



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

14 Μαρτίου 2004

ΠΡΟΗΓΜΕΝΑ ΘΕΜΑΤΑ ΟΡΓΑΝΩΣΗΣ ΥΠΟΛΟΓΙΣΤΩΝ
Εξετάσεις Φεβρουαρίου 2004

Επώνυμο:	
Όνομα:	
Εξάμηνο:	

Θέμα	Βαθμός
1	
2	
3	
4	

Μην ξεχάσετε να γράψετε το ονοματεπώνυμο σας σε όλες τις κόλλες. Η εξέταση γίνεται με ανοικτά βιβλία και σημειώσεις. Διάρκεια εξέτασης 2 ½ ώρες. Ο βαθμός του διαγωνίσματος είναι το 70% του τελικού βαθμού προαγωγής στο μάθημα.

Θέμα 1^ο (30%):

Έστω το τμήμα του C κώδικα:

```
for (i = 0; i < 12; i++)
    if (array[i] == k)
        array[i] = j;
```

Θεωρούμε ότι οι τιμές των μεταβλητών $4 \cdot i$, j , k βρίσκονται στους καταχωρητές $\$s1$, $\$s2$, $\$s3$ και η διεύθυνση του πίνακα `array` στον $\$s4$. Το περιεχόμενο των καταχωρητών $\$s1$ και $\$s5$ είναι αρχικά 0 και 48, αντίστοιχα. Ο αντίστοιχος κώδικας σε assembly είναι ο εξής:

```
Loop:    add    $t1, $s1, $s4
         lw    $t0, 0($t1)
         bne   $t0, $s3, Next
         sw    $s2, 0($t1)
Next:    addi   $s1, $s1, 4
         bne   $s1, $s5, Loop
Exit:
```

α) Δείξτε το διάγραμμα εκτέλεσης των επιμέρους φάσεων για κάθε εντολή, μέσα στην αρχιτεκτονική αγωγού, χωρίς την ύπαρξη κανενός σχήματος προώθησης (διάγραμμα εκτέλεσης όπως στις διαφάνειες του μαθήματος: IF - ID -...κλπ). Υποθέστε ότι κάθε φορά θεωρούμε ότι δεν πρόκειται να γίνει διακλάδωση (branch not taken) και ότι στο 75% των περιπτώσεων ισχύει η ισότητα `array[i]==k`. Πόσους κύκλους χρειάζεται για να εκτελεστεί ο παραπάνω βρόχος;

β) Δείξτε το διάγραμμα εκτέλεσης των επιμέρους φάσεων για κάθε εντολή, μέσα στην αρχιτεκτονική αγωγού, υποθέτοντας την ύπαρξη σχήματος προώθησης. Ισχύουν οι υποθέσεις του ερωτήματος (α) σε ό,τι αφορά τις εντολές διακλάδωσης και τον έλεγχο της ισότητας `array[i]==k`. Πόσους κύκλους

χρειάζεται για να εκτελεστεί ο παραπάνω βρόχος; Μπορείτε να βελτιστοποιήσετε τον κώδικα (αποφυγή τυχόν pipeline stalls);

(γ) Η τεχνική loop unrolling (ξεδίπλωμα βρόχου) είναι μια από τις πιο διαδεδομένες τεχνικές βελτιστοποίησης κώδικα. Στην τεχνική αυτή, διαδοχικές επαναλήψεις του αρχικού βρόχου γράφονται σαν ξεχωριστές εντολές του ίδιου βρόχου (π.χ. για 2 διαδοχικές επαναλήψεις, ο unrolled βρόχος έχει πλέον βήμα 2). Στο παράδειγμά μας, ο αρχικός βρόχος ξεδιπλώνεται ως εξής:

```
for (i = 0; i < 12; i+=2) {
    if (array[i] == k)
        array[i] = j;
    if (array[i+1] == k)
        array[i+1] = j;
}
```

Εδώ, θέλουμε να αξιολογήσουμε τα πλεονεκτήματα υλοποίησης της τεχνικής ξεδιπλώματος βρόχων σε κώδικα που εκτελείται σε αρχιτεκτονική MIPS με χρήση αγωγού (pipeline datapath) και ύπαρξη σχήματος προώθησης (forwarding). Παρακάτω είναι το τμήμα του αρχικού κώδικα, όπου ο βρόχος είναι ξεδιπλωμένος μία φορά:

```
Loop:      add  $t1, $s1, $s4
           lw   $t0, 0($t1)
           bne  $t0, $s3, Next1
           sw   $s2, $0($t1)
Next1:     lw   $t2, 4($t1)
           bne  $t2, $s3, Next2
           sw   $s2, 4($t1)
Next2:     addi $s1, $s1, 8
           bne  $s1, $s5, Loop
Exit:
```

Σχεδιάστε ξανά τον κώδικα ώστε να αποφύγετε τις υποχρεωτικές καθυστερήσεις (pipeline stalls). Λαμβάνοντας υπόψη τα αναγκαία stalls συγκρίνετε τη διαφορά επίδοσης μεταξύ του αρχικού μη-ξεδιπλωμένου κώδικα (A) και του δικού σας (μετά το ξεδίπλωμα και τη βελτιστοποίηση του κώδικα B).

Θέμα 2^ο (15%):

Έστω ότι επιθυμούμε να έχουμε τη δυνατότητα απευθείας προσπέλασης στη μνήμη από τις αριθμητικές εντολές, μέσω της εντολής:

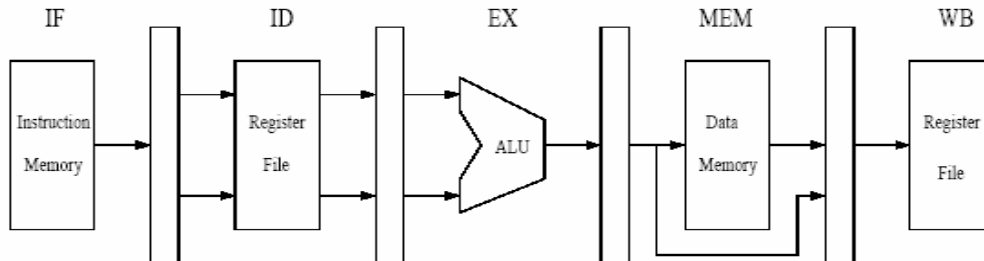
```
addm $t1, 100($t3)
```

δηλαδή: $\$t1 = \$t1 + \text{Mem}[\$t3+100]$

Γιατί θα ήταν δύσκολο να προσθέσουμε την εντολή αυτή σε αρχιτεκτονική MIPS με χρήση αγωγού (pipelined datapath), όπως αυτή περιγράφεται στο κεφ. 6 των σημειώσεων; Τι συνέπειες θα είχαν οι αλλαγές που απαιτούνται για την υλοποίηση της νέας εντολής (σε throughput, latency και hazards);

Θέμα 3^ο (25%):

Έστω η παρακάτω αρχιτεκτονική αγωγού (pipelined datapath) 5 σταδίων (IF-ID-EX-MEM-WB), όπως του επεξεργαστή MIPS του μαθήματος, όπου όμως το Register File **γράφεται στο δεύτερο μισό** του κύκλου ρολογιού και **διαβάζεται στο πρώτο μισό**. Οι εντολές μπορούν να καθυστερήσουν (stall) μέσα στην αρχιτεκτονική αγωγού (pipeline stalls) μόνο στα στάδια IF και ID.



α) για την παρακάτω ακολουθία εντολών, δείξτε την εκτέλεσή τους μέσα στο pipelined datapath, γεμίζοντας τον παρακάτω πίνακα. Όπου συμβαίνει καθυστέρηση (stall), βάλτε στο αντίστοιχο στάδιο τη λέξη bubble. (10%)

```
lw   $s2, 0($s1)    ; lw1
add  $s2, $s1, $s2  ; add1
add  $s3, $s1, $s3  ; add2
sw   $s2, 0($s4)    ; sw1
```

cycles	IF	ID	EX	MEM	WB
1	lw1				
2	add1	lw1			
3	add2	add1	lw1		
..					

β) προσπαθήστε να εξαφανίσετε καθυστερήσεις (stalls) που εμφανίζονται στην ακολουθία εντολών, προσθέτοντας σχήμα προώθησης (forwarding-bypass path), ώστε η παραπάνω αρχιτεκτονική αγωγού να συμπεριφέρεται όπως μια αρχιτεκτονική αγωγού όπου το register file γράφεται στο πρώτο μισό του κύκλου και διαβάζεται στο δεύτερο μισό. Δείξτε σε αναλυτικό διάγραμμα τα εναλλακτικά μονοπάτια δεδομένων που δημιουργεί το σχήμα προώθησης που επιλέξατε. Στον πίνακα του α) ερωτήματος, βάλτε το γράμμα B δίπλα σε κάθε bubble που εξαφανίζεται με το σχήμα προώθησης που επιλέξατε (Δηλ. bubble, B). Τέλος, γράψτε τις συνθήκες που πρέπει να ισχύουν για να γίνει το προτεινόμενο σχήμα προώθησης. (15%)

Θέμα 4^ο (30%):

Δίνεται μια σειρά αναφορών σε διευθύνσεις λέξεων στη μνήμη ενός υπολογιστή: 5, 9, 12, 28, 32, 25, 6, 8, 11, 28, 31, 24, 5, 25, 4, 12. Υποθέτουμε ότι έχουμε κρυφή μνήμη με οργάνωση:

- απευθείας απεικόνισης (direct mapped) με 16 blocks, όπου κάθε block έχει μέγεθος μια λέξη (word),
- απευθείας απεικόνισης (direct mapped), με 16 λέξεις (words) συνολικό μέγεθος cache, όπου κάθε block έχει μέγεθος τέσσερις (4) λέξεις.
- συνόλου συσχέτισης 2-δρόμων (2-way set associative) με συνολικό μέγεθος 16 λέξεις (words) όπου κάθε block έχει μέγεθος δύο (2) λέξεις (Υποθέστε LRU αλγόριθμο αντικατάστασης).

Δείξτε για τις παραπάνω περιπτώσεις οργάνωσης της κρυφής μνήμης, για κάθε αναφορά, αν είναι επιτυχής (hit) ή όχι (miss) καθώς και την τελικά περιεχόμενα της κρυφής μνήμης.