



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

ΕΡΓΑΣΤΗΡΙΟ ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

www.cslab.ece.ntua.gr

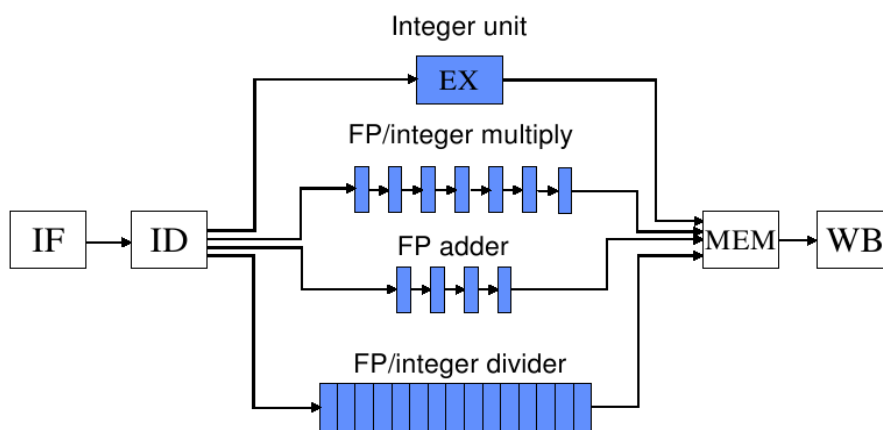
3η ΑΣΚΗΣΗ ΣΤΗΝ ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΥΠΟΛΟΓΙΣΤΩΝ

Ακ. έτος 2008-2009, 5ο Εξάμηνο Σχολή ΗΜ&ΜΥ

Τελική Ημερομηνία Παράδοσης: 1/2/2009

Στο πλαίσιο αυτής της άσκησης θα χρησιμοποιηθεί ο DLX, ένα πρότυπο επεξεργαστή που αναπτύχθηκε για εκπαιδευτικούς σκοπούς από τους J. Hennessy και D. Patterson και διαθέτει σωλήνωση παρόμοια με αυτή του MIPS. Συγκεκριμένα, θα χρησιμοποιηθεί ο προσομοιωτής WinDLX, ο οποίος τρέχει σε περιβάλλον Windows. Ο WinDLX προσομοιώνει με ακρίβεια κύκλο την εκτέλεση ενός προγράμματος στον DLX, και μεταξύ άλλων οπτικοποιεί την ροή των εντολών από τα στάδια της σωλήνωσης, μαζί με τα όποια stalls, προωθήσεις, κ.λ.π.

Η σωλήνωση του DLX παρουσιάζεται στο ακόλουθο διάγραμμα. Για τους αριθμούς κινητής υποδιαστολής, οι πράξεις της πρόσθεσης, αφαίρεσης και πολλαπλασιασμού διαρκούν περισσότερους του ενός κύκλους και είναι πλήρως επικαλυπτόμενες (fully pipelined), ενώ η πράξη της διαίρεσης δεν είναι επικαλυπτόμενη. Αυτό σημαίνει ότι η εντολή που έπεται μιας εντολής πρόσθεσης, αφαίρεσης ή πολλαπλασιασμού η οποία έχει εισέλθει στο στάδιο EX, μπορεί να εισέλθει στο στάδιο EX από τον επόμενο κύκλο. Αν έπεται μιας εντολής διαίρεσης, αντίθετα, μπορεί να εισέλθει στο στάδιο EX μόνο όταν η εντολή διαίρεσης εξέλθει από αυτό.



Σκοπός αυτής της εργασίας είναι η καλύτερη κατανόηση της λειτουργίας της σωλήνωσης και των εννοιών που σχετίζονται με αυτήν.

Θεωρείστε το ακόλουθο πρόγραμμα σε C το οποίο ανανεώνει τα στοιχεία ενός διανύσματος.

```
double x[100];

double pi = 3.14159;

for (i=0; i<100; i++) {

    x[i] = x[i] + pi;

}
```

Ο αντίστοιχος κώδικας DLX είναι ο εξής:

	.data	;data segment
x:	.space 800	;δέσμευσε χώρο για 100 doubles (800 bytes). ;ο 'x' συμβολίζει τη διεύθυνση από όπου ;αρχίζει ο χώρος αυτός
pi:	.double 3.14159	;αποθήκευσε τον αριθμό 3.14159 στη μνήμη. ;το 'pi' συμβολίζει την αρχική διεύθυνση του ;αριθμού αυτού στη μνήμη
	.text	;code segment
	.global main	
main:		
	addi r1,r0,x	;φόρτωσε στον r1 την αρχική διεύθυνση του ;διανύσματος 'x'
	ld f2,pi(r0)	;φόρτωσε στον f2 τον αριθμό που βρίσκεται ;στη διεύθυνση 'pi' (σημ: επειδή είναι ;double, ο αριθμός θα φορτωθεί στον f2 αλλά ;και στον f3)
	addi r4,r1,#800	;αποθήκευσε στον r4 την τελική ;διεύθυνση του διανύσματος 'x'
loop:		
	ld f0,0(r1)	;φόρτωσε στον f0 (και στον f1) ένα στοιχείο ;από το διάνυσμα
	addd f4,f0,f2	;πρόσθεσέ του τον 3.14159
	sd 0(r1),f4	;αποθήκευσε το αποτέλεσμα
	addi r1,r1,#8	;αύξησε τον r1 κατά 8 για να δείχνει στο ;επόμενο στοιχείο του 'x'
	slt r2,r1,r4	;είναι ο r1 μικρότερος από την τελική ;διεύθυνση του διανύσματος?
	bnez r2,loop	;αν ναι, συνέχισε την εκτέλεση του loop
	trap 0	;τερματισμός προγράμματος

A) Εκτελέστε το παραπάνω πρόγραμμα στον προσομοιωτή, απενεργοποιώντας αρχικά την προώθηση. Αξιολογείστε την απόδοσή του:

- δείξτε τις εξαρτήσεις που προκύπτουν ανάμεσα στις εντολές (για μία επανάληψη του βρόχου)
- αναφέρατε τα διάφορα είδη stalls που εμφανίζονται στο pipeline, καθώς και το ποσοστό επί του συνολικού αριθμού των κύκλων εκτέλεσης που αντιστοιχούν σε stalls.

B) Ενεργοποιείτε την προώθηση. Αξιολογείστε την απόδοση του κώδικα (απαντήστε στα ίδια υποερωτήματα όπως και πριν). Ποια είναι η χρονική βελτίωση (επιτάχυνση-speedup) που δίνει η προώθηση;

Γ) Αναδιατάξτε τον αρχικό κώδικα (κάνοντας όποιες τροποποιήσεις θεωρείτε αναγκαίες για τη διατήρηση της ορθής λειτουργίας του) ώστε να μειωθούν όπου είναι δυνατόν τα stalls. Αξιολογείστε την απόδοσή του. Ποια είναι η χρονική βελτίωση σε σχέση με το (β);

Παρατηρήσεις:

1. Για την εκτέλεση του κώδικα, αρκεί να δημιουργήσετε ένα .s αρχείο (χρησιμοποιώντας οποιοδήποτε editor) με τις εντολές assembly και τα διάφορα assembler directives (ό,τι αναγράφεται δηλαδή παραπάνω με έντονα γράμματα), το οποίο φορτώνετε στην προσομοιούμενη κύρια μνήμη (File|Load Code or Data) και εκτελείτε (Execution|Run, για εκτέλεση μέχρι τέλους, ή Execution|Single Cycle, για βηματική εκτέλεση).
2. Σε κάθε διαφορετική έκδοση του κώδικα που φτιάχνετε για τα διάφορα υποερωτήματα, να ελέγχετε πάντα την ορθότητα του προγράμματος, εξετάζοντας τα τελικά περιεχόμενα του διανύσματος x.
3. Μερικά χρήσιμα links για τον DLX και τον WinDLX:
 - <http://cs.uns.edu.ar/~jechaiz/arquitectura/windlx/windlx.html>
 - <http://cs.uns.edu.ar/~jechaiz/arquitectura/windlx/DLXinst.html>
 - <http://www.ndsu.nodak.edu/instruct/tareski/ee774f96/notes/windlx/wdlxtut.htm>
 - <http://atc2.aut.uah.es/~acebron/tutorial2.pdf>

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 ηλεκτρονικά στην ιστοσελίδα:

<http://www.cslab.ece.ntua.gr/courses/comparch/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.