

## Άσκηση 4: Γλώσσα Μηχανής, Format Εντολών

κάντε την έως Τρίτη 28 Φεβρουαρίου 2017 (βδ. 4.1) (από βδ. 2.3)

**Βιβλίο:** Διαβάστε την §2.5: σελίδες 134-142.

### 4.1 Περίληψη Θεμάτων Γλώσσας Μηχανής του MIPS:

Εσωτερικά, ο υπολογιστής λειτουργεί μόνο με δυαδικές τιμές και σύμβολα. Έτσι, για να μπορέσει να εκτελεστεί ένα πρόγραμμα Assembly πρέπει αυτό να μεταφραστεί σε *Γλώσσα Μηχανής*, δηλ. σε δυαδικά σύμβολα. Τη μετάφραση αυτή κάνει ένα πρόγραμμα, ο *Assembler*. Όλες οι εντολές του MIPS, όπως και των άλλων υπολογιστών συγλ RISC, έχουν σταθερό μέγεθος 32 bits, και είναι ευθυγραμμισμένες σε λέξεις των 32 bits, δηλαδή οι διευθύνσεις τους είναι ακέραια πολλαπλάσια του 4 (άρα τα 2 LS bits τους είναι πάντα 0). Μέσα τους οι εντολές περιέχουν από 2 έως 6 πεδία (fields). Υπάρχουν 3 διαφορετικές μορφές εντολών· η μορφή (format) της κάθε εντολής καθορίζεται από το πρώτο πεδίο της εντολής, που ονομάζεται **op** (από "operation code" - κωδικός πράξης), έχει μέγεθος 6 bits, και είναι πάντα στην ίδια θέση (MS bits) για όλες τις εντολές. Φυσικά, με τα 6 bits που έχει, το πεδίο αυτό καθορίζει, εκτός από το format της εντολής, και ένα μέρος ή και ολόκληρο το **opcode** της εντολής, δηλαδή το τι πράξη ή ενέργεια πρέπει να εκτελέσει αυτή η εντολή. Οι τρεις μορφές εντολών είναι:

#### R-format:

Αυτό φαίνεται στη σελίδα 137 του βιβλίου (Α' τόμος Ελληνικής μετάφρασης από 4η Αμ. έκδοση), και αποτελείται, μετά το πεδίο **op**, από τρία πεδία των 5 bits καθένα που επιλέγουν έναν από τους 32 καταχωρητές καθένα, ένα πεδίο των 5 bits που δεν θα το χρησιμοποιήσουμε εμείς στο υποσύνολο των εντολών που θα μελετήσουμε, και ένα πεδίο μεγέθους 6 bits που αποτελεί επέκταση του **op** και λέγεται **funct** (function code).

#### I-format:

Αυτό φαίνεται στη σελίδα 138 του βιβλίου (Α' τόμος, 4η εκδ.), και αποτελείται, μετά το πεδίο **op**, από δύο πεδία των 5 bits καθένα που επιλέγουν έναν από τους 32 καταχωρητές καθένα, και ένα πεδίο μεγέθους 16 bits που λέγεται **imm** (immediate, ή offset, ή constant, ή address) και που περιέχει μία "άμεση" σταθερή ποσότητα (συχνά προσημασμένη - signed).

#### J-format:

Αυτό φαίνεται στη σελίδα 171 του βιβλίου (Α' τόμος, 4η εκδ.), και περιέχει, μετά το πεδίο **op**, ένα πεδίο "**target**" μεγέθους 26 bits που προσδιορίζει μεγάλο μέρος μιας πλήρους διεύθυνσης μνήμης.

Οι εντολές του MIPS, το format τους, και τα opcodes τους φαίνονται και στις σελίδες A-49 έως A-70 του παραρτήματος Α του βιβλίου, από παλαιότερη Αμερικανική έκδοση, που υπάρχει στο αρχείο: [~hy225/spim/documentation/HP\\_AppA.pdf](http://www.csd.uoc.gr/~hy225/spim/documentation/HP_AppA.pdf) Καταχωρητές "πηγής" (source registers) λέμε αυτούς που η εντολή τους διαβάζει κατά τη διάρκεια της εκτέλεσής της· ανάλογα με την εντολή, οι καταχωρητές πηγής είναι δύο, ένας, ή κανένας. Καταχωρητή "προορισμού" (destination register), εάν υπάρχει, λέμε αυτόν στον οποίο η εντολή γράφει το αποτέλεσμα της –εάν γράφει σε καταχωρητή το αποτέλεσμά της. Στον MIPS, οι καταχωρητές πηγής, όταν υπάρχουν, προσδιορίζονται **πάντα** από τα ίδια πεδία της εντολής, σε **σταθερές θέσεις**, προκειμένου να επιταχυνθεί η ανάγνωσή τους αμέσως μόλις γίνει γνωστή η εντολή, χωρίς να περιμένουμε την αποκωδικοποίηση του opcode για να μάθουμε το format της εντολής: οι καταχωρητές πηγής, όταν υπάρχουν, είναι πάντα το δεύτερο και το τρίτο πεδίο της εντολής, των 5 bits καθένα, αμέσως μετά τα 6 bits του opcode. Ο καταχωρητής προορισμού, όταν υπάρχει, μπορεί να είναι σε μεταβλητή θέση μέσα στο format της εντολής, δεδομένου ότι υπάρχει χρόνος να έχει αποκωδικοποιηθεί το opcode μέχρι να χρειαστεί να ξέρουμε ποιός είναι ο καταχωρητής προορισμού: είναι άλλοτε το τέταρτο και άλλοτε το τρίτο πεδίο της εντολής. Ο τρόπος που το format και το opcode καθορίζονται από τα πεδία op και funct με μια κωδικοποίηση μεταβλητού μεγέθους αποτελεί αντικείμενο της άσκησης 4.2.

## Άσκηση 4.2: Γλώσσα Μηχανής και Κώδικες Μεταβλητού Μήκους

Σ' αυτή την άσκηση θα μελετήσετε τον τρόπο που το format και το opcode στον MIPS καθορίζονται από τα πεδία `op` και `funct` με μια κωδικοποίηση μεταβλητού μεγέθους. Για να μην παιδευόμαστε όμως με μεγάλο πλήθος εντολών, θα χρησιμοποιήσουμε ένα δικό μας, φανταστικό format εντολών, ενός φανταστικού υπολογιστή, του MIPS\_8. Όλες οι εντολές του MIPS\_8 έχουν μέγεθος 8 bits. Ο MIPS\_8 έχει μόνο 8 καταχωρητές, τους \$0, \$1, ..., \$7, και οι σταθερές του ποσότητες (immediates) μπορούν να είναι μόνο οι 32 μη-αρνητικοί ακέραιοι 0, 1, 2, ..., 31. Οι εντολές του MIPS\_8 έχουν μόνον έναν τελεστέο –είτε καταχωρητή είτε σταθερή ποσότητα immediate– και έχουν μόνο δύο format, τα εξής:

### R-format:

- **op** (3 MS bits): πρώτο μέρος του opcode,
- **R** (3 επόμενα bits): ο καταχωρητής - τελεστέος,
- **funct** (2 LS bits): δεύτερο μέρος του opcode.

### I-format:

- **op** (3 MS bits): ο (μοναδικός) opcode,
- **Imm** (5 LS bits): η σταθερά - τελεστέος.

**4(a):** Εστω ότι "ξοδεύουμε" και τους οκτώ (8) διαθέσιμους συνδυασμούς του πεδίου `op` για 8 εντολές I-format, τις εντολές **mi**, **ni**, **pi**, **qi**, **ri**, **si**, **ti**, **vi**. Σ' αυτή την περίπτωση, μπορούμε να έχουμε καμία εντολή R-format; Γιατί όχι; Έστω πως επιμέναμε να έχουμε την εντολή R-format "**RR**" με `op=001` και `funct=10`. Τότε, η εντολή "**RR** \$6" με ποιάν άλλη εντολή I-format (και με τι τελεστέο) θα ήταν ίδια κι απαράλλακτη, με συνέπεια να μη μπορούμε να τις έχουμε και τις δύο στον MIPS\_8; Αποδείξτε εν συντομία αλλά με "μαθηματική" ακρίβεια και σαφήνεια ότι το ίδιο θα ίσχυε για οιαδήποτε άλλη δυνατή εντολή R-format.

**4(b):** Εστω τώρα ότι "ξοδεύουμε" μόνο τους 7 από τους 8 διαθέσιμους συνδυασμούς του πεδίου `op` –τους 001, 010, 011, 100, 101, 110, και 111– για 7 εντολές I-format –τις **ni**, **pi**, **qi**, **ri**, **si**, **ti**, **vi**. Σ' αυτή την περίπτωση, πόσες εντολές R-format μπορούμε να έχουμε; Τι κώδικες **op** και **funct** θα έχει καθεμιά τους; Γιατί δεν μπορούμε να έχουμε περισσότερες από τόσες εντολές R-format;

**4(c):** Για την περίπτωση 4(b), σχεδιάστε ένα συνδυαστικό κύκλωμα, χρησιμοποιώντας πύλες NOT, AND, OR (οσωνδήποτε εισόδων) (μόνο τέτοιες, και όχι έτοιμους αποκωδικοποιητές), που να δέχεται σαν είσοδο μιαν εντολή (8 bits) και να παράγει σαν εξόδους:

- ένα σήμα I που ανάβει όταν και μόνον όταν η εντολή είναι I-format,
- ένα σήμα R που ανάβει όταν και μόνον όταν η εντολή είναι R-format,
- 7 σήματα, ένα για κάθε εντολή I-format, που να ανάβει όταν και μόνον όταν βλέπει τη συγκεκριμένη εντολή, και
- ένα σήμα για κάθε εντολή R-format, που να ανάβει όταν και μόνον όταν βλέπει τη συγκεκριμένη εντολή.

**4(d):** Ανάλογη ερώτηση με την 4(b), αλλά έστω ότι τώρα έχουμε μόνο 6 εντολές I-format. Πόσες εντολές R-format μπορούμε να έχουμε; Γράψτε τα opcodes όλων των εντολών, και των δύο format. Προσπαθήστε να επιλέξετε τα opcodes έτσι ώστε να απλοποιείται η αποκωδικοποίηση των σημάτων I και R (σύμφωνα με την εμπειρία σας από την ερώτηση 4(c), χωρίς πάντως να ζητείται εδώ σχεδίαση κυκλώματος).

### Τρόπος Παράδοσης:

Κάντε την απλή αυτή άσκηση από τώρα, για να την έχετε έτοιμη, παρ' ότι θα την παραδώσετε λίγο αργότερα, μαζί με την επόμενη σειρά ασκήσεων 5. Ο τρόπος παράδοσης, για τις ασκήσεις 4 και 5 μαζί, θα είναι on-line, σε μορφή PDF (μπορεί να είναι κείμενο μηχανογραφημένο ή/και "σκαναρισμένο" χειρόγραφο).