

!A

LLOAD SCAN.L,A\$4000

\*\*\* End of Pass 1

LLOAD SCAN1.L,A\$4000

LLOAD SCAN2.L,A\$4000

LLOAD SCAN3.L,A\$4000

LLOAD SCAN4.L,A\$4000

LLOAD SCAN5.L,A\$4000

LLOAD SCAN.L,A\$4000

\*\*\* End of Pass 2

```
0800      1      ttl "Track Scan Source Code, SCAN.L"
0800      2      src "SCAN.L"
0800      3      ;
0800      4      ;
0800      5      ; SCAN.L
0800      6      ;
0800      7      ;
0800      8      ; Track Scan Source Code
0800      9      ;
0800     10      ; 2024 February 14
0800     11      ;
0800     12      ;
0800     13      ; DOS 4.5, Build 06
0800     14      ;
0800     15      ; 2024 February 14
0800     16      ;
0800     17      ;
0800     18      ; Start of Source Code: 0x4000
0800     19      ; Start of Symbol List: 0x7800
0800     20      ;
0800     21      ;
0800     22      ; Copyright (c) 2024 February 14 by
0800     23      ; Walland Philip Vrbancic Jr
0800     24      ;
0800     25      ; 6223 East Peabody Street
0800     26      ; Long Beach, California 90808
0800     27      ; United States of America
0800     28      ;
0800     29      ; All Rights Reserved
0800     30      ;
0800     31      ; This software is the confidential and
0800     32      ; proprietary intellectual property of
0800     33      ; Walland Philip Vrbancic Jr
0800     34      ;
0800     35      ;
0000     36      LOC0      epz $00
0800     37      ;
0020     38      WNDLFT      epz $20
0021     39      WNDWDTH      epz $21
0022     40      WNDTOP      epz $22
0023     41      WNDBTM      epz $23
0024     42      CH          epz $24
0025     43      CV          epz $25
0026     44      TEMPZ       epz $26
0027     45      TEMP2Z      epz $27
002A     46      CURTRKZ     epz $2A
002B     47      SLOT16Z    epz $2B
002C     48      DRVFLAG     epz $2C
002C     49      ADRDATMK    epz $2C
002C     50      ADRFIELD    epz $2C
002D     51      SECFNDZ     epz $2D
002E     52      TRKFNDZ     epz $2E
002F     53      VOLFNDZ     epz $2F
0800     54      ;
0032     55      INVFLG      epz $32
0033     56      PROMPT      epz $33
0034     57      PHASE       epz $34
0035     58      PAGECNT     epz $35
0035     59      SYNCNT      epz $35
0036     60      CSWL        epz $36
```

003C	61	ROMTEMPZ	epz	\$3C
003C	62	MOTORTIM	epz	\$3C
003E	63	BUFADR2Z	epz	\$3E
003E	64	ODDBITSZ	epz	\$3E
003F	65	SECTORZ	epz	\$3F
0800	66	;		
0040	67	TRACKZ	epz	\$40
0041	68	VOLUMEZ	epz	\$41
004A	69	IOBADR	epz	\$4A
0800	70	;		
00EE	71	GENPTR	epz	\$EE
00FA	72	DATAPTR	epz	\$FA
00FC	73	PRNTPTR	epz	\$FC
0800	74	;		
0800	75		enz	
0800	76	;		
0000	77	ZERO	equ	\$00
00FF	78	NEGONE	equ	\$FF
0800	79	;		
0000	80	SEEKCMD	equ	\$00
0001	81	READCMD	equ	\$01
0002	82	WRITCMD	equ	\$02
0004	83	FRMTCMD	equ	\$04
0800	84	;		
0000	85	TEXTMODE	equ	\$00
0001	86	GRPHMODE	equ	\$01
0002	87	TX80MODE	equ	\$02
0003	88	LV80MODE	equ	\$03
0800	89	;		
0000	90	NORMDISP	equ	\$00
0001	91	INVRDISP	equ	\$01
0800	92	;		
0000	93	INITSCRN	equ	\$00
0001	94	HOMESCRN	equ	\$01
0800	95	;		
0000	96	EOLCLR	equ	\$00
0001	97	EOPCLR	equ	\$01
0800	98	;		
0000	99	DIRECT	equ	\$00
0001	100	INDIRECT	equ	\$01
0800	101	;		
0000	102	NOPAD	equ	\$00
0020	103	HIGHLOW	equ	\$20
0040	104	ZEROPAD	equ	\$40
0080	105	SPCPAD	equ	\$80
0800	106	;		
0004	107	NEXTLINE	equ	\$04
0007	108	CHARCELL	equ	\$07
000F	109	PCMDMASK	equ	\$0F
000F	110	NIBLMASK	equ	\$0F
0014	111	VSNLEN	equ	\$14
0018	112	NAMELEN	equ	\$18
001F	113	CVMASK	equ	\$1F
0028	114	MAXWDTH	equ	\$28
0032	115	MAXTRKS	equ	\$32
003F	116	TRKMASK	equ	\$3F
0050	117	MAXCH	equ	\$50
0060	118	MINCV	equ	\$60
007F	119	ASCIMASK	equ	\$7F
007F	120	INVRMASK	equ	\$7F
0080	121	MSBMASK	equ	\$80

0080	122	ASCIFLAG	equ	\$80
0800	123	;		
0050	124	RTNCMD	equ	\$50
0051	125	MODECMD	equ	\$51
0052	126	DISPCMD	equ	\$52
0053	127	SCRNCMD	equ	\$53
0054	128	CLRCMD	equ	\$54
0055	129	CNTRCMD	equ	\$55
0056	130	BUFRCMD	equ	\$56
0057	131	NIBLCMD	equ	\$57
0058	132	BYT1CMD	equ	\$58
0059	133	BYT2CMD	equ	\$59
005A	134	BYTNCMD	equ	\$5A
005B	135	ADRCMD	equ	\$5B
005C	136	DEC1CMD	equ	\$5C
005D	137	DEC2CMD	equ	\$5D
005E	138	DEC3CMD	equ	\$5E
005F	139	DECNCMD	equ	\$5F
0800	140	;		
0060	141	FLASH	equ	\$60
0084	142	CTRLD	equ	\$84
0087	143	BELLCHAR	equ	\$87
0088	144	LARROW	equ	\$88
008A	145	DARROW	equ	\$8A
008B	146	UARROW	equ	\$8B
008D	147	RETURN	equ	\$8D
0091	148	CTRLQ	equ	\$91
0095	149	RARROW	equ	\$95
009B	150	ESCAPE	equ	\$9B
00A0	151	SPACE	equ	" "
00AE	152	PERIOD	equ	". "
00AF	153	SLASH	equ	" / "
00BA	154	COLON	equ	" : "
00DF	155	LWRMASK	equ	\$DF
00E0	156	LWRCASE	equ	\$E0
00FF	157	DELETE	equ	\$FF
0800	158	;		
0004	159	DFLTPHAS	equ	4
0010	160	DFLTSEC	equ	16
0023	161	DFLTRACK	equ	35
0030	162	MAXTRACK	equ	48
0800	163	;		
0010	164	PHASMAX	equ	16
0800	165	;		
0006	166	HDRSYNC	equ	6
0008	167	MINSYNC	equ	8
0020	168	MAXSYNC	equ	32
0020	169	MAXRETRY	equ	32
0500	170	SYNCBITS	equ	MAXSYNC*40
0800	171	;		
0001	172	DRIVE1	equ	\$01
0002	173	DRIVE2	equ	\$02
0800	174	;		
0000	175	RWTSSEEK	equ	\$00
0001	176	RWTSREAD	equ	\$01
0002	177	RWTSWRIT	equ	\$02
0004	178	RWTSFRMT	equ	\$04
0800	179	;		
0000	180	RWNOERR	equ	\$00
0008	181	RWINITER	equ	\$08
0010	182	RWPROTER	equ	\$10

0020	183	RWVOLERR	equ	\$20
0030	184	RWSYNERR	equ	\$30
0040	185	RWDRVERR	equ	\$40
0080	186	RWREADER	equ	\$80
0800	187	;		
0040	188	WNIBLSIZ	equ	\$40
0056	189	NBUF2SIZ	equ	\$56
006A	190	RNIBLSIZ	equ	\$6A
0800	191	;		
00AA	192	ODDBITS	equ	\$AA
00D5	193	ADRMARK1	equ	\$D5
00AA	194	ADRMARK2	equ	\$AA
0096	195	ADRMARK3	equ	\$96
00D5	196	DATMARK1	equ	\$D5
00AA	197	DATMARK2	equ	\$AA
00AD	198	DATMARK3	equ	\$AD
00DE	199	SLPMARK1	equ	\$DE
00AA	200	SLPMARK2	equ	\$AA
00EB	201	SLPMARK3	equ	\$EB
00FF	202	SYNCMARK	equ	\$FF
0800	203	;		
D8EF	204	MOTONTIM	equ	!-10000-1
0800	205	;		
0100	206	STACK	equ	\$100
0100	207	PAGESIZE	equ	\$100
0800	208	;		
03D0	209	DOSWARM	equ	\$3D0
03D3	210	DOSCOLD	equ	\$3D3
03D9	211	CALLRWTS	equ	\$3D9
03E1	212	RDCLKVSN	equ	\$3E1
03E3	213	GETIOB	equ	\$3E3
03EA	214	HOOKDOS	equ	\$3EA
0800	215	;		
0478	216	DRV0TRK	equ	\$478
04F8	217	DRV1TRK	equ	\$4F8
0678	218	DRV0PHAS	equ	\$678
06F8	219	DRV1PHAS	equ	\$6F8
0800	220	;		
0478	221	FINDTRK	equ	\$478
04F8	222	RECALCNT	equ	\$4F8
0800	223	;		
0578	224	SEEKCNT	equ	\$578
05F8	225	RETRYCNT	equ	\$5F8
0800	226	;		
0678	227	NEXTON	equ	\$678
06F8	228	NEXTOFF	equ	\$6F8
0800	229	;		
0778	230	ALLOCNTR	equ	\$778
07F8	231	MSLOT	equ	\$7F8
0800	232	;		
1C00	233	BUFSIZE	equ	\$1C00
0800	234	;		
BFF2	235	MNGDISK	equ	\$BFF2
0800	236	;		
C000	237	PAGEC0	equ	\$C000
D000	238	PAGED0	equ	\$D000
0800	239	;		
C000	240	KEY	equ	\$C000
C010	241	CLRKEY	equ	\$C010
C050	242	TXTCLR	equ	\$C050
C051	243	TXTSET	equ	\$C051

C052	244	MIXCLR	equ	\$C052
C054	245	LOWSCR	equ	\$C054
C057	246	HIRES	equ	\$C057
C082	247	ROM2WP	equ	\$C082
0800	248	;		
C080	249	PHASEOFF	equ	\$C080
C088	250	MOTOROFF	equ	\$C088
C089	251	MOTORON	equ	\$C089
C08A	252	DRV0EN	equ	\$C08A
C08B	253	DRV1EN	equ	\$C08B
C08C	254	STROBE	equ	\$C08C
C08D	255	LATCH	equ	\$C08D
C08E	256	DATAIN	equ	\$C08E
C08F	257	DATAOUT	equ	\$C08F
0800	258	;		
CFFF	259	CLRROM	equ	\$CFFF
0800	260	;		
F941	261	PRNTAX	equ	\$F941
F94A	262	PRBL2	equ	\$F94A
FB2F	263	INIT	equ	\$FB2F
FB5B	264	TABV	equ	\$FB5B
FC22	265	VTAB	equ	\$FC22
FC42	266	CLREOP	equ	\$FC42
FC58	267	HOME	equ	\$FC58
FC9C	268	CLREOL	equ	\$FC9C
FD8E	269	CROUT	equ	\$FD8E
FDDA	270	PRBYTE	equ	\$FDDA
FDE3	271	PRHEX	equ	\$FDE3
FDED	272	COUT	equ	\$FDED
FE95	273	OUTPORT	equ	\$FE95
FF3A	274	BELL	equ	\$FF3A
FF69	275	MONITOR	equ	\$FF69
0800	276	;		
0800	277	;		
0800	278		icl	"SCAN1.L"

LLOAD SCAN1.L,A\$4000

```

0800          1          ttl "Track Scan Source Code, SCAN1.L"
0800          2          ;
0800          3          ;
0800          4          ; SCAN1.L
0800          5          ;
0800          6          ;
0000          7  DISPLAY equ TEXTMODE
0800          8          ;
0800          9          ;
0900         10          org $900
0900         11          obj $900
0900         12          usr
0900         13          ;
0900         14          ;
0900 20 EA 03         15          jsr HOOKDOS
0903         16          ;
0903 20 CB 16         17          jsr PRINT
0906         18          .if DISPLAY=TEXTMODE
0906 51 00           19          byt MODECMD,TEXTMODE
0908         20          .fi
0908         21          .if DISPLAY=GRPHMODE
0908         22          byt RETURN
0908         23          byt MODECMD,GRPHMODE
0908         24          .fi
0908         25          .if DISPLAY=TX80MODE
0908         26          byt MODECMD,TX80MODE
0908         27          .fi
0908 52 00           28          byt DISPCMD,NORMDISP
090A 53 00           29          byt SCRNCMD,INITSCRN
090C 53 01           30          byt SCRNCMD,HOMESCRN
090E 50             31          byt RTNCMD
090F             32          ;
090F 20 71 15        33          jsr INITPGM
0912             34          ;
0912             35          ;
0912 38             36          ^0 sec
0913             37          ;
0913 20 71 0F        38          ^1 jsr SHOWMENU
0916             39          ;
0916 20 43 09        40          jsr GETOPTN
0919 B0 25           41          bcs >6
091B             42          ;
091B C9 01           43          cmp #1
091D D0 03           44          bne >2
091F             45          ;
091F 20 6F 09        46          jsr GETSDP
0922             47          ;
0922 C9 02           48          ^2 cmp #2
0924 D0 03           49          bne >3
0926             50          ;
0926 20 64 0A        51          jsr GETRKRNG
0929             52          ;
0929 C9 03           53          ^3 cmp #3
092B D0 05           54          bne >4
092D             55          ;
092D 20 EF 0A        56          jsr DOSCAN
0930 B0 E1           57          bcs <1
0932             58          ;
0932 C9 04           59          ^4 cmp #4
0934 D0 05           60          bne >5

```

```

0936      61 ;
0936 20 FF 0C 62      jsr DOFORMAT
0939 F0 D7 63      beq <0          ; always taken
093B      64 ;
093B C9 05 65      ^5      cmp #5
093D      66 ;
093D 18 67      clc
093E      68 ;
093E D0 D3 69      bne <1
0940      70 ;
0940 4C 14 0E 71      ^6      jmp QUIT
0943      72 ;
0943      73 ;
0943 20 CB 16 74      GETOPTN  jsr PRINT
0946 00 76 75      hex 0076
0948 54 00 76      byt CLRCMD,EOLCLR
094A D3 E5 EC 77      asc "Select Option:  "
094D E5 E3 F4
0950 A0 CF F0
0953 F4 E9 EF
0956 EE BA A0
0959 A0
095A 50      78      byt RTNCMD
095B      79 ;
095B 20 C0 15 80      jsr GETDEC1
095E B0 0E 81      bcs >1
0960      82 ;
0960 AD 72 19 83      lda DECVALS
0963 C9 01 84      cmp #1
0965 90 DC 85      bcc GETOPTN
0967      86 ;
0967 C9 06 87      cmp #5+1
0969 B0 D8 88      bcs GETOPTN
096B      89 ;
096B 8D 5F 19 90      sta OPTION
096E      91 ;
096E 60 92      ^1      rts
096F      93 ;
096F      94 ;
096F 20 84 09 95      GETSDP  jsr GETSLOT
0972 B0 0D 96      bcs >1
0974      97 ;
0974 20 71 0F 98      jsr SHOWMENU
0977      99 ;
0977 20 EB 09 100     jsr GETDRV
097A      101 ;
097A 18 102     clc
097B      103 ;
097B 20 71 0F 104     jsr SHOWMENU
097E      105 ;
097E 20 26 0A 106     jsr GETPHS
0981      107 ;
0981 A9 00 108     ^1      lda #ZERO
0983      109 ;
0983 60 110     rts
0984      111 ;
0984      112 ;
0984 20 CB 16 113     GETSLOT  jsr PRINT
0987 00 76 114     hex 0076
0989 54 00 115     byt CLRCMD,EOLCLR
098B C5 EE F4 116     asc "Enter "

```



```

098E E5 F2 A0
0991 52 01      117      byt DISPCMD,INVRDISP
0993 D3 CC CF   118      asc "SLOT"
0996 D4
0997 52 00      119      byt DISPCMD,NORMDISP
0999 A0 EE F5   120      asc " number ("
099C ED E2 E5
099F F2 A0 A8
09A2 5C          121      byt DEC1CMD
09A3 4E 19      122      adr SLOT
09A5 A9 BA A0   123      asc "): "
09A8 A0
09A9 50          124      byt RTNCMD
09AA          125      ;
09AA 20 C0 15   126      jsr GETDEC1
09AD B0 17      127      bcs >1
09AF          128      ;
09AF F0 15      129      beq >1
09B1          130      ;
09B1 AD 72 19   131      lda DECVALS
09B4 C9 01      132      cmp #1
09B6 90 CC      133      bcc GETSLOT
09B8          134      ;
09B8 C9 08      135      cmp #7+1
09BA B0 C8      136      bcs GETSLOT
09BC          137      ;
09BC 8D 4E 19   138      sta SLOT
09BF          139      ;
09BF 0A          140      asl
09C0 0A          141      asl
09C1 0A          142      asl
09C2 0A          143      asl
09C3          144      ;
09C3 8D B6 19   145      sta SLOT16
09C6          146      ;
09C6 08          147      ^1 php
09C7          148      ;
09C7 AE 4E 19   149      ldx SLOT
09CA A9 00      150      lda #ZERO
09CC          151      ;
09CC 20 E7 09   152      jsr GETDISK
09CF          153      ;
09CF C9 C0      154      cmp /PAGEC0
09D1 90 12      155      bcc >2
09D3          156      ;
09D3 C9 D0      157      cmp /PAGED0
09D5 B0 0E      158      bcs >2
09D7          159      ;
09D7 48          160      pha
09D8          161      ;
09D8 AD 4E 19   162      lda SLOT
09DB 0A          163      asl
09DC AA          164      tax
09DD          165      ;
09DD 68          166      pla
09DE 9D 02 13   167      sta DISKADRS-1,X
09E1          168      ;
09E1 98          169      tya
09E2 9D 01 13   170      sta DISKADRS-2,X
09E5          171      ;
09E5 28          172      ^2 plp

```

```

09E6          173 ;
09E6 60       174      rts
09E7          175 ;
09E7          176 ;
09E7 38       177 GETDISK sec
09E8          178 ;
09E8 6C F2 BF 179      jmp (MNGDISK)
09EB          180 ;
09EB          181 ;
09EB 20 CB 16 182 GETDRV jsr PRINT
09EE 00 76     183      hex 0076
09F0 54 00     184      byt CLRCMD,EOLCLR
09F2 C5 EE F4 185      asc "Enter "
09F5 E5 F2 A0
09F8 52 01     186      byt DISPCMD,INVRDISP
09FA C4 D2 C9 187      asc "DRIVE"
09FD D6 C5
09FF 52 00     188      byt DISPCMD,NORMDISP
0A01 A0 EE F5 189      asc " number ("
0A04 ED E2 E5
0A07 F2 A0 A8
0A0A 5D        190      byt DEC2CMD
0A0B B7 19     191      adr DRIVE
0A0D A9 BA A0 192      asc "): "
0A10 A0
0A11 50        193      byt RTNCMD
0A12          194 ;
0A12 20 C3 15 195      jsr GETDEC2
0A15 B0 0E     196      bcs >1
0A17          197 ;
0A17 F0 0C     198      beq >1
0A19          199 ;
0A19 AD 72 19 200      lda DECVALS
0A1C F0 CD     201      beq GETDRV
0A1E          202 ;
0A1E C9 52     203      cmp #82
0A20 B0 C9     204      bcs GETDRV
0A22          205 ;
0A22 8D B7 19 206      sta DRIVE
0A25          207 ;
0A25 60        208 ^1    rts
0A26          209 ;
0A26          210 ;
0A26 20 CB 16 211 GETPHS jsr PRINT
0A29 00 76     212      hex 0076
0A2B 54 00     213      byt CLRCMD,EOLCLR
0A2D C5 EE F4 214      asc "Enter "
0A30 E5 F2 A0
0A33 52 01     215      byt DISPCMD,INVRDISP
0A35 D0 C8 C1 216      asc "PHASE"
0A38 D3 C5
0A3A 52 00     217      byt DISPCMD,NORMDISP
0A3C A0 EE F5 218      asc " number ("
0A3F ED E2 E5
0A42 F2 A0 A8
0A45 5D        219      byt DEC2CMD
0A46 BF 19     220      adr IOCBPHAS
0A48 A9 BA A0 221      asc "): "
0A4B A0
0A4C 50        222      byt RTNCMD
0A4D          223 ;

```

```

0A4D 20 C3 15      224      jsr GETDEC2
0A50 B0 11        225      bcs >1
0A52              226      ;
0A52 F0 0F        227      beq >1
0A54              228      ;
0A54 AD 72 19     229      lda DECVALS
0A57 F0 CD        230      beq GETPHS
0A59              231      ;
0A59 C9 11        232      cmp #17
0A5B B0 C9        233      bcs GETPHS
0A5D              234      ;
0A5D 8D BF 19     235      sta IOCBPHAS
0A60 8D 42 19     236      sta VALSPHAS
0A63              237      ;
0A63 60           238      ^1    rts
0A64              239      ;
0A64              240      ;
0A64 20 72 0A     241      GETRKRNG jsr GETSTART
0A67 B0 06        242      bcs >1
0A69              243      ;
0A69 20 71 0F     244      jsr SHOWMENU
0A6C              245      ;
0A6C 20 B3 0A     246      jsr GETEND
0A6F              247      ;
0A6F A9 00        248      ^1    lda #ZERO
0A71              249      ;
0A71 60           250      rts
0A72              251      ;
0A72              252      ;
0A72 20 CB 16     253      GETSTART jsr PRINT
0A75 00 76        254      hex 0076
0A77 54 00        255      byt CLRCMD,EOLCLR
0A79 C5 EE F4     256      asc "Enter "
0A7C E5 F2 A0
0A7F 52 01        257      byt DISPCMD,INVRDISP
0A81 D3 D4 C1     258      asc "START"
0A84 D2 D4
0A86 52 00        259      byt DISPCMD,NORMDISP
0A88 A0 D4 F2     260      asc " Track ("
0A8B E1 E3 EB
0A8E A0 A8
0A90 5D           261      byt DEC2CMD
0A91 4A 19        262      adr STRTRACK
0A93 A9 BA A0     263      asc "): "
0A96 A0
0A97 50           264      byt RTNCMD
0A98              265      ;
0A98 20 C3 15     266      jsr GETDEC2
0A9B B0 15        267      bcs >1
0A9D              268      ;
0A9D F0 13        269      beq >1
0A9F              270      ;
0A9F AD 72 19     271      lda DECVALS
0AA2 C9 31        272      cmp #49
0AA4 B0 CC        273      bcs GETSTART
0AA6              274      ;
0AA6 8D 4A 19     275      sta STRTRACK
0AA9              276      ;
0AA9 CD 4B 19     277      cmp ENDTRACK
0AAC 90 04        278      bcc >1
0AAE              279      ;

```

```

0AAE 8D 4B 19      280      sta ENDTRACK
0AB1              281      ;
0AB1 18            282      clc
0AB2              283      ;
0AB2 60            284      ^1      rts
0AB3              285      ;
0AB3              286      ;
0AB3 20 CB 16      287      GETEND    jsr PRINT
0AB6 00 76          288      hex 0076
0AB8 54 00          289      byt CLRCMD,EOLCLR
0ABA C5 EE F4      290      asc "Enter "
0ABD E5 F2 A0
0AC0 52 01          291      byt DISPCMD,INVRDISP
0AC2 CC C1 D3      292      asc "LAST"
0AC5 D4
0AC6 52 00          293      byt DISPCMD,NORMDISP
0AC8 A0 D4 F2      294      asc " Track ("
0ACB E1 E3 EB
0ACE A0 A8
0AD0 5D            295      byt DEC2CMD
0AD1 4B 19          296      adr ENDTRACK
0AD3 A9 BA A0      297      asc "): "
0AD6 A0
0AD7 50            298      byt RTNCMD
0AD8              299      ;
0AD8 20 C3 15      300      jsr GETDEC2
0ADB B0 11          301      bcs >1
0ADD              302      ;
0ADD F0 0F          303      beq >1
0ADF              304      ;
0ADF AD 72 19      305      lda DECVALS
0AE2 CD 4A 19      306      cmp STRTRACK
0AE5 90 CC          307      bcc GETEND
0AE7              308      ;
0AE7 C9 31          309      cmp #49
0AE9 B0 C8          310      bcs GETEND
0AEB              311      ;
0AEB 8D 4B 19      312      sta ENDTRACK
0AEE              313      ;
0AEE 60            314      ^1      rts
0AEF              315      ;
0AEF              316      ;
0AEF 20 CB 16      317      DOSCAN    jsr PRINT
0AF2 62            318      hex 62
0AF3 54 01          319      byt CLRCMD,EOPCLR
0AF5 8D 8D          320      byt RETURN,RETURN
0AF7 D6 EF EC      321      asc "Vol Trk Sec Gap Hdr Gap Data Err"
0AFA A0 A0 D4
0AFD F2 EB A0
0B00 A0 D3 E5
0B03 E3 A0 A0
0B06 C7 E1 F0
0B09 A0 A0 C8
0B0C E4 F2 A0
0B0F A0 C7 E1
0B12 F0 A0 A0
0B15 C4 E1 F4
0B18 E1 A0 A0
0B1B C5 F2 F2
0B1E 8D            322      byt RETURN
0B1F AD AD AD      323      asc "----"

```

```

0B22 A0 A0 AD
0B25 AD AD A0
0B28 A0 AD AD
0B2B AD A0 A0
0B2E AD AD AD
0B31 A0 A0 AD
0B34 AD AD A0
0B37 A0 AD AD
0B3A AD A0 A0
0B3D AD AD AD
0B40 AD A0 A0
0B43 AD AD AD
0B46 50          324      byt RTNCMD
0B47          325      ;
0B47 AD 4A 19    326      lda STRTRACK
0B4A 8D B9 19    327      sta TRACK
0B4D          328      ;
0B4D A9 08       329      lda #8                ; 1 sector before 0x0F
0B4F 8D BA 19    330      sta SECTOR
0B52          331      ;
0B52 A9 00       332      lda #BUFFER
0B54 8D BD 19    333      sta DATABUFR
0B57          334      ;
0B57 A9 1C       335      lda /BUFFER
0B59 8D BE 19    336      sta DATABUFR+1
0B5C          337      ;
0B5C A9 01       338      lda #READCMD
0B5E 8D C1 19    339      sta CMDCODE
0B61          340      ;
0B61 A9 00       341      ^1      lda #ZERO
0B63 8D 5A 19    342      sta SYNCTOTL
0B66 8D 5B 19    343      sta SYNCTOTL+1
0B69          344      ;
0B69 20 CB 16    345      jsr PRINT
0B6C 00 66       346      hex 0066
0B6E 54 01       347      byt CLRCMD,EOPCLR
0B70 50          348      byt RTNCMD
0B71          349      ;
0B71 A0 B5       350      ld y #TBLTYPE
0B73 A9 19       351      lda /TBLTYPE
0B75          352      ;
0B75 20 3E 10    353      jsr DORWTS
0B78 B0 64       354      bcs >6
0B7A          355      ;
0B7A 20 4C 0C    356      jsr READTRK
0B7D          357      ;
0B7D 20 93 0E    358      jsr FINDEND
0B80 B0 7D       359      bcs >7
0B82          360      ;
0B82 A9 00       361      lda #ZERO
0B84 8D 48 19    362      sta SECCNT
0B87          363      ;
0B87 A9 01       364      ^3      lda #1
0B89 8D 47 19    365      sta STAGE
0B8C          366      ;
0B8C 20 8F 0C    367      jsr SHOWADR
0B8F B0 14       368      bcs >4
0B91          369      ;
0B91 A9 05       370      lda #5
0B93 8D 47 19    371      sta STAGE
0B96          372      ;

```

```

0B96 20 D9 0C    373      jsr SHOWDATA
0B99 B0 0A      374      bcs >4
0B9B          375      ;
0B9B A9 08      376      lda #8
0B9D 8D 47 19   377      sta STAGE
0BA0          378      ;
0BA0 A9 00      379      lda #ZERO
0BA2 8D 46 19   380      sta ERRFLAG
0BA5          381      ;
0BA5 20 3E 0C   382      ^4      jsr SHOWERR
0BA8          383      ;
0BA8 EE 48 19   384      inc SECCNT
0BAB          385      ;
0BAB AD 48 19   386      lda SECCNT
0BAE C9 10      387      cmp #16
0BB0 D0 D5      388      bne <3
0BB2          389      ;
0BB2 20 A1 0D   390      jsr SHOWAVG
0BB5 20 95 16   391      jsr GETKEY
0BB8          392      ;
0BB8 C9 9B      393      cmp #ESCAPE
0BBA F0 7F      394      beq >9
0BBC          395      ;
0BBC C9 D2      396      cmp #"R"
0BBE F0 A1      397      beq <1
0BC0          398      ;
0BC0 C9 C2      399      cmp #"B"
0BC2 D0 0D      400      bne >5
0BC4          401      ;
0BC4 AD B9 19   402      lda TRACK
0BC7 CD 4A 19   403      cmp STRTRACK
0BCA F0 95      404      beq <1
0BCC          405      ;
0BCC CE B9 19   406      dec TRACK
0BCF 10 90      407      bpl <1          ; always taken
0BD1          408      ;
0BD1 AD B9 19   409      ^5      lda TRACK
0BD4 CD 4B 19   410      cmp ENDTRACK
0BD7 F0 62      411      beq >9
0BD9          412      ;
0BD9 EE B9 19   413      inc TRACK
0BDC 10 83      414      bpl <1          ; always taken
0BDE          415      ;
0BDE 20 CB 16   416      ^6      jsr PRINT
0BE1 00 76      417      hex 0076
0BE3 C5 F2 F2   418      asc "Error from DORWTS, 0x"
0BE6 EF F2 A0
0BE9 E6 F2 EF
0BEC ED A0 C4
0BEF CF D2 D7
0BF2 D4 D3 AC
0BF5 A0 B0 F8
0BF8 58          419      byt BYT1CMD
0BF9 C2 19      420      adr ERRCODE
0BFB AE          421      asc "."
0BFC 50          422      byt RTNCMD
0BFD          423      ;
0BFD 90 38      424      bcc >8          ; always taken
0BFF          425      ;
0BFF 20 CB 16   426      ^7      jsr PRINT
0C02 00 76      427      hex 0076

```

```

0C04 D5 EE E1      428      asc "Unable to find an Address footer"
0C07 E2 EC E5
0C0A A0 F4 EF
0C0D A0 E6 E9
0C10 EE E4 A0
0C13 E1 EE A0
0C16 C1 E4 E4
0C19 F2 E5 F3
0C1C F3 A0 E6
0C1F EF EF F4
0C22 E5 F2
0C24 8D      429      byt RETURN
0C25 EF F2 A0    430      asc "or a Data footer."
0C28 E1 A0 C4
0C2B E1 F4 E1
0C2E A0 E6 EF
0C31 EF F4 E5
0C34 F2 AE
0C36 50      431      byt RTNCMD
0C37      432      ;
0C37 20 95 16    433      ^8      jsr GETKEY
0C3A      434      ;
0C3A 38      435      sec
0C3B      436      ;
0C3B A9 00      437      ^9      lda #ZERO
0C3D      438      ;
0C3D 60      439      rts
0C3E      440      ;
0C3E      441      ;
0C3E 20 CB 16    442      SHOWERR jsr PRINT
0C41 24      443      hex 24
0C42 5C      444      byt DEC1CMD
0C43 47 19      445      adr STAGE
0C45 AD      446      asc "-"
0C46 5C      447      byt DEC1CMD
0C47 46 19      448      adr ERRFLAG
0C49 8D      449      byt RETURN
0C4A 50      450      byt RTNCMD
0C4B      451      ;
0C4B 60      452      rts
0C4C      453      ;
0C4C      454      ;
0C4C AE B6 19    455      READTRK  ldx SLOT16
0C4F      456      ;
0C4F 20 81 13    457      jsr SETREAD
0C52      458      ;
0C52 BD 89 C0    459      lda MOTORON,X
0C55      460      ;
0C55 A9 00      461      lda #ZERO
0C57 85 EE      462      sta GENPTR
0C59      463      ;
0C59 A9 1D      464      lda /TRACKBUF
0C5B 85 EF      465      sta GENPTR+1
0C5D      466      ;
0C5D A9 1C      467      lda /BUFSIZE
0C5F 8D 49 19    468      sta TRACKSIZ
0C62      469      ;
0C62      470      ;
0C62      471      ; Read and toss 32 disk bytes.
0C62      472      ;
0C62 A0 08      473      ldy #8

```

```

0C64          474 ;
0C64 BD 8C C0 475 ^1      lda STROBE,X
0C67 10 FB    476          bpl <1
0C69          477 ;
0C69 20 70 15 478          jsr WAIT12
0C6C          479 ;
0C6C 88       480          dey
0C6D D0 F5    481          bne <1
0C6F          482 ;
0C6F          483 ;
0C6F          484 ; Now read 0x1C00 disk bytes.
0C6F          485 ;
0C6F 48       486 ^2      pha
0C70 68       487          pla
0C71          488 ;
0C71 BD 8C C0 489 ^3      lda STROBE,X
0C74 10 FB    490          bpl <3
0C76          491 ;
0C76 91 EE    492          sta (GENPTR),Y
0C78          493 ;
0C78 C8       494          iny
0C79 D0 F4    495          bne <2
0C7B          496 ;
0C7B E6 EF    497          inc GENPTR+1
0C7D          498 ;
0C7D CE 49 19 499          dec TRACKSIZ
0C80 D0 EF    500          bne <3
0C82          501 ;
0C82 BD 88 C0 502          lda MOTOROFF,X
0C85          503 ;
0C85 A9 1D    504          lda /TRACKBUF
0C87 85 EF    505          sta GENPTR+1
0C89          506 ;
0C89 A9 1C    507          lda /BUFSIZE
0C8B 8D 49 19 508          sta TRACKSIZ
0C8E          509 ;
0C8E 60       510          rts
0C8F          511 ;
0C8F          512 ;
0C8F A9 00    513 SHOWADR  lda #ZERO
0C91          514 ;
0C91 8D 51 19 515          sta HDRVOL
0C94 8D 52 19 516          sta HDRTRK
0C97 8D 53 19 517          sta HDRSEC
0C9A 8D 4D 19 518          sta HDRCNT
0C9D 8D 50 19 519          sta CHKSUM
0CA0 8D 4C 19 520          sta SYNCNUM
0CA3          521 ;
0CA3 20 22 0E 522          jsr FINDADR
0CA6 08       523          php
0CA7          524 ;
0CA7 A9 04    525          lda #4
0CA9 8D 47 19 526          sta STAGE
0CAC          527 ;
0CAC 20 CB 16 528          jsr PRINT
0CAF 00       529          hex 00
0CB0 5E       530          byt DEC3CMD
0CB1 51 19    531          adr HDRVOL
0CB3 06       532          hex 06
0CB4 5D       533          byt DEC2CMD
0CB5 52 19    534          adr HDRTRK

```



```

0CB7 0B          535          hex 0B
0CB8 5D          536          byt DEC2CMD
0CB9 53 19       537          adr HDRSEC
0CBB 0F          538          hex 0F
0CBC 5E          539          byt DEC3CMD
0CBD 4C 19       540          adr SYNCNUM
0CBF 15          541          hex 15
0CC0 5C          542          byt DEC1CMD
0CC1 4D 19       543          adr HDRCNT
0CC3 50          544          byt RTNCMD
0CC4             545          ;
0CC4 AD 53 19    546          lda HDRSEC
0CC7 F0 0E       547          beq >1
0CC9             548          ;
0CC9 AD 4C 19    549          lda SYNCNUM
0CCC 6D 5A 19    550          adc SYNCNTOTL
0CCF             551          ;
0CCF 8D 5A 19    552          sta SYNCNTOTL
0CD2 90 03       553          bcc >1
0CD4             554          ;
0CD4 EE 5B 19    555          inc SYNCNTOTL+1
0CD7             556          ;
0CD7 28          557          ^1 plp
0CD8             558          ;
0CD8 60          559          rts
0CD9             560          ;
0CD9             561          ;
0CD9 A9 00       562          SHOWDATA lda #ZERO
0CDB             563          ;
0CDB 8D 50 19    564          sta CHKSUM
0CDE 8D 4C 19    565          sta SYNCNUM
0CE1 8D 58 19    566          sta DATACNT
0CE4 8D 59 19    567          sta DATACNT+1
0CE7             568          ;
0CE7 20 B7 0E    569          jsr FINDATA
0CEA 08          570          php
0CEB             571          ;
0CEB A9 07       572          lda #7
0CED 8D 47 19    573          sta STAGE
0CF0             574          ;
0CF0 20 CB 16    575          jsr PRINT
0CF3 1A          576          hex 1A
0CF4 5D          577          byt DEC2CMD
0CF5 4C 19       578          adr SYNCNUM
0CF7 1D          579          hex 1D
0CF8 5F 80       580          byt DECNCMD,SPCPAD
0CFA 58 19       581          adr DATACNT
0CFC 50          582          byt RTNCMD
0CFD             583          ;
0CFD 28          584          plp
0CFE             585          ;
0CFE 60          586          rts
0CFF             587          ;
0CFF             588          ;
0CFF 20 CB 16    589          DOFORMAT jsr PRINT
0D02 00 76       590          hex 0076
0D04 54 00       591          byt CLRCMD,EOLCLR
0D06 C5 EE F4    592          asc "Enter "
0D09 E5 F2 A0
0D0C 52 01       593          byt DISPCMD,INVRDISP
0D0E CC C1 D3    594          asc "LAST"

```

```

0D11 D4
0D12 52 00      595      byt DISPCMD,NORMDISP
0D14 A0 D4 F2   596      asc " Track (35/36/40/48):  "
0D17 E1 E3 EB
0D1A A0 A8 B3
0D1D B5 AF B3
0D20 B6 AF B4
0D23 B0 AF B4
0D26 B8 A9 BA
0D29 A0 A0
0D2B 50          597      byt RTNCMD
0D2C             598      ;
0D2C 20 C3 15   599      jsr GETDEC2
0D2F B0 6D      600      bcs >3
0D31             601      ;
0D31 F0 1A      602      beq >2
0D33             603      ;
0D33 AD 72 19   604      lda DECVALS
0D36             605      ;
0D36 C9 23      606      cmp #35
0D38 F0 0C      607      beq >1
0D3A             608      ;
0D3A C9 24      609      cmp #36
0D3C F0 08      610      beq >1
0D3E             611      ;
0D3E C9 28      612      cmp #40
0D40 F0 04      613      beq >1
0D42             614      ;
0D42 C9 30      615      cmp #48
0D44 D0 B9      616      bne DOFORMAT
0D46             617      ;
0D46 8D 43 19   618      ^1 sta ENDTRK
0D49             619      ;
0D49 18          620      clc
0D4A             621      ;
0D4A 20 71 0F   622      jsr SHOWMENU
0D4D             623      ;
0D4D A9 00      624      ^2 lda #ZERO
0D4F 8D B9 19   625      sta TRACK
0D52 8D BA 19   626      sta SECTOR
0D55             627      ;
0D55 A9 00      628      lda #BUFFER
0D57 8D BD 19   629      sta DATABUFR
0D5A             630      ;
0D5A A9 1C      631      lda /BUFFER
0D5C 8D BE 19   632      sta DATABUFR+1
0D5F             633      ;
0D5F A9 04      634      lda #FRMTCMD
0D61 8D C1 19   635      sta CMDCODE
0D64             636      ;
0D64 A0 B5      637      ldy #TBLTYPE
0D66 A9 19      638      lda /TBLTYPE
0D68             639      ;
0D68 20 3E 10   640      jsr DORWTS
0D6B             641      ;
0D6B A5 35      642      lda SYNCNT
0D6D 8D 45 19   643      sta SYNCOUNT
0D70             644      ;
0D70 20 F1 0D   645      jsr SHOWUSE
0D73             646      ;
0D73 20 CB 16   647      jsr PRINT

```

```

0D76 00 76      648      hex 0076
0D78 54 00      649      byt CLRCMD,EOLCLR
0D7A D2 E5 F4   650      asc "Return code from Format:  0x"
0D7D F5 F2 EE
0D80 A0 E3 EF
0D83 E4 E5 A0
0D86 E6 F2 EF
0D89 ED A0 C6
0D8C EF F2 ED
0D8F E1 F4 BA
0D92 A0 A0 B0
0D95 F8
0D96 58      651      byt BYT1CMD
0D97 C2 19      652      adr ERRCODE
0D99 AE      653      asc "."
0D9A 50      654      byt RTNCMD
0D9B      655      ;
0D9B 20 95 16   656      jsr GETKEY
0D9E      657      ;
0D9E A9 00      658      ^3      lda #ZERO
0DA0      659      ;
0DA0 60      660      rts
0DA1      661      ;
0DA1      662      ;
0DA1 A9 00      663      SHOWAVG  lda #ZERO
0DA3 8D 55 19   664      sta HDRGAP
0DA6      665      ;
0DA6 38      666      ^1      sec
0DA7      667      ;
0DA7 AD 5A 19   668      lda SYNCOTL
0DAA E9 0F      669      sbc #15
0DAC      670      ;
0DAC 8D 5A 19   671      sta SYNCOTL
0DAF B0 03      672      bcs >2
0DB1      673      ;
0DB1 CE 5B 19   674      dec SYNCOTL+1
0DB4      675      ;
0DB4 EE 55 19   676      ^2      inc HDRGAP
0DB7      677      ;
0DB7 AD 5B 19   678      lda SYNCOTL+1
0DBA D0 EA      679      bne <1
0DBC      680      ;
0DBC AD 5A 19   681      lda SYNCOTL
0DBF C9 0F      682      cmp #15
0DC1 B0 E3      683      bcs <1
0DC3      684      ;
0DC3 20 CB 16   685      jsr PRINT
0DC6 8D      686      byt RETURN
0DC7 C1 F6 E7   687      asc "Avg Header Gap = "
0DCA A0 C8 E5
0DCD E1 E4 E5
0DD0 F2 A0 C7
0DD3 E1 F0 A0
0DD6 BD A0
0DD8 5D      688      byt DEC2CMD
0DD9 55 19      689      adr HDRGAP
0DDB AC A0 D2   690      asc ", Remainder = "
0DDE E5 ED E1
0DE1 E9 EE E4
0DE4 E5 F2 A0
0DE7 BD A0

```

```

0DE9 5D          691          byt DEC2CMD
0DEA 5A 19      692          adr SYNCTOTL
0DEC AF B1 B5   693          asc "/15"
0DEF 50         694          byt RTNCMD
0DF0           695          ;
0DF0 60         696          rts
0DF1           697          ;
0DF1           698          ;
0DF1 20 CB 16   699  SHOWUSE  jsr PRINT
0DF4 1C 73      700          hex 1C73
0DF6 D3 D9 CE   701          asc "SYNCNT=0x"
0DF9 C3 CE D4
0DFC BD B0 F8
0DFF 58         702          byt BYT1CMD
0E00 45 19      703          adr SYNCOUNT
0E02 1A 74      704          hex 1A74
0E04 D2 C5 D4   705          asc "RETRYCNT=0x"
0E07 D2 D9 C3
0E0A CE D4 BD
0E0D B0 F8
0E0F 58         706          byt BYT1CMD
0E10 F8 05      707          adr RETRYCNT
0E12 50         708          byt RTNCMD
0E13           709          ;
0E13 60         710          rts
0E14           711          ;
0E14           712          ;
0E14 20 CB 16   713  QUIT      jsr PRINT
0E17           714          .if DISPLAY=GRPHMODE
0E17           715          byt MODECMD,TEXTMODE
0E17           716          .fi
0E17           717          .if DISPLAY=TX80MODE
0E17           718          byt MODECMD,LV80MODE
0E17           719          .fi
0E17 52 00      720          byt DISPCMD,NORMDISP
0E19 53 00      721          byt SCRNCMD,INITSCRN
0E1B 8D         722          byt RETURN
0E1C 53 01      723          byt SCRNCMD,HOMESCRN
0E1E 50         724          byt RTNCMD
0E1F           725          ;
0E1F 4C D3 03   726          jmp DOSCOLD
0E22           727          ;
0E22           728          ;
0E22           729          ; If CHKSUM not zero set ERRFLAG to 2, CHKSUM error.
0E22           730          ;
0E22           731          ; If SYNCNUM is zero set ERRFLAG to 3, SYNCNUM Overflow.
0E22           732          ;
0E22 A0 00      733  FINDADR  ldy #ZERO
0E24           734          ;
0E24 A9 02      735          lda #2
0E26 8D 47 19   736          sta STAGE
0E29           737          ;
0E29 20 5C 0F   738  ^1      jsr GETDATA
0E2C B0 64      739          bcs >9
0E2E           740          ;
0E2E C9 D5      741  ^2      cmp #$D5
0E30 F0 07      742          beq >3
0E32           743          ;
0E32 EE 4C 19   744          inc SYNCNUM
0E35 D0 F2      745          bne <1
0E37           746          ;

```

```

0E37 F0 53      747      beq >7
0E39           748      ;
0E39 20 5C 0F   749      ^3      jsr GETDATA
0E3C B0 54      750      bcs >9
0E3E           751      ;
0E3E C9 AA      752      cmp #$AA
0E40 F0 07      753      beq >4
0E42           754      ;
0E42 EE 4C 19   755      inc SYNCNUM
0E45 D0 E7      756      bne <2
0E47           757      ;
0E47 F0 43      758      beq >7
0E49           759      ;
0E49 20 5C 0F   760      ^4      jsr GETDATA
0E4C B0 44      761      bcs >9
0E4E           762      ;
0E4E C9 96      763      cmp #$96
0E50 F0 0C      764      beq >5
0E52           765      ;
0E52 EE 4C 19   766      inc SYNCNUM
0E55 F0 35      767      beq >7
0E57           768      ;
0E57 EE 4C 19   769      inc SYNCNUM
0E5A D0 D2      770      bne <2
0E5C           771      ;
0E5C F0 2E      772      beq >7
0E5E           773      ;
0E5E A2 00      774      ^5      ldx #ZERO
0E60           775      ;
0E60 20 5C 0F   776      ^6      jsr GETDATA
0E63 B0 2D      777      bcs >9
0E65           778      ;
0E65 38         779      sec
0E66           780      ;
0E66 2A         781      rol
0E67 8D 4F 19   782      sta TEMP
0E6A           783      ;
0E6A 20 5C 0F   784      jsr GETDATA
0E6D B0 23      785      bcs >9
0E6F           786      ;
0E6F 2D 4F 19   787      and TEMP
0E72 9D 51 19   788      sta HDRVOL,X
0E75           789      ;
0E75 4D 50 19   790      eor CHKSUM
0E78 8D 50 19   791      sta CHKSUM
0E7B           792      ;
0E7B EE 4D 19   793      inc HDRCNT
0E7E           794      ;
0E7E E8         795      inx
0E7F           796      ;
0E7F E0 04      797      cpx #4
0E81 D0 DD      798      bne <6
0E83           799      ;
0E83 AD 50 19   800      lda CHKSUM
0E86 F0 0B      801      beq FINDEND
0E88           802      ;
0E88 A9 02      803      lda #2
0E8A D0 02      804      bne >8      ; always taken
0E8C           805      ;
0E8C A9 03      806      ^7      lda #3
0E8E           807      ;

```

```

0E8E 8D 46 19      808  ^8      sta ERRFLAG
0E91              809  ;
0E91 38           810      sec
0E92              811  ;
0E92 60           812  ^9      rts
0E93              813  ;
0E93              814  ;
0E93 A0 00        815  FINDEND  ldy #ZERO
0E95              816  ;
0E95 A9 03        817      lda #3
0E97 8D 47 19     818      sta STAGE
0E9A              819  ;
0E9A 20 5C 0F     820  ^1      jsr GETDATA
0E9D B0 17        821      bcs >3
0E9F              822  ;
0E9F C9 DE        823  ^2      cmp #$DE
0EA1 D0 F7        824      bne <1
0EA3              825  ;
0EA3 20 5C 0F     826      jsr GETDATA
0EA6 B0 0E        827      bcs >3
0EA8              828  ;
0EA8 C9 AA        829      cmp #$AA
0EAA D0 F3        830      bne <2
0EAC              831  ;
0EAC 20 5C 0F     832      jsr GETDATA
0EAF B0 05        833      bcs >3
0EB1              834  ;
0EB1 C9 EB        835      cmp #$EB
0EB3 D0 EA        836      bne <2
0EB5              837  ;
0EB5 18           838      clc
0EB6              839  ;
0EB6 60           840  ^3      rts
0EB7              841  ;
0EB7              842  ;
0EB7              843  ; If SYNCNUM is zero set ERRFLAG to 4, SYNCNUM Overflow.
0EB7              844  ;
0EB7              845  ; If DATAcnt is zero set ERRFLAG to 5, DATAcnt Overflow.
0EB7              846  ;
0EB7 A0 00        847  FINDDATA  ldy #ZERO
0EB9              848  ;
0EB9 A9 06        849      lda #6
0EBB 8D 47 19     850      sta STAGE
0EBE              851  ;
0EBE              852  ;
0EBE              853  ; Find start data marks.
0EBE              854  ;
0EBE 20 5C 0F     855  ^1      jsr GETDATA
0EC1 B0 7A        856      bcs >9
0EC3              857  ;
0EC3 C9 D5        858  ^2      cmp #$D5
0EC5 F0 07        859      beq >3
0EC7              860  ;
0EC7 EE 4C 19     861      inc SYNCNUM
0ECA D0 F2        862      bne <1
0ECC              863  ;
0ECC F0 66        864      beq >7
0ECE              865  ;
0ECE 20 5C 0F     866  ^3      jsr GETDATA
0ED1 B0 6A        867      bcs >9
0ED3              868  ;

```

```

0ED3 C9 AA      869      cmp #$AA
0ED5 F0 07      870      beq >4
0ED7           871      ;
0ED7 EE 4C 19   872      inc SYNCNUM
0EDA F0 E7      873      beq <2
0EDC           874      ;
0EDC F0 56      875      beq >7
0EDE           876      ;
0EDE 20 5C 0F   877      ^4      jsr GETDATA
0EE1 B0 5A      878      bcs >9
0EE3           879      ;
0EE3 C9 AD      880      cmp #$AD
0EE5 F0 0C      881      beq >1
0EE7           882      ;
0EE7 EE 4C 19   883      inc SYNCNUM
0EEA F0 48      884      beq >7
0EEC           885      ;
0EEC EE 4C 19   886      inc SYNCNUM
0EEF D0 D2      887      bne <2
0EF1           888      ;
0EF1 F0 41      889      beq >7
0EF3           890      ;
0EF3           891      ;
0EF3           892      ; Read data and look for slip marks.
0EF3           893      ;
0EF3 20 5C 0F   894      ^1      jsr GETDATA
0EF6 B0 45      895      bcs >9
0EF8           896      ;
0EF8 C9 DE      897      ^2      cmp #$DE
0EFA F0 07      898      beq >3
0EFC           899      ;
0EFC 20 3E 0F   900      jsr DOCOUNT
0EFF 90 F2      901      bcc <1
0F01           902      ;
0F01 B0 34      903      bcs >8
0F03           904      ;
0F03 20 5C 0F   905      ^3      jsr GETDATA
0F06 B0 35      906      bcs >9
0F08           907      ;
0F08 C9 AA      908      cmp #$AA
0F0A F0 0B      909      beq >4
0F0C           910      ;
0F0C 48         911      pha
0F0D           912      ;
0F0D A9 DE      913      lda #$DE
0F0F           914      ;
0F0F 20 3E 0F   915      jsr DOCOUNT
0F12 B0 23      916      bcs >8
0F14           917      ;
0F14 68         918      pla
0F15           919      ;
0F15 90 E1      920      bcc <2      ; always taken
0F17           921      ;
0F17 20 5C 0F   922      ^4      jsr GETDATA
0F1A B0 21      923      bcs >9
0F1C           924      ;
0F1C C9 EB      925      cmp #$EB
0F1E F0 12      926      beq >5
0F20           927      ;
0F20 48         928      pha
0F21           929      ;

```

```

0F21 A9 DE      930      lda #$DE
0F23           931      ;
0F23 20 3E 0F   932      jsr DOCOUNT
0F26 B0 0F      933      bcs >8
0F28           934      ;
0F28 A9 AA      935      lda #$AA
0F2A           936      ;
0F2A 20 3E 0F   937      jsr DOCOUNT
0F2D B0 08      938      bcs >8
0F2F           939      ;
0F2F 68         940      pla
0F30           941      ;
0F30 90 C6      942      bcc <2                ; always taken
0F32           943      ;
0F32 18         944      ^5      clc
0F33           945      ;
0F33 60         946      rts
0F34           947      ;
0F34 A9 04      948      ^7      lda #4
0F36           949      ;
0F36 2C 00 00   950      bit *-*
0F39           951      dfs !-2
0F37           952      ;
0F37 A9 05      953      ^8      lda #5
0F39           954      ;
0F39 8D 46 19   955      sta ERRFLAG
0F3C           956      ;
0F3C 38         957      sec
0F3D           958      ;
0F3D 60         959      ^9      rts
0F3E           960      ;
0F3E           961      ;
0F3E 8E 5C 19   962      DOCOUNT stx SAVX
0F41           963      ;
0F41 AA         964      tax
0F42           965      ;
0F42 BD 00 1B   966      lda RDNIBL-$96,X
0F45           967      ;
0F45 4D 50 19   968      eor CHKSUM
0F48 8D 50 19   969      sta CHKSUM
0F4B           970      ;
0F4B AE 5C 19   971      ldx SAVX
0F4E           972      ;
0F4E EE 58 19   973      inc DATACNT
0F51 D0 05      974      bne >1
0F53           975      ;
0F53 EE 59 19   976      inc DATACNT+1
0F56 F0 02      977      beq >2
0F58           978      ;
0F58 18         979      ^1      clc
0F59           980      ;
0F59 60         981      rts
0F5A           982      ;
0F5A 38         983      ^2      sec
0F5B           984      ;
0F5B 60         985      rts
0F5C           986      ;
0F5C           987      ;
0F5C           988      ; If data runs out set ERRFLAG to 1, GENPTR Overflow.
0F5C           989      ;
0F5C 18         990      GETDATA clc

```



```
0F5D          991  ;
0F5D B1 EE    992          lda (GENPTR),Y
0F5F          993  ;
0F5F E6 EE    994          inc GENPTR
0F61 D0 0D    995          bne >1
0F63          996  ;
0F63 E6 EF    997          inc GENPTR+1
0F65          998  ;
0F65 CE 49 19 999          dec TRACKSIZ
0F68 D0 06    1000         bne >1
0F6A          1001 ;
0F6A A9 01    1002         lda #1
0F6C 8D 46 19 1003         sta ERRFLAG
0F6F          1004 ;
0F6F 38       1005         sec
0F70          1006 ;
0F70 60       1007 ^1      rts
0F71          1008 ;
0F71          1009 ;
0F71          1010         icl "SCAN2.L"
```

```
LLOAD SCAN2.L,A$4000
```

```

0F71          1          ttl "Track Scan Source Code, SCAN2.L"
0F71          2      ;
0F71          3      ;
0F71          4      ; SCAN2.L
0F71          5      ;
0F71          6      ;
0F71 90 36     7  SHOWMENU bcc >1
0F73          8      ;
0F73 A0 76     9          ldy #VSNBUFR
0F75 A9 19    10         lda /VSNBUFR
0F77          11     ;
0F77 20 3A 10 12         jsr READVSN
0F7A          13     ;
0F7A 20 CB 16 14         jsr PRINT
0F7D 53 01    15         byt SCRNCMD,HOMESCRN
0F7F 8D       16         byt RETURN
0F80 55       17         byt CNTRCMD
0F81 C1 F0 F0 18         asc "Apple ][ Track Scan"
0F84 EC E5 A0
0F87 DD DB A0
0F8A D4 F2 E1
0F8D E3 EB A0
0F90 D3 E3 E1
0F93 EE
0F94 8D 8D     19         byt RETURN,RETURN
0F96 0A       20         hex 0A
0F97 56 00    21         byt BUFRCMD,DIRECT
0F99 76 19    22         adr VSNBUFR
0F9B 66       23         hex 66
0F9C 55       24         byt CNTRCMD
0F9D CF F0 F4 25         asc "Option Menu"
0FA0 E9 EF EE
0FA3 A0 CD E5
0FA6 EE F5
0FA8 50       26         byt RTNCMD
0FA9          27     ;
0FA9 20 CB 16 28     ^1   jsr PRINT
0FAC 69 04    29         hex 6904
0FAE 54 00    30         byt CLRCMD,EOLCLR
0FB0 B1 A0 AD 31         asc "1 - Set Slot/Drive/Phase ("
0FB3 A0 D3 E5
0FB6 F4 A0 D3
0FB9 EC EF F4
0FBC AF C4 F2
0FBF E9 F6 E5
0FC2 AF D0 E8
0FC5 E1 F3 E5
0FC8 A0 A8
0FCA 5C       32         byt DEC1CMD
0FCB 4E 19    33         adr SLOT
0FCD AF       34         asc "/"
0FCE 5D       35         byt DEC2CMD
0FCF B7 19    36         adr DRIVE
0FD1 AF       37         asc "/"
0FD2 5D       38         byt DEC2CMD
0FD3 BF 19    39         adr IOCBPHAS
0FD5 A9       40         asc ")"
0FD6 6B 04    41         hex 6B04
0FD8 54 00    42         byt CLRCMD,EOLCLR
0FDA B2 A0 AD 43         asc "2 - Set Track Start/End ("

```

```

0FDD A0 D3 E5
0FE0 F4 A0 D4
0FE3 F2 E1 E3
0FE6 EB A0 D3
0FE9 F4 E1 F2
0FEC F4 AF C5
0FEF EE E4 A0
0FF2 A8
0FF3 5D          44      byt DEC2CMD
0FF4 4A 19       45      adr STRTRACK
0FF6 AF         46      asc "/"
0FF7 5D         47      byt DEC2CMD
0FF8 4B 19       48      adr ENDTRACK
0FFA A9         49      asc ")"
0FFB 6D 04       50      hex 6D04
0FFD B3 A0 AD    51      asc "3 - Scan Tracks"
1000 A0 D3 E3
1003 E1 EE A0
1006 D4 F2 E1
1009 E3 EB F3
100C 6F 04       52      hex 6F04
100E 54 00       53      byt CLRCMD,EOLCLR
1010 B4 A0 AD    54      asc "4 - Format Volume ("
1013 A0 C6 EF
1016 F2 ED E1
1019 F4 A0 D6
101C EF EC F5
101F ED E5 A0
1022 A8
1023 5D          55      byt DEC2CMD
1024 43 19       56      adr ENDTRK
1026 A0 F4 F2    57      asc " tracks)"
1029 E1 E3 EB
102C F3 A9
102E 71 04       58      hex 7104
1030 B5 A0 AD    59      asc "5 - Quit"
1033 A0 D1 F5
1036 E9 F4
1038 50          60      byt RTNCMD
1039             61      ;
1039 60          62      rts
103A             63      ;
103A             64      ;
103A 38          65      READVSN  sec
103B             66      ;
103B 6C E1 03    67      jmp (RDCLKVSN)
103E             68      ;
103E             69      ;
103E             70      ; Page 3 entry point.  A-reg and Y-reg point to the IOB.
103E             71      ; Copy the USRBUF address for use by all disk controllers.
103E             72      ;
103E 08          73      DORWTS   php
103F             74      ;
103F 78          75      sei
1040             76      ;
1040 84 4A        77      sty IOBADR
1042 85 4B        78      sta IOBADR+1
1044             79      ;
1044 A0 08        80      ldy #USRBUF-TBLTYPE
1046             81      ;
1046 B1 4A        82      lda (IOBADR),Y

```

```

1048 85 3E      83      sta BUFADR2Z
104A           84      ;
104A C8        85      iny
104B           86      ;
104B B1 4A     87      lda (IOBADR),Y
104D 85 3F     88      sta BUFADR2Z+1
104F           89      ;
104F           90      ;
104F           91      ; One cannot assume finding a different slot number is
104F           92      ; really for a Disk ][ device. Other devices must set
104F           93      ; ERRCODE, VOLFND, and DRVFND, and mask track with
104F           94      ; TRKMASK (0x3F).
104F           95      ;
104F A0 01     96      ldy #SNUM16-TBLTYPE
1051           97      ;
1051 B1 4A     98      lda (IOBADR),Y
1053 85 2B     99      sta SLOT16Z
1055           100     ;
1055 A0 0F     101     ldy #SLOTFND-TBLTYPE
1057           102     ;
1057 91 4A     103     sta (IOBADR),Y
1059           104     ;
1059 4A        105     lsr
105A 4A        106     lsr
105B 4A        107     lsr
105C 4A        108     lsr
105D           109     ;
105D AA        110     tax
105E           111     ;
105E 38        112     sec
105F           113     ;
105F 2A        114     rol
1060 8D 01 13  115     sta DISKJMP+1
1063           116     ;
1063 BD B0 12  117     lda SCRCHTBL-1,X
1066 D0 31     118     bne >4
1068           119     ;
1068 DE B0 12  120     dec SCRCHTBL-1,X
106B           121     ;
106B 9D 78 04  122     sta DRV0TRK,X
106E 9D F8 04  123     sta DRV1TRK,X
1071           124     ;
1071 9D 78 06  125     sta DRV0PHAS,X
1074 9D F8 06  126     sta DRV1PHAS,X
1077           127     ;
1077           128     ;
1077           129     ; Build RDNIBL and WRNIBL tables. For a given X-reg, if
1077           130     ; A-reg becomes zero, save the X-reg indexed by the Y-reg
1077           131     ; for WRNIBL and save the Y-reg indexed by the X-reg for
1077           132     ; RDNIBL. The useful X-reg values are from 0x16 to 0x7F.
1077           133     ;
1077 A8         134     tay
1078           135     ;
1078 86 3C     136     ^1 stx ROMTEMPZ
107A           137     ;
107A 8A       138     txa
107B 0A       139     asl
107C           140     ;
107C           141     ;
107C           142     ; X-reg values that fail here:
107C           143     ;

```

```

107C      144 ; 00 01 02 04 05 08 09 0A 10 11 12 14 15 20 21 22 24
107C      145 ; 25 28 29 2A 40 41 42 44 45 48 49 4A 50 51 52 54 55
107C      146 ;
107C 24 3C 147          bit ROMTEMPZ
107E F0 16 148          beq >3
1080      149 ;
1080 05 3C 150          ora ROMTEMPZ
1082 49 FF 151          eor #NEGONE
1084 29 7E 152          and #$7E
1086      153 ;
1086      154 ;
1086      155 ; X-reg values that fail here:
1086      156 ;
1086      157 ; 03 06 07 0B 0C 0D 0E 0F 13 18 19 1C 23 30 31
1086      158 ; 38 43 46 47 4C 58 60 61 62 63 64 68 70 71 78
1086      159 ;
1086 B0 0E 160 ^2          bcs >3
1088      161 ;
1088 4A      162          lsr
1089 D0 FB 163          bne <2
108B      164 ;
108B 8A      165          txa
108C 09 80 166          ora #$80
108E 99 56 1B 167          sta WRNIBL,Y
1091      168 ;
1091 98      169          tya
1092 9D 80 1B 170          sta RDNIBL-$16,X
1095      171 ;
1095 C8      172          iny
1096      173 ;
1096 E8      174 ^3          inx
1097 10 DF 175          bpl <1
1099      176 ;
1099      177 ;
1099      178 ; Enter the RWTS routine for the requested slot number.
1099      179 ;
1099 A6 2B 180 ^4          ldx SLOT16Z
109B      181 ;
109B A4 4A 182          ldy IOBADR
109D A5 4B 183          lda IOBADR+1
109F      184 ;
109F 20 00 13 185          jsr DISKJMP
10A2      186 ;
10A2 2C FF CF 187          bit CLRROM
10A5      188 ;
10A5 A0 0D 189          ldy #ERRCODE-TBLTYPE
10A7      190 ;
10A7 B1 4A 191          lda (IOBADR),Y
10A9      192 ;
10A9 B0 03 193          bcs >5
10AB      194 ;
10AB 28      195          plp
10AC      196 ;
10AC 18      197          clc
10AD      198 ;
10AD 60      199          rts
10AE      200 ;
10AE 28      201 ^5          plp
10AF      202 ;
10AF 38      203          sec
10B0      204 ;

```

```

10B0 60          205          rts
10B1          206      ;
10B1          207      ;
10B1          208      ; RWTS handler routine for Disk ][.
10B1          209      ;
10B1          210      ; Check whether data is changing on this controller card
10B1          211      ; even though the drive motor is currently off but still
10B1          212      ; possibly spinning.
10B1          213      ;
10B1 8E B6 19    214  RWTSENT  stx SNUM16
10B4          215      ;
10B4 20 81 13    216          jsr SETREAD          ; ensure read mode is enabled
10B7          217      ;
10B7 A9 EF      218          lda #MOTONTIM
10B9 85 3C      219          sta MOTORTIM
10BB          220      ;
10BB A9 D8      221          lda /MOTONTIM
10BD 85 3D      222          sta MOTORTIM+1
10BF          223      ;
10BF A9 00      224          lda #ZERO
10C1 85 2C      225          sta DRVFLAG
10C3          226      ;
10C3 A0 08      227          ldy #8
10C5          228      ;
10C5 BD 8C C0    229      ^1      lda STROBE,X
10C8          230      ;
10C8 20 6D 15    231          jsr WAIT24
10CB          232      ;
10CB DD 8C C0    233          cmp STROBE,X
10CE D0 05      234          bne >2
10D0          235      ;
10D0 88          236          dey
10D1 D0 F2      237          bne <1
10D3          238      ;
10D3 E6 2C      239          inc DRVFLAG          ; data not changing
10D5          240      ;
10D5          241      ;
10D5          242      ; Start the motor and select the requested drive.
10D5          243      ;
10D5 BD 89 C0    244      ^2      lda MOTORON,X
10D8          245      ;
10D8 A0 02      246          ldy #DNUM-TBLTYPE
10DA          247      ;
10DA B1 4A      248          lda (IOBADR),Y
10DC 8D B7 19    249          sta DNUM
10DF          250      ;
10DF C9 01      251          cmp #DRIVE1
10E1 F0 01      252          beq >3
10E3          253      ;
10E3 E8          254          inx
10E4          255      ;
10E4 BC 8A C0    256      ^3      ldy DRV0EN,X
10E7          257      ;
10E7          258      ;
10E7          259      ; If the drive number has changed save the new drive
10E7          260      ; number in the IOB and wait 180 msec for the old drive
10E7          261      ; to come to rest.
10E7          262      ;
10E7 A0 10      263          ldy #DRVFND-TBLTYPE
10E9          264      ;
10E9 D1 4A      265          cmp (IOBADR),Y

```

```

10EB F0 0C      266      beq >5
10ED           267      ;
10ED 91 4A      268      sta (IOBADR),Y
10EF           269      ;
10EF A0 08      270      ldy #8
10F1           271      ;
10F1 20 5C 15   272      ^4      jsr MSWAIT
10F4           273      ;
10F4 88         274      dey
10F5 D0 FA      275      bne <4
10F7           276      ;
10F7 E6 2C      277      inc DRVFLAG          ; drive changed
10F9           278      ;
10F9           279      ;
10F9           280      ; If there was no data changing on this controller card or
10F9           281      ; the drive number has changed wait for this drive motor
10F9           282      ; to come up to speed.
10F9           283      ;
10F9 A5 2C      284      ^5      lda DRVFLAG
10FB F0 07      285      beq >7
10FD           286      ;
10FD 20 5C 15   287      ^6      jsr MSWAIT
1100           288      ;
1100 24 3D      289      bit MOTORTIM+1
1102 30 F9      290      bmi <6
1104           291      ;
1104           292      ;
1104           293      ; Initialize PHASE from the selected saved PHASE location
1104           294      ; if it is not zero. If it is zero, initialize the saved
1104           295      ; PHASE location with #DFLTPHAS.
1104           296      ;
1104 20 C9 14    297      ^7      jsr GENINDEX
1107           298      ;
1107 BD 78 06    299      lda DRV0PHAS,X
110A D0 05      300      bne >8
110C           301      ;
110C A9 04      302      lda #DFLTPHAS
110E 9D 78 06    303      sta DRV0PHAS,X
1111           304      ;
1111 85 34      305      ^8      sta PHASE
1113           306      ;
1113           307      ;
1113           308      ; Obtain the requested IOCB PHASE value. If it is zero,
1113           309      ; use PHASE. Range check the value in A-reg.
1113           310      ;
1113 A0 0A      311      ldy #IOCBPHAS-TBLTYPE
1115           312      ;
1115 B1 4A      313      lda (IOBADR),Y
1117 D0 02      314      bne >9
1119           315      ;
1119 A5 34      316      lda PHASE
111B           317      ;
111B C9 11      318      ^9      cmp #PHASMAX+1
111D B0 7A      319      bcs SYNERR
111F           320      ;
111F           321      ;
111F           322      ; Compare the requested PHASE value to the saved PHASE
111F           323      ; value in PHASE. If they differ move the disk head to
111F           324      ; track 0 using the value already in PHASE. Then set PHASE
111F           325      ; to the requested PHASE value.
111F           326      ;

```

```

111F 91 4A      327      sta (IOBADR),Y
1121           328      ;
1121 C5 34      329      cmp PHASE
1123 F0 0A      330      beq >1
1125           331      ;
1125 48          332      pha
1126 9D 78 06   333      sta DRV0PHAS,X
1129           334      ;
1129 20 DB 14    335      jsr MOVHEAD0
112C           336      ;
112C 68          337      pla
112D 85 34      338      sta PHASE
112F           339      ;
112F           340      ;
112F           341      ; Position the disk head over the requested track using
112F           342      ; the PHASE value, verify the requested command, and
112F           343      ; process it.
112F           344      ;
112F 20 D4 14    345      ^1      jsr MOVHEADN
1132           346      ;
1132 A0 0C       347      ldy #CMDCODE-TBLTYPE
1134           348      ;
1134 B1 4A       349      lda (IOBADR),Y
1136 F0 0F       350      beq >2          ; RWTSSEEK
1138           351      ;
1138 C9 01       352      cmp #RWTSREAD
113A F0 32       353      beq >7
113C           354      ;
113C C9 02       355      cmp #RWTSWRIT
113E F0 0A       356      beq >3
1140           357      ;
1140 C9 04       358      cmp #RWTSFRMT
1142 D0 03       359      bne >2
1144           360      ;
1144 4C 01 12     361      jmp DISKFMT
1147           362      ;
1147 4C F0 11     363      ^2      jmp RWTSEXIT
114A           364      ;
114A           365      ;
114A           366      ; A write command, so preinibblize the data first.
114A           367      ;
114A           368      ; The preinibblize routine converts 256 bytes pointed at by
114A           369      ; BUFADR2Z to 342 6-bit nibbles of the form 00XXXXXX.
114A           370      ;
114A           371      ; First clear NBUF2.
114A           372      ;
114A A2 55       373      ^3      ldx #NBUF2SIZ-1
114C           374      ;
114C A9 00       375      lda #ZERO
114E           376      ;
114E 9D 00 1B    377      ^4      sta NBUF2,X
1151           378      ;
1151 CA          379      dex
1152 10 FA       380      bpl <4
1154           381      ;
1154           382      ;
1154           383      ; Process BUFADR2Z.
1154           384      ;
1154 A0 02       385      ldy #2
1156           386      ;
1156 A2 00       387      ^5      ldx #ZERO

```



```

1158          388 ;
1158 88        389 ^6      dey
1159          390 ;
1159 B1 3E     391          lda (BUFADR2Z),Y
115B          392 ;
115B          393 ;
115B          394 ; Shift low order two bits into NBUF2.
115B          395 ;
115B 4A        396          lsr
115C 3E 00 1B 397          rol NBUF2,X
115F          398 ;
115F 4A        399          lsr
1160 3E 00 1B 400          rol NBUF2,X
1163          401 ;
1163          402 ;
1163          403 ; Put low order six bits into NBUF1.
1163          404 ;
1163 99 00 1A   405          sta NBUF1,Y
1166          406 ;
1166 E8        407          inx
1167          408 ;
1167 E0 56     409          cpx #NBUF2SIZ
1169 D0 ED     410          bne <6
116B          411 ;
116B 98        412          tya
116C D0 E8     413          bne <5
116E          414 ;
116E          415 ;
116E          416 ; Set up for one disk head recalibration, 2 track seeks,
116E          417 ; and 32 sector retries.
116E          418 ;
116E A9 02     419 ^7      lda #2
1170 8D F8 04  420          sta RECALCNT
1173          421 ;
1173 A9 02     422 ^1      lda #2
1175 8D 78 05  423          sta SEEKCNT
1178          424 ;
1178 A9 20     425          lda #32
117A 8D F8 05  426          sta RETRYCNT
117D          427 ;
117D          428 ;
117D          429 ; The time it takes WRITADR to write SLPMARK3 and a
117D          430 ; SYNCMARK and call WRITSCTR is 80 usecs. The time it
117D          431 ; takes READADR to return from reading SLPMARK2 and
117D          432 ; check CMDCODE for RWTSREAD before calling WRITSCTR is
117D          433 ; 72 usecs. Both READSCTR and WRITSCTR have enough time.
117D          434 ;
117D 20 40 14   435 ^2      jsr READADR
1180 90 1B     436          bcc >5
1182          437 ;
1182 CE F8 05   438 ^3      dec RETRYCNT
1185 D0 F6     439          bne <2
1187          440 ;
1187          441 ;
1187          442 ; Recalibrate the disk head. Set the disk head track as
1187          443 ; if it was on track 48. Move the disk head to track 0,
1187          444 ; then to the requested track.
1187          445 ;
1187 A9 40       446 ^4      lda #RWDRVERR          ; get bad drive error
1189          447 ;
1189 CE F8 04   448          dec RECALCNT

```

```

118C F0 43      449      beq RWTSEERR
118E           450      ;
118E 20 B6 14   451      jsr SAVETRKK
1191 20 DB 14   452      jsr MOVHEAD0
1194           453      ;
1194 20 D4 14   454      jsr MOVHEADN
1197 10 DA      455      bpl <1                ; always taken
1199           456      ;
1199           457      ;
1199 A9 30      458      SYNERR      lda #RWSYNERR
119B B0 56      459      bcs ERREXIT
119D           460      ;
119D           461      ;
119D           462      ; Check address field for requested track.
119D           463      ;
119D A5 2E      464      ^5      lda TRKFNDZ
119F C5 2A      465      cmp CURTRKZ
11A1 F0 0D      466      beq >6
11A3           467      ;
11A3 CE 78 05   468      dec SEEKCNT
11A6 F0 DF      469      beq <4
11A8           470      ;
11A8           471      ;
11A8           472      ; Set the disk head to the track found, then to requested
11A8           473      ; track.
11A8           474      ;
11A8 20 A9 14   475      jsr SAVETRKK
11AB           476      ;
11AB 20 D4 14   477      jsr MOVHEADN
11AE 10 CD      478      bpl <2                ; always taken
11B0           479      ;
11B0           480      ;
11B0           481      ; Save the found volume number in the IOB.  If the
11B0           482      ; requested volume is not 0 check for volume mismatch.
11B0           483      ;
11B0 A0 0E      484      ^6      ldy #VOLFND-TBLTYPE
11B2           485      ;
11B2 A5 2F      486      lda VOLFNDZ
11B4 91 4A      487      sta (IOBADR),Y
11B6           488      ;
11B6           489      ;
11B6           490      ; Check for correct sector using the interleave table.
11B6           491      ;
11B6 A0 05      492      ldy #SNUM-TBLTYPE
11B8           493      ;
11B8 B1 4A      494      lda (IOBADR),Y
11BA A8         495      tay
11BB           496      ;
11BB B9 A5 19   497      lda INTRLEAV,Y
11BE C5 2D      498      cmp SECFNDZ
11C0 D0 BB      499      bne <2
11C2           500      ;
11C2           501      ;
11C2           502      ; Now at requested sector for read or write operation.
11C2           503      ;
11C2 A0 0C      504      ldy #CMDCODE-TBLTYPE
11C4           505      ;
11C4 B1 4A      506      lda (IOBADR),Y
11C6 C9 01      507      cmp #RWTSREAD
11C8 F0 0A      508      beq >1
11CA           509      ;

```

```

11CA      510 ;
11CA      511 ; Write the nibblized data to the requested sector.
11CA      512 ;
11CA 20 11 13 513      jsr WRITSCTR
11CD 90 21 514      bcc RWTSEXIT
11CF      515 ;
11CF      516 ;
11CF A9 10 517 RWPERR    lda #RWPROTER      ; write protect error
11D1      518 ;
11D1 38 519 RWTSEERR    sec
11D2 B0 1F 520      bcs ERREXIT      ; always taken
11D4      521 ;
11D4      522 ;
11D4      523 ; Read the requested sector and convert the disk nibbles to
11D4      524 ; bytes. The post-nibblize routine converts 342 nibbles of
11D4      525 ; the form 00XXXXXX to eight bit data bytes. The nibbles
11D4      526 ; are stored in NBUF1 and NBUF2, and the 8-bit bytes are
11D4      527 ; stored at BUFADR2Z.
11D4      528 ;
11D4 20 03 14 529 ^1      jsr READSCTR
11D7 B0 A9 530      bcs <3
11D9      531 ;
11D9 A0 00 532      ldy #ZERO
11DB      533 ;
11DB A2 56 534 ^2      ldx #NBUF2SIZ
11DD      535 ;
11DD CA 536 ^3      dex
11DE 30 FB 537      bmi <2
11E0      538 ;
11E0      539 ;
11E0      540 ; Get nibble and shift in low order two bits from NBUF2.
11E0      541 ;
11E0 B9 00 1A 542      lda NBUF1,Y
11E3      543 ;
11E3 5E 00 1B 544      lsr NBUF2,X
11E6 2A 545      rol
11E7      546 ;
11E7 5E 00 1B 547      lsr NBUF2,X
11EA 2A 548      rol
11EB      549 ;
11EB      550 ;
11EB      551 ; Store 8-bit byte in data buffer and continue until done.
11EB      552 ;
11EB 91 3E 553      sta (BUFADR2Z),Y
11ED      554 ;
11ED C8 555      iny
11EE D0 ED 556      bne <3
11F0      557 ;
11F0      558 ;
11F0      559 ; Exit RWTS with no error. Clear C-flag.
11F0      560 ;
11F0 18 561 RWTSEXIT clc
11F1      562 ;
11F1 A9 00 563      lda #RWNOERR
11F3      564 ;
11F3      565 ;
11F3      566 ; Exit RWTS with error in A-reg and C-flag set.
11F3      567 ;
11F3 A0 0D 568 ERREXIT    ldy #ERRCODE-TBLTYPE
11F5      569 ;
11F5 91 4A 570      sta (IOBADR),Y

```

```

11F7          571 ;
11F7 A6 2B    572         ldx SLOT16Z
11F9          573 ;
11F9 BC 88 C0 574         ldy MOTOROFF,X
11FC          575 ;
11FC 60       576         rts
11FD          577 ;
11FD          578 ;
11FD          579 ; Handle write protect sense error.
11FD          580 ;
11FD 68       581 DORWPERR pla
11FE 68       582         pla
11FF          583 ;
11FF B0 CE    584         bcs RWPERR           ; always taken
1201          585 ;
1201          586 ;
1201          587 ; Routine to format a volume in the requested drive.
1201          588 ; Initialize VOLUMEZ from VOLEXPT, ODDBITSZ for WRITADR,
1201          589 ; and RECALCNT for one recalibration.
1201          590 ;
1201 A0 03     591 DISKFMT ldy #VOLEXPT-TBLTYPE
1203          592 ;
1203 B1 4A     593         lda (IOBADR),Y
1205 85 41     594         sta VOLUMEZ
1207          595 ;
1207 A9 AA     596         lda #ODDBITS
1209 85 3E     597         sta ODDBITSZ
120B          598 ;
120B A9 02     599         lda #2
120D 8D F8 04 600         sta RECALCNT
1210          601 ;
1210          602 ;
1210          603 ; Begin with MAXSYNC sync bytes and set the disk head
1210          604 ; location to track 48. Move the disk head to track 0
1210          605 ; and then to track 2 to ensure disk head stability.
1210          606 ;
1210 A9 20     607 ^1      lda #MAXSYNC
1212 85 35     608         sta SYNCNT
1214          609 ;
1214 20 B6 14   610         jsr SAVETRKKX
1217 20 DB 14   611         jsr MOVHEAD0
121A          612 ;
121A A9 02     613         lda #2
121C 20 DD 14   614         jsr MOVHEAD
121F          615 ;
121F          616 ;
121F          617 ; Initialize the NBUF1 and NBUF2 nibble buffers. NBUF2
121F          618 ; must follow NBUF1. The time to clear these buffers will
121F          619 ; help to stabilize the disk head on track 2.
121F          620 ;
121F A9 00     621         lda #ZERO
1221 A8        622         tay
1222          623 ;
1222 99 00 1A   624 ^2      sta NBUF1,Y
1225 99 56 1A   625         sta NBUF1+NBUF2SIZ,Y
1228          626 ;
1228 C8        627         iny
1229 D0 F7      628         bne <2
122B          629 ;
122B          630 ;
122B          631 ; Initialize TRACKZ and format only track 0 to establish a

```

```

122B      632 ; working value for SYNCNT. TRACKFMT returns A-reg = 0.
122B      633 ; Give SYNCNT a final adjustment and begin the disk format.
122B      634 ;
122B 85 40 635      sta TRACKZ
122D      636 ;
122D 20 4F 12 637      jsr TRACKFMT
1230 B0 14 638      bcs >4
1232      639 ;
1232 C6 35 640      dec SYNCNT
1234 C6 35 641      dec SYNCNT
1236      642 ;
1236      643 ;
1236      644 ; Format the selected track and increment track number
1236      645 ; until ENDTRK.
1236      646 ;
1236 20 4F 12 647 ^3      jsr TRACKFMT
1239 B0 0B 648      bcs >4
123B      649 ;
123B E6 40 650      inc TRACKZ
123D      651 ;
123D A5 40 652      lda TRACKZ
123F CD 43 19 653      cmp ENDTRK
1242 D0 F2 654      bne <3
1244      655 ;
1244 F0 AA 656      beq RWTSEXIT      ; always taken
1246      657 ;
1246      658 ;
1246      659 ; Error from TRACKFMT. Try a recalibration, then exit.
1246      660 ;
1246 CE F8 04 661 ^4      dec RECALCNT
1249 D0 C5 662      bne <1
124B      663 ;
124B A9 08 664      lda #RWINITER      ; track init error
124D      665 ;
124D B0 A4 666      bcs ERREXIT      ; always taken
124F      667 ;
124F      668 ;
124F      669 ; Routine to format the selected track. Move the disk head
124F      670 ; to the selected track and allow for 32 track retries.
124F      671 ; Initialize SECTORZ to 0 and mark all sectors as
124F      672 ; unformatted.
124F      673 ;
124F 20 DD 14 674 TRACKFMT jsr MOVHEAD
1252      675 ;
1252 A9 20 676      lda #MAXRETRY
1254 8D F8 05 677      sta RETRYCNT
1257      678 ;
1257 A9 00 679 ^1      lda #ZERO
1259 85 3F 680      sta SECTORZ
125B      681 ;
125B AC 44 19 682      ldy ENDSEC
125E      683 ;
125E 99 8A 19 684 ^2      sta SECMAP-1,Y
1261      685 ;
1261 88 686      dey
1262 D0 FA 687      bne <2
1264      688 ;
1264      689 ;
1264      690 ; Begin with 128 sync bytes before sector 0 address field,
1264      691 ; then use SYNCNT sync bytes before all other sector
1264      692 ; address fields.

```

```

1264          693 ;
1264 A0 80      694      ldy #128
1266          695 ;
1266 2C 00 00    696      bit *-*
1269          697      dfs !-2
1267          698 ;
1267          699 ;
1267          700 ; Write an address field and exit if disk is write
1267          701 ; protected. Then write the data field. Increment sector
1267          702 ; number until NUMSECS.
1267          703 ;
1267 A4 35      704 ^3      ldy SYNCNT
1269          705 ;
1269 20 88 13    706      jsr WRITADR
126C B0 8F      707      bcs DORWPERR
126E          708 ;
126E 20 11 13   709      jsr WRITSCTR
1271          710 ;
1271 E6 3F      711      inc SECTORZ
1273          712 ;
1273 A5 3F      713      lda SECTORZ
1275 CD 44 19   714      cmp ENDSEC
1278 D0 ED      715      bne <3
127A          716 ;
127A          717 ;
127A          718 ; Ensure the time between the last and first sectors is at
127A          719 ; least ( SYNCNT / 2 ) * 100 usecs.
127A          720 ;
127A A5 35      721      lda SYNCNT
127C 4A         722      lsr
127D          723 ;
127D 20 5C 15   724      jsr MSWAIT
1280          725 ;
1280          726 ;
1280          727 ; Read the first address field found. If sector 0 is not
1280          728 ; found, reduce SYNCNT and try again.
1280          729 ;
1280 20 40 14    730      jsr READADR
1283 B0 0C      731      bcs >4
1285          732 ;
1285 A5 2D      733      lda SECFNDZ
1287 F0 14      734      beq >7
1289          735 ;
1289 C6 35      736      dec SYNCNT
128B          737 ;
128B A5 35      738      lda SYNCNT
128D C9 08      739      cmp #MINSYNC
128F 90 05      740      bcc >5          ; cannot retry this error
1291          741 ;
1291          742 ;
1291          743 ; Any error found reading an address or data field, or
1291          744 ; finding a duplicate address field or change in SYNCNT
1291          745 ; will cause the entire track to be reformatted.
1291          746 ;
1291 CE F8 05    747 ^4      dec RETRYCNT
1294 D0 C1      748      bne <1
1296          749 ;
1296 38         750 ^5      sec
1297          751 ;
1297 60         752      rts
1298          753 ;

```

```
1298          754 ;
1298          755 ; Read the next address field and data field.
1298          756 ;
1298 20 40 14 757 ^6      jsr READADR
129B B0 F4   758          bcs <4
129D          759 ;
129D 20 03 14 760 ^7      jsr READSCTR
12A0 B0 EF   761          bcs <4
12A2          762 ;
12A2          763 ;
12A2          764 ; Mark the sector map with the sector found and verify this
12A2          765 ; is not a duplicate sector found.
12A2          766 ;
12A2 A6 2D   767          ldx SECFNDZ
12A4          768 ;
12A4 BD 8B 19 769          lda SECMAP,X
12A7 30 E8   770          bmi <4
12A9          771 ;
12A9 DE 8B 19 772          dec SECMAP,X
12AC          773 ;
12AC C6 3F   774          dec SECTORZ
12AE D0 E8   775          bne <6
12B0          776 ;
12B0 60      777          rts                      ; C-flag is clear
12B1          778 ;
12B1          779 ;
12B1          780          icl "SCAN3.L"
```

LLOAD SCAN3.L,A\$4000

```

12B1      1          ttl "Track Scan Source Code, SCAN3.L"
12B1      2      ;
12B1      3      ;
12B1      4      ; SCAN3.L
12B1      5      ;
12B1      6      ;
12B1 00    7      SCRCHTBL byt ZERO
12B2 00    8          byt ZERO
12B3 00    9          byt ZERO
12B4 00   10          byt ZERO
12B5 00   11          byt ZERO
12B6 00   12          byt ZERO
12B7 00   13          byt ZERO
12B8      14      ;
12B8      15      ;
12B8      16          dfs PAGESIZE-*)&NEGONE,ZERO
1300      17      ;
1300      18      ;
1300 6C 03 13 19      DISKJMP  jmp (DISKADRS)
1303      20      ;
1303      21      ;
1303 B1 10 22      DISKADRS  adr RWTSENT
1305 B1 10 23          adr RWTSENT
1307 B1 10 24          adr RWTSENT
1309 B1 10 25          adr RWTSENT
130B B1 10 26          adr RWTSENT
130D B1 10 27          adr RWTSENT
130F B1 10 28          adr RWTSENT
1311      29      ;
1311      30      ;
1311      31      ; Write a sector data field routine.  A sector data field
1311      32      ; consists of preinibbilized data contained in NBUF1 and
1311      33      ; NBUF2.  Check write protect sense from the Disk ][.
1311      34      ;
1311 38    35      WRITSCTR sec
1312      36      ;
1312 BD 8D C0 37          lda LATCH,X          ; enable sequencer latch
1315 BD 8E C0 38          lda DATAIN,X        ; read sequencer latch
1318      39      ;
1318 30 67   40          bmi SETREAD
131A      41      ;
131A      42      ;
131A      43      ; Prepare TEMPZ for later and write HDRSYNC sync bytes and
131A      44      ; data marks 0xD5, 0xAA, and 0xAD to disk.
131A      45      ;
131A AD 00 1B 46          lda NBUF2
131D 85 26   47          sta TEMPZ
131F      48      ;
131F A0 06   49          ldy #HDRSYNC
1321      50      ;
1321 A9 AD   51          lda #DATMARK3
1323 20 B6 13 52          jsr WRITSYNC
1326      53      ;
1326      54      ;
1326      55      ; Checksum is cleared by writing the last byte of NBUF2.
1326      56      ; Apple originally published this routine using a 36 usec
1326      57      ; byte for this value.  I have changed this logic.
1326      58      ;
1326 AE 55 1B 59          ldx NBUF2+NBUF2SIZ-1
1329      60      ;

```



```

1329 A0 55      61      ldy #NBUF2SIZ-1
132B D0 07      62      bne >2                ; always taken
132D           63      ;
132D           64      ;
132D           65      ; Get previous 6-bit nibble and XOR with current nibble to
132D           66      ; form index into the write translate table.
132D           67      ;
132D B9 01 1B    68      ^1      lda NBUF2+1,Y
1330           69      ;
1330 59 00 1B    70      eor NBUF2,Y
1333 AA         71      tax
1334           72      ;
1334 BD 56 1B    73      ^2      lda WRNIBL,X
1337           74      ;
1337 AE B6 19    75      ldx SNUM16
133A           76      ;
133A 9D 8D C0    77      sta LATCH,X          ; load sequencer latch
133D BD 8C C0    78      lda STROBE,X        ; enable sequencer to shift
1340           79      ;
1340 88         80      dey
1341 10 EA       81      bpl <1
1343           82      ;
1343           83      ;
1343           84      ; Write NBUF1 to disk using the same logic.
1343           85      ;
1343 C8         86      iny
1344           87      ;
1344 A5 26       88      lda TEMPZ
1346           89      ;
1346 59 00 1A    90      ^3      eor NBUF1,Y
1349 AA         91      tax
134A           92      ;
134A BD 56 1B    93      lda WRNIBL,X
134D           94      ;
134D AE B6 19    95      ldx SNUM16
1350           96      ;
1350 9D 8D C0    97      sta LATCH,X          ; load sequencer latch
1353 BD 8C C0    98      lda STROBE,X        ; enable sequencer to shift
1356           99      ;
1356 B9 00 1A   100      lda NBUF1,Y
1359          101      ;
1359 C8         102      iny
135A D0 EA      103      bne <3
135C           104      ;
135C F0 00     105      beq >4                ; always taken, waste 3 cycles
135E           106      ;
135E           107      ;
135E           108      ; Write the checksum to disk.
135E           109      ;
135E A8        110      ^4      tay
135F           111      ;
135F B9 56 1B   112      lda WRNIBL,Y
1362           113      ;
1362 20 F0 13   114      jsr WNIBL
1365           115      ;
1365 90 00     116      bcc WRITEXIT          ; always taken, waste 3 cycles
1367           117      ;
1367           118      ;
1367           119      ; Write slip marks 0xDE, 0xAA, and 0xEB to disk.
1367           120      ;
1367 EA       121      WRITEXIT nop          ; waste 2 cycles

```

```

1368 EA          122          nop                      ; waste 2 cycles
1369              123      ;
1369 A9 DE        124          lda #SLPMARK1
136B 20 EF 13    125          jsr WNIBL2
136E              126      ;
136E A9 AA        127          lda #SLPMARK2
1370 20 ED 13    128          jsr WNIBL9
1373              129      ;
1373 A9 EB        130          lda #SLPMARK3
1375 20 ED 13    131          jsr WNIBL9
1378              132      ;
1378              133      ;
1378              134      ; Terminate the address/data footer and clear the C-flag.
1378              135      ;
1378 A9 FF        136          lda #SYNCMARK
137A 20 ED 13    137          jsr WNIBL9
137D              138      ;
137D 20 6D 15    139          jsr WAIT24                ; allow all bits to be written
1380 EA          140          nop                      ; 32 cycles have now passed
1381              141      ;
1381              142      ;
1381              143      ; Return to read mode.
1381              144      ;
1381 BD 8E C0     145  SETREAD  lda DATAIN,X            ; enable sequencer to READ
1384 BD 8C C0     146          lda STROBE,X              ; enable sequencer to shift
1387              147      ;
1387 60           148          rts
1388              149      ;
1388              150      ;
1388              151      ; Write an address data field routine. An address data
1388              152      ; field consists of the volume, track, sector, and checksum
1388              153      ; values. Check write protect sense from the Disk ][.
1388              154      ;
1388 38           155  WRITADR  sec
1389              156      ;
1389 A6 2B        157          ldx SLOT16Z
138B              158      ;
138B BD 8D C0     159          lda LATCH,X                ; enable sequencer latch
138E BD 8E C0     160          lda DATAIN,X              ; read sequencer latch
1391              161      ;
1391 30 EE        162          bmi SETREAD
1393              163      ;
1393              164      ;
1393              165      ; Prepare TEMPZ for later with the address header checksum
1393              166      ; and write Y-reg number of sync bytes. Then write data
1393              167      ; marks 0xD5, 0xAA, and 0x96 to disk.
1393              168      ;
1393 A5 41        169          lda VOLUMEZ
1395 45 40        170          eor TRACKZ
1397 45 3F        171          eor SECTORZ
1399 85 26        172          sta TEMPZ
139B              173      ;
139B A9 96        174          lda #ADRMARK3
139D 20 B6 13    175          jsr WRITSYNC
13A0              176      ;
13A0              177      ;
13A0              178      ; Write volume, track, sector, and checksum to disk.
13A0              179      ;
13A0 A5 41        180          lda VOLUMEZ
13A2 20 DE 13    181          jsr WBYTE
13A5              182      ;

```

```

13A5 A5 40      183      lda TRACKZ
13A7 20 DE 13   184      jsr WBYTE
13AA           185      ;
13AA A5 3F      186      lda SECTORZ
13AC 20 DE 13   187      jsr WBYTE
13AF           188      ;
13AF A5 26      189      lda TEMPZ
13B1 20 DE 13   190      jsr WBYTE
13B4           191      ;
13B4 90 B1      192      bcc WRITEXIT          ; always taken
13B6           193      ;
13B6           194      ;
13B6           195      ; These routines are all time critical. Be careful with
13B6           196      ; page boundries and do not change any instructions.
13B6           197      ;
13B6           198      ; Y-reg contains the number of sync bytes to write.
13B6           199      ;
13B6           200      ; Configure firmware to generate sync bytes. Sync bytes
13B6           201      ; are 40 usecs in length and are written as %1111111100.
13B6           202      ; The first sync byte must be written exactly 40 usecs
13B6           203      ; later and continually written in a 40 usec loop.
13B6           204      ;
13B6 85 2C      205      WRITSYNC sta ADRDATMK
13B8           206      ;
13B8 A9 FF      207      lda #SYNCMARK
13BA           208      ;
13BA 9D 8F C0   209      sta DATAOUT,X      ; enable sequencer to WRITE
13BD 1D 8C C0   210      ora STROBE,X        ; enable sequencer to shift
13C0           211      ;
13C0 48         212      pha                ; waste 3 cycles
13C1 68         213      pla                ; waste 4 cycles
13C2           214      ;
13C2 20 6D 15   215      ^1 jsr WAIT24        ; waste 24 cycles
13C5           216      ;
13C5 9D 8D C0   217      sta LATCH,X        ; load sequencer latch
13C8 1D 8C C0   218      ora STROBE,X        ; enable sequencer to shift
13CB           219      ;
13CB EA        220      nop                ; waste 2 cycles
13CC           221      ;
13CC 88         222      dey
13CD D0 F3      223      bne <1
13CF           224      ;
13CF           225      ;
13CF           226      ; Write address/data marks 0xD5, 0xAA, and TEMP2Z to disk.
13CF           227      ;
13CF A9 D5      228      lda #DATMARK1      ; same as ADRMARK1
13D1 20 ED 13   229      jsr WNIBL9
13D4           230      ;
13D4 A9 AA      231      lda #DATMARK2      ; same as ADRMARK2
13D6 20 ED 13   232      jsr WNIBL9
13D9           233      ;
13D9 A5 2C      234      lda ADRDATMK        ; recall 3rd address/data mark
13DB           235      ;
13DB 18         236      clc
13DC 90 0F      237      bcc WNIBL9        ; always taken
13DE           238      ;
13DE           239      ;
13DE           240      ; Write a byte as two four-bit nibbles to the disk
13DE           241      ; starting with the odd bits.
13DE           242      ;
13DE 48         243      WBYTE pha

```

```

13DF      244 ;
13DF 4A    245      lsr
13E0 05 3E 246      ora ODDBITSZ
13E2      247 ;
13E2 9D 8D C0 248      sta LATCH,X      ; load sequencer latch
13E5 BD 8C C0 249      lda STROBE,X      ; enable sequencer to shift
13E8      250 ;
13E8      251 ;
13E8      252 ; Now write the even bits.
13E8      253 ;
13E8 68    254      pla
13E9 05 3E 255      ora ODDBITSZ
13EB      256 ;
13EB 48    257      pha      ; waste 3 cycles
13EC 68    258      pla      ; waste 4 cycles
13ED      259 ;
13ED      260 ;
13ED      261 ; Wait 9 clock cycles, then write to disk.
13ED      262 ;
13ED 48    263 WNIBL9   pha      ; waste 3 cycles
13EE 68    264      pla      ; waste 4 cycles
13EF      265 ;
13EF      266 ;
13EF      267 ; Wait 2 clock cycles, then write to disk.
13EF      268 ;
13EF 18    269 WNIBL2   clc      ; waste 2 cycles
13F0      270 ;
13F0      271 ;
13F0      272 ; Write nibble to disk.
13F0      273 ;
13F0 9D 8D C0 274 WNIBL   sta LATCH,X      ; load sequencer latch
13F3 BD 8C C0 275      lda STROBE,X      ; enable sequencer to shift
13F6      276 ;
13F6 60    277      rts
13F7      278 ;
13F7      279 ;
13F7      280 ; PHASEON/PHASEOFF table. Time delay uses MSWAIT.
13F7      281 ;
13F7 40 32 2A 282 ONOFFTBL hex 40322A242120
13FA 24 21 20
13FD 1F 1E 1E 283      hex 1F1E1E1D1D1C
1400 1D 1D 1C
1403      284 ;
000C      285 OTBLLEN   equ *-ONOFFTBL
1403      286 ;
1403      287 ;
1403      288 ; READ routine reads nibblized data from the disk and
1403      289 ; stores the data in NBUF1 and NBUF2. Fail after N read
1403      290 ; attempts where N corresponds to the possibility that
1403      291 ; there may be MAXSYNC sync bytes before the data header.
1403      292 ;
1403 A9 56   293 READSCTR  lda #NBUF2SIZ
1405 85 26   294      sta TEMPZ
1407      295 ;
1407 A2 AD   296      ldx #DATMARK3      ; for ADRDATMK
1409      297 ;
1409 A0 28   298      ldy #SYNCBITS/32
140B A9 00   299      lda #ZERO      ; for TEMP2Z
140D      300 ;
140D 20 7B 14 301      jsr READMRKS
1410 30 2C   302      bmi READERR

```

```

1412          303 ;
1412          304 ;
1412          305 ; Read nibbles into NBUF2.
1412          306 ;
1412 C6 26      307 ^6      dec TEMPZ
1414          308 ;
1414 BC 8C C0   309 ^7      ldy STROBE,X          ; read sequencer latch
1417 10 FB     310          bpl <7
1419          311 ;
1419 59 00 1B   312          eor RDNIBL-$96,Y
141C          313 ;
141C A4 26     314          ldy TEMPZ
141E          315 ;
141E 99 00 1B   316          sta NBUF2,Y
1421          317 ;
1421 D0 EF     318          bne <6
1423          319 ;
1423          320 ;
1423          321 ; Now read nibbles into NBUF1.
1423          322 ;
1423 BC 8C C0   323 ^8      ldy STROBE,X          ; read sequencer latch
1426 10 FB     324          bpl <8
1428          325 ;
1428 59 00 1B   326          eor RDNIBL-$96,Y
142B          327 ;
142B A4 26     328          ldy TEMPZ
142D          329 ;
142D 99 00 1A   330          sta NBUF1,Y
1430          331 ;
1430 E6 26     332          inc TEMPZ
1432 D0 EF     333          bne <8
1434          334 ;
1434          335 ;
1434          336 ; Verify the checksum byte.
1434          337 ;
1434 BC 8C C0   338 ^9      ldy STROBE,X          ; read sequencer latch
1437 10 FB     339          bpl <9
1439          340 ;
1439 D9 00 1B   341          cmp RDNIBL-$96,Y
143C F0 28     342          beq READEXIT
143E          343 ;
143E          344 ;
143E 38        345 READERR  sec
143F          346 ;
143F 60        347          rts
1440          348 ;
1440          349 ;
1440          350 ; Read address field. Address field nibbles are odd/even
1440          351 ; encoded. Read over 0x400 disk nibbles before giving up.
1440          352 ;
1440 A2 96      353 READADR  ldx #ADRMARK3          ; for ADRDATMK
1442          354 ;
1442 A0 04      355          ldy #4
1444 98        356          tya          ; for TEMP2Z
1445          357 ;
1445 20 7B 14   358          jsr READMRKS
1448 30 F4     359          bmi READERR
144A          360 ;
144A          361 ;
144A          362 ; Read the four-byte address data field. First read the
144A          363 ; 'odd' bit nibble. C-flag was set previously from the

```

```

144A          364 ; ADRDATMK comparison in READMRKS.
144A          365 ;
144A 85 27     366 ^6      sta TEMP2Z
144C          367 ;
144C BD 8C C0  368 ^7      lda STROBE,X          ; read sequencer latch
144F 10 FB     369          bpl <7
1451          370 ;
1451 2A        371          rol
1452 85 26     372          sta TEMPZ
1454          373 ;
1454          374 ;
1454          375 ; Now read the 'even' bit nibble and merge the two nibbles.
1454          376 ; Store the data byte, then update the checksum and repeat
1454          377 ; until the entire address field is read.
1454          378 ;
1454 BD 8C C0  379 ^8      lda STROBE,X          ; read sequencer latch
1457 10 FB     380          bpl <8
1459          381 ;
1459 25 26     382          and TEMPZ
145B          383 ;
145B 99 2C 00  384          sta ADRFIELD,Y
145E 45 27     385          eor TEMP2Z
1460          386 ;
1460 88        387          dey
1461 10 E7     388          bpl <6
1463          389 ;
1463          390 ;
1463          391 ; Checksum in A-reg must be zero for no error.
1463          392 ;
1463 A8        393          tay
1464 D0 D8     394          bne READERR
1466          395 ;
1466          396 ;
1466          397 READEXIT:
1466          398 ;
1466          399 ; Check for slip mark 1.
1466          400 ;
1466 BD 8C C0  401 ^1      lda STROBE,X          ; read sequencer latch
1469 10 FB     402          bpl <1
146B          403 ;
146B C9 DE     404          cmp #SLPMARK1
146D D0 CF     405          bne READERR
146F          406 ;
146F EA        407          nop                ; waste 2 cycles
1470          408 ;
1470          409 ;
1470          410 ; Check for skip mark 2. Slip mark 3 is not checked.
1470          411 ;
1470 BD 8C C0  412 ^2      lda STROBE,X          ; read sequencer latch
1473 10 FB     413          bpl <2
1475          414 ;
1475 C9 AA      415          cmp #SLPMARK2
1477 D0 C5     416          bne READERR
1479          417 ;
1479 18        418          clc
147A          419 ;
147A 60        420          rts
147B          421 ;
147B          422 ;
147B 86 2C     423 READMRKS stx ADRDATMK          ; save address/data mark 3
147D 85 27     424          sta TEMP2Z          ; save MSB disk nibble counter

```

```

147F      425 ;
147F A6 2B      426      ldx SLOT16Z      ; recall SLOT*16 value
1481      427 ;
1481 88      428 ^1      dey      ; LSB disk nibble counter
1482 D0 04      429      bne >2
1484      430 ;
1484 C6 27      431      dec TEMP2Z      ; MSB disk nibble counter
1486 30 20      432      bmi >6
1488      433 ;
1488      434 ;
1488      435 ; Check for address/data mark 1.
1488      436 ;
1488 BD 8C C0      437 ^2      lda STROBE,X      ; read sequencer latch
148B 10 FB      438      bpl <2
148D      439 ;
148D C9 D5      440 ^3      cmp #ADRMARK1      ; same as DATMARK1
148F D0 F0      441      bne <1
1491      442 ;
1491 EA      443      nop      ; waste 2 cycles
1492      444 ;
1492      445 ;
1492      446 ; Check for address/data mark 2.
1492      447 ;
1492 BD 8C C0      448 ^4      lda STROBE,X      ; read sequencer latch
1495 10 FB      449      bpl <4
1497      450 ;
1497 C9 AA      451      cmp #ADRMARK2      ; same as DATMARK2
1499 D0 F2      452      bne <3
149B      453 ;
149B A0 03      454      ldy #3      ; needed for READADR
149D      455 ;
149D      456 ;
149D      457 ; Check for address/data mark 3 saved in ADRDATMK.
149D      458 ;
149D BD 8C C0      459 ^5      lda STROBE,X      ; read sequencer latch
14A0 10 FB      460      bpl <5
14A2      461 ;
14A2 C5 2C      462      cmp ADRDATMK
14A4 D0 E7      463      bne <3
14A6      464 ;
14A6      465 ;
14A6      466 ; Initialize the checksum to zero.
14A6      467 ;
14A6 A9 00      468      lda #ZERO
14A8      469 ;
14A8 60      470 ^6      rts
14A9      471 ;
14A9      472 ;
14A9      473 ; Routine to save the half-phase value of a drive in a
14A9      474 ; slot dependant location.
14A9      475 ;
14A9      476 ; Do not modify the TRK, DRV, or PHAS table addresses.
14A9      477 ;
14A9      478 ; First calculate the half-phase value based on track
14A9      479 ; and phase value.
14A9      480 ;
14A9 85 26      481 SAVETRK sta TEMPZ
14AB      482 ;
14AB 18      483      clc
14AC      484 ;
14AC A4 34      485      ldy PHASE

```

```

14AE          486 ;
14AE A9 00    487     lda #ZERO
14B0          488 ;
14B0 65 26    489 ^1     adc TEMPZ
14B2          490 ;
14B2 88       491     dey
14B3 D0 FB    492     bne <1
14B5          493 ;
14B5 2C 00 00 494     bit *-*
14B8          495     dfs !-2
14B6          496 ;
14B6 A9 C0    497 SAVETRKY lda #DFLTPHAS*MAXTRACK
14B8          498 ;
14B8 8D 78 04 499     sta FINDTRK
14BB A8       500     tay
14BC          501 ;
14BC          502 ;
14BC          503 ; Retrieve the current track position and save the
14BC          504 ; requested track position in half-phase values.
14BC          505 ;
14BC 20 C9 14 506     jsr GENINDEX
14BF          507 ;
14BF BD 78 04 508     lda DRV0TRK,X
14C2 85 2A    509     sta CURTRKZ
14C4          510 ;
14C4 98       511     tya
14C5 9D 78 04 512     sta DRV0TRK,X
14C8          513 ;
14C8 60       514     rts
14C9          515 ;
14C9          516 ;
14C9          517 ; Calculate the index based on slot and drive.
14C9          518 ;
14C9 A5 2B    519 GENINDEX lda SLOT16Z
14CB          520 ;
14CB 4A       521     lsr
14CC          522 ;
14CC 0D B7 19 523     ora DNUM
14CF          524 ;
14CF 4A       525     lsr
14D0 4A       526     lsr
14D1          527 ;
14D1 6A       528     ror
14D2          529 ;
14D2 AA       530     tax
14D3          531 ;
14D3 60       532     rts
14D4          533 ;
14D4          534 ;
14D4          535 ; Routine to move the Disk ][ head to the track in the
14D4          536 ; A-reg on the current drive.
14D4          537 ;
14D4 A0 04    538 MOVHEADN ldy #TNUM-TBLTYPE
14D6          539 ;
14D6 B1 4A    540     lda (IOBADR),Y
14D8 29 3F    541     and #TRKMASK
14DA          542 ;
14DA 2C 00 00 543     bit *-*
14DD          544     dfs !-2
14DB          545 ;
14DB A9 00    546 MOVHEAD0 lda #ZERO

```



```

14DD          547 ;
14DD 48        548 MOVHEAD pha
14DE          549 ;
14DE 20 E5 14  550          jsr MOVEHD
14E1          551 ;
14E1 68        552          pla
14E2 85 2A     553          sta CURTRKZ
14E4          554 ;
14E4 60        555 RTN16      rts
14E5          556 ;
14E5          557 ;
14E5          558 ; Move the disk head to the requested track in half-phase
14E5          559 ; steps.
14E5          560 ;
14E5 20 A9 14  561 MOVEHD    jsr SAVETRK
14E8          562 ;
14E8 C5 2A     563          cmp CURTRKZ
14EA F0 F8     564          beq RTN16
14EC          565 ;
14EC          566 ;
14EC          567 ; Determine the offsets to the next motor phase based on
14EC          568 ; direction the head needs to move.
14EC          569 ;
14EC A2 FF     570          ldx #NEGONE
14EE 8A        571          txa
14EF          572 ;
14EF B0 03     573          bcs >0
14F1          574 ;
14F1 E8        575          inx
14F2 A9 01     576          lda #1
14F4          577 ;
14F4 8E 78 06  578 ^0        stx NEXTON
14F7 8D F8 06  579          sta NEXTOFF
14FA          580 ;
14FA          581 ;
14FA          582 ; Determine the direction to move and increment/decrement
14FA          583 ; the current position.
14FA          584 ;
14FA A0 00     585          ldy #ZERO
14FC 84 26     586          sty TEMPZ
14FE          587 ;
14FE 20 41 15  588 ^1        jsr CHKPOS
1501          589 ;
1501 38        590          sec
1502          591 ;
1502 A5 2A     592          lda CURTRKZ
1504 ED 78 04  593          sbc FINDTRK
1507 F0 1B     594          beq >6
1509          595 ;
1509 B0 06     596          bcs >2
150B          597 ;
150B 49 FF     598          eor #NEGONE
150D          599 ;
150D E6 2A     600          inc CURTRKZ
150F 90 04     601          bcc >3          ; always taken
1511          602 ;
1511 69 FE     603 ^2        adc #!-2
1513          604 ;
1513 C6 2A     605          dec CURTRKZ
1515          606 ;
1515 C5 26     607 ^3        cmp TEMPZ

```

```

1517 90 02      608      bcc >4
1519           609      ;
1519 A5 26      610      lda TEMPZ
151B           611      ;
151B C9 0C      612      ^4      cmp #OTBLLEN
151D B0 01      613      bcs >5
151F           614      ;
151F A8         615      tay
1520           616      ;
1520 E6 26      617      ^5      inc TEMPZ
1522 D0 DA      618      bne <1      ; always taken
1524           619      ;
1524           620      ;
1524           621      ; At final destination. Lock in half-phase or phase
1524           622      ; position.
1524           623      ;
1524 20 5C 15    624      ^6      jsr MSWAIT
1527           625      ;
1527 A5 2A      626      lda CURTRKZ
1529           627      ;
1529 29 06      628      and #6
152B 05 2B      629      ora SLOT16Z
152D           630      ;
152D AA        631      tax
152E           632      ;
152E A5 2A      633      lda CURTRKZ
1530 4A        634      lsr
1531           635      ;
1531 69 00      636      adc #ZERO
1533 29 03      637      and #3
1535           638      ;
1535 0A        639      asl
1536 05 2B      640      ora SLOT16Z
1538           641      ;
1538 A8         642      tay
1539           643      ;
1539 BD 80 C0    644      lda PHASEOFF,X
153C B9 80 C0    645      lda PHASEOFF,Y
153F           646      ;
153F 90 18      647      bcc >9      ; always taken
1541           648      ;
1541           649      ;
1541           650      ; Select the next motor phase to turn off or turn on.
1541           651      ; Fall into MSWAIT.
1541           652      ;
1541 A5 2A      653      CHKPOS      lda CURTRKZ
1543           654      ;
1543 4A        655      lsr
1544 90 06      656      bcc >7
1546           657      ;
1546 ED 78 06    658      sbc NEXTON
1549           659      ;
1549 38         660      sec
154A B0 04      661      bcs >8
154C           662      ;
154C 6D F8 06    663      ^7      adc NEXTOFF
154F           664      ;
154F 18         665      clc
1550           666      ;
1550 2A         667      ^8      rol
1551           668      ;

```

```
1551 29 07      669      and #7
1553 05 2B      670      ora SLOT16Z
1555           671      ;
1555 AA         672      tax
1556           673      ;
1556 BD 80 C0    674      lda PHASEOFF,X
1559           675      ;
1559 B9 F7 13    676      ^9      lda ONOFFTBL,Y
155C           677      ;
155C           678      ;
155C           679      ; Routine to delay A-reg * 99 + 7 usecs.  This routine
155C           680      ; must reside on the same page.
155C           681      ;
155C 38          682      MSWAIT    sec
155D           683      ;
155D A2 11       684      ^1      ldx #17
155F           685      ;
155F CA         686      ^2      dex
1560 D0 FD       687      bne <2
1562           688      ;
1562 E6 3C       689      inc MOTORTIM
1564 D0 02       690      bne >3
1566           691      ;
1566 E6 3D       692      inc MOTORTIM+1
1568           693      ;
1568 E9 01       694      ^3      sbc #1
156A D0 F1       695      bne <1
156C           696      ;
156C 60          697      rts
156D           698      ;
156D           699      ;
156D           700      ; Wait 12/24 clock cycles.
156D           701      ;
156D 20 70 15    702      WAIT24    jsr WAIT12
1570           703      ;
1570 60          704      WAIT12    rts
1571           705      ;
1571           706      ;
1571           707      icl "SCAN4.L"
```

LLOAD SCAN4.L,A\$4000

```

1571          1          ttl "Track Scan Source Code, SCAN4.L"
1571          2          ;
1571          3          ;
1571          4          ; SCAN4.L
1571          5          ;
1571          6          ;
1571 20 E3 03    7  INITPGM  jsr  GETIOB
1574          8          ;
1574 84 4A      9          sty  IOBADR
1576 85 4B     10         sta  IOBADR+1
1578          11         ;
1578 A0 01     12         ldy  #SLOT16-TBLTYPE
157A          13         ;
157A B1 4A     14 ^1      lda  (IOBADR),Y
157C 99 B5 19  15         sta  TBLTYPE,Y
157F          16         ;
157F C8        17         iny
1580          18         ;
1580 C0 04     19         cpy  #TRACK-TBLTYPE
1582 D0 F6     20         bne  <1
1584          21         ;
1584 A0 0E     22         ldy  #VOLFND-TBLTYPE
1586          23         ;
1586 B1 4A     24 ^2      lda  (IOBADR),Y
1588 99 B5 19  25         sta  TBLTYPE,Y
158B          26         ;
158B C8        27         iny
158C          28         ;
158C C0 11     29         cpy  #TBLSIZE
158E D0 F6     30         bne  <2
1590          31         ;
1590 AD B6 19  32         lda  SLOT16
1593          33         ;
1593 4A        34         lsr
1594 4A        35         lsr
1595 4A        36         lsr
1596 4A        37         lsr
1597          38         ;
1597 8D 4E 19  39         sta  SLOT
159A          40         ;
159A A9 04     41         lda  #DFLTPHAS
159C 8D 42 19  42         sta  VALSPHAS
159F 8D BF 19  43         sta  IOCBPHAS
15A2          44         ;
15A2 A0 00     45         ldy  #ZERO
15A4 98        46         tya
15A5          47         ;
15A5 99 00 1A  48 ^3      sta  NBUF1,Y
15A8 99 00 1B  49         sta  NBUF2,Y
15AB          50         ;
15AB C8        51         iny
15AC D0 F7     52         bne  <3
15AE          53         ;
15AE 60        54         rts
15AF          55         ;
15AF          56         ;
15AF A2 02     57  GETHEX2  ldx  #2
15B1          58         ;
15B1 2C 00 00  59         bit  *-*
15B4          60         dfs  !-2

```

```

15B2          61 ;
15B2 A2 03    62 GETHEX3 ldx #3
15B4          63 ;
15B4 2C 00 00 64          bit *-*
15B7          65          dfs !-2
15B5          66 ;
15B5 A2 04    67 GETHEX4 ldx #4
15B7          68 ;
15B7 2C 00 00 69          bit *-*
15BA          70          dfs !-2
15B8          71 ;
15B8 A2 06    72 GETHEX6 ldx #6
15BA          73 ;
15BA 2C 00 00 74          bit *-*
15BD          75          dfs !-2
15BB          76 ;
15BB A2 08    77 GETHEX8 ldx #8
15BD          78 ;
15BD 38       79          sec
15BE B0 0C    80          bcs >1
15C0          81 ;
15C0          82 ;
15C0 A2 01    83 GETDEC1 ldx #1
15C2          84 ;
15C2 2C 00 00 85          bit *-*
15C5          86          dfs !-2
15C3          87 ;
15C3 A2 02    88 GETDEC2 ldx #2
15C5          89 ;
15C5 2C 00 00 90          bit *-*
15C8          91          dfs !-2
15C6          92 ;
15C6 A2 03    93 GETDEC3 ldx #3
15C8          94 ;
15C8 2C 00 00 95          bit *-*
15CB          96          dfs !-2
15C9          97 ;
15C9 A2 05    98 GETDEC5 ldx #5
15CB          99 ;
15CB 18       100         clc
15CC          101 ;
15CC 08       102 ^1      php
15CD          103 ;
15CD 8E 63 19 104          stx NNUMS
15D0          105 ;
15D0 A9 00    106          lda #ZERO
15D2          107 ;
15D2 A0 0E    108          ldy #CHARBUFR-NENTRY+1
15D4          109 ;
15D4 99 65 19 110 ^2      sta NENTRY,Y
15D7          111 ;
15D7 88       112          dey
15D8 10 FA    113          bpl <2
15DA          114 ;
15DA 20 AD 16 115 NUMLOOP jsr READKEY
15DD 90 03    116          bcc >3
15DF          117 ;
15DF 28       118          plp
15E0          119 ;
15E0 38       120          sec
15E1          121 ;

```

```

15E1 60          122      rts
15E2            123      ;
15E2 D0 0A      124      ^3      bne >4
15E4            125      ;
15E4 AC 65 19   126      ldy NENTRY
15E7 D0 51      127      bne NUMPROC
15E9            128      ;
15E9 28         129      plp
15EA            130      ;
15EA A9 00      131      lda #ZERO
15EC            132      ;
15EC 18         133      clc
15ED            134      ;
15ED 60         135      rts
15EE            136      ;
15EE C9 88      137      ^4      cmp #LARROW
15F0 F0 06      138      beq >5
15F2            139      ;
15F2 C9 FF      140      cmp #DELETE
15F4 D0 15      141      bne >6
15F6            142      ;
15F6 A9 88      143      lda #LARROW
15F8            144      ;
15F8 AC 65 19   145      ^5      ldy NENTRY
15FB F0 DD      146      beq NUMLOOP
15FD            147      ;
15FD A9 A0      148      lda #SPACE
15FF 20 BF 16   149      jsr PRNTCHAR
1602            150      ;
1602 C6 24      151      dec CH
1604 C6 24      152      dec CH
1606            153      ;
1606 CE 65 19   154      dec NENTRY
1609 10 CF      155      bpl NUMLOOP      ; always taken
160B            156      ;
160B C9 E0      157      ^6      cmp #LWRCASE
160D 90 02      158      bcc >7
160F            159      ;
160F 29 DF      160      and #LWRMASK
1611            161      ;
1611 AC 65 19   162      ^7      ldy NENTRY
1614 CC 63 19   163      cpy NNUMS
1617 B0 C1      164      bcs NUMLOOP
1619            165      ;
1619 AA         166      tax
161A            167      ;
161A E9 AF      168      sbc #"0"-1
161C 30 BC      169      bmi NUMLOOP
161E            170      ;
161E C9 0A      171      cmp #10
1620 90 0C      172      bcc >8
1622            173      ;
1622 28         174      plp
1623 08         175      php
1624            176      ;
1624 90 B4      177      bcc NUMLOOP
1626            178      ;
1626 E9 11      179      sbc #17
1628            180      ;
1628 C9 06      181      cmp #6
162A B0 AE      182      bcs NUMLOOP

```

```

162C      183 ;
162C 69 0A      184      adc #10
162E      185 ;
162E 99 66 19   186 ^8      sta NUMVALS,Y
1631      187 ;
1631 8A      188      txa
1632      189 ;
1632 20 BF 16   190      jsr PRNTCHAR
1635      191 ;
1635 EE 65 19   192      inc NENTRY
1638 D0 A0      193      bne NUMLOOP      ; always taken
163A      194 ;
163A 28      195 NUMPROC  plp
163B 90 2F      196      bcc DECPROC
163D      197 ;
163D CC 63 19   198 HEXPROC  cpy NNUMS
1640 B0 13      199      bcs >2
1642      200 ;
1642 A2 06      201      ldx #6
1644      202 ;
1644 BD 66 19   203 ^1      lda NUMVALS,X
1647 9D 67 19   204      sta NUMVALS+1,X
164A      205 ;
164A CA      206      dex
164B 10 F7      207      bpl <1
164D      208 ;
164D A9 00      209      lda #ZERO
164F 8D 66 19   210      sta NUMVALS
1652      211 ;
1652 C8      212      iny
1653 D0 E8      213      bne HEXPROC
1655      214 ;
1655 A2 06      215 ^2      ldx #6
1657 A0 03      216      ldy #3
1659      217 ;
1659 BD 66 19   218 ^3      lda NUMVALS,X
165C      219 ;
165C 0A      220      asl
165D 0A      221      asl
165E 0A      222      asl
165F 0A      223      asl
1660      224 ;
1660 1D 67 19   225      ora NUMVALS+1,X
1663 99 6E 19   226      sta HEXVALS,Y
1666      227 ;
1666 CA      228      dex
1667 CA      229      dex
1668      230 ;
1668 88      231      dey
1669 10 EE      232      bpl <3
166B      233 ;
166B 60      234      rts
166C      235 ;
166C AE 65 19   236 DECPROC  ldx NENTRY
166F CA      237      dex
1670      238 ;
1670 A0 00      239      ldy #ZERO
1672      240 ;
1672 BD 66 19   241 ^1      lda NUMVALS,X
1675 F0 19      242      beq >3
1677      243 ;

```

```

1677 AD 72 19      244  ^2      lda DECVALS
167A 79 9B 19      245      adc DECTBLL,Y
167D 8D 72 19      246      sta DECVALS
1680              247      ;
1680 AD 73 19      248      lda DECVALS+1
1683 79 A0 19      249      adc DECTBLH,Y
1686 8D 73 19      250      sta DECVALS+1
1689              251      ;
1689 B0 09          252      bcs >4
168B              253      ;
168B DE 66 19      254      dec NUMVALS,X
168E D0 E7          255      bne <2
1690              256      ;
1690 C8              257      ^3      iny
1691              258      ;
1691 CA              259      dex
1692 10 DE          260      bpl <1
1694              261      ;
1694 60              262      ^4      rts
1695              263      ;
1695              264      ;
1695 2C 10 C0        265  GETKEY    bit CLRKEY
1698              266      ;
1698 AD 00 C0        267      ^1      lda KEY
169B              268      ;
169B EA              269      nop
169C              270      ;
169C 10 FA          271      bpl <1
169E              272      ;
169E 2C 10 C0        273      bit CLRKEY
16A1              274      ;
16A1 C9 E1          275      cmp #"a"
16A3 90 07          276      bcc >3
16A5              277      ;
16A5 C9 FB          278      cmp #"z"+1
16A7 B0 02          279      bcs >2
16A9              280      ;
16A9 29 DF          281      and #LWRMASK
16AB              282      ;
16AB 18              283      ^2      clc
16AC              284      ;
16AC 60              285      ^3      rts
16AD              286      ;
16AD              287      ;
16AD A9 60          288  READKEY    lda #FLASH
16AF 20 ED FD        289      jsr COUT
16B2              290      ;
16B2 C6 24          291      dec CH
16B4              292      ;
16B4 20 95 16        293      jsr GETKEY
16B7              294      ;
16B7 C9 9B          295      cmp #ESCAPE
16B9 F0 03          296      beq >1
16BB              297      ;
16BB C9 8D          298      cmp #RETURN
16BD              299      ;
16BD 18              300      clc
16BE              301      ;
16BE 60              302      ^1      rts
16BF              303      ;
16BF              304      ;

```



```

16BF 8D 74 19    305 PRNTCHAR sta CHARBUFR
16C2             306 ;
16C2 20 CB 16    307         jsr PRINT
16C5 56 00        308         byt BUFRCMD,DIRECT
16C7 74 19        309         adr CHARBUFR
16C9 50           310         byt RTNCMD
16CA             311 ;
16CA 60           312         rts
16CB             313 ;
16CB             314 ;
16CB 8D AB 17    315 PRINT      sta PRNTSAVA+1
16CE 8E A9 17    316             stx PRNTSAVX+1
16D1 8C A7 17    317             sty PRNTSAVY+1
16D4             318 ;
16D4 68           319         pla
16D5 85 FC        320         sta PRNTPTR
16D7             321 ;
16D7 68           322         pla
16D8 85 FD        323         sta PRNTPTR+1
16DA             324 ;
16DA E6 FC        325 PRNTLOOP  inc PRNTPTR
16DC D0 02        326             bne >1
16DE             327 ;
16DE E6 FD        328             inc PRNTPTR+1
16E0             329 ;
16E0 A0 00        330 ^1         ldy #ZERO
16E2             331 ;
16E2 B1 FC        332             lda (PRNTPTR),Y
16E4 10 10        333             bpl >3
16E6             334 ;
16E6 C9 A0        335             cmp #SPACE
16E8 90 06        336             bcc >2
16EA             337 ;
16EA 20 45 17    338             jsr PRNTOUT
16ED             339 ;
16ED 4C DA 16    340             jmp PRNTLOOP
16F0             341 ;
16F0 20 49 17    342 ^2         jsr PRNTOUT2
16F3             343 ;
16F3 4C DA 16    344             jmp PRNTLOOP
16F6             345 ;
16F6 C9 50        346 ^3         cmp #MAXCH
16F8 B0 04        347             bcs >4
16FA             348 ;
16FA 85 24        349             sta CH
16FC             350 ;
16FC 90 DC        351             bcc PRNTLOOP
16FE             352 ;
16FE C9 60        353 ^4         cmp #MINCV
1700 90 0A        354             bcc >5
1702             355 ;
1702 29 1F        356             and #CVMASK
1704 85 25        357             sta CV
1706             358 ;
1706 20 22 FC    359 PRNTMOD1 jsr VTAB
1709             360 ;
1709 4C DA 16    361             jmp PRNTLOOP
170C             362 ;
170C 29 0F        363 ^5         and #PCMDMASK
170E AA          364             tax
170F             365 ;

```

```

170F BD 6B 17 366          lda PRNTBL,X
1712 8D 22 17 367          sta PRNTMOD2+1
1715          368          ;
1715 BD 7B 17 369          lda PRNTBLL,X
1718 8D 40 17 370          sta PRNTMOD3+1
171B          371          ;
171B BD 8B 17 372          lda PRNTBLH,X
171E 8D 41 17 373          sta PRNTMOD3+2
1721          374          ;
1721 90 19      375 PRNTMOD2 bcc PRNTBR4
1723          376          ;
1723 C8          377 PRNTBR1  iny
1724          378          ;
1724 B1 FC      379          lda (PRNTPTR),Y
1726 8D 6A 17 380          sta FRMTVAL
1729          381          ;
1729 C8          382 PRNTBR2  iny
172A          383          ;
172A B1 FC      384          lda (PRNTPTR),Y
172C 85 FA      385          sta DATAPTR
172E          386          ;
172E C8          387 PRNTBR3  iny
172F          388          ;
172F B1 FC      389          lda (PRNTPTR),Y
1731 85 FB      390          sta DATAPTR+1
1733          391          ;
1733 98          392          tya
1734          393          ;
1734 65 FC      394          adc PRNTPTR
1736 85 FC      395          sta PRNTPTR
1738 90 02      396          bcc PRNTBR4
173A          397          ;
173A E6 FD      398          inc PRNTPTR+1
173C          399          ;
173C 18          400 PRNTBR4  clc
173D          401          ;
173D A0 00      402          ldy #ZERO
173F          403          ;
173F 20 00 00  404 PRNTMOD3 jsr *-*
1742          405          ;
1742 4C DA 16  406          jmp PRNTLOOP
1745          407          ;
1745          408          ;
1745          409 PRNTOUT:
1745          410          ;
1745 09 00      411 OUTMOD1  ora #ZERO
1747 49 00      412 OUTMOD2  eor #ZERO
1749          413          ;
1749 4C ED FD  414 PRNTOUT2 jmp COUT
174C          415          ;
174C          416          ;
174C          417          ; Notes on DISPCMD as index
174C          418          ;
174C          419          ; 0 - Normal display
174C          420          ; 1 - Inverse display
174C          421          ;
174C 00 40      422 OUTTBL1  hex 0040          ; TEXT
174E 00 00      423          hex 0000          ; GRPH
1750 00 00      424          hex 0000          ; TX80
1752          425          ;
1752 00 C0      426 OUTTBL2  hex 00C0

```

```

1754 00 80      427      hex 0080
1756 00 00      428      hex 0000
1758           429      ;
1758 FF 3F      430 OUT80COL hex FF3F
175A           431      ;
175A           432      ;
175A 22 FC      433 VTABADRS adr VTAB
175C AD 17      434      adr PRINTRTN
175E 22 FC      435      adr VTAB
1760           436      ;
1760           437      ;
1760 ED FD      438 OUTADRS  adr COUT
1762 42 19      439      adr PRNTGRPH
1764 ED FD      440      adr COUT
1766           441      ;
1766 00         442 PRNTSAV  hex 00
1767 00 00      443 PRNTNUM  hex 0000
1769           444      ;
1769           445      ;
1769           446 ; Notes on MODEVAL and FRMTVAL
1769           447      ;
1769           448 ; 0 - 40 column TEXT mode
1769           449 ; 1 - GRAPHICS mode
1769           450 ; 2 - 80 column TEXT mode
1769           451 ; 3 - exit 80 TEXT, enter 40 TEXT
1769           452      ;
1769           453 ; 0x00 - no left padding
1769           454 ; 0x20 data in high/low order, otherwise low/high order
1769           455 ; 0x40 - zero left padding
1769           456 ; 0x80 - space left padding
1769           457      ;
1769 00         458 MODEVAL  hex 00
176A 00         459 FRMTVAL  hex 00
176B           460      ;
176B           461      ;
176B           462 ; Branch table of command routines.
176B           463      ;
176B           464 PRNTBL:
176B 19         465      byt PRNTBR4-PRNTBR1 ; 50
176C 0B         466      byt PRNTBR3-PRNTBR1 ; 51
176D 0B         467      byt PRNTBR3-PRNTBR1 ; 52
176E 0B         468      byt PRNTBR3-PRNTBR1 ; 53
176F 0B         469      byt PRNTBR3-PRNTBR1 ; 54
1770 19         470      byt PRNTBR4-PRNTBR1 ; 55
1771 00         471      byt PRNTBR1-PRNTBR1 ; 56
1772 06         472      byt PRNTBR2-PRNTBR1 ; 57
1773 06         473      byt PRNTBR2-PRNTBR1 ; 58
1774 06         474      byt PRNTBR2-PRNTBR1 ; 59
1775 00         475      byt PRNTBR1-PRNTBR1 ; 5A
1776 00         476      byt PRNTBR1-PRNTBR1 ; 5B
1777 06         477      byt PRNTBR2-PRNTBR1 ; 5C
1778 06         478      byt PRNTBR2-PRNTBR1 ; 5D
1779 06         479      byt PRNTBR2-PRNTBR1 ; 5E
177A 00         480      byt PRNTBR1-PRNTBR1 ; 5F
177B           481      ;
177B           482      ;
177B           483 ; Address tables of command routines.
177B           484      ;
177B           485 PRNTBLL:
177B 9B         486      byt PRNTRTN      ; 50
177C AE         487      byt PRNTMODE   ; 51

```

```

177D F9          488          byt PRNTDISP          ; 52
177E 18          489          byt PRNTSCRN          ; 53
177F 28          490          byt PRNTCLR           ; 54
1780 32          491          byt PRNTCNTR          ; 55
1781 44          492          byt PRNTBUFR          ; 56
1782 5F          493          byt PRNTNIBL          ; 57
1783 68          494          byt PRNT1BYT          ; 58
1784 6B          495          byt PRNT2BYT          ; 59
1785 64          496          byt PRNTNBYT          ; 5A
1786 77          497          byt PRNTADR           ; 5B
1787 8F          498          byt PRNT1DEC          ; 5C
1788 95          499          byt PRNT2DEC          ; 5D
1789 A2          500          byt PRNT3DEC          ; 5E
178A AC          501          byt PRNTNDEC          ; 5F
178B            502          ;
178B            503          PRNTBLH:
178B 17          504          hby PRNTRTN           ; 50
178C 17          505          hby PRNTMODE          ; 51
178D 17          506          hby PRNTDISP          ; 52
178E 18          507          hby PRNTSCRN          ; 53
178F 18          508          hby PRNTCLR           ; 54
1790 18          509          hby PRNTCNTR          ; 55
1791 18          510          hby PRNTBUFR          ; 56
1792 18          511          hby PRNTNIBL          ; 57
1793 18          512          hby PRNT1BYT          ; 58
1794 18          513          hby PRNT2BYT          ; 59
1795 18          514          hby PRNTNBYT          ; 5A
1796 18          515          hby PRNTADR           ; 5B
1797 18          516          hby PRNT1DEC          ; 5C
1798 18          517          hby PRNT2DEC          ; 5D
1799 18          518          hby PRNT3DEC          ; 5E
179A 18          519          hby PRNTNDEC          ; 5F
179B            520          ;
179B            521          ;
179B            522          ; RTNCMD (0x50)
179B            523          ;
179B BA          524          PRNTRTN   tsx
179C            525          ;
179C A5 FC          526          lda PRNTPTR
179E 9D 01 01      527          sta STACK+1,X
17A1            528          ;
17A1 A5 FD          529          lda PRNTPTR+1
17A3 9D 02 01      530          sta STACK+2,X
17A6            531          ;
17A6 A0 00          532          PRNTSAVY ldy #ZERO
17A8 A2 00          533          PRNTSAVX ldx #ZERO
17AA A9 00          534          PRNTSAVA lda #ZERO
17AC            535          ;
17AC 18            536          clc
17AD            537          ;
17AD 60            538          PRINTRTN rts
17AE            539          ;
17AE            540          ;
17AE            541          ; MODECMD (0x51)
17AE            542          ;
17AE            543          ; 0 - 40 column TEXT mode
17AE            544          ; 1 - GRAPHICS mode
17AE            545          ; 2 - 80 column TEXT mode
17AE            546          ; 3 - exit 80 TEXT, enter 40 TEXT
17AE            547          ;
17AE A5 FB          548          PRNTMODE lda DATAPTR+1

```

```

17B0 29 03      549      and #3
17B2           550      ;
17B2 C9 02      551      cmp #TX80MODE
17B4 D0 09      552      bne >1
17B6           553      ;
17B6 A9 B3      554      lda #"3"
17B8 20 95 FE    555      jsr OUTPORT
17BB           556      ;
17BB A9 02      557      lda #TX80MODE
17BD D0 1A      558      bne >2
17BF           559      ;
17BF C9 03      560      ^1 cmp #LV80MODE
17C1 D0 16      561      bne >2
17C3           562      ;
17C3 AD 58 17    563      lda OUT80COL
17C6 85 32      564      sta INVFLG
17C8           565      ;
17C8 A9 9B      566      lda #ESCAPE
17CA 20 ED FD    567      jsr COUT
17CD           568      ;
17CD A9 91      569      lda #CTRLQ
17CF 20 ED FD    570      jsr COUT
17D2           571      ;
17D2 A9 B0      572      lda #"0"
17D4 20 95 FE    573      jsr OUTPORT
17D7           574      ;
17D7 A9 00      575      lda #TEXTMODE
17D9           576      ;
17D9 8D 69 17    577      ^2 sta MODEVAL
17DC           578      ;
17DC 0A          579      asl
17DD A8          580      tay
17DE           581      ;
17DE B9 5A 17    582      lda VTABADRS,Y
17E1 8D 07 17    583      sta PRNTMOD1+1
17E4           584      ;
17E4 B9 5B 17    585      lda VTABADRS+1,Y
17E7 8D 08 17    586      sta PRNTMOD1+2
17EA           587      ;
17EA B9 60 17    588      lda OUTADRS,Y
17ED 8D 4A 17    589      sta PRNTOUT2+1
17F0           590      ;
17F0 B9 61 17    591      lda OUTADRS+1,Y
17F3 8D 4B 17    592      sta PRNTOUT2+2
17F6           593      ;
17F6 4C EA 03    594      jmp HOOKDOS
17F9           595      ;
17F9           596      ;
17F9           597      ; DISPCMD (0x52)
17F9           598      ;
17F9           599      ; 0 - Normal display
17F9           600      ; 1 - Inverse display
17F9           601      ;
17F9 A4 FB      602      PRNTDISP ldy DATAPTR+1
17FB           603      ;
17FB AD 69 17    604      lda MODEVAL
17FE C9 02      605      cmp #TX80MODE
1800 D0 05      606      bne >1
1802           607      ;
1802 BE 58 17    608      ldx OUT80COL,Y
1805 86 32      609      stx INVFLG

```

```
1807          610 ;
1807 0A        611 ^1      asl
1808 65 FB     612          adc DATAPTR+1
180A          613 ;
180A A8       614          tay
180B          615 ;
180B B9 4C 17 616          lda OUTTBL1,Y
180E 8D 46 17 617          sta OUTMOD1+1
1811          618 ;
1811 B9 52 17 619          lda OUTTBL2,Y
1814 8D 48 17 620          sta OUTMOD2+1
1817          621 ;
1817 60        622          rts
1818          623 ;
1818          624 ;
1818          625 ; SCRNCMD (0x53)
1818          626 ;
1818          627 ; 0 - INIT
1818          628 ; 1 - HOME
1818          629 ;
1818 2C 54 C0   630 PRNTSCRN bit LOWSCR
181B          631 ;
181B          632          .if DISPLAY=GRPHMODE
181B          633 ;
181B          634          lda MODEVAL
181B          635          cmp #GRPHMODE
181B          636          bne >2
181B          637 ;
181B          638          lda DATAPTR+1
181B          639          bne >1
181B          640 ;
181B          641          jmp SCRNNINIT
181B          642 ;
181B          643 ^1      jmp SCRNHOME
181B          644 ;
181B          645          .fi
181B          646 ;
181B A5 FB     647 ^2      lda DATAPTR+1
181D D0 06     648          bne >3
181F          649 ;
181F 2C 51 C0   650          bit TXTSET
1822          651 ;
1822 4C 2F FB   652          jmp INIT
1825          653 ;
1825 4C 58 FC   654 ^3      jmp HOME
1828          655 ;
1828          656 ;
1828          657 ; CLRCMD (0x54)
1828          658 ;
1828          659 ; 0 - EOL
1828          660 ; 1 - EOP
1828          661 ;
1828          662 PRNTCLR:
1828          663          .if DISPLAY=GRPHMODE
1828          664 ;
1828          665          lda MODEVAL
1828          666          cmp #GRPHMODE
1828          667          bne >2
1828          668 ;
1828          669          lda DATAPTR+1
1828          670          bne >1
```

```

1828      671 ;
1828      672      jmp SCRNEOL
1828      673 ;
1828      674 ^1      jmp SCRNEOP
1828      675 ;
1828      676      .fi
1828      677 ;
1828 A5 FB      678 ^2      lda DATAPTR+1
182A D0 03      679      bne >3
182C      680 ;
182C 4C 9C FC      681      jmp CLREOL
182F      682 ;
182F 4C 42 FC      683 ^3      jmp CLREOP
1832      684 ;
1832      685 ;
1832      686 ; CNTRCMD (0x55)
1832      687 ;
1832 A9 9F      688 PRNTCNTR lda #SPACE-1
1834      689 ;
1834 C8      690 ^1      iny
1835      691 ;
1835 D1 FC      692      cmp (PRNTPTR),Y
1837 90 FB      693      bcc <1
1839      694 ;
1839 98      695      tya
183A      696 ;
183A 49 FF      697      eor #NEGONE
183C 65 21      698      adc WNDWDTH
183E      699 ;
183E 4A      700      lsr
183F      701 ;
183F 65 20      702      adc WNDLFT
1841 85 24      703      sta CH
1843      704 ;
1843 60      705      rts
1844      706 ;
1844      707 ;
1844      708 ; BUFRCMD (0x56)
1844      709 ;
1844      710 ; 0 - direct address
1844      711 ; 1 - indirect address
1844      712 ;
1844 AD 6A 17      713 PRNTBUFR lda FRMTVAL
1847 F0 0B      714      beq >1
1849      715 ;
1849 B1 FA      716      lda (DATAPTR),Y
184B AA      717      tax
184C      718 ;
184C C8      719      iny
184D      720 ;
184D B1 FA      721      lda (DATAPTR),Y
184F      722 ;
184F 86 FA      723      stx DATAPTR
1851 85 FB      724      sta DATAPTR+1
1853      725 ;
1853 88      726      dey
1854      727 ;
1854 B1 FA      728 ^1      lda (DATAPTR),Y
1856 F0 06      729      beq >2
1858      730 ;
1858 20 45 17      731      jsr PRNTOUT

```

```

185B          732 ;
185B C8       733      iny
185C D0 F6    734      bne <1
185E          735 ;
185E 60       736 ^2      rts
185F          737 ;
185F          738 ;
185F          739 ; NIBLCMD (0x57)
185F          740 ;
185F B1 FA    741 PRNTNIBL lda (DATAPTR),Y
1861          742 ;
1861 4C FC 18 743      jmp PRNTHX
1864          744 ;
1864          745 ;
1864          746 ; BYT1CMD (0x58)
1864          747 ; BYT2CMD (0x59)
1864          748 ; BYTNCMD (0x5A)
1864          749 ;
1864 AE 6A 17 750 PRNTNBYT ldx FRMTVAL
1867          751 ;
1867 2C 00 00 752      bit *-*
186A          753      dfs !-2
1868          754 ;
1868 A2 01     755 PRNT1BYT ldx #1
186A          756 ;
186A 2C 00 00 757      bit *-*
186D          758      dfs !-2
186B          759 ;
186B A2 02     760 PRNT2BYT ldx #2
186D          761 ;
186D B1 FA    762 PRNTBYT  lda (DATAPTR),Y
186F          763 ;
186F 20 F3 18 764      jsr PRNTBYTE
1872          765 ;
1872 C8       766      iny
1873          767 ;
1873 CA       768      dex
1874 D0 F7    769      bne PRNTBYT
1876          770 ;
1876 60       771      rts
1877          772 ;
1877          773 ;
1877          774 ; ADRCMD (0x5B)
1877          775 ;
1877          776 ; 0 - direct address
1877          777 ; 1 - indirect address
1877          778 ;
1877 AD 6A 17 779 PRNTADR  lda FRMTVAL
187A D0 06    780      bne >1
187C          781 ;
187C A6 FA    782      ldx DATAPTR
187E A5 FB    783      lda DATAPTR+1
1880          784 ;
1880 90 06    785      bcc >2          ; always taken
1882          786 ;
1882 B1 FA    787 ^1      lda (DATAPTR),Y
1884 AA       788      tax
1885          789 ;
1885 C8       790      iny
1886          791 ;
1886 B1 FA    792      lda (DATAPTR),Y

```



```

1888      793 ;
1888 20 F3 18 794 ^2      jsr PRNTBYTE
188B      795 ;
188B 8A      796      txa
188C      797 ;
188C 4C F3 18 798      jmp PRNTBYTE
188F      799 ;
188F      800 ;
188F      801 ; DEC1CMD (0x5C)
188F      802 ;
188F 20 09 19 803 PRNT1DEC jsr HEXTODEC
1892      804 ;
1892 4C FC 18 805      jmp PRNTHex
1895      806 ;
1895      807 ;
1895      808 ; DEC2CMD (0x5D)
1895      809 ;
1895 20 09 19 810 PRNT2DEC jsr HEXTODEC
1898      811 ;
1898      812 ;
1898 8A      813 PRNTDEC  txa
1899 20 FC 18 814      jsr PRNTHex
189C      815 ;
189C AD 66 17 816      lda PRNTSAV
189F      817 ;
189F 4C FC 18 818      jmp PRNTHex
18A2      819 ;
18A2      820 ;
18A2      821 ; DEC3CMD (0x5E)
18A2      822 ;
18A2 20 09 19 823 PRNT3DEC jsr HEXTODEC
18A5      824 ;
18A5 98      825      tya
18A6      826 ;
18A6 20 FC 18 827      jsr PRNTHex
18A9      828 ;
18A9 4C 98 18 829      jmp PRNTDEC
18AC      830 ;
18AC      831 ;
18AC      832 ; DECNCMD (0x5F)
18AC      833 ;
18AC      834 ; 0x00 - no left padding
18AC      835 ; 0x20 - data in high/low order, otherwise low/high order
18AC      836 ; 0x40 - zero left padding
18AC      837 ; 0x80 - space left padding
18AC      838 ;
18AC      839 ; data in high/low order
18AC      840 ;
18AC B1 FA    841 PRNTNDEC lda (DATAPTR),Y
18AE AA      842      tax
18AF      843 ;
18AF C8      844      iny
18B0      845 ;
18B0 B1 FA    846      lda (DATAPTR),Y
18B2 A8      847      tay
18B3      848 ;
18B3 AD 6A 17 849      lda FRMTVAL
18B6 29 20    850      and #HIGHLOW
18B8 D0 09    851      bne >1
18BA      852 ;
18BA 8E 67 17 853      stx PRNTNUM

```

```

18BD 8C 68 17      854      sty PRNTNUM+1
18C0              855      ;
18C0 4C C9 18      856      jmp >2
18C3              857      ;
18C3 8C 67 17      858      ^1      sty PRNTNUM
18C6 8E 68 17      859      stx PRNTNUM+1
18C9              860      ;
18C9 A2 03          861      ^2      ldx #3
18CB              862      ;
18CB 2C 6A 17      863      bit FRMTVAL
18CE 70 18          864      bvs >6
18D0              865      ;
18D0 20 23 19      866      ^3      jsr GETDIGIT
18D3 D0 16          867      bne >7
18D5              868      ;
18D5 2C 6A 17      869      bit FRMTVAL
18D8 10 05          870      bpl >4
18DA              871      ;
18DA A9 A0          872      lda #SPACE
18DC 20 45 17      873      jsr PRNTOUT
18DF              874      ;
18DF CA            875      ^4      dex
18E0 10 EE          876      bpl <3
18E2              877      ;
18E2 AD 67 17      878      ^5      lda PRNTNUM
18E5              879      ;
18E5 4C FC 18      880      jmp PRNTHEx
18E8              881      ;
18E8 20 23 19      882      ^6      jsr GETDIGIT
18EB              883      ;
18EB 20 FC 18      884      ^7      jsr PRNTHEx
18EE              885      ;
18EE CA            886      dex
18EF 10 F7          887      bpl <6
18F1              888      ;
18F1 30 EF          889      bmi <5                ; always taken
18F3              890      ;
18F3              891      ;
18F3 48            892      PRNTBYTE pha
18F4              893      ;
18F4 4A            894      lsr
18F5 4A            895      lsr
18F6 4A            896      lsr
18F7 4A            897      lsr
18F8              898      ;
18F8 20 FE 18      899      jsr PRNTHEx2
18FB              900      ;
18FB 68            901      pla
18FC              902      ;
18FC              903      ;
18FC 29 0F          904      PRNTHEx and #NIBLMASK
18FE              905      ;
18FE 09 B0          906      PRNTHEx2 ora #"0"
1900              907      ;
1900 C9 BA          908      cmp #"9"+1
1902 90 02          909      bcc >1
1904              910      ;
1904 69 06          911      adc #6
1906              912      ;
1906 4C 45 17      913      ^1      jmp PRNTOUT
1909              914      ;

```

```
1909          915 ;
1909 A2 00      916 HEXTODEC ldx #ZERO
190B          917 ;
190B B1 FA     918          lda (DATAPTR),Y
190D          919 ;
190D C9 64     920 ^1      cmp #100
190F 90 05     921          bcc >2
1911          922 ;
1911 E9 64     923          sbc #100
1913          924 ;
1913 C8         925          iny
1914 D0 F7     926          bne <1
1916          927 ;
1916 C9 0A     928 ^2      cmp #10
1918 90 05     929          bcc >3
191A          930 ;
191A E9 0A     931          sbc #10
191C          932 ;
191C E8         933          inx
191D D0 F7     934          bne <2
191F          935 ;
191F 8D 66 17  936 ^3      sta PRNTSAV
1922          937 ;
1922 60         938          rts
1923          939 ;
1923          940 ;
1923 A0 00      941 GETDIGIT ldy #ZERO
1925          942 ;
1925 38         943 ^1      sec
1926          944 ;
1926 AD 67 17  945          lda PRNTNUM
1929 FD 9C 19  946          sbc DECTBLL+1,X
192C 48        947          pha
192D          948 ;
192D AD 68 17  949          lda PRNTNUM+1
1930 FD A1 19  950          sbc DECTBLH+1,X
1933 90 0A     951          bcc >2
1935          952 ;
1935 8D 68 17  953          sta PRNTNUM+1
1938          954 ;
1938 68         955          pla
1939 8D 67 17  956          sta PRNTNUM
193C          957 ;
193C C8         958          iny
193D D0 E6     959          bne <1
193F          960 ;
193F 68         961 ^2      pla
1940          962 ;
1940 98         963          tya
1941          964 ;
1941 60         965          rts
1942          966 ;
1942          967 ;
1942          968 PRNTGRPH:
1942          969          .if DISPLAY=GRPHMODE
1942          970 ;
1942          971          cmp #SPACE
1942          972          bcs >3
1942          973 ;
1942          974          cmp #ASCIFLAG
1942          975          bcc >3
```

```
1942      976 ;
1942      977      cmp #BELLCHAR
1942      978      bne >1
1942      979 ;
1942      980      jmp BELL
1942      981 ;
1942      982 ^1      cmp #RETURN
1942      983      beq >2
1942      984 ;
1942      985      cmp #LARROW
1942      986      bne >1
1942      987 ;
1942      988      dec CH
1942      989 ;
1942      990 ^1      cmp #DARROW
1942      991      bne >1
1942      992 ;
1942      993      inc CV
1942      994 ;
1942      995 ^1      cmp #UARROW
1942      996      bne >1
1942      997 ;
1942      998      dec CV
1942      999 ;
1942     1000 ^1      cmp #RARROW
1942     1001      bne >1
1942     1002 ;
1942     1003      inc CH
1942     1004 ;
1942     1005 ^1      rts
1942     1006 ;
1942     1007 ^2      lda WNDLFT
1942     1008      sta CH
1942     1009 ;
1942     1010      inc CV
1942     1011 ;
1942     1012      lda CV
1942     1013      cmp WNDBTM
1942     1014      bcc >8
1942     1015 ;
1942     1016      jmp SCROLL
1942     1017 ;
1942     1018 ^3      stx SCRNSAVX+1
1942     1019      sty SCRNSAVY+1
1942     1020 ;
1942     1021      ldx /CHARTBL
1942     1022 ;
1942     1023      ldy #ZERO
1942     1024 ;
1942     1025      asl
1942     1026      bcs >4
1942     1027 ;
1942     1028      ldy #INVRMASK
1942     1029 ;
1942     1030      sec
1942     1031 ;
1942     1032 ^4      sty SCRNMOD2+1
1942     1033 ;
1942     1034      sbc #$40
1942     1035 ;
1942     1036      asl
```

```
1942      1037      bcc >5
1942      1038      ;
1942      1039      ldx /CHARTBL+$200
1942      1040      ;
1942      1041      ^5      asl
1942      1042      bcc >6
1942      1043      ;
1942      1044      inx
1942      1045      ;
1942      1046      clc
1942      1047      ;
1942      1048      ^6      adc #CHARTBL
1942      1049      sta SCRNMOD1+1
1942      1050      bcc >7
1942      1051      ;
1942      1052      inx
1942      1053      ;
1942      1054      ^7      stx SCRNMOD1+2
1942      1055      ;
1942      1056      clc
1942      1057      ;
1942      1058      ldx CV
1942      1059      ;
1942      1060      lda YBASELO,X
1942      1061      sta SCRNMOD3+1
1942      1062      ;
1942      1063      lda YBASEHI,X
1942      1064      sta SCRNMOD3+2
1942      1065      ;
1942      1066      ldy CH
1942      1067      ldx #CHARCELL
1942      1068      ;
1942      1069      SCRNMOD1 lda *-,X
1942      1070      ;
1942      1071      SCRNMOD2 eor #ZERO
1942      1072      ;
1942      1073      SCRNMOD3 sta *-,Y
1942      1074      ;
1942      1075      lda SCRNMOD3+2
1942      1076      adc #NEXTLINE
1942      1077      sta SCRNMOD3+2
1942      1078      ;
1942      1079      dex
1942      1080      bpl SCRNMOD1
1942      1081      ;
1942      1082      iny
1942      1083      ;
1942      1084      SCRNMOD4 cpy #MAXWDTH
1942      1085      bcs <2
1942      1086      ;
1942      1087      sty CH
1942      1088      ;
1942      1089      SCRNSAVY ldy #ZERO
1942      1090      SCRNSAVX ldx #ZERO
1942      1091      ;
1942      1092      ^8      rts
1942      1093      ;
1942      1094      ;
1942      1095      SCRNINIT bit HIRES
1942      1096      bit MIXCLR
1942      1097      bit TXTCLR
```

```
1942      1098 ;
1942      1099      clc
1942      1100 ;
1942      1101      lda #MAXWDTH
1942      1102      adc WNDLFT
1942      1103      sta SCRNMOD4+1
1942      1104      sta EOLMOD1+1
1942      1105      sta SCRLMOD3+1
1942      1106 ;
1942      1107      rts
1942      1108 ;
1942      1109 ;
1942      1110 SCROLL      lda WNDLFT
1942      1111      sta CH
1942      1112 ;
1942      1113      ldx WNDBTM
1942      1114      dex
1942      1115      stx SCRLMOD4+1
1942      1116 ;
1942      1117      ldx WNDTOP
1942      1118      stx CV
1942      1119 ;
1942      1120 ^1      lda YBASELO,X
1942      1121      sta SCRLMOD2+1
1942      1122 ;
1942      1123      lda YBASEHI,X
1942      1124      sta SCRLMOD2+2
1942      1125 ;
1942      1126      inx
1942      1127      stx CV
1942      1128 ;
1942      1129      lda YBASELO,X
1942      1130      sta SCRLMOD1+1
1942      1131 ;
1942      1132      lda YBASEHI,X
1942      1133      sta SCRLMOD1+2
1942      1134 ;
1942      1135      ldx #CHARCELL
1942      1136 ;
1942      1137 ^2      ldY CH
1942      1138 ;
1942      1139 SCRLMOD1      lda *-,Y
1942      1140 SCRLMOD2      sta *-,Y
1942      1141 ;
1942      1142      iny
1942      1143 ;
1942      1144 SCRLMOD3      cpy #MAXWDTH
1942      1145      bcc SCRLMOD1
1942      1146 ;
1942      1147      lda SCRLMOD1+2
1942      1148      adc #NEXTLINE-1
1942      1149      sta SCRLMOD1+2
1942      1150 ;
1942      1151      lda SCRLMOD2+2
1942      1152      adc #NEXTLINE
1942      1153      sta SCRLMOD2+2
1942      1154 ;
1942      1155      dex
1942      1156      bpl <2
1942      1157 ;
1942      1158      ldx CV
```

```
1942      1159  SCRLMOD4  cpx #LOC0
1942      1160                bne <1
1942      1161      ;
1942      1162      ;
1942      1163  SCRNEOL   ldx CV
1942      1164      ;
1942      1165                lda YBASELO,X
1942      1166                sta EOLMOD2+1
1942      1167      ;
1942      1168                lda YBASEHI,X
1942      1169                sta EOLMOD2+2
1942      1170      ;
1942      1171                ldx #CHARCELL
1942      1172      ;
1942      1173      ^1      ldy CH
1942      1174      ;
1942      1175                lda #ZERO
1942      1176      ;
1942      1177  EOLMOD1   cpy #MAXWDTH
1942      1178                bcs >2
1942      1179      ;
1942      1180  EOLMOD2   sta *-*,Y
1942      1181      ;
1942      1182                iny
1942      1183                bne EOLMOD1
1942      1184      ;
1942      1185      ^2      lda EOLMOD2+2
1942      1186                adc #NEXTLINE-1
1942      1187                sta EOLMOD2+2
1942      1188      ;
1942      1189                dex
1942      1190                bpl <1
1942      1191      ;
1942      1192                rts
1942      1193      ;
1942      1194      ;
1942      1195  SCRNHOM   lda WNDLFT
1942      1196                sta CH
1942      1197      ;
1942      1198                lda WNDTOP
1942      1199                sta CV
1942      1200      ;
1942      1201      ;
1942      1202  SCRNEOP   jsr SCRNEOL
1942      1203      ;
1942      1204                lda CH
1942      1205                pha
1942      1206      ;
1942      1207                lda CV
1942      1208                pha
1942      1209      ;
1942      1210                lda WNDLFT
1942      1211                sta CH
1942      1212      ;
1942      1213      ^1      inc CV
1942      1214      ;
1942      1215                lda CV
1942      1216                cmp WNDBTM
1942      1217                bcs >2
1942      1218      ;
1942      1219                jsr SCRNEOL
```

```
1942      1220      bmi <1
1942      1221      ;
1942      1222      ^2      pla
1942      1223      sta CV
1942      1224      ;
1942      1225      pla
1942      1226      sta CH
1942      1227      ;
1942      1228      rts
1942      1229      ;
1942      1230      ;
1942      1231      YBASELO:
1942      1232      hex 0080008000800080
1942      1233      hex 28A828A828A828A8
1942      1234      hex 50D050D050D050D0
1942      1235      ;
1942      1236      YBASEHI:
1942      1237      hex 2020212122222323
1942      1238      hex 2020212122222323
1942      1239      hex 2020212122222323
1942      1240      ;
1942      1241      ;
1942      1242      CHARTBL:
1942      1243      hex 0000000000000000 ;
1942      1244      hex 0008000808080808 ; !
1942      1245      hex 0000000000141414 ; "
1942      1246      hex 0014143E143E1414 ; #
1942      1247      hex 00081E281C0A3C08 ; $
1942      1248      hex 0030320408102606 ; %
1942      1249      hex 002C122A040A0A04 ; &
1942      1250      hex 0000000000080808 ; ^
1942      1251      hex 0010080404040810 ; (
1942      1252      hex 0004081010100804 ; )
1942      1253      hex 00082A1C081C2A08 ; *
1942      1254      hex 000008083E080800 ; +
1942      1255      hex 0408080000000000 ; ,
1942      1256      hex 000000003E000000 ; -
1942      1257      hex 0008000000000000 ; .
1942      1258      hex 0000020408102000 ; /
1942      1259      ;
1942      1260      hex 001C22262A32221C ; 0
1942      1261      hex 001C080808080C08 ; 1
1942      1262      hex 003E02041820221C ; 2
1942      1263      hex 001C22201810203E ; 3
1942      1264      hex 0010103E12141810 ; 4
1942      1265      hex 001C2220201E023E ; 5
1942      1266      hex 001C22221E020438 ; 6
1942      1267      hex 000404040810203E ; 7
1942      1268      hex 001C22221C22221C ; 8
1942      1269      hex 000E10203C22221C ; 9
1942      1270      hex 0000080008000000 ; :
1942      1271      hex 0408080008000000 ; ;
1942      1272      hex 0020100804081020 ; <
1942      1273      hex 0000003E003E0000 ; =
1942      1274      hex 0002040810080402 ; >
1942      1275      hex 000800089820221C ; ?
1942      1276      ;
1942      1277      hex 003C021A2A3A221C ; @
1942      1278      hex 0022223E22221408 ; A
1942      1279      hex 001E22221E22221E ; B
1942      1280      hex 001C22020202221C ; C
```



```
1942      1281      hex 001E222222222221E ; D
1942      1282      hex 003E02021E02023E ; E
1942      1283      hex 000202021E02023E ; F
1942      1284      hex 003C22320202023C ; G
1942      1285      hex 002222223E222222 ; H
1942      1286      hex 001C08080808081C ; I
1942      1287      hex 001C222020202020 ; J
1942      1288      hex 0022120A060A1222 ; K
1942      1289      hex 003E020202020202 ; L
1942      1290      hex 002222222A2A3622 ; M
1942      1291      hex 002222322A262222 ; N
1942      1292      hex 001C22222222221C ; O
1942      1293      ;
1942      1294      hex 000202021E22221E ; P
1942      1295      hex 002C122A2222221C ; Q
1942      1296      hex 0022120A1E22221E ; R
1942      1297      hex 001C22201C02221C ; S
1942      1298      hex 000808080808083E ; T
1942      1299      hex 001C222222222222 ; U
1942      1300      hex 0008142222222222 ; V
1942      1301      hex 0022362A2A222222 ; W
1942      1302      hex 0022221408142222 ; X
1942      1303      hex 0008080808142222 ; Y
1942      1304      hex 003E02040810203E ; Z
1942      1305      hex 003C04040404043C ; [
1942      1306      hex 0000201008040200 ; \
1942      1307      hex 001E10101010101E ; ]
1942      1308      hex 0000000000221408 ; ^
1942      1309      hex 7F00000000000000 ; _
1942      1310      ;
1942      1311      hex 0000000000100804 ; `
1942      1312      hex 003C223C201C0000 ; a
1942      1313      hex 001E2222221E0202 ; b
1942      1314      hex 003C0202023C0000 ; c
1942      1315      hex 003C2222223C2020 ; d
1942      1316      hex 003C023E221C0000 ; e
1942      1317      hex 000404041E042418 ; f
1942      1318      hex 1C203C22221C0000 ; g
1942      1319      hex 00222222221E0202 ; h
1942      1320      hex 001C0808080C0008 ; i
1942      1321      hex 0C12101010180010 ; j
1942      1322      hex 0022120E12220202 ; k
1942      1323      hex 001C08080808080C ; l
1942      1324      hex 00222A2A2A360000 ; m
1942      1325      hex 00222222221E0000 ; n
1942      1326      hex 001C2222221C0000 ; o
1942      1327      ;
1942      1328      hex 02021E22221E0000 ; p
1942      1329      hex 20203C22223C0000 ; q
1942      1330      hex 00020202063A0000 ; r
1942      1331      hex 001E201C023C0000 ; s
1942      1332      hex 00182404041E0404 ; t
1942      1333      hex 002C322222220000 ; u
1942      1334      hex 0008142222220000 ; v
1942      1335      hex 00362A2A22220000 ; w
1942      1336      hex 0022140814220000 ; x
1942      1337      hex 1C203C2222220000 ; y
1942      1338      hex 003E0408103E0000 ; z
1942      1339      hex 0030080804080830 ; {
1942      1340      hex 0808080808080808 ; |
1942      1341      hex 0006080810080806 ; }
```

1942	1342	hex 00000000000001A2C ; ~
1942	1343	hex 00002A142A142A00 ;
1942	1344 ;	
1942	1345 ;	
1942	1346	.fi
1942	1347 ;	
1942	1348 ;	
1942	1349	icl "SCAN5.L"

LLOAD SCAN5.L,A\$4000

```
1942          1          ttl "Track Scan Source Code, SCAN5.L"
1942          2          ;
1942          3          ;
1942          4          ; SCAN5.L
1942          5          ;
1942          6          ;
1942          7          VALSPHAS dfs 1,DFLTPHAS
1943          8          ENDTRK  dfs 1,DFLTRACK
1944          9          ENDSEC   dfs 1,DFLTSEC
1945         10          SYNCOUNT dfs 1,NEGONE
1946         11          ;
1946         12          ERRFLAG  dfs 1,ZERO
1947         13          STAGE     dfs 1,ZERO
1948         14          ;
1948         15          SECCNT    dfs 1,ZERO
1949         16          TRACKSIZ  dfs 1,ZERO
194A         17          STRTRACK  dfs 1,ZERO
194B         18          ENDTRACK  dfs 1,ZERO
194C         19          SYNCNUM  dfs 1,ZERO
194D         20          HDRCNT    dfs 1,ZERO
194E         21          SLOT      dfs 1,ZERO
194F         22          TEMP      dfs 1,ZERO
1950         23          CHKSUM    dfs 1,ZERO
1951         24          ;
1951         25          HDRVOL     dfs 1,ZERO
1952         26          HDRTRK     dfs 1,ZERO
1953         27          HDRSEC     dfs 1,ZERO
1954         28          HDRCHK     dfs 1,ZERO
1955         29          HDRGAP     dfs 1,ZERO
1956         30          ;
1956         31          NUMBUF     dfs 2,ZERO
1958         32          DATAcnt   dfs 2,ZERO
195A         33          SYNCTOTL  dfs 2,ZERO
195C         34          ;
195C         35          SAVX       dfs 1,ZERO
195D         36          SAVY       dfs 1,ZERO
195E         37          SAVA       dfs 1,ZERO
195F         38          ;
195F         39          OPTION    dfs 1,ZERO
1960         40          ;
1960         41          SAVNUM     dfs 1,ZERO
1961         42          ;
1961         43          COUNT      dfs 1,ZERO
1962         44          COUNT2     dfs 1,ZERO
1963         45          ;
1963         46          NNUMS      dfs 1,ZERO
1964         47          SAVECH     dfs 1,ZERO
1965         48          NENTRY     dfs 1,ZERO
1966         49          ;
1966         50          NUMVALS    dfs 8,ZERO
196E         51          ;
196E         52          HEXVALS   dfs 4,ZERO
1972         53          DECVALS   dfs 2,ZERO
1974         54          ;
1974 A0 00      55          CHARBUFR byt SPACE,ZERO
1976         56          ;
1976         57          VSNBUFR    dfs VSNLEN+1,ZERO
198B         58          ;
198B         59          SECMAP    dfs 16,ZERO
199B         60          ;
```

```

199B          61 ;
199B 01 0A 64 62 DECTBL  byt 1,10,100,1000,10000
199E E8 10
19A0 00 00 00 63 DECTBLH hby 1,10,100,1000,10000
19A3 03 27
19A5          64 ;
19A5          65 ;
19A5 00 0D 0B 66 INTRLEAV hex 000D0B0907050301
19A8 09 07 05
19AB 03 01
19AD 0E 0C 0A 67          hex 0E0C0A080604020F
19B0 08 06 04
19B3 02 0F
19B5          68 ;
19B5          69 ;
19B5          70 TBLTYPE dfs 1,1
19B6          71 SNUM16:
19B6          72 SLOT16  dfs 1,ZERO
19B7          73 DNUM:
19B7          74 DRIVE    dfs 1,ZERO
19B8          75 VOLEXPT:
19B8          76 VOLUME   dfs 1,ZERO
19B9          77 TNUM:
19B9          78 TRACK    dfs 1,ZERO
19BA          79 SNUM:
19BA          80 SECTOR   dfs 1,ZERO
19BB 00 00 81 DCTADR    adr *-*
19BD          82 USRBUF:
19BD 00 00 83 DATABUFR  adr *-*
19BF          84 IOCBPHAS  dfs 1,ZERO
19C0          85 BYTCNT   dfs 1,ZERO
19C1          86 CMDCODE   dfs 1,ZERO
19C2          87 ERRCODE   dfs 1,ZERO
19C3          88 VOLFND    dfs 1,ZERO
19C4          89 SLOTFND   dfs 1,ZERO
19C5          90 DRVFND    dfs 1,ZERO
19C6          91 ;
0011          92 TBLSIZE   equ *-TBLTYPE
19C6          93 ;
19C6          94 ;

```

BSAVE SCAN,A\$0900,B,L\$10C6

```

19C6          95          usr SCAN
19C6          96 ;
19C6          97 ;
19C6          98          dfs PAGESIZE-*&NEGONE
1A00          99 ;
1A00         100 ;
1A00         101 NBUF1    dfs PAGESIZE
1B00         102 ;
1B00         103 NBUF2    dfs NBUF2SIZ
1B56         104 ;
1B56         105 WRNIBL   dfs WNIBLSIZ
1B96         106 ;
1B96         107 RDNIBL   dfs RNIBLSIZ
1C00         108 ;
1C00         109 ;
1C00         110 BUFFER    dfs PAGESIZE
1D00         111 ;
1D00         112 ;

```

```
1D00          113  TRACKBUF:
1D00          114  ;
1D00          115  ;
1D00          116          stt "VTOC Manager Symbol Table"
1D00          117  ;
1D00          118  ;
1D00          119          end 111
```

```
*** End of Assembly
```

Symbol List starts at 0x7800, ends at 0x87FA, used 0x0FFA, remaining 0x2CC6

## Symbols unsorted:

LOC0	0000	WNDLFT	0020	WNDWDTH	0021	WNDTOP	0022	WNDBTM	0023
CH	0024	CV	0025	TEMPZ	0026	TEMP2Z	0027	CURTRKZ	002A
SLOT16Z	002B	DRVFLAG	002C	ADRDATMK	002C	ADRFIELD	002C	SECENDZ	002D
TRKFNDZ	002E	VOLFNDZ	002F	INVFLG	0032	PROMPT	0033	PHASE	0034
PAGECNT	0035	SYNCRNT	0035	CSWL	0036	ROMTEMPZ	003C	MOTORTIM	003C
BUFADR2Z	003E	ODDBITSZ	003E	SECTORZ	003F	TRACKZ	0040	VOLUMEZ	0041
IOBADR	004A	GENPTR	00EE	DATAPTR	00FA	PRNTPTR	00FC	ZERO	0000
NEGONE	00FF	SEEKCMD	0000	READCMD	0001	WRITCMD	0002	FRMTCMD	0004
TEXTMODE	0000	GRPHMODE	0001	TX80MODE	0002	LV80MODE	0003	NORMDISP	0000
INVRDISP	0001	INITSCRN	0000	HOMESCRN	0001	EOLCLR	0000	EOPCLR	0001
DIRECT	0000	INDIRECT	0001	NOPAD	0000	HIGHLOW	0020	ZEROPAD	0040
SPCPAD	0080	NEXTLINE	0004	CHARCELL	0007	PCMDMASK	000F	NIBLMASK	000F
VSNLEN	0014	NAMELEN	0018	CVMASK	001F	MAXWDTH	0028	MAXTRKS	0032
TRKMASK	003F	MAXCH	0050	MINCV	0060	ASCIMASK	007F	INVRMASK	007F
MSBMASK	0080	ASCIFLAG	0080	RTNCMD	0050	MODECMD	0051	DISPCMD	0052
SCRNCMD	0053	CLRCMD	0054	CNTRCMD	0055	BUFRCMD	0056	NIBLCMD	0057
BYT1CMD	0058	BYT2CMD	0059	BYTNCMD	005A	ADRCMD	005B	DEC1CMD	005C
DEC2CMD	005D	DEC3CMD	005E	DECNCMD	005F	FLASH	0060	CTRLD	0084
BELLCHAR	0087	LARROW	0088	DARROW	008A	UARROW	008B	RETURN	008D
CTRLQ	0091	RARROW	0095	ESCAPE	009B	SPACE	00A0	PERIOD	00AE
SLASH	00AF	COLON	00BA	LWRMASK	00DF	LWRCASE	00E0	DELETE	00FF
DFLTPHAS	0004	DFLTSEC	0010	DFLTRACK	0023	MAXTRACK	0030	PHASMAX	0010
HDRSYNC	0006	MINSYNC	0008	MAXSYNC	0020	MAXRETRY	0020	SYNCRNT	0050
DRIVE1	0001	DRIVE2	0002	RWTSSEEK	0000	RWTSREAD	0001	RWTSWRIT	0002
RWTSFRMT	0004	RWNOERR	0000	RWINITER	0008	RWPROTER	0010	RWVOLERR	0020
RWSYNERR	0030	RWDRVERR	0040	RWREADER	0080	WNIBLSIZ	0040	NBUF2SIZ	0056
RNIBLSIZ	006A	ODDBITS	00AA	ADRMARK1	00D5	ADRMARK2	00AA	ADRMARK3	0096
DATMARK1	00D5	DATMARK2	00AA	DATMARK3	00AD	SLPMARK1	00DE	SLPMARK2	00AA
SLPMARK3	00EB	SYNCRMARK	00FF	MOTONTIM	D8EF	STACK	0100	PAGESIZE	0100
DOSWARM	03D0	DOSCOLD	03D3	CALLRWTS	03D9	RDCLKVSN	03E1	GETIOB	03E3
HOOKDOS	03EA	DRV0TRK	0478	DRV1TRK	04F8	DRV0PHAS	0678	DRV1PHAS	06F8
FINDTRK	0478	RECALCNT	04F8	SEEKCRNT	0578	RETRYCNT	05F8	NEXTON	0678
NEXTOFF	06F8	ALLOCNTR	0778	MSLOT	07F8	BUFSIZE	1C00	MNGDISK	BFF2
PAGEC0	C000	PAGED0	D000	KEY	C000	CLRKEY	C010	TXTCCLR	C050
TXTCSET	C051	MIXCLR	C052	LOWSCR	C054	HIRES	C057	ROM2WP	C082
PHASEOFF	C080	MOTOROFF	C088	MOTORON	C089	DRV0EN	C08A	DRV1EN	C08B
STROBE	C08C	LATCH	C08D	DATAIN	C08E	DATAOUT	C08F	CLRROM	CFFF
PRNTAX	F941	PRBL2	F94A	INIT	FB2F	TABV	FB5B	VTAB	FC22
CLREOP	FC42	HOME	FC58	CLREOL	FC9C	CROUT	FD8E	PRBYTE	FDDA
PRHEX	FDE3	COUT	FDED	OUTPORT	FE95	BELL	FF3A	MONITOR	FF69
DISPLAY	0000	GETOPTN	0943	GETSDP	096F	GETSLOT	0984	GETDISK	09E7
GETDRV	09EB	GETPHS	0A26	GETRKRNG	0A64	GETSTART	0A72	GETEND	0AB3
DOSCAN	0AEF	SHOWERR	0C3E	READTRK	0C4C	SHOWADR	0C8F	SHOWDATA	0CD9
DOFORMAT	0CFF	SHOWAVG	0DA1	SHOWUSE	0DF1	QUIT	0E14	FINDADR	0E22
FINDEND	0E93	FINDATA	0EB7	DOCOUNT	0F3E	GETDATA	0F5C	SHOWMENU	0F71
READVSN	103A	DORWTS	103E	RWTSENT	10B1	SYNERR	1199	RWPERR	11CF
RWTSERR	11D1	RWTSEXIT	11F0	ERREXIT	11F3	DORWPERR	11FD	DISKFMT	1201
TRACKFMT	124F	SCRCHTBL	12B1	DISKJMP	1300	DISKADRS	1303	WRITSCTR	1311
WRITEXIT	1367	SETREAD	1381	WRITADR	1388	WRITSYNCR	13B6	WBYTE	13DE
WNIBL9	13ED	WNIBL2	13EF	WNIBL	13F0	ONOFFTBL	13F7	OTBLLEN	000C
READSCTR	1403	READERR	143E	READADR	1440	READEXIT	1466	READMRKS	147B
SAVETRKR	14A9	SAVETRKRX	14B6	GENINDEX	14C9	MOVHEADN	14D4	MOVHEAD0	14DB
MOVHEAD	14DD	RTN16	14E4	MOVEHD	14E5	CHKPOS	1541	MSWAIT	155C
WAIT24	156D	WAIT12	1570	INITPGM	1571	GETHEX2	15AF	GETHEX3	15B2
GETHEX4	15B5	GETHEX6	15B8	GETHEX8	15BB	GETDEC1	15C0	GETDEC2	15C3
GETDEC3	15C6	GETDEC5	15C9	NUMLOOP	15DA	NUMPROC	163A	HEXPROC	163D

DECPROC	166C	GETKEY	1695	READKEY	16AD	PRNTCHAR	16BF	PRINT	16CB
PRNTLOOP	16DA	PRNTMOD1	1706	PRNTMOD2	1721	PRNTBR1	1723	PRNTBR2	1729
PRNTBR3	172E	PRNTBR4	173C	PRNTMOD3	173F	PRNTOUT	1745	OUTMOD1	1745
OUTMOD2	1747	PRNTOUT2	1749	OUTTBL1	174C	OUTTBL2	1752	OUT80COL	1758
VTABADRS	175A	OUTADRS	1760	PRNTSAV	1766	PRNTNUM	1767	MODEVAL	1769
FRMTVAL	176A	PRNTBL	176B	PRNTBLL	177B	PRNTBLH	178B	PRNTRTN	179B
PRNTSAVY	17A6	PRNTSAVX	17A8	PRNTSAVA	17AA	PRINTRTN	17AD	PRNTMODE	17AE
PRNTDISP	17F9	PRNTSCRN	1818	PRNTCLR	1828	PRNTCNTR	1832	PRNTBUFR	1844
PRNTNIBL	185F	PRNTNBYT	1864	PRNT1BYT	1868	PRNT2BYT	186B	PRNTBYT	186D
PRNTADR	1877	PRNT1DEC	188F	PRNT2DEC	1895	PRNTDEC	1898	PRNT3DEC	18A2
PRNTNDEC	18AC	PRNTBYTE	18F3	PRNTHex	18FC	PRNTHex2	18FE	HEXTODEC	1909
GETDIGIT	1923	PRNTGRPH	1942	VALSPHAS	1942	ENDTRK	1943	ENDSEC	1944
SYNCOUNT	1945	ERRFLAG	1946	STAGE	1947	SECCNT	1948	TRACKSIZ	1949
STRTRACK	194A	ENDTRACK	194B	SYNCNUM	194C	HDRCNT	194D	SLOT	194E
TEMP	194F	CHKSUM	1950	HDRVOL	1951	HDRTRK	1952	HDRSEC	1953
HDRCHK	1954	HDRGAP	1955	NUMBUF	1956	DATAcnt	1958	SYNCTOTL	195A
SAVX	195C	SAVY	195D	SAVA	195E	OPTION	195F	SAVNUM	1960
COUNT	1961	COUNT2	1962	NNUMS	1963	SAVECH	1964	NENTRY	1965
NUMVALS	1966	HEXVALS	196E	DECVALS	1972	CHARBUFR	1974	VSNBUFR	1976
SECMAP	198B	DECTBLL	199B	DECTBLH	19A0	INTRLEAV	19A5	TBLTYPE	19B5
SNUM16	19B6	SLOT16	19B6	DNUM	19B7	DRIVE	19B7	VOLEXPT	19B8
VOLUME	19B8	TNUM	19B9	TRACK	19B9	SNUM	19BA	SECTOR	19BA
DCTADR	19BB	USRBUF	19BD	DATABUFR	19BD	IOCBPHAS	19BF	BYTCNT	19C0
CMDCODE	19C1	ERRCODE	19C2	VOLFND	19C3	SLOTFND	19C4	DRVFND	19C5
TBLSIZE	0011	NBUF1	1A00	NBUF2	1B00	WRNIBL	1B56	RDNIBL	1B96
BUFFER	1C00	TRACKBUF	1D00						

## Symbols alphabetically sorted:

ADRCMD	005B	ADRDATMK	002C	ADRFIELD	002C	ADRMARK1	00D5	ADRMARK2	00AA
ADRMARK3	0096	ALLOCNTR	0778	ASCIFLAG	0080	ASCIMASK	007F	BELL	FF3A
BELLCHAR	0087	BUFADR2Z	003E	BUFFER	1C00	BUFRCMD	0056	BUFSIZE	1C00
BYT1CMD	0058	BYT2CMD	0059	BYTCNT	19C0	BYTNCMD	005A	CALLRWTS	03D9
CH	0024	CHARBUFR	1974	CHARCELL	0007	CHKPOS	1541	CHKSUM	1950
CLRCMD	0054	CLREOL	FC9C	CLREOP	FC42	CLRKEY	C010	CLRROM	CFFF
CMDCODE	19C1	CNTRCMD	0055	COLON	00BA	COUNT	1961	COUNT2	1962
COUT	FD0D	CROUT	FD8E	CSWL	0036	CTRLD	0084	CTRLQ	0091
CURTRKZ	002A	CV	0025	CVMASK	001F	DARROW	008A	DATABUFR	19BD
DATAcnt	1958	DATAIN	C08E	DATAOUT	C08F	DATAPTR	00FA	DATMARK1	00D5
DATMARK2	00AA	DATMARK3	00AD	DCTADR	19BB	DEC1CMD	005C	DEC2CMD	005D
DEC3CMD	005E	DECNCMD	005F	DECPROC	166C	DECTBLH	19A0	DECTBLL	199B
DECVALS	1972	DELETE	00FF	DFLTPHAS	0004	DFLTRACK	0023	DFLTSEC	0010
DIRECT	0000	DISKADRS	1303	DISKFMT	1201	DISKJMP	1300	DISPCMD	0052
DISPLAY	0000	DNUM	19B7	DOCOUNT	0F3E	DOFORMAT	0CFF	DORWPERR	11FD
DORWTS	103E	DOSCAN	0AEF	DOSCOLD	03D3	DOSWARM	03D0	DRIVE	19B7
DRIVE1	0001	DRIVE2	0002	DRV0EN	C08A	DRV0PHAS	0678	DRV0TRK	0478
DRV1EN	C08B	DRV1PHAS	06F8	DRV1TRK	04F8	DRVFLAG	002C	DRVFND	19C5
ENDSEC	1944	ENDTRACK	194B	ENDTRK	1943	EOLCLR	0000	EOPCLR	0001
ERRCODE	19C2	ERREXIT	11F3	ERRFLAG	1946	ESCAPE	009B	FINDADR	0E22
FINDATA	0EB7	FINDEND	0E93	FINDTRK	0478	FLASH	0060	FRMTCMD	0004
FRMTVAL	176A	GENINDEX	14C9	GENPTR	00EE	GETDATA	0F5C	GETDEC1	15C0
GETDEC2	15C3	GETDEC3	15C6	GETDEC5	15C9	GETDIGIT	1923	GETDISK	09E7
GETDRV	09EB	GETEND	0AB3	GETHEX2	15AF	GETHEX3	15B2	GETHEX4	15B5
GETHEX6	15B8	GETHEX8	15BB	GETIOB	03E3	GETKEY	1695	GETOPTN	0943
GETPHS	0A26	GETRKRNG	0A64	GETSDP	096F	GETSLOT	0984	GETSTART	0A72
GRPHMODE	0001	HDRCHK	1954	HDRCNT	194D	HDRGAP	1955	HDRSEC	1953
HDRSYNC	0006	HDRTRK	1952	HDRVOL	1951	HEXPROC	163D	HEXTODEC	1909
HEXVALS	196E	HIGHLOW	0020	HIRES	C057	HOME	FC58	HOMESCRN	0001
HOOKDOS	03EA	INDIRECT	0001	INIT	FB2F	INITPGM	1571	INITSCRN	0000
INTRLEAV	19A5	INVFLG	0032	INVRDISP	0001	INVRMASK	007F	IOBADR	004A

IOCBPHAS	19BF	KEY	C000	LARROW	0088	LATCH	C08D	LOC0	0000
LOWSCR	C054	LV80MODE	0003	LWRCASE	00E0	LWRMASK	00DF	MAXCH	0050
MAXRETRY	0020	MAXSYNC	0020	MAXTRACK	0030	MAXTRKS	0032	MAXWIDTH	0028
MINCV	0060	MINSYNC	0008	MIXCLR	C052	MNGDISK	BFF2	MODECMD	0051
MODEVAL	1769	MONITOR	FF69	MOTONTIM	D8EF	MOTOROFF	C088	MOTORON	C089
MOTORTIM	003C	MOVEHD	14E5	MOVHEAD	14DD	MOVHEAD0	14DB	MOVHEADN	14D4
MSBMASK	0080	MSLOT	07F8	MSWAIT	155C	NAMELEN	0018	NBUF1	1A00
NBUF2	1B00	NBUF2SIZ	0056	NEGONE	00FF	NENTRY	1965	NEXTLINE	0004
NEXTOFF	06F8	NEXTON	0678	NIBLCMD	0057	NIBLMASK	000F	NNUMS	1963
NOPAD	0000	NORMDISP	0000	NUMBUF	1956	NUMLOOP	15DA	NUMPROC	163A
NUMVALS	1966	ODDBITS	00AA	ODDBITSZ	003E	ONOFFTBL	13F7	OPTION	195F
OTBLEN	000C	OUT80COL	1758	OUTADRS	1760	OUTMOD1	1745	OUTMOD2	1747
OUTPORT	FE95	OUTTBL1	174C	OUTTBL2	1752	PAGEC0	C000	PAGECNT	0035
PAGED0	D000	PAGESIZE	0100	PCMDMASK	000F	PERIOD	00AE	PHASE	0034
PHASEOFF	C080	PHASMAX	0010	PRBL2	F94A	PRBYTE	FDDA	PRHEX	FDE3
PRINT	16CB	PRINTRTN	17AD	PRNT1BYT	1868	PRNT1DEC	188F	PRNT2BYT	186B
PRNT2DEC	1895	PRNT3DEC	18A2	PRNTADR	1877	PRNTAX	F941	PRNTBL	176B
PRNTBLH	178B	PRNTBLL	177B	PRNTBR1	1723	PRNTBR2	1729	PRNTBR3	172E
PRNTBR4	173C	PRNTBUFR	1844	PRNTBYT	186D	PRNTBYTE	18F3	PRNTCHAR	16BF
PRNTCLR	1828	PRNTCNTR	1832	PRNTDEC	1898	PRNTDISP	17F9	PRNTGRPH	1942
PRNTHX	18FC	PRNTHX2	18FE	PRNTLOOP	16DA	PRNTMOD1	1706	PRNTMOD2	1721
PRNTMOD3	173F	PRNTMODE	17AE	PRNTNBYT	1864	PRNTNDEC	18AC	PRNTNIBL	185F
PRNTNUM	1767	PRNTOUT	1745	PRNTOUT2	1749	PRNTPTR	00FC	PRNTRTN	179B
PRNTSAV	1766	PRNTSAVA	17AA	PRNTSAVX	17A8	PRNTSAVY	17A6	PRNTSCRN	1818
PROMPT	0033	QUIT	0E14	RARROW	0095	RDCLKVSN	03E1	RDNIBL	1B96
READADR	1440	READCMD	0001	READERR	143E	READEXIT	1466	READKEY	16AD
READMRKS	147B	READSCTR	1403	READTRK	0C4C	READVSN	103A	RECALCNT	04F8
RETRYCNT	05F8	RETURN	008D	RNIBLSIZ	006A	ROM2WP	C082	ROMTEMPZ	003C
RTN16	14E4	RTNCMD	0050	RWDRVERR	0040	RWINITER	0008	RWNOERR	0000
RWPERR	11CF	RWPROTER	0010	RWREADER	0080	RWSYNERR	0030	RWTSENT	10B1
RWTSEERR	11D1	RWTSEXIT	11F0	RWTSFRMT	0004	RWTSREAD	0001	RWTSSEEK	0000
RWTSWRIT	0002	RWVOLERR	0020	SAVA	195E	SAVECH	1964	SAVETRК	14A9
SAVETRКX	14B6	SAVNUM	1960	SAVX	195C	SAVY	195D	SCRCHTBL	12B1
SCRNCMD	0053	SECCNT	1948	SECFNDZ	002D	SECMAP	198B	SECTOR	19BA
SECTORZ	003F	SEEKCMD	0000	SEEKCNT	0578	SETREAD	1381	SHOWADR	0C8F
SHOWAVG	0DA1	SHOWDATA	0CD9	SHOWERR	0C3E	SHOWMENU	0F71	SHOWUSE	0DF1
SLASH	00AF	SLOT	194E	SLOT16	19B6	SLOT16Z	002B	SLOTFND	19C4
SLPMARK1	00DE	SLPMARK2	00AA	SLPMARK3	00EB	SNUM	19BA	SNUM16	19B6
SPACE	00A0	SPCPAD	0080	STACK	0100	STAGE	1947	STROBE	C08C
STRTRACK	194A	SYNCBITS	0500	SYNCMARK	00FF	SYNCNT	0035	SYNCNUM	194C
SYNCOUNT	1945	SYNCTOTL	195A	SYNERR	1199	TABV	FB5B	TBLSIZE	0011
TBLTYPE	19B5	TEMP	194F	TEMP2Z	0027	TEMPZ	0026	TEXTMODE	0000
TNUM	19B9	TRACK	19B9	TRACKBUF	1D00	TRACKFMT	124F	TRACKSIZ	1949
TRACKZ	0040	TRKFNDZ	002E	TRKMASK	003F	TX80MODE	0002	TXTCLR	C050
TXTSET	C051	UARROW	008B	USRBUF	19BD	VALSPHAS	1942	VOLEXPT	19B8
VOLFND	19C3	VOLFNDZ	002F	VOLUME	19B8	VOLUMEZ	0041	VSNBUFR	1976
VSNLEN	0014	VTAB	FC22	VTABADRS	175A	WAIT12	1570	WAIT24	156D
WBYTE	13DE	WNDBTM	0023	WNDLFT	0020	WNDTOP	0022	WNDWIDTH	0021
WNIBL	13F0	WNIBL2	13EF	WNIBL9	13ED	WNIBLSIZ	0040	WRITADR	1388
WRITCMD	0002	WRITEXIT	1367	WRITSCTR	1311	WRITSYNC	13B6	WRNIBL	1B56
ZERO	0000	ZEROPAD	0040						

## Symbols numerically sorted:

ZERO	0000	TEXTMODE	0000	SEEKCMD	0000	RWTSSEEK	0000	RWNOERR	0000
NORMDISP	0000	NOPAD	0000	LOC0	0000	INITSCRN	0000	EOLCLR	0000
DISPLAY	0000	DIRECT	0000	RWTSREAD	0001	READCMD	0001	INVRDISP	0001
INDIRECT	0001	HOMESCRN	0001	GRPHMODE	0001	EOPCLR	0001	DRIVE1	0001
WRITCMD	0002	TX80MODE	0002	RWTSWRIT	0002	DRIVE2	0002	LV80MODE	0003
RWTSFRMT	0004	NEXTLINE	0004	FRMTCMD	0004	DFLTPHAS	0004	HDRSYNC	0006



CHARCELL	0007	RWINITER	0008	MINSYNC	0008	OTBLLEN	000C	PCMDMASK	000F
NIBLMASK	000F	RWPROTER	0010	PHASMAX	0010	DFLTSEC	0010	TBLSIZE	0011
VSNLEN	0014	NAMELEN	0018	CVMASK	001F	WNDLFT	0020	RWVOLERR	0020
MAXSYNC	0020	MAXRETRY	0020	HIGHLOW	0020	WNDWDTH	0021	WNDTOP	0022
WNCBTM	0023	DFLTRACK	0023	CH	0024	CV	0025	TEMPZ	0026
TEMP2Z	0027	MAXWDTH	0028	CURTRKZ	002A	SLOT16Z	002B	DRVFLAG	002C
ADRFIELD	002C	ADRDATMK	002C	SECFNDZ	002D	TRKFNDZ	002E	VOLFNDZ	002F
RWSYNERR	0030	MAXTRACK	0030	MAXTRKS	0032	INVFLG	0032	PROMPT	0033
PHASE	0034	SYNCNT	0035	PAGECNT	0035	CSWL	0036	ROMTEMPZ	003C
MOTORTIM	003C	ODDBITSZ	003E	BUFADR2Z	003E	TRKMASK	003F	SECTORZ	003F
ZEROPAD	0040	WNIBLSIZ	0040	TRACKZ	0040	RWDRVERR	0040	VOLUMEZ	0041
IOBADR	004A	RTNCMD	0050	MAXCH	0050	MODECMD	0051	DISPCMD	0052
SCRNCMD	0053	CLRCMD	0054	CNTRCMD	0055	NBUF2SIZ	0056	BUFRCMD	0056
NIBLCMD	0057	BYT1CMD	0058	BYT2CMD	0059	BYTNCMD	005A	ADRCMD	005B
DEC1CMD	005C	DEC2CMD	005D	DEC3CMD	005E	DECNCMD	005F	MINCV	0060
FLASH	0060	RNIBLSIZ	006A	INVRMASK	007F	ASCIMASK	007F	SPCPAD	0080
RWREADER	0080	MSBMASK	0080	ASCIFLAG	0080	CTRLD	0084	BELLCHAR	0087
LARROW	0088	DARROW	008A	UARROW	008B	RETURN	008D	CTRLQ	0091
RARROW	0095	ADRMARK3	0096	ESCAPE	009B	SPACE	00A0	SLPMARK2	00AA
ODDBITS	00AA	DATMARK2	00AA	ADRMARK2	00AA	DATMARK3	00AD	PERIOD	00AE
SLASH	00AF	COLON	00BA	DATMARK1	00D5	ADRMARK1	00D5	SLPMARK1	00DE
LWRMASK	00DF	LWRCASE	00E0	SLPMARK3	00EB	GENPTR	00EE	DATAPTR	00FA
PRNTPTR	00FC	SYNCMARK	00FF	NEGONE	00FF	DELETE	00FF	STACK	0100
PAGESIZE	0100	DOSWARM	03D0	DOSCOLD	03D3	CALLRWTS	03D9	RDCLKVSN	03E1
GETIOB	03E3	HOOKDOS	03EA	FINDTRK	0478	DRV0TRK	0478	RECALCNT	04F8
DRV1TRK	04F8	SYNCBITS	0500	SEEKCNT	0578	RETRYCNT	05F8	NEXTON	0678
DRV0PHAS	0678	NEXTOFF	06F8	DRV1PHAS	06F8	ALLOCNTR	0778	MSLOT	07F8
GETOPTN	0943	GETSDP	096F	GETSLOT	0984	GETDISK	09E7	GETDRV	09EB
GETPHS	0A26	GETRKRNG	0A64	GETSTART	0A72	GETEND	0AB3	DOSCAN	0AEF
SHOWERR	0C3E	READTRK	0C4C	SHOWADR	0C8F	SHOWDATA	0CD9	DOFORMAT	0CFF
SHOWAVG	0DA1	SHOWUSE	0DF1	QUIT	0E14	FINDADR	0E22	FINDEND	0E93
FINDATA	0EB7	DOCOUNT	0F3E	GETDATA	0F5C	SHOWMENU	0F71	READVSN	103A
DORWTS	103E	RWTSENT	10B1	SYNERR	1199	RWPERR	11CF	RWTSERR	11D1
RWTSEXIT	11F0	ERREXIT	11F3	DORWPERR	11FD	DISKFMT	1201	TRACKFMT	124F
SCRCHTBL	12B1	DISKJMP	1300	DISKADRS	1303	WRITSCTR	1311	WRITEEXIT	1367
SETREAD	1381	WRITADR	1388	WRITSYNC	13B6	WBYTE	13DE	WNIBL9	13ED
WNIBL2	13EF	WNIBL	13F0	ONOFFTBL	13F7	READSCTR	1403	READERR	143E
READADR	1440	READEXIT	1466	READMRKS	147B	SAVETRK	14A9	SAVETRFX	14B6
GENINDEX	14C9	MOVHEADN	14D4	MOVHEAD0	14DB	MOVHEAD	14DD	RTN16	14E4
MOVEHD	14E5	CHKPOS	1541	MSWAIT	155C	WAIT24	156D	WAIT12	1570
INITPGM	1571	GETHEX2	15AF	GETHEX3	15B2	GETHEX4	15B5	GETHEX6	15B8
GETHEX8	15BB	GETDEC1	15C0	GETDEC2	15C3	GETDEC3	15C6	GETDEC5	15C9
NUMLOOP	15DA	NUMPROC	163A	HEXPROC	163D	DECPROC	166C	GETKEY	1695
READKEY	16AD	PRNTCHAR	16BF	PRINT	16CB	PRNTLOOP	16DA	PRNTMOD1	1706
PRNTMOD2	1721	PRNTBR1	1723	PRNTBR2	1729	PRNTBR3	172E	PRNTBR4	173C
PRNTMOD3	173F	PRNTOUT	1745	OUTMOD1	1745	OUTMOD2	1747	PRNTOUT2	1749
OUTTBL1	174C	OUTTBL2	1752	OUT80COL	1758	VTABADRS	175A	OUTADRS	1760
PRNTSAV	1766	PRNTNUM	1767	MODEVAL	1769	FRMTVAL	176A	PRNTBL	176B
PRNTBLL	177B	PRNTBLH	178B	PRNTRTN	179B	PRNTSAVY	17A6	PRNTSAVX	17A8
PRNTSAVA	17AA	PRINTRTN	17AD	PRNTMODE	17AE	PRNTDISP	17F9	PRNTSCRN	1818
PRNTCLR	1828	PRNTCNTR	1832	PRNTBUFR	1844	PRNTNIBL	185F	PRNTNBYT	1864
PRNT1BYT	1868	PRNT2BYT	186B	PRNTBYT	186D	PRNTADR	1877	PRNT1DEC	188F
PRNT2DEC	1895	PRNTDEC	1898	PRNT3DEC	18A2	PRNTNDEC	18AC	PRNTBYTE	18F3
PRNTHEx	18FC	PRNTHEx2	18FE	HEXTODEC	1909	GETDIGIT	1923	VALSPHAS	1942
PRNTGRPH	1942	ENDTRK	1943	ENDSEC	1944	SYNCOUNT	1945	ERRFLAG	1946
STAGE	1947	SECCNT	1948	TRACKSIZ	1949	STRTRACK	194A	ENDTRACK	194B
SYNCNUM	194C	HDRCNT	194D	SLOT	194E	TEMP	194F	CHKSUM	1950
HDRVOL	1951	HDRTRK	1952	HDRSEC	1953	HDRCHK	1954	HDRGAP	1955
NUMBUF	1956	DATACNT	1958	SYNCTOTL	195A	SAVX	195C	SAVY	195D
SAVA	195E	OPTION	195F	SAVNUM	1960	COUNT	1961	COUNT2	1962
NNUMS	1963	SAVECH	1964	NENTRY	1965	NUMVALS	1966	HEXVALS	196E

DECVALS	1972	CHARBUFR	1974	VSNBUFR	1976	SECMAP	198B	DECTBLL	199B
DECTBLH	19A0	INTRLEAV	19A5	TBLTYPE	19B5	SNUM16	19B6	SLOT16	19B6
DRIVE	19B7	DNUM	19B7	VOLUME	19B8	VOLEXPT	19B8	TRACK	19B9
TNUM	19B9	SNUM	19BA	SECTOR	19BA	DCTADR	19BB	USRBUF	19BD
DATABUFR	19BD	IOCBPHAS	19BF	BYTCNT	19C0	CMDCODE	19C1	ERRCODE	19C2
VOLFND	19C3	SLOTFND	19C4	DRVFND	19C5	NBUF1	1A00	NBUF2	1B00
WRNIBL	1B56	RDNIBL	1B96	BUFSIZE	1C00	BUFFER	1C00	TRACKBUF	1D00
MNGDISK	BFF2	PAGEC0	C000	KEY	C000	CLRKEY	C010	TXTCLR	C050
TXTSET	C051	MIXCLR	C052	LOWSCR	C054	HIRES	C057	PHASEOFF	C080
ROM2WP	C082	MOTOROFF	C088	MOTORON	C089	DRV0EN	C08A	DRV1EN	C08B
STROBE	C08C	LATCH	C08D	DATAIN	C08E	DATAOUT	C08F	CLRROM	CFFF
PAGED0	D000	MOTONTIM	D8EF	PRNTAX	F941	PRBL2	F94A	INIT	FB2F
TABV	FB5B	VTAB	FC22	CLREOP	FC42	HOME	FC58	CLREOL	FC9C
CROUT	FD8E	PRBYTE	FDDA	PRHEX	FDE3	COUT	FDED	OUTPORT	FE95
BELL	FF3A	MONITOR	FF69						