

Hochschule Bremerhaven

Fachbereich II – Management und Informationssysteme

Informatik

Modul Rechnerarchitektur

Protokoll zu Aufgabenblatt Nr.07

von

Fabian Tober Matrikel-Nr. 40607

Dipok Chandra Roy Matrikel-Nr. 40783

24.05.2024

Inhaltsverzeichnis

1	Einleitung	3
1.1	Material und Methode	3
1.2	Gegeben	3
1.3	Gesucht	3
1.4	Geplante Durchführung und Erwartung	3
2	Aufgabe 1 - Takte und Assambler	3
2.1	Aufgabe 1.1 - Takte und Assambler P ₁	3
2.2	Aufgabe 1.2 - Takte und Assambler P ₂	4
3	Aufgabe 2 - Programmiercode einbetten	5
3.1	Aufgabe 2.1 - Code ausführen	5
3.2	Aufgabe 2.2 - Oszilloskop Messung und vergleichen	9
4	Aufgabe 3 - Funktion einbetten	10
4.1	Aufgabe 3.1 - Planung	10
4.2	Aufgabe 3.2 - Einbetten, Simulieren, Messen	10
5	Aufgabe 4 - Arithmetik und Sprünge	10
5.1	Aufgabe 4.1 - Planung	10
5.2	Aufgabe 4.2 - Einbetten, Simulieren, Messen	10
6	Ergebnis	10

1 Einleitung

Dieses Versuchsprotokoll dient der Dokumentation der Bearbeitung von Aufgabenblatt07, welches sich mit dem Arbeiten von Assamblercode beschäftigt. Im Folgenden werden die Aufgaben abgearbeitet.

1.1 Material und Methode

Zur Verfügung stehen uns Laptops (oder Laborrechner), ein Server (Hopper), sowie das Internet. Die Vorlesungsfolien sind natürlich auch jederzeit einsehbar und dienen als Hilfsmittel. Das Ziel erreichen wir ebenfalls nur durch Nutzung unserer gegebenen Materialien.

1.2 Gegeben

Wir können unser eigenes Wissen und/oder das Internet benutzen, um die Aufgaben abzuschließen. Außerdem haben wir noch die Vorlesungsfolien. Diesesmal verwenden wir auch noch ein uns gegebenes Mustercode.

1.3 Gesucht

Unser Ziel besteht darin, den Aufgabenzettel 07 vollständig zu bearbeiten, indem wir alle Fragen umfassend beantworten.

1.4 Geplante Durchführung und Erwartung

Um die Aufgaben zu lösen, haben wir uns die Methoden von der Vorlesung zuvor und/oder dem Internet zu nutzen gemacht. Wir gehen davon aus, dass die Aufgaben erfolgreich bearbeitet werden.

2 Aufgabe 1 - Takte und Assamblere

In Aufgabe 1 sollen wir die Taktrate berechnen und dann als Assamblerecode schreiben.

2.1 Aufgabe 1.1 - Takte und Assamblere P_1

Hier sollen wir nun die Takte für P_1 pro Durchlauf bestimmen und daraus eine Assamblere schleife bilden. Zuerst einmal die beiden Werte die wir von Aufgabe04 haben $S=1$; $T=39$ und folgendes richtig zu berechnen:

$$P_1 = (282823 \bmod(4T+5S)) + 128$$

$$P_1 = (282823 \bmod(4*39+5*1)) + 128$$

$$P_1 = (282823 \bmod 161)+ 128$$

$$P_1 = 107 + 128$$

$$P_1 = 235$$

Damit sind es bei $P_1 = 235$ Takte.

Nun machen wir eine Schleife die genau auf 235 Takte kommt:

Listing 1: Assamblercode

```

1  block_A_start:  <- 0Takte
2  ;setze R16=0    <- 1Takt
3
4  marke:         <- 0Takte
5  inc R16        <- 1Takt
6  cpi R16,47     <- 1Takt
7  nop            <- 1Takt
8  brne marke     <- 2Takte*46+1Takt
9  block_A_ende:  <- 0Takte

```

2.2 Aufgabe 1.2 - Takte und Assambler P_2

Hier sollen wir nun die Takten für P_2 pro Durchlauf bestimmen und daraus eine Assambler schleife bilden. Zuerst einmal die beiden werte die wir von Aufgabe04 haben $S=1$; $T=39$ un folgendes richtig zu berechnen:

$$P_2 = (728441 \bmod(6T+7S)) + 8192$$

$$P_2 = (728441 \bmod(6*39+7*1)) + 8192$$

$$P_2 = (728441 \bmod 241)+ 8192$$

$$P_2 = 139 + 8192$$

$$P_2 = 8331$$

Damit sind es bei $P_2 = 8331$ Takte.

Nun machen wir eine Schleife die genau auf 8331 Takte kommt:

Listing 2: Assamblercode

```
1 block_A_start:
2     ldi R17, 35                <- 1 Takt
3
4 aeußere_schleife:
5     ldi R16, 0                <- 1 Takt
6
7 marke:
8     inc R16                   <- 1 Takt
9     cpi R16, 47              <- 1 Takt
10    nop                       <- 1 Takt
11    brne marke                <- 2 Takte*46+1Takt
12
13    dec R17                   <- 1 Takt
14    brne aeußere_schleife    <- 2 Takte*34+1Takt
15 block_A_ende:
16    nop                       <- 1 Takt
17
```

3 Aufgabe 2 - Programmiercode einbetten

Hier wird nun der Code eingebettet und ausgeführt, danach nachgemessen und geschaut wo die Unterschiede sind.

3.1 Aufgabe 2.1 - Code ausführen

Wir haben den Code von P_1 genommen und gemäß der Aufgabe erweitert. Dadurch ist folgender code entstanden:

Listing 3: Assamblercode

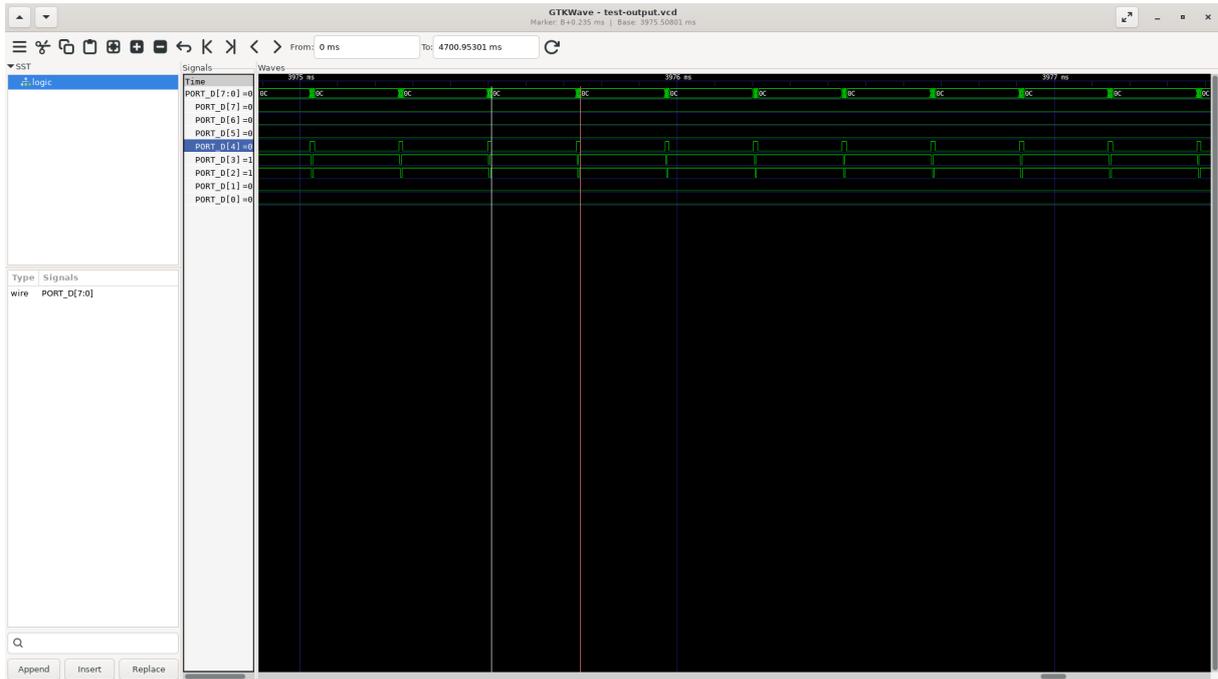
```
1      block_A_start:
2
3      clr r16
4      marke:
5      inc r16
6      cpi r16,55
7      brne marke
8
9      block_A_ende:
10
11     ;Folgendes ist von der Aufgabe ergänzt worden:
12     sbi PORTD,PD4
13     cbi PORTD,PD3
14     cbi PORTD,PD2
15     sbi PORTD,PD3
16     sbi PORTD,PD2
17     nop
18     cbi PORTD,PD4
19     rjmp block_A_start
```

Wir haben nun den Code von P₂ genommen und gemäß der Aufgabe erweitert. Dadurch ist folgender code entstanden:

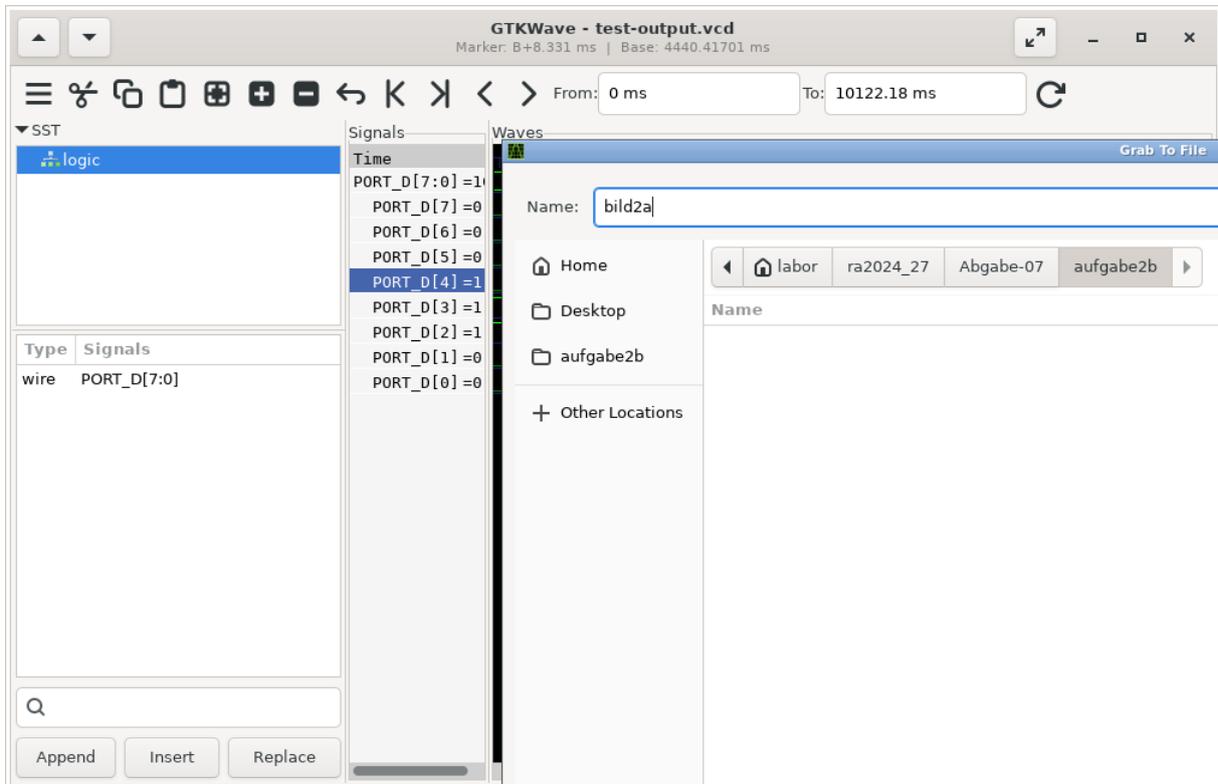
Listing 4: Assamblercode

```
1      unnotige_block_start:
2      ldi R18,0
3      marke1:
4          inc R18
5          cpi R18,7
6          brne marke1
7      unnotige_bolock_end:
8
9      block_B_start:
10         ldi R17, 40
11
12     out_schleife:
13         ldi R16, 0
14
15     marke:
16         inc R16
17         cpi R16, 51
18         brne marke
19         dec R17
20         brne out_schleife
21     block_B_ende:
22
23     ;Folgendes ist von der Aufgabe ergänzt worden:
24     sbi PORTD,PD4
25     cbi PORTD,PD3
26     cbi PORTD,PD2
27     sbi PORTD,PD3
28     sbi PORTD,PD2
29     nop
30     cbi PORTD,PD4
31     rjmp main
32
```

Folgend noch ein Bild aus GTKWave vom P₁ code:



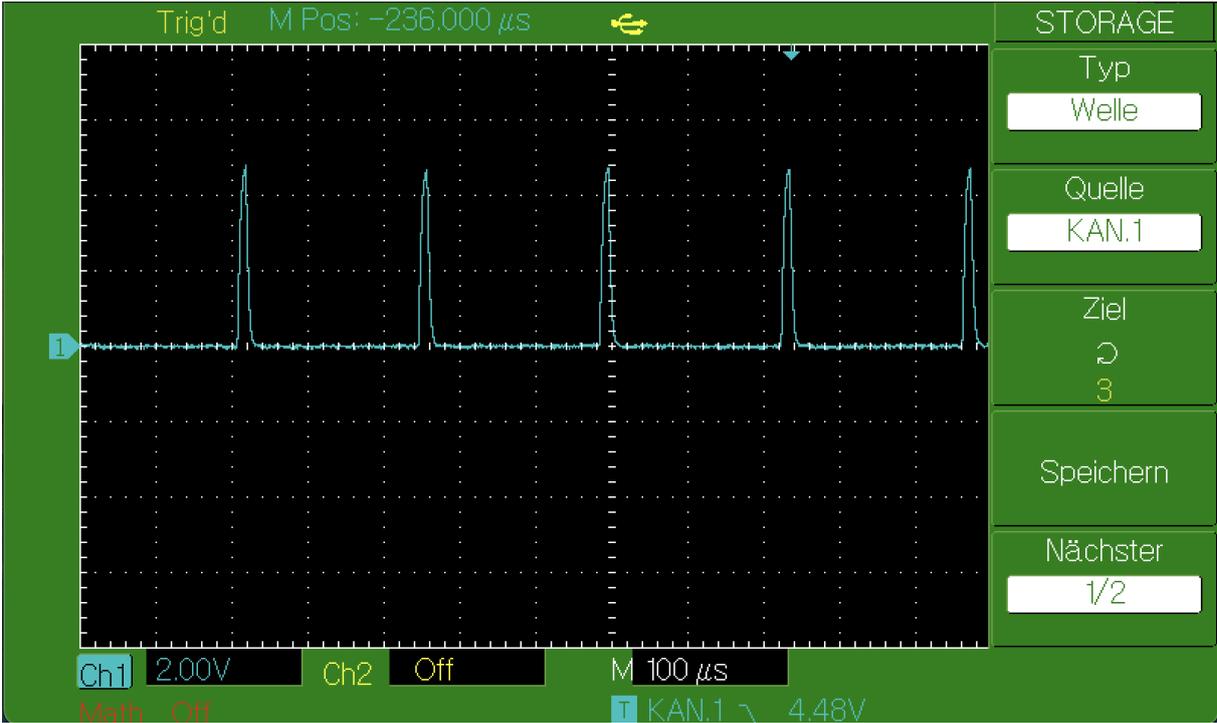
Folgend noch ein Bild aus GTKWave vom P₂ code:



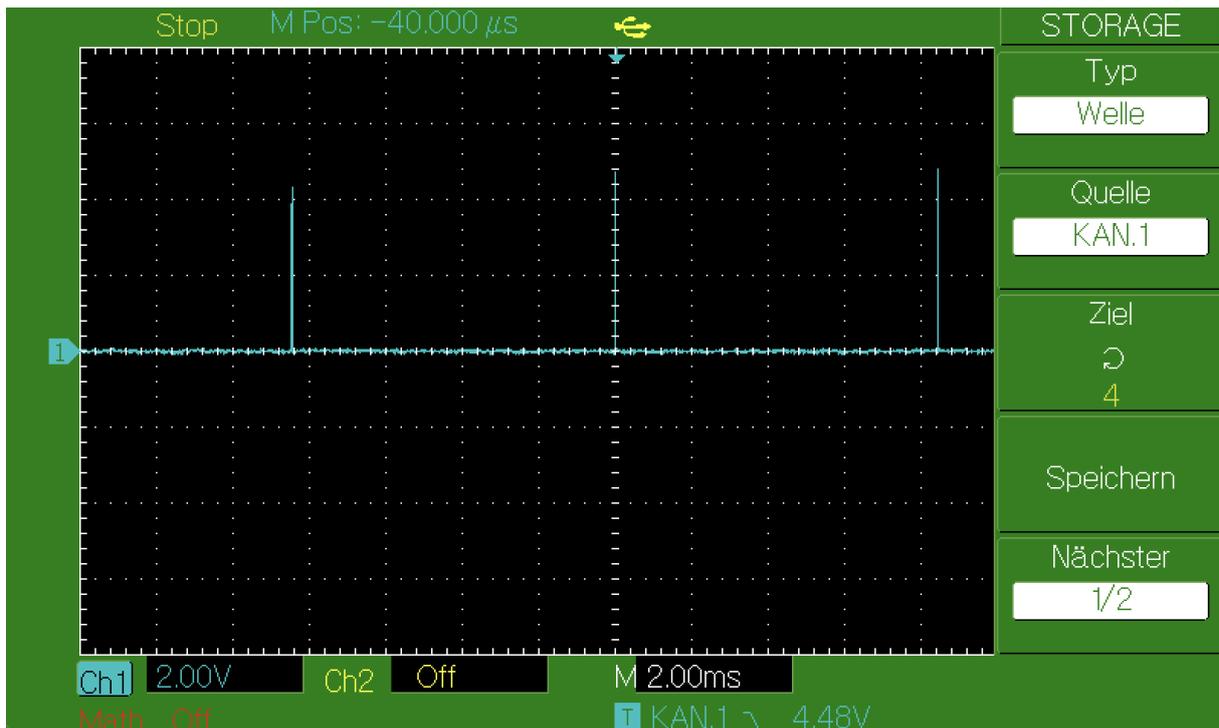
3.2 Aufgabe 2.2 - Oszilloskop Messung und vergleichen

Nachdem wir den Code mit der richtigen Taktzahl mittels GTKWave ermitteln konnten, haben wir eine Messung über das Oszilloskop gemacht. Hier die Ergebnisse von P₁ und P₂.

Folgende Bilder sind von P₁:



Nun folgen die Bilder von P₂:



Hierbei stellen wir fest das ...

4 Aufgabe 3 - Funktion einbetten

In Aufgabe 3 sollen wir für jeden Code-Abschnitt eine Funktion erstellen (Befehle aufrufen und beenden).

4.1 Aufgabe 3.1 - Planung

4.2 Aufgabe 3.2 - Einbetten, Simulieren, Messen

5 Aufgabe 4 - Arithmetik und Sprünge

Hier sollen wir `cp` und `cpi` entfernen und durch Zero- und Carrywerte ersetzen. Dabei sollen wir `adc` und `subc` benutzen (in der Verschachtelten Schleife).

5.1 Aufgabe 4.1 - Planung

5.2 Aufgabe 4.2 - Einbetten, Simulieren, Messen

6 Ergebnis

Die Aufgaben wurden alle bearbeitet. Damit ist das Ergebnis Positiv.