ΗΥ-225: Οργάνωση Υπολογιστών
Άνοιξη 2009

Τμ. Επ. Υπολογιστών
© Πανεπιστήμιο Κρήτης

Σειρά Ασκήσεων 14:
Επίδοση Επεξεργαστών

Προθεσμία έως Δευτέρα 27 Απριλίου 2009, πριν το μάθημα (βδομάδα 11)

Άσκηση 14.1: Επίδοση Επεξεργαστών

Βιβλίο (Ελλ. εκδ.): διαβάστε τις § 4.1 - 4.3 (σελ. 258-276), και σελ. 348-349· επίσης, εγκυκλοπαιδικά, είναι καλό να διαβάστε και το υπόλοιπο κεφ. 4 (σελ. 277-299). Η επίδοση (performance) ενός υπολογιστή, κατά την εκτέλεση δοθέντος προγράμματος, είναι αντίστροφα ανάλογη προς τον χρόνο εκτέλεσης αυτού του προγράμματος σε αυτόν τον υπολογιστή. Η επίδοση ενός υπολογιστή, γενικά και αόριστα, ανεξαρτήτως εκτελουμένου προγράμματος, δεν μπορεί να οριστεί επακριβώς και επιστημονικά, μπορεί δε να ποικίλλει ευρέως ανάλογα με τα χαρακτηριστικά των διαφόρων προγραμμάτων. Ο οργανισμός "SPEC" (System Performance Evaluation Cooperative) ιδρύθηκε το 1989 από μερικές εταιρείες υπολογιστών, και ορίζει ένα σύνολο προγραμμάτων (SPEC benchmarks) που χρησιμοποιούνται από κοινού στη βιομηχανία γιά τη μέτρηση των επιδόσεων των υπολογιστών.

Εάν ο υπολογιστής *A* εκτελεί ένα δοθέν πρόγραμμα σε χρόνο *tA*, ο δε υπολογιστής *B* το εκτελεί σε χρόνο *tB*, όπου *tB > tA* και *(tB / tA) = 1.xy*, τότε λέμε ότι "ο υπολογιστής Α είναι ταχύτερος του Β κατά *xy %* γιά το δοθέν πρόγραμμα". Παραδείγματος χάριν, αν tA = 4s και tB = 5s, τότε (tB/tA) = 1.25, και ο Α είναι ταχύτερος του Β κατά 25 % γιά το δοθέν πρόγραμμα. Ο χρόνος t_{exec} εκτέλεσης ενός προγράμματος σ' έναν υπολογιστή μπορεί συχνά να εκφραστεί σαν:

$$t_{exec} = N_{instructions} * CPI_{average} * T_{clock}$$

όπου $N_{instructions}$ είναι το πλήθος (ο αριθμός) των εντολών που ο υπολογιστής εκτελεί προκειμένου να ολοκληρωθεί η εκτέλεση του δοθέντος προγράμματος, $CPI_{average}$ είναι το μέσο πλήθος (μέσος αριθμός) των κύκλων ρολογιού που απαιτούνται γιά την εκτέλεση μιάς εντολής (Cycles Per Instruction --CPI), και T_{clock} είναι ο χρόνος που διαρκεί ένας κύκλος ρολογιού, δηλαδή η περίοδος του ρολογιού, δηλαδή το αντίστροφο της συχνότητας ρολογιού.

Ερώτηση: Θεωρήστε έναν υπολογιστή Α (τύπου RISC), που γιά να τελειώσει ένα δοθέν πρόγραμμα πρέπει να εκτελέσει 2,400,000 εντολές, με μέσο CPI = 3.5 κύκλους ρολογιού ανά εντολή, και με ρολόϊ 600 MHz. Ενας άλλος υπολογιστής Β (τύπου CISC --complex instruction set computer) έχει πιό "πλούσιο" ρεπερτόριο εντολών, κι έτσι του αρκεί να εκτελέσει μόνο 1,800,000 εντολές γιά να τελειώσει το ίδιο πρόγραμμα. Ομως, λόγω της αυξημένης πολυπλοκότητάς του, έχει μέσο CPI = 5.2 κύκλους ρολογιού ανά εντολή, και ρολόϊ 500 MHz. Πόσους κύκλους ρολογιού και πόσα δευτερόλεπτα χρειάζεται ο κάθε υπολογιστής γιά να εκτελέσει το δοθέν πρόγραμμα; Ποιός από τους δύο υπολογιστές είναι ταχύτερος από τον άλλον γιά το δοθέν πρόγραμμα, και πόσο ταχύτερος;

Άσκηση 14.2: Μέσο CPI του Επεξεργαστή του Μαθήματος

Στα προγράμματα ακεραίων (όχι κινητής υποδιαστολής) μεταξύ των SPEC benchmarks, όταν αυτά εκτελούνται στον MIPS, η συχνότητα εκτέλεσης των διαφόρων εντολών είναι **περίπου** η εξής:

- 26 % load (lw, κλπ),
- 11 % store (sw, κλπ),
- 40 % ALU (add, addi, sub, and, or, slt, slti, κλπ),
- 3 % load upper immediate (lui),
- 16 % conditional branches (beq, bne, κλπ),
- 4 % unconditional jumps (j, jr, jal, κλπ).

(α) Στην υλοποίησή σας της άσκησης 12.1 (δηλαδή χωρίς τις βελτιστοποιήσεις της άσκησης 12.2), πόσους κύκλους ρολογιού παίρνει η εκτέλεση μιάς εντολής του κάθε τύπου; Βασει της διάρκειας αυτής εκτέλεσης του κάθε τύπου εντολής, και βάσει των παραπάνω ποσοστών εκτέλεσης των διαφόρων τύπων εντολών, πόσο θα είναι το μέσο CPI αυτού του επεξεργαστή γιά αυτά τα προγράμματα; (Προφανώς, το

μέσο CPI είναι ο σταθμισμένος μέσος όρος των κύκλων ρολογιού ανά εντολή, όπου οι συντελεστές στάθμισης είναι τα ποσοστά (συχνότητα) εκτέλεσης του κάθε τύπου εντολών).
(β) Έστω τώρα ότι κάνουμε βελτιστοποιήσεις ανάλογες με αυτές της άσκησης 12.2 αλλά περισσότερες: έστω ότι κάνουμε όλες τις εντολές άλματος (j, jr, jal, κλπ) καθώς και την εντολή load upper immediate (lui) να εκτελούνται σε δύο αντί τριών ή περισσοτέρων κύκλων ρολογιού καθεμία. Πόσο θα είναι τότε το νέο μέσο CPI του επεξεργαστή γιά αυτά τα προγράμματα;
(γ) Αν η βελτιστοποίηση (β) έχει σαν αρνητική παρενέργεια να αυξήσει τον κύκλο ρολογιού από 2.4 ns σε 2.6 ns, ποιός από τους δύο επεξεργαστές (α) και (β) θα είναι ταχύτερος, και κατά πόσο; (Υπόδειξη: προφανώς, το πλήθος των εκτελούμενων εντολών $N_{instructions}$ δεν αλλάζει από τον (α) στον (β)).

Άσκηση 14.3: Μέσο CPI Επεξεργαστή με Ομοχειρία (Pipelining)

Βιβλίο (Ελλ. εκδ.): διαβάστε την § 6.1 (σελ. 388-402). Κατά τη σύντομη εισαγωγή μας στην τεχνική της ομοχειρίας (pipelining), είδαμε ότι, όσο ο επεξεργαστής βρίσκει **ανεξάρτητες** μεταξύ τους εντολές, τις εκτελεί με ρυθμό μία εντολή ανά κύκλο ρολογιού. Έτσι, παρά ο γεγονός ότι η εκτέλεση της κάθε εντολής διαρκεί περισσότερους του ενός κύκλους ρολογιού, το συνολικό πλήθος κύκλων ρολογιού γιά την εκτέλεση ενός ολόκληρου προγράμματος --που είναι και αυτό που μας ενδιαφέρει-- είναι περίπου τόσο όσες και οι εκτελούμενες εντολές (υπό τις παραπάνω προϋποθέσεις ανεξαρτησίας). Αρα, το μέσο CPI υπό τις συνθήκες αυτές θα είναι 1, δηλαδή η κάθε εντολή μας "κοστίζει" 1 κύκλο ρολογιού, όσο δηλαδή μας "καθυστερεί" μέχρι να ξεκινήσουμε και την επόμενή της.

Ομως, όπως είπαμε, δυστυχώς, υπάρχουν και αλληλεξαρτήσεις εντολών, οι οποίες προκαλούν απώλεια κύκλου ή κύκλων ρολογιού επιπλέον του παραπάνω ενός "βασικού" κύκλου ανά εντολή. Χωρίς να μπούμε σε πολλές λεπτομέρειες του πώς και γιατί, ας θεωρήσουμε, σε σχέση και με τα ποσοστά εντολών που αναφέρθηκαν στην παραπάνω άσκηση 14.2, ότι:

- Το 33 % των εκτελουμένων load (άρα το 33% του 26% ίσον 8.6% του συνόλου των εκτελουμένων εντολών) ακολουθείται αμέσως από εξαρτημένη εντολή (εντολή που χρειάζεται τα loaded data στην 3η βαθμίδα της pipeline της), και άρα προκαλούν (αυτές οι 33% των load ίσον 8.6% του συνόλου των εντολών) την απώλεια ενός κύκλου ρολογιού επιπλέον του ενός βασικού.
- Το 25 % των εκτελουμένων διακλαδώσεων υπό συνθήκη (branch) (άρα το 25% του 16% ίσον 4% του συνόλου των εκτελουμένων εντολών) δεν καθίσταται εφικτό να περιέχει μία από τις πολλές βελτιστοποιήσεις που υπάρχουν, και κατά συνέπεια προκαλούν (και αυτές οι 4% του συνόλου των εντολών) την απώλεια ενός κύκλου ρολογιού επιπλέον του ενός βασικού.

Με αυτά τα δεδομένα, πόσο θα είναι το μέσο CPI (γιά τα παραπάνω προγράμματα της άσκησης 14.2) ενός επεξεργαστή MIPS που χρησιμοποιεί αυτή τη μορφή ομοχειρίας; Θεωρώντας ότι αυτός ο επεξεργαστής έχει το ίδιο ρολόϊ με εκείνον της άσκησης 14.2(α), και αφού, φυσικά, εκτελεί τις ίδιες εντολές ανά πρόγραμμα, πόσο ταχύτερος θα είναι αυτός ο επεξεργαστής από εκείνον της άσκησης 14.2(α);

Άσκηση 14.4: Μέσο CPI Επεξεργαστή Superscalar με Ομοχειρία

Οι σημερινοί (μικρο-) επεξεργαστές του εμπορίου χρησιμοποιούν τόσο την τεχνική της ομοχειρίας (pipelining) που είδαμε παραπάνω, όσο και τεχνικές εκτέλεσης πολλαπλών εντολών ταυτόχρονα --συνήθως με τη μορφή που αποκαλείται "superscalar". Θεωρήστε ένα τέτοιον επεξεργαστή που σε κάθε κύκλο ρολογιού διαβάζει (fetch) από τη μνήμη τις τέσσερεις (4) επόμενες εντολές, και εκτελεί ταυτόχρονα (εν παραλλήλω) **όσες** από αυτές είναι ανεξάρτητες μεταξύ τους. Το μέσο CPI ενός τέτοιου επεξεργαστή είναι όσο θα ήταν με χρήση ομοχειρίας μόνο (χωρίς superscalarity), διηρημένο διά το μέσο πλήθος ταυτόχρονα εκτελούμενων εντολών, δεδομένου ότι τώρα ο κάθε κύκλος ρολογιού που περνάει "χρεώνεται" σε όλες τις ταυτόχρονα εκτελούμενες εντολές, άρα στην κάθε εντολή χρεώνονται αντίστοιχα λιγότεροι κύκλοι ρολογιού (λιγοστεύουν οι cycles per instruction --CPI).

Θεωρήστε ότι κάνουμε τον επεξεργαστή της άσκησης 14.3 superscalar, και ας πούμε γιά απλότητα ότι αυτό γίνεται χωρίς να αλλάξουμε την δομή της pipeline του και χωρίς να αλλάξει το ρολόϊ (στην πράξη δεν είναι έτσι). Εάν το μέσο πλήθος ταυτόχρονα εκτελούμενων ανεξάρτητων εντολών είναι μιάμιση (1.5) εντολή, τότε ποιό θα είναι το μέσο CPI του νέου επεξεργαστή; Πόσο γρηγορότερος θα είναι αυτός από εκείνον της άσκησης 14.2(α), και πόσο από εκείνον της άσκησης 14.3;

Τρόπος Παράδοσης: Δώστε όλες τις απαντήσεις σας **σε χαρτί**, στο μάθημα, πριν αυτό αρχίσει. (Εάν γράψετε την απάντηση σε υπολογιστή, παρακαλείστε να την τυπώσετε και να παραδώσετε μόνο χαρτί, γιά ομοιομορφία και διευκόλυνση διόρθωσης).