Architettura degli Elaboratori (modulo II e B) (Compito 12 Gennaio 2011)

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

Esercizio 1

Siano date le due sequenze di accessi alla memoria:

A: 1, 134, 135, 65

B: 6, 214, 174, 175

Gli accessi riportati sono indirizzi alla "word". Quindi per ottenete gli indirizzi al Byte, essi devono essere moltiplicati per 4=2^2 (che in binario corrisponde ad aggiungere ....). La dimensione dell'indirizzo fisico finale è di 16 b.

1. Data una cache diretta, contenente solo 8 blocchi da 1 word ciascuno, identificare per ciascun accesso l'indirizzo binario, il Tag e l'Index.
2. Riportare per ogni accesso se questo si risolve in una hit o miss, supponendo che all'inizio di ciascuna sequenza la cache sia vuota.
3. Cosa cambierebbe se la cache diretta avesse solo 4 blocchi da 2 word?
4. Rispetto ad una memoria virtuale paginata, con indirizzo virtuale di 32 b (e fisico da 16 b, come definito in precedenza) e pagine da 1 KB, calcolare la dimensione di ciascun ingresso della Page Table, considerando che tale ingresso contiene anche i due bit aggiuntivi di Valid e Dirty. Qual è la dimension totale della Page Table?

Soluzione

A: 0001, 10000110, 10000111, 1000001

B: 1010, 11010110, 10101110, 10101111

Cache con 8 blocchi da 1 word:
INDEX=$\log 8 = 3$ b
OFFSET = $\log 4 = 2$ b
TAG = 32 -5 = 27 b

Nelle liste seguenti il carattere ':' separa l'index (parte bassa) dalla tag (parte alta). I 2 b di OFFSET (00) sono assenti nelle rappresentazioni binarie precedenti, in quanto gli indirizzi sono alla word.

A: 0:001 (miss compulsory), 10000:110 (miss compulsory), 10000:111 (miss compulsory), 1000:001 (miss compulsory con conflitto)

B: 1:010 (miss compulsory), 11010:110 (miss compulsory), 10101:110 (miss compulsory con conflitto) , 10101:111 (miss compulsory)

Cache con 4 blocchi da 2 word:
INDEX=$\log 4 = 2$ b
OFFSET = $\log 8 = 3$ b
TAG = 32 -5 = 27 b

Nelle liste seguenti il carattere ':' separa parte dell'Offset (parte bassa), dall'Index (parte intermedia) e dalla Tag (parte alta).

A: 0:00:1 (miss compulsory), 10000:11:0 (miss compulsory), 10000:11:1 (hit), 1000:00:1 (miss compulsory con conflitto)

B: 1:01:0 (miss compulsory), 11010:11:0 (miss compulsory), 10101:11:0 (miss compulsory con conflitto) , 10101:11:1 (hit)

Page Offset = $\log 2^{10} = 10$ b.
Virtual Page Number = 32 - 10 = 22 b.
Physical Page Number = 16 - 10 = 6 b.
Dimensione della PT entry = 6 + 2 = 8 b = 1 B.
Dimensione dell PT = $2^{22} * 1$ B = 4 MB.

Esercizio 2

Considerando che all'inizio `$s3` vale 2^{10}, mentre `$t0` vale 0, si calcoli il numero di volte che viene eseguito il loop seguente, individuando il tipo di calcolo eseguito (`$s0` è l'indirizzo di un array).

```
start:
  add  $t1, $s0, $t0
  lw   $t2, 0($t1)
  add  $t2, $t2, $t3
  sw   $t2, 0($t1)
  addi $t0, $t0, 4
  bne  $t0, $s3, start
end_loop:
  ...
```

Facendo riferimento all'architettura MIPS vista a lezione con pipeline a 5 stadi e considerando il forwarding e il delayed branch, quanti sono i cicli di clock totali per l'esecuzione del loop precedente se l'accesso alla memoria (ovvero alla cache) effettuato nello stadio MEM della pipeline costa sempre 1 ciclo di clock?
(Non si consideri a questo proposito la fase di startup della pipeline e non si ottimizzi il codice)

Qual è il CPI di questa porzione di codice?

Considerare infine la presenza di una cache ad accesso diretto con blocco di dimensione 8B e dimensione dei dati totale di 8KB. Calcolare il numero di cache miss nel caso in cui nessuna delle word accedute sia presente in cache all'inizio dell'esecuzione del loop.

Se si considera che nel caso in cui abbiamo un miss in cache lo stadio MEM necessita di 4 cicli di clock per il completamento, quanti sono i cicli totali impiegati per l'esecuzione del loop? Qual è il nuovo CPI?

Soluzione

Il loop viene quindi eseguito per $2^{10}/4 = 256$ volte. Il registro `$s0` contiene quindi l'indirizzo di una array di 256 interi (word). Il programma accede tutti gli interi dell'array incrementandoli del valore di `$t3`.

Considerando forwarding e delayed branch, e lo stadio MEM che completa in un ciclo di clock, abbiamo un'istruzione di `nop` dopo la `beq`, e uno stallo (esemplificato qui da una `nop`) dopo la `lw`. Quindi 8 cicli a regime per eseguire 6 istruzioni:

```
start:
   add  $t1, $s0, $t0
   lw   $t2, 0($t1)
   nop
   add  $t2, $t2, $t3
   sw   $t2, 0($t1)
   addi $t0, $t0, 4
   bne  $t0, $s3, start
   nop
end_loop:
   ...
```

Quindi abbiamo un numero di cicli di clock totali uguali a $8 * 256 = 2048$.
Il CPI è uguale a $\frac{8\ cicli}{6\ istr} = 1.33$.

Infine, se l'accesso alla memoria in presenza di miss costa 4 cicli (3 cicli in più dell'hit), abbiamo che la `sw` completa sempre in un ciclo (hit), mentre la `lw` può dar luogo a miss.
Il diagramma temporale in caso di miss è il seguente:

```
lw    | IF | ID  | EXE | MEM |<MEM>|<MEM>|<MEM>|WB  |
add        | IF  | ID  |<ID> |<ID> |<ID> |<ID> | EXE | MEM | WB  |
sw               | IF  |<IF> |<IF> |<IF> |<IF> | ID  | EXE | MEM | WB  |
```

Il diagramma temporale in caso di hit è il seguente:

```
lw    | IF | ID  | EXE | MEM | WB  |
add        | IF  | ID  |<ID> | EXE | MEM | WB  |
sw               | IF  |<IF> | ID  | EXE | MEM | WB  |
```

Si noti che la fase di EXE della `add` non può iniziare prima del completamento della fase di MEM (4 cicli) della `lw` precedente.
Si consideri però che la situazione di sopra dovuta al cache miss si verifica solo quando viene letto la prima word di ogni blocco della cache, la cui lunghezza è 8 B. Quando si accede invece la seconda word, si verifica un hit. Siccome il codice accede in totale 256 word, il numero totale di cicli diventa: 8 * 128 + (8+3) * 128 = 2432 cicli.
Il CPI è uguale a $\frac{2432\ cicli}{256*6\ istr} = 1.58$.