



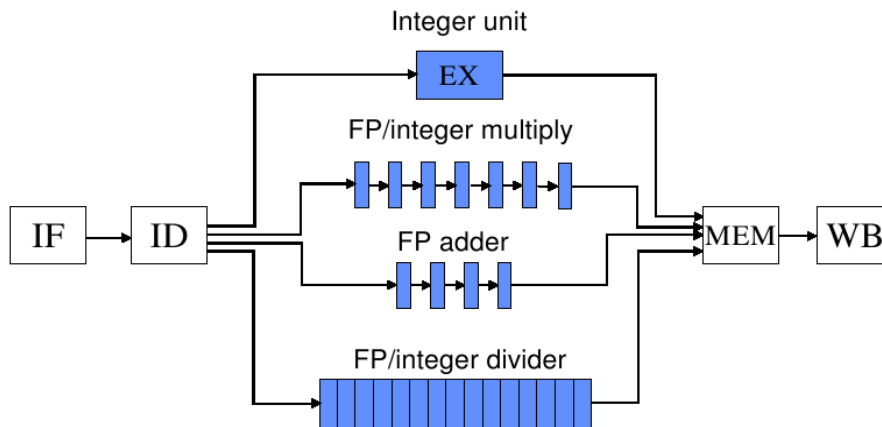
ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ
ΕΡΓΑΣΤΗΡΙΟ ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ
www.cslab.ece.ntua.gr

3η ΑΣΚΗΣΗ ΣΤΗΝ ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΥΠΟΛΟΓΙΣΤΩΝ
Ακ. έτος 2011-2012, 5ο Εξάμηνο Σχολή ΗΜ&ΜΥ

Τελική Ημερομηνία Παράδοσης: 12/02/2012

Στο πλαίσιο αυτής της άσκησης θα χρησιμοποιηθεί ο DLX, ένας πρότυπος επεξεργαστής 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 μπλοκ διάγραμμα. Οι πράξεις για τους αριθμούς κινητής υποδιαστολής διαρκούν περισσότερους του ενός κύκλους. Επιπλέον, οι πράξεις της πρόσθεσης, αφαίρεσης και πολλαπλασιασμού είναι *πλήρως επικαλυπτόμενες* (fully pipelined), ενώ η πράξη της διαίρεσης είναι *μη επικαλυπτόμενη* (non pipelined). Αυτό σημαίνει ότι αν έχουμε π.χ. δύο διαδοχικές εντολές πρόσθεσης, τότε μπορούν να αρχίσουν να εκτελούνται στη μονάδα πρόσθεσης σε διαδοχικούς κύκλους. Αντίθετα, αν έχουμε δύο διαδοχικές εντολές διαίρεσης, τότε η δεύτερη εντολή μπορεί να εισέλθει στο στάδιο EX μόνο όταν η πρώτη εξέλθει από αυτό. Η σωλήνωση του DLX παρουσιάζεται στο ακόλουθο διάγραμμα.



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], y[100];

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

    x[i] = x[i] + y[i];

}
```

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
y:              .space 800           ;δέσμευσε χώρο για 100 doubles (800 bytes).
                ;ο 'y' συμβολίζει τη διεύθυνση από όπου
                ;αρχίζει ο χώρος αυτός

                .text                ;code segment
                .global main

main:

    addi r1,r0,x    ;φόρτωσε στον r1 την αρχική διεύθυνση του
                   ;διανύσματος 'x'
    addi r2,r0,y    ;φόρτωσε στον r2 την αρχική διεύθυνση του
                   ;διανύσματος 'y'
    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 'x'
    ld f2,0(r2)    ;φόρτωσε στον f2 (και στον f3) ένα στοιχείο
                   ;από το διάνυσμα 'y'

    addd f4,f0,f2   ;αποθήκευσε το αποτέλεσμα
    sd 0(r1),f4     ;αύξησε τον r1 κατά 8 για να δείχνει στο
                   ;επόμενο στοιχείο του 'x'
    addi r1,r1,#8   ;αύξησε τον r2 κατά 8 για να δείχνει στο
                   ;επόμενο στοιχείο του 'y'
    addi r2,r2,#8

    slt r5,r1,r4    ;είναι ο r1 μικρότερος από την τελική
                   ;διεύθυνση του διανύσματος?
    bnez r5,loop    ;αν ναι, συνέχισε την εκτέλεση του loop
    trap 0          ;τερματισμός προγράμματος
```

A) Εκτελέστε το παραπάνω πρόγραμμα στον προσομοιωτή απενεργοποιώντας αρχικά την προώθηση. Χρησιμοποιήστε τις προκαθορισμένες (default) παραμέτρους του προσομοιωτή για τη μονάδα πρόσθεσης ($Count=1$, $Delay=2$). Παρακολουθήστε την εξέλιξη της εκτέλεσης και αξιολογήστε την απόδοσή της:

- i.** Δείξτε τις εξαρτήσεις που προκύπτουν ανάμεσα στις εντολές (για μία επανάληψη του βρόχου).
- ii.** Πόσοι κύκλοι απαιτούνται για την εκτέλεση του προγράμματος;
- iii.** Αναφέρατε τα διάφορα είδη stalls που εμφανίζονται, καθώς και το ποσοστό τους επί του συνολικού αριθμού των κύκλων.

B) Είναι δυνατή η εμφάνιση κινδύνων WAR στο pipeline του ερωτήματος A, και αν ναι, πώς επιλύονται; Δικαιολογήστε την απάντησή σας και επαληθεύστε την με έναν παράδειγμα κώδικα τον οποίο εκτελέστε στον προσομοιωτή (μπορείτε να χρησιμοποιήσετε τον αρχικό κώδικα σαν βάση).

Γ) Μειώστε την καθυστέρηση της μονάδας πρόσθεσης στον 1 κύκλο. Τί βελτίωση πετυχαίνετε σε σχέση με την περίπτωση A;

Δ) Ενεργοποιήστε την προώθηση, θεωρώντας το pipeline του ερωτήματος Γ. Όπως και στο ερώτημα A:

- i.** Δείξτε τις εξαρτήσεις που προκύπτουν για μία επανάληψη του βρόχου.
- ii.** Πόσοι κύκλοι απαιτούνται για την εκτέλεση του προγράμματος;
- iii.** Αναφέρατε τα διάφορα είδη stalls που εμφανίζονται, καθώς και το ποσοστό τους επί του συνολικού αριθμού των κύκλων.
- iv.** Τι βελτίωση πετυχαίνετε σε σχέση με την περίπτωση Γ;

Ε) Στον αρχικό κώδικα, εφαρμόστε την τεχνική του ξεδιπλώματος βρόχων (loop unrolling) 2 φορές. Αξιολογήστε την απόδοσή του χρησιμοποιώντας το pipeline του ερωτήματος Δ.

- i.** Πόσοι κύκλοι απαιτούνται για την εκτέλεση του προγράμματος;
- ii.** Τι βελτίωση πετυχαίνετε σε σχέση με την περίπτωση Δ;
- iii.** Μπορείτε να πετύχετε καλύτερη απόδοση αυξάνοντας τον αριθμό των μονάδων πρόσθεσης σε 2; Δικαιολογήστε την απάντησή σας.

Ζ) Χωρίς να εφαρμόσετε περαιτέρω unrolling, τροποποιήστε καταλλήλως τον κώδικα του ερωτήματος Ε ώστε να πετύχετε ακόμη καλύτερη απόδοση. Υποθέστε το pipeline του ερωτήματος Ε.

- i.** Πόσοι κύκλοι απαιτούνται για την εκτέλεση του προγράμματος;
- ii.** Τι βελτίωση πετυχαίνετε σε σχέση με την περίπτωση Ε;
- iii.** Μπορείτε να πετύχετε καλύτερη απόδοση αυξάνοντας τον αριθμό των μονάδων πρόσθεσης σε 2; Δικαιολογήστε την απάντησή σας.

Παρατηρήσεις:

1. Τον προσομοιωτή μπορείτε να τον βρείτε στην ακόλουθη διεύθυνση:

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

2. Για την προσομοίωση ενός προγράμματος αρκεί να δημιουργήσετε ένα .s αρχείο (χρησιμοποιώντας οποιοδήποτε editor) με τον πηγαίο κώδικα του προγράμματος σε assembly του DLX, το οποίο φορτώνετε στην προσομοιώμενη κύρια μνήμη (File → Load Code or Data) και εκτελείτε (Execution → Run για εκτέλεση μέχρι τέλους, ή Execution → Single Cycle για βηματική εκτέλεση).
3. 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.
4. 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://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 από άλλους συμφοιτητές σας.