

Calcolatori Elettronici 0.5 - Appello del 3 giugno 2003
L. Tarantino - a.a. 2002/2003

(1) add \$s0,\$zero,\$zero
(2) add \$t1,\$zero,100
(3) loop: slt \$t2,\$s0,\$t1
(4) beq \$t2,\$zero,exit
(5) addi \$t4,\$t4,4
(6) lw \$t3,0(\$t4)
(7) add \$s1,\$s1,\$t3
(8) addi \$s0,\$s0,1
(9) j loop
(10) exit: sw \$s1,0(\$s4)

Soluzione (1)

Criticità sui dati

Ogni istruzione può avere criticità con le due successive

antipico (1 volta, solo alla prima iterazione)

antipico (101 volte)

stallo + antipico (100 volte)

Osservazioni:
- La (8) sembra avere criticità con la (3). In realtà, per effetto dei ritardi dovuti al salto della j, la slt trova nel banco il valore di \$s0 calcolato dalla (8)

Calcolatori Elettronici 0.5 - Appello del 3 giugno 2003
L. Tarantino - a.a. 2002/2003

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 (perché è necessaria questa precisazione?)
- 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 (SPIEGARLA)
- Si supponga di non poter effettuare il riordinamento del codice nel caso di criticità di tipo carica-e-usa

Esercizio sulle prestazioni con pipelining

add \$s0,\$zero,\$zero
add \$t1,\$zero,100
loop: slt \$t2,\$s0,\$t1
beq \$t2,\$zero,exit
addi \$t4,\$t4,4
lw \$t3,0(\$t4)
add \$s1,\$s1,\$t3
addi \$s0,\$s0,1
j loop
exit: sw \$s1,0(\$s4)

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

Calcolatori Elettronici 0.5 - Appello del 3 giugno 2003
L. Tarantino - a.a. 2002/2003

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

$N_{cicl} = CI + Num-ritardi =$

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

$705 + 100 + 2 + 2 \times 100 = 1007$

Soluzione (3)

Calcolatori Elettronici 0.5 - Appello del 3 giugno 2003
L. Tarantino - a.a. 2002/2003

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

(1) add \$0,\$zero,\$zero

(2) add \$t1,\$zero,100

(3) loop: sit \$t2,\$s0,\$t1

(4) beq \$t2,\$zero,exit

(5) add \$t4,\$t4,4

(6) lw \$t3,0(\$t4)

(7) add \$s1,\$s1,\$t3

(8) add \$s0,\$s0,1

(9) j loop

(10) exit: sw \$s1,0(\$s4)

Soluzione (2)

n Criticità sul controllo

Soluzione (4)

ⁿ Effetti sulla pipeline

