AIM-65 Assembler und Disassembler

AIM-65-Assembler und -Disassembler

In seiner Grundausstattung implementiert der AIM 65 die 6502-Maschinensprache (Objekt-Code) und im ROM-residenten Monitor die Disassemblersprache. Bei optionaler Beschaffung eines (bzw. weiterer) ROMs werden auch Assembler (und BASIC) verfügbar.

Der 6502 Maschinencode steht voll zur Verfügung und kann mittels Monitor-M-Befehl direkt in den Speicher eingegeben werden. Diesen stupiden Weg wird man aber in der Praxis nicht gehen, da schon die Grundausstattung über einen hochkomfortablen Monitor verfügt, der das Programmieren im Disassembler ermöglicht und zur Freude macht.

Der Disassembler benutzt die mnemonischen Kürzel für die Maschinenbefehle, absolute Adressen und grundsätzlich Hexadezimalzahlen. In Speicherstelle 00 wird inkrementiert und der jeweilige Zählerstand im Display angezeigt. (Es empfiehlt sich, hierbei den Drucker auszuschalten). Auch die Inhalte von X- und Y-Register werden ineinandergeschachtelt, schrittweise um eins erhöht (inkrementiert), um so eine Verzögerung des Ablaufs zu bewirken. (NB: Der AIM zählt also zur bloßen Verzögerung in jedem Durchgang von 0 bis 65535, zutreffender gesagt von 0000 bis FFFF, und benötigt dazu rund eine halbe Sekunde). Abschließend Rücksprung zum Programmanfang (endlose Schleife).

```
ROCKWELL AIM 65
< 1 >
0000          *=0200
0200  E6  INC  00
0202  A5  LDA  00
0204  20  JSR  EA46
0207  E8  INX
0208  C8  INY
0209  D0  BNE  0208
020B  8A  TXA
020C  D0  BNE  0207
020E  20  JSR  EA13
0211  4C  JMP  0200
0214
```

Bild 1: Das Disassemblerprogramm

Bild 1 zeigt das Listing dieses Programms, so wie es im Disassembler eingegeben wurde. (NB: Im ROM steht ab FA46 die Routine NUMA; sie bewirkt eine Ausgabe des Akkumulatorinhalts

als zweistellige Hexadezimalzahl. CRL0W (ab FA13) cleart das Display und bewirkt bei eingeschaltetem Drucker einen Papiervorschub).

```
< > ?
< M > = 0200 E6  00  A5  00
< I  > 0204 20  46  EA  E8
< >   0208 C8  D0  FD  8A
< >   020C D0  F9  20  13
< >   0210 EA  4C  00  02
```

Bild 2: Der Speicherauszug

Bild 2 zeigt den Speicherauszug dieses Programms, also den echten Maschinencode. Er wurde mittels Monitor-M-Befehl gewonnen. Das Maschinenprogramm hätte aber auch auf eben diesem Wege eingegeben werden können. (Bei genauem Hinsehen erkennt man in der Codierung das Disassemblerprogramm wieder).

Zur Darstellung des Programmierens im Assembler müssen wir etwas weiter ausholen. Zunächst einmal erfordert die Assemblersprache nicht unbedingt die Verwendung von Hexadezimalzahlen; je nach vorangestelltem Kennzeichen können auch Dezimal-, Oktal- oder Binärzahlen verwendet werden.

§ ist beispielsweise das Kennzeichen für eine Hexadezimalzahl. Sodann erlaubt der Assembler die Benutzung symbolischer Adressen:

a) Der erste Befehl erhält das Label START und kann z.B. vom Programmende her mit der Anweisung JMP START angesprungen werden. Entsprechende VERZ1, VERZ2.

b) Die in 00 zu erhöhende Zahl läßt sich z.B. als ZAHL ansprechen (nach Definition in der Symboltabelle) und die Subroutinen lassen sich (ebenfalls nach Definition) unter ihrem Namen aufrufen.

Nach entsprechender Eingewöhnung werden die Programme hiermit leichter lesbar. Außerdem bietet der Assembler die Möglichkeit, das Programm mit Kommentaren zu versehen, und eine Reihe weiterer Befehle z.B. .OPT LIS, GEN, ERR beispielsweise veranlassen den Assembler, das Programm zu listen, das Maschinenprogramm zu generieren und die Fehler mittels eines besonderen Fehlercodes anzuzeigen. Das Programm endet mit der Anweisung .END.

```

< E >
EDITOR
FROM=0400 TO = 0700

IN =
 PAGE 'ZAEHLPROGRAMM
'
 PAGE 'SYMBOLTABELLE
'
ZAHL = $00
CRLOW = $EA13
NUMA = $EA46
.PAGE 'STARTADRESSE'
*= $0200
.PAGE
START INC ZAHL
LDA ZAHL
JSR NUMA
VERZ1 INX
VERZ2 INY
BNE VERZ2
TXA
BNE VERZ1
JSP CRLOW
JMP START
.PAGE
.OPT LIS
.END
= < Q >

< N >
ASSEMBLER
PROM=0800 TO=0800
IN=M
LIST?Y
LIST-OUT=P

```

```

OBJ?N
PASS 1

PASS 2

ERRORS = 0000

```

**Bild 3:
Der absichtlich ungeordnete Quelltext und der Assembleraufruf**

Bild 3 zeigt das Assemblerprogramm, wie es mittels Textaufbereitungsprogramm (Editor) (hier absichtlich unübersichtlich) eingegeben wurde. Das zur Grundausstattung gehörende Textaufbereitungsprogramm (im ROM) ermöglicht nicht nur eine bequeme Eingabe des Programmtextes in den Textpuffer, sondern auch eine schnelle Korrektur, insbesondere das Löschen oder Einfügen von Zeilen, Korrigieren von Eingabefehlern. Hierin liegt möglicherweise

der Hauptvorteil des Assemblers: Man muß nicht mehr, wie z.B. beim Disassembler, beim Einfügen neuer Zeilen neu durchnummerieren; man korrigiert den Fehler und assembliert neu. Alles weitere geschieht automatisch. Es entsteht zusätzlicher Speicherbedarf: Page 0 steht für Variable bis 00E0 zur Verfügung; das reicht gut aus. Page 1 wird ohnehin vom Stack benutzt, steht dem Anwender also praktisch nur herfür zur Verfügung. Üblicherweise beginnt das Maschinenprogramm in 0200. Man kann dann z.B. von 0400 bis 0700 den Textpuffer einrichten und für die Symboltabelle die Speicherplätze 0800 bis 0B00 reservieren. Man wird also mit der 1KB-Version des AIM kaum auskommen.

Um einen kleinen Einblick vom praktischen Ablauf zu geben, seien die wichtigsten Schritte aufgezeigt:

- 1.) Aufruf des Textaufbereitungsprogramms durch E-Befehl
- 2.) Eingabe der Textpuffergrenzen
- 3.) Eingabe des Quellprogramms
- 4.) Ausstieg aus dem Editor mittels Q-Befehle

Der Programmtext steht jetzt im Textpuffer.

- 5.) Aufruf des Assemblers mittels N-Befehl
- 6.) Bereitstellen von Speicherraum für die Symboltabelle; wir haben im Beispiel mit 769 Bytes erheblich überzogen!
- 7.) Eingabe diverser Parameter, z.B. Wo steht das Quellprogramm, auf Band?, im Speicher (memory)?, wohin soll der Objektcode geleitet werden, zum Drucker?, auf Band? Soll zum Beispiel der Objektcode nicht nach aussen sondern in den Speicher geleitet werden, dann heißt der Parameter N (No).
- 8.) Der Assembler startet automatisch zwei Durchläufe, zeigt ggf. die Fehler an und leitet, wenn alles in Ordnung ist, das Maschinenprogramm zum zugewiesenen Baustein.
- 9.) Waren noch Fehler vorhanden, so ruft man mit T das Textaufbereitungsprogramm zum zweiten Male auf, korrigiert und assembliert neu.

Je nach Parametern und zusätzlichen Assembleranweisungen läßt sich auch das wohlgeordnete und kommentierte Listing von Bild 4 erhalten. Man beachte, daß es sowohl Maschinencode als auch Assemblercode wiedergibt. NB: Das abschließende Objektlisting beginnt mit (hexadezimal) 14, der Anzahl der Maschinenbefehle. Es folgen die Startadresse 0200 und 20 (= \$14) Maschinenbefehle. Die abschließende Zahl 0A2C ist die (Kontroll-)Summe aller vorstehenden zweistelligen Hexadezimalzahlen.

<N>
 ASSEMBLER
 FROM=0800 TO=0B00
 IN=M
 LIST?Y
 LIST-OUT=P

OBJ?Y
 OBJ-OUT=P

J
 PASS 1

PASS 2

 ZAEHLPROGRAMM

 SYMBOLTABELLE

== 0000 ZAHL
 = \$00
 == 0000 CRLOW
 = \$EA13
 == 0000 NUMA
 = \$EA46

 STARTADRESSE

== 0000
 *=\$0200

 == 0200 START
 E600 INC ZAHL
 A500 LDA ZAHL
 2046EA JSR NUMA
 == 0207 VERZ1
 E8 INX
 ==0208 VERZ2
 C8 INY
 D0FD BNE VERZ2
 8A TXA
 D0F9 BNE VERZ1
 2013EA JSR CRLOW
 400002 JMP START

.OPT LIS
 .END

ERRORS= 0000
 ; 140200E600A5002046E
 AE8C8D0FD8AD0F92013E
 A4C00020A2C

Bild 4: Die fertige Programmdokumentation

H. Ostermann ■

TOPP -aktuell

Buchreihe Elektronik

Aus mehr als 100 Bänden

Best.-Nr. 455
 G. Abeldt
BASIC
 80 Seiten, kartoniert
 Besondere Eigenschaften von
 BASIC: Sprachumfang, Zusam-
 menstellung der Anweisungen
 und Befehle. BASIC Sprach-
 elements. Konstante, Variable,
 Mathematische Ausdrücke,
 BASIC-Befehle, Befehle von
 Ein- und Ausgabegerät (Terminal).

DM 9,--

finden Sie immer das Richtige

Best.-Nr. 166/167
 H. Bernstein
TV-Computerspiele
 206 Seiten, 120 Abb., kart.
 Aufbau, Funktion und Bauan-
 leitungen für Computerspiele
 über das Fernsehgerät.

DM 23,--

für Ihr Hobby, zum Nachschlagen und zur

Best.-Nr. 478
 Schwarz
Heim-Schachcomputer
 181 Seiten, 96 Abb., kart.
 Eine Auswahl der auf dem
 Markt erhältlichen Schach-
 computer, deren Leistung und
 Möglichkeiten. Schon fast ein
 Testbericht.

DM 19,80

beruflichen Weiterbildung

wir sind Mitglied
 der Informationsgemeinschaft
 elektro

Informieren Sie sich!
 Kostenlos erhalten Sie das Gesamtverzeichnis
 das Heft „Elektronik-Probleme?“

frech-verlag

7000 Stuttgart 31, Turbinenstraße 7