ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ
ΕΡΓΑΣΤΗΡΙΟ ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ
www.cslab.ece.ntua.gr

19 Απριλίου 2005

ΠΡΟΗΓΜΕΝΑ ΘΕΜΑΤΑ ΟΡΓΑΝΩΣΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

1η Σειρά Ασκήσεων (παράδοση μέχρι 31/5/05)

Θέλουμε να διερευνήσουμε την αποδοτικότητα διαφορετικών μεθόδων πρόβλεψης διακλάδωσης, σχετικά με την επίδοση προγραμμάτων που εκτελούνται σε μία αρχιτεκτονική αγωγού. Στο εξής θεωρούμε ότι διαθέτουμε μία σωλήνωση 5 σταδίων, στην οποία η απόφαση διακλάδωσης πραγματοποιείται στον $2^{ο}$ κύκλο του αγωγού (ID) όταν πρόκειται για εντολή διακλάδωσης χωρίς συνθήκη (`j`, κλπ) και στον $3^{ο}$ κύκλο του αγωγού (EX) όταν πρόκειται για εντολή διακλάδωσης υπό συνθήκη (`beq`, `bne`, κλπ). Θεωρούμε, επίσης ότι υπάρχουν όλα τα δυνατά σχήματα προώθησης.

α) Υποθέτουμε ότι δεν υπάρχει κανενός είδους πρόβλεψη διακλάδωσης (πρέπει να επιλυθεί πρώτα το άλμα πριν συνεχίσει η εκτέλεση της επόμενης εντολής). Πόση καθυστέρηση (σε κύκλους ρολογιού) εισάγουν οι εντολές διακλάδωσης χωρίς συνθήκη και πόση οι εντολές διακλάδωσης υπό συνθήκη;

β) Έστω το παρακάτω τμήμα του κώδικα (ρουτίνα bubblesort):

Θεωρούμε ότι η διεύθυνση του προς ταξινόμηση πίνακα `array` είναι η `100` και το μήκος του πίνακα είναι `Length`. Η τιμή της μεταβλητής `4·i` είναι αρχικά ίση με `Length` και βρίσκεται στον καταχωρητή `$r3`.

AO-AE-Label	**Assembly**	**Σχόλια**
1-1 loop1:	slt $t1, $zero, $r3	#while (i > 0) {
2	beq $t1, $zero, end	
2-3	addi $r5, $zero, 0	# j = 0;
3-4 loop2:	slt $t2, $r5, $r3	# while (j < i) {
5	beq $t2, $zero, cont	
4-6	lw $r6, 100($r5)	# temp = array[j];
7	lw $r7, 104($r5)	# temp2 = array[j+1];
8	slt $t3, $r7, $r6	# if (temp > temp2) {
9	beq $t3, $zero, skip	
5-10	sw $r7, 100($r5)	# array[j] = temp2;
11	sw $r6, 104($r5)	# array[j+1] = temp;
		}
6-12 skip:	addi $r5, $r5, 4	# j++;
13	j loop2	# }
7-14 cont:	addi $r3, $r3, -4	# i--;
15	j loop1	#}
8-16 end:	hlt	

Ομαδοποιούμε τις εντολές σύμφωνα με την αρίθμηση της στήλης ΑΟ (αριθμός ομάδας). Για κάποιον συγκεκριμένο πίνακα `array`, η σειρά εκτέλεσης των ομάδων εντολών είναι η ακόλουθη:
1 2 3 4 5 6 3 4 6 3 4 6 3 7 1 2 3 4 5 6 3 4 6 3 7 1 2 3 4 5 6 3 7 1 2 3 7 1 8

Συμπληρώστε τον ακόλουθο πίνακα με ***N*** όταν δεν πραγματοποιείται άλμα (branch Not taken) και με ***T*** στην περίπτωση που πραγματοποιείται άλμα (branch Taken).

ΑΕ	**Διαδοχικές εκτελέσεις εντολών άλματος**									
	1η	2η	3η	4η	5η	6η	7η	8η	9η	10η
2	***N***									
5										
9										
13										
15										

Με βάση τον προηγούμενο πίνακα, συμπληρώστε τον πίνακα που ακολουθεί:

ΑΕ	**Αρ.επαναληπτικών Εκτελέσεων**	**Πραγματοποίηση Άλματος (Taken)**	**Όχι Άλμα (Not taken)**	**% Taken**	**% Not taken**
2					
5					
9					
13					
15					

γ) Δείξτε το διάγραμμα εκτέλεσης των επιμέρους φάσεων για κάθε εντολή (κατά την πρώτη σειρά επαναλήψεων: ***1 2 3 4 5 6*** και ***3 7 1 2***). Υποθέστε ότι υπάρχουν όλα τα δυνατά σχήματα προώθησης (εκτός από τις περιπτώσεις εντολών άλματος), και σχεδιάστε τις θέσεις όπου χρειάζεται να πραγματοποιηθεί προώθηση δεδομένων μεταξύ εντολών. Πόσους κύκλους χρειάζεται για να ολοκληρωθεί η εκτέλεση της ρουτίνας; Πόσοι από τους κύκλους αυτούς σπαταλούνται λόγω εξαρτήσεων δεδομένων και πόσοι λόγω εντολών άλματος (με και χωρίς συνθήκη);

δ) Θεωρείστε μία στατική τεχνική πρόβλεψης διακλάδωσης κατά την οποία ο μεταγλωττιστής υποθέτει πάντοτε ότι οι εντολές άλματος είναι " not taken ".
Ακολουθώντας την ίδια σειρά εκτέλεσης με το ερώτημα (β), πόσους κύκλους καθυστέρησης γλιτώνουμε;

ε) Θεωρείστε εναλλακτικά τη στατική τεχνική πρόβλεψης διακλάδωσης κατά την οποία ο μεταγλωττιστής υποθέτει πάντοτε ότι οι εντολές άλματος είναι "taken".
Ακολουθώντας την ίδια σειρά εκτέλεσης με το ερώτημα (β), πόσους κύκλους καθυστέρησης γλιτώνουμε;

στ) Θεωρείστε μία τεχνική πρόβλεψης διακλάδωσης κατά την οποία ο μεταγλωττιστής «σημαδεύει» κάθε εντολή άλματος ως "taken" ή "not taken" βάσει μίας υποθετικής εκτέλεσης της ρουτίνας, και σύμφωνα με τις στατιστικές που προκύπτουν από αυτήν. Αν, για παράδειγμα, κατά την θεωρητική εκτέλεση η εντολή #2 βρεθεί ότι στο 65% των περιπτώσεων είναι " not taken", θεωρείται καθ' όλη τη διάρκεια της πραγματικής εκτέλεσης ότι μετά την εντολή δεν θα πραγματοποιηθεί άλμα, οπότε συνεχίζει η εκτέλεση των εντολών που την ακολουθούν, πριν ακόμα ολοκληρωθεί η εκτέλεση της εντολή άλματος.
Σύμφωνα με τις στατιστικές του ερωτήματος (β) «σημαδέψτε» κάθε μία από τις 5 εντολές άλματος ως "taken" ή "not taken".
Ακολουθώντας την ίδια σειρά εκτέλεσης με το ερώτημα (β), πόσους κύκλους καθυστέρησης γλιτώνουμε;