

**ARCHITETTURE AVANZATE DEI CALCOLATORI A.A. 2006/07**  
**Prima prova in itinere del 30/10/2006**  
**Compito A**

**Domanda 1**

Cosa è il superpipelining? Quali sono i suoi vantaggi e svantaggi?

**Domanda 2**

Definire la criticità sui dati di tipo WAW e mostrare un esempio. Spiegare perché non si verifica nella pipeline del processore MIPS. Spiegare come questo tipo di criticità viene gestito nei processori superscalari con scheduling dinamico mediante:

- A) Scoreboard
- B) Algoritmo di Tomasulo

**Domanda 3**

Spiegare la predizione statica di tipo Branch Always Taken; quando conviene applicarla?

Spiegare la predizione dinamica con 2-bit BHT; quale vantaggio presenta rispetto alla 1-bit-BHT?

Si supponga che l'esecuzione di un salto condizionale abbia il seguente risultato (N: not taken, T: taken):

N N N T T T N N N T T T

Quale è l'accuratezza del predittore statico Branch Always Taken?

Quale è l'accuratezza del predittore dinamico 2-bit BHT, assumendo che sia inizializzato a weakly predict not taken (01)?

**Esercizio 1**

Si consideri il seguente frammento di codice in linguaggio C, dove A e B sono due vettori di interi:

```
for (i=0; i<500; i++)  
    A[i] = A[i] - B[i]/2;
```

Il sorgente sia stato compilato nel seguente codice in assembler MIPS. Si assuma che \$t0 sia inizializzato a 0 e che \$s1 contenga 500. I simboli BASEA e BASEB sono costanti a 16 bit, prefissate.

```
Loop: lw $t1, BASEA($s0)  
      lw $t2, BASEB($s0)  
      srl $t2, $t2, 1  
      sub $t2, $t1, $t2  
      sw $t2, BASEA($s0)  
      addi $t0, $t0, 1  
      addi $s0, $s0, 4  
      bne $t0, $s1, Loop
```

- a) Schedulare in modo efficiente il ciclo sul processore MIPS con pipeline statica two-issue.
  - Calcolare il valore del CPI ottenuto e confrontarlo con quello ideale.
- b) Effettuare il loop unrolling (con un fattore di srotolamento pari a 2) e riordinare il codice sul processore MIPS scalare con pipeline ottimizzata allo scopo di massimizzare le prestazioni.
  - Calcolare il valore del CPI per una generica iterazione del ciclo originale prima di effettuare il loop unrolling.
  - Calcolare il valore del CPI per una generica iterazione del ciclo modificato dopo aver effettuato il loop unrolling ed il riordinamento.
  - Calcolare il numero di cicli di clock necessari ad eseguire l'intero ciclo prima di effettuare il loop unrolling (trascurare i cicli di clock necessari a riempire la pipeline).
  - Calcolare il numero di cicli di clock necessari ad eseguire l'intero ciclo dopo aver effettuato il loop unrolling ed il riordinamento (trascurare i cicli di clock necessari a riempire la pipeline).
- c) (*facoltativo*) Conviene srotolare il ciclo con un fattore di srotolamento maggiore di 2? Perché?

**Esercizio 2**

Sia dato il seguente ciclo di un programma in assembler MIPS. Si assuma che i registri \$t6 e \$t7 siano stati inizializzati rispettivamente ai valori 0 e 4N. I simboli BASEA, BASEB, BASEC, BASED, BASEE e BASEF sono costanti a 16 bit, prefissate. Il ciclo di clock del processore vale 1 ns.

```
Loop: lw $t2, BASEA($t6)  
      sw $t2, BASEB($t6)  
      lw $t3, BASEC($t6)  
      sw $t3, BASED($t6)  
      lw $t4, BASEE($t6)  
      add $t4, $t4, $t2  
      add $t4, $t4, $t3  
      sw $t4, BASEF($t6)  
      addi $t6, $t6, 4
```

bne \$t6, \$t7, Loop

Si supponga che il ciclo venga iterato N volte (con N grande a piacere) tramite una pipeline MIPS a 5 stadi. Si consideri una generica iterazione del ciclo.

- a) Si supponga che la pipeline sia priva di ottimizzazioni.
- Disegnare il diagramma a ciclo multiplo della pipeline ed individuare le criticità sui dati e sul controllo.
  - Inserire gli stalli necessari a risolvere le criticità individuate.
  - Calcolare il numero totale di stalli inseriti, il CPI asintotico ed il throughput espresso in MIPS.
- b) Si supponga che nella pipeline siano state introdotte le seguenti ottimizzazioni:
- nel banco dei registri è possibile la lettura e la scrittura nello stesso ciclo di clock;
  - forwarding dei dati;
  - anticipazione del salto nello stadio ID.
- Disegnare il diagramma a ciclo multiplo della pipeline ed individuare le criticità sui dati e sul controllo rimaste; indicare i percorsi di forwarding utilizzati.
  - Inserire gli stalli necessari a risolvere le criticità rimaste.
  - Calcolare il numero totale di stalli inseriti, il CPI asintotico ed il throughput espresso in MIPS.

### Esercizio 3

Si consideri la sequenza di istruzioni

```
addi $3, $3, -1
bne $3, $0, L1
```

Si individui la criticità sui dati presente. Si descriva se e come tale criticità viene risolta dal processore MIPS considerato a lezione con pipeline ottimizzata, il cui schema è riportato in Figura 1. Se la criticità viene risolta, si mostri come sono utilizzate dal processore MIPS le condizioni per il rilevamento e la gestione delle criticità esaminate a lezione. Se la criticità non viene risolta, si determinino le opportune condizioni per il suo rilevamento e la sua gestione e si mostri l'eventuale hardware da aggiungere ed il percorso di forwarding usato.

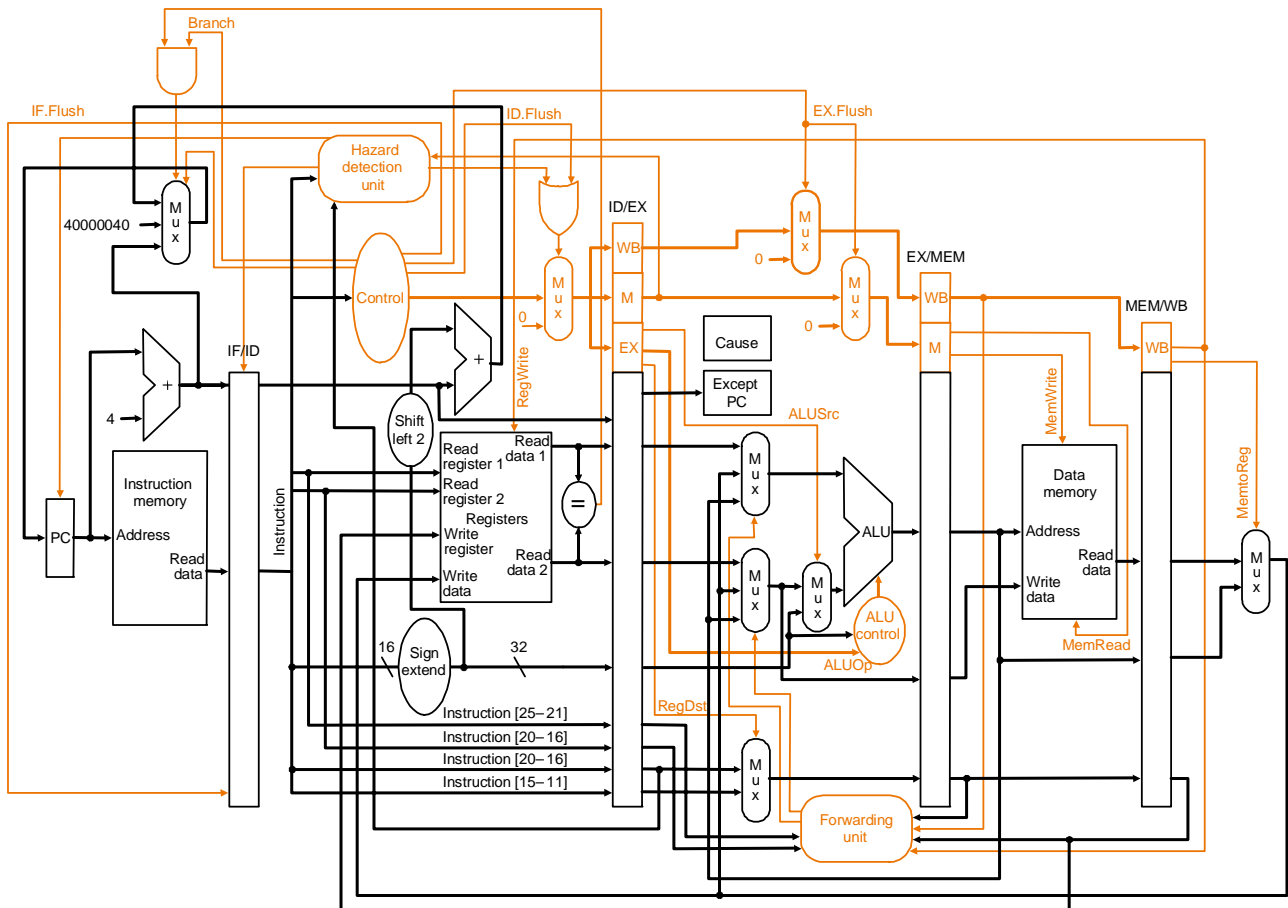


Figura 1 Unità di elaborazione e controllo del processore MIPS con pipeline