



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΕΡΓΑΣΤΗΡΙΟ ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

Όνοματεπώνυμο: _____

ΑΜ: _____

ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΥΠΟΛΟΓΙΣΤΩΝ

(τμήμα Μ - Ω)

Κανονική εξεταστική Φεβρουαρίου 2016

Διάρκεια 2,5 ώρες

Οι εξετάσεις θα πραγματοποιηθούν ΧΩΡΙΣ την παρουσία βιβλίων, βοηθημάτων ή άλλου είδους σημειώσεων. Το μόνο που επιτρέπεται να φέρετε είναι ένα φύλλο Α4 με προσωπικές σημειώσεις.

ΘΕΜΑ 1^ο (30%)

A(10%). Να χαρακτηριστούν οι ακόλουθες προτάσεις ως αληθείς (Σ) ή ψευδείς (Λ):

1. Για το ίδιο datapath, ο κύκλος ενός single cycle επεξεργαστή είναι μικρότερος από τον αντίστοιχο ενός pipelined.
2. Είναι δυνατόν να παρατηρηθεί αστοχία TLB όταν μια σελίδα είναι απύσχα από την κύρια μνήμη.
3. Η χρονική τοπικότητα αναφέρεται στην πιθανότητα αντικείμενα κοντά σε αυτά που προσπελάστηκαν πρόσφατα να προσπελαστούν σύντομα.
4. Με την τεχνική της διοχέτευσης (pipeline) επιτυγχάνεται υψηλότερη απόδοση επειδή μειώνεται ο λανθάνων χρόνος κάθε εντολής (latency).
5. Οι εντολές `add $t1, $t0, $t0` και `sll $t1, $t0, 1` παράγουν το ίδιο αποτέλεσμα στον καταχωρητή \$t1.
6. Σε έναν υπολογιστή έχει μετρηθεί Ρυθμός αστοχίας στην I-Cache 0,4%, στην D-Cache 11,4%, και σταθμισμένος μέσος ρυθμός αστοχίας 7,4%.

B(10%). Δίνεται ένα πρότυπο κινητής υποδιαστολής 24 bits αντίστοιχο με το IEEE 754, όπου όμως ο εκθέτης έχει μήκος 9 bits και το κλάσμα 14 bits. Να βρεθεί η αναπαράσταση του αριθμού $-57/2^{25}$ στο πρότυπο αυτό.

Γ(10%). Δίνεται ο ακόλουθος κώδικας σε C καθώς και η ημιτελής απόδοσή του σε MIPS assembly. Δίνεται επίσης ότι η διεύθυνση του πρώτου στοιχείου του πίνακα A βρίσκεται αποθηκευμένη στον καταχωρητή \$s1, η διεύθυνση του πρώτου στοιχείου του πίνακα B στον καταχωρητή \$s2, ενώ ο καταχωρητής \$s0 περιέχει την τιμή του N. Υπενθυμίζεται τέλος πως ο τύπος int της C έχει μέγεθος 4 bytes (1 word για την αρχιτεκτονική MIPS). Ζητείται να συμπληρωθούν καταλλήλως τα κενά.

<pre>int A[N], B[N], i; B[0] = A[0]; for (i = 1; i < N; i++) B[i] = A[i] + B[i-1];</pre>	<pre>lw \$t0, 0(____) # load A[0] sw ____, 0(____) # store B[0] addi \$t1, ____, ____ # i = 1 LOOP: slt \$t0, \$t1, ____ beq ____, \$zero, END # if (i >= N) goto END sll \$t2, \$t1, ____ # i * 4 add \$t6, \$t2, ____ ____ \$t3, 0(____) # A[i] add \$t6, \$t2, \$s2 lw \$t4, ____ (____) # B[i-1] add \$t5, ____, ____ # A[i] + B[i-1] ____ \$t5, 0(____) addi \$t1, \$t1, ____ # i++ ____ ____ END:</pre>
--	--

ΘΕΜΑ 2° (30%)

Δίνεται ο παρακάτω κώδικας:

```
LOOP: lw $s1, 0($t1)
      lw $s2, 2000($t1)
      add $s0, $s1, $s1
      add $s0, $s0, $s1
      sub $s0, $s0, $s2
      sw $s0, 0($t2)
      addi $t1, $t1, 4
      addi $t2, $t2, 4
      slti $t0, $t1, 1024
      bne $t0, $zero, LOOP
```

Υποθέστε ότι η αρχική τιμή του καταχωρητή \$t1 είναι 512 και ότι ο κώδικας εκτελείται σε επεξεργαστή MIPS που διαθέτει την κλασική σωλήνωση 5 σταδίων (IF, ID, EX, MEM, WB) όπου η εγγραφή σε έναν καταχωρητή γίνεται στο πρώτο μισό του κύκλου ενώ η ανάγνωση από αυτόν στο δεύτερο μισό του ίδιου κύκλου. Υποθέστε, επίσης, ότι όλες οι αναφορές στη μνήμη ικανοποιούνται στον ίδιο κύκλο (δεν υπάρχουν δηλαδή αστοχίες), καθώς και ότι η απόφαση για μία εντολή διακλάδωσης λαμβάνεται στο στάδιο MEM.

A(12%). Αν δεν υπάρχουν σχήματα προώθησης, εκτελέστε την 1^η επανάληψη του βρόχου (μέχρι και το load της 2^{ης} επανάληψης). Δείξτε σε ένα διάγραμμα χρονισμού τα διάφορα στάδια της σωλήνωσης από τα οποία διέρχονται οι παραπάνω εντολές, σχολιάζοντας παράλληλα το λόγο οποιασδήποτε παρατηρούμενης καθυστέρησης. Πόσοι κύκλοι απαιτούνται για την εκτέλεση ολόκληρου του βρόχου;

B(10%). Για την ίδια ακολουθία εντολών δείξτε και πάλι το χρονισμό της σωλήνωσης, θεωρώντας όμως τώρα ότι υπάρχουν όλα τα δυνατά σχήματα προώθησης που είδαμε στο μάθημα. Πόσοι κύκλοι απαιτούνται τώρα για να ολοκληρωθεί η εκτέλεση του βρόχου;

Γ(8%). Προκειμένου να βελτιώσουμε περαιτέρω την απόδοση του δοθέντος κώδικα, εξετάζεται η υιοθέτηση κάποιου από τους παρακάτω branch predictors στην αρχιτεκτονική μας:

- στατικός, λαμβανόμενης διακλάδωσης.
- στατικός, μη λαμβανόμενης διακλάδωσης

- δυναμικός, με διάταξη πρόβλεψης 1-bit και αρχικοποιημένος στο NOT TAKEN.

Για τους σκοπούς αυτού του ερωτήματος, θεωρούμε πάλι ότι διατίθενται όλα τα δυνατά σχήματα προώθησης. Οι εντολές που προκύπτουν από την πρόβλεψη προσκομίζονται άμεσα χωρίς περαιτέρω ποινές. Ο υπολογισμός της διεύθυνσης προορισμού για μια εντολή διακλάδωσης συνεχίζει να γίνεται στο στάδιο MEM της σωλήνωσης. Σε περίπτωση που η πρόβλεψη είναι σωστή, η εκτέλεση του κώδικα συνεχίζεται κανονικά, ενώ στην αντίθετη περίπτωση, η σωλήνωση εκκενώνεται και στον επόμενο κύκλο μεταφέρεται η σωστή εντολή. Θεωρήστε ακόμα πως δεν υπάρχουν άλλες εντολές διακλάδωσης που να επηρεάζουν την πρόβλεψη (οι predictors αναφέρονται μόνο στην εντολή `bne $t0, $zero, LOOP`).

Με βάση τα παραπάνω, καλείστε να σχολιάσετε ποια διάταξη επιφέρει τα καλύτερα αποτελέσματα στην εκτέλεση του συγκεκριμένου βρόχου, υπολογίζοντας τις λανθασμένες προβλέψεις που παρατηρούνται ανά περίπτωση.

Παρατήρηση: Δεν είναι απαραίτητο να παρουσιάσετε εκ νέου το χρονοδιάγραμμα της σωλήνωσης ή να υπολογίσετε τους συνολικούς κύκλους εκτέλεσης.

ΘΕΜΑ 3^ο (20%)

A(15%). Έστω ότι διαθέτουμε τους τέσσερις παρακάτω δίσκους σε διάταξη RAID 5, με τα παρακάτω περιεχόμενα:

	DISK0	DISK1	DISK2	DISK3
STRIPE0	1011	0001	0110	
STRIPE1	1101	1111		0100
STRIPE2	0000		1101	1000
STRIPE3		1001	1010	1111

1. Να συμπληρωθούν τα ψηφία ισοτιμίας στον παραπάνω πίνακα (σκιασμένα κελιά).
2. Να ορισθεί πως θα γίνει η εγγραφή της νέας τιμής 1011 στο STRIPE2 του DISK0.
3. Υποθέστε ότι ο DISK3 καταστρέφεται και το STRIPE1 εμφανίζει την παρακάτω εικόνα:

	DISK0	DISK1	DISK2	DISK3
STRIPE1	1101	1100	0101	****

Να εξηγηθεί πως θα γίνει η ανάγνωση-ανάκτηση του STRIPE1 του κατεστραμμένου δίσκου.

B(5%). Ο μέσος χρόνος προσπέλασης μνήμης (AMAT) για έναν επεξεργαστή με ένα επίπεδο κρυφής μνήμης (L1) είναι ίσος με 3.2 κύκλους ρολογιού:

- Αν τα ζητούμενα δεδομένα βρίσκονται στην κρυφή μνήμη και είναι έγκυρα, προσπελάζονται σε 1 κύκλο ρολογιού.
- Αν τα δεδομένα δε βρεθούν ή δεν είναι έγκυρα, 80 επιπλέον κύκλοι ρολογιού απαιτούνται για την προσπέλασή τους στην κύρια μνήμη.

Σχεδιαστές επιχειρούν να βελτιώσουν περαιτέρω το μέσο χρόνο προσπέλασης έτσι ώστε να αποκομίσουν μια βελτίωση της τάξης του 55%. Για το σκοπό αυτό εξετάζουν την εισαγωγή και ενός δεύτερου επιπέδου κρυφής μνήμης (L2).

- Το δεύτερο αυτό επίπεδο μπορεί να είναι προσβάσιμο σε 6 επιπλέον κύκλους ρολογιού.
- Η εισαγωγή του δεν επηρεάζει το πρώτο επίπεδο όσον αφορά το χρόνο ευστοχίας και το μοτίβο των προσβάσεων.
- Η κύρια μνήμη εξακολουθεί να απαιτεί 80 επιπλέον κύκλους ρολογιού για πρόσβαση.

Καλείστε να υπολογίσετε το ελάχιστο ποσοστό ευστοχίας του δεύτερου επιπέδου κρυφής μνήμης έτσι ώστε ο στόχος για 55% βελτίωση να είναι επιτεύξιμος.

ΘΕΜΑ 4^ο (30%)

Εξετάζεται η εκτέλεση του ακόλουθου κώδικα C:

```
int i, j;
double A[16][32], B[32][32], C[16][32];

for (i = 0; i < 16; i++)
    for (j = 0; j < 32; j++)
        A[i][j] = C[i][j] * B[i+16][j%16];
```

Οι πίνακες περιέχουν στοιχεία κινητής υποδιαστολής διπλής ακρίβειας μεγέθους 8 bytes το καθένα. Κάνουμε τις εξής υποθέσεις:

- Το πρόγραμμα εκτελείται σε έναν επεξεργαστή με ένα επίπεδο κρυφής μνήμης δεδομένων, η οποία είναι αρχικά άδεια. Η κρυφή μνήμη είναι direct-mapped, write-allocate και αποτελείται από 128 blocks. Το μέγεθος του block είναι 64 bytes, ενώ η μικρότερη μονάδα που μπορεί να διευθυνσιοδοτηθεί είναι το 1 byte.
- Όλες οι μεταβλητές, πλην των στοιχείων των πινάκων, αποθηκεύονται σε καταχωρητές του επεξεργαστή και επομένως οποιαδήποτε αναφορά σε αυτές δεν συνεπάγεται προσπέλαση στην κρυφή μνήμη.
- Σε επίπεδο εντολών assembly, οι αναγνώσεις γίνονται με τη σειρά που εμφανίζονται στον κώδικα.
- Οι πίνακες είναι αποθηκευμένοι στην κύρια μνήμη κατά γραμμές και είναι ευθυγραμμισμένοι. Το πρώτο στοιχείο του πίνακα A βρίσκεται στη διεύθυνση 0x00000280, του πίνακα B στη διεύθυνση 0x00002280 και του πίνακα C στη διεύθυνση 0x000AE280.

A(15%). Βρείτε το συνολικό αριθμό των hits και misses για όλη την εκτέλεση του παραπάνω κώδικα, καθώς επίσης και το ποσοστό ευστοχίας. Υποδείξτε τον τύπο των misses.

B(15%). Προκειμένου να βελτιωθεί περαιτέρω η εκτέλεση του παραπάνω κώδικα, ζητείται να επιλεγεί μία από τις κάτωθι λύσεις:

- Για τον ίδιο τύπο και χαρακτηριστικά κρυφής μνήμης, αλλαγή της πολιτικής από write-allocate σε write-no-allocate.
- Αντικατάσταση της κρυφής μνήμης με μια άλλη τύπου 2-way associative, με ίδιο μέγεθος και αριθμό blocks που χρησιμοποιεί πολιτική εγγραφής write-allocate και αντικατάστασης LRU.

Η απάντησή σας καλείται να είναι τεκμηριωμένη, δηλαδή να περιέχει τον υπολογισμό των διαφορετικών, ανά περίπτωση, μοτίβων πρόσβασης και ποσοστών ευστοχίας.