



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

ΠΡΟΗΓΜΕΝΑ ΘΕΜΑΤΑ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ ΥΠΟΛΟΓΙΣΤΩΝ
Εξετάσεις Οκτωβρίου 2011
Διάρκεια 2:45' ώρες

Οι εξετάσεις θα πραγματοποιηθούν ΧΩΡΙΣ την παρουσία βιβλίων, βοηθημάτων ή άλλου είδους σημειώσεων. Το μόνο που επιτρέπεται να έχετε μαζί σας είναι ένα φύλλο Α4 στο οποίο μπορείτε να έχετε γράψει ό,τι έχετε κρίνει πιο σημαντικό για το μάθημα και θέλετε να το έχετε ως βοήθημά σας. Απαγορεύεται η ανταλλαγή οποιουδήποτε αντικειμένου κατά την ώρα της εξέτασης, ούτε και των φύλλων Α4 που είναι ατομικά.

Θέμα 1ο (15%)

A. Θεωρήστε ένα loop το οποίο εκτελείται αρκετές φορές σε ένα πρόγραμμα. Κάθε εκτέλεση του loop περιλαμβάνει 8 επαναλήψεις. Κάθε επανάληψη περιλαμβάνει 4 εντολές άλματος υπό συνθήκη, τα αποτελέσματα των οποίων φαίνονται στον παρακάτω πίνακα.

	Επανάληψη							
	1	2	3	4	5	6	7	8
Branch 1	N	T	T	N	N	T	T	N
Branch 2	T	T	T	T	T	N	N	N
Branch 3	T	N	N	T	T	T	T	N
Branch 4	T	T	T	T	T	T	T	N

Στην 8^η επανάληψη το Branch 4 είναι NT και η εκτέλεση φεύγει από το loop. Υποθέστε ότι δεν υπάρχουν άλλες εντολές άλματος υπό συνθήκη στον κώδικα, καθώς και ότι το global branch prediction history είναι όλο NT στην αρχή κάθε επανάληψης του loop. Υποθέστε ότι τα predictor tables έχουν απεριόριστη χωρητικότητα και ότι το loop εκτελείται αρκετές φορές ώστε να μη μας νοιάζει η αρχική κατάσταση των predictors.

Για κάθε μία από τις παρακάτω εντολές διακλάδωσης, διαλέξτε τον dynamic predictor με το καλύτερο ποσοστό πρόβλεψης, εξηγήστε γιατί αυτός δουλεύει καλά για τη συγκεκριμένη εντολή και παρουσιάστε την κατάσταση του στο τέλος της 1000ής επανάληψης του loop. Σε περίπτωση που παραπάνω από ένας predictors δίνουν το καλύτερο αποτέλεσμα, διαλέξτε το φθηνότερο. Δικαιολογήστε τις επιλογές σας.

- (i) Branch 1
- (ii) Branch 2
- (iii) Branch 3

B. Σε μια αρχιτεκτονική που υλοποιεί τον αλγόριθμο Tomasulo, τόσο στα reservation stations όσο και στον Reorder Buffer υπάρχουν τα πεδία "V" για την αποθήκευση τιμών που προκύπτουν από διάφορες εντολές. Γιατί χρειάζεται να υπάρχει το πεδίο αυτό και στα reservation stations, εφόσον υπάρχει ήδη στον ROB;

Γ. Θεωρήστε έναν out-of-order, superscalar επεξεργαστή που υποστηρίζει μετονομασία καταχωρητών και μηχανισμούς πρόβλεψης διακλάδωσης. Υποθέτουμε επιπλέον τις εξής τροποποιήσεις:

- χρήση μεταγλωττιστή με περισσότερες βελτιστοποιήσεις
- αύξηση φυσικών καταχωρητών
- αύξηση entries στον Branch Target Buffer

Για κάθε μία από τις τροποποιήσεις αυτές, αναφέρατε πώς θα μεταβληθούν τα ακόλουθα μεγέθη (αν θα υπάρξει δηλαδή αύξηση, μείωση ή καμία μεταβολή). Δικαιολογήστε τις απαντήσεις σας.

- (i) Instructions/Program
- (ii) Cycles/Instruction

Θέμα 2^ο (25%)

A. Υποθέστε ένα σύστημα παράλληλης επεξεργασίας που χρησιμοποιεί ένα write-invalidate cache coherence protocol, καθώς και την παρακάτω σειρά προσβάσεων στη μνήμη από 2 επεξεργαστές. Τα R_x και W_x περιγράφουν ανάγνωση και εγγραφή αντίστοιχα της μεταβλητής X. Δίνεται επίσης ότι οι μεταβλητές A, B ανήκουν στο ίδιο cache block. Για αυτή τη σειρά προσβάσεων βρείτε τον αριθμό των hits και misses του κάθε επεξεργαστή. Υποδείξτε ποια misses είναι compulsory, true sharing ή false sharing misses. Αρχικά οι caches των επεξεργαστών είναι άδειες.

P1	P2
R_A	
	R_B
W_A	R_B
	R_A
	W_B
R_A	

B. Ποια η διαφορά μεταξύ των invalidate και των update snooping πρωτοκόλλων; Περιγράψτε ένα σενάριο όπου το κάθε είδος έχει χαμηλή απόδοση και εξηγήστε γιατί.

Γ. Σε ένα σύστημα παράλληλης επεξεργασίας με μοιραζόμενη μνήμη, οι 2 επεξεργαστές εκτελούν τον παρακάτω κώδικα:

Processor A	Processor B
A1: Load R, (X)	B1: Store (X), 2
A2: Store (Y), 11	B2: Load R, (X)
A3: R = R+1	B3: Store (Y), R
A4: Store (X), R	

Ο R είναι καταχωρητής (ο κάθε επεξεργαστής έχει τον δικό του) ενώ οι X, Y είναι διευθύνσεις στη μνήμη, των οποίων η αρχική τιμή είναι μηδέν ($M[X] = M[Y] = 0$). Αν υποτεθεί ότι το σύστημα είναι ακολουθιακά συνεπές, ποια από τα παρακάτω ζεύγη τελικών τιμών είναι σωστά και ποια λάθος; Δικαιολογήστε την απάντησή σας.

- (i) $(X, Y) = (1, 1)$ (ii) $(X, Y) = (1, 3)$ (iii) $(X, Y) = (1, 11)$
- (iv) $(X, Y) = (2, 1)$ (v) $(X, Y) = (2, 3)$ (vi) $(X, Y) = (3, 2)$

Θέμα 3ο (30%)

Δίνεται αρχιτεκτονική η οποία υλοποιεί τον αλγόριθμο Tomasulo χρησιμοποιώντας ROB. Το pipeline του επεξεργαστή περιέχει τα στάδια IS, EX, WR και CMT, αγνοούμε δηλαδή τα IF και ID.

- Τα IS, WR, CMT απαιτούν 1 κύκλο ρολογιού.
- Το σύστημα διαθέτει *περιορισμένο* αριθμό reservation stations (RS). Διαθέτει 1 RS για προσθήσεις/αφαιρέσεις και 1 RS για πολλαπλασιασμούς/διαιρέσεις Floating Point αριθμών. Για εντολές Integer αριθμών διαθέτει 4 RS.
- Το σύστημα περιλαμβάνει 1 non-pipelined FU για πράξεις Integer αριθμών. Όλες οι εντολές μεταξύ Integer αριθμών διαρκούν 1 κύκλο.
- Το σύστημα περιλαμβάνει 2 non-pipelined Floating Point FUs, ένα για ADDD/SUBD και ένα για MULD/DIVD. Οι εντολές πρόσθεσης/αφαίρεσης διαρκούν 3 κύκλους, ενώ οι εντολές πολλαπλασιασμού/διαίρεσης 6 κύκλους.
- Οι εντολές αναφοράς στη μνήμη πραγματοποιούν στο στάδιο EX τόσο τον υπολογισμό της διεύθυνσης αναφοράς όσο και την προσπέλαση στη μνήμη. Το σύστημα διαθέτει ένα Load και ένα Store Queue, 2 θέσεων το καθένα. Οι εντολές χρησιμοποιούν ένα ξεχωριστό pipelined FU για τον υπολογισμό της διεύθυνσης και διαρκούν 2 ή 5 κύκλους σε περίπτωση Hit ή Miss στην cache.
- Οι εντολές διακλάδωσης υπό συνθήκη χρησιμοποιούν τα Integer RS και FU προκειμένου να υπολογίσουν αν ισχύει η συνθήκη. Η πρόβλεψη για μια εντολή διακλάδωσης υπό συνθήκη γίνεται *ταυτόχρονα* με τη δρομολόγηση της εντολής. Ο έλεγχος της πρόβλεψης γίνεται μόλις γίνει γνωστό το αποτέλεσμα της εντολής (στο στάδιο WR). Σε περίπτωση σφάλματος, σταματά η εκτέλεση των εντολών του miss-predicted execution path και στον επόμενο κύκλο δρομολογείται η σωστή εντολή.
- Ο ROB έχει 8 θέσεις.
- Το σύστημα περιλαμβάνει 1 CDB. Όταν παραπάνω από μια εντολές θέλουν να το χρησιμοποιήσουν, προτεραιότητα αποκτά η “*παλαιότερη*” εντολή (αυτή που έγινε issued πρώτη). Θεωρήστε ότι τα branches δεν χρησιμοποιούν το CDB κατά τη διάρκεια του WR σταδίου τους.
- Για τις εντολές διακλάδωσης υπό συνθήκη, το σύστημα χρησιμοποιεί έναν (1,1) global history predictor. Ο προβλέπτης αυτός διαθέτει ένα Pattern History Table (PHT) με συνολικά 2 εγγραφές. Οι 1-bit predictors είναι αρχικοποιημένοι στο NT.
- Το σύστημα διαθέτει μια *άδεια*, fully associative cache με 32 cache lines και μέγεθος block 16 bytes.
- Οι καταχωρητές R1, R2 περιέχουν την διεύθυνση του πρώτου στοιχείου των πινάκων A και B αντίστοιχα, οι οποίοι έχουν αποθηκευμένους αριθμούς διπλής ακρίβειας (μήκους 8 bytes ο καθένας). Οι πίνακες είναι ευθυγραμμισμένοι.

Δίνεται ο παρακάτω κώδικας :

```
0x00880000      LOOP:      LD      F0, 8(R1)
0x00880004              LD      F1, 8(R2)
0x00880008              ADDD   F4, F4, F0
0x0088000C              MULD   F4, F4, F1
0x00880010              ADDI   R4, R4, #1
0x00880014              ANDI   R5, R4, #1
0x00880018              SUBI   R5, R5, #1
0x0088001C              BEZ    R5, L1
0x00880020              ADDI   R1, R1, #8
0x00880024              LD      F0, 8(R1)
0x00880028              ADDD   F4, F4, F0
0x0088002C      L1:      ADDI   R1, R1, #8
0x00880030              ADDI   R2, R2, #8
0x00880034              SUBI   R6, R6, #1
0x00880038              BNEZ   R6, LOOP
0x0088003C              ADDD   F4, F4, F0
0x00880040              ADDD   F4, F4, F5
0x00880044              ADDI   R8, R8, #1
```

Δίνονται οι αρχικές τιμές $R4=0$ και $R6=2$. Εκτελέστε τον παραπάνω κώδικα για όλες τις επαναλήψεις του loop και δώστε τους χρόνους δρομολόγησης, εκτέλεσης και ολοκλήρωσης των εντολών σε έναν πίνακα όπως ο παρακάτω :

OP	IS	EX	WR	CMT	Σχόλιο
L.D F0, 8(R1)	1	2-??	??	??	

Στο πεδίο “Σχόλιο” δικαιολογήστε τυχόν καθυστερήσεις μεταξύ IS-EX, EX-WR και WR-CMT καθώς και ακυρώσεις εντολών.

Θέμα 4ο (20%)

Εξετάζουμε την εκτέλεση του παρακάτω κώδικα:

```
#define N 8
#define M 16

long y[N][M], z[M][N], x[M];

for ( i=0; i<N; i++ )
    for ( j=0; j<M; j++ )
        x[j] += y[i][j] + z[j][i];
```

Ισχύουν τα εξής:

- Το πρόγραμμα εκτελείται σε έναν επεξεργαστή με μόνο ένα επίπεδο κρυφής μνήμης δεδομένων, η οποία αρχικά είναι άδεια. Η κρυφή μνήμη είναι fully-associative, write-allocate με LRU πολιτική αντικατάστασης, και έχει χωρητικότητα 512 bytes.
- Το μέγεθος του block είναι 32 bytes, ενώ το μέγεθος ενός long είναι 8 bytes.
- Δήλωση διαδοχικών μεταβλητών στο πρόγραμμα συνεπάγεται αποθήκευσή τους σε διαδοχικές θέσεις στη μνήμη. Οι πίνακες είναι ευθυγραμμισμένοι.
- Η σειρά με την οποία γίνονται οι αναγνώσεις είναι: x, y, z
- Αρχικά η κρυφή μνήμη δεδομένων είναι άδεια, ενώ μόνο οι προσβάσεις στους πίνακες προκαλούν πρόσβαση στην cache. Οι υπόλοιπες μεταβλητές είναι αποθηκευμένες σε καταχωρητές.

A. Βρείτε το συνολικό ποσοστό αστοχίας (miss rate) για τις αναφορές που γίνονται στη μνήμη στο παραπάνω κώδικα.

B. Ποιες από τις παρακάτω τεχνικές βελτιστοποίησης θα ακολουθούσατε προκειμένου να βελτιώσετε την απόδοση; Δικαιολογήστε την επιλογή ή μη επιλογή κάθε τεχνικής. Για αυτές που επιλέξατε, υπολογίστε το νέο ποσοστό αστοχίας.

- (i) loop blocking
- (ii) loop unrolling
- (iii) προσθήκη victim cache 4 θέσεων με πολιτική FIFO.