

Επίπεδο Μικροαρχιτεκτονικής (The Microarchitecture Level)

Απαντήσεις

1. Μόνο ένας καταχωρητής κάθε φορά μπορεί να οδηγηθεί στον δίαυλο B, έτσι ένα απλό πεδίο 4-bit επαρκεί για να καθορίσει ποίος. Από την άλλη μπορεί να γράφεται η έξοδος του διαύλου C σε περισσότερους από έναν καταχωρητές και γι' αυτό το λόγο ο δίαυλος C αναπαρίσταται με χάτη bit.
2. Ο κώδικας είναι ο ακόλουθος:

```
ILOAD j
ILOAD k
IADD
ISTORE i
ILOAD i
BIPUSH 3
IF_CMPEQ L1
ILOAD j
BIPUSH 1
ISUB
ISTORE j
GOTO L2
L1: BIPUSH 0
    ISTORE k
L2: BIPUSH 5
    ISTORE k
```

3. Είναι $i = (j - n - 7) + (j - n - 7)$.
4. Οι εντολές και ο αριθμός των μικροεντολών για κάθε μία είναι ILOAD (6), ILOAD (6), IADD (4), ISTORE (7). Αυτό μας κάνει σύνολο 23 μικροεντολών. Στα 2.5 GHz, κάθε μικροεντολή απαιτεί 0.4 nsec, έτσι και οι 23 θα χρειαστούν 9.2 nsec.
5. Οι εντολές και ο αριθμός των μικροεντολών για κάθε μία είναι ILOAD (3), ILOAD (3), IADD (3), ISTORE (5). Αυτό μας κάνει σύνολο 14 μικροεντολών. Στα 2.5 GHz, κάθε μικροεντολή απαιτεί 0.4 nsec, έτσι και οι 14 θα χρειαστούν 5.6 nsec.

Με βάση αυτό το αποτέλεσμα και αυτού της προηγούμενης άσκησης, βλέπουμε ότι η Mic-2 υλοποίηση σε σχέση με την Mic-1 είναι 5.6/9.2. Συνεπώς, ένα πρόγραμμα το οποίο χρειάζεται 100 sec στην Mic-1, θα χρειαζόταν 60.87 sec στην Mic-2.

6.

```
pop2a SP = SP - 1
pop2b MAR = SP = SP - 1; rd
pop2c
pop2d TOS = MDR; goto Main1
```
7. Επί του παρόντος, η τοπική μεταβλητή 0 είναι ένας δείκτης, ο οποίος σπάνια χρησιμοποιείται. Έτσι, έχοντας περιορισμένη πρόσβαση σε αυτό δεν είναι ιδιαίτερα χρήσιμο. Για το πλήρες μηχάνημα JVM, θα ήταν προτιμότερο να έχουμε το LV σημείο στην πρώτη τοπική μεταβλητή (η πρώτη παράμετρος), με σχετική διεύθυνση offset -1 από αυτό.
8. Χρειάζεται να εντοπίσει την λέξη OBJREF (ένας δείκτης προς το αντικείμενο που θα κληθεί), έτσι ώστε να ξέρει πόσες παραμέτρους θα προσπεράσει για να βρει την στοίβα όπου αποθηκεύονται οι τοπικές μεταβλητές της καλούσας μεθόδου.
9. Μία πιθανή υλοποίηση είναι η παρακάτω:

```

dload1    MAR = LV + MBR1U; rd
dload2    H = MAR + 1
dload3    MAR = SP = SP + 1; wr
dload4    MAR = H; rd
dload5    MAR = SP = SP + 1; wr
dload6    TOS = MDR; goto (MBR)

```

10. Ο μέσος χρόνος προσπέλασης είναι $0.6 \times 5 + 0.35 \times 20 + 0.05 \times 80$. Αυτό δίνει 14 nsec.
11. Όχι ιδιαίτερα. Αν η μνήμη cache είναι τοποθέτησης κατά την εγγραφή (write allocation), τότε η γραμμή θα τεθεί στην cache κατά τη στιγμή της εγγραφής, έτσι ώστε οι μετέπειτα αναγνώσεις να είναι ελεύθερες. Αν η μνήμη cache δεν είναι τοποθέτησης κατά την εγγραφή, η εγγραφή θα γίνει απευθείας στην μνήμη, αλλά η πρώτη ανάγνωση θα φέρει την γραμμή μας στην cache, και το υπόλοιπο των αναγνώσεων θα είναι και πάλι ελεύθερο. Το μόνο κέρδος εδώ είναι ότι εδώ η εξοικονόμηση μιας εγγραφής στη μνήμη. Το μειονέκτημα της τοποθέτησης κατά την εγγραφή (write allocation) είναι ότι αν δεν υπάρχουν ακολουθούμενες αναγνώσεις, η γραμμή θα έχει έρθει χωρίς κάποιο λόγο. Έτσι, η τοποθέτησης κατά την εγγραφή, ποντάρει στο ότι η γραμμή η γραμμή θα ξανά γραφτεί ξανά σύντομα.
12. Η προκαταβολική προσκόμιση θα είναι μόνο επιτυχημένη εάν και οι τέσσερις εντολές διακλάδωσης είναι σωστά προβλέψιμες. Η πιθανότητα να συμβεί αυτό είναι 0.94, η οποία είναι περίπου 0.66.
13. Μια εξάρτηση WAW παρουσιάζεται όταν ένας καταχωρητής ο οποίος γράφεται έχει ξαναγραφτεί πριν από κάποια άλλη εντολή. Εάν δεν υπάρχουν ενδιάμεσες αναγνώσεις του καταχωρητή και δεν υπάρχουν ανεπιθύμητες ενέργειες, η πρώτη εντολή αποτυγχάνει, έτσι ώστε να μπορεί να αφαιρεθεί, να ματαιωθεί νωρίς, ή να έχει το αποτέλεσμα της επικαλυμμένο (overwritten).