Descrizione Sistema

La cpu dispone di una cache dati a 2 vie da 16KiB complessivi e linee da 64 byte, gestita con stato MESI e con politica di scrittura Write-Around in caso di miss.
Si considerino A,B,C immagazzinate in memoria a partire rispettivamente dagli indirizzi: 0x0100 8000, 0x0100 A000, 0x0100 C000

1) Si disegni la mappa della memoria

2) Si analizzi la dinamica della cache dati, e, tenendo ben presente che il sistema ha un solo caching agent, si risponda in modo preciso, schematico, conciso e tabellare ai seguenti quesiti:

Quali sono gli indici di set e linea, e i tag associati ad A,B,C per il primo e l'ultimo blocco contenenti le matrici?

Quante linee di cache occuperanno nel loro insieme? Possono i due vettori e la variabile stare per intero e simultaneamente in cache?

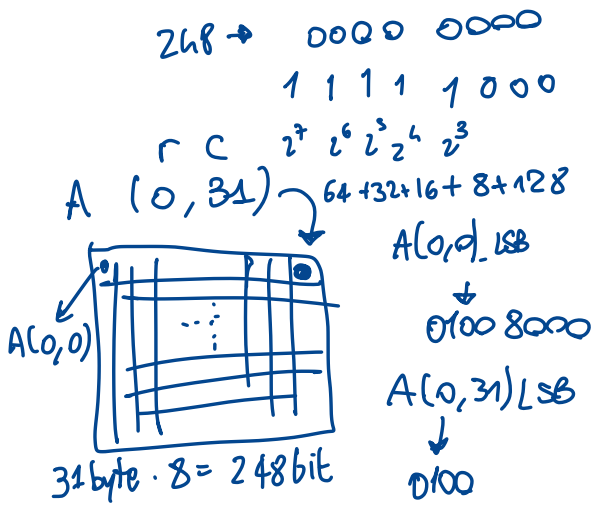
3) Si consideri la dinamica della cache nel calcolo della prima iterazione i=0, j=0, k=0 nel calcolo del primo elmento di C, disegnando lo stato MESI, il contenuto della cache e il valore del bit LRU dopo ogni operazione elementare (Load e Store) e si indichi il numero di accessi, il numero di miss e il numero di cicli di writeback e gli eventuali cicli di write allocate.

4) Si indichi il numero di accessi, il numero di miss e il numero di cicli di writeback e gli eventuali cicli di write allocate, nonché lo stato MESI della cache al termine del calcolo del primo elemento di C[0][0]. (si riporti il contenuto del primo ed ultimo set della cache)

a) Si disegni la mappa della memoria. **Punti 2**

BA[31..0]	Byte8	Byte6	Byte5	Byte4	Byte3	Byte2	Byte1	Byte0	BA[31..0]
0x0100 DFFF	c[31][31]_MSB							c[31][31]_LSB	0x0100 CFF8
0x0100 C007	c[0][0]_MSB							c[0][0]_LSB	0x0100 C000
0x0100 BFFF	b[31][31]_MSB							b[31][31]_LSB	0x0100 BFF8
0x0100 A007	b[0][0]_MSB							b[0][0]_LSB	0x0100 A000
0x0100 9FFF	a[31][31]_MSB							a[31][31]_LSB	0x0100 9FF8
0x0100 9F07	a[31][0]_LSB							a[31][0]_LSB	0x0100 8F00
0x0100 80FF	a[0][31]_MSB							a[0][31]_LSB	0x0100 80F8
0x0100 800F	a[0][1]_MSB							a[0][1]_LSB	0x0100 8008
0x0100 8007	a[0][0]_MSB							a[0][0]_LSB	0x0100 8000

Si compilino i campi in arancione con le risposte	
Nome	
Cognome	
Matricola	



byte + significativo e - sign. dell'ultimo elemento di A.

Bit meno significativo dell'ultimo elemento della 1ª riga di A

8 byte int

0100 8007 0100 80FF

0000 0001 0000 0000 1000 0000 0000 0000

0000 0001 0000 0000 1000 0000 0000 0111

1 + 2 + 4 = 7

A(0,0)_LSB: 0100 8000

A(0,31)_LSB: ?

0100 80F8

F 8

1111 1000

quindi faccio A(0,0) LSB → 0100 8000

a cui aggiungo 248 bit (31 × 8 byte),

ovvero vado all'ultima colonna

248 → 1111 1000

F 8

→

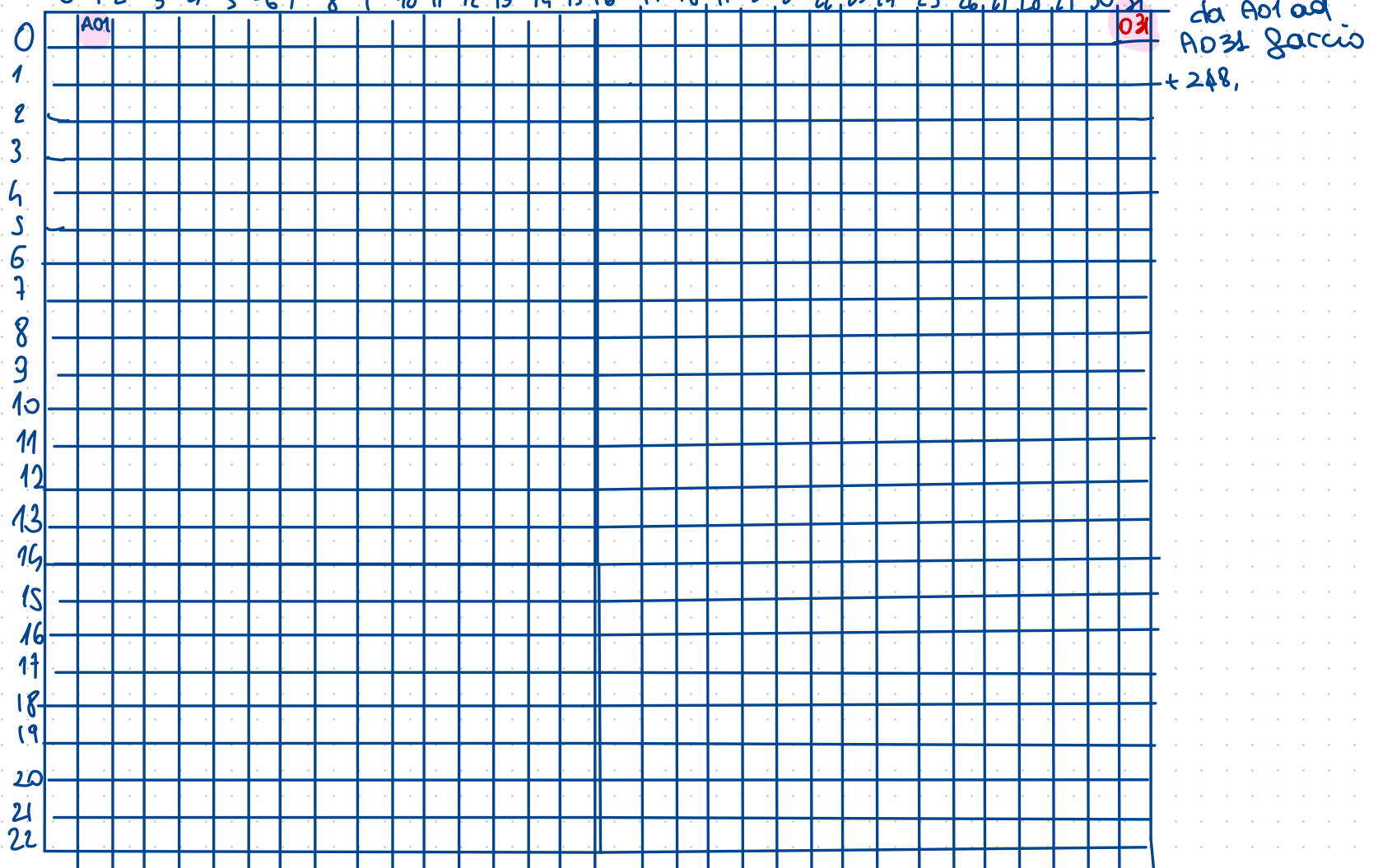
0100 8000 +

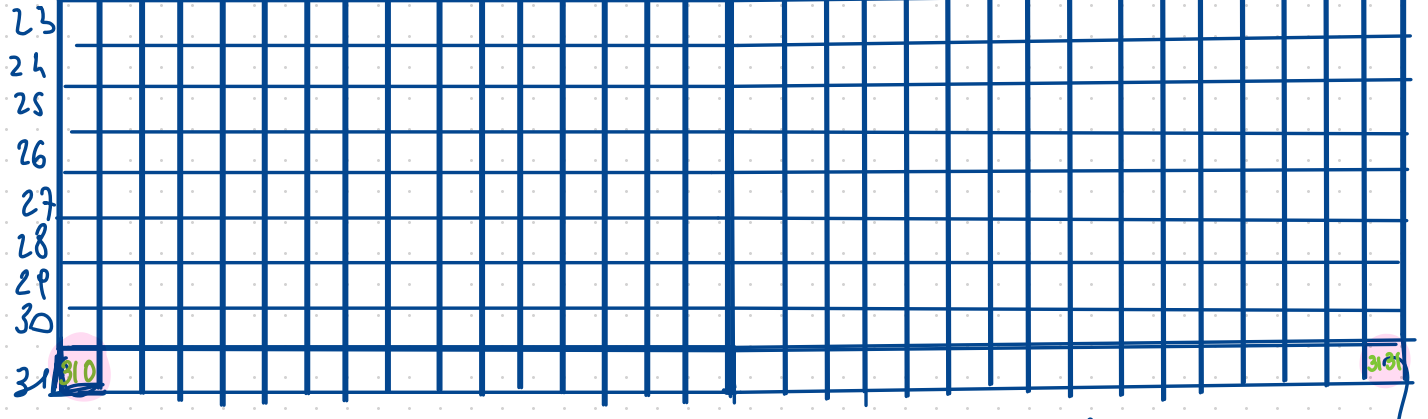
0000 00F8 =

0100 80F8

A(0,31)_LSB

ciascuna riga ha 32 elementi


0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
A01
A31
da A01 ad A31 saccio +248,



la matrice A ha 32 righe e 32 colonne

$2^5 \cdot 2^5 = 2^{10}$ questo con 8 byte

$\downarrow$

2^3

32 bit
4 byte

A[31,31]
+
64 bit =
8 byte

$\Downarrow$

$2^5 \cdot 2^5 \cdot 2^3 = 2^{13}$

il MSB lo sottrae con l'indirizzo base a cui sommare $2^{13} - 1$ $\rightarrow$

A CHE INDIRIZZO SI TROVA IL LSB di A[0][31] ?

31,31 LSB 0100 9FF8

31,31 MSB 0100 9FFF

↓

LSB 1001 1111 1111 1000

MSB 1001 1111 1111 1111) + 0111

↓

7

A CHE INDIRIZZO SI TROVA NON CHE VALORE HA.

0,0 MSB

31,0 MSB

0100 8007

0100 9007

⊕ 31

0100 8007 +

0 0 0 1 1 1 1 1

0101 9118

80FF

8007

B(0)(0) ↘ + $2^{13}-1$

B(31)(31) ↘

Testo: La cpu dispone di una cache dati a 2 vie da 16KiB complessivi e linee da 64 byte, gestita con stato MESI e con politica di scrittura Write-Around in caso di miss.

Capacità:		*2^14*	**16KiB**
Dimensione Blocco:		*2^6*	**64B**
Numero di blocchi nella cache:	*2^14 / 2^6*	*2^8*	**256**
Numero di set	*2^8/2*	*2^7*	**128**
Numero di bit index			**7**
Numero di bit di tag	*32-7-6*		**19**
	Scomposizione indirizzo:		
TAG BA[31..13]	Index BA[12...6]	Offset /Bytes in Blocks BA[5...0]	

16KB → blocchi da 64 byte che sono 2^6 byte

$\frac{2^{14}}{2^6} = 2^8$ blocchi → 2 lettere

0100 1111

n° blocchi / n° vie

BIT INDICE / BIT DI BYTE IN BLOCK

$\log_2 128$

Address	(Address >> 13) & 0x7 FFFF	(Address >> 6) & 111 1111	Address & 11 1111	0x	0	1	0	0	8	0	0	0
0x0100 8000	0x0 0804	0x00 - b000 0000 - d0	0x00		0000	0001	0000	0000	0000	1000	0000	0000
					0	1	0	0	8	0	F	8
0x0100 80F8	0x00804	0x03 - b000 0011 - d3	0x38		0000	0001	0000	0000	1000	0000	1111	1000

Quanti elementi di una matrice sono contenuti in una linea di cache? 8
perché 8 elementi da 8 byte fanno 64 byte

3. Si analizzi la dinamica della **cache** dati, e, tenendo ben presente **che il sistema ha un solo caching agent,** si risponda in modo preciso, **schematico, conciso e tabellare** ai seguenti quesiti: **11 Punti**

a) Quali sono gli indici di set e linea, e i tag associati ad A,B,C per il primo e l'ultimo blocco contenenti le matrici?

	Contenuto Blocco di Cache			BA[31..0]		TAG	Set_id/indice	*Numero progressivo delle linee/blocchi di cache occupati da A,B,C e D.*
Indirizzo primo elemento della linea/ blocco di cache	Da		A	Indirizzo ultimo elemento della linea/ blocco di cache		BA[?..?]	BA[?..?]	
	Si completi inserendo le variabili agli estremi del blocco di cache (Byte più significativo e meno significativo).					*Si completino sotto i "?" identificando i bit utilizzati per tag e indice.*		
0x0100 8000	A[0][0]		A[0][7]	0x0100 801F		0x0804	0x00	*1*
0x0100 9FE0	A[31][24]		A[31][31]	0x0100 9FFF		0x0804	0x3F	*127*
0x0100 A000	B[0][0]		B[0][7]	0x0100 A01F		0x0805	0x00	*128*
0x0100 BFE0	B[31][24]		B[31][31]	0x0100 BFFF		0x0805	0x3F	*255*
0x0100 C000	C[0][0]		C[0][7]	0x0100 C01F		0x0806	0x00	*128*
0x0100 DFE0	C[31][24]		C[31][31]	0x0100 DFFF		0x0806	0x3F	255

a) Quante linee di cache occuperanno nel loro insieme? Possono i due vettori e la variabile stare per intero e simultaneamente in cache?

No in quanto la cache ha capacità di 16KiB ma le tre matrici occupano 24KiB

3. Si analizzi la dinamica della cache dati, e, tenendo ben presente che il sistema ha un solo caching agent, si risponda in modo preciso, schematico, conciso e tabellare ai seguenti quesiti: 11 Punti

b) Si consideri la dinamica della cache nel calcolo della prima iterazione i=0, j=0, k=0 nel calcolo del primo elmento di C, disegnando lo stato MESI, il contenuto della cache e il valore del bit LRU dopo ogni operazione elementare (Load e Store) e si indichi il numero di accessi, il numero di miss e il numero di cicli di writeback e gli eventuali cicli di write allocate.

Si consideri la dinamica della cache nel calcolo della prima iterazione i=0, j=0, k=0 nel calcolo del primo elmento di C, disegnando lo stato MESI, il contenuto della cache e il valore del bit LRU dopo ogni operazione elementare (Load e Store).

Stato MESI	Passo:	0	Miss	1		Hit	0
	Operazione elementare:			f0 = C[i][k]			
	Via0			LRU	Via1		
set_id	TAG	Data	MESI		TAG	Data	MESI
0	0x0 0806	C[0][0]...C[0][7]	E	1			I

Stato MESI	Passo:	1	Miss	2		Hit	0
	Operazione elementare:			f1 = B[i][k]			
	Via0			LRU	Via1		
set_id	TAG	Data	MESI		TAG	Data	MESI
0	0x0 0806	C[0][0]...C[0][7]	E	0	0x0 0805	B[0][0]...B[0][7]	E

Stato MESI	Passo:	2	Miss	3		Hit	1
	Operazione elementare:			f2 = A[i][k]			
	Via0			LRU	Via1		
set_id	TAG	Data	MESI		TAG	Data	MESI
0	0x0 0804	A[0][0]...A[0][7]	E	1	0x0 0805	B[0][0]...B[0][7]	E

Stato MESI	Passo:	3	Miss	4		Hit	1
	Operazione elementare:			C[i][k] = f0			
	Via0			LRU	Via1		
set_id	TAG	Data	MESI		TAG	Data	MESI
0	0x0 0804	A[0][0]...A[0][7]	E	1	0x0 0805	B[0][0]...B[0][7]	E

Si indichi il numero di accessi, il numero di miss e il numero di cicli di writeback e gli eventuali cicli di write allocate.

nella prima iterazione ci sono 3 accessi in lettura, 3 missi in lettura

c) Si indichi il numero di accessi, il numero di miss e il numero di cicli di writeback e gli eventuali cicli di write allocate, nonché lo stato MESI della cache al termine del calcolo del primo elemento di C[0][0]. (si riporti il contenuto del primo ed ultimo set della cache)

Stato MESI	**Passo:**		**WB:**	**0**			
	Via0			**LRU**	Via1		
set_id	TAG	Data	MESI		TAG	Data	MESI
0	0x0 0804	A[0][0]...A[0][7]	E	1	0x0 0805	B[0][0]...B[0][7]	E
1	0x0 0804	A[0][8]...A[0][15]	E	1			
2	0x0 0804	A[0][16]...A[0][23]	E	1			
3	0x0 0804	A[0][24]...A[0][31]	E	1			
4	0x00805	B[1][0-7]	E	0			
127	0x00805	B[31][0-7]	E	0			

Si indichi il numero di accessi, il numero di miss e il numero di cicli di writeback e gli eventuali cicli di write allocate? Si giustifichi la domanda

Per il calcolo di C[0][0] vengono eseguite 32 iterazioni in k. A ciascuna iterazione corrispondono 2 accessi in lettura agli elementi di A[0][k] e B[k][0]. Per un totale di 32*2 accessi in lettura. Al termine delle 32 iterazioni viene eseguita la scrittura di C[0][0] ed un corrispondente accesso in scrittura all'elemento.

Ogni 8 accessi di A danno origine ad una miss in lettura e 7 hit in lettura per un totale di 4 miss e 28 hit.
Tutti gli accessi a B danno origine ad una miss in lettura per un totale di 32 miss.

La scrittura di C da origine ad una miss in scrittura, viene gestita in write around e non comporta uno spostamento di una linea di cache dalla memoria alla cache. Non ci sono cicli di WB.

Accessi 65

0x 0100 8000 →

0000 0001 0000 0000 1000 0000 0000 0000

Set ID 0X 00

tag ID 0X 00804 si ottiene shiftando a dx di 1 bit ovvero dividere per 2