

Esercitazione su pipeline semplice

Esercitazione

Calcolatori Elettronici

Esercizio

ⁿ Valutare il numero di cicli di clock necessari a completare l'esecuzione di questo programma nel caso di pipeline semplice (si ipotizzi che il ciclo si esegue 100 volte).

- Si supponga di essere nel caso ideale di assenza di fallimenti in cache
- Si adotti la tecnica basata sulla predizione di fallimento per la gestione delle criticità sul controllo
- Si adotti la tecnica di propagazione per la risoluzione delle criticità sui dati
- Si supponga di non poter effettuare il riordinamento del codice nel caso di criticità di tipo carica-e-usa

```
add $s1,$zero,$zero  
addi $t4,$zero,100  
add $t2,$s0,$s0  
loop: slt $t0,$t2,$t4  
beq $t0,$zero,exit  
lw $t3,0($t2)  
add $s1,$s1,$t3  
add $s2,$s2,$t3  
addi $t2,$t2,4  
j loop  
exit: sw $s1,0($s2)
```

ⁿ N.B. Evidenziare tutti gli anticipi e gli stalli necessari, specificando il numero di volte per cui si ripetono

Calcolatori Elettronici - Esercitazione sulla pipeline - Slide 4

L. Tarantino - a.a. 2004/2005

```

(1) add $s1,$zero,$zero
(2) addi $t4,$zero,100
(3) add $t2,$t2,$s0
(4) Loop: sll $t0,$t2,$t4
(5) beq $t0,$zero,exit
(6) lw $t3,0($t2)
(7) add $s1,$s1,$t3
(8) add $s2,$s2,$t3
(9) addi $t2,$t2,4
(10) j Loop
(11) exit: sw $s1,0($s2)
    
```

La beq non esegue il salto per 100 volte (la lw viene prelevata correttamente, niente ritardo). La 101-esima volta il salto si esegue: 2 cicli di ritardo 1 sola volta

J si comporta come beq nel caso di salto eseguito: 2 cicli di ritardo per 100 volte

Soluzione (2)

Criticità sul controllo

Calcolatori Elettronici - Esercitazione sulla pipeline - Slide 3

```

(1) add $s1,$zero,$zero
(2) addi $t4,$zero,100
(3) add $t2,$t2,$s0
(4) Loop: sll $t0,$t2,$t4
(5) beq $t0,$zero,exit
(6) lw $t3,0($t2)
(7) add $s1,$s1,$t3
(8) add $s2,$s2,$t3
(9) addi $t2,$t2,4
(10) j Loop
(11) exit: sw $s1,0($s2)
    
```

antecipazione (1 volta, solo alla prima iterazione)

antecipazione (101 volte)

stallo + anticipo (100 volte)

osservazioni:

- La (9) sembra avere criticità con la (4). In realtà, per effetto dei ritardi dovuti al salto della j, la sll trova nel banco il valore di \$t2 calcolato dalla (9)
- La (8) non entra in criticità con la (6) sul registro \$t3 perché, per effetto dello stallo introdotto, il ciclo ID della (8) si trova in corrispondenza col ciclo WB della (6) - vedi anche la situazione della pipeline nell'ultima slide

Soluzione (1)

Criticità sui dati

Ogni istruzione può avere criticità con le due successive

Calcolatori Elettronici - Esercitazione sulla pipeline - Slide 6

L. Trantino - a.a. 2004/2005

Soluzione (4)

Effetti sulla pipeline

```

(1) add $s1,$zero,$zero
(2) add $t4,$zero,100
(3) add $t2,$t2,$s0
(4) Loop: sll $t0,$t2,$t4
(5) beq $t0,$zero,exit
(6) lw $t3,0($t2)
(7) add $s1,$s1,$t3
(8) add $s2,$s2,$t3
(9) addi $t2,$t2,4
(10) j Loop
(11) exit: sw $s1,0($s2)
    
```

Quando il salto si esegue, 2 ritardi

Calcolatori Elettronici - Esercitazione sulla pipeline - Slide 5

L. Trantino - a.a. 2004/2005

Soluzione (3)

(1)	1
(2)	1
(3)	1
(4)	101
(5)	101
(6)	100
(7)	100
(8)	100
(9)	100
(10)	100
(11)	1

N_i

$N^{cicl} = CI + Num-ritardi =$

$CI + ritardi-critica-dati + ritardi-critica-controllo =$

$706 + 100 + 2 + 2 \cdot 100 = 1008$