primo blocco della cache (quello con indice 0)
 - le word agli indirizzi 2 e 3 saranno caricate (dopo il secondo miss) nel blocco con indice 1
 - La word all'indirizzo 4 sarà caricata nel blocco con indice 2, dopo il terzo miss
 - Dopodichè, ripetendo il ciclo, ci saranno solo hit
- In totale si avranno quindi 3 miss

Esercizio 5 – soluzione alternativa

- In alternativa:
- ricaviamo da indirizzi di parola gli indirizzi di double word (dividendo per 2): 0123401234..... → 00112 00112.....
- Avremo solo MISS di primo load, ovvero MHMHHHH.....

Esercizio 5 – soluzione alternativa

- Convertiamo gli indirizzi in binario e consideriamo che il bit meno significativo è usato come **offset** per individuare la parola nel blocco, preceduti da 2 bit per l'**indice**, a sua volta preceduto dal tag

0...0**000**

0...0**001** (nella verifica della presenza del blocco in cache non

0...0**010** consideriamo l'offset)

0...0**011**

0...0**100**

- In totale si avranno quindi 3 miss
- P.S. In alternativa possiamo procedere come nell'esercizio 4b, ricavando la sequenza di accessi ai blocchi (dividendo per 2)

Esercizio 6

- Considerando indirizzi di memoria (**al byte**) a 32 bit per una cache direct-mapped con dimensione di ogni blocco di 4 parole e dimensione totale di **64 KiB**, qual è il numero di bit per i TAG?

- N.B. Se non è specificato diversamente, si assume **indirizzo al byte**

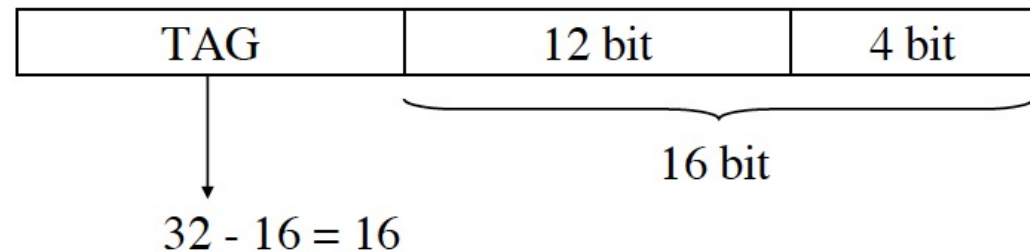
Esercizio 6 – Soluzione

- Ricaviamo il numero dei blocchi dividendo la dimensione totale (in byte) per il numero di byte contenuti in ogni blocco

$$64 \text{ KiByte} / 4 * 4 = 2^6 * 2^{10} / 2^2 * 2^2 = 2^{12} \text{ blocchi}$$

→ 12 bit per indice

- 4 bit sono usati per individuare uno dei 4 byte all'interno di una delle 4 parole
 - blocchi da 4 parole → 2 bit per individuare la parola
 - indirizzo al byte → 2 bit per individuare il byte all'interno della parola
- indirizzi di 32 bit → 16 bit per etichetta



Esercizio 7

Quanti bit sono necessari per una cache a mappaggio diretto con 16 KiB di dati e blocchi da 4 word, assumendo un indirizzamento a 32 bit?

Esercizio 7 – soluzione

Ogni blocco contiene 4 word (4*4 byte) di dati, quindi la cache è composta da
 $16 \text{ KiB} / (4 * 4 \text{ byte}) = 1024 \text{ blocchi} = 2^{10} \text{ blocchi}$
→ 10 bit dei 32 bit dell'indirizzo sono usati per l'indice

Numero di bit per TAG = $32 - (10 + \log_2 \# \text{word} + \log_2 \# \text{byte}) = 18$

oltre ai bit per l'indice, sono usati 2 bit per individuare la singola word tra le 4 di ogni blocco e 2 bit per individuare il singolo byte tra i 4 della parola)

Quindi per la cache (in bit) sono necessari

$$\begin{aligned} & 2^{10} * (4 * 32 \text{ bit di dati} + 18 \text{ bit per TAG} + 1 \text{ bit di validità}) = \\ & = 2^{10} * (128 + 18 + 1) \text{ bit} \\ & = 147 \text{ Kibit} \end{aligned}$$

Circa 18 KiByte

Kibibyte

Da Wikipedia, l'enciclopedia libera.

Il **kibibyte** è un'unità di misura dell'informazione o della quantità di dati, il termine deriva dalla contrazione di **kilo binary byte** ed ha per simbolo **KiB** (*KB* o *kIB* sono errati). Questo standard è stato definito dalla [Commissione Elettrotecnica Internazionale](#) (IEC) nel dicembre 1998 per stabilire i nuovi multipli del [byte](#).^[1]

Il kibibyte è collegato strettamente al [chilobyte](#), il quale è usato (creando spesso ambiguità) o per riferirsi a 10^3 byte (1 000 byte) oppure come sinonimo per kibibyte creando in quest'ultimo caso un errore del 2,4%.

1 kibibyte = 1 024 B = 2^{10} byte

1 kibibyte \approx 1 000 byte → 2,4% di errore

Esercizio 8

Data una macchina che richiede 5 cicli per accessi solo in cache (hit), 15 cicli di miss penalty e hit probability 0,75, qual è il numero medio di cicli per accesso alla memoria ?

Esercizio 8 – soluzione

Data una macchina che richiede 5 cicli per accessi solo in cache (hit), 15 cicli di miss penalty e hit probability 0,75, qual è il numero medio di cicli per accesso alla memoria ?

CPI = 5 cicli per istruzione quando non ci sono miss

miss penalty = 15 cicli

hit prob = 0,75

miss prob = 0,25

Num medio di cicli per accesso in memoria:

$$0,75*5 + 0,25*15 = 3,75 + 3,75 = 7,5$$

Esercizio 9

Data una macchina con frequenza di clock di 600 MHz, 2 CPI per accessi solo in cache, 22 cicli di miss penalty e hit probability 0,95, qual è velocità media in istruzioni/sec?

Esercizio 9 – soluzione

- 600 MHz di frequenza di clock
- CPI = 2 cicli/istr quando non ci sono miss (prob. del 95%)
- prob di miss al 5% (0,05) e 22 cicli di miss penalty

Velocità media (istr/sec) sarà pari alla frequenza (cicli/sec) divisa per il numero medio dei cicli eseguiti per ogni istruzione, ovvero:

$$\begin{aligned} 600 \text{ MHz} / (0,95 * 2 + 0,05 * 22) &= 600 \text{ milioni} / (1,9 + 1,1) \text{ istr/sec} \\ &= 200 \text{ milioni istr/sec} \end{aligned}$$

Esercizio 10

- Si considerino due cache, C1 e C2, costituite entrambe da **4 blocchi di due word**. La **cache C1 è direct-mapped**, la **cache C2 è fully-associative con politica di rimpiazzamento LRU**
- Determinare, nei due casi, il numero di miss per la seguente sequenza di operazioni di lettura di una word agli indirizzi:
508, 1016, 510, 24, 540, 1050, 1020, 24, 538
- **Gli indirizzi sono al byte, espressi in decimale**
- Si assuma che all'inizio le cache siano vuote
- Spiegare il ragionamento che ha portato alla risposta

Esercizio 10 – soluzione (C1 direct-mapped)

Essendo i blocchi da 2 word e gli indirizzi al byte, dividiamo tutti gli indirizzi per 8 (e consideriamo la parte intera del risultato), dopodichè calcoliamo il blocco in cui è mappato l'indirizzo richiesto attraverso l'operazione di modulo 4 (numero dei blocchi di cache)

- $508/8 = 63$ $63\%4 = 3 \rightarrow \text{miss}$
- $1016/8 = 127$ $127\%4=3 \rightarrow \text{miss}$
- $510/8 = 63$ $63\%4 = 3 \rightarrow \text{miss}$
- $24/8 = 3$ $3\%4 = 3 \rightarrow \text{miss}$
- $540/8 = 67$ $67\%4 = 3 \rightarrow \text{miss}$
- $1050/8 = 131$ $131\%4 = 3 \rightarrow \text{miss}$
- $1020/8 = 127$ $127\%4 = 3 \rightarrow \text{miss}$
- $24/8 = 3$ $3\%4 = 3 \rightarrow \text{miss}$
- $538/8 = 67$ $67\%4 = 3 \rightarrow \text{miss}$
- Si sarebbe potuto concludere che si hanno solo miss, osservando che tutti gli indirizzi sono mappati all'indice 3 e che non ci sono due accessi consecutivi che richiedono lo stesso indirizzo

Esercizio 10 – soluzione (C2 fully-associative)

Essendo i blocchi da 2 word e gli indirizzi al byte, dividiamo tutti gli indirizzi per 8 (e consideriamo la parte intera del risultato), dopodichè verificiamo se abbiamo miss o hit e nel caso di miss applichiamo la politica LRU

- $508/8 = 63$ miss ($63 \rightarrow 0$)
- $1016/8 = 127$ miss ($127 \rightarrow 1$)
- $510/8 = 63$ hit $\rightarrow 0$
- $24/8 = 3$ miss ($3 \rightarrow 2$)
- $540/8 = 67$ miss ($67 \rightarrow 3$)
- $1050/8 = 131$ miss ($131 \rightarrow 1$) LRU
- $1020/8 = 127$ miss ($127 \rightarrow 0$) LRU
- $24/8 = 3$ hit
- $538/8 = 67$ hit