|  |
| --- |
| Εθνικό & Καποδιστριακο Πανεπιστημιο Αθηνων  Τμημα Πληροφορικησ & Τηλεπικοινωνιων  Μαθημα: **Αρχιτεκτονικη Υπολογιστων II**  **Εργασία 2 (*υποχρεωτική*) – Κρυφές Μνήμες** **Ακαδημαϊκο Ετοσ 2021 – 2022**  (εκφωνηση) Δευτερα 20 Δεκεμβριου 2021  (παραδοση στο eclass μεχρι) **Παρασκευη 21 Ιανουαριου 2022** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Επώνυμο** | **Όνομα** | **Αριθμός Μητρώου** | **Email** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# **Πληροφορίες για τις Υποχρεωτικές Εργασίες του μαθήματος**

* Οι υποχρεωτικές εργασίες του μαθήματος είναι **δύο**. Σκοπός τους είναι η κατανόηση των εννοιών του μαθήματος με χρήση αρχιτεκτονικών προσομοιωτών. Η πρώτη υποχρεωτική εργασία αφορά τη διοχέτευση (pipelining) και η δεύτερη (αυτή) αφορά τις κρυφές μνήμες (cache memories).
* Οι δύο εργασίες είναι υποχρεωτικές και η βαθμολογία του μαθήματος θα προκύπτει από το γραπτό (60%), την εργασία της διοχέτευσης (20%), και την εργασία των κρυφών μνημών (20%).
* Κάθε ομάδα μπορεί να αποτελείται **από 1 έως και 3 φοιτητές** (η υποβολή να γίνει μόνο από έναν φοιτητή εκ μέρους όλους της ομάδας). 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 Σεπτεμβρίου δε θα δοθούν άλλες εργασίες. Το Σεπτέμβριο εξετάζεται μόνο το γραπτό.
* Σε περίπτωση αντιγραφής θα μηδενίζονται όλες οι ομάδες που μετέχουν σε αυτή.
* Η παράδοση της **Εργασίας Κρυφών Μνημών** πρέπει να γίνει **ηλεκτρονικά μέχρι τα μεσάνυχτα της προθεσμίας και μόνο στο eclass** (να ανεβάσετε ένα μόνο αρχείο zip ή rar με την τεκμηρίωσή σας σε PDF και τους κώδικές σας). **Μην περιμένετε μέχρι την τελευταία στιγμή – κάθε εργασία απαιτεί τον χρόνο της**.

# **Ζητούμενο**

Το ζητούμενο της εργασίας είναι η **σχεδίαση μιας «οικογένειας» τριών μικροεπεξεργαστών** που να διαφέρουν στην απόδοση και το κόστος για την ίδια υπολογιστική εργασία. Η σχεδίαση και η αξιολόγηση των επεξεργαστών θα γίνει στον προσομοιωτή QtMips.

Η υπολογιστική εργασία αποτελείται από δύο διακριτά βήματα (sort and search):

(α) Ένας αρχικά μη ταξινομημένος πίνακας 40 000 θετικών ακεραίων αριθμών (n[1], …, n[40 000]) μεγέθους 1 byte ο καθένας πρέπει να ταξινομηθεί με αύξουσα σειρά. Πρέπει να γίνεται έλεγχος ότι οι αριθμοί είναι πράγματι θετικοί και το πρόγραμμα να σταματάει εάν βρεθεί αρνητικός αριθμός ή το μηδέν. Ο πίνακας βρίσκεται στο τμήμα δεδομένων του προγράμματός σας και δεν απαιτείται είσοδος/έξοδος. Η ταξινόμηση να χρησιμοποιεί τον αλγόριθμο quicksort.

(β) Στον ταξινομημένο πίνακα των ακεραίων του 1 byte γίνεται αναζήτηση ενός συγκεκριμένου αριθμού X (που επίσης βρίσκεται στο τμήμα δεδομένων του προγράμματός σας). Η αναζήτηση να χρησιμοποιεί τον αλγόριθμο δυαδικής αναζήτησης (binary search). Αν ο αριθμός βρεθεί τότε μια μεταβλητή POS (και πάλι στο τμήμα δεδομένων του προγράμματος) πρέπει να περιέχει την πρώτη θέση στην οποία βρέθηκε ο αριθμός (1 μέχρι 40 000). Αν ο αριθμός δεν βρεθεί τότε η POS πρέπει να έχει την τιμή 0.

Οι τρεις επεξεργαστές έχουν κοινή σχεδίαση στον πυρήνα του pipeline: διαθέτουν branch predictor των 2-bit με BHT που προσπελαύνεται με 5 bit και επίλυση των διακλαδώσεων στο στάδιο EX και πλήρη μονάδα κινδύνων με προώθηση. Διαφέρουν στο σύστημα της μνήμης.

**Ο πρώτος επεξεργαστής** (codename: *turtle*) είναι μια φθηνή έκδοση της οικογένειας με το ελάχιστο δυνατό κόστος που σημαίνει ότι δεν χρησιμοποιεί κρυφή μνήμη (cache). Η προσπέλαση της κύριας μνήμης διαρκεί 40 κύκλους ρολογιού. Το κόστος του επεξεργαστή είναι 20 ευρώ και ο ρυθμός του ρολογιού του είναι 500 MHz.

**Ο δεύτερος επεξεργαστής** (codename: *rabbit*) είναι μια μέση έκδοση της οικογένειας που χρησιμοποιεί μόνο ένα επίπεδο κρυφής μνήμης (L1) για εντολές και δεδομένα. Η προσπέλαση της κύριας μνήμης και σε αυτόν τον επεξεργαστή διαρκεί 40 κύκλους. Η κρυφή μνήμη εντολών/προγράμματος (L1 program cache στην ορολογία του QtMips) έχει μέγεθος 8ΚΒ και μπορεί να περιέχει 4, 8, 16, ή 32 λέξεις ανά μπλοκ. Μπορείτε να επιλέξετε όποια συσχετιστικότητα (associativity) και όποια πολιτική αντικατάστασης επιθυμείτε. Η κρυφή μνήμη δεδομένων (L1 data cache) μπορεί να έχει μέγεθος μεταξύ 4, 8, ή 16ΚΒ και μπορεί να περιέχει 4, 8, 16, ή 32 λέξεις ανά μπλοκ. Μπορείτε και πάλι να επιλέξετε όποια συσχετιστικότητα και όποια πολιτική αντικατάστασης επιθυμείτε. Η πολιτική εγγραφής και κατανομής πρέπει να είναι ετερόχρονη (write back) και με κατανομή σε εγγραφή (write allocate). Το κόστος αυτού του επεξεργαστή είναι 20, 25, ή 30 ευρώ υψηλότερο από τον προηγούμενο αν η κρυφή μνήμη δεδομένων έχει μέγεθος 4, 8, ή 16ΚΒ, αντίστοιχα. Ο ρυθμός ρολογιού είναι 500 MHz αν η συσχετιστικότητα είναι 1 (direct mapped) αλλά μειώνεται κατά 10 MHz για κάθε διπλασιασμό της συσχετιστικότητας.

**Ο τρίτος επεξεργαστής** (codename: *puma*) είναι μια προηγμένη έκδοση της οικογένειας που χρησιμοποιεί δύο επίπεδα κρυφής μνήμης (ξεχωριστή L1 για εντολές και δεδομένα και ενιαία L2). Η προσπέλαση της κύριας μνήμης και σε αυτόν τον επεξεργαστή διαρκεί 40 κύκλους και η προσπέλαση της L2 διαρκεί 5 κύκλους. Οι κρυφές μνήμες L1 είναι ίδιες με του προηγούμενου επεξεργαστή (θα κρατήσετε τη σχεδίαση στην οποία καταλήξατε – συνεπώς και το κόστος και τον ρυθμό ρολογιού). Η κρύφη μνήμη L2 έχει μέγεθος 16, 32, ή 64 ΚΒ και έχει ίδιο μέγεθος μπλοκ με την L1 data cache του προηγούμενου επεξεργαστή. Μπορείτε να επιλέξετε όποια συσχετιστικότητα και όποια πολιτική αντικατάστασης επιθυμείτε για την L2. Η πολιτική εγγραφής και κατανομής πρέπει να είναι ετερόχρονη (write back) και με κατανομή σε εγγραφή (write allocate). Το κόστος αυτού του επεξεργαστή είναι 50, 75 ή 100 ευρώ υψηλότερο από τον προηγούμενο αν η κρυφή μνήμη L2 έχει μέγεθος 16, 32, ή 64 ΚΒ, αντίστοιχα. Ο ρυθμός ρολογιού είναι είναι ίσος με αυτόν του προηγούμενου επεξεργαστή.

* Να συμπληρώσετε τον παρακάτω πίνακα με **τα χαρακτηριστικά της σχεδίασής σας** για τον καθένα από τους τρεις επεξεργαστές.
* Να μετρήσετε **τον χρόνο εκτέλεσης** του προγράμματός σας στις τρεις σχεδιάσεις σας χρησιμοποιώντας τον αριθμό των κύκλων ρολογιού από τον QtMips και τον ρυθμό ρολογιού της σχεδίασής σας. Εάν χρησιμοποιήσετε διάφορα σύνολα δεδομένων να περιγράψετε τον τρόπο με τον οποίο παίρνετε τις μετρήσεις σας (μπορείτε να χρησιμοποιήσετε μέσες τιμές για παράδειγμα).
* Να υπολογίσετε **τον λόγο απόδοσης προς κόστος** του προγράμματός σας και στις τρεις σχεδιάσεις σας (όπως κάνατε τον υπολογισμό και στην Εργασία 1).
* Να περιγράψετε **τρία διαφορετικά υπολογιστικά συστήματα** στα οποία θα χρησιμοποιούσατε καθέναν από τους τρεις επεξεργαστές. Ποιοι είναι οι λόγοι για τους οποίους θεωρείτε κατάλληλο τον καθένα για το συγκεκριμένο σύστημα.

|  |  |  |  |
| --- | --- | --- | --- |
| **Χαρακτηριστικό** | **Πρώτος  επεξεργαστής  (*turtle*)** | **Δεύτερος  επεξεργαστής  (*rabbit*)** | **Τρίτος  επεξεργαστής  (*puma*)** |
| L1 program cache (μέγεθος, μέγεθος μπλοκ, συσχετιστικότητα, πολιτική αντικατάστασης) |  |  |  |
| L1 data cache (μέγεθος, μέγεθος μπλοκ, συσχετιστικότητα, πολιτική αντικατάστασης) |  |  |  |
| L2 cache (μέγεθος, μέγεθος μπλοκ, συσχετιστικότητα, πολιτική αντικατάστασης) |  |  |  |
| Κόστος |  |  |  |
| Ρυθμός ρολογιού |  |  |  |
| Κύκλοι εκτέλεσης |  |  |  |
| Χρόνος εκτέλεσης |  |  |  |
| Λόγος απόδοσης/κόστους |  |  |  |
| Σύστημα για το οποίο είναι κατάλληλος (~150 λέξεις) |  |  |  |

# **Τεκμηρίωση**

[ Σύντομη τεκμηρίωση της λύσης σας μέχρι **10 σελίδες ξεκινώντας από την επόμενη σελίδα** – μην αλλάζετε τη μορφοποίηση του κειμένου (**και παραδώστε την τεκμηρίωση σε αρχείο PDF**). Η τεκμηρίωσή σας πρέπει να περιλαμβάνει παραδείγματα ορθής εκτέλεσης και σχολιασμό για την επίλυση του προβλήματος και την επίτευξη του ζητούμενου. Μπορείτε να χρησιμοποιήσετε εικόνες, διαγράμματα και ό,τι άλλο μπορεί να βοηθήσει στην εξήγηση της δουλειάς σας. ]