

LEHRSTUHL FÜR INFORMATIK III — PROF. DR. M. GÖSSEL	
Rechnerarchitektur (Sommersemester 2000)	
Übungen: M. Seuring, A. Dmitriev, P. Vogel	
Übungsblatt Nr. 4	2.5.2000
Abgabetermin: 9.5.2000	

Wichtiger Hinweis

Die Vorlesung am 9. Mai fällt wegen dies academicus zum Leibniz-Kolleg aus. Zu der regulären Übungszeit am Donnerstag von 15.15 -16.45 Uhr findet am üblichen Ort (Großer Physikhörsaal) eine Vorlesung statt. Die Rechnerübungen am 10.5, 11.5. 13.30 -15.00 und 12.5. finden wie geplant statt. Die Übungsgruppe 4 (Donnerstag 15.15. - 16.45) bitte ich, die Freitagsgruppen mit zu nutzen.

Beachten Sie bitte die Einführung zum SPIM-Simulator des MIPS R2000 Prozessors und beantworten Sie folgende Fragen. Zur Beantwortung der Fragen müssen Sie mit dem SPIM-Simulator vertraut sein und mit ihm arbeiten können.

Aufgabe 13

Geben Sie das Programm `test01.s` (s.u.) mit dem Editor Ihrer Wahl ein und laden Sie dann das Programm in den SPIM-Simulator.

- a) Was ist im Speicher unter der Adresse `0x00400020` abgelegt?
- b) Was passiert im Programmspeicherbereich von Adresse `0x00400000` bis zum eigentlichen Beginn des geladenen Programms? (Erklärung, kurz)
- c) Setzen Sie den Wert von Register 20 auf 123 und den Wert von R21 auf 7. Setzen Sie einen Breakpoint auf die Adresse `0x0040002c` und einen zweiten auf die Adresse der Anweisung `1a $a0, msg3`. Arbeiten Sie mit **GO** bis zum ersten Break. Geben Sie an:
 - Die Adresse der Zeile `1a $a0, msg3`
 - Den Inhalt von Register R20 hexadezimal und dezimal
 - Den Inhalt von Register R21 hexadezimal und dezimal
 - Die Adresse, ab der die Zeichenkette `msg1` im Datenspeicher beginnt. Notieren Sie die Daten, die beginnend ab dieser Adresse bis zum Ende der ersten Zeichenkette im Datenspeicher abgelegt sind und dekodieren Sie diese mit Hilfe einer ASCII-Tabelle.
- d) Arbeiten Sie das Programm bis zum Ende ab. Betrachten Sie die Konsolenausgabe. Was ist das Ergebnis der Abarbeitung? Wo steht das Resultat?
- e) Was passiert, wenn die Zeile `.text` durch `.text 0x00400000` ersetzt wird?

```

#
#   Erstes      Testprogramm
#               test01.s
#
        .text
        .globl main
main:
        addu    $s6, $s4, $s5
        la     $a0, mesg1
        li     $v0,4
        syscall
        move   $a0, $s4
        li     $v0,1
        syscall
        la     $a0, mesg2
        li     $v0,4
        syscall
        move   $a0, $s5
        li     $v0,1
        syscall
        la     $a0, mesg3
        li     $v0,4
        syscall
        move   $a0, $s6
        li     $v0,1
        syscall
        li     $v0, 10
        syscall                                #exit

        .data
mesg1: .asciiz  "\nSie haben Register 20 belegt mit:"
mesg2: .asciiz  "\nSie haben Register 21 belegt mit:"
mesg3: .asciiz  "\nDas Ergebnis der Berechnung ist:"

```

Aufgabe 14

Geben Sie bitte den Inhalt des Registers $\$sp$ an, nachdem Sie den SPIM-Simulator aufgerufen und ein beliebiges Assemblerprogramm geladen haben.

Wie sieht zu diesem Zeitpunkt die Belegung des Stacks aus?

Schreiben Sie die Integer-Werte 1, 3, 11, 7 und 17 auf den Stack. Dazu können Sie ein kleines Assemblerprogrammfragment schreiben. Auf welche Speicherzelle zeigt nun der Stackpointer $\$sp$. Unter welchen Adressen sind die Werte im Stack gespeichert?

Aufgabe 15

Geben Sie das Programm mit dem Editor ein und arbeiten Sie es mit dem Simulator PC-Spim schrittweise ab, beobachten Sie dabei die Veränderungen in den Registern und protokollieren Sie diese. Verändern Sie das Programm, indem MEM1 durch MEM2 ersetzt wird. Wie verändern sich

die Ergebnisse?

```
        .text    0x400000
        .globl  main
main:    lui     $7, 0x55AA
        ori     $2, 0,0xAA55
        srl     $3,$2,16
        sll     $4,$2,16
        or      $7,$7,$2
        add     $2,$0,$7      # MOV $2,$7
        la     $5,MEM1
        lw     $6,0($5)
        sw     $2,4($5)
ENDE:   j      ENDE
        .data
MEM1:   .word  0x12345678
MEM2:   .word  0
```

Aufgabe 16

```
# Test des SLTUI
#
        .text
        .globl  main
main:   li      $16,9
        li      $17,0x0ffffff
        sltiu   $18,$16,20
        sltiu   $19,$16,-20
        sltiu   $20,$16,40000
        sltiu   $21,$17,0x9c40 # := +40000
        sltiu   $22,$16,100000
        sltiu   $23,$17,100000

        jr     $31
```

Testen Sie die Arbeitsweise des Kommandos *sltiu* mit obigem Programm. Protokollieren Sie die Veränderungen in den Registern. Was ist bei diesem Kommando zu beachten? Beschreiben Sie die Wirkungsweise des Programms (kurz).