

ΣΕΙΡΑ ΑΣΚΗΣΕΩΝ 4

ΑΣΚΗΣΗ 4.2

Ερώτημα 4(a):

Εχουμε ως δεδομένο ότι ξοδεύουμε και τις 8 εντολές του πεδίου `op` για εντολές `I format`. Αυτό σημαίνει ότι αν θέλαμε εμείς να εκτελέσουμε μια εντολή `R format` δε θα μπορούσαμε, καθώς θα ήταν όλες "κατειλημμένες" από τις `I-formated`. Με μόνο 3 bits στον `opcode` έχουμε μόνο $2^3=8$ πιθανούς συνδυασμούς, κανέναν όμως ελεύθερο για τις εντολές του `R format`.

Ακόμη όμως κι αν επιμενάμε να εκτελέσουμε μια εντολή τύπου `R` θα καταλήγαμε σε μια επαναληψη εντολής και θα παρατηρούσαμε ότι η ίδια εντολή είχε εκτελεστεί προηγουμένως από τις εντολές με `I format`.

Π.Χ.: εντολή: `RR, op=001 & funct=10`

Αυτό σημαίνει για τον υπολογιστή, "αν δεις 001 βρίσκεσαι σε `R format` γι' αυτό δες και το `funct`" αυτό σημαίνει ουσιαστικά όμως ότι μιλάμε για την εντολή με `op=110` στο `I format` (εφόσον είναι όλες "κατειλημμένες" από τις `I-formated` εντολές).

Είναι ευνοητό λοιπόν, πως αφού έχουμε 8 δυνατές εντολές να εκτελέσουμε και τις έχουμε εκτελέσει όλες με `I format`, δε μένει καμία για `R format` και αν προσπαθήσουμε βεβιασμένα να εκτελέσουμε κάποια έτσι, θα καταλήξουμε σε επαναληψη εντολής, αρα και σε λάθος.

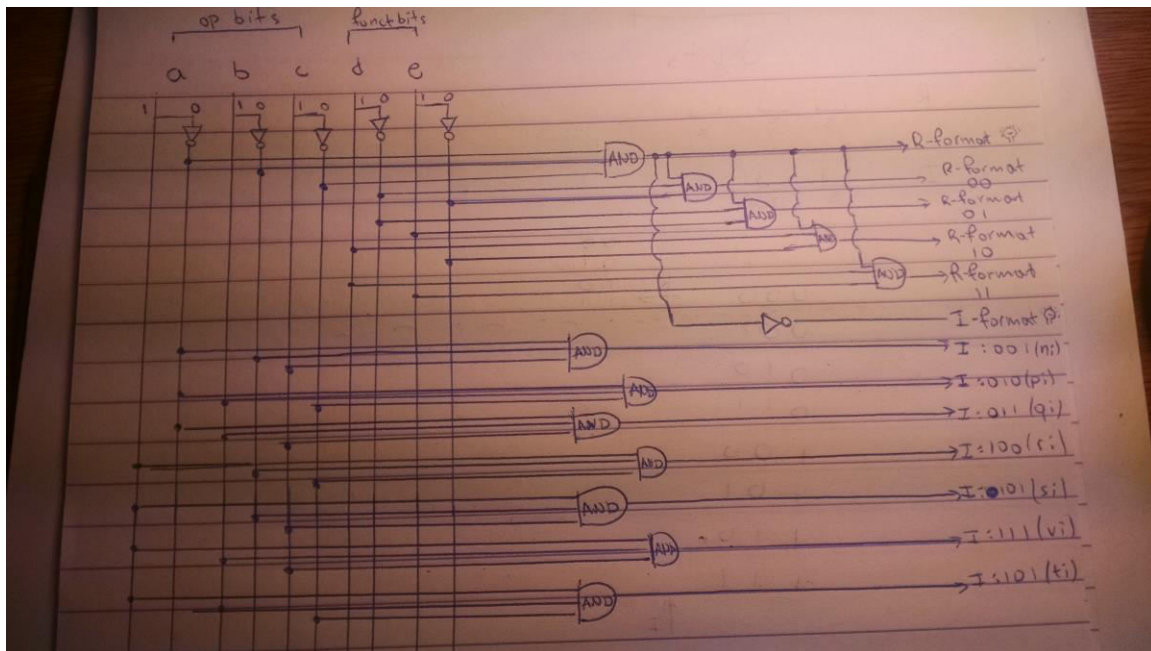
Ερώτημα 4(b):

Εχουμε ως δεδομένο ότι ξοδεύουμε τις 7 από τις 8 εντολές του πεδίου `op` για εντολές `I format`. Αυτό σημαίνει ότι μπορούμε να εκτελέσουμε εντολές με `R format`, μόνο όμως όταν έχουμε `op = 000`. Για να δούμε πόσες τέτοιες εντολές έχουμε τη δυνατότητα να εκτελέσουμε θα κοιτάξουμε τον `funct` και πόσα bits έχει αυτός. Έχει 2, αρα οι δυνατοί συνδυασμοί είναι $2^2=4$. Αρα έχουμε στη διάθεση μας 4 εντολές με `R format`.

Θα μπορούσαμε να ειχαμε παραπάνω `R-formated` εντολές αν ειχαμε παραπάνω συνδυασμούς `opcode`, για την ακρίβεια για κάθε έναν παραπάνω `opcode` θα ειχαμε στη διάθεση μας 4 παραπάνω εντολές ενώ παραλληλα θα χαναμε 1 τύπου `I`.

Αρα δε μπορούμε να έχουμε παραπάνω καθοτι 7 συνδυασμοί είναι πιασμένοι στις εντολές με `I format`. Συνολικά $7+4=11$

Ερώτημα 4(c):



Ερωτημα 4 (d) :

Αντιστοιχα με το b αφου εχουν δεσμευθει μονο 6 εντολες ως I format μενουν αλλοι 2 συνδυασμοι op, γεγονος το οποιο σημαινει οτι για R format εντολες τωρα εχουμε $2 \cdot 4 = 8$ διαθεσιμες. Αρα συνολικα $6 + 8 = 14$

Μια καλη επιλογη opcode που θα αντιστοιχουν στις R format εντολες θα ηταν α: op=000 και β: op=111 ωστε το συνδυαστικο κυκλωμα να μη περιπλεκεται πολυ παραπανω.

opcode	funct
#---R-FORMAT---	
000	00
000	01
000	10
000	11
111	00
111	01
111	10
111	11
#---I-FORMAT---	
001	
010	
011	
100	
101	
110	

ΣΕΙΡΑ ΑΣΚΗΣΕΩΝ 5

Ασκηση 5.2

Στον MIPS έγινε μεγάλη προσπάθεια να είναι όσο γίνεται μικρότερος ο αριθμός των εντολών ώστε αντιστοιχά να χρειαζονται και λιγότερα bits στον op code. Ωστε αντιστοιχά να είναι περισσότερα τα bits σε καταχωρητές και immediates. Για τον λόγο αυτό προτιμήθηκε να μην υπάρχουν οι 2 εντολές beq & bne για immediates και απλώς να γίνεται η τιμή τους load σε καταχωρητές και η σύγκριση να γίνεται ανάμεσα στους καταχωρητές. Αυτό ίσως είναι λίγο πιο χρονοβόρο αλλά τελικά γλιτώνονται κάποια bits που μας είναι απαραίτητα άλλου.

Ασκηση 5.3

- i. beq \$12, \$13, L1
- ii. bne \$12, \$13, L1
- iii. slt \$at, \$12, \$13
 bne \$at, \$0, L1
- iv. slt \$at, \$13, \$12
 beq \$at, \$0, L1
- v. slt \$at, \$13, \$12
 bne \$at, \$0, L1
- vi. slt \$at, \$12, \$13
 bne \$at, \$0, L1
- vii. addi \$at, CONST
 beq \$12, \$at, L1
- viii. addi \$at, CONST
 bne \$12, \$at, L1
- ix. slti \$at, \$12, CONST
 bne \$at, , \$0. L1
- x. addi \$at, CONST
 stl \$at, \$at, \$12
 beq \$at, \$0, L1
- xi. addi \$at, CONST
 slt \$at, \$at, \$12
 bne \$at, \$0, L1
- xii. slti \$at, \$12, CONST
 bne \$at, \$0, L1

Ασκηση 5.4

```

        add $s3, $0, $0          # i=0;

Loop:    sll $t1, $s3, 2          # t1 = 4 * i
        add $t1, $t1, $s6        # t1 = διεύθυνση του table[i]
        lw $t0, 0($t1)          # t0 = table[i]
        addi $s3, $s3, 1        # i = i+1
        bne $t0, $s5, Loop      # if table[i]!=value goto Loop

        j Exit                  # επανάληψη: goto Exit

Exit:    ....                    # υπόλοιπο πρόγραμμα
-----

```

Στον προηγούμενο κωδικά απαιτούνται,
 επαναληψεις: 10
 εντολες ανα επαναληψη: 6
 Αρα $6 \cdot 10 = 60$
 Συνολικά όμως 58 καθώς μετά την beq στο τελευταίο loop δεν
 εκτελούνται οι εντολες addi & jump.

Στον νέο κωδικά,
 επαναληψεις: 10
 εντολες ανα επαναληψη: 5
 Αρα $5 \cdot 10 = 50$