Lehrstuhl für Informatik III — Prof. Dr. M. Gössel	
Rechnerarchitektur (Sommersemester 2000)	
Übungen: M. Seuring, A. Dmitriev, P. Vogel	
Übungsblatt Nr. 5	9.5.2000
Abgabetermin: 16.5.2000	

Die Programme bzw. Programmfragmente bitte auf Diskette abgeben. Die Disketten gut beschriften (Name, Matrikelnummer, Übungsblattnr., wo programiert und getestet (Win/Linux). Die Disketten erhalten Sie mit den korrigierten Übungsblättern zurück. Eine ausgedruckte Version der Programmtexte den Lösungen beifügen.

Aufgaben zu den Transferbefehlen: Lade- und Speicherbefehle

`la Rdest, address` *Lade die Adresse*

Bei den folgenden Befehlen wird das Format des zu ladenden Datums vorgegeben.

`lb Rdest, address` *Lade Byte*

`lbu Rdest, address` *Lade vorzeichenloses Byte*

`ld Rdest, address` *Lade Double-Word*

`lh Rdest, address` *Lade Halfword*

`sw Rsrc, address` *Speichere Wort*

Speichere Inhalt des Registers `Rsrc` unter *address*.

`sb Rsrc, address` *Speichere Byte*

`sh Rsrc, address` *Speichere Halbwort*

`swl Rsrc, address` *Speichere Wort links*

Speichert die linken Bytes aus Register `Rsrc` an die möglicherweise unaligned *address*

`swr Rsrc, address` *Speichere Wort rechts*

Analog zu `swl`

Die anderen Lade- und Speicherbefehle finden Sie in der Befehlstabelle des Beiblatts. Sie sind in ihrer Bezeichnung selbsterklärend.

Aufgabe 17

Geben Sie das untenstehende Programm zum Retten der Register ein und testen Sie dieses schrittweise aus. Beschreiben Sie das Vorgehen im Programm. Testen Sie für verschiedene Belegungen der angegebenen Register. Nutzen Sie dabei die Möglichkeiten des SPIM-Simulators zur Veränderung der Registerbelegungen. Unter welcher Adresse wird *mem6* im Datensegment abgelegt? Welche anderen Möglichkeiten der Adressierung der Daten gibt es?

```
        .text       0x400000
start:  la          $2, mem2
        lw          $3, 0($2)
        lw          $4, 4($2)
        lw          $5, 8($2)
        lw          $6, 12($2)
        lw          $7, 16($2)
        lw          $8, 20($2)
        lw          $9, 24($2)
        sw          $3, 4($2)
        sw          $4, 8($2)
        sw          $5,12($2)
        sw          $6,16($2)
        sw          $7,20($2)
        sw          $8,24($2)
        sw          $9,28($2)
        nop
        nop
        nop
ende:   j           ende
        nop

        .data
mem2:   .word 4
mem3:   .word 8
mem4:   .word 16
mem5:   .word 32
mem6:   .word 64
mem7:   .word 128
mem8:   .word 256
mem9:   .word 512
```

Aufgabe 18

Schreiben Sie ein Programm, daß Daten über die Konsole einliest und nacheinander im Datenspeicher ablegt.

a) Es sollen fünf Integerwerte eingelesen werden.

b) Es sollen zwei Zeichenketten der Länge 20 eingelesen werden.

Verzweigungs- und Vergleichinstruktionen

`b label` *Sprunginstruktion*
Unbedigter Sprung zur markierten Instruktion.
`beq Rsrc1, Src2, label` *Sprung bei Gleichheit*
Bedingter Sprung, falls Inhalt von `Rsrc1` und `Src2` gleich sind.
`beqz Rsrc, label` *Sprung bei Null*
`bge Rsrc1, Src2, label` *Sprung bei Größer-gleich*
`bgt Rsrc1, Src2, label` *Sprung bei Größer-als*

`bgtz Rsrc1, label` *Sprung bei Größer-als Null*
`bltz Rsrc1, label` *Sprung bei Kleiner Null*
und weitere siehe Beiblatt.

Aufgabe 19

Bilden Sie folgende Kontrollstrukturen in Assemblercode nach:

a) Case-Anweisung: In Abhängigkeit von der Registerbelegung sollen unterschiedliche Ausgaben erfolgen.

Belegung Register $t0	Ausgabe auf Konsole
1	Erster Fall
2	Zweiter Fall
3	Dritter Fall
4	Vierter Fall
"sonst"	Nicht definiert

b) If-then-else-Anweisung: Bilden Sie folgende Anweisung nach:
IF $t0 = 27 THEN print(Register T0 = 27)
ELSE print(Register T0 ist ungleich 27)

c) For-Schleife: Bilden Sie folgende Anweisung nach:
k=0;
FOR i=1 UNTIL 27 STEP 2 DO
k=k+i
END;

Aufgabe 20

Schreiben Sie ein Assemblerprogramm für die Lösung folgender Aufgabe:

In Abhängigkeit vom Preis soll Rabatt gewährt werden. Über die Konsole wird der Preis erfragt und in Abhängigkeit davon soll der neue rabattierte Preis berechnet und ausgegeben werden. Ist der Preis < 10 DM wird kein Rabatt gewährt. Ist der Preis >= 10 und < 50 DM, dann werden 2,00 DM Rabatt gewährt, bei einem Preis >= 50 DM beträgt der Rabatt 5,00 DM. Wie sieht das Programm in einer höheren Programmiersprache aus? (Skizze genüegt)