

## Rechnerstrukturen im WS 2013/2014 Übungsblatt 9 (Block C-1)

**Hinweis: Kommentieren Sie immer Ihre selbstgeschriebenen Assemblerprogramme !!!**

### Aufgabe 1 (Assemblerprogrammierung) (4 Punkte)

In einer Assemblerprozedur soll die Register-Transfer-Anweisung durchgeführt werden:

```
Speicher[0x10010000] := (Speicher[0x10010004] + Speicher[0x10010008])
                     * Speicher[0x1001000B] - Speicher[0x10010010] + 8
```

Vervollständigen Sie die nachfolgende Sequenz bis einschließlich Programmende. Die Verwendung von lw- und sw-Befehlen mit *Offset*-Werten, die nicht mit 16 Bit dargestellt werden können, ist nicht zulässig. Verwenden Sie möglichst wenige Register und möglichst wenige Befehle!

```
li    $2, 0x10010000
lw    $3, 0x4($2)
lw    $4, 0x8($2)
```

---



---



---



---



---



---



---



---



---



---

### Aufgabe 2 (Einfache Fallunterscheidung) (4 Punkte)

Ein MIPS-Programm soll bei der Auswertung der Klausurkorrekturen helfen. Es soll feststellen, ob jemand bestanden hat (Punkte  $\geq 40$ ) oder nicht (Punkte  $< 40$ ). Zunächst ist die Variable „Bestanden“ mit 99 belegt, was soviel bedeutet wie „die Note steht noch nicht fest“.

Ergänzen Sie das Programmfragment so, dass in „Bestanden“ eine 1 für bestanden und eine 0 für „nicht bestanden“ steht.

```
.data
Punkte:    .word 42          # Klausurpunkte
Bestanden: .word 99         # 0 Nein, 1 Ja, 99 weiß nicht
Grenze:    .word 40         # Bestehensgrenze 40 Punkte

.text
.globl main
```

**Aufgabe 3** (Adressierungsarten) (4 Punkte)

Gegeben sei eine abstrakte Maschine und folgender Auszug aus der Speicherbelegung einiger Register und Speicherzellen.

Register	Wert
Register 1	12
Register 2	36
Register 3	-12
Register 4	68

Speicherzelle	Wert
Speicherzelle 12	6
Speicherzelle 24	98
Speicherzelle 36	24
Speicherzelle 52	10
Speicherzelle 80	42
Speicherzelle 98	-5

Geben Sie jeweils an, welcher Wert jeweils durch folgende Adressierungsbefehle geladen wird. Die Adressierungsart ist jeweils vorgegeben und die Parameter (Konstanten / Adressen bzw. Register) sind in Klammern angegeben. Wie sieht der dazugehörige MIPS-Befehl aus? Das Ergebnis soll jeweils im Register \$t0 stehen. Die Register 1 bis 4 und die Speicherzellen 12 bis 98 seien schon entsprechend vorbelegt.

- a) Registeradressierung (Register 2)
- b) Unmittelbare Adressierung (24)
- c) Direkte Adressierung (52)
- d) Register-indirekte Adressierung (Register 1)

**Aufgabe 4** (Assemblerprogrammierung) (4 Punkte)

Vervollständigen Sie das gegebene Programmfragment der Art, dass nach Ablauf an der Stelle „ergebnis“ der gespiegelte Inhalt von „wert“ erscheint. D.h. dass die Bits in „ergebnis“ seitenverkehrt dargestellt werden. Der Inhalt von „wert“ darf nicht verändert werden. Sie sollten mit maximal 7 verschiedenen Registern auskommen.

```
.data
wert:      .word 0x0A080D03 # 0A080D03 Hex # umzuwandelnder 32 bit Wert
ergebnis:  .word 0          # Ergebnis wird C0B01050

.text
.globl main
main:     li    $2,32          # Laufindex i in R2
```

**Die Abgaben sollen bis Mittwoch den 18. Dezember 2013 um 18.00 Uhr in die Briefkästen in der Otto-Hahn-Strasse 20 eingeworfen werden. Bitte Name (bei einem 3er-Team alle), Matrikel- und Gruppennummer oben auf der ersten Seite der Lösungen angeben. Die Besprechung erfolgt in der 2. Januarwoche 2014.**