ESSENTIAL DATA DUPLICATOR

4

SOURCE CODE LISTING

UTILICO MICROWARE
NOTICE

Utilico Microware reserves the right to make improvements in the product described in this manual at any time and without notice.

DISCLAIMER OF WARRANTIES AND LIABILITIES

Utilico Microware makes no warranties, either expressed or implied, with respect to this manual or with respect to the software described in this manual, its quality, performance, merchantability, or fitness for any particular purpose. Utilico Microware software and manuals are sold "AS IS". The entire risk as to the quality and performance is with the buyer. Should this manual or software described in this manual prove defective following its purchase, the buyer (and not Utilico Microware, its distributor, or its retailer) assumes the entire cost of all necessary servicing, repair, or correction and any incidental or consequential damages. In no event will Utilico Microware be liable for direct, indirect, incidental, or consequential damages resulting from any defect in the software or this manual, even if Utilico Microware has been advised of the possibility of such damages. Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you.

(C) Copyright 1986 by Utilico Microware
All Rights Reserved

This document may not, in whole or part, be copied, photocopied, reproduced, translated, entered in, or reduced to any electronic medium or machine readable form without prior consent, in writing, from Utilico Microware.

UTILICO MICROWARE
3377 Solano Avenue, Suite 352
Napa, California 94558
(707) 257-2420

**  **  **
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>1</td>
</tr>
<tr>
<td>Plus Card Interfacing</td>
<td>3</td>
</tr>
<tr>
<td>Memory Allocation</td>
<td>5</td>
</tr>
<tr>
<td>EDD Source Code Listing</td>
<td>7</td>
</tr>
<tr>
<td>Main Program Module</td>
<td>7</td>
</tr>
<tr>
<td>Option 4 &amp; 5 Module</td>
<td>33</td>
</tr>
<tr>
<td>Option 3 &amp; 7 Module</td>
<td>37</td>
</tr>
<tr>
<td>Analyze Routines</td>
<td>43</td>
</tr>
<tr>
<td>Control Routines</td>
<td>53</td>
</tr>
<tr>
<td>EDD 4 Text</td>
<td>61</td>
</tr>
<tr>
<td>Program Buffers &amp; Parameters</td>
<td>71</td>
</tr>
<tr>
<td>Drive Routines</td>
<td>81</td>
</tr>
</tbody>
</table>

**   **   **
Written by Donald Anthony Schnapp using an apple IIe computer, Essential Data Duplicator 4 is a powerful computer program for allowing the user to make back up copies of his "copy-protected" software. The 6502 assembler which was used to enter EDD 4 is named "BIG MAC", which is similar to the "MERLIN" assembler written by Glen Bredon (available from Roger Wagner Publishing Inc.).

The main purpose of us providing this listing, is to let you see the "insides" of a sophisticated 6502 computer program for educational purposes. Buried in this listing are subroutines for managing routines, text output routines, hi-resolution graphic output routines, multiply and divide routines, interfacing disk drives routines, plus many more. Perhaps our routines will give you ideas for creating your own.

The section on the PLUS card explains how it works and how to interface it with your own 6502 programs.

Since every "batch" of EDD disks are programmed differently, this listing may not be an exact listing of an EDD disk which you may own. For reference purposes, this source code listing was printed out from the batch of EDD 4 disks recorded on April 23, 1986.

**  **  **

\[Signature\]
The purpose of the PLUS card is to read the bits of data coming from the disk drive and to make them available to computer software. This is done by first taking eight bits of data from the raw bit stream coming from the disk drive, combining them to form a raw disk byte, at which time a flag is set. Software detects the flag, reads the raw disk byte, at which time the flag is cleared. And the whole process is repeated until all the information is gathered. Using this process, every bit of data coming from the raw bit stream can be accurately read into the computer's memory.

The PLUS card has two valid memory locations in which software can access. They are $C080,X and $C081,X where "X" equals the slot number times $10. For example, if the PLUS card has been installed in slot #5, "X" would need to equal "$50" or if slot #7 was used, "X" would need to equal "$70".

A description of EDD PLUS card memory locations are as follows:

$C080,X contains the 8 bit raw disk byte.
(only valid if READY FLAG is minus)

$C081,X contains the READY FLAG.

Here is a sample routine, which takes one byte from the disk drive (note: drive must already be ON, in the read mode, and the read/write head must be located on the track which needs to be read):

```assembly
LDX #$50 ;Slot number of PLUS card.
chkready LDA $C081,X ;See if READY FLAG is set.
BPL CHKREADY ;No, go back & check again.
LDA $C080,X ;Yes, take the byte.
```

To go one step further, here is a routine which takes $2000 bytes from the disk drive and stores them in memory locations from $6000 through $7FFF (note: disk drive must already be set up as before):

```assembly
LDY #$60 ;Set up memory
STY $1 ;pointers to start
LDY #$0 ;storing at address $6000.
STY $0 ;
LDX #$50 ;Slot number of PLUS card.
chkready LDA $C081,X ;See if READY FLAG is set.
BPL CHKREADY ;No, go back & check again.
LDA $C080,X ;Yes, take the byte.
STA ($0),Y ;Store the byte in memory.
```
INY          ;Move pointer to point
BNE CHKREADY ;at next memory byte.
INC $1       ;If not at mem $8000 yet,
BPL CHKREADY ;go back and do again.

The first example below, is what a raw bit stream looks like when a "normal" timing gap containing $FF disk bytes with 2 timing bits attached is read by the disk drive. Below this, is what the disk bytes look like after they pass through a "standard" apple drive controller which "strips off" all the timing bits. Lastly, is what the raw disk bytes look like after they pass through the PLUS card (from the raw bit stream), which keeps all bits intact:

Raw bit stream coming from disk drive:
1111111100111111110111111111001111111111111111111111

Raw disk bytes read using the apple drive controller:
FF FF FF FF FF FF
11111111 11111111 11111111 11111111 11111111 11111111

Raw disk bytes read using the PLUS card:
FF FF 3F CF F3 FC FF
11111111 00111111 11001111 11100111 1111100 1111111

Since raw disk bytes read using the PLUS card are not usually very useful in their "raw" format, you will often need to create a routine which separates the data bytes from the timing bits. The routine which EDD uses to read the disk and separate the timing bits is called "DCCDUMP", and is contained in the "DRIVE ROUTINES" listing.

Although it is usually not very useful, it is, of course, possible to read in a whole buffer full of raw disk bytes, and then write them out without ever decoding or analyzing them.

ADDITIONAL PLUS CARD PROGRAMMING INFORMATION:

When writing programs for use with the PLUS card, keep in mind that one raw disk byte will be ready approximately every 32 cycles (one bit in 4 cycles equals 8 bits in 32 cycles). If your routine doesn't read the raw disk byte shortly after the READY flag is set, you will miss it. KEEP YOUR LOOPS SHORT!

The book named "Beneath Apple DOS" has all of the detailed information needed for accessing the disk drive through the drive controller.

**  **  **
$0000 - $00FF = ZERO PAGE VARIABLES
$0100 - $01FF = STACK POINTER
$0200 - $03FF = MISC. POINTERS/VALUES
$0400 - $07FF = EDD LOADER
$0800 - $0BFF = TEXT DISPLAY SCREEN
$0C00 - $17C2 = MAIN EDD ROUTINES
$17C3 - $1CFF = CHANGE PARAMETERS & BACK UP A DISK MODULE
$1D00 - $20FF = SCAN DISK & CERTIFY A DISK OPTIONS MODULE
$2100 - $25FF = DRIVE SPEED & DRIVE EXAMINE OPTIONS MODULE
$2600 - $2C29 = ANALYZE A DISK MODULE
$2C2A - $2FFF = CONTROL MODULE
$3000 - $3FFF = TEXT TABLES & ADDRESSES
$4000 - $77FF = RAW DISK BYTE BUFFER
$7800 - $AFFF = TIMING BIT BUFFER
$B000 - $B0FF = PREANALYZE ROUTINE BUFFER
$B100 - $B1FF = PREWRITE ROUTINE BUFFER
$B200 - $B2FF = PROGRAM VARIABLE STORAGE BUFFER
$B300 - $B3FF = PARAMETER BUFFER
$B400 - $B4FF = PARAMETER BUFFER (reserved)
$B500 - $B6FF = EXPANSION SPACE
$B700 - $BFFF = DISK DRIVE ROUTINES MODULE

**  **  **
**ESSENTIAL DATA DUPLICATOR**

**VERSION 4.2 STANDARD/PLUS**

**6502 ASSEMBLY SOURCE CODE**

**COPYRIGHT (c) 1984**

**All Rights Reserved**

**UTILICO MICROWARE**

**DONALD ANTHONY SCHNAPP**

**PRINTED APRIL 23, 1986**

---

**MEMORY ALLOCATIONS**

- PAGE = 0000
- EDD4 = 0000
- CERTDISK = 0100
- SCANDISK = 0200
- CHECKMEM = 0200
- DROEXAM = 0200
- DRVSPEED = 0200
- ANALYZE = 0200
- CONTROL = 0200
- TXTABLE = 0200
- PREALN = 0200
- CONTRLP = 0200
- VAR = 0200
- PAR = 0200
- PARMSET = 0200
- DRVR = 0200
- * zero page usage *

---

**VARIABLES**

- LTS = VAR+00
- LIT = VAR+01
- LIP = VAR+02
- LTP = VAR+03
- LTE = VAR+04
- LTV = VAR+05
- LTV = VAR+06
- LG = VAR+07
- LG = VAR+08
- RX = VAR+09
- LX = VAR+0A
- RP = VAR+0B
- RX = VAR+0C
- RX = VAR+0D
- RX = VAR+0E
- RX = VAR+0F

---

Page 7
LUP = VAR$+72
LASTHP = VAR$+73
KVALUE = VAR$+74
TWINDTOP = VAR$+75
SRMBR = VAR$+76
SCP = VAR$+77
SCPB = VAR$+78
SCHYVLU = VAR$+79
SCRT = VAR$+7A
SCFLT = VAR$+7B
SCULPNT = VAR$+7C
PTANGE = VAR$+7D
POLPNTR = VAR$+7E
PDS = VAR$+7F
POD = VAR$+80
DDD = VAR$+81
DDS = VAR$+82
DDO = VAR$+83
DDC = VAR$+84
DRYCOUNT = VAR$+85
CTRKO = VAR$+86
CTR = VAR$+87
STARTK = VAR$+88
ENDTRK = VAR$+89
STEP = VAR$+8A
TRACK = VAR$+8B
SYNCFLG = VAR$+8C
NBLF = VAR$+8D
TIMEFLG = VAR$+8E
DRVLETTR = VAR$+8F
TLEN = VAR$+90
TLENH = VAR$+91
WRKPNTR = VAR$+92
WS1 = VAR$+93
WS2 = VAR$+94
WSH = VAR$+95
WSH2 = VAR$+96
WRKPENTR2 = VAR$+97
WS2 = VAR$+98
WSH2 = VAR$+99
EP = VAR$+9A
EPF = VAR$+9B
ENP = VAR$+9C
ENHF = VAR$+9D
EMF = VAR$+9E
EMHF = VAR$+9F
TLEN = VAR$+A0
TLENH = VAR$+A1
TLEN2 = VAR$+A2
TLEN3 = VAR$+A3
TLEN4 = VAR$+A4
TLEN5 = VAR$+A5
TLEN6 = VAR$+A6
TLEN7 = VAR$+A7
TLEN8 = VAR$+A8
TLEN9 = VAR$+A9
HI = VAR$+AA
M = VAR$+AB
W = VAR$+AC
W1 = VAR$+AD
W2 = VAR$+AE
W3 = VAR$+AF
W4 = VAR$+B0
W5 = VAR$+B1
W6 = VAR$+B2
W7 = VAR$+B3
W8 = VAR$+B4
W9 = VAR$+B5
W10 = VAR$+B6
W11 = VAR$+B7
W12 = VAR$+B8
W13 = VAR$+B9
W14 = VAR$+BA
W15 = VAR$+BB
W16 = VAR$+BC
W17 = VAR$+BD
W18 = VAR$+BE
W19 = VAR$+BF
W20 = VAR$+C0
W21 = VAR$+C1
W22 = VAR$+C2
W23 = VAR$+C3
W24 = VAR$+C4
W25 = VAR$+C5
W26 = VAR$+C6
W27 = VAR$+C7
W28 = VAR$+C8
W29 = VAR$+C9
W30 = VAR$+CA
*PARAMETERS*
SYNSTSL = PARM$+50
TIMEBITS = PARM$+51
SPCNTL = PARM$+52
ABSNT = PARM$+53
MINL = PARM$+54
MAXL = PARM$+55
PINCTL = PARM$+56
PDINTL = PARM$+57
PGAPHIN = PARM$+58
ERRORS = PARM$+59
TRES = PARM$+5A
MINBTEGLN = PARM$+5B
Page 8
* EDD4 COMMON ROUTINE HOOKS *

** COMMON ANALYZE ROUTINE HOOKS **

** COMMON DRIVE ROUTINE HOOKS **

** EDD 4 MANAGER **

ORG EDD4 ;$0C00

LC00: 4C 18 0C

LC03: 4C 0A 0B

LC06: 4C 0B 0D

LC09: 4C 06 0E

LC0C: 4C AA AC

LC0F: 4C 03 14

LC12: 4C 09 0D

LC15: 4C 00 18

EDD: LDY #0, INITIALIZE EDD

LDY SCYP, EDD

STY SCYP, EDD

STY TWINTOP, EDD

STA PARMSET,Y, PARMSET

INT 1, TO SCREEN

BNE E, DUMP

LDA #$0D51, DUMP

LDA #$0C55, SCREEN

LDA POS, CURRENT SLOT

ASLA

ASLA

LDA SCLT, TITLE

LDA SCLT, TITLE

LDA SCLT, TITLE

LDA SCLT, TITLE

LDA SCLT, TITLE

E0: BPL E0

LDA #$0C00

E1: BIT #$010

JSR #1, DISPLAY/PCH

JSR #33, MENU

JSR #31, TITLE

JSR #34, TITLE

JSR #35, TITLE

JSR #36, TITLE

JSR #37, TITLE

JSR #38, TITLE

JSR #39, TITLE

DE: JSR MTROFF, DOESC
**COMMON SUBROUTINES**

---

**KYPRMPT**

```
JMP K1
```

**KYGLF**

```
LDA #$10
STA TWINDTOP
```

**KYSUB**

```
JSR KYSUB
```

**KYPROMPT**

```
JMP K1
```

**KYVALUE**

```
LDA #VALUE
STA TWINDTOP
```

**KYPROMPT**

```
LDA #VALUE
STA TWINDTOP
```

**KYGLF**

```
LDA #$10
STA TWINDTOP
```

**KYSUB**

```
JSR KYSUB
```

**KYPROMPT**

```
JMP K1
```

**KYVALUE**

```
LDA #VALUE
STA TWINDTOP
```

**KYPROMPT**

```
LDA #VALUE
STA TWINDTOP
```

**KYGLF**

```
LDA #$10
STA TWINDTOP
```

**KYSUB**

```
JSR KYSUB
```

**KYPROMPT**

```
JMP K1
```

**KYVALUE**

```
LDA #VALUE
STA TWINDTOP
```

**KYPROMPT**

```
LDA #VALUE
STA TWINDTOP
```

**KYGLF**

```
LDA #$10
STA TWINDTOP
```

**KYSUB**

```
JSR KYSUB
```

**KYPROMPT**

```
JMP K1
```

**KYVALUE**

```
LDA #VALUE
STA TWINDTOP
```

**KYPROMPT**

```
LDA #VALUE
STA TWINDTOP
```

**KYGLF**

```
LDA #$10
STA TWINDTOP
```

**KYSUB**

```
JSR KYSUB
```

**KYPROMPT**

```
JMP K1
```

**KYVALUE**

```
LDA #VALUE
STA TWINDTOP
```

**KYPROMPT**

```
LDA #VALUE
STA TWINDTOP
```

**KYGLF**

```
LDA #$10
STA TWINDTOP
```

**KYSUB**

```
JSR KYSUB
```

**KYPROMPT**

```
JMP K1
```

**KYVALUE**

```
LDA #VALUE
STA TWINDTOP
```

**KYPROMPT**

```
LDA #VALUE
STA TWINDTOP
```

**KYGLF**

```
LDA #$10
STA TWINDTOP
```

**KYSUB**

```
JSR KYSUB
```

**KYPROMPT**

```
JMP K1
```

**KYVALUE**

```
LDA #VALUE
STA TWINDTOP
```

**KYPROMPT**

```
LDA #VALUE
STA TWINDTOP
```

**KYGLF**

```
LDA #$10
STA TWINDTOP
```

**KYSUB**

```
JSR KYSUB
```

**KYPROMPT**

```
JMP K1
```

**KYVALUE**

```
LDA #VALUE
STA TWINDTOP
```

**KYPROMPT**

```
LDA #VALUE
STA TWINDTOP
```

**KYGLF**

```
LDA #$10
STA TWINDTOP
```

**KYSUB**

```
JSR KYSUB
```

**KYPROMPT**

```
JMP K1
```

**KYVALUE**

```
LDA #VALUE
STA TWINDTOP
```

**KYPROMPT**

```
LDA #VALUE
STA TWINDTOP
```

**KYGLF**

```
LDA #$10
STA TWINDTOP
```

**KYSUB**

```
JSR KYSUB
```

**KYPROMPT**

```
JMP K1
```

**KYVALUE**

```
LDA #VALUE
STA TWINDTOP
```

**KYPROMPT**

```
LDA #VALUE
STA TWINDTOP
```

**KYGLF**

```
LDA #$10
STA TWINDTOP
```

**KYSUB**

```
JSR KYSUB
```

**KYPROMPT**

```
JMP K1
```

**KYVALUE**

```
LDA #VALUE
STA TWINDTOP
```

**KYPROMPT**

```
LDA #VALUE
STA TWINDTOP
```

**KYGLF**

```
LDA #$10
STA TWINDTOP
```

**KYSUB**

```
JSR KYSUB
```

**KYPROMPT**

```
JMP K1
```

**KYVALUE**

```
LDA #VALUE
STA TWINDTOP
```

**KYPROMPT**

```
LDA #VALUE
STA TWINDTOP
```

**KYGLF**

```
LDA #$10
STA TWINDTOP
```

**KYSUB**

```
JSR KYSUB
```

**KYPROMPT**

```
JMP K1
```

**KYVALUE**

```
LDA #VALUE
STA TWINDTOP
```

**KYPROMPT**

```
LDA #VALUE
STA TWINDTOP
```

**KYGLF**

```
LDA #$10
STA TWINDTOP
```

**KYSUB**

```
JSR KYSUB
```

**KYPROMPT**

```
JMP K1
```

**KYVALUE**

```
LDA #VALUE
STA TWINDTOP
```

**KYPROMPT**

```
LDA #VALUE
STA TWINDTOP
```

**KYGLF**

```
LDA #$10
STA TWINDTOP
```

**KYSUB**

```
JSR KYSUB
```

**KYPROMPT**

```
JMP K1
```

**KYVALUE**

```
LDA #VALUE
STA TWINDTOP
```

**KYPROMPT**

```
LDA #VALUE
STA TWINDTOP
```

**KYGLF**

```
LDA #$10
STA TWINDTOP
```

**KYSUB**

```
JSR KYSUB
```

**KYPROMPT**

```
JMP K1
```

**KYVALUE**

```
LDA #VALUE
STA TWINDTOP
```
1140: C0 A0 860: CMP #A0
1141: E9 C9 861: BEO LNS2
1142: D2 04 862: CMP #4
1143: 25 BF 863: CMP #A0
1144: 5A 30 864: CMP #3F
1145: 4E 29 865: AND #3F
1146: 50 23 B2 866: STA WS
1147: 5C 0C 867: STA (SAL),Y
1148: 5E 09 DE 868: LNS1 ORA WS
1149: 5F 0A 869: LNS2 DEY
1150: 60 6C 870: BNE LNL1
1151: 62 6E 871: SCODSPLY
1152: 64 0A 872: SCOV
1153: 66 04 873: LDA LNS2
1154: 68 05 15 874: JSR TOUT
1155: 6A 06 B2 875: STA SCNMBR
1156: 6C 00 876: ORA #80
1157: 6E 20 16 877: JSR COUT
1158: 70 53 15 878: PLA SCVTAB1
1159: 72 60 879: RTS
1160: 74 6A 880: GTDRV001
1161: 76 C4 881: LDA #D
1162: 78 05 11 882: JSR SCGTRV
1163: 7A 00 883: JSR SETDRV
1164: 7C 01 11 884: JSR SCDINSRT
1165: 7E 60 885: LDA BLANK
1166: 80 68 886: RTS
1167: 82 6A 887: GTDRV002
1168: 84 C4 888: LDA #D
1169: 86 0E 11 889: JSR SCDINSRT1
1170: 88 60 890: LDA BLANK
1171: 8A 6A 891: BLNKind
1172: 8C 08 11 892: JSR SETDRV
1173: 8E 00 893: JSR INTO
1174: 90 60 894: LDA SCDINSRT
1175: 92 68 895: SCGTRV
1176: 94 0F 896: STA DRVLETTR
1177: 96 02 897: LDA #9
1178: 98 05 15 898: JSR TOUT
1179: A0 05 00 899: JMP SS10
1180: A2 05 00 89A: JSR BLN
1181: A4 05 00 89B: JSR #0
1182: A6 05 00 89C: JSR SETDRV
1183: AD 05 00 89D: JSR #0
1184: AE 05 00 89E: JSR #D
1185: AF 05 00 89F: JSR #D
1186: B0 05 00 8A0: JSR SETDRV
1187: B1 05 00 8A1: JSR SETDRV
1188: B2 05 00 8A2: JSR SETDRV
1189: B3 05 00 8A3: JSR SETDRV
1190: B4 05 00 8A4: JSR SETDRV
1191: B5 05 00 8A5: JSR SETDRV
1192: B6 05 00 8A6: JSR SETDRV
1193: B7 05 00 8A7: JSR SETDRV
1194: B8 05 00 8A8: JSR SETDRV
1195: B9 05 00 8A9: JSR SETDRV
1196: BA 05 00 8AA: JSR SETDRV
1197: BB 05 00 8AB: JSR SETDRV
1198: BC 05 00 8AC: JSR SETDRV
1199: BD 05 00 8AD: JSR SETDRV
1200: BE 05 00 8AE: JSR SETDRV
1201: BF 05 00 8AF: JSR SETDRV
1202: C0 05 00 8B0: JSR SETDRV
1203: C1 05 00 8B1: JSR SETDRV
1204: C2 05 00 8B2: JSR SETDRV
1205: C3 05 00 8B3: JSR SETDRV
1206: C4 05 00 8B4: JSR SETDRV
1207: C5 05 00 8B5: JSR SETDRV
1208: C6 05 00 8B6: JSR SETDRV
1209: C7 05 00 8B7: JSR SETDRV
1210: C8 05 00 8B8: JSR SETDRV
1211: C9 05 00 8B9: JSR SETDRV
1212: CA 05 00 8BA: JSR SETDRV
1213: CB 05 00 8BB: JSR SETDRV
1214: CC 05 00 8BC: JSR SETDRV
1215: CD 05 00 8BD: JSR SETDRV
1216: CE 05 00 8BE: JSR SETDRV
1217: CF 05 00 8BF: JSR SETDRV
1218: D0 05 10 8C0: STA CSLT
1219: D2 06 8C1: STA DRV
1220: D4 06 8C2: STA DISK
1221: D6 06 8C3: STA A
1222: D8 06 8C4: STA B
1223: DA 06 8C5: STA C
1224: DC 06 8C6: STA D
1225: DD 05 8C7: RTS
1226: DE 0F B2 8C8: SCDDINSRT1
1227: EF 0A 8C9: LDA DRVLETTR
1228: F0 04 8CA: LDA TOUT
1229: F2 05 15 8CB: JSR SCDINDO
1230: F4 06 12 8CC: BCC SS14
1231: F6 05 05 8CD: JSR F
1232: F8 05 05 8CE: JSR F
1233: FA 05 05 8CF: JSR F
1234: FB 05 05 8D0: JSR F
1235: FD 05 05 8D1: JSR F
1236: FF 05 05 8D2: JSR F

Page 17
21F:  A0 8B B2 1032  LDA      TRACK
220:  4A  4A      1034  LDA      TRACK
223:  20 7C 0E 1036  JSR      CNVHAD
226:  AD 03 B2 1038  LDA CNVDEC+4
229:  90 93      1039  BCC    GNBS2
22C:  AD 80 00 1039  LDA    #BS0
22F:  80 04 B2 1040  STA CNVDEC+3
232:  90 93      1041  LDA    #BS4
235:  20 03 00E 1043  LSR    CNVDEC+4
238:  AD 8F B2 1044  LDA    CNVHEX
23B:  5A 0A      1045  ASLA   ASLA
23E:  4C 93 12 1048  JMP    GNBL
241:  A0 8B B2 1049  GETDEC  LDA    TRACK
244:  90 93      1049  LDA    #BS3
247:  60 06      1050  BNE    #DL1
24A:  16 01      1052  LDA    #BS4
24D:  8C 00      1053  JSR    COUT
24E:  6D 04      1054  JSR    KYPRMPT
251:  AD 03      1055  LDA    #BS3
254:  90 93      1056  LDA    #BS1
257:  D9 45 B2 1057  CMP    LQV1
25A:  88 06      1058  BNO   GDS1
25D:  E0 8F      1059  JMP    GDL1
25E:  4C 0F 12 1061  JMP    GDL1
262:  A0 8B B2 1062  GDS1  LDA    TRACK
265:  68 06      1063  LDA    #BS3
268:  90 93      1064  STA    TRACK
26B:  8C 00      1065  JSR    #BS4
26E:  90 93      1066  STA    TRACK
271:  8C 00      1067  JSR    #BS8
274:  90 93      1068  STA    TRACK
277:  8C 00      1069  JSR    #BS3
27A:  90 93      1070  STA    TRACK
27D:  8C 00      1071  JSR    GDL1
27E:  90 93      1072  STA    GDL1
281:  C0 8D      1073  GNBCHKD  CMP    #BS3
284:  F0 68      1074  BEQ    GNBD2
287:  90 93      1075  JMP    GDL1
28A:  6D 04      1076  CMP    GBS1
28D:  78 00      1077  CMP    GBS1
28E:  78 00      1078  CMP    GBS1
291:  6D 04      1079  CMP    GBS1
294:  78 00      1080  CMP    GBS1
297:  6D 04      1081  CMP    GBS1
29A:  78 00      1082  CMP    GBS1
29D:  6D 04      1083  CMP    GBS1
29E:  78 00      1084  CMP    GBS1
2A1:  A0 8B B2 1085  GETSYNC  LDA    #35
2A4:  90 93      1086  JSR    GETSYNO
2A7:  8C 02      1087  STY    SYNCFL6
2A9:  8C 02      1088  RTS
2AC:  A0 2D      1091  RTS
2AD:  8C 02      1092  RTS
2AF:  8C 02      1093  RTS
2B1:  8C 02      1094  RTS
2B3:  8C 02      1095  RTS
2B5:  8C 02      1096  RTS
2B7:  8C 02      1097  RTS
2B9:  8C 02      1098  RTS
2BA:  8C 02      1099  RTS
2BB:  8C 02      109A  RTS
2BC:  8C 02      109B  RTS
2BD:  8C 02      109C  RTS
2BE:  8C 02      109D  RTS
2BF:  8C 02      109E  RTS
2C0:  8C 02      109F  RTS
2C1:  8C 02      10A0  RTS
2C2:  8C 02      10A1  RTS
2C3:  8C 02      10A2  RTS
2C4:  8C 02      10A3  RTS
2C5:  8C 02      10A4  RTS
2C6:  8C 02      10A5  RTS
2C7:  8C 02      10A6  RTS
2C8:  8C 02      10A7  RTS
2C9:  8C 02      10A8  RTS
2CA:  8C 02      10A9  RTS
2CB:  8C 02      10AA  RTS
2CC:  8C 02      10AB  RTS
2CD:  8C 02      10AC  RTS
2CE:  8C 02      10AD  RTS
2CF:  8C 02      10AE  RTS
2D0:  AD 80 00 1040  JSR    TOUT
2D3:  90 93      1041  LDA    #BS4
2D6:  60 06      1042  BNE    #DL1
2D9:  8C 00      1043  JSR    KYPRMPT
2DA:  6D 04      1044  JSR    #BS1
2DB:  8C 00      1045  JSR    #BS8
2DE:  90 93      1046  STA    TRACK
2DF:  90 93      1047  STA    #BS4
2E2:  6D 04      1048  JSR    TRKOUT
2E5:  AD 80 00 1049  LDA    #BS4
2E8:  90 93      104A  LDA    #BS1
2EA:  6D 04      104B  JSR    #BS3
2EB:  AD 80 00 104C  LDA    #BS4
2EE:  90 93      104D  LDA    #BS1
2F1:  6D 04      104E  JSR    #BS8
2F4:  90 93      104F  LDA    #BS1
2F7:  6D 04      1050  JSR    #BS3
2F8:  90 93      1051  LDA    #BS1
2F9:  6D 04      1052  JSR    #BS8
2FA:  90 93      1053  LDA    #BS1
2FB:  6D 04      1054  JSR    #BS3
2FC:  90 93      1055  LDA    #BS1
2FD:  6D 04      1056  JSR    #BS8
2FE:  90 93      1057  LDA    #BS1
2FF:  6D 04      1058  JSR    #BS3

DLNGR STA DFSIGN
LDA #30
JSR COUT
BNE OLNGR
STA MJERR
FORCE GOOD
ERROR

OILNGR STA DFSIGN
LDA #38
JSR COUT
BNE OONUGR
STA MJERR
FORCE GOOD
ERROR

SCHLN STA US2
LDA #2
JSR COUT
BNE SCLN
LDA #2
JSR COUT
DISPLY A LINE
OF RAW
TRACK BYTES

SCS1 PLA
BNE SCL2
JR Z1
INC #4
INC #3
BNE SCL1
INC #2
INC #1
DISPLY IT ON
SCREEN
--End assembly--

4079 bytes
Errors: 0
LDA "X" ; UNSET
STA FLG ; FLAG

CES1 JSR EPOUT
BCC CELB

LDY #41
LDA FLG
BEG CES2

CES2 JSR TOUT
VRA PDONE

*---------------------* HGR2 BASE ADDRESS TABLE *
*---------------------*

LTVHRS DA $5DD0

LTHRS DFB $18,$00 ;00

LTSCPT DFB $18,$18,$3C

SYNC DFB 0
HGRVPOS DFB 0

--End assembly--
4737 bytes

Errors: 0
**ASM**

```
   1620  ****************************
   1621  * ESSENTIAL DATA DUPLICATOR *
   1622  * VERSION 4.2 STANDARD/PLUS *
   1623  * 6502 ASSEMBLY SOURCE CODE *
   1624  * COPYRIGHT (C) 1984 *
   1625  * ALL RIGHTS RESERVED *
   1626  * UTILICO MICROWARE *
   1627  * DONALD ANTHONY SCHNAPP *
   1628  * PRINTED APRIL 23, 1986 *
   1629  ****************************
   1630  
   1631  **FLAG DS \$2100-FLAG**
   1632  
   1633  2100: 4C 32 21
   1634  DREXAM JMP DEXAM
   1635  
   1636  2103: 20 0C 1F
   1637  DRSPEED JSR CHECKMEM ;MANAGER
   1638  LDR #21
   1639  JSR TOUT ;SPEED
   1640  JSR GTDRV01 ;OPTION
   1641  LDR #22
   1642  JSR SETAB1
   1643  JSR SCDINDO2
   1644  LDR #19
   1645  JSR TOUT
   1646  JSR PLY
   1647  STY SPDPLNTR ;SET
   1648  JSR SETBS ;CHECK
   1649  JSR CHRSPD ;SPEED
   1650  JSR PLYLNE ;DISPLAY
   1651  JMP DRL1 ;SPEED
   1652  DRL1
   1653  ****************************
   1654  * EXAMINE DRIVE OPTION *
   1655  ****************************
   1656  
   1657  2132: 20 0C 1F
   1658  DEXAM JSR CHECKMEM ;MANAGER
   1659  LDR #21
   1660  JSR TOUT ;EXAMINE
   1661  JSR GTDRV01 ;DRIVE
   1662  LDR #22
   1663  JSR SETAB1
   1664  JSR SCDINDO2
   1665  LDR #23
   1666  JSR TOUT
   1667  JSR PLY
   1668  JSR SSRCM ;R/W
   1669  JSR EXFSX ;ABILITY
   1670  JSR GSCSDF ;AVG
   1671  JSR SETBS ;FLUCT
   1672  JSR CTRSK ;SPEED
   1673  JSR SCDNET ;COUNT
   1674  JSR HSCSL ;TRK
   1675  JSR SCDV9 ;BLEED
   1676  JSR SDIPARM ;OF ARM
   1677  JMP DREXAM
   1678  DREXAM
   1679  ****************************
   1680  **END**
   1681  ;FIND
   1682  JSR CTRKW ;THE
   1683  JSR SDIPARM ;OF ARM
   1684  JSR SDIPARM ;OF ARM
   1685  JSR SDIPARM ;OF ARM
```

Page 37
2245: A9 00 841 EXMSPO D LDA #0 :EXAMINE
LDY #7
LDA HXF, Y
DRIVE SPEED TO GET AVERAGE
AND FLUCT
THEN DISPLAY BOTH

LDA EXL1
JMP CALCNL
AND #80F

LDA EXL2
JMP CALCNL
AND #80F

LDA EXL3
JMP CALCNL
AND #80F

LDA "EXIT"
CAS5  LDA  WSL2
INC  STA  HXL
INC  ADC  HXL
STA  HXL
ADC  HSL2
STA  HXL
ADC  HXL
STA  HXH

EXRDWT  LDA  #0  CHECK
STA  WRKPNTR
STA  WZPAGE2
WRITE
JMP  EXS3
JR  TRKURT
JR  TRKV1
JMP  EPBOUT
JR  RESULTS

EXL4  JSR  EXS2
JMP  TRKURT
JR  TRKV1
JMP  EPBOUT
JR  RESULTS

EXS2  LDA  WSL2
INC  STA  WPAGE2
WRITE
JSR  SETBDEX
BEQ  DCPOUT
JSR  TRKURT
JSR  TRKV1
JSR  EPBOUT

EXL5  JSR  EXS3
JSR  WPAGE2
BEQ  DCPOUT
DEC  WPAGE2
DEC  WRKPNTR2
JR  DCPOUT
JR  WAOUT
LDA  #"."
JSR  COUT

EXS3  LDA  WPAGE2
INC  STA  WPAGE2
JSR  WRKPNTR2
JSR  SETBDEX
BEQ  DCPOUT
JSR  TRKURT
JSR  TRKV1
JSR  EPBOUT

DCPOUT  JSR  EXS3
JSR  WPAGE2
JSR  WAOUT
LDA  #"."
JSR  COUT

SPDOUTFH  JSR  NMBPOUT2
JSR  SHTAB
JSR  NMBFOUT2
DISPLAY
LDA  #20
JSR  SHTAB
JSR  NMBFOUT2
HEX
LDA  #20
JSR  NMBFOUT2
NUMBER
SPL1  LDA  #20
JSR  SHTAB
JSR  NMBFOUT2
& FRCNTN
JSR  TOUT
JSR  SHTAB
JSR  RHTAB
HEN
JSR  TOUT
JSR  SHTAB
JSR  RHTAB
RPM*

SPDOUTFD  JSR  SHTAB
JSR  TRKURT
DECIMAL
LDA  WSL2
JSR  TRKURT
NUMBER
JSR  TRKDS
TERMS
JSR  CALC
JSR  TRKDS
SPEED
LDA  WSL2
JSR  TRKDS
DRIVER

CHKSPD  JSR  TRKURT
JSR  TRKDS
CHECK
JSR  CALC
JSR  TRKDS
DRIVE
LDA  WSL2
JSR  TRKDS
SPEED

SPDLINE  JSR  TRKDS
JSR  WSL2
DISPLAY
LDA  #"."
JSR  TRKDS
DRIVE
JSR  WSL2
JSR  TRKDS
SPEED
LDA  #25
JSR  TRKDS
IN A
LDA  #6
JSR  TRKDS
"LINE"
LDA  #8
JSR  TRKDS
FORMAT
LDA  #8
JSR  TRKDS
DIVDAXY
LDA  TEMP
LDA  TEMP

Page 40
20E8: D0 9C B2 265
20E9: 20 9D 2B 269
20EA: A9 43 2C 26F
20EB: 20 12 0C 275
20EC: A9 2A 2C 27B
20ED: 20 20 2C 27F
20EE: 20 12 0C 285
20EF: 20 0D 27 28B
20F0: 18 29 2F
20F1: 60 29 2F

20F2: A0 5E B2 299
20F3: 8C 55 305
20F4: A0 76 B2 299
20F5: 8C C6 B2 299
20F6: A0 00 B2 299
20F7: 8C EE B2 299
20F8: A0 67 B2 299
20F9: 8C C7 B2 299
20FA: A0 6B B2 299
20FB: 8C C0 B2 299
20FC: 38 40 299
20FD: 60 E9 05 299
20FE: 8C 4C B2 299
20FF: A0 67 B2 299

2100: 38 40 299
2101: 60 E9 05 299
2102: 8C 4C B2 299
2103: A0 67 B2 299
2104: 38 02 299
2105: 60 E9 05 299
2106: 8C 4C B2 299
2107: A0 67 B2 299
2108: 38 02 299
2109: 60 E9 05 299
210A: 8C 4C B2 299
210B: A0 67 B2 299
210C: 38 02 299
210D: 60 E9 05 299
210E: 8C 4C B2 299
210F: A0 67 B2 299

2110: 38 40 299
2111: 60 E9 05 299
2112: 8C 4C B2 299
2113: A0 67 B2 299
2114: 38 02 299
2115: 60 E9 05 299
2116: 8C 4C B2 299
2117: A0 67 B2 299

2118: 85 08 299

2119: 20 51 2A
211A: 25 5C 2A
211B: 20 06 2A
211C: 85 05 2A
211D: 85 07 2A
211E: 85 03 2A
211F: 85 01 2A

2120: D0 01 2A
2121: E6 0D 2A
2122: 09 31 2A
2123: E6 03 2A
2124: 09 31 2A
2125: EC 79 2A
2126: D0 01 2A
2127: 09 31 2A
2128: 18 0C 2A
2129: 00 24 2A
212A: 00 07 2A

Page 44
2965: D0 07 2C 629 BNE FTGS3
2966: EC 10 2C 630 CPX TGLL
2967: E0 01 2C 631 BCC FTGS5
2968: GE 10 2C 632 BEX TGLL
2969: 80 11 2C 633 STA TGLL
2970: AD 11 2C 634 STA TGAL
2971: 8D 04 2C 635 STA TRKDEL
2972: 6B 04 2C 636 STA TTDAH
2973: 6C 04 2C 637 STA TRKTEH
2974: 38 04 2C 638 SEC
2975: #38
2984: A5 01 B2 642 CMP #$80
2985: D0 80 B2 643 BNE FTGL3
2986: C9 B0 644 JMP F78L4
2987: 4C 19 29 646

2988: A9 02 2C 647 LDA #2
2989: 20 2A 2C 648 JSR CONTROL
2990: 09 04 2C 649 LDR #0
2991: A0 05 2C 64A JSR PATTERN
2992: 89 02 2C 64B LDA #1
2993: 20 CB 29 64C JSR FTMARKR
2994: 20 05 29 64D BCC TDPE2
2995: 20 B0 29 64E JSR CTRKMR
2996: 38 04 29 64F RTS
2997: 8C B2 650 TDPE2
2998: 01 B2 651 LDA #1
2999: A9 B2 652 STA TRKDEL
299A: 00 04 B2 653 RTS
299B: 40 04 B2 654

299C: A0 00 2C 661 LDY #0
299D: A0 00 2C 662 STA TRKDEH
299E: A5 01 B2 663 LDA #1
299F: B1 00 2C 664 STA TRKDEH
29A0: 10 04 B2 665 RTS
29A1: BF 00 2C 666

29A2: 00 F5 670 LDA #0
29A3: E0 01 C6 671 STA TRKTEH
29A4: C9 B0 673 LDA #1
29A5: CF B0 674 CMP #$80
29A6: 20 ED 675 BNE TDPE3
29A7: 1C 18 676
29A8: 2D 60 677

29A9: 84 00 2C 679 LDA #0
29AA: 85 01 B2 67A STA TRKDEH
29AB: 20 F8 67B LDA ($0),Y
29AC: 20 00 2C 67C STA TRKDEH
29AD: 4E 01 B2 67D LDA ($0),Y
29AE: 20 BC 67E LDA ($0),Y
29AF: 0C 00 AF 67F STA TRKDEH
29B0: 20 F1 680 LDA ($0),Y
29B1: 30 00 AF 681 STA TRKDEH
29B2: 20 08 682 LDA ($0),Y
29B3: 30 00 2C 683 STA TRKDEH
29B4: 4E 01 2C 684 LDA ($0),Y
29B5: 20 FF 685 LDA ($0),Y
29B6: 0C 00 2C 686 STA TRKDEH
29B7: 20 0A 687 LDA ($0),Y
29B8: 20 0A 2C 688 LDA ($0),Y
29B9: 20 05 09 689 STA TRKDEH
29BA: 4E 01 2C 690 LDA ($0),Y
29BB: 20 05 AF 691 LDA ($0),Y
29BC: 20 FF 692 LDA ($0),Y
29BD: 0C 00 2C 693 STA TRKDEH
29BE: 20 0A 694 LDA ($0),Y
29BF: 20 0A 2C 695 LDA ($0),Y
29C0: 20 05 09 696 STA TRKDEH
29C1: 20 05 AF 697 LDA ($0),Y
29C2: 20 FF 698 LDA ($0),Y
29C3: 0C 00 2C 699 STA TRKDEH
29C4: 20 0A 69A LDA ($0),Y
29C5: 20 0A 2C 69B LDA ($0),Y
29C6: 20 05 09 69C STA TRKDEH
29C7: 20 05 AF 69D LDA ($0),Y
29C8: 20 FF 69E LDA ($0),Y
29C9: 0C 00 2C 69F STA TRKDEH
29CA: 20 0A 6A LDA ($0),Y
29CB: 20 0A 2C 6AB LDA ($0),Y
29CC: 20 05 09 6AC STA TRKDEH
29CD: 20 05 AF 6AD LDA ($0),Y
29CE: 20 FF 6AE LDA ($0),Y
29CF: 0C 00 2C 6AF STA TRKDEH
29D0: 20 0A 6B LDA ($0),Y
29D1: 20 0A 2C 6B LDA ($0),Y
29D2: 20 05 09 6BC STA TRKDEH
29D3: 20 05 AF 6BD LDA ($0),Y
29D4: 20 FF 6BE LDA ($0),Y
29D5: 0C 00 2C 6BE STA TRKDEH
29D6: 20 0A 6C LDA ($0),Y
29D7: 20 0A 2C 6C LDA ($0),Y
29D8: 20 05 09 6BD STA TRKDEH
29D9: 20 05 AF 6C LDA ($0),Y
29DA: 20 FF 6C LDA ($0),Y
29DB: 0C 00 2C 6C STA TRKDEH
29DC: 20 0A 6D LDA ($0),Y
29DD: 20 0A 2C 6D LDA ($0),Y
29DE: 20 05 09 6D STA TRKDEH
29DF: 20 05 AF 6E LDA ($0),Y
29E0: 20 FF 6E LDA ($0),Y
29E1: 0C 00 2C 6E STA TRKDEH
29E2: 20 0A 6F LDA ($0),Y
29E3: 20 0A 2C 6F LDA ($0),Y
29E4: 20 05 09 6F STA TRKDEH
29E5: 20 05 AF 6F STA TRKDEH
29E6: 20 FF 6F LDA ($0),Y
29E7: 0C 00 2C 6F STA TRKDEH
29E8: 20 0A 70 LDA ($0),Y
29E9: 20 0A 2C 70 LDA ($0),Y
29EA: 20 05 09 70 STA TRKDEH
29EB: 20 05 AF 70 LDA ($0),Y
29EC: 20 FF 70 LDA ($0),Y
29ED: 0C 00 2C 70 STA TRKDEH
29EE: 20 0A 71 LDA ($0),Y
29EF: 20 0A 2C 71 LDA ($0),Y
29F0: 20 05 09 71 STA TRKDEH
29F1: 20 05 AF 71 STA TRKDEH
29F2: 20 FF 71 LDA ($0),Y
29F3: 0C 00 2C 71 STA TRKDEH
29F4: 20 0A 72 LDA ($0),Y
29F5: 20 0A 2C 72 LDA ($0),Y
29F6: 20 05 09 72 STA TRKDEH
29F7: 20 05 AF 72 STA TRKDEH
29F8: 20 FF 72 LDA ($0),Y
29F9: 0C 00 2C 72 STA TRKDEH
29FA: 20 0A 73 LDA ($0),Y
29FB: 20 0A 2C 73 LDA ($0),Y
29FC: 20 05 09 73 STA TRKDEH
29FD: 20 05 AF 73 STA TRKDEH
29FE: 20 FF 73 LDA ($0),Y
29FF: 0C 00 2C 73 STA TRKDEH
Page 48
**DOPSUBO: DO PARM JSR ORDER**

- **A/Y = HIGH/LOW = JSR TABLE**
- **X = KEY_ORDER PATRN (3213 213X)**

**EXAMPLES:**
- ORDER 1, 2, 3 = 0010 1010 ($2A)
- ORDER 3, 1, 2 = 1110 0000 ($E0)
- ORDER 2 ONLY = 0100 0000 ($40)

---

**DOPSUBO**

```assembly
DOPSUBO: STA $5
          STY $4
          STX ODPRTR ; ORDER
          LDA #2
          STA ODCTR ; @=LTAB
          LDA #2
          STA ODNTR ; Y=LTAB
          LDA ODPRTR
          ASL ODPRTR
          ASLA
          TAY
          LDA ($4),Y
          STA $0
          INY
          LDA ($4),Y
          STA $1
          JMP DOPD
          BCC ODPR
          STA ERRORCD
          DOPS1: DEC ODCTR
          BSR DOPDE
          DEC ODNTR
          BPL DOPL2
          BNE ERRORCD
          DOPDE: SEC
          LDA ERRORCD
          BMI DOPL1
          DOPD: RTS

; MANAGER
NCAUTO: LDA TRKDSL
          CLC
          ADC #$10 ; AUTO
          TAY
          CMP #0
          BNE ERRORCD
          CMP #5
          BNE ERRORCD
          LDA TRKTSH
          STA $1
          DPH
          BMI CHOP
          CREATE: LDA #$1
          STA $3
          LDA #$4
          STA $4
          CRTL2: LDA #$8
          CRTL1: JSR NXTIME
```

---

**Page 50**
**--End assembly--**

1577 bytes

Errors: 0
199 ****************************
200  * ESSENTIAL DATA DUPLICATOR*
201  * VERSION 4.2 STANDARD/PLUS*
202  * 2502 ASSEMBLY SOURCE CODE*
203  * COPYRIGHT (C) 1986*
204  * ALL RIGHTS RESERVED*
205  * UTILICO MICROWARE*
206  * DONALD ANTHONY SCHNAPP*
207  * PRINTED APRIL 23, 1986*
208  ****************************
209 ****************************
210  * CONTROL ROUTINES *
211  * $2C2A-$2FFF *
212  ****************************
213
214 RDPRP EQU $0
215 RDWRKP EQU $2
216 RTWRKP EQU $4
217 CPP EQU $6
218 CPRP EQU $8
219 ZWRKSPC EQU $FE
220
221 ****************************
222  * INSTRUCTION BYTES:*
223  *
224  * IXXXX XXXXX = DATA*
225  * 0000 0XXX = TIMING*
226  *
227  * $03 MATCH OR FIND ONE TIMING*
228  * BYTE DURING SEARCH*
229  *
230  * $10 ANLDONE; ROUTINE DONE*
231  *
232  * $20 START SUB FLAG (TCPSPU5)
233  * $21 RETURN SUB (TSUBCPP)
234  * IF SUB IS DONE, RESET*
235  * SUB FLAG*
236  *
237  * $30 SEARCH MODE, SET DATA*
238  * BUFF POINTER TO 4000*
239  *
240  * $31 SEARCH MODE, CONTINUE DATA*
241  * BUFF PTR FROM S/R END*
242  *
243  * $32 SEARCH MODE, SET DATA BUFF*
244  * POINTER TO S/R START*
245  *
246  * $35 REPLACE MODE, CONTINUE DATA*
247  * BUFF PTR FROM S/R END*
248  *
249  * $36 REPLACE MODE, SET DATA*
250  * BUFF PTR TO S/R START*
251  *
252  * $40 TRANSFER RDP TO WORK*
253  * $41 TRANSFER WORK TO RDP*
254  * $42 TRANSFER RDP TO STORAGE*
255  * $43 TRANSFER STORAGE TO RDP*
256  *
257  * $50 FIND NEXT INVALID BYTE*
258  * $51 FIND NEXT NON-$EF*
259  *
260  * $55 RPLC DATA WITH RANDOM VALD*
261  * $56 RPLC DATA WITH RAND INVLD*
262  * $57 RPLC DATA WITH RAND MINVLN*
263  * $58 RPLC DATA WITH ZERO*
264  *
265  * $60 ADD NXT 2 BYTES TO RDWRKP*
266  * $61 SUB NXT 2 BYTES FRM RDWRKP*
267  * $62 ADD NEXT TWO BYTES TO RDP*
268  * $63 SUB NEXT TWO BYTES FRM RDP*
269  *
270  * $70 SINGLE CHAR WILDCARD*
271  * $71 MULTIPLE CHAR WILDCARD*
272  * AND RESET RDP*
273  *
274  * $73 SET TIMING BYTE HIGH BIT*
275  * OR LOCATING TEND/LENGTH*
276  *
277  * $75 BREAK INTO MONITOR*
278  *
279  * $76 NON-COMMANDS ARE IGNORED*
280  *
281  ****************************
282
283 ORG CONTROL ;2C2A

Page 53
SUBBYT INC CPUP ;INSTRUCT
    LDA (CPUP),Y
    STA RDP+1
    RTS

SWLDCD JSR INCDT ;INSTRUCT
    JSR SETINC
    RTS

MILDCD LDA #0 ;INSTRUCT
    STA MIFLAG
    JRE INCDT
    RTS

STHIGH JSR INCDT2 ;INSTRUCT
    LDA (RTWRKP),Y
    ORA #$80
    STA (RTWRKP),Y
    JRE INCDT
    RTS

QUIT JMP $FF59 ;INSTRUCT

*COMMON ROUTINES*

RSETUP LDA #$40 ;SETUP
    LDA #$0 ;RAW
    STA RDP
    STA IMODE
    STA MIFLAG
    IRE ENDFTAG
    JSR TRPDWRK
    RTS

CLRMPFLG LDX #0 ;CLEAR
    STX MIFLAG
    RTS

SETINCF LDX #$80 ;SET
    BNE SIS1
    RTS

SETINCN LDX #$00 ;SET
    BNE SIS1
    RTS

INCDT3 LDX MNBFLAG ;INC
    BMI INCDT
    RTX MNBFLAG
    RIS I1
    RTS

INCDT2 LDX MNBFLAG ;INC
    BMI INCDT
    RTX MNBFLAG
    BEQ 8IS
    RTS

INCDT LDX #0 ;INC
    RIS I1
    RTS

IL1 BEQ 8I1
    RTS

IS1 BEQ 8I1
    RTS

HITBEND LDX #1 ;INC
    RTX MNBFLAG
    RTX MNBFLAG
    RTS

Page 57
LDX #4F
RTSRKP+1

LDX MPFLAG
BNE PATSL
PATRN

INC MPFLAG
SET &

LDX RDWRKP
TRNRK

STX RDP
RDRK

LDX RDWRKP+1
TO RDP

JMP PATSL

TRA &R

LDA MNBNAG
STA RDWRKP
TRANSFER

LDA RDP
RDP TO

STA RDWRKP
DATA

LDA RDP+1
WORK

STA RDWRKP+1
AND

CLC
CALC

DPL
TIME

STA RTWRKP+1
WORK

LDA RTWRKP
CALC

STA RTWRKP+1
WORK

STA RTWRKP+1
RTSRKP+1

LDA RTWRKP
DATA

LDA RTWRKP+1
WORK

STA RDWRKP
FROM

STA RDWRKP+1
WORK

LDA RDWRKP+1
WORK

LDX CPP
TRANSFER

STX CPP+1
CNTRL

LDX CPP+1
WORK

STX CPP+1
TO CPP

LDX RDP
INSTRUCT

STX STORAGEL
#42

STX RDP+1
TRANSFER

STX STORAGEH
RDP TO

STX RDP+1
STORAGE

LDA STORAGEL
INSTRUCT

LDA STORAGEH
TRANSFER

LDX STORAGEH
TRANSF

LDX STORAGEH
TSTRDRP

RTS

LDX RDP
RTO

IMODE DFB 0

MPFLAG DFB 0

MNBNAG DFB 0

MNBFLAG DFB 0

MNBFLAG DFB 0

STORAGEH DFB 0

STORAGEL DFB 0

SUBPNTR DFB 0

SUBPNTR DFB 0

RNDMVLD DFB 0

RNDMVLD DFB 0

RNDMVLD DFB 0

RNDMVLD DFB 0

RNDMVLD DFB 0

* LOOK UP TABLE FOR COMMANDS *

*-----------------------------*

LTCINST DFB $10
ANLNDONE

DEB $20
STRSUB

DEB $20
STRSUB

DEB $20
STRSUB

DEB $20
STRSUB

DEB $20
STRSUB

*-----------------------------*

Page 58
### Lookup Table for Command

<table>
<thead>
<tr>
<th>Address</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>00 0D</td>
<td>ANLDONE</td>
</tr>
<tr>
<td>01 20</td>
<td>STRTSUB</td>
</tr>
<tr>
<td>02 2D</td>
<td>RTNRTSUB</td>
</tr>
<tr>
<td>03 21</td>
<td>RTNRTD</td>
</tr>
<tr>
<td>04 22</td>
<td>GSTMOD0</td>
</tr>
<tr>
<td>05 23</td>
<td>GSTMOD1</td>
</tr>
<tr>
<td>06 24</td>
<td>GSTMOD2</td>
</tr>
<tr>
<td>07 25</td>
<td>GSTMOD3</td>
</tr>
<tr>
<td>08 26</td>
<td>CTRNTOP</td>
</tr>
<tr>
<td>09 27</td>
<td>TCFLVLD</td>
</tr>
<tr>
<td>0A 28</td>
<td>TCFZERD</td>
</tr>
<tr>
<td>0B 29</td>
<td>TCFADDW</td>
</tr>
<tr>
<td>0C 2A</td>
<td>TCFSUBYT</td>
</tr>
<tr>
<td>0D 2B</td>
<td>TCFADDY</td>
</tr>
<tr>
<td>0E 2C</td>
<td>LFMLDCD</td>
</tr>
<tr>
<td>0F 2D</td>
<td>LFSRHIGH</td>
</tr>
<tr>
<td>10 2E</td>
<td>LFQUIT</td>
</tr>
</tbody>
</table>

### Lookup Table of Invalid Disk

<table>
<thead>
<tr>
<th>Address</th>
<th>Byte Patterns</th>
</tr>
</thead>
<tbody>
<tr>
<td>07 0E</td>
<td>01C3870E080</td>
</tr>
</tbody>
</table>

### Lookup Table for Replacing

<table>
<thead>
<tr>
<th>Address</th>
<th>Hexadecimal</th>
</tr>
</thead>
<tbody>
<tr>
<td>96 06</td>
<td>9E6979A9B9D9E9FA6</td>
</tr>
<tr>
<td>97 07</td>
<td>A7A7BACADAEAFB2B3</td>
</tr>
<tr>
<td>98 08</td>
<td>B4B5B4B7B9BABBBC</td>
</tr>
<tr>
<td>99 09</td>
<td>BDBEBFBCDCD0CE5D</td>
</tr>
<tr>
<td>9A 0A</td>
<td>D67D9DADBDCDDDE</td>
</tr>
<tr>
<td>9B 0B</td>
<td>DFE5E67E9EAE8BC</td>
</tr>
<tr>
<td>9C 0C</td>
<td>EDEEEFF2F3F4F5F6</td>
</tr>
<tr>
<td>9D 0D</td>
<td>F7F9FAFBCFDFEEFF</td>
</tr>
<tr>
<td>9E 0E</td>
<td>AAD5</td>
</tr>
</tbody>
</table>

### Look Up Table for Replacing

<table>
<thead>
<tr>
<th>Address</th>
<th>Hexadecimal</th>
</tr>
</thead>
<tbody>
<tr>
<td>9F 0F</td>
<td>AAD5</td>
</tr>
</tbody>
</table>

---

Page 59
2FCB: 80 81 82 83 84 85 86 87 88 89 8A 8B 8C 8D 8E 8F 90 91 92 93 94 95 96 97 98 99 9A 9B 9C 9D 9E 9F A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 AA AB AC AD AE AF B0 B1 B2 B3 B4 B5 B6 B7 B8 B9 BA BB BC BD BE BF C0 C1 C2 C3 C4 C5 C6 C7 C8 C9 CA CB CC CD CE CF D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 DA DB DC DD DE DF E0 E1 E2 E3 E4 E5 E6 E7 E8 E9 EA EB EC ED EE EF F0 F1 F2 F3 F4 F5 F6 F7 F8 F9 FA FB FC FD FE FF

LTRINVLD HEX 8081828384858687

---End assembly---

981 bytes

Errors: 0
TXMNU ASC "&J" - OPTION MENU -J"

1. BACK UP A DISK*?

2. CHANGE PARAMETERS*

3. CHECK DRIVE SPEED*

4. DISK SCAN*

5. CERTIFY AND ERASE DISK*

6. CHANGE SLOTS AND DRIVES*

7. EXAMINE DISK DRIVE*

8. CLEAR ERROR CODES*

9. QUIT^?

TX00 ASC "_ PRESS <"

RETURN

> TO SELECT #[?]  ""

D" OR ""

>ORIGINAL OR ""

>DUPLICATE DRIVE? PRESS <"

RETURN

> FOR ?  ""

INSERT ?"
ASC "&j] - CHANGE PARAMETERS -"

ASC "&j] 1. CHANGE PARAMETER VALUES"

ASC "&j] 2. REPROGRAM PREANAYLZE ROUTINE"

ASC "&j] 3. REPROGRAM PREWRITE ROUTINE"

ASC "&j] 4. RESET PARAMETERS TO DEFAULT"

ASC "&j] 5. RETURN TO OPTION MENU?

ASC "&j] REFER TO YOUR EDD DOCUMENTATION AND"

ASC "&j] PROGRAM INFORMATION LISTS IF NECESSARY.

ASC "&j] PRESS (Q) TO QUIT."

ASC "&j] CURRENT NUMBER /"
CURRENT VALUE"^?"

"CHANGE"

"CHANGE"

"CHANGE"

">>> PARAMETERS HAVE BEEN RESET <<<[

">>> TO THEIR DEFAULT VALUE <<<[

"FLOPPY DISK UTILITIES -"

"1. DISK SCAN"

"2. CERTIFY AND ERASE DISK"

"3. RETURN TO OPTION MENU"^?"

"DISK DRIVE UTILITIES -[

"1. CHECK DRIVE SPEED"

"2. EXAMINE DRIVE"

"3. RETURN TO OPTION MENU"^?"
"SPEED IS: ?"

"RPM[?"

"LOOPS?"

"&]] - EXAMINE DISK DRIVE -]]]

"READ/WRITE TRACK ABILITY =]

"AVERAGE DRIVE SPEED =]

"DRIVE SPEED FLUCTUATION =]

"HIGHEST TRACK ACCESSIBLE =]

"QUARTER TRACK BLEED OVER =]

"MINIMUM ARM PHASE TIME =]^?"

"EXAMINING THE ?"

"(THIS OPTION TAKES ABOUT 60 SECONDS)[?"

"PRESS <"

"RETURN"

">, PROCESS IS "

"FINISHED"

"[?"

"WRITE-PROTECT"

"THE "

"ORIGINAL"

"DISK[?"

"WRITE-PROTECT FROM "

"REMOVE"
3CEF: A6 4C 41 275 FLS "BLANK"
3CEF: A6 4C 41 276 ASC " DISK[?"
3CF8: D3 C8 C6 DB 277
3CFC: A6 DD DD 278 FXSC ASC "&?]" - DISK SCAN -^?
3D01: A6 00 00 00 280 XXCE ASC "&?]" - CERTIFY AND ERASE DISK -^?
3D25: A6 00 00 00 282 TXCE6 ASC " " SELECTED TRACKS CERTIFIED O.K.[?"
3D46: AF 4C 4C 4D 283 ASC " " BAD DISK; CERTIFY HAS "
3D49: E8 00 00 00 284 ASC " " BACK UP A DISK -^?
3D4C: A6 DD DD 288 TXCPYDISK ASC "&?]" - BACK UP A DISK -^?
3D54: A6 00 00 00 289 ASC " " TRACK START: (RAW DISK BYTES)"?
3D61: AF 00 00 00 290 ASC " " TRACK END: (RAW DISK BYTES)"?
3D6E: A6 00 00 00 291 ASC " " TRACK: TRACK LENGTH:"?
3D7B: A6 00 00 00 292 ASC " " DIFFERENCE:[?"
3D84: A6 DD DD 293 PCSER ASC "&?]"
3D89: 4C 55 296 FLS "PLUS"
3D94: A6 41 52 297 ASC " " FLS "CARD"
3D9F: A6 41 52 298 ASC " " FLS "ERROR;"
3DB7: A6 DD DD 300 ASC " } EJDER THE EDD PLUS CARD IS NOT]"
3DBE: A6 AF CC 302 ASC " LOCATED IN SLOT# , OR THE PLUS CARD]"
3DF4: A6 AF CC 303 ASC " IS NOT CONNECTED TO THE "

Page 69
INV "ORIGINAL"
ASC "DISK"
ASC "DRIVE CONTROLLER LOCATED IN SLOT #."
ASC "REFER TO YOUR EDD MANUAL FOR PROPER"
ASC "PLUS CARD INSTALLATION - PRESSING"
ASC "<RETURN> WILL ENTER YOU INTO THE"
ASC "CHANGE SLOTS AND DRIVES OPTION. ^3"

End assembly
3952 bytes
Errors: 0
**PREANLYZE BUFFER**
**B000 - $B0FF**
**ORG $B000**

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Address</th>
<th>Hex Code</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>B000: 01</td>
<td></td>
<td>01</td>
<td></td>
</tr>
<tr>
<td>B001: 10</td>
<td></td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>B002: 20</td>
<td></td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>B003: 10</td>
<td></td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>B004: 20</td>
<td></td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>B005: 10</td>
<td></td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>B006: 20</td>
<td></td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>B007: 10</td>
<td></td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>B008: 20</td>
<td></td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>B009: 10</td>
<td></td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>B00A: 20</td>
<td></td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>B00B: 10</td>
<td></td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>B00C: 20</td>
<td></td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>B00D: 10</td>
<td></td>
<td>10</td>
<td></td>
</tr>
</tbody>
</table>

* $B0 = CHANGE INVALID TO FF'S*

* $90 = GENERIC BITSLIP PROTECTN 2
  USED ON SOME EPYX SFTW*

* $C8 = LOCK-IT-UP PROTECTION:
* PARAMETER BUFFER *
* $B300 - $B3FF *
*----------------------------------------------------------*
  **SPAGE** DS $B300**SPAGE

*----------------------------------------------------------*
  **PARMS $00-$08 = SYNTAX**
  **THESE 9 PARAMETERS MAKE UP THE**
  **RAW DISK BYTE PATTERN WHICH**
  **EDD USES FOR SEARCHING, TO**
  **SYNCHRONIZE EACH TRACK.**
  **THIS PATTERN MUST BE FOUND ON**
  **THE TRACK SPECIFIED BY THE**
  **"TRKSYNC" PARAMETER. (PARM$1B)**
  **WHICH HAS A DEFAULT VALUE OF**
  **TRACK 0.**
  **USING THE VALUE "?F" IN THIS**
  **TABLE, REPRESENTS A "WILDCARD"**
  **AND WILL MATCH ANY DISK BYTE.**
  **THIS TABLE HAS BEEN PRESET TO**
  **SYNCHRONIZE OFF OF SECTOR 0**
  **OF TRACK 0.**

  **B300: D5 AA 96**
  **B304: 7F AA AA**
  **B307: AA**
  **B308: AA**

  **SYNTAX** DFB $D5,$ AA,$96,$7F
  **DFB** $7F,$ AA,$ AA,$ AA
  **DFB** $ AA

* Parm $09 = TIMEBITS *
* THIS IS THE AMOUNT OF TIMING **
* BITS THAT A TIMING BYTE IS **
* GIVEN WHEN EDD READS THE TRACK **
* USING THE "NORMAL" MODE. SINCE **
* MANY DISKS USE TWO TIMING BITS **
* INSTEAD OF ONE, YOU MAY NEED **
* TO USE A VALUE OF "2" HERE, IF **
* THE COPY-PROTECTION IS **
* CHECKING TIMING BITS.

  **B309: 01**
  **B30A: 80**
  **B30B: 00**

  **TIMEBITS** DFB $1

* Parm $0A = SPECIAL CONTROL *
* THIS PARAMETER AFFECTS THE **
* AMOUNT OF BYTES WRITTEN FOR **
* EACH TRACK:

  **$00 = WRITE THE AMOUNT OF**
  **BYTES DISPLAYED FOR THE**
  **TRACK LENGTH.**

  **$80 = WRITE A FULL TRACK OF**
  **$1BFF BYTES WHEN POSSIBLE**

  **B30C: 80**

  **SPCLNTL** DFB $80

* Parm $0C-$0D = ABSLNGT *
* IF THE TRACK LENGTH IS TO **
* BE AN ABSOLUTE FORCED LENGTH.
303  * (SEE PARM $12), USE THESE TWO
306  * VALUES AS THE ABSOLUTE TRACK
309  * LENGTH.
311  *
312  B30C:  72
313  ABSLNGTH DFB $72
314  ABSLNGTH DFB $18
315  B30D: 00
316  DFB $0  !UNUSED
317  DFB $0  !UNUSED
318  *
319  PARM $10 = MINLENGTH
320  *
321  * THIS VALUE REPRESENTS THE
324  * HIGH BYTE OF THE LOWEST TRACK
327  * LENGTH ACCEPTABLE.
328  *
329  B310: 14
330  MINLENGTH DFB $14
331  *
332  PARM $11 = MAXLENGTH
333  *
334  * THIS VALUE REPRESENTS THE
337  * HIGH BYTE OF THE LARGEST
340  * TRACK LENGTH ACCEPTABLE.
341  *
342  B311: 1C
343  MAXLENGTH DFB $1C
344  *
345  PARM $12 = PLNGCNTL
346  *
347  * WHICH TRACK LENGTH ROUTINE(S)
348  * TO USE AND ORDER OF ROUTINE IF
349  * AN ERROR OCCURS:
350  *
351  * BIT POSITION: 7654 3210
352  * ROUTINE: 3213 213X
353  *
354  * ROUTINES AVAILABLE:
355  *
356  1 = DATA PATTERN
357  * LOCATE TWO MATCHING DATA
358  * PATTERNS (SEE PARMS $80-$FF) THEN SUBTRACT THEIR
359  * DISTANCE TO OBTAIN TRACK
360  *
361  2 = UNIQUE DATA PATTERN
362  * LOCATE A UNIQUE DATA
363  * PATTERN THEN SEARCH FOR A
364  * MATCHING PATTERN, THEN
365  * SUBTRACT THEIR DISTANCE
366  * TO OBTAIN TRACK LENGTH
367  (USES NO ADDITIONAL PARMS)
368  *
369  3 = ABSOLUTE FORCED LENGTH
370  * THE VALUES FOUND AT PARMS
371  * $1C-$0D ARE USED FOR THE
372  * TRACK LENGTH, NO TRACK
373  * SEARCHING IS NECESSARY.
374  *
375  * EXAMPLES:
376  * 0111 0000 = ORDER: 2,1,3
377  * 1000 0000 = 3 ONLY
378  *
379  * EDD HAS BEEN PRESET TO USE THE
380  * "UNIQUE DATA PATTERN" ROUTINE
381  * $40 = 01000000 = ROUTINE #2
382  *
383  B312: 40
384  PLNGCNTL DFB $40
385  *
386  PARM $13 = PENDCNTL
387  *
388  * WHICH TRACK END ROUTINE(S) TO
389  * USE AND ORDER OF ROUTINES IF
390  * AN ERROR OCCURS:
391  *
392  * BIT POSITION: 7654 3210
393  * ROUTINE: 3213 213X
394  *
395  * ROUTINES AVAILABLE:
396  *
397  1 = TIMING GAP
398  * POINT TO THE BYTE LOCATED
399  * JUST BEFORE THE 1ST BYTE
400  *
401  *
OF THE LARGEST TIMING GAP.

2 = DATA PATTERN

USE THE TRKSRC INSTRCTN

ROUTINE (PARAMETERS $80-

$FF) FOR LOCATING THE

TRACK END.

3 = DATA GAP

POIN T TO THE BYTE LOCATED

JUST BEFORE THE 1ST BYTE

OF THE LARGEST DATA GAP.

EXAMPLES:

0010 1010 = ORDER 1, 2, 3

0100 0000 = 2 ONLY

EDD HAS BEEN PRESET TO USE THE

"TIMING GAP" ROUTINE:

$20 = 00100000 = ROUTINE#1

PENDCNTL DFB $20

DFB $0 ; UNUSED

DFB $0 ; UNUSED

DFB $0 ; UNUSED

PARM $17 = PTGAPMIN

WHEN FINDING THE TRACK END BY

SEARCHING FOR THE LARGEST

TIMING GAP (SEE PARM $13) THIS

PARM CONTAINS THE MINIMUM

AMOUNT OF BYTES THE GAP MUST

CONTAIN. IF NOT, THE TIMING

ROUTINE FAILS.

PTGAPMIN DFB $2

DFB $0 ; UNUSED

PARM $19 = RERRORS

THE MAXIMUM AMOUNT OF ATTEMPTS

TO READ THE TRACK PROPERLY,

BEFORE A READ ERROR IS FORCED

RERRORS DFB $3

PARM $1A = WERRORS

THE MAXIMUM AMOUNT OF ATTEMPTS

OF WRITING THE TRACK PROPERLY,

BEFORE A WRITE ERROR IS FORCED

WERRORS DFB $3

PARM $1B = TRKSYNC

THIS IS THE TRACK TO SYNCHRONIZE

FROM WHEN USING THE SYNC TRACK

MODE.

THIS TRACK VALUE IS THE ACTUAL

QUARTER TRACK TO USE. (IE: IF

THE VALUE OF $04 IS USED, THIS

REPRESENTS 4 QUARTER TRACKS, &

TRACK #1 WILL BE THE "SYNC"

FROM TRACK.

TRKSYNC DFB $0

PARM $1C = MNBTQGLN

WHEN LOCATING THE TRACK LENGTH

- IF MORE THAN THIS AMOUNT OF

BYTES DON'T MATHC UP, THE

LENGTH IS CONSIDERED INVALID,

WHICH A READ ERROR OCCURS.

MNBTQGLN DFB $10

---------------------------------------------

Page 77
* PARM $80-$BF = TBLEND

* WHEN USING A RAW DISK BYTE
* PATTERN TO LOCATE THE TRACK'S
* END (PARM $13), THIS TABLE
* IS USED FOR THE RAW DISK BYTE
* PATTERNS.

* THIS TABLE IS USED IN THE SAME
* FASHION AS A PREANALYZE OR
* PREURITE ROUTINE (AS DESCRIBED
* IN THE EDD OPERATING MANUAL)
* THE MAIN DIFFERENCE HERE,
* IS THIS ROUTINE IS DONE
* DURING THE ANALYZE ROUTINE
* AND THE INSTRUCTION "$73" SETS
* A "FLAG" FOR FINDING THE END
* TRACK.

* PARAMETER $80 POINTS TO THE
* FIRST POSITION OF THE TABLE
* WHICH HAS A DEFAULT VALUE OF
* $81, MEANING THAT PARAMETER
* $81 IS THE FIRST INSTRUCTION
* BYTE FOLLOWED.

* THE PRESET ROUTINE STARTS AT
* PARM $81 AND IT FINDS THE
* TRACK END BY LOCATING A DATA
* FIELD, THEN POINTING TO THE
* BYTE LOCATED AFTER A DATA
* FIELD EPILOGUE.

* REFER TO THE "INSTRUCTION
* BYTES" SECTION OF THE EDD 4
* MANUAL FOR A COMPLETE DESCRIP-
* TION (INCLUDING SOME EXAMPLES)
* OF INSTRUCTION BYTES.

TBLEND DS *B380-TBLEND
HEX 81D5A83D71DEAA73
HEX 2110101010101010
HEX 1010101010101010
HEX 1010101010101010
HEX 1010101010101010
HEX 1010101010101010
HEX 1010101010101010

* PARM $C0-$FF = TRKLENGTH

* WHEN USING A RAW DISK BYTE
* PATTERN TO LOCATE THE TRACK'S
* LENGTH (PARM $12), THIS TABLE
* IS USED FOR THE RAW DISK BYTE
* PATTERNS.

* THIS TABLE IS USED IN THE SAME
* FASHION AS THE TBLEND TABLE
* ABOVE.

* PARAMETER $C0 POINTS TO THE
* FIRST POSITION OF THE TABLE
* WHICH HAS A DEFAULT VALUE OF
* $C1, MEANING THAT PARAMETER
* $C1 IS THE FIRST INSTRUCTION
* BYTE FOLLOWED.

* THE PRESET ROUTINE HERE,
* LOCATES THE ADDRESS FIELD
* OF SECTOR ZERO, THEN LOOKS
* FOR THE REPEAT, SUBTRACTS
* THE DISTANCE FROM THEIR
* POSITIONS TO CALCULATE TRACK
* LENGTH.

B3C0: C1 31 D5 602 TBLLENGTH HEX C131D5AA96707070
<table>
<thead>
<tr>
<th>Address</th>
<th>Value</th>
<th>Hexadecimal Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>B3C3:</td>
<td>AA 96 70 70 70</td>
<td>70AAAA3673211010</td>
</tr>
<tr>
<td>B3CB:</td>
<td>70 AA AA 603</td>
<td>1010101010101010</td>
</tr>
<tr>
<td>B3C9:</td>
<td>36 73 21 10 10</td>
<td>1010101010101010</td>
</tr>
<tr>
<td>B3D6:</td>
<td>10 10 10 10 10</td>
<td>1010101010101010</td>
</tr>
<tr>
<td>B3D3:</td>
<td>10 10 10 10 10</td>
<td>1010101010101010</td>
</tr>
<tr>
<td>B3D0:</td>
<td>10 10 10 10 10</td>
<td>1010101010101010</td>
</tr>
<tr>
<td>B3C7:</td>
<td>10 10 10 10 10</td>
<td>1010101010101010</td>
</tr>
<tr>
<td>B3EF:</td>
<td>10 10 10 10 10</td>
<td>1010101010101010</td>
</tr>
</tbody>
</table>

---End assembly---

1024 bytes

Errors: 0
************

* ESSENTIAL DATA DUPLICATOR *

* VERSION 4.2 STANDARD/PLUS *

* 6502 ASSEMBLY SOURCE CODE *

* COPYRIGHT (C) 1982 *

* ALL RIGHTS RESERVED *

* UTILICO MICROWARE *

* DONALD ANTHONY SCHNAPP *

* PRINTED APRIL 23, 1986 *

************

* DRIVE ROUTINES *

************

* JUMP TABLE *

************

8700: ORG B700

8701: DRVR ;D700

8704: B700: JMP TDUMPW

8705: B703: JMP TDUMPP

8706: B709: JMP ARMV

8707: B70C: JMP SYNTKR2

8708: B70F: JMP TRK1

8710: B712: JMP TRK2

8713: B716: JMP ARMSPD

8714: B718: JMP WRTETRKR

8715: B71E: JMP DCDUMP

8716: B723: JMP TDUMPV

8717: B729: JMP TRKOS

8718: B72B: JMP PCDCHK

8719: B72D: DCCDUMP

8720: LDA DCCSLOT

8721: DUMP A

8722: ASL A

8723: TRACK

8724: ASL A

8725: USING

8726: ASL A

8727: THE EDD

8728: ASL A

8729: PLUS

872A: CARD

872B: AD 84 B2

872C: DCCDUMP

872D: LDA DCCSLOT

872E: DUMP A

872F: ASL A

8730: TRACK

8731: ASL A

8732: THE EDD

8733: ASL A

8734: PLUS

8735: CARD

8736: LOOP1

8737: DB $40

8738: STA $40

8739: LDY $0

873A: STY $0

873B: LOOP1

873C: DB $7E

873D: BRQ DNDUMP

873E: CHKRDY

873F: LDA @081,X

8740: BPL CHKRDY

8741: LDA @080,X

8742: STA ($0),Y

8743: INY

8744: DNDUMP

8745: LDA $40

8746: CONVERT DUMPED

8747: STA $3

8748: TRACK

8749: STA $3

874A: DUMP INTO

874B: STA $9

874C: STABSTANDRD

874D: LDY $0

874E: EDD

874F: STY $4

8750: TRACK

8751: STY $4

8752: FORMAT

8753: STY $4

8754: LOOP3

8755: LDA ($4),Y

8756: LOOP2

8757: INC $1

8758: BNE LOOP3

8759: INC $2

875A: BNE LOOP2

875B: 50 00 02

875C: 53 00

875D: 56 00

875E: 59 02

875F: 5C 00

8760: DB 01 E

8761: STORE

8762: LDA ($0)

8763: BNE LOOP4

8764: STA ($6),Y

8765: STA ($2)

8766: STA ($8),Y

8767: STA $0

8768: STY $0

8769: STY $0

876A: STY $0

876B: STY $0

876C: STY $0

876D: STY $0

876E: STY $0

876F: STY $0

8770: STY $0

8771: STY $0

8772: STY $0

8773: STY $0

8774: STY $0

8775: STY $0

8776: STY $0

8777: STY $0

8778: STY $0

8779: STY $0

Page 81
<table>
<thead>
<tr>
<th>Address</th>
<th>Instruction</th>
<th>Immediate</th>
<th>Register</th>
<th>Accumulator</th>
<th>Stack</th>
<th>Other</th>
</tr>
</thead>
</table>
| B840   | A6 16      | 491      |          |             | MF
| B841   | B9 04      | 492      |          |             | MF
| B842   | 6F BE      | 493      |          |             | MF
| B843   | 6F D0      | 494      |          |             | MF
| B844   | 6F 00      | 495      |          |             | MF
| B845   | 6F 00      | 496      |          |             | MF
| B846   | 6F 00      | 497      |          |             | MF
| B847   | 6F 00      | 498      |          |             | MF
| B848   | 6F 00      | 499      |          |             | MF
| B849   | 6F 00      | 500      |          |             | MF
| B84A   | 6F 00      | 501      |          |             | MF
| B84B   | 6F 00      | 502      |          |             | MF
| B84C   | 6F 00      | 503      |          |             | MF
| B84D   | 6F 00      | 504      |          |             | MF
| B84E   | 6F 00      | 505      |          |             | MF
| B84F   | 6F 00      | 506      |          |             | MF
| B850   | 6F 00      | 507      |          |             | MF
| B851   | 6F 00      | 508      |          |             | MF
| B852   | 6F 00      | 509      |          |             | MF
| B853   | 6F 00      | 510      |          |             | MF
| B854   | 6F 00      | 511      |          |             | MF
| B855   | 6F 00      | 512      |          |             | MF
| B856   | 6F 00      | 513      |          |             | MF
| B857   | 6F 00      | 514      |          |             | MF
| B858   | 6F 00      | 515      |          |             | MF
| B859   | 6F 00      | 516      |          |             | MF
| B85A   | 6F 00      | 517      |          |             | MF
| B85B   | 6F 00      | 518      |          |             | MF
| B85C   | 6F 00      | 519      |          |             | MF
| B85D   | 6F 00      | 520      |          |             | MF
| B85E   | 6F 00      | 521      |          |             | MF
| B85F   | 6F 00      | 522      |          |             | MF
| B860   | 6F 00      | 523      |          |             | MF
| B861   | 6F 00      | 524      |          |             | MF
| B862   | 6F 00      | 525      |          |             | MF
| B863   | 6F 00      | 526      |          |             | MF
| B864   | 6F 00      | 527      |          |             | MF
| B865   | 6F 00      | 528      |          |             | MF
| B866   | 6F 00      | 529      |          |             | MF
| B867   | 6F 00      | 530      |          |             | MF
| B868   | 6F 00      | 531      |          |             | MF
| B869   | 6F 00      | 532      |          |             | MF
| B86A   | 6F 00      | 533      |          |             | MF
| B86B   | 6F 00      | 534      |          |             | MF
| B86C   | 6F 00      | 535      |          |             | MF
| B86D   | 6F 00      | 536      |          |             | MF
| B86E   | 6F 00      | 537      |          |             | MF
| B86F   | 6F 00      | 538      |          |             | MF
| B870   | 6F 00      | 539      |          |             | MF
| B871   | 6F 00      | 540      |          |             | MF
BAF4: 60 00 715 PC3 LDA $C080,X
BAF6: 60 80 C0 716 PC4 LDY $080,X
BAFA: F0 17 718 BEQ PCERR
BAC: 60 01 719 LDA $C081,X
BACC: 60 01 720 BPL PC4
BACD: 60 00 721 LDA $C080,X
BD07: 10 04 722 BPL PC5
BD08: 60 01 723 INC $1
BD10: 60 06 724 BEQ PCERR
BD13: 80 FF 725 INC #0
BD17: 60 01 726 BNE PC3
BD19: 60 10 727 BNE PC6
BD1B: 60 728 PCERR LDA #$FF
BD1D: 60 00 729 LDA $1
BD1F: 60 00 730 CMP #$20
BD21: 20 81 B9 731 RTS TRKV3 JSR TRKV2
BD23: 60 06 732 TRKV3ER LDA #$BC
BD25: 18 733 ARM TRKV3ER
BD27: 60 8C 734 ARM ARMV
BD29: 60 12 735 ARMV
BD2B: 60 00 736 ;WORK OK
BD2D: 60 00 737 ;TOSMALL
BD30: 38 7D B8 738 WANTRK DFB 0
BD31: 60 739 WANTHALF DFB 0
BD32: 00 740 PHSLST DFB 0
BD35: 00 741 ARMWOKL DFB 0
BD36: 00 742 ARMWOKH DFB 0
BD38: 00 743 ARMFSLH DFB 0
BD39: 00 744 ARMTFSH DFB 0
BD74: 0A 745 MODWRITE DS $BB74-MODWRITE
BD75: 0A 746 WRITETRK ASLA ACTUAL
BD76: AA 747 WRITE LDA $C081,X
BD7C: BD A5 BB 748 WRITE LDA $C081,X
BD7F: BD A5 BB 749 WRITE LDA $C081,X
BD82: DD 10 750 STA WRTJMP+2
BD87: BD BC CO 751 STA WRTJMP+1
BD94: 10 02 752 LDA C8 H
BD9C: 38 753 SEC RTS
BD9E: 96 754 S1 TYS
BE06: 49 FF 755 STY $FF
BE21: 68 756 INC Y
BE35: 60 757 NOP
BE44: EA 758 NOP
BE55: EA 759 NOP
BE64: 60 CO 760 STA $C081,X
BE6D: 60 CO 761 STA $C081,X
BE9A: 9D BF CO 762 ORA $C081,X
BB00: 00 763 DA LAB9
BB04: BF 764 LTWRITE DA LAF
BB08: 74 765 DA LAF
BB0C: BF 766 DA LALC
BB1C: 24 FF 767 DA LAB
BB24: 00 BE 768 DA LAA
BB28: 80 00 769 DA LA9
Page 87
BCE6:  F0 03 9F34  BEQ S9B
BCE7:  A5 00 2F35  DEX $0
BCE8:  9F 0F 7F36  LDX $F000,Y
BCE9:  9F 00 BD CO 0F36  STA $0BD,X
BCEA:  0D 08 C0 0C39  ORA $0B8C,X
BCEB:  9C 00 DE 0F39  INY L9D
BCEC:  00 00 00 0F39  JMP $9C
BCEF:  00 00 00 0F39  DFB 0,0,0,0,0,0,0,0

BDD0:  6A 00 9C 0F39  NOP
BDD1:  0C 00 9F39  LDA $FC00,Y
BDD2:  09 00 9F39  TAX
BDD3:  06 00 9F39  BEQ S9C
BDD4:  03 00 9F39  LDA S9C
BDD5:  03 00 9F39  DEX S9C
BDD6:  06 00 9F39  BEQ S9C
BDD7:  09 00 9F39  LDA $0
BDD8:  00 00 80 9F39  LDY $8000,Y
BDD9:  00 00 80 9F39  STA $0800,X
BDDA:  00 00 80 9F39  ORA $080C,X
BDB0:  00 00 80 9F39  INY L9C
BDB1:  00 00 80 9F39  JMP W9D

BDB2:  4C 29 BD 00 0F39  LDA W9D
BDB3:  00 00 9D 0F39  NOP
BDB4:  00 00 9D 0F39  TAX
BDB5:  00 00 9D 0F39  BEQ S9D
BDB6:  00 00 9D 0F39  DEX S9D
BDB7:  00 00 9D 0F39  BEQ S9D
BDB8:  00 00 9D 0F39  LDA $0
BDB9:  00 00 81 9D 0F39  LDY $800,Y
BDBA:  00 00 81 9D 0F39  STA $0800,X
BDBB:  00 00 81 9D 0F39  ORA $080C,X
BDBC:  00 00 81 9D 0F39  INY L9D
BDBD:  00 00 81 9D 0F39  JMP W9E

BDBE:  4C 4D BD 00 0F39  LDA W9E
BDBF:  00 00 9E 0F39  NOP
BDC0:  00 00 9E 0F39  TAX
BDC1:  00 00 9E 0F39  BEQ S9E
BDC2:  00 00 9E 0F39  DEX S9E
BDC3:  00 00 9E 0F39  BEQ S9E
BDC4:  00 00 9E 0F39  LDA $0
BDC5:  00 00 82 9E 0F39  LDY $8200,Y
BDC6:  00 00 82 9E 0F39  STA $0800,X
BDC7:  00 00 82 9E 0F39  ORA $080C,X
BDC8:  00 00 82 9E 0F39  INY L9E
BDC9:  00 00 82 9E 0F39  JMP W9F

BDCD:  00 00 71 BD 00 0F39  LDA W9F
BDD0:  00 00 9F 0F39  NOP
BDD1:  00 00 9F 0F39  TAX
BDD2:  00 00 9F 0F39  BEQ S9F
BDD3:  00 00 9F 0F39  DEX S9F
BDD4:  00 00 9F 0F39  BEQ S9F
BDD5:  00 00 9F 0F39  LDA $0
BDD6:  00 00 82 1F 0F39  LDY $8200,Y
BDD7:  00 00 82 1F 0F39  STA $0800,X
BDD8:  00 00 82 1F 0F39  ORA $080C,X
BDD9:  00 00 82 1F 0F39  INY L9F
BDDA:  00 00 82 1F 0F39  JMP $9F

Page 90
<table>
<thead>
<tr>
<th>Line</th>
<th>Assembly</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>LDX C3</td>
<td>Load X from C3</td>
</tr>
<tr>
<td>2</td>
<td>STA $3000,Y</td>
<td>Store A from X at $3000,Y</td>
</tr>
<tr>
<td>3</td>
<td>STA *C080,X</td>
<td>Store A at address *C080,X</td>
</tr>
<tr>
<td>4</td>
<td>ORA *C08C,X</td>
<td>OR A with value at *C08C,X</td>
</tr>
<tr>
<td>5</td>
<td>INC L9F</td>
<td>Increment L9F</td>
</tr>
<tr>
<td>6</td>
<td>JMP QA0</td>
<td>Jump to QA0</td>
</tr>
<tr>
<td>7</td>
<td>NOP</td>
<td>No Operation</td>
</tr>
<tr>
<td>8</td>
<td>LDA $A000,Y</td>
<td>Load A from $A000,Y</td>
</tr>
<tr>
<td>9</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>10</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>11</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>12</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>13</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>14</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>15</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>16</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>17</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>18</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>19</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>20</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>21</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>22</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>23</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>24</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>25</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>26</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>27</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>28</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>29</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>30</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>31</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>32</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>33</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>34</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>35</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>36</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>37</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>38</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>39</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>40</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>41</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>42</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>43</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>44</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>45</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>46</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>47</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>48</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>49</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>50</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>51</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>52</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>53</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>54</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>55</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>56</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>57</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>58</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>59</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>60</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>61</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>62</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>63</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>64</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>65</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>66</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>67</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>68</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>69</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>70</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>71</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>72</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>73</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>74</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>75</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>76</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>77</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>78</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>79</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>80</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>81</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>82</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>83</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>84</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>85</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>86</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>87</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>88</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>89</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>90</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>91</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>92</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>93</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>94</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>95</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>96</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>97</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>98</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>99</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>100</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>101</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>102</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>103</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>104</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>105</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>106</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>107</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>108</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>109</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>110</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>111</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>112</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>113</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>114</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>115</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>116</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>117</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>118</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>119</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>120</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>121</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>122</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>123</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>124</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>125</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>126</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>127</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>128</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>129</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>130</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>131</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>132</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>133</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>134</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>135</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>136</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>137</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>138</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>139</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>140</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>141</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>142</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
<tr>
<td>143</td>
<td>STA SA0</td>
<td>Store A at SA0</td>
</tr>
</tbody>
</table>

Page 91
<table>
<thead>
<tr>
<th>BE48</th>
<th>EB 00 A4</th>
<th>44 LA4</th>
<th>NOP</th>
<th>$A400,Y ;$8800</th>
</tr>
</thead>
<tbody>
<tr>
<td>BE49</td>
<td>EB 00 A5</td>
<td>44 LA5</td>
<td>LDA</td>
<td>$A500,Y ;$8900</td>
</tr>
<tr>
<td>BE50</td>
<td>EB 00 A6</td>
<td>44 LA6</td>
<td>LDA</td>
<td>$A600,Y ;$8A00</td>
</tr>
<tr>
<td>BE51</td>
<td>EB 00 A7</td>
<td>44 LA7</td>
<td>LDA</td>
<td>$A700,Y ;$8B00</td>
</tr>
</tbody>
</table>

Page 92