

```

(1)
    addi $s4, $0, 15
    addi $s5, $0, 16
(2)
main: addi $s1, $0, 5
      addi $t2, $0, 3
      addi $s3, $0, 4
      add  $t4, $s1, $t2
      add  $s5, $s1, $s3
      sub  $t2, $t4, $s3
      sub  $s3, $s5, $s2
      slt  $s4, $s1, $t2
      and  $s1, $s3, $s5
      or   $s5, $t2, $s1
exit:

```

Παρατηρώ ότι τα σήματα ελέγχου είναι ίδια σε κάθε εντολή τύπου R, διότι έχουν ίδιο opcode. Η διαφορά τους είναι στο σήμα ελέγχου ALUOp που περιέχει και την πράξη που θα κάνει η ALU. Στο R-format είναι το function κομμάτι.

```

(3)
main: addi $s1, $0, 5
      lw   $t2, 0($s0)
      sw   $s1, 0($s0)
exit:

```

Οι εντολές αναφοράς στη μνήμη χρησιμοποιούν την ALU για έναν υπολογισμό διεύθυνσης. Στη συγκεκριμένη περίπτωση αυτό γίνεται με την πρόσθεση της τιμής του καταχωρητή και του offset. Για την lw έχουμε: Η ALUOp στον κεντρικό πρώτο ελεγχό είναι 00 και στον δεύτερο 010. Για την sw έχουμε: Η ALUOp στον κεντρικό πρώτο ελεγχό είναι 00 και στον δεύτερο 010.

```

(4) main: addi $s1, $0, 5
      andi $t2, $s1, 8
      ori  $s4, $s0, 0
exit:

```

	addi	andi	ori
RegDst	0	0	0
Jump	0	0	0
Branch	0	0	0
MemtoReg	0	0	0
MemRead	0	0	0
ALUSrc	1	1	1
ALUOp	010	000	001
RegWrite	1	1	1

Παρατηρούμε ότι η διαφορά τους είναι στο σήμα ελέγχου ALUOp όπως είναι και αναμενόμενο διότι αυτό καθορίζει και την πράξη της ALU. Η διαφορά στις εισόδους της ALU είναι ότι στις εντολές R-format οι είσοδοι είναι δυο καταχωρητές ενώ στις I-format είναι ένας καταχωρητής και μία σταθερή ποσότητα. Επίσης στις εντολές με R-format η πράξη φαίνεται στα 5 τελευταία bits (στο function μέρος) ενώ στις I-format φαίνεται στο opcode. Οι τελεστές για κάθε πράξη βρίσκονται στο ALUOp και ο πολυπλέκτης που το καθορίζει είναι ο ALUSrc. Στις εντολές R-format το RegDst είναι 1 διότι γράφουμε σε καταχωρητή, το ALUSrc είναι 0 διότι δεν χρειάζεται.

(5)

```
main: addi $s1, $0, 5
      addi $s2, $0, 8
      beq  $s1, $s2, exit
      j    main
exit:
```

Όπως γνωρίζουμε και από προηγούμενη θεωρία ο PC μετά από επιτυχημένο branch παίρνει την τιμή  $(PC+4)+4*Imm$ , όπως και για κάθε εντολή jump  
Αν δεν πετύχει το branch παίρνει απλά την τιμή της επόμενης εντολής  $PC+4$ . Το σήμα branch και ο MUX2 καθορίζουν ποιο σήμα θα φορτωθεί.