Data transfer instructions

load from and store to memory

lw, sw, lb και sd: χρησιμοποιούνται για τη φόρτωση/αποθήκευση δεδομένων μεταξύ μνήμης και καταχωρητών, κάτι που αποτελεί βασικό μέρος του MIPS.

Όνομα Εντολής	Περιγραφή	Μέγεθος	Μεταφορά
lw	φόρτωση λέξης	φόρτωση λέξης φόρτωση μιας λέξης 32-bit από τη μνήμη σε έναν καταχωρητή	4 bytes
sw	αποθήκευση λέξης	αποθήκευση λέξης 32-bit από έναν καταχωρητή στη μνήμη	4 bytes

Όνομα Εντολής	Περιγραφή	Μέγεθος	Μεταφορά
lb	φόρτωση Byte	Φόρτωση 1 byte από τη μνήμη	1 byte
sd	αποθήκευση Byte	μεταφέρει το μικρότερο byte δεδομένων από έναν καταχωρητή στην κύρια μνήμη	1 byte

1. lw - load word

∑ύνταξη: **lw $t0, offset($s1)**

Φορτώνει μια λέξη 32-bit από τη μνήμη στη διεύθυνση offset + $s1 στον καταχωρητή $t0. Η διεύθυνση μνήμης πρέπει να είναι ευθυγραμμισμένη με 4 bytes (δηλαδή, διαιρούμενη με το 4).

Παράδειγμα: lw $t0, 8($sp) # Φόρτωση λέξης από τη διεύθυνση ($sp + 8) στον $t0

2. sw - store word

∑ύνταξη: **sw $t0, offset($s1)**

Αποθηκεύει την τιμή 32-bit στο $t0 στη διεύθυνση μνήμης offset + $s1. Η διεύθυνση πρέπει να είναι ευθυγραμμισμένη με λέξεις (διαιρούμενη με το 4).

Παράδειγμα: sw $t0, 0($sp) # Αποθήκευση $t0 στη μνήμη στο ($sp + 0)

3. lb – load byte

Σύνταξη: **lb $t0, offset($s1)**

Παράδειγμα: lb $t0, 2($a0) # Φόρτωση byte από ($a0 + 2)

4. sb – store byte

Σύνταξη: **sb $t0, offset($s0)**
#Αποθήκευση του χαμηλότερου byte του $t0 στη μνήμη στη διεύθυνση ($s0 + offset).

example1: print number of memory

```
.data
number: .word 42          # Store the integer 42 in memory

.text
main:
        lw $a0, number    # Load the value from memory into $a0
        li $v0, 1         # syscall code 1 = print integer
        syscall           # print the integer in $a0

        # Exit program
        li $v0, 10
        syscall
```

example2:

```
data
myVar: .word 10        # Reserve a word in memory and initialize it to 10

.text
main:
        # Load the value from memory (myVar) into register $t0
        lw $t0, myVar          # $t0 = 10

        # Modify the value in $t0
        addi $t0, $t0, 5       # $t0 = $t0 + 5 = 15
        # Store the new value back into memory
        sw $t0, myVar          # myVar = 15

        # To demonstrate it's stored correctly, load it into another register
        lw $t1, myVar          # $t1 = 15

        # End of program (optional depending on environment)
        li $v0, 10       # Exit syscall
        syscall
```

example3: print char of memory

```
.data
char_val: .asciiz "A"        # Store character 'A' (null-terminated string)

.text
main:
        la $a0, char_val     # Load address of the character
        li $v0, 11           # syscall code 11 = print character
        lb $a0, 0($a0)       # load byte (the character) into $a0
        syscall              # print the character

        # Exit program
        li $v0, 10
        syscall
```

Βιβλιογραφεία: MIPS Assembly Language Programmer’s Guide (October 1992)
MIPS Assembly Language using QtSpim (Version 1.0 January 2013)