

15

1. Γράψτε ένα πρόγραμμα σε assembly του MIPS, χρησιμοποιώντας επανάληψη/βρόγχο, το οποίο υπολογίζει το **το ελάχιστο και το μέγιστο** ενός πίνακα 10 ακεραίων με όνομα **invec** που βρίσκονται στην μνήμη και αποθηκεύει το αποτέλεσμα σε έναν πίνακα στην μνήμη με όνομα **minval, maxval**. Θεωρήστε ότι οι αριθμοί μπορεί να είναι θετικοί και αρνητικοί. Αποφύγετε την χρήση ψευδοεντολών **bge, bgt, blt, ble**. Αρχικοποιήστε τον **invec** μέσω της δήλωσης:
`invec: .word -2,20,7,19,-3,0,0,-2,19,22`

Παραδοτέα: ο κώδικας του προγράμματος (`ask1.s`), στιγμιότυπο (εικόνα) της μνήμης με το περιεχόμενο μνήμης (μνήμη: `invec, minval, maxval`), πριν την εκτέλεση και μετά.

2. Έστω η οργάνωση του MIPS δίχως pipeline, όπου ο επεξεργαστής λειτουργεί στα 2 GHz και το κόστος των εντολών σε κύκλους ρολογιού είναι ως εξής: `add/addi: 4cc, beq/bne: 3cc, j/jal/jr: 2cc, slt/slti: 4cc, lw/lb: 5cc, sw/sb: 4cc`.

10

- (α) Για το πρόγραμμα που αναπτύξατε στο προηγούμενο ερώτημα, το (1), ποιά είναι το CPI αυτού του επεξεργαστή αν υποθέσουμε ότι κάθε εντολή `load/store` βρίσκει τις εντολές/δεδομένα στην κρυφή μνήμη εντολών/δεδομένων, και αν υποθέσουμε ότι δεν υπάρχουν αλληλοεξαρτήσεις (ιδανικό pipeline)?

7

- (β) ποιός είναι ο χρόνος ολοκλήρωσης της εφαρμογής?

8

- (γ) ποιά είναι το throughput σε εντολές ανά δευτερόλεπτο?

10

- (δ) Αν ο χρόνος προσπέλασης στην κύρια μνήμη είναι 25 κύκλοι ρολογιού και το ποσοστό επιτυχίας για δεδομένα και εντολές είναι 80%, τότε ποιός είναι ο χρόνος ολοκλήρωσης της εφαρμογής αυτής?

3. Έστω ότι ένα πρόγραμμα σε assembly MIPS το οποίο με βάση τη συμβολοσειρά `str: .asciiz "Have a Nice Day!"` έχει μια συνάρτηση `int find(str)`, η οποία επιστρέφει αριθμό κεφαλαίων χαρακτήρων. Χρησιμοποιήστε Στοιβά (Stack), ώστε πριν την κλήση της συνάρτησης να βάζετε στο stack την διεύθυνση του `str` και όταν επιστρέφει η συνάρτηση να βάζετε το αποτέλεσμα.

15

- α. Δώστε ένα κώδικα καθώς και στιγμιότυπο (εικόνα) της μνήμης με το περιεχόμενο μνήμης, πριν την εκτέλεση και μετά.

25

- β. Έστω ότι το `.text` ξεκινά από την διεύθυνση `0x0400` και το `.data` από την διεύθυνση `0x0A00`, υπολογίστε το ποσοστό επιτυχίας της κρυφής μνήμης (θεωρήστε ξεχωριστή κρυφή μνήμη εντολών και δεδομένων) (i) (η κάθε μία κρυφή μνήμη) άμεσης απεικόνισης 8 θέσεων όπου το cache block είναι 4 bytes, (ii) πλήρους προσεταιριστικής 8 θέσεων όπου το cache block είναι 4 bytes και πολιτική αντικατάστασης `least-recently-used`. Υποθέστε `write-through (no-allocate)` πολιτική εγγραφής. Δείξτε σε κάθε πρόσβαση στην μνήμη τι επίδραση έχει στην κρυφή μνήμη (σε V, TAG και DATA).

10

- γ. Ποιος είναι ο μέσος χρόνος πρόσβασης στην μνήμη σε καθεμία από τις περιπτώσεις (i), (ii) αν ο χρόνος πρόσβασης στην cache είναι 1 κύκλος ρολογιού και ο χρόνος πρόσβασης στην κύρια μνήμη είναι 25 κύκλοι ρολογιού?