

Σύνολο μονάδων 100. Απαντήστε σύντομα και περιεκτικά όλες τις ερωτήσεις. Αν χρειαστεί να κάνετε κάποιες υποθέσεις για να προχωρήσετε στην λύση, γράψτε καθαρά όλες αυτές τις υποθέσεις. Δείξτε και την διαδικασία της λύσης μαζί με το τελικό αποτέλεσμα. Καλή επιτυχία!

Ερώτηση 1: (5 Μονάδες) Τι περιγράφει ο νόμος του Amdahl και ποια η εξίσωση του ;

Ερώτηση 2: (6 Μονάδες) Αναφέρετε 3 τρόπους βελτιστοποίησης των κρυφών μνημών που επιτυγχάνουν μείωση του χρόνου αποτυχίας (Miss penalty). Περιγράψτε πως επιτυγχάνεται αυτή η μείωση σε τρεις γραμμές το πολύ για κάθε τρόπο βελτιστοποίησης.

Ερώτηση 3: (10 Μονάδες) Σχεδιάστε έναν επιλεκτικό μηχανισμό πρόβλεψης (tournament predictor) και περιγράψτε πως λειτουργεί και γιατί μπορεί να αυξήσει την απόδοση ενός επεξεργαστή σε 10 γραμμές το πολύ.

Ερώτηση 4: (9 Μονάδες) Αναφέρετε και περιγράψτε i) 1 ομοιότητα, ii) 1 πλεονέκτημα και iii) 1 μειονέκτημα των πρωτοκόλλων κατασκοπίας συνοχής μνήμης (snoopy cache coherence protocols) σε σύγκριση με τα πρωτοκόλλα καταλόγου συνοχής μνήμης (directory-based cache coherence protocols).

Ερώτηση 5: (10 Μονάδες) Αναφέρετε :

A) Ποιο είναι το κοινό χαρακτηριστικό των VLIW και superscalar αρχιτεκτονικών;

B) Δώστε 2 λόγους για τους οποίους μια αρχιτεκτονική VLIW είναι πιο εύκολο να υλοποιηθεί σε υλικό από μια αρχιτεκτονική superscalar όταν και οι δύο επεξεργάζονται τον ίδιο αριθμό παράλληλων απλών εντολών

Γ) Δώστε 2 λόγους για τους οποίους μια αρχιτεκτονική VLIW έχει χαμηλότερη απόδοση από μια αρχιτεκτονική superscalar όταν και οι δύο επεξεργάζονται τον ίδιο αριθμό παράλληλων απλών εντολών

Ερώτηση 6: (20 Μονάδες) Για την παρακάτω ακολουθία εντολών του MIPS

- 1) LW \$t2, 0(\$t3)
- 2) ADD \$t2, \$t0, \$t2
- 3) SUB \$t8, \$t2, \$t0
- 4) LW \$t4, 0(\$t5)
- 5) ADD \$t4, \$t0, \$t4

A) Δείξτε τους κινδύνους δεδομένων (data hazards) και εφαρμόστε την τεχνική της αλλαγής ονομάτων των καταχωρητών (register renaming) ώστε να απαλειφθούν όσες γίνεται. Έχετε στην διάθεση σας όσους καταχωρητές θέλετε.

B) Υποθέστε ότι έχετε έναν επεξεργαστή με 5 επίπεδα pipeline που λειτουργεί ως εξής

- Επίπεδο 1 : Διαβάζει (Fetch) 2 εντολές από τη μνήμη
- Επίπεδο 2 : Αποκωδικοποιεί (Decode) και εφαρμόζει την τεχνική της αλλαγής ονομάτων των καταχωρητών (register renaming) και ξεκίνα (issue) έως και 2 εντολές στις μονάδες ALU
- Επίπεδο 3 : Εκτελεί έως και 2 εντολές στις 2 ALUs
- Επίπεδο 4 : Γράφει στη μνήμη τα αποτελέσματα έως και 2 εντολών (στον ίδιο κύκλο)
- Επίπεδο 5 : Γράφει στους καταχωρητές (register file) τα αποτελέσματα έως και δύο εντολών

Με ποια σειρά θα ξεκινήσουν (issue) οι παραπάνω εντολές; Με άλλα λόγια με ποια σειρά θα φτάσουν στο 3ο επίπεδο ?

Ερώτηση 7: (40 Μονάδες) : Έστω ότι έχετε έναν VLIW επεξεργαστή ο οποίος μπορεί να εκτελεί οποιεσδήποτε N εντολές του MIPS ταυτόχρονα αν δεν υπάρχουν κίνδυνοι/εξαρτήσεις (hazards/dependencies). Έστω το παρακάτω πρόγραμμα και έστω ότι N=3.

- (01) lw \$t0 0(\$a0)
- (02) lw \$t2 8(\$a0)
- (03) lw \$t1 4(\$a0)
- (04) add \$t6 \$t0, \$t1
- (05) lw \$t3 12(\$a0)

```

(06) sub $t7    $t1, $t2
(07) lw  $t4    16($a0)
(08) lw  $t5    20($a0)
(09) srlv $s2    $t6, $t7
(10) sub $s1    $t4, $t5
(11) add $s0    $t3, $t4
(12) sllv $s4    $t7, $s1
(13) srlv $s3    $t6, $s0
(14) sllv $s5    $s0, $s1
(15) add $s6    $s3, $s4
(16) add $s7    $s4, $s6
(17) srlv $t0    $s6, $s7
(18) srlv $t1    $t0, $s7

```

Υπενθυμίζεται ότι στον MIPS το αποτέλεσμα κάθε εντολής γράφεται στον 1^ο καταχωρητή της εντολής

Έστω ότι ο assembler του επεξεργαστή μπορεί να αλλάζει την σειρά των εντολών ώστε να εκτελεί όλες περισσότερες εντολές μπορεί ταυτόχρονα χωρίς φυσικά να αλλάζει η συμπεριφορά του προγράμματος (δηλ. το τελικό αποτέλεσμα του προγράμματος θα είναι το ίδιο με αυτό που θα προέκυπτε αν το πρόγραμμα εκτελούνταν σε έναν απλό MIPS σειριακά με μία εντολή σε κάθε κύκλο)

A) Δείξτε το βέλτιστο χρονο-προγραμματισμό (scheduling) ώστε το πρόγραμμα να εκτελεστεί με τις λιγότερες δυνατές VLIW εντολές

B) Αν ο επεξεργαστής VLIW επεκταθεί ώστε να εκτελεί όλες εντολές MIPS ταυτόχρονα θέλετε, πόσες εντολές κατ' ελάχιστον θα χρειαζόταν ώστε να εκτελείται αυτό το πρόγραμμα με τις λιγότερες VLIW εντολές (δηλαδή ποιο είναι το μικρότερο δυνατό N που επιτυγχάνει την βέλτιστη απόδοση ώστε να ελαχιστοποιηθεί ο αριθμός των VLIW εντολών που απαιτούνται για την εκτέλεση αυτού του προγράμματος);

Γ) Έστω ότι ο επεξεργαστής VLIW εκτελεί τις N εντολές που βρήκατε στο ερώτημα B), δείξτε το βέλτιστο χρονο-προγραμματισμό (scheduling) ώστε το πρόγραμμα να εκτελεστεί με τις λιγότερες νέες VLIW εντολές

Δ) Αν η κάθε VLIW εντολή του αρχικού επεξεργαστή χρειάζεται 4 κύκλους ρολογιού για να ολοκληρωθεί σε πόσους κύκλους εκτελείται το πρόγραμμα σύμφωνα με τον χρονο-προγραμματισμό του ερωτήματος A ; Αν η κάθε VLIW εντολή του «βέλτιστου» επεξεργαστή από το ερώτημα B, χρειάζεται 5 κύκλους για να ολοκληρωθεί σε πόσους κύκλους θα εκτελεστεί το πρόγραμμα σύμφωνα με τον χρονο-προγραμματισμό του ερωτήματος Γ ;