



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 2013-2014, 8ο εξάμηνο, Σχολή ΗΜ&ΜΥ

2η ΕΡΓΑΣΙΑ

Τελική Ημερομηνία Παράδοσης: 27/07/2014

1. Σκοπός της άσκησης

Αντικείμενο της εργασίας αποτελεί η μελέτη διαφορετικών συστημάτων πρόβλεψης καθώς και η αξιολόγηση διαφορετικών επιλογών για την υλοποίηση του μηχανισμού πρόβλεψης 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.

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 θα χρησιμοποιηθεί από το configuration αρχείο που έχει δοθεί με το όρισμα `-c` κατά την εκτέλεση του προσομοιωτή και καλείται ο κατάλληλος constructor του predictor.

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

Για τη μελέτη των branch predictors θα χρησιμοποιήσετε **small** size ενώ για τα window και dispatch size **test** size.

4. Μελέτη Branch Predictors

4.1. 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 8 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;

4.2. Σύγκριση διαφορετικών predictors

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

1. **Static Not-Taken**
2. **Static BTNT (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 P₀, P₁ μπορούν να είναι n-bit, local-history ή global-history predictors.
 - Οι P₀, P₁ 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;

5. Μελέτη dispatch width και window size

Σε αυτό το μέρος της άσκησης θα χρησιμοποιήσετε τον sniper simulator και στις προσομοιώσεις θα μεταβάλετε χαρακτηριστικά του datapath του επεξεργαστή και θα μελετήσετε τον τρόπο που επηρεάζεται η απόδοση του επεξεργαστή.

5.1. Config file

Για να φτιάξετε το config file για τις προσομοιώσεις αυτής της άσκησης θα στηριχτείτε στο config file που χρησιμοποιήσατε στην 1^η 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 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.

5.2. 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;

- (ii) 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 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 κάνανε 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.