

Usare un foglio separato per rispondere ai due esercizi seguenti, specificando nell'intestazione: **Titolo del corso** (*Architettura degli Elaboratori – modulo II* oppure *Architettura degli Elaboratori B*), **Data esame**, **Cognome e Nome**, **Matricola**

Esercizio 1

Considerare un sistema di memoria virtuale di cui siano noti i seguenti dati: indirizzo virtuale di 32 b, page size di 4 KB, TLB associativa a 2 vie con 256 ingressi, dove la dimensione di ogni ingresso (TAG + Num. Pag. Fisica) è 4 B. Qual è la dimensione dell'indirizzo fisico?

Tramite l'indirizzo fisico così ottenuto, si accede a una cache associativa a 4 vie, composta da 2^{12} insiemi, e i cui blocchi sono grandi 16 B.

Qual è la dimensione (dati) della cache in KB?

Qual è la dimensione della TAG?

Rispetto all'esecuzione di un'istruzione che riferisce l'indirizzo fisico 0x100FAEA0, e provoca il conseguente caricamento di un blocco in cache, quali degli indirizzi seguenti corrisponde a un riferimento a: (1) stesso insieme e stesso blocco, oppure (2) stesso insieme ma blocco diverso, oppure (3) insieme (e anche blocco) diverso:

0x100FAEA4
0x100FAEB0
0x1000AEA0

Soluzione

Il page offset è $\log 4K = \log 2^{12} = 12$ b, da cui

$$VIRT_PAGE_NO = VIRT_ADDR - PAGE_OFFSET = 32 - 12 = 20$$

Il numero di insiemi della TLB è uguale a $256/2 = 128$.

L'index della TLB è $INDEX = \log NUM_SET = \log 2^7 = 7$ b, da cui:

$$TAG = VIRT_PAGE_NO - INDEX = 20 - 7 = 13 \text{ b.}$$

Quindi $PHIS_PAGE_NO = \dim_entry_TLB - TAG = 32 - 13 = 19$ b, da cui:

$$PHIS_ADDR = PHIS_PAGE_NO + PAGE_OFFSET = 19 + 12 = 31 \text{ b.}$$

Il numero di blocchi dati della cache è $NUM_SET * NUM_VIE = 2^{12} = 2^{14}$. Poiché ogni blocco è 2^4 B, abbiamo che:

$$CACHE_SIZE = 2^{14} * 2^4 = 2^8 * 2^{10} \text{ B} = 2^8 \text{ KB} = 256 \text{ KB.}$$

Poiché $BLOCK_OFFSET = \log 16 = 4$ b, e $INDEX = \log NUM_SET = \log 2^{12} = 12$ b, abbiamo che:

$$TAG = PHIS_ADDR - BLOCK_OFFSET - INDEX = 31 - 4 - 12 = 15 \text{ b.}$$

Per rispondere all'ultima domanda, basta porre una maschera sugli indirizzi in modo da evidenziare TAG, INDEX e BLOCK_OFFSET.

| TAG | IND | OFF | |
|------|-----|-----|---|
| 100F | AEA | 0 | |
| 100F | AEA | 4 | ---- TAG e INDEX uguali, e OFFSET diversi. Caso (1) |
| 100F | AEB | 0 | ---- TAG uguali e INDEX diversi. Caso (3) |
| 1000 | AEA | 0 | ---- TAG diversi ma INDEX uguali. Caso (2) |

Esercizio 2

Considerare la seguente porzione di programma assembler MIPS:

```

Loop:
    lw    $s0, 0($t0)
    addi  $s0, $s0, 100
    sw    $s0, 0($t0)
    addi  $t0, $t0, 4
    bne   $t0, $t1, Loop

```

1. Cosa calcola il *loop*?
2. Supporre che il *loop* sia eseguito su un sistema, la cui gerarchia di memoria include una cache con blocchi da 4 B. Calcolare i *data miss* per l'esecuzione di ogni iterazione del loop. Calcolare il *data miss rate* complessivo. Come cambierebbe la risposta se il blocco fosse di 32 B?

3. **(solo Mod. II)** Inglobare il codice di sopra in una funzione, aggiungendo il prologo e l'epilogo della funzione, rispettando le convenzioni di chiamata MIPS. I parametri sono passati alla funzione tramite registri, in accordo alle solite convenzioni (\$a0, \$a1, ...). In particolare, i due valori passati alla funzione sono quelli contenuti in \$t0 e \$t1.
- 3'. **(solo Arch. B)** Determinare le dipendenze RAW. Rispetto ad un'architettura pipeline a cinque stadi, senza forwarding ma con *register file speciale*, disegnare il diagramma di esecuzione. Cosa succede al ciclo di clock 9?

Soluzione

- Incrementa della costante 100 un vettore di interi.
- Ad ogni iterazione del loop abbiamo un miss (lw) e un hit (sw), poiché il blocco è esattamente 1 word. Quindi il *data miss rate*, calcolato rispetto alle istruzioni di load/store, è il 50 %.

Se il blocco fosse 8 volte più grande, in corrispondenza di una certa iterazione del loop, verrà caricato un blocco da 32 B in corrispondenza della lw. La sw successiva sarà un hit, e le istruzioni di lw/sw contenute nelle successive 7 iterazioni del loop sarebbero tutte hit. Quindi in media, per ogni $8 * 2 = 16$ di istruzioni di lw/sw, abbiamo 1 solo miss, da cui $\text{data miss rate} = 100 * 1/16 = 6.25 \%$.

3. **(solo Mod. II)**

La procedura è la seguente:

```
foo:
    addi    $sp, $sp, -4
    sw      $s0, 0($sp)
    addi    $t0, $a0, 0
    addi    $t1, $a1, 0

Loop:
    lw      $s0, 0($t0)
    addi    $s0, $s0, 100
    sw      $s0, 0($t0)
    addi    $t0, $t0, 4
    bne     $t0, $t1, Loop

    lw      $s0, 0($sp)
    addi    $sp, $sp, 4
    jr      $ra
```

- 3'. **(solo Arch. B)**

```
1.    lw      $s0, 0($t0)
2.    addi    $s0, $s0, 100
3.    sw      $s0, 0($t0)
4.    addi    $t0, $t0, 4
5.    bne     $t0, $t1, Loop
```

Le dipendenze rispetto a codice di sopra sono $1 \rightarrow 2$, $2 \rightarrow 3$, $4 \rightarrow 5$.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|------|----|----|----|------|------|----|------|------|----|----|------|------|----|----|----|
| lw | IF | ID | EX | ME | WB | | | | | | | | | | |
| addi | | IF | ID | <ID> | <ID> | EX | ME | WB | | | | | | | |
| sw | | | IF | <IF> | <IF> | ID | <ID> | <ID> | EX | ME | WB | | | | |
| addi | | | | | | IF | <IF> | <IF> | ID | EX | ME | WB | | | |
| bne | | | | | | | | | IF | ID | <ID> | <ID> | EX | ME | WB |

Al ciclo di clock 9, EX esegue sw, ID esegue il secondo addi, ID esegue bne, mentre ME e WB sono attraversati da una bolla.