



# 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 2011-2012, 8ο εξάμηνο, Σχολή ΗΜ&ΜΥ

### 3η ΕΡΓΑΣΙΑ

Τελική Ημερομηνία Παράδοσης: **10/06/2012** (δεν θα δοθεί παράταση)

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

0x00880000	LOOP:	LD	F0, 8(R1)
0x00880004		LD	F1, 8(R2)
0x00880008		ADDD	F4, F4, F0
0x0088000C		MULD	F4, F4, F1
0x00880010		ADDI	R4, R4, #1
0x00880014		ANDI	R5, R4, #1
0x00880018		SUBI	R5, R5, #1
0x0088001C		BEZ	R5, L1
0x00880020		ADDI	R1, R1, #8
0x00880024		LD	F0, 8(R1)
0x00880028		ADDD	F4, F4, F0
0x0088002C	L1:	ADDI	R1, R1, #8
0x00880030		ADDI	R2, R2, #8
0x00880034		SUBI	R6, R6, #1
0x00880038		BNEZ	R6, LOOP
0x0088003C		ADDD	F4, F4, F0
0x00880040		ADDD	F4, F4, F5
0x00880044		ADDI	R8, R8, #1

και οι αρχικές τιμές  $R4=0$  και  $R6=2$ . Όσον αφορά την αρχιτεκτονική του συστήματος κάνουμε τις εξής παραδοχές:

- Το pipeline περιέχει τα στάδια IS, EX, WR αγνοούμε δηλαδή τα IF και ID. Όλα τα στάδια (εκτός του EX προφανώς) διαρκούν 1 κύκλο.
- Το σύστημα διαθέτει *περιορισμένο* αριθμό reservation stations (RS). Διαθέτει 1 RS για προσθέσεις/αφαιρέσεις και 1 RS για πολλαπλασιασμούς/διαιρέσεις Floating Point αριθμών. Αντίστοιχα, για εντολές Integer αριθμών διαθέτει 4 RS.
- Το σύστημα περιλαμβάνει 1 non-pipelined FU για πράξεις Integer αριθμών. Όλες οι εντολές μεταξύ Integer αριθμών διαρκούν 1 κύκλο.
- Το σύστημα περιλαμβάνει 2 non-pipelined Floating Point FUs, ένα για ADDD/SUBD και ένα για MULD/DIVD. Οι εντολές πρόσθεσης/αφαίρεσης διαρκούν 3 κύκλους, ενώ οι εντολές πολλαπλασιασμού/διαίρεσης 6 κύκλους.
- Οι εντολές αναφοράς στη μνήμη πραγματοποιούν στο στάδιο EX τόσο τον υπολογισμό της διεύθυνσης αναφοράς όσο και την προσπέλαση στη μνήμη. Το σύστημα διαθέτει ένα Load και ένα Store Queue, 2 θέσεων το καθένα. Οι εντολές χρησιμοποιούν ένα ξεχωριστό pipelined FU για τον

υπολογισμό της διεύθυνσης και διαρκούν συνολικά 2 ή 5 κύκλους σε περίπτωση Hit ή Miss στην cache αντίστοιχα.

- 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 WR).
- Το σύστημα διαθέτει μια άδεια, fully associative cache με 32 cache lines και μέγεθος block 16 bytes.
- Οι καταχωρητές R1, R2 περιέχουν την διεύθυνση του πρώτου στοιχείου των πινάκων A και B αντίστοιχα, οι οποίοι έχουν αποθηκευμένους αριθμούς διπλής ακρίβειας (μήκους 8 bytes ο καθένας).
- Όταν μία εντολή ολοκληρώνεται και γράφει το αποτέλεσμά της στο CDB στον κύκλο k (κατά το WR), 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 το reservation station αλλά όχι και την μονάδα εκτέλεσής της.

**A.1)** Εκτελέστε τον παραπάνω κώδικα χρησιμοποιώντας 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 του κώδικα;

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

**A.2)** Για τον κύκλο 22, 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;

**B.1)** Υποθέστε τώρα ότι επεκτείνουμε το προηγούμενο pipeline ώστε να υποστηρίζει **υποθετική εκτέλεση με χρήση ROB**. Αυτό συνεπάγεται τις εξής αλλαγές:

- Προσθήκη ROB 8 θέσεων, και 1 επιπλέον στάδιο "Commit" στο pipeline (IS, EX, WR, CMT).
- Επιπλέον, για τα branches χρησιμοποιείται πλέον πρόβλεψη διακλάδωσης. Συγκεκριμένα, το σύστημα χρησιμοποιεί έναν (1,1) global history predictor. Ο προβλέπτης αυτός διαθέτει ένα Pattern History Table (PHT) με συνολικά 2 εγγραφές. Οι 1-bit predictors είναι αρχικοποιημένοι στο NT, που είναι TAKEN.

- Οι εντολές διακλάδωσης υπό συνθήκη χρησιμοποιούν τα Integer RS και FU προκειμένου να υπολογίσουν αν ισχύει η συνθήκη. Η πρόβλεψη για μια εντολή διακλάδωσης υπό συνθήκη γίνεται *ταυτόχρονα* με τη δρομολόγηση της εντολής. Ο έλεγχος της πρόβλεψης γίνεται μόλις γίνει γνωστό το αποτέλεσμα της εντολής (στο στάδιο WR). Σε περίπτωση σφάλματος, σταματά η εκτέλεση των εντολών του miss-predicted execution path και στον επόμενο κύκλο δρομολογείται η σωστή εντολή.

Εκτελέστε ολόκληρο τον παραπάνω κώδικα χρησιμοποιώντας 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 ολόκληρου του κώδικα;

**B.2)** Για τον κύκλο 22, 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 από άλλους συμφοιτητές σας.*