

RI 201

Arhitektura računara

Auditorne vježbe 2



Fakultet elektrotehnike Univerziteta u Tuzli



Plan za vježbe

- Organizacija i pisanje programa na nivou MIPS ISA (asembli programiranje)
- Konverzija standardnog C koda u MIPS asembli kod
- Prevođenje MIPS asebli koda u MIPS mašinski kod
- Princip rada i komponente RISC procesora
- Implementacija MIPS procesora u Logisim simulatoru



Software

- Elk kompajlerski lanac:
 - clang kompajler
 - qemu simulator
 - gdb debugger
- Logisim
- <http://sourceforge.net/projects/circuit/>

Instrukcije

- Rad sa aritmetikom
 - add, addi, addu, addiu, sub, subu, lui, ori, ...
- Pristup memoriji
 - lb, sb, lh, sh, lw, sw, ...
- Promjena toka
 - bne, beq, j, jal, ...

Registri

- Sekvencijalni uređaji direktno ugrađeni u procesor
- Omogućavaju veoma brzo čitanje i pisanje
- MIPS arhitektura sadrži 32 registra sa po 32 bit-a
- Registri se označavaju brojevima \$0 ... \$31 ili imenima
- MIPS operacije se izvode samo na vrijednostima snimljenim u registrima

Format 1 zapisa instrukcija

- 1 2, 3, 4
- 1 → ime instrukcije
- 2 → destinacijski registar
- 3, 4 → operandi ili izvori
- `add $s0, $0, $s1` → `a = b+c` `a` → `$s0`, `b` → `$0`, `c` → `$s1`
- Registar `$0` ima specijalno značenje, sadrži vrijednost 0 i vrijednost mu se ne može promijeniti

Format 2 zapisa instrukcija

- 1 2, 3(4)
- 1 → ime instrukcije
- 3 → konstantna vrijednost (offset)
- 2, 4 → registri

lb \$s2, 20(\$a2)

\$a2 se naziva bazni registar,
20 predstavlja offset

Instrukcija uzima pointer snimljen u registar \$a2, na tu vrijednost dodaje broj 20, te sa memorijske adrese dobivene ovom sumom čita 1 byte podataka i upisuje u registar \$s2

sb \$s3, 24(\$a3)

\$a3 se naziva bazni registar 24 predstavlja offset

Instrukcija uzima pointer snimljen u registar \$a3, na tu vrijednost dodaje broj 24, te na memorijsku adresu dobivenu ovom sumom upisuje 1 byte iz registra \$s3, i to byte najniže vrijednosti

Zadatak 1

Napisati pomoću MIPS instrukcija sljedeću naredbu: $a = b + c - d$,

$a \rightarrow \$s3$,

$b \rightarrow \$s2$,

$c \rightarrow \$s1$,

$d \rightarrow \$s0$



Rješenje

.section .text

.set noreorder

.global main

main:

addi \$s2, \$0, 14

→ $b = 14$

addi \$s1, \$0, 25

→ $c = 25$

addi \$s0, \$0, 9

→ $d = 9$

add \$s3, \$s2, \$s1

→ $a = b + c$

sub \$s3, \$s3, \$s0

→ $a = a - d$

addi \$v0, \$0, 0

jr \$ra

nop

- Napomena: instrukcije koje rade sa konstantama kao što su addi, lui, ori, itd., kao vrijednost konstante mogu poprimiti broj sa maksimalno 16 bit-a

Zadatak 2

Izračunati vrijednost izraza:

- $(100-20)|5$



Fakultet elektrotehnike Univerziteta u Tuzli



Rješenje:

```
.section .text  
.set noreorder  
.global main  
main:  
addi $s0, $0, 100  
addi $s1, $0, 20  
sub $s2, $s0, $s1  
ori $s2, $s2, 5  
addi $v0, $0, 0  
jr $ra  
nop
```

-Koji bi bio drugi način za rješavanje datog problema?

Zadatak 3

- Izračunati vrijednost izraza:
- $100 + 40000 + 0xFFFFFFFF$
- Kako trebamo učitati vrijednost 40000 u registar?
- Kako upisati vrijednost 0xFFFFFFFF u registar? – Redoslijed instrukcija?
- Koju instrukciju trebamo koristiti za sabiranje?

Rješenje

```
.section .text  
.set noreorder  
.global main  
main:  
addi $s0, $0, 200  
ori $s1, $0, 40000  
lui $s3, 0xFFFF  
ori $s3, $s3, 0xFFFF  
add $s4, $s0, $s1  
add $s4, $s4, $s3  
addi $v0, $0, 0  
jr $ra  
nop
```