



### 4. Übungsblatt

Abgabetermin: 19. Mai 2005, bis 13:00 Uhr

Prof. Dr. U. Brinkschulte

Geb. 40.28, D-76131 Karlsruhe

Email: brinks@ira.uka.de

Dr.-Ing. T. Asfour

Telefon: +49-721-608-7379

Fax: +49-721-608-8270

Email: asfour@ira.uka.de

http://i61www.ira.uka.de/users/asfour/TI

Zur Erleichterung der MIMA-Programmierung wird jetzt die im folgenden Beispiel dargestellte Assemblernotation eingeführt.

\* setzt den Ladepunkt für die folgenden Anweisungen

Marken definieren Bezeichner für die aktuelle Adresse

Konstanten kann ein Name zugeordnet werden

```

*          = $10
DS        = 1
DS        = -1
          = 100
DS        ;Zählervariable

```

Zahlkonstanten sind hexadezimal oder dezimal

DS (define storage) reserviert einen Speicherplatz und kann optional einen Anfangswert festlegen

```

*          = 128
;Hauptprogramm fuer eine Zählschleife
START     LDV  EINS      ;initialisiere ZAEHLER mit 1
          STV  ZAEHLER   ;entspricht STV $12 oder STV 18
SCHLEIFE LDV  ZAEHLER   ;erhoehe ZAEHLER um 1
          ADD  EINS
          STV  ZAEHLER   ;entspricht LDC 100 oder LDC $64
          LDC  MAX       ;falls MAX erreicht ist
          EQL  ZAEHLER   ;
          JMN  FERTIG    ;dann beende die Schleife
          JMP  SCHLEIFE  ;sonst nächster Schleifendurchlauf
FERTIG    HALT          ;HALT beendet das Programm

```

Die reservierte Marke START legt den Programmstartpunkt fest

Kommentare

#### Aufgabe 1

(10 Punkte)

Der Befehlssatz der MIMA wird um zwei Befehle mit indirekter Adressierung erweitert.

OpCode	Mnemonik	Beschreibung
A	LDIV a	«a» → Akku
B	STIV a	Akku → «a»

LDIV lädt den Wert an der Adresse <a> in den Akkumulator. STIV schreibt den Inhalt des Akkumulators an der Adresse <a>.

- Geben Sie die Mikroprogramme für die Execute-Phasen der Befehle LDIV und STIV an (jeweils ab dem 7. Takt).

2. Schreiben Sie ein MIMA-Programm, das ein Array von Zahlen sortiert. Die Zahlen stehen im Speicher ab Adresse 0. Der Einfachheit halber seien alle Werte im Bereich 0x00000 bis 0x7FFFF. Die Adresse des letzten Eintrags, d. h. die Array-Größe minus 1, stehe in der Speicherzelle 0x00040 bereit. Der Adressbereich 0x00080 bis 0x000FF steht für temporäre Variablen des Programms zur Verfügung. Das Programm soll bei Adresse 0x00100 beginnen. Nach Ablauf des Programms sollen die Zahlen in aufsteigender Reihenfolge, beginnend ab Adresse 0, im Speicher stehen.

**Hinweis:** Beginnen Sie mit der Sortierung beim letzten Eintrag und formulieren Sie die Bedingung für den Schleifenabbruch entsprechend.

### Aufgabe 2

(6 Punkte)

Schreiben Sie ein MIPS-Assemblerprogramm, welches die Fließkommazahlen A, B und C in doppelter Genauigkeit einliest und anschließend

- $A + B + C$
- $A - B - C$
- $(A * B) + C$
- $(A \div B)$

berechnet. Achten Sie darauf, dass bei der Division durch 0 eine geeignete Meldung ausgegeben wird.

### Aufgabe 3

(6 Punkte)

1. Schreiben Sie die folgenden Kontrollstrukturen in MIPS-Assembler um. Die Variablen i, j und k stehen in den Registern \$s3, \$s4 und \$s5.

i.)     if ( i==j )  
          k = i+j;

ii.)    if ( i!=j )  
          k = i+j;  
          else  
          k = i-j;

iii.)   if ( i < j )  
          k = 1;  
          else  
          k = 0;