

Praktikum 1 – Einstieg und Befehle – Hilfestellung

Vorbereitung

1. Wie ist ein Assemblerbefehl aufgebaut? Erläutern Sie die Begriffe: Maschinenbefehl, Op-Code, Daten, Konstante, Variable, Quelloperand (Source), Zieloperand (Destination).
 - „Befehlsliste des 8086“, S. 2 und S. 14 (Registersatz, Abkürzungen, Befehlsliste und Adressierungsarten)
 - „Anleitung – Programmieren mit dem 8086“, S. 3 - 5
2. Wie funktionieren folgende Befehle prinzipiell [...]
 - siehe PDF „Befehlsliste des 8086“, Kapitel Befehlsliste, S. 2 – 11
3. Informieren Sie sich, wie folgende Baugruppen des SBC86 verwendet werden:
 - a. Abfrage der Schalterstellung: siehe Beispielprogramm Zeile 3

```
org 100h
; -----
start: in al,0      ; Schalterstellung einlesen
out 0,al           ; und auf die LED-Zeile ausgeben
mov cx,-1          ; Zaehler für Zeitschleife laden
schl1: loop schl1   ; loop: zwei Befehle in einem
                  ; Schritt1: Dekrementiere <CX>
                  ; Schritt2: Sprung, wenn <CX> ungleich 0

mov al,0
out 00,al          ; alle LED „ausschalten“
mov cx,-1
schl2: loop schl2
jmp start          ; Endlosschleife
; -----
```

start:	in	al,	0	;Schalterstellung einlesen
[label]	[Befehl]	[Ziel]	[Quelle]	[; Kommentar]

Das **Label** start wird am Ende des Programms als Sprungpunkt zum Erzeugen einer Endlosschleife genutzt (jmp start).

Der **Befehl** in nimmt als Argumente das Ziel (al) und die Quelle (0). Er liest die Daten aus der Quelle ein und legt sie im Zielregister ab.

Als **Ziel** wird das Register al, ein allgemeines 8-Bit-Register und Teil des Akkumulators (siehe *Befehlssatz des 8086*, S.2) genutzt. Hier werden die eingelesenen Daten für die spätere Nutzung zwischengelagert.

Als **Quelle** wird in diesem Fall 0 mitgegeben, da sich der Binäre Input/Output des SBC86 auf Port 00H (Hexadezimal) befindet wie im Blockschaltbild zu sehen ist (siehe *Anleitung – Programmieren mit dem 8086*, S.8).

- b. Direkte Ansteuerung der 7-Segment-Anzeige
 - siehe „Anleitung – Programmieren mit dem 8086“, S. 9 und S. 8

Praktikum 1 – Einstieg und Befehle – Hilfestellung

Einführung des SBC86-Editors und Emulators

Hinweis zum Erzeugen eines List-Files:

Um ein List-File zu erzeugen, muss dem Compiler das entsprechende Kommando übergeben werden. Da Sie mit Geany kompilieren, muss diese Einstellung in Geany vorgenommen werden. Die Anleitung hierzu finden Sie in der Anleitung – Programmieren mit dem 8086 auf Seite 6-7.

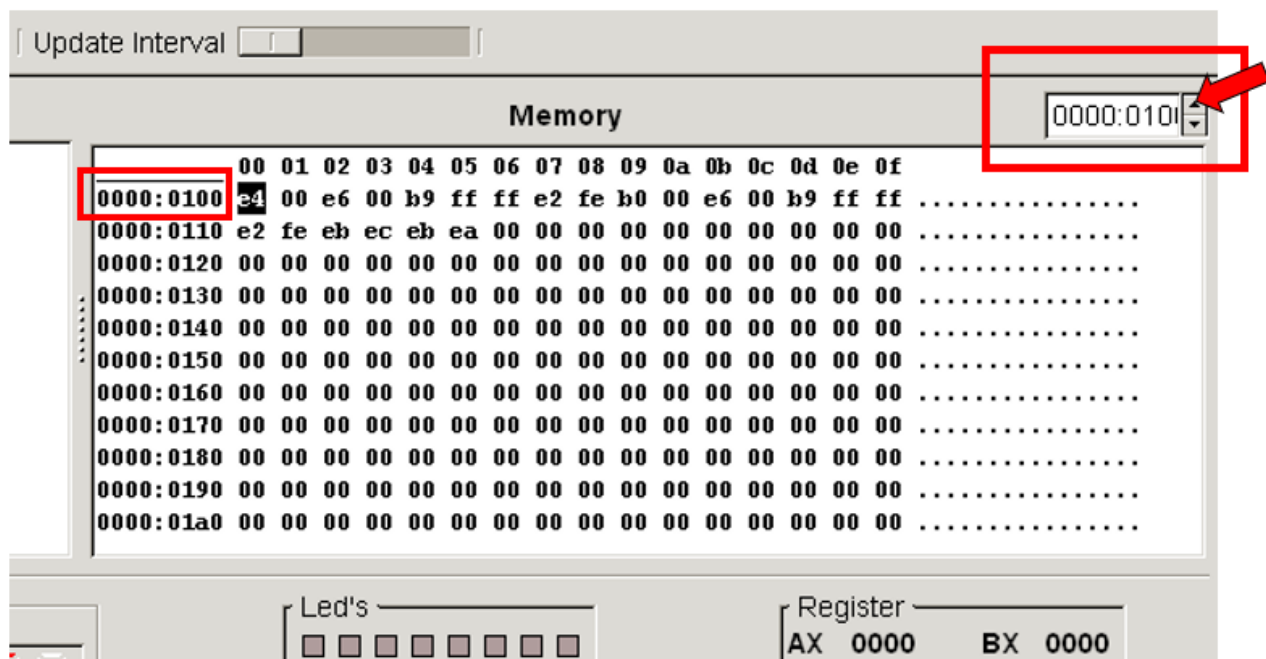
Bestimmen von OP-Codes und relativen Adressen aus dem List File

einstieg.asm		einstieg.lst	
1	1		org 100h
2	2		
3	3	00000000	E400
4	4	00000002	E600
5	5	00000004	B9FFFF
6	6		

		start: in al,00
		out 0,al
		mov cx,-1

Auslesen des Speichers und Disassemblers

Um an die Speicherstelle 0000:0100 zu gelangen, muss diese lediglich über die Schaltflächen rechts oberhalb des Memory eingestellt werden. Das Memory-Feld scrollt bei Betätigung der Pfeile an die eingestellte Speicherstelle.



Praktikum 1 – Einstieg und Befehle – Hilfestellung

Beispiel: Auslesen der Speicherstelle 0000:0115

	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f
0000:0100	e4	00	e6	00	b9	ff	ff	e2	fe	b0	00	e6	00	b9	ff	ff
0000:0110	e2	ff	eb	eb	b	ea	00	00	00	00	00	00	00	00	00	00
0000:0120	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0000:0130	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0000:0140	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0000:0150	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0000:0160	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0000:0170	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0000:0180	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0000:0190	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0000:01a0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

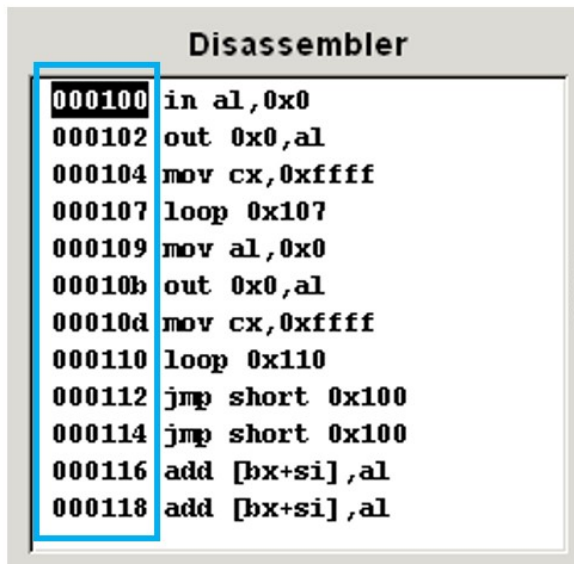
Die Beschriftung der Spalten hat eine führende 0, die lediglich der sauberen Darstellung dient. Es wird immer die letzte Ziffer der Zeilenzahl durch die zweite Ziffer der Spaltenzahl ersetzt. Aus 0000:0110 und 05 wird damit die Speicherstelle 0000:0115, in der in diesem Beispiel der Hex-Wert ea hinterlegt ist.

Tipp zur Interpretation der Werte an den Adressen 010e und 010f

Schauen Sie sich die Speicherstelle direkt vor den beiden Adressen ebenfalls an und vergleichen Sie mit dem List-File. Überlegen Sie als nächstes, welchen Wert „ff ff“ in anderen Zahlensystemen darstellt und wie dies zu Ihren Funden aus dem List-File passt.

Praktikum 1 – Einstieg und Befehle – Hilfestellung

Interpretation des Disassembler-Fensters

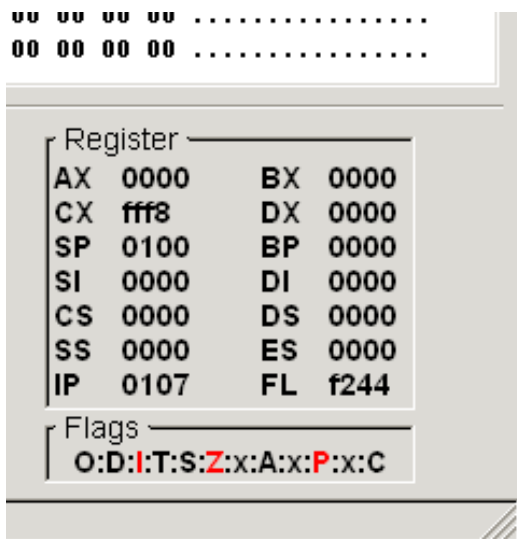


```
Disassembler
000100 in al,0x0
000102 out 0x0,al
000104 mov cx,0xffff
000107 loop 0x107
000109 mov al,0x0
00010b out 0x0,al
00010d mov cx,0xffff
000110 loop 0x110
000112 jmp short 0x100
000114 jmp short 0x100
000116 add [bx+si],al
000118 add [bx+si],al
```

Speicheradresse
des Befehls

Im Disassembler-Fenster wird Ihnen die tatsächliche Adresse der Befehle Ihres geladenen Programms im Speicher angezeigt.

Allgemeine Register



```
00 00 00 00 .....
00 00 00 00 .....

Register
AX 0000    BX 0000
CX fff8    DX 0000
SP 0100    BP 0000
SI 0000    DI 0000
CS 0000    DS 0000
SS 0000    ES 0000
IP 0107    FL f244

Flags
O:D:I:T:S:Z:X:A:X:P:X:C
```

Am rechten unteren Rand des Emulators können Sie die allgemeinen Register finden. Hierzu zählt auch der für das Praktikum gesuchte Instruction Pointer (IP). Dieser zeigt auf die Adresse des nächsten auszuführenden Befehls. Siehe auch *Befehlssatz des 8086*, S. 2

Praktikum 1 – Einstieg und Befehle – Hilfestellung

Denkanstöße für die Lösung der Programmieraufgaben

Schauen Sie sich noch einmal das Beispielprogramm am Anfang dieses Praktikums an. Es wird zu Beginn die Schalterstellung eingelesen und dann auf der LED-Zeile ausgegeben. Vollziehen Sie noch einmal nach, wie genau dies funktioniert und auch, aus welchem Grund an zwei Stellen ein Loop ohne Inhalt ausgeführt wird. Zuletzt: laut Blockschaltbild (*Anleitung – Programmieren mit dem 8086*, S. 8) ist die LED-Zeile ein binärer Output.

Haben Sie nun eine Idee, wie man dieser vielleicht einen anderen binären Output als die eingelesene Schalterstellung zum Anzeigen übergeben könnte?

FAQ

Warum blinken keine LEDs, wenn ich das Beispielprogramm laufen lasse?

- das Programm lässt die zu den Schaltern zugehörigen LEDs blinken → sind Schalter betätigt? Falls nein, bleiben die zugehörigen LEDs natürlich dunkel
- Auf Fehler beim Kopieren des Codes prüfen

Ich kriege eine Fehlermeldung beim Ausführen des Programms

- Überprüfen, dass `~/MPT86/einstieg` (kompilierte Datei ohne Dateiendung) oder unter Windows `~/MPT86/einstieg.com` geladen wurde
- Überprüfen Sie, dass die erste Zeile des Programms „`org 100h`“ lautet
- Code auf Syntaxfehler überprüfen