

Άσκηση 4: Γλώσσα Μηχανής, Format Εντολών

Προθεσμία έως Τετάρτη 10 Μαρτίου 2010, 23:59 (βδομάδα 5.2)(από βδομάδα 3.2)

[[Up](#) - Table of Contents]
[[Prev](#) - 3. Memory Accesses]

[printer version - [PDF](#)]
[5. Compare and Branch - [Next](#)]

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 στυλ RISC, έχουν σταθερό μέγεθος 32 bits. Μέσα τους περιέχουν από 2 έως 6 πεδία (fields). Υπάρχουν 3 διαφορετικές μορφές που οι εντολές μπορεί να έχουν. Η μορφή (format) της κάθε εντολής καθορίζεται από το πρώτο πεδίο της εντολής, που ονομάζεται **op**, 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 81 (επάνω) του βιβλίου (Α' τόμος Ελληνικής έκδοσης), 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 81 (κάτω) του βιβλίου (Α' τ. Ελλ. 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.

J-format:

Αυτό φαίνεται στη σελίδα 115 (επάνω) του βιβλίου (Α' τ. Ελλ. 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 στις σελίδες 117-145 (παράρτημα Α) του βιβλίου (Β' τόμος Ελληνικής έκδοσης). Ο τρόπος που το format και το opcode καθορίζονται από τα πεδία **op** και **funct** με μια κωδικοποίηση μεταβλητού μεγέθους αποτελεί αντικείμενο της άσκησης 5.2.

Ειδική αναφορά χρήζει η κωδικοποίηση των εντολών μεταφοράς ελέγχου με συνθήκη στον MIPS. Οι εντολές μεταφοράς ελέγχου (CTI, control transfer instructions) καθορίζουν να εκτελεστεί σαν επόμενη εντολή – πάντοτε ή υπό ορισμένες συνθήκες μόνο– μια έντολη άλλη από την "επόμενη από κάτω" τους εντολή. Όταν η μεταφορά ελέγχου γίνεται υπό συνθήκη, οι εντολές συνήθως ονομάζονται **διακλαδώσεις** (branch). Όταν η μεταφορά γίνεται πάντοτε, οι εντολές συνήθως λέγονται **άλματα** (jump). Επίσης υπάρχουν κλήσεις διαδικασιών και λειτουργικού συστήματος, και επιστροφές από αυτές. Στον MIPS, η συνθήκη διακλάδωσης μπορεί να έχει περιορισμένες μόνο μορφές, για λόγους ταχύτητας. Οι μόνες συνθήκες διακλαδώσεων που υπάρχουν αφορούν συγκρίσεις ενός καταχωρητή με το μηδέν (δεν θα ασχοληθούμε με αυτές στο "δικό μας" υποσύνολο του MIPS), και συγκρίσεις δύο καταχωρητών *μόνο για ισότητα ή ανισότητα* και όχι για άλλων μορφών σχέσεις (μικρότερος, μεγαλύτερος, κλπ). Οι εντολές αυτές είναι:

```
• beq $rs, $rt, label # διακλάδωση εάν: $rs == $rt
• bne $rs, $rt, label # διακλάδωση εάν: $rs != $rt
```

Στη γλώσσα Assembly, η διεύθυνση προορισμού της διακλάδωσης δηλώνεται απλά με μια ετικέτα (label), και αναλαμβάνει ο Assembler να υπολογίσει και να βάλει τη σωστή δυαδική τιμή. Στη γλώσσα μηχανής, οι εντολές διακλάδωσης ακολουθούν το I-format, και η διεύθυνση προορισμού προκύπτει ως εξής:

$$PC_new := (PC_br + 4) + 4 * ImmOffset$$

όπου **PC_br** είναι η διεύθυνση της ίδιας της εντολής διακλάδωσης, **ImmOffset** είναι η σταθερή ποσότητα των 16 bits του I-format θεωρούμενη ως προσημασμένος αριθμός σε συμπλήρωμα ως προς 2 (δηλαδή sign-extended), και **PC_new** είναι η διεύθυνση της εντολής προορισμού σε περίπτωση επιτυχίας της

διακλάδωσης. Η αύξηση (**PC_br+4**) γίνεται για λόγους ευκολίας του hardware (όλες οι εντολές αυξάνουν τον PC κατά 4). Ο πολλαπλασιασμός του ImmOffset επί 4 γίνεται για να εκμεταλλευτούμε το γεγονός ότι η διεύθυνση όλων των εντολών του MIPS είναι ακέραιο πολλαπλάσιο του 4, κι έτσι να τετραπλασιάσουμε το "βεληνικές" των διακλαδώσεων –με άλλα λόγια, ο αριθμός ImmOffset μετράει πλήθος εντολών μπροστά (θετικών) ή πίσω (αρνητικός), αντί να μετρά πλήθος bytes μπροστά ή πίσω. Στην πραγματικότητα, ο MIPS έχει "καθυστερημένες διακλαδώσεις" (delayed branches), για λόγους καλύτερης εκμετάλλευσης της ομοχειρίας (pipelining) του hardware, αλλά εμείς θα το αγνοήσουμε σε αυτό το μάθημα –το θέμα αυτό συζητείται εν εκτάσει στο HY-425. Στην υλοποίηση με καθυστερημένες διακλαδώσεις ο επεξεργαστής εκτελεί πάντα και χωρίς συνθήκη την εντολή που ακολουθεί το branch, για αυτό ο μετρητής προγράμματος αυξάνεται πάντα κατά 4 bytes μετά το branch πριν υπολογιστεί η απόσταση από την εντολή προορισμού. Εδώ πρέπει να σημειώσουμε ότι εξ'ορισμού ο προσομοιωτής SPIM δεν προσομοιώνει τις καθυστερημένες διακλαδώσεις και κατά συνέπεια η απόσταση από την εντολή προορισμού υπολογίζεται σαν **PC_new := PC_br+4*ImmOffset**.

Άσκηση 4.2: Αποκωδικοποίηση Γλώσσας Μηχανής – Disassembler

Σε αυτή την άσκηση στόχος σας είναι να αποκωδικοποιήσετε το παρακάτω απόσπασμα κώδικα μηχανής του MIPS και να βρείτε τον ισοδύναμο κώδικα σε assembly, ο οποίος περιλαμβάνει εντολές που έχετε διδαχθεί μέχρι στιγμής στο μάθημα. Σας δίνουμε δύο στοιχεία, τη διεύθυνση κάθε εντολής και την κωδικοποίηση της εντολής, η οποία ακολουθεί το instruction format του επεξεργαστή MIPS.

[0x00400038]	0x8d4c0000
[0x0040003c]	0x218c0001
[0x00400040]	0xad6c0000
[0x00400044]	0x21080001
[0x00400048]	0x216b0004
[0x0040004c]	0x218c0004
[0x00400050]	0x1509fffa

Γράψτε σε ένα αρχείο text με όνομα **ask4.txt** τον ισοδύναμο κώδικα σε assembly του MIPS. Εξηγήστε λεπτομερώς πώς κάνετε την αποκωδικοποίηση όλων των πεδίων της κάθε εντολής. Τέλος, περιγράψτε με μία πρόταση τι λειτουργία εκτελεί αυτός ο κώδικας.

Τρόπος Παράδοσης:

Για να παραδώσετε την άσκηση χρησιμοποιήστε την ακόλουθη διαδικασία. Συνδεθείτε σε ένα μηχάνημα Linux του τμήματος με το login και password που χρησιμοποιείτε στο Τμήμα. Η λίστα μηχανημάτων Linux του Τμήματος είναι διαθέσιμη [εδώ](#). Ετοιμάστε ένα directory με το αρχείο που σας ζητάει η άσκηση (ask4.txt). Ας υποθέσουμε ότι το όνομα του directory είναι <somepath>/mydir. Μετακινηθείτε στο directory <somepath> και εκτελέστε την εντολή:

```
/usr/local/bin/submit exercise4@hy225 mydir
```

Η διαδικασία submit θα σας ζητήσει να επιβεβαιώσετε την αποστολή των αρχείων.

Περισσότερες πληροφορίες και αναλυτικές οδηγίες για τη νέα διαδικασία submit είναι διαθέσιμες εκτελώντας:

```
man submit
```

σε κάποιο από τα μηχανήματα Linux του Τμήματος. Σημειώστε ότι η απομακρυσμένη πρόσβαση στα μηχανήματα του Τμήματος γίνεται από 1.2.2010 με νέο ασφαλή τρόπο μέσω gateways και με χρήση tunnel.