



HAPPINESS CONSISTS OF A
SOLID FAITH, GOOD HEALTH,
AND A BAD MEMORY.



Εργασία 2 (υποχρεωτική) – Κρυφές Μνήμες

ΑΚΑΔΗΜΑΪΚΟ ΕΤΟΣ 2017 – 2018

(ΕΚΦΩΝΗΣΗ) ΔΕΥΤΕΡΑ 18 ΔΕΚΕΜΒΡΙΟΥ 2017

(ΠΑΡΑΔΟΣΗ ΣΤΟ ECLASS ΜΕΧΡΙ) **ΤΕΤΑΡΤΗ 24 ΙΑΝΟΥΑΡΙΟΥ 2018**

Επώνυμο	Όνομα	Αριθμός Μητρώου	Email
Μακρυγεώργος	Νικόλαος	1115201500238	sdi1500238@di.uoa.gr
Απλαδάς	Αγαμέμνων	1115201300013	sdi1300013@di.uoa.gr

Πληροφορίες για τις Υποχρεωτικές Εργασίες του μαθήματος

- Οι υποχρεωτικές εργασίες του μαθήματος είναι δύο. Σκοπός τους είναι η πλήρης κατανόηση των εννοιών του μαθήματος με χρήση αρχιτεκτονικών προσομοιωτών (architectural simulators). Η πρώτη υποχρεωτική εργασία αφορούσε τη διοχέτευση (pipelining) και η δεύτερη (αυτή) αφορά τις κρυφές μνήμες (cache memories).
- Η παράδοση των δύο εργασιών του μαθήματος είναι υποχρεωτική για τους φετινούς τριτοετείς φοιτητές καθώς επίσης και για όσους φοιτητές έχουν αριθμό μητρώου 2009 και μεταγενέστερο (δηλαδή πήραν το μάθημα για πρώτη φορά ως τριτοετείς από το εαρινό εξάμηνο του 2012 και μετά). Η βαθμολογία του μαθήματος θα προκύπτει από το γραπτό (60%), την εργασία της διοχέτευσης (20%), και την εργασία των κρυφών μνημών (20%). Καθένας από τους τρεις βαθμούς πρέπει να είναι προβιβάσιμος για να περαστεί προβιβάσιμος βαθμός στη γραμματεία. Οι προαιρετικές ασκήσεις που κατά καιρούς δίνονται δε βαθμολογούνται και δίνονται μόνο για να διευκολυνθεί η κατανόηση του μαθήματος.
- Για τους παλαιότερους φοιτητές (με αριθμό μητρώου 2008 και παλαιότερο) οι δύο εργασίες (pipeline, cache) είναι προαιρετικές. Αν κάποιος παλαιότερος φοιτητής δεν τις παραδώσει θα βαθμολογηθεί με ποσοστό 100% στο γραπτό. Αν τις παραδώσει, θα βαθμολογηθεί με τον παραπάνω τρόπο.
- Κάθε ομάδα μπορεί να αποτελείται **από 1 έως και 3 φοιτητές**. Συμπληρώστε τα στοιχεία όλων των μελών της ομάδας στον παραπάνω πίνακα. Όλα τα μέλη της ομάδας πρέπει να έχουν ισότιμη συμμετοχή και να γνωρίζουν τις λεπτομέρειες της υλοποίησης της ομάδας.
- Για την εξεταστική του Σεπτεμβρίου δε θα δοθούν άλλες εργασίες. Το Σεπτέμβριο θα εξεταστεί μόνο το γραπτό.
- Σε περίπτωση αντιγραφής θα μηδενίζονται όλες οι ομάδες που μετέχουν σε αυτή.
- Η παράδοση της **Εργασίας Κρυφών Μνημών** πρέπει να γίνει μέχρι τα μεσάνυχτα της **Τετάρτης 24 Ιανουαρίου 2018** (όπως συμφωνήσαμε για να διευκολυνθείτε με το φόρτο εργασίας σας) ηλεκτρονικά στο eclass (να ανεβάσετε ένα αρχείο zip ή rar με την τεκμηρίωσή σας σε Word ή PDF και/ή τους κώδικές σας). Μόνο ένα από τα μέλη της ομάδας να ανεβάσει την εργασία στο eclass.

Ζητούμενο

Γράψτε πρόγραμμα assembly MIPS για εκτέλεση στον προσομοιωτή Spim-Cache το οποίο να εκτελεί τον παρακάτω απλό υπολογισμό:

- Επεξεργάζεται έναν τυχαίο τετραγωνικό πίνακα ακεραίων $T[i][j]$ (4 byte κάθε ακεραίος – ο T αποθηκεύεται στο τμήμα δεδομένων του προγράμματος από την αρχή) με διαστάσεις 45×45 (δηλαδή $i, j = 0, 1, \dots, 44$) και αναζητά τετραγωνικούς υπο-πίνακες του T με διαστάσεις 5×5 που να είναι *διαγώνιοι* ή *συμμετρικοί*. Στο τέλος εκτυπώνει ως αποτέλεσμα σε διαδοχικές γραμμές την λίστα των υπο-πινάκων που έχουν την μία ή και τις δύο παραπάνω ιδιότητες καθώς και τις συντεταγμένες του πρώτου στοιχείου του κάθε υπο-πίνακα. Για παράδειγμα αν υπάρχει συμμετρικός υπο-πίνακας 5×5 με πρώτο στοιχείο το $T[2][3]$ (δηλαδή ο πίνακας αποτελείται από τα 5×5 στοιχεία $T[2][3] \dots T[2][7], T[3][3] \dots T[3][7], \dots, T[6][3] \dots T[6][7]$) να εκτυπώνει «Sub-matrix at position $i=2, j=3$ is symmetric» κ.ο.κ για κάθε υπο-πίνακα που διαθέτει τουλάχιστον μία από τις ιδιότητες.
- Ο χρόνος εκτέλεσης της εκτύπωσης του αποτελέσματος θα συνυπολογίζεται στο συνολικό χρόνο εκτέλεσης του προγράμματός σας.

Φυσικά, το πρωταρχικό ζητούμενο είναι το πρόγραμμα να εκτελείται σωστά για οποιοδήποτε περιεχόμενο του πίνακα T . Πέρα όμως από την ορθή εκτέλεση πρέπει να βελτιστοποιήσετε τη σχέση απόδοσης-κόστους του προγράμματός σας *γράφοντας κατάλληλα τον κώδικά σας αλλά και ρυθμίζοντας τις παραμέτρους* της κρυφής μνήμης στον Spim-Cache με βάση τα παρακάτω δεδομένα.

- Ως απόδοση P ορίζουμε τον αριθμό των κύκλων ρολογιού για την πλήρη εκτέλεση του προγράμματός σας (το μικρότερο P είναι το καλύτερο).

- Ως κόστος C ορίζουμε το κόστος της κρυφής μνήμης εντολών και δεδομένων αθροιστικά. Κάθε 128 byte μνήμης εντολών ή μνήμης δεδομένων αντιστοιχούν σε 1 μονάδα κόστους (το μικρότερο C είναι το καλύτερο). Για παράδειγμα, ένα σύστημα με μνήμη εντολών 256 byte (2x128) και μνήμη δεδομένων 512 byte (4x128) αντιστοιχεί σε 6 μονάδες κόστους.
- Υπάρχει ξεχωριστή κρυφή μνήμη εντολών (instruction cache) και ξεχωριστή κρυφή μνήμη δεδομένων (data cache) δηλαδή Harvard architecture σύμφωνα και με την ορολογία του προσομοιωτή.
- Δεν χρησιμοποιούνται delayed branches ή delayed loads (ρυθμίστε τον προσομοιωτή κατάλληλα).
- Κάθε εντολή που εκτελείται διαρκεί 1 κύκλο ρολογιού αν ευστοχεί (hit) στην κρυφή μνήμη εντολών (instruction cache) και επιβαρύνεται με ποινή 25/28/31 επιπλέον κύκλων ρολογιού αν αστοχεί (miss) σε αυτή αν το μέγεθος του μπλοκ είναι 4B/8B/16B αντίστοιχα.
- Εάν η εντολή έχει και αναφορά δεδομένων και ευστοχεί (hit) στη data cache δεν έχει καμία πρόσθετη επιβάρυνση. Εάν όμως η εντολή με αναφορά δεδομένων αστοχήσει (miss) στη data cache επιβαρύνεται με ποινή 25/28/31 επιπλέον κύκλων ρολογιού αν το μέγεθος του μπλοκ είναι 4B/8B/16B αντίστοιχα.
- Το μέγεθος των κρυφών μνημών μπορεί να είναι οποιοδήποτε από αυτά που υποστηρίζει ο προσομοιωτής. Η ποινή αστοχίας δεν επηρεάζεται από το μέγεθος των κρυφών μνημών.
- Το μέγεθος του μπλοκ κάθε κρυφής μνήμης μπορεί να είναι οποιοδήποτε από αυτά που υποστηρίζει ο προσομοιωτής αλλά η ποινή αστοχίας είναι διαφορετική σε κάθε περίπτωση όπως αναφέρεται παραπάνω.
- Η οργάνωση κάθε κρυφής μνήμης μπορεί να είναι οποιαδήποτε από αυτές που υποστηρίζει ο προσομοιωτής.
- Η πολιτική εγγραφής της κρυφής μνήμης δεδομένων πρέπει να είναι write-back-allocate.
- Ο αλγόριθμος αντικατάστασης πρέπει να είναι LRU και στις δύο κρυφές μνήμες (όταν φυσικά υπάρχει θέμα αντικατάστασης).

Συμπληρώστε τον παρακάτω πίνακα με τις ρυθμίσεις που επιλέξατε.

Κρυφή μνήμη εντολών: μέγεθος οργάνωση μέγεθος μπλοκ	Κρυφή μνήμη δεδομένων: μέγεθος οργάνωση μέγεθος μπλοκ
1024B Direct Mapping 16B	1024B Direct Mapping 16B

Συμπληρώστε τον παρακάτω πίνακα με τις πληροφορίες του τελικού σας προγράμματος (με βάση τα στατιστικά από τον προσομοιωτή και τα παραπάνω δεδομένα της εργασίας).

Συνολικό πλήθος εντολών που εκτελούνται	Συνολικοί κύκλοι ρολογιού για την πλήρη εκτέλεση του προγράμματος (P)	CPI	Ρυθμός ευστοχίας κρυφής μνήμης εντολών	Ρυθμός ευστοχίας κρυφής μνήμης δεδομένων	Κόστος συστήματος (C)	Απόδοση* κόστος (P*C)
57779	79496	1,37	0,99	0,93	16	1271936

Τεκμηρίωση

[Σύντομη τεκμηρίωση της λύσης σας μέχρι 7 σελίδες – μην αλλάζετε τη μορφοποίηση του κειμένου. Η τεκμηρίωσή σας μπορεί να περιλαμβάνει παραδείγματα ορθής εκτέλεσης του προγράμματος, διαγράμματα και γενικά οποιονδήποτε σχολιασμό για την ανάπτυξη του κώδικά σας.]

Το πρόγραμμα αποτελείται από δύο loop το βασικό που ελέγχει αν κάποιος υποπίνακας είναι διαγώνιος η συμμετρικός και ένα στο τέλος για τις εκτυπώσεις. Το βασικό loop πρώτα ελέγχει αν κάποιος υποπίνακας είναι διαγώνιος, εάν είναι αποθηκεύει τη θέση του πρώτου του στοιχείου και προχωράει στον επόμενο υποπίνακα (αφού ένας διαγώνιος πίνακας είναι και συμμετρικός). Αν δεν είναι διαγώνιος ελέγχει αν είναι συμμετρικός και εάν είναι αποθηκεύει τη θέση του πρώτου του στοιχείου και προχωράει στον επόμενο υποπίνακα. Αφού βρεί όλους τους υποπίνακες τους εκτυπώνει.

Ο πίνακας είναι μονοδιάστατος αλλά τον αντιμετωπίζουμε σαν διδιάστατο.

Για παράδειγμα αν θέλουμε ένα πίνακα 2 διαστάσεων 5x5 θα είναι της μορφής

.word -10, -9, -8, -7, -6,

.word -5, -4, -3, -2, -1,

.word 0, 1, 2, 3, 4

.word 5, 6, 7, 8, 9,

.word 10, 11, 12, 13, 14

όπου το στοιχείο 2,3 του πίνακα θα είναι το 3.

Ο δείκτης για τον πίνακα πηγαίνει μέχρι τη γραμμή και στήλη 40.

Για να βρούμε αν είναι διαγώνιος ένας υποπίνακας :

Ελέγχουμε αν τα στοιχεία της κύριας διαγωνίου είναι διάφορα του 0, αν ένα από αυτά είναι 0 τότε ο πίνακας δεν είναι διαγώνιος. Αντίστοιχα ελέγχουμε αν τα στοιχεία πάνω και κάτω από την κύρια διαγώνιο είναι 0, αν ένα από αυτά δεν είναι τότε ο πίνακας δεν είναι διαγώνιος. Όταν φτάσουμε στο τελευταίο στοιχείο της πρώτης γραμμής κάποιου υποπίνακα τότε η θέση του επόμενου δηλαδή αυτού που βρίσκεται στη πρώτη στήλη και δεύτερη γραμμή ισούται με το δείκτη του πίνακα + 164 . . . Στην περίπτωση που ο πίνακας είναι διαγώνιος αποθηκεύουμε σε διαδοχικές θέσεις σε ένα πίνακα τη γραμμή, τη στήλη του πρώτου στοιχείου και ένα αριθμό (το 2) που συμβολίζει ότι πίνακας είναι διαγώνιος .

Για να βρούμε αν είναι συμμετρικός ένας υποπίνακας :

Ελέγχουμε αν τα συμμετρικά στοιχεία του υποπίνακα είναι ίσα. Η συμμετρική θέση του στοιχείου που βρίσκεται στη πρώτη γραμμή και δεύτερη στήλη οποιουδήποτε υποπίνακα έχει απόσταση $t + 176$ από αυτό, όπου t είναι η θέση του στοιχείου στη πρώτη γραμμή και δεύτερη στήλη. Η συμμετρική θέση του στοιχείου που βρίσκεται στη πρώτη γραμμή και 3 στήλη οποιουδήποτε υποπίνακα έχει απόσταση $t + 352$, . . .

Στην περίπτωση που ο πίνακας είναι συμμετρικός αποθηκεύουμε σε διαδοχικές θέσεις σε ένα πίνακα τη γραμμή,τη στήλη του πρώτου στοιχείου και ένα αριθμό (το 1) που συμβολίζει οτι πίνακας είναι συμμετρικός .

Όταν βρίσκουμε οτι ένας υποπίνακας έχει μια απο τις 2 ιδιότητες αυξάνουμε ένα μετρητή για να ξέρουμε το μέγεθος του πίνακα που αποθηκεύουμε τις θέσεις των πρώτων στοιχείων των υποπινάκων.Στο τέλος με τη βοήθεια αυτού του μετρητή και ένα loop εκτυπώνουμε τα στοιχεία του πίνακα και την κατάλληλη ιδιότητα.

Ο πίνακας που χρησιμοποιώ είναι κάποιος τυχαίος ο οποίος περιέχει 53 υποπίνακες με μία απο τις δύο ιδιότητες