



**ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ**  
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ  
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ  
ΕΡΓΑΣΤΗΡΙΟ ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ  
www.cslab.ece.ntua.gr

## **ΠΡΟΗΓΜΕΝΑ ΘΕΜΑΤΑ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ ΥΠΟΛΟΓΙΣΤΩΝ**

Ακ. έτος 2012-2013, 8ο εξάμηνο, Σχολή ΗΜ&ΜΥ

### **2η ΕΡΓΑΣΙΑ**

**Τελική Ημερομηνία Παράδοσης: 26/05/2013**

#### **1. Σκοπός της άσκησης**

Σκοπό της άσκησης αποτελεί η μελέτη σημαντικών χαρακτηριστικών των σύγχρονων superscalar, out-of-order επεξεργαστών και ο τρόπος με τον οποίο επηρεάζουν την απόδοση του συστήματος.

#### **2. Μελέτη με χρήση simulator**

##### **2.1 Εισαγωγή**

Σε αυτό το μέρος της άσκησης θα χρησιμοποιήσετε τον sniper simulator που χρησιμοποιήσατε και στην προηγούμενη άσκηση. Αυτή τη φορά όμως στις προσομοιώσεις θα μεταβάλετε χαρακτηριστικά του datapath του επεξεργαστή και θα μελετήσετε τον τρόπο που επηρεάζεται η απόδοση του επεξεργαστή.

##### **2.2 Config file**

Για να φτιάξετε το config file για τις προσομοιώσεις αυτής της άσκησης θα στηριχτείτε στο config file που χρησιμοποιήσατε στην 1<sup>η</sup> άσκηση. Θα πρέπει να το παραμετροποιήσετε κατάλληλα έτσι ώστε:

- Οι L1 instruction και data caches θα είναι 32KB, 8-way associative με block size 64 bytes.
- Η L2 cache θα είναι 1024KB, 8-way associative με block size 128 bytes.
- Ο branch predictor θα πρέπει να έχει τα εξής χαρακτηριστικά (μπορείτε να δείτε αντίστοιχα entries και στο base.cfg) :
  - type = one\_bit
  - mispredict\_penalty = 17
  - size=1024

Οι παράμετροι τις οποίες θα μελετήσετε είναι οι **dispatch\_width** και **window\_size**. Η πρώτη είναι ο αριθμός των εντολών που μπορούν να γίνουν ταυτόχρονα (δηλαδή το “πόσο” superscalar είναι ο επεξεργαστής μας), ενώ η δεύτερη είναι το μέγεθος του ROB (reorder buffer). Οι παράμετροι αυτοί βρίσκονται στο `[perf_model/core/interval_timer]` τμήμα του config file.

##### **2.3 Μετροπρογράμματα**

Για τους σκοπούς της άσκησης αυτής θα χρησιμοποιήσουμε και πάλι τα PARSEC benchmarks. Πιο συγκεκριμένα, θα χρησιμοποιήσετε τα παρακάτω:

1. parsec-blackscholes
2. parsec-bodytrack

3. parsec-canneal
4. parsec-dedup
5. parsec-fluidanimate
6. parsec-freqmine
7. parsec-swaptions
8. parsec-vips

Όλα τα benchmarks θα πρέπει να εκτελεστούν με input size **test**. Επίσης αυτή τη φορά δεν χρειάζεται να χρησιμοποιήσετε το `periodicins-stat script` μιας και μας ενδιαφέρουν τα στατιστικά από την αρχή έως το τέλος της προσομοίωσης.

## 2.4 Πειραματική Αξιολόγηση

Εκτελέστε όλα τα benchmarks για κάθε διαφορετικό επεξεργαστή που προκύπτει από το συνδυασμό των παρακάτω τιμών για τις παραμέτρους `dispatch_width` και `window_size`:

- `dispatch_width` = 1, 2, 4, 8, 16, 32
- `window_size` = 1, 2, 4, 8, 16, 32, 64, 128, 256, 512

## 2.5 Ζητούμενο

i) Χρειάζεται πραγματικά να προσομοιώσετε και τους 60 διαφορετικούς επεξεργαστές που προκύπτουν με βάση τις παραπάνω τιμές; Αν όχι, εξηγήστε ποιες περιπτώσεις μπορείτε να παραλείψετε και γιατί. Δικαιολογήστε την απάντησή σας όχι μόνο θεωρητικά, αλλά και αποδεικνύοντας για ένα μικρό αριθμό αυτών των περιπτώσεων ότι καλώς τις παραλείψατε.

ii) Πως επηρεάζει η κάθε παράμετρος την απόδοση του επεξεργαστή; Σε ποια συμπεράσματα μπορείτε να καταλήξετε ως προς το σχεδιασμό ενός superscalar, out-of-order επεξεργαστή;

iii) Βρείτε τα αντίστοιχα χαρακτηριστικά (`dispatch_width`, `window_size`) για τον επεξεργαστή του προσωπικού σας υπολογιστή (όχι του VM) ή για κάποιον από τους σύγχρονους επεξεργαστές (π.χ Intel Sandy Bridge). Δικαιολογούνται οι τιμές που επιλέξαν σε αυτά τα συστήματα οι αρχιτέκτονες με βάση τις προσομοιώσεις που εκτελέσατε και τα συμπεράσματα στα οποία καταλήξατε; Θα είχε νόημα να ήταν διαφορετικές (π.χ. μεγαλύτερο `window_size`); Για ποιο λόγο πιστεύετε δεν κάνανε κάποια άλλη επιλογή;

Σαν βασική μετρική απόδοσης θα χρησιμοποιήσετε το IPC (Instructions Per Cycle). Με την προϋπόθεση ότι ο κύκλος μηχανής και ο εκτελούμενος αριθμός εντολών παραμένουν σταθεροί κάθε φορά, μεγαλύτερες τιμές στο IPC υποδεικνύουν καλύτερη απόδοση (σημείωση: αυτό ισχύει μόνο στα πλαίσια της προσομοίωσης. Στην πράξη, οι διάφορες τροποποιήσεις στα μικροαρχιτεκτονικά χαρακτηριστικά του επεξεργαστή επιφέρουν συνήθως αλλαγές και στην διάρκεια του κύκλου ρολογιού).

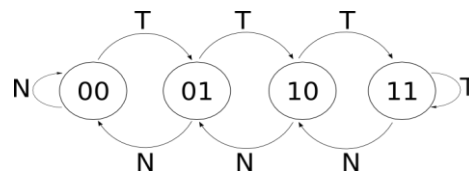
## 3. Άσκηση Tomasulo (Θέμα Σεπτέμβριος 2012)

Δίνεται αρχιτεκτονική η οποία υλοποιεί τον αλγόριθμο Tomasulo χρησιμοποιώντας ROB για in-order commit εντολών. Το pipeline του επεξεργαστή περιέχει τα στάδια Issue (IS), Execute (EX), Write Result (WR) και Commit (CMT), αγνοούμε δηλαδή τα IF και ID. Ισχύουν επίσης τα ακόλουθα :

- Τα IS, WR, CMT απαιτούν 1 κύκλο.
- Για integer αριθμούς το σύστημα περιέχει 4 RS για εντολές διακλάδωσης, αριθμητικές και λογικές εντολές καθώς και 4 non-pipelined functional units. Οι integer εντολές διαρκούν 1 κύκλο.

- Για floating point αριθμούς, το σύστημα περιέχει 1 RS για προσθέσεις/αφαιρέσεις, 1 RS για πολλαπλασιασμούς/διαιρέσεις και αντίστοιχο αριθμό non-pipelined functional units. Οι εντολές πρόσθεσης/αφαίρεσης διαρκούν 1 κύκλο, ενώ οι εντολές πολλαπλασιασμού/διαίρεσης 5 κύκλους.
- Για τις εντολές αναφοράς στη μνήμη, στο στάδιο EX γίνεται τόσο ο υπολογισμός της διεύθυνσης αναφοράς όσο και η προσπέλαση στη μνήμη. Το σύστημα περιλαμβάνει ένα Load και ένα Store Queue, το καθένα από τα οποία διαθέτει 2 θέσεις. Οι εντολές χρησιμοποιούν ένα ξεχωριστό, pipelined functional unit για τον υπολογισμό της διεύθυνσης και διαρκούν 1 κύκλο στην περίπτωση Hit στην cache και 5 κύκλους σε περίπτωση Miss.
- Η πρόβλεψη μιας εντολής διακλάδωσης υπό συνθήκη γίνεται *ταυτόχρονα* με τη δρομολόγηση της εντολής. Ο έλεγχος της πρόβλεψης γίνεται μόλις γίνει γνωστό το αποτέλεσμα της εντολής, δηλαδή στο στάδιο WR (κύκλος k). Σε περίπτωση σφάλματος, σταματά η εκτέλεση των εντολών του miss-predicted execution path και στον επόμενο κύκλο (κύκλος k+1) δρομολογείται η σωστή εντολή.
- Ο ROB έχει 7 θέσεις.
- Το σύστημα περιλαμβάνει 1 CDB. Σε περίπτωση που παραπάνω από μια εντολές θέλουν να το χρησιμοποιήσουν, τότε προτεραιότητα αποκτά η “παλαιότερη” εντολή (αυτή που έγινε issued πρώτη). Θεωρήστε ότι τα branches και τα stores δεν δημιουργούν conflicts στο CDB.
- Για τις εντολές διακλάδωσης υπό συνθήκη, το σύστημα χρησιμοποιεί τον παρακάτω πίνακα από 2-bit predictors. Δίνεται επίσης το FSM διάγραμμα του 2-bit predictor.

Index	Value
0	00
1	11
2	10
3	01



Η δεικτοδότηση του πίνακα γίνεται χρησιμοποιώντας τον κατάλληλο αριθμό low order bits από το PC της εντολής. Ο επεξεργαστής υλοποιεί το ISA του MIPS, οι εντολές απέχουν μεταξύ τους 4 bytes και επομένως κατά τη δεικτοδότηση θα πρέπει να αγνοήσετε τα 2 λιγότερα σημαντικά bits του PC.

- Το σύστημα περιλαμβάνει μια fully associative cache με 2 cache lines και μέγεθος block = 16 bytes, η οποία χρησιμοποιεί πολιτική αντικατάστασης LRU. Αρχικά η cache είναι άδεια.
- Οι καταχωρητές R<sub>1</sub>, R<sub>2</sub> δείχνουν και οι δύο στο πρώτο στοιχείο ενός πίνακα A, ο οποίος περιέχει στοιχεία μήκους 8 bytes και είναι ευθυγραμμισμένος.

Δίνεται ο παρακάτω κώδικας :

```

0x00880004 LOOP: LD      F0, 0(R1)
0x00880008      ADDI    R3, R1, 0x10
0x0088000C      LD      F1, 0(R3)
0x00880010      ADDD    F0, F0, F1          /* Ο αντίστοιχος κώδικας σε C */
0x00880014      DIVD    F0, F0, 0x2        k = 0;
0x00880018      SD      F0, 0(R2)        for(int i=0; i < 6; i+=3) {
0x0088001C      ADDI    R2, R2, 0x8          A[k] = (A[i] + A[i+2])/2;
0x00880020      ADDI    R1, R1, 0x18        k ++ ;
0x00880024      SUBI    R8, R8, 0x1        }
0x00880028      BNEZ    R8, LOOP          A[k] = A[k+2] - A[k+1];
0x0088002C      ADDI    R3, R2, 0x10
0x00880030      LD      F0, 0(R3)
0x00880034      ADDI    R4, R2, 0x8
0x00880038      LD      F1, 0(R4)
0x0088003C      SUBD    F0, F0, F1
0x00880040      SD      F0, 0(R2)

```

Δίνεται επίσης ότι R<sub>8</sub> = 2. Εκτελέστε τον κώδικα και δώστε τους χρόνους δρομολόγησης, εκτέλεσης και ολοκλήρωσης των εντολών σε έναν πίνακα όπως ο παρακάτω :

OP	IS	EX	WR	CMT	Σχόλιο
L.D F0, 0(R1)	1	2-??	??	??	

Στο πεδίο “Σχόλιο” δικαιολογήστε τις καθυστερήσεις μεταξύ σταδίων καθώς και ακυρώσεις εντολών.

*Παραδοτέο της άσκησης θα είναι ένα ηλεκτρονικό κείμενο (pdf, doc ή odt). Στο ηλεκτρονικό κείμενο να αναφέρετε στην αρχή τα στοιχεία σας (Όνομα, Επώνυμο, ΑΜ). Η άσκηση να παραδοθεί ηλεκτρονικά στην ιστοσελίδα:*

<http://www.cslab.ece.ntua.gr/courses/advcomparch/submit>

*Δουλέψτε ατομικά. Έχει ιδιαίτερη αξία για την κατανόηση του μαθήματος να κάνετε μόνοι σας την εργασία. Μην προσπαθήσετε να την αντιγράψετε από άλλους συμφοιτητές σας.*

*Μην αφήσετε την εργασία για το τελευταίο Σαββατοκύριακο, απαιτεί αρκετό χρόνο για την εκτέλεση των benchmarks, ξεκινήστε αμέσως!*