

Πανεπιστήμιο Θεσσαλίας  
Τμήμα Πληροφορικής

Οργάνωση Η/Υ

17 Δεκεμβρίου 2018

Δεύτερη Σειρά Ασκήσεων

παράδοση: τέλος εξεταστικής Ιανουαρίου ή Σεπτεμβρίου

**Άσκηση 1**

Θεωρήστε τη ΜΕΔ MIPS μερικά επικαλυπτόμενων εντολών που μελετήσαμε στο μάθημα. Υποθέστε ότι οι εντολές διακλάδωσης εκτελούνται και ολοκληρώνουν τα άλματά τους στη φάση εκτέλεσης (Φ3) του κύκλου εντολής χωρίς καθυστέρηση, ότι η ανάγνωση του ΦΚ γίνεται αμέσως μετά την εγγραφή του και στον ίδιο κύκλο μηχανής, και ότι δεν υπάρχουν δομικές εξαρτήσεις.

Έστω ο πιο κάτω κώδικας MIPS, ο οποίος εκτελείται στην εν λόγω ΜΕΔ:

```
          addi  $16, $0, 0
          lw    $21, 304($28)
L:        lw    $8, 4($21)
          add   $10, $8, $17
          lw    $8, 8($21)
          addi  $9, $10, -1
B1:       beq  $8, $0, T
          addi  $16, $16, 1
          lw    $10, 0($8)
          sub   $9, $10, $9
B2:       beq  $9, $0, T
          lw    $8, 4($8)
          add   $9, $9, $8
T:        sw   $9, 12($21)
          lw    $21, 0($21)
B3:       bne  $21, $0, L
          add   $2, $16, $0
          jr   $31
```

Σε μια συγκεκριμένη εκτέλεση, η διακλάδωση B3 δημιουργεί βρόχο 1000000 επαναλήψεων, ενώ η B1 εκτελεί άλμα σε ποσοστό 25% των επαναλήψεων και η B2 εκτελεί άλμα σε ποσοστό 2% των επαναλήψεων όπου εμφανίζεται.

A. Βρείτε όλες τις εξαρτήσεις από δεδομένα στον κώδικα, ως εξής: Να σχηματίσετε έναν πίνακα με γραμμές που αντιστοιχούν στις εντολές του κώδικα και στήλες που αντιστοιχούν στους καταχωρητές που μπορούν να συμμετέχουν σε εξαρτήσεις, και να σημειώσετε σ' αυτόν τις λειτουργίες ανάγνωσης και εγγραφής του κάθε καταχωρητή για κάθε εντολή. Στη συνέχεια να εντοπίσετε και να δώσετε όλες τις εξαρτήσεις τύπου AME, EMA και EME που βλέπετε, συμπεριλαμβανομένων αυτών που δρουν μεταξύ επαναλήψεων του βρόχου.

B. Ποιες από τις εξαρτήσεις που βρήκατε αποτελούν κίνδυνο στο μηχανισμό επικάλυψης, εάν η ΜΕΔ δεν διαθέτει μηχανισμό παροχέτευσης; Υποθέτοντας ότι οι εντολές διακλάδωσης εκτελούνται με στατική πρόβλεψη με βάση το πρόσρημο της μετατόπισης και ότι οι διευθύνσεις προορισμού των διακλαδώσεων υπολογίζονται στη φάση αποκωδικοποίησης (Φ2) του κύκλου εντολής, δώστε τα απαραίτητα διαγράμματα χρονισμού που καλύπτουν όλες τις δυνατές περιπτώσεις εκτέλεσης των διακλαδώσεων B1 και B2. Με βάση τα διαγράμματα αυτά, υπολογίστε

την επιτάχυνση του μέσου ρυθμού ολοκλήρωσης εντολών<sup>1</sup> στην παραπάνω ΜΕΔ, σε σύγκριση με την εκτέλεση σε ΜΕΔ της ίδιας συχνότητας αλλά χωρίς μηχανισμό μερικής επικάλυψης, στην οποία κάθε εντολή ολοκληρώνεται στον ελάχιστο αριθμό κύκλων μηχανής, σύμφωνα με το διάγραμμα καταστάσεων που μελετήσαμε στο μάθημα.

Γ. Θεωρήστε τώρα ότι η ΜΕΔ διαθέτει το μηχανισμό παροχέτευσης που μελετήσαμε στο μάθημα. Να επαναλάβετε τα παραπάνω: Εντοπίστε εξαρτήσεις που *ακόμα* αποτελούν κίνδυνο, δώστε τα διαγράμματα χρονισμού για την εκτέλεση του κώδικα, και υπολογίστε την επιτάχυνση του μέσου ρυθμού ολοκλήρωσης εντολών σε σύγκριση με τη ΜΕΔ χωρίς επικάλυψη. Πώς μπορείτε να αντιμετωπίσετε εξαρτήσεις δεδομένων που αποτελούν κίνδυνο παρά το μηχανισμό παροχέτευσης, χωρίς να παγώσετε το μηχανισμό επικάλυψης; Είναι δυνατό να εξαλειφτούν με αυτόν τον τρόπο *όλοι* οι κίνδυνοι στον παραπάνω κώδικα; Δείξτε πώς θα μπορούσατε να το επιτύχετε!

Δ. Οι διαδικασιακές εξαρτήσεις εντοπίζονται σε εντολές που ακολουθούν κάποια εντολή διακλάδωσης και εκτελούνται μόνο στη μία κατεύθυνση της διακλάδωσης, αλλά όχι και στις δύο. Ποιες από τις εντολές που ακολουθούν τις διακλαδώσεις B1 και B2 στον παραπάνω κώδικα είναι διαδικασιακά εξαρτημένες από αυτές; Εάν η ΜΕΔ διαθέτει μηχανισμό παροχέτευσης για την αντιμετώπιση εξαρτήσεων από δεδομένα, αλλά και μηχανισμό δυναμικής πρόβλεψης διακλαδώσεων όπου η διεύθυνση προορισμού γίνεται διαθέσιμη στο τέλος της φάσης ανάκλησης (Φ1) από κατάλληλη μνήμη διευθύνσεων προορισμού, δώστε τα διαγράμματα χρονισμού για την εκτέλεση του κώδικα, για όλους τους συνδυασμούς πρόβλεψης και αποτελέσματος εκτέλεσης για τις δύο διακλαδώσεις B1 και B2. Αν η πρόβλεψη της B1 έχει επιτυχία σε ποσοστό 70% για πρόβλεψη εκτέλεσης και σε ποσοστό 85% για πρόβλεψη μη εκτέλεσης, η πρόβλεψη της B2 έχει επιτυχία σε ποσοστό 99% και για τις δύο προβλέψεις, και αγνοώντας τις αποτυχίες στην πρόβλεψη της B3, υπολογίστε την επιτάχυνση του μέσου ρυθμού ολοκλήρωσης εντολών σχετικά με τη ΜΕΔ χωρίς επικάλυψη. Υπόδειξη: Το ποσοστό *πρόβλεψης* εκτέλεσης *άλματος* προκύπτει ως συνάρτηση του ποσοστού επιτυχίας πρόβλεψης και του ποσοστού εκτέλεσης του *άλματος*.

Ε. Θεωρήστε τώρα μια παραλλαγή της τελευταίας ΜΕΔ, όπου προστίθεται μια υπομονάδα σύγκρισης στις δύο εξόδους του ΦΚ, ώστε οι διακλαδώσεις να εκτελούνται στη φάση αποκωδικοποίησης (Φ2) του κύκλου εντολής, και όπου οι διακλαδώσεις έχουν έναν κύκλο καθυστέρησης στην εκτέλεση *άλματος*. Μετασηματίστε τον παραπάνω κώδικα για εκτέλεση στη νέα ΜΕΔ, εισάγοντας θέσεις καθυστέρησης, και συμπληρώνοντάς τες με το βέλτιστο δυνατό τρόπο. Στη χειρότερη περίπτωση μια θέση καθυστέρησης συμπληρώνεται με εντολή *nop*, ενώ στην καλύτερη περίπτωση συμπληρώνεται με κάποια άλλη εντολή που προκύπτει από αναδιάταξη του κώδικα, χωρίς όμως να παραβιάζονται οι εξαρτήσεις που ορίζουν την ορθή εκτέλεσή του. Αν υποστηρίζεται παροχέτευση και προς τη νέα υπομονάδα σύγκρισης, να ξαναδώσετε τα διαγράμματα χρονισμού για όλους τους συνδυασμούς πρόβλεψης και αποτελέσματος εκτέλεσης των B1 και B2 για το νέο κώδικα, και να ξαναυπολογίσετε την επιτάχυνση του μέσου ρυθμού ολοκλήρωσης εντολών με τις ίδιες υποθέσεις ποσοστών επιτυχίας πρόβλεψης διακλαδώσεων. Τι παρατηρείτε;

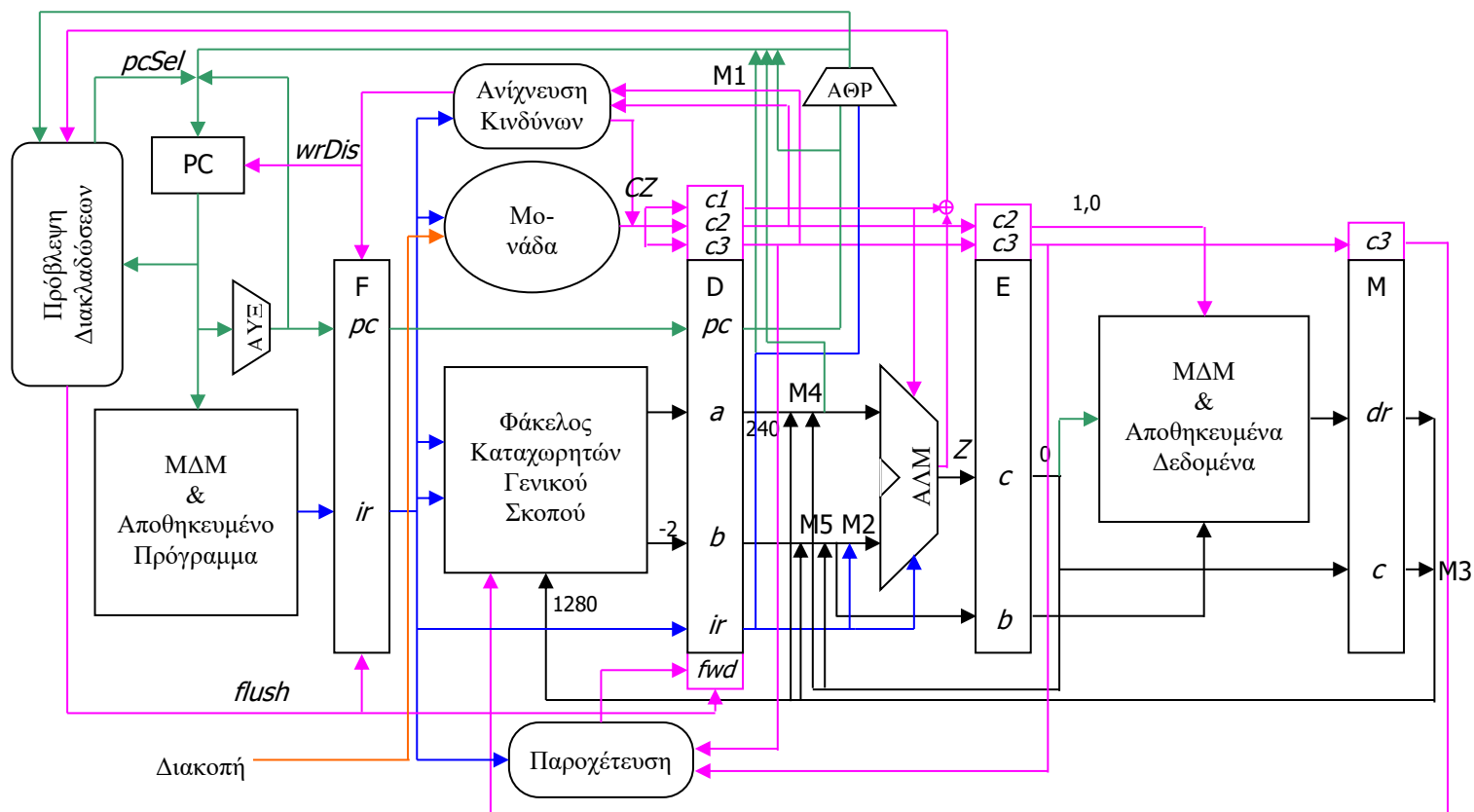
## Άσκηση 2

Έστω η ΜΕΔ MIPS μερικά επικαλυπτόμενων εντολών που μελετήσαμε στο μάθημα, με παροχέτευση και πρόβλεψη διακλαδώσεων, η οποία υποστηρίζει τις εντολές *add*, *sub*, *addi*, *lw*, *sw*, *bne*, *beq*, *j* και *jr*. Η ΜΕΔ αυτή ξαναδίνεται πιο κάτω, με μερικές συμπληρωματικές λεπτομέρειες όσο αφορά τη μεταφορά πληροφορίας μεταξύ των υπομονάδων της καθώς και μεταξύ των καταχωρητών επικάλυψης F, D, E και M.

Η πληροφορία εξόδου για κάθε έναν από τους καταχωρητές επικάλυψης είναι η πληροφορία που αποθηκεύτηκε σε αυτόν στον προηγούμενο κύκλο μηχανής, και αποτελεί ενδιάμεση είσοδο

<sup>1</sup> Ρυθμός ολοκλήρωσης εντολών = Αριθμός εντολών που ολοκληρώνουν την εκτέλεσή τους – δηλαδή δεν ακυρώνονται για οποιονδήποτε λόγο – στη μονάδα χρόνου.

για την εντολή στη φάση που περιορίζεται μεταξύ αυτού και του επόμενου καταχωρητή επικάλυψης. Για παράδειγμα, η τιμή 240 – που είναι η τιμή εξόδου του πεδίου D.a – αφορά την εντολή στη φάση εκτέλεσης (Φ3), ενώ η τιμή 0 – που είναι η τιμή εξόδου του πεδίου E.c – αφορά την εντολή στη φάση προσπέλασης μνήμης (Φ4). Αντίστοιχα, η πληροφορία εισόδου για καθέναν από τους καταχωρητές επικάλυψης αποθηκεύεται στο τέλος του παρόντος κύκλου, και αποτελεί ενδιάμεση έξοδο της εντολής στη φάση που περιορίζεται μεταξύ αυτού και του προηγούμενου καταχωρητή επικάλυψης. Για παράδειγμα, η τιμή -2 – που είναι η τιμή εισόδου του πεδίου D.b – αφορά την εντολή στη φάση αποκωδικοποίησης (Φ2). Οι δρόμοι πληροφορίας από δεξιά προς τα αριστερά μεταφέρουν τιμές που παράγονται από την εντολή που βρίσκεται στο δεξί άκρο τους. Για παράδειγμα, η τιμή 1280 – που είναι τιμή εισόδου δεδομένων του ΦΚ – αφορά την εντολή στη φάση αποθήκευσης αποτελέσματος (Φ5). Κάθε κύκλος εντολής θεωρείται ότι ξεκινά από τον PC και τελειώνει στο ΦΚ.



Επιπλέον των γραμμών δεδομένων, οι γραμμές σημάτων ελέγχου που φεύγουν από τη ME – και απεικονίζονται ως μία γραμμή για κάθε φάση επικάλυψης – θεωρούνται ότι μεταφέρουν όσα ψηφία είναι απαραίτητα για κάθε φάση. Έτσι, η γραμμή c1 περιέχει ψηφία που απευθύνονται στη Φ3, η γραμμή c2 περιέχει ψηφία που απευθύνονται στη Φ4 και η γραμμή c3 περιέχει ψηφία που απευθύνονται στη Φ5. Τα ψηφία ελέγχου δίνονται συμβολικά ως πεδία των c1, c2 και c3 στον πίνακα που ακολουθεί. Έτσι, με την ίδια λογική απεικόνιση στο διάγραμμα όπως για τις γραμμές δεδομένων, η τιμή (1,0) – που αποτελεί την έξοδο του πεδίου E.c2 – αναφέρεται σε σήματα για την εντολή στη Φ4, και ειδικότερα, η τιμή 1 αναφέρεται στο σήμα MemRead (πεδίο E.c2.MemRead) και η τιμή 0 αναφέρεται στο σήμα MemWrite (πεδίο E.c2.MemWrite). Οι επιμέρους τιμές αναγράφονται με τη σειρά των σημάτων στον πίνακα. Για απλούστευση, στο διάγραμμα φαίνονται μόνο οι κύριες γραμμές των σημάτων ελέγχου.

| Γραμμή | Πεδίο Ελέγχου | Λειτουργία  |
|--------|---------------|---|
| c1     | M1 (PCSrc)    | Επιλογή διεύθυνσης από D.ir (προορισμός άμεσου άλματος), D.a (έμμεσου άλματος), ΑΘΡ (άλματος διακλάδωσης) ή D.pc (επαναφορά διεύθυνσης σε λάθος πρόβλεψη άλματος).  |
|        | pcSel         | Επιλογή εισόδου του PC από την έξοδο του πολυπλέκτη M1 (όταν στη Φ3 εκτελείται άλμα χωρίς πρόβλεψη ή έχουμε διαπίστωση λανθασμένης πρόβλεψης), την έξοδο της Μονάδας Πρόβλεψης Διακλαδώσεων (όταν υπάρχει πρόβλεψη στη Φ1 και δεν συμβαίνει κάτι από τα προηγούμενα στη Φ3), ή από την έξοδο του αθροιστή ΑΥΞ (στις υπόλοιπες περιπτώσεις). |
|        | M2 (ALUSrc)   | Επιλογή εισόδου ALU2 μεταξύ της εξόδου του M5 (βλέπε πιο κάτω) και του πεδίου D.ir (άμεσο τελούμενο ή μετατόπιση).  |
|        | ALUop         | Επιλογή πράξης ΑΛΜ μεταξύ add (πρόσθεση από κωδικό λειτουργίας), sub (αφαίρεση από κωδικό λειτουργίας) και funct (πράξη από κωδικό τελεστή).  |
| c2     | MemRead       | Αν 1, τότε ανάγνωση μνήμης δεδομένων.   |
|        | MemWrite      | Αν 1, τότε εγγραφή μνήμης δεδομένων.  |
| c3     | M3 (MemtoReg) | Επιλογή δεδομένου προς αποθήκευση στο ΦΚ μεταξύ M.c (από την ΑΛΜ) και M.dr (από τη μνήμη).  |
|        | RegWrite      | Αν 1, τότε εγγραφή ΦΚ.  |
|        | rg            | Αριθμός καταχωρητή εγγραφής (ένας από τους rt και rd της αντίστοιχης εντολής).  |

Για την υποστήριξη παροχέτευσης και πρόβλεψης διακλαδώσεων, η ΜΕΔ περιλαμβάνει τις τρεις υπομονάδες που περιγράφονται στη συνέχεια:

- Η υπομονάδα «Παροχέτευση» εξετάζει τη ΜΕΔ για κινδύνους δεδομένων που αντιμετωπίζονται με παροχέτευση. Σε κάθε κύκλο μηχανής δέχεται τον κωδικό λειτουργίας της εντολής στη Φ2 και τους αριθμούς καταχωρητών που διαβάζονται από το ΦΚ, καθώς και τα σήματα regWrite και rg των δύο προηγούμενων εντολών που βρίσκονται στις δύο επόμενες φάσεις της ΜΕΔ. Για κάθε ταύτιση καταχωρητή που διαβάζεται – και είναι απαραίτητος για την εντολή που τον διαβάζει – με καταχωρητή που γράφεται, ενεργοποιείται το κατάλληλο σήμα παροχέτευσης, το οποίο αποθηκεύεται στο πεδίο fwd του καταχωρητή D, για να χρησιμοποιηθεί στον αμέσως επόμενο κύκλο μηχανής για επιλογή σε έναν από τους πολυπλέκτες M4 και M5.
- Η υπομονάδα «Ανίχνευση Κινδύνων» εξετάζει τη ΜΕΔ για κινδύνους από παραμένουσες εξαρτήσεις από δεδομένα που δημιουργούνται μετά από εντολές φόρτωσης. Σε κάθε κύκλο μηχανής δέχεται τον κωδικό λειτουργίας της εντολής στη Φ2 και τους αριθμούς καταχωρητών που διαβάζονται από το ΦΚ, καθώς και τα σήματα memRead και rg της προηγούμενης εντολής που βρίσκεται στην επόμενη φάση της ΜΕΔ. Αν το σήμα memRead έχει τιμή 1 και υπάρχει ταύτιση του καταχωρητή που γράφεται με έναν καταχωρητή που διαβάζεται – και είναι απαραίτητος για την εντολή που τον διαβάζει, η ΜΕΔ παγώνει για έναν κύκλο μηχανής, με μη επίτρεψη εγγραφής του PC και του καταχωρητή επικάλυψης F, καθώς και με μηδενισμό των σημάτων ελέγχου που παράγονται στον ίδιο κύκλο. Αυτό επιτυγχάνεται με ενεργοποίηση των σημάτων wrDis και CZ.
- Η υπομονάδα «Πρόβλεψη Διακλαδώσεων» υλοποιεί δυναμική πρόβλεψη διακλαδώσεων με δύο ψηφία ιστορίας ανά διακλάδωση, με τη βοήθεια μιας τοπικής μνήμης που αποθηκεύει την ιστορία και τη διεύθυνση προορισμού για κάθε διακλάδωση που απαντάται, και με βάση τον αλγόριθμο δυναμική πρόβλεψης που μελετήσαμε στο μάθημα και περιγράφεται στο βιβλίο των Patterson-Hennessy. Σε έλλειψη πληροφορίας στην υπομονάδα πρόβλεψης, αποδίδονται στην διακλάδωση μηδενικά αρχικά ψηφία ιστορίας. Σε κάθε κύκλο μηχανής, η υπομονάδα πρόβλεψης δέχεται την τιμή του PC, και αν η τιμή αυτή αντιστοιχεί

σε διεύθυνση διακλάδωσης που έχει συναντηθεί στο παρελθόν και έχει πρόβλεψη εκτέλεσης άλματος, παρέχει μια διεύθυνση προορισμού στον πολυπλέκτη pcSel. Η Φ3 ενημερώνει την υπομονάδα για την έκβαση κάθε διακλάδωσης, και σε περίπτωση άλματος παρέχει σε αυτήν τη διεύθυνση προορισμού για να αποθηκευτεί στη μνήμη διευθύνσεων προορισμού. Σε κάθε λανθασμένη πρόβλεψη, ή για εκτέλεση αλμάτων χωρίς πρόβλεψη (διακλάδωσης που δεν έχουν ξανασυναντηθεί, καθώς και άμεσων ή έμμεσων αλμάτων), ενεργοποιείται το σήμα flush, που μηδενίζει τις τιμές στις εισόδους των καταχωρητών επικάλυψης F και D, συμπεριλαμβανομένων των σημάτων που παράγει η μονάδα ελέγχου – εκτός του pcSel, μέσω του οποίου ο PC λαμβάνει τη σωστή διεύθυνση από τον πολυπλέκτη M1.

Θεωρήστε τον κώδικα MIPS της Άσκησης 1, ο οποίος φορτώνεται στη μνήμη, με την πρώτη εντολή στη θέση 0x40038b08, με σκοπό να εκτελεστεί στην παραπάνω ΜΕΔ. Πριν την εκτέλεση του κώδικα, υποθέστε ότι οι καταχωρητές του ΦΚ που μας ενδιαφέρουν περιέχουν τις ακόλουθες τιμές:

```

$2:   0x1
$8:   0x0d2f
$9:   0x0
$10:  0xa
$16:  0x40b41c00
$17:  0x4
$21:  0x0d30
$28:  0x100a8200
$31:  0x40a00f0c

```

και οι θέσεις μνήμης που αναφέρονται από τον κώδικα περιέχουν:

```

mem[0x1003ab00]: 0x32
mem[0x1003ab04]: 0xc
mem[0x1003d328]: 0x11
mem[0x1003d32c]: 0x3
mem[0x10088b00]: 0x10088e04
mem[0x10088b04]: 0x1a
mem[0x10088b08]: 0x1003ab00
mem[0x10088b0c]: 0x0
mem[0x10088b28]: 0x0
mem[0x10088b2c]: 0xe
mem[0x10088b30]: 0x1003d328
mem[0x10088b34]: 0xffffffff801
mem[0x10088e04]: 0x10088b28
mem[0x10088e08]: 0x11
mem[0x10088e0c]: 0x0
mem[0x10088e10]: 0xfffffffffff
mem[0x100a8330]: 0x10088b00

```

Επίσης, η ΜΕΔ είναι άδεια και τα σήματα ελέγχου, όπως και όλα τα ψηφία ιστορίας της υπομονάδας πρόβλεψης διακλαδώσεων, είναι μηδενισμένα. Υποθέστε ότι οι διευθύνσεις των εντολών διακλάδωσης δεν εμφανίζουν σύγκρουση στην προσπέλαση της μνήμης ιστορίας και διευθύνσεων προορισμού. Δείτε πώς ο κώδικας εκτελείται στην παραπάνω ΜΕΔ, ως εξής:

A. Να δώσετε το ενιαίο διάγραμμα χρονισμού που προκύπτει από την εκτέλεση, με βάση τις τιμές καταχωρητών και μνήμης που αναφέρονται πιο πάνω.

B. Να δώσετε όλες τις τιμές που παράγονται μέσα στη ΜΕΔ, είτε ως δεδομένα είτε ως σήματα ελέγχου, ξεχωριστά για κάθε κύκλο μηχανής, από τη φάση ανάκλησης της πρώτης εντολής, μέχρι τη φάση αποθήκευσης αποτελέσματος της τελευταίας εντολής. Για το σκοπό αυτό, να χρησιμοποιήσετε το επισυναπτόμενο αρχείο «datapath.pdf», αντιγράφοντάς το όσες φορές χρειαστεί, δημιουργώντας σε κάθε κενή ΜΕΔ ένα στιγμιότυπο ενός κύκλου μηχανής, τοποθετώντας πάνω στη καθεμία τις ζητούμενες τιμές σημάτων και δεδομένων.

Υπόδειξη: Στην παρούσα άσκηση η ροή του κώδικα είναι πραγματική, και όχι στατιστική, και ο βρόχος εκτελεί ακριβώς 3 επαναλήψεις.

### Άσκηση 3

Θεωρήστε τη ΜΕΔ MIPS μερικά επικαλυπτόμενων εντολών της Άσκησης 2. Σ' αυτή τη ΜΕΔ, η διάρκεια του κύκλου μηχανής καθορίζεται από τη φάση με τη μέγιστη διάρκεια, που συνήθως αντιστοιχεί στη μικρολειτουργία με τη μέγιστη διάρκεια. Κάποιες φορές, μια μικρολειτουργία μεγάλης διάρκειας μπορεί να υλοποιηθεί με μικρότερες επικαλυπτόμενες φάσεις, έτσι ώστε η ΜΕΔ να υλοποιείται συνολικά με ένα μεγαλύτερο αριθμό σύντομων φάσεων, επιτυγχάνοντας αύξηση στη συχνότητα λειτουργίας του επεξεργαστή. Υποθέστε ότι στα πλαίσια μιας τέτοιας βελτιστοποίησης, η προσπέλαση της κρυφής μνήμης εντολών υλοποιείται σε δύο επικαλυπτόμενες φάσεις, ενώ η προσπέλαση της κρυφής μνήμης δεδομένων υλοποιείται σε τρεις επικαλυπτόμενες φάσεις, έτσι ώστε και οι πέντε αυτές φάσεις να έχουν ίση διάρκεια, και καμία άλλη μικρολειτουργία της ΜΕΔ να μην απαιτεί χρόνο μεγαλύτερο από τη διάρκεια αυτή. Επομένως, η ΜΕΔ υλοποιείται συνολικά με οκτώ επικαλυπτόμενες φάσεις, με κύκλο μηχανής διάρκειας μίας φάσης.

Α. Δείξτε πώς διαμορφώνεται το σχεδιάγραμμα της ΜΕΔ με την παραπάνω αλλαγή στις προσπελάσεις της κρυφής μνήμης.

Β. Μελετήστε τις ανάγκες παροχέτευσης και βρείτε σε πόσο βάθος εντολών πρέπει να υπάρχει υποστήριξη παροχέτευσης σε αυτή τη ΜΕΔ. Σε ποιες περιπτώσεις εξαρτήσεις από δεδομένα δεν αντιμετωπίζονται με παροχέτευση, και μέχρι ποια απόσταση εντολών;

Γ. Ποιο είναι το κόστος αποτυχίας στην πρόβλεψη διακλαδώσεων σε αυτή τη ΜΕΔ, όταν αυτές εκτελούνται κανονικά στην ΑΛΜ και, όταν χρειάζεται, ενημερώνουν τον PC στο τέλος της ίδιας φάσης;

Δ. Θεωρήστε τον κώδικα της Άσκησης 1, ο οποίος εκτελείται στη νέα ΜΕΔ, με αρχικά δεδομένα εκτέλεσης όπως δίνονται στην Άσκηση 2. Δώστε το ενιαίο διάγραμμα χρονισμού των τριών επαναλήψεων που εκτελούνται. Πόσο πιο γρήγορο πρέπει να είναι το ρολόι της νέας ΜΕΔ, ώστε αυτή να έχει καλύτερη απόδοση για τη συγκεκριμένη εκτέλεση;

Ε. Έστω ότι ο κώδικας της Άσκησης 1 είναι μέρος μιας εφαρμογής, με εντολές που είναι κατά 27% εντολές φόρτωσης, 12% εντολές αποθήκευσης, 14% εντολές διακλάδωσης, 2% άλλες εντολές άλματος και κατά 45% εντολές ΑΛΜ. Υπολογίστε το μέσο αριθμό κύκλων μηχανής ανά εντολή (CPI) για την εφαρμογή όταν εκτελείται στην παραπάνω ΜΕΔ, και: (α) για κάθε εντολή φόρτωσης, υπάρχει πιθανότητα  $1/2$  για την πρώτη,  $1/4$  για τη δεύτερη,  $1/8$  για την τρίτη, κόκ, εντολή που ακολουθεί να είναι εξαρτημένη από αυτή, και (β) οι διακλαδώσεις προβλέπονται σωστά σε ποσοστό 80%. Θεωρήστε ότι τα άλματα χωρίς συνθήκη εκτελούνται όπως και οι διακλαδώσεις στη φάση εκτέλεσης.

ΣΤ. Για να μειώσουμε τον αριθμό κύκλων μηχανής που χάνονται, επειδή οι διακλαδώσεις και τα άλλα άλματα εκτελούνται στη φάση εκτέλεσης, τροποποιούμε τη ΜΕΔ, εισάγοντας καθυστέρηση  $N$  κύκλων μηχανής στην εκτέλεση άλματος, για κάποιο συγκεκριμένο  $N$ . Έτσι, οι  $N$  εντολές που ακολουθούν εκτελούνται πάντα, είτε το άλμα πραγματοποιείται είτε όχι. Δυστυχώς, ο μεταγωγτιστής δε μπορεί πάντα να βρίσκει τις  $N$  εντολές που απαιτούνται και να τις τοποθετεί αμέσως μετά την αντίστοιχη εντολή, οπότε αναγκάζεται να τοποθετήσει στις θέσεις καθυστέρησης εντολές nop, που δεν κάνουν τίποτα χρήσιμο, και διατηρούν επομένως την απώλεια κύκλων μηχανής. Επαναλάβετε τον υπολογισμό του ερωτήματος Ε, αν τα άλματα εκτελούνται με  $N$  κύκλους καθυστέρησης, για τις περιπτώσεις  $N = 1, 2$  και  $3$ , και αν ο μεταγωγτιστής μπορεί να τοποθετεί κάποια χρήσιμη εντολή στην πρώτη θέση καθυστέρησης με πιθανότητα 75%, και, αν  $N > 1$  και συμπληρώσει την πρώτη θέση, στη δεύτερη θέση καθυστέρησης με πιθανότητα 20%, και τέλος, αν  $N > 2$  και συμπληρώσει τη δεύτερη θέση, στην τρίτη θέση καθυστέρησης με πιθανότητα 2%, αφήνοντας εντολές nop στις υπόλοιπες θέσεις καθυστέρησης.

## Άσκηση 4

Υποθέστε ότι έχουμε τη ΜΕΔ MIPS μερικά επικαλυπτόμενων εντολών του ερωτήματος ΣΤ της Άσκησης 3 με  $N = 2$ . Σε αυτή τη ΜΕΔ θέλουμε να εκτελέσουμε τον κώδικα της Άσκησης 1, ο οποίος διαμορφώνεται αρχικά ως εξής:

```

        addi   $16, $0, 0
        lw     $21, 304($28)
L:      lw     $8, 4($21)
        add    $10, $8, $17
        lw     $8, 8($21)
        addi   $9, $10, -1
B1:     beq   $8, $0, T
        nop
        nop
        addi   $16, $16, 1
        lw     $10, 0($8)
        sub   $9, $10, $9
B2:     beq   $9, $0, T
        nop
        nop
        lw     $8, 4($8)
        add   $9, $9, $8
T:      sw    $9, 12($21)
        lw     $21, 0($21)
B3:     bne   $21, $0, L
        nop
        nop
        add   $2, $16, $0
        jr    $31
        nop
        nop

```

Στον παραπάνω κώδικα οι εντολές `nop` βρίσκονται στις θέσεις καθυστέρησης των αντίστοιχων διακλαδώσεων και έμμεσου άλματος. Ο μεταγλωττιστής δεν έκανε καμία προσπάθεια να τοποθετήσει στις θέσεις αυτές χρήσιμες εντολές, και γι' αυτό τις συμπλήρωσε με εντολές `nop`, που εκτελούνται κανονικά, χωρίς όμως να αλλάζουν την κατάσταση της ΜΕΔ.

Με τις υποθέσεις ποσοστών επιτυχίας πρόβλεψης διακλαδώσεων της Άσκησης 1:

A. Υπολογίστε το μέσο χρόνο ολοκλήρωσης μιας επανάληψης του παραπάνω βρόχου, δηλαδή το μέσο χρονικό διάστημα σε κύκλους μηχανής από την ολοκλήρωση μιας επανάληψης μέχρι την ολοκλήρωση της επόμενης, δίνοντας τα διαγράμματα χρονισμού που απαιτούνται για όλους τους συνδυασμούς πρόβλεψης και αποτελέσματος των διακλαδώσεων B1 και B2, υποθέτοντας πάντα επιτυχημένη πρόβλεψη εκτέλεσης άλματος για τη διακλάδωση B3.

B. Επειδή οι εντολές `nop` δεν κάνουν κάτι χρήσιμο, ισοδυναμούν με παγώματα στο μηχανισμό επικάλυψης, με τη διαφορά ότι αυτά είναι φανερά και δεν προκύπτουν αυτόματα από τη ΜΕΔ. Επαναδιατάξτε τώρα τις εντολές του κώδικα, ώστε και να τοποθετήσετε χρήσιμες εντολές στις θέσεις καθυστέρησης των διακλαδώσεων, αλλά και να εξαλείψετε τα παγώματα στις φορτώσεις, αν αυτό είναι εφικτό. Η αναδιάταξη πρέπει να γίνει με προσοχή, ώστε να μην αλλάζει το αποτέλεσμα της εκτέλεσης του κώδικα.

Για το σκοπό αυτό, δεν πρέπει σε καμία περίπτωση να παραβιαστούν εξαρτήσεις δεδομένων τύπου ΑΜΕ. Αντίθετα, εξαρτήσεις δεδομένων τύπου ΕΜΑ και ΕΜΕ μπορούν να παραβιαστούν, μόνο αν μπορείτε να μετονομάσετε τους εμπλεκόμενους καταχωρητές, ώστε οι εξαρτήσεις ουσιαστικά να εξαλειφθούν. Για παράδειγμα, στην ακολουθία εντολών:

```

E1:    lw     $8, 4($21)
E2:    add   $12, $8, $17
E3:    lw     $8, 8($21)

```

η εξάρτηση E1 από την E2 προς την E3 λόγω του \$8 μπορεί να εξαλειφτεί, αν στις εντολές E1 και E2 ο καταχωρητής \$8 μετονομαστεί σε κάποιον άλλο διαθέσιμο καταχωρητή, έστω τον \$13:

```
E1:  lw    $13, 4($21)
E2:  add   $12, $13, $17
E3:  lw    $8, 8($21)
```

Τώρα η εντολή E3 μπορεί να μετακινηθεί και να τοποθετηθεί πριν την E2, με αποτέλεσμα να γλυτώσουμε έναν κύκλο παγώματος.

Όσο αφορά τις διαδικασιακές εξαρτήσεις, θεωρήστε τρεις περιπτώσεις:

(1) Δεν επιτρέπεται να παραβιάσετε τις διαδικασιακές εξαρτήσεις στο σώμα του βρόχου. Καμία εντολή που είναι εξαρτημένη διαδικασιακά από κάποια διακλάδωση δε μπορεί να μετακινηθεί και να τοποθετηθεί σε θέση που δεν είναι διαδικασιακά εξαρτημένη από τη διακλάδωση, είτε πριν είτε μετά από αυτήν.

(2) Μπορείτε να παραβιάσετε διαδικασιακές εξαρτήσεις, αλλά όχι για εντολές που μπορεί να προκαλέσουν ειδική περίπτωση στον επεξεργαστή. Υποθέστε για τους σκοπούς της άσκησης, ότι οι μόνες τέτοιες εντολές είναι οι εντολές προσπέλασης μνήμης.

(3) Μπορείτε να παραβιάσετε διαδικασιακές εξαρτήσεις για οποιαδήποτε εντολή.

Για τις δύο τελευταίες περιπτώσεις, πρέπει να προσέξετε ώστε η όποια παραβίαση διαδικασιακής εξάρτησης να μην παραβιάζει εξάρτηση δεδομένων τύπου AME, είτε σε καταχωρητή είτε στη μνήμη. Για παράδειγμα, οι εντολές φόρτωσης και πρόσθεσης μεταξύ της B2 και της T δε μπορούν να τοποθετηθούν στις δύο θέσεις καθυστέρησης της B2, διότι έτσι παραβιάζεται η εξάρτηση AME μεταξύ της εντολής αφαίρεσης και της εντολής αποθήκευσης, καθώς η εντολή πρόσθεσης δίνει άλλη τιμή στον καταχωρητή \$9 από αυτή που κανονικά θα έφθανε στην εντολή αποθήκευσης σε περίπτωση εκτέλεσης άλματος της B2.

Για κάθε μία από τις τρεις περιπτώσεις ξεχωριστά, εφαρμόστε αναδιάταξη στις εντολές του σώματος του βρόχου, και ξαναυπολογίστε το μέσο χρόνο ολοκλήρωσης μιας επανάληψης του βρόχου, δίνοντας πάλι τα αναγκαία διαγράμματα χρονισμού. Τι παρατηρείτε;

Γ. Μια μέθοδος επιτάχυνσης στην εκτέλεση ενός βρόχου είναι το *ζεδίπλωμα* του βρόχου (loop unrolling). Ξεδίπλωμα βρόχου  $n$  φορές σημαίνει ότι το σώμα του βρόχου γράφεται  $n$  διαδοχικές φορές, με την τελευταία διακλάδωση να οδηγεί προς την αρχή του ξεδιπλωμένου βρόχου. Έτσι, ένας αρχικός βρόχος  $N$  επαναλήψεων μεταβάλλεται σε βρόχο  $\lceil N/n \rceil$  επαναλήψεων. Για παράδειγμα, το ξεδίπλωμα 2 φορές του βρόχου του ερωτήματος Α οδηγεί στον κώδικα:

```

      addi  $16, $0, 0
      lw    $21, 304($28)
L:    lw    $8, 4($21)
      add   $10, $8, $17
      lw    $8, 8($21)
      addi  $9, $10, -1
B11:  beq   $8, $0, T1
      nop
      nop
      addi  $16, $16, 1
      lw    $10, 0($8)
      sub   $9, $10, $9
B21:  beq   $9, $0, T1
      nop
      nop
      lw    $8, 4($8)
      add   $9, $9, $8
T1:   sw    $9, 12($21)
      lw    $21, 0($21)
B31:  beq   $21, $0, E
      nop
      nop
      lw    $8, 4($21)
      add   $10, $8, $17
```



```

        lw      $8, 8($21)
        addi   $9, $10, -1
B12:    beq    $8, $0, T2
        nop
        nop
        addi   $16, $16, 1
        lw    $10, 0($8)
        sub   $9, $10, $9
B22:    beq    $9, $0, T2
        nop
        nop
        lw    $8, 4($8)
        add   $9, $9, $8
T2:     sw    $9, 12($21)
        lw    $21, 0($21)
B32:    bne   $21, $0, L
        nop
        nop
E:      add   $2, $16, $0
        jr   $31
        nop
        nop

```

Συνήθως το ξεδίπλωμα συνοδεύεται από αναδιάταξη των εντολών στο σώμα του νέου βρόχου, ώστε να εξαλειφθούν προβλήματα από παγώματα ή εντολές nop στον αρχικό βρόχο. Όμως, η αναδιάταξη αυτή πρέπει να γίνεται με προσοχή, επειδή το ξεδίπλωμα οδηγεί στην εμφάνιση καινούργιων εξαρτήσεων στο σώμα του νέου βρόχου, καθώς οι ίδιοι καταχωρητές χρησιμοποιούνται σε κάθε επανάληψη του αρχικού βρόχου. Πολλές από αυτές τις εξαρτήσεις είναι τύπου EMA ή EME, και αντιμετωπίζονται με μετονομασία καταχωρητών.

Για καθεμία από τις τρεις περιπτώσεις του ερωτήματος Β, τροποποιήστε τον πιο πάνω ξεδιπλωμένο κώδικα με αναδιάταξη των εντολών του βρόχου, ώστε να εξαλείψετε τα παγώματα κατά την εκτέλεση του βρόχου και τις εντολές nop των θέσεων καθυστέρησης των διακλαδώσεων, αν αυτό είναι εφικτό. Υποθέστε ότι έχετε διαθέσιμους όλους τους υπόλοιπους καταχωρητές της αρχιτεκτονικής, ώστε να αλλάξετε όποιους καταχωρητές θέλετε για την αναδιάταξη αυτή. Δώστε τα διαγράμματα χρονισμού που προκύπτουν από την εκτέλεση του κώδικα για όλους τους συνδυασμούς πρόβλεψης και αποτελέσματος των διακλαδώσεων στο σώμα του νέου βρόχου. Υπολογίστε το νέο μέσο χρόνο ολοκλήρωσης μιας επανάληψης του βρόχου και συγκρίνετέ τον με αυτόν που βρήκατε νωρίτερα, δεδομένου ότι μία επανάληψη του νέου βρόχου αντιστοιχεί σε δύο επαναλήψεις του αρχικού. Τι παρατηρείτε; Μπορείτε να καταλάβετε αν με μεγαλύτερο βαθμό ξεδιπλώματος θα εξαλείψετε περισσότερα παγώματα και εντολές nop;

## Άσκηση 5

Θεωρήστε τη ΜΕΔ MIPS μερικά επικαλυπτόμενων εντολών. Υποθέστε ότι εκτός από την υπομονάδα εκτέλεσης πράξεων σταθερής υποδιαστολής, η ΜΕΔ αυτή διαθέτει και δύο υπομονάδες εκτέλεσης πράξεων κινητής υποδιαστολής, οι οποίες είναι συνδεδεμένες με ένα ΦΚ κινητής υποδιαστολής:

- (i) Η υπομονάδα πρόσθεσης αριθμών κινητής υποδιαστολής απαιτεί 3 κύκλους μηχανής για την εκτέλεση μιας πρόσθεσης, που αντιστοιχούν σε 3 διαδοχικές φάσεις εκτέλεσης Φ31-Φ33.
- (ii) Η υπομονάδα πολλαπλασιασμού αριθμών κινητής υποδιαστολής απαιτεί 5 κύκλους μηχανής για την εκτέλεση ενός πολλαπλασιασμού, που αντιστοιχούν σε 5 διαδοχικές φάσεις εκτέλεσης Φ31-Φ35.

Εντολές MIPS εκτελούνται στην πιο πάνω ΜΕΔ με τους ακόλουθους περιορισμούς:

1. Οι δύο πρώτες φάσεις του κύκλου εντολής είναι κοινές για όλες τις εντολές, ενώ οι υπόλοιπες είναι διαφορετικές για κάθε υπομονάδα εκτέλεσης.

2. Οι φάσεις εκτέλεσης σε διαφορετικές υπομονάδες της ΜΕΔ επικαλύπτονται πλήρως. Έτσι, μια πράξη πρόσθεσης κινητής υποδιαστολής μπορεί να εκτελείται ταυτόχρονα με μια λογική πράξη, αλλά και με μια πράξη πολλαπλασιασμού κινητής υποδιαστολής.
3. Ο μηχανισμός παροχέτευσης μπορεί να προωθεί δεδομένα είτε μέσα σε κάθε υπομονάδα εκτέλεσης, είτε μεταξύ των υπομονάδων εκτέλεσης.
4. Όλες οι εντολές προσπέλασης στη μνήμη, ακόμα κι αν αφορούν καταχωρητές κινητής υποδιαστολής, εκτελούνται στην υπομονάδα σταθερής υποδιαστολής, με τον τρόπο που γνωρίζουμε. Για παράδειγμα, η εντολή `lwc1`, αν και αποθηκεύει το αποτέλεσμα της σε κάποιο καταχωρητή κινητής υποδιαστολής, εκτελείται σαν εντολή σταθερής υποδιαστολής. Σαν αποτέλεσμα των παραπάνω, ο κύκλος εντολής κινητής υποδιαστολής δεν περιλαμβάνει φάση προσπέλασης μνήμης.
5. Οι εντολές μεταφοράς δεδομένων μεταξύ των δύο ΦΚ εκτελούνται στην υπομονάδα σταθερής υποδιαστολής. Έτσι, η εντολή `mfc1` διαβάζει το τελούμενο εισόδου από έναν καταχωρητή κινητής υποδιαστολής κατά τη φάση αποκωδικοποίησης, και αν δεν υπάρξει παροχέτευση από κάποια από τις δύο υπομονάδες κινητής υποδιαστολής κατά τη φάση εκτέλεσης, το αποθηκεύει κατά τη φάση αποθήκευσης αποτελέσματος σε έναν καταχωρητή σταθερής υποδιαστολής.
6. Δύο εγγραφές σε διαφορετικό ΦΚ μπορούν να γίνονται ταυτόχρονα. Μ' άλλα λόγια, οι φάσεις αποθήκευσης αποτελέσματος μιας εντολής σταθερής και μιας εντολής κινητής υποδιαστολής μπορούν να επικαλύπτονται. Αντίθετα, σε περίπτωση που δύο εντολές κινητής υποδιαστολής ολοκληρώσουν ταυτόχρονα την εκτέλεσή τους, θα προχωρήσει μόνο η παλαιότερη στη φάση αποθήκευσης αποτελέσματος και η άλλη θα παγώσει για έναν κύκλο μηχανής.

A. Μελετήστε την εκτέλεση του παρακάτω κώδικα MIPS σε αυτή τη ΜΕΔ και κατασκευάστε το αντίστοιχο διάγραμμα χρονισμού, δείχνοντας τις εξαρτήσεις από δεδομένα και εξηγώντας πώς αυτές αντιμετωπίζονται:

```
lwc1  $f1, -4($7)
lwc1  $f4, 0($8)
add.s $f2, $f1, $f4
ori   $6, $0, 0xfff
sll   $6, $6, 23
mfc1  $10, $f2
swc1  $f2, 0($7)
and   $10, $10, $6
srl   $10, $10, 23
sw    $10, 0($4)
```

B. Να επαναλάβετε για τον παρακάτω κώδικα:

```
lwc1  $f1, -4($7)
lwc1  $f4, 0($8)
ori   $6, $0, 0xfff
sll   $6, $6, 23
lwc1  $f3, -4($9)
mul.s $f2, $f1, $f3
mfc1  $11, $f2
add.s $f2, $f1, $f4
and   $11, $11, $6
srl   $11, $11, 23
sw    $11, 0($5)
mfc1  $10, $f2
and   $10, $10, $6
add.s $f2, $f2, $f3
srl   $10, $10, 23
sw    $10, 0($4)
swc1  $f2, 0($7)
```

## Άσκηση 6

Θεωρήστε τη ΜΕΔ MIPS μερικά επικαλυπτόμενων εντολών της Άσκησης 2, με τη διαφορά ότι αντί διαχωρισμένης, διαθέτουμε μια ενοποιημένη κρυφή μνήμη εντολών και δεδομένων μεγέθους 4 Kbytes. Η μνήμη αυτή δημιουργεί δομικούς κινδύνους μεταξύ εντολών που κάνουν προσπέλαση μνήμης για δεδομένα και επόμενων εντολών που βρίσκονται στη φάση ανάκλησης, με αποτέλεσμα οι δεύτερες να παγώνουν όσο γίνεται προσπέλαση δεδομένων.

A. Στην παραπάνω ΜΕΔ εκτελούμε τον κώδικα της Άσκησης 1, με αρχικά δεδομένα εκτέλεσης αυτά που δίνονται στην Άσκηση 2. Δώστε πάλι το ενιαίο διάγραμμα χρονισμού που προκύπτει, και βρείτε την ακολουθία διευθύνσεων μνήμης που παράγεται κατά την εκτέλεση αυτή.

B. Υποθέτοντας ότι η κρυφή μνήμη είναι αρχικά άδεια (μ' άλλα λόγια όλα τα πλαίσια είναι αρχικά άκυρα), και ότι τα δεδομένα προσκομίζονται με τακτική «όταν απαιτείται», βρείτε πόσες και ποιες είναι οι επιτυχίες και οι αποτυχίες από τις προσπελάσεις μνήμης που κάνει ο κώδικας, ποιες αντικαταστάσεις πλαισίων συμβαίνουν, καθώς και τι τιμές υπάρχουν μετά την τελευταία από τις προσπελάσεις (α) στο ψηφίο εγκυρότητας κάθε πλαισίου και (β) στο πεδίο ετικέτας κάθε πλαισίου, όταν:

- (1) Η κρυφή μνήμη είναι άμεσης απεικόνισης με μέγεθος πλαισίου 16 bytes.
- (2) Η κρυφή μνήμη είναι άμεσης απεικόνισης με μέγεθος πλαισίου 256 bytes.
- (3) Η κρυφή μνήμη είναι 2-τρόπων συνόλου συσχέτισης με μέγεθος πλαισίου 16 bytes και LRU στρατηγική απελευθέρωσης.
- (4) Η κρυφή μνήμη είναι 2-τρόπων συνόλου συσχέτισης με μέγεθος πλαισίου 16 bytes και FIFO στρατηγική απελευθέρωσης.
- (5) Η κρυφή μνήμη είναι 2-τρόπων συνόλου συσχέτισης με μέγεθος πλαισίου 256 bytes και LRU στρατηγική απελευθέρωσης.
- (6) Η κρυφή μνήμη είναι 2-τρόπων συνόλου συσχέτισης με μέγεθος πλαισίου 256 bytes και FIFO στρατηγική απελευθέρωσης.
- (7) Η κρυφή μνήμη είναι 4-τρόπων συνόλου συσχέτισης με μέγεθος πλαισίου 16 bytes και LRU στρατηγική απελευθέρωσης.
- (8) Η κρυφή μνήμη είναι 4-τρόπων συνόλου συσχέτισης με μέγεθος πλαισίου 16 bytes και FIFO στρατηγική απελευθέρωσης.
- (9) Η κρυφή μνήμη είναι 4-τρόπων συνόλου συσχέτισης με μέγεθος πλαισίου 256 bytes και LRU στρατηγική απελευθέρωσης.
- (10) Η κρυφή μνήμη είναι 4-τρόπων συνόλου συσχέτισης με μέγεθος πλαισίου 256 bytes και FIFO στρατηγική απελευθέρωσης.
- (11) Η κρυφή μνήμη είναι πλήρους συσχέτισης με μέγεθος πλαισίου 16 bytes και LRU στρατηγική απελευθέρωσης.
- (12) Η κρυφή μνήμη είναι πλήρους συσχέτισης με μέγεθος πλαισίου 256 bytes και LRU στρατηγική απελευθέρωσης.

## Άσκηση 7

Θεωρήστε ένα σύστημα ιεραρχίας μνήμης με κρυφή (ενός επιπέδου), κύρια, και εικονική μνήμη. Η κρυφή μνήμη δέχεται λογικές διευθύνσεις, και σε περίπτωση αποτυχίας απαιτείται η φυσική διεύθυνση για την προσπέλαση του επόμενου επιπέδου της ιεραρχίας. Η φυσική διεύθυνση βρίσκεται με τη βοήθεια (α) μιας συσχετιστικής μνήμης TLB, η οποία διατηρεί τις πιο πρόσφατες απεικονίσεις από λογικές σε φυσικές διευθύνσεις, και (β) ενός πίνακα σελίδων που χρησιμοποιεί απεικόνιση δύο επιπέδων για την εύρεση των φυσικών διευθύνσεων από τις λογικές. Επειδή η κρυφή μνήμη της ιεραρχίας μνήμης δέχεται λογικές διευθύνσεις, μπορεί να αναζητήσει κάποιο δεδομένο την ίδια στιγμή που η TLB αναζητάει τη φυσική διεύθυνση.

Έστω η εκτέλεση του κώδικα που μελετήσατε στις 12 περιπτώσεις κρυφής μνήμης στην Άσκηση 6. Πέρα από τη συμπεριφορά της κρυφής μνήμης, την οποία λαμβάνετε από την επίλυση της Άσκησης 6, θεωρήστε επιπλέον ότι οι προσπελάσεις που αστοχούν στην κρυφή μνήμη βρίσκουν την απεικόνιση από λογική σε φυσική διεύθυνση μνήμης στην TLB σε ποσοστό 98%.

Στις υπόλοιπες περιπτώσεις δημιουργείται ειδική περίπτωση στον επεξεργαστή, ώστε να εξεταστεί ο πίνακας σελίδων και να αναζητηθεί σε αυτόν η απεικόνιση σε φυσική διεύθυνση. Έστω ότι σε ένα 0,001% των τελευταίων περιπτώσεων έχουμε σφάλμα σελίδας, οπότε παρεμβάλλεται το λειτουργικό σύστημα, ώστε να φέρει τη ζητούμενη σελίδα από το δίσκο. Τότε, και αν χρειαστεί η σελίδα αυτή να αντικαταστήσει κάποια άλλη, η απεικόνιση της τελευταίας στην TLB ακυρώνεται.

Υποθέστε ότι ο χρόνος προσπέλασης της TLB είναι 2ns και της κρυφής μνήμης είναι 1ns. Ο χρόνος προσπέλασης της κύριας μνήμης είναι 50ns. Ο πίνακας σελίδων δε μεταφέρεται ποτέ στην κρυφή μνήμη. Το πρώτο επίπεδο απεικόνισης του πίνακα σελίδων βρίσκεται μόνιμα στην κύρια μνήμη, αλλά το δεύτερο όχι, κι έτσι μπορεί να οδηγήσει σε σφάλμα σελίδας, με την πιθανότητα που αναγράφτηκε παραπάνω. Κάθε περίπτωση σφάλματος σελίδας έχει κόστος 1ms μέχρι η σελίδα να φορτωθεί από το δίσκο στην κύρια μνήμη.

Με βάση τα παραπάνω, να υπολογίσετε το μέσο χρόνο προσπέλασης μνήμης για το συγκεκριμένο πρόγραμμα, για κάθε περίπτωση κρυφής μνήμης της Άσκησης 6, και να υπολογίσετε το συνολικό χρόνο εκτέλεσης που προκύπτει, λαμβάνοντας υπόψη το αντίστοιχο κόστος προσπέλασης μνήμης. Υποθέστε ότι η διάρκεια κύκλου μηχανής καθορίζεται από το χρόνο προσπέλασης της κρυφής μνήμης, και ότι σε κάθε περίπτωση αστοχίας στην προσπέλαση της κρυφής μνήμης, η ΜΕΔ παγώνει μέχρι να ολοκληρωθεί η προσπέλαση.

## Άσκηση 8

Ακολουθεί μια λίστα δομικών στοιχείων, τα οποία χρησιμοποιούνται σε ένα σύστημα E/E με σύγχρονο δίαυλο επεξεργαστή-μνήμης συχνότητας 1200 MHz και έναν ή περισσότερους προσαρμογείς E/E που διασυνδέουν διαύλους E/E στο δίαυλο επεξεργαστή-μνήμης.

- Σύστημα μνήμης: Το σύστημα μνήμης έχει διασύνδεση εύρους 32 bits στην οποία εκτελεί μεταφορές τεσσάρων λέξεων τη φορά. Το σύστημα μνήμης χρησιμοποιεί ξεχωριστές γραμμές διευθύνσεων και δεδομένων και για εγγραφές στη μνήμη δέχεται μια λέξη σε κάθε κύκλο ρολογιού για 4 κύκλους ρολογιού, και στη συνέχεια απαιτεί επιπλέον 4 κύκλους ρολογιού πριν ολοκληρώσει την αποθήκευση των λέξεων και μπορέσει να δεχθεί την επόμενη συναλλαγή.
- Διασυνδέσεις άμεσης προσπέλασης μνήμης (DMA): Οι προσαρμογείς E/E χρησιμοποιούν DMA για τη μεταφορά των δεδομένων μεταξύ των διαύλων E/E και του διαύλου επεξεργαστή-μνήμης. Η μονάδα DMA επιτελεί τη διαιτησία στο δίαυλο επεξεργαστή-μνήμης και στέλνει και λαμβάνει μπλοκ τεσσάρων λέξεων από και προς το σύστημα μνήμης. Ο ελεγκτής DMA μπορεί να εξυπηρετήσει έως 8 δίσκους. Η εκκίνηση μια νέας λειτουργίας E/E απαιτεί 0.1 ms, και στη διάρκεια του χρόνου αυτού δε μπορεί να ξεκινήσει άλλη E/E από αυτόν τον ελεγκτή (αλλά μπορεί να γίνει χειρισμός λειτουργιών που είναι σε εκκρεμότητα).
- Δίαυλος E/E: Ο δίαυλος E/E είναι ένας σύγχρονος δίαυλος με εύρος ζώνης 160 MB/sec σε σταθερή λειτουργία. Κάθε μεταφορά έχει μήκος μια λέξη.
- Δίσκοι: Οι δίσκοι έχουν μέσο χρόνο καθυστέρησης για αναζήτηση και περιστροφή 8 ms, ενώ έχουν εύρος ζώνης ανάγνωσης/εγγραφής 64 MB/sec, όταν εκτελούν μεταφορά.

Βρείτε το χρόνο που απαιτείται για την ανάγνωση ενός τομέα 32 KB από ένα δίσκο στη μνήμη, με την παραδοχή ότι αυτή είναι η μόνη δραστηριότητα στο δίαυλο.

**Προσοχή:** Οι θεωρητικές ασκήσεις παραδίνονται μόνο χειρόγραφες, κάθε άσκηση σε ξεχωριστές σελίδες. Η παράδοση συνοδεύεται από υποχρεωτική σύντομη εξέταση.

Τρίτη Σειρά Ασκήσεων (Εργαστηριακές Ασκήσεις)  
παράδοση: τέλος εξεταστικής Ιανουαρίου ή Σεπτεμβρίου

#### Άσκηση 4

Σχεδιάστε μια μονάδα διαίρεσης σταθερής υποδιαστολής για μη προσημασμένους αριθμούς, με τη βοήθεια μιας γλώσσας περιγραφής υλικού (VHDL ή Verilog) και του προγράμματος ModelSim, η οποία να υλοποιεί τον επαναληπτικό αλγόριθμο διαίρεσης με διαδοχικές αφαιρέσεις και ολισθήσεις. Ο διαιρετέος να είναι μεγέθους 64 bits, ενώ ο διαιρέτης να είναι μεγέθους 32 bits. Οι έξοδοι της μονάδας, το πηλίκο και το υπόλοιπο, να είναι επίσης μεγέθους 32 bits. Μια πρόσθετη έξοδος της μονάδας, μεγέθους 1 bit, σηματοδοτεί προσπάθεια διαίρεσης με το 0.

Θεωρήστε σα βάση τον κλασικό αλγόριθμο, ο οποίος σε κάθε επανάληψη, κι αφού εκτελέσει αφαίρεση του διαιρέτη από το μερικό υπόλοιπο, επιλέγει σα νέα τιμή του μερικού υπολοίπου το αποτέλεσμα της αφαίρεσης, μόνο αν αυτό είναι μη αρνητικό, αλλιώς διατηρεί την παλιά τιμή του, ενώ θέτει αντίστοιχα τιμές 1 ή 0 στο επόμενο ψηφίο του πηλίκου. Ο αλγόριθμος αυτός υπολογίζει το πηλίκο με ρυθμό 1 bit ανά επανάληψη, επειδή κάθε φορά εξετάζει αν ο διαιρέτης χωράει *το πολύ μία* φορά στο μερικό υπόλοιπο.

Στην άσκηση αυτή σας ζητείται να υλοποιήσετε έναν αλγόριθμο, ο οποίος να προχωράει με ρυθμό 2 bits ανά επανάληψη, εκτελώντας ταυτόχρονα πολλαπλές αφαιρέσεις από το μερικό υπόλοιπο, τόσο του διαιρέτη, όσο και πολλαπλασίων αυτού – διπλάσιου και τριπλάσιου, ώστε να βρίσκει 2 νέα ψηφία πηλίκου. Η επιλογή των ψηφίων αυτών θα γίνεται με βάση τα πρόσημα που έχουν τα αποτελέσματα των αφαιρέσεων.

Για την άσκηση αυτή, μπορείτε να προχωρήσετε με τα ακόλουθα βήματα:

1. Σχεδιάστε τη ΜΕΔ της μονάδας. Θα χρησιμοποιήσετε πολλαπλές υπομονάδες αφαίρεσης, έναν πολυπλέκτη για την επιλογή του αποτελέσματος, και έναν καταχωρητή ολίσθησης, που θα πρέπει να κάνει ολίσθηση κατά δύο ψηφία σε κάθε παλμό ρολογιού. Θα χρειαστείτε ακόμα κι έναν ή περισσότερους επιπλέον καταχωρητές, για να κρατάτε το ή τα πολλαπλάσια του διαιρέτη που χρησιμοποιείτε στις πολλαπλές αφαιρέσεις. Κάποιο από, ή και τα δύο πολλαπλάσια θα πρέπει να υπολογίζονται έναν κύκλο ρολογιού πριν αρχίσει η εφαρμογή του επαναληπτικού αλγόριθμου. Υλοποιήστε σε VHDL ή Verilog τη ΜΕΔ της μονάδας με όλα τα πιο πάνω στοιχεία.
2. Προσθέστε τα σήματα ελέγχου που χρειάζεστε. Καθορίστε τις καταστάσεις της μονάδας και σχεδιάστε τη μηχανή καταστάσεων. Με βάση τη μηχανή καταστάσεων, υλοποιήστε το κύκλωμα ελέγχου της μονάδας, ώστε να παράγει τα σήματα που προσθέσατε, στο σωστό κύκλο μηχανής.
3. Προσθέστε τη δυνατότητα ελέγχου για διαίρεση με 0, ο οποίος θα δίνει τιμή σε κατάλληλο σήμα εξόδου της μονάδας. Πιθανά να χρειαστείτε μια νέα κατάσταση για την ενεργοποίηση αυτού του σήματος.
4. Ελέγξτε την ορθότητα του κυκλώματός σας με κατάλληλα διανύσματα δοκιμής.

Να γράψετε δομημένο (structural) πρόγραμμα, με βασικά δομικά στοιχεία καταχωρητές και πύλες. Σε όσες από τις πιο πάνω φάσεις σχεδίασης χρειάζεστε κάποια υπομονάδα πρόσθεσης ή αφαίρεσης, χρησιμοποιήστε την υπομονάδα πρόσθεσης πρόβλεψης κρατουμένου που υλοποιήσατε στην Άσκηση 3, αφού την επεκτείνετε για αφαίρεση.

#### Άσκηση 5

Θεωρήστε ένα φάκελο καταχωρητών (ΦΚ), μιας υποθετικής αρχιτεκτονικής AmA (Amazing Architecture), όπως περιγράφεται στη συνέχεια:

- Περιλαμβάνει 56 καταχωρητές γενικού σκοπού, με μήκος λέξης 32 bits, οι οποίοι προσπελαύνονται με 32 διευθύνσεις των 5 bits, από \$0 έως \$31, και με τη βοήθεια ειδικών σημάτων ελέγχου που περιγράφονται στη συνέχεια.

- Ο καταχωρητής με διεύθυνση \$0 έχει πάντα τιμή 0.
- Διαθέτει 2 θύρες ανάγνωσης A και B, και 1 θύρα εγγραφής D, κι επομένως θα δέχεται συνολικά 3 διευθύνσεις καταχωρητών για ανάγνωση και εγγραφή, έστω addrA, addrB και addrD αντίστοιχα, καθώς και ένα σήμα επίτρεψης ανάγνωσης και ένα σήμα επίτρεψης εγγραφής, έστω RegRead και RegWrite αντίστοιχα.
- Οι διευθύνσεις \$16 έως και \$23 αντιστοιχούν σε τέσσερις ομάδες καταχωρητών, που επιλέγονται με τη βοήθεια ειδικού σήματος ελέγχου, έστω threadID, εύρους 2 bits. Το σήμα αυτό δεν επηρεάζει την προσπέλαση άλλων διευθύνσεων.
- Οι καταχωρητές με διεύθυνση \$23 είναι τέσσερις μετρητές προγράμματος PC1, PC2, PC3 και PC4, και ο φάκελος θα πρέπει να υποστηρίζει αύξηση της τιμής κατά 1 του καταχωρητή που καθορίζει το σήμα threadID, όταν κατάλληλο σήμα επίτρεψης, έστω PCinc, είναι ενεργοποιημένο. Όταν συμβαίνει εγγραφή στη διεύθυνση \$23, η εγγραφή αυτή έχει προτεραιότητα έναντι της αύξησης κατά 1, ανεξάρτητα της τιμής του σήματος PCinc.
- Οι καταχωρητές \$24 έως και \$31 υλοποιούνται σε αντίστοιχες δομές ουράς βάθους 4 θέσεων η καθεμία. Σε κάθε τέτοιο καταχωρητή, ένας βοηθητικός καταχωρητής ολίσθησης των 4 bits δείχνει το ενεργό βάθος της ουράς, ώστε μία εγγραφή να εισάγει μία λέξη από το πίσω μέρος της ουράς, στη θέση που υποδεικνύεται από τον βοηθητικό καταχωρητή. Μία ανάγνωση αφαιρεί μία λέξη από το μπροστινό μέρος της ουράς, οπότε όλες οι υπόλοιπες λέξεις μέχρι το ενεργό βάθος της ουράς ολισθαίνουν προς τα εμπρός. Επειδή η ανάγνωση τροποποιεί την κατάσταση του καταχωρητή, γίνεται απαραίτητο το σήμα επίτρεψης ανάγνωσης που αναφέρθηκε. Το σήμα αυτό δεν αφορά άλλους καταχωρητές.
- Μη επιτρεπτές προσπελάσεις των καταχωρητών \$24 έως και \$31 περιλαμβάνουν εγγραφή σε γεμάτη ουρά, ανάγνωση σε άδεια ουρά ή ταυτόχρονη ανάγνωση και εγγραφή στην ίδια ουρά. Τέτοιες προσπελάσεις ενεργοποιούν κατάλληλα σήματα εξόδου του ΦΚ, έστω inAccessA, inAccessB, inAccessD, για καθεμία από τις τρεις προσπελάσεις που μπορεί να δέχεται ο ΦΚ σε έναν κύκλο μηχανής, καθώς και inAccess για την τελευταία περίπτωση. Τα σήματα αυτά μπορούν να σηματοδοτούν αντίστοιχες εξαιρέσεις στον επεξεργαστή.
- Όλες οι αλλαγές κατάστασης του ΦΚ, συμπεριλαμβανομένων των ολισθήσεων των βοηθητικών καταχωρητών ολίσθησης και της αύξησης των μετρητών προγράμματος κατά 1, συμβαίνουν σύγχρονα, με τη βοήθεια εισόδου ωρολογιακών παλμών, έστω CLK.

Σχεδιάστε τον παραπάνω ΦΚ σε κάποια γλώσσα περιγραφής υλικού (VHDL ή Verilog) με το πρόγραμμα ModelSim. Να γράψετε δομημένο (structural) πρόγραμμα, με βασικά δομικά στοιχεία καταχωρητές μεγέθους 32 bits, τέσσερις καταχωρητές-μετρητές ίδιου μεγέθους, 8 δομές ουράς ίδιου μεγέθους των 4 θέσεων, 3 αποκωδικοποιητές  $5 \times 32$  και πύλες τριών καταστάσεων. Ειδικότερα, μπορείτε να προχωρήσετε με τα ακόλουθα βήματα:

1. Υλοποιήστε τους 48 καταχωρητές με διευθύνσεις από \$0 έως και \$23. Προσέξτε ιδιαίτερα την υλοποίηση των καταχωρητών \$0 και \$23.
2. Υλοποιήστε τους 8 καταχωρητές ουράς με διευθύνσεις από \$24 έως και \$31. Προσέξτε ότι κάθε ουρά μπορεί να δεχτεί μέχρι δύο αναγνώσεις στον ίδιο κύκλο, αλλά δε μπορεί να δεχτεί ταυτόχρονη ανάγνωση και εγγραφή, ούτε ανάγνωση όταν είναι άδεια, ούτε εγγραφή όταν είναι γεμάτη.
3. Προσθέστε τους αποκωδικοποιητές για τη διεθυνσιοδότηση του ΦΚ.
4. Προσθέστε τα σήματα ελέγχου PCinc για αύξηση των καταχωρητών με διεύθυνση \$23, RegRead και RegWrite για επίτρεψη ανάγνωσης και εγγραφής του ΦΚ, threadID για την επιλογή της οκτάδας \$16-\$23, και CLK. Προσέξτε ιδιαίτερα την περίπτωση εγγραφής στη διεύθυνση \$23.
5. Προσθέστε τις εξόδους μη επιτρεπτών προσπελάσεων για τους καταχωρητές \$24-\$31.
6. Ελέγξτε την ορθότητα του κυκλώματός σας με κατάλληλα προγράμματα δοκιμής, τα οποία να γράφουν τιμές στο ΦΚ και στη συνέχεια να τις διαβάζουν. Δώστε αυθαίρετες αρχικές τιμές στους τέσσερις καταχωρητές \$23.

## Άσκηση 6

Να γράψετε ένα πρόγραμμα C ή Java που να προσομοιώνει τη ΜΕΔ MIPS με μερική επικάλυψη που μελετήσαμε στο μάθημα σε επίπεδο κύκλου μηχανής. Το πρόγραμμα θα πρέπει να οδηγείται από μια μεταβλητή ρολογιού, σε κάθε νέα τιμή της οποίας θα πρέπει να ενεργοποιούνται οι υπομονάδες που βρίσκονται στα πέντε στάδια επικάλυψης της ΜΕΔ, και με είσοδο τις εξόδους των καταχωρητών επικάλυψης και του PC, θα παράγουν έξοδο που θα αποθηκεύεται πίσω στους καταχωρητές επικάλυψης, στον PC, στο ΦΚ ή στη μνήμη. Μ' άλλα λόγια, το πρόγραμμα θα προσομοιώνει ως ανεξάρτητες συναρτήσεις όλες τις υπομονάδες της ΜΕΔ από καθένα από τα πέντε στάδια επικάλυψης, και οι συναρτήσεις αυτές θα καλούνται μέσα από ένα βρόχο που θα υλοποιεί την προσομοίωση κύκλο-κύκλο. Έτσι:

1. Δηλώστε τη μεταβλητή ρολογιού, τη μεταβλητή PC, τους πίνακες που θα υλοποιούν το ΦΚ και τις κρυφές μνήμες, καθώς και τις δομές που θα υλοποιούν τους καταχωρητές επικάλυψης.
2. Γράψτε τις συναρτήσεις που θα υλοποιούν τη λειτουργία κάθε υπομονάδας της ΜΕΔ, συμπεριλαμβανομένων των PC-ΑΥΞ, μνήμης εντολών, ΦΚ, μονάδας ελέγχου, ΑΛΜ, ΑΘΡ, μνήμης δεδομένων, μονάδας ανίχνευσης κινδύνων, μονάδας παροχέτευσης και μονάδας πρόβλεψης διακλαδώσεων (στατικής με βάση το πρόσημο της μετατόπισης). Για παράδειγμα, η μνήμη δεδομένων θα λειτουργεί με βάση μια συνάρτηση που με παραμέτρους τα σήματα ελέγχου memRead και memWrite, τη διεύθυνση προσπέλασης και την τιμή εγγραφής, θα επιστρέφει την τιμή ανάγνωσης. Μπορείτε να ενσωματώσετε σε κάθε συνάρτηση τους όποιους πολυπλέκτες εισόδου της υπομονάδας ή να γράψετε ξεχωριστές συναρτήσεις υλοποίησης κάθε πολυπλέκτη.
3. Γράψτε τον κύριο βρόχο του προσομοιωτή που για κάθε κύκλο μηχανής θα καλεί τις παραπάνω συναρτήσεις με τις κατάλληλες παραμέτρους. Οι έξοδοι των συναρτήσεων θα αποθηκεύονται πίσω στους καταχωρητές επικάλυψης. Προσέξτε τη σειρά κλήσης των συναρτήσεων, ώστε να μην χρησιμοποιείτε ως είσοδο κάποια συνάρτησης την έξοδο άλλης από διαφορετικό στάδιο επικάλυψης στον ίδιο κύκλο μηχανής.
4. Προσθέστε στον κώδικά σας μεταβλητές συλλογής στατιστικών στοιχείων, όπως συνολικό πλήθος εντολών, πλήθος εντολών ανά κατηγορία, πλήθος παγωμάτων από κινδύνους δεδομένων, πλήθος αποτυχιών πρόβλεψης διακλαδώσεων, πλήθος επιτυχημένων παροχέτευσεων, καθώς και πλήθος εντολών nop που εκτελούνται.
5. Γράψτε κάποιο πρόγραμμα δοκιμής σε γλώσσα MIPS και κωδικοποιήστε το με οποιονδήποτε τρόπο σας βολεύει στη μνήμη εντολών. Αρχικοποιήστε τη μνήμη δεδομένων, αν σας είναι απαραίτητο, καθώς και τον PC.
6. Επεξηγήστε ό,τι κάνετε με κατάλληλα σχόλια στον κώδικά σας.

Ακόμα μπορείτε προαιρετικά να υλοποιήσετε (α) κάποιο μοντέλο δυναμικής πρόβλεψης διακλαδώσεων, και (β) κάποιο απλό μοντέλο ιεραρχίας μνήμης.

Το πρόγραμμά σας θα πρέπει να υποστηρίζει τουλάχιστον τις εντολές lw, sw, add, sub, addi, slt, slti, beq, bne, j, jr, jal, jalr, nop και break, με την τελευταία να σημαίνει τερματισμό εκτέλεσης. Εκτός αν υλοποιήσετε ιεραρχία μνήμης, υποθέστε ότι κάθε προσπέλαση μνήμης ολοκληρώνεται επιτυχώς σε έναν κύκλο μηχανής.

Με βάση τα στατιστικά στοιχεία που θα συγκεντρώσετε, να υπολογίσετε το μέσο αριθμό κύκλων μηχανής ανά εντολή (μέσο CPI) για το πρόγραμμα δοκιμής που εκτελείται στον προσομοιωτή. Αν εκτελέστηκαν εντολές nop, να τις μετρήσετε σαν παγώματα. Με την ολοκλήρωση της εκτέλεσης θα πρέπει να εκτυπώνονται όλα τα παραπάνω στατιστικά στοιχεία, μαζί με το μέσο CPI που προκύπτει.

Καλό θα είναι το πρόγραμμα δοκιμής που γράψατε να εκτελείται για αρκετές χιλιάδες κύκλων.

**Προσοχή:** Οι εργαστηριακές ασκήσεις παραδίνονται μόνο ηλεκτρονικά. Η παράδοση συνοδεύεται από υποχρεωτική σύντομη επίδειξη.