


Προηγμένη Αρχιτεκτονική Υπολογιστών

Εισαγωγή

Μιχάλης Ψαράκης



Βιβλιογραφία

- Εύδοξος:
 - Βιβλίο Α:
 - «Αρχιτεκτονική Υπολογιστών», 4η Έκδοση, J.H. Hennessy , D.A. Patterson Εκδόσεις Τζιόλα, 2011 (Μεταφρασμένο)
 - Βιβλίο Β
 - «Οργάνωση και Σχεδίαση Υπολογιστών: Η Διασύνδεση Υλικού και Λογισμικού», 4η έκδοση D.A.Patterson, J.H.Hennessy, Εκδόσεις Κλειδάριθμος, 2010 (Μεταφρασμένο)

Προηγμένη Αρχιτεκτονική Υπολογιστών Εισαγωγή 2

Ιστοσελίδα

- Ιστοσελίδα μαθήματος στη νέα πλατφόρμα τηλεκπαίδευσης: <http://gunet2.cs.unipi.gr/eclass>
 - Εγγραφείτε ως χρήστες
- Διαφάνειες
 - Μπορείτε να τις κατεβάσετε από την ιστοσελίδα του μαθήματος
- Επιπλέον βιβλιογραφία
 - Στη βιβλιοθήκη του Πανεπιστημίου Πειραιώς
 - Σχετική βιβλιογραφία στην ιστοσελίδα του μαθήματος

Βαθμολογία μαθήματος

- Η εξέταση του μαθήματος γίνεται μόνο με την υποβολή και εξέταση εργασιών
 - Δεν υπάρχει γραπτή εξέταση
- Οι εργασίες είναι ομαδικές
 - Κάθε ομάδα μπορεί να αποτελείται από 1 ή 2 άτομα
- Θα πρέπει να παραδώσετε 3 εργασίες
 - Οι εργασίες είναι υποχρεωτικές
 - Τελικός βαθμός:
 $0.4 * \text{Εργασία 1} + 0.3 * \text{Εργασία 2} + 0.3 * \text{Εργασία 3}$

Διευκρινίσεις:

- Σε περίπτωση που μια ομάδα δεν παραδώσει και τις 3 εργασίες, μπορεί να διατηρήσει τον βαθμό των εργασιών που παρέδωσε και στην εξεταστική του Σεπτεμβρίου να παραδώσει και τις υπόλοιπες εργασίες
- Ο βαθμός των εργασιών δεν διατηρείται για την επόμενη ακαδημαϊκή χρονιά

Περιεχόμενο μαθήματος

- **Τεχνική διοχέτευσης (pipelining)**
 - Υλοποίηση της διοχέτευσης στον επεξεργαστή MIPS
 - Κίνδυνοι διοχέτευσης/εξαιρέσεις
- **Ιεραρχία μνήμης (memory hierarchy)**
 - Κρυφή μνήμη (cache memory)
 - Εικονική μνήμη (virtual memory)
- **Παραλληλία Επιπέδου Εντολής (Instruction Level Parallelism)**
 - Δυναμικός χρονοπρογραμματισμός (dynamic scheduling)
 - Εικασία (speculation)

Τι πρέπει να θυμάστε ...

- Από το μάθημα
«Αρχιτεκτονική Υπολογιστών»
 - Συμβολική γλώσσα (assembly) του επεξεργαστή MIPS
 - Υλοποίηση της διαδρομής δεδομένων (datapath) και της μονάδας ελέγχου (control unit) του επεξεργαστή MIPS

Βασικές εντολές MIPS

Κατηγορία	Εντολή	Παράδειγμα	Σημασία	Σχόλια
Αριθμητικές πράξεις	add	add \$s1,\$s2,\$s3	$\$s1 = \$s2 + \$s3$	Τρεις τελεστέοι· δεδομένα σε καταχωρητές
	subtract	sub \$s1,\$s2,\$s3	$\$s1 = \$s2 - \$s3$	Τρεις τελεστέοι· δεδομένα σε καταχωρητές
	add immediate	addi \$s1,\$s2,100	$\$s1 = \$s2 + 100$	Χρησιμοποιείται για να προσθέσει σταθερές
Λογικές πράξεις	and	and \$s1,\$s2,\$s3	$\$s1 = \$s2 \& \$s3$	Τρεις τελεστέοι καταχωρητή. AND bit προς bit
	or	or \$s1,\$s2,\$s3	$\$s1 = \$s2 \$s3$	Τρεις τελεστέοι καταχωρητή. OR bit προς bit
	and immediate	andi \$s1,\$s2,100	$\$s1 = \$s2 \& 100$	AND bit προς bit καταχωρητή με σταθερά
	or immediate	ori \$s1,\$s2,100	$\$s1 = \$s2 100$	OR bit προς bit καταχωρητή με σταθερά

Βασικές εντολές MIPS

Κατηγορία	Εντολή	Παράδειγμα	Σημασία	Σχόλια
Μεταφορά δεδομένων	load word	lw \$s1,100(\$s2)	$\$s1 = \text{Memory}[\$s2+100]$	Λέξη από τη μνήμη σε καταχωρητή
	store word	sw \$s1,100(\$s2)	$\text{Memory}[\$s2+100] = \$s1$	Λέξη από καταχωρητή στη μνήμη
Διακλάδωση με συνθήκη	branch on equal	beq \$s1,\$s2,L	αν ($\$s1 == \$s2$) πήγαινε στο L	Έλεγχος ισότητας και διακλάδωση
	set on less than	slt \$s1,\$s2,\$s3	αν ($\$s2 < \$s3$) τότε $\$s1 = 1$ · αλλιώς $\$s1 = 0$	Σύγκριση μικρότερο από· χρήση με τις beq, bne
Άλμα χωρίς συνθήκη	Jump	j L	πήγαινε στο L	Άλμα στη διεύθυνση προορισμού

Πεδία των εντολών του MIPS

Μορφή R (R-format)

op	rs	rt	rd	shamt	funct
----	----	----	----	-------	-------

6 bit 5 bit 5 bit 5 bit 5 bit 6 bit

- **op**: opcode (operation code — κώδικας λειτουργίας)
- **rs**: Τελεστής προέλευσης πρώτου καταχωρητή
- **rt**: Τελεστής προέλευσης δεύτερου καταχωρητή
- **rd**: Τελεστής καταχωρητή προορισμού
- **shamt**: Ποσότητα ολίσθησης (shift amount)
- **funct**: Παραλλαγή πράξης (λειτουργίας)

Πεδία των εντολών του MIPS

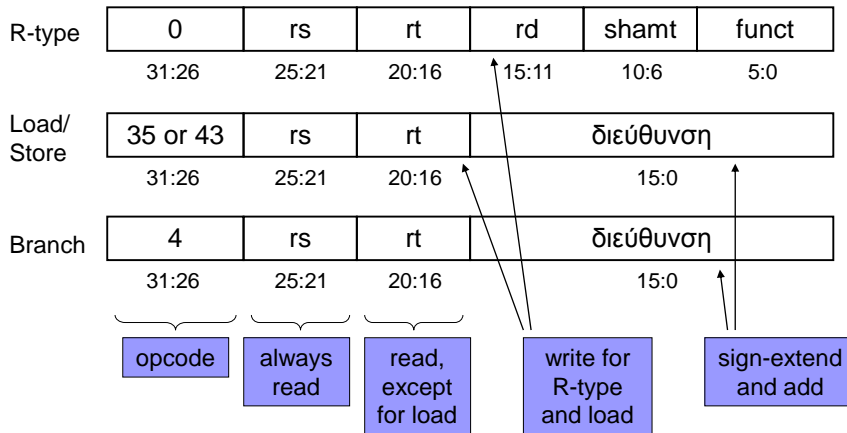
Μορφή I (I-format)

op	rs	rt	Σταθερά ή διεύθυνση
----	----	----	---------------------

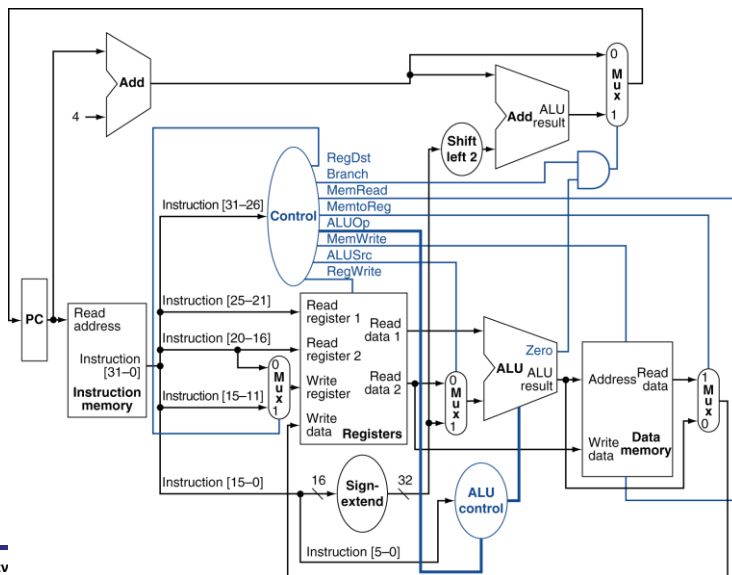
6 bit 5 bit 5 bit 16 bit

- Παράδειγμα εντολής μορφής I
lw \$1,32(\$2)
 - \$2 → rs
 - \$1 → rt
 - 32(άμεσο πεδίο) → διεύθυνση

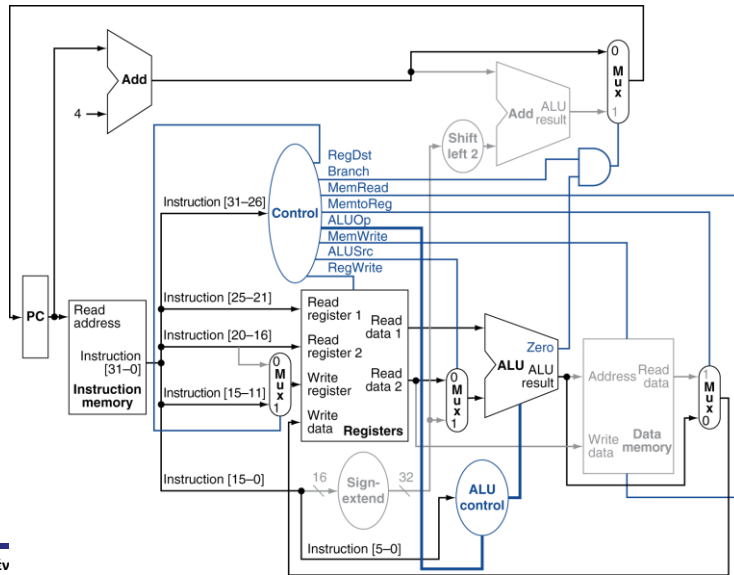
Πεδία των εντολών του MIPS



Διαδρομή δεδομένων ενός κύκλου



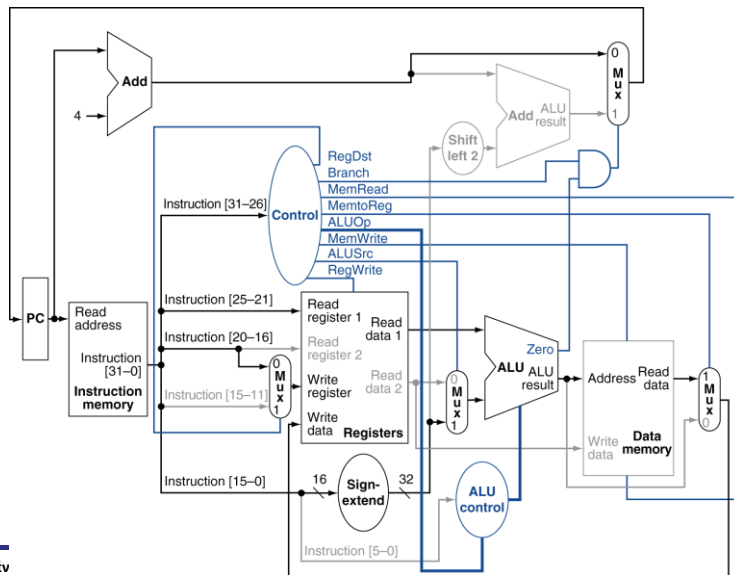
Εντολή R



Προηγμέν

13

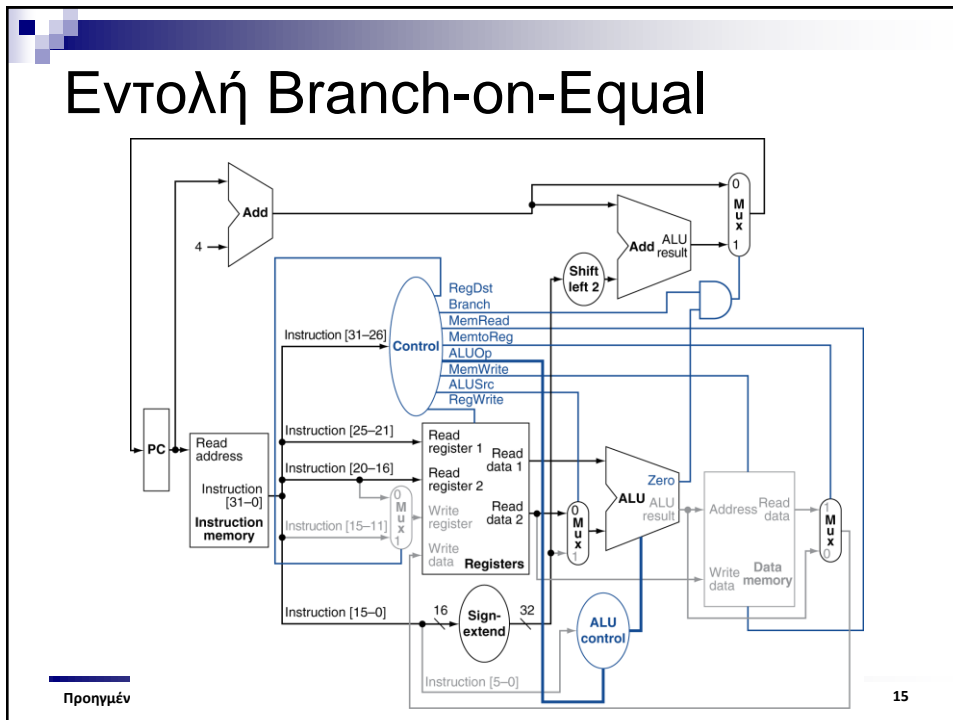
Εντολή Load



Προηγμέν

14

Εντολή Branch-on-Equal



Απόδοση υλοποίησης ενός κύκλου

- Ποιο είναι το CPI της υλοποίησης ενός κύκλου;
- **Παράδειγμα:** Απόδοση Μηχανής Ενός Κύκλου
- Υποθέτουμε ότι οι χρόνοι καθυστέρησης των λειτουργικών μονάδων είναι:
 - Μονάδες μνήμης: 200 picosecond (ps)
 - ALU και αθροιστές: 100 ps
 - Αρχείο καταχωρητών (ανάγνωση ή εγγραφή): 50 ps
 - Πολυπλέκτες, μονάδα ελέγχου, προσβάσεις του PC, επέκταση προσήμου, και συνδέσεις = καθυστέρηση 0

Απόδοση υλοποίησης ενός κύκλου

- Ποια από τις παρακάτω μηχανές θα είναι ταχύτερη;
- **Μηχανή A:** κάθε εντολή λειτουργεί σε 1 κύκλο ρολογιού σταθερού μήκους
- **Μηχανή B:** κάθε εντολή εκτελείται σε 1 κύκλο ρολογιού με ρολόι μεταβλητού μήκους, το οποίο για κάθε εντολή έχει τόση διάρκεια όσο χρειάζεται.
 - Μια τέτοια προσέγγιση δεν είναι και πολύ πρακτική!
- Υποθέστε το παρακάτω μίγμα εντολών:
 - 25% φορτώσεις
 - 10% αποθηκεύσεις
 - 45% εντολές ALU
 - 15% διακλαδώσεις
 - 5% άλματα.

Απόδοση υλοποίησης ενός κύκλου

Κατηγορία εντολής	Μήμη εντολών	Ανάγνωση καταχωρητή	Λειτουργία ALU	Μήμη δεδομένων	Εγγραφή καταχωρητή	Συνολικός χρόνος
R-format (add, sub, and, or, slt)	200	50	100	0	50	400 ps
Load word (lw)	200	50	100	200	50	600 ps
Store word (sw)	200	50	100	200		550 ps
Branch (beq)	200	50	100			350 ps
Jump (j)	200					200 ps

Απόδοση υλοποίησης ενός κύκλου

- Κύκλος ρολογιού CPU-A = **600 ps**
- Κύκλος ρολογιού CPU-B =
 $600 \times 25\% + 550 \times 10\% + 400 \times 45\% +$
 $350 \times 15\% + 200 \times 5\% =$
= 447,5 ps