

Rechnerorganisation im WS 2017/18

5. Übungsblatt

Abgabetermin: 11. Dezember, 13:15 Uhr

Prof. Dr. Wolfgang Karl
Haid-und-Neu-Str. 7

Dr.-Ing. Ömer Terlemez
Adenauerring 2, Geb. 50.20

Email: ti@ira.uka.de

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

Hinweise zur Bearbeitung der MIPS-Programmieraufgaben:

- Die Programmieraufgaben dürfen in einer Gruppe von bis zu drei Studenten (desselben Tutoriums) bearbeitet werden.
- Testen Sie Ihr Programm mit dem MIPS-Simulator MARS¹. Bei Fragen oder Problemen mit MARS können Sie sich an Ihren Tutor wenden.
- Programme, die aufgrund von Syntaxfehlern nicht ausführbar sind, werden mit null Punkten bewertet.
- Versehen Sie Ihr Programm mit einem Programmkopf (Kommentar), der die Namen und Matrikelnummern aller beteiligten Studenten enthält. Weiterhin muss Ihr Programm sinnvoll kommentiert sein. Fehlender Programmkopf oder Kommentare führen zu Punktabzug.
- Senden Sie Ihre Programmdatei per E-Mail an Ihren Tutor **und** geben Sie einen Ausdruck oder eine Abschrift mit Ihrer Ausarbeitung ab. Die E-Mail Adresse ihres Tutors finden Sie auf der TI-Homepage.
- Damit MARS mit der Ausführung des Programms bei der Marke `main` beginnt, muss die Option `Settings - Initialize Program Counter to global 'main' if defined` gesetzt sein.

¹Siehe <http://ti.ira.uka.de/TI-2/Mips/Mips.php>

Aufgabe 1

(6 Punkte)

1. Was versteht man unter einem Pseudobefehl? 2 P.

Warum muss es sich bei dem Befehl `li` (`load immediate`, lädt eine 32-Bit-Konstante in ein Register) um einen Pseudobefehl handeln?

2. Was versteht man unter einer Assemblerdirektive? 2 P.

Was bewirken die Assemblerdirektiven `.data` und `.float 20.12`?

3. Wie unterscheiden sich die Register `$t0-$t7` und die Register `$s0-$s7` in der Konvention ihrer Verwendung? 2 P.

Gibt es auch auf Hardwareebene Unterschiede zwischen diesen Registern? Falls ja, geben Sie diese Unterschiede an.

Aufgabe 2

(4 Punkte)

Folgendes Programm soll mit einem Assembler übersetzt werden, der keine Pseudobefehle des MIPS-Befehlssatzes unterstützt.

Ersetzen Sie hierzu alle Pseudobefehle durch nativ ausführbare Befehle.

Tip: In MARS kann die Unterstützung für Pseudobefehle deaktiviert werden (Settings - Permit extended (pseudo) instructions and formats).

```

        .text
        .globl main

main:   li $v0, 5           # Read number
        syscall
        move $s0, $v0

        neg $a0, $s0      # Output negated value
        li $v0, 1
        syscall

        abs $a0, $s0      # Output absolute value
        li $v0, 1
        syscall

        li $t0, 3         # Output tripled value
        mul $a0, $s0, $t0
        li $v0, 1
        syscall

        li $v0, 10        # Exit
        syscall

```

Aufgabe 3

(7 Punkte)

1. Gegeben ist das folgende MIPS-Programmstück:

2 P.

```
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.

2 P.

- `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:

3 P.

- i.) `lw $s1, 100($s2)`
- ii.) `sw $s1, 100($s2)`
- iii.) `jal mystery`

Aufgabe 4

(6 Punkte)

Eine vollkommene Zahl entspricht der Summe ihrer Teiler (die Zahl selbst ist ausgenommen).
Beispiel:

- 6 ist eine vollkommene Zahl, denn $1 + 2 + 3 = 6$
- 8 ist keine vollkommene Zahl, denn $1 + 2 + 4 \neq 8$
- 1 ist keine vollkommene Zahl, da sie nur durch sich selbst teilbar ist

Schreiben Sie ein Programm, das

- vom Benutzer eine Zahl einliest
- alle Teiler der eingelesenen Zahl ausgibt
- ausgibt, ob es sich um eine vollkommene Zahl handelt

Aufgabe 5

(6 Punkte)

Schreiben Sie ein MIPS-Assemblerprogramm, welches den Wert der Konstanten π in doppelter Genauigkeit mit der Bailey-Borwein-Plouffe-Formel² berechnet und ausgibt. Für eine hinreichende Genauigkeit genügt hierbei die Iteration bis $k = 10$.

$$\pi \approx \sum_{k=0}^{10} \frac{1}{16^k} \left(\frac{4}{8k+1} - \frac{2}{8k+4} - \frac{1}{8k+5} - \frac{1}{8k+6} \right) = 3.141592653589793$$

Die Ausgabe sollte wie folgt aussehen:

Pi: 3.141592653589793

Tipp: Der MIPS-Befehlssatz stellt keinen Befehl zur Potenzberechnung bereit. Überlegen Sie sich, wie Sie unter Verwendung einer Hilfsvariable (zusätzlich zum Schleifenzähler k) die explizite Berechnung von 16^k umgehen können.

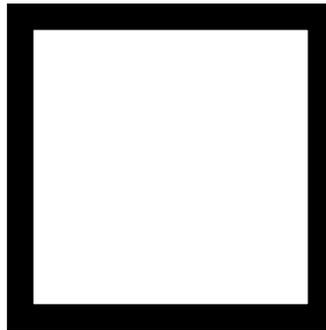
Abgabeort: Briefkasten im Untergeschoss im Informatikgebäude am Fasanengarten (Geb. 50.34)

²<http://de.wikipedia.org/wiki/Bailey-Borwein-Plouffe-Formel>

Vorlesung Rechnerorganisation Wintersemester 2017/18

- Übungsblatt 5 -

Tutoriumsnummer



Name, Vorname: _____

Matrikelnummer: _____

Studiengang: _____

Name des Tutors: _____

/29 Punkte