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

ΠΡΟΗΓΜΕΝΑ ΘΕΜΑΤΑ ΟΡΓΑΝΩΣΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

Εξετάσεις Οκτωβρίου 2007
Διάρκεια 3 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 Α4 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 Α4 που είναι ατομικά.

Επώνυμο:	
Όνομα:	
Εξάμηνο:	

Θέμα	Βαθμός
1	
2	
3	

Θέμα 1ο (35%):

Εξετάζουμε την εκτέλεση του εξής κώδικα:

```
Loop: add r1, r3, 50(r2)
      mul r6, r4, 100(r1)
      st r6, 50(r2)
      addi r3, r3, #2
      subi r2, r2, #8
      bnez r2, Loop
Exit:
```

Υποθέτουμε ότι έχουμε αρχιτεκτονική σωλήνωσης (pipelining) με τα εξής στάδια: IF ID ALU1 MEM ALU2 WB. Το στάδιο ALU1 χρησιμοποιείται για τον υπολογισμό των τελικών διευθύνσεων μνήμης (effective address calculation) για αριθμητικές πράξεις, loads και stores, όπως επίσης και για τον υπολογισμό των διευθύνσεων-στόχων (targets) των εντολών διακλάδωσης. Το στάδιο ALU2 χρησιμοποιείται για όλους τους υπόλοιπους υπολογισμούς, καθώς και για την επίλυση των διακλαδώσεων υπό συνθήκη. Όλες οι αριθμητικές πράξεις (MUL, ADD, SUB) χρειάζονται έναν κύκλο στο στάδιο ALU2 για να εκτελεστούν. Υποθέτουμε ότι όλες οι αναφορές στη μνήμη ικανοποιούνται από την κρυφή μνήμη σε 1 κύκλο (δεν υπάρχουν δηλαδή αστοχίες). Έστω ότι η αρχική τιμή του r2 είναι 800.

α) Για την 1η επανάληψη του παραπάνω βρόχου μέχρι και τη 1η εντολή της 2ης επανάληψης, βρείτε όλες τις εξαρτήσεις που υπάρχουν καθώς και την κατηγορία στην οποία ανήκουν (true dependence, output dependence, anti-dependence, control dependence).

β) Αρχικά, υποθέτουμε ότι η αρχιτεκτονική σωλήνωσης δε διαθέτει σχήμα προώθησης (forwarding). Επίσης, η εγγραφή σε κάποιον καταχωρητή γίνεται στο πρώτο μισό ενός κύκλου, ενώ η ανάγνωση από τον ίδιον καταχωρητή στο δεύτερο μισό του ίδιου κύκλου. Για την 1η επανάληψη του παραπάνω βρόχου, μέχρι και τη 1η εντολή της 2ης επανάληψης, χρησιμοποιείστε ένα διάγραμμα χρονισμού για να δείξετε τα διάφορα στάδια του pipeline από τα οποία διέρχονται οι εντολές σε αυτό το διάστημα εκτέλεσης. Ποιες από τις εξαρτήσεις που βρήκατε στο ερώτημα α δημιουργούν κινδύνους και πώς αντιμετωπίζονται αυτοί; Πόσοι κύκλοι απαιτούνται συνολικά για να ολοκληρωθεί ο παραπάνω βρόχος (για όλες τις επαναλήψεις του, όχι μόνο για την 1η);

γ) Για την ίδια ακολουθία εντολών, δείξτε όπως και πριν τον χρονισμό του pipeline, θεωρώντας όμως τώρα ότι υπάρχουν όλα τα δυνατά σχήματα προώθησης. Υποδείξτε και εξηγείστε τα σημεία όπου γίνονται προωθήσεις. Πόσοι κύκλοι απαιτούνται συνολικά για να ολοκληρωθεί ο βρόχος;

δ) Επεκτείνουμε την παραπάνω αρχιτεκτονική σωλήνωσης διπλασιάζοντας το εύρος της σωλήνωσης (εύρος 2 σωληνώσεων). Στο παραπάνω κομμάτι κώδικα, εφαρμόστε την τεχνική του ξεδιπλώματος βρόχων (loop unrolling) 2 φορές. Για την 1η επανάληψη του ξεδιπλωμένου βρόχου μέχρι και την 1η εντολή της 2ης επανάληψης, δείξτε τον χρονισμό του pipeline, θεωρώντας πάλι όλα τα δυνατά σχήματα προώθησης, και υποδείξτε όπως και πριν τα σημεία όπου γίνονται προωθήσεις. Για να μειωθούν οι εξαρτήσεις, αναδιατάξτε όπου είναι δυνατόν τις εντολές ώστε ανά 2 να είναι ανεξάρτητες μεταξύ τους και να μπορούν έτσι να εκτελεστούν παράλληλα, φροντίζοντας να μην επηρεάζεται η σημασιολογία του προγράμματος. Πόσοι κύκλοι απαιτούνται συνολικά για να ολοκληρωθεί ο βρόχος;

ε) Ορίζουμε ως branch penalty τους επιπλέον κύκλους που πρέπει να καθυστερήσει το pipeline για να φορτωθεί η εντολή που έπεται μιας εντολής διακλάδωσης.
- Πόσο είναι το branch penalty για το pipeline στο ερώτημα α της άσκησης;
- Έστω ότι αλλάζουμε το pipeline ώστε η επίλυση διακλαδώσεων να γίνεται στο ALU1 αντί του ALU2. Πόσο γίνεται τώρα το branch penalty; Υποθέτουμε ότι η βελτιστοποίηση αυτή βρίσκει αντίκρυσμα στο 65% των εντολών διακλάδωσης, ενώ για την υλοποίησή της απαιτείται αύξηση του κύκλου του ρολογιού κατά 10% (για όλες τις εντολές). Οι εντολές διακλάδωσης στατιστικά αντιστοιχούν στο 35% του συνόλου των εντολών. Αν το ιδανικό CPI (Clock-cycles Per Instruction) είναι 1, πόση είναι η συνολική χρονοβελτίωση (speedup) που μπορούμε να επιτύχουμε με αυτή τη τροποποίηση;

Θέμα 2ο (35%):

Εξετάζουμε τις ακόλουθες περιπτώσεις οργάνωσης της cache:
1. Πλήρως συσχετιστική (fully associative), χωρητικότητας 64KB, με μέγεθος block ίσο με 64 bytes και LRU πολιτική αντικατάστασης.
2. Ευθείας αντιστοίχισης (direct mapped), χωρητικότητας 8KB και μέγεθος block ίσο με 64 bytes.
3. Συσχέτισης 2 δρόμων (2-way set associative), χωρητικότητας 8KB, με μέγεθος block ίσο με 64 bytes και LRU πολιτική αντικατάστασης.

Σε όλες τις παραπάνω οργανώσεις, η μικρότερη μονάδα δεδομένων που μπορεί να διευθυνσιοδοτηθεί είναι το 1 byte, ενώ οι διευθύνσεις μνήμης έχουν εύρος 32 bits.

Για κάθε μία από τις οργανώσεις αυτές:
α.1) Υπολογίστε τον αριθμό των bits καθενός από τα επιμέρους πεδία στα οποία χωρίζεται μία διεύθυνση μνήμης. Παρουσιάστε ένα διάγραμμα που να δείχνει πώς διαχωρίζεται η διεύθυνση στα πεδία αυτά, και εξηγείστε τη σημασία του καθενός.
α.2) Παρουσιάστε σε block διάγραμμα την cache, τα επιμέρους πεδία μιας διεύθυνσης, και τον τρόπο που διασυνδέονται μεταξύ τους ώστε να οδηγήσουν στην προσπέλαση (εγγραφή/ανάγνωση) δεδομένων της cache. Εξηγείστε εν συντομία τη σημασία της κάθε διασύνδεσης στη διαδικασία εύρεσης δεδομένων στην

cache. Λάβετε υπόψη ότι κάθε cache line έχει ένα επιπλέον bit valid/invalid. Τι ποσοστό του συνολικού μεγέθους της cache αφιερώνεται για τα bits του tag σε κάθε μία από τις περιπτώσεις;
β) Βρείτε το συνολικό ποσοστό αστοχίας (miss rate) για τις αναφορές που γίνονται στη μνήμη από την εκτέλεση του ακόλουθου προγράμματος:

```
double a[1024], b[1024], c[1024];
int i;
for(i=0; i<1000; i++)
      a[i] = 35.0*b[i] + c[i+1];
```

Οι πίνακες περιέχουν στοιχεία κινητής υποδιαστολής διπλής ακρίβειας, μεγέθους 8 bytes το καθένα. Κάνουμε τις εξής υποθέσεις:

- Το πρόγραμμα εκτελείται σε έναν επεξεργαστή με μόνο ένα επίπεδο κρυφής μνήμης δεδομένων.
- Όλες οι μεταβλητές, πλην των στοιχείων των πινάκων, μπορούν να αποθηκευτούν σε καταχωρητές του επεξεργαστή, οπότε οποιαδήποτε αναφορά σε αυτές δεν συνεπάγεται προσπέλαση στην κρυφή μνήμη.
- Σε επίπεδο εντολών assembly, η σειρά με την οποία γίνονται οι αναφορές στους πίνακες είναι η εξής: b, c, a.
- Οι πίνακες a, b, c είναι αποθηκευμένοι με αυτή τη σειρά συνεχόμενα στη μνήμη, δηλαδή εκεί που τελειώνει ο ένας, αρχίζει ο επόμενος από την επόμενη θέση μνήμης.
- Αρχικά, η κρυφή μνήμη δεδομένων είναι άδεια.

Θέμα 3ο (30%):

Εξετάζουμε την εκτέλεση του ακόλουθου πολλαπλασιασμού πίνακα με διάνυσμα:

```
#define N 4
#define M 8
double c[N], a[N][M], b[M] ;
int i,j;

for(i=0; i<N; i++)
      for(j=0; j<M; j++)
            c[i] = c[i] + a[i][j]*b[j];
```

Όλοι οι πίνακες περιέχουν στοιχεία κινητής υποδιαστολής διπλής ακρίβειας, μεγέθους 8 bytes το καθένα. Κάνουμε τις εξής υποθέσεις:

- Το πρόγραμμα εκτελείται σε έναν επεξεργαστή με μόνο ένα επίπεδο κρυφής μνήμης δεδομένων. Η κρυφή μνήμη είναι πλήρως συσχετιστική (fully associative) με LRU πολιτική αντικατάστασης, και αποτελείται από 6 blocks των 16 bytes το καθένα.
- Όλες οι μεταβλητές, πλην των στοιχείων των πινάκων, μπορούν να αποθηκευτούν σε καταχωρητές του επεξεργαστή, οπότε οποιαδήποτε αναφορά σε αυτές δεν συνεπάγεται προσπέλαση στην κρυφή μνήμη.
- Σε επίπεδο εντολών assembly, η σειρά με την οποία γίνονται οι αναφορές στους πίνακες είναι η εξής: c, a, b, c.
- Αρχικά, η κρυφή μνήμη δεδομένων είναι άδεια.

α) Βρείτε το συνολικό ποσοστό αστοχίας (miss rate) για τις αναφορές που γίνονται στη μνήμη στον παραπάνω κώδικα.
β) Εφαρμόστε την τεχνική του blocking στον παραπάνω κώδικα για να μειώσετε τον αριθμό των misses. Πόσο γίνεται τώρα το ποσοστό αστοχίας; Εξηγείστε σε κάθε περίπτωση την επιλογή που κάνετε για την τιμή του blocking factor.

γ) Το σύνολο εντολών της αρχιτεκτονικής του επεξεργαστή διαθέτει μία ειδική εντολή prf(*addr). Η εντολή αυτή προ-φορτώνει στην κρυφή μνήμη ολόκληρο το μπλοκ που περιέχει τη λέξη που βρίσκεται στη διεύθυνση μνήμης addr. Χωρίς να αλλάξετε τη σειρά των loads για τις αναφορές στα στοιχεία των πινάκων, εισάγετε κλήσεις στην prf (1 ή περισσότερες) στον κώδικα που προέκυψε από το ερώτημα β ώστε να μειωθούν περαιτέρω τα misses. Πόσο γίνεται τώρα το ποσοστό αστοχίας;
Υποθέστε ότι 2 εκτελέσεις του βασικού σώματος του βρόχου (συμπεριλαμβανομένων των όποιων κλήσεων στην prf) είναι αρκετές ώστε να "καλυφθεί" ο χρόνος που απαιτείται για να έρθουν τα δεδομένα που ζητά η prf στην cache. Επιπλέον, μη λάβετε ειδική μέριμνα για την προφόρτωση δεδομένων στις αρχικές επαναλήψεις του loop, ούτε για τις έξτρα προφορτώσεις στις τελευταίες επαναλήψεις.