



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 2007-2008, 8ο εξάμηνο, Σχολή ΗΜ&ΜΥ

3η ΕΡΓΑΣΙΑ

Τελική Ημερομηνία Παράδοσης: **31/5/2008** (δεν θα δοθεί παράταση)

Η υπορουτίνα DAXPY (**D**ouble precision **AX** Plus **Y**), που προδιαγράφεται στο πρότυπο βασικών υπορουτινών γραμμικής άλγεβρας BLAS-1 (Level-1 Basic Linear Algebra Subprogramms), υλοποιεί τη πράξη $Y = a * X + Y$, όπου X, Y διανύσματα με στοιχεία κινητής υποδιαστολής διπλής ακρίβειας, και a κάποια scalar σταθερά. Ο αντίστοιχος MIPS κώδικας που υλοποιεί αυτή την πράξη για διανύσματα 100 στοιχείων φαίνεται στη συνέχεια. Αρχικά, ο R1 είναι 0, ο R3 είναι 0, ενώ η σταθερά a βρίσκεται στον F0.

```
f00:      L.D      F2,0(R1)          ;φόρτωσε το X(i)
          MUL.D   F4,F2,F0        ;πολλαπλασίασε a*X(i)
          L.D      F6,0(R2)        ;φόρτωσε το Y(i)
          ADD.D   F6,F4,F6        ;πρόσθεσε a*X(i)+Y(i)
          S.D      F6,0(R2)        ;αποθήκευσε το Y(i)
          DADDUI  R1,R1,#8         ;επόμενο στοιχείο στο X
          DADDUI  R2,R2,#8         ;επόμενο στοιχείο στο Y
          DSGTUI  R3,R1,#800      ;θέσε R3=1 αν R1>800
          BEQZ    R3,f00
```

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 adder	4	1	3
FP multiplier	15	1	2

Επίσης, κάνουμε τις εξής παραδοχές:

- οι μονάδες εκτέλεσης δεν είναι pipelined. Αυτό σημαίνει ότι η εκτέλεση μιας εντολής που απαιτεί περισσότερους από έναν κύκλους, δεν μπορεί να επικαλυφθεί από την εκτέλεση μιας δεύτερης εντολής στην ίδια μονάδα. Η δεύτερη εντολή μπορεί να αρχίσει να εκτελείται στην μονάδα αυτή μόνο όταν ολοκληρώσει την εκτέλεσή της η πρώτη 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 αποτέλεσμα της στο CDB στον κύκλο k, τότε μία εξαρτώμενη εντολή που περιμένει το αποτέλεσμα αυτό θα αρχίσει να εκτελείται από τον κύκλο k+1
- στο CDB μπορεί να γράφει το αποτέλεσμα της κάθε φορά μία μόνο εντολή. Αν στον ίδιο κύκλο επιχειρήσουν 2 εντολές να γράψουν στο CDB, τότε η μία από τις δύο (π.χ. η μεταγενέστερη) πρέπει να stall-άρει
- για τα loads και τα stores, θεωρούμε ότι στο στάδιο EX γίνεται και ο υπολογισμός των τελικών διευθύνσεων μνήμης (effective address calculation) και η πρόσβαση στη μνήμη, τα οποία διαρκούν και τα δύο 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 υπάρχει πρόβλεψη διακλάδωσης που είναι TAKEN. Η εντολή στη διεύθυνση στόχο επομένως αρχίζει να φορτώνεται στον επόμενο κύκλο από αυτόν στον οποίο φορτώνεται η εντολή branch.
- οι εντολές DADDUI, DSGTUI και BEQZ εκτελούνται στην integer μονάδα

α.1) Για τις τρεις 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, 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 3 πρώτων επαναλήψεων του loop;

#επανάληψης loop	εντολή	IF	ID	IS	EX	WB	Σχόλια
1	L.D F2,0(R1)	1	2	3	4	5	
1	MUL.D F4,F2,F0	2	3	4	6-20	21	stall στον κύκλο 5: αναμονή για τον F2 από την L.D
...							
...							

α.2) Για τον κύκλο 30, 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.

α.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 προσαυξάνουμε το προηγούμενο pipeline ώστε να υποστηρίζεται η ανανέωση της αρχιτεκτονικής κατάστασης του επεξεργαστή με τη σωστή σειρά προγράμματος, με τη σειρά δηλαδή με την οποία γίνονται issue οι εντολές. Για το σκοπό αυτό, θεωρούμε τώρα ότι εφαρμόζεται ο αλγόριθμος Tomasulo χρησιμοποιώντας Reorder Buffer 12 θέσεων. Έτσι, η εκτέλεση

των εντολών προσαυξάνεται στην ουσία κατά 1 επιπλέον στάδιο στο pipeline, το “Commit” (IF, ID, IS, EX, WB, CO).

Για τις τρεις 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 3 πρώτων επαναλήψεων του loop;

β.2) Για τον κύκλο 30, 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.

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.