

# Αρχιτεκτονική Υπολογιστών

## Κεφάλαιο 2

### Ασκήσεις

## Άσκηση 1

- Κώδικας C:

```
f = g + h + B[4];
```

```
f = g - A[B[4]];
```

f, g, h, στους \$s0, \$s1, \$s2, και διευθύνσεις βάσης των πινάκων A και B στους \$s6 και \$s7

Ποιος είναι ο αντίστοιχος κώδικας MIPS;

```
lw $s0, 16($s7)      #f = B[4]
add $s0, $s0, $s1     #f = B[4] + g
add $s0, $s0, $s2     #f = B[4] + g + h
```

- 

```
lw $t0, 16($s7)
add $t0, $t0, $s6
lw $s0, 0($t0)
sub $s0, $s1, $s0
```

Ποιο είναι το λάθος;

- Σωστή λύση

```
lw $t0, 16($s7)
sll $t0, $t0, 2
add $t0, $t0, $s6
lw $s0, 0($t0)
sub $s0, $s1, $s0
```

## Άσκηση 2

- Κώδικας C:

```
f = -g + h + B[1];
```

```
f = A[B[g]+1];
```

f, g, h, στους \$s0, \$s1, \$s2, και διευθύνσεις βάσης των πινάκων A και B στους \$s6 και \$s7

Ποιος είναι ο αντίστοιχος κώδικας MIPS;

- 

```
lw  $s0, 4($s7)      # f = B[1]
sub  $s0, $s0, $s1   # f = B[1] - g
add  $s0, $s0, $s2   # f = B[1] - g + h
```

- 

```
sll  $s1, $s1, 2     # $s1 = 4*g
add  $t0, $s7, $s1   # $t0 = διεύθυνση του B[g]
lw   $t0, 0($t0)     # $t0 = B[g]
sll  $t0, $t0, 2     # $t0 = 4*B[g]
add  $t0, $t0, $s6   # $t0 = διεύθυνση του A[B[g]]
lw   $s0, 4($t0)     # f = A[B[g]+1]
```

Εντολές: η γλώσσα του υπολογιστή — 3

## Άσκηση 3

- Καταχωρητές:

a. \$s0=0x70000000 \$s1=0x0FFFFFFF

b. \$s0=0x40000000 \$s1=0x40000000

Ποιο είναι το αποτέλεσμα (\$t0) των εντολών;

- add \$t0, \$s0, \$s1

a. 7FFFFFFF, no overflow

b. 80000000, overflow

- sub \$t0, \$s0, \$s1

a. 60000001, no overflow

b. 0, no overflow

Εντολές: η γλώσσα του υπολογιστή — 4

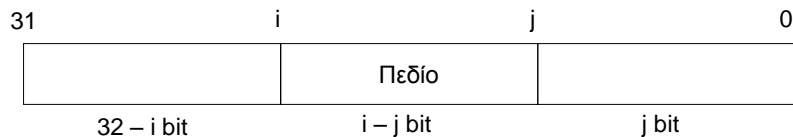
## Άσκηση 4

- Καταχωρητές:
  - a.  $\$t0=0x55555555$   $\$t1=0x12345678$
  - b.  $\$t0=0xBEADFEED$   $\$t1=0xDEADFADE$
 Ποιο είναι το αποτέλεσμα ( $\$t2$ ) των εντολών;
- `sll $t2, $t0, 4`  
`or $t2, $t2, $t1`
  - a.  $0x57755778$
  - b.  $0xFEFFFFEED$
- `srl $t2, $t0, 3`  
`andi $t2, $t2, 0xFFEF`
  - a.  $0x0000AAAA$
  - b.  $0x0000BFCD$

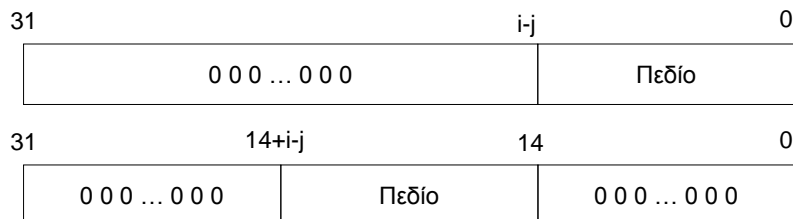
Εντολές: η γλώσσα του υπολογιστή — 5

## Άσκηση 5

- Καταχωρητής  $\$t0$



- Καταχωρητής  $\$t1$



Εντολές: η γλώσσα του υπολογιστή — 6

## Άσκηση 5 (συν.)

- Ακολουθία εντολών για την εξαγωγή του πεδίου από τον \$t0 και την μεταφορά του στον \$t1 για  $i=22$  &  $j=5$

```

add $t1, $t0, $0
srl $t1, $t1, 5
andi $t1, $t1, 0x0001ffff

```

```

add $t1, $t0, $0
sll $t1, $t1, 9
andi $t1, $t1, 0x7fffc000

```

- Ποιο είναι το λάθος στους παραπάνω κώδικες;
  - Η άμεση τιμή της εντολής andi δεν μπορεί να είναι μεγαλύτερη από 16 bit

Εντολές: η γλώσσα του υπολογιστή — 7

## Άσκηση 6

- Υλοποιήστε σε συμβολική γλώσσα MIPS τους παρακάτω κώδικες C:
  - Οι μεταβλητές a, b, c, d βρίσκονται στους καταχωρητές \$S0, \$S1, \$S2, \$S3

```

if (a < b)
    c = c + 1;
d = d + 1;

```

```

if (a > b)
    c = c + 1;
d = d + 1;

```

```

if (a <= b)
    c = c + 1;
d = d + 1;

```

```

if (a >= b)
    c = c + 1;
d = d + 1;

```

Εντολές: η γλώσσα του υπολογιστή — 8

## Άσκηση 6 (συν.)

- Μετάφραση ψευδοεντολών

<code>bge \$rt, \$rs, LABEL</code>	<code>slt \$t0, \$rt, \$rs</code> <code>beq \$t0, \$zero, LABEL</code>
<code>bgt \$rt, \$rs, LABEL</code>	<code>slt \$t0, \$rs, \$rt</code> <code>bne \$t0, \$zero, LABEL</code>
<code>ble \$rt, \$rs, LABEL</code>	<code>slt \$t0, \$rs, \$rt</code> <code>beq \$t0, \$zero, LABEL</code>
<code>blt \$rt, \$rs, LABEL</code>	<code>slt \$t0, \$rt, \$rs</code> <code>bne \$t0, \$zero, LABEL</code>

Εντολές: η γλώσσα του υπολογιστή — 9

## Άσκηση 6 (συν.)

Με χρήση των ψευδοεντολών

<code>if (a &lt; b)</code>	<code>bge \$s0, \$s1, fcond</code>
<code>c = c + 1;</code>	<code>addi \$s2, \$s2, 1</code>
<code>d = d + 1;</code>	<code>fcond: addi \$s3, \$s3, 1</code>
<code>if (a &lt;= b)</code>	<code>bgt \$s0, \$s1, fcond</code>
<code>c = c + 1;</code>	<code>addi \$s2, \$s2, 1</code>
<code>d = d + 1;</code>	<code>fcond: addi \$s3, \$s3, 1</code>
<code>if (a &gt; b)</code>	<code>ble \$s0, \$s1, fcond</code>
<code>c = c + 1;</code>	<code>addi \$s2, \$s2, 1</code>
<code>d = d + 1;</code>	<code>fcond: addi \$s3, \$s3, 1</code>
<code>if (a &gt;= b)</code>	<code>blt \$s0, \$s1, fcond</code>
<code>c = c + 1;</code>	<code>addi \$s2, \$s2, 1</code>
<code>d = d + 1;</code>	<code>fcond: addi \$s3, \$s3, 1</code>

## Άσκηση 7

- Υλοποιήστε σε συμβολική γλώσσα MIPS τον παρακάτω βρόχο:
  - $a = \$s0$ ,  $b = \$s1$ ,  $i = \$t0$
  - Ποιος είναι ο συνολικός αριθμός των εντολών;
- **for** ( $i=0$ ;  $i<10$ ;  $i++$ )  
 $a = a + b$ ;

```

addi $t0, $0, 0           # i = 0
LOOP: add $s0, $s0, $s1   # a = a + b
      addi $t0, $t0, 1    # i++
      slti $t2, $t0, 10   # if i < 10
      bne $t2, $0, LOOP   # go to Loop

```

Συνολικός αριθμός εντολών: 41

Εντολές: η γλώσσα του υπολογιστή — 11

## Άσκηση 8

- Υλοποιήστε σε συμβολική γλώσσα MIPS τον παρακάτω βρόχο:
  - $a = \$s0$ ,  $b = \$s1$ , base address  $D = \$s2$
  - Ποιος είναι ο συνολικός αριθμός των εντολών που εκτελούνται εάν  $a = 10$ ;

```

while (a < 10) {
    D[a] = b + a;
    a += 1;
}

```

```

LOOP:    slti $t2, $s0, 10           # if a >= 10
          beq  $t2, $0, DONE          # go to DONE
          add  $t3, $s1, $s0          # $t3 = b + a
          sll  $t2, $s0, 2            # $t2 = 4*a
          add  $t2, $s2, $t2          # διευθ. του D[a]
          sw   $t3, ($t2)             # D[a] = b + a
          addi $s0, $s0, 1            # a++
          j    LOOP                  # jump to Loop

```

DONE:

Συνολικός αριθμός εντολών: 2

Εντολές: η γλώσσα του υπολογιστή — 12

## Άσκηση 9

```

        addi $t1, $0, 100
LOOP: lw   $s1, 0($s0)
        add  $s2, $s2, $s1
        addi $s0, $s0, 4
        subi $t1, $t1, 1
        bne $t1, $0, LOOP

        addi $t1, $s0, 400
LOOP: lw   $s1, 0($s0)
        add  $s2, $s2, $s1
        lw   $s1, 4($s0)
        add  $s2, $s2, $s1
        addi $s0, $s0, 8
        bne $t1, $s0, LOOP

```

- Ποιος είναι ο συνολικός αριθμός των εντολών που εκτελούνται;
- Μεταφράστε τους παραπάνω βρόχους σε C.
  - Υποθέστε ότι \$t1 → variable i, \$s2 → result, \$s0 → διεύθυνσης βάσης του πίνακα MemArray
- Ξαναγράψτε τους βρόχους ώστε να μειώσετε τον αριθμό των εντολών.

Εντολές: η γλώσσα του υπολογιστή — 13

## Άσκηση 9 (συν.)

```

        addi $t1, $0, 100
LOOP: lw   $s1, 0($s0)
        add  $s2, $s2, $s1
        addi $s0, $s0, 4
        subi $t1, $t1, 1
        bne $t1, $0, LOOP

        addi $t1, $s0, 400
LOOP: lw   $s1, 0($s0)
        add  $s2, $s2, $s1
        lw   $s1, 4($s0)
        add  $s2, $s2, $s1
        addi $s0, $s0, 8
        bne $t1, $s0, LOOP

```

Συνολικός αριθμός εντολών: 501

Συνολικός αριθμός εντολών: 301

```

j=0;
for(i=100; i>0; i--){
result += MemArray[j];
j = j+1;
}

```

```

for(i=0; i<100; i+=2){
result += MemArray[i];
result += MemArray[i+1];
}

```

Εντολές: η γλώσσα του υπολογιστή — 14

## Άσκηση 9 (συν.)

```

    addi $t1, $0, 100
LOOP: lw  $s1, 0($s0)
    add  $s2, $s2, $s1
    addi $s0, $s0, 4
    subi $t1, $t1, 1
    bne $t1, $0, LOOP

```

```

    addi $t1, $s0, 400
LOOP: lw  $s1, 0($s0)
    add  $s2, $s2, $s1
    lw   $s1, 4($s0)
    add  $s2, $s2, $s1
    addi $s0, $s0, 8
    bne $t1, $s0, LOOP

```

Να χρησιμοποιήσουμε τον ίδιο καταχωρητή (\$s0) για τον δείκτη του βρόχου (i) και του πίνακα (s0) και να ελέγχουμε εάν φτάσαμε στο τελευταίο στοιχείο του πίνακα

Είναι ήδη βέλτιστος

```

    addi $t1, $s0, 400
LOOP: lw  $s1, 0($s0)
    add  $s2, $s2, $s1
    addi $s0, $s0, 4
    bne $s0, $t1, LOOP

```

Εντολές: η γλώσσα του υπολογιστή — 15

## Άσκηση 10

- Υλοποιήστε τον κώδικα σε συμβολική γλώσσα MIPS :

```

int compare(int a, int b){
    if (sub(a,b)>=0)
        return 1;
    else
        return 0;
}

int sub (int a,int b){
    return a-b;
}

```

```

compare:
    addi $sp, $sp, -4
    sw   $ra, 0($sp)
    jal  sub
    slt  $v0, $v0, $zero
    slti $v0, $v0, 1
    lw   $ra, 0($sp)
    addi $sp, $sp, 4
    jr   $ra

sub:    sub  $v0, $a0, $a1
    jr   $ra

```

Εντολές: η γλώσσα του υπολογιστή — 16