RI 201
Arhitektura računara

Auditorne vježbe 3

Load/Store & MIPS

- MIPS je load/store arhitektura
- Instrukcije koje rade sa memorijom spadaju u jednu od dvije kategorije
  - Load - Učitavanje podataka iz memorije
  - Store - Pisanje podataka u memoriju
- MIPS arhitektura podržava load/store u veličini
  - Byte
  - Half word
  - Word
- Pristup memoriji mora biti poravnat

Instrukcije za pristup memoriji

- Load instrukcije
  - `lb(u) dest, Imm(src)`
  - `lh(u) dest, Imm(src)`
  - `lw dest, Imm(src)`
- Store instrukcije
  - `sb dest, Imm(src)`
  - `sh dest, Imm(src)`

Primjer 1

Napisati MIPS assembly program koji sa memorijske lokacije ***x*** dohvata vrijednost, dijeli je sa 2 i smiješta na memorijsku lokaciju ***rez***. Memorijske lokacije ***x*** i ***rez*** definirati u data sekciji programa.

C kod:

```
int x = 100;
int rez = 0;
int main() {
    rez = x / 2;
    return 0;
}
```

Primjer 1 - riješenje

```
.section .data
x: .word 100
rez: .word 0

.section .text
.set noreorder
.global main
```

```
main:
    la $t0, x
    lw $t1, ($t0)
    sra $t1, $t1, 1
    la $t0, rez
    sw $t1, 0($t0)

    addi $v0, $zero, 0
    jr $ra
    nop
```

Primjer 2

Napisati MIPS assembly program koji sa memorijskih lokacija ***addr1*** i ***addr2*** učitava 8 bitne brojeve -5 i 245. Rezultat sabiranja ova dva broja smiješta na memorijsku lokaciju ***rez***.

C kod:

```
int8_t addr1 = -5;
uint8_t addr2 = 245;
int16_t rez = 0;
int main() {
  rez = addr1 + addr2;
  return 0;
}
```

Primjer 2 - riješenje

```
.section .data
addr1: .byte -5
addr2: .byte 245
rez: .hword 0

.section .text
.set noreorder
.global main
```

```
main:
    la $t0, addr1
    lb $t1, ($t0)
    la $t0, addr2
    lbu $t2, ($t0)
    add $t1, $t1, $t2
    la $t0, rez
    sh $t1, ($t0)

    addi $v0, $zero, 0
    jr $ra
    nop
```