

ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΥΠΟΛΟΓΙΣΤΩΝ

Χειμερινό Εξάμηνο 2022/2023

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ & ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ

3^η Εργαστηριακή Άσκηση

Energy-Delay-Area Product Optimization (gem5 + McPAT)

Η εργασία αυτή αποτελεί συνέχεια της δεύτερης εργαστηριακής άσκησης. Θα χρειαστεί να χρησιμοποιήσετε τα αποτελέσματα που έχουν προκύψει από τα πειράματα που έχετε εκτελέσει και για τούτο θα παρατηρήσετε ότι αναφέρεται ως Μέρος Τρίτο. Συνεπώς, αν για κάποιο λόγο δεν έχετε ολοκληρώσει τη δεύτερη εργαστηριακή άσκηση, καλείστε να την ολοκληρώσετε προτού προχωρήσετε σε αυτήν εδώ.

ΜΕΡΟΣ ΤΡΙΤΟ

Βήμα 1°. Εξοικείωση με τον εξομοιωτή McPAT

Το McPAT¹ είναι ένα εργαλείο που μπορεί να μοντελοποιήσει αρχιτεκτονικές επεξεργαστών και να παράσχει εκτιμήσεις για τη κατανάλωση ισχύος τους, το μέγεθος ολοκλήρωσης τους και τον χρονισμό τους. Ως είσοδο δέχεται μια περιγραφή της μικροαρχιτεκτονικής ενός συστήματος μέσω μιας σειράς από παραμέτρους με βάση την οποία κατασκευάζει ένα μοντέλο. Επιπρόσθετα λαμβάνει ως είσοδο και μια σειρά από στατιστικά στοιχεία τα οποία χρησιμοποιεί για να μπορεί να παράξει εκτιμήσεις για την κατανάλωση του συστήματος συνολικά και κάθε κομματιού του ξεχωριστά. Το εργαλείο έχει τη δυνατότητα να μοντελοποιήσει κυκλώματα τεχνολογίας από 90nm έως και 22nm ενώ μεταγενέστερες εργασίες έχουν προσπαθήσει να το τροποποιήσουν ώστε να μπορεί να μοντελοποιήσει και κυκλώματα ακόμα μικρότερης κλίμακας και τεχνολογιών FinFET πέρα από Bulk CMOS.

Στο link που παρέχεται από την HP Labs στο κάτω μέρος αυτής της σελίδας, θα βρείτε τα βασικά papers που περιγράφουν το McPAT. Συνιστάται να τα διαβάσετε ώστε να αποκτήσετε μια όσο το δυνατόν πληρέστερη εικόνα για το εργαλείο.

Για τη συγκεκριμένη εργασία θα χρησιμοποιήσετε μία έκδοση του McPAT που έχει τροποποιηθεί από εμάς. Για να την κατεβάσετε θα πρέπει να κάνετε clone το σχετικό repository:

```
$ git clone https://github.com/kingmouf/cmcpat.git my_mcpat
```

Για να κάνετε build το εργαλείο μπείτε στο φάκελο mcpat και πατήστε απλά make. Είναι πολύ πιθανό ανάλογα την έκδοση του gcc/g++ που διαθέτετε να παρουσιαστούν κάποια θέματα κατά το make ή κάποια λάθη που μπορεί να οδηγήσουν στην αποτυχία του build. Στη περίπτωση αυτή θα χρειαστεί να εγκαταστήσετε τα ακόλουθα πακέτα (υποθέτοντας ότι χρησιμοποιείται κάποιο Ubuntu-based σύστημα):

```
$ sudo apt install libc6-dev-i386  
$ sudo apt install gcc-7-multilib g++-7-multilib
```

Στο σύστημα ανάπτυξης που χρησιμοποιήσαμε για την ανάπτυξη της εργασίας ήταν εγκατεστημένη η έκδοση 7.4 του gcc/g++ για τούτο και το «7» στην τελευταία εντολή. Αν χρησιμοποιείτε κάποια άλλη version του gcc/g++ αλλάξτε κατάλληλα τον αριθμό αυτό.

Μέσα στο φάκελο ProcessorDescriptionFiles θα βρείτε μια σειρά από αρχεία xml. Τα αρχεία αυτά χρησιμοποιούνται ως αρχεία εισόδου στο McPAT και είναι περιγραφές επεξεργαστών. Χρησιμοποιείτε κάποιο από αυτά που αναφέρονται στους επεξεργαστές Xeon, Niagara, ARM A9 για να δείτε πώς τρέχει το εργαλείο και τι αποτελέσματα παράγει. Για παράδειγμα για να τρέξετε το McPAT για τον επεξεργαστή Xeon χρησιμοποιείτε την ακόλουθη εντολή:

¹ McPAT - Multicore Power, Area, and Timing, <http://www.hpl.hp.com/research/mcpat/>

```
$ ./mcpat -infile ProcessorDescriptionFiles/Xeon.xml -print_level 1
```

Μπορείτε να αλλάξετε το `print_level` από 1 μέχρι 5. Όσο μεγαλύτερο νούμερο, τόσο πιο αναλυτικά αποτελέσματα θα σας βγάλει.

Αν ανοίξετε το σχετικό αρχείο `xml` θα παρατηρήσετε ότι υπάρχουν δύο ειδών εγγραφές:

- Παράμετροι (`param`) και
- Στατιστικά (`stats`)

Οι παράμετροι χρησιμοποιούνται για να προσδιορίσουν το σύστημα που θέλετε να μοντελοποιήσετε και να ορίσουν τα βασικά χαρακτηριστικά του. Από την άλλη τα στατιστικά χρησιμοποιούνται για να ποσοτικοποιήσουν τη χρήση κάθε μονάδας (π.χ. πόσες φορές έγινε `access` μια μνήμη κατά την εκτέλεση ενός προγράμματος).

Ερωτήματα

1. Κοιτάζοντας τα αποτελέσματα που σας δίνει η έξοδος του McPAT, θα δείτε αποτελέσματα για **dynamic power** και **leakage**. Ανατρέξτε στη βιβλιογραφία και προσπαθήστε να εξηγήσετε τί είναι το ένα και τί είναι το άλλο. Αν τρέξετε διαφορετικά προγράμματα σε έναν επεξεργαστή ποιο θα επηρεαστεί και πώς; Έχει σημασία πόσο μεγάλο (σε χρονική διάρκεια εκτέλεσης) είναι ένα πρόγραμμα;
2. Έστω ότι έχετε ένα σύστημα που τροφοδοτείται από μπαταρία συγκεκριμένης χωρητικότητας και έχετε την επιλογή να βάλετε στο σύστημα αυτό δύο διαφορετικούς επεξεργαστές. Έστω ότι ο ένας καταναλώνει 5 Watt και ο άλλος 50 Watt. Υπάρχει περίπτωση ο δεύτερος επεξεργαστής να δίνει στο σύστημα μεγαλύτερη διάρκεια μπαταρίας; Το McPAT μπορεί να μας δώσει τη σχετική απάντηση με τα αποτελέσματα που παράγει; Αν ναι γιατί και αν όχι τι άλλο θα χρειαζόμασταν για να μπορέσουμε να απαντήσουμε;
3. Τρέξτε το McPAT για τον Xeon (`Xeon.xml`) και για τον ARM A9 (`ARM_A9_2GHz.xml`). Ας κάνουμε μια εντελώς αυθαίρετη υπόθεση και ας υποθέσουμε ότι ο Xeon μπορεί να τρέξει την ίδια εφαρμογή με τον A9, 50 φορές γρηγορότερα. Υποθέτοντας ότι δεν διακόπτεται η λειτουργία του συστήματος μετά την ολοκλήρωση εκτέλεσης της εφαρμογής (δλδ δεν απενεργοποιείται με κάποιο τρόπο το σύστημα), εξηγήστε χρησιμοποιώντας το McPAT γιατί ο Xeon δεν μπορεί να είναι πιο *energy efficient* από τον A9 παρά τη διαφορά στην απόδοση.

Βήμα 2°. gem5 + McPAT: αναζητώντας τη βελτιστοποίηση του γινομένου EDP

Για να μπορέσει το McPAT να υπολογίσει τις δυναμικές πτυχές ενός συστήματος κατά την εκτέλεση ενός προγράμματος χρειάζεται δεδομένα που πρέπει να προέλθουν από την εκτέλεση του προγράμματος. Στη βιβλιογραφία, ένα πρόγραμμα που μπορεί να προσφέρει τέτοιου είδους δεδομένα ονομάζεται *performance simulator* και ο `gem5` είναι ένας τέτοιος εξομοιωτής.

Παρόλα αυτά, η έξοδος του `gem5` και η είσοδος του McPAT δεν είναι σε συμβατή μορφή. Χρειάζεται λοιπόν ένας τρόπος να αντληθούν τα αποτελέσματα της εκτέλεσης ενός προγράμματος στο `gem5` (καθώς και οι βασικές πληροφορίες που περιγράφουν το σύστημα που εξομοίωσε ο `gem5`) ώστε να παραχθεί το κατάλληλο αρχείο `xml` εισόδου που χρειάζεται το McPAT. Στο φάκελο `Scripts` του repository που κάνατε `clone` με το McPAT υπάρχει ένα πρόγραμμα `rython` που χρησιμοποιείται για αυτό ακριβώς το σκοπό.

Για να λειτουργήσει αυτό το πρόγραμμα χρειάζεται ένα βασικό template xml αρχείο που μοντελοποιεί τα βασικά στοιχεία του μοντέλου CPU που εκτελεί ο gem5 μαζί με έναν αριθμό από πληροφορίες σχετικά με την τεχνολογία υλοποίησης (π.χ. το μέγεθος της λιθογραφίας). Τις παραμέτρους αυτού του μοντέλου (π.χ. το μέγεθος της κάθε cache, το associativity, τον αριθμό των execution units κτλ) μπορεί να τις πάρει από το αρχείο config.json που παράγει ο gem5 ενώ τα στατιστικά από την εκτέλεση του προγράμματος μπορεί να τα αντλήσει από το αρχείο stats.txt. Αυτό που θα παράξει είναι ένα αρχείο xml που μπορεί να δοθεί ως είσοδος στο McPAT.

Ανατρέξτε στο φάκελο Scripts και βρείτε τις σχετικές πληροφορίες για το πώς μπορείτε να εκτελέσετε το πρόγραμμα GEM5ToMcPAT.py. Μέσα στο φάκελο ProcessorDescriptionFiles θα βρείτε ένα αρχείο με το όνομα inorder_arm.xml. Αυτό το αρχείο έχει φτιαχτεί για να μπορείτε να το χρησιμοποιήσετε ως template για τα μοντέλα minorCPU του gem5 που χρησιμοποιήσατε στη προηγούμενη άσκηση.

Αν ανοίξετε το template xml αρχείο θα παρατηρήσετε ότι οι τιμές (value) που αποδίδονται σε ένα μεγάλο αριθμό από params αντί να έχουν κάποια συγκεκριμένη τιμή έχουν ένα string που ξεκινάει με το όνομα config. Αυτό σημαίνει ότι το πρόγραμμα μετατροπής της εξόδου του gem5 σε είσοδο του McPAT θα αναζητήσει τις σχετικές πληροφορίες στα κατάλληλα πεδία του config.json αρχείου. Αντίστοιχα, σε stat καταχωρήσεις, οι τιμές value παραπέμπουν σε τιμές του αρχείου stats.txt.

Ερωτήματα

Στη δεύτερη άσκηση σας ζητήθηκε να προσπαθήσετε να βελτιστοποιήσετε τον επεξεργαστή με βάση μια συνάρτηση κόστους που θα φτιάχνατε. Στην άσκηση αυτή πρέπει να χρησιμοποιήσετε τα αποτελέσματα που πήρατε από την εκτέλεση των πειραμάτων της δεύτερης άσκησης και σας ζητείται να προσπαθήσετε να δώσετε τη καλύτερη λύση που να βελτιστοποιεί το γινόμενο Energy-Delay-Area-Product (EDAP) ή Energy-Delay -Product (EDP). Τρέξτε το McPAT με print_level 5 και χρησιμοποιήστε τα αποτελέσματα για το core και για την L2 (αθροιστικά). Χρησιμοποιήστε τα Area, Runtime Dynamic, Subthreshold Leakage και Gate Leakage.

1. Υπολογίστε την κατανάλωση ενέργειας για όλες τις περιπτώσεις.
2. Παρουσιάστε με σχετικά γραφήματα τα αποτελέσματα σας. Δείξτε την επίδραση κάθε παραμέτρου που αλλάξατε στη προηγούμενη άσκηση (cache size, associativity, block size) σε σχέση με το power (χρησιμοποιήστε το peak power για αυτά τα γραφήματα).
3. Αν μπορούσατε να συνδυάσετε χαρακτηριστικά από τις διαφορετικές αρχιτεκτονικές τις οποίες έχετε δει στα πλαίσια των εργαστηρίων, τι θα επιλέγατε από την κάθε μια για δημιουργήσετε την «ιδανική» αρχιτεκτονική?

Ολοκληρώνοντας τις μετρήσεις σας και την ανάλυση σας, θα θέλαμε να σκεφτείτε και να παρουσιάσετε πιθανούς λόγους γιατί τα αποτελέσματα σας μπορεί να μην είναι αξιόπιστα ή η ανάλυση που πραγματοποιήσατε να έχει αυξημένο βαθμό σφάλματος. Πού μπορεί να οφείλονται σημαντικές αποκλίσεις στα αποτελέσματα των εξομοιώσεων και στα πραγματικά συστήματα; Η χρήση δύο διαφορετικών προγραμμάτων για την εκτίμηση διαφορετικών πτυχών του ίδιου συστήματος μπορεί να είναι πηγή επιπλέον σφαλμάτων; Αν ναι, γιατί;

Ανατρέξτε στη σχετική βιβλιογραφία και προσπαθήστε να δικαιολογήσετε τις απαντήσεις σας.

Μικρή παρατήρηση: το McPAT τυπώνει κατευθείαν την έξοδο του στην οθόνη και όχι σε κάποιο αρχείο. Αν εσείς θέλετε να κρατήσετε τα αποτελέσματα σε αρχεία για να τα επεξεργαστείτε στη

```
$ ./mcpat -infile my_cpu.xml -print_level 1 > results.txt
```

συνέχεια μπορείτε να χρησιμοποιήσετε ένα redirect σε αρχείο κατά την κλήση του McPAT. Έτσι αν γράψετε:

τα αποτελέσματα αντί για την οθόνη θα γραφτούν στο αρχείο results.txt.

Παραδοτέα

Χρησιμοποιήστε τον λογαριασμό στο GitHub που δημιουργήσατε στις προηγούμενες ασκήσεις και δημιουργήστε ένα νέο repository για την άσκηση αυτή. Ανεβάστε:

1. ό,τι κώδικες και αρχεία αποτελεσμάτων έχουν προκύψει από τα ερωτήματα που σας ζητούνται να απαντήσετε. Για τους κώδικες που θα ανεβάσετε απαιτείται να έχετε οπωσδήποτε επαρκή σχόλια που να επεξηγούν τι έχετε κάνει και αν αυτό απαιτείται κάποιο documentation.
2. μια αναλυτική αναφορά με τις απαντήσεις στα ερωτήματα. Η αναφορά θα είναι γραμμένη σε ένα αρχείο README.md που θα είναι στο top level του repository σας. Η αναφορά σας θα πρέπει να περιλαμβάνει οπωσδήποτε τις πηγές που χρησιμοποιήσατε για να ολοκληρώσετε την εργασία σας.

ΣΗΜΑΝΤΙΚΟ: Όταν ολοκληρώσετε επιτυχώς την εργασία σας, μπορείτε να πάρετε έναν έξτρα βαθμό, γράφοντας μια κριτική για την εργασία αυτή καθώς και συνολικά για το πώς κρίνετε όλες τις ασκήσεις τώρα που ολοκληρώθηκαν και το εργαστήριο. Επικεντρωθείτε στο αν μάθατε κάτι, που δυσκολευτήκατε και αν ήταν υπερβολικά απλές οι εργασίες, καθώς και αν σας έχει κινηθεί το ενδιαφέρον να ασχοληθείτε περαιτέρω με την αρχιτεκτονική υπολογιστών. Βάλτε αυτή την κριτική μέσα στην αναφορά σας που είναι στο README.md

Για την παράδοση της άσκησης σας, θα κάνετε submit μέσω του elearning την διεύθυνση του repository σας (όχι τον κώδικα ή οτιδήποτε άλλο ΜΟΝΟ την διεύθυνση στο github).

Να έχετε υπόψη σας τα εξής:

1. εφόσον όλα τα repositories είναι public, όλοι μπορούν να δουν το κώδικα σας και την εργασία σας. Αυτό σημαίνει ότι θα πρέπει να φροντίσετε να είναι προσεγμένα (θα κριθείτε και για την ποιότητα της εργασίας όσον αφορά την παρουσίαση της)
2. με την ίδια λογική εφόσον όλα είναι public γίνεται προφανές ότι είναι εξαιρετικά εύκολο να αναδειχθούν οι αντιγραφές. Καλό θα ήταν λοιπόν να το λάβετε αυτό υπόψη σας.

ΚΑΛΗ ΕΠΙΤΥΧΙΑ