

# Esercizi Cache

## Architettura degli elaboratori

*Laurea in Informatica*

*Docente: Federico Corò*

# Esercizio 3 cache

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, e che la dimensione di parola sia di un singolo byte, 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.

# Esercizio 3-a cache - soluzione

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**, e che la dimensione di parola sia di un singolo byte, 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;

## Soluzione:

- procedendo come visto negli esercizi precedenti, calcoliamo le dimensioni dei campi tag, set e parola
- 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:

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

$$17A87 = (\text{su 18 bit}) \ 01011110\textcolor{red}{1010}000111$$

- non essendo identici, la risposta è no.

# Esercizio 3-b cache - soluzione

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**, e che la dimensione di parola sia di un singolo byte, si dica:

- b) se in cache è presente la locazione con indirizzo **05ABC**, quali altre locazioni sono sicuramente presenti nella cache.

**Soluzione:**

- 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) **000101101010111100** , tutte le locazioni con indirizzo da **000101101010000000** (hex **05A80**) a **000101101010111111** (hex **05ABF**) si troveranno simultaneamente in cache.

# Esercizio 4 cache

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

	Indirizzo	I/S	Dato scritto (in esadecimale)
1	0001 0000 0000	I	
2	0001 0000 1000	I	
3	0001 0000 1100	s	B1
4	0001 0000 1100	I	
5	0001 0001 0000	s	B4
6	0001 0001 0000	I	
7	0001 0001 0100	s	B7

Si assume 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 assume 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

Si mostri come cambiano sia il contenuto della cache che il contenuto della memoria

# Esercizio 4 cache - soluzione

Soluzione:

- Poiché un blocco è costituito da 4B, e la cache è di 16B, si avranno in cache  $2^4/2^2 = 2^2$  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 (**word**);
  - il **terzo bit** da destra individuerà l'insieme (set 0 o set 1);
  - i restanti bit costituiranno il campo **tag**.

# Esercizio 4 cache - soluzione

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

E' necessario fare assunzioni aggiuntive 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.

Indirizzo	Hit/ miss	Cache <i>(per ogni linea di cache indicare il contenuto del campo tag)</i>				Modifica memoria $M[\text{ind.}] = \text{contenuto}$
Binario HEX	Hit/ miss		Linea 0	Linea 1		Mem[\text{ind.}] = contenuto

# Soluzione Esercizio 4 cache – inizio

Indirizzo	Hit/ miss	Cache <i>(per ogni linea di cache indicare il contenuto del campo tag)</i>			Modifica memoria $M[ind.] = \text{contenuto}$
0001 0000 0000 100 HEX	miss		Linea 0	Linea 1	
		Set 0	[0C 00 07 02] tag:000100000		
		Set 1			
0001 0000 1000 108 HEX	miss		Linea 0	Linea 1	
		Set 0	[0C 00 07 02] tag:000100000	[AE 13 A1 23] tag: 0000100001	
		Set 1			
000100001100 10C HEX	miss		Linea 0	Linea 1	
		Set 0	[0C 00 07 02] tag:000100000	[AE 13 A1 23] tag: 000100001	Mem[10C]=B1
		Set 1	[A1 42 90 75] → [B1 42 90 75] write allocate	tag:000100001	
0001 0000 1100 10C HEX	hit		Linea 0	Linea 1	
		Set 0	[0C 00 07 02] tag:000100000	[AE 13 A1 23] tag: 000100001	
		Set 1	[B1 42 90 75] tag:000100001		

# Soluzione Esercizio 4 cache – continua

Indirizzo	Hit/ miss	Cache <i>(per ogni linea di cache indicare il contenuto del campo tag)</i>				Modifica memoria $M[ind.] = \text{contenuto}$
0001 0001 0000 110 HEX	miss			Linea 0	Linea 1	Mem[110]=B4
		Set 0	[B9 16 00 00] write allocate LRU	[B4 16 00 00] tag:000100010	[AE 13 A1 23] tag: 000100001	
0001 0001 0000 110 HEX	hit			Linea 0	Linea 1	
		Set 0	[B4 16 00 00] tag:000100010	[AE 13 A1 23] tag: 000100001		
0001 0001 0100 114 HEX	miss			Linea 0	Linea 1	Mem[114]=B7
		Set 0	[B4 16 00 00] tag:000100010	[AE 13 A1 23] tag: 000100001		
		Set 1	[B1 42 90 75] tag:000100001	[0A 07 03 71] write allocate	[B7 07 03 71] tag:000100010	

# **Esercizi Cache**

## **Architettura degli elaboratori**

*Laurea in Informatica*

*Docente: Federico Corò*

# Esercizio 5 cache

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

	Indirizzo	I/s	Dato scritto (HEX)
1	0001 0000 1000	s	43
2	0001 0000 1100	s	3F
3	0001 0000 1111	l	
4	0001 0000 1101	l	
5	0001 0001 0100	l	
6	0001 0001 1111	s	AE
7	0001 0000 0111	s	CD
8	0001 0010 0110	l	

ind	byte	ind	byte	ind	byte	ind	byte
100	8	101	D0	102	7	103	2
104	0	105	0	106	0	107	0
108	AE	109	13	10A	A1	10B	23
10C	A1	10D	42	10E	90	10F	75
110	BB	111	16	112	0	113	0
114	0A	115	87	116	3	117	71
118	3E	119	13	11A	A1	11B	23
11C	A1	11D	82	11E	90	11F	15
120	F9	121	86	122	A0	123	0
124	E9	125	16	126	5	127	0

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 cambiano sia il contenuto della cache che il contenuto della memoria**

# Esercizio 5 cache – inizio soluzione

Soluzione:

- 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 <i>(per ogni linea di cache indicare il contenuto del campo tag)</i>				Modifica memoria $M[\text{ind.}] = \text{contenuto}$
0001 0000 1000 108 HEX	miss		Linea 0		Linea 1	
		Set 00	[AE13] write allocate	→ [4313]* tag:000100001		
		Set 01				
		Set 10				
		Set 11				

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

# Esercizio 5 cache – soluzione

Indirizzo	Hit/ miss	Cache <i>(per ogni linea di cache indicare il contenuto del campo tag)</i>		Modifica memoria $M[ind.] = \text{contenuto}$															
0001 0000 1100 10C HEX	miss	<table border="1"> <thead> <tr> <th></th> <th>Linea 0</th> <th>Linea 1</th> </tr> </thead> <tbody> <tr> <td>Set 00</td><td>[4313]* tag:000100001</td><td></td></tr> <tr> <td>Set 01</td><td></td><td></td></tr> <tr> <td>Set 10</td><td>[A142] write allocate</td><td>[3F42]* tag:000100001</td></tr> <tr> <td>Set 11</td><td></td><td></td></tr> </tbody> </table>		Linea 0	Linea 1	Set 00	[4313]* tag:000100001		Set 01			Set 10	[A142] write allocate	[3F42]* tag:000100001	Set 11				
	Linea 0	Linea 1																	
Set 00	[4313]* tag:000100001																		
Set 01																			
Set 10	[A142] write allocate	[3F42]* tag:000100001																	
Set 11																			
0001 0000 1111 10F HEX	miss	<table border="1"> <thead> <tr> <th></th> <th>Linea 0</th> <th>Linea 1</th> </tr> </thead> <tbody> <tr> <td>Set 00</td><td>[4313]* tag:000100001</td><td></td></tr> <tr> <td>Set 01</td><td></td><td></td></tr> <tr> <td>Set 10</td><td>[3F42]* tag:000100001</td><td></td></tr> <tr> <td>Set 11</td><td>[9075] tag:000100001</td><td></td></tr> </tbody> </table>		Linea 0	Linea 1	Set 00	[4313]* tag:000100001		Set 01			Set 10	[3F42]* tag:000100001		Set 11	[9075] tag:000100001			
	Linea 0	Linea 1																	
Set 00	[4313]* tag:000100001																		
Set 01																			
Set 10	[3F42]* tag:000100001																		
Set 11	[9075] tag:000100001																		

# Esercizio 5 cache – soluzione

Indirizzo	Hit/ miss	Cache <i>(per ogni linea di cache indicare il contenuto del campo tag)</i>		Modifica memoria $M[ind.] = \text{contenuto}$
0001 0000 1101 10D HEX	hit	Linea 0	Linea 1	
		Set 00 [4313]* tag:000100001		
		Set 01		
		Set 10 [3F42]* tag:000100001		
		Set 11 [9075] tag:000100001		
0001 0001 0100 114 HEX	miss	Linea 0	Linea 1	
		Set 00 [4313]* tag:000100001		
		Set 01		
		Set 10 [3F42]* tag:000100001	[0A87] tag:000100010	
		Set 11 [9075] tag:000100001		

# Esercizio 5 cache – soluzione

Indirizzo	Hit/ miss	Cache <i>(per ogni linea di cache indicare il contenuto del campo tag)</i>		Modifica memoria $M[ind.] = \text{contenuto}$
0001 0001 1111 11F HEX	miss	Linea 0	Linea 1	
		Set 00      [4313]* tag:000100001	Set 01	
		Set 10      [3F42]* tag:000100001	[0A87] tag:000100010	
		Set 11      [9075] tag:000100001	[9015] write allocate	[90AE]* tag:000100011
0001 0000 0111 107 HEX	miss	Linea 0	Linea 1	
		Set 00      [4313]* tag:000100001		
		Set 01		
		Set 10      [3F42]* tag:000100001	[0A87] tag:000100010	
		Set 11      [0000] write allocate <b>LRU</b>	[00CD]* tag:000100000	[90AE]* tag:000100011

# Esercizio 5 cache – soluzione

Indirizzo	Hit/ miss	Cache <i>(per ogni linea di cache indicare il contenuto del campo tag)</i>		Modifica memoria $M[ind.] = \text{contenuto}$															
0001 0010 0110 126 HEX	miss		<table border="1"> <thead> <tr> <th></th> <th>Linea 0</th> <th>Linea 1</th> </tr> </thead> <tbody> <tr> <td>Set 00</td><td>[4313]* tag:000100001</td><td></td></tr> <tr> <td>Set 01</td><td></td><td></td></tr> <tr> <td>Set 10</td><td>[3F42]* tag:000100001</td><td>[0A87] tag:000100010</td></tr> <tr> <td>Set 11</td><td>[00CD]* tag:000100000</td><td>[0500] tag:000100010 <b>LRU</b></td></tr> </tbody> </table>		Linea 0	Linea 1	Set 00	[4313]* tag:000100001		Set 01			Set 10	[3F42]* tag:000100001	[0A87] tag:000100010	Set 11	[00CD]* tag:000100000	[0500] tag:000100010 <b>LRU</b>	M[11E] = 90 M[11F] = AE
	Linea 0	Linea 1																	
Set 00	[4313]* tag:000100001																		
Set 01																			
Set 10	[3F42]* tag:000100001	[0A87] tag:000100010																	
Set 11	[00CD]* tag:000100000	[0500] tag:000100010 <b>LRU</b>																	

# Esercizio 5 cache – inizio soluzione

Indirizzo	Hit/ miss	Cache <i>(per ogni linea di cache indicare il contenuto del campo tag)</i>		Modifica memoria $M[ind.] = \text{contenuto}$
0001 0000 1100 10C HEX	miss	Linea 0	Linea 1	
		Set 00                    [4313]* tag:000100001		
		Set 01		
		Set 10                    [A142] → [3F42]* write allocate            tag:000100001		
		Set 11		
0001 0000 1111 10F HEX	miss	Linea 0	Linea 1	
		Set 00                    [4313]* tag:000100001		
		Set 01		
		Set 10                    [3F42]* tag:000100001		
		Set 11                    [9075] tag:000100001		

## Esercizio 6 Cache

Corso di Architettura degli Elaboratori

cdl in Informatica, a.a. 2024/25, docente: Prof. Nicolò Navarin

### Esercizio 6 Cache

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

#	indirizzo (binario)	l/s	parola scritta (HEX)
1	000100001000	s	34A2239E
2	000100001100	l	
3	000100001100	l	
4	000100011000	s	9F33ABC1
5	000100011000	l	
6	000100011100	s	91DD39FA
7	000100000100	l	
8	000100000000	l	

ind byte ind byte ind byte ind byte

100	08	101	0A	102	D7	103	02	blocco
104	1F	105	00	106	80	107	E0	blocco
108	AE	109	73	10A	AF	10B	23	blocco
10C	A1	10D	42	10E	90	10F	75	blocco
110	B9	111	16	112	FD	113	D0	blocco
114	0A	115	07	116	03	117	71	blocco
118	3E	119	D3	11A	71	11B	23	blocco
11C	A1	11D	8A	11E	90	11F	15	blocco
120	F9	121	86	122	A0	123	00	
124	E9	125	16	126	05	127	00	

Si assuma che la dimensione di parola sia di **4 byte** (con memoria indirizzata al byte), e la presenza di una cache di ampiezza **32B**, dimensione di **blocco 8B**, 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**).

### Soluzione (da compilare)

- Indicare di seguito in quali campi (e la loro dimensione) gli indirizzi emessi dalla CPU sono suddivisi:



4 campi: etichetta, insieme, parola, byte. Si osserva che i 2 bit più a destra indicano la posizione del byte all'interno della parola, il campo parola sarà costituito da 1 bit poiché un blocco è costituito da 2 parole (8B/4B), il campo set da 1 bit in quanto la cache contiene  $32B/8B = 4$  linee, raggruppate in 2 gruppi da 2 linee, e il campo etichetta dai restanti 8 bit più a sinistra.

- Indicare di seguito in quante linee/set la cache `e suddivisa

2 set, ognuno composto da 2 linee

- Si mostri come cambiano sia il contenuto della cache che il contenuto della memoria

Indirizzo	Hit/ miss	Cache (per ogni linea di cache indicare il contenuto del campo tag)			Modifica memoria $M[ind.] = \text{contenuto}$									
0001 0000 1000 108 HEX	miss	<table border="1"> <tr> <td></td><td>Linea 0</td><td>Linea 1</td></tr> <tr> <td>Set 0</td><td></td><td></td></tr> <tr> <td>Set 1</td><td> <p>[AE 73 AF 23 A1 42 90 75] write allocate</p> <p>[34 A2 23 9E A1 42 90 75]* tag:10 HEX</p> </td><td></td></tr> </table>				Linea 0	Linea 1	Set 0			Set 1	<p>[AE 73 AF 23 A1 42 90 75] write allocate</p> <p>[34 A2 23 9E A1 42 90 75]* tag:10 HEX</p>		
	Linea 0	Linea 1												
Set 0														
Set 1	<p>[AE 73 AF 23 A1 42 90 75] write allocate</p> <p>[34 A2 23 9E A1 42 90 75]* tag:10 HEX</p>													
0001 0000 1100 10C HEX	<table border="1"> <tr> <td></td><td>Linea 0</td><td>Linea 1</td></tr> <tr> <td>Set 0</td><td></td><td></td></tr> <tr> <td>Set 1</td><td> <p>[34 A2 23 9E A1 42 90 75]* tag:10 HEX</p> </td><td></td></tr> </table>					Linea 0	Linea 1	Set 0			Set 1	<p>[34 A2 23 9E A1 42 90 75]* tag:10 HEX</p>		
	Linea 0	Linea 1												
Set 0														
Set 1	<p>[34 A2 23 9E A1 42 90 75]* tag:10 HEX</p>													
0001 0000 1100 10C HEX	<table border="1"> <tr> <td></td><td>Linea 0</td><td>Linea 1</td></tr> <tr> <td>Set 0</td><td></td><td></td></tr> <tr> <td>Set 1</td><td> <p>[34 A2 23 9E A1 42 90 75]* tag:10 HEX</p> </td><td></td></tr> </table>					Linea 0	Linea 1	Set 0			Set 1	<p>[34 A2 23 9E A1 42 90 75]* tag:10 HEX</p>		
	Linea 0	Linea 1												
Set 0														
Set 1	<p>[34 A2 23 9E A1 42 90 75]* tag:10 HEX</p>													
0001 0001 1000 118 HEX	miss	<table border="1"> <tr> <td></td><td>Linea 0</td><td>Linea 1</td></tr> <tr> <td>Set 0</td><td></td><td></td></tr> <tr> <td>Set 1</td><td> <p>[34 A2 23 9E A1 42 90 75]* tag:10 HEX</p> </td><td> <p>[3E D3 71 23 A1 8A 90 15] write allocate</p> <p>[9F 33 AB C1 A1 8A 90 15]* tag:11 HEX</p> </td></tr> </table>				Linea 0	Linea 1	Set 0			Set 1	<p>[34 A2 23 9E A1 42 90 75]* tag:10 HEX</p>	<p>[3E D3 71 23 A1 8A 90 15] write allocate</p> <p>[9F 33 AB C1 A1 8A 90 15]* tag:11 HEX</p>	
	Linea 0	Linea 1												
Set 0														
Set 1	<p>[34 A2 23 9E A1 42 90 75]* tag:10 HEX</p>	<p>[3E D3 71 23 A1 8A 90 15] write allocate</p> <p>[9F 33 AB C1 A1 8A 90 15]* tag:11 HEX</p>												
0001 0001 1000 118 HEX	<table border="1"> <tr> <td></td><td>Linea 0</td><td>Linea 1</td></tr> <tr> <td>Set 0</td><td></td><td></td></tr> <tr> <td>Set 1</td><td> <p>[34 A2 23 9E A1 42 90 75]* tag:10 HEX</p> </td><td> <p>[9F 33 AB C1 A1 8A 90 15]* tag:11 HEX</p> </td></tr> </table>					Linea 0	Linea 1	Set 0			Set 1	<p>[34 A2 23 9E A1 42 90 75]* tag:10 HEX</p>	<p>[9F 33 AB C1 A1 8A 90 15]* tag:11 HEX</p>	
	Linea 0	Linea 1												
Set 0														
Set 1	<p>[34 A2 23 9E A1 42 90 75]* tag:10 HEX</p>	<p>[9F 33 AB C1 A1 8A 90 15]* tag:11 HEX</p>												
0001 0001 1100 11C HEX	<table border="1"> <tr> <td></td><td>Linea 0</td><td>Linea 1</td></tr> <tr> <td>Set 0</td><td></td><td></td></tr> <tr> <td>Set 1</td><td> <p>[34 A2 23 9E A1 42 90 75]* tag:10 HEX</p> </td><td> <p>[9F 33 AB C1 91 DD 39 FA]* tag:11 HEX</p> </td></tr> </table>					Linea 0	Linea 1	Set 0			Set 1	<p>[34 A2 23 9E A1 42 90 75]* tag:10 HEX</p>	<p>[9F 33 AB C1 91 DD 39 FA]* tag:11 HEX</p>	
	Linea 0	Linea 1												
Set 0														
Set 1	<p>[34 A2 23 9E A1 42 90 75]* tag:10 HEX</p>	<p>[9F 33 AB C1 91 DD 39 FA]* tag:11 HEX</p>												
0001 0000 0100 104 HEX	miss	<table border="1"> <tr> <td></td><td>Linea 0</td><td>Linea 1</td></tr> <tr> <td>Set 0</td><td> <p>[08 0A D7 02 1F 00 80 E0] tag:10 HEX</p> </td><td></td></tr> <tr> <td>Set 1</td><td> <p>[34 A2 23 9E A1 42 90 75]* tag:10 HEX</p> </td><td> <p>[9F 33 AB C1 91 DD 39 FA]* tag:11 HEX</p> </td></tr> </table>				Linea 0	Linea 1	Set 0	<p>[08 0A D7 02 1F 00 80 E0] tag:10 HEX</p>		Set 1	<p>[34 A2 23 9E A1 42 90 75]* tag:10 HEX</p>	<p>[9F 33 AB C1 91 DD 39 FA]* tag:11 HEX</p>	
	Linea 0	Linea 1												
Set 0	<p>[08 0A D7 02 1F 00 80 E0] tag:10 HEX</p>													
Set 1	<p>[34 A2 23 9E A1 42 90 75]* tag:10 HEX</p>	<p>[9F 33 AB C1 91 DD 39 FA]* tag:11 HEX</p>												
0001 0000 0000 100 HEX	<table border="1"> <tr> <td></td><td>Linea 0</td><td>Linea 1</td></tr> <tr> <td>Set 0</td><td> <p>[08 0A D7 02 1F 00 80 E0] tag:10 HEX</p> </td><td></td></tr> <tr> <td>Set 1</td><td> <p>[34 A2 23 9E A1 42 90 75]* tag:10 HEX</p> </td><td> <p>[9F 33 AB C1 91 DD 39 FA]* tag:11 HEX</p> </td></tr> </table>					Linea 0	Linea 1	Set 0	<p>[08 0A D7 02 1F 00 80 E0] tag:10 HEX</p>		Set 1	<p>[34 A2 23 9E A1 42 90 75]* tag:10 HEX</p>	<p>[9F 33 AB C1 91 DD 39 FA]* tag:11 HEX</p>	
	Linea 0	Linea 1												
Set 0	<p>[08 0A D7 02 1F 00 80 E0] tag:10 HEX</p>													
Set 1	<p>[34 A2 23 9E A1 42 90 75]* tag:10 HEX</p>	<p>[9F 33 AB C1 91 DD 39 FA]* tag:11 HEX</p>												

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