ΠΛΗ10

ΕΝΟΤΗΤΑ 1: Εισαγωγή στους Η/Υ

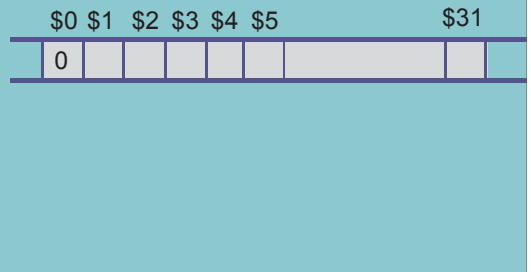
Μάθημα 1.5:
Προγραμματιστικά Στοιχεία του Επεξεργαστή MIPS

Δημήτρης Ψούνης

Περιεχόμενα Μαθήματος

Α. Θεωρία
1. Ο επεξεργαστής MIPS
1. Εισαγωγή

Ο **επεξεργαστής MIPS** χρησιμοποιεί αρχιτεκτονική RISC και μπορεί:
- Να τρέξει εντολές Assembly που έχουν οριστεί σε αυτόν.
- Χρησιμοποιεί ως αποθηκευτικό χώρο 32 καταχωρητές με ονόματα $0, $1, …, $32
  - Ο καταχωρητής $0 έχει αποθηκευμένη την τιμή 0.
  - Τους χρησιμοποιούμε ως μεταβλητές στο πρόγραμμα μας.
  - Κάθε καταχωρητής είναι 4bytes (32 bits)
- Σε μία αρκετά απλοποιημένη εκδοχή, ο τρόπος εκτέλεσης ενός προγράμματος φαίνεται στο σχήμα:



Σημείωση:
Η παρουσίαση αυτή είναι ένα βοήθημα για ερωτήματα εργασιών της ΠΛΗ10 και δεν φιλοδοξεί να είναι ολοκληρωμένη παρουσίαση του επεξεργαστή MIPS.

Α. Θεωρία
1. Ο επεξεργαστής MIPS
2. Εντολές Διαχείρισης Σταθερών

Οι Εντολές Διαχείρισης Σταθερών (li-load intermediate) χρησιμεύουν στο να αποθηκευτεί μια σταθερά σε έναν καταχωρητή.

Σύνταξη: `li Rdest, Imm`
όπου:
- `Imm`: ένας αριθμός
- `Rdest`: καταχωρητής στον οποίον γίνεται η αποθήκευση

Παράδειγμα:
Η εντολή `li $1, 10` αποθηκεύει στον καταχωρητή $1 την τιμή 10.

Άσκηση: Κατασκευάστε πρόγραμμα που αποθηκεύει στον καταχωρητή $2 την τιμή 5 και στον καταχωρητή $8 την τιμή 12.

Α. Θεωρία
1. Ο επεξεργαστής MIPS
3. Αριθμητικές Εντολές

Οι Αριθμητικές Εντολές κάνουν πρόσθεση και αφαίρεση

Σύνταξη:

```
add Rdest, Rsrc1, Rsrc2
addi Rdest, Rsrc1, Imm
```

```
sub Rdest, Rsrc1, Rsrc2
```

όπου:
- `Rdest,Rsrc1,Rsrc2`: καταχωρητές, `imm` αριθμός
- `add`: πρόσθεση καταχωρητών (ισοδύναμα σε ψευδογλώσσα) `Rdest:=Rsrc1+Rsrc2`
- `addi`:πρόσθεση καταχωρητή-αριθμού (ισοδύναμα σε ψευδογλώσσα) `Rdest:=Rsrc1+Imm`
- `sub`:αφαίρεση καταχωρητών (ισοδύναμα σε ψευδογλώσσα) `Rdest:=Rsrc1-Rsrc2`

Παράδειγμα:

Αν ο `$1` έχει την τιμή 5 και ο `$2` έχει την τιμή 3.

$0	$1	$2	$3
0	5	3	

Η εντολή `add $3, $1, $2` αποθηκεύει στον καταχωρητή `$3` την τιμή 8.

Η εντολή `add $1, $1, $2` αποθηκεύει στον καταχωρητή `$1` την τιμή 8.

Η εντολή `addi $1, $1, 2` αποθηκεύει στον καταχωρητή `$1` την τιμή 7.

Η εντολή `sub $3, $1, $2` αποθηκεύει στον καταχωρητή `$3` την τιμή 2.

Α. Θεωρία
1. Ο Επεξεργαστής MIPS
3. Αριθμητικές Εντολές

Άσκηση: Ποια η κατάσταση των καταχωρητών $1,$2, $3 με το πέρας του ακόλουθου προγράμματος:

```
li $1, 7
li $2, 10
addi $3, $1, 2
add $3, $3, $3
sub $3, $2, $3
addi $2, $0, 8
```


Α. Θεωρία
1. Ο επεξεργαστής MIPS
4. Λογικές Εντολές (Λογικές Πράξεις)

Οι Λογικές Εντολές εκτελούν τις λογικές πράξεις **and, or, nand, nor, xor, not** στις δυαδικές αναπαραστάσεις των αριθμών που είναι αποθηκευμένοι στους καταχωρητές

Σύνταξη:

```
and Rdest, Rsrc1, Rsrc2
andi Rdest, Rsrc, Imm
```

```
not Rdest, Rsrc
```

όπου:
- `Rdest,Rsrc,Rsrc1,Rsrc2`: καταχωρητές, `Imm` αριθμός
- `and`: λογικό ΚΑΙ στις δυαδικές συμβολοσειρές Rsrc1 και Rsrc2 και αποθήκευση στο `Rdest`
  - Αντίστοιχα και οι `or, nand, nor, xor, not`
- `andi`: λογικό ΚΑΙ στις δυαδικές συμβολοσειρές `Rsrc` και `Imm` και αποθήκευση στο `Rdest`

Παράδειγμα:

Αν ο `$1` έχει την τιμή 5 και ο `$2` έχει την τιμή 3.

$0	$1	$2	$3
0	5	3	

Η εντολή `and $3, $1, $2` αποθηκεύει στον καταχωρητή `$3` την τιμή 1.

Η εντολή `or $3, $1, $2` αποθηκεύει στον καταχωρητή `$3` την τιμή 7.

Η εντολή `andi $3, $1, 2` αποθηκεύει στον καταχωρητή `$1` την τιμή 7.

Α. Θεωρία
1. Ο επεξεργαστής MIPS
4. Λογικές Εντολές (Εντολές Ολίσθησης)

Οι Εντολές Ολίσθησης sll και srl εκτελούν αριστερή και δεξιά ολίσθηση αντίστοιχα του περιεχομένου του καταχωρητή.

Σύνταξη:

```
sll Rdest, Rsrc1, Rsrc2
```

```
srl Rdest, Rsrc1, Rsrc2
```

όπου:
- `Rdest,Rsrc1` καταχωρητές, `Rsrc2` αριθμός ή καταχωρητής
- `sll`: Μετακινεί όλα τα bits του `Rsrc1` κατά `Rsrc2` θέσεις αριστερά
  - Αν `Rsrc2=1` τότε ισοδυναμεί με διπλασιασμό του αριθμού
- `srl`: Μετακινεί όλα τα bits του `Rsrc1` κατά `Rsrc2` θέσεις αριστερά
  - Αν `Rsrc2=1` τότε ισοδυναμεί με υποδιπλασιασμό του αριθμού (αριθμός div 2)

Παράδειγμα:

Αν ο `$1` έχει την τιμή 5

$1

00000000	00000000	00000000	00000101

Η εντολή `sll $1, $1, 1` αποθηκεύει στον καταχωρητή `$1` την τιμή 10.

$1

00000000	00000000	00000000	00001010

.

Α. Θεωρία
1. Ο Επεξεργαστής MIPS
4. Λογικές Εντολές

Άσκηση: Ποιο το περιεχόμενο των καταχωρητών $1-$5 με το πέρας της εκτέλεσης του προγράμματος;

```
li $1, 11
andi $2, $1, 1
srl $1, $1, 1
andi $3, $1, 1
srl $1, $1, 1
andi $4, $1, 1
srl $1, $1, 1
andi $5, $1, 1
srl $1, $1, 1
```


Α. Θεωρία
1. Ο επεξεργαστής MIPS
5. Εντολές Μεταφοράς Ελέγχου (Εντολές Σύγκρισης)

Οι Εντολές Μεταφοράς Ελέγχου αλλάζουν τη σειριακή ροή εκτέλεσης των εντολών ανάλογα με το αν ισχύει κάποια συνθήκη. Θα μελετήσουμε μία από αυτές (διακλάδωση αν ίσα):

Σύνταξη: `beq Rsrc1, Rsrc2, label`

όπου:
- `Rdest,Rsrc,Rsrc1,Rsrc2`: καταχωρητές
- `label`: ετικέτα εντολής
- Ελέγχεται αν οι τιμές των καταχωρητών `Rsrc1` και `Rsrc2` είναι ίσες.
  - Αν ναι γίνεται διακλάδωση στην ετικέτα `label`
  - Αν όχι γίνεται μετάβαση στην επόμενη εντολή

Αντίστοιχα ορίζονται εντολές μεταφοράς ελέγχου για άλλες συγκρίσεις:

Σύνταξη:

`bge Rsrc1, Rsrc2, label`	$Rsrc1 \geq Rsrc2$	`bgt Rsrc1, Rsrc2, label`	$Rsrc1 > Rsrc2$
`ble Rsrc1, Rsrc2, label`	$Rsrc1 \leq Rsrc2$	`blt Rsrc1, Rsrc2, label`	$Rsrc1 < Rsrc2$
`beq Rsrc1, Rsrc2, label`	$Rsrc1 = Rsrc2$	`bne Rsrc1, Rsrc2, label`	$Rsrc1 \neq Rsrc2$
`j label`	Προσοχή! Διακλάδωση χωρίς συνθήκη!		

Α. Θεωρία
1. Ο Επεξεργαστής MIPS
5. Εντολές Μεταφοράς Ελέγχου (Εντολές Σύγκρισης)

Άσκηση 1: Τι κάνει το ακόλουθο τμήμα κώδικα;

```
  bge $4, 0, label1
  sub $4, $0, $4
label1:
  addi $4, $4, 1
```


Α. Θεωρία
1. Ο Επεξεργαστής MIPS
5. Εντολές Μεταφοράς Ελέγχου (Εντολές Σύγκρισης)

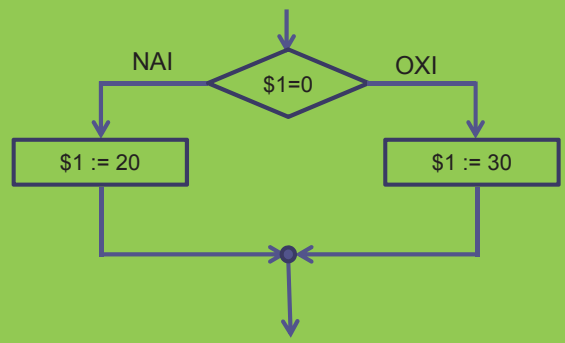
Παρατήρηση:
Με τις εντολές διακλάδωσης μπορούμε να υλοποιήσουμε την **εντολή απόφασης** (όπως την είδαμε και στο διάγραμμα ροής προγράμματος)
Π.χ. το ακόλουθο τμήμα κώδικα εκτελεί έναν απλό έλεγχο για το αν το περιεχόμενου καταχωρητή R1 είναι ίσο με το 0 και αν ναι αλλάζει το περιεχόμενο και το κάνει ίσο με 20. Αν όχι τότε αλλάζει το περιεχόμενο και το κάνει ίσο με 30. Το πρόγραμμα έπειτα συνεχίζει την εκτέλεση του.

```
…
  beq $1, $0, ltrue
  j lfalse

ltrue:
  li $1, 20
  j lexit:

lfalse:
  li $1, 30
  j lexit:

lexit:
….
```



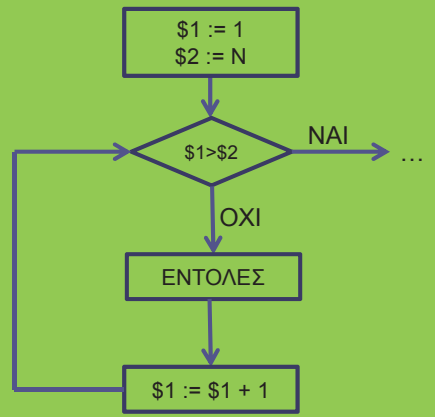
Α. Θεωρία

1. Ο Επεξεργαστής MIPS

5. Εντολές Μεταφοράς Ελέγχου (Εντολές Σύγκρισης)

Παρατήρηση:
Με τις εντολές διακλάδωσης μπορούμε να υλοποιήσουμε την **εντολή επανάληψης** (όπως την είδαμε και στο διάγραμμα ροής προγράμματος)
Π.χ. το ακόλουθο τμήμα κώδικα εκτελεί ένα πλήθος εντολών N φορές

```
…
  li $1, 1
  li $2, N
loop:
  bgt $1, $2, lexit
  ΕΝΤΟΛΕΣ
  addi $1, $1, 1
  j loop
lexit:
….
```



Α. Θεωρία

1. Ο Επεξεργαστής MIPS

5. Εντολές Μεταφοράς Ελέγχου (Εντολές Σύγκρισης)

Άσκηση 2: Τι κάνει το ακόλουθο τμήμα κώδικα;

```
  li $1, 0
  li $2, 3
  li $4, 0
  li $5, 5
loop:
  add $1, $1, $2
  addi $4, $4, 1
  bne $5, $4, loop
```


Α. Θεωρία

1. Ο επεξεργαστής MIPS

6. Εντολές Μεταφοράς Δεδομένων

Η εντολή μεταφοράς δεδομένων move μεταφέρει τα περιεχόμενα ενός καταχωρητή σε έναν άλλο καταχωρητή.

Σύνταξη: `move Rdest, Rsrc`

όπου:
- `Rdest,Rsrc` καταχωρητές
  - Αντιγράφει τα περιεχόμενα του καταχωρητή Rsrc στον καταχωρητή Rdest

Άσκηση: Τι κάνει το ακόλουθο τμήμα κώδικα;

```
li $5, 0
move $4, $3
Loop1:
  add $4, $4, $3
  addi $5, $5, 1
  bne $5, 4, Loop1
```