

Technische Universität  
 Clausthal Institut für Informatik  
 Prof. G. Kemnitz

13. November 2018

### Rechnerarchitektur: Laborübung 3 (Arithmetik)

**Hinweise:** Schreiben Sie die Lösungen, so weit es möglich ist, auf die Aufgabenblätter. Tragen Sie Namen, Matrikelnummer und Studiengang in die nachfolgende Tabelle ein und schreiben Sie auf jedes zusätzlich abgegebene Blatt ihre Matrikelnummer. Lassen Sie für vorgeführte Experimente vom Betreuer die Punkte auf dem Aufgabenblatt eintragen und geben Sie, wenn Sie fertig sind, alle Blätter ab. Für eine Bescheinigung der erfolgreichen Teilnahme sind in jeder bis auf einer Laborübung mindestens 60% der Punkte zu erreichen.

Name	Matrikelnummer	Studiengang	Punkte von 20	≥ 60%

**Aufgabe 3.1:** Das nachfolgende Assemblerprogramm zur Addition von zwei 16-Bit-Zahlen

```
.global main
main:          ; Adresszuordnung
lds r18, 0x200 ; 0x200: a Byte 0
lds r19, 0x201 ; 0x201: a Byte 1
lds r20, 0x202 ; 0x202: b Byte 0
lds r21, 0x203 ; 0x203: b Byte 1
add r18, r20
add r19, r21
sts 0x200, r18
sts 0x201, r19
ret
```

soll folgendes C-Programm nachbilden:

```
#include <avr/io.h>

uint16_t a,b;
int main(){
  a += b;
}
```

Es enthält aber einen Fehler<sup>1</sup>.

- Suchen Sie den Fehler durch disassemblieren des übersetzten C-Programms und Vergleich beider Programme.
- Suchen Sie einen Test, mit dem der Fehler nachweisbar ist. Ein Test ist hier eine Wertepaar (a, b), bei dem das Assemblerprogramm einen anderen Wert für a berechnet als das C-Programm.

Der Betreuer kontrolliert:

- die vorgenommene Korrektur im Assemblerprogramm

2P

<sup>1</sup>Unterschiede, die keinen Einfluss auf die Funktion haben, wie eine andere Anordnung der Variablen im Speicher und die Nutzung anderer Register für Zwischenwerte sind keine Fehler.

- und das Testbeispiel.

2P

Bei Zweifel an der Richtigkeit ist vorzuführen, dass das korrigierte Programm für das Testbeispiel das richtige und das Originalprogramm ein falsches Ergebnis berechnet.

Punkte Aufgabe 3.1	
--------------------	--

**Aufgabe 3.2:** Gegeben sind die Bezeichner, Adressen und Anfangswerte der globalen Variablen und das disassemblierte Programm:

Name	Value	Type
<a href="#">a</a>		uint16_t(data)@0x0204
<a href="#">b</a>	0x3842	uint16_t(data)@0x0200
<a href="#">c</a>	0x5fa3	uint16_t(data)@0x0202

  

	r25	r24	r19	r18
0085 LDS R18,0x0200				
0087 LDS R19,0x0201				
0089 LDS R24,0x0202				
008B LDS R25,0x0203				
008D ADD R24,R18				
008E ADC R25,R19				
008F SUBI R24,0x0D				
0090 SBCI R25,0xFF				
0091 STS 0x0205,R25				
0093 STS 0x0204,R24				

a) Tragen Sie in der Tabelle neben dem disassemblierten Programm die Registerwerte nach Abarbeitung des Befehls links daneben ein und in die Tabelle mit den Bezeichnern und Adressen den Wert, der nach Abarbeitung aller Befehle in die Variable a gespeichert wird.

b) Schreiben Sie ein C-Programm, das dieselbe Funktion nachbildet.

Der Betreuer kontrolliert:

- für Teil a die acht zugewiesenen Registerwerte und das Ergebnis für Variable a. 4P
- Für Teil c das C-Programm, in Zweifelsfällen die korrekte Abarbeitung des Beispiels. 2P

Punkte Aufgabe 3.1	
--------------------	--

**Aufgabe 3.3:** Für das nachfolgende Multiplikationsprogramm sind zusätzlich die Adressen zu den Variablen und das disassemblierte Programm gegeben:

