

Micro Complex
25651 Minos Street
Mission Viejo, CA. 92691
(714) 770-2168

05 May 1980

Documetation on Micro Complex's "PHASE LOCK II" Dual Density Disk Controller and McDos (TM) Disk Operating System.

1. The Micro Complex disk controller has been designed to be Plug Compatible with the North Star, Inc. controller board. Therefore, it is capable of executing software programs which will execute on the North Star, Inc. controller. This compatibility has been intentionally maintained in order to support the large number of software packages which are being written by many other companies and users.

2. The Micro Complex "PHASE LOCK II" has several features which are extentions beyond the capiblities of the North Star, Inc. controller board. These are;

a. The controller's memory address can be re-located by an on-board dip switch. This is provided as an optional addition and should be specified when odering the board. The switch is not normally installed. This is due to the fact that most users will operate the board at the standard address of E800H. The "PHASE LOCK II" is delivered with the address E800H selected by circuit etch connections. However, the Dip Switch Option can be easily installed by cutting the two traces on the component side of the board. These traces presently run vertical between pins 4 & 9 and pins 6 & 7 at the dip switch location E3. Installation of the Dip Switch Option consists of cutting the two traces and soldering in the 6 pole dip switch. The memory address of the board may then be set by the switch. The switch sets the upper six bits of the address. Since the lines are normally pulled up to a logical one, the open position on the switch represents a 'logical one' for the address. Therefore, if the switch is installed, the E800H address would be set by a switch selection (reading left to right) of

OPEN/OPEN/OPEN/CLOSED/OPEN/CLOSED

Operation of the board at a different address also requiries that the Boot Prom software and DOS software be modified. Source listings of both are available at additional costs, either printed-out or on diskette. A simple means of re-loction this is being developed, and will be available very soon. If you are interested in re-addressing your board, please advise us.

b. There is some confusion in the industry as which pin on the ribbon cable header should be used to select drive #4 if installed. North Star, Inc. uses pin # 34. Pertec, Inc. and Tandon, Inc. uses pin #6. Suggart, Inc. uses pin #34 with user modification required for selection of a forth drive. On the "PHASE LOCK II" controller, the select line can be changed to pin #6 by jumpering between Pin #6 and Pin #34. Note: when looking at the front of the board, the header pins are numbered from "right to left, all odd numbered pins are grounded. The even numbered pins are the row nearest to you. Pin #6 is the third one from the right, in the lower row.

c. The interrupt circuit may be connected by simply jumpering between which ever of the interrupts you select (VI0 thru VI7 or NMI) as marked on the board. The other end of the jumper should connect to the plate through hole located between the V and I letters at the VI5 interrupt select label. After the jumper is added, the software must arm the interrupt in a manner similar to North Star, Inc.'s board.

d. The Automatic Motor-Off timer is user selectable to 4, 8, 16, or 32 seconds. As delivered it is set to 32 seconds. The option selection holes are located below the IC at location 3B on the board. The four holes in a row are left to right the 4, 8, 16 , & 32 second selection holes. The single hole, centered beneath these holes is for the other end of the timer selection jumper. You may note that this hole is jumpered, on the back of the board, to the 32 second selection holed. If you desired to change the selection, then cut this circuit, near the 32 second hole, on the backside, and then solder in a jumper to the desired timing hole.

***** "McDos" Disk Operating System Features *****

1. The McDos provides all the functions of the North Star, Inc. DOS. However, it's structure is considerably different. It uses no memory above 2A00H, so compatibility with all old programs is maintained. Instead, it uses a buffer area at 1D00H and 100H at 1F00H. Both of these areas will automatically re-load themselves if they are destroyed by user program execution. The re-load is not done until the function at that address is required for McDos to complete it's assigned command.

2. Copies of McDos can be made using either the LF/SF sequence or the RD/WR sequence of commands. It will not function properly if you save a copy of it's self from it's operating location in memory. If modifications to McDos are desired, they can be made by loading the code into memory and modifying it. Then re-save it on the disk. It should be noted that the McDos code for addresses 2000H to 2A00H is in sectors 4 thru 8 and that the code for 1E00H thru 1FFFH is in sector 9. This inverse order is necessary due to the method of bootstrapping which is employed. Therefore, if you are assembling a new version of McDos, it is necessary to load the new version with an off-set, for example with a bias of 2000H. Then McDos will be at addresses 3E00H to 4A00H. Then put it on the Disk by typing;

```
WR 4 4000 10
WR 9 3E00 2
```

3. A full source listing of the McDos is available for \$40.00. Source code of McDos in CP/M assembly format is available for an additional \$40.00. It includes code for the following special features;

a. Auto-Load of a user designated file at boot-up is triggered simply by putting it's name in the Keyboard Buffer, at 28B0H. You can easily do this using the commands; LF DOS 4000. Then insert the ASCII code for the filename at address 48B0H, followed by a RETURN code, 0DH. Then type; SF DOS 4000 and the DOS will automatically load and execute the file when you boot-up.

b. GO commands are not required, typing the file followed by a RETURN will automatically load and execute a file.

c. All six of the disk function error messages are output in plain text;

- (1) SYNC
- (2) CRC
- (3) VERIFY
- (4) INDEX
- (5) DENS
- (6) WRIT-P

d. An RN command is provided to rename files. It's form is;

RN Filename1,<Unit#> Filename2

e. An exit from the LI command is provided to escape in the middle of long file directory listings. Either the mode key or Control-C will abort the listing.

f. Additional jump vector commands are available;

B = Jump to Re-Boot
BA = Return to Basic at 2A00H
WP = Return to Word Processor
at 2A00H
M = Jump to the Monitor
(Solos at C004H)

These commands may be modified to different two letter commands and/or vector addresses by changing the table which begins at address 280AH

g. The I/O area is at 2900H and can be modified to suit your requirements.

h. The start-up initialization and Banner may be modified by changing the code at 1E00H. This code gets over-written after boot-up is completed.

i. The SOL computer McDos is set up to provide protocol handshake with a Diablo printer on the serial port. If this is not desired, the replace the RNZ code at (SOL version) 2910H with a RET code, i.e., the C0 becomes C9.

j. McDos commands and Hexidecimal numbers may now be entered in either upper or lower case. Note: Commands CO,CF,CD, etc. are really calls to files stored on the disk and therefore the upper/lower case must be the same as the filename as listed in the disk directory.

k. Two new file routines are provided on the system disk to allow you to MERGE DISKS. They are, MD.S and MD.D, which will merge the files from a single or double density disk in drive #2 onto a disk in drive #1. This is very useful when converting from single density systems to double density systems.

That's about all the information we have prepared at present. Schematics for the boards are available.

Micro Complex will repair all boards returned, if user pre-pays postage both ways. Subject to our evaluation as to wheather the board has been abused, the repair will be free except for the postage, until 90 days after purchase date.

More detailed information and coding to provide board address re-location is available for those of you who would like to re-map your memory configurations.

If you need additional aid in the operation of McDos, please call me at Micro Complex (714) 770-2168,

Robert Lee Hogg.

```

1 0000
2 0000 ;0080
3 0000 ;QUAD 7.0
4 0000 ;04/26/80
5 0000 ;QUAD DOS WITH N.S. COMPATABLE BYTE AT 2034H
6 0000 ;
7 0000 ;
8 0000 ;*****
9 0000 ;
10 0000 ; DOUBLE DENSITY CONTROLLER *
11 0000 ; DISK OPERATING SYSTEM *
12 0000 ; by Robert Lee Hogg *
13 0000 ; Mission Viejo, CA. *
14 0000 ; (714) 770-2168 *
15 0000 ; Mc Dos (TM) / COPYRIGHTS (C) MICRO COMPLEX 1980 *
16 0000 ;*****
17 0000
18 0000 ;Controller Codes
19 0000
20 0000 2000 DBASE EQU 02000H ;DOS Base Assignment
21 0000 E000 PBASE EQU 0E800H ;FROM Base Assignment
22 0000 E900 CWRITE EQU PBASE+100H
23 0000 EA00 CORDER EQU PBASE+200H
24 0000 EB00 COMAND EQU PBASE+300H
25 0000 ;
26 0000 EB10 STATA EQU COMAND+10H
27 0000 EB20 STATB EQU COMAND+20H
28 0000 EB30 STATC EQU COMAND+30H
29 0000 EB40 READD EQU COMAND+40H
30 0000 ;
31 0000 0010 ORDST EQU 10H ;Order Step control High
32 0000 0020 ORDDP EQU 20H ;Order Step-In or Write PreComp
33 0000 0040 ORDSS EQU 40H ;Order Second Side of Disk
34 0000 0080 ORDDD EQU 80H ;Order Double Density
35 0000 ;
36 0000 0000 NO.OP EQU 0
37 0000 0001 RSTSF EQU 1
38 0000 0002 DARMI EQU 2
39 0000 0003 ARMI EQU 3
40 0000 0004 SETBODY EQU 4
41 0000 0005 MOTRON EQU 5
42 0000 0006 BWRITE EQU 6
43 0000 0007 RESET EQU 7
44 0000 ;
45 0000 0000 DS0 EQU 0 ;Drive Select - NONE
46 0000 0001 DS1 EQU 1 ;Drive Select 1
47 0000 0002 DS2 EQU 2 ;Drive Select 2
48 0000 0004 DS3 EQU 4 ;Drive Select 3
49 0000 0008 DS4 EQU 8 ;Drive Select 4
50 0000 ;
51 0000 0010 MO EQU 10H
52 0000 0020 DD EQU 20H
53 0000 0040 IX EQU 40H
54 0000 0080 SF EQU 80H
55 0000 ;

```

```

56 0000 0001 A.BD EQU 01H
57 0000 0002 A.SP EQU 02H
58 0000 0004 A.RE EQU 04H
59 0000 0008 A.WI EQU 08H
60 0000 ;
61 0000 0001 B.T0 EQU 01H
62 0000 0002 B.WP EQU 02H
63 0000 0004 B.SP EQU 04H
64 0000 0008 B.WR EQU 08H
65 0000 ;
66 0000 00FB SYNC EQU 0FBH
67 0000
68 0000
69 0000 ;System Definitions
70 0000
71 0000 1D00 BUFFER EQU 01D00H ;DOS Buffer Assignment
72 0000 1E00 STAK2 EQU 01E00H ;Second BOOT Stack Address
73 0000 1F00 ISTACK EQU 01F00H ;Initial BOOT Stack Address
74 0000 2A00 WP EQU 02A00H ;Word Processor/Basic Vector
75 0000 2A04 BASIC EQU 02A04H ;Basic ReEntry, NO Clear
76 0000 C004 MONTR EQU 0C004H ;SOLOS Monitor Entry
77 0000 C019 SOUT EQU 0C019H ;SOLOS SOUT
78 0000 C01C AOUT EQU 0C01CH ;SOLOS AOUT
79 0000 C01F SINP EQU 0C01FH ;SOLOS SINP
80 0000 C022 AINP EQU 0C022H ;SOLOS AINP
81 0000 C807 OPORT EQU 0C807H ;SOLOS Output Psuedo Port Store
82 0000 C80B SPEED EQU 0C80BH ;SOLOS Speed Control Byte
83 0000
84 0000
85 0000 0020 KBFSZ EQU 20H ;Define Keyboard Buffer Size
86 0000 0030 STKSZ EQU 30H ;Define Stack Size
87 0000 0004 DIRSZ EQU 04H ;Define Directory Size in Blocks
88 0000 000A NSECTR EQU 10 ;Number of Sectors per Track
89 0000 0023 NTRKS EQU 35 ;Number of Tracks/Side of Disk
90 0000 0059 TRK.? EQU 059H ;Unknown Head Track Flag
91 0000 002B PROMPT EQU '+'
92 0000 008B CLRSCN EQU 08BH
93 0000 0040 ABORT EQU 'Q'
94 0000 0080 MODE EQU 080H
95 0000 008C LOAD EQU 08CH
96 0000 0020 SPACE EQU 020H
97 0000 002C COMMA EQU ','
98 0000 0001 SETX EQU 001H
99 0000 0002 SETY EQU 002H
100 0000 0008 BS EQU 008H
101 0000 0009 TAB EQU 009H
102 0000 0003 CTL0C EQU 003H
103 0000 001B ESC EQU 01BH
104 0000 00FF EOT EQU 0FFH
105 0000 0003 ETX EQU 03H
106 0000 0006 ACK EQU 06H
107 0000 000D CR0 EQU 0DH
108 0000 000A LFC EQU 0AH
109 0000
110 0000 PAGE

```

```

111 0000
112 0000
113 0000      2000      ORG      DBASE
114 2000
115 2000 20      BOOTAD: DB      DBASE/100H ;High Byte of DOS Boot-Up Address
116 2001      2000      HEAD1: EQU      $-1      ;HEAD1 Position Counter
117 2001 59      HEAD2: DB      TRK.?      ;HEAD2 Position Counter
118 2002 59      HEAD3: DB      TRK.?      ;HEAD3 Position Counter
119 2003 59      HEAD4: DB      TRK.?      ;HEAD4 Position Counter
120 2004 00      SINGFLG DB      00H      ;Single Den Controller Flag if 0C3H
121 2005 00      DB      00H      ;*****SPARE*****
122 2006 01      UNIT:  DB      001H      ;DOS Unit Selection Store
123 2007 C9      OFTEN: DB      0C9H      ;Often Re-visited Vector
124 2008 0000      DW      0000H      ;Routine Address to be placed here.
125 200A
126 200A C33920      BOOTV: JMP      BOOT      ;DOS Initialization
127 200D C30029      COUTV: JMP      OUTPR      ;Console Output
128 2010 C33029      CINV:  JMP      INPUT      ;Console Input
129 2013 C3A01E      TINTV: JMP      TINIT      ;Terminal Initialization (ReBoot)
130 2016 C38029      CNTCV: JMP      CNTLC      ;Control-C Routine
131 2019 C34D22      HDERV: JMP      HDERR      ;Hard Disk Error Routine
132 201C C3DE23      DLOKV: JMP      DLOOK      ;Disk Directory Look-Up
133 201F C3C522      DWRTV: JMP      DWRT      ;Disk Directory Update
134 2022 C35323      DCOMV: JMP      DCOM      ;Disk Command Routine
135 2025 C31D26      LISTV: JMP      LIST      ;List Disk Directory
136 2028 C33727      DOSV:  JMP      DOS      ;DOS Entry Vector
137 202B 01      PWCHK:  DB      001H      ;Read after Write Flag, 0=OFF, 1=ON
138 202C C32C27      QMARK: JMP      QMRK0      ;Error Question Mark Routine Vector
139 202F 80      DEN:    DB      080H      ;Normal DENsity Flag, 0=S, 80=D
140 2030 01      GOTYPE: DB      001H      ;"GO TYPE" File Identifier
141 2031 B028      BUFPT: DW      KEYBF      ;Key BUFFER Location Pointer
142 2033 10      SCRSZ:  DB      16      ;SCREEN Size in # of Lines
143 2034 00      CONFG:  DB      000H      ;Drive Configuration Byte
144 2035
145 2035      PAGE

```



```

146 2035
147 2035
148 2035
149 2035
150 2035
151 2035
152 2035 0001 INITCT: DW 00100H ;Control Word for
153 2037 2036 RWVCTR: EQU $-1 ; Disk Initialization
154 2037
155 2037 00EA CNTLWD: DW CORDER ;Control Word Store
156 2039
157 2039 2039 DATA: EQU $
158 2039 2039 NAME1: EQU DATA+00 ;First Key Buffer Name Store
159 2039 2048 NAME2: EQU DATA+15 ;Second Key Buffer Name Store
160 2039 2049 TYPE: EQU DATA+16 ;Type File store
161 2039 2049 SEC2RW: EQU DATA+16 ;also # Sectors 2 Read/Write store
162 2039 204A SECCTR: EQU DATA+17 ;Sector Counter
163 2039 204B ERRSTR: EQU DATA+18 ;Error Flag Store
164 2039 204C BOTFLG: EQU DATA+19 ;Boot ReLoad Flag
165 2039 204D STKSAV: EQU DATA+20 ;Stack Pointer Store
166 2039 204F DCMS1: EQU DATA+22 ;DCOM Store #1
167 2039 2051 DCMS2: EQU DATA+24 ;DCOM Store #2
168 2039 2053 DEVICE: EQU DATA+26 ;List Device Store
169 2039 2054 SINDEXT: EQU DATA+27 ;Sector Index
170 2039 2056 UNIT#: EQU DATA+29 ;Unit # Store
171 2039 2057 NLSWH: EQU DATA+30 ;No List Switch
172 2039 2058 DIRSEC: EQU DATA+31 ;Directory Sector Counter
173 2039 2059 DIRCTR: EQU DATA+32 ;Directory File Counter
174 2039 205A TIMER: EQU DATA+33 ;Delay Timer Number
175 2039 205B DATA: EQU DATA+34 ;Last Data Entry + 1 Address
176 2039
177 2039

```

PAGE

```

178 2039
179 2039
180 2039 ;*****
181 2039 ; The Boot Prom reads into 2000H To 2200H *
182 2039 ; from sector 4. *
183 2039 ; *
184 2039 ; BOOT: reads into 2200H to 2A00H *
185 2039 ; from sectors 5,6,7, & 8 *
186 2039 ; *
187 2039 ; INIT: then completes DOS read-in *
188 2039 ; at 1E00H to 2000H from sector 9 *
189 2039 ; This initialization technique will *
190 2039 ; allow greatly expanded initialization *
191 2039 ; activity. *
192 2039 ; *
193 2039 ;*****
194 2039
195 2039 31001E BOOT: LXI SP,ISTACK ;Set Initial Stack
196 2039 3E59 MVI A,TRK.?
197 2039 320020 STA HEAD1 ;Reset HEAD1 Posit Counter
198 2041 3E04 MVI A,004H ;A = 4 sectors to read, 200H each
199 2043 018100 LXI B,0081H ;B = Track#, C = Density & Unit #
200 2046 110105 LXI D,((DIRSZ+1)*100H)+1 ;D=1st sec, E=Read
201 2049 210022 LXI H,DBASE+200H ;HL = Load Address, 2200H
202 204C CD5B20 CALL DSKFNC ;Go read 2200H to 2A00H
203 204F C23920 JNZ BCOT ;If error, go try again
204 2052 2100E9 LXI H,PBASE
205 2055 220B20 SHLD BOOTV+1 ;Reset Boot Vector to PROM address
206 2058 C3C029 JMP INIT ;Go initialize & complete DOS read-
207 205B
208 205B PAGE

```

```

209 205B
210 205B      205B          ORG      DATA
211 205B
212 205B      ;*****
213 205B      ;      DISK FUNCTION ROUTINE      *
214 205B      ;      -----      *
215 205B      ;      A = Transfer Length in Sectors      *
216 205B      ;      B = Track #      *
217 205B      ;      C = Density/Side/Unit# Select      *
218 205B      ;      Bit-7=Double, Bit-6=Side B Select      *
219 205B      ;      D = Sector # (limits 0 thru 9)      *
220 205B      ;      E = 0 to Write, 1 to Read, 2 to Verify      *
221 205B      ;      -1 to Init Single, -2 to Init Double      *
222 205B      ;      HL= Transfer Memory Address      *
223 205B      ;      *
224 205B      ;      On Return;      *
225 205B      ;      -----      *
226 205B      ;      Z-Set = Transfer was OK!      *
227 205B      ;*****
228 205B
229 205B F5      DSKFNC: PUSE      PSW
230 205C E5      PUSH      H
231 205D 79      MOV      A,C
232 205E E6C0    ANI      0C0H
233 2060 67      MOV      H,A
234 2061 3E14    MVI      A,014H
235 2063 B8      CMP      B
236 2064 1F      RAR
237 2065 1F      RAR
238 2066 1F      RAR
239 2067 E620    ANI      020H
240 2069 B4      ORA      H
241 206A 67      MOV      H,A
242 206B 79      MOV      A,C
243 206C E63F    ANI      3FH
244 206E 4F      MOV      C,A
245 206F FE03    CPI      003H
246 2071 DA7720  JC      DSKF1
247 2074 17      RAL
248 2075 E60C    ANI      00CH
249 2077
250 2077 B4      DSKF1: ORA      H
251 2078 323720  STA      CNTLWD
252 207B F1      POP      H
253 207C E5      PUSH      H
254 207D 7B      MOV      A,E
255 207E 323620  STA      RWVCTR ;Set Read/Write/Verify Control Byte
256 2081 B7      ORA      A
257 2082 F28720  JP      W RTP?
258 2085 1E00    MVI      E,000H
259 2087
260 2087 D5      W RTP?: PUSH      D
261 2088 C5      PUSH      B
262 2089 CDA221  CALL     MOTON
263 208C C1      POP      B

```

264	208D	D1	POP	D
265	208E	7B	MOV	A,E
266	208F	B7	ORA	A
267	2090	C29D20	JNZ	HOME?
268	2093	3A20EB	LDA	STATB
269	2096	E602	ANI	B,WP
270	2098	3E06	MVI	A,006H ;Flag, "WRITE PROTECTED"
271	209A	C21E21	JNZ	EEXT6
272	209D			
273	209D	D5	HOME?: PUSH	D
274	209E	C5	PUSH	B
275	209F	21FF1F	LXI	H,HEAD1-1
276	20A2	0600	MVI	B,000E
277	20A4	09	DAD	B
278	20A5	7E	MOV	A,M
279	20A6	EE59	XRI	TRK.?
280	20A8	E5	PUSH	H
281	20A9	CC2221	CZ	UPDATE ;Go Home Head to Track 0
282	20AC	E1	POP	H
283	20AD	F1	POP	PSW
284	20AE	CD2221	CALL	UPDATE
285	20B1	2A3720	LHLD	CNTLWD
286	20B4	7E	MOV	A,M
287	20B5	7D	MOV	A,L
288	20B6	E680	ANI	080H
289	20B8	EE80	XRI	080H
290	20BA	4F	MOV	C,A
291	20BB	EE80	XRI	080H
292	20BD	1F	RAR	
293	20BE	1F	RAR	
294	20BF	1F	RAR	
295	20C0	C60F	ADI	00FH
296	20C2	47	MOV	B,A
297	20C3	D1	POP	D
298	20C4	E1	POP	H
299	20C5	F1	POP	PSW
300	20C6	3C	INR	A
301	20C7	F5	PUSH	PSW
302	20C8	15	DCR	D
303	20C9	D5	PUSH	D
304	20CA	C5	PUSH	B
305	20CB			
306	20CB	C1	NXTSEC: POP	B
307	20CC	D1	POP	D
308	20CD	14	INR	D
309	20CE	F1	POP	PSW
310	20CF	3D	DCR	A
311	20D0	C8	RZ	
312	20D1	F5	PUSH	PSW
313	20D2	D5	PUSH	D
314	20D3	C5	PUSH	B
315	20D4			
316	20D4	D5	SEEK: PUSH	D
317	20D5	CD7121	CALL	SEEK1
318	20D8	D1	POP	D

319	20D9	3A35EB	LDA	STATC+MOTRON ;Gate Status-B
320	20DC	E60F	ANI	00FH ;Mask down to Sector Bits
321	20DE	BA	CMP	D ;Check if On Sector Yet?
322	20DF	C2D420	JNZ	SEEK
323	20E2	1D	DCR	E
324	20E3	FA0C22	JM	WRITE
325	20E6	C2E921	JNZ	VERIFY
326	20E9			
327	20E9		PAGE	

```

328 20E9
329 20F9
330 20E9
331 20E9
332 20E9 CD0921
333 20EC 0600
334 20EE 1A
335 20EF 77
336 20F0 A8
337 20F1 07
338 20F2 47
339 20F3 23
340 20F4 37
341 20F5 1A
342 20F6 77
343 20F7 A8
344 20F8 07
345 20F9 47
346 20FA 23
347 20FB 0D
348 20FC C2EE20
349 20FF 1A
350 2100 A8
351 2101 CACB20
352 2104 3E02
353 2106 C31D21
354 2109
355 2109 068C
356 210B 1140EB
357 210E CD8821
358 2111 3A10EB
359 2114 0F
360 2115 D8
361 2116 05
362 2117 C21121
363 211A E1
364 211B 3E01
365 211D E1
366 211E E1
367 211F C1
368 2120 B7
369 2121 C9
370 2122
371 2122

;*****
READS:  CALL  GETSYNC ;Get Sync and Density
RSECT:  MVI   B,000H
RBYTE:  LDAX  D
        MOV   M,A
        XRA   B
        RLC
        MOV   B,A
        INX   H
        STC
        LDAX  D
        MOV   M,A
        XRA   B
        RLC
        MOV   B,A
        INX   H
        DCR   C
        JNZ   RBYTE
RBYTX:  LDAX  D
        XRA   B
        JZ    NXTSEC
CRCERR: MVI   A,002H ;Flag, "CRC COMPARE ERROR"
        JMP   EEXT2

GETSYNC: MVI   B,140D ;Sync Search Limit (16 Byte times)
        LXI   D,READD
        CALL  DD?
SYNC?:  LDA   STATA
        RRC
        RC
        DCR   B
        JNZ   SYNC?
        POP   H ;Clear Stack RET Address
EEXT1:  MVI   A,001H ;Flag, "SYNC BYTE NOT FOUND"
EEXT2:  POP   H
EEXT6:  POP   H
        POP   B
        ORA   A
        RET

PAGE

```

```

372 2122
373 2122
374 2122
375 2122 57      UPDATE: MOV      D,A
376 2123 96      SUB      M
377 2124 72      MOV      M,D
378 2125 C8      RZ
379 2126 2120EA   LXI      H,CORDER+ORDDP
380 2129 4F      MOV      C,A
381 212A F23721   JP      UPDT1
382 212D 2F      CMA
383 212E 3C      INR      A
384 212F 4F      MOV      C,A
385 2130 3A20EB   LDA      STATB
386 2133 E601     ANI      E.T0
387 2135 C0      RNZ
388 2136 6F      MOV      L,A      ;00H to 'L'
389 2137 3A0620   UPDT1:  LDA      UNIT
390 213A FE03     CPI      003H
391 213C DA4221   JC      UPDT2
392 213F 17      RAL
393 2140 E60C     ANI      00CE
394 2142 B5      UPDT2:  ORA      L
395 2143 6F      MOV      L,A
396 2144 56      MOV      D,M
397 2145 EE10     XRI      010H
398 2147 6F      MOV      L,A
399 2148 56      MOV      D,M
400 2149 EE10     XRI      010H
401 214B 6F      MOV      L,A
402 214C 56      MOV      D,M
403 214D 3A5A20   LDA      TIMER      ;Get Timer Number
404 2150 1638     UPDT3:  MVI      D,38H
405 2152 15      UPDT4:  DCR      D
406 2153 C25221   JNZ      UPDT4
407 2156 3D      DCR      A
408 2157 C25021   JNZ      UPDT3
409 215A 3A3420   LDA      CONFG
410 215D A5      ANA      L
411 215E E60F     ANI      00FH
412 2160 1602     MVI      D,002H
413 2162 CC7321   CZ      SEEK.D
414 2165 3A25EB   LDA      STATB+MOTRON
415 2168 E601     ANI      001H
416 216A C27121   JNZ      SEEK1
417 216D 0D      DCR      C
418 216E C23721   JNZ      UPDT1
419 2171 1601     SEEK1:  MVI      D,001H
420 2173 CD0720   SEEK.D: CALL    OFTEN
421 2176 3A11EB   LDA      STATA+RSTSF
422 2179 3A10EB   SEK2:   LDA      STATA
423 217C B7      ORA      A
424 217D F27921   JP      SEK2
425 2180 15      DCR      D
426 2181 3A11EB   LDA      STATA+RSTSF

```

427 2184 027321
428 2187 09
429 2188
430 2188

JNZ SEEK.D
RET

PAGE


```

431 2188
432 2182
433 2188
434 2188 3A10EB DD?: LDA STATA
435 218B E604 ANI A.RE
436 218D CA8821 JZ DD?
437 2190 3E09 MVI A,09H ;Delay Timer
438 2192 3D DD?: DCR A
439 2193 C29221 JNZ DD?: ;Loop for 45 us @ 4MHZ
440 2196 3A10EB LDA STATA
441 2199 17 RAL
442 219A 17 RAL
443 219B A9 XRA C
444 219C 3E05 MVI A,005H ;Flag, "DENSITY MIS-MATCH"
445 219E F8 RM
446 219F C3E421 JMP INDXX
447 21A2
448 21A2 CD7121 MOTON: CALL SEEK1
449 21A5 E610 ANI 010H
450 21A7 3A15EB LDA STATA+MOTRON
451 21AA C2BE21 JNZ MOTO2
452 21AD 1630 MVI D,'0'
453 21AF 3A3720 LDA CNTLWD
454 21B2 B7 ORA A
455 21B3 F2B821 JP MOTO1
456 21B6 1617 MVI D,017H
457 21B8 CD7321 MOTO1: CALL SEEK.D
458 21BB C3C321 JMP MOTO3
459 21BE
460 21BE 3A0620 MOTO2: LDA UNIT
461 21C1 B9 CMP C
462 21C2 C8 RZ
463 21C3 79 MOTO3: MOV A,C
464 21C4 320620 STA UNIT
465 21C7 16EA MVI D,CORDER/100H
466 21C9 3A3720 LDA CNTLWD
467 21CC 5F MOV E,A
468 21CD 1A LDAX D
469 21CE 1602 MVI D,002H
470 21D0 CD7321 CALL SEEK.D
471 21D3 060C MVI B,00CH
472 21D5 CD7121 INDXX?: CALL SEEK1
473 21D8 3A10EB LDA STATA
474 21DB E640 ANI IX
475 21DD C0 RNZ
476 21DE 05 DCR B
477 21DF C2D521 JNZ INDXX?
478 21E2 3E04 MVI A,004H ;Flag, "NO INDEX PULSE"
479 21E4 C1 INDXX: POP B
480 21E5 C1 POP B
481 21E6 C31D21 JMP EEXT2
482 21E9
483 21E9 PAGE

```

```
484 21E9
485 21E9 ;*****
486 21E9
487 21E9 CD0921 VERIFY: CALL GETSYNC ;Go get Sync and Density
488 21EC 0600 VRFY1: MVI B,000H
489 21EE 1A VRFY2: LDAX D
490 21EF BE CMP M
491 21F0 C20722 JNZ VRFYX
492 21F3 A8 XRA B
493 21F4 07 RLC
494 21F5 47 MOV B,A
495 21F6 23 INX H
496 21F7 1A LDAX D
497 21F8 BE CMP M
498 21F9 C20722 JNZ VRFYX
499 21FC A8 XRA B
500 21FD 07 RLC
501 21FE 47 MOV B,A
502 21FF 23 INX H
503 2200 0D DCR C
504 2201 C2EE21 JNZ VRFY2
505 2204 C3FF20 JMP RBYTX
506 2207
507 2207 3E03 VRFYX: MVI A,003H ;Flag, "VERIFY COMPARE ERROR"
508 2209 C31D21 JMP EEXT2
509 220C
510 220C PAGE
```

```

511 220C
512 220C
513 220C ;*****
514 220C
515 220C 3A16EB WRITE: LDA STATA+BWRITE
516 220F 3A15EB WRIT1: LDA STATA+MOTRON
517 2212 E608 ANI A,WI
518 2214 C20F22 JNZ WRIT1
519 2217 3A00E9 WRIT2: LDA CWRITE+00H ;Write ZERO Byte
520 221A 7F MOV A,A
521 221B 16E9 MVI D,0E9H
522 221D 05 DCR B
523 221E C21722 JNZ WRIT2 ;Loop Until 15 Leading ZEROS Done!
524 2221 3AFBE9 LDA CWRITE+SYNC
525 2224 3A3720 LDA CNTLWD
526 2227 07 RLC
527 2228 D22E22 JNC WRIT3
528 222B 3AFBE9 WRIT3: LDA CWRITE+SYNC ;Write SYNC Byte (0FBH)
529 222E 49 MOV C,C ;Delay a little
530 222F B7 ORA A ; and a bit more!
531 2230 7E WRIT4: MOV A,M ;Get byte to write
532 2231 5F MOV E,A ;Put it in 'E' for output
533 2232 A8 XRA B ;Calculate
534 2233 07 RLC ; the CRC
535 2234 47 MOV B,A ; and accumulate it in 'B'
536 2235 1A LDAX D ;Write the byte
537 2236 23 INX H ;Point to next byte
538 2237 7E MOV A,M ;Do
539 2238 5F MOV E,A ; another
540 2239 A8 XRA B ; byte
541 223A 07 RLC ; in this loop
542 223B 47 MOV B,A ; so we can write 200H
543 223C 1A LDAX D ; bytes while using only
544 223D 23 INX H ; a 100H counter, i.e.
545 223E 0D DCR C ;Decrement the counter
546 223F C23022 JNZ WRIT4 ;Loop until sector is complete
547 2242 58 MOV E,B ;Put CRC byte in 'E' for output
548 2243 03 INX B ;Bump it one for proper 2's complim
549 2244 1A LDAX D ;Output the CRC byte
550 2245 EB XCHG ;Decrement Memory Pointer back to t
551 2246 2A3520 LHLD INITCT ; beginning of the Dos Buffer if
552 2249 19 DAD D ; Disk Initialization Mode
553 224A C3CB20 JMP NXTSEC ;Go do next sector
554 224D
555 224D PAGE

```

```
556 224D
557 224D ;*****
558 224D
559 224D 114928 HDERR: LXI D,ERRTBL-1
560 2250 83 ADD E
561 2251 5F MOV E,A
562 2252 1A LDAX D
563 2253 5F MOV E,A ;MESSAGE ADDR IS IN DE
564 2254 CDBE25 CALL MESSAGE ;Output Error Type
565 2257 CD9B25 CALL OUTSP ;Output following Space
566 225A 3E44 MVI A,'D' ;'D'rive #
567 225C CDB725 CALL CONOUT ;Output to Consol
568 225F 79 MOV A,C ;Get Drive #
569 2260 C630 ADI '0' ;Add ASCII Bias
570 2262 CDB725 CALL CONOUT ;Output to Consol
571 2265 CD9B25 CALL OUTSP
572 2268 3E53 MVI A,'S' ;'S'ector #
573 226A CDB725 CALL CONOUT ;Output to Consol
574 226D CD6725 CALL OUTDEC ;Output Sector# in H,L in Decimal
575 2270 C33127 JMP CR&DOS ;Output CR & return to DOS
576 2273
577 2273 PAGE
```

```

578 2273
579 2273
580 2273
581 2273 CD7F22
582 2276 DA2C20
583 2279 CDEE23
584 227C C3CD22
585 227F
586 227F
587 227F
588 227F 3E01
589 2281 325620
590 2284 1604
591 2286 214920
592 2289 CDBA22
593 228C 54
594 228D 5D
595 228E CD3B26
596 2291 37
597 2292 C8
598 2293 FE20
599 2295 CAEE22
600 2298 FE2C
601 229A 37
602 229B 0608
603 229D C8
604 229E FE20
605 22A0 C8
606 22A1 FE2C
607 22A3 CAB122
608 22A6 05
609 22A7 37
610 22A8 F8
611 22A9 77
612 22AA 23
613 22AB CD3B26
614 22AE C39D22
615 22B1
616 22B1 CDC925
617 22B4 CD2F26
618 22B7 C8
619 22B8 37
620 22F9 C9
621 22BA
622 22BA 2B
623 22BB 3620
624 22BD 2B
625 22BE 3620
626 22C0 15
627 22C1 C2BA22
628 22C4 C9
629 22C5
630 22C5

;*****
FFILE:  CALL      MOV81
        JC        QMARK
        CALL      NLFSR
        JMP       DWR2T

;*****
MOV81:  MVI        A,001H
        STA        UNIT#
MOV8:   MVI        D,004H
        LXI        H,NAME2+1
        CALL       CLRSPC ;Go clear 8 char buffer
        MOV        D,H
        MOV        E,L
MOV83:  CALL       GETCH
        STC
        RZ
        CPI        SPACE
        JZ         MOV83
        CPI        COMMA
        STC
        MVI        B,000EH
MOV84:  RZ
        CPI        SPACE
        RZ
        CPI        COMMA
        JZ         MOV8X
        DCR        B
        STC
        RM
        MOV        M,A
        INX        H
        CALL       GETCH
        JMP        MOV84
MOV8X:  CALL       UNT2A
        CALL       GET.CH
        RZ
        STC
        RET
CLRSPC: DCX        H
        MVI        M,SPACE
        DCX        H
        MVI        M,SPACE
        DCR        D
        JNZ        CLRSPC
        RET
PAGE

```

```

631 22C5
632 22C5 ;*****
633 22C5
634 22C5 0600 DWRIT: MVI B,000E ;Set to Write
635 22C7 CDD122 CALL SET.UP
636 22CA CD2220 DWR0T: CALL DCOMV
637 22CD DA2C20 DWR2T: JC QMARK
638 22D0 C9 RET
639 22D1
640 22D1 212F20 SET.UP LXI E,DEN
641 22D4 3A5620 LDA UNIT#
642 22D7 B6 ORA M
643 22D8 4F MOV C,A
644 22D9 3A5820 LDA DIRSEC
645 22DC 6F MOV L,A
646 22DD AF XRA A
647 22DE 67 MOV E,A
648 22DF 3C INR A
649 22E0 11001D LXI D,DOSBF
650 22E3 C9 RET
651 22E4
652 22E4 ;*****
653 22E4
654 22E4 7B VERCHK: MOV A,E
655 22E5 B7 ORA A
656 22E6 CAEE22 JZ VERC2 ; Jump if we just Wrote
657 22E9
658 22E9 D1 VERC1: POP D
659 22EA D1 POP D
660 22EB C37121 JMP SEEK1
661 22EE
662 22EE 3A2B20 VERC2: LDA RWCHK
663 22F1 B7 ORA A
664 22F2 CAE922 JZ VERC1 ; Jump if NO Read After Write
665 22F5 E1 POP H
666 22F6 F1 POP PSW
667 22F7 3A4920 LDA SEC2RW
668 22FA 1E02 MVI E,002H ; Set to Verify
669 22FC
670 22FC 324920 RDWR1: STA SEC2RW
671 22FF 3E0A MVI A,NSECTR
672 2301 F5 RDWR2: PUSE PSW
673 2302 E5 PUSH H
674 2303 D5 PUSH D
675 2304 C5 PUSH B
676 2305 3E45 MVI A,(2*NTRKS)-1
677 2307 90 SUB B
678 2308 FE23 CPI NTRKS
679 230A D21223 JNC RDWR3 ; Jump if Sector is on Front Side
680 230D 47 MOV B,A
681 230E 3E40 MVI A,40H
682 2310 B1 ORA C
683 2311 4F MOV C,A ; Select Backside of Disk
684 2312 3A4920 RDWR3: LDA SEC2RW
685 2315 CD5B20 CALL DSKFNC

```

686 2318 324B20	STA	ERRSTR
687 231B 78	MOV	A,B
688 231C 324A20	STA	SECCTR
689 231F C1	POP	B
690 2320 D1	POP	D
691 2321 CAE422	JZ	VERCHK
692 2324 E1	POP	H
693 2325 3A4B20	LDA	ERRSTR
694 2328 FE04	CPI	004H
695 232A D23223	JNC	RDWRX
696 232D F1	POP	PSW
697 232E 3D	DCR	A
698 232F C20123	JNZ	RDWR2
699 2332 2A4D20	RDWRX: LHLD	STKSAV
700 2335 F9	SPHL	
701 2336 3A4920	LDA	SEC2RW
702 2339 214A20	LXI	E,SECCTR
703 233C 96	SUB	M
704 233D 82	ADD	D
705 233E 68	MOV	L,B
706 233F 2600	MVI	H,000H
707 2341 29	DAD	H
708 2342 54	MOV	D,H
709 2343 5D	MOV	E,L
710 2344 29	DAD	H
711 2345 29	DAD	H
712 2346 19	DAD	D
713 2347 85	ADD	L
714 2348 6F	MOV	L,A
715 2349 7C	MOV	A,H
716 234A CE00	ACI	000H
717 234C 67	MOV	H,A
718 234D 3A4B20	LDA	ERRSTR
719 2350 C31920	JMP	HDERV
720 2353		
721 2353	PAGE	

722 2353

723 2353

724 2353

725 2353

726 2353

727 2353

728 2353

729 2353

730 2353

731 2353

732 2353

733 2353

734 2353

735 2353

736 2353

737 2353

738 2353

739 2353

740 2353 E5

741 2354 210400

742 2357 39

743 2358 224D20

744 235B E1

745 235C D5

746 235D C5

747 235E F5

748 235F E5

749 2360 B7

750 2361 CA8E23

751 2364 F5

752 2365 EB

753 2366 CDEB25

754 2369 F1

755 236A 19

756 236B DA8E23

757 236E 5F

758 236F 1600

759 2371 1B

760 2372 19

761 2373 DA8E23

762 2376 79

763 2377 E67F

764 2379 FE01

765 237B DA8E23

766 237E FE05

767 2380 D28E23

768 2383 78

769 2384 C602

770 2386 FA8E23

771 2389 FE05

772 238B DA9423

773 238E

774 238E E1

775 238F F1

776 2390 C1

```

;*****
;
;                                DISK COMMAND ROUTINE
;-----
; On Entry;
;   A = Number of Sectors to Transfer
;   B = Command (0=Write, 1=Read, 2=Verify
;           -1=Single Init., -2=Double Init.))
;   C = Unit#, Bit7=Density; Double=1, Single=0
;   DE = Starting Address of Memory Block
;   HL = Starting Disk Sector Address
;
; On Return;
;   'CARRY SET' - Means that Arguments were illegal
;*****

```

```

DCOM:  PUSH    H
        LXI     H,0004H
        DAD     SP
        SHLD    STKSAV
        POP     H
        PUSH    D
        PUSH    E
        PUSH    PSW
        PUSH    H
        ORA     A
        JZ      PROBX    ;If NO Sectors to transfer
        PUSH    PSW
        XCHG
        CALL    SECLIM   ;Get - (Sector Max Limit to D,E)
        POP     PSW
        DAD     D
        JC      PROBX    ;If File's Disk Addr is too large
        MOV     E,A
        MVI     D,000EH
        DCX     D
        DAD     D
        JC      PROBX    ;If File's End Addr is too large
        MOV     A,C
        ANI     07FH
        CPI     001H
        JC      PROBX    ;If Unit# is Less Than 1
        CPI     005H
        JNC     PROBX    ;If Unit# is More Than 4
        MOV     A,B
        ADI     002H
        JM      PROBX    ;If Control Byte is Less Than (-2)
        CPI     005H
        JC      DCOM2    ;If No Errors, Control Byte < +2

PROBX:  POP     H
        POP     PSW
        POP     B

```


777	2391	D1	POP	D
778	2392	37	STC	
779	2393	C9	RET	
780	2394			
781	2394	E1	DCOM2: POP	H
782	2395	F1	POP	PSW
783	2396	11F6FF	LXI	D,-NSECTR
784	2399	06FF	MVI	B,-1
785	239B	04	DCOM3: INR	B
786	239C	19	DAD	D
787	239D	DA9B23	JC	DCOM3
788	23A0			
789	23A0	85	DCOM5: ADD	L
790	23A1	D2D023	JNC	DCOM6
791	23A4	CAD023	JZ	DCOM6
792	23A7	67	MOV	H,A
793	23A8	7D	MOV	A,L
794	23A9	C60A	ADI	NSECTR
795	23AB	57	MOV	D,A
796	23AC	D60B	SUI	NSECTR+1
797	23AE	2F	CMA	
798	23AF	324A20	STA	SECCTR
799	23B2	68	MOV	L,B
800	23B3	224F20	SHLD	DCMS1
801	23B6	E1	POP	H
802	23B7	225120	SHLD	DCMS2
803	23BA	4D	MOV	C,L
804	23BB	5C	MOV	E,H
805	23BC	E1	POP	H
806	23BD	CDFC22	CALL	RDWR1
807	23C0	E5	PUSH	H
808	23C1	2A5120	LHLD	DCMS2
809	23C4	E5	PUSH	H
810	23C5	2A4F20	LHLD	DCMS1
811	23C8	7C	MOV	A,H
812	23C9	45	MOV	B,L
813	23CA	04	INR	B
814	23CB	2EF6	MVI	L,-NSECTR
815	23CD	C3A023	JMP	DCOM5
816	23D0			
817	23D0	95	DCOM6: SUB	L
818	23D1	F5	PUSH	PSW
819	23D2	7D	MOV	A,L
820	23D3	C60A	ADI	NSECTR
821	23D5	57	MOV	D,A
822	23D6	F1	POP	PSW
823	23D7	E1	POP	H
824	23D8	5C	MOV	E,H
825	23D9	4D	MOV	C,L
826	23DA	E1	POP	H
827	23DB	C3FC22	JMP	RDWR1
828	23DE			
829	23DE		PAGE	

```

830 23DE
831 23DE
832 23DE
833 23DE
834 23DE 223120      DLOOK:  SHLD      BUFPT
835 23E1 325620      STA      UNIT#
836 23E4 7E          MOV      A,M
837 23E5 FE20        CPI      SPACE
838 23E7 113920      LXI      D,NAME1
839 23EA C48422      CNZ      MOV8
840 23ED D8          RC
841 23EE
842 23EE AF          NLFSR:  XRA      A
843 23EF 325720      DFSR:   STA      NLSWE
844 23F2 3E80        MVI      A,080E
845 23F4 322F20      STA      DEN
846 23F7 210400      LXI      H,DIRSZ ;Set Director Limit
847 23FA 225420      SHLD     SINDE
848 23FD 010004      LXI      B,DIRSZ*100H ;B=#Sec in Dir. / C=counter
849 2400 EB          DFSR1:  XCHG
850 2401 E5          PUSH     H
851 2402 C5          PUSH     B
852 2403 CD1624      CALL     RDSEC
853 2406 C1          POP      B
854 2407 D1          POP      D
855 2408 3A5620      LDA      UNIT#
856 240B C8          RZ
857 240C 0C          INR      C
858 240D 05          DCR      B
859 240E C20024      JNZ      DFSR1 ;Do all Directory Sectors
860 2411 37          STC
861 2412 2A5420      LHLD     SINDE
862 2415 C9          RET
863 2416
864 2416 E5          RDSEC:   PUSH     H
865 2417 79          MOV      A,C
866 2418 325820      STA      DIRSEC
867 241B 2A1A20      RDS1:   LHLD     HDERV+1
868 241E E5          PUSH     H
869 241F 219B24      LXI      H,HDERPC
870 2422 221A20      SHLD     EDERV+1
871 2425 0601        MVI      B,001H ;Set to Read
872 2427 CDD122      CALL     SET.UP ;Go Set-Up to Read Sector
873 242A CD2220      CALL     DCOMV ;Go Read Sector
874 242D E1          POP      E
875 242E 221A20      SHLD     HDERV+1
876 2431 0620        MVI      B,020H
877 2433 3A2F20      LDA      DEN
878 2436 B7          ORA      A
879 2437 FA3C24      JM       RDS2
880 243A 0610        MVI      B,010H
881 243C 21001D      RDS2:   LXI      E,DOSBF
882 243F D1          POP      D
883 2440 D5          RDS3:   PUSH     D
884 2441 E5          PUSH     E

```

885	2442	0E09		MVI	C,008H	
886	2444	3A5720		LDA	NLSWH	
887	2447	B7		ORA	A	
888	2448	C2A024		JNZ	PRTL1	
889	244B					
890	244B	1A	RDS4:	LDAX	D	
891	244C	BE		CMP	M	
892	244D	C25A24		JNZ	BLANK?	
893	2450	13		INX	D	
894	2451	23		INX	H	
895	2452	0D		DCR	C	
896	2453	C24B24		JNZ	RDS4	
897	2456	C1		POP	B	
898	2457	C1		POP	B	
899	2458	AF		XRA	A	
900	2459	C9		RET		
901	245A					
902	245A	E1	BLANK?:	POP	H	
903	245B	E5		PUSH	H	
904	245C	7E		MOV	A,M	
905	245D	FE20		CPI	SPACE	
906	245F	CA7E24		JZ	LNLST	
907	2462	C5		PUSH	B	
908	2463	110800		LXI	D,0008H	
909	2466	19		DAD	D	
910	2467	CDB225		CALL	M2DE	
911	246A	4E		MOV	C,M	
912	246B	23		INX	H	
913	246C	46		MOV	B,M	
914	246D	EB		XCHG		
915	246E	09		DAD	B	
916	246F	EB		XCHG		
917	2470	215420		LXI	H,SINDEX	
918	2473	CD4B26		CALL	CMPHD	
919	2476	C1		POP	B	
920	2477	DA7E24		JC	LNLST	
921	247A	EB		XCHG		
922	247B	225420		SHLD	SINDEX	
923	247E	E1	LNLST:	POP	H	
924	247F	111000		LXI	D,0010H	
925	2482	19		DAD	D	
926	2483	D1		POP	D	
927	2484	05		DCR	B	
928	2485	C24024		JNZ	RDS3	
929	2488	AF		XRA	A	
930	2489	3C		INR	A	
931	248A	C9		RET		
932	248B					
933	248B	FE05	EDERPC:	CPI	005H	;Check if Density Mis-Match?
934	248D	E3		XTHL		
935	248E	221A20		SHLD	EDERV+1	;Put EDERR address back in EDER
936	2491	E1		POP	H	
937	2492	C21920		JNZ	EDERV	
938	2495	3A2F20		LDA	DEN	
939	2498	EE80		XRI	080H	

940	249A	322F20	STA	DEN	
941	249D	C31B24	JMP	RDS1	
942	24A0				
943	24A0	7E	PRTL1:	MOV	A,M
944	24A1	FE20		CPI	SPACE
945	24A3	CA7E24		JZ	LNLST
946	24A6	C5		PUSH	B
947	24A7	3A5320		LDA	DEVICE
948	24AA	B7		ORA	A
949	24AB	C2C324		JNZ	PRTL3
950	24AE	3A5920		LDA	DIRCTR
951	24B1	B7		ORA	A
952	24B2	CAC324		JZ	PRTL3
953	24B5	3D		DCR	A
954	24B6	C2C024		JNZ	PRTL2
955	24B9	CD2525		CALL	POUSE
956	24BC	3A3320		LDA	SCRSZ
957	24BF	3D		DCR	A
958	24C0	325920	PRTL2:	STA	DIRCTR
959	24C3	46	PRTL3:	MOV	B,M
960	24C4	CD9D25		CALL	OUTBR
961	24C7	23		INX	H
962	24C9	0D		DCR	C
963	24C9	C2C324		JNZ	PRTL3
964	24CC	CD9B25		CALL	OUTSP
965	24CF	CDB225		CALL	M2DE
966	24D2	E5		PUSH	H
967	24D3	EB		XCHG	
968	24D4	CD9825		CALL	OUTDGC
969	24D7	E1		POP	H
970	24D8	CDB225		CALL	M2DE
971	24DB	E5		PUSH	H
972	24DC	7E		MOV	A,M
973	24DD	F5		POUSE	PSW
974	24DE	EB		XCHG	
975	24DF	B7		ORA	A
976	24E0	F2E424		JP	PRTL5
977	24E3	29		DAD	H
978	24E4	CD9825	PRTL5:	CALL	OUTDGC
979	24E7	F1		POP	PSW
980	24E8	F5		PUSH	PSW
981	24E9	E680		ANI	080H
982	24EB	0653		MVI	B,'S'
983	24ED	CAF224		JZ	PRTL6
984	24F0	0644		MVI	B,'D'
985	24F2	CD9D25	PRTL6:	CALL	OUTBR
986	24F5	CD9B25		CALL	OUTSP
987	24F8	F1		POP	PSW
988	24F9	E67F		ANI	07FE
989	24FB	6F		MOV	L,A
990	24FC	2600		MVI	H,000H
991	24FE	CD6725		CALL	OUTDEC
992	2501	E1		POP	H
993	2502	3A3020		LDA	GOTYPE
994	2505	47		MOV	B,A

;Get TYPE FILE from Directory Entry

;Get TYPE back

;Test if DOUBLE DENSITY type file?

;S'ingle Density ASCII Print-out

;D'ouble Density ASCII Print-out

;Get TYPE back

;Stripoff Double Density Bit

```

995 2506 7E      MOV      A,M
996 2507 E67F     ANI      07FH
997 2509 B8       CMP      B      ;Check if "GO TYPE" File?
998 250A C21825   JNZ      PRTL4
999 250D CD9B25   CALL     OUTSP
1000 2510 23      INX      H
1001 2511 CDB225   CALL     M2DE
1002 2514 EB      XCHG
1003 2515 CD5325   CALL     OUTHEX
1004 2518 CDA325   PRTL4:  CALL     OUTCR      ;Output a RETURN
1005 251B CD1620   CALL     CNTCV
1006 251E C1      POP      B
1007 251F CA2C20   JZ       QMARK
1008 2522 C37E24   JMP      LNLST
1009 2525
1010 2525 113725   PGUSE:  LXI      D,HITKEY
1011 2528 CDBE25   CALL     MESSAGE
1012 252B AF      XRA      A
1013 252C CD1020   CALL     CINV
1014 252F FE03     CPI      CTLGC
1015 2531 CA2C20   JZ       QMARK
1016 2534 C3A325   JMP      OUTCR
1017 2537
1018 2537 2A2A2A2A HITKEY:  DB      '*****<< HIT KEY >>*****',EOT
      253B 2A2A2A3C
      253F 3C204849
      2543 54204B45
      2547 59203E3E
      254B 2A2A2A2A
      254F 2A2A2AFF
1019 2553
1020 2553

```

PAGE

```

1021 2553
1022 2553
1023 2553
1024 2553 1100F0 OUTHEX: LXI D,-1000H
1025 2556 0E30 MVI C,'0'
1026 2558 CD7925 CALL CONV4
1027 255B 1100FF LXI D,-100H
1028 255E CD7925 CALL CONV4
1029 2561 11F0FF LXI D,-010H
1030 2564 C37225 JMP CONV3
1031 2567
1032 2567 119CFF OUTDEC: LXI D,-100D
1033 256A 0E30 MVI C,'0'
1034 256C CD7925 CALL CONV4
1035 256F 11F6FF LXI D,-10D
1036 2572 CD7925 CONV3: CALL CONV4
1037 2575 11FFFF LXI D,-1
1038 2578 4B MOV C,E
1039 2579 062F CONV4: MVI B,'0'-1
1040 257B 22B028 CONV5: SHLD KEYBF ;Temp store in KEYBF, wipes out CMD
1041 257E 04 INR B
1042 257F 19 DAD D
1043 2580 DA7B25 JC CONV5
1044 2583 2AB028 LHLD KEYBF
1045 2586 78 MOV A,B
1046 2587 B9 CMP C
1047 2588 CA9B25 JZ OUTSP
1048 258B 0EFF MVI C,-1
1049 258D FE3A CPI '9'+1
1050 258F DA9D25 JC OUTBR
1051 2592 C607 ADI 007H
1052 2594 47 MOV B,A
1053 2595 C39D25 JMP OUTBR
1054 2598
1055 2598 PAGE

```

```

1056 2598
1057 2598
1058 2598
1059 2598
1060 2598 CD6725 OUTDQC: CALL OUTDEC ;Output Decimal & Space
1061 259B 0620 OUTSP: MVI B,SPACE ;Output Space
1062 259D 3A5320 OUTBR: LDA DEVICE ;Output 'B' Register
1063 25A0 C30D20 JMP COUTV
1064 25A3
1065 25A3 060D OUTCR: MVI B,CRO
1066 25A5 CD9D25 CALL OUTBR
1067 25A8 060A MVI B,LFO
1068 25AA C39D25 JMP OUTBR
1069 25AD
1070 25AD 73 DE2M: MOV M,E
1071 25AE 23 INX H
1072 25AF 72 MOV M,D
1073 25B0 23 INX E
1074 25B1 C9 RET
1075 25B2
1076 25B2 5E M2DE: MOV E,M
1077 25B3 23 INX H
1078 25B4 56 MOV D,M
1079 25B5 23 INX H
1080 25B6 C9 RET
1081 25B7
1082 25B7 E67F CONOUT: ANI 7FE ;Prevent reverse video
1083 25B9 47 MOV B,A
1084 25BA AF XRA A
1085 25BB C30D20 JMP COUTV
1086 25BE
1087 25BE 1A MESSAGE: LDAX D
1088 25FF FEFF CPI EOT
1089 25C1 C8 RZ
1090 25C2 CDB725 CALL CONOUT
1091 25C5 13 INX D
1092 25C6 C3BE25 JMP MESSAGE
1093 25C9
1094 25C9 PAGE

```

```

1095 25C9
1096 25C9
1097 25C9
1098 25C9
1099 25C9 0E01
1100 25CB CD3F26
1101 25CE CAE625
1102 25D1 CD3526
1103 25D4 CAC925
1104 25D7 D630
1105 25D9 DA2C20
1106 25DC
1107 25DC FE05
1108 25DE D22C20
1109 25E1 B7
1110 25E2 CA2C20
1111 25E5 4F
1112 25E6 79
1113 25E7 325620
1114 25EA C9
1115 25EB
1116 25EB
1117 25EB
1118 25EB
1119 25EB
1120 25EB
1121 25EB 79
1122 25EC E607
1123 25EE 67
1124 25EF 3A3420
1125 25F2 25
1126 25F3 07
1127 25F4 C2F225
1128 25F7 3E23
1129 25F9 21A2FE
1130 25FC D0
1131 25FD 07
1132 25FE 29
1133 25FF C9
1134 2600
1135 2600

;*****
UNT2A: MVI      C,001H
        CALL    GETCH
        JZ      UNT1A
        CALL    CMP.1
        JZ      UNT2A
        SUI     '0'
        JC      QMARK

UNT3A: CPI      005H
        JNC     QMARK      ;No such Unit #!
        ORA     A
        JZ      QMARK      ;No 0 Unit # either!
        MOV     C,A
UNT1A: MOV     A,C
        STA     UNIT#
        RET

;*****
;Get Negative Sector Limit to D,E
; and # of Tracks to 'A'
;-----
SECLIM: MOV     A,C      ;Get Density & Unit#
SECL2M: ANI     007H     ;Strip to Unit#
        MOV     H,A      ;Save it in 'D'
        LDA     CONFG    ;Get Quad Configuration Flag
SECL3M: DCR     H        ;Move Drive
        RLC      ; Configuration Bit
        JNZ     SECL3M   ; to Carry Flag
        MVI     A,NTRKS  ;Set Registers to
        LXI     H,-(NTRKS*NSECTR) ; Single Head Values
        RNC      ;Return if Single Headed Drive
        RLC      ;Multiple # of Tracks by 2 = 70
        DAD     H        ;Set Single Head Track Limit
        RET      ;Return, D,E=Sector Lim & 'A'=#TRKS

PAGE

```



```

1136 2600
1137 2600
1138 2600
1139 2600 2A3120
1140 2603 7E
1141 2604 23
1142 2605 CD3526
1143 2608 CA0326
1144 260B FE23
1145 260D 3E00
1146 260F C21926
1147 2612 7E
1148 2613 23
1149 2614 223120
1150 2617 E607
1151 2619 6F
1152 261A CDC925
1153 261D CDDC25
1154 2620 7D
1155 2621 325320
1156 2624 3A3320
1157 2627 325920
1158 262A 3E01
1159 262C C3EF23
1160 262F
1161 262F
1162 262F
1163 262F CD3B26
1164 2632 FE0D
1165 2634 C8
1166 2635 FE20
1167 2637 C8
1168 2638 FE2C
1169 263A C9
1170 263B
1171 263B E5
1172 263C 2A3120
1173 263F 7E
1174 2640 FE0D
1175 2642 CA4926
1176 2645 23
1177 2646 223120
1178 2649 E1
1179 264A C9
1180 264B
1181 264B
1182 264B
1183 264B 23
1184 264C 7A
1185 264D BE
1186 264E 2B
1187 264F C0
1188 2650 7B
1189 2651 BE
1190 2652 C9

;*****
LI:      LHLD      BUFPT
LI1:     MOV       A,M
         INX       H
         CALL      CMP.1
         JZ        LI1
         CPI       '#'
         MVI       A,000H
         JNZ       LI2
         MOV       A,M
         INX       H
         SHLD      BUFPT
         ANI       007H
LI2:     MOV       L,A
         CALL      UNT2A
LIST:    CALL      UNT3A      ;Check if Unit# is legit?
         MOV       A,L
         STA       DEVICE
         LDA       SCRSZ
         STA       DIRCTR
         MVI       A,001H      ;Set "No List Switch"
         JMP       DFSR

;*****
GET.CH:  CALL      GETCH
CMP.:    CPI       CRQ
         RZ
CMP.1:   CPI       SPACE
         RZ
         CPI       COMMA
         RET

GETCH:   PUSH      H
         LHLD      BUFPT
         MOV       A,M
         CPI       CRQ
         JZ        GETCX
         INX       H
         SHLD      BUFPT
GETCX:   POP       H
         RET

;*****
CMPED:   INX       H
         MOV       A,D
         CMP       M
         DCX       H
         RNZ
         MOV       A,E
         CMP       M
         RET

```

1191 2653

1192 2653

PAGE

```

1193 2653
1194 2653
1195 2653
1196 2653 21B029 KEYN: LXI H,KEYBF ;Set Buffer Pointer
1197 2656 110120 LXI D,(KBFSZ*100H)+1 ;Set KEYBF limit
1198 2659 AF KEYN1: XRA A ; & Next Char Posit Pointer
1199 265A CD1020 CALL CINV ;Get Character
1200 265D FE03 CPI CTLQC
1201 265F CA6C26 JZ KEYN2
1202 2662 FE40 CPI ABORT
1203 2664 CA6C26 JZ KEYN2
1204 2667 FE0E CPI 'N'-40H
1205 2669 C27126 JNZ BS?
1206 266C 0640 KEYN2: MVI B,ABORT
1207 266E C32E27 JMP QMRK1 ;Go Put Abort Char to Screen
1208 2671
1209 2671 065F BS?: MVI B,'.'
1210 2673 FE5F CPI
1211 2675 CA8426 JZ BS!
1212 2678 FE11 CPI 'Q'-40H
1213 267A CA8426 JZ BS!
1214 267D 0608 MVI B,BS
1215 267F FE08 CPI BS
1216 2681 C21A27 JNZ KEYN5
1217 2684 CD9D25 BS!: CALL OUTER ;Output BACKSPACE
1218 2687 14 INR D ;Increment Limit Counter
1219 2688 1D DCR E ;Decrement Next Character
1220 2689 ; Position Pointer
1221 2689 CA3127 JZ CR&DOS ;Return to DOS,
1222 268C ; if no character to delete!
1223 268C 2B DCX H ;Decrement KEYBF Pointer
1224 268D C35926 JMP KEYN1 ;Loop for next character
1225 2690
1226 2690 PAGE

```

```

1227 2690
1228 2690
1229 2690
1230 2690 CD3B26      GETHEX: CALL    GETCH    ;Get character from buffer
1231 2693 FE20          CPI      SPACE
1232 2695 CA9026        JZ       GETHEX ;Bypass leading spaces
1233 2698 210000        LXI      H,0000H ;ReSet Digit Bytes
1234 269B CDD027      NXTHEX: CALL  TOUPP    ;Allow lower case Hex Numbers
1235 269E CDB126        CALL     ASC2H    ;Go strip ASCII bias
1236 26A1 DA2C20        JC       QMARK    ;Error exit, if not a digit
1237 26A4 29           DAD       H        ; X2
1238 26A5 29           DAD       H        ; X4
1239 26A6 29           DAD       H        ; X8
1240 26A7 29           DAD       H        ; X16
1241 26A8 85           ADD       L
1242 26A9 6F           MOV       L,A      ; + New Digit
1243 26AA CD2F26        CALL     GET.CH    ;Go get next digit
1244 26AD C29B26        JNZ      NXTHEX   ;Go get another digit
1245 26B0 C9           RET
1246 26B1
1247 26B1 FE30          ASC2H:  CPI      '0'
1248 26B3 D8           RC
1249 26B4 FE3A          CPI      '9'+1
1250 26B6 DAC226        JC       ASC2X
1251 26B9 FE41          CPI      'A'
1252 26BB D8           RC
1253 26BC FE47          CPI      'F'+1
1254 26BE 3F           CMC
1255 26BF D8           RC
1256 26C0 D607          SUI       7
1257 26C2 D630          ASC2X:  SUI      '0'
1258 26C4 C9           RET
1259 26C5
1260 26C5              PAGE

```

```

1261 26C5
1262 26C5
1263 26C5
1264 26C5
1265 26C5 CD3B26 GETDEC: CALL GETCH
1266 26C8 37 STC
1267 26C9 C8 RZ
1268 26CA FE20 CPI SPACE
1269 26CC CAC526 JZ GETDEC
1270 26CF 210000 LXI H,0000H
1271 26D2
1272 26D2 FE3A NXTDEC: CPI '9'+1
1273 26D4 D22C20 JNC QMARK
1274 26D7 D630 SUI '0'
1275 26D9 DA2C20 JC QMARK
1276 26DC 54 MOV D,H
1277 26DD 5D MOV E,L
1278 26DE 29 DAD H ; X2
1279 26DF 29 DAD H ; X4
1280 26E0 19 DAD D ; X5
1281 26E1 29 DAD H ; X10
1282 26E2 5F MOV E,A
1283 26E3 1600 MVI D,0000H
1284 26E5 19 DAD D ; + New Digit
1285 26E6 CD2F26 CALL GET.CH
1286 26E9 C8 RZ
1287 26EA C3D226 JMP NXTDEC
1288 26ED
1289 26ED CDC526 GETDEC: CALL GETDEC
1290 26F0 C3CD22 JMP DWR2T
1291 26F3
1292 26F3 PAGE

```

```

1293 26F3
1294 26F3
1295 26F3
1296 26F3 CDED25
1297 26F6 7C
1298 26F7 B7
1299 26F8 7D
1300 26F9 C31627
1301 26FC
1302 26FC
1303 26FC
1304 26FC 0601
1305 26FE CD3B26
1306 2701 C8
1307 2702 FE44
1308 2704 CA1327
1309 2707 FE20
1310 2709 CAFE26
1311 270C 0600
1312 270E FE53
1313 2710 C22C20
1314 2713
1315 2713
1316 2713
1317 2713 CD3B26
1318 2716 C22C20
1319 2719 C9
1320 271A
1321 271A
1322 271A
1323 271A 77
1324 271B 47
1325 271C CD9D25
1326 271F FE0D
1327 2721 060A
1328 2723 CA9D25
1329 2726 23
1330 2727 1C
1331 2728 15
1332 2729 C25926
1333 272C
1334 272C
1335 272C
1336 272C 063F
1337 272E CD9D25
1338 2731 CDA325
1339 2734 C32820
1340 2737
1341 2737

;*****
GETPN:  CALL    GGTDEC
        MOV     A,H
        ORA     A
        MOV     A,L
        JMP     CR?1

;*****
DENS1?: MVI     B,001H ;Flag Double Density
DENS2?: CALL    GETCH
        RZ
        CPI     'D'    ;Double Density ?
        JZ      CR?
        CPI     SPACE
        JZ      DENS2?
        MVI     B,000H ;Flag Single Density
        CPI     'S'    ;Single Density ?
        JNZ     QMARK

;*****
CR?:    CALL    GETCH
CR?1:   JNZ     QMARK
        RET

;*****
KEYN5:  MOV     M,A      ;Put character in KEYBF
        MOV     B,A
        CALL    OUTBR    ;Echo character
        CPI     CR?
        MVI     B,LF?
        JZ      OUTBR    ;Output LF after CR
        INX     H        ;Increment KEYBF Pointer
        INR     E        ;Increment Next Char Posit Pointer
        DCR     D        ;Decrement Limit Counter
        JNZ     KEYN1    ;Loop for next character

;*****
QMRK0:  MVI     B,'?'
QMRK1:  CALL    OUTBR
CR&DOS: CALL    OUTCR
        JMP     DOSV     ;Enable DOSV Trap

PAGE

```

```

1342 2737
1343 2737
1344 2737
1345 2737 310029
1346 273A 214D22
1347 273D 221A20
1348 2740 212C27
1349 2743 222D20
1350 2746 AF
1351 2747 325320
1352 274A 3C
1353 274B 323020
1354 274E 062B
1355 2750 CD9D25
1356 2753 CD5326
1357 2756 21B229
1358 2759 223120
1359 275C 2B
1360 275D 2B
1361 275E 3E0D
1362 2760 BE
1363 2761 CA3727
1364 2764 23
1365 2765 BE
1366 2766 CA6F27
1367 2769 23
1368 276A 7E
1369 276B CD3226
1370 276E 2B
1371 276F 2B
1372 2770 C29227
1373 2773 CDC427
1374 2776 210A29
1375 2779 0610
1376 277B 78
1377 277C D60B
1378 277E 324C20
1379 2781 CD4B26
1380 2784 23
1381 2785 23
1382 2786 CAD927
1383 2789 23
1384 278A 23
1385 278B 05
1386 278C C27B27
1387 278F 21B029
1388 2792 223120
1389 2795 CD7322
1390 2798 4F
1391 2799 CDB225
1392 279C D5
1393 279D CDB225
1394 27A0 7E
1395 27A1 E680
1396 27A3 B1

;*****
DOS:    LXI    SP,STACK
GETCMD: LXI    H,EDERR
        SHLD   EDERV+1
        LXI    H,QMRK0
        SHLD   QMARK+1
        XRA    A
        STA    DEVICE ;Clear List Device Assignment
        INR    A
        STA    GOTYPE ;ReSet Go Type Ident
        MVI    B,PROMPT
        CALL   OUTBR
        CALL   KEYN
AUTO:   LXI    H,KEYBF+2
        SHLD   BUFPT ;PreSet BUFPT to
        DCX    H      ; next posit after command
        DCX    H      ;Point to command again
        MVI    A,CRG
        CMP    M
        JZ     DOS    ;Quick loop back, if no entry
        INX    H
        CMP    M
        JZ     DOS01  ;Allow single character command
        INX    H
        MOV    A,M
        CALL   CMP.
DOS0:   DCX    H
DOS01:  DCX    H
        JNZ    AUTOGO
        CALL   UCASE  ;Convert command to uppercase
        LXI    H,CMDTB
        MVI    B,(CMDTX-CMDTB)/4 ;Number of table entries
HIT?:   MOV    A,B
        SUI    ((CMDTX-CMDTB)/4)+1-6 ;Check if by first 6
        STA    BOTFLG ;Clear Flag, we're ok!
        CALL   CMPHD
        INX    H
        INX    H
        JZ     HIT!
        INX    H
        INX    H
        DCR    B
        JNZ    HIT?
        LXI    H,KEYBF ;Point to Filename again
AUTOGO: SHLD   BUFPT  ;Set the buffer pointer
GO:     CALL   FFILE
        MOV    C,A
        CALL   M2DE
        PUSH   D
        CALL   M2DE
        MOV    A,M
        ANI    080H
        ORA    C

```

```

1397 27A4 4F      MOV      C,A
1398 27A5 7E      MOV      A,M
1399 27A6 E67F     ANI      07FH
1400 27A8 23      INX      H
1401 27A9 FE01     CPI      001H
1402 27AB C22C20  JNZ      QMARK
1403 27AE 7B      MOV      A,E
1404 27AF CDB225  CALL     M2DE
1405 27B2 E1      POP      H
1406 27B3 D5      PUSH     D
1407 27B4 0601     MVI      B,001H
1408 27B6 CD2220  CALL     DCOMV
1409 27B9 2A3120  G02:    LHLD     BUFPT
1410 27BC C9      RET
1411 27BD
1412 27BD CD9026  EX:     CALL     GETHEX
1413 27C0 E5      PUSH     H
1414 27C1 C3B927  JMP      G02
1415 27C4
1416 27C4 7E      UCASE:  MOV      A,M
1417 27C5 23      INX      H
1418 27C6 CDD027  CALL     TOUPP    ;Go check/convert to UPPER case
1419 27C9 5F      MOV      E,A
1420 27CA 7E      MOV      A,M
1421 27CB CDD027  CALL     TOUPP    ;Do second command character too!
1422 27CE 57      MOV      D,A
1423 27CF C9      RET
1424 27D0
1425 27D0 FE61     TOUPP:  CPI      'a'    ;Test
1426 27D2 D8      RC          ; the
1427 27D3 FE7B     CPI      'z'+1    ; range
1428 27D5 D0      RNC          ;Return if not lower case ASCII
1429 27D6 E6DF     ANI      0DFH    ;Remove lower case bias
1430 27D8 C9      RET
1431 27D9
1432 27D9 3A4C20  HIT!:   LDA      BOTFLG ;Check BOOT ONE SECTOR flag
1433 27DC A7      ANA      A
1434 27DD FA0120  JM      HIT12
1435 27E0 E5      PUSH     H
1436 27E1 21FF1F  LXI      H,DBASE-1
1437 27E4 AF      XRA      A
1438 27E5 86      CHECK:  ADD      M
1439 27E6 2D      DCR      L
1440 27E7 C2E527  JNZ      CHECK
1441 27EA D600     SUI      00      ;Checksum stored here by boot
1442 27EC 27EB     CKSUM:  EQU      $-1
1443 27EC CA0029  JZ       HIT10    ;Jump if code still intact
1444 27EF 3E01     MVI      A,1      ;A=1 Sector to Read
1445 27F1 018100  LXI      B,0001H  ;Track/Density&Unit
1446 27F4 110109  LXI      D,0001H  ;Sector/Read
1447 27F7 21001E  LXI      H,BUFFER+100H ;Memory address
1448 27FA CD5B20  CALL     DSKFNC    ;Go re-boot back sector
1449 27FD C200E3  JNZ      PBASE     ;If problem, go try to ReBoot it all
1450 2800 E1      HIT10:  POP      H
1451 2801 CDB225  HIT12:  CALL     M2DE

```


1452	2804	EB	ICHG	
1453	2805	112820	LXI	D,DOSV
1454	2808	D5	PUSH	D
1455	2809	E9	PCHL	
1456	280A			
1457	280A		PAGE	

```

1458 280A
1459 280A
1460 280A
1461 280A 494F
1462 280C A81F
1463 280E 4352
1464 2810 401F
1465 2812 524E
1466 2814 E11F
1467 2816 4445
1468 2818 321F
1469 281A 5244
1470 281C 001F
1471 281E 5752
1472 2820 031F
1473 2822
1474 2822 4558
1475 2824 BD27
1476 2826 474F
1477 2828 9527
1478 282A 4049
1479 282C 0026
1480 282E 5459
1481 2830 D329
1482 2832 4046
1483 2834 7229
1484 2836 5346
1485 2838 7528
1486 283A 5750
1487 283C 002A
1488 283E 420D
1489 2840 00E8
1490 2842 4241
1491 2844 042A
1492 2846 4D0D
1493 2848 04C0
1494 284A
1495 284A
1496 284A
1497 284A

```

;*****
CMTDB: DB 'IN'
DW IN
DB 'CR'
DW CR
DB 'RN'
DW RN
DB 'DE'
DW DE
DB 'RD'
DW RD
DB 'WR'
DW WR
;
DB 'EX'
DW EX
DB 'GO'
DW GO
DB 'LI'
DW LI
DB 'TY'
DW TY
DB 'LF'
DW LF
DB 'SF'
DW SFG
DB 'WP'
DW WP
DB 'B',CRQ
DW PBASE
DB 'BA'
DW BASIC
DB 'M',CRQ
DW MONTR

CMTX:

PAGE

1498	284A			
1499	284A	50	ERRTBL: DB	MSYNC AND 0FFH
1500	284B	55	DB	MCRC AND 0FFH
1501	284C	59	DB	MVERIFY AND 0FFH
1502	284D	60	DB	MINDEX AND 0FFH
1503	284E	66	DB	MDENS AND 0FFH
1504	284F	6B	DB	MWRTP AND 0FFH
1505	2850			
1506	2850	53594E43	MSYNC: DB	'SYNC',EOT
	2854	FF		
1507	2855	435243FF	MCRC: DB	'CRC',EOT
1508	2859	56455249	MVERIFY: DB	'VERIFY',EOT
	285D	4659FF		
1509	2860	494E4445	MINDEX: DB	'INDEX',EOT
	2864	58FF		
1510	2866	44454E53	MDENS: DB	'DENS',EOT
	286A	FF		
1511	286B	57524954	MWRTP: DB	'WRIT-P',EOT
	286F	2D50FF		
1512	2872			
1513	2872		PAGE	

```

1514 2872
1515 2872
1516 2872
1517 2872 0601
1518 2874 3E
1519 2875 0600
1520 2877 C5
1521 2878 CD7322
1522 287B C1
1523 287C 3A5620
1524 287F 4F
1525 2880 CDB225
1526 2883 D5
1527 2884 CDB225
1528 2887 7A
1529 2888 B7
1530 2889 C22C20
1531 288C 7E
1532 288D E680
1533 288F B1
1534 2890 4F
1535 2891 C5
1536 2892 7B
1537 2893 F5
1538 2894 CD9026
1539 2897 EB
1540 2898 CD1327
1541 289B F1
1542 289C C1
1543 289D E1
1544 289E C3CA22
1545 28A1
1546 28A1

;*****
LF:      MVI      B,001H  ;Flag LOAD FILE!
          DB       3EH    ;Tricky code, saves 2 cells
SFG:     MVI      B,000H  ;Flag SAVE FILE1
LFSF1:   PUSH     B       ;Save LOAD/SAVE FILE Flag
          CALL     FFIL    ;Go look-up file in directory
          POP      B
          LDA      UNIT#   ;Get Unit
          MOV      C,A     ;   to 'C'
          CALL     M2DE
          PUSH     D
          CALL     M2DE
          MOV      A,D
          ORA      A
          JNZ      QMARK
          MOV      A,M
          ANI      080H
          ORA      C
          MOV      C,A
          PUSH     B
          MOV      A,E
          PUSH     PSW
          CALL     GETHEX
          XCHG
          CALL     CR?
          POP      PSW
          POP      B
LFSF3:   POP      H
LFSF2:   JMP      DWR0T

PAGE

```

```

1547 28A1
1548 28A1
1549 28A1
1550 28A1      28B0
1551 28B0
1552 28B0
1553 28C0
1554 28C0 31001E
1555 28C3 3E01
1556 28C5 018100
1557 28C8 110109
1558 28CB 21001E
1559 28CE CD5B20
1560 28D1 C2C028
1561 28D4 213920
1562 28D7 0608
1563 28D9 3620
1564 28DB 23
1565 28DC 05
1566 28DD C2D928
1567 28E0 C3A11E
1568 28E3
1569 28E3
1570 28E3      2900
1571 2900
1572 2900
1573 2900
1574 2900

;*****
ORG      DBASE+900H-STKSZ-KPFSZ

KEYBUF:  DS      10H

INIT:    LXI      SP,STAK2 ;Set Secondary Stack Location
          MVI      A,01      ;One sector, 200H long
          LXI      B,0081H ;B = Track 0, C = DD/Unit#
          LXI      D,((DIRSZ+5)*100H)+1 ;D=Sec, E=Read
          LXI      E,BUFFER+100H ;Load address 1E00H to 2000H
          CALL     DSKFNC ;Go read-in the remainder of DOS
          JNZ      INIT      ;Loop Until Success!
          LXI      H,NAME1
          MVI      B,009H
N1CLR:   MVI      M,SPACE
          INX      H
          DCR      B
          JNZ      N1CIR
          JMP      INIT2      ;Go complete boot load

;-----
ORG      DBASE+900H
STACK:   ;Stack Address Assignment
;-----

PAGE

```

```

1575 2900
1576 2900
1577 2900
1578 2900 2900          ORG      DEASE+900H
1579 2900
1580 2900 FE01          OUTPB:  CPI      01H      ;Check if Serial Port
1581 2902 CA0A29        JZ        PORT1    ;If so, JUMP to servive Diablo
1582 2905 CD1CC0        CALL     AOUT      ;Otherwise, output the character
1583 2908 78           MOV      A,B
1584 2909 C9           RET
1585 290A
1586 290A CD1CC0        ;
1587 290D 78           PORT1:  CALL     AOUT      ;Output Char to Diablo
1588 290E FE0D        MOV      A,B
1589 2910 C0           CPI      CR0        ;Check if "RETURN"
1590 2911 C5           RNZ        ;Return if no buffer delay required
1591 2912 0603        PUSH     B           ;Save 'BC' Reg's
1592 2914 3E01        MVI      B,ETX      ;Diablo Protocol Character
1593 2916 CD1CC0        MVI      A,01H     ;Serial Port Select
1594 2919              CALL     AOUT      ;Output it to Diablo
1595 2919 3E01        ;
1596 291B CD22C0        ACK?:   MVI      A,01H ;Serial Port Select
1597 291E CA1929        CALL     AINP      ;Get char from Diablo
1598 2921 FE06        JZ        ACK?      ;Loop until we get a char
1599 2923 C21929        CPI      ACK      ;Is it an ACK character
1600 2926 C1          JNZ        ACK?      ;Loop until it is!
1601 2927 78          POP      B           ;ReStore 'BC' Reg's
1602 2928 C9          MOV      A,B        ;Required to return with char in 'A'
1603 2929              RET
1604 2929
1605 2929 2930          ;
1606 2930              ORG      DEASE+930H
1607 2930 C5          INPUT:  PUSH     B
1608 2931 CD1FC0        L0173:  CALL     SINP
1609 2934 CA3129        JZ        L0173
1610 2937 FE7F        CPI      07FH
1611 2939 C23E29        JNZ        LOAD?
1612 293C 3E5F        MVI      A,' '
1613 293E FE8C        LOAD?:   CPI      LOAD  ;LOAD?
1614 2940 CA9E29        JZ        ECHO2
1615 2943 FE80        CPI      MODE      ;MODE?
1616 2945 C24A29        JNZ        LAROW?
1617 2948 3E03        MVI      A,0E3H    ;Substitute Control-C for MODE!
1618 294A FE81        LAROW?:  CPI      081H ;LEFT ARROW?
1619 294C C25129        JNZ        RAROW?
1620 294F 3E5F        MVI      A,' '
1621 2951 FE93        RAROW?:  CPI      093H ;RIGHT ARROW?
1622 2953 C25829        JNZ        NORM?
1623 2956 3E20        MVI      A,SPACE
1624 2958 FE80        NORM?:   CPI      MODE ;Check if not Special Character?
1625 295A D25F29        JNC      ECHO1    ;Jump if SPECIAL
1626 295D C1          POP      B
1627 295E C9          RET
1628 295F
1629 295F 47          ECHO1:  MOV      B,A      ;Echo Special Character

```

1630	2960	AF	XRA	A	
1631	2961	CD1CC0	CALL	AOUT	
1632	2964	C33129	JMP	L0173	;Go get another character
1633	2967				
1634	2967		PAGE		

```

1635 2967
1636 2967
1637 2967
1638 2967 2980          ORG      DBASE+980H
1639 2980
1640 2980 CD1FC0  CNTLC:  CALL    SINP
1641 2983 C29829          JNZ     SPAC?
1642 2986 3C           INR     A
1643 2987 C9           RET
1644 2988
1645 2988 FE20  SPAC?:  CPI     SPACE  ;SPACE?
1646 298A C29329       JNZ     CTLC?
1647 298D
1648 298D CD1FC0  PAUSE?: CALL    SINP
1649 2990 CA8D29       JZ      PAUSE?
1650 2993
1651 2993 FE03  CTLC?:  CPI     CTLC?  ;Control-C ?
1652 2995 C8           RZ
1653 2996 FE80          CPI     MODE
1654 2998 C8           RZ
1655 2999 FE8C          CPI     LOAD
1656 299B C2AB29       JNZ     MODE?
1657 299E
1658 299E 060D  ECHO2:  MVI     B,CRC
1659 29A0 CD19C0       CALL    SOUT
1660 29A3 060A          MVI     B,LFQ
1661 29A5 CD19C0       CALL    SOUT
1662 29A8 C32820       JMP     DOSV
1663 29AB
1664 29AB FE80  MODE?:  CPI     MODE
1665 29AD DAB729       JC      SPDSET
1666 29B0 47          MOV     B,A
1667 29B1 CD19C0       CALL    SOUT
1668 29B4 C38029       JMP     CNTLC
1669 29B7
1670 29B7 FE30  SPDSET: CPI     '0'
1671 29B9 DA8029       JC      CNTLC
1672 29BC FE3A          CPI     '9'+1
1673 29BE D28029       JNC     CNTLC
1674 29C1 D630          SUI     '0'
1675 29C3 320BC8       STA     SPEED
1676 29C6 C38029       JMP     CNTLC
1677 29C9
1678 29C9          PAGE

```



```

1679 29C9
1680 29C9
1681 29C9
1682 29C9      29D3      ORG      DBASE+9D3H
1683 29D3
1684 29D3
1685 29D3
1686 29D3 CD7322      TY:      CALL      FFILE      ;Find File Directory
1687 29D6 110400      LXI      D,0004H      ;Increment to Byte 12D
1688 29D9 19          DAD      D      ;Point to TYPE byte address
1689 29DA E5          PUSH     H      ;Save it
1690 29DB CDF326      CALL     GETPN      ;Get Parameter Number
1691 29DE B7          ORA      A
1692 29DF FA2C20      JM      QMARK      ;Don't allow minus TYPE input
1693 29E2 47          MOV      B,A      ;Save TYPE in 'B'
1694 29E3 E1          POP      H      ;Get TYPE byte address back
1695 29E4 7E          MOV      A,M      ;Check if current
1696 29E5 E680      ANI      080H      ; is double density?
1697 29E7 B0          ORA      B      ;OR in new TYPE assignment
1698 29E8              ; with old density
1699 29E8 77          MOV      M,A      ;Make new TYPE entry in directory
1700 29E9 E67F      ANI      07FH      ;Strip off density bit
1701 29EB FE01      CPI      001H      ;Check if TYPE 1?
1702 29ED C2FA29      JNZ      WRAPUP      ;Bypass GO address if NOT
1703 29F0 E5          PUSH     H
1704 29F1 CD9026      CALL     GETHEX      ;Get GO address
1705 29F4 EB          XCHG
1706 29F5 E1          POP      H
1707 29F6 23          INX      H
1708 29F7 CDAD25      CALL     DE2M      ;Put it in the directory
1709 29FA CD1327      WRAPUP: CALL     CR?      ;Get trailing CR
1710 29FD C31F20      JMP      DWRTV      ;Go write out updated Directory
1711 2A00
1712 2A00      PAGE

```

```

1713 2A00
1714 2A00 1D00 ORG BUFFER
1715 1D00
1716 1D00 DOSBF:
1717 1D00
1718 1D00 1E00 ORG BUFFER+100H
1719 1E00
1720 1E00 1B02001B MCDOS: DB ESC,SETY,00,ESC,SETX,1FH
1E04 011F
1721 1E06 28432920 DB '(C) Micro Complex 1980'
1E0A 4D696372
1E0E 6F20436F
1E12 6D706C65
1E16 78203139
1E1A 3830
1722 1E1C 1B0131 DB ESC,SETX,31H
1723 1E1F 28373134 DB '(714) 770-2168'
1E23 29202037
1E27 37302D32
1E2B 313638
1724 1E2E 1B02051B DB ESC,SETY,05H,ESC,SETX,13H
1E32 0113
1725 1E34 20202020 DB ' Micro Complex '
1E38 20202020
1E3C 4D696372
1E40 6F20436F
1E44 6D706C65
1E48 78202020
1E4C 20202020
1E50 20
1726 1E51 0A1B0113 DB LFG,ESC,SETX,13H
1727 1E55 2D3D2A3C DB '--*< Dual Density Mc Dos >*--'
1E59 20447561
1E5D 6C204465
1E61 6E736974
1E65 79204D63
1E69 20446F73
1E6D 203E2A3D
1E71 2D
1728 1E72 0D1B0208 DB CRQ,ESC,SETY,08H,EOT
1E76 FF
1729 1E77
1730 1E77 ;*****
1731 1E77
1732 1E77 1EA0 ORG BUFFER+1A0H
1733 1EA0
1734 1EA0 C9 TINIT: RET
1735 1EA1
1736 1EA1 310029 INIT2: LXI SP,STACK ;Set Stack Location
1737 1EA4 21FF1F LXI H,DBASE-1
1738 1EA7 AF XRA A
1739 1EA8 86 CHECK: ADD M
1740 1EA9 2D DCR L
1741 1EAA C2A81E JNZ CHECK
1742 1EAD 32EB27 STA CKSUM ;Set low DOS block checksum

```

```

1743 1EB0 CD7121          CALL    SEEK1
1744 1EB3 1600            MVI     D,000H
1745 1EB5 14              SETIMR: INR    D
1746 1EB6 3EE2            MVI     A,0E2H
1747 1EB8 3D              SET2MR: DCR    A
1749 1EB9 C2B81E          JNZ     SET2MR
1749 1EB9 C2B81E          LDA     STATA
1750 1EBF B7              ORA     A
1751 1EC0 F2B51E          JP      SETIMR
1752 1EC3 7A              MOV     A,D
1753 1EC4 325A20          STA     TIMER
1754 1EC7 CD1320          CALL    TINTV    ;Go Initialize the Terminal
1755 1ECA 2100E9          LXI     H,PBASE
1756 1ECD 221420          SHLD    TINTV+1 ;Set to Re-Boot
1757 1ED0 11E61E          LXI     D,VER.
1758 1ED3 CDBE25          CALL    MESSAGE ;Output Version #
1759 1ED6 3AB028          AUTO?: LDA     KEYBF ;Check if there is a START-UP comma
1760 1ED9 B7              ORA     A ;Check if Zero?
1761 1EDA C25627          JNZ     AUTO ;Jump to AUTO, first character Not=
1762 1EDD 11001E          LXI     D,MCDOS
1763 1EE0 CDBE25          CALL    MESSAGE
1764 1EE3 C32820          JMP     DOSV
1765 1EE6
1766 1EE6 8B              VER.:  DB     CLRSCN
1767 1EE7 51756164        DB     'Quad Version 7.0',CR0,LF0,EOT
1767 1EEB 20566572
1767 1EEF 73696F6E
1767 1EF3 20372E30
1767 1EF7 0D0AFF
1768 1EFA
1769 1EFA

```

PAGE

```

1770 1EFA
1771 1EFA      1F00      ORG      BUFFER+200H
1772 1F00
1773 1F00      ;*****
1774 1F00
1775 1F00 0601      RD:      MVI      B,001H
1776 1F02 3E              DB      3EH      ;Tricky code, saves 2 cells
1777 1F03 0600      WR:      MVI      B,000H
1778 1F05 CDED26      R&W01:  CALL    GETDEC
1779 1F08 E5              PUSH    H
1780 1F09 0E01              MVI      C,001H
1781 1F0B FE20              CPI      SPACE
1782 1F0D CA181F          JZ      R&W02
1783 1F10 FE2C              CPI      COMMA
1784 1F12 C22C20          JNZ      QMARK
1785 1F15 CDC925          CALL    UNT2A
1786 1F18 CD9026      R&W02:  CALL    GETHEX
1787 1F1B E5              PUSH    H
1788 1F1C C5              PUSH    B
1789 1F1D CDF326          CALL    GETPN
1790 1F20 CDFC26          CALL    DENS17
1791 1F23 78              MOV      A,B
1792 1F24 0F              RRC
1793 1F25 B1              ORA      C
1794 1F26 C1              POP      B
1795 1F27 4F              MOV      C,A
1796 1F28 7D              MOV      A,L
1797 1F29 F22E1F          JP      R&W03
1798 1F2C 3C              INR      A
1799 1F2D 1F              RAR
1800 1F2E D1      R&W03:  POP      D
1801 1F2F C39D28          JMP      LFSF3
1802 1F32
1803 1F32 CD7322      DE:      CALL    FFILE
1804 1F35 CD1327          CALL    CR?
1805 1F38 1604          MVI      D,04H
1806 1F3A CDBA22          CALL    CLRSPC
1807 1F3D C31F20          JMP      DWRTV
1808 1F40
1809 1F40      PAGE

```

```

1810 1F40
1811 1F40
1812 1F40
1813 1F40 CD7F22
1814 1F43 CDEE23
1815 1F46 D22C20
1816 1F49 E5
1817 1F4A 113920
1818 1F4D CDEE23
1819 1F50 DA2C20
1820 1F53 E3
1821 1F54 E5
1822 1F55 CDED25
1823 1F58 E3
1824 1F59 E5
1825 1F5A 2A3120
1826 1F5D 7E
1827 1F5E FE41
1828 1F60 E1
1829 1F61 DCC526
1830 1F64 CDFC26
1831 1F67 EB
1832 1F68 78
1833 1F69 0F
1834 1F6A 324920
1835 1F6D E1
1836 1F6E D2791F
1837 1F71 23
1838 1F72 B7
1839 1F73 7C
1840 1F74 1F
1841 1F75 67
1842 1F76 7D
1843 1F77 1F
1844 1F78 6F
1845 1F79 3A5620
1846 1F7C 44
1847 1F7D 4D
1848 1F7E CDEC25
1849 1F81 2B
1850 1F82 09
1851 1F83 DA2C20
1852 1F86 19
1853 1F87 DA2C20
1854 1F8A E1
1855 1F8B E5
1856 1F8C CDAD25
1857 1F8F 71
1858 1F90 23
1859 1F91 70
1860 1F92 23
1861 1F93 3A4920
1862 1F96 77
1863 1F97 E1
1864 1F98 114820

```

```

CR:  CALL  MOV81
     CALL  NLFSR
     JNC   QMARK
     PUSH  H
     LXI   D,NAME1
     CALL  NLFSR
     JC    QMARK
     XTHL
     PUSH  H
     CALL  GETDEC
     XTHL
     PUSH  H
     LHLD  BUFPT
     MOV   A,M
     CPI   'A'
     POP   H
     CC    GETDEC
     CALL  DENS1?
     XCHG
     MOV   A,B
     RRC
     STA   TYPE
     POP   H
     JNC   CR1
     INX   H
     ORA   A
     MOV   A,H
     RAR
     MOV   E,A
     MOV   A,L
     RAR
     MOV   L,A
CR1:  LDA   UNIT#
     MOV   B,H
     MOV   C,L
     CALL  SECL2M
     DCX   H
     DAD   B
     JC    QMARK
     DAD   D
     JC    QMARK
     POP   H
     PUSH  H
     CALL  DE2M ;Disk Address to the Directory
     MOV   M,C ;Number of
     INX   H ; Sectors for
     MOV   M,B ; File Length to Directory
     INX   H
     LDA   TYPE ;File Type
     MOV   M,A ; to Directory
CR2:  POP   H ;Get Directory address for Name+8
     LXI   D,NAME2 ;Point to New File's Name+7

```

1865	1F9B	0608		MVI	B,008H	;Eight characters to move
1866	1F9D	2B	CR3:	DCX	H	;File Name
1867	1F9E	1A		LDAX	D	; to
1868	1F9F	77		MOV	M,A	; Directory
1869	1FA0	1B		DCX	D	
1870	1FA1	05		DCR	B	
1871	1FA2	C29D1F		JNZ	CR3	
1872	1FA5	C31F20		JMP	DWRTV	;Go write New Directory
1873	1FA8					
1874	1FA8			PAGE		

```

1875 1FA8
1876 1FA8
1877 1FA8
1878 1FA8 CDC925
1879 1FAB F5
1880 1FAC FE01
1881 1FAE CC2525
1882 1FB1 CDFC26
1883 1FB4 78
1884 1FB5 0F
1885 1FB6 E1
1886 1FB7 B4
1887 1FB8 4F
1888 1FB9 78
1889 1FBA 2F
1890 1FBB 47
1891 1FBC 21001F
1892 1FEF 1600
1893 1FC1 CDBA22
1894 1FC4 CDFB25
1895 1FC7 210000
1896 1FCA F5
1897 1FCB 3E0A
1898 1FCD C5
1899 1FCE E5
1900 1FCF 11001D
1901 1FD2 CDCA22
1902 1FD5 E1
1903 1FD6 010A00
1904 1FD9 09
1905 1FDA C1
1906 1FDB F1
1907 1FDC 3D
1908 1FDD C2CA1F
1909 1FE0 C9
1910 1FE1
1911 1FE1 CD7322
1912 1FE4 E5
1913 1FE5 F5
1914 1FE6 CD8422
1915 1FE9 F1
1916 1FEA 325620
1917 1FED 1A
1918 1FEE FE20
1919 1FF0 CA2C20
1920 1FF3 C3971F
1921 1FF6
1922 1FF6 0000

;*****
IN:  CALL  UNT2A
     PUSH  PSW
     CPI   01      ;If Unit# 1, pause to change disk
     CZ    PCUSE
     CALL  DENS1?
     MOV   A,B
     RRC
     POP   H
     ORA   H
     MOV   C,A
     MOV   A,B
     CMA
     MOV   B,A
     LXI   E,DOSBF+200H
     MVI   D,000H   ;Set #bytes to clear = 2*(0FFH+1)
     CALL  CLRSPC   ;Go clear 200H DOSBF
     CALL  SECLIM
     LXI   H,0000H
IN2:  PUSH  PSW      ; Save # of Tracks remaining
     MVI   A,NSECTR ; to be Initialized
     PUSH  B
     PUSH  H
     LXI   D,DOSBF
     CALL  DWR0T
     POP   H
     LXI   B,NSECTR
     DAD   B
     POP   B
     POP   PSW      ; Get # of Tracks left to do
     DCR   A
     JNZ   IN2
     RET

RN:  CALL  FFILE   ;Find File Directory Entry
     PUSH  H       ;Save Directory Address
     PUSH  PSW     ;Save File's Unit#
     CALL  MOV8    ;Put new name in NAME2 buffer
     POP   PSW
     STA   UNIT#   ;Make sure unit is not changed!
     LDAX  D       ;Get first character of new name
     CPI   SPACE   ;Check if no new name?
     JZ    QMARK   ;Got to have a new name to change!
     JMP   CR2     ;Go update directory
END

```

TOTAL ERRORS=00

TOTAL ERRORS=00

```

1 0000
2 0000      ;0005
3 0000      ;08/10/79
4 0000      ;DOUBLE DENSITY CONTROLLER
5 0000      ;PROM PROGRAM
6 0000      ;
7 0000      ;
8 0000      ;*****
9 0000      ;      DUAL DENSITY CONTROLLER PROM BOOTSTRAP      *
10 0000     ;*****
11 0000
12 0000      ;
13 0000     E800      BOOT      EQU      0E800H      ;BOOT PROGRAM ADDRESS
14 0000     E800      BASE      EQU      0E800H      ;CONTROLLER ADDRESS
15 0000     EB10      STATA     EQU      BASE+310H
16 0000     EB11      RSTSF     EQU      BASE+311H
17 0000     EB15      MOTOR     EQU      BASE+315H
18 0000     EB20      STATB     EQU      BASE+320H
19 0000     EB35      STATC     EQU      BASE+335H
20 0000     EB40      RDATA     EQU      BASE+340H
21 0000      ;
22 0000      ;
23 0000     E800      ORG      BOOT
24 E800      ;
25 E800 3E01      START: MVI      A,01      ;DEFAULT TO DRIVE #1
26 E802 5F        ALTER: MOV      E,A      ;SAVE BOOTLOAD DRIVE ASSIGNED I
27 E803 0E0A      MVI      C,12      ;SET ReTRY COUNTER
28 E805 3A15EB     LDA      MOTOR     ;TURN ON MOTORS
29 E808 1630      MVI      D,30H     ;SET FOR 600 ms DELAY
30 E80A 2110E9     LXI      H,SELECT  ;SET RETURN ADDRESS
31 E80D C3EBE9     JMP      DSECTR   ;GO DELAY UNTIL UP TO SPEED
32 E810      ;
33 E810 16EA      SELECT: MVI      D,(BASE/100H)+2 ;SELECT
34 F812 1A        LDAX     D          ;BOOTLOAD ASSIGNED DRIVE
35 E813 2119E8     LXI      H,SETLMT  ;SET RETURN ADDRESS
36 E816 C34AE8     JMP      DLY40    ;GO DELAY 40 ms MORE!
37 E819      ;
38 E819 060C      SETLMT: MVI      B,12      ;SET SECTOR LIMIT, INDEX SEARCH
39 E81B      ;
40 F81B 2121E3     INDEX?: LXI      H,INDX2 ;SET RETURN ADDRESS
41 E81E C3E9E8     JMP      NEXTS    ;GO FIND NEXT SECTOR
42 E821      ;
43 E821 3A10EB     INDX2:  LDA      STATA   ;GET STAT-A
44 E824 E640      ANI      40H      ;CHECK IF INDEX (DISK ROTATING?)
45 E826 C230E9     JNZ      MOVIN    ;JUMP IF INDEX IS FOUND
46 E829 05        DCR      B          ;DEC TRY COUNTER
47 E82A C21BE9     JNZ      INDEX?   ;LOOP FOR 12 TRIES
48 E82D      ;
49 E82D C32DE8     HANG1:  JMP      HANG1   ;HANG COMPUTER, CAN'T FIND AN INDEX
50 E830      ;
51 E830 2136E9     MOVIN:  LXI      H,MOVI2 ;SET RETURN ADDRESS
52 E833 C3A8E8     JMP      STEPIN   ;GO STEP-IN ON ASSIGNED DRIVE
53 E836 C347E8     MOVI2:  JMP      FIND0   ;GO GET ON TRACK #0
54 E839      ;
55 E839 3A20EB     MOVOUT: LDA      STATB   ;GET STAT-B

```



```

56 E83C E601      ANI      001H      ;CHECK FOR TRACK 0
57 E83E C24FEB    JNZ      NXTSEC   ;JUMP WHEN ON TRACK 0
58 E841 2147EB    LXI      H,FIND0   ;SET RETURN ADDRESS
59 E844 C3ACEB    JMP      STEPOUT  ;GO STEP-OUT ON ASSIGNED DRIVE
60 E847           ;
61 E847 2139EB    FIND0: LXI      H,MOVOUT ;SET RETURN ADDRESS
62 E84A           ;
63 E84A 1602      DLY40: MVI      D,002H ;SET FOR 40 ms DELAY
64 E84C C3EBEB    JMP      DSECTR   ;GO DELAY UNTIL SETTLED ON TRACK
65 E84F           ;
66 E84F 2155EB    NXTSEC: LXI      H,DOS? ;SET RETURN ADDRESS
67 E852 C3E9EB    JMP      NEXTS    ;GO FIND NEXT SECTOR
68 E855           ;
69 E855 3A35EB    DOS?:  LDA      STATC   ;STAT-C / TURN ON MOTORS
70 E858 E60F      ANI      00FH      ;GET SECTOR ADDRESS
71 E85A FE04      CPI      004H      ;CHECK IF POSSIBLY 1st DOS SECTOR?
72 E85C C24FEB    JNZ      NXTSEC   ;LOOP UNTIL AT SECTOR 4
73 E85F           ;
74 E85F 3A10EB    RE?:   LDA      STATA   ;GET STAT-B
75 E862 E604      ANI      004H      ;CHECK IF READ ENABLED YET?
76 E864 CA5FEB    JZ       RE?       ;LOOP UNTIL READ ENABLED
77 E867 3E09      MVI      A,009H     ;SET FOR DELAY
78 E869           ;
79 E869 3D        DD?:   DCR      A       ;DELAY
80 E86A C269EB    JNZ      DD?       ; LOOP
81 E86D 3A10EB    LDA      STATA   ;GET STAT-B
82 E870 E620      ANI      020H      ;CHECK IF DOUBLE DENSITY?
83 E872 C291EB    JNZ      SYNC?    ;JUMP IF DOUBLE DENSITY
84 E875 217BEB    LXI      H,SINGLE   ;SET RETURN ADDRESS
85 E878 C3A8EB    JMP      STEPIN   ;GO STEP-IN ON ASSIGNED DRIVE
86 E87B           ;
87 E87B 2181EB    SINGLE: LXI      H,FXNXTS ;SET RETURN ADDRESS
88 E87E C34AEB    JMP      DLY40    ;GO WAIT UNTIL SETTLED ON TRACK
89 E881           ;
90 E881 2187EB    FXNXTS: LXI      H,SDOS? ;SET RETURN ADDRESS
91 E884 C3E9EB    JMP      NEXTS    ;GO FIND NEXT SECTOR
92 E887           ;
93 E887 3A35EB    SDOS?:  LDA      STATC   ;STAT-C / TURN ON MOTORS
94 E88A E60F      ANI      00FH      ;GET SECTOR ADDRESS
95 E88C FE08      CPI      008H      ;CHECK IF ON FIRST SECTOR SINGLE DO
96 E88E C281EB    JNZ      FXNXTS   ;LOOP UNTIL READY TO READ SINGLE DO
97 E891           ;
98 E891 068C      SYNC?:  MVI      B,140   ;SET SYNC SEARCH LIMIT
99 E893 1140EB    LXI      D,RDATA   ;READ DATA
100 E896           ;
101 E896 3A10EB    SYNC2:  LDA      STATA   ;GET STAT-B
102 E899 0F        RRC             ;CHECK IF BODY DETECTED YET?
103 E89A DABEEB    JC       DOSADR    ;JUMP WHEN BODY IS FOUND
104 E89D 05        DCR      B         ;DEC LIMIT COUNTER
105 E89E C296EB    JNZ      SYNC2    ;LOOP UNTIL OUT OF TIME
106 E8A1           ;
107 E8A1 0D        RETRY?: DCR      C      ;DEC TRY COUNTER
108 E8A2 C230EB    JNZ      MOVIN    ;GO TRY AGAIN, IF NOT TRIED OUT
109 E8A5           ;
110 E8A5 C3A5EB    HANG2:  JMP      HANG2   ;HANG COMPUTER, CAN'T READ

```

```

111 E8A8      ;
112 E8A8 3E20  STEPIN: MVI      A,20H      ;BUILD STEP-IN CODE
113 E8AA 83      ADD      E      ;
114 E8AB 5F      MOV      E,A      ;
115 E8AC 16EA  STEPOUT: MVI     D,(BASE/100H)+2 ;STEP SELECT CODE IN 'DE'
116 E8AE 1A      LDAX     D      ;STEP
117 E8AF 3E10  MVI      A,10H      ; PULSE
118 E8B1 83      ADD      E      ; TO
119 E8B2 5F      MOV      E,A      ; ASSIGNED
120 E8B3 1A      LDAX     D      ; BOOTLOAD
121 E8B4 3EF0  MVI      A,-10H     ; DRIVE,
122 E8B6 83      ADD      E      ; NUMBER
123 E8B7 5F      MOV      E,A      ; SAVED IN
124 E8B8 1A      LDAX     D      ; 'E' REG
125 E8B9 7B      MOV      A,E      ;STRIP OFF
126 E8BA E607  ANI      7      ; DRIVE #
127 E8BC 5F      MOV      E,A      ;AND SAVE IT IN 'E' AGAIN
128 E8BD E9      PCHL      ;RETURN
129 E8BE
130 E8BE 3A40EB  DOSADR: LDA     RDATA    ;READ BYTE COMMAND
131 E8C1 67      MOV      H,A      ;SET DOS LOAD ADDRESS FROM 1st BYTE
132 E8C2 2E01  MVI      L,001H     ;2nd BYTE MEMORY ADDRESS IN 'HL'
133 E8C4 77      MOV      M,A      ;DELAY
134 E8C5 07      RLC      ;CALCULATE
135 E8C6 47      MOV      B,A      ; CRC IN 'B'
136 E8C7
137 E8C7 3A40EB  ;
138 E8CA 77      BLCK1: LDA     RDATA    ;GET NEXT BYTE
139 E8CB A8      MOV      M,A      ; TO MEMORY
140 E8CC 07      XRA      B      ;CALCULATE
141 E8CD 47      RLC      ; CRC
142 E8CE 2C      MOV      B,A      ; IN 'B'
143 E8CF C2C7E9  INR      L      ;POINT TO NEXT ADDRESS
144 E8D2 24      JNZ      BLCK1     ;LOOP UNTIL 1st BLOCK IS IN MEMORY
145 E8D3      INR      H      ;POINT TO NEXT BLOCK ADDRESS
146 E8D3 3A40EB  ;
147 E8D6 77      BLCK2: LDA     RDATA    ;READ BYTE
148 E8D7 A8      MOV      M,A      ; TO MEMORY
149 E8D8 07      XRA      B      ;CALCULATE
150 E8D9 47      RLC      ; CRC
151 E8DA 2C      MOV      B,A      ; IN 'B'
152 E8DB C2D3E9  INR      L      ;CHECK IF ALL 200H BYTES ARE IN MEMO
153 E8DE 1A      JNZ      BLCK2     ;LOOP UNTIL THEY ARE ALL IN!
154 E8DF A8      LDAX     D      ;GET RECORDED CRC BYTE FROM DISK
155 E8E0 C2A1E8  XRA      E      ;CHECK FOR COMPARE
156 E8E3 25      JNZ      RETRY?    ;IF IN ERROR, GO TRY TO RE-READ SEC
157 E8E4 2E0A  DCR      H      ;POINT BACK TO DOS BOOTLOAD
158 E8E6 16EF  MVI      L,00AH     ; ENTRY ADDRESS
159 E8E8 E9      MVI      D,(BASE/100H)+3 ;SET 'DE' TO BASE+3 & DRIVE
160 E8E9      PCHL      ;JUMP TO DOS BOOTLOAD
161 E8E9 1601  ;
162 E8EB      NEXTS: MVI      D,001H   ;SET TO FIND NEXT SECTOR
163 E8EB 3A11EB  ;
164 E8EE      DSECTR: LDA     RSTSF    ;RESET SECTOR FLAG
165 E8EE 3A10EB  ;
165 E8EE 3A10EB  SECT?: LDA     STATA   ;GET STAT-A

```

166	E8F1	B7	ORA	A	;CHECK FOR SECTOR FLAG
167	E8F2	F2EEEE	JP	SECT?	;LOOP UNTIL SECTOR HOLE IS FOUND
168	E8F5	15	DCR	D	;DEC SECTOR SEEK COUNTER
169	E8F6	3A11EB	LDA	RSTSF	;RESET SECTOR FLAG
170	E8F9	C2EBEB	JNZ	DSECTR	;LOOP UNTIL ON DESIRED SECTOR
171	E8FC	E9	PCHL		;RETURN
172	E8FD	0000	END		

TOTAL ERRORS=00

TOTAL ERRORS=00