

Rechnerorganisation im WS 2020/21

## 5. Übungsblatt

Abgabetermin: 21. Dezember, 13:15 Uhr

Prof. Dr. Jörg Henkel  
Dr.-Ing. Lars Bauer  
Roman Lehmann, M. Sc.  
Haid-und-Neu-Str. 7,  
Geb. 07.21 (Technologiefabrik)  
Email: [roman.lehmann@kit.edu](mailto:roman.lehmann@kit.edu)

### Hinweise zur Bearbeitung der MIPS-Programmieraufgaben:

- Die Programmieraufgaben dürfen in einer Gruppe von bis zu drei Studierenden desselben Tutoriums bearbeitet werden.
- Testen Sie Ihr Programm mit dem MIPS-Simulator MARS<sup>1</sup>. 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.
- Versetzen 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.
- Erstellen Sie mit der Programmdatei und der PDF Ihrer Abgabe ein Zip-Archiv und laden Sie dieses als Abgabe im Übungsmodul des Ilias-Kurses hoch.
- 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.

---

<sup>1</sup>Siehe <http://courses.missouristate.edu/kenvollmar/mars/>

Aufgabe 1

(4 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. 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)

Bestimmen Sie das Befehlsformat der nachfolgend aufgelisteten MIPS-Befehle. Bei Pseudobefehlen kreuzen Sie unabhängig von der erzeugten Befehlssequenz bitte „Pseudobefehl“ an.

Richtige Antworten werden mit 0.5 Punkten bewertet, falsche mit −0.5 Punkten. Nicht ausgefüllte Felder werden nicht bewertet.

	R-Typ	I-Typ	J-Typ	Pseudobefehl
<code>add</code>				
<code>j</code>				
<code>beq</code>				
<code>nop</code>				
<code>ori</code>				
<code>subu</code>				
<code>negu</code>				
<code>lb</code>				

Aufgabe 3

(8 Punkte)

Eine sehr einfache Art der Textkodierung ist das ROT13-Verfahren. Dieses ersetzt jeden der 26 Buchstaben des lateinischen Alphabets (A-Z) durch den jeweils 13 davor bzw. danach liegenden Buchstaben.

Schreiben Sie ein MIPS-Programm, das eine Zeichenkette vom Benutzer einliest und diese ROT13-kodiert wieder ausgibt. Hierbei soll die Groß-/Kleinschreibung erhalten bleiben und alle anderen Zeichen, wie z.B. Ziffern, unverändert bestehen bleiben.

Verwenden Sie als Vorlage für Ihre Implementierung das Programmskelett in `blatt05_rot13.asm`.

**Tipp:** Verwenden Sie eine ASCII-Tabelle, um festzustellen, welche Zeichen Sie auf welche Weise verändern müssen.

Aufgabe 4

(11 Punkte)

Der Lagerroboter Kobi hat eine wichtige Datendisc verloren. Helfen Sie dem Roboter, diese im vorgegebenen Lagerraum wieder zu finden.

Verwenden Sie als Vorlage für Ihre Implementierung das Programmskelett in `blatt05_robot.asm`.

Der Roboter startet in dem im Programmskelett kodierten Lagerraum an der Position  $(x, y) = (1, 1)$ . Den Weg versperrende Wände sind durch das ASCII-Zeichen '#' kodiert. Die Position der Datendisc ist mit durch '0' kodiert.

Kobi kann<sup>2</sup> pro Zeitschritt ein Feld vorwärts fahren, eine Linksdrehung vollziehen, an seiner momentanen Position nach der Datendisc sehen sowie prüfen, ob sich in Blickrichtung auf dem Feld vor ihm eine Wand befindet.

1. Finden Sie die Markierung MYSTIC CODE im Code. Beschreiben Sie in kurzen Worten, welche Ihnen bekannten Kontrollstruktur höherer Programmiersprachen hier umgesetzt wird und wie dies auf Assemblerebene erreicht wird. 2 P.
2. Vervollständigen Sie an der markierten Stelle im Programmskelett die Subroutine `kobi_turnLeft`. 2 P.
3. Vervollständigen Sie an der markierten Stelle im Programmskelett die Subroutine `kobi_checkDisc`. 3 P.
4. Implementieren Sie in der `main`-Funktion den Suchalgorithmus nach der Rechte-Hand-Regel<sup>3</sup> wie im Kommentar im Programmskelett als Pseudo-Code vorgegeben. 4 P.

**Abgabeort:** Online im Ilias-Kurs in dem Übungsmodul Ihrer Tutoriumsgruppe

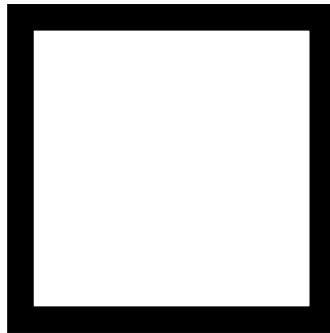
<sup>2</sup>In Anlehnung an [http://de.wikipedia.org/wiki/Niki\\_%E2%80%93\\_der\\_Roboter](http://de.wikipedia.org/wiki/Niki_%E2%80%93_der_Roboter)

<sup>3</sup>[http://en.wikipedia.org/wiki/Maze\\_solving\\_algorithm#Wall\\_follower](http://en.wikipedia.org/wiki/Maze_solving_algorithm#Wall_follower)

# Vorlesung Rechnerorganisation Wintersemester 2020/21

## - Übungsblatt 5 -

Tutoriumsnummer



Name, Vorname: \_\_\_\_\_

Matrikelnummer: \_\_\_\_\_

Studiengang: \_\_\_\_\_

Name des Tutors: \_\_\_\_\_

**/27 Punkte**