



# 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 2014-2015, 8ο εξάμηνο, Σχολή ΗΜ&ΜΥ

### 2η ΕΡΓΑΣΙΑ

Τελική Ημερομηνία Παράδοσης: 31/05/2015

#### 1. Σκοπός της άσκησης

Αντικείμενο της εργασίας αποτελεί η μελέτη διαφορετικών συστημάτων πρόβλεψης αλμάτων καθώς και η αξιολόγηση τους με δεδομένο το διαθέσιμο χώρο πάνω στο τσιπ. Στο δεύτερο μέρος θα εξεταστούν σημαντικά χαρακτηριστικά των σύγχρονων *superscalar*, *out-of-order* επεξεργαστών και ο τρόπος με τον οποίο επηρεάζουν την απόδοση του συστήματος, την κατανάλωση ενέργειάς του καθώς και το μέγεθος του *chip* του επεξεργαστή.

#### 2. Προσθήκη *branch predictor* στον *sniper*

Η γενική κλάση που υλοποιεί έναν *branch predictor* στον προσομοιωτή ορίζεται στα αρχεία *sniper/common/performance\_model/branch\_predictor.{cc,h}*. Η αρχικοποίηση του *branch predictor* γίνεται στη μέθοδο `create()`. Εκεί, διαβάζεται το όνομα του *predictor* που έχει δοθεί είτε μέσω του *configuration file* είτε μέσω του ορίσματος `-g` της εντολής `run-sniper`.

Για την υλοποίηση ενός νέου *predictor* απαιτείται η δημιουργία μίας υπο-κλάσης της κλάσης **BranchPredictor** και ο ορισμός των δύο μεθόδων `predict()` και `update()`. Η πρώτη συνάρτηση δέχεται ως ορίσματα το PC της εντολής και τη διεύθυνση προορισμού και καλείται να προβλέψει αν το άλμα θα εκτελεστεί ή όχι (Taken / Not Taken). Η δεύτερη μέθοδος καλείται να αποθηκεύσει τις πληροφορίες εκείνες που απαιτούνται για τις μελλοντικές προβλέψεις. Τα ορίσματα της είναι η πρόβλεψη που έκανε ο *predictor*, το πραγματικό αποτέλεσμα της εντολής διακλάδωσης, το PC της εντολής και η διεύθυνση προορισμού.

Για την εκτέλεση της άσκησης, δίνεται υλοποιημένος ένας n-bit *predictor* καθώς και το αρχείο *branch\_predictor.cc* το οποίο περιέχει όλες τις απαραίτητες προσθήκες προκειμένου να είναι δυνατή η προσομοίωση του n-bit *predictor*. Για να μπορέσετε να χρησιμοποιήσετε τον n-bit *predictor* (καθώς και τους υπόλοιπους *predictors* που θα υλοποιήσετε) θα πρέπει μετά τις αλλαγές στο *branch\_predictor.cc* να κάνετε `make` στον φάκελο *sniper* ώστε να μεταγλωττιστούν εκ νέου τα αρχεία του προσομοιωτή.

#### 3. Μετροπρογράμματα

Για τους σκοπούς αυτής της άσκησης θα χρησιμοποιήσετε το περιβάλλον προσομοίωσης του “Sniper Multicore Simulator” και τα PARSEC benchmarks, όπως και στην πρώτη άσκηση. Πιο συγκεκριμένα, θα χρησιμοποιήσετε τα παρακάτω:

1. parsec-blackscholes
2. parsec-canoe

3. parsec-fluidanimate
4. parsec-freqmine
5. parsec-streamcluster
6. parsec-swaptions
7. parsec-vips

Και στα δύο μέρη της άσκησης θα χρησιμοποιήσετε **small** size.

## 4. McPAT

Το McPAT (**M**ulti-**c**ore **P**ower, **A**rea, **T**iming) είναι ένα εργαλείο το οποίο μοντελοποιεί χαρακτηριστικά ενός επεξεργαστή, όπως η κατανάλωση ενέργειας και το μέγεθος που καταλαμβάνουν στο τσιπ οι διαφορετικές δομικές μονάδες του επεξεργαστή. Στην υπάρχουσα εγκατάσταση του *sniper* περιλαμβάνεται το McPAT και μπορεί να χρησιμοποιηθεί για την εξαγωγή αντίστοιχων στατιστικών μέσα από τον φάκελο μίας προσομοίωσης που έχει ολοκληρωθεί. Στο βοηθητικό κώδικα της άσκησης, δίνεται επιπλέον το script *advcomparch\_mcpat.py* το οποίο είναι μία τροποποιημένη έκδοση του *mcpat.py* που περιέχεται στον *sniper*. Αφού αντιγράψετε το *advcomparch\_mcpat.py* στον φάκελο *scripts* μπορείτε να το χρησιμοποιήσετε ως εξής:

```
user@snf-xxx:~/advcomparch$ advcomparch_mcpat.py -h
Usage: /home/user/advcomparch/sniper/tools/advcomparch_mcpat.py [-h (help)] [-j <jobid> | -d
<resultsdir (default: .)>] [-t <type: total|dynamic|static|peak|peakdynamic|area>] [-c <override-
config>] [-o <output-file (power{.png, .txt, .py})>]
```

Από τις παραμέτρους που δέχεται αυτό το script μας ενδιαφέρουν κυριώς οι εξής:

- `-d <resultsdir>` : ο φάκελος της προσομοίωσης.
- `-t <type>` : το είδος των στατιστικών που θέλουμε να εξάγουμε.
- `-o <output-file>` : το όνομα που θα έχουν τα αρχεία που θα παραχθούν.

Η εκτέλεση του script παράγει 4 αρχεία, τα **power.png**, **power.py**, **power.txt** και **power.xml**. Στο πρώτο περιέχεται η γραφική αναπαράσταση (με μπάρα) των διαφόρων στατιστικών. Το δεύτερο περιλαμβάνει όλα τα στατιστικά που έχουν παραχθεί, σε μορφή κατάλληλη ώστε να μπορεί να χρησιμοποιηθεί απευθείας σε python script, ενώ το τρίτο περιλαμβάνει τα ίδια στατιστικά σε μορφή κειμένου. Το .xml αρχείο είναι το αρχείο που παράγεται από τον *sniper* και δίνεται ως είσοδος στο McPAT.

Εκτός από τα προαναφερθέντα αρχεία το *advcomparch\_mcpat.py* τυπώνει στην οθόνη (ή όπου κάνετε redirect την έξοδο) συνοπτικά τα στατιστικά που ορίσαμε με το όρισμα `-t`. Ένα παράδειγμα χρήσης του δίνεται παρακάτω:

```
user@snf-xxx:~/advcomparch$ advcomparch_mcpat.py -d bench_dir -t total -o bench_dir/power >
bench_dir/power.total.out
user@snf-xxx:~/advcomparch$ ll bench_dir
total 172
drwxrwxr-x    2 user user  4096 Μάι  11 13:24 ./
drwxrwxr-x 458 user user 36864 Μάι  11 13:08 ../
-rw-rw-r--    1 user user   6602 Μάι  11 13:13 power.png
-rw-rw-r--    1 user user  18287 Μάι  11 13:13 power.py
-rw-rw-r--    1 user user  12572 Μάι  11 13:13 power.total.out
-rw-rw-r--    1 user user  12572 Μάι  11 13:13 power.txt
-rw-rw-r--    1 user user  19221 Μάι  11 13:13 power.xml
-rw-rw-r--    1 user user   6121 Μάι   7 21:02 sim.cfg
-rw-rw-r--    1 user user   2481 Μάι   7 21:11 sim.info
-rw-rw-r--    1 user user   1900 Μάι   7 21:11 sim.out
-rw-r--r--    1 user user  45056 Μάι   7 21:11 sim.stats.sqlite3
```

```
user@snf-xxx:~/advcomparch$ cat bench_dir/power.total.out
```

	Power	Energy	Energy %
core-core	10.15 W	0.92 J	41.45%
core-ifetch	1.78 W	0.16 J	7.26%
core-alu	0.40 W	0.04 J	1.65%
core-int	1.07 W	0.10 J	4.37%
core-fp	1.05 W	0.10 J	4.30%
core-mem	0.63 W	0.06 J	2.59%
core-other	0.95 W	0.09 J	3.89%
icache	0.82 W	0.08 J	3.37%
dcache	1.36 W	0.12 J	5.56%
l2	1.98 W	0.18 J	8.09%
l3	0.00 W	0.00 J	0.00%
nuca	0.00 W	0.00 J	0.00%
noc	0.02 W	2.26 mJ	0.10%
dram	4.25 W	0.39 J	17.37%
core	16.04 W	1.46 J	65.52%
cache	4.17 W	0.38 J	17.02%
total	24.49 W	2.23 J	100.00%

## 5. Energy-Delay Product

Παραδοσιακά, για την αξιολόγηση της κατανάλωσης ενός επεξεργαστή χρησιμοποιείται ως μετρική η συνολική κατανάλωση ενέργειας σε Joules. Ωστόσο, πολλές φορές απαιτείται η μελέτη της επίδρασης διαφόρων χαρακτηριστικών του επεξεργαστή όχι μόνο στην κατανάλωση αλλά ταυτόχρονα και στην επίδοσή του. Για το σκοπό αυτό έχει προταθεί ως μετρική το energy-delay product (EDP). Το EDP για την εκτέλεση ενός benchmark ορίζεται ως το γινόμενο της ενέργειας επί τον χρόνο εκτέλεσης του benchmark:

$$EDP = Energy(J) * runtime(sec)$$

Αντίστοιχα, αν θέλουμε να δώσουμε περισσότερο βάρος στον χρόνο εκτέλεσης μπορούμε να υψώσουμε το *runtime* στο τετράγωνο, στον κύβο κ.λ.π. Έτσι προκύπτουν τα  $ED^2P$ ,  $ED^3P$  και ούτω καθεξής:

$$ED^2P = Energy(J) * runtime^2(sec)$$

$$ED^3P = Energy(J) * runtime^3(sec)$$

## 6. Μελέτη Branch Predictors

### 6.1. n-bit predictors

Θα μελετήσετε την απόδοση των n-bit predictors, χρησιμοποιώντας την υλοποίηση που σας παρέχουμε (**nbit\_branch\_predictor.h**). Η συγκεκριμένη υλοποίηση του predictor απαιτεί τον ορισμό στο config file του μήκους του counter καθώς και των bits που απαιτούνται για το indexing (ίσα με  $\log_2(\text{entries})$ ).

- (i) Διατηρώντας σταθερό τον αριθμό των BHT entries και ίσο με 16K, προσομοιώστε τους n-bit predictors, για  $N=1, 2, \dots, 7$ . Συγκρίνετε τους predictors χρησιμοποιώντας τα direction Mispredictions Per Thousand Instructions (direction MPKI) καθώς και τα Instructions Per Cycle (IPC).
- (ii) Στο προηγούμενο ερώτημα η αύξηση του αριθμού των bits ισοδυναμούσε με αύξηση του απαιτούμενου hardware, αφού κρατούσαμε σταθερό τον αριθμό των entries του BHT. Διατηρώντας τώρα σταθερό το hardware και ίσο με 32K bits, επαναλάβετε τις προσομοιώσεις για τα 7 benchmarks, θέτοντας  $N=1,2,4$  και τον κατάλληλο αριθμό entries. Δώστε το κατάλληλο διάγραμμα και εξηγήστε τις μεταβολές που παρατηρείτε. Ποιον predictor θα διαλέγατε ως την βέλτιστη επιλογή;

## 6.2. Σύγκριση διαφορετικών predictors

Στο κομμάτι αυτό θα συγκρίνετε τους παρακάτω predictors (οι predictors σε **bold** δε δίνονται):

1. **Static Not-Taken**
2. **Static BTFNT (Backward Taken Forward Not Taken)**
3. n-bit predictor (η επιλογή που κάνατε στο 4.1(ii))
4. Pentium-M predictor (το hardware overhead είναι περίπου 30K)
5. **Local-History two-level predictors** (βλ. διαφάνειες μαθήματος) με τα εξής χαρακτηριστικά :
  - PHT entries = 8192
  - PHT n-bit counter length = 2
  - BHT entries = X
  - BHT entry length = Z

Υπολογίστε το Z ώστε το απαιτούμενο hardware να είναι σταθερό και ίσο με 32K, όταν X = 2048 και X = 4096.

6. **Global History two-level predictors** με τα εξής χαρακτηριστικά:

- PHT entries = Z
- PHT n-bit counter length = X
- BHR length = 4, 8

Υπολογίστε το Z ώστε το απαιτούμενο hardware να είναι σταθερό και ίσο με 32K όταν X = 2 και X = 4. Το κόστος του Branch History Register (4 και 8 bits) θεωρείται αμελητέο.

7. **Tournament Hybrid predictors** (βλ. διαφάνειες μαθήματος) με τα εξής χαρακτηριστικά:

- Ο meta-predictor M είναι ένας 2-bit predictor με 512 entries (το overhead του μπορείτε να το αγνοήσετε στην ανάλυση σας).
- Οι P<sub>0</sub>, P<sub>1</sub> μπορούν να είναι n-bit, local-history ή global-history predictors.
- Οι P<sub>0</sub>, P<sub>1</sub> έχουν overhead 16K ο καθένας.
- Υλοποιήστε **τουλάχιστον 2** διαφορετικούς tournament predictors.

Προσομοιώστε και συγκρίνετε τους παραπάνω (τουλάχιστον 12) predictors. Δώστε τα κατάλληλα διαγράμματα. Ποιον predictor θα διαλέγατε τελικά να υλοποιήσετε;

## 7. Μελέτη dispatch width και window size

Σε αυτό το μέρος της άσκησης θα χρησιμοποιήσετε τον sniper simulator και στις προσομοιώσεις αλλάζοντας χαρακτηριστικά του datapath του επεξεργαστή και θα μελετήσετε τον τρόπο που επηρεάζεται η απόδοση του. Επίσης, θα χρησιμοποιήσετε το McPAT για να μελετήσετε την μεταβολή της κατανάλωσης κάθε επεξεργαστή καθώς και το μέγεθός του.

### 7.1. Config file

Για αυτό το κομμάτι της άσκησης θα χρησιμοποιήσετε το config file που χρησιμοποιήσατε στην 1<sup>η</sup> άσκηση. Συγκεκριμένα:

- Οι L1 instruction και data caches θα είναι 64KB, 8-way associative με block size 64 bytes.
- Η L2 cache θα είναι 1024KB, 8-way associative με block size 64 bytes.
- Ο branch predictor θα πρέπει να έχει τα εξής:
  - type = one\_bit
  - mispredict\_penalty = 14
  - 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 issue 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.

## 7.2. 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

- (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) Πώς 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) Πώς επηρεάζει η κάθε παράμετρος την κατανάλωση ενέργειας και το μέγεθος του τσιπ;
- (iv) 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 ή Haswell). Δικαιολογούνται οι τιμές που επιλέξατε 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 κάνατε κάποια άλλη επιλογή;

Παραδοτέο θα είναι ένα ηλεκτρονικό κείμενο (pdf, doc ή odt) που θα περιέχει την αναφορά με τα διαγράμματα και τα συμπεράσματά σας, καθώς και τον κώδικα που υλοποιήσατε. Μη ξεχάσετε να αναφέρετε τα στοιχεία σας (Όνομα, Επώνυμο, ΑΜ). Η άσκηση θα παραδοθεί μόνο 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 την αντιγράψετε απλά από άλλους συμφοιτητές σας.*