

## Übungen zur „Technischen Informatik I“, WS 2004/05

Nr. 11, Abgabe: Dienstag, 18. Januar vor der Vorlesung

---

### A. Hausaufgaben

55. Multiplikation im Ein-Befehlsrechner

3 Punkte

Schreiben Sie für den in Aufgabe 53 definierten Ein-Befehlsrechner SIC ein Programm zur Multiplikation zweier positiver Zahlen, die in `a` und `b` gespeichert sind. Das Produkt soll in `c` geschrieben werden. Die Inhalte von `a` und `b` dürfen modifiziert werden. Unter der Adresse `one` stehe der Wert 1 zur Verfügung.

56. Logische Instruktion `xor`

4 Punkte

(a) Geben Sie eine MIPS-Instruktionssequenz an, die die logische MIPS-Instruktion `xor reg1 reg2 reg3` simuliert, die im Register `reg1` bitweise das exklusive Oder der Inhalte der Register `reg2` und `reg3` berechnet.

(b) Geben Sie eine Folge von drei Instruktionen an, die die Inhalte zweier Register vertauscht, ohne dabei ein weiteres Register zu benutzen.

**Hinweis:** Verwenden Sie die in (a) definierte `xor`-Instruktion.

57. Stringverarbeitung

5 Punkte

Schreiben Sie eine MIPS-Assemblerprozedur `bfind`, die als einziges Argument in `$a0` die Adresse eines null-terminierten ASCII-Strings erhält. Die Prozedur sucht das erste Auftreten des Zeichens `b` in dem String und liefert in `$v0` dessen Adresse. Sollte die Zeichenkette kein `b` enthalten, so soll als Resultat die Adresse des abschließenden null-Zeichens zurückgegeben werden. Zum Beispiel soll für den String "imbibe" die Adresse des dritten Zeichens bestimmt werden.

---

### B. Mündliche Aufgaben

58. Branch on Overflow

Simulieren Sie mit MIPS-Instruktionen einen Branch-on-Overflow-Befehl für den vorzeichenlosen Fall einer Addition `$t0 = $t1 + $t2`. Verwenden Sie logische Operationen zum Test, ob `$t0 = $t1 + $t2 > 232 - 1`.

59. Analyse einer MIPS-Instruktionsfolge

Die folgende MIPS Instruktionsfolge implementiert einen neuen Befehl `new $s0 $s1`. Das Register `$t0` wird lediglich als Hilfsregister verwendet.

```
srl $s1, $s1, 1
sll $t0, $s0, 31
srl $s0, $s0, 1
or  $s1, $s1, $t0
```

Analysieren Sie die Sequenz und beschreiben Sie, was die neue Instruktion bewirkt.

60. Java-Strings

Java verwendet Unicode zur Zeichenkodierung. Dabei werden 16 Bits pro Zeichen verwendet. Die Speicherung von Zeichenketten (Strings) erfolgt ohne Nullterminierung. Stattdessen enthält das erste Wort die Länge der Zeichenkette. Entwickeln Sie eine MIPS-Assemblerprozedur zum Kopieren einer Java-Zeichenkette von einer Speicherposition `source` zu einer Speicherposition `sink`.