Technische Universität Clausthal
Institut für Informatik
Prof. G. Kemnitz

01/14/2016

Laborübung 2 Rechnerarchitektur

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 vorgeführte Experimente vom Betreuer gegenzeichnen. 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	$\geq 40\%$

Aufgabe 1:

Gegeben ist folgendes C-Programm:

```
void main(void){

    DDRC  |= (1<<3) | (1<<5);
    PORTC |= (1<<3);

}
```

a) Welche Funktion hat das Programm? 2P

Antwort:

b) Untersuchen Sie, ob das Programm bei Übersetzung mit Complieroptimierung -O1 Bitsetzbefehle nutzt. Tragen Sie die disassemblierte Befehlsfolgen in die nachfolgende Tabelle ein. Schauen Sie bei Befehlen, die Sie nicht kennen, in »AVR Befehlssatz« unten auf der Web-Seite nach, was sie bewirken. Lösen Sie die Aufgabe mit dem Simulator. 2P

Übersetzung mit -O1		Übersetzung -O0	
Adresse	Befehl	Adresse	Befehl

c) Übersetzen Sie das Programm noch einmal mit Compileroptimierung -O0 und ergänzen Sie rechts in der Tabelle nur die Befehle, die die beiden EA-Register lesen, die Bitverknüpfung durchführen und die Wert zurückschreiben. (Was Z ist und wie da die richtigen EA-Adressen reinkommt, wird noch in der Vorlesung behandelt.) 2P

Nutzung von Bitsetzbefehlen:

Aufgabe 2:

Das nachfolgende Programm gibt auf PORTC am Ende von jedem Schleifendurchlauf einen Pseudo-Zufallswert aus:

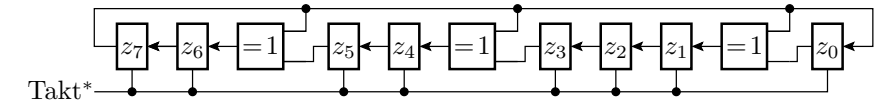
```
uint8_t z=0x6C;              // Startvektor
void main(){
 DDRC = 0xFF;
 while (1){
  if (z&0x80)
   z = (z<<1) ^ 0b01100011;
  else
   z = z << 1;
 PORTC = z;
 }
}
```

a) Testen Sie das Programm mit einem LED-Modul an JB (PORTC) im Debug-Modus mit einem Unterbrechungspunkt nach jeder Ausgabe. 2P

b) Die ausgegebene Wertefolge wiederholt sich zyklisch. Überlegen Sie sich eine Möglichkeit zur Bestimmung der Zykluslänge und bestimmen Sie diese[1]. 3P

Zykluslänge:	

c) Die nachfolgende Schaltung bildet fast die Funktion des Programms nach, hat aber einen kleinen Fehler. Tragen Sie in die nachfolgende Tabelle für den gegebenen Startzustand die Zustandsfolgen, die das Programm und die Schaltung durchlaufen, ein. Ändern Sie das Programm oben so, das seine Funktion mit der der Schaltung übereinstimmt (Änderungen oben im Programmtext eintragen). 3P



Schritt	Programm								Schaltung							
	z_7	z_6	z_5	z_4	z_3	z_2	z_1	z_0	z_7	z_6	z_5	z_4	z_3	z_2	z_1	z_0
1	0	1	1	0	1	1	0	0	0	1	1	0	1	1	0	0
2																
3																
4																
5																

$*$ die Register z_0 bis z_7 schalten in jedem Takt einen Schritt weiter

[1] Man kann z.B. einen Zähler für die Anzahl der Schleifendurchläufe in das Programm einfügen und die Schleifenabbruchbedingung so ändern, dass nur genau ein Zyklus durchlaufen wird.

d) Stecken Sie an PORTA obere Reihe das Modul mit den Tastern und erweitern Sie das Programm aus Aufgabenteil a so, dass es zu Beginn von jedem Schleifendurchlauf erst auf die Betätigung und dann auf das Loslassen von Taster 1 wartet. 2P

Aufgabe 3:

Das nachfolgende Assemblerprogramm zur Addition von zwei 16-Bit-Zahlen

```
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:

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

Es enthält aber einen Fehler[2].

a) Suchen Sie den Fehler durch dissasemblieren des übersetzten C-Programms und Vergleich beider Programme. 2P

Unterschiede beider Programm:

b) 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. 2P

Tests für den Fehlernachweis:

[2] 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.