



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 2010-2011, 8ο εξάμηνο, Σχολή ΗΜ&ΜΥ

3η ΕΡΓΑΣΙΑ

Τελική Ημερομηνία Παράδοσης: **29/05/2011** (δεν θα δοθεί παράταση)

Αντικείμενο της εργασίας αυτής είναι η εξοικείωση με τον αλγόριθμο Tomasulo για δυναμική δρομολόγηση εντολών. Σας δίνεται ο ακόλουθος κώδικας MIPS, που υλοποιεί την εκτέλεση ενός απλού βρόχου:

```
foo:      L.D      F0, 0(R1)
          ADD.D   F2, F0, F6
          MUL.D   F4, F0, F2
          DIV.D   F6, F0, F8
          ADD.D   F8, F0, F8
          S.D     F4, 1024(R1)
          DADDI  R1, R1, #-4
          BNEZ   R1, foo
```

Αρχικά ισχύει ότι $R1 \gg 8$. Υποθέτουμε ότι έχουμε τις μονάδες εκτέλεσης με τα χαρακτηριστικά που φαίνονται στον ακόλουθο πίνακα:

Τύπος μονάδας εκτέλεσης	Απαιτούμενοι κύκλοι εκτέλεσης (στάδιο EX)	Αριθμός μονάδων	Αριθμός entries στο αντίστοιχο res. station
Integer	1	1	5
FP add/sub	5	1	4
FP mul/div	15	1	4

Κάνουμε τις εξής παραδοχές:

- Η μονάδα FP mul/div δεν είναι pipelined, ενώ η FP add/subtract είναι πλήρως pipelined με διάρκεια σταδίου ενός κύκλου ρολογιού. Στις non-pipelined μονάδες η εκτέλεση μιας εντολής που απαιτεί περισσότερους από έναν κύκλους δεν μπορεί να επικαλυφθεί από την εκτέλεση μιας δεύτερης εντολής, και έτσι η τελευταία θα πρέπει να περιμένει την ολοκλήρωση της πρώτης. Αντίθετα, στις pipelined μονάδες η εκτέλεση των εντολών επικαλύπτεται. Διάρκεια σταδίου c κύκλων συνεπάγεται ότι οι εντολές μπορούν να εισέρχονται για εκτέλεση στη μονάδα ανά c κύκλους.
- Ανάμεσα σε δύο ή περισσότερες εντολές που περιμένουν να γίνει διαθέσιμη κάποια μονάδα, προτεραιότητα στη χρήση της μόλις αυτή απελευθερωθεί έχει η εντολή που τη ζήτησε πρώτη. Σε ταυτόχρονη ζήτηση, προτεραιότητα έχει η προγενέστερη εντολή.
- Το pipeline συνολικά αποτελείται από τα εξής στάδια: Fetch (IF), Decode (ID), Issue (IS), Execute (EX), Write result (WB).
- Τα στάδια IF, ID, IS, WB διαρκούν 1 κύκλο το καθένα.

- Σε κάθε κύκλο μπορεί να γίνει issue μία εντολή.
- Όταν μία εντολή ολοκληρώνεται και γράφει το αποτέλεσμά της στο CDB στον κύκλο k (κατά το WB), τότε μία εξαρτώμενη εντολή που περιμένει το αποτέλεσμα της θα αρχίσει να εκτελείται από τον κύκλο $k+1$.
- Το CDB είναι κοινό για όλους τους τύπους εντολών (integer και floating point). Τα branches και τα stores δεν το χρησιμοποιούν.
- Στο CDB μπορεί να γράφει το αποτέλεσμά της κάθε φορά μία μόνο εντολή. Αν στον ίδιο κύκλο επιχειρήσουν 2 εντολές να γράψουν, τότε η μεταγενέστερη από τις δύο θα πρέπει να περιμένει. Αυτή που περιμένει εξακολουθεί να καταλαμβάνει την μονάδα εκτέλεσής της καθώς και το αντίστοιχο reservation station.
- Για τα loads και τα stores, θεωρούμε ότι στο στάδιο EX γίνονται και ο υπολογισμός των τελικών διευθύνσεων μνήμης (χρησιμοποιείται διαφορετική μονάδα από την integer) και η πρόσβαση στη μνήμη, τα οποία διαρκούν μαζί 1 κύκλο.
- Οι load και store buffers διαθέτουν από 5 entries ο καθένας
- Η μνήμη διαθέτει 1 port, που σημαίνει ότι μόνο μία εντολή πρόσβασης στη μνήμη (load ή store) μπορεί να εκτελείται κάθε φορά.
- Υποθέτουμε ότι για τα branches δεν υπάρχει πρόβλεψη διακλάδωσης. Αυτό σημαίνει ότι η επόμενη προς εκτέλεση εντολή αρχίζει να φορτώνεται στον επόμενο κύκλο από αυτόν στον οποίο γίνεται γνωστό το αποτέλεσμα της διακλάδωσης (στάδιο EX).
- Οι εντολές DADDI και BNEZ εκτελούνται στην integer μονάδα

a.1) Για τις δύο πρώτες επαναλήψεις του παραπάνω loop εκτελέστε τον αλγόριθμο Tomasulo στην απλή μορφή του (χωρίς Reorder Buffer). Για τις εντολές που εκτελούνται σε αυτό το διάστημα (συμπεριλαμβανομένων των integer), συμπληρώστε έναν πίνακα χρονισμού όπως αυτός που παρουσιάζεται παρακάτω, δείχνοντας τους κύκλους στους οποίους η κάθε εντολή διέρχεται από τα διάφορα στάδια, καθώς και τους λόγους για τους οποίους αυτή stall-άρει για μία ή περισσότερες φορές κατά την εκτέλεσή της (π.χ. stalls στους κύκλους A-B περιμένοντας το περιεχόμενο του καταχωρητή X από την εντολή Y, ή stall στον κύκλο C λόγω του ότι το CDB χρησιμοποιείται από την εντολή Z, ή stall στους κύκλους D-E λόγω μη διαθέσιμου entry στο res. station R, κ.ο.κ.). Για το στάδιο EX (το μόνο το οποίο διαρκεί περισσότερους από 1 κύκλο), υποδείξτε τόσο τον κύκλο στον οποίο αρχίζει να εκτελείται μία εντολή, όσο και τον κύκλο στον οποίο ολοκληρώνεται η εκτέλεση.

Πόσοι κύκλοι απαιτούνται συνολικά για την εκτέλεση των 2 πρώτων επαναλήψεων του loop;

#επανάληψης loop	εντολή	IF	ID	IS	EX	WB	Σχόλια
1	L.D F0,0(R1)	1	2	3	4-4	5	...
...

a.2) Για τον κύκλο 40, παρουσιάστε ένα “στιγμιότυπο” όλων των δομών που χρησιμοποιεί ο αλγόριθμος μαζί με τα περιεχόμενά τους. Σε αυτόν τον κύκλο δηλαδή, δείξτε ποια είναι τα περιεχόμενα των διαφόρων reservation stations και των πεδίων τους, τα περιεχόμενα των load/store buffers και των πεδίων τους, τα περιεχόμενα της δομής register status, κ.λπ., όπως ακριβώς γίνεται στα δύο παραδείγματα χρήσης του αλγορίθμου στις αντίστοιχες διαφάνειες του μαθήματος.

a.3) Ποια είναι τα hazards που εμφανίζονται στη ζητούμενη ακολουθία εντολών και πώς τελικά αυτά αντιμετωπίζονται; Πόσες και ποιες μετονομασίες καταχωρητών γίνονται κατά την εκτέλεση της ακολουθίας;

β.1) Υποθέστε τώρα ότι επεκτείνουμε το προηγούμενο pipeline ώστε να υποστηρίζει *υποθετική εκτέλεση με χρήση ROB* και *2-way υπερβαθμωτή εκτέλεση*. Αυτό συνεπάγεται τις εξής αλλαγές:

- Προσθήκη ROB 12 θέσεων, και 1 επιπλέον στάδιο "Commit" στο pipeline (IF, ID, IS, EX, WB, CO).
- Τα στάδια IF, ID, IS και CO μπορούν πλέον να φορτώνουν, αποκωδικοποιούν, δρομολογούν και ολοκληρώνουν, αντίστοιχα, 2 εντολές σε κάθε κύκλο (εφόσον υπάρχουν διαθέσιμες εντολές για την εκάστοτε λειτουργία).
- Επιπλέον, για τα branches υπάρχει πρόβλεψη διακλάδωσης που είναι TAKEN. Η εντολή στην προβλεφθείσα διεύθυνση-στόχο υποθέτουμε ότι αρχίζει να φορτώνεται στον επόμενο κύκλο από αυτόν που φορτώθηκε η εντολή branch. *Παρατήρηση:* αυτό ισχύει ακόμα και αν τύχει η εντολή που έπεται του branch να πρέπει (λόγω υπερβαθμωτής εκτέλεσης) να φορτωθεί στον ίδιο κύκλο με το branch.

Για τις δύο πρώτες επαναλήψεις του παραπάνω loop, εκτελέστε τον αλγόριθμο Tomasulo και συμπληρώστε έναν πίνακα χρονισμού αντίστοιχο με αυτόν του ερωτήματος α.1, υποδεικνύοντας μεταξύ άλλων τους λόγους για τους οποίους stall-άρει μια εντολή. Πόσοι κύκλοι απαιτούνται συνολικά για την εκτέλεση των 2 πρώτων επαναλήψεων του loop;

β.2) Για τον κύκλο 40, παρουσιάστε ένα “στιγμιότυπο” όλων των δομών που χρησιμοποιεί ο αλγόριθμος μαζί με τα περιεχόμενά τους. Όπως κάνατε και στο ερώτημα α.2, δείξτε ποια είναι τα περιεχόμενα του Reorder Buffer, των διαφόρων reservation stations, των load/store buffers, του register status, κ.λπ.

β.3) Βοήθησε, κατά τη γνώμη σας, η υπερβαθμωτή εκτέλεση την απόδοση του συγκεκριμένου κώδικα; Πού εντοπίζετε τα όποια bottlenecks;

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 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 προσπαθήσετε να την αντιγράψετε απλά από άλλους συμφοιτητές σας.