



Technische Informatik II im SS 2005

## 5. Übungsblatt

Abgabetermin: 02. Juni 2005, bis 13:00 Uhr

Prof. Dr. U. Brinkschulte

Geb. 40.28, D-76131 Karlsruhe

Email: brinks@ira.uka.de

Dr.-Ing. T. Asfour

Telefon: +49-721-608-7379

Fax: +49-721-608-8270

Email: asfour@ira.uka.de

<http://i61www.ira.uka.de/users/asfour/TI>

### Aufgabe 1

(6 Punkte)

- Bei der MIPS-Architektur können nur 16-Bit Operanden durch MIPS-Befehle in die Register geladen werden. Geben Sie eine Folge von MIPS-Befehlen an, mit welcher der 32-Bit Operand `1111 0000 0011 1101 0000 1001 0000 1001` ins Register `$s0` geladen werden kann. Dabei dürfen Sie den Pseudobefehl `li $s0, Imm` nicht verwenden.
- Welche Werte haben die 2 niedrigstwertigen Bits einer Wortadresse?
- Führen Sie den folgenden MIPS-Code aus und geben Sie die Änderungen in den Register- und Speicherinhalten an. Verwenden Sie die im Lösungsblatt angegebenen Tabellen.

```

addi $t3, $0, 0x20
lw   $t1, 0x08($t3)
add  $t4, $t3, $t0
sw   $t4, 0x10($t3)

```

Registersatz

Register	Inhalt
\$t0	0x10
\$t1	0x14
\$t2	0x16
\$t3	0x28
\$t4	0x1234

Hauptspeicher

Adresse	Inhalt
\$0x20	0x22
\$0x24	0x30
\$0x28	0x40
\$0x2C	0x50
\$0x30	0x60

### Aufgabe 2

(3 Punkte)

Welche Funktion hat das folgende MIPS-Programm:

```

.data
a:      .word 36, 20, 27, 15, 1, 62, 41
n:      .word 7
max:    .word 0

.text
.globl main

```

```

main:   li $t0, 0
        li $s0, 0
        lw $s1, n

m1:     bge $t0, $s1, m3
        mul $t1, $t0, 4
        lw $t2, a($t1)
        ble $t2, $s0, m2
        move $s0, $t2
m2:     addi $t0, $t0, 1
        b m1
m3:     move $a0, $s0
        li $v0, 1
        syscall
        li $v0, 10
        syscall

```

Aufgabe 3

(7 Punkte)

1. Gegeben ist das folgende MIPS-Programmstück:

```

anfang:   addi $t0, $zero, 0
          addi $t1, $zero, 1
schleife: slt  $t2, $a0, $t1      # if ($a0<$t1) then $t2=1 else $t2=0
          bne $t2, $zero, ende
          add  $t0, $t0, $t1
          addi $t1, $t1, 2
          j  schleife
ende:     add $v0, $t0, $zero

```

Das Register \$a0 ist mit einer positiven Integerzahl  $n$  initialisiert. Das Register \$v0 wird für die Ausgabe des Ergebnisses verwendet.

- i.) Welche Funktion erfüllt das angegebene Programmstück?
- ii.) Welche Werte stehen im Register \$v0 nach Abarbeitung des Programmstücks, wenn das Register \$a0 mit 9 bzw. mit 10 initialisiert wird.

2. Geben Sie die echten MIPS-Befehle bzw. Befehlsfolgen zu den folgenden MIPS-Pseudobefehlen an. Verwenden Sie das Register \$at zur Speicherung temporärer Variablen.

- `move $t5, $t3`           #  $\$t5 = \$t3$
- `clear $t5`               #  $\$t5 = 0$
- `bgt $t5, $t3, marke`   # if ( $\$t5 > \$t3$ ) goto marke
- `bge $t5, $t3, marke`   # if ( $\$t5 \geq \$t3$ ) goto marke

3. Beschreiben Sie die Funktion der folgenden MIPS-Befehle:

- i.) `lw $s1, 100($s2)`

- ii.) sw \$s1, 100(\$s2)
- iii.) jal mystery

Aufgabe 4

(6 Punkte)

Gegeben sei das folgende MIPS-Programmstück:

```
.data
vec: .word 8, 12, 16, 19, 2002, 0, 0, 0, 0, 0

.text
main: lw $t1, vec
      lw $t2, vec+4
      lw $t3, vec($t1)
      lw $t4, vec+4($t1)
```

1. Geben Sie die Inhalte der Register `$t1`, `$t2`, `$t3` und `$t4` in hexadezimaler Schreibweise nach der Ausführung des obigen Programmcodes an.
2. Geben Sie den MIPS-Code an, mit dem man die Adresse von `vec` im Register `$s0` speichert.
3. Schreiben Sie eine Programmschleife, welche die ersten fünf Elemente aus `vec` ausgibt. Die Elemente sollen dabei durch ein Leerzeichen getrennt sein.

Aufgabe 5

(6 Punkte)

Schreiben Sie ein MIPS-Assemblerprogramm, welches zwei Integer-Zahlen  $n$  und  $k$  einliest und  $n$  über  $k$  ( $n, k \in \mathbb{N}_0, n \geq k$ ) berechnet und ausgibt:

$$\binom{n}{k} := \frac{n!}{(n-k)! \cdot k!} = \frac{n \cdot (n-1) \cdot (n-2) \cdot \dots \cdot (n-k+1)}{k \cdot (k-1) \cdot (k-2) \cdot \dots \cdot 1}$$

Dazu werden Zähler und Nenner jeweils separat, aber in einer Schleife berechnet und anschließend die Division durchgeführt. Wenn zwei Zahlen mit  $n < k$  eingegeben werden, soll eine Fehlermeldung ausgegeben werden. Prüfen Sie auch, ob Ihr Programm  $\binom{0}{0} := 1$  richtig berechnet.