

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

## Esercizio 1

Considerare una TLB 4-way associative. La dimensione della TAG della TLB è 14 b, la TLB ha 512 entry, mentre  $Page\ size = 2\ KB$ . Calcolare l'indirizzo virtuale.

Se il numero di pagina fisica è grande 13 b, stabilire la dimensione dell'indirizzo fisico.

Considerare una cache 2 way set associative, con dimensione del blocco di 16 B, INDEX = 8 b. Calcolare la dimensione della cache (parte dati).

Considerare la seguente lista di accessi alla cache (sequenza di ind. fisici espressi in esadecimale) i cui ingressi sono all'inizio tutti non validi. Quali degli indirizzi provoca miss e hit? Quali miss provocano un conflitto, e quindi richiedono un rimpiazzamento di un ingresso della cache?

```
123 410
123 424
124 414
123 418
125 410
```

## Soluzione

Il numero di set della TLB è  $512 / 4\ way = 2^9 / 2^2 = 2^7$ .

L'INDEX della TLB è  $\log 2^7 = 7\ b$ .

L'OFFSET è dato da  $\log page\_size = \log 2^{11} = 11\ b$ .

Dim. ind. virtuale: TAG + INDEX + OFFSET =  $14 + 7 + 11 = 32\ b$ .

Dim. ind. fisico: Num\_Pagina\_Fisica + OFFSET =  $13 + 11 = 24\ b$ .

Dim. cache =  $dim\_blocco * 2^8 * 2\ way = 2^4 * 2^8 * 2 = 2^{13}\ B = 8\ KB$ .

```
TAG=123  INDEX=41  OFFSET=0      : miss
TAG=123  INDEX=42  OFFSET=4      : miss
TAG=124  INDEX=41  OFFSET=4      : miss (conflitto sul set)
TAG=123  INDEX=41  OFFSET=8      : hit
TAG=125  INDEX=41  OFFSET=0      : miss (conflitto)
```

L'ultimo accesso provoca il rimpiazzo di uno dei due blocchi presenti in cache nel set identificato da INDEX=41.

## Esercizio 2

Si consideri il seguente programma assembly MIPS:

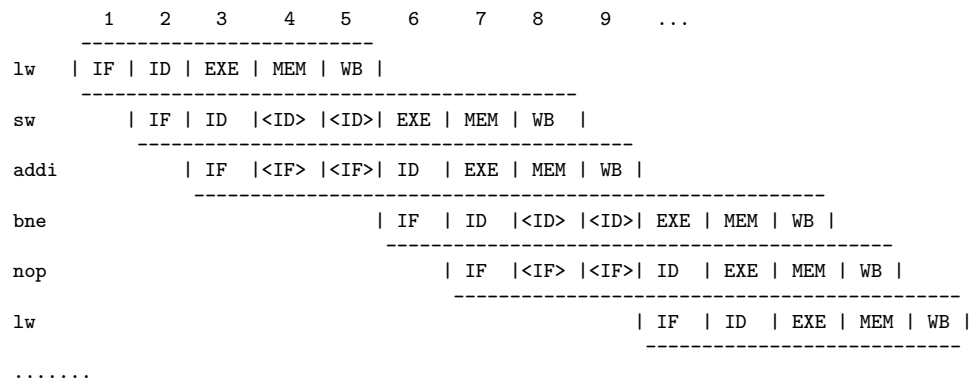
```
loop: lw  $3, a($4)
      sw  $3, b($4)
      addi $4, $4, 4
      bne $4, $6, loop
```

dove \$4 è inizializzato a 0 e \$6 a 10.000. Le costanti a e b sono rappresentabili su 16 bit.

1. Calcolare il CPI per l'architettura pipeline a 5 stadi, senza tecniche di forwarding, ma con il register file speciale che scrive nella prima parte del ciclo e legge nella seconda parte dello stesso ciclo. Supporre inoltre che le istruzioni di salto condizionato siano gestiti con la tecnica del *delay branch* (delay slot = 1 ciclo).
2. Calcolare il miglioramento della performance (speed-up) nel caso in cui vengano introdotte tecniche di forwarding.

## Soluzione

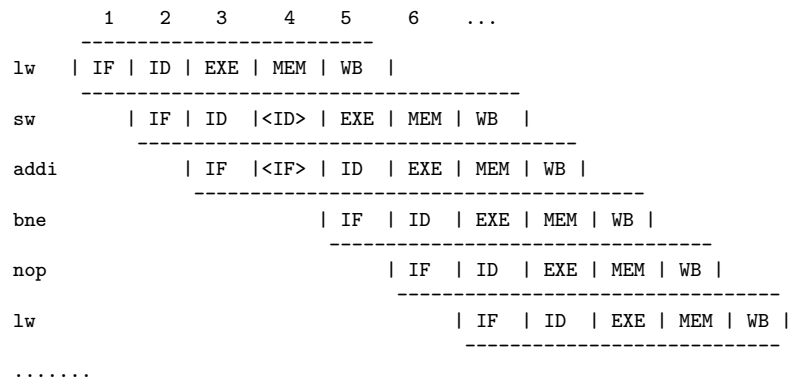
Abbiamo una dipendenza tra `lw` e `sw` a causa del registro \$3. Inoltre abbiamo una dipendenza tra la `addi` e `bne` a causa del registro \$4. Otteniamo quindi la seguente esecuzione:



Ogni iterazione loop richiede quindi 9 cicli di clock (più i cicli finali per svuotare la pipeline, che sono irrilevanti per questo calcolo, in quanto trattasi di una piccola costante).

Poiché vengono eseguiti  $10000 / 4 = 2500$  iterazioni del loop (ad ogni iterazione \$6 viene decrementato fino a diventare zero), otteniamo un numero totale di cicli di  $9 * 2500 = 22500$ . Poiché le istruzioni sono  $IC = 4 * 2500 = 10000$  otteniamo  $CPI = \text{cicli}/IC = 22500 / 10000 = 2,25$ .

Con il forwarding abbiamo il seguente diagramma di esecuzione:



Ogni iterazione loop richiede quindi 6 cicli di clock (più i cicli finali per svuotare la pipeline, che sono irrilevanti per questo calcolo, in quanto trattasi di una piccola costante).

Poiché vengono eseguiti  $10000 / 4 = 2500$  iterazioni del loop, otteniamo un numero totale di cicli di  $6 * 2500 = 15000$ .

Poiché le istruzioni sono  $IC = 4 * 2500 = 10000$  otteniamo  $CPI' = \text{cicli}/IC = 15000 / 10000 = 1,5$ .

Per calcolare lo Speedup, poiché IC e la Frequenza rimangono inalterati, basta considerare il rapporto tra i CPI:

$$Speedup = 2,25/1,5 = 1,5.$$