

Disassembler

Rechner: Atari XL, XE
 Programmname: Disassembler
 Programmiersprache: BASIC
 Programmlänge: 4500 Bytes
 Zubehör: Drucker optional

Dieser Disassembler soll in der Hauptsache BASIC-Programmierern, die gelegentlich Maschinenroutinen in ihren Programmen nutzen, um die Leistungsfähigkeit ihrer Programme zu steigern, eine Programmierhilfe sein.

Es ist empfehlenswert, aber nicht von Nöten, den Disassembler mit der Diskettenstation zu verwenden, da beim Gebrauch des Disassemblers das im Speicher stehende BASIC-Programm dem Disassembler sehr oft weichen muß.

Bei der Arbeit mit dem Disassembler gehen Sie am Besten so vor:
 Speichern Sie das BASIC-Programm ab. Laden Sie anschließend den Disassembler in den Rechner. Nach dem Start des Disassemblers geben Sie die Anfangsadresse und die Endadresse des Speicherbereichs, den Sie ansehen möchten ein. Auf dem Bildschirm wird jeder Assembler-Befehl mit dezimaler Adresse, Mnemonic und Code in dezimaler und hexadezimaler Form ausgegeben. Trifft der Disassembler dabei auf

leeren Speicher, also auf mehrere aufeinander folgende 'BRK' mit dem Code Null, so stellt er nur das Erste dar. Der Computer beginnt das Disassemblieren erst wieder beim nächsten von 'BRK' verschiedenen Befehl. Wenn ein Drucker angeschlossen ist, findet auch dort eine Ausgabe der Daten statt. Jetzt können Sie in aller Ruhe die Maschinenroutine betrachten. Eventuelle Verbesserungen führen Sie dann "von Hand" in Ihrem vorher abgespeicherten BASIC-Programm aus.

Hinweis zur Programmeingabe

Da einige Zeilen sehr lang sind, wird vor Abschreiben des Programmes empfohlen, den Befehl

POKE 82,0
 im Direktmodus einzugeben. Dieser Befehl verlängert die Zeilen um sechs Zeichen.

Ergeben sich beim Gebrauch des Druckers Probleme, müssen eventuell in Zeile 96 und 97 die Druckersteuerungszeichen Ihrem Drucker entsprechend geändert werden. Sie können sie aber auch weglassen, da die Zeilen lediglich einer Verschönerung des Druckbildes dienen.

Jan Spitzkowsky

```

1 REM *****
2 REM DISASS
3 REM FUER CHIP-SPECIAL
4 REM VON JAN SPITZKOWSKY
5 REM *****
10 ? "Startadresse";:INPUT ST
15 ST1=ST:LN=J1
20 ? "Endadresse";:INPUT EN
30 RESTORE
31 DIM BEF$(4),A$(70),C$(1000),HEX$(
16),MC$(100)
32 MC$="{CODE169/0/133/18/133/19/133
/20}hh{CODE133/204}h{CODE133/203/
160/0/177/203/153/206/0/200/192/4
/208/246/165/207/201}?{CODE208/1/
96/24/165/203}i{CODE4/133/203/144
/2/230/204/160/0/177/203/197/205/
240/19/200/192/13/208/245/24/165/
203}i{CODE13/133/203/144/2/230/20

```

```

4/24/144/199/152/133/18/96}"
36 C$=" ADC{CODE2}ieu{CODE2}m{CODE12
5}y{CODE2/2}aq{CODE2} AND{CODE2})
%5{CODE2}=-9{CODE2/2}!i{CODE2} AS
L{CODE10/2/6/22/2/14/30/2/2/2/2/2
/2/165}BCC{CODE2/2/2/2/2/2/2/2/2
144/2/2/2/165}BCS{CODE2/2/2/2/2/2
/2/2/2/176/2/2/2/165}BEQ{CODE2/2/
2/2/2/2/2/2/2/2/2/2/2/2/2/2/2/2
/165}BMI{CODE2/2/2/2/2/2/2/2/2/2/2
CODE2/2/2/165}BNE{CODE2/2/2/2/2/2
/2/2/2/208/2/2/2/165}BPL{CODE2/2/
2/2/2/2/2/2/2/2/16/2/2/2} BRK{CODE2
/2/2/2/2/2/2/2/2/0/2/2/2/165}BVC{
CODE2/2/2/2/2/2/2/2/2/2/2/2/2/2/2}
38 C$(LEN(C$)+1)="P{CODE2/2/2/165}BV
S{CODE2/2/2/2/2/2/2/2/2/2/2/2/2/2/2}p{CODE2/2

```



```
670 A$(J18)=STR$(PEEK(ST+J1)+PEEK(ST+J2)*256):A$(23,24)=",X":ST=ST+J3:FL=J2:GOTO J9
700 A$(J18)=STR$(PEEK(ST+J1)+PEEK(ST+J2)*256):A$(23,24)=",Y":ST=ST+J3:FL=J2:GOTO J9
730 ST=ST+J1:FL=J3:GOTO J9
760 A=PEEK(ST+J1):IF A>127 THEN A=A-256:A$(13,13)="{CODE164}"
770 A$(J18)=STR$(A+ST+J2):ST=ST+J2:FL=J1:GOTO J9
790 A$(J18,J18)="(:A$(J18+J1)=STR$(PEEK(ST+J1)):A$(22)=",X":ST=ST+J2:FL=J1:GOTO J9
820 A$(J18,J18)="(:A$(J18+J1)=STR$(PEEK(ST+J1)):A$(22)=",Y":ST=ST+J2:FL=J1:GOTO J9
850 A$(J18)="(:A$(J18+J1)=STR$(PEEK(ST+J1)+PEEK(ST+J2)*256):A$(24)="):ST=ST+J3:FL=J2:GOTO J9
880 FL=J3:ST=ST+J1
900 RESTORE
910 IF ST>EN+J1 THEN END
915 A=PEEK(ST-FL)
917 IF A=J0 AND FL<J3 THEN A$(35,35)="0":A$(49,50)="00":GOTO 925
920 IF FL<J3 THEN A$(35-INT(CLOG(A)),35)=STR$(A):C=J2:GOSUB 395:C=J1:GOSUB 390:A$(49,50)=BEF$(J1,J2)
925 A=PEEK(ST-J1)
927 IF A=J0 AND FL=J2 THEN A$(39,39)="0":A$(52,53)="00":GOTO 980
930 IF FL=J2 THEN A$(39-INT(CLOG(A)),39)=STR$(A):C=J2:GOSUB 395:C=J1:GOSUB 390:A$(52,53)=BEF$(J1,J2)
950 A$(56)=" "
955 IF BRKF=J1 THEN ? :? "{CODE127/30}BRK":? :BRKF=J0:IF DRCKFLG THEN LPRINT :LPRINT "{CODE9}{6SPACE S}BRK":LPRINT
960 ? A$(J1,5);A$(13,24);A$(28,39);A$(45,53)
980 IF DRCKFLG THEN LPRINT A$(J1,11);"{3SPACES}";A$(12,28);"{8SPACES}";A$(29)
990 GOTO 100
```

Zahlensysteme rasch umgerechnet

Rechner: Atari 400/800 und XL
Programmname: Dehebino
Programmlänge: 3050 Bytes
Programmiersprache: BASIC
Zubehör: ---
Beim Programmieren ist es immer wieder von Nutzen über ein Hilfsprogramm zu verfügen, das die Umrechnung einer Zahl aus einem Zahlensystem in ein Anderes ermöglicht. Dies spart Zeit und Nerven. Das vorliegende Programm erlaubt Umrechnungen zwischen dem Dezimalen, Hexadezimalen, Binären und Octalen Zahlensystem. Es besteht allerdings eine Begrenzung der Zahlenbereiche:
Dez 65535
Hex FFFF
Bin 16 Stellen

Oct 177777
Nach dem Start des Programmes erscheint die Kopfzeile. Das "?" fordert zur Eingabe einer Zahl in der Spalte DEZ auf. Möchten Sie die Eingabe in einer anderen Spalte machen, drücken Sie solange "RETURN" bis das "?" in der gewünschten Spalte erscheint. Geben Sie nun Ihre Zahl ein. Auch Spaces werden bei der Eingabe akzeptiert, und Nullstellen können durch Punkte ersetzt werden, um die Übersichtlichkeit zu erhöhen. Das Programm rechnet zuerst immer in das Dezimal-System zurück, und von dort in die Anderen. Dies erscheint auf den ersten Blick umständlich, verringert aber den Programmumfang.
Werner Warthmann

```
50 REM *****
55 REM *{6SPACES}{CODE160/196/197/200/197/194/201/206/207/160}{7SPACES}*
60 REM * DEZ<->HEX<->BIN<->OCT *
65 REM *{6SPACES}UMRECHNUNG{7SPACES}*
70 REM * fuer CHIP-Spezial von *
75 REM *{3SPACES}Werner{2SPACES}Warthmann{3SPACES}*
80 REM *****
90 GRAPHICS 0:SETCOLOR 4,1,10:SETCOLOR 1,0,12
```