



Technische Informatik II im SS 2007

5. Übungsblatt

Abgabetermin: 31. Mai 2007, 13:00 Uhr

Prof. Dr. J. Henkel

Am Zirkel 2, Geb. 20.20
D-76131 Karlsruhe

Dr.-Ing. T. Asfour

Telefon: +49-721-608-7379

Fax: +49-721-608-8270

Email: asfour@ira.uka.de

<http://ti.ira.uka.de>

Aufgabe 1

(6 Punkte)

Schreiben Sie ein MIPS-Assemblerprogramm, welches die Fließkommazahlen A, B und C in doppelter Genauigkeit einliest und anschließend

- $A + B + C$
- $A - B - C$
- $(A * B) + C$
- $(A \div B)$

berechnet. Achten Sie darauf, dass bei der Division durch 0 eine geeignete Meldung ausgegeben wird.

Aufgabe 2

(6 Punkte)

1. 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.
2. Welche Werte haben die 2 niedrigstwertigen Bits einer Wortadresse?
3. Führen Sie den folgenden MIPS-Code aus und geben Sie die Änderungen in den Register- und Speicherinhalten an.

```

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 3

(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 4

(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 angegeben 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 >= $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`