

# Esercizi Cache



## organizzazione e tecniche di allocazione

**Es1:** Si consideri una cache di 4KB con associazione a gruppi a 8 vie ( 8-way set associative) in congiunzione con una memoria centrale di 1MB.

Supponendo che un blocco sia di dimensione 64B, si dica come un indirizzo di memoria è suddiviso in campi e a quanto ammonta la dimensione di ogni campo.

### Soluz.:



- trattandosi di una cache con associazione a gruppi, l'indirizzo di memoria centrale deve essere suddiviso nei campi tag, set, e parola.
- la memoria centrale è di 1MB, cioè  $2^{20}$  byte; pertanto un indirizzo di memoria centrale è espresso in 20 bit.
- la dimensione del campo parola è individuato univocamente dalla dimensione del blocco, che è di 64B, cioè  $2^6$  byte; pertanto il campo parola è di 6 bit.
- una cache di 4KB possiede  $2^{12}$  byte; ogni linea deve contenere un blocco e quindi impegna  $2^6$  byte; quindi la cache contiene  $2^{12} / 2^6 = 2^6$  linee. Poiché un insieme deve contenere 8 linee, il numero di insiemi della cache è pari a  $2^6 / 2^3 = 2^3$ . Pertanto il campo set è di 3 bit.
- la dimensione del campo tag sarà dunque:  $20 - 3 - 6 = 11$  bit

# Esercizi Cache



## organizzazione e tecniche di allocazione

**Es2:** Si consideri una cache di 16KB con associazione a gruppi a 4 vie ( 4-way set associative) e dimensione di linea di 32B.

Supponendo che il campo tag sia di 12 bit, si dica quale è la dimensione massima (in byte) di memoria principale che la cache è in grado di gestire, assumendo il singolo byte come unità di indirizzamento della memoria.

### Soluz.:



- per calcolare la quantità massima di memoria principale gestibile, bisogna calcolare il numero di bit totali che esprimono una generica locazione di memoria.
- trattandosi di una cache con associazione a gruppi, l'indirizzo di memoria centrale deve essere suddiviso nei campi tag, set, e parola.
- sappiamo che il campo tag è di 12 bit; quindi occorre calcolare la dimensione dei campi set e parola.
- la dimensione del campo parola è individuato univocamente dalla dimensione del blocco, che è di 32B, cioè  $2^5$  byte; pertanto il campo parola è di 5 bit.
- una cache di 16KB possiede  $2^{14}$  byte; ogni linea deve contenere un blocco e quindi impegna  $2^5$  byte; quindi la cache contiene  $2^{14}/2^5 = 2^9$  linee. Poiché un insieme deve contenere 4 linee, il numero di insiemi della cache è pari a  $2^9/2^2 = 2^7$ . Pertanto il campo set è di 7 bit.
- quindi la dimensione massima di memoria gestibile è:  $2^{12+7+5}$ , cioè 16MB

# Esercizi Cache



## organizzazione e tecniche di allocazione

**Es3:** Si consideri una cache di 4KB con associazione a gruppi a 4 vie ( 4-way set associative) in congiunzione con una memoria centrale di 256KB.

Supponendo che un blocco sia di dimensione 64B, si dica:

- a) se le locazioni di memoria con indirizzi (in esadecimale) **30E5C** e **17A87** hanno la possibilità di essere caricate all'interno dello stesso set di linee;
- b) se in cache è presente la locazione con indirizzo **05ABC**, quali altre locazioni sono sicuramente presenti nella cache.

### Soluz.:



a) procedendo come visto negli esercizi precedenti, abbiamo che un indirizzo di memoria è decomponibile in un campo parola di 6 bit, un campo set di 4 bit, ed un campo tag di 8 bit.

- le due locazioni di indirizzo **30E5C** e **17A87** possono trovarsi nello stesso insieme se il loro campo set è identico. Quindi basta controllare se i bit da 10 a 7 (a partire da destra) sono identici:

$30E5C = (\text{su } 18 \text{ bit}) 11000011\textcolor{red}{1001}011100$

$17A87 = (\text{su } 18 \text{ bit}) 01011110\textcolor{red}{1010}0000111$

- non essendo identici, la risposta è no.

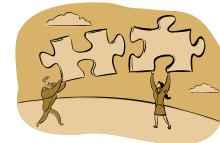
b) le altre locazioni che necessariamente saranno presenti con la locazione di indirizzo **05ABC** sono quelle all'interno del medesimo blocco.

- poiché **05ABC** = (su 18 bit)  $000101101010\textcolor{blue}{111100}$ , tutte le locazioni con indirizzo da  $000101101010\textcolor{blue}{000000}$  (hex **05A80**) a  $000101101010\textcolor{blue}{111111}$  (hex **05ABF**) si troveranno simultaneamente in cache.

**Es4a:** Siano date le seguenti 2 sequenze di indirizzi di memoria (parole) emessi dalla CPU

[illegible]

ed una cache ad associazione diretta di 16 blocchi di una parola. Dire come si decompongono gli indirizzi e determinare le corrispondenti sequenze di hit e miss



**Soluz.:**

- trattandosi di una cache con associazione diretta, l'indirizzo di memoria centrale deve essere suddiviso nei campi etichetta, linea, e parola, ed in particolare:
  - poiché un blocco è costituito da una sola parola, il campo parola è a dimensionalità nulla (0 bit)
  - il campo linea sarà costituito da 4 bit in quanto occorre indirizzare 16 ( $= 2^4$ ) linee (blocchi) di cache
  - il campo tag sarà quindi costituito dai rimanenti 28 bit più significativi
- di seguito si mostra, per ogni sequenza di indirizzi, la corrispondente sequenza di hit e miss



# Esercizi Cache



## organizzazione e tecniche di allocazione

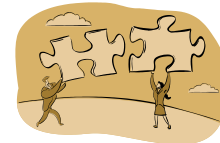
**Es4b:** Ripetere l'esercizio precedente nel caso di una cache ad associazione diretta di 8 blocchi, ognuno costituito da 2 parole.

**Es4c:** Con riferimento alle sequenze mostrate nell'es4a, supponendo di avere una cache ad associazione diretta in grado di memorizzare 8 parole, quale fra le seguenti dimensioni di blocco

- a) 1 parola
- b) 2 parole
- c) 4 parole

è la più conveniente (minimizza il numero di miss) ?

**Es4d:** Ripetere l'esercizio 4a nel caso di una cache ad associazione a 2 vie in grado di memorizzare 16 blocchi, ognuno costituito da 1 parola.



### Soluz. Es4b:

- trattandosi di una cache con associazione diretta, l'indirizzo di memoria centrale deve essere suddiviso nei campi etichetta, linea, e parola, ed in particolare:
  - poiché un blocco è costituito da 2 parole, il campo parola è di 1 bit
  - il campo linea sarà costituito da 3 bit in quanto occorre indirizzare 8 ( $= 2^3$ ) linee (blocchi) di cache
  - il campo tag sarà quindi costituito dai rimanenti 28 bit più significativi
- di seguito si mostra, per ogni sequenza di indirizzi, la corrispondente sequenza di hit e miss

```
l = linea
p = parola
```

	Sequenza 1	h/m	cache
	[tag   1 p]		
1	00000000000000000000000000000001	miss	blocco $1_{dec}/2 = 0_{dec}$ in linea 000
2	00000000000000000000000000001000	miss	blocco $134_{dec}/2 = 67_{dec}$ in linea 011
3	00000000000000000000000000001101	miss	blocco $212_{dec}/2 = 106_{dec}$ in linea 010
4	00000000000000000000000000000001	hit	
5	00000000000000000000000000001000	hit	
6	00000000000000000000000000001101	hit	
7	00000000000000000000000000001010	miss	blocco $162_{dec}/2 = 81_{dec}$ in linea 001
8	00000000000000000000000000001010	miss	blocco $161_{dec}/2 = 80_{dec}$ in linea 000 a
9	00000000000000000000000000000001	miss	blocco $2_{dec}/2 = 1_{dec}$ in linea 001 b
10	00000000000000000000000000001011	miss	blocco $44_{dec}/2 = 22_{dec}$ in linea 110
11	00000000000000000000000000001010	miss	blocco $41_{dec}/2 = 20_{dec}$ in linea 100
12	00000000000000000000000000001101	miss	blocco $221_{dec}/2 = 110_{dec}$ in linea 110 c

$$\begin{aligned} a &= [0_{\text{dec}} \text{ out}] \\ b &= [81_{\text{dec}} \text{ out}] \\ c &= [22_{\text{dec}} \text{ out}] \end{aligned}$$

```
l = linea
p = parola
```

[illegible]
$$a = [2_{\text{dec}} \text{ out}]$$

# Esercizi Cache



## organizzazione e tecniche di allocazione

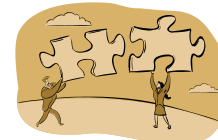
**Es4c:** Con riferimento alle sequenze mostrate nell'es4a, supponendo di avere una cache ad associazione diretta in grado di memorizzare 8 parole, quale fra le seguenti dimensioni di blocco

- a) 1 parola
- b) 2 parole
- c) 4 parole

è la più conveniente (minimizza il numero di miss) ?

**Es4d:** Ripetere l'esercizio 4a nel caso di una cache ad associazione a 2 vie in grado di memorizzare 16 blocchi, ognuno costituito da 1 parola. Usare la politica FIFO per il rimpiazzo dei blocchi.

### Soluz. 4c:



- trattandosi di una cache con associazione diretta, l'indirizzo di memoria centrale deve essere suddiviso nei campi etichetta, linea, e parola, ed in particolare:

- a) blocco costituito da una sola parola:  
il campo parola ha 0 bit; il campo linea sarà costituito da 3 bit in quanto occorre indirizzare 8 ( $= 2^3$ ) linee (blocchi) di cache; il campo tag sarà quindi costituito da  $(32 - 3 - 0) = 29$  bit;
- b) blocco costituito da 2 parole:  
il campo parola ha 1 bit ( $2^1$  parole); il campo linea sarà costituito da 2 bit in quanto occorre indirizzare 4 ( $= 2^2$ ) linee (blocchi) di cache; il campo tag sarà quindi costituito da  $(32 - 2 - 1) = 29$  bit;
- c) blocco costituito da 4 parole:  
il campo parola ha 2 bit ( $2^2$  parole); il campo linea sarà costituito da 1 bit in quanto occorre indirizzare 2 ( $= 2^1$ ) linee (blocchi) di cache; il campo tag sarà quindi costituito da  $(32 - 1 - 2) = 29$  bit.



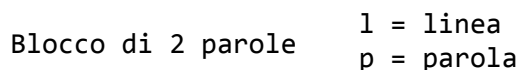


Blocco di 1 parola

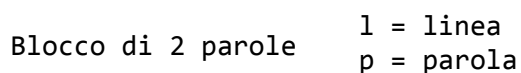
[illegible]

Blocco di 1 parola

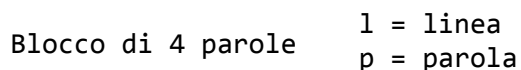
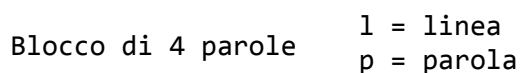
	Sequenza 2	h/m	cache
	[ tag   linea ]		
1	0000000000000000000000000000100	miss	blocco 4 <sub>dec</sub> in linea 100
2	000000000000000000000000000011010110	miss	blocco 214 <sub>dec</sub> in linea 110
3	000000000000000000000000000010101111	miss	blocco 175 <sub>dec</sub> in linea 111
4	000000000000000000000000000011010110	hit	
5	000000000000000000000000000000000000100	hit	
6	0000000000000000000000000000001010100	miss	blocco 84 <sub>dec</sub> in linea 100 [ 4 <sub>dec</sub> out]
7	0000000000000000000000000000000000001000001	miss	blocco 65 <sub>dec</sub> in linea 001
8	00000000000000000000000000000010101110	miss	blocco 174 <sub>dec</sub> in linea 110 [214 <sub>dec</sub> out]
9	00	miss	blocco 64 <sub>dec</sub> in linea 000
10	0000000000000000000000000000000000001101001	miss	blocco 105 <sub>dec</sub> in linea 001 [ 65 <sub>dec</sub> out]
11	0000000000000000000000000000001010101	miss	blocco 85 <sub>dec</sub> in linea 101
12	00000000000000000000000000000011010111	miss	blocco 215 <sub>dec</sub> in linea 111 [175 <sub>dec</sub> out]



```
a = [ 0dec out]
b = [ 81dec out]
c = [106dec out]
d = [ 80dec out]
e = [ 22dec out]
```



```
a = [102dec out]
b = [ 87dec out]
c = [  2dec out]
d = [102dec out]
e = [ 32dec out]
f = [ 87dec out]
```


$$\begin{array}{lcl} \mathbf{a} & = & \begin{bmatrix} 33_{\text{dec}} & \text{out} \end{bmatrix} \\ \mathbf{b} & = & \begin{bmatrix} 53_{\text{dec}} & \text{out} \end{bmatrix} \\ \mathbf{c} & = & \begin{bmatrix} 33_{\text{dec}} & \text{out} \end{bmatrix} \\ \mathbf{d} & = & \begin{bmatrix} 0_{\text{dec}} & \text{out} \end{bmatrix} \\ \mathbf{e} & = & \begin{bmatrix} 40_{\text{dec}} & \text{out} \end{bmatrix} \end{array} \qquad \begin{array}{lcl} \mathbf{f} & = & \begin{bmatrix} 53_{\text{dec}} & \text{out} \end{bmatrix} \\ \mathbf{g} & = & \begin{bmatrix} 0_{\text{dec}} & \text{out} \end{bmatrix} \\ \mathbf{h} & = & \begin{bmatrix} 11_{\text{dec}} & \text{out} \end{bmatrix} \end{array}$$

$$\begin{array}{ll} \mathbf{a} = \begin{bmatrix} 1_{\text{dec}} & \text{out} \end{bmatrix} & \mathbf{f} = \begin{bmatrix} 21_{\text{dec}} & \text{out} \end{bmatrix} \\ \mathbf{b} = \begin{bmatrix} 53_{\text{dec}} & \text{out} \end{bmatrix} & \mathbf{g} = \begin{bmatrix} 16_{\text{dec}} & \text{out} \end{bmatrix} \\ \mathbf{c} = \begin{bmatrix} 43_{\text{dec}} & \text{out} \end{bmatrix} & \mathbf{h} = \begin{bmatrix} 43_{\text{dec}} & \text{out} \end{bmatrix} \\ \mathbf{d} = \begin{bmatrix} 53_{\text{dec}} & \text{out} \end{bmatrix} & \mathbf{i} = \begin{bmatrix} 21_{\text{dec}} & \text{out} \end{bmatrix} \\ \mathbf{e} = \begin{bmatrix} 1_{\text{dec}} & \text{out} \end{bmatrix} & \end{array}$$



Riassumendo i risultati ottenuti abbiamo:

Dimensione blocco	Numero totale miss	Numero totale hit	Migliore
1	21	3	X
2	18	6	
4	21	3	



## Esercizi Cache



### organizzazione e tecniche di allocazione

**Es4c:** Con riferimento alle sequenze mostrate nell'es4a, supponendo di avere una cache ad associazione diretta in grado di memorizzare 8 parole, quale fra le seguenti dimensioni di blocco

- a) 1 parola
- b) 2 parole
- c) 4 parole

è la più conveniente (minimizza il numero di miss) ?

**Es4d:** Ripetere l'esercizio 4a nel caso di una cache ad associazione a 2 vie in grado di memorizzare 16 blocchi, ognuno costituito da 1 parola. Usare la politica FIFO per il rimpiazzo dei blocchi.

# Esercizi Cache



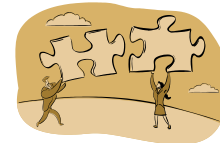
## organizzazione e tecniche di allocazione

**Es4c:** Con riferimento alle sequenze mostrate nell'es4a, supponendo di avere una cache ad associazione diretta in grado di memorizzare 8 parole, quale fra le seguenti dimensioni di blocco

- a) 1 parola
- b) 2 parole
- c) 4 parole

è la più conveniente (minimizza il numero di miss) ?

**Es4d:** Ripetere l'esercizio 4a nel caso di una cache ad associazione a 2 vie in grado di memorizzare 16 blocchi, ognuno costituito da 1 parola. Usare la politica FIFO per il rimpiazzo dei blocchi.



### Soluz. Es4d:

- trattandosi di una cache con associazione a 2 vie, l'indirizzo di memoria centrale deve essere suddiviso nei campi etichetta, set, e parola, ed in particolare:
  - poiché un blocco è costituito da 1 parola, il campo parola è di 0 bit
  - il campo set sarà costituito da 3 bit in quanto: il numero totale di linee della cache è 16 ( $= 2^4$ ), che devono essere organizzate in gruppi di 2 linee (vie) ognuno; pertanto il numero di gruppi (set) è 8 ( $= 2^4 / 2$ )
  - il campo tag sarà quindi costituito dai rimanenti  $(32 - 3 - 0) = 29$  bit più significativi
- di seguito si mostra, per ogni sequenza di indirizzi, la corrispondente sequenza di hit e miss

	Sequenza 1	h/m	cache
	[ tag   set ]		
1	00000000000000000000000000000001001	miss	blocco 1 <sub>dec</sub> in set 001[0]
2	0000000000000000000000000000010000110	miss	blocco 134 <sub>dec</sub> in set 110[0]
3	000000000000000000000000000011010100	miss	blocco 212 <sub>dec</sub> in set 100[0]
4	00000000000000000000000000000000001	hit	
5	0000000000000000000000000000010000111	miss	blocco 135 <sub>dec</sub> in set 111[0]
6	0000000000000000000000000000011010101	miss	blocco 213 <sub>dec</sub> in set 101[0]
7	0000000000000000000000000000010100010	miss	blocco 162 <sub>dec</sub> in set 010[0]
8	0000000000000000000000000000010100001	miss	blocco 161 <sub>dec</sub> in set 001[1]
9	00000000000000000000000000000000000010	miss	blocco 2 <sub>dec</sub> in set 010[1]
10	000000000000000000000000000001011000	miss	blocco 44 <sub>dec</sub> in set 100[1]
11	000000000000000000000000000001010001	miss	blocco 41 <sub>dec</sub> in set 001[0] a
12	0000000000000000000000000000011011101	miss	blocco 221 <sub>dec</sub> in set 101[1]

set **001**[0] si riferisce alla linea 0 del set 001

**Es5:** Sia data la seguente sequenza di indirizzi in lettura (l) o scrittura (s) emessi dalla CPU:

	Indirizzo	l/s	dato scritto (in esadecimale)
1	000100000000	l	B1  B4  B7
2	000100001000	l	
3	000100001100	s	
4	000100001100	l	
5	000100010000	s	
6	000100010000	l	
7	000100010100	s	

Si assuma che la dimensione di parola coincida con un byte, e la presenza di una cache di ampiezza 16B, dimensione di blocco 4B, inizialmente vuota, e ad associazione a 2 vie (con politica di rimpiazzo LRU e politica di scrittura write-through).

Si assuma che la memoria abbia il contenuto esadecimale mostrato di seguito:

ind	byte	ind	byte	ind	byte	ind	byte
100	0C	101	00	102	07	103	02
104	00	105	00	106	00	107	00
108	AE	109	13	10A	A1	10B	23
10C	A1	10D	42	10E	90	10F	75
110	B9	111	16	112	00	113	00
114	0A	115	07	116	03	117	71

ind = indirizzo

**Si mostri come sia il contenuto della cache che il contenuto della memoria cambia.**

### Soluzione:

Poiché un blocco è costituito da 4B, e la cache è di 16B, si avranno in cache  $16/4 = 4$  linee.

Essendo l'associatività a due linee (2 vie), la cache sarà costituita da due insiemi (set 0 e set 1) ognuno di 2 linee.

Quindi i 12 bit di indirizzo saranno suddivisi nel seguente modo:

- i 2 bit meno significativi individueranno il byte all'interno del blocco;
- il terzo bit da destra individuerà l'insieme (set 0 o set 1);
- i restanti bit costituiranno il campo tag.

Mostriamo di seguito l'evoluzione del contenuto della cache e della memoria.

Per la cache, nel caso in cui tutte e due le linee di un insieme (set) siano libere, si sceglie la linea con indirizzo minore per la allocazione (scelta arbitraria: si poteva usare un criterio diverso).

In caso di miss per una operazione di scrittura, si assume la politica "write allocate", cioè si porta prima in cache il blocco che contiene la parola da scrivere e poi si effettua la scrittura.

### Codifica della soluzione

ind. rif. memoria	cache dati		modifica memoria mem[ind.] = cont.
	set 0	set 1	
hex	[ linea 0 ]	[ linea 2 ]	
binario	t: tag	t: tag	
	r: rif.	r: rif.	
	[ linea 1 ]	[ linea 3 ]	
	t: tag	t: tag	
	r: rif.	r: rif.	

```

100      [0C|00|07|02]
000100000000 t:000100000
r:miss
[          ]
t:
r:

```

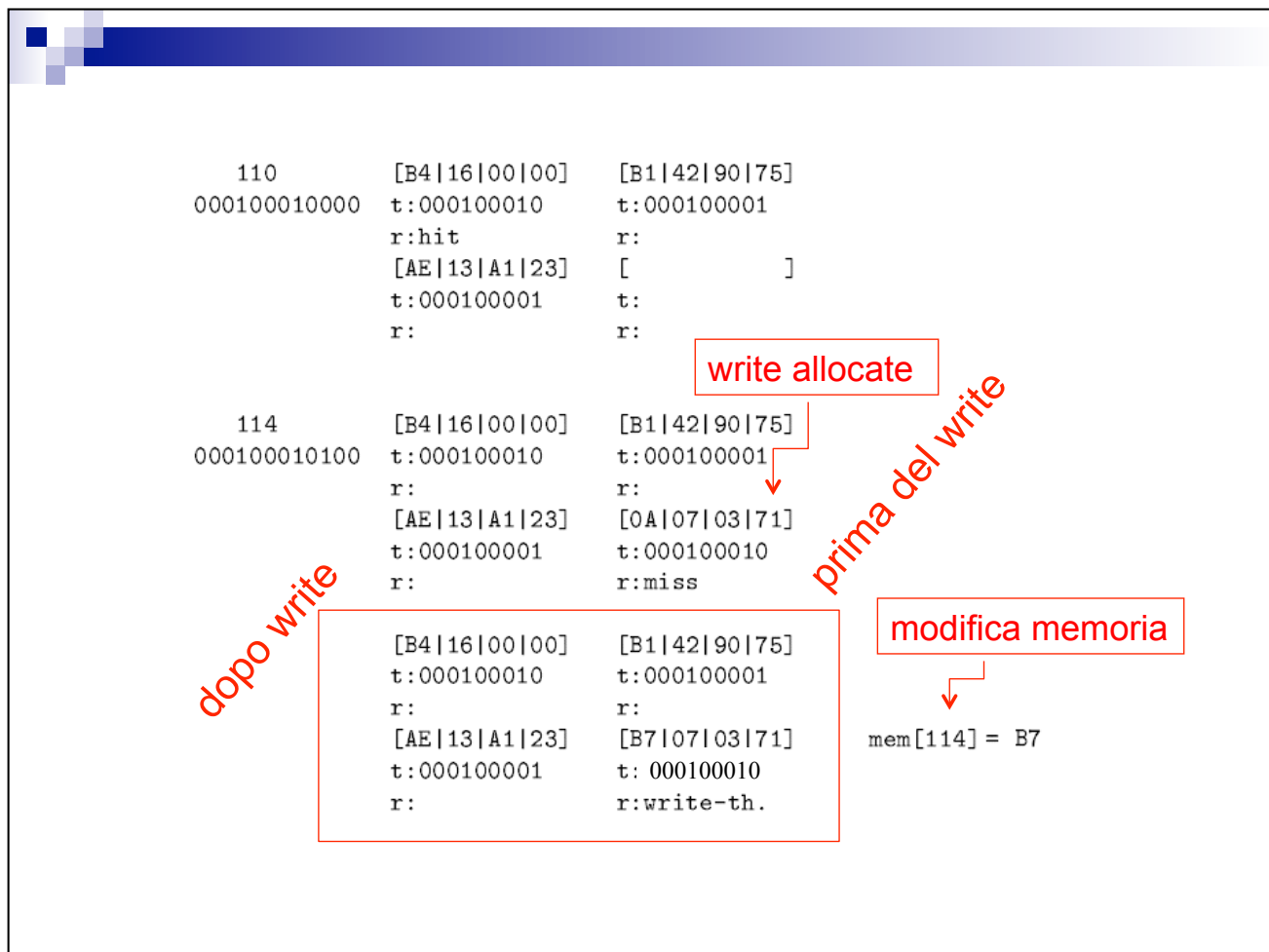
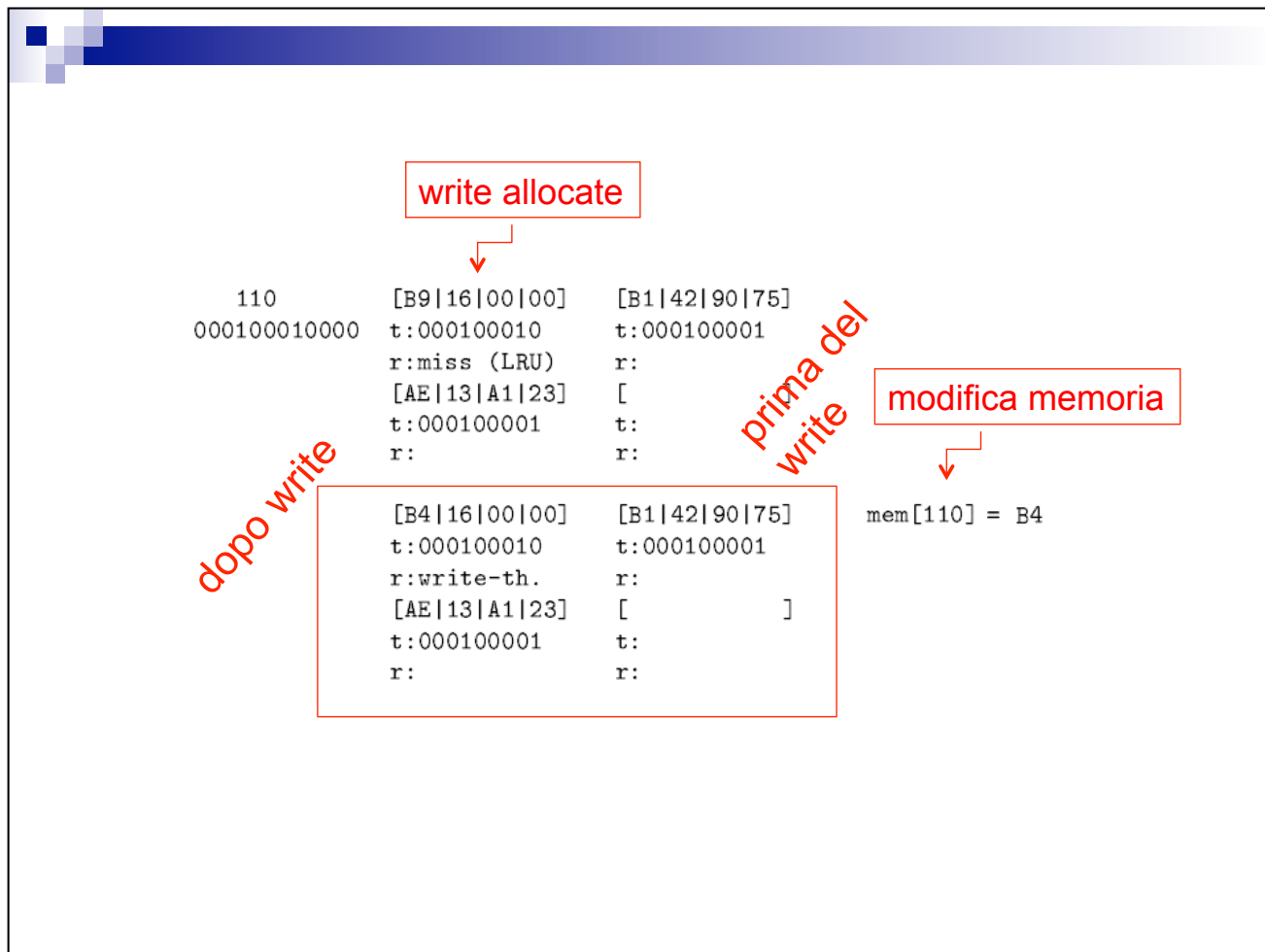
```

108      [0C|00|07|02]
000100001000 t:000100000
r:
[AE|13|A1|23]
t:000100001
r:miss

```







## Esercizio

es8

Sia data la seguente sequenza di indirizzi in lettura (l) o scrittura (s) emessi dalla CPU e che la memoria abbia il contenuto esadecimale mostrato di seguito:

#	indirizzo (binario)	l/s	byte scritto (HEX)	ind	byte	ind	byte	ind	byte	ind	byte
1	000100001000	s	43	100	08	101	D0	102	07	103	02
2	000100001100	s	3F	104	00	105	00	106	00	107	00
3	000100001111	l		108	AE	109	13	10A	A1	10B	23
4	000100001101	l		10C	A1	10D	42	10E	90	10F	75
5	000100010100	l		110	BB	111	16	112	00	113	00
6	000100011111	s	AE	114	0A	115	87	116	03	117	71
7	000100000111	s	CD	118	3E	119	13	11A	A1	11B	23
8	000100100110	l		11C	A1	11D	82	11E	90	11F	15
				120	F9	121	86	122	A0	123	00
				124	E9	125	16	126	05	127	00

Si assuma che la dimensione di parola coincida con un byte, e la presenza di una cache di ampiezza 16B, dimensione di blocco 2B, inizialmente vuota, e ad associazione a 2 vie (politica di rimpiazzo LRU, politica di scrittura write-back e gestione dei miss in scrittura con la politica write allocate).

Si mostri come sia il contenuto della cache che il contenuto della memoria cambia.

**Soluzione** (da compilare)

- Indicare di seguito in quali campi (e la loro dimensione) gli indirizzi emessi dalla CPU sono suddivisi:  
tag (o etichetta) da 9 bit, set (o insieme) da 2 bit, word (o parola) da 1 bit
- Indicare di seguito in quante linee/set la cache è suddivisa:  
La cache è costituita da 4 set, ognuno di 2 linee da 2B

Indicare l'evoluzione della cache e della modifica della memoria nello schema sottostante:

Indirizzo	hit/ miss	Cache (per ogni linea di cache indicare il contenuto del campo tag)	Modifica memoria $M[ind.] = \text{contenuto}$
$108_{HEX}$ 000100001000	miss	<div> set 00                      set 01                      set 10                      set 11 </div> <div> linea 0 [AE13] write allocate ↓ linea 0 [4313]* tag:000100001 </div>	
$10C_{HEX}$ 000100001100	miss	<div> linea 0 [4313]* tag:000100001 </div> <div> linea 0 [A142] write allocate ↓ linea 0 [3F42]* tag:000100001 </div>	

continuare nella pagina seguente

Indirizzo	hit/ miss	Cache (per ogni linea di cache indicare il contenuto del campo tag)				Modifica memoria $M[ind.] = contenuto$
		set 00	set 01	set 10	set 11	
$10F_{HEX}$ 000100001111	miss	linea 0 [4313]* tag:000100001		linea 0 [3F42]* tag:000100001	linea 0 [9075] tag:000100001	
$10D_{HEX}$ 000100001101	hit	linea 0 [4313]* tag:000100001		linea 0 [3F42]* tag:000100001	linea 0 [9075] tag:000100001	
$114_{HEX}$ 000100010100	miss	linea 0 [4313]* tag:000100001		linea 0 [3F42]* tag:000100001	linea 0 [9075] tag:000100001	
				linea 1 [0A87] tag:000100010		
$11F_{HEX}$ 000100011111	miss	linea 0 [4313]* tag:000100001		linea 0 [3F42]* tag:000100001	linea 0 [9075] tag:000100001	
				linea 1 [0A87] tag:000100010	linea 1 [9015] write allocate ↓ linea 1 [90AE]* tag:000100011	
$107_{HEX}$ 000100000111	miss	linea 0 [4313]* tag:000100001		linea 0 [3F42]* tag:000100001	linea 0 [LRU] [0000] write allocate	
				linea 1 [0A87] tag:000100010	linea 1 [90AE]* tag:000100011	
					↓	
		linea 0 [4313]* tag:000100001		linea 0 [3F42]* tag:000100001	linea 0 [LRU] [00CD]* tag:000100000	
				linea 1 [0A87] tag:000100010	linea 1 [90AE]* tag:000100011	
$126_{HEX}$ 000100100110	miss	linea 0 [4313]* tag:000100001		linea 0 [3F42]* tag:000100001	linea 0 [00CD]* tag:000100000	
				linea 1 [0A87] tag:000100010	linea 1 [LRU] [0500] tag:000100100	$M[11E] = 90$ $M[11F] = AE$

\* indica linea sporca a causa della politica write-back

## Esercizio

es8

Sia data la seguente sequenza di indirizzi in lettura (l) o scrittura (s) emessi dalla CPU e che la memoria abbia il contenuto esadecimale mostrato di seguito:

#	indirizzo (binario)	l/s	byte scritto (HEX)	ind	byte	ind	byte	ind	byte	ind	byte
1	000100000100	l	3F	100	08	101	D0	102	07	103	02
2	000100001100	s		104	00	105	00	106	00	107	00
3	000100001111	l	A9	108	AE	109	13	10A	A1	10B	23
4	000100001101	s		10C	A1	10D	42	10E	90	10F	75
5	000100010100	l	5E	110	BB	111	16	112	00	113	00
6	000100011111	s		114	0A	115	87	116	03	117	71
7	000100000111	s	66	118	3E	119	13	11A	A1	11B	23
8	000100100110	l		11C	A1	11D	82	11E	90	11F	15
				120	F9	121	86	122	A0	123	00
				124	E9	125	16	126	05	127	00

Si assuma che la dimensione di parola coincida con un byte, e la presenza di una cache di ampiezza 32B, dimensione di blocco 4B, inizialmente vuota, e ad associazione a 2 vie (politica di rimpiazzo FIFO, politica di scrittura write-through e gestione dei miss in scrittura con la politica write allocate).

Si mostri come sia il contenuto della cache che il contenuto della memoria cambia.

**Soluzione** (da compilare)

- Indicare di seguito in quali campi (e la loro dimensione) gli indirizzi emessi dalla CPU sono suddivisi:  
tag (o etichetta) da 8 bit, set (o insieme) da 2 bit, word (o parola) da 2 bit
- Indicare di seguito in quante linee/set la cache è suddivisa:  
La cache è costituita da 4 set, ognuno di 2 linee da 4B

Indicare l'evoluzione della cache e della modifica della memoria nello schema sottostante:

Indirizzo	hit/ miss	Cache (per ogni linea di cache indicare il contenuto del campo tag)				Modifica memoria $M[ind.] = \text{contenuto}$
		set 00	set 01	set 10	set 11	
$104_{HEX}$ 000100000100	miss		linea 0 [00000000] tag:00010000			
$10C_{HEX}$ 000100001100	miss		linea 0 [00000000] tag:00010000		linea 0 [A1429075] write allocate ↓ linea 0 [3F429075] tag:00010000	$M[10C] = 3F$
$10F_{HEX}$ 000100001111	hit		linea 0 [00000000] tag:00010000		linea 0 [3F429075] tag:00010000	

continuare nella pagina seguente

Indirizzo	hit/ miss	Cache (per ogni linea di cache indicare il contenuto del campo tag)				Modifica memoria $M[ind.] = \text{contenuto}$
		set 00	set 01	set 10	set 11	
$10D_{HEX}$ 000100001101	hit		linea 0 [00000000] tag:00010000		linea 0 [3FA99075] tag:00010000	$M[10D] = A9$
$114_{HEX}$ 000100010100	miss		linea 0 [00000000] tag:00010000  linea 1 [0A870371] tag:00010001		linea 0 [3FA99075] tag:00010000	
$11F_{HEX}$ 000100011111	miss		linea 0 [00000000] tag:00010000  linea 1 [0A870371] tag:00010001		linea 0 [3FA99075] tag:00010000  linea 1 [A1829015] write allocate ↓ linea 1 [A182905E] tag:00010001	$M[11F] = 5E$
$107_{HEX}$ 000100000111	hit		linea 0 [00000066] tag:00010000  linea 1 [0A870371] tag:00010001		linea 0 [3FA99075] tag:00010000  linea 1 [A182905E] tag:00010001	$M[107] = 66$
$126_{HEX}$ 000100100110	miss		linea 0 [FIFO] [E9160500] tag:00010010  linea 1 [0A870371] tag:00010001		linea 0 [3FA99075] tag:00010000  linea 1 [A182905E] tag:00010001	