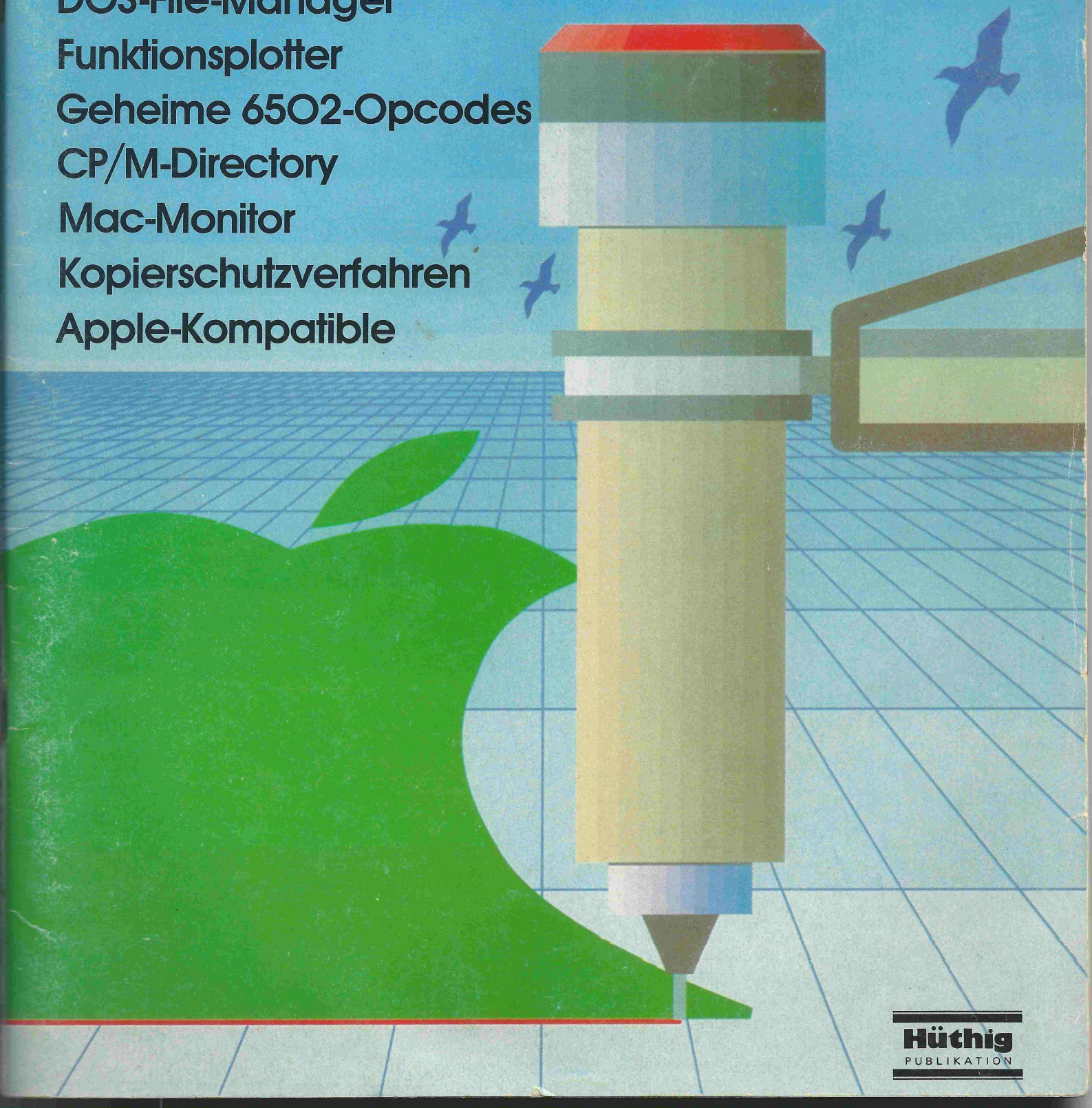
Nr. 5/85 Mai
DM 6,50, sfr 6,50, öS 50, Lit 5900, hfl 7,50

PEEKER

MAGAZIN FÜR APPLE-COMPUTER

DOS-File-Manager
Funktionsplotter
Geheime 6502-Opcodes
CP/M-Directory
Mac-Monitor
Kopierschutzverfahren
Apple-Kompatible

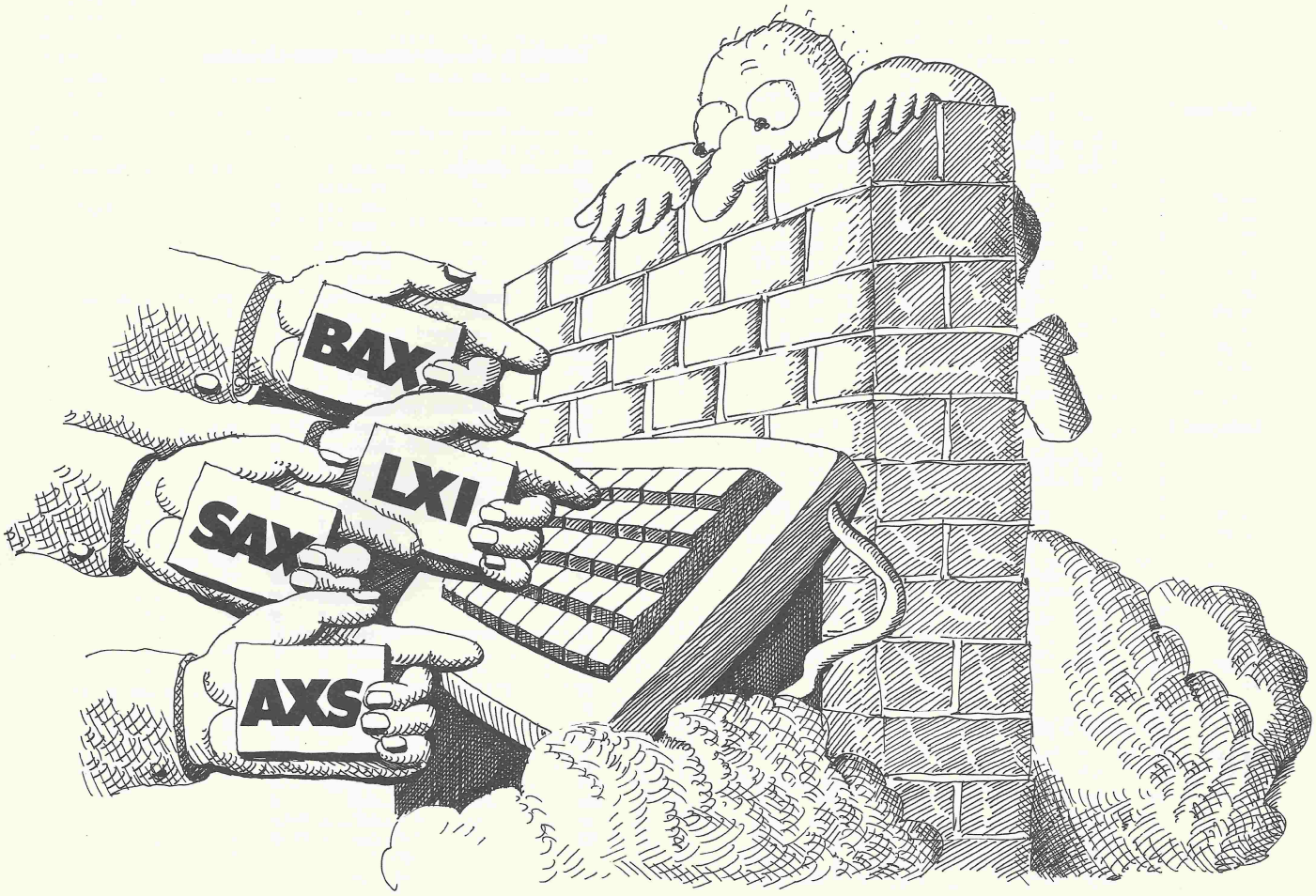


Hüthig
PUBLIKATION

Disassembler für „geheime“ 6502-Opcodes

von Christoph Bregler

Der hier vorgestellte Disassembler ist eine Erweiterung des alten Monitor-Disassemblers um 93 zusätzliche 6502-NMOS-Opcodes, die in den Datenblättern und in den meisten Assembler-Büchern nicht beschrieben werden.
Die Befehle sind auf dem Apple II, II+ und IIe verfügbar, da diese Geräte im Gegensatz zum IIc, der den 65C02-CMOS-Prozessor enthält, noch die alte NMOS-Version benutzen.



Bei dem normalen 6502-Prozessor sind von 256 möglichen Codes nur 151 als Assembler-Befehle definiert. Die CPU interpretiert aber aus den restlichen Codes auch Befehle, die zwar zum Teil unsinnig sind, aber auch manche nützliche Operationen enthalten.
Mit dem vorhandenen Monitor-Disassembler kann man ein Programm mit solchen Befehlen nicht auslisten, da dann nur „???“ ausgedruckt wird. Ich habe nun einen „Extended Disassembler“ (**EDA**) programmiert, der die zusätzlichen Befehle richtig ausgibt.
Nach dem Eintippen des Programms sollte es mit „BSAVE EDA, A$94C8, L$0138“ auf Diskette gespeichert werden. Durch „BRUN EDA“ initialisiert sich die Routine selbst und kann dann vom Monitor aus wie der „L“-Befehl aufgerufen werden, wobei das „L“ durch Ctrl-Y zu ersetzen ist.
EDA ist keine gepatchte Language-Card-Version des alten Disassemblers, sondern wird in der Page 3 und direkt unter dem DOS-Puffer abgespeichert, so daß er verschwindend wenig vom freien Speicherplatz in Anspruch nimmt.
Somit läuft EDA auch auf Apple-II-Versionen mit nur 48K RAM oder unter ProDOS, das die Language-Card benutzt.

Im folgenden sind zwei Anwendungsbeispiele der zusätzlichen Op-Codes näher erläutert:

DEC/CMP adr.
INC/SBC adr.

Diese Befehle erniedrigen oder erhöhen den Inhalt einer Speicheradresse und vergleichen das Ergebnis mit dem Akkumulator.
Solche Operationen sind oft für die Ausführung von Schleifen nützlich, in denen z.B. ein Pointer von einer Anfangsadresse auf eine Endadresse heruntergezählt wird. Dies ist nun mit einem einzigen Befehl möglich.
Beispiel 1 zeigt eine solche Schleife, die einen Speicherbereich löscht.

AXM #Operand

Dieser Befehl verknüpft Akkumulator und X-Register durch die AND-Funktion, zieht davon den Operanden ab und speichert das Ergebnis in das X-Register. Das Carry-Flag muß nicht gesetzt oder gelöscht werden. Wird der Akkumulator vorher mit $FF geladen, so schafft dieser Befehl die Möglichkeit, direkt im X-Register zu addieren oder zu subtrahieren.
Beispiel 2 demonstriert diesen Befehl und den Zeitgewinn gegenüber der Benutzung der „echten“ Codes.

Eine Auflistung aller 93 zusätzlichen Codes kann der **Tabelle 1** entnommen werden*.
Bei vielen Befehlen handelt es sich nur um eine Aneinanderreihung zweier normaler 6502-Befehle, wobei die Mnemonics dann aus den zwei Befehlen und einem Schrägstrich bestehen.
Wegen der fehlenden Festlegung habe ich für die restlichen Codes die Mnemonics frei definiert. Bei ihnen druckt EDA dann noch ein „§“ davor, um sie von den anderen Befehlen abzusetzen.
Alle übrigen nicht aufgelisteten Codes „hängen den Prozessor auf“ (z.B. $02). Man könnte sie höchstens als HALT-Befehle benutzen, bei denen der Prozessor dann wartet, bis ein Reset oder Interrupt erfolgt.

* Eine analoge Matrixtabelle findet man in „incider“, 4/85, S. 121

Beispiel 1

```
                1   * X = High-Byte der Anfangsadresse-1
                2   * A = High-Byte der Endadresse
                3
A0 00           4           LDY  #0
84 06           5           STY  $6
85 07           6           STA  $7
98              7   LP1     TYA
91 06           8   LP2     STA  ($6),Y
C8              9           INY
D0 FB           10          BNE  LP2
8A              11          TXA
C7 07           12          DEC/CMP $7
90 F5           13          BCC  LP1
```

Beispiel 2

```
                1   * Subtraktion (z.B. X - 7 -> X)
                2   * 4 Mikrosekunden
                3
A9 FF           4           LDA  #$FF
CB 07           5           AXM  #7         ;C=0 bei Überlauf
                6
                7   * Subtraktion mit "echten" Opcodes
                8   * 8 Mikrosekunden
                9
8A              10          TXA
38              11          SEC
E9 07           12          SBC  #7
AA              13          TAX
                14
                15  * Addition (z.B. X + 13 -> X)
                16
A9 FF           17          LDA  #$FF
CB F3           18          AXM  #-13       ;C=1 bei Überlauf
                19  * Im Zweierkomplemet ist -13 = $F3
```

Tabelle 1: Die „geheimen“ 6502-Opcodes

Code:	Mnemonic:	Adress.:	Zeit:	Funktion:
03	ASL/ORA	(ind.,X)	8	ASL Adr. & ORA Adr.
07	-"-	zeropage	5	-"-
0F	-"-	absolute	6	-"-
13	-"-	(ind.),Y	8	-"-
17	-"-	z.page,X	6	-"-
1B	-"-	absol.,Y	7	-"-
1F	-"-	absol.,X	7	-"-
23	ROL/AND	(ind.,X)	8	ROL Adr. & AND Adr.
27	-"-	zeropage	5	-"-
2F	-"-	absolute	6	-"-
33	-"-	(ind.),Y	8	-"-
37	-"-	z.page,X	6	-"-
3B	-"-	absol.,Y	7	-"-
3F	-"-	absol.,X	7	-"-
43	LSR/EOR	(ind.,X)	8	LSR Adr. & EOR Adr.
47	-"-	zeropage	5	-"-
4F	-"-	absolute	6	-"-
53	-"-	(ind.),Y	8	-"-
57	-"-	z.page,X	6	-"-
5B	-"-	absol.,Y	7	-"-
5F	-"-	absol.,X	7	-"-
63	ROR/ADC	(ind.,X)	8	ROR Adr. & ADC Adr.
67	-"-	zeropage	5	-"-
6F	-"-	absolute	6	-"-
73	-"-	(ind.),Y	8	-"-
77	-"-	z.page,X	6	-"-
7B	-"-	absol.,Y	7	-"-
7F	-"-	absol.,X	7	-"-
83	STX/STA	(ind.,X)	6	Store (A AND X) Adr.
87	-"-	zeropage	3	-"-
8F	-"-	absolute	4	-"-
97	-"-	z.page,Y	4	-"-
A3	LDX/LDA	(ind.,X)	6	LDX Adr. & LDA Adr.
A7	-"-	zeropage	3	-"-
AB	-"-	immed.	2	-"-
AF	-"-	absolute	4	-"-
B3	-"-	(ind.),Y	5	-"-

B7	–"–	z.page,Y	4	–"–
BF	–"–	absol.,Y	5	–"–
C3	DEC/CMP	(ind.,X)	8	DEC Adr. & CMP Adr.
C7	–"–	zeropage	5	–"–
CF	–"–	absolute	6	–"–
D3	–"–	(ind.),Y	8	–"–
D7	–"–	z.page,X	6	–"–
DB	–"–	absol.,Y	7	–"–
DF	–"–	absol.,X	7	–"–
E3	INC/SBC	(ind.,X)	8	INC Adr. & SBC Adr.
E7	–"–	zeropage	5	–"–
EF	–"–	absolute	6	–"–
F3	–"–	(ind.),Y	8	–"–
F7	–"–	z.page,X	6	–"–
FB	–"–	absol.,Y	7	–"–
FF	–"–	absol.,X	7	–"–
93	BAX	(ind.),Y	6	Speichere <B>it0/3 von <A> AND <X> nach Adr.
4B	LSA	immed.	2	AND Adr. & LSR A
6B	ROA	immed.	2	AND Adr. & ROR A
8B	LXI	immed.	2	<L>ade A mit <X> AND <I>mmedediate
9B	SAX	impl.	5	Lade <S> mit <A> AND <X> (besteht aus 3 Bytes, von denen die letzten 2 Bytes beliebig sein können)
BB	AXS	absol.,Y	4	Lade <A>,<X>,<S> mit S AND Adr.
CB	AXM	immed.	2	Lade X mit <A> AND <X> <M>inus immediate
9C	SBY	absol.,X	5	<S>peichere <B>it0/3 von <Y> nach Adr.
9E	SHX	absolute	5	<S>peichere <H>igh Byte+1 der Adr. AND <X> nach Adr.
9F	HXA	absolute	5	Speichere <H>igh Byte+1 der Adr. AND <X> AND <A> nach Adr.
0B,2B	AND	immed.		
EB	SBC	immed.		
1A,3A,5A 7A,DA,FA	NOP	impl.		
80,82,C2 E2,04,14 34,44,54 64,74,D4 F4,89	NOP	impl.		besteht aus 2 Bytes, von denen das letzte Byte beliebig sein kann
0C,1C,3C 5C,7C,DC FC	NOP	impl.		besteht aus 3 Bytes, von denen die letzten 2 Bytes beliebig sein können

EDA

```
1   ***********************************
2   *                                 *
3   * E D A :                         *
4   *                                 *
5   * (E)xtended (D)is-(A)ssembler    *
6   *                                 *
7   * von Christoph Bregler           *
8   *       Oktober 1984              *
9   *                                 *
10  *                                 *
11  * für die zusätzlichen            *
12  * 6502-Codes                      *
13  * (keine 65C02-Codes!)            *
14  *                                 *
15  * Speicherbelegung: $0300-$03CC   *
16  *                   $95B8-$95FF   *
17  *                                 *
18  * Benutzung:                      *
19  * Wie den Monitor-List-Befehl.    *
20  * Anstatt 'L' wird Ctrl-Y ein-    *
21  * gegeben.                        *
22  *                                 *
23  ***********************************
24
25
26  LMNEM    EQU  $2C
27  FORMAT   EQU  $2E
28  LENGTH   EQU  $2F
29  CSWL     EQU  $36
30  CSWH     EQU  $37
31  PCL      EQU  $3A
32  PCH      EQU  $3B
33  HIMEM    EQU  $73
34  XSAVE    EQU  $EE
35  XSAV1    EQU  $EF
36  LCOUNT   EQU  $FC
37  COUNT    EQU  $FD
38  CSWL1    EQU  $FE
39  CSWH1    EQU  $FF
40
41  INSDS1   EQU  $F882
42  INSDS2   EQU  $F88C
43  INSTDSP  EQU  $F8D0
44  NXTCOL   EQU  $F8F5
45  PCADJ    EQU  $F953
46  COUT     EQU  $FDED
47  A1PC     EQU  $FE75
48
50           ORG  $94C8
51
52  * Schiebt EDA in Page 3,
53  * setzt Ctrl-Y-Pointer und HIMEM
54  * für die Opcode-Tabellen
55
94C8: A2 CD  56           LDX  #FMT3-ROUT
94CA: BD EA 94 57  INITLP  LDA  ROUT-1,X
94CD: 9D FF 02 58          STA  $2FF,X
94D0: CA     59           DEX
94D1: D0 F7  60           BNE  INITLP
94D3: A9 4C  61           LDA  #$4C
94D5: 8D F8 03 62         STA  $3F8
94D8: A9 00  63           LDA  #<EXTLIST
94DA: 8D F9 03 64         STA  $3F9
94DD: A9 03  65           LDA  #>EXTLIST
94DF: 8D FA 03 66         STA  $3FA
94E2: A9 B8  67           LDA  #<FMT3
94E4: 85 73  68           STA  HIMEM
94E6: A9 95  69           LDA  #>FMT3
94E8: 85 74  70           STA  HIMEM+1
94EA: 60     71           RTS
72
73  ROUT
74
75  * EDA-Beginn bei $300
76
77           ORG  $300
78
79  * Ruft 20mal (20 Zeilen) INSTDSP2
80  * auf und erhöht mit PCADJ den
81  * Befehlszähler
82
0300: A5 36  83  EXTLIST  LDA  CSWL
0302: 85 FE  84           STA  CSWL1
0304: A5 37  85           LDA  CSWH
0306: 85 FF  86           STA  CSWH1
0308: A9 14  87           LDA  #20
030A: 85 FC  88           STA  LCOUNT
030C: 20 75 FE 89         JSR  A1PC
030F: 20 47 03 90  EXTLIST1 JSR  INSTDSP2
0312: 20 2A 03 91         JSR  STCSW2
0315: 20 53 F9 92         JSR  PCADJ
0318: 85 3A  93           STA  PCL
031A: 84 3B  94           STY  PCH
031C: C6 FC  95           DEC  LCOUNT
031E: D0 EF  96           BNE  EXTLIST1
0320: 60     97           RTS
98
99  * Biegt CSWL-Pointer zu COUT2 um
100
0321: A2 33  101 STCSW1   LDX  #<COUT2
0323: 86 36  102          STX  CSWL
0325: A2 03  103          LDX  #>COUT2
0327: 86 37  104          STX  CSWH
0329: 60     105          RTS
106
107 * Setzt CSWL zurück, so daß ein
108 * eventuelles DOS nichts "merkt"
109
032A: A6 FE  110 STCSW2   LDX  CSWL1
032C: 86 36  111          STX  CSWL
032E: A6 FF  112          LDX  CSWH1
0330: 86 37  113          STX  CSWH
0332: 60     114          RTS
115
116 * COUT-Routine, die sowohl ohne
117 * DOS als auch mit DOS nicht
118 * "abzuhängen" ist und nach dem
119 * Xten Aufruf (X=COUNT) wieder zu
120 * EDA springt
121
0333: 86 EE  122 COUT2    STX  XSAVE
0335: 20 2A 03 123        JSR  STCSW2
0338: 20 ED FD 124        JSR  COUT
033B: 20 21 03 125        JSR  STCSW1
```

```
033E: A6 EE     126           LDX  XSAVE
0340: C6 FD     127           DEC  COUNT
0342: D0 02     128           BNE  RET
0344: 68        129           PLA
0345: 68        130           PLA
0346: 60        131 RET       RTS
                132
                133 * Prüft, ob ein nicht definierter
                134 * Befehl anliegt
                135
0347: 20 82 F8  136 INSTDSP2  JSR  INSDS1
034A: C9 10     137           CMP  #$10       ;?-CODE
034C: D0 28     138           BNE  CONT
034E: 20 21 03  139           JSR  STCSW1
                140
                141 * Prüft, ob ein Befehl anliegt,
                142 * der durch die folgende De-
                143 * codierung falsch ausgedruckt wird
                144
0351: B1 3A     145           LDA  (PCL),Y
0353: A2 10     146           LDX  #16
0355: DD EF 95  147 EXTLP     CMP  EXTBL,X
0358: F0 50     148           BEQ  EXT
035A: CA        149           DEX
035B: 10 F8     150           BPL  EXTLP
                151
035D: 4A        152           LSR
035E: B0 19     153           BCS  ODD
                154
                155 * Decodierung der NOP- und CRASH-
                156 * Befehle
                157
0360: 4A        158           LSR
0361: 29 03     159           AND  #3
0363: AA        160           TAX
0364: F0 0C     161           BEQ  CRASH
0366: BD B8 95  162 NOP       LDA  FMT3,X
0369: 85 2E     163           STA  FORMAT
036B: 29 03     164           AND  #3
036D: 85 2F     165           STA  LENGTH
036F: A9 3E     166           LDA  #$3E       ;NOP+?
0371: 38        167           SEC
0372: 90 F2     168 CRASH     BCC  NOP
0374: 49 10     169           EOR  #$10       ;?-CODE
0376: 4C D3 F8  170 CONT      JMP  INSTDSP+3
                171
                172 * Decodierung der "Doppel-Mnemo-
                173 * nic"-Befehle (z.B. INC/SBC)
                174
0379: 2A        175 ODD       ROL
037A: 29 F0     176           AND  #$F0
037C: 09 05     177           ORA  #5
037E: 20 8E F8  178           JSR  INSDS2+2
0381: 48        179           PHA
0382: B1 3A     180           LDA  (PCL),Y
0384: 29 F0     181           AND  #$F0
0386: 09 06     182           ORA  #6
0388: 20 8E F8  183           JSR  INSDS2+2
038B: 48        184           PHA
038C: B1 3A     185           LDA  (PCL),Y
038E: AA        186           TAX
038F: 4A        187           LSR
0390: 4A        188           LSR
0391: 4A        189           LSR
0392: 8A        190           TXA
0393: 69 FE     191           ADC  #-2
0395: 20 8E F8  192           JSR  INSDS2+2
0398: A9 0F     193           LDA  #15        ;COLM.-8
039A: 85 FD     194           STA  COUNT
039C: 68        195           PLA
039D: 20 D3 F8  196           JSR  INSTDSP+3
03A0: A9 AF     197           LDA  #"/"
03A2: 20 ED FD  198           JSR  COUT
03A5: A2 03     199           LDX  #3
03A7: 4C E9 F8  200           JMP  $F8E9
                201
                202 * Decodierung der "unregelmäßi-
                203 * gen" Codes nach den Opcode-
                204 * Tabellen in PAGE $95
                205
03AA: BD BC 95  206 EXT       LDA  FMT4,X
03AD: 86 EF     207           STX  XSAV1
03AF: 85 2E     208           STA  FORMAT
03B1: 29 03     209           AND  #3
03B3: 85 2F     210           STA  LENGTH
03B5: A9 0D     211           LDA  #13        ;COLM.-8
03B7: 85 FD     212           STA  COUNT
03B9: A9 52     213           LDA  #$52       ;§-CODE
```

```
03BB: 20 D3 F8  214           JSR  INSTDSP+3
03BE: A6 EF     215           LDX  XSAV1
03C0: BD CD 95  216           LDA  MNEML1,X
03C3: 85 2C     217           STA  LMNEM
03C5: BD DE 95  218           LDA  MNEMR1,X
03C8: A2 03     219           LDX  #3
03CA: 4C F3 F8  220           JMP  NXTCOL-2
                221
                222 RTEND
                223
                224 * Tabellen, die in PAGE $95
                225 * stehen:
                226
                227           ORG  ROUT+RTEND-EXTLIST
                228
                229 * Tabelle für die Länge der
                230 * NOP-Befehle
                231
95B8: 81 81 00  232 FMT3      HEX  81810082
95BB: 82
                233
                234 * Tabelle für die Adressierungs-
                235 * arten:
                236
95BC: 81 81 81  237 FMT4      HEX  818181     ;$XX
95BF: 4D        238           HEX  4D         ;($XX),Y
95C0: 81        239           HEX  81         ;$XX
95C1: 21 21 21  240           HEX  2121212121 ;#$XX
95C4: 21 21
95C6: 82        241           HEX  82         ;$XXXX
95C7: 86        242           HEX  86         ;$XXXX,Y
95C8: 21 21     243           HEX  2121       ;#$XX
95CA: 92        244           HEX  92         ;$XXXX,X
95CB: 82 82     245           HEX  8282       ;$XXXX
                246
                247 * Tabelle für die zusätzlichen
                248 * Mnemonics:
                249
95CD: 7C 7C 7C  250 MNEML1    HEX  7C7C7C     ;NOP
95D0: 18        251           HEX  18         ;BAX
95D1: 7C        252           HEX  7C         ;NOP
95D2: 13 13     253           HEX  1313       ;AND
95D4: 6D        254           HEX  6D         ;LSA
95D5: 9C        255           HEX  9C         ;ROA
95D6: 6E        256           HEX  6E         ;LXI
95D7: A0        257           HEX  A0         ;SAX
95D8: 16        258           HEX  16         ;AXS
95D9: 16        259           HEX  16         ;AXM
95DA: A0        260           HEX  A0         ;SBC
95DB: A0        261           HEX  A0         ;SBY
95DC: A2        262           HEX  A2         ;SHX
95DD: 4E        263           HEX  4E         ;HXA
                264
95DE: 22 22 22  265 MNEMR1    HEX  222222     ;NOP
95E1: B2        266           HEX  B2         ;BAX
95E2: 22        267           HEX  22         ;NOP
95E3: CA CA     268           HEX  CACA       ;AND
95E5: 04        269           HEX  04         ;LSA
95E6: 04        270           HEX  04         ;ROA
95E7: 54        271           HEX  54         ;LXI
95E8: B2        272           HEX  B2         ;SAX
95E9: 68        273           HEX  68         ;AXS
95EA: 5C        274           HEX  5C         ;AXM
95EB: C8        275           HEX  C8         ;SBC
95EC: F4        276           HEX  F4         ;SBY
95ED: 72        277           HEX  72         ;SHX
95EE: 44        278           HEX  44         ;HXA
                279
                280 * Tabelle der Codes mit den
                281 * Mnemonics von oben:
                282
95EF: 82 C2 E2  283 EXTBL     HEX  82C2E2     ;NOP
95F2: 93        284           HEX  93         ;BAX
95F3: 89        285           HEX  89         ;NOP
95F4: 0B 2B     286           HEX  0B2B       ;AND
95F6: 4B        287           HEX  4B         ;LSA
95F7: 6B        288           HEX  6B         ;ROA
95F8: 8B        289           HEX  8B         ;LXI
95F9: 9B        290           HEX  9B         ;SAX
95FA: BB        291           HEX  BB         ;AXS
95FB: CB        292           HEX  CB         ;AXM
95FC: EB        293           HEX  EB         ;SBC
95FD: 9C        294           HEX  9C         ;SBY
95FE: 9E        295           HEX  9E         ;SHX
95FF: 9F        296           HEX  9F         ;HXA
                297
                298 DOSBUF

312 Bytes
```