



## Εργασία 1 (υποχρεωτική) – Διοχέτευση

ΑΚΑΔΗΜΑΪΚΟ ΕΤΟΣ 2017 – 2018

(ΕΚΦΩΝΗΣΗ) ΔΕΥΤΕΡΑ 13 ΝΟΕΜΒΡΙΟΥ 2017

(ΠΑΡΑΔΟΣΗ ΣΤΟ ECLASS ΜΕΧΡΙ) ΤΡΙΤΗ 5 ΔΕΚΕΜΒΡΙΟΥ 2017

Επώνυμο  
Μακρυγέρως

Όνομα  
Νικόλαος

Αριθμός Μητρώου  
1115201500238

Email  
sdi1500238@di.uoa.gr

### 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 (αυτή) αφορά τη διοχέτευση (pipelining) και η δεύτερη θα αφορά τις κρυφές μνήμες (cache memories).
- Οι δύο αυτές 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%). Αυτός είναι ο τρόπος εξέτασης του μαθήματος για όσους φοιτητές έχουν αριθμό μητρώου 2009 και μεταγενέστερο (δηλαδή πήραν το μάθημα για πρώτη φορά το εαρινό εξάμηνο του 2012 και μετά). Καθένας από τους τρεις βαθμούς πρέπει να είναι προβιβάσιμος για να περαστεί προβιβάσιμος βαθμός στη γραμματεία.
- 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 (γραπτό + 2 εργασίες).
- 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 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 Διοχέτευσης** πρέπει να γίνει μέχρι τα **μεσάνυχτα της προθεσμίας ηλεκτρονικά** και μόνο στο eclass (να ανεβάσετε ένα μόνο αρχείο zip ή rar με την τεκμηρίωσή σας σε Word ή PDF και τους κώδικές σας). **Μην περιμένετε μέχρι την τελευταία στιγμή. Δεν θα υπάρξει παράταση στην προθεσμία παράδοσης ώστε να διατεθεί αρκετός χρόνος και για την εργασία των κρυφών μνημών της οποίας η εκφώνηση θα δοθεί αμέσως μετά.**

### Ζητούμενο

Το ζητούμενο της εργασίας είναι να υλοποιήσετε ένα πρόγραμμα σε συμβολική γλώσσα για τον προσομοιωτή WinMIPS64 και να επιλέξετε τη βέλτιστη διαμόρφωση της CPU για την εκτέλεσή του στον μικρότερο δυνατό χρόνο.

Το πρόγραμμα πρέπει να εκτελεί την εξής λειτουργία: επεξεργάζεται έναν μη ταξινομημένο πίνακα 400 προσημασμένων ακεραίων αριθμών  $A[i]$  ( $i = 0, 1, \dots, 399$ ) και υπολογίζει: το πλήθος (X) των ακεραίων που έχουν ίσο πλήθος άσσων με τον αντίθετό τους (στις δυαδικές τους αναπαράστάσεις), το πλήθος (Z) των ακεραίων που έχουν περισσότερα από 10 μηδενικά στη δυαδική τους αναπαράσταση, το πλήθος (B) των ακεραίων που έχουν απόλυτη τιμή μεγαλύτερη του 1000, το πλήθος (C) των ακεραίων που έχουν την ίδια αριθμητική τιμή αν γραφούν αντίστροφα δυαδικά, τη θέση (P, από 0 μέχρι 399) που βρίσκεται ο μεγαλύτερος ακεραίος της λίστας (αν εμφανίζεται περισσότερες από μία φορές να κρατάει τη μεγαλύτερη θέση). Τα A, X, Z, B, C, P πρέπει να είναι μεταβλητές στο τμήμα δεδομένων του προγράμματός σας. Στο τέλος της εκτέλεσης του προγράμματος να εκτυπώνονται στο Terminal οι τιμές των αποτελεσμάτων X, Z, B, C, P σε μία γραμμή: "X=..., Z=..., B=..., C=..., P=...".

Μπορείτε να κάνετε αλλαγές στο Configuration του προσομοιωτή (Enable Forwarding, Enable Branch Target Buffer, Enable Delay Slot) και αλλαγές στον κώδικα ώστε να μειώσετε τις καθυστερήσεις (stalls) που αυξάνουν το χρόνο εκτέλεσης του προγράμματος.

Θα πρέπει να υποθέσετε ότι ο ρυθμός του ρολογιού του μικροεπεξεργαστή είναι διαφορετικός ανάλογα με την διαμόρφωση που θα επιλέξετε. Ο πίνακας που ακολουθεί δίνει το ρυθμό του ρολογιού σε συνάρτηση με την κάθε διαμόρφωση.

Forwarding	Branch Target Buffer	Branch Delay Slot	Ρυθμός ρολογιού
-	-	-	500 MHz
-	-	✓	485 MHz
-	✓	-	470 MHz
✓	-	-	445 MHz
✓	-	✓	430 MHz
✓	✓	-	415 MHz

Μεταξύ των προγραμμάτων που εκτελούνται σωστά, τα ταχύτερα θα βαθμολογηθούν με μεγαλύτερο βαθμό. Ως ταχύτερο πρόγραμμα εννοείται αυτό που σε μια μηχανή MIPS με τη συχνότητα του παραπάνω πίνακα που αντιστοιχεί στη διαμόρφωση που επιλέξατε θα εκτελεστεί στον μικρότερο χρόνο (πλήθος κύκλων x χρόνος κύκλου ρολογιού).

Συμπληρώσετε τον ακόλουθο πίνακα για το πρόγραμμά σας.

	Επιλογή Forwarding (Ναι / Όχι)	Επιλογή Branch Target Buffer (Ναι / Όχι)	Επιλογή Branch Delay Slot (Ναι / Όχι)	Περίοδος ρολογιού (nsec) – με βάση τον παραπάνω πίνακα και τη διαμόρφωσή σας	Κύκλοι ρολογιού εκτέλεσης	Χρόνος εκτέλεσης (sec)
Απαντήσεις για το Πρόγραμμα μου	Ναι	Όχι	Όχι	445 MHz	204836	474

### Τεκμηρίωση

[ Σύντομη τεκμηρίωση της λύσης σας μέχρι **5 σελίδες ξεκινώντας από την επόμενη** – μην αλλάζετε τη μορφοποίηση του κειμένου. Η τεκμηρίωσή σας πρέπει να περιλαμβάνει παραδείγματα ορθής εκτέλεσης του προγράμματος και σχολιασμό για την επίλυση του προβλήματος. Μπορείτε να χρησιμοποιήσετε εικόνες, διαγράμματα και ό,τι άλλο μπορεί να βοηθήσει στην εξήγηση της δουλειάς σας. ]

Το πρόγραμμα αποτελείται απο ενα βασικό loop που κάνει 400 πεναλήψεις.Μέσα σε αυτό το loop γίνεται ο υπολογισμός όλων των απαιτήσεων για καθε στοιχείο του πινακα.Έξω απο το loop αρχικοποιούνται κάποιοι καταχωρητές,αποθηκεύονται στη μνήμη τα αποτελέσματα και εκτυπώνονται τα αποτελέσματα.

Στο τμήμα data υπάρχουν οι πέντε μεταβλητές X,Z,B,C,P, οι διευθύνσεις των control,data,οι συμβολοσειρές που πρέπει να εκτυπωθούν και ένας τυχαίος πινακας ακεραίων τετρακοσίων στοιχείων.Ο πίνακας προέκυψε απο ένα πίνακα 100 στοιχείων που υπάρχει μέσα σε ενα πρόγραμμα με όνομα isort.s στον φάκελο του WinMips.Τον αντέγραψα τέσσερις φορές στο δικό μου πρόγραμμα.

Υπολογισμός μονάδων σε ενα ακέραιο : Η ίδια διαδικασία και για το άθροισμα μονάδων του θετικού και για του αρνητικού.

Η διαδικασία γίνεται σε ενα loop δύο επαναλήψεων πέντε βημάτων. Ο τρόπος υπολογισμού είναι μια επαναλαμβανόμενη διαδικασία αθροίζοντας γειτονικά bit και τα αποθηκεύουμε σε πεδία 2 bit,στη συνέχεια αθροίζουμε γειτονικές ακολουθίες 2 bit και τις αποθηκεύουμε σε πεδία 4 bit, αθροίζοντας γειτονικές ακολουθίες 4 bit και αποθηκεύοντας σε πεδία 8 bit, αυτό γίνεται μέχρι να καταλήξουμε σε ενα 32 bit αριθμό ο οποίος είναι το άθροισμα τον μονάδων του αριθμού.Επαναλαμβάνεται η διαδικασία για τα 32 αριστερά bit. Για τον υπολογισμό αυτο χρησιμοποιώ κατάλληλες μάσκες 16 bit.

Παράδειγμα :

για ενα ακέραιο 32 bit π.χ. ο οποίος έχει 14 άσσους αν αθροίσουμε ανα δυο τα bit : (1)	0 1 0 0 1 0 1 0 1 0 1 0 1 0 1 1 1 1 0 0 0 0 1 0 0 0 1 1 0 0 0
αθροίσουμε ανα τέσσερα(2)	01   00   01   01   01   01   01   10   10   01   00   01   00   01   01   00
αθροίσουμε ανα οχτώ(3)	0001   0010   0010   0011   0011   0001   0001   0001
ανα δεκαέξι(4)	00000011   00000101   00000100   00000010
αθροίζουμε τα τελευταία 16 και 16 bit(5)	0000000000000100   0000000000000110

τα τελευταία bit ισούτε με τον αριθμό 14 όσα και οι άσσοι του αρχικού αριθμού.

Στην περίπτωση που ο αριθμός είναι 64 bit γίνεται 2 φορές αυτή η διαδικασία κάνοντας τον 64 bit αριθμό 32 θέσεις ολίσθηση δεξιά.Επομένως στην πρώτη επανάληψη του loop υπολογίζουμε τους άσσους στα 32 δεξιά bit και στην δεύτερη επανάληψη τους άσσους στα 32 αριστερά bit.

Στο πρόγραμμα για να υλοποιηθεί η πρώτη γραμμή δηλαδή να αθροίσουμε ανα δύο τα bit γίνεται με το να κάνουμε αντιγραφή του αριθμού σε άλλους 3 καταχωρητές άρα σύνολο 4 καταχωρητές με τον ιδιο αριθμό.Κάνουμε τον πρώτο καταχωρητή and με 0x5555,τον δεύτερο τον κάνουμε ολίσθηση 16 bit δεξιά τον κάνουμε and με το 0x5555 και ολίσθηση 16 bit αριστερά και αυτους τους 2 καταχωρητές τους κάνουμε or σε έναν απο τους 2 αυτους καταχωρητές.Τους άλλους 2 κάνουμε δεξιά ολίσθηση κατα 1 θεση και την ίδια διαδικασία όπως προηγουμένως με το 0x5555.Οι 2 τελικοί καταχωρητές απο τα or προστίθονται σε ένα καταχωρητή,αυτόν τον καταχωρητή τον αντιγράφουμε στους άλλους τέσσερις για το βήμα (2).Αυτό το αποτέλεσμα αντιστοιχεί στο αποτέλεσμα του βήματος (1).Αντίστοιχα κάνουμε τα βήματα 2,3,4,5.Στο βήμα 2 κάνουμε and με 0x3333,στο 3 με 0xf0f,στο 4 με 0x00ff και στο πέμπτο με το 0xffff. Το αποτέλεσμα είναι το άθροισμα των άσσων στα 32 δεξιά bit του αριθμού,για τα 32 αριστερά κάνουμε τον αρχικό αριθμό ολίσθηση 32 bit δεξιά και επαναλαμβάνουμε την ίδια διαδικασία.

Η ίδια διαδικασία γίνεται και για τον αρνητικό.

Για να βρούμε αν έχουν ίσο αριθμό άσσον κάνουμε bne τους δύο καταχωρητές με τα αθροίσματα απο τους άσσους του αριθμού και του αντιθέτου του και αν είναι ίσα αυξάνει ενα counter αλλιώς προχωράει παρακάτω.

Εφόσον ξέρουμε πόσους άσσους έχει ενας αριθμός ξέρουμε και πόσα μηδενικά εχει, κάνουμε μια αφαίρεση απο το 64 – καταχωρητής με άσσους και βάζω το αποτέλεσμα σε ενα καταχωρητή.Συγκρίνω με slti τον καταχωρητή με το 10 και αν το αποτέλεσμα είναι 0 αυξάνω τον αντίστοιχο counter.

Για να δω αν έχει απόλυτη τιμή μεγαλύτερη απο 1000 κοιτάω αρχικά αν είναι μηδέν,αν είναι τότε προχωράω παρακατω. Αν δεν ειναι 0 κοιτάω αν είναι θετικός η αρνητικός και ανάλογα τι ισχύει συγκρίνω με το 1000 ή με το - 1000 και αν ισχύει ενα απο τα δύο αυξάνω τον counter.

