

Esercizio cache

Sia data la seguente sequenza di istruzioni assembler, dove i dati immediati sono espressi in esadecimale ed il registro R0 contiene il valore 0:

```
LB   R3, 100(R0)    ! load byte da mem[100+[R0]]
ADD  R2, R0, R0     ! R2 = R0 + R0
LB   R1, 108(R2)    ! load byte da mem[108+[R2]]
ADDI R2, R2, 5      ! R2 = R2 + 5
SUB  R4, R3, R2     ! R4 = R3 - R2
ADDI R1, R1, 3      ! R1 = R1 + 3
SB   R1, 108(R2)    ! store byte in mem[108+[R2]]
BGTZ R4, -6         ! PC = PC - 6 se [R4] > 0
                        ! cioe' salta alla istruzione LB   R1, 108(R2)
```

Si assuma la presenza di due cache, una dati ed una istruzioni. La cache dati, in particolare, è di ampiezza 8B, con dimensione di blocco 4B, inizialmente vuota, ed associazione diretta (con politica di scrittura write-through). Si assuma che la memoria abbia il contenuto esadecimale mostrato di seguito (si esprimano gli indirizzi su 12 bit):

Indirizzo	byte	byte	byte	byte
100	0F	00	07	02
104	00	00	00	00
108	AE	13	A1	23
10C	A1	42	90	75
110	B9	16	00	00
114	0A	07	03	71

Si mostri come sia il contenuto della cache dati che il contenuto della memoria cambia a causa della esecuzione del codice assembler.

Soluzione

Poiché un blocco è costituito da 4B, e la cache è di 8B si avranno $8/4 = 2$ 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 la linea (0 o 1), ed i restanti bit costituiranno il tag. Mostriamo di seguito l'evoluzione del contenuto dei registri, dei riferimenti a memoria, della cache dati (solo quando cambia) e della memoria (solo quando cambia).

codice eseguito	[R1]	[R2]	[R3]	[R4]	ind. rif. memoria hex binario	cache dati		modifica memoria mem[ind.] = cont.
	hex	hex	hex	hex		[linea 0]	[linea 1]	
						t: tag	t: tag	
						r: rif.	r: rif.	
LB R3, 100(R0)	?	?	F	?	100 000100000000	[0F 00 07 02] t:000100000 r:miss		
ADD R2, R0, R0	?	0	F	?				
LB R1, 108(R2)	AE	0	F	?	108 000100001000	[AE 13 A1 23] t:000100001 r:miss		
ADDI R2, R2, 5	AE	5	F	?				
SUB R4, R3, R2	AE	5	F	A				
ADDI R1, R1, 3	B1	5	F	A				
SB R1, 108(R2)	B1	5	F	A	10D 000100001101	[AE 13 A1 23] t:000100001 r: [AE 13 A1 23] t:000100001 r:	[A1 42 90 75] t:000100001 r:miss [A1 B1 90 75] t:000100001 r:write-th.	mem[10D] = [R1] = B1
BGTZ R4, -6	B1	5	F	A				
LB R1, 108(R2)	B1	5	F	A	10D 000100001101	[AE 13 A1 23] t:000100001 r:	[A1 B1 90 75] t:000100001 r:hit	
ADDI R2, R2, 5	B1	A	F	A				
SUB R4, R3, R2	B1	A	F	5				
ADDI R1, R1, 3	B4	A	F	5				
SB R1, 108(R2)	B4	A	F	5	112 000100010010	[B9 16 00 00] t:000100010 r:miss [B9 16 B4 00] t:000100010 r:write-th.	[A1 B1 90 75] t:000100001 r: [A1 B1 90 75] t:000100001 r:	mem[112] = [R1] = B4
BGTZ R4, -6	B4	A	F	5				
LB R1, 108(R2)	B4	A	F	5	112 000100010010	[B9 16 B4 00] t:000100010 r:hit	[A1 B1 90 75] t:000100001 r:	
ADDI R2, R2, 5	B4	F	F	5				
SUB R4, R3, R2	B4	F	F	0				
ADDI R1, R1, 3	B7	F	F	0				
SB R1, 108(R2)	B7	F	F	0	117 000100010111	[B9 16 B4 00] t:000100010 r: [B9 16 B4 00] t:000100010 r:	[0A 07 03 71] t:000100010 r:miss [0A 07 03 B7] t:000100010 r:write-th.	mem[117] = [R1] = B7
BGTZ R4, -6	B7	F	F	0				