



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
Εξετάσεις Ιουλίου 2013
Διάρκεια 2:45' ώρες

Οι εξετάσεις θα πραγματοποιηθούν ΧΩΡΙΣ την παρουσία βιβλίων, βοηθημάτων ή άλλου είδους σημειώσεων. Το μόνο που επιτρέπεται να έχετε μαζί σας είναι ένα φύλλο A4 στο οποίο μπορείτε να έχετε γράψει ό,τι έχετε κρίνει πιο σημαντικό για το μάθημα και θέλετε να το έχετε ως βοήθημά σας. Απαγορεύεται η ανταλλαγή οποιουδήποτε αντικειμένου κατά την ώρα της εξέτασης, ούτε και των φύλλων A4 που είναι ατομικά.

Θέμα 1ο (15%)

A. Έστω επεξεργαστής που υλοποιεί Register Renaming στο ID στάδιο χρησιμοποιώντας ένα physical register file με περισσότερους φυσικούς καταχωρητές από ότι ορίζει το ISA του. Πότε είναι ασφαλές να θεωρήσουμε ότι ένας φυσικός καταχωρητής είναι πλέον ελεύθερος και μπορούμε να τον επαναχρησιμοποιήσουμε;

B. Δίνεται ο παρακάτω κώδικας σε C, ο οποίος διατρέχει έναν πίνακα, του οποίου οι τιμές είναι πραγματικά τυχαίοι θετικοί ακέραιοι. Ο κώδικας περιέχει 4 branches (B1, B2, B3, B4).

```
for(int i = 0; i < N; i++) { /* B1 */
    val = array[i];          /* Taken path for B1 */
    if (val % 2 == 0) {      /* B2 */
        sum += val;         /* Taken path for B2 */
    }
    if (val % 3 == 0) {      /* B3 */
        sum += val;         /* Taken path for B3 */
    }
    if (val % 6 == 0) {      /* B4 */
        sum += val;         /* Taken path for B4 */
    }
}
```

- (i) Υπάρχει κάποιο branch που να εμφανίζει *local correlation* (να μπορεί δηλαδή να προβλεφθεί ιδανικά με κάποιο local history predictor); Αν ναι, ποιο; Εξηγήστε την απάντησή σας.
- (ii) Υπάρχει κάποιο branch που να είναι *globally correlated* (να μπορεί δηλαδή να προβλεφθεί ιδανικά με κάποιο global history predictor); Αν ναι, ποιο; Εξηγήστε την απάντησή σας.

Γ. Ποια η διαφορά μεταξύ Reservation Stations και Reorder Buffer;

Θέμα 2ο (20%)

A. Δίνεται συμμετρικό, πολυεπεξεργαστικό σύστημα 4 επεξεργαστών, το οποίο χρησιμοποιεί πρωτόκολλο συνάφειας MESI και επιτρέπει cache-to-cache transfers. Το σύστημα χρησιμοποιεί διευθύνσεις μήκους 32 bits, είναι byte-addressable ενώ κάθε επεξεργαστής έχει μία 32B cache, direct mapped, write-back, με block μεγέθους 8 bytes. Για την ακολουθία των εντολών μνήμης που παρατίθενται στη συνέχεια, παρουσιάστε την κατάσταση των caches μετά από την εκτέλεση κάθε εντολής. Για όλες τις caches, δείξτε την κατάσταση MESI της cache line που εμπλέκεται σε κάθε

αναφορά, υποθέτοντας αρχικά ότι όλες οι cache lines είναι σε κατάσταση Invalid. Υποδείξτε επίσης ποιες αναφορές οδηγούν σε μεταφορά δεδομένων μεταξύ των caches ή/και από/προς τη κύρια μνήμη.

Εντολή	Cache state				Μεταφορά Δεδομένων	
	P0	P1	P2	P3	Από/Προς κύρια μνήμη	Μεταξύ caches
P0: read 0x00000000						
P0: write 0x00000002						
P0: write 0x00000000						
P1: read 0x00000004						
P2: read 0x0000000A						
P3: read 0x00000000						
P2: write 0x00000002						
P0: write 0x00000002						

B. Θεωρήστε ένα πολυεπεξεργαστικό σύστημα κοινής μνήμης 4 επεξεργαστών, καθώς και τον ακόλουθο κώδικα για 4 διεργασίες που εκτελούνται παράλληλα στο σύστημα. Οι μεταβλητές A και B είναι αποθηκευμένες στην κοινή μνήμη, με αρχική τιμή ίση με 0.

P1	P2	P3	P4
$a_1: A = 1$	$b_1: u=A$ $b_2: v=B$	$c_1: B=1$	$d_1: w=B$ $d_2: x=A$

(i) Βρείτε ποιοι συνδυασμοί τελικών τιμών για την τετράδα (u,v,w,x) είναι ακολουθιακά συνεπείς (sequentially consistent) και ποιοι όχι. Εξηγήστε τις απαντήσεις σας δείχνοντας πώς μπορούν να παραχθούν ή πώς παραβιάζουν το SC.

(ii) Θεωρήστε ότι το σύστημα υλοποιεί το πλέον relaxed memory model (π.χ. RMO). Ποιοι από τους συνδυασμούς του ερωτήματος (i) που δεν επιτρέπονται στο SC (αν υπάρχουν) μπορούν να εμφανιστούν τώρα; Επιλέξτε έναν από αυτούς και εισάγετε τον ελάχιστο αριθμό εντολών memory fence στον παραπάνω κώδικα ώστε να αποτρέπεται η εμφάνισή του.

Θέμα 3ο (30%)

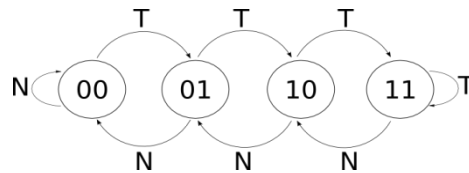
Δίνεται αρχιτεκτονική η οποία υλοποιεί τον αλγόριθμο Tomasulo χρησιμοποιώντας ROB για in-order commit εντολών. Το pipeline του επεξεργαστή περιέχει τα στάδια Issue (IS), Execute (EX), Write Result (WR) και Commit (CMT), αγνοούμε δηλαδή τα IF και ID. Ισχύουν επίσης τα ακόλουθα:

- Τα IS, WR, CMT απαιτούν 1 κύκλο.
- Ο επεξεργαστής είναι superscalar και μπορεί να δρομολογεί (Issue) και να ολοκληρώνει (Commit) μέχρι 2 εντολές ταυτόχρονα.
- Το σύστημα περιέχει 2 RS για προσθέσεις/αφαιρέσεις και 1 RS για πολλαπλασιασμούς/διαίρεσεις floating point αριθμών. Αντίστοιχα, για integer αριθμούς περιλαμβάνονται 4 RS για εντολές διακλάδωσης, αριθμητικές και λογικές εντολές.
- Το σύστημα περιλαμβάνει 4 non-pipelined functional units για πράξεις integer αριθμών. Όλες οι εντολές μεταξύ integer αριθμών διαρκούν 1 κύκλο.
- Το σύστημα περιλαμβάνει 2 non-pipelined floating point functional units, ένα για ADDD / SUBD και ένα για MULD / DIVD. Οι εντολές πρόσθεσης/αφαίρεσης διαρκούν 2 κύκλους, ενώ οι εντολές πολλαπλασιασμού/διαίρεσης 5 κύκλους.
- Για τις εντολές αναφοράς στη μνήμη, στο στάδιο EX γίνεται τόσο ο υπολογισμός της διεύθυνσης αναφοράς όσο και η προσπέλαση στη μνήμη. Το σύστημα περιλαμβάνει ένα Load και ένα Store Queue, το καθένα από τα οποία διαθέτει 2 θέσεις. Οι εντολές χρησιμοποιούν ένα ξεχωριστό,

pipelined functional unit για τον υπολογισμό της διεύθυνσης και διαρκούν 1 κύκλο στην περίπτωση Hit στην cache και 5 κύκλους σε περίπτωση Miss.

- Ο ROB έχει 7 θέσεις.
- Το σύστημα περιλαμβάνει 1 CDB. Σε περίπτωση που παραπάνω από μια εντολές θέλουν να το χρησιμοποιήσουν, τότε προτεραιότητα αποκτά η “παλαιότερη” εντολή (αυτή που προηγείται στο program order). Θεωρήστε ότι τα branches και τα stores δεν χρησιμοποιούν το CDB κατά τη διάρκεια του WR σταδίου τους.
- Η πρόβλεψη μιας εντολής διακλάδωσης υπό συνθήκη γίνεται ταυτόχρονα με τη δρομολόγηση της εντολής. Ο έλεγχος της πρόβλεψης γίνεται μόλις γίνει γνωστό το αποτέλεσμα της εντολής, δηλαδή στο στάδιο WR (κύκλος k). Σε περίπτωση σφάλματος, σταματά η εκτέλεση των εντολών του miss-predicted execution path και στον επόμενο κύκλο (κύκλος k+1) δρομολογείται η σωστή εντολή.
- Για τις εντολές διακλάδωσης υπό συνθήκη, το σύστημα χρησιμοποιεί τον παρακάτω πίνακα από 2-bit predictors. Δίνεται επίσης το FSM διάγραμμα του 2-bit predictor, ο οποίος προβλέπει T για value ≥ 2 και NT για τις υπόλοιπες.

Index	Value
0	11
1	10
2	00
3	01



Η δεικτοδότηση του πίνακα γίνεται χρησιμοποιώντας τον κατάλληλο αριθμό low order bits από το PC της εντολής. Ο επεξεργαστής υλοποιεί το ISA του MIPS, οι εντολές απέχουν μεταξύ τους 4 bytes και επομένως κατά τη δεικτοδότηση θα πρέπει να αγνοήσετε τα 2 λιγότερα σημαντικά bits του PC.

- Το σύστημα περιλαμβάνει μια fully associative cache με 2 cache lines και μέγεθος block = 32 bytes, η οποία χρησιμοποιεί πολιτική αντικατάστασης LRU. Αρχικά η cache είναι άδεια.
- Οι καταχωρητές R1, R2 περιέχουν τις διευθύνσεις των πρώτων στοιχείων δύο πινάκων A, B αριθμών διπλής ακρίβειας (μήκους 8 bytes ο καθένας). Οι πίνακες είναι ευθυγραμμισμένοι.

Δίνεται ο παρακάτω κώδικας :

```

0x0088000C LOOP: LD      F1, 0(R1)
0x00880010      LD      F2, 0(R2)
0x00880014      ADDD   F4, F1, F2
0x00880018      ADDD   F4, F4, F4
0x0088001C      LD      F1, 8(R1)
0x00880020      LD      F2, 8(R2)
0x00880024      ADDD   F5, F1, F2
0x00880028      ADDD   F5, F4, F5
0x0088002C      MULD   F6, F6, F5
0x00880030      ADDI   R1, R1, #16
0x00880034      ADDI   R2, R2, #16
0x00880038      SUBI   R4, R4, #1
0x0088003C      BNEZ   R4, LOOP
0x00880040      LD      F1, 16(R1)
0x00880044      ADDD   F1, F1, F6
0x00880048      SD      F1, 16(R1)

```

/* Ο αντίστοιχος κώδικας σε C */

```

for(int i=0; i<4; i+=2)
    x *= (2*(A[i]+B[i])+A[i+1]+B[i+1]);
A[6] = A[6] + x;

```

Δίνεται επίσης ότι $R_4 = 2$. Εκτελέστε τον παραπάνω κώδικα και δώστε τους χρόνους δρομολόγησης, εκτέλεσης και ολοκλήρωσης των εντολών σε έναν πίνακα όπως ο παρακάτω. Στο πεδίο “Σχόλιο” δικαιολογήστε τυχόν καθυστερήσεις μεταξύ IS-EX, EX-WR και WR-CMT καθώς και ακυρώσεις εντολών.

OP	IS	EX	WR	CMT	Σχόλιο
LD F1, 0(R1)	1	2-??	??	??	??

Θέμα 4ο (25%)

Θεωρήστε ένα σύστημα παράλληλης επεξεργασίας 2 επεξεργαστών, το οποίο χρησιμοποιεί το πρωτόκολλο συνάφειας MESI επιτρέποντας cache-to-cache transfers. Κάθε επεξεργαστής διαθέτει μία μόνο L1 data cache μεγέθους 64B, direct-mapped, write-back αποτελούμενη από 8 blocks. Αρχικά όλα τα blocks σε όλες τις caches είναι σε κατάσταση invalid (I). Στους επεξεργαστές εκτελείται παράλληλα ένα C πρόγραμμα το οποίο περιλαμβάνει τα εξής δεδομένα:

```
struct X {
    int a;
    int b;
}
struct X x[32];
```

Δίνεται ότι ο πίνακας x είναι ευθυγραμμισμένος, καθώς και ότι το μέγεθος ενός ακεραίου είναι 4 bytes. Ο κώδικας που εκτελείται στους 2 επεξεργαστές είναι ο ακόλουθος:

P1	P2
for (int i=0; i<32; i++) x[i].a += 1;	for (int i=0; i<32; i++) x[i].b += 1;

Το πρόγραμμα εκτελείται με τέτοιο τρόπο ώστε η εκτέλεση κάθε λειτουργίας μνήμης (read ή write) να εναλλάσσεται ανάμεσα στους επεξεργαστές με round-robin τρόπο (κυκλικά). π.χ. P1:read x[0].a, P2:read x[0].b, P1:write x[0].a, P2:write x[0].b, κ.ο.κ. Γενικά η εκτέλεση του προγράμματος είναι τέτοια ώστε να μην υπάρχουν επικαλύψεις (overlaps) μεταξύ των λειτουργιών. Έτσι, πρώτα ολοκληρώνεται η εκτέλεση μιας λειτουργίας του ενός επεξεργαστή (καθώς και ότι αυτή απαιτεί) και μετά ακολουθεί η λειτουργία μνήμης του άλλου επεξεργαστή.

Τέλος, ισχύουν οι ακόλουθοι χρόνοι για τις διάφορες λειτουργίες της μνήμης και του πρωτοκόλλου:

Λειτουργία	Κύκλοι
Cache hit (read ή write)	1
Μεταφορά 1 cache line (προς cache ή προς κύρια μνήμη)	8
BusRdX/BusRd bus transactions	2

(i) Ποιος είναι ο συνολικός χρόνος εκτέλεσης του προγράμματος; Εξηγήστε αναλυτικά την απάντησή σας παρουσιάζοντας την κατάσταση των caches μετά από την εκτέλεση κάθε αναφοράς. Συγκεκριμένα, για όλες τις caches, δείξτε την κατάσταση MESI της cache line που εμπλέκεται σε κάθε αναφορά καθώς και το χρόνο που χρειάζεται για την εκτέλεση της αναφοράς.

(ii) Χωρίς να τροποποιήσετε τον ορισμό της δομής, αλλάξτε τον κώδικα ώστε να επιτύχετε καλύτερο χρόνο εκτέλεσης. Φροντίστε οι αλλαγές που κάνετε να μην επηρεάζουν τη σημασιολογία του προγράμματος. Ποιος είναι τώρα ο συνολικός χρόνος εκτέλεσης του προγράμματος;