

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	
2	000100001000	l	
3	000100001100	s	B1
4	000100001100	l	
5	000100010000	s	B4
6	000100010000	l	
7	000100010100	s	B7

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

```

write allocate

↓

<pre> 10C [0C 00 07 02] [A1 42 90 75] 000100001100 t:000100000 t:000100001 r: r:miss [AE 13 A1 23] [] t:000100001 t: r: r: </pre>	<p><i>prima del write</i></p> <p>modifica memoria</p> <p>↓</p> <p>mem[10C] = B1</p>
<p><i>dopo write</i></p> <div style="border: 1px solid red; padding: 5px; width: fit-content; margin: 5px auto;"> <pre> [0C 00 07 02] [B1 42 90 75] t:000100000 t:000100001 r: r:write-th. [AE 13 A1 23] [] t:000100001 t: r: r: </pre> </div>	
<pre> 10C [0C 00 07 02] [B1 42 90 75] 000100001100 t:000100000 t:000100001 r: r:hit [AE 13 A1 23] [] t:000100001 t: r: r: </pre>	

