

## Übungsblatt 8 (Block B - 4)

(16 Punkte)

Abgabe bis spätestens Mittwoch, 16. Dezember 2015, 16:00 Uhr  
Besprechung ab Montag, 4. Januar 2016

Um die Aufgaben zu lösen, sollten Sie den in der Vorlesung vorgestellten MARS Simulator installieren und mit ihm arbeiten. Sie finden die Software unter:

<http://courses.missouristate.edu/KenVollmar/MARS/index.htm>

Installieren Sie den Simulator auf Ihrem Rechner.

### 8.1 Simulatoroberfläche (4 Punkte)

Wählen Sie unter „File“ „New“ und geben Sie im Editorfeld das nachfolgende Assemblerprogramm ein. Speichern Sie ihre Eingabe als Datei. (z.B. Blatt8A1.asm)

```
.data                                # 01 die Zeilenzahl ist zum
x: .word 4711                        # 02 Besprechen in der Uebung
y: .word 10                          # 03
z: .word 0x0a92                      # 04
e: .word 0                           # 05 Ergebnisvariable
                                     # 06
.text                                 # 07
.globl main                          # 08
main:                                # 09
    lw  $2,x                          # 10
    lw  $3,y                          # 11
    lw  $4,z                          # 12
    add $2,$2,$3                      # 13
    sub $3,$2,$4                      # 14
    sw  $3,e                          # 15
    li  $2,10                         # 16 Programm ordnungsgemaess beenden
    syscall                           # 17
```

Lassen Sie das Programm assemblieren (F3). Ihr Programm erscheint nun unter „Execute“ im Textsegment. Es beginnt mit der Zeile: **lui \$1,0x1001** (unter „Basic“). Ihr Eingabetext steht unter „Source“. Lassen Sie das Programm schrittweise ablaufen (F7). Achten Sie bei jedem Schritt auf Veränderungen in den Registern 2 bis 4 (rechts unter Register „Number“ und „Value“).

- Nach der Abarbeitung Ihres Programms erscheint unter „Run I/O“ „– program is finished running –“. Welcher Wert steht im Register **\$3**? Geben Sie dieses Ergebnis zusätzlich als Dezimalzahl an.
- Was berechnet das Programm im Register Reg[3] (bezogen auf die Variablen x,y,z)? Geben Sie sowohl die Register-Transfer-Notation für die Berechnung als auch das berechnete Ergebnis an. (in der Register-Transfer-Notation soll x,y,z und das Ergebnisregister erscheinen)
- Die erste Programmzeile, die im Simulator angezeigt wird, lautet: **lui \$1,0x1001** (unter dem Reiter „Basic“). Warum fügt der Simulator diese Zeile ein und woher kommt die 0x1001?

- d. Wo (Segment in MARS) findet man nach Abarbeitung von Zeile 15 das Ergebnis im Speicher? Geben Sie den Bereich der Simulatoroberfläche und sowohl die genaue Speicheradresse als auch deren Inhalt im Format 0x..... an. Was bedeutet das Präfix 0x?

### 8.2 Laden von Konstanten (4 Punkte)

Mit dem MIPS-Pseudobefehl „**li** r, k“ kann eine 32-Bit Konstante k in ein Register r geladen werden. Je nach Wertebereich von k erzeugt der Assembler verschiedene Instruktionen für den Befehl **li** r, k. Diese verschiedenen Instruktionen sollen hier untersucht werden. Laden Sie dazu geeignete Konstanten k mit dem **li** Befehl. Geben Sie mit je einem Beispiel an, welche Anweisungen der Assembler (in der MARS-Umgebung unter „Execute“-> „Basic“) erzeugt. Verwenden Sie **\$3** für r und **\$1**, um Zwischenwerte zu speichern. Wählen Sie ein geeignetes k aus den folgenden Intervallen:

- a.  $-2^{15} \leq k \leq 2^{15}$  : 16 Bit Zweierkomplement (Pattern: 0x?000????)
- b.  $0 \leq k \leq 2^{16}$  : 16 Bit Betragszahl (Pattern: 0x0000????)
- c.  $k = k_1 \cdot 2^{16}$  mit  $2^{16} \leq k_1 \leq 2^{32}$ : 32 Bit Betragszahl (Tipp: 0x????0000)
- d.  $0 \leq k \leq 2^{32}$ : 32 Bit Betragszahl (Tipp:  $k = k_1 \cdot 2^{16} + k_2$  mit  $0 \leq k_1 \leq 2^{16}$  und  $0 \leq k_2 \leq 2^{16}$ ) (Pattern: 0x????????)

Das Fragezeichen (?) steht jeweils für eine hexadezimale Ziffer ({0 ... F})

### 8.3 Multiplikation (4 Punkte)

Schreiben Sie ein Assemblerprogramm zum Testen der Multiplikationsvarianten mit den Befehlen **mul**, **mult**, **multu** und **mulo** nach dem folgenden Schema: Legen Sie in zwei Speicherzellen (wert1 und wert2 im Bereich .data) die Konstanten +1 und -1 ab und multiplizieren Sie die Werte zunächst mit **mult** und dann mit **multu**. Des Weiteren berechnen Sie das Produkt der Zahlen 1030 und 4721471 mit den Befehlen **mul** und **mulo**.

- a. Geben Sie die Ergebnisse der Multiplikationen tabellarisch an (Register Hi und Lo und ggf. Reg[4]). Das Reg[4] soll dabei, wenn benötigt, als Zielregister benutzt werden. Schreiben Sie nicht einfach die internen (möglicherweise falschen) Werte der Registerinhalte des Simulators ab, sondern geben Sie das an, was ein fertiges kompiliertes Programm sinnvollerweise anzeigen würde.

Befehl	Hi	Lo	\$4
mult			
multu			
mul			
mulo			

- b. Welches Ergebnis liefern die Befehle mult und multu dezimal? Warum liefert der Befehl mulo kein Ergebnis?

### 8.4 Befehlssequenz (4 Punkte)

Gegeben sei folgende Sequenz von MIPS-Befehlen. Ergänzen Sie die folgende Tabelle: (Geben Sie ab der 2. Zeile nur Veränderungen an).

	Registerinhalte nach Ausführung des Befehls			
	\$2	\$3	\$4	\$lo
bei Programmstart	?	?	?	?
<b>li \$2, 0x05</b>				
<b>ori \$4, \$0, 48</b>				
<b>mul \$3, \$4, \$2</b>				
<b>mfhi \$3</b>				
<b>add \$2, \$2, \$4</b>				
<b>addi \$4, \$2, 0xAB39</b>				
<b>and \$3, \$4, 0x7F56</b>				
<b>ori \$3, \$3, 0x87BA</b>				

**Hinweise:**

Es wird empfohlen, im Hexsystem zu rechnen. Logische Immediate-Befehle nutzen die zero-extend-, nicht die sign-extend-Funktion. Sie sollten diese Aufgabe ohne Benutzung des Simulators lösen können.

**Hinweise:**

Die Abgaben sollen bis Mittwoch, 16. Dezember 2015, 16:00 Uhr in die Briefkästen in der Otto-Hahn-Straße 12 eingeworfen werden.

Die Briefkästen finden Sie in der ersten Etage der Otto-Hahn-Straße 12 am Übergang zum Erdgeschoss der Otto-Hahn-Straße 14. Die Briefkästen sind mit dem Namen der Veranstaltung, der Gruppennummer sowie Zeit und Ort der Übung gekennzeichnet.

Schreiben Sie unbedingt Ihren **Namen**, Ihre **Matrikelnummer** und Ihre **Gruppennummer rechts oben** auf Ihre Abgabe. Sie dürfen als Team mit bis zu zwei weiteren Personen abgeben. Geben Sie dann nur eine einzige Lösung ab und schreiben Sie alle Namen und Matrikelnummern des Teams auf die gemeinsame Abgabe.

Heften Sie die Abgabe zusammen. (Tacker oder notfalls Büroklammer). Falten Sie aber nicht ihre Abgabe. Stecken Sie die Abgabe nicht in einen Umschlag. Benutzen Sie den richtigen Briefkasten. Dazu benötigen Sie ihre Gruppennummer.

Es gibt insgesamt 12 Übungsblätter, die in 3 Blöcke (A, B, C) aufgeteilt sind. In jedem Block müssen Sie 30 Punkte von 64 möglichen Punkten erreichen, um zur Prüfung zugelassen zu werden.