

# ΗΥ 232

## Οργάνωση και Σχεδίαση Υπολογιστών

### Διάλεξη 2

#### Οργάνωση μνήμης Καταχωρητές του MIPS Εντολές του MIPS 1

**Νίκος Μπέλλας**  
Τμήμα Μηχανικών Η/Υ, Τηλεπικοινωνιών και Δικτύων

# Σύνολο Εντολών

- Το ρεπερτόριο των εντολών ενός υπολογιστή
- Διαφορετικοί υπολογιστές έχουν διαφορετικά σύνολα εντολών
  - Αλλά με πολλά κοινά χαρακτηριστικά
- Οι πρώτοι υπολογιστές είχαν πολύ απλά σύνολα εντολών
  - Απλοποιημένη υλοποίηση
- Πολλοί μοντέρνοι υπολογιστές έχουν επίσης απλά σύνολα εντολών
- Ο σχεδιασμός ενός επιτυχημένου συνόλου εντολών για έναν επεξεργαστή είναι η σημαντικότερη δουλειά ενός αρχιτέκτονα υπολογιστών (Computer Architect)

# Είδη εντολών και τελούμενων (*operands*)

- Οι μονάδες του υλικού στις οποίες δρούν οι εντολές τις *assembly* (δηλ. αποτελούν τα τελούμενά τους) είναι οι *καταχωρητές* (*registers*) και οι *θέσεις μνήμης*
  - Καταχωρητές: χώρος προσωρινής αποθήκευσης δεδομένων προς επεξεργασία
  - Μνήμη: κύριος χώρος αποθήκευσης των δεδομένων
  - Οι Καταχωρητές και η Κύρια Μνήμη είναι δύο διαφορετικά πράγματα

# Το Σύνολο Εντολών του MIPS

- Ο MIPS χρησιμοποιείται ως παράδειγμα σε όλο το βιβλίο
- Μεγάλο μέρος της αγοράς ενσωματωμένων συστημάτων.
  - Εφαρμογές σε ηλεκτρονικές συσκευές, εξοπλισμό δικτύων/αποθήκευσης, φωτογραφικές μηχανές, εκτυπωτές, ...
- Τυπικό παράδειγμα πολλών μοντέρνων συνόλων εντολών

# Καταχωρητές του MIPS

- 32 καταχωρητές των 32-bits ο καθένας
- 32-bits αποτελούν μια λέξη (word) του Η/Υ
- Ονόματα καταχωρητών: \$0 , \$1 , ... , \$31
- Συμβολικά ονόματα καταχωρητών:

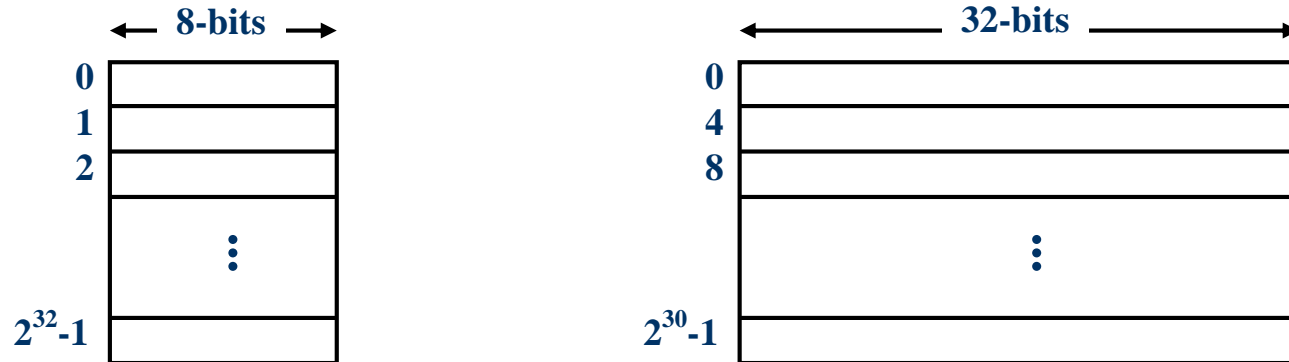
\$0	\$2-\$3	\$4-\$7	\$8-\$15	\$16-\$23	\$24-\$25	\$28	\$29	\$30	\$31
\$zero	\$v0-\$v1	\$a0-\$a3	\$t0-\$t7	\$s0-\$s7	\$t8-\$t9	\$gp	\$sp	\$fp	\$ra

- Οι καταχωρητές \$1 (συμβολικό \$at) και \$26-\$27 (συμβολικά \$k0-\$k1) δεν είναι προσβάσιμοι στον προγραμματιστή, ενώ κάποιοι άλλοι επιτελούν ειδικές λειτουργίες
- Ο καταχωρητής \$0 (ή \$zero) έχει πάντοτε την τιμή 0

# Οργάνωση μνήμης

- Η μνήμη είναι οργανωμένη σε *bytes*
- Διευθυνσιοδότηση μνήμης για κάθε byte ξεχωριστά (byte addressing)

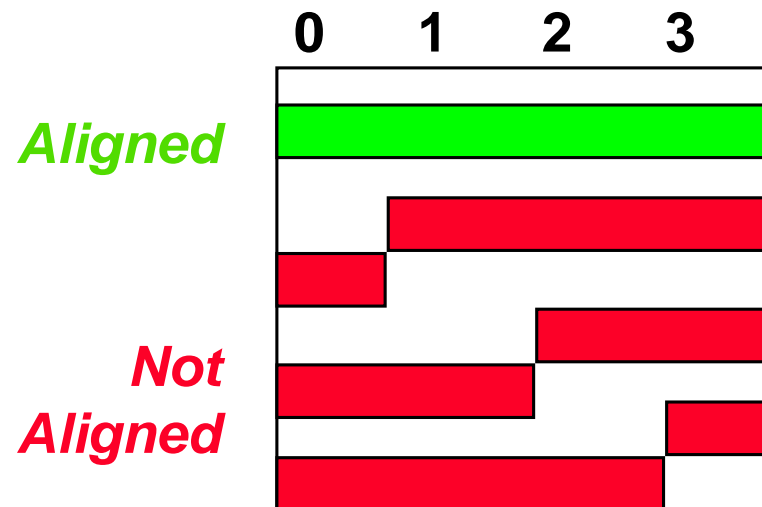
=> οι λέξεις (32-bits ή 4 bytes) στη μνήμη έχουν διευθύνσεις που διαφέρουν ανά 4 μεταξύ τους



- 32-bits διευθύνσεων =>  $2^{32}$  bytes ή  $2^{30}$  λέξεις συνολικά

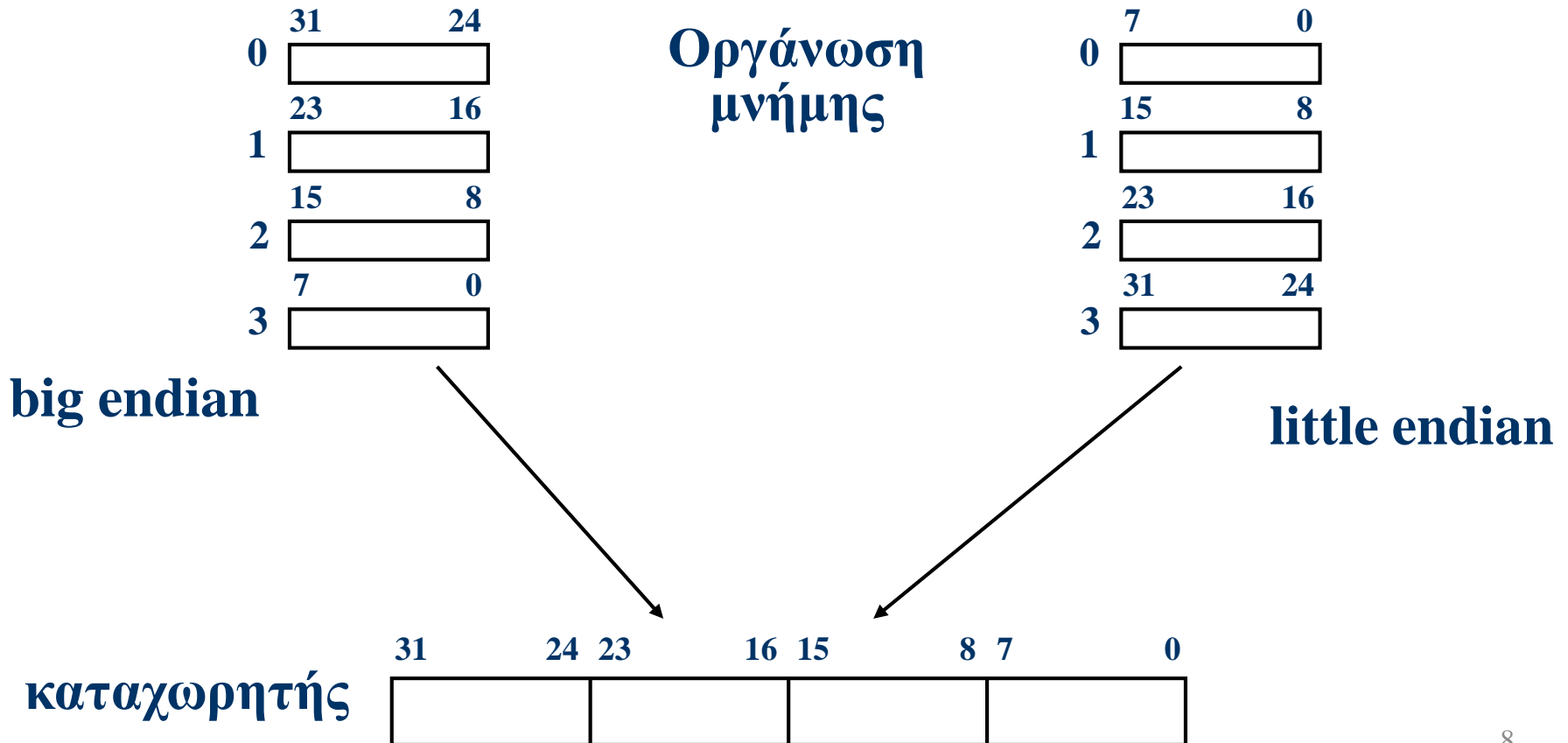
# Οργάνωση μνήμης

- Η τοποθέτηση των λέξεων στη μνήμη πρέπει να γίνεται με *ευθυγραμμισμένο* τρόπο (alignment), δηλ. η διεύθυνση του πρώτου byte να είναι πολλαπλάσιο του 4



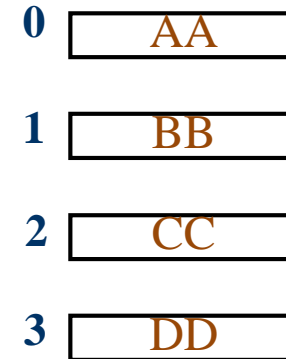
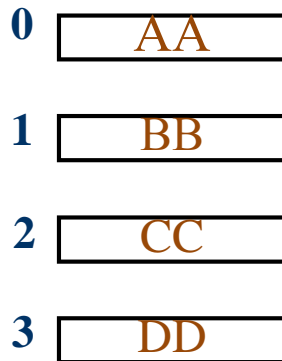
# Οργάνωση μνήμης

- Υπάρχουν δύο τρόποι διάταξης των bytes σε μια λέξη της μνήμης
- Το πρότυπο του MIPS ακολουθεί τη διάταξη *big endian* (αντίθετα με τον Intel 80x86)





# Οργάνωση μνήμης



Οργάνωση  
μνήμης



**big endian**



**little endian**

# Επέκταση προσήμου

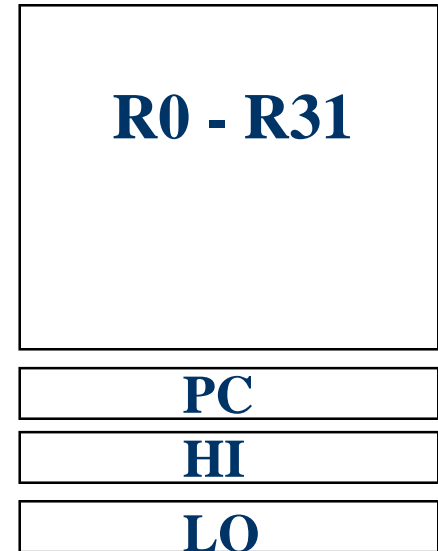
- Αναπαράσταση αριθμού με περισσότερα bits
  - Διατηρώντας την αριθμητική τιμή του
- Αντιγράψτε το πρόσημο αριστερά
  - π.χ. απρόσημες τιμές: επεκτείνονται με μηδενικά
- Παραδείγματα: 8-bit σε 16-bit
  - +2: 0000 0010 => 0000 0000 0000 0010
  - -2: 1111 1110 => 1111 1111 1111 1110

# Μοντέλο μηχανής MIPS

- Είδη εντολών στον MIPS:

- Εντολές πράξεων (ακεραίων ή κινητής υποδιαστολής)
- Εντολές μεταφοράς δεδομένων (από και προς την μνήμη)
- Εντολές διακλάδωσης (με ή χωρίς συνθήκη)
- Ειδικές εντολές

Καταχωρητές



## 3 μορφές εντολών του MIPS: όλες 32 bits

<b>op</b>	<b>rs</b>	<b>rt</b>	<b>rd</b>	<b>sa</b>	<b>funct</b>
-----------	-----------	-----------	-----------	-----------	--------------

Μορφή R

<b>op</b>	<b>rs</b>	<b>rt</b>	<b>immediate</b>
-----------	-----------	-----------	------------------

Μορφή I

<b>op</b>	<b>jump target</b>
-----------	--------------------

Μορφή J

# Αναπαράσταση Εντολών

- Οι εντολές αναπαριστούνται στο δυαδικό σύστημα
  - Το αποτέλεσμα λέγεται κώδικας μηχανής
- Εντολές MIPS
  - Κωδικοποιούνται ως λέξεις (32-bits)
  - Υπάρχει μικρός αριθμός από μορφές (formats) που κωδικοποιούν κώδικες λειτουργίας (opcodes), αριθμούς καταχωρητών,...

# Εντολές πράξεων στον MIPS

- Όλες οι εντολές πράξεων γίνονται αποκλειστικά μεταξύ καταχωρητών  
=> για επεξεργασία θέσεων μνήμης θα πρέπει πρώτα τα δεδομένα να μεταφερθούν σε κατάλληλους καταχωρητές
- Στις εντολές πράξεων χρησιμοποιούνται συνήθως οι καταχωρητές  $\$s0-\$s7$  για τελικά αποτελέσματα (τα οποία φυλάσσονται ως μεταβλητές), και οι καταχωρητές  $\$t0-\$t9$  για ενδιάμεσα (προσωρινά) αποτελέσματα

# Εντολές πράξεων στον MIPS

## Παράδειγμα

- Κώδικας C:

```
f = (g + h) - (i + j);
```

– Οι f, g, h, i, j είναι στους \$s0, \$s1, \$s2, \$s3, \$s4

- Μεταγλωτισμένος κώδικας MIPS:

```
add $t0, $s1, $s2
```

```
add $t1, $s3, $s4
```

```
sub $s0, $t0, $t1
```

destination ← source1 op source2

- Η επιλογή των καταχωρητών γίνεται είτε από τον προγραμματιστή σε assembly είτε από τον μεταφραστή γλώσσας ανωτέρου επιπέδου

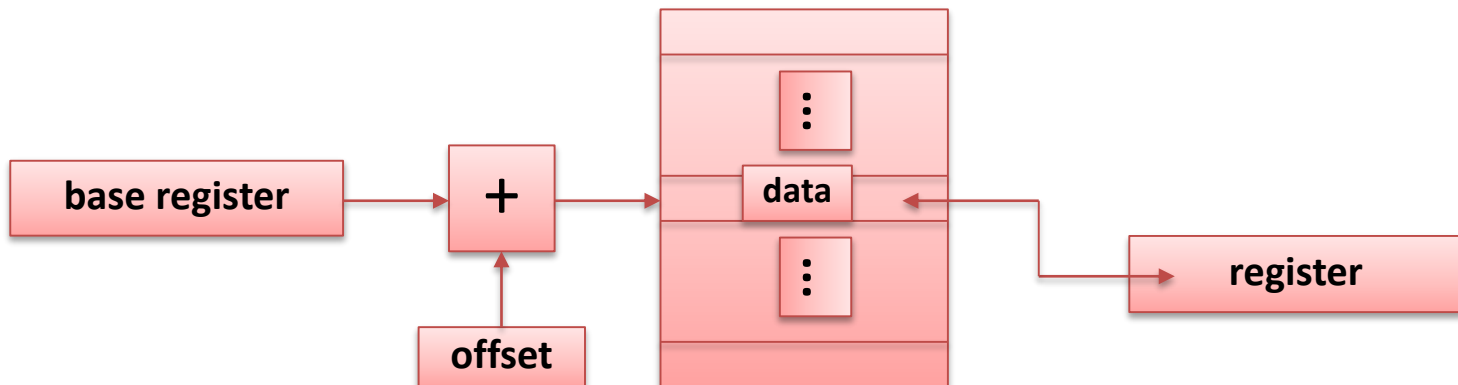
# Εντολές πράξεων στον MIPS

- Οι εντολές πράξεων καλύπτουν όλους τους διαφορετικούς αριθμητικούς τελεστές (arithmetic operators) μιάς γλώσσας προγραμματισμού
  - Όχι πάντα 1-1 αντιστοίχιση
- Παράδειγμα:

C	MIPS assembly
+	add, addu, addi, addiu
-	sub, subu
*	mult, multu
/	div, divu
<<	sll, sllv
&	and, andi
	or, ori

# Εντολές μεταφοράς

- Αναφέρονται ως εντολές φόρτωσης (load) καταχωρητή και αποθήκευσης (store) καταχωρητή:  
 $lw \$s0, 100(\$s1) \iff \$s0 = Mem[\$s1+100]$   
 $sw \$s0, 100(\$s1) \iff Mem[\$s1+100] = \$s0$
- Αποτελούν τον μοναδικό τρόπο πρόσβασης στη μνήμη στον MIPS
- Η διεύθυνση πρόσβασης σχηματίζεται από το άθροισμα μιας διεύθυνσης βάσης που περιέχεται σε έναν *καταχωρητή βάσης* (base register), και μιας 16-bit σταθεράς μετατόπισης (offset) που δίνεται άμεσα ως όρισμα της εντολής





# Εντολές μεταφοράς

- Παράδειγμα προσπέλασης σύνθετης δομής.

Εάν έχουμε έναν integer πίνακα A (`int A[100]`) του οποίου η διεύθυνση βάσης (δηλ. η διεύθυνση του πρώτου στοιχείου) βρίσκεται στον καταχωρητή βάσης `$s1`, τότε η εντολή:

`lw $s0,32($s1)`

φορτώνει το όγδοο στοιχείο του πίνακα στον καταχωρητή `$s0`, ενώ η εντολή:

`sw $s0,48($s1)`

αποθηκεύει το εν λόγω στοιχείο - από τον `$s0` - στη θέση του δωδέκατου στοιχείου (δηλ. `A[12] = A[8]`)

# Παράδειγμα

$\$s3 = \&A[0]$



Απόσταση (Offset)

- Κώδικας C:

$g = h + A[8];$

- Το  $g$  είναι στον  $\$s1$ , το  $h$  στον  $\$s2$ , και η διεύθυνση (βάση) του  $A$  στον  $\$s3$

- Μεταγλωττισμένος κώδικας MIPS:

- Η θέση 8 βρίσκεται 32 bytes μετά την αρχή του  $A$ 
  - 4 bytes ανά λέξη

$lw \ \$t0, 32(\$s3) \quad \# \text{ load word}$

$add \ \$s1, \$s2, \$t0$

# Τελεστέοι μνήμης (Memory Operands)

- Η μνήμη χρησιμοποιείται για σύνθετα δεδομένα
  - Πίνακες, δομές, δυναμικά δεδομένα
- Για να εφαρμόσετε αριθμητικές πράξεις
  - Φορτώστε (load) τις τιμές από τη μνήμη σε καταχωρητές.
  - Αποθηκεύστε (store) το αποτέλεσμα από τον καταχωρητή στη μνήμη.
- Η μνήμη είναι οργανωμένη σε bytes
  - Κάθε διεύθυνση προσδιορίζει ένα byte (8 bits)
- Οι λέξεις είναι ευθυγραμμισμένες στη μνήμη.
  - Κάθε διεύθυνση πρέπει να είναι πολ/σιο του 4.
- Ο MIPS είναι Big Endian
  - Το πιο σημαντικό byte (most-significant byte) βρίσκεται στη μικρότερη διεύθυνση