



## ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΕΡΓΑΣΤΗΡΙΟ ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

Όνοματεπώνυμο: \_\_\_\_\_

ΑΜ: \_\_\_\_\_

### ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΥΠΟΛΟΓΙΣΤΩΝ

(τμήμα Λ - Ω)

Εξέταση Μαρτίου 2014

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.

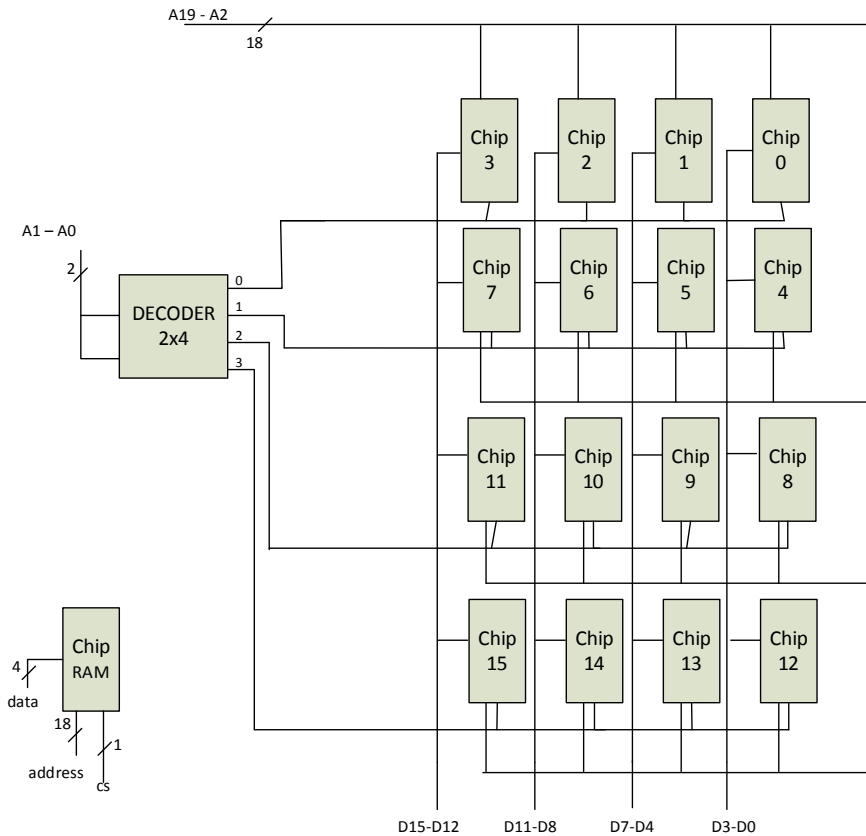
#### Θέμα 1<sup>ο</sup>

1. Ο υπολογισμός του παραγοντικού ενός αριθμού μπορεί να γίνει επαναληπτικά, ως εξής:

```
int factorial(x){
    int i, fact = 1;
    for(i=1; i<=x; i++){
        fact = fact * i;
    }
    return fact;
}
```

Να γραφεί ισοδύναμη ρουτίνα σε assembly MIPS που υλοποιεί αυτή τη συνάρτηση. Η ρουτίνα factorial του MIPS θα λαμβάνει το όρισμά της (x) μέσω του καταχωρητή \$a1 και θα επιστρέφει την υπολογισμένη τιμή στον καταχωρητή \$v1. Να αγνοηθεί η περίπτωση υπερχείλισης.

2. Να αναφέρετε τους «κινδύνους δεδομένων» σε μία σωλήνωση και να εξηγήσετε ποιους από αυτούς είναι δυνατόν να καταλήξουν σε stall στην αρχιτεκτονική MIPS, και με ποιες προϋποθέσεις συμβαίνει αυτό.
3. Στην ακόλουθη διάταξη μνήμης RAM, (α) να βρεθεί το μέγεθος της συνολικής μνήμης και (β) να εντοπισθούν οι ψηφίδες που φιλοξενούν τις διευθύνσεις μνήμης: 0xF1452, 0xF1451, 0xF1455, 0xFFFF47



4. Να εξηγηθεί ποιοι από τους παρακάτω συνδυασμούς είναι πιθανοί και ποιοι όχι:

TLB	Πίνακας Σελίδων	Κρυφή Μνήμη	ΝΑΙ/ΟΧΙ
miss	hit	hit	
miss	miss	miss	
hit	miss	miss	
hit	miss	hit	

5. Αιτιολογείστε εάν συμφωνείτε ή όχι με την παρακάτω πρόταση:

«Όσο μεγαλύτερο είναι το ποσοστό των αναγνώσεων στη μνήμη τόσο αποδοτικότερη γίνεται η πολιτική της ετερόχρονης εγγραφής (write-back) σε σχέση με την πολιτική της ταυτόχρονης εγγραφής (write-through) στην κρυφή μνήμη»

## Θέμα 2<sup>ο</sup>

Δίνεται το ακόλουθο τμήμα προγράμματος σε assembly MIPS:

```

addi $2, $zero, 64
add $1, $zero, $zero
Loop: lw $4, 400($1)
      addi $3, $1, 4
      lw $5, 400($2)
      sub $6, $5, $4

```

```

srl $6, $6, 1
sw $6, 800($3)
addi $1, $1, 4
bne $1, $2, Loop

```

Υποθέτουμε εκτέλεση του παραπάνω τμήματος κώδικα σε επεξεργαστή MIPS με αρχιτεκτονική σωλήνωσης (pipeline) 5 σταδίων (IF, ID, EX, MEM, WB) όπου η εγγραφή σε κάποιον καταχωρητή γίνεται στο πρώτο μισό ενός κύκλου, ενώ η ανάγνωση στο δεύτερο μισό του ίδιου κύκλου. Υποθέτουμε επίσης, ότι όλες οι αναφορές στη μνήμη ικανοποιούνται σε 1 κύκλο (δεν υπάρχουν αστοχίες), ενώ η απόφαση για μια εντολή διακλάδωσης λαμβάνεται στο στάδιο MEM.

1. Αρχικά υποθέτουμε έλλειψη σχήματος προώθησης. Συμπληρώστε το διάγραμμα χρονισμού για την πρώτη επανάληψη του βρόχου, επισημαίνοντας κάθε φορά το λόγο που προκαλείται η οποιαδήποτε καθυστέρηση. Υπολογίστε το συνολικό αριθμό κύκλων που απαιτούνται για την ολοκλήρωση του προγράμματος.
2. Στη συνέχεια υποθέτουμε την ύπαρξη σχήματος προώθησης. Παρουσιάστε το διάγραμμα χρονισμού της πρώτης επανάληψης του βρόχου. Υπολογίστε το νέο συνολικό αριθμό κύκλων που απαιτούνται μέχρι το πέρας του προγράμματος.
3. Είναι δυνατόν να επιτευχθεί καλύτερη απόδοση χωρίς να αλλάξει η σημασιολογία του προγράμματος, όταν παρέχεται σχήμα προώθησης; Αν ναι, υποδείξτε τον τρόπο και υπολογίστε το νέο πλήθος απαιτούμενων κύκλων για την ολοκλήρωση του βελτιστοποιημένου πλέον κώδικα.

### Θέμα 3<sup>ο</sup>

Στο πλαίσιο αυτού του θέματος καλείσθε να εκτιμήσετε την επίδοση ενός τμήματος προγράμματος συναρτήσεως της εσωτερικής οργάνωσης της κρυφής μνήμης (**L1 cache**). Διατίθεται μία μόνο cache μεγέθους **δεδομένων 256 Kbytes** εφοδιασμένη με **write back (1 "dirty" bit), write allocate** και **LRU (1 bit, όπου είναι απαραίτητο) πολιτικές, διευθύνσεις μεγέθους 32 bit** και **διευθυνσιοδότηση 1 byte**. Το μέγεθος κάθε **block** είναι **64 bytes δεδομένων**. Υποθέτουμε τις ακόλουθες οργανώσεις:

- Άμεση οργάνωση
- Οργάνωση συνόλου 2 δρόμων

1. Για κάθε μία από τις παραπάνω οργανώσεις ζητείται να βρεθεί το μέγεθος των πεδίων **tag, index, byte\_offset** σε bits, το **συνολικό μέγεθος της cache**, καθώς και το **ποσοστό ωφέλιμου** χώρου αυτής.
2. Δίνεται το ακόλουθο τμήμα κώδικα σε C:

```

double A[64], B[64], C[64];
for (i=0; i<64; i++)
    A[i] = (B[i] + C[i]) / 2;

```

Οι μονοδιάστατοι πίνακες A, B, C περιέχουν στοιχεία διπλής ακριβείας (double precision) μεγέθους **8 byte** το καθένα. Βρίσκονται στη μνήμη RAM, ξεκινώντας από τις θέσεις **0xFF004F40**, **0xFF124F40**, **0xFF804F40**, αντίστοιχα. **Η L1 cache είναι αρχικά άδεια** και η οποιαδήποτε αναφορά σε μεταβλητή πέραν των τριών πινάκων εξυπηρετείται από τους καταχωρητές του επεξεργαστή.

A. Για κάθε μία από τις παραπάνω οργανώσεις ζητείται το πλήθος των misses που θα συμβούν κατά την εκτέλεση του κώδικα, καθώς και το είδος αυτών.

B. Πώς θα μεταβληθεί ο συνολικός αριθμός misses σε κάθε μία από τις παραπάνω οργανώσεις μνήμης αν αλλάξουμε την πολιτική write allocate σε write no-allocate; Δικαιολογήστε την απάντησή σας .

#### Θέμα 4<sup>ο</sup>

Έστω ότι διαθέτουμε τους πέντε παρακάτω δίσκους σε διάταξη RAID 5, με τα παρακάτω περιεχόμενα (τα σκιασμένα κελιά περιέχουν τα ψηφία ισοτιμίας):

	Disk 0	Disk 1	Disk 2	Disk 3	Disk 4
STRIPE 0	00101	11101	01001	01100	
STRIPE 1	11100	01101	10001	01111	01111
STRIPE 2	10101	01100	00100	10001	01100
STRIPE 3	00101		11001	11111	00101
STRIPE 4	00111	11101	11111	00000	00101

1. Να συμπληρωθούν τα δεδομένα στον παραπάνω πίνακα.
2. Να ορισθεί πως θα γίνει η εγγραφή της νέας τιμής 01011 στο STRIPE 1 του Disk 2.
3. Υποθέστε ότι ο Disk 3 καταστράφηκε και το σύστημα εμφανίζει την παρακάτω εικόνα

	Disk 0	Disk 1	Disk 2	Disk 3	Disk 4
STRIPE 0	10100	10101	01001	*****	11001
STRIPE 1	10100	01111	10001	*****	01011
STRIPE 2	10101	01100	01101	*****	01100
STRIPE 3	10110	11001	11101	*****	10100
STRIPE 4	11101	00101	10011	*****	01101

Να εξηγηθεί πως θα γίνει η ανάγνωση-ανάκτηση του STRIPE 3 του κατεστραμμένου Disk 3.

**ΚΑΛΗ ΕΠΙΤΥΧΙΑ!**