

Technische Universität
 Clausthal Institut für Informatik
 Prof. G. Kemnitz

3. Januar 2017

Rechnerarchitektur: Laborübung 2 (LS)

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 2.1: Gegeben ist das nachfolgende C-Programm:

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

uint8_t a, b;
int main(void){
    DDRJ = 0xFF;
    while(1){
        a = PINA;
        b = a << 3;
        PORTJ = b;
    }
}
```

Übersetzen Sie das Programm mit »-O0«. Stecken Sie wie bei der 1. Laborübung an Port A oben das Schaltermodul und starten Sie das Programm im Debugger. Öffnen Sie zusätzlich folgende Fenster

- I/O (Debug > Windows > I/O) zur Beobachtung der Ports A und J,
- Watch 1 (Debug > Windows > Watch > Watch 1) zur Beobachtung der Variablen a und b,
- Disassembly (Debug > Windows > Disassembly) zur Visualisierung des Maschinenprogramms.

a) Lesen Sie aus den Beobachtungsfenstern die Adressen der genutzten Variablen und Register ab:

Datenobjekt	DDRJ	PINA	PORTJ	a	b
Adresse					

b) Schauen Sie im disassemblierten Programm nach, welche Befehle der Compiler zum Lesen und Schreiben der Register und Variablen verwendet und kennzeichnen Sie, ob direkte (D) oder indirekte (I) Adressierung verwendet wird:

Datenobjekt	DDRJ	PINA	PORTJ	a	b
Lesebefehl	—		—		
Schreibbefehl		—			
Adressierung (D/I)					

- c) In welche Befehlsfolge wird die Anweisung »b = a << 3;« übersetzt? Ergänzen Sie im nachfolgenden Listing zusätzlich das Zuweisungsziel (Register, Speicheradresse) sowie den zugewiesenen Wert für die Schaltereingabe 0b1011 und kontrollieren Sie das Ergebnis im Schrittbetrieb.

```

;Assembleranweisung / Zuweisungsziel / zugewiesener Wert
;                   /                   / 0b00001011
;                   /
;                   /
;                   /
;                   /
;                   /

```

- d) Erhöhen Sie die Verschiebung auf 5 »b = a << 5;« und untersuchen Sie in derselben Weise, wie der Compiler diese Anweisung übersetzt sowie das Testbeispiel mit derselben Schaltereingabe 0b1011:

```

;Assembleranweisung / Zuweisungsziel / zugewiesener Wert
;                   /                   / 0b00001011
;                   /
;                   /
;                   /
;                   /
;                   /

```

Dem Betreuer sind vorzuführen:

- vier richtig ausgefüllte Tabellen 8P
- Vorführung der Abarbeitung eines Schleifendurchlaufs des Assemblerprogramms im Schrittbetrieb mit einer vom Betreuer vorgegebenen Schaltereingabe und Verschiebung. 2P

Punkte Aufgabe 2.1	
--------------------	--

Aufgabe 2.2: Das nachfolgende Programm gibt auf Port J am Ende von jedem Schleifendurchlauf einen Pseudo-Zufallswert aus:

```

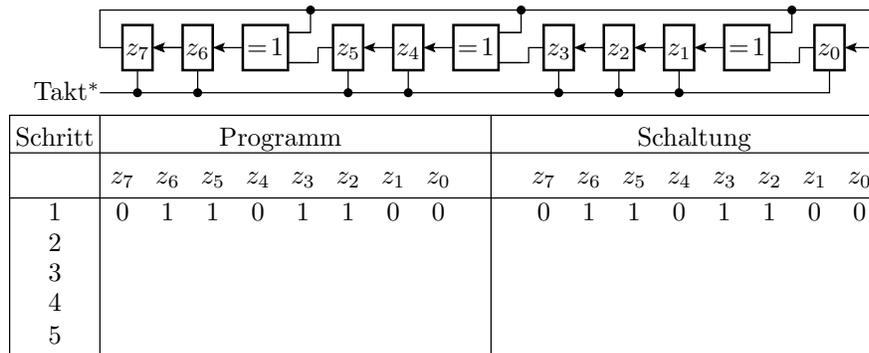
#include <avr/io.h>

uint8_t z=0x6C;
void main(){
  DDRJ = 0xFF;
  while (1){
    if (z&0x80)
      z = (z<<1) ^ 0b01100011;
    else
      z = z << 1;
    PORTJ = z;
  }
}

```

- a) Testen Sie das Programm mit einem Unterbrechungspunkt nach der Ausgabe an Port J und notieren Sie in der Tabelle unten die ersten vier Folgewerte.

- b) Die ausgegebene Wertefolge wiederholt sich zyklisch. Bestimmen Sie die Zykluslänge. Erweitern Sie dazu das Programm um einen 16-Bit-Zähler für die Anzahl der Schleifendurchläufe und die Möglichkeit, einen Unterbrechungspunkt so zu setzen, dass es immer, wenn an Port J der Startwert $z=0x6C$ zugewiesen wird, anhält.
- c) Die nachfolgende Schaltung bildet fast die Funktion des Programms nach, hat aber einen kleinen Fehler. Tragen Sie in die nachfolgende Tabelle zusätzlich die Folgezustände der Schaltung ein. Ändern Sie das Programm oben so, das seine Funktion mit der der Schaltung übereinstimmt (Änderungen oben im Programmtext eintragen).



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

- d) Stecken Sie an Port A obere Reihe das Modul mit den Tastern und erweitern Sie das Programm aus Aufgabenteil a so, dass es nach jedem Schleifendurchlauf erst dann weiterläuft, nachdem ein Taster gedrückt und wieder losgelassen wurde.

Dem Betreuer sind vorzuführen:

- Tabelle mit je vier Folgezuständen zu Aufgabenteil a und c. 4P
- Vorführung der Bestimmung der Zykluslänge zu Aufgabenteil b. 2P
- Vorführung des Programms zu Aufgabenteil d incl. der Änderung aus Aufgabenteil c. 4P

Punkte Aufgabe 2.2	
--------------------	--