

4(a): Εφόσον, έχουμε χρησιμοποιήσει και τους 8 συνδυασμούς του op δεν μπορούμε έχουμε κάποια εντολή R-format, γιατί δεν μας περισσεύουν συνδυασμοί των τριών bits, ώστε να την περιγράψουμε. Η εντολή RR \$5 είναι θα ήταν ίδια και απαράλλακτη με την li, επειδή θα έχουν το ίδιο op code.

4(b): Μας μένει ένας συνδυασμός άρα μπορούμε να έχουμε μία εντολή R-Format με τα 3 bits του op code που μένουν δηλαδή 111. Θα έχουμε ως funct την 00. Δεν μπορούμε να έχουμε περισσότερες, γιατί θα έχουν καλυφτεί όλοι οι συνδυασμοί των τριών bits του opcode.

4(d): Τώρα μπορούμε να έχουμε 2 εντολές R-Format. Οι 6 εντολές I-Format θα έχουν opcodes 000,001,010,011,100,101 και οι 2 εντολές R-Format 110 και 111.

5.2: Ο λόγος που δεν υπάρχουν εντολές για σύγκριση τιμής καταχωρητή με σταθερά είναι, ότι με τους καταχωρητές μπορούμε να κάνουμε και τις δουλειές με τις σταθερές. Δηλαδή εάν θέλουμε να συγκρίνουμε τιμές που χρειάζεται να τις έχουμε σε καταχωρητή γιατί τις επεξεργαζόμαστε συνεχώς, αυτό μπορεί να γίνει κάλλιστα με τις beq bne, ενώ αν επίσης χρειαζόμαστε να συγκρίνουμε με σταθερά πάλι μπορούμε να βάλουμε την σταθερά σε ένα καταχωρητή και να κάνουμε την σύγκριση.

5.3:

i)

```
beq $16,$17,L1
L1:...
```

ii)

```
bne $16,$17,L1
L1:...
```

iii)

```
slt $at,$16,$17
bne $at,$0,L1
L1:...
```

iv)

```
beq $16,$17,L1
slt $at,$16,$17
bne $at,$0,L1
L1:
```

v)

```
slt $at,$16,$17
beq $at,$0,L1
L1:
```

vi)

```
beq $16,$17,L1
slt $at,$16,$17
beq $at,$0,L1
L1:
```

vii)

```
addi $at,$0,CONST
beq $16,$at,L1
L1:
```

```
viii)
addi $at,$0,CONST
bne $16,$at,L1
L1:
```

```
ix)
slti $at,$16,CONST
bne $at,$0,L1
L1:
```

```
x)
addi $at,$0,CONST
beq $16,$at,L1
slti $at,$16,CONST
bne $at,$0,L1
L1:
```

```
xi)
slti $at,$16,CONST
beq $at,$0,L1
L1:
```

```
xii)
addi $at,$0,CONST
beq $16,$at,L1
slti $at,$16,CONST
beq $at,$0,L1
L1:
```

5.4:

```
addi $s3,$0,0
Loop:
add $t1,$s6,$s3
lw $t0,0($t1)
addi $s3,$s3,1
bne $t0,$s5,Loop
```

4c)

