

JADE COMPUTER PRODUCTS

CP/M 2.2 -- DOUBLE D

SOFTWARE MANUAL

IOD-1201M

Release 2

Copyright (C) 1980
Jade Computer Products
4901 Rosecrans
Hawthorne, California
90250

All Rights Reserved

The following is a list of specifications for this release of DOUBLE D CP/M 2.2.

1. Supports single and double density diskettes. Single and double density diskettes may be mixed on a drive-by-drive basis. The FORMAT program allows for density selection when formatting a diskette.

2. Supports single and double sided drives and diskettes. Single and double sided diskettes may be mixed on a drive-by-drive basis. The FORMAT program automatically determines the number of sides of the diskette and formats accordingly.

3. Supports the following Western Digital controller chips: FD1971-01, FD1793-01 and the entire FD179x-02 family. This software will operate both the true and inverted data bus controller chips and is controlled by setting USER SWITCH 0 on the DOUBLE D DISK CONTROLLER.

4. Supports the Serial Interface of the DOUBLE D DISK CONTROLLER as the LIST DEVICE. The EIA OUT is the serial output. In the distribution diskette, this is set to run at 9600 baud. The EIA IN is used to monitor the PRINTER READY signal. A positive signal level indicates ready. A single byte change in DCM allows for different baud rates.

The following is a list of files present on the JADE DOUBLE D diskette. A brief description is also included.

/ ASM.COM	CP/M ASSEMBLER - Provided by Digital Research.
BIOS.ASM	BASIC I/O SYSTEM - Provided by Digital Research. Bios for MDS development system.
BIOSGEN.COM	BIOS GENERATOR UTILITY - Used to read and write an image of the users CBIOS from and to system track 0. The image of DDBIOS resides at 1000H to 13FFH. Use DDT to put the DDBIOS image at that address.
/ BLT.ASM	BIOS LOADER TRANSIENT - Source code for "COLD START LOADER" as generated by MOVCPM.COM.
CPM20.COM	IMAGE of COLD START LOADER, CCP and BDOS in the same format as generated by MOVCPM 20 * and then SAVE 34 CPM20.COM.
/ DCM.ASM	DISK CONTROLLER MODULE - Source code (TDL Z80) for the DOUBLE D onboard Z80A.
DCM.HEX	DISK CONTROLLER MODULE - Intel Hex format of DCM.
DCMGEN.COM	DCM GENERATOR UTILITY - Used to read and write an image of the users DCM from or to system track 0. The image of DCM resides at 1000H to 13FFH. Use DDT to put DCM image at this address and to make patches.
/ DDBIOS.ASM	DOUBLE D BIOS - CP/M Assembler format source code for Double D BIOS. This assemble is listed in the software manual.
DDBIOS.HEX	DOUBLE D BIOS - Intel Hex format DDBIOS file.
DDBOOT.ASM	DOUBLE D BOOTSTRAP - CP/M Assembler format source code for the bootstrap.
DDT.COM	DYNAMIC DEBUG TOOL - Digital Research.
DEBLOCK.ASM	DEBLOCKING SOURCE CODE - Digital Research.
DISKDEF.LIB	DISK DEFINITION LIBRARY - Digital Research.
DUMP.ASM	FILE DUMP UTILITY - Source by Digital Research.
DUMP.COM	FILE DUMP UTILITY - COM by Digital Research.
ED.COM	EDITOR UTILITY - Digital Research.
FORMAT.ASM	FORMAT UTILITY - DOUBLE D format program source.
FORMAT.COM	FORMAT UTILITY - DOUBLE D format program. Formats on any drive A through D in single and double density.
LOAD.COM	LOAD UTILITY - Digital Research.
MOVCPM.COM	CP/M RELOCATION UTILITY - Generates CP/M system with BLT for Jade Double D.
OLDSYS.COM	SYSGEN UTILITY - Digital Research SYSGEN.COM as documented in CP/M manuals.
PIP.COM	FILE TRANSFER UTILITY - Digital Research.
STAT.COM	SYSTEM STATUS UTILITY - Digital Research.
SUBMIT.COM	CP/M BATCH SUBSYSTEM - Digital Research.
SYSGEN.COM	CSL/CCP/BDOS GENERATOR UTILITY - Double D system tracks compatible. Similar to SYSGEN.COM described in CP/M manuals but does not read or write BIOS. Use BIOSGEN for your CBIOS.
XSUB.COM	EXTENDED BATCH SUBSYSTEM - Digital Research.

The SYSTEM TRACKS have a different layout than the diskettes distributed by DIGITAL RESEARCH. This section presents a discription of the system tracks (0 and 1) as distributed for the JADE DOUBLE D disk controller board. Those modules residing on the SYSTEM TRACKS which often need to be modified for a specific system are on track 0, which is in single density. CCP and BDOS, which are not modified by the user are on track 1 in double density. All data tracks are in single density such that the DOUBLE D distribution diskette can be read and modified on most 8" single density CP/M systems.

The following table shows the layout of SYSTEM TRACK 0. This track is formatted in single density with 26 sequentially numbered sectors.

Sector Number	Execution Address	Format Ld Addr	Module Name	GRH Name	GRH ID
01	n.a.		IDT	IDT	1000
02	1380H (DD)	1080H	BLT	BLT	1001
03		1100H			1002
04	4A00H+b	1180H	BIOS	DCM	1003
05	4A80H+b	1200H	BIOS	DCM	
06	4B00H+b	1280H	BIOS	DCM	
07	4B80H+b	1300H	BIOS	DCM	
08	4C00H+b	1380H	BIOS	DCM	
09	4C80H+b	1400H	BIOS	DCM	
10	4D00H+b	1480H	BIOS	DCM	
11	4D80H+b	1500H	BIOS	DCM	
12		1580H	RSV	RSV	
13	1000H (DD)	1600H	DCM	BIOS	1004
14	1080H (DD)	1680H	DCM	BIOS	1005
15	1100H (DD)	1700H	DCM	BIOS	1006
16	1180H (DD)	1780H	DCM	BIOS	1007
17	1200H (DD)	1800H	DCM	BIOS	1008
18	1280H (DD)	1880H	DCM	BIOS	1009
19	1300H (DD)	1900H	DCM	BIOS	1010
20	1380H (DD)	1980H	DCM	BIOS	1011
21	1400H (DD)	1A00H	RSV	BIOS	RSV
22	1480H (DD)	1A80H	RSV	BIOS	RSV
23	1500H (DD)	1B00H	RSV	BIOS	RSV
24	1580H (DD)	1B80H	RSV	BIOS	RSV
25	1600H (DD)	1C00H	RSV	BIOS	RSV
26	1680H (DD)	1C80H	RSV	BIOS	RSV

The following table shows the layout of SYSTEM TRACK 1. This track is formatted in double density with 50 physically staggered sectors.

Sector Number	Execution Address	Format Ld Addr	Module Name
01		1D00H	SPARE
02	3400H+b	1D80H	CCP
03	3480H+b	1E00H	CCP
04	3500H+b	1E80H	CCP
05	3580H+b	1F00H	CCP
06	3600H+b	1F80H	CCP
07	3680H+b	2000H	CCP
08	3700H+b	2080H	CCP
09	3780H+b	2100H	CCP
10	3800H+b	2180H	CCP
11	3880H+b	2200H	CCP
12	3900H+b	2280H	CCP
13	3980H+b	2300H	CCP
14	3A00H+b	2380H	CCP
15	3A80H+b	2400H	CCP
16	3B00H+b	2480H	CCP
17	3B80H+b	2500H	CCP
18	3C00H+b	2580H	BDOS
19	3C80H+b	2600H	BDOS
20	3D00H+b	2680H	BDOS
21	3D80H+b	2700H	BDOS
22	3E00H+b	2780H	BDOS
23	3E80H+b	2800H	BDOS
24	3F00H+b	2880H	BDOS
25	3F80H+b	2900H	BDOS
26	4000H+b	2980H	BDOS
27	4080H+b	2A00H	BDOS
28	4100H+b	2A80H	BDOS
29	4180H+b	2B00H	BDOS
30	4200H+b	2B80H	BDOS
31	4280H+b	2C00H	BDOS
32	4300H+b	2C80H	BDOS
33	4380H+b	2D00H	BDOS
34	4400H+b	2D80H	BDOS
35	4480H+b	2E00H	BDOS
36	4500H+b	2E80H	BDOS
37	4580H+b	2F00H	BDOS
38	4600H+b	2F80H	BDOS
39	4680H+b	3000H	BDOS
40	4700H+b	3080H	BDOS
41	4780H+b	3100H	BDOS
42	4800H+b	3180H	BDOS
43	4880H+b	3200H	BDOS
44	4900H+b	3280H	BDOS
45	4980H+b	3300H	BDOS
46		3380H	SPARE
47		3400H	SPARE
48		3480H	SPARE
49			SPARE
50			SPARE

ADD A130

SYSTEM TRACK GENERATOR UTILITIES

The three generator utilities SYSGEN.COM, BIOSGEN.COM, and DCMGEN.COM provide the end user the ability to extract and rewrite various sections of the system tracks. The following table shows which sections of memory are used by each program and which system track modules are read or rewritten. SYSGEN.COM is similar to the SYSGEN.COM described in the CP/M 2.0 manual set. Notice the difference is the BIOS module.

UTILITY	MODULE	TRACK	SECTORS	SYSTEM ADDRESS
SYSGEN.COM	BLT	0	2	0900-097FH
	CCP	1	2-17	0980-117FH
	BDOS	1	18-45	1180-1F7FH
BIOSGEN.COM	BIOS	0	4-11	1000-13FFH
DCMGEN.ASM	DCM	0	13-20	1000-13FFH

CHANGING SYSTEM SIZE

The following section is intended to lead the customer through the sequence of operations needed to change the operating system size of a diskette. A 32K system is generated in this example.

Make a copy of DDBIOS.ASM, calling the new copy DDBIOS32.ASM indicating that this is to be a 32K DDBIOS. Edit this file changing the equate CPM\$NK from 20 to 32. See example below. Assemble this new program (expecting zero errors). In the PRN file of this assembly note the value generated for the name BIOS\$R. This value is used when loading DDBIOSnn.HEX. See the PRN section below.

```

; DECLARE CP/M 2.2 SYSTEM SIZE
;*****

0020 =      CPM$NK EQU      32      ;SYSTEM SIZE K BYTES.

;*****
; DOUBLE D HARDWARE PARAMETER - SYSTEM PORT AD
;*****

0043 =      D$PORT EQU      043H    ;DOUBLE D PORT ADDRESS

;*****
; SELECT NUMBER OF DISK DRIVES USED
;*****

0002 =      N$DRVS EQU      2       ;SELECT 1 TO 4 DRIVES.

;*****
; DISK OPERATING SYSTEM ADDRESSES.
;*****

0400 =      K$B      EQU      1024      ;1K BYTE SIZE.
8000 =      CPM$SZ   EQU      CPM$NK * K$B ;TOP SYSTEM AD
3000 =      CPM$BS   EQU      CPM$SZ-(20*K$B) ;CP/M BIAS VAL

0100 =      TPA      EQU      0100H     ;ADDRESS OF TP
6400 =      CCP      EQU      CPM$BS+3400H ;ADDRESS OF CC
6C00 =      BDOS     EQU      CPM$BS+3C00H ;ADDRESS OF BD
7A00 =      BIOS     EQU      CPM$BS+4A00H ;ADDRESS OF BI
9600 =      BIOS$R   EQU      1000H-BIOS ;DDT OFFSET 10
F000 =      BOOT     EQU      0F000H    ;BOOT PROM JUM
0003 =      IO$LOC   EQU      0003H    ;I/O BYTE LOCA
0004 =      DF$LOC   EQU      0004H    ;DRIVE ASSIGN

;*****
; DOUBLE D SYSTEM PARAMETERS

```

The following section displays the system interaction as viewed from the console when writing DDBIOSnn.HEX to the system tracks. The left side of the example is the console interaction. The SYSTEM is printing in UPPER CASE while the user is typing in lower case. The right side of the example contains comments. PLEASE NOTE that the value for LOAD WITH OFFSET is the value set for BIOS#R.

CONSOLE INTERACTION	COMMENTS
A>ddt	EXECUTE DDT
DDT VERS 2.2	DDT SIGNON
-f1000,13ff,0	CLEAR MEMORY
-iddbios32.hex	ENTER FILENAME
-r9600	LOAD WITH OFFSET
NEXT PC	DDT RESPONDS
1309 0000	
-l1000	DISSASSEMBLE
1000 JMP 7A36	TO VERIFY LOAD
1003 JMP 7A42	
1006 JMP F006	
1009 JMP F009	
100C JMP F00C	
100F JMP 7AB0	
1012 JMP 7AAF	
1015 JMP 7AAC	
1018 JMP 7AD0	
101B JMP 7AD5	
101E JMP 7AF4	LOOKS GOOD
-s0	REBOOT SYSTEM
A>biosgen	EXECUTE BIOSGEN
JADE COMPUTER PRODUCTS	BIOSGEN SIGNS ON
BIOSGEN 2.2 - DOUBLE D	
EXTRACT BIOS FROM DRIVE (CR TO BYPASS)?	DO NOT EXTRACT
WRITE BIOS ON DRIVE (CR TO EXIT)? b	SELECT DRIVE
TYPE CR WHEN DRIVE B READY.	TYPE CR WHEN READY.
WRITE BIOS ON DRIVE (CR TO EXIT)?	WRITING TO DRIVE
A>	TYPE CR TO EXIT
	BACK TO CP/M

Now that DDBIOS has been written to the system tracks we will proceed to load CCP/BDOS. The following section displays system interaction as viewed from the console when generating a new size CCP/BDOS and writing this to the system tracks.

CONSOLE INTERACTION	COMMENTS
A>movcpm 32 *	EXECUTE MOVCPM USE '*' OPTION
CONSTRUCTING 32K CP/M VERS 2.2 READY FOR "SYSGEN" OR "SAVE 34 CPM32.COM"	MOVCPM SIGNON
A>sysgen	MOVCPM FINISHED EXECUTE SYSGEN
JADE COMPUTER PRODUCTS SYSGEN 2.2 - DOUBLE D	SYSGEN SIGNON
EXTRACT SYS FROM DRIVE (CR TO BYPASS)?	DO NOT EXTRACT
WRITE SYS ON DRIVE (CR TO EXIT)? b TYPE CR WHEN DRIVE B READY.	SELECT DRIVE CR WHEN READY
WRITE SYS ON DRIVE (CR TO EXIT)? A>	WRITING ON DRIVE CR TO EXIT

This completes the steps needed to generate the system tracks for a different system size.

Besides containing CCP/BDOS and DBBIOS, the system tracks must also contain DCM (Disk Controller Module). The following sequence display system interaction as viewed from the console when writing DCM to the system tracks. Please note that after verifying a proper load (by displaying some of DCM) that any timing or other modifications to DCM should be made before exiting DDT.

 CONSOLE INTERACTION

COMMENTS

A>ddt	EXECUTE DDT
DDT VERS 2.2	DDT SIGNS ON
-f1000,13ff,0	CLEAR MEMORY
-idcm2.hex	ENTER FILENAME
-r	LOAD ABSOLUTE
NEXT PC	DDT RESPONDS
13AF 0000	
-d1000,103f	TO VERIFY LOAD
1000 C3 00 00 C3 80 17 41 10 06 19 10 FE C3 74 10 00	A..
1010 5E 01 50 00 50 00 01 00 FD E1 DB 05 D3 07 78 A9 ^.P.P....	
1020 D3 04 08 FE 3E D0 A9 D3 04 E3 E3 E3 E3 DB 04 A9	>.....
1030 C9 00 00 00 00 00 00 00 DB 20 D1 2A 06 10 E9 FB	
-s0	REBOOT SYSTEM
 A>dcmgen	 EXECUTE DCMMGEN
JADE COMPUTER PRODUCTS	DCMGEN SIGNS ON
DCMGEN 2.2 - DOUBLE D	
EXTRACT DCM FROM DRIVE (CR TO BYPASS)?	DO NOT EXTRACT
WRITE DCM ON DRIVE (CR TO EXIT)?	SELECT A DRIVE
TYPE CR WHEN DRIVE B READY.	TYPE CR WHEN READY.
WRITE DCM ON DRIVE (CR TO EXIT)?	TYPE CR TO EXIT
A>	BACK TO CP/M

NEW CP/M 2.2
BDOS FUNCTIONS

```
*****  
* FUNCTION 37:  RESET DRIVE      *  
*                               *  
*****  
* Entry Parameters:             *  
*   Register C:  25H            *  
*   Register DE: Drive Vector  *  
*                               *  
* Returned Value :              *  
*   Register A:  00H            *  
*****
```

The RESET DRIVE function allows resetting of specified drive(s). The passed parameter is a 16 bit vector of drives to be reset, the least significant bit is drive A:.

In order to maintain compatibility with MP/M, CP/M returns a zero value.

```
*****  
* FUNCTION 40:  WRITE RANDOM WITH*  
*              ZERO FILL        *  
*****  
* Entry Parameters:             *  
*   Register C:  28H            *  
*   Register DE: FCB Address    *  
* Returned Value:               *  
*   Register A:  Return Code    *  
*****
```

The WRITE RANDOM WITH ZERO FILL operation is similar to FUNCTION 34: with the exception that a previously unallocated block is filled with zeros before the data is written.

DIGITAL RESEARCH CP/M^R 2.2 FIELD SOFTWARE CHANGE

Copyright © 1980 Digital Research

ID# CPM22-0001 PROGRAM: BDOS ISSUE DATE: 02/19/80

Error Description: The following change affects only those CP/M systems which are using the optional blocking and deblocking algorithms listed in Appendix G of the CP/M Alteration Guide. If you are in doubt as to the applicability of this field change, please contact Digital Research or your CP/M distributor.

Patch Procedure: Ensure you have an archive copy of the distributed MOVCPM.COM file. Make changes to a version of MOVCPM.COM by carefully following the steps shown below: MOVCPM.COM is loaded into memory using DDT and the changes are made using the Assemble (A) and Set (S) commands. After making the changes, return to the console command processor using the Go (G) command, and SAVE the altered memory image on disk. The memory image on tracks 0 and 1 must also be updated. This can be accomplished by executing the new MOVCPM program, as described in the CP/M Alteration Guide, and integrating your customized I/O system.

```
ddt movepm.com
DDT VERS 2.0
NEXT PC
2700 0100
-a1cd2
1CD2 nop
1CD3 nop
1CD4 lxi h,0
1CD7
```

-G0

```
save 38 movepm.com
```

NOTE: This Field Software change is not installed in the CP/M version 2.2. It must be installed in all systems which use the deblocking algorithms listed in Appendix G of the CP/M Alteration Guide.

```

; *****
;
; PROGRAM ID: DOUBLE D BIOS (DDBIOS)
;
; *****
;
; VERSION: CP/M 2.2 8" RELEASE 2A
;
; *****
;
; PRESENTED BY: JADE COMPUTER PRODUCTS
; 4901 W. ROSECRANS BLVD.
; HAWTHORNE, CALIFORNIA
; 90250, U.S.A.
;
; ***** SK *****

```

```

; *****
; DECLARE CP/M 2.2 SYSTEM SIZE
; *****

```

```

0014 = CPM$NK EQU 20 ;SYSTEM SIZE K BYTES.

```

```

; *****
; DOUBLE D HARDWARE PARAMETER - SYSTEM PORT ADDRESS
; *****

```

```

0043 = D$PORT EQU 043H ;DOUBLE D PORT ADDRESS.

```

```

; *****
; SELECT NUMBER OF DISK DRIVES USED
; *****

```

```

0002 = N$DRVS EQU 2 ;SELECT 1 TO 4 DRIVES.

```

```

; *****
; DISK OPERATING SYSTEM ADDRESSES.
; *****

```

```

0400 = K$B EQU 1024 ;1K BYTE SIZE.
5000 = CPM$SZ EQU CPM$NK * K$B ;TOP SYSTEM ADDRESS.
0000 = CPM$BS EQU CPM$SZ-(20*K$B) ;CP/M BIAS VALUE.

```

```

0100 = TPA EQU 0100H ;ADDRESS OF TPA.
3400 = CCP EQU CPM$BS+3400H ;ADDRESS OF CCP.
3C00 = BDOS EQU CPM$BS+3C00H ;ADDRESS OF BDOS.
4A00 = BIOS EQU CPM$BS+4A00H ;ADDRESS OF BIOS
C600 = BIOS$R EQU 1000H-BIOS ;DDT OFFSET 1000H LOAD.
F000 = BOOT EQU 0F000H ;BOOT PROM JUMP TABLE.
0003 = IO$LOC EQU 0003H ;I/O BYTE LOCATION.
0004 = DF$LOC EQU 0004H ;DRIVE ASSIGN LOCATION.

```

```

; *****
; DOUBLE D SYSTEM PARAMETERS
; *****

```

```

0000 = IOBYTE EQU 00000000B ;INITIAL IOBYTE VALUE.
0000 = DF$DRV EQU 0 ;INITIAL DEFAULT DRV.

```

```

0080 =          SEC$SZ EQU      0080H          ;BYTES PER SECTOR.
0100 =          FMT$SZ EQU      0100H          ;FORMAT BUFF SIZE.

;*****
; DOUBLE D HARDWARE COMMANDS *
;*****

0001 =          DC$SIN EQU      00000001B     ;SWITCH DD BANK 0 INTO SYSTEM.
0001 =          DC$MBO EQU      00000001B     ;SELECT DOUBLE D BANK 0.
0003 =          DC$MB1 EQU      00000011B     ;SELECT DOUBLE D BANK 1.
0000 =          DC$SOT EQU      00000000B     ;SWITCH DD MEM OUT OF SYSTEM.
0002 =          DC$INT EQU      00000010B     ;ISSUE DD Z80A INTERRUPT.

;*****
; DISK CONTROLLER MODULE LINKAGE (DCM - VER 2.2) *
;*****

;***** ( DCM ADDRESSES DEFINED )*****

0370 =          DD$CBT EQU      0370H          ;COMMAND BLOCK (BANK 0).
0380 =          DD$BUF EQU      0380H          ;SECTOR BUFFER (BANK 0).
0300 =          DD$FBF EQU      0300H          ;FORMAT BUFFER (BANK 1).
03A0 =          DD$DPB EQU      03A0H          ;ID SEC DPB (BANK 0).
03B1 =          DD$DDF EQU      03B1H          ;ID SEC FLAGS (BANK 0).

;***** ( DCM COMMANDS )*****

0000 =          DC$LOG EQU      000H           ;LOG ON DISKETTE.
0001 =          DC$RDS EQU      001H           ;READ SECTOR.
0002 =          DC$WRS EQU      002H           ;WRITE SECTOR.
0003 =          DC$FMT EQU      003H           ;FORMAT TRACK.
0005 =          DC$LST EQU      005H           ;LIST CHARACTER.
0006 =          DC$LCK EQU      006H           ;LIST STATUS.

;*****
; ASSEMBLER DIRECTIVES *
;*****

4A00                                ORG      BIOS

;*****
; BIOS JUMP VECTOR TABLE *
;*****

4A00 C3364A          JMP      INIT           ;COLD START ENTRY
4A03 C3424A          JMP      WARM           ;RELOAD CCP/BDOS
4A06 C306F0          JMP      CNS$CK         ;GET CONSOLE STATUS
4A09 C309F0          JMP      CNS$IN         ;CONSOLE INPUT
4A0C C30CF0          JMP      CNS$OT         ;CONSOLE OUTPUT
4A0F C3B04A          JMP      LIST           ;PRINTER OUTPUT
4A12 C3AF4A          JMP      PUNCH          ;PUNCH OUTPUT
4A15 C3AC4A          JMP      READER         ;READER INPUT
4A18 C3D04A          JMP      HOME           ;HOME SELECTED DRIVE
4A1B C3D54A          JMP      SELDSK         ;SELECT DISK DRIVE
4A1E C3F44A          JMP      SETTRK         ;SET TRACK NUMBER
4A21 C3F94A          JMP      SETSEC         ;SET SECTOR NUMBER
4A24 C3FE4A          JMP      SETDMA         ;SET TRANSFER ADDRESS
4A27 C3044B          JMP      DISKRD         ;PERFORM DISK READ
4A2A C3244B          JMP      DISKWR         ;PERFORM DISK WRITE
4A2D C3C04A          JMP      LISTST        ;RETURN LIST STAT

```

```

4A30 C3514B      JMP      SECTRN      ;TRANSLATE SECTOR
4A33 C3614B      JMP      FORMAT      ;FORMAT A TRACK

```

```

;*****
; COLD START ENTRY - ISSUE SIGN ON MESSAGE      *
;*****

```

```

4A36 318000     INIT:   LXI      SP,0080H      ;SET UP STACK AREA.
4A39 21964C     LXI      H,MSG$S0      ;SIGN ON MSG ADDR.
4A3C CD4C4C     CALL     MSG$OT        ;ISSUE MESSAGE.
4A3F C3574A     JMP      CPM$LD        ;LOAD CCP/BDOS.

```

```

;*****
; WARM BOOT ENTRY - LOADS CCP/BDOS - INITIALIZES  *
;*****

```

```

;***** ( SET UP FOR CCP/BDOS LOAD )*****

```

```

4A42 3A0300     WARM:   LDA      IO$LOC      ;GET I/O BYTE VALUE.
4A45 32914C     STA      IO$IMG      ;STORE I/O VALUE.
4A48 3A0400     LDA      DF$LOC      ;GET DEFAULT DRIVE.
4A4B FE02       CPI      N$DRVS      ;CHECK LEGAL DRIVE.
4A4D DA514A     JC      WRM$OK        ;IF LEGAL, GO OK.
4A50 AF        XRA      A            ;SET DRIVE TO A.
4A51 32924C     WRM$OK: STA      DF$IMG      ;STORE IN IMAGE.
4A54 318000     LXI      SP,0080H      ;SET UP STACK.
4A57 3E00       CPM$LD: MVI      A,DF$DRV    ;INIT DEFAULT DRIVE.
4A59 32594C     STA      BT$DRV      ;SELECT DISK.
4A5C 010034     LXI      B,CCP        ;CP/M CCP ADDRESS.
4A5F CDFE4A     CALL     SETDMA      ;SET DMA ADDR.
4A62 0E02       MVI      C,2          ;CCP 1ST SECTOR.
4A64 CDF94A     CALL     SETSEC      ;SET SECTOR NMBR.
4A67 0E01       MVI      C,1          ;CCP/BDOS TRACK.
4A69 CDF44A     CALL     SETTRK      ;SET TRACK NUMBER.

```

```

;***** ( LOAD CCP/BDOS )*****

```

```

4A6C CD044B     W$READ: CALL     DISKRD      ;READ ONE SECTOR.
4A6F A7        ANA      A            ;SET FLAGS.
4A70 C28C4A     JNZ     W$EROR      ;EXIT IF ERROR.
4A73 3A5B4C     LDA      BT$SEC      ;GET SECTOR NMBR.
4A76 FE2D       CPI      45          ;LAST SECTOR CHECK.
4A78 CA934A     JZ      W$ZRPG      ;GOTO ZERO PAGE SET.
4A7B 3C        INR      A            ;INCREMENT SECTOR.
4A7C 325B4C     STA      BT$SEC      ;STORE NEXT SECTOR.
4A7F 118000     LXI      D,SEC$SZ    ;GET SECTOR SIZE.
4A82 2A604C     LHLD    BT$DMA      ;GET TRANSFER ADDR.
4A85 19        DAD      D            ;CALCULATE NEW ADDR.
4A86 22604C     SHLD    BT$DMA      ;SET NEW ADDRESS.
4A89 C36C4A     JMP      W$READ      ;DO ANOTHER WARM READ.

```

```

;***** ( READ ERROR DETECTED )*****

```

```

4A8C 21CB4C     W$EROR: LXI      H,MSG$LE    ;GET ERROR MESSAGE.
4A8F CD4C4C     CALL     MSG$OT      ;ISSUE MESSAGE.
4A92 76        HLT                      ;OR GOTO MONITOR

```

```

;***** ( INITIALIZE SYSTEM PARAMETERS )*****

```

```

4A93 010800     W$ZRPG: LXI      B,S      ;BASE IMAGE SIZE.

```

```

4A96 110000      LXI      D,0           ;BASE ADDRESS SET.
4A99 218E4C      LXI      H,BS$IMG     ;BASE IMAGE ADDR.
4A9C CD414C      CALL     BLOCK        ;BLOCK MOVE ROUTINE.
4A9F 218000      LXI      H,0080H     ;DEFAULT SECTOR BUFF.
4AA2 22604C      SHLD    BT$DMA       ;SET TRANSFER ADDRESS.

```

```

;***** ( JUMP TO CCP )*****

```

```

4AA5 3A0400      LDA      DF$LOC       ;GET CURRENT DSK NMBR.
4AA8 4F          MOV      C,A         ;SEND TO THE CCP.
4AA9 C30034      JMP      CCP          ;JUMP INTO CCP CP/M.

```

```

;*****
; CONSOLE LINKAGE DEFINITIONS - BOOT PROM ADDRESSES *
;*****

```

```

F006 =          CNS$CK EQU      BOOT+006H     ;CHECK CONSOLE INPUT.
F009 =          CNS$IN EQU      BOOT+009H     ;READ CONSOLE INPUT.
F00C =          CNS$OT EQU      BOOT+00CH     ;CHARACTER TO CONSOLE.

```

```

;*****
; READER AND PUNCH DRIVERS - USER SHOULD DEFINED *
;*****

```

```

4AAC 3E1AC9      READER: MVI  A,CNTL$Z!RET ;RETURN END OF FILE.
4AAF C9          PUNCH:  RET           ;NOT IMPLEMENTED.

```

```

;*****
; PRINTER DRIVER AREA - DCM SERIAL PORT LINKAGE *
;*****

```

```

4AB0 79          LIST:   MOV      A,C           ;LIST CHAR TO ACUM.
4AB1 325D4C      STA      BT$CHR        ;STORE LIST CHARACTER.
4AB4 3E01        MVI      A,DC$SIN      ;LOAD SWITCH MEM CMND.
4AB6 D343        OUT      D$PORT       ;ISSUE HARDWARE CMND.
4AB8 3E05        MVI      A,DC$LST      ;DCM LIST COMMAND.
4ABA CD8A4B      CALL     DSK$EX        ;CALL DISK EXECUTE.
4ABD C3444B      JMP      DSK$OK        ;RETURN TO CALLER.

```

```

4AC0 3E01        LISTST: MVI      A,DC$SIN ;LOAD SWITCH MEM CMND.
4AC2 D343        OUT      D$PORT       ;ISSUE HARDWARE CMND.
4AC4 3E06        MVI      A,DC$LCK      ;DCM LIST STAT CMND.
4AC6 CD8A4B      CALL     DSK$EX        ;CALL DISK EXECUTE.
4AC9 CD444B      CALL     DSK$OK        ;SWITCH DD MEM OUT.
4ACC 3A5F4C      LDA      BT$STS        ;LOAD RETURN STATUS.
4ACF C9          RET           ;RETURN TO CALLER.

```

```

;*****
; HOME - SET TRACK TO ZERO *
;*****

```

```

4AD0 0E00        HOME:   MVI      C,0           ;C REGISTER TO ZERO.
4AD2 C3F44A      JMP      SETTRK        ;PERFORM SET TRACK.

```

```

;*****
; SELECT DISK DRIVE - CHECK FOR LOGON *
;*****

```

```

4AD5 210000      SELDSK: LXI      H,0           ;ERROR RETURN CODE.
4AD8 79          MOV      A,C           ;PUT DRIVE NMBR IN A.

```



```

4AD9 FE02          CPI      N$DRVS      ;CHECK IF LEGAL DRIVE.
4ADB D0           RNC          ;NO CARRY IF ILLEGAL.
4ADC 32594C      STA      BT$DRV      ;STORE DRIVE NUMBER.
4ADF 7B          MOV      A,E        ;CHECK IF LOG-ON REQ.
4AE0 32644C      STA      LOG$RQ      ;STORE LOGON REGISTER.
4AE3 3A594C      RETDISK: LDA     BT$DRV      ;GET DRIVE NUMBER.
4AE6 6F          MOV      L,A        ;L SET DISK NUMBER.
4AE7 2600        MVI      H,0        ;ZERO H REGISTER.
4AE9 29          DAD      H          ;*2.
4AEA 29          DAD      H          ;*4.
4AEB 29          DAD      H          ;*8.
4AEC 29          DAD      H          ;*16 (SIZE OF HEADER).
4AED 11E94C      LXI      D,D0$DPH    ;DRIVE 0 D$P$H.
4AF0 19          DAD      D          ;HLSET DRIVE N DPH.
4AF1 C3B54B      JMP      LOG$ON     ;GO CHECK LOG-ON.

```

```

;*****
; SET TRACK NUMBER *
;*****

```

```

4AF4 79          SETTRK: MOV     A,C        ;MOVE TRACK NUMBER.
4AF5 325A4C      STA      BT$TRK     ;SAVE TRACK NUMBER.
4AF8 C9          RET          ;RETURN TO CALLER.

```

```

;*****
; SET SECTOR NUMBER *
;*****

```

```

4AF9 79          SETSEC: MOV     A,C        ;MOVE SECTOR NUMBER.
4AFA 325B4C      STA      BT$SEC     ;SAVE SECTOR NUMBER.
4AFD C9          RET          ;RETURN TO CALLER.

```

```

;*****
; SET MEMORY ADDRESS FOR DISK TRANSFER *
;*****

```

```

4AFE 60          SETDMA: MOV     H,B      ;HIGH ORDER MOVE.
4AFF 69          MOV     L,C      ;LOW ORDER MOVE.
4B00 22604C      SHLD   BT$DMA     ;SAVE TRANSFER ADDRESS.
4B03 C9          RET          ;RETURN TO CALLER.

```

```

;*****
; READ A DISK SECTOR ROUTINE *
;*****

```

```

4B04 3E01      DISKRD: MVI     A,DC$SIN ;SWITCH DD INTO SYSTEM.
4B06 D343      OUT     D$PORT     ;ISSUE DD COMMAND.
4B08 3E01      MVI     A,DC$RDS   ;READ SECTOR COMMAND.
4B0A CD8A4B      CALL   DSK$EX     ;PERFORM OPERATION.
4B0D C24A4B      JNZ     DSK$ER     ;ERROR EXIT.
4B10 2A604C      LHLD   BT$DMA     ;LOAD USER BUF ADDRESS
4B13 EB          XCHG           ;MOVE HL TO DE.
4B14 018003      LXI     B,DD$BUF   ;LOAD BUFFER OFFSET.
4B17 2A4000      LHLD   D$ADDR     ;LOAD DD WINDOW ADDR.
4B1A 09          DAD     B          ;HL NOW SECTOR BUFFER.
4B1B 018000      LXI     B,SEC$SZ   ;LOAD SECTOR SIZE.
4B1E CD414C      CALL   BLOCK      ;BLOCK MOVE ROUTINE.
4B21 C3444B      JMP     DSK$OK     ;NORMAL RETURN.

```

```

;*****

```

; WRITE A DISK SECTOR ROUTINE *

```
4B24 3E01    DISKWR: MVI    A,DC$SIN    ;SWITCH DD INTO SYSTEM.
4B26 D343    OUT     D$PORT    ;ISSUE HARDWARE CMND.
4B28 018000 LXI     B,SEC$SZ    ;LOAD SECTOR SIZE.
4B2B 2A4000 LHL    D$ADDR     ;DD SYSTEM ADDRESS.
4B2E 118003 LXI     D,DD$BUF    ;DD BUFFER OFFSET.
4B31 19      DAD     D        ;HL NOW DD BUF ADDR.
4B32 EB      XCHG    ;DE NOW DD BUF ADDR.
4B33 2A604C LHL    BT$DMA     ;HL NOW USER BUF ADDR.
4B36 CD414C CALL   BLOCK      ;BLOCK MOVE ROUTINE.
4B39 3E02    MVI    A,DC$WRS   ;LOAD WRITE SEC CMND.
4B3B CD8A4B CALL   DSK$EX     ;CALL DISK EXECUTIVE.
4B3E CA444B JZ     DSK$OK     ;JUMP IF WRITE OK.
4B41 C34A4B JMP    DSK$ER     ;ERROR EXIT.
```

; DISK READ/WRITE EXITS *

```
4B44 3E00    DSK$OK: MVI    A,DC$SOT    ;SWITCH DD OUT OF SYS.
4B46 D343    OUT     D$PORT    ;ISSUE HARDWARE CMND.
4B48 AF      XRA     A        ;ZERO A REGISTER.
4B49 C9      RET     ;NORMAL EXIT.
```

```
4B4A 3E00    DSK$ER: MVI    A,DC$SOT    ;SWITCH DD OUT OF SYS.
4B4C D343    OUT     D$PORT    ;ISSUE HARDWARE CMND.
4B4E 3EFF    MVI    A,OFFH     ;LOAD ERROR FLAGS.
4B50 C9      RET     ;ERROR EXIT.
```

; TRANSLATE SECTOR NUMBER *

```
4B51 7A      SECTRN: MOV     A,D        ;TESTING TBL ADDR.
4B52 B3      ORA     E        ;ADDR IN REG DE.
4B53 CA5C4B JZ     NOTRAN     ;IF ZERO, NO TRANS.
4B56 EB      XCHG    ;(HL) NOW TRANS TBL.
4B57 09      DAD     B        ;(HL) NOW TRANS SECTOR.
4B58 6E      MOV     L,M      ;L IS TRANSLATED SEC.
4B59 2600    MVI    H,0       ;HIGH ORDER BYTE ZERO.
4B5B C9      RET     ;RETURN TO CALLER.
4B5C 210100 NOTRAN: LXI    H,1    ;SET HL TO ONE.
4B5F 09      DAD     B        ;ADD SEC NMBR TO HL.
4B60 C9      RET     ;RETURN TO CALLER.
```

; FORMAT A DISK TRACK ROUTINE *

```
4B61 3E01    FORMAT: MVI    A,DC$SIN    ;SWITCH DD INTO SYSTEM.
4B63 D343    OUT     D$PORT    ;ISSUE HARDWARE CMND.
4B65 3E03    MVI    A,DC$MB1   ;SELECT DD BANK 1.
4B67 D343    OUT     D$PORT    ;ISSUE HARDWARE CMND.
4B69 010001 LXI    B,FMT$SZ   ;FORMAT PROG SIZE.
4B6C 2A4000 LHL    D$ADDR     ;DD SYSTEM ADDRESS.
4B6F 110003 LXI    D,DD$FBF   ;DD FORMAT BUF OFFSET.
4B72 19      DAD     D        ;HL NOW DD FBUF ADDR.
4B73 EB      XCHG    ;DE NOW DD FBUF ADDR.
```

```

4B74 2A604C      LHLD    BT$DMA      ;FORMAT PROGRAM ADDR.
4B77 CD414C      CALL    BLOCK       ;BLOCK MOVE ROUTINE.
4B7A 3E01        MVI     A,DC$MBO    ;RESELECT DD BANK 0.
4B7C D343        OUT     D$PORT      ;ISSUE TO DD HARDWARE.
4B7E 3E03        MVI     A,DC$FMT    ;LOAD FORMAT TRK CMND.
4B80 CD8A4B      CALL    DSK$EX      ;CALL DISK EXECUTIVE.
4B83 CD444B      CALL    DSK$OK      ;SWITCH DD MEM OUT.
4B86 3A5F4C      LDA     BT$STS      ;LOAD FORMAT STATUS.
4B89 C9          RET                ;FORMAT EXIT.

```

```

;*****
; DOUBLE D EXECUTION SUBROUTINE *
;*****

```

```

;***** ( COMMAND BLOCK TO DOUBLE D AND EXEC )*****

```

```

4B8A 32584C      DSK$EX: STA     BT$CMD      ;STORE DCM COMMAND.
4B8D 010700      LXI     B,7          ;NMBR BYTE TO MOVE.
4B90 117003      LXI     D,DD$CBT     ;COMMAND BYTE OFFSET.
4B93 2A4000      LHLD   D$ADDR       ;DD SYS ADDRESS.
4B96 19          DAD     D            ;HL NOW PTS CMND BLK.
4B97 EB          XCHG                    ;NOW ADDR IN DE.
4B98 21584C      LXI     H,BT$CMD     ;BIOS CMND BLOCK.
4B9B CD414C      CALL    BLOCK       ;PERFORM BLOCK MOVE.
4B9E 3E02        MVI     A,DC$INT    ;LOAD DD INTERRUPT.
4BA0 D343        OUT     D$PORT      ;ISSUE DD INTERRUPT.

```

```

;***** ( WAIT FOR DOUBLE D HALT )*****

```

```

4BA2 3A4200      LDA     D$HALT      ;LOAD HALT BIT MASK.
4BA5 47          MOV     B,A         ;MASK IN B REGISTER.
4BA6 DB43      DSK$WT: IN     D$PORT      ;READ DD STATUS.
4BA8 A0          ANA     B           ;TEST HALT* FLAG.
4BA9 C2A64B      JNZ    DSK$WT      ;TEST UNTIL HALTED.

```

```

;***** ( GET DOUBLE D STATUS )*****

```

```

4BAC 3E01        MVI     A,DC$SIN    ;SWITCH DD INTO SYS.
4BAE D343        OUT     D$PORT      ;ISSUE HARDWARE CMND.
4BB0 EB          XCHG                    ;EXCHANGE SRC/DSTN.
4BB1 7E          MOV     A,M         ;STATUS INTO A REG.
4BB2 12          STAX   D            ;STORE STATUS BYTE.
4BB3 A7          ANA     A           ;TEST FOR ERRORS.
4BB4 C9          RET                ;RETURN TO CALLER.

```

```

;*****
; LOG-ON - SET DISK PARAMETER BLOCK *
;*****

```

```

;***** ( CHECK IF LOG-ON REQUESTED )*****

```

```

4BB5 3A644C      LOG$ON: LDA     LOG$RQ    ;CHECK LOG REQUEST.
4BB8 E601        ANI     001H        ;LOG ON BIT TEST.
4BBA C2444B      JNZ    DSK$OK      ;RETURN, NO LOG-ON.

```

```

;***** ( READ IDENTITY SECTOR )*****

```

```

4BBD 22624C      SHLD   DT$PTR       ;STORE DRV TBL PNTR.
4BC0 3E01        MVI     A,DC$SIN    ;SWITCH DD INTO SYS.
4BC2 D343        OUT     D$PORT      ;ISSUE HARDWARE CMND.

```

```

4BC4 3E00          MVI      A,DC$LOG          ;LOAD DCM LOG-ON CMND.
4BC6 CD8A4B       CALL     DSK$EX            ;PERFORM DISK OP.
4BC9 CAD24B       JZ       LOG$CK           ;GO TO LOGON ERROR.
4BCC 210000       LXI     H,0                ;ERROR, BAD LOG ON.
4BCF C34A4B       JMP     DSK$ER            ;BIOS EXIT.
;
;***** ( CHECK FOR JADE ID )*****
4BD2 118003       LOG$CK: LXI     D,DD$BUF     ;DD BUFFER OFFSET.
4BD5 2A4000       LHL     D,$ADDR           ;DD SYS ADDRESS.
4BD8 19           DAD     D                 ;HL NOW PNTS BUFFER.
4BD9 11E14C       LXI     D,JADEID          ;DE PNTS BIOS ID.
4BDC 0608          MVI     B,ID$SIZE         ;SET LABEL SIZE.
4BDE 1A13       LOG$ID: LDAX   D!           INX D ;GET LABEL CHARACTER.
4BE0 BE23          CMP     M!           INX H ;DOES ID SECTOR MATCH.
4BE2 C20F4C       JNZ     LG3740           ;ASSUME DISKETTE 3740.
4BE5 05           DCR     B               ;DECREMENT COUNT.
4BE6 C2DE4B       JNZ     LOG$ID           ;CHECK IF ANOTHER CHR.
;
;***** ( DISKETTE CONTAINS ID )*****
4BE9 CD2E4C       CALL    TRNONE           ;ASSUME DDENS.
4BEC CD364C       CALL    DPB$AD           ;GET DPB ADDR IN DE.
4BEF 01A003       LXI     B,DD$DPB         ;DPB ADDR OFFSET.
4BF2 2A4000       LHL     D,$ADDR           ;DD SYSTEM ADDRESS.
4BF5 09           DAD     B                 ;HL NOW AT ID DPB.
4BF6 010F00       LXI     B,DPB$SZ         ;DPB SIZE IN BYTES.
4BF9 CD414C       CALL    BLOCK            ;MOVE INTO DPB.
4BFC 11B103       LXI     D,DD$DDF         ;ID DTA DNS OFFSET.
4BFF 2A4000       LHL     D,$ADDR           ;DD SYSTEM ADDR.
4C02 19           DAD     D                 ;HL POINTS FLAGS.
4C03 7E           MOV     A,M              ;LOAD FLAGS.
4C04 E604          ANI     04H              ;TEST DATA DENSITY.
4C06 CC244C       CZ      TR3740           ;IF 0 USE 3740 TRN.
4C09 2A624C       LHL     DT$PTR           ;RELOAD POINTER.
4C0C C3444B       JMP     DSK$OK           ;EXIT BIOS JUMP.
;
;***** ( ASSUME 3740 DISKETTE )*****
4C0F CD244C       LG3740: CALL    TR3740     ;SET SECTOR TRANSLATE.
4C12 CD364C       CALL    DPB$AD           ;SET REGISTER DE.
4C15 010F00       LXI     B,DPB$SZ         ;DPB SIZE IN BYTES.
4C18 217F4C       LXI     H,SD$PBK         ;ADDRESS OF BLK IMAGE.
4C1B CD414C       CALL    BLOCK            ;MOVE INTO DPB.
4C1E 2A624C       LHL     DT$PTR           ;RELOAD POINTER.
4C21 C3444B       JMP     DSK$OK           ;EXIT BIOS JUMP.
;
;***** ( SET 3740 SECTOR TRANSLATION )*****
4C24 11654C       TR3740: LXI     D,SDTRAN    ;SECTOR TRAN TBL ADDR.
4C27 2A624C       LHL     DT$PTR           ;ADDR DISK PARA HDR.
4C2A 73           MOV     M,E              ;LOW ORDER ADDR.
4C2B 23           INX     H                 ;POINT NEXT BYTE.
4C2C 72           MOV     M,D              ;HIGH ORDER ADDR.
4C2D C9           RET                      ;RETURN TO LOG USER.
;
;***** ( SET NO SECTOR TRANSLATION )*****
4C2E AF          TRNONE: XRA     A         ;ZERO A REGISTER.
4C2F 2A624C       LHL     DT$PTR           ;ADDR OF PARA HDR.

```

```

4032 77          MOV      M,A          ;ZERO LOW ORDER ADDR.
4033 23          INX      H            ;NEXT BYTE.
4034 77          MOV      M,A          ;ZERO HIGH BYTE.
4035 09          RET                    ;RETURN TO LOG USER.

```

```

;***** ( GET DRIVE PARA BLK ADDR )*****

```

```

4036 2A624C     DPB$AD: LHL D      DT$PTR      ;ADDR DISK PARA HDR.
4039 110A00     LXI      D,10          ;DPB TBL PNTR OFFSET.
403C 19         DAD      D            ;NOW AT DPB PNTR.
403D 5E         MOV      E,M          ;LOW ORDER ADDR.
403E 23         INX      H            ;NEXT BYTE.
403F 56         MOV      D,M          ;HIGH ORDER ADDR.
4040 09         RET                    ;RETURN TO LOG USER.

```

```

;*****
; BLOCK MOVE SUBROUTINE - Z80 LDIR WILL FUNCTION HERE *
;*****

```

```

4041 7E23     BLOCK: MOV      A,M!      INX H      ;GET EACH BYTE.
4043 1213     STAX      D!          INX D      ;STORE EACH BYTE.
4045 0B78B1   DCX B!      MOV A,B!      ORA C      ;DEC LENGTH (MAX 64K).
4048 C2414CC9 JNZ        BLOCK!      RET          ;FINISH BLOCK AND RET.

```

```

;*****
; MESSAGE DISPLAY ROUTINE - HL REG POINTS TO STRING *
;*****

```

```

404C 7E      MSG$OT: MOV      A,M          ;LOAD CHARACTER/BYTE.
404D FE24C8  CPI      EOM!      RZ          ;EXIT IF TERMINATOR.
4050 4FC00CF0 MOV C,A!      CALL   CNS$OT      ;DISPLAY CHARACTER.
4054 23C34C4C INX H!      JMP      MSG$OT      ;REPEAT FOR NEXT BYTE.

```

```

;*****
; DOUBLE D - DCM COMMAND BLOCK BUFFER *
;*****

```

```

4058 00     BT$CMD: DB      0          ;DCM COMMAND.
4059 00     BT$DRV: DB      0          ;DRIVE NUMBER.
405A 00     BT$TRK: DB      0          ;TRACK NUMBER.
405B 00     BT$SEC: DB      0          ;SECTOR NUMBER.
405C 00     BT$SPO: DB      0          ;SPARE BYTE 0.
405D 00     BT$CHR: DB      0          ;LIST CHARACTER.
405E 00     BT$MOD: DB      00000000B ;MODE CONTROLS.
405F 00     BT$STS: DB      0          ;COMMAND STATUS.

```

```

;*****
; BIOS VARIABLE STORAGE *
;*****

```

```

4060 0000     BT$DMA: DW      0          ;SYSTEM TRANSFER ADDR.
4062 0000     DT$PTR: DW      0          ;DRIVE TABLE POINTER.
4064 00       LOG$RQ: DB      0          ;LOG ON REQUEST REG.

```

```

;*****
; DOUBLE D - MEMORY ASSIGNMENTS (40-4F HEX) *
;*****

```

```

0040 =      D$ADDR EQU      0040H      ;DD SYSTEM WINDOW ADDR POINTER.
0042 =      D$HALT EQU      0042H      ;DD HALT STATUS MASK LOCATION.

```

```
*****
; 3740 FORMAT PARAMETERS ***** CP/M SINGLE DENSITY *
*****
```

```
***** ( SINGLE DENSITY CP/M SECTORING )*****
```

```
4C65 01070D1319SDTRAN: DB      01H,07H,0DH,13H,19H,05H,0BH,11H,17H,03H
4C6F 090F150208      DB      09H,0FH,15H,02H,08H,0EH,14H,1AH,06H,0CH
4C79 1218040A10      DB      12H,18H,04H,0AH,10H,16H
```

```
***** ( DEFAULT DISK PARAMETER BLOCK )*****
```

```
4C7F 1A00      SD$PBK: DW      26      ;SECTORS PER TRACK.
4C81 03        DB      3        ;BLOCK SHIFT FACTOR.
4C82 07        DB      7        ;BLOCK MASK.
4C83 00        DB      0        ;NULL MASK.
4C84 F200      DW      242      ;DISK SIZE - 1.
4C86 3F00      DW      63      ;DIRECTORY MAX.
4C88 C0        DB      1100000B ;ALLOC 0.
4C89 00        DB      0        ;ALLOC 1.
4C8A 1000      DW      16      ;CHECK SIZE.
4C8C 0200      DW      2        ;TRACK OFFSET.
```

```
*****
; ZERO PAGE IMAGE - BLOCK MOVED TO BASE PAGE *
*****
```

```
4C8E C3034A    BS$IMG: JMP      BIOS+03H      ;WARM BOOT VECTOR.
4C91 00        IO$IMG: DB      IOBYTE      ;I/O BYTE IMAGE.
4C92 00        DF$IMG: DB      DF$DRV      ;DEFAULT DRIVE IMG.
4C93 C3063C    JMP      BDOS+06H      ;BDOS CALL VECTOR.
```

```
*****
; MESSAGES *
*****
```

```
4C96 0D0A0D0A  MSG$S0: DB      CR,LF,CR,LF
4C9A 4A41444520 DB      'JADE COMPUTER PRODUCTS',CR,LF
4CB2 3230      DB      '0' + CPM$NK / 10,'0' + CPM$NK MOD 10
4CB4 4B2043502F DB      'K CP/M 2.2 DDBIOS2',CR,LF,CR,LF,EOM
```

```
4CCB 0D0A434350MSG$LE: DB      CR,LF,'CCP/BDOS LOAD ERROR',EOM
```

```
000A =      LF      EQU      00AH      ;ASCII LINE FEED.
000D =      CR      EQU      00DH      ;CARRIAGE RETURN.
0024 =      EOM     EQU      '$'      ;END OF STRING.
001A =      CNTL$Z EQU      01AH      ;CONTROL-Z (EOF).
```

```
*****
; ID LABEL DEFINITIONS *
*****
```

```
4CE1 4A61646520JADEID: DB      'JADE DD '      ;ID LABEL.
0008 =      ID$SIZE EQU      $-JADEID      ;LABEL SIZE.
```

```
*****
; DRIVE PARAMETER HEADER AREA *
*****
```

```

4CE9 0000      D0$DPH: DW      0          ;SECTOR TRAN TBL.
4CEB 0000      DW      0          ;SCRATCH AREA.
4CED 0000      DW      0          ;SCRATCH AREA.
4CEF 0000      DW      0          ;SCRATCH AREA.
4CF1 004E      DW      DIR$BF      ;DIRECTORY BUFFER.
4CF3 804E      DW      D0$DPB      ;DRIVE PARAM BLK.
4CF5 B04E      DW      D0$CHK      ;DRIVE CHANGE BLK.
4CF7 8F4E      DW      D0$ALL      ;DRIVE ALLOCATION.

                                IF      (1-N$DRVS) SHR 15      ;TEST SIGN BIT.
4CF9 00000000D1$DPH: DW      0,0,0,0
4D01 004ED04E  DW      DIR$BF,D1$DPB
4D05 004FDF4E  DW      D1$CHK,D1$ALL
                                ENDIF

                                IF      (2-N$DRVS) SHR 15
D2$DPH: DW      0,0,0,0
                                DW      DIR$BF,D2$DPB
                                DW      D2$CHK,D2$ALL
                                ENDIF

                                IF      (3-N$DRVS) SHR 15
D3$DPH: DW      0,0,0,0
                                DW      DIR$BF,D3$DPB
                                DW      D3$CHK,D3$ALL
                                ENDIF

;*****
; BIOS PROGRAM AREA REMAINING *
;*****

4E00 =          BIOS$U EQU      BIOS + K$B      ;BEGIN SCRATCH AREA.
00F7 =          BIOS$F EQU      BIOS$U - $      ;NUMBER OF BYTES FREE.

                                IF      BIOS$F SHR 15      ;TEST PROG OVERFLOW.
ERROR      EQU      1/0          ;GENERATE ERROR.
                                ENDIF

4D09          F$AREA: DS          BIOS$F          ;USE UP PROG AREA.

;*****
; DIRECTORY BUFFER AREA - BEGINNING OF SCRATCH AREA *
;*****

4E00          DIR$BF: DS          SEC$SZ

;*****
; DRIVE TABLE ENTRY - SIZES *
;*****

000F =          DPB$SZ EQU      15          ;PARAMETER BLOCK SIZE.
0021 =          ALL$SZ EQU      33         ;ALLOCATION INFO SIZE.
0020 =          CHK$SZ EQU      32         ;CHANGED DISK SCRATCH.

;*****
; DRIVE TABLES - SCRATCH AREAS *
;*****

4E80          D0$DPB: DS          DPB$SZ      ;DISK PARAMETER BLOCK.
4E8F          D0$ALL: DS          ALL$SZ      ;DISK ALLOCATION INFO.

```

```
4EB0          D0$CHK: DS          CHK$SZ          ;DISK CHANGED SCRATCH.
              IF          (1-N$DRVS) SHR 15
4ED0          D1$DPB: DS          DPB$SZ          ;DISK PARAMETER BLOCK.
4EDF          D1$ALL: DS          ALL$SZ          ;DISK ALLOCATION INFO.
4F00          D1$CHK: DS          CHK$SZ          ;DISK CHANGED SCRATCH.
              ENDIF

              IF          (2-N$DRVS) SHR 15
          D2$DPB: DS          DPB$SZ          ;DISK PARAMETER BLOCK.
          D2$ALL: DS          ALL$SZ          ;DISK ALLOCATION INFO.
          D2$CHK: DS          CHK$SZ          ;DISK CHANGED SCRATCH.
              ENDIF

              IF          (3-N$DRVS) SHR 15
          D3$DPB: DS          DPB$SZ          ;DISK PARAMETER BLOCK.
          D3$ALL: DS          ALL$SZ          ;DISK ALLOCATION INFO.
          D3$CHK: DS          CHK$SZ          ;DISK CHANGED SCRATCH.
              ENDIF
```

```
4F20          END
*
```


DBSLDR
ADDR CODE

STMT SOURCE STATEMENT

```
0001          NAME      DBSLDR
0003 ; *****
0004 ;
0005 ;          PROGRAM ID:      DDBIOS LOADER
0006 ;
0007 ;          VERSION:        2.2          RELEASE 2
0008 ;
0009 ; *****
0010 ;
0011 ;          PROPERTY OF:      JADE COMPUTER PRODUCTS
0012 ;                          4901 W. ROSECRANS BLVD.
0013 ;                          HAWTHORNE, CALIFORNIA
0014 ;                          90250, U.S.A.
0015 ;
0016 ; *****
0017 ; THE BIOS LOADER IS READ INTO THE DCM SECTOR BUFFER *
0018 ; AFTER DCM HAS INITIALIZED. THE BIOS LOADER PROGRAM *
0019 ; IS THEN EXECUTED WHICH READS THE DDBIOS MODULE *
0020 ; INTO BANK 1. THE COMMAND BLOCK (IN DCM) IS SET TO *
0021 ; INDICATE DDBIOS MODULE SIZE AND THE SYSTEM LOAD *
0022 ; ADDRESS. THE BIOS LOADER PROGRAM IS GENERATED BY *
0023 ; MOVCPM.COM AS THE COLD START LOADER (900-97F HEX). *
0024 ; THIS MODULE IS PROVIDED FOR REFERENCE PURPOSES. *
0025 ; *****
0026 ; THE DDBIOS LOADER IS COMPATABLE WITH REV B AND C *
0027 ; DOUBLE D CONTROLLER BOARDS. IT IS COMPATABLE WITH *
0028 ; FD1791-01 / FD1793-01. IT WILL ALSO FUNCTION WITH *
0029 ; THE CURRENT FD179X-02 SERIES.
0030 ; *****
```

```

ADDR  CODE      STMT SOURCE STATEMENT

0032 ;*****
0033 ; CONTROLLER PORT ASSIGNMENTS *
0034 ;*****
0035
>0000 0036 BL$STS EQU 000H ;BOARD STATUS
>0000 0037 BL$CTL EQU 000H ;BOARD CONTROLS
>0004 0038 WD$CMD EQU 004H ;179X-02 COMMAND REGISTER
>0004 0039 WD$STS EQU 004H ;179X-02 STATUS REGISTER
>0006 0040 WD$SEC EQU 006H ;179X-02 SECTOR REGISTER
>0007 0041 WD$DTA EQU 007H ;179X-02 DATA REGISTER
>0010 0042 XP$MTO EQU 010H ;MOTOR TIME OUT
>0040 0043 XP$MTX EQU 040H ;MOTOR TIME EXTEND
>0080 0044 XP$DSH EQU 080H ;DATA SYNC HOLD
0045
0046 ;*****
0047 ; 179X-02 COMMAND AND MASK. *
0048 ;*****
0049
>0088 0050 DC$RDS EQU 1000100B ;READ SECTOR.
>009C 0051 DM$RER EQU 10011100B ;READ ERROR MASK.
0052
0053 ;*****
0054 ; SYSTEM ASSIGNMENTS *
0055 ;*****
0056
>0014 0057 NMBR$K EQU 20 ;SYSTEM SIZE IN K.
>0400 0058 LNG$1K EQU 1024 ;TOTAL BYTES IN 1K.
>5000 0059 CPM$SZ EQU NMBR$K*LNG$1K ;TOTAL SYSTEM BYTES.
>0600 0060 BIOS$S EQU LNG$1K*3/2 ;BIOS ALLOCATED SIZE.
>4A00 0061 BIOS$A EQU CPM$SZ-BIOS$S ;BIOS LOAD ADDRESS.
0062
0063 ;*****
0064 ; INTERNAL MEMORY ASSIGNMENTS *
0065 ;*****
0066
>1000 0067 BANK$0 EQU 1000H ;LOWER BANK ADDRESS.
>0400 0068 BANK$L EQU 0400H ;1K BANK LENGTH.
>1400 0069 BANK$1 EQU BANK$0+BANK$L ;UPPER BANK ADDRESS.
>1370 0070 IO$BLK EQU BANK$0+0370H ;I/O BLOCK ADDRESS.
>1377 0071 CB$STS EQU IO$BLK+0007H ;COMMAND STATUS BYTE.
>1378 0072 CW$LAD EQU IO$BLK+0008H ;BIOS LOAD ADDR LOC.
>137A 0073 CW$LNG EQU IO$BLK+000AH ;BIOS LOAD LENGTH LOC.
>1380 0074 SEC$BF EQU BANK$0+0380H ;SECTOR BUFFER AREA.
0075
0076 ;*****
0077 ; BIOS PROGRAM LINKAGE. *
0078 ;*****
0079
>0004 0080 SEC$BG EQU 4 ;FIRST BIOS SECTOR.
>0008 0081 SEC$NM EQU 8 ;NUMBER OF SECTORS.
>000B 0082 SEC$EX EQU SEC$BG+SEC$NM-1 ;LAST BIOS SECTOR.
0083
0084 ;*****

```

```

0086 ;*****
0087 ; ASSEMBLER DIRECTIVES *
0088 ;*****
0089
>1380 0090 PSECT ABS ;ABSOLUTE ADDRESSING.
0091 ORG SEC$BF ;PROGRAM START POINT.
0092
0093 ;*****
0094 ; INITIALIZE BIOS READ OPERATION *
0095 ;*****
0096
1380 210004 0097 BEGIN: LD HL,LNG$1K ;BIOS LOAD LENGTH.
1383 227A13 0098 LD (CW$LNG),HL ;LOAD LENGTH SET.
1386 21004A 0099 LD HL,BIOS$A ;BIOS SYSTEM ADDR.
1389 227813 0100 LD (CW$LAD),HL ;LOAD ADDRESS SET.
138C 210014 0101 LD HL,BANK$1 ;BIOS LOAD POINT.
0102
0103 ;*****
0104 ; SET-UP FOR EACH READ SECTOR COMMAND *
0105 ;*****
0106
138F FD21A813 0107 RD$SEC: LD IY,RD$TST ;SET NMI VECTOR.
1393 3AC413 0108 LD A,(SECTOR) ;FIRST BIOS SECTOR.
1396 A9 0109 XOR C ;INVERT (1791-01).
1397 D306 0110 OUT (WD$SEC),A ;SET 179X-02 SEC REG.
1399 3E88 0111 LD A,DC$RDS ;READ SECTOR CMND.
139B A9 0112 XOR C ;INVERT (1791-01).
139C D304 0113 OUT (WD$CMD),A ;ISSUE 179X-02 COMMAND.
0114
0115 ;*****
0116 ; READ SECTOR OPERATION *
0117 ;*****
0118
139E DB80 0119 RD$BYT: IN A,(XP$DSH) ;WAIT FOR DATA.
13A0 DB07 0120 IN A,(WD$DTA) ;INPUT INV DATA.
13A2 A9 0121 XOR C ;INVERT (1791-01).
13A3 77 0122 LD (HL),A ;STORE DCM BYTE.
13A4 23 0123 INC HL ;INCREMENT POINTER.
13A5 C39E13 0124 JP RD$BYT ;REPEAT OPERATION.
0125
0126 ;*****

```

```

ADDR CODE      STMT SOURCE STATEMENT
0128 ;*****
0129 ; CHECK READ SECTOR STATUS, REPEAT UNTIL BIOS LOADED *
0130 ;*****
0131
13A8 E69C      0132 RD#TST: AND      DM#RER      ;TEST FOR ERRORS.
13AA 200D      0133          JR          NZ,ERRORS    ;ERROR DETECTED.
13AC 3AC413    0134          LD          A,(SECTOR)    ;GET SECTOR NMBR.
13AF FE0B      0135          CP          SEC#EX        ;CHECK IF LAST SEC.
13B1 280F      0136          JR          Z,FINISH      ;GO IF FINISHED.
13B3 3C        0137          INC         A             ;INCREMENT.
13B4 32C413    0138          LD          (SECTOR),A    ;STORE SECTOR NUMBER.
13B7 18D6      0139          JR          RD#SEC        ;READ NEXT SECTOR.
0140
0141 ;*****
0142 ; READ ERROR HAS BEEN DETECTED *
0143 ;*****
0144
13B9 327713    0145 ERRORS: LD      (CB#STS),A    ;DISPLAY ERROR STATUS.
13BC AF        0146          XOR         A             ;ZERO A REGISTER.
13BD D300      0147          OUT        (BL#CTL),A    ;DESELECT DRIVE.
13BF DB10      0148          IN         A,(XP#MTO)    ;MOTOR OFF!
13C1 76        0149          HALT                    ;TERMINATE.
0150
0151 ;*****
0152 ; BIOS SECTOR HAVE BEEN LOADED *
0153 ;*****
0154
13C2 FB        0155 FINISH: EI                    ;ENABLE INTERRUPTS.
13C3 76        0156          HALT                    ;SHUTDOWN BOARD.
0157
0158 ;*****
0159 ; SECTOR NUMBER STORAGE *
0160 ;*****
0161
13C4 04        0162 SECTOR: DEFB    SEC#BG      ;SECTOR COUNTER.
0163
0164 ;*****
0165          END

```

```

; *****
;
; PROGRAM ID: DOUBLE D BOOTSTRAP (DDBOOT)
;
; *****
;
; VERSION: CP/M 2.2 RELEASE 2B
;
; BOOT IN 2708: P/N SFC-58001200E
;
; *****
;
; DISTRIBUTOR: JADE COMPUTER PRODUCTS
; 4901 W. ROSECRANS BLVD.
; HAWTHORNE, CALIFORNIA
; 90250, U.S.A.
;
; *****SK*****

```

```

; *****
; THE DOUBLE D BOOTSTRAP PROGRAM (DDBOOT) IS USED TO
; INITIATE THE SYSTEM TRACKS LOAD SEQUENCE FROM DRIVE
; A (OR 0) AND TO PROVIDE CONSOLE I/O SUBROUTINES FOR
; THE DISK OPERATING SYSTEM (CP/M). THIS PROM SHOULD
; BE LOCATED AT F000 HEX. THE SOURCE CODE FOR DDBOOT
; CAN BE ASSEMBLED WITH DIGITAL RESEARCH ASSEMBLER
; ASM.COM. MACHINE CODE IS 8080/8085/Z80 COMPATIBLE
; *****

```

```

; *****
; DDBOOT INJECTION MODULE IS COMMAND COMPATIBLE WITH
; THE FOLLOWING WESTERN DIGITAL CONTROLLER CHIPS.
; DOUBLE D USER SWITCH 0 (U0 OR R0) MUST BE SET TO
; INDICATE THE CONTROLLER CHIP DATA BUS POLARITY.
; *****
; CONTROLLER IC USER SW0
; -----
; FD1791-02 (01) CLOSED
; FD1793-02 (01) OPENED
; FD1795-02 CLOSED
; FD1797-02 OPENED
; *****
; THE FD1795-02 AND FD1797-02 PROVIDE ENHANCED SINGLE
; DENSITY PERFORMANCE IN THAT THESE CHIPS ARE FULLY
; COMPATIBLE WITH FD1771-01 3740 FORMATS.
; *****

```

```

; *****
; CBIOS SCRATCH ***** SYSTEM MEMORY ALLOCATION
; *****
; ADDRESS NAME FUNCTION
; -----
; 0040-41 D$ADDR ADDRESS POINTER TO DOUBLE D
; 0042 D$MASK STATUS PORT HALT BIT MASK
; 0043 D$TEMP TEMPORARY, INSERT DISK MSG
; *****

```

MODIFICATION MAY BE NEEDED. THIS PROM PROVIDES THE *
CONSOLE STATUS, INPUT, AND OUTPUT SUBROUTINES. MANY *
END USER SYSTEMS REQUIRE UART / USART AND BAUD RATE *
GENERATOR INITIALIZATION. THESE ROUTINES MAY NEED *
TO BE PATCHED TO PROVIDE FOR PROPER CONSOLE LINKAGE *
PATCHING MAY ALSO BE DONE FOR SOME DISK DRIVES. *

FROM LOCATIONS THAT MAY NEED PATCHING *

SYSTEM INITIALIZATION - UART/USART, BAUD RATE, AND *
AND POWER ON JUMP CIRCUITRY MAY REQUIRE SOFTWARE *
INITIALIZATION. A PATCH AREA IS RESERVED AT "INIT" *

CONSOLE STATUS CHECK - RETURNS KEYBOARD STATUS TO *
CP/M OPERATING SYSTEM. THIS SUBROUTINE MUST USE *
THE CORRECT PORT ADDRESS AND TEST PROPER STATUS BIT.*
ROUTINE IS LABELED "CNS\$CK" AND CONTAINS PATCH AREA.*

CONSOLE INPUT - RETURNS KEYBOARD CHARACTER TO CP/M *
OPERATING SYSTEM. THIS ROUTINE MUST ALSO USE THE *
CORRECT PORT ADDRESSING. THIS ROUTINE IS LABELED *
"CNS\$IN" AND CONTAINS A PATCH AREA. *

CONSOLE OUTPUT - DISPLAYS CHARACTER TO CONSOLE UNIT.*
THIS ROUTINE MUST USE THE CORRECT PORT ADDRESS FOR *
BOTH THE OUTPUT STATUS AND OUTPUT DATA PORTS. THIS *
ROUTINE MUST TEST THE CONSOLE OUTPUT STATUS BIT. *
THIS ROUTINE IS LABELED "CNS\$OT" AND CONTAINS A *
PATCH AREA. *

BOARD REVISION - NOTE SOURCE FILE HAS CONDITIONAL *
STATEMENTS FOR BOARD REVISION. REVISION B BOARDS *
MAY BE MODIFIED TO ACT AS REV-C (CALL JADE) OR *
DDBOOT MAY BE PATCHED. PROM IS FOR REV-C. *

Table with 6 columns: ADDRESS, NAME, FUNCTION, REV-C, REV-B. Rows include DS\$ASW, D\$BASE, DS\$HLT.

** SHOULD BE "E0" IF MA10 JUMPER IS INSTALLED. *

DISK DRIVES - DDBOOT IS USING A 10 MILLISECOND STEP *
WHICH WILL HOME THE R/W HEAD ON MOST DRIVES. IF THE *
USER DRIVE IS MUCH FASTER (SHUGART SA850 OR SIEMENS *
FD100-8D) THEN THE STEP RATE CONSTANT MAY BE PATCHED*
NOTE: A DDBOOT PROM PATCHED FOR FAST DRIVES WILL *
NOT FUNCTION PROPERLY IF LATER USED WITH SLOWER *
DRIVES. SLOWER STEPS SHOULD ALWAYS WORK. A DELAY *
BEFORE READING HAS BEEN PROVIDED FOR HEAD LOAD TIME *
AND CAN BE USED FOR DRIVE-MOTOR START UP TIME WHEN *
THE DRIVE MOTORS ARE CONTROLLED BY THE DOUBLE-D. *

Table with 3 columns: ADDRESS, NAME, FUNCTION. Header row only.

```

; F1D1-F1D2  IM$TM$STP  STEP TIME          *
; F1FA-F1FB  IM$TM$DBR  DELAY BEFORE READ  *
; *****

; *****
; DOUBLE D BOOTSTRAP SYSTEM ADDRESS          *
; *****

F000 =      PROM$ADDR      EQU      0F000H  ;DDBOOT SYSTEM ADDRESS.

; *****
; SET DOUBLE D SYSTEM PORT ADDRESS          *
; *****

0043 =      D$PORT      EQU      043H    ;DOUBLE D PORT ADDRESS.

; *****
; SET USER DOUBLE D BOARD REVISION          *
; *****

0001 =      TRUE      EQU      1        ;SET TRUE TO LOGIC ONE.
0000 =      FALSE     EQU      0        ;SET FALSE TO LOGIC ZERO
0000 =      REV$B     EQU      FALSE    ;SET TRUE FOR REV B BOARDS.
0001 =      REV$C     EQU      TRUE     ;SET TRUE FOR REV C BOARDS.
0000 =      MA10      EQU      FALSE    ;TRUE IF MA10 JUMPED (REV-B).

; *****
; DEFINE HALT MASK AND BASE ADDRESS OF DOUBLE D          *
; *****

                IF      REV$B AND NOT MA10
DS$HLT EQU      002H    ;STATUS PORT HALT INDICATOR.
DS$ASW EQU      00CH    ;STATUS PORT ADDR SW MASK.
D$BASE EQU      0E400H  ;SYSTEM WINDOW BASE ADDRESS.
                ENDIF

                IF      REV$B AND MA10
DS$HLT EQU      002H    ;STATUS PORT HALT INDICATOR.
DS$ASW EQU      00CH    ;STATUS PORT ADDR SW MASK.
D$BASE EQU      0E000H  ;SYSTEM WINDOW BASE ADDRESS.
                ENDIF

0001 =      DS$HLT     EQU      001H    ;STATUS PORT HALT INDICATOR.
000E =      DS$ASW     EQU      00EH    ;STATUS PORT ADDR SW MASK.
E000 =      D$BASE     EQU      0E000H  ;SYSTEM WINDOW BASE ADDRESS.
                ENDIF

; *****
; BOOTSTRAP LINKAGE ADDRESS.          *
; *****

0080 =      BSTACK    EQU      0080H    ;BOOTSTRAP TOP OF STACK.
0040 =      D$ADDR     EQU      0040H    ;DOUBLE D ADDRESS POINTER.
0042 =      D$MASK     EQU      0042H    ;DOUBLE D HALT BIT ADDR.
0043 =      D$TEMP     EQU      0043H    ;DDBOOT TEMPORARY LOCATION.
0377 =      BL$DCS     EQU      0377H    ;DCM DISK CONTROLLER STATUS.
0378 =      BL$ADR     EQU      0378H    ;DCM LOAD AND JUMP ADDR FNTR.
037A =      BL$BSZ     EQU      037AH    ;DCM BLOCK LOAD SIZE.

```

```

0080 =          BL$DNR EQU          0080H      ;DRIVE NOT READY BIT.

;*****
; DOUBLE D HARDWARE COMMANDS
;*****

0080 =          DC$BGN EQU          080H      ;RESET Z80A AND EXECUTE.
0001 =          DC$MRQ EQU          001H      ;REQUEST MEMORY WINDOW.
0000 =          DC$MRT EQU          000H      ;RELEASE MEMORY WINDOW.
0001 =          DC$MBO EQU          001H      ;SELECT MEMORY BANK 0.
0003 =          DC$MB1 EQU          003H      ;SELECT MEMORY BANK 1.
0002 =          DC$EXC EQU          002H      ;ISSUE DOUBLE D INTERRUPT.

;*****
; ASSEMBLER DIRECTIVES
;*****

F000                                ORG          FROM$ADDR      ;MODULE ADDRESS.

;*****
; DDBOOT FUNCTIONS VECTOR TABLE
;*****

F000 C312F0          JMP          INIT          ;INITIALIZE AND BOOT.
F003 C33AF0          JMP          BOOT          ;REBOOT DISK SYSTEM.
F006 C3DAF0          JMP          CNS$CK          ;CONSOLE STATUS.
F009 C3F6F0          JMP          CNS$IIN          ;CONSOLE INPUT.
F00C C313F1          JMP          CNS$OT          ;CONSOLE OUTPUT.
F00F C332F1          JMP          MSG$OT          ;MESSAGE TO CONSOLE.

;*****
; INITIALIZE SYSTEM HARDWARE - USER PATCH AREA
;*****

F012 00000000      INIT:      NOP!NOP!NOP!NOP          ;PATCH AREA.
F016 00000000          NOP!NOP!NOP!NOP          ;PATCH AREA.
F01A 00000000          NOP!NOP!NOP!NOP          ;PATCH AREA.
F01E 00000000          NOP!NOP!NOP!NOP          ;PATCH AREA.
F022 00000000          NOP!NOP!NOP!NOP          ;PATCH AREA.
F026 00000000          NOP!NOP!NOP!NOP          ;PATCH AREA.
F02A 00000000          NOP!NOP!NOP!NOP          ;PATCH AREA.
F02E 00000000          NOP!NOP!NOP!NOP          ;PATCH AREA.
F032 00000000          NOP!NOP!NOP!NOP          ;PATCH AREA.
F036 00000000          NOP!NOP!NOP!NOP          ;PATCH AREA.

;*****
; SET STACK AND DETERMINE CONTROLLER ADDRESS
;*****

F03A 318000      BOOT:      LXI          SP,BSTACK          ;SET STACK POINTER.
F03D DB43          IN          D,$PORT          ;INPUT STATUS PORT.
F03F E60E          ANI          DS$ASW          ;MASK FOR ADDR SWS.
F041 07          RLC          ;POSITION BITS.
F042 F&E0          ORI          D,$BASE SHR $          ;OR IN BASE ADDR.
F044 67          MOV          H,A          ;HIGH BYTE VALUE.
F045 2E00          MVI          L,0          ;LOW BYTE VALUE.
F047 224000      SHLD         D,$ADDR          ;STORE THE ADDRESS
F04A 3E01          MVI          A,DS$HLT          ;LOAD HALT BIT MASK.
F04C 324200      STA          D,$MASK          ;STORE FOR BIOS USE.
F04F 324300      STA          D,$TEMP          ;SET REPEAT FLAG NZ.

```



```

; *****
; INJECT BOOT MODULE INTO CONTROLLER
; *****

```

```

F052 3E01      INJECT: MVI      A,DC#MBO      ;REQUEST DD MEM BANK 0.
F054 D343      OUT        D$PORT      ;ISSUE COMMAND.
F056 01C600    LXI        B,IM$END-IM$BGN ;INJECTION MODULE SIZE.
F059 2A4000    LHL      D$ADDR      ;LOAD DOUBLE D ADDR.
F05C EB        XCHG        ;D$ADDR HL TO DE.
F05D 2189F1    LXI        H,IM$BGN      ;INJECTION MODULE ADDR.
F060 CDA7F0    CALL     BLOCK      ;BLOCK MOVE.

```

```

; *****
; RESET AND START THE DISK PROCESSOR
; *****

```

```

F063 3E80      MVI      A,DC#BGN      ;BEGIN DD PROCESSOR.
F065 D343      OUT        D$PORT      ;ISSUE COMMAND.
F067 E3        XTHL      ;ALLOW DOUBLE D TIME
F068 E3        XTHL      ;TO START UP.

```

```

; *****
; WAIT FOR TASK COMPLETION
; *****

```

```

F069 3A4200    LDA      D$MASK      ;HALT BIT MASK.
F06C 47        MOV      B,A          ;MOVE INTO B REG.
F06D DB43      WAIT:  IN        D$PORT ;INPUT DD STATUS.
F06F A0        ANA      B          ;TEST HALT* STATUS.
F070 C26DF0    JNZ     WAIT        ;WAIT TILL HALTED.

```

```

; *****
; SWITCH CONTROLLER MEMORY INTO SYSTEM BUS
; *****

```

```

F073 3E01      MVI      A,DC#MRQ      ;REQUEST MEM (BANK 0).
F075 D343      OUT        D$PORT      ;ISSUE COMMAND.

```

```

; *****
; CHECK FOR BOOTSTRAP MALFUNCTION
; *****

```

```

F077 2A4000    LHL      D$ADDR      ;CONTROLLER ADDRESS.
F07A 117703    LXI      D,BL$DCS     ;ERROR CODE IM$BGN.
F07D 19        DAD      D          ;SET HL POINTER.
F07E 7E        MOV      A,M          ;GET ERROR CODE.
F07F E680      ANI      BL$DNR      ;TEST DRIVE NOT READY.
F081 C2B4F0    JNZ     INSERT      ;IF DRIVE NOT READY.
F084 7E        MOV      A,M          ;GET ERROR CODE.
F085 A7        ANA      A          ;TEST REGISTER.
F086 C2C8F0    JNZ     BAD$LD      ;BAD LOAD.

```

```

; *****
; PERFORM BLOCK TRANSFER FROM DISK MEMORY
; *****

```

```

F089 2A4000    LHL      D$ADDR      ;CONTROLLER ADDRESS.
F08C 117803    LXI      D,BL$ADR     ;LOAD ADDRESS FNTR.
F08F 19        DAD      D          ;SET HL POINTER.

```

```

F090 5E      MOV      E,M      ;LOW ORDER ADDR.
F091 23      INX      H        ;INCREMENT HL.
F092 56      MOV      D,M      ;HIGH ORDER ADDR.
F093 23      INX      H        ;REQUIRES BL,BSZ NEXT.
F094 4E      MOV      C,M      ;LOW ORDER LENGTH.
F095 23      INX      H        ;INCREMENT HL.
F096 46      MOV      B,M      ;HIGH ORDER LENGTH.
F097 D5      PUSH     D        ;USE AS JUMP ADDR.
F098 3E03     MVI      A,DC$MB1  ;SWITCH TO MEM BANK 1.
F09A D343     OUT      D$PORT    ;ISSUE COMMAND.
F09C 2A4000   LHLD    D$ADDR     ;DOUBLE D MEM ADDRESS.
F09F CDA7F0   CALL    BLOCK      ;MOVE BIOS MODULE.

```

```

;*****
; TRANSFER CONTROL TO OPERATING SYSTEM *
;*****

```

```

FOA2 3E01     MVI      A,DC$MBO  ;SWITCH TO BANK.0
FOA4 D343     OUT      D$PORT    ;ISSUE COMMAND.
FOA6 C9      RET                ;GOTO BIOS COLD ENTRY.

```

```

;*****
; BLOCK MOVE SUBROUTINE (Z80 BLOCK MOVE REGISTERS) *
;*****

```

```

FOA7 7E      BLOCK: MOV      A,M      ;GET BYTE.
FOA8 23      INX      H        ;INC POINTER
FOA9 EB      XCHG                     ;GET DESTINATION.
FOAA 77      MOV      M,A      ;PUT BYTE.
FOAB 23      INX      H        ;INC POINTER
FOAC EB      XCHG                     ;GET SOURCE.
FOAD 0B      DCX      B        ;ONE LESS TO DO.
FOAE 78      MOV      A,B      ;GET HI COUNT.
FOAF B1      ORA      C        ;GET LO COUNT.
FOB0 C2A7F0   JNZ     BLOCK      ;FINISH LOADING.
FOB3 C9      RET

```

```

;*****
; DISK DRIVE IS NOT READY *
;*****

```

```

FOB4 3A4300   INSERT: LDA      D$TEMP  ;LOAD INIT FLAG.
FOB7 A7      ANA      A        ;TEST FOR INITIAL.
FOB8 CA52F0   JZ       INJECT      ;TRY BOOTING AGAIN.
FOBB AF      XRA      A        ;ZERO A REGISTER.
FOBC 324300   STA      D$TEMP  ;CLEAR INITIAL FLAG.
FOBF 2156F1   LXI      H,MSG$IN  ;INSERT MESSAGE ADDR.
FOC2 CD32F1   CALL    MSG$OT    ;OUTPUT MESSAGE.
FOC5 C352F0   JMP     INJECT      ;TRY BOOTING AGAIN.

```

```

;*****
; DOUBLE D BOOTSTRAP MALFUNCTION *
;*****

```

```

FOC8 324300   BAD$LD: STA      D$TEMP  ;STORE ERROR CODE.
FOCB 2171F1   LXI      H,MSG$ER  ;ERROR MESSAGE ADDEESS.
FOCE CD32F1   CALL    MSG$OT    ;DISPLAY MESSAGE.
FOD1 3A4300   LDA      D$TEMP  ;LOAD ERROR CODE.
FOD4 CD3EF1   CALL    HXB$OT    ;DISPLAY HEX BYTE.
FOD7 760000   HLT!NOP!NOP      ;HALT OR JUMP MONITOR.

```

```

;*****
;  CONSOLE INPUT AND OUTPUT
;*****
;
;  XXX$SP:  STATUS PORT ADDRESS
;  XXX$SB:  STATUS READY BIT
;  XXX$SI:  IF READY TRUE IS "1" USE "00" ELSE "FF"
;  XXX$DP:  DATA PORT ADDRESS
;
;*****

```

```

0000 =      CNI$SP  EQU      000H      ;INPUT STATUS PORT.
0002 =      CNI$SB  EQU      002H      ;INPUT STATUS BIT.
0000 =      CNI$SI  EQU      000H      ;INPUT STATUS INVERT.
0001 =      CNI$DP  EQU      001H      ;INPUT DATA PORT.

0000 =      CNO$SP  EQU      000H      ;OUTPUT STATUS PORT.
0004 =      CNO$SB  EQU      004H      ;OUTPUT STATUS BIT.
0000 =      CNO$SI  EQU      000H      ;OUTPUT STATUS INVERT.
0001 =      CNO$DP  EQU      001H      ;OUTPUT DATA PORT.

```

```

;*****
;  CONSOLE INPUT STATUS CHECK
;*****

```

```

FODA DB00      CNS$CK: IN      CNI$SP      ;INPUT STATUS PORT.
FODC EE00      XRI      CNI$SI      ;ADJUST POLARITY.
FODE E602      ANI      CNI$SB      ;TEST READY BIT.
FOE0 C8        RZ                ;ZERO IS NOT READY.
FOE1 3EFF      MVI      A,0FFH      ;SET CONSOLE READY.
FOE3 C9        RET                ;ONES INDICATE READY.

FOE4 000000000 DB      0,0,0,0,0,0  ;PATCHING AREA.
FOEA 000000000 DB      0,0,0,0,0,0  ;PATCHING AREA.
FOF0 000000000 DB      0,0,0,0,0,0  ;PATCHING AREA.

```

```

;*****
;  CONSOLE DATA INPUT
;*****

```

```

FOF6 CDDAF0    CNS$IN: CALL     CNS$CK      ;TEST INPUT READY.
FOF9 CAF6F0    JZ      CNS$IN      ;REPEAT TEST FOR RDY.
FOFC DB01      IN      CNI$DP      ;INPUT CONSOLE DATA.
FOFE E67F      ANI      07FH       ;SEVEN BITS OF ASCII.
F100 C9        RET                ;RETURN WITH DATA.

F101 000000000 DB      0,0,0,0,0,0  ;PATCHING AREA.
F107 000000000 DB      0,0,0,0,0,0  ;PATCHING AREA.
F10D 000000000 DB      0,0,0,0,0,0  ;PATCHING AREA.

```

```

;*****
;  CONSOLE DATA OUTPUT
;*****

```

```

F113 DB00      CNS$OT: IN      CNO$SP      ;OUTPUT STATUS PORT.
F115 EE00      XRI      CNO$SI      ;ADJUST POLARITY.
F117 E604      ANI      CNO$SB      ;TEST READY BIT.
F119 CA13F1    JZ      CNS$OT      ;TEST AGAIN FOR RDY.
F11C 79        MOV      A,C        ;OUTPUT SETUP.

```

```

F11D D301      OUT      CND$DP      ;OUTPUT CONSOLE DATA.
F11F C9        RET              ;RETURN COMPLETE.

F120 0000000000 DB      0,0,0,0,0,0 ;PATCHING AREA.
F126 0000000000 DB      0,0,0,0,0,0 ;PATCHING AREA.
F12C 0000000000 DB      0,0,0,0,0,0 ;PATCHING AREA.

```

```

;*****
; MESSAGE DISPLAY ROUTINE - HL REG POINTS TO STRING *
;*****

```

```

F132 7E      MSG$OT: MOV      A,M      ;LOAD CHARACTER/BYTE.
F133 FE24      CPI      '$'        ;CHECK FOR TERMINATOR.
F135 C8        RZ              ;EXIT IF TERMINATOR.
F136 4F        MOV      C,A        ;PASS BYTE IN C REG.
F137 CD13F1    CALL     CNS$OT      ;DISPLAY CHARACTER.
F13A 23        INX      H          ;POINT TO NEXT BYTE.
F13B C332F1    JMP      MSG$OT      ;REPEAT SEQUENCE.

```

```

;*****
; DISPLAY A REGISTER IS HEXIDECIMAL *
;*****

```

```

F13E F5      HXB$OT: PUSH     PSW      ;SAVE A REGISTER.
F13F OFOF0F0F RRC!RRC!RRC!RRC ;SHIFT 4 PLACES.
F143 CD47F1    CALL     HXN$OT      ;DISPLAY HEX NIBBLE.
F144 F1        POP      PSW      ;RESTORE A REGISTER.
F147 E60F      HXN$OT: ANI      00FH   ;MASK LOWER NIBBLE.
F149 FE0A      CPI      00AH   ;TEST IF LETTER HEX.
F14B DA50F1    JC       HXN$NM      ;DISPLAY NUMBER.
F14E C607      ADI      '$'-1     ;ADD LETTER OFFSET.
F150 C630      HXN$NM: ADI      '0'   ;START WITH ASCII 0.
F152 4F        MOV      C,A        ;OUTPUT BYTE TO C REG.
F153 C313F1    JMP      CNS$OT      ;CONSOLE OUTPUT.

```

```

;*****
; SYSTEM BOOTSTRAP MESSAGE AREA *
;*****

```

```

F156 0D0A0A494EMSG$IN: DB      CR,LF,LF,'INSERT SYSTEM DISKETTE ', '$'
F171 0D0A0A4444MSG$ER: DB      CR,LF,LF,'DDBOOT LOAD ERROR - ', '$'

```

```

000D =      CR      EQU      00DH      ;CARRIAGE RETURN.
000A =      LF      EQU      00AH      ;LINE FEED COMMAND.

```

```

;*****
;*****
; INJECTION MODULE *** THE FOLLOWING EXECUCUTES IN DD *
;*****
; THIS SECTION OF CODE IS HAS BEEN WRITTEN WITH AN *
; ADDRESS OFFSET SO AS TO ASSEMBLE WITH IM$BGN AT *
; LOCATION ZERO. *
;*****
;*****

```

```

;*****
; DOUBLE D INTERNAL PORT ASSIGNMENTS *
;*****

```

```

0000 = IM$BL$STS EQU 000H ;BOARD STATUS
0000 = IM$BL$CTL EQU 000H ;BOARD CONTROLS
0004 = IM$WD$CMD EQU 004H ;179X COMMAND REGISTER
0004 = IM$WD$STS EQU 004H ;179X STATUS REGISTER
0005 = IM$WD$TRK EQU 005H ;179X TRACK REGISTER
0006 = IM$WD$SEC EQU 006H ;179X SECTOR REGISTER
0007 = IM$WD$DTA EQU 007H ;179X DATA REGISTER
0008 = IM$XP$STP EQU 008H ;STEPPER PULSE
0010 = IM$XP$MTO EQU 010H ;MOTOR TIME OUT
0040 = IM$XP$MTX EQU 040H ;MOTOR TIME EXTEND
0080 = IM$XP$DSH EQU 080H ;DATA SYNC HOLD

```

```

; *****
; 179X-02 COMMAND CODES *
; *****

```

```

0018 = IM$DC$HDL EQU 00011000B ;LOAD R/W HEAD.
0098 = IM$DC$RMS EQU 10011000B ;READ MULTI-SECTOR.
00D0 = IM$DC$STS EQU 11010000B ;SET TYPE 1 STATUS.

```

```

; *****
; Z80 INSTRUCTION HEX CODES - NOTE HI/LOW ORDER SWAP *
; *****

```

```

21FD = IM$LXIY EQU 021FDH ;LOAD Y REG IMED.
45ED = IM$RETN EQU 045EDH ;RETN (NMI RETURN).
E3FD = IM$XTIY EQU 0E3FDH ;EXCHANGE (SP) <> IY.

```

```

; *****
; BOARD STATUS AND CONTROL PORTS *
; *****

```

```

0001 = IM$BS$USO EQU 001H ;179X-02 POLARITY TEST.
0004 = IM$BC$DRO EQU 004H ;DRIVE 0 SELECT/ENABLE.

```

```

; *****
; DISK STATUS MASKS *
; *****

```

```

009C = IM$DM$RER EQU 10011100B ;READ ERROR TEST MASK.
0004 = IM$DM$TKO EQU 00000100B ;TRACK 0 TEST.
0080 = IM$DM$DNR EQU 10000000B ;DRIVE NOT READY.

```

```

; *****
; DISK DRIVE PARAMETERS *
; *****

```

```

000A = IM$TM$STP EQU 10 ;STEPPER INTERVAL - MS.
0028 = IM$TM$DBR EQU 40 ;DELAY BEFORE READ- MS.
004D = IM$NB$TRK EQU 77 ;NUMBER OF TRACKS.

```

```

; *****
; INTERNAL MEMORY ASSIGNMENTS *
; *****

```

```

0000 = IM$BKO EQU 0000H ;LOWER BANK ADDRESS.
0400 = IM$BKL EQU 0400H ;1K BANK LENGTH.
0400 = IM$BK1 EQU IM$BKO+IM$BKL ;UPPER BANK ADDRESS.
0066 = IM$NMI EQU IM$BKO+0066H ;NON-MASKABLE INT ADDR.

```

```

0000 = IM$BL$DC EQU IM$BK0+0376H ;ERROR CODE LOCATION.
0001 = IM$BL$DCS EQU IM$BK0+BL$DCS ;DISK CONTROLLER STAT.

;*****
; BOOTSTRAP COMMUNICATION *
;*****

0001 = BE$HOM EQU 001H ;HOME ERROR.
0002 = BE$RDA EQU 002H ;READ ERROR A.
0004 = BE$RDB EQU 004H ;READ ERROR B.

;*****
; DISK CONTROLLER MODULE (DCM) LINKAGE *
;*****

000D = DCM$SS EQU 13 ;FIRST DCM SECTOR = 13.
0403 = DCM$BG EQU IM$BK1+3 ;DCM COLD START ENTRY.
0400 = DCM$LN EQU 0400H ;DCM LENGTH

;*****
; SET STACK, START DRIVE MOTOR, AND SET INVERT SW (C) *
;*****

F189 310004 IM$BGN: LXI SP,IM$BK1 ;SET UP STACK.
F18C DB40 IN IM$XP$MTX ;TURN ON MOTOR.
F18E 0E00 MVI C,0 ;ASSUME 1793.
F190 DB00 IN IM$BL$STS ;INPUT STATUS.
F192 E601 ANI IM$BS$USO ;TEST USER SW 0.
F194 C21000 JNZ IM$DRV-IM$BGN ;GOTO SELECT DRV.
F197 0EFF MVI C,OFFH ;1791-01 INVERTS.

;*****
; CLEAR 179X-01 INTERRUPT AND SELECT DRIVE 0 *
;*****

F199 CD5000 IM$DRV: CALL IM$STS-IM$BGN ;179X-01 FORCED CLEAR.
F19C 3E04 MVI A,IM$BC$DRO ;DRIVE 0, ENABLED.
F19E D300 OUT IM$BL$CTL ;OUTPUT CONTROLS.

;*****
; CHECK FOR DRIVE READY SIGNAL *
;*****

F1A0 CD5000 CALL IM$STS-IM$BGN ;GET DRIVE STATUS.
F1A3 327703 STA IM$BL$DCS ;STORE DRIVE STATUS.
F1A6 E680 ANI IM$DM$DNR ;CHECK DRIVE NOT RDY.
F1A8 CA2600 JZ IM$HDL-IM$BGN ;IF READY, BOOT SYSTEM.
F1AB AF XRA A ;ZERO A REGISTER.
F1AC C3B100 JMP IM$HLT-IM$BGN ;DOUBLE D SHUTDOWN.

;*****
; LOAD R/W HEAD ON SELECTED DRIVE *
;*****

F1AF 79 IM$HDL: MOV A,C ;GET TRACK 0 VALUE.
F1B0 D305 OUT IM$WD$TRK ;SET TRACK REGISTER.
F1B2 D307 OUT IM$WD$DTA ;SEEK SAME TRACK.
F1B4 FD21 DW IM$LXIY ;Z80 LXIY HEX CODE.
F1B6 3700 DW IM$HME-IM$BGN ;SET NMI RETURN ADDR.
F1B8 3E18 MVI A,IM$DC$HDL ;HEAD LOAD COMMAND.

```

```

F1BA A9          XRA      C          ;INVERT (1791-01).
F1BB D304       OUT      IM$WD$CMD   ;ISSUE COMMAND.
F1BD C33400     IM$WFI: JMP      IM$WFI-IM$BGN ;WAIT FOR INTERRUPT.

```

```

; *****
; POSITION R/W HEAD AT TRACK ZERO
; *****

```

```

F1C0 2E4D       IM$HME: MVI      L,IM$NB$TRK   ;SET MAX TRACKS.
F1C2 CD5000     IM$STP: CALL     IM$STS-IM$BGN   ;GET 179X STATUS.
F1C5 E604              ANI      IM$DM$TKO   ;TEST TRACK 0 BIT.
F1C7 C27000              JNZ      IM$RSU-IM$BGN   ;TRACK 0 EXIT.
F1CA 2D              DCR      L          ;DEC ATTEMPTS.
F1CB CAAF00              JZ       IM$EHM-IM$BGN   ;CANT FIND TRK 0?
F1CE DB08              IN       IM$XP$STP   ;ISSUE STEP PULSE.
F1D0 110A00     LXI      D,IM$TM$STP   ;STEP INTERVAL TIME.
F1D3 CDB800     CALL     IM$TMR-IM$BGN   ;PAUSE FOR PERIOD.
F1D6 C33900     JMP      IM$STP-IM$BGN   ;TRY ANOTHER TIME.

```

```

; *****
; GET UPDATED 179X-01 STATUS
; *****

```

```

F1D9 3ED0       IM$STS: MVI      A,IM$DC$STS   ;TYPE 4 - STATUS.
F1DB A9          XRA      C          ;INVERT (1791-01).
F1DC D304       OUT      IM$WD$CMD   ;ISSUE COMMAND.
F1DE E3          XTHL     ;DELAY
F1DF E3          XTHL     ;DELAY
F1E0 E3          XTHL     ;DELAY
F1E1 E3          XTHL     ;DELAY
F1E2 DB04       IN       IM$WD$STS   ;GET STATUS
F1E4 A9          XRA      C          ;INVERT (1791-01).
F1E5 C9          RET      ;RETURN TO CALLER.

```

```

; *****
; DISK INTERRUPT "NMI" ROUTINE
; *****

```

```

F1EF              ORG      IM$BGN+IM$NMI

F1EF DB04       IN       IM$WD$STS   ;GET 179X STATUS.
F1F1 A9          XRA      C          ;INVERT (1791-01).
F1F2 327703     STA      IM$BL$DCS   ;MAKE STATUS VISIBLE.
F1F5 FDE3       DW       IM$XTIY   ;EXCHANGE (SF) <> IY.
F1F7 ED45       DW       IM$RETN   ;NMI RETURN (RETN).

```

```

; *****
; SET-UP FOR DCM READ OPERATION
; *****

```

```

F1F9 112800     IM$RSU: LXI      D,IM$TM$DBR   ;DELAY BEFORE READ.
F1FC CDB800     CALL     IM$TMR-IM$BGN   ;CALL MS. TIMER.
F1FF 110004     LXI      D,IM$BKL   ;SET BANK LENGTH
F202 210004     LXI      H,IM$BK1   ;DCM LOAD ADDRESS
F205 FD21       DW       IM$LXIY   ;Z80 LXI Y HEX CODE.
F207 A500       DW       IM$REA-IM$BGN   ;READ ERROR TRAP.
F209 3E0D       MVI      A,DCM$SS   ;FIRST SEC OF DCM.
F20B A9          XRA      C          ;INVERT (1791-01)
F20C D306       OUT      IM$WD$SEC   ;SET 179X SEC REG.
F20E 3E98       MVI      A,IM$DC$RMS ;READ MULTI-SECTOR.

```

```

F210 A9          XRA      C          ;INVERT (1791-01).
F211 D304       OUT      IM$WD$CMD   ;ISSUE 179X COMMAND.

```

```

;*****
; ACCEPT EACH BYTE AND STORE IN MEMORY *
;*****

```

```

F213 DB80       IM$RBT: IN      IM$XF$DSH   ;WAIT FOR DATA.
F215 DB07       IN      IM$WD$DTA   ;INPUT INV DATA.
F217 A9         XRA      C          ;INVERT (1791-01).
F218 77         MOV      M,A        ;STORE DCM BYTE.
F219 23         INX     H          ;INCREMENT POINTER.
F21A 1B         DCX     D          ;DECREMENT LENGTH.
F21B 7A         MOV      A,D        ;GET HIGH REG.
F21C B3         ORA     E          ;THEN OR-IN LOW REG.
F21D C28A00     JNZ     IM$RBT-IM$BGN ;READ ANOTHER BYTE.

```

```

;*****
; TEST READ STATUS, TERMINATE OPERATION, GO DCM *
;*****

```

```

F220 DB04       IM$TRS: IN      IM$WD$STS   ;INPUT READ STATUS.
F222 A9         XRA      C          ;INVERT (1791-01).
F223 E69C       ANI     IM$DM$RER   ;TEST FOR ERRORS.
F225 C2AA00     JNZ     IM$REB-IM$BGN ;READ ERROR TRAP.
F228 CD5000     CALL    IM$STS-IM$BGN ;TERMINATE READ.
F22B C30304     JMP     DCM$BG        ;TRANSFER TO DCM.

```

```

;*****
; READ ERROR HAS BEEN DETECTED *
;*****

```

```

F22E 3E02       IM$REA: MVI     A,BE$RDA   ;LOAD READ ERROR CODE.
F230 C3B100     JMP     IM$HLT-IM$BGN ;GO TO ERROR HALT.
F233 3E04       IM$REB: MVI     A,BE$RDB   ;LOAD READ ERROR CODE.
F235 C3B100     JMP     IM$HLT-IM$BGN ;GO TO ERROR HALT.
F238 3E01       IM$EHM: MVI     A,BE$HOM   ;HOME ERROR CODE.
F23A 327603     IM$HLT: STA     IM$BL$ERC ;DISPLAY ERROR CODE.
F23D AF         XRA      A          ;ZERO A REG.
F23E D300       OUT     IM$BL$STS   ;DESELECT DRIVE.
F240 76         HLT     ;TERMINATE.

```

```

;*****
; TIMER - WAIT FOR (BC * 1.0) MILLISECONDS *
;*****

```

```

F241 3EDC       IM$TMR: MVI     A,220      ;LOAD INT MS VALUE.
F243 3D         IM$TMX: DCR     A          ;DEC FOR 1 MS.
F244 00         NOP     ;EXTRA TIMING DELAY.
F245 C2BA00     JNZ     IM$TMX-IM$BGN ;REPEAT FOR 1 MS.
F248 1B         DCX     D          ;TEST FOR ANOTHER MS.
F249 7A         MOV     A,D        ;CHECK REG D.
F24A B3         ORA     E          ;AND REGISTER E.
F24B C2B800     JNZ     IM$TMR-IM$BGN ;DO ANOTHER 1 MS.
F24E C9         RET     ;TIME PERIOD EXPIRED!

```

```

;*****

```

```

F24F           IM$END: END      BEGIN      ;END OF ASSEMBLY.

```


?

A>DDT

DDT VERS 2.2

-DF000,F2FF

```

F000 C3 12 F0 C3 3A F0 C3 DA F0 C3 F6 F0 C3 13 F1 C3 .....:.....
F010 32 F1 00 00 00 00 00 00 00 00 00 00 00 00 00 2.....
F020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
F030 00 00 00 00 00 00 00 00 00 00 31 80 00 DB 43 E6 .....1...C.
F040 0E 07 F6 E0 67 2E 00 22 40 00 3E 01 32 42 00 32 ....G.."@.>.2B.2
F050 43 00 3E 01 D3 43 01 C6 00 2A 40 00 EB 21 89 F1 C.>..C...*@...!..
F060 CD A7 F0 3E 80 D3 43 E3 E3 3A 42 00 47 DB 43 A0 ...>..C...:B.G.C.
F070 C2 6D F0 3E 01 D3 43 2A 40 00 11 77 03 19 7E E6 .M.>..C*@..W..^
F080 80 C2 B4 F0 7E A7 C2 C8 F0 2A 40 00 11 78 03 19 ....^.....*e...X..
F090 5E 23 56 23 4E 23 46 D5 3E 03 D3 43 2A 40 00 CD ^#V#N#F.>..C*@..
FOA0 A7 F0 3E 01 D3 43 C9 7E 23 EB 77 23 EB 0B 78 B1 ..>..C.^#.W#...X.
FOB0 C2 A7 F0 C9 3A 43 00 A7 CA 52 F0 AF 32 43 00 21 .....:C...R...2C.!
FOC0 56 F1 CD 32 F1 C3 52 F0 32 43 00 21 71 F1 CD 32 V..2..R.2C.!0..2
FOD0 F1 3A 43 00 CD 3E F1 76 00 00 DB 00 EE 00 E6 02 .:C.>.V.....
FOE0 C8 3E FF C9 00 00 00 00 00 00 00 00 00 00 00 .>.....
FOF0 00 00 00 00 00 00 CD DA F0 CA F6 F0 DB 01 E6 7F .....
F100 C9 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
F110 00 00 00 DB 00 EE 00 E6 04 CA 13 F1 79 D3 01 C9 .....Y...
F120 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
F130 00 00 7E FE 24 C8 4F CD 13 F1 23 C3 32 F1 F5 0F ..^$.0...#.2...
F140 0F 0F 0F CD 47 F1 F1 E6 0F FE 0A DA 50 F1 C6 07 ....G.....P...
F150 C6 30 4F C3 13 F1 0D 0A 0A 49 4E 53 45 52 54 20 .00.....INSERT
F160 53 59 53 54 45 4D 20 44 49 53 4B 45 54 54 45 20 SYSTEM DISKETTE
F170 24 0D 0A 0A 44 44 42 4F 4F 54 20 4C 4F 41 44 20 $...DDBOOT LOAD
F180 45 52 52 4F 52 20 2D 20 24 31 00 04 DB 40 0E 00 ERROR - $1...@..
F190 DB 00 E6 01 C2 10 00 0E FF CD 50 00 3E 04 D3 00 .....P.>...
F1A0 CD 50 00 32 77 03 E6 80 CA 26 00 AF C3 B1 00 79 .P.2W....&.....Y
F1B0 D3 05 D3 07 FD 21 37 00 3E 18 A9 D3 04 C3 34 00 .....!7.>.....4.
F1C0 2E 4D CD 50 00 E6 04 C2 70 00 2D CA AF 00 DB 08 .M.P....P.-.....
F1D0 11 0A 00 CD B8 00 C3 39 00 3E D0 A9 D3 04 E3 E3 .....9.>.....
F1E0 E3 E3 DB 04 A9 C9 FF FF FF FF FF FF FF FF DB .....
F1F0 04 A9 32 77 03 FD E3 ED 45 11 28 00 CD B8 00 11 ..2W....E.(.....
F200 00 04 21 00 04 FD 21 A5 00 3E 0D A9 D3 06 3E 98 ..!...!..>...>.
F210 A9 D3 04 DB 80 DB 07 A9 77 23 1B 7A B3 C2 8A 00 .....W#.Z....
F220 DB 04 A9 E6 9C C2 AA 00 CD 50 00 C3 03 04 3E 02 .....P...>.
F230 C3 B1 00 3E 04 C3 B1 00 3E 01 32 76 03 AF D3 00 ...>.....>.2V....
F240 76 3E DC 3D 00 C2 BA 00 1B 7A B3 C2 B8 00 C9 FF V>.=.....Z.....
F250 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF .....
F260 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF .....
F270 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF .....
F280 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF .....
F290 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF .....
F2A0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF .....
F2B0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF .....
F2C0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF .....
F2D0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF .....
F2E0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF .....
F2F0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF .....

```

-

JADE COMPUTER PRODUCTS
4901 W. ROSECRANS BLVD
HAWTHORNE, CALIF 90250

Subject: Engineering Change Notice # 1.
Product: Double D Disk Controller.
Revision: B and C revision boards.
Date: August 4, 1980.

It has been reported that insertion of the Double D Disk Controller into of some S100 systems prevents normal operation. Usually on these systems the common characteristic is that they just will not operate. Please note S100 Bus pins #20, #53, and #70 are connected to ground, as per S100 Standards, IEEE Task 696.1/D2. These pin connections do cause interference with IMSAI front panel systems or cpu boards designed to operate with front panels. It is permissible to cut the foil links connecting pins #20, 53, and 70 to their respective plate-thru-holes. Please verify in your system documentation that these pins are causing interference before cutting.

Subject: Engineering Change Notice # 2.
Product: Double D Disk Controller.
Revision: B and C revision boards.
Date: August 4, 1980.

A review of the Double D 8" phase locked loop has been completed. This has resulted in a reselection of some component values. Enhanced operation, particularly in double density, will be realized with the following modification. This modification will double the loop capture range and also eliminate a cause of loop instability.

With the exception of R1, just change those resistors listed for the new values as shown in the list. R1 does have a changed value but also must be installed so that it will connect to +5 volts regulated instead of the previous connection to Vx. With careful lead bending and resistor placement, one lead can solder to the +5V foil running from pin #16 of IC 1A to pin #16 of IC 1B. It would help to scrape some of the solder mask away before soldering R1 to this foil. Vx will now measure about +5.0 volts. Installation of the modification will require retuning the PLL.

R1	6.8K 1/4W (TO +5v)	R42	470K 1/4W
R3	12K 1/4W	R43	2.7K 1/4W
R4	10K 1/4W	R49	JUMPER
R38	20K 1/4W		

Subject: Engineering Change Notice # 3.
Product: Double D Disk Controller.
Revision: B and C revision boards.
Date: August 4, 1980.

The Double D Disk Controller uses S-100 signal Swo*. CPUs such as SBC-100 and SBC-200 do not generate these signals and therefore present an interface problem. The following modification has solved the problem with the above mentioned boards.

1. On the solder side of the board: Cut the foil link from S-100 pin # 97 to the plate-thru-hole.
2. On the Solder Side of the Board: Using a small gauge wire jumper IC 1H pin #2 to IC 3J pin #1.

Subject: Engineering Change Notice # 4.
Product: Double D Disk Controller.
Revision: B and C revision boards.
Date: August 4, 1980.

The Double D Disk Controller exhibits erratic operation when run with the Big-Z Z80 CPU board. The problem exists on the Big-Z board. The following modification fixes this problem. Note: The Big-Z does not send out write data to the S-100 Bus until it actually sends the write strobe. This modification allows the write data to settle on the S-100 Bus before the write strobe is issued.

1. On the solder side Big-Z: Cut the foil from IC 22 pin # 13.
 2. On the solder side Big-Z: Jump IC 22 pin # 13 to pin # 3.
-

Subject: Engineering Change Notice # 5B.
Product: Double D Disk Controller.
Revision: B and C revision boards.
Date: August 5, 1980.

IMPORTANT NOTICE! BOARD MODIFICATION NEEDED on revision C.
This ECN pertains to the use of the Double D disk controller with
JADE RELEASE # 2 of CP/M 2.2. Connector J3 pin #48 was
designated ILLEGAL PACK*. It has been redefined and it is now
designated TWO SIDED*.

Many Shugart SA800/801, Siemens FD100-8, and other models of
disk drives have optional data separators installed. Disk drives
using these options use pin #48 of the 50 line ribbon for the
SEPERATED DATA* signal. As release #2 (specifically DCM2)
monitors this signal line for TWO SIDED* , ERRATIC DISK OPERATION
would be expected. Please cut the foil link between the two
plate-thru-holes at J3 pin #48 (Revision C). For use with the
SA850/851 disk drive a jumper should be installed on the J3
patching area from the lower pin # 48 plate-thru-hole to the
upper pin #10 plate-thru-hole. This completes the path for the
TWO SIDED* signal from the SA850/851.

Subject: Engineering Change Notice # 6.
Product: Double D Disk Controller.
Revision: B and C revision boards.
Date: August 5, 1980.

NOTE: Concerning the Double D Disk Controller when used
with 64K of system memory or any other memory arrangement where
the Double D memory window overlaps assigned memory space.

When used in this configuration the Phantom Block must be
jumpered to complete the PHAN* signal path to the S100 bus. The
Phantom Block is located below IC 4D. It appears as two plate-
thru-holes enclosed by a silkscreen border labeled PHAN*. Add a
jumper connecting these two holes together. Any memory board
that the Double D is to overlap must be configured so as to be
disabled when responding to the Phantom signal (PHAN*).

Subject: Engineering Change Notice # 7.
Product: Double D Disk Controller.
Revision: B and C revision boards.
Date: October 6, 1980.

NOTE: Use National Semiconductor 74LS123 one-shots on the Double-D controller board. The resistor / capacitor combinations have been selected to provide proper pulse periods when used with this one-shot. Double D disk controller boards (A&T and kit) are now supplied with National Semiconductor 74LS123s. Customers who build Double D bare boards take note.

Subject: Engineering Change Notice # 8.
Product: Double D Disk Controller.
Revision: C revision boards.
Date: October 6, 1980.

The following list contains corrections to the Double D revision C schematic of 3/7/80. Please make these corrections to your diagrams.

1. The output of IC 1M (7406) pin 6 to the plate-through-hole in the Interrupt Block should be labeled DINT*. (Page 1)
 2. A section of IC 3L (pins 5 and 15) has been drawn and labeled as a 74LS244. This is shown on page 1 connected to the 1791. This part is a 74LS240.
 3. An inverting buffer, part of IC 3L (74LS240) pins 11 and 9, is not shown in the diagram. Pin 11 is the input and is connected to DDEN. Pin 9 is the output and is the source for DDEN*.
 4. Four pin assignments of IC 3H (8131) are in error. Change pin 13 to pin 11, pin 12 to pin 10, pin 11 to pin 13, and pin 10 to pin 12. (Page 1)
 5. The input to IC 4A on pin 13 is labeled as BPWR*. This label should read as BPWR. (Page 2)
-

Subject: Engineering Change Notice # 9.
Product: Double D Disk Controller.
Revision: B and C revision boards.
Date: October 6, 1980.

The following jumper configuration can be used with the Shugart SA800/801 model disk drive.

EACH DRIVE: A, B, C, Y, T2, HL, 800

DRIVE A: DS1
DRIVE B: DS2
DRIVE C: DS3
DRIVE D: DS4

LAST DRIVE: T1, T3, T4, T5, T6

THE L JUMPER IS SET DEPENDING ON THE -5V SUPPLY.
CONSULT YOU SA800 MANUAL. USE NO OTHER JUMP PLUGS
IN THIS CONFIGURATION.

Siemens disk drive models FD120-8B and the newer FD100-8D can be used the Jade Double D. Each drive must have the Radial select option plug set to the proper drive number. 0 selects drive A, 1 selects drive B, 2 selects drive C, and 3 selects drive D. Only the last drive on the ribbon should contain the resistor pack. Be sure to review ECN #5. No other changes are needed.

The following page describes a tested jumper configuration for the Shugart SA850/851 when used with JADE RELEASE # 2 of CP/M 2.2.

SHUGART SA850/851

Start with the disk drives(s) set to factory configuration as described in the service and maintenance manual. Then perform the following alterations to the drive(s).

1. Remove the 'IW' plug. This allows for lower write current on the inside tracks.
 2. Remove the 'RS' plug and install this plug at 'RM'. This allows DRIVE READY to be true when DIRECTION (SIDE SELECT) is selecting the wrong side of a single sided diskette.
 3. Break connection 'X' on the shorting plug and install a plug at 'C'. This allows the drive to be selected without enabling the stepper or loading the R/W head.
 4. Break connection 'Z' on the shorting plug and install a plug at 'Y'. Activity light will be on when the R/W head is loaded.
 5. Remove the 'SZ' plug and install at 'S1'. This allows side select from the DIRECTION signal.
 6. Remove the '851' plug and install at '850'. This is done for soft sector diskettes.
 7. Install a plug at '2S'. This allows the drive to issue the TWO SIDED signal when double sided diskettes are being used.
 8. Remove the 'DL' plug.
 9. Set -5/-15 V according to the negative supply voltage you are using. This applies to early SA850/851.
 10. Drive A: Install plug at 'DS1'.
Drive B: Install plug at 'DS2'.
Drive C: Install plug at 'DS3'.
Drive D: Install plug at 'DS4'.
 11. Remove the terminator from all but the last drive on the ribbon cable.
 12. Install Jade Double D ECN #5.
-

Subject: Engineering Change Notice # 10.
Product: Double D Disk Controller - QUME DATATRAK 8 DRIVES
Revision: B and C boards, Release 2 software
Date: May 8, 1981.

The following jumped options should be changed on the QUME DATATRACK 8 disk drive for use with the DOUBLE D disk controller.

1. Remove programmable shunt from P.C. board socket. Bend pins B, HL, and Z to prevent making contact. Replace shunt back into socket.
2. Insert shunt plugs at locations C, DS, Y, and 2S.
3. Please read and perform Engineering Change Notice # 5. It applies to QUME DATATRACK 8 as well as the SHUGART SA850/851.

The following patch is required in DCM2. Location 111D hex was OFB hex and should be changed to ODB hex. The new source code for this line is:

```
TDL:      ANI  #(BC.DSE!BC.SD1)
ASM:      ANI  NOT ( BC$DSE OR BC$SD1 )
```

Double D CP/M 2.2 distributed after Nov 30, 1981 include this modification (starting with S/N 2-187-1410).

ENJOY YOUR QUMES.

```
*****
;
; PROGRAM ID: DISK CONTROLLER MODULE
;
; VERSION: 2.2 8" RELEASE 2A
;
*****
;
; PRESENTED BY: JADE COMPUTER PRODUCTS INC.
; 4901 W. ROSECRANS BLVD.
; HAWTHORNE, CALIFORNIA
; 90250, U.S.A.
;
*****
;
; WRITTEN BY: STAN KRUMME
;
*****
; THE DISK CONTROLLER MODULE (DCM2) EXECUTES INTERNAL
; TO THE JADE DOUBLE D DISK CONTROLLER BOARD. THIS
; PROGRAM PROVIDES A FACILITY TO READ/WRITE DISKETTE
; SECTORS AND FORMAT DISKETTE TRACKS (IN SINGLE AND
; DOUBLE DENSITY). THIS DCM SETS THE PARAMETERS FOR
; EACH DRIVE DURING THE "LOG-ON" OPERATION. THE
; FORMAT.COM PROGRAM WRITES AN IDENTIFICATION SECTOR
; WHICH PROVIDES THE NEEDED INFORMATION. IF THIS
; IDENTITY SECTOR IS NOT PRESENT ON THE DISKETTE,
; IT IS ASSUMED TO BE A STANDARD 8" 3740 FORMAT.
; THIS PROGRAM CONTAINS A 4 WORD TIMING BLOCK WHICH
; SHOULD BE PATCHED TO MATCH THE USERS DISK DRIVES.
; THIS HAS NORMALLY BEEN SET FOR SHUGART SA800/801.
;
*****
;
; DISK CONTROLLER MODULE IS COMMAND COMPATABLE WITH
; THE FOLLOWING WESTERN DIGITAL CONTROLLER CHIPS.
; DOUBLE D USER SWITCH 0 (U0 OR R0) MUST BE SET TO
; INDICATE THE CONTROLLER CHIP DATA BUS POLARITY.
;
*****
; CONTROLLER IC USER SW0
; -----
; FD1791-02 (01) CLOSED
; FD1793-02 (01) OPENED
; FD1795-02 CLOSED
; FD1797-02 OPENED
;
*****
; THE FD1795-02 AND FD1797-02 PROVIDE ENHANCED SINGLE
; DENSITY PERFORMANCE IN THAT THESE CHIPS ARE FULLY
; COMPATABLE WITH FD1771-01 3740 FORMATS.
;
*****
```

```
*****  
; THE FOLLOWING IS A LIST OF THE INTERNAL I/O ADDRESS *  
; ASSIGNMENTS. THESE PORTS AND CONTROLS CAN ONLY BE *  
; USED BY THE ONBOARD Z80A. THESE PORTS AND CONTROLS *  
; ARE NOT IN THE S100 BUS ADDRESS SPACE. *  
*****
```

```
***** ( CONTROLLER PORT ASSIGNMENTS ) *****
```

```
0000 BL.STS == 000H ;BOARD STATUS PORT.  
0000 BL.CTL == 000H ;BOARD CONTROL PORT.  
0004 WD.CMD == 004H ;179X COMMAND REGISTER.  
0004 WD.STS == 004H ;179X STATUS REGISTER.  
0005 WD.TRK == 005H ;179X TRACK REGISTER.  
0006 WD.SEC == 006H ;179X SECTOR REGISTER.  
0007 WD.DTA == 007H ;179X DATA REGISTER.
```

```
***** ( CONTROLLER FUNCTION ASSIGNMENTS ) *****
```

```
0008 XP.STP == 008H ;ISSUE STEP PULSE.  
0010 XP.MTO == 010H ;MOTOR TURN OFF.  
0020 XP.IRR == 020H ;S100 INT-REQ RESET.  
0040 XP.MTX == 040H ;MOTOR TIME EXTEND.  
0080 XP.DSH == 080H ;DATA SYNC HOLD.
```

```
*****  
; THE FOLLOWING LIST ASSIGNS EACH BIT POSITION AND *  
; FUNCTION OF THE BOARD CONTROL PORT (BL.CTL). *  
*****
```

```
***** ( BIT ASSIGNMENTS ) *****
```

```
0001 BC.DSA == 00000001B ;DRIVE SELECT A (2*0).  
0002 BC.DSB == 00000010B ;DRIVE SELECT B (2*1).  
0004 BC.DSE == 00000100B ;DRIVE SELECT ENABLE.  
0008 BC.EIA == 00001000B ;EIA SIGNAL LEVEL OUT.  
0010 BC.DDE == 00010000B ;DOUBLE DENSITY ENABLE.  
0020 BC.DAS == 00100000B ;DIRECTION AND SIDE  
0040 BC.PCA == 01000000B ;PRECOMP SELECT A.  
0080 BC.PCB == 10000000B ;PRECOMP SELECT B.
```

```
***** ( FUNCTION ASSIGNMENTS ) *****
```

```
0003 BC.DSN == BC.DSA!BC.DSB ;DRIVE NMBR MASK.  
0000 BC.SDS == 0 ;SINGLE DENSITY.  
0010 BC.BDS == BC.DDE ;DOUBLE DENSITY.  
0040 BC.PCH == BC.PCA ;PRECOMP - HEAVY.  
0080 BC.PCM == BC.PCB ;PRECOMP - MEDIUM.  
00C0 BC.PCL == BC.PCA!BC.PCB ;PRECOMP - LIGHT.  
0020 BC.SD1 == BC.DAS ;SELECT SIDE ONE.  
0020 BC.INW == BC.DAS ;STEP INWARD DIRC.
```

```
*****
```

```
*****  
; THE FOLLOWING LIST DEFINES EACH BIT AND FUNCTION OF *  
; THE BOARD STATUS PORT (BL.STS). *  
*****
```

```
0001 BS.US0 == 00000001B ;USER SWITCH 0.  
0002 BS.US1 == 00000010B ;USER SWITCH 1.  
0004 BS.TST == 00000100B ;TEST MODE SWITCH.  
0008 BS.INT == 00001000B ;HOST INT REQUEST.  
0010 BS.EIA == 00010000B ;EIA SIGNAL LEVEL IN.  
0020 BS.MOF == 00100000B ;MOTOR OFF INIDCATOR.  
0040 BS.TSD == 01000000B ;TWO SIDED DRIVE FLAG.  
0080 BS.DCN == 10000000B ;DISK CHANGE INDICATOR.
```

```
*****  
; THE FOLLOWING IS A LIST OF COMMAND CODES ISSUED TO *  
; THE 179X-02 DISK CONTROLLER. *  
*****
```

```
0018 DC.HDL == 00011000B ;SEEK/LOAD RW HEAD.  
0010 DC.HDU == 00010000B ;SEEK/UNLD RW HEAD.  
0088 DC.RDS == 10001000B ;READ SECTOR.  
00A8 DC.WRS == 10101000B ;WRITE SECTOR.  
00F0 DC.WRT == 11110000B ;WRITE TRACK FORMAT.  
00C0 DC.RDA == 11000000B ;READ TRACK ADDRESS.  
00D0 DC.STS == 11010000B ;SET TYPE 1 STATUS  
00D8 DC.IFI == 11011000B ;FORCED INTERRUPT.
```

```
*****  
; THE FOLLOWING LIST CONTAINS ALL THE MASKS USED TO *  
; TEST THE 179X-02 STATUS CODES (PORT WD.STS). *  
*****
```

```
009D DM.RER == 10011101B ;READ ERROR TEST.  
00FD DM.WER == 11111101B ;WRITE ERROR TEST.  
00E4 DM.FER == 11100100B ;FORMAT ERROR TEST.  
0004 DM.TK0 == 00000100B ;TRACK 0 TEST.  
0020 DM.HDL == 00100000B ;HEAD LOAD TEST.  
0080 DM.DNR == 10000000B ;DRIVE NOT READY.  
0004 DM.LDE == 00000100B ;LOST DATA ERROR.
```

```
*****
```

```
*****
; THE FOLLOWING LIST DEFINES INTERNAL MEMORY.
*****

***** ( BASE ADDRESS FOR DCM ) *****

1000      BASE      ==      1000H      ; BASE ADDRESS.

***** ( MEMORY BANKS ) *****

1000      BANK.0    ==      BASE+0000H    ; BANK 0 DEFINED.
0400      BANK.L    ==      0400H        ; BANK LENGTH.
1400      BANK.1    ==      BANK.0+BANK.L  ; BANK 1 DEFINED.

***** ( RESTART VECTORS ) *****

1000      RST.0     ==      BANK.0+0000H    ; RESTART 0.
1008      RST.1     ==      BANK.0+0008H    ; RESTART 1.
1010      RST.2     ==      BANK.0+0010H    ; RESTART 2.
1018      RST.3     ==      BANK.0+0018H    ; RESTART 3.
1020      RST.4     ==      BANK.0+0020H    ; RESTART 4.
1028      RST.5     ==      BANK.0+0028H    ; RESTART 5.
1030      RST.6     ==      BANK.0+0030H    ; RESTART 6.
1038      RST.7     ==      BANK.0+0038H    ; RESTART 7.

***** ( INTERRUPT VECTORS ) *****

1038      HR.INT    ==      RST.7          ; MASKABLE.
1066      NM.INT    ==      BANK.0+0066H    ; NON MASKABLE.

***** ( I/O COMMUNICATION ) *****

1370      IO.BLK    ==      BANK.0+0370H    ; I/O BLOCK BEGIN.
1370      TP.STK    ==      IO.BLK+0000H    ; TOP OF STACK.
1370      CMD.BK    ==      IO.BLK+0000H    ; COMMAND BLOCK.
1380      BUF.BG    ==      IO.BLK+0010H    ; SECTOR BUFFER.
1700      FMT.BG    ==      BANK.1+0300H    ; FORMAT BUFFER.
1708      FMT.PS    ==      FMT.BG+0008H    ; FORMAT PROGRAM.

*****
```

```

;*****
; WAIT IS A "RESTART" TO THE TIMER SUBROUTINE ENTRY. *
; THIS SUBROUTINE PROVIDES MOST OF THE TIMING USED BY *
; THE DOUBLE D CONTROLLER. *
;*****

```

```

        .DEFINE WAIT = [
            RST     1]

```

```

;*****
; ASSEMBLER DIRECTIVES *
;*****

```

```

        .PABS           ;ABSOLUTE ADDRESSING.
        .PHEX          ;INTEL HEX OBJECT FILE.
        .XLINK         ;NO LINKAGE REQUIRED.

```

```

;*****
; TENTH MILLESECOND TIMING CONSTANTS / 0.2 MS FOR 5" *
;*****

```

```

0019      TMR.FC ==      0019H  ;TIMING CONSTANT, FIRST PASS.
001C      TMR.NC ==      001CH  ;TIMING CONSTANT, REPEAT PASS.

```

```

;*****
;      BAUD RATE GENERATOR - TIMING CONSTANTS
;*****
;      BAUDRATE      US/BIT      8" SYS      5 " SYS      *
;      -----      -
;      19200          52.1         9           N.A.         *
;      9600           104.2        25           9             *
;      4800           208.3        57           25            *
;      2400           416.6        121          57            *
;      1200           833.3        248          121           *
;      600            1666.6       N.A.         248           *
;*****

```

```

0019      BAUD.C ==      25.      ;BAUD RATE CONSTANT 9600 8".

```

```

;*****
; ERROR RECOVERY VALUES *
;*****

```

```

0005      RTY.SK ==      5        ;REPOSITION R/W HEAD ON RETRY.
0009      RTY.LS ==      9        ;LAST REPEATED RETRY.

```

```

001A      TRK.OB ==      26       ;AT FIRST THIRD TRACK OF DISK.
0034      TRK.IB ==      52       ;AT SECOND THIRD TRACK.

```

```

;*****

```

```

;*****
; THE FOLLOWING AREA IS THE INITIAL START JUMP TABLE. *
; THE FIRST JUMP IS EXECUTED WHEN THE ONBOARD Z80A IS *
; RESET. THE SECOND JUMP IS THE DCM ENTRY FROM A *
; BOOTSTRAP LOADER. THIS ENTRY ASSUMES DCM HAS BEEN *
; LOADED INTO DOUBLE D BANK 1 BY THE LOADER ROUTINE. *
; THE LAST TWO BYTES HOLD THE JUMP ADDRESS USED BY *
; RESTART INTERRUPT ROUTINE AT BANK 0 + 0380H. *
;*****
    
```

```

1000          .LOC      RST.0          ;MODULE BEGINNING.

1000      C3 0000          JMP      0          ;NOT IMPLEMENTED.
1003      C3 1780          JMP      INIT.B+BANK.L ;BOOTSTRAPPED ENTRY.
1006      1041      HR.VEC: .WORD      X.CUTE      ;HOST INTERRUPT VECTOR.
    
```

```

;*****
; THE FOLLOWING SUBROUTINE IS THE ENTRY POINT FOR THE *
; DISK CONTROLLER TIMING MODULE. THIS MODULE PROVIDES *
; DELAYS WHICH ARE MULTIPLES OF 100 MICROSECONDS. THE *
; CONTENTS OF REGISTER PAIR DE DETERMINES THE TOTAL *
; PERIOD. (DELAY = (DE) * 100 MICROSECONDS). THIS *
; SUBROUTINE IS ENTERED BY THE MACRO "WAIT". *
;*****
    
```

```

1008          .LOC      RST.1          ;TIMING ENTRY POINT.

1008      0619          MVI      B,TMR.FC      ;FIRST TICK CONSTANT.
100A      10FE          DJNZ     .          ;AUTO DEC UNTIL ZERO.
100C      C3 1074          JMP      TICK.E      ;JUMP TO TICK ENTRY.
    
```

```

;*****
; THE FOLLOWING SECTION IS THE DISK DRIVE TIMING AREA.*
; THE TIMES ARE SET FOR THE SHUGART SAS00. THIS AREA *
; SHOULD BE MODIFIED FOR THE END USERS DRIVE TYPE. *
;*****
    
```

***** (TIMING VALUES IN 0.1 MS)*****

```

1010          .LOC      RST.2

1010      015E      TM.HLD: .WORD      350      ;HEAD ENGAGE TIME.
1012      0050      TM.STP: .WORD      80       ;STEPPER INTERVAL.
1014      0050      TM.ALS: .WORD      80       ;AFTER LAST STEP.
1016      0001      TM.MTG: .WORD      1        ;MOTOR START UP.
    
```

```

;*****
    
```

```

;*****
; THE FOLLOWING SUBROUTINE PROVIDES THE R/W HEAD CNTL *
; FUNCTION. AS THE FD179X-02 DOES NOT OFFER THIS *
; EXPLICIT COMMAND, THE SEEK COMMAND (TYPE-1) IS USED *
; WITH THE HEAD LOAD BIT SET / RESET. THE DESTINATION *
; TRACK IS SET EQUAL TO THE TRACK REGISTER TO BYPASS *
; THE FD179X-02 STEPPING FUNCTION. PLEASE REFER TO *
; THE FD179X-02 FLOW-CHART FOR TYPE-1 COMMANDS. *
;*****
    
```

```

1018   FDE1      EX.HCF: POP      Y           ;RETURN ADDR IN REG Y.
101A   DB05      IN          WD,TRK        ;READ PRESENT TRACK.
101C   D307      OUT         WD,DTA        ;SET DESTINATION TRK.
101E   78        MOV        A,B          ;LOAD TYPE-1 COMMAND.
101F   A9        XRA        C           ;INVERT (1791-01).
1020   D304      OUT         WD,CMD        ;ISSUE COMMAND.
1022   18FE      JMPR       ;WAIT FOR INTERRUPT.
    
```

```

;*****
; THE FOLLOWING SUBROUTINE UPDATES THE FD179X-02 *
; STATUS PORT TO REFLECT CURRENT TYPE-1 STATUS CODES. *
; NOTE: THIS IS A TYPE-4 COMMAND WITH NO INTERRUPT *
; CONDITIONS SET. *
;*****
    
```

```

1024   3ED0      EX.STS: MVI      A,DC.STS  ;LOAD SET-STATUS CMND.
1026   A9        XRA        C           ;INVERT (1791-01).
1027   D304      OUT         WD,CMD        ;ISSUE COMMAND.
1029   E3        XTHL       ;PAUSE FOR FD179X-02.
102A   E3        XTHL       ;PAUSE MORE.
102B   E3        XTHL       ;PAUSE STILL MORE.
102C   E3        XTHL       ;PAUSE LAST TIME.
102D   DB04      IN          WD,STS        ;INPUT STATUS PORT.
102F   A9        XRA        C           ;INVERT (1791-01).
1030   C9        RET        ;RETURN TO USER.
    
```

```

;*****
; THE FOLLOWING SECTION IS THE MASKABLE INTERRUPT *
; ROUTINE. THIS ROUTINE IS EXECUTED WHEN RESTARTING *
; THE Z80 FROM A HALT. THE FUNCTIONS ARE RESET THE *
; DOUBLE D INT REQ FLIP-FLOP, PUT THE INTERRUPTED *
; ADDR IN REG DE, AND JUMP ADDRESS AT "HR.VEC". *
;*****
    
```

```

1038           .LOC      HR.INT          ;HOST INTERRUPT ADDR.
1038   DB20      IN          XF,IRR        ;RESET INTERRUPT REQ FF
103A   D1        POP      D           ;PURGE INTERRUPTED ADDR
103B   2A 1006   LHLD      HR,VEC        ;LOAD RETURN ADDRESS
103E   E9        PCHL       ;JUMP RETURN ADDRESS
    
```

```

;*****
    
```

```

;*****
; THE FOLLOWING SECTION HALTS EXECUTION OF THE
; ONBOARD Z80A PROCESSOR. DURING THIS TIME THE HOST
; SYSTEM CAN SWITCH THE CONTROLLER MEMORY INTO THE
; S100 BUS FOR STATUS CHECK, SETTING COMMAND BLOCK,
; AND SECTOR DATA TRANSFERS.
;*****

```

```

103F    FB    FETCH:  EI          ;ENABLE INTERRUPT START
1040    76    HLT          ;HALT ON-BOARD PROCESSOR

```

```

;*****
; THE FOLLOWING SECTION GAINS CONTROL AFTER THE DISK
; CONTROLLER IS INTERRUPTED FROM THE HALT CONDITION.
; THIS SECTION BRANCHES TO THE INDIVIDUAL COMMAND
; ROUTINES. THE COMMAND TABLE CONTAINS THE ADDRESSES
; FOR THIS DISTRIBUTION.
;*****

```

```

1041    3A 1370 X.CUTE: LDA      CB.CMD      ;LOAD HOST COMMAND.
1044    E607    ANI      CM.MSK      ;MASK ANY OPTIONS.
1046    87      ADD      A           ;GET 2*A VALUE.
1047    1600    MVI      D,0         ;ZERO D REGISTER.
1049    5F      MOV      E,A        ;DE NOW TABLE OFFSET.
104A    21 1053 LXI      H,CM.DTA    ;LOAD TABLE ADDRESS.
104D    19      DAD      D           ;NOW POINTS TO ENTRY.
104E    5E      MOV      E,M        ;LOW ORDER ADDR LOAD.
104F    23      INX      H           ;POINT TO NEXT BYTE.
1050    56      MOV      D,M        ;HI ORDER ADDRESS.
1051    EB      XCHG          ;BRANCH ADDR IN HL.
1052    E9      PCHL          ;BRANCH TO COMMAND.

```

```

;*****
; THE FOLLOWING AREA IS THE COMMAND DRIVER TABLE.
; EACH ENTRY POINTS TO THE COMMAND DRIVER ROUTINE.
;*****

```

```

1053    CM.DTA ==          .          ;COMMAND TABLE.

1053    10AC    ..CM0A: .WORD  $.LGON  ;LOG-ON DRIVE.
1055    107C    ..CM1A: .WORD  $.READ  ;READ SECTOR.
1057    108A    ..CM2A: .WORD  $.WRIT  ;WRITE SECTOR.
1059    1098    ..CM3A: .WORD  $.FORM  ;FORMAT TRACK.
105B    10C7    ..CM4A: .WORD  $.ADDR  ;READ ADDRESS.
105D    10CF    ..CM5A: .WORD  $.LIST  ;LIST OUTPUT.
105F    10D5    ..CM6A: .WORD  $.LSTT  ;LIST STATUS.
1061    10E4    ..CM7A: .WORD  $.IDLE  ;BACKGROUND.

0007    CM.MSK ==          007H      ;COMMAND MASK.

```

```

;*****

```



```

;*****
; THE FOLLOWING SECTION IS THE NON-MASKABLE INTERRUPT *
; ROUTINE. UPON 179X-02 COMMAND TERMINATION THE Z80 *
; RECIEVES A NON-MASKABLE INTERRUPT. THE STATUS PORT *
; IS INTERROGATED AND SAVED (SV.STS). REGISTER IY *
; CONTAINS THE RETURN ADDRESS. *
;*****

```

```

1066          .LOC      NM.INT          ;NON-MASKABLE INT.

1066  DB04      WD.INT: IN      WD.STS      ;GET 179X STATUS.
1068  A9                XRA      C          ;INVERT (1791).
1069  32 1334      STA      SV.STS      ;SAVE STATUS.
106C  FDE3        XTIY                ;EXCHANGE (SP)<>IY.
106E  ED45        RETN                 ;RETURN AT OLD IY.

```

```

;*****
; THIS SECTION IS THE REMAINDER OF THE TIMING *
; SECTION ENTERED BY A RESTART 1. SEE THAT SECTION *
; FOR THE DESCRIPTION. *
;*****

```

```

1070  061C      TICK.R: MVI      B,TMR.NC    ;NORMAL TICK CONSTANT.
1072  10FE                DJNZ      .        ;AUTO DEC UNTIL ZERO.
1074  1B                TICK.E: DCX      D    ;DECREMENT AMOUNT.
1075  7A                MOV      A,D        ;GET HIGH ORDER.
1076  B3                ORA      E          ;AND LOW ORDER.
1077  00                NOP                ;TIMING ADJUST.
1078  00                NOP                ;TIMING ADJUST.
1079  20F5       JRNZ      TICK.R          ;REPEAT UNTIL ZERO.
107B  C9                RET                ;RETURN TO USER.

```

```

;*****

```

```

;*****
; $.READ IS THE READ-SECTOR COMMAND CONTROLLER.      *
;*****
    
```

```

107C    CD 10EF    $.READ: CALL    SELECT    ;SELECT DRIVE ROUTINE.
107F    CD 1131          CALL    SEEK      ;SEEK TRACK, SET CTLS.
1082    2003          JRNZ    ..EXIT    ;DRIVE OR SEEK ERROR.
1084    CD 1204          CALL    RD.SEC    ;READ DISK SECTOR.
1087    C3 103F    ..EXIT: JMP     FETCH    ;GET NEXT COMMAND.
    
```

```

;*****
; $.WRIT IS THE WRITE-SECTOR COMMAND CONTROLLER.      *
;*****
    
```

```

108A    CD 10EF    $.WRIT: CALL    SELECT    ;SELECT DRIVE ROUTINE.
108D    CD 1131          CALL    SEEK      ;SEEK TRACK, SET CTLS.
1090    2003          JRNZ    ..EXIT    ;DRIVE OR SEEK ERROR.
1092    CD 122F          CALL    WR.SEC    ;WRITE DISK SECTOR.
1095    C3 103F    ..EXIT: JMP     FETCH    ;GET NEXT COMMAND.
    
```

```

;*****
; $.FORM IS THE FORMAT-TRACK COMMAND CONTROLLER.      *
;*****
    
```

```

1098    CD 10EF    $.FORM: CALL    SELECT    ;SELECT DRIVE NUMBER.
109B    3A 1373          LDA     CB.SEC    ;LOAD FORMAT FLAGS.
109E    DD7702          MOV     DV.FLG(X),A ;RESET DRIVE FLAGS.
10A1    CD 1131          CALL    SEEK      ;SEEK TRACK, SET CTLS.
10A4    2003          JRNZ    ..EXIT    ;DRIVE OR SEEK ERROR.
10A6    CD 125A          CALL    WR.TRK    ;WRITE DISK TRACK.
10A9    C3 103F    ..EXIT: JMP     FETCH    ;GET NEXT COMMAND.
    
```

```

;*****
; $.LGON IS THE DRIVE LOG-ON COMMAND CONTROLLER      *
;*****
    
```

```

10AC    CD 10EF    $.LGON: CALL    SELECT    ;SELECT DRIVE NUMBER.
10AF    AF          XRA     A           ;ZERO REGISTER A.
10B0    32 1372          STA     CB.TRK    ;SET TRACK AT 0.
10B3    3C          INR     A           ;NOW A REG IS 1.
10B4    32 1373          STA     CB.SEC    ;SET SECTOR TO ID.
10B7    CD 1131          CALL    SEEK      ;SEEK TRACK, SET CTLS.
10BA    2008          JRNZ    ..EXIT    ;DRIVE OR SEEK ERROR.
10BC    CD 1204          CALL    RD.SEC    ;READ ID SECTOR.
10BF    2003          JRNZ    ..EXIT    ;READ ERROR DETECTED.
10C1    CD 12D7          CALL    LOG.ON    ;LOG ON DISK DRIVE.
10C4    C3 103F    ..EXIT: JMP     FETCH    ;GET NEXT COMMAND.
    
```

```

;*****
    
```

```

;*****
; $.ADDR IS THE READ-ADDRESS COMMAND CONTROLLER.      *
;*****
    
```

```

10C7    3EFF    $.ADDR: MVI    A,OFFH    ;LOAD ALL ONES.
10C9    32 1377    STA    CB.STS    ;STORE ERRORS.
10CC    C3 103F    JMP    FETCH    ;NOT IMPLEMENTED.
    
```

```

;*****
; $.LIST IS A LIST DEVICE COMMAND CONTROLLER.          *
;*****
    
```

```

10CF    CD 12F4    $.LIST: CALL   LST.OT    ;SEND CHAR TO LIST.
10D2    C3 103F    JMP    FETCH    ;GET NEXT COMMAND.
    
```

```

;*****
; $.LSTT CHECKS LIST DEVICE STATUS                      *
;*****
    
```

```

10D5    DB00    $.LSTT: IN     BL.STS    ;GET BOARD STATUS.
10D7    E610    ANI    BS.EIA    ;TEST READY BIT.
10D9    CA 10DE    JZ     ..EXIT    ;IF ZERO GOTO EXIT.
10DC    3EFF    MVI    A,OFFH    ;LOAD ALL ONES.
10DE    32 1377    ..EXIT: STA   CB.STS    ;STORE STATUS.
10E1    C3 103F    JMP    FETCH    ;GET NEXT COMMAND.
    
```

```

;*****
; $.IDLE IS THE IDLE COMMAND CONTROLLER.                *
;*****
    
```

```

10E4    DB00    $.IDLE: IN     BL.STS    ;INPUT BOARD STATUS.
10E6    E608    ANI    BS.INT    ;CHECK HOST INTERRUPT.
10E8    28FA    JRZ   $.IDLE    ;REPEAT IDLE CHECK.
10EA    DB20    IN     XP.IRR    ;RESET INTERRUPT REQ.
10EC    C3 103F    JMP    FETCH    ;GET NEXT COMMAND.
    
```

```

;*****
    
```

```

;*****
; THE FOLLOWING SUBROUTINE SELECTS REQUESTED DRIVE *
; NUMBER 0-3 (A-D). BEFORE DRIVE SELECTION, THE DRIVE *
; MOTOR CONTROL STATE IS TESTED AND ENABLED IF NEEDED.*
; INDEX REGISTER X IS SET POINTING TO THE REQUESTED *
; DRIVE TABLE ENTRY. THE DRIVE IS THEN SELECTED. *
;*****

```

```

;***** ( MOTOR CHECK ROUTINE )*****

```

```

10EF    DB00          SELECT: IN      BL.STS      ;BOARD LEVEL STATUS.
10F1    E620          ANI      BS.MOF      ;CHECK MOTOR STATE.
10F3    DB40          IN      XP.MTX      ;START OR EXTEND TIMER.
10F5    2805          JRZ      ..CKDV     ;IF WAS ON, NO STARTUP.
10F7    ED5B 1016    LDED     TM.MTO      ;MOTOR STARTUP DELAY.
10FB    CF           WAIT           ;PROGRAMMABLE DELAY.

```

```

;***** ( NEW SELECTION CHECK )*****

```

```

10FC    3A 1371      ..CKDV: LDA     CB.DRV      ;LOAD DRIVE NUMBER.
10FF    E603          ANI     BC.DSN      ;GET DRIVE NUMBER.
1101    DDBE00       CMP     DV.NBR(X)   ;CURRENTLY SELECTED?
1104    C8           RZ           ;RETURN IF DRV SAME.

```

```

;***** ( SET TABLE POINTER )*****

```

```

1105    DD21 1342    LXI     X,DV.TBL    ;DRIVE TABLE ADDR.
1109    11 0004      LXI     D,DV.DES     ;DRIVE ENTRY SIZE.
110C    3D          ..NEXT: DCR     A           ;DECREMENT DRV NO.
110D    FA 1114      JM      ..DSLTL    ;IF S=1 EXIT.
1110    DD19          DADX     D           ;POINT NEXT DRIVE.
1112    18F8          JMPR    ..NEXT      ;TRY THIS DRIVE.

```

```

;***** ( DESELECT OLD DRIVE )*****

```

```

1114    0610          ..DSLTL: MVI     B,DC.HDU    ;LOAD UNLOAD R/W HEAD.
1116    CD 1018      CALL    EX.HCF      ;FD179X-02 TYPE 1 CMND.
1119    3A 1333      LDA     SV.CTL      ;BL.CTL LAST ISSUED.
111C    E6FB          ANI     #BC.DSE     ;DRIVE SELECT DSBLD.
111E    D300          OUT     BL.CTL      ;ISSUE DESELECT.

```

```

;***** ( SELECT NEW DRIVE )*****

```

```

1120    E6FC          ANI     #BC.DSN      ;STRIP OFF DRIVE NMBR.
1122    DDB600       ORA     DV.NBR(X)   ;OR IN NEW DRIVE NMBR.
1125    D300          OUT     BL.CTL      ;OUTPUT DRIVE NMBR.
1127    F604          ORI     BC.DSE      ;SET DRV ENABLE BIT.
1129    D300          OUT     BL.CTL      ;ENABLE NEW DRIVE.
112B    E607          ANI     BC.DSN!BC.DSE ;NOW JUST DRIVE ENBLD.
112D    32 1331      STA     SV.DRV      ;SAVE DRIVE SELECT.
1130    C9           RET           ;DRIVE IS SELECTED.

```

```

;*****

```

```

;*****
; THE FOLLOWING SUBROUTINE PERFORMS THE TRACK SEEK *
; OPERATION. AFTER THE SEEK OPERATION, THE DENSITY *
; AND PRE-COMPENSATION CONTROLS ARE SET. *
;*****
    
```

```

;***** ( HEAD LOADING ) *****
    
```

```

1131 CD 1024 SEEK: CALL EX.STS ;GET DRIVE STATUS.
1134 E6A0 ANI DM.HDL!DM.DNR ;CHECK HEAD AND READY.
1136 FA 1174 JM ..NRDY ;DRIVE NOT READY EXIT.
1139 C2 1146 JNZ ..DTAS ;BYPASS IF HEAD LOADED.
113C 0618 MVI B,DC.HDL ;HEAD-LOAD COMMAND.
113E CD 1018 CALL EX.HCF ;EXEC FD179X-02 TYPE 1.
1141 ED5B 1010 LDED TM.HLD ;SET HEAD-LOAD DELAY.
1145 CF WAIT ;PROGRAMMABLE DELAY.
    
```

```

;***** ( DETERMINE TRACK NMBR AND SIDE ) *****
    
```

```

1146 DB00 ..DTAS: IN BL.STS ;INPUT BOARD STATUS.
1148 E640 ANI BS.TSD ;TEST DISK SIDES FLAG.
114A 3A 1372 LDA CB.TRK ;GET LOGICAL TRACK NO.
114D 6F MOV L,A ;SAVE LOGICAL TRACK.
114E C2 1152 JNZ ..NDBL ;SKIP IF NOT DBL SIDED
1151 1F RAR ;DIV BY 2 DOUBLE SIDE
1152 32 1335 ..NDBL: STA PH.TRK ;STORE PHYSICAL TRACK.
1155 67 MOV H,A ;SAVE PHYSICAL NUMBER.
1156 3A 1331 LDA SV.DRV ;LOAD DRV NMBR ENABLED.
1159 3002 JRNC ..SIDO ;SKIP NEXT IF SIDE 0.
115B F620 ORI BC.SD1 ;OR IN SELECT SIDE 1.
115D 32 1332 ..SIDO: STA SV.DAS ;STORE DRV AND SIDE EN.
1160 57 MOV D,A ;SAVE DRV AND SIDE EN.
1161 7C MOV A,H ;LOAD PHYSICAL NUMBER.
1162 DD9601 SUB DV.TRK(X) ;TRACK OFFSET TESTED.
1165 2021 JRNZ ..SEEK ;IF OFFTRACK, DO SEEK.
1167 DB00 IN BL.STS ;INPUT BOARD STATUS.
1169 E640 ANI BS.TSD ;TEST DISK SIDES FLAG
116B CA 117B JZ ..DSID ;GOTO DOUBLE SIDE CTL.
    
```

```

;***** ( SINGLE SIDED DISKETTE ) *****
    
```

```

116E DD7E03 ..SSID: MOV A,DV.CTL(X) ;GET PREVIOUS CONTROLS.
1171 C3 11EA JMP ..EXIT ;SET CONTROLS / EXIT.
    
```

```

;***** ( DRIVE NOT READY EXIT ) *****
    
```

```

1174 3E80 ..NRDY: MVI A,CS.DNR ;DRIVE NOT READY FLAG.
1176 32 1377 STA CB.STS ;STORE ERROR STATUS.
1179 A7 ANA A ;SET NOT ZERO FLAG.
117A C9 RET ;ERROR EXIT.
    
```

```

;*****
    
```

***** (DISKETTE IS DOUBLE SIDED)*****

```

117B 7C      ..DSID: MOV    A,H      ;GET PHYSICAL TRK NMBR.
117C A7      ANA    A          ;TEST IF TRACK ZERO.
117D 283B    JRZ    ..DCTL     ;IF ZERO, RESET CNTLS.
117F DD7E03   MOV    A, DV.CTL(X) ;LOAD OLD DRV CTLS.
1182 E6DF    ANI    #BC.SD1    ;STRIP OFF SIDE CMND.
1184 B2      ORA    D          ;OR IN NEW SIDE CMND.
1185 C3 11EA  JMP    ..EXIT    ;SET CONTROLS / EXIT.
    
```

***** (SET DIRECTION AND COUNT STEPS)*****

```

1188 F5      ..SEEK: PUSH   PSW      ;SAVE REG A AND FLGS.
1189 ED5B 1338 LDED  TM.SAW    ;STEP AFTER WRITE.
118D CF      WAIT                    ;PROGRAMMABLE DELAY.
118E F1      POP    PSW      ;RESTORE A AND FLGS.
118F 380A    JRC    ..SOUT    ;IF CARRY STEP OUT.
1191 6F      ..SIN: MOV    L,A      ;MOVE OFFSET TO L.
1192 3A 1331 LDA  SV.DRV    ;DRIVE SELECT BITS.
1195 F620    ORI    BC.INW   ;SET STEP DIRC IN.
1197 D300    OUT   BL.CTL  ;OUTPUT CONTROL.
1199 180B    JMPR  ..STEP   ;GOTO STEP ROUTINE.
119B ED44    ..SOUT: NEG                    ;COMPLEMENT OFFSET.
119D FA 11FD JM    ..HOME   ;BETTER HOME DRV.
11A0 6F      MOV    L,A      ;MOVE OFFSET TO L.
11A1 3A 1331 LDA  SV.DRV    ;DRIVE SELECT BITS.
11A4 D300    OUT   BL.CTL  ;SET DIRECTION OUT.
11A6 DB08    ..STEP: IN    XP.STP  ;ISSUE STEP PULSE.
11A8 ED5B 1012 LDED  TM.STP   ;STEP DELAY TIME.
11AC CF      WAIT                    ;PROGRAMMABLE DELAY.
11AD 2D      DCR    L          ;DECREMENT STEPS.
11AE 20F6    JRNZ  ..STEP   ;REPEAT OPERATION.
11B0 3A 1332 LDA  SV.DAS   ;LOAD DRV AND SIDE.
11B3 D300    OUT   BL.CTL  ;OUTPUT CONTROL.
11B5 ED5B 1014 LDED  TM.ALS   ;MORE AFTER LAST STP.
11B9 CF      WAIT                    ;PROGRAMMABLE DELAY.
    
```

***** (CONTROL DETERMINATION)*****

```

11BA 3A 1372  ..DCTL: LDA    CB,TRK      ;LOAD LOGICAL TRACK.
11BD FE01      CPI    1            ;COMPARE AGAINST 1.
11BF 3820      JRC    ..SDEN      ;TRACK 0 IS SDENS.
11C1 3E04      MVI    A,DF.DTD    ;DATA TRK DENS FLG.
11C3 C2 11C8   JNZ    ..DTST      ;GOTO TEST DENSITY.
11C6 3E02      MVI    A,DF.T1D    ;TRACK 1 DENS FLAG.
11C8 DDA602   ..DTST: ANA    DV,FLG(X) ;TEST DENSITY FLAGS.
11CB CA 11E1   JZ     ..SDEN      ;IF ZERO, THEN SDENS.
11CE 3A 1335   ..DDEN: LDA    PH,TRK  ;LOAD PHYSICAL TRACK.
11D1 FE1A      CPI    TRK.OB     ;TEST OUTSIDE BOUNDRY.
11D3 06D0      MVI    B,BC.DDS!BC.PCL ;DDENS AND LOW PRECOMP.
11D5 380C      JRC    ..CTLS     ;SET FOR OUTSIDE TRKS.
11D7 FE34      CPI    TRK.IB     ;TEST INSIDE BOUNDRY.
11D9 0690      MVI    B,BC.DDS!BC.PCM ;DDENS AND MED PRECOMP.
11DB 3806      JRC    ..CTLS     ;JUMP TO CONTROLS SET.
11DD 0650      MVI    B,BC.DDS!BC.PCH ;DDENS AND MAX PRECOMP.
11DF 1802      JMPR   ..CTLS     ;JUMP TO CONTROLS SET.

11E1 06C0      ..SDEN: MVI    B,BC.SDS!BC.PCL ;SDEN AND PC-LOW.
    
```

***** (SET CONTROL VALUES AND EXIT)*****

```

11E3 3A 1332   ..CTLS: LDA    SV,DAS      ;GET DRIVE AND SIDE.
11E6 B0         ORA    B          ;SET PRECOMP AND DENS.
11E7 DD7703    MOV    DV,CTL(X),A   ;SAVE CONTROLS FOR DRV.
11EA D300      ..EXIT: OUT   BL,CTL  ;OUTPUT CONTROLS.
11EC 32 1333   STA    SV,CTL  ;SAVE THESE CONTROLS.
11EF 3A 1335   LDA    PH,TRK  ;PHYSICAL TRACK NMBR.
11F2 DD7701    MOV    DV,TRK(X),A   ;SET DRIVE TABLE.
11F5 3A 1372   LDA    CB,TRK  ;LOGICAL TRACK NMBR.
11F8 A9         XRA    C          ;INVERT (1791-01).
11F9 D305      OUT    WD,TRK  ;SET TRACK REGISTER.
11FB AF        XRA    A          ;SET ZERO FLAG.
11FC C9         RET                    ;RETURN TO CALLER.
    
```

***** (CALIBRATE TRACK NUMBER)*****

```

11FD CD 12A6   ..HOME: CALL   HOME.D     ;HOME SELECTED DRIVE.
1200 C0        RNZ                    ;EXIT SEEK, HOME BAD.
1201 C3 1146   JMP    ..DTAS     ;NOW SEEK TRACK.
    
```

```

;*****
; RD.SEC IS THE SUBROUTINE THAT INTERACTS WITH THE *
; 179X-02 DURING READ SECTOR OPERATIONS. THIS SECTION *
; INITIATES THE DISK TRANSFER, SERVICES THE CONTROLLER*
; CHIP DURING DATA TRANSFER, AND TERMINATES OPERATION *
; WHEN FINISHED. ERROR DETECTION IS IMPLEMENTED AND *
; RETRIES ARE EXECUTED IF DATA ERRORS ARE DETECTED. *
;*****

```

```

;***** ( INITIALIZE READ OPERATION )*****

```

```

1204 AF RD.SEC: XRA A ;ZERO A REGISTER.
1205 32 1330 STA ERR.CT ;ZERO ERROR COUNT.
1208 3A 1373 LDA CB.SEC ;LOAD SECTOR NMBR.
120B A9 XRA C ;INVERT (1791-01).
120C D306 OUT WD.SEC ;SET SECTOR REGISTER.
120E FD21 1223 ..RTRY: LXI Y,..NMI ;LOAD NMI VECTOR.
1212 2A 132E LHLD BUF,ST ;BUFFER START.
1215 3E88 MVI A,DC.RDS ;READ SECTOR COMMA
1217 A9 XRA C ;INVERT (1791-01).
1218 D304 OUT WD.CMD ;ISSUE READ COMMAND

```

```

;***** ( DATA TRANSFER LOOP )*****

```

```

121A DB80 ..REPT: IN XP.DSH ;HOLD FOR DATA
121C DB07 IN WD.DTA ;INPUT DATA.
121E A9 XRA C ;INVERT (1791-01).
121F 77 MOV M,A ;PUT INTO BUFFER
1220 23 INX H ;BUMP BUFF POINTER
1221 18F7 JMPR ..REPT ;GO FOR ANOTHER

```

```

;***** ( CHECK STATUS )*****

```

```

1223 E69D ..NMI: ANI DM.RER ;TEST FOR ERRORS.
1225 32 1377 STA CB.STS ;SAVE READ STATUS.
1228 C8 RZ ;RETURN COMPLETE.
1229 CD 1279 CALL CHK.RT ;CHECK ABOUT RETRYS.
122C 28E0 JRZ ..RTRY ;PERFORM RETRY.
122E C9 RET ;ERROR RETURN.

```

```

;*****

```



```

;*****
; WR.SEC SUBROUTINE INTERACTS WITH THE FD179X-02 *
; DURING WRITE SECTOR OPERATIONS, THIS SECTION *
; INITIATES THE DISK TRANSFER, SERVICES THE CONTROLLER*
; CHIP, AND TERMINATES THE OPERATION. ERROR DETECTION *
; IS IMPLEMENTED. *
;*****

```

```

;***** ( INITIALIZE WRITE OPERATION )*****

```

```

122F    AF          WR.SEC: XRA      A          ;ZERO REGISTER.
1230    32 1330     STA      ERR.CT      ;SET ERROR COUNTER.
1233    3A 1373     LDA      CB.SEC      ;LOAD SECTOR NMBR.
1236    A9          XRA      C          ;INVERT (1791-01).
1237    D306       OUT      WD.SEC      ;SET SECTOR REGISTER.
1239    FD21 124E   ..RTRY: LXI      Y,..NMI ;SET NMI RETURN.
123D    2A 132E     LHLD     BUF.ST      ;BUFFER START.
1240    3EAB       MVI      A,DC.WRS    ;LOAD WRITE SECTOR CMD.
1242    A9          XRA      C          ;INVERT (1791-01).
1243    D304       OUT      WD.CMD      ;ISSUE COMMAND.

```

```

;***** ( DATA TRANSFER LOOP )*****

```

```

1245    DB80       ..REPT: IN       XP.DSH ;HOLD FOR DATA REQ.
1247    7E         MOV      A,M        ;GET DATA BYTE.
1248    A9         XRA      C          ;INVERT (1791-01).
1249    D307       OUT      WD.DTA      ;OUTPUT DATA BYTE.
124B    23         INX      H          ;INCREMENT BUFF POINTER
124C    18F7       JMPR     ..REPT     ;REPEAT SEQUECE

```

```

;***** ( CHECK STATUS )*****

```

```

124E    E6FD       ..NMI: ANI      DM.WER ;TEST FOR WRITE ERRORS.
1250    32 1377     STA      CB.STS    ;STORE WRITE STATUS.
1253    C8         RZ          ;RETURN COMPLETE.
1254    CD 1279     CALL     CHK.RT    ;CHECK ABOUT RETRYS.
1257    28E0       JRZ      ..RTRY    ;PERFORM RETRY.
1259    C9         RET          ;ERROR RETURN.

```

```

;*****

```

```
*****  
; WR.TRK IS THE SUBROUTINE WHICH INITIATES A FORMAT *  
; TRACK COMMAND (WRITE-TRACK 179X-02 TYPE 3). THE *  
; FORMATTING BYTE STREAM IS PROVIDED BY A PROGRAM *  
; WHICH MUST BE PRESENT IN THE FORMAT BUFFER. *  
*****
```

```
***** ( INITIALIZE WRITE TRACK )*****
```

```
125A FD21 1266 WR.TRK: LXI Y,..NMI ;LOAD NMI VECTOR.  
125E 3EFO MVI A,DC.WRT ;WRITE TRACK CMND.  
1260 A9 XRA C ;INVERT (1791-01).  
1261 D304 OUT WD.CMD ;ISSUE COMMAND.  
1263 C3 1708 JMP FMT.PS ;FORMAT PROG START.
```

```
***** ( CHECK COMPLETION STATUS )*****
```

```
1266 E6E4 ..NMI: ANI DM.FER ;TEST FOR ERRORS.  
1268 47 MOV B,A ;HOLD THIS STATUS.  
1269 DB00 IN BL.STS ;INPUT BOARD STATUS.  
126B E640 ANI BS.TSD ;TEST TWO SIDED BIT.  
126D 78 MOV A,B ;RESTORE STATUS TO A.  
126E 2002 JRNZ ..EXIT ;NOT ZERO IS ONE SIDED.  
1270 F601 ORI CS.TSD ;OR IN TWO SIDED FLAG.  
1272 32 1377 ..EXIT: STA CB.STS ;STORE FORMAT STATUS.  
1275 22 137A SHLD CW.LNG ;DISPLAY TRAIL BYTES.  
1278 C9 RET ;RETURN TO USER.
```

```
*****
```

```

;*****
; CHK.RT IS THE SUBROUTINE USED BY RD.SEC AND
; WR.SEC TO COUNT RETRY OPERATIONS AND PERFORM A
; RE-SEEK OPERATION WHEN NEEDED.
;*****

```

```

;***** ( CHECK IF RECOVERABLE )*****

```

```

1279 E680 CHK.RT: ANI DM.DNR ;TEST NOT READY BIT.
127B 2028 JRNZ ..EXIT ;CAN NOT RECOVER.
127D 3A 1376 LDA CB.MOD ;GET COMMAND MODE.
1280 E680 ANI CM.NRT ;NO RETRYS CHECK.
1282 2021 JRNZ ..EXIT ;SHOULD NOT RECOVER.
1284 DB40 IN XP.MTX ;MOTOR TIME EXTEND.

```

```

;***** ( RECORD RETRY )*****

```

```

1286 3A 1330 LDA ERR.CT ;GET ERROR COUNT.
1289 3C INR A ;INCREMENT.
128A 32 1330 STA ERR.CT ;STORE NEW COUNT.
128D FE05 CPI RTY.SK ;SHOULD TRY SEEK?
128F 2008 JRNZ ..CKLS ;IF NOT, CHECK LAST.

```

```

;***** ( REPOSITION R/W HEAD )*****

```

```

1291 CD 12A6 CALL HOME.D ;HOME SELECTED DRIVE.
1294 200F JRNZ ..EXIT ;ERROR EXIT.
1296 CD 1131 CALL SEEK ;SEEK DESIRED TRACK.

```

```

;***** ( HOLD READ GATE FOR 3/4 REVOLUTION )*****

```

```

1299 FE09 ..CKLS: CPI RTY.LS ;WAS THIS THE LAST.
129B 2807 JRZ ..STNZ ;ERROR LAST RETRY.
129D ED5B 1336 LDED TM.PLD ;PHASE LOCK DELAY.
12A1 CF WAIT ;PROGRAMMABLE DELAY.
12A2 AF XRA A ;CLEAR FOR RETRY.
12A3 C9 RET ;TRY AGAIN EXIT.

```

```

;***** ( ERROR EXIT )*****

```

```

12A4 3C ..STNZ: INR A ;SET NOT ZERO.
12A5 C9 ..EXIT: RET ;ERROR EXIT.

```

```

;*****

```

```

;*****
; HOME.D IS THE SUBROUTINE THAT STEPS THE DISK DRIVE *
; R/W HEAD OUTWARD UNTIL THE TRACK 0 FLAG BECOMES *
; ACTIVE OR 255 STEPS HAVE BEEN ISSUED. *
;*****

```

```

;***** ( RESTORE R/W HEAD )*****

```

```

12A6      3A 1331      HOME.D: LDA      SV.DRV      ;LOAD DRV NMBR ENABLED.
12A9      D300                OUT      BL.CTL      ;ISSUE CONTROLS.
12AB      32 1333      STA      SV.CTL      ;AND SAVE THESE.
12AE      2EFF                MVI      L,255      ;SET STEP COUNTER.
12B0      CD 1024      ..STEP: CALL     EX.STS      ;CHECK DISK STATUS.
12B3      E604                ANI      DM.TKO      ;INSPECT TRACK 0 FLG.
12B5      200C                JRNZ     ..EXIT      ;IF SET, GO ..EXIT.
12B7      2D                DCR      L          ;DECREMENT STEP COUNT.
12B8      2816                JRZ      ..EROR      ;ERROR IF 255 STEPS.
12BA      DB08                IN       XP.STP      ;ISSUE STEP PULSE.
12BC      ED5B 1012      LDED     TM.STP      ;LOAD STEP DELAY.
12C0      CF                WAIT     ;PROGRAMMABLE DELAY.
12C1      18ED                JMPR     ..STEP      ;TRY STEPPING AGAIN.

```

```

;***** ( DRIVE IS RESTORED )*****

```

```

12C3      ED5B 1014      ..EXIT: LDED     TM.ALS      ;TIME AFTER LAST STEP.
12C7      CF                WAIT     ;PROGRAMMABLE DELAY.
12C8      79                MOV      A,C          ;GET WD TRK 0 VALUE.
12C9      D305                OUT     WD.TRK      ;ZERO TRACK REGISTER.
12CB      AF                XRA     A            ;ZERO A REG, SET FLAG.
12CC      DD7701           MOV     DV.TRK(X),A    ;SET TRACK VALUE.
12CF      C9                RET      ;RETURN TO CALLER.

```

```

;***** ( TRACK 0 NOT FOUND )*****

```

```

12D0      3E02      ..EROR: MVI     A,CS.HME      ;LOAD HOME ERROR FLAG.
12D2      32 1377      STA     CB.STS      ;STORE ERROR STATUS.
12D5      A7                ANA     A            ;SET RETURN FLAGS.
12D6      C9                RET      ;RETURN TO CALLER.

```

```

;*****

```

```

;*****
; LOG.ON IS THE SUBROUTINE THAT READS THE IDENTITY *
; SECTOR FROM THE DISKETTE AND MAKES THE NEEDED *
; ENTRIES INTO THE DRIVE TABLE. THE SECTOR DATA IS *
; ALSO LEFT IN THE SECTOR BUFFER FOR BIOS TO FINISH *
; THE LOG-ON OPERATION. *
;*****
    
```

***** (CHECK JADE IDENTITY)*****

```

12D7      11 133A      LOG.ON: LXI      D,JADEID      ;ID ADDRESS LOADED.
12DA      21 1380      LXI      H,ID.LBL      ;SECTOR ID ADDRESS.
12DD      0608      MVI      B,ID.SZE      ;ID LABEL SIZE.
12DF      1A      ..CKJI: LDAX      D      ;GET CHARACTER.
12E0      BE      CMP      M      ;CHECK AGAINST DISK.
12E1      200B      JRNZ      ..3740      ;IF DIFFERENT: 3740.
12E3      13      INX      D      ;CHECK NEXT.
12E4      23      INX      H      ;CHECK NEXT.
12E5      10F8      DJNZ      ..CKJI      ;REPEAT OPERATION.
    
```

***** (LOG-ON JADE FORMAT)*****

```

12E7      3A 13B1      LDA      ID.FLG      ;SIDE AND DENSITIES.
12EA      DD7702      MOV      DV.FLG(X),A      ;STORE IN DRIVE TBL.
12ED      C9      RET      ;RETURN TO CALLER.
    
```

***** (ASSUME 3740 FORMAT)*****

```

12EE      3E00      ..3740: MVI      A,ID.FLD      ;SIDE AND DENSITIES.
12F0      DD7702      MOV      DV.FLG(X),A      ;STORE IN DRIVE TBL.
12F3      C9      RET      ;RETURN TO CALLER.
    
```

```

;*****
; THE FOLLOWING ROUTINE SENDS ONE 8 BIT CHARACTER OUT *
; THE EIA LEVEL TRANSMISSION BIT, SET FOR BAUD RATE. *
;*****

```

```

;***** ( SET UP FOR TRANSMISSION )*****

```

```

12F4 DB00 LST.OT: IN BL.STS ;GET BOARD STATUS.
12F6 E610 ANI BS.EIA ;TEST LIST READY BIT.
12F8 CA 12F4 JZ LST.OT ;WAIT READY (JZ/JNZ).
12FB 3A 1375 LDA CB.CHR ;GET LIST CHARACTER.
12FE 2F CMA ;COMPLEMENT ACUMULATOR.
12FF 5F MOV E,A ;CHARACTER TO E REG.
1300 3A 1333 LDA SV.CTL ;LAST CONTROLS USED.

```

```

;***** ( SEND THE START BIT )*****

```

```

1303 37 STC ;SET CARRY BIT.
1304 CD 131A CALL BIT.OT ;OUTPUT START BIT.
1307 00 NOP ;EQUALIZE TIMING.
1308 00 NOP ;EQUALIZE TIMING.
1309 1608 MVI D,8 ;NUMBER OF DATA BITS.

```

```

;***** ( SEND EACH DATA BIT )***** ( 39 CYCLE LOOP )***

```

```

130B CB0B ..DATA: RRCR E ;ROTATE E REG RIGHT.
130D CD 131A CALL BIT.OT ;SEND ONE DATA BIT.
1310 15 DCR D ;ONE LESS BIT TO DO.
1311 C2 130B JNZ ..DATA ;REPEAT IF MORE BITS.

```

```

;***** ( SEND STOP BIT )*****

```

```

1314 00 NOP ;EQUALIZE TIMING.
1315 A7 ANA A ;CLEAR CARRY FLAG.
1316 CD 131A CALL BIT.OT ;SEND STOP BIT.
1319 C9 RET ;RETURN TO CALLER.

```

```

;***** ( SET EIA BIT AND OUTPUT )***** ( 39 CYCLES )****

```

```

131A DA 1322 BIT.OT: JC ..ONE ;IF CARRY, SET TO ONE.
131D CB9F RES 3,A ;ZERO EIA IN ACUM REG.
131F C3 1327 JMP ..OUT ;GO TO OUTPUT PORT.
1322 CBDF ..ONE: SET 3,A ;SET EIA IN ACUM.
1324 C3 1327 JMP ..OUT ;EQUALIZE TIMING.
1327 D300 ..OUT: OUT BL.CTL ;SEND ACUM TO PORT.

```

```

;***** ( SET DELAY FOR BAUDRATE )*****

```

```

1329 0619 MVI B,BAUD.C ;LOAD TIMING CSNT.
132B 10FE DJNZ . ;DELAY FOR BIT.
132D C9 RET ;RETURN TO LST CALL.

```

```

;*****

```

```

;*****
; PROGRAM STORAGE LOACTIONS
;*****

132E    1380    BUF.ST: .WORD    BUF.BG    ;BUFFER STARTING ADDRESS.
1330    00      ERR.CT: .BYTE    0      ;RETRY ERROR COUNTER.

1331    00      SV.DRV: .BYTE    0      ;BL.CTL DRIVE BITS.
1332    00      SV.DAS: .BYTE    0      ;BL.CTL DRIVE AND SIDE BITS.
1333    00      SV.CTL: .BYTE    0      ;BL.CTL LAST ISSUED.
1334    00      SV.STS: .BYTE    0      ;FD179X-02 STATUS VALUE.

1335    00      PH.TRK: .BYTE    0      ;PHYSICAL TRACK NUMBER.

;*****
; TIMING VALUES - 0.1 MS INCREMENTS
;*****

1336    04B0    TM.PLD: .WORD    1200    ;PHASE LOCK RECOVERY.
1338    000A    TM.SAW: .WORD    10      ;STEP AFTER WRITING.
0018    0000    TM.SDD  ==      24      ;SIDE SELECT DELAY.

;*****
; DISKETTE IDENTITY LABEL
;*****

133A    4A6164652044 JADEID: .ASCII  "JADE DD "    ;DISKETTE ID LABEL.
0008                                ID.SZE  ==      (. - JADEID) ;ID LABEL SIZE.

1380                                ID.LBL  ==    BUF.BG+0000H    ;ID SECTOR LABEL.
13A0                                ID.BLK  ==    ID.LBL+0020H    ;ID BLOCK AREA.
13B1                                ID.FLG  ==    ID.BLK+0011H    ;DISKETTE FLAGS.
0000                                ID.FLD  ==    00000000B    ;3740 FLAGS.

;*****

```

```
*****  
; DRIVE TABLE AREA DEFINED  
*****  
  
***** ( DRIVE TABLE ENTRIES ) *****  
  
0000      DV.NBR ==      0      ;CURRENT DRIVE NUMBER.  
0001      DV.TRK ==      1      ;CURRENT TRACK NUMBER.  
0002      DV.FLG ==      2      ;SIDE AND DENSITY FLAGS  
0003      DV.CTL ==      3      ;LAST CONTROLS USED.  
  
***** ( DRIVE TABLE AREA ) *****  
  
1342      DV.TBL ==      .      ;DRIVE TABLE BEGGINING ADDRESS.  
  
1342      00FF02C4      DT.DE0: .BYTE 0,255,DF.DFL,0C4H      ;DRIVE 0.  
1346      01FF02C5      DT.DE1: .BYTE 1,255,DF.DFL,0C5H      ;DRIVE 1.  
134A      02FF02C6      .BYTE 2,255,DF.DFL,0C6H      ;DRIVE 2.  
134E      03FF02C7      .BYTE 3,255,DF.DFL,0C7H      ;DRIVE 3.  
1352      04FF0000      DT.DED: .BYTE 4,255,0,0      ;DUMMY.  
  
0004      DV.DES ==      DT.DE1-DT.DE0      ;EACH DRIVE ENTRY SIZE.  
  
***** ( FLAG BIT DEFINITIONS ) *****  
  
0002      DF.T1D == 00000010B      ;TRACK 1 DENSITY (1 = DOUBLE).  
0004      DF.DTD == 00000100B      ;DATA TRACKS DENSITY (1 = DD).  
0008      DF.TSD == 00001000B      ;TWO SIDED ( 1 = TWO SIDES).  
0002      DF.DFL == DF.T1D      ;DEFAULT FLAGS.  
  
*****
```



```

;*****
; THE FOLLOWING AREA IS DEFINED AS THE COMMAND BLOCK. *
; THIS AREA IS RESERVED FOR SPECIFICATION BY THE HOST *
; SYSTEM FOR ALL DISK OPERATIONS.  CONTROLLER STATUS *
; AT COMPLETION OF OPERATION IS PRESENT IN THIS AREA. *
;*****

```

```

1370          .LOC      CMD.BK  ;COMMAND BLOCK.

1370      00      CB.CMD: .BYTE  0      ;CONTROL COMMAND.
1371      00      CB.DRV: .BYTE  0      ;DRIVE NUMBER.
1372      00      CB.TRK: .BYTE  0      ;LOGICAL TRACK NUMBER.
1373      00      CB.SEC: .BYTE  0      ;SECTOR NUMBER.
1374      00      CB.FFG: .BYTE  0      ;FORMAT FLAGS.
1375      00      CB.CHR: .BYTE  0      ;EIA CHARACTER.
1376      00      CB.MOD: .BYTE  0      ;MODE SELECTS.
1377      00      CB.STS: .BYTE  0      ;CONTROLLER STATUS.

1378      0000    CW.LAD: .WORD  0      ;LOAD ADDRESS.
137A      0000    CW.LNG: .WORD  0      ;LOAD LENGTH

```

```

;***** ( MODE BIT DEFINITIONS )*****

```

```

0080      CM.NRT == 1000000B  ;NO RETRYS ( = 1 ).

```

```

;***** ( STATUS BIT DEFINITIONS )*****

```

```

0080      CS.DNR == 1000000B  ;DRIVE NOT READY.
0040      CS.WRP == 0100000B  ;WRITE PROTECTED.
0020      CS.BT5 == 0010000B  ;NOT ASSIGNED.
0010      CS.RNF == 0001000B  ;RECORD NOT FOUND.
0008      CS.CRC == 0000100B  ;CRC ERROR.
0004      CS.LDE == 0000010B  ;LOST DATA ERROR.
0002      CS.HME == 0000001B  ;DRIVE HOME ERROR.
0001      CS.TSD == 00000001B ;TWO SIDES FLAG (FORMAT).

```

```

;*****

```

```

;*****
; THIS SECTION RESIDES IN THE DCM SECTOR BUFFER. THIS *
; SECTION MOVES DCM FROM BANK 1 DOWN TO BANK 0. THE *
; C REGISTER IS SET FOR 1791-01 OR 1793-01. THE LAST *
; OPERATION IS TO READ THE BIOS LOADER SECTOR TO *
; OVERLAY THIS INITIALIZATION SEQUENCE. BIOS LOADER *
; THEN READ BIOS INTO BANK 1 AND HALTS. *
;*****

```

```

;***** ( EXECUTES IN BANK 1 )*****

```

```

1380          .LOC      BUF,BG          ;RESIDES IN BUFFER.
1380      01 0400      INIT.B: LXI      B,BANK.L      ;SET BANK LENGTH.
1383      11 1000          LXI      D,BANK.0      ;SET DESTINATION.
1386      21 1400          LXI      H,BANK.1      ;SET SOURCE ADDR.
1389      EDB0          LDIR          ;MOVE BLOCK.
138B      C3 138E          JMP       ..DOWN      ;JUMP TO NEW IMAGE.

```

```

;***** ( NOW IN BANK 0, SET INT MODE )*****

```

```

138E      31 1370          ..DOWN: LXI      SP,TP.STK  ;SET STACK PNTR.
1391      ED56          IM1          ;INTERRUPT MODE 1.

```

```

;***** ( SET 1791-01/1793-01 )*****

```

```

1393      0E00          MVI      C,0          ;LOAD C REG ZERO.
1395      DB00          IN       BL.STS      ;BOARD STATUS.
1397      E601          ANI      BS.US0      ;TEST USER SW #1.
1399      2002          JRNZ     LD.BLT      ;SW OPEN - 1793.
139B      0EFF          MVI      C,OFFH      ;SW CLOSED - 1791.

```

```

;***** ( OVERLAY WITH BIOS LOADER TRANSIENT )*****

```

```

139D      DD21 1352      LD.BLT: LXI      X,DT.DED      ;INIT DRIVE TBL.
13A1      3E02          MVI      A,2          ;BIOS LOADER SECTOR.
13A3      32 1373          STA      CB.SEC      ;SET SECTOR VALUE.
13A6      DB40          IN       XP.MTX      ;MOTOR TIME EXTEND.
13A8      21 1380          LXI      H,BUF.BG      ;SET RETURN ADDR.
13AB      E5          PUSH     H          ;PUSH INTO STACK.
13AC      C3 1204          JMP       RD.SEC      ;GET BIOS LOADER.

```

```

;*****
.END

```

BANK.0 1000	BANK.1 1400	BANK.L 0400	BASE 1000
BAUD.C 0019	BC.DAS 0020	BC.DDE 0010	BC.DDS 0010
BC.DSA 0001	BC.DSB 0002	BC.DSE 0004	BC.DSN 0003
BC.EIA 0008	BC.INW 0020	BC.FCA 0040	BC.PCB 0080
BC.PCH 0040	BC.PCL 00C0	BC.FCM 0080	BC.SD1 0020
BC.SDS 0000	BIT.OT 131A	BL.CTL 0000	BL.STS 0000
BS.DCN 0080	BS.EIA 0010	BS.INT 0008	BS.MOF 0020
BS.TSD 0040	BS.TST 0004	BS.US0 0001	BS.US1 0002
BUF.BG 1380	BUF.ST 132E	CB.CHR 1375	CB.CMD 1370
CB.DRV 1371	CB.FFG 1374	CB.MOD 1376	CB.SEC 1373
CB.STS 1377	CB.TRK 1372	CHK.RT 1279	CMD.BK 1370
CM.DTA 1053	CM.MSK 0007	CM.NRT 0080	CS.BT5 0020
CS.CRC 0008	CS.DNR 0080	CS.HME 0002	CS.LDE 0004
CS.RNF 0010	CS.TSD 0001	CS.WRP 0040	CW.LAD 1378
CW.LNG 137A	DC.HDL 0018	DC.HDU 0010	DC.IFI 00D8
DC.RDA 00C0	DC.RDS 0088	DC.STS 00D0	DC.WRS 00A8
DC.WRT 00F0	DF.DFL 0002	DF.DTD 0004	DF.TID 0002
DF.TSD 0008	DM.DNR 0080	DM.FER 00E4	DM.HDL 0020
DM.LDE 0004	DM.RER 009D	DM.TKO 0004	DM.WER 00FD
DT.DE0 1342	DT.DE1 1346	DT.DED 1352	DV.CTL 0003
DV.DES 0004	DV.FLG 0002	DV.NBR 0000	DV.TBL 1342
DV.TRK 0001	ERR.CT 1330	EX.HCF 1018	EX.STS 1024
FETCH 103F	FMT.BG 1700	FMT.PS 1708	HOME.D 12A6
HR.INT 1038	HR.VEC 1006	ID.BLK 13A0	ID.FLD 0000
ID.FLG 13B1	ID.LBL 1380	ID.SZE 0008	INIT.B 1380
ID.BLK 1370	JADEID 133A	LD.BLT 139D	LOG.ON 12D7
LST.OT 12F4	NM.INT 1066	PH.TRK 1335	RD.SEC 1204
RST.0 1000	RST.1 1008	RST.2 1010	RST.3 1018
RST.4 1020	RST.5 1028	RST.6 1030	RST.7 1038
RTY.LS 0009	RTY.SK 0005	SEEK 1131	SELECT 10EF
SV.CTL 1333	SV.DAS 1332	SV.DRV 1331	SV.STS 1334
TICK.E 1074	TICK.R 1070	TMR.FC 0019	TMR.NC 001C
TM.ALS 1014	TM.HLD 1010	TM.MTD 1016	TM.PLD 1336
TM.SAW 1338	TM.SDD 0018	TM.STP 1012	TP.STK 1370
TRK.IB 0034	TRK.OB 001A	WD.CMD 0004	WD.DTA 0007
WD.INT 1066	WD.SEC 0006	WD.STS 0004	WD.TRK 0005
WR.SEC 122F	WR.TRK 125A	XP.DSH 0080	XP.IRR 0020
XP.MTD 0010	XP.MTX 0040	XP.STP 0008	X.CUTE 1041
\$.ADDR 10C7	\$.FORM 1098	\$.IDLE 10E4	\$.LGON 10AC
\$.LIST 10CF	\$.LSTT 10D5	\$.READ 107C	\$.WRIT 108A

F1000,13FF,0

-IDCM2.HEX

-R

NEXT PC

13AF 0000

-D1000,12FF

1000 C3 00 00 C3 80 17 41 10 06 19 10 FE C3 74 10 00A.....T..
1010 5E 01 50 00 50 00 01 00 FD E1 DB 05 D3 07 78 A9 ^.P.P.....X.
1020 D3 04 18 FE 3E D0 A9 D3 04 E3 E3 E3 E3 DB 04 A9>.....
1030 C9 00 00 00 00 00 00 00 DB 20 D1 2A 06 10 E9 FB*.....
1040 76 3A 70 13 E6 07 87 16 00 5F 21 53 10 19 5E 23 V:P.....!S..^#
1050 56 EB E9 AC 10 7C 10 8A 10 98 10 C7 10 CF 10 D5 V.....\.....
1060 10 E4 10 00 00 00 DB 04 A9 32 34 13 FD E3 ED 4524....E
1070 06 1C 10 FE 1B 7A B3 00 00 20 F5 C9 CD EF 10 CDZ.....
1080 31 11 20 03 CD 04 12 C3 3F 10 CD EF 10 CD 31 11 1.?.....1.
1090 20 03 CD 2F 12 C3 3F 10 CD EF 10 3A 73 13 DD 77 ../?...:S..W
10A0 02 CD 31 11 20 03 CD 5A 12 C3 3F 10 CD EF 10 AF ..1. ...Z...?.....
10B0 32 72 13 3C 32 73 13 CD 31 11 20 08 CD 04 12 20 2R.<ZS..1.
10C0 03 CD D7 12 C3 3F 10 3E FF 32 77 13 C3 3F 10 CD?>.2W..?..
10D0 F4 12 C3 3F 10 DB 00 E6 10 CA DE 10 3E FF 32 77 ...?.....>.2W
10E0 13 C3 3F 10 DB 00 E6 08 28 FA DB 20 C3 3F 10 DB ..?.....(.. ?..
10F0 00 E6 20 DB 40 28 05 ED 5B 16 10 CF 3A 71 13 E6 .. .@([...:Q..
1100 03 DD BE 00 C8 DD 21 42 13 11 04 00 3D FA 14 11!B.....=
1110 DD 19 18 F8 06 10 CD 18 10 3A 33 13 E6 FB D3 00:3.....
1120 E6 FC DD B6 00 D3 00 F6 04 D3 00 E6 07 32 31 1321..
1130 C9 CD 24 10 E6 A0 FA 74 11 C2 46 11 06 18 CD 18 ..\$....T..F.....
1140 10 ED 5B 10 10 CF DB 00 E6 40 3A 72 13 6F C2 52 ..[.....@:R.O.R
1150 11 1F 32 35 13 67 3A 31 13 30 02 F6 20 32 32 13 ..25.G:1.0.. 22.
1160 57 7C DD 96 01 20 21 DB 00 E6 40 CA 7B 11 DD 7E W\... !...@.[...^
1170 03 C3 EA 11 3E 80 32 77 13 A7 C9 7C A7 28 3B DD>.2W...\.(:.
1180 7E 03 E6 DF B2 C3 EA 11 F5 ED 5B 38 13 CF F1 38 ^.....18...8
1190 0A 6F 3A 31 13 F6 20 D3 00 18 0B ED 44 FA FD 11 .0:1..D...
11A0 6F 3A 31 13 D3 00 DB 08 ED 5B 12 10 CF 2D 20 F6 0:1.....[...- .
11B0 3A 32 13 D3 00 ED 5B 14 10 CF 3A 72 13 FE 01 38 :2....[...:R...8
11C0 20 3E 04 C2 C8 11 3E 02 DD A6 02 CA E1 11 3A 35 >.....>.....:5
11D0 13 FE 1A 06 D0 38 0C FE 34 06 90 38 06 06 50 188..4..8..P.
11E0 02 06 C0 3A 32 13 B0 DD 77 03 D3 00 32 33 13 3A ...:2...W...23.:
11F0 35 13 DD 77 01 3A 72 13 A9 D3 05 AF C9 CD A6 12 5..W.:R.....
1200 C0 C3 46 11 AF 32 30 13 3A 73 13 A9 D3 06 FD 21 ..F..20.:S.....!
1210 23 12 2A 2E 13 3E 88 A9 D3 04 DB 80 DB 07 A9 77 #.*..>.....W
1220 23 18 F7 E6 9D 32 77 13 C8 CD 79 12 28 E0 C9 AF #....2W...Y.(...
1230 32 30 13 3A 73 13 A9 D3 06 FD 21 4E 12 2A 2E 13 20.:S.....!N.*..
1240 3E A8 A9 D3 04 DB 80 7E A9 D3 07 23 18 F7 E6 FD >.....^...#....
1250 32 77 13 C8 CD 79 12 28 E0 C9 FD 21 66 12 3E F0 2W...Y.(...!F.>..
1260 A9 D3 04 C3 08 17 E6 E4 47 DB 00 E6 40 78 20 02G...@X .
1270 F6 01 32 77 13 22 7A 13 C9 E6 80 20 28 3A 76 13 ..2W."Z.... (:V.
1280 E6 80 20 21 DB 40 3A 30 13 3C 32 30 13 FE 05 20 .. !.@:0.<20...
1290 08 CD A6 12 20 0F CD 31 11 FE 09 28 07 ED 5B 361...([6
12A0 13 CF AF C9 3C C9 3A 31 13 D3 00 32 33 13 2E FF<.:1...23...
12B0 CD 24 10 E6 04 20 0C 2D 28 16 DB 08 ED 5B 12 10 .\$....-([...[...
12C0 CF 18 ED ED 5B 14 10 CF 79 D3 05 AF DD 77 01 C9[...Y....W..
12D0 3E 02 32 77 13 A7 C9 11 3A 13 21 80 13 06 08 1A >.2W....:!......
12E0 BE 20 0B 13 23 10 F8 3A B1 13 DD 77 02 C9 3E 00 . .#...:W..>..
12F0 DD 77 02 C9 DB 00 E6 10 CA F4 12 3A 75 13 2F 5F .W.....:U./..

D1300,13FF

1300 3A 33 13 37 CD 1A 13 00 00 16 08 CB 0B CD 1A 13 :3.7.....
1310 15 C2 0B 13 00 A7 CD 1A 13 C9 DA 22 13 CB 9F C3"
1320 27 13 CB DF C3 27 13 D3 00 06 19 10 FE C9 80 13 ^.....^.....
1330 00 00 00 00 00 00 B0 04 0A 00 4A 61 64 65 20 44JADE D
1340 44 20 00 FF 02 C4 01 FF 02 C5 02 FF 02 C6 03 FF D
1350 02 C7 04 FF 00 00 00 00 00 00 00 00 00 00 00 00
1360 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
1370 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
1380 01 00 04 11 00 10 21 00 14 ED B0 C3 8E 13 31 70!.....1P
1390 13 ED 56 0E 00 DB 00 E6 01 20 02 0E FF DD 21 52 ..V.....!R
13A0 13 3E 02 32 73 13 DB 40 21 80 13 E5 C3 04 12 00 .>.2S..@!.....
13B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
13C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
13D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
13E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
13F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
-

```
*****  
; PROGRAM ID:      FORMAT  
; *****  
; PRESENTED BY:   JADE COMPUTER PRODUCTS  
;                4901 W. ROSECRANS BLVD.  
;                HAWTHORNE, CALIFORNIA  
;                90250, U.S.A.  
; *****  
; VERSION:       CP/M 2.2  RELEASE 2A  
; *****  
; WRITTEN BY:    STAN KRUMME  
; *****  
; FORMAT IS A SYSTEM UTILITY WHICH PROVIDES A MEANS  
; TO WRITE A SINGLE OR DOUBLE DENSITY FORMAT ON ANY  
; OF DRIVES A THROUGH D. THIS UTILITY ALSO PROVIDES  
; A COPY-SYSTEM-TRACKS FEATURE. THIS IS A USEFUL  
; FUNCTION FOR FORMAT AS THE SYSTEM TRACKS CAN BE  
; WRITTEN WITH THE OPERATING SYSTEM WHEN FORMATTED.  
; FORMAT IS 8080/8085/Z80 COMPATABLE.  
; *****  
; *****  
; FORMAT INJECTION MODULES ARE COMMAND COMPATABLE WITH  
; THE FOLLOWING WESTERN DIGITAL CONTROLLER CHIPS.  
; DOUBLE D USER SWITCH 0 (U0 OR R0) MUST BE SET TO  
; INDICATE THE CONTROLLER CHIP DATA BUS POLARITY.  
; *****  
; CONTROLLER IC      USER SW0  
; -----  
; FD1791-02 (01)     CLOSED  
; FD1793-02 (01)     OPENED  
; FD1795-02         CLOSED  
; FD1797-02         OPENED  
; *****  
; *****  
; RELEASE 2A:  SINGLE AND DOUBLE SIDED DRIVES CAN BE  
; FORMATED.  INSPECTION OF TWO SIDED* SIGNAL FROM THE  
; DISK DRIVE DETERMINES NUMBER OF SIDES.  WITH DOUBLE  
; SIDED DISKETTES, BOTH SIDES FORM ONE LOGICAL DISK.  
; EACH DOUBLE DENSITY TRACK NOW CONTAINS 50 SECTORS.  
; *****
```

; DRIVER MODULE DEFINITIONS *

000A	LF	==	00AH	;ASCII LINE FEED.
000D	CR	==	00DH	;CARRIAGE RETURN.
0024	EOM	==	'\$'	;STRING TERMINATOR.
0100	TPA	==	0100H	;TRANSIENT PROGRAM.
0000	TRK.0	==	0	;TRACK 0.
0001	TRK.1	==	1	;TRACK 1.
0002	TRK.2	==	2	;TRACK 2.
0080	SEC.SZ	==	128	;128 BYTES PER SECTOR.
0001	ID.SEC	==	1	;ID SECTOR NUMBER.
0000	REBOOT	==	0	;REBOOT ADDRESS.
0001	BS.PTR	==	0001H	;WARM ADDR POINTER.
0001	NO.LOG	==	01H	;REQUEST NO LOG-ON.
00FE	FT.ERC	==	1111110B	;FORMAT ERROR MASK.
0001	FT.TSM	==	00000001B	;TWO SIDED MASK.

; INJECTION MODULE DEFINITIONS *

1700	FMT.EA	==	1700H	;FORMAT EXEC ADDRESS.
0005	WD.TRK	==	005H	;DOUBLE D TRACK PORT.
0007	WD.DTA	==	007H	;DOUBLE D DATA PORT.
0080	XP.DSH	==	80H	;DATA SYNC HOLD PORT.
0000	ZEROS	==	00000000B	;ALL ZERO BYTE.
00FF	ONES	==	11111111B	;ALL ONES BYTE.

; BDOS CALL - VECTOR NUMBERS *

0005	BDOS	==	0005H	;SYSTEM CALL ADDR.
0009	BC.PTX	==	009H	;PRINT STRING CONSOLE.
000A	BC.RCB	==	00AH	;READ CONSOLE BUFFERD.

; ASSEMBLER DIRECTIVES *

	.I8080			
	.FABS			
	.PHEX			
	.XLINK			
0100	.LOC	TPA		

```
*****  
; PROGRAM BEGINS *  
*****  
0100    C3 0146    BEGIN:  JMP      INIT          ;GO TO INITIALIZE.  
  
*****  
; ASCII IDENTIFICATION INSERT *  
*****  
0103    4A4144452043    .ASCII  'JADE COMPUTER PRODUCTS '  
011A    444F55424C45    .ASCII  'DOUBLE D - FORMAT 8" '  
012F    56455253494F    .ASCII  'VERSION 2.2 RELEASE 2A '  
  
*****  
; SET STACK POINTER AND ISSUE LOG-ON *  
*****  
0146    31 0500    INIT:   LXI      SP,SP.TOP      ;SET STACK POINTER.  
0149    11 0716    LXI      D,MSG.BG        ;LOAD MESSAGE ADDR.  
014C    CD 0297    CALL     MSG.OT         ;ISSUE MESSAGE,  
  
*****  
; LOAD BIOS VECTORS JUMP TABLE - WARM THRU FORMAT *  
*****  
014F    01 0033    LXI      B,BS.VSZ      ;SET BIOS VECTORS SIZE.  
0152    11 02A4    LXI      D,BS.WRM      ;SET FORMAT TABLE.  
0155    2A 0001    LHLD    BS.PTR        ;WARM VECTOR POINTER.  
0158    CD 02D7    CALL     B.MOVE        ;BLOCK MOVE VECTORS.  
  
*****  
; SELECT DRIVE TO FORMAT ON *  
*****  
015B    21 0781    LXI      H,MSG.FD      ;FORMAT ON DRIVE MSG.  
015E    CD 0485    CALL     SEL.DR        ;CALL SELECT DRIVE.  
0161    32 04DF    STA     FD.NBR        ;FORMAT DRIVE NMBR.
```

```
*****
```



```
*****  
; DISPLAY FUNCTIONS LIST *  
*****
```

```
0164 11 07BB LIST: LXI D,MSG.FL ;FUNCTIONS MSG ADDR.  
0167 CD 0297 CALL MSG.OT ;ISSUE THIS MESSAGE.
```

```
*****  
; INQUIRE SELECTION *  
*****
```

```
016A 11 08F4 SELECT: LXI D,MSG.SF ;SELECT FUNCTION MSG.  
016D CD 0297 CALL MSG.OT ;ISSUE THIS MESSAGE.  
0170 CD 029C CALL CNS.IN ;GET CONSOLE CHARACTER.  
0173 3A 0501 LDA RC.NBR ;LOAD BUFFER SIZE.  
0176 FE01 CPI 1 ;CHECK FOR 1 CHARACTER.  
0178 C2 016A JNZ SELECT ;OTHER THAN 1 TOO BAD.
```

```
*****  
; SELECT FUNCTION DRIVER *  
*****
```

```
017B 3A 0502 LDA RC.TXT+0 ;LOAD CONSOLE CHAR.  
017E FE31 CPI "1"  
0180 CA 01AA JZ FUN.1 ;FMT DOUBLE DENSITY.  
0183 FE32 CPI "2"  
0185 CA 0208 JZ FUN.2 ;FMT SINGLE DENSITY.  
0188 FE33 CPI "3"  
018A CA 01D2 JZ FUN.3 ;FMT 3740.  
018D FE34 CPI "4"  
018F CA 01EB JZ FUN.4 ;READ SYSTEM TRACKS.  
0192 FE35 CPI "5"  
0194 CA 025D JZ FUN.5 ;WRITE SYSTEM TRACKS.  
0197 FE2A CPI "*" ;  
0199 CA 023B JZ FMT.ST ;FORMAT SYSTEM TRACKS.  
019C FE27 CPI "~"  
019E CA 04CC JZ RST.7 ;DDT TRAP.
```

```
*****  
; MUST BE A BAD CHOICE *  
*****
```

```
01A1 11 07A2 LXI D,MSG.SE ;SELECT ERROR MESSAGE.  
01A4 CD 0297 CALL MSG.OT ;ISSUE MESSAGE.  
01A7 C3 0164 JMP LIST ;DIPLAY LIST AGAIN.
```

```
*****
```

```

;*****
; FUNCTION 1 - FORMAT IN DOUBLE DENSITY *
;*****

```

```

01AA      3A 0647      FUN.1:  LDA      DD.FLG      ;LOAD DDENS FLAGS.
01AD      32 04DC      STA      F.FLAG      ;STORE FORMAT FLAGS.
01B0      3E00         MVI      A,TRK.0      ;TRACK 0.
01B2      32 04D9      STA      TRK.NO      ;SET TRACK NUMBER.
01B5      CD 031B      CALL     FMT.SD       ;FORMAT TRACK SDENS.
01B8      C2 028E      JNZ      TRK.ER      ;JUMP ERROR DETECTED.
01BB      3E01         MVI      A,TRK.1      ;TRACK 1 VALUE.
01BD      32 04D9      STA      TRK.NO      ;SET TRACK NUMBER.
01C0      CD 0321      ;.REPT: CALL     FMT.DD       ;FORMAT TRACK DDENS.
01C3      C2 028E      JNZ      TRK.ER      ;JUMP ERROR DETECTED.
01C6      CD 0279      CALL     TRK.NX       ;SET FOR NEXT TRACK.
01C9      CA 01C0      JZ       ;.REPT       ;FORMAT NEXT TRACK.
01CC      CD 02F2      ;.ID:  CALL     WDD.ID       ;WRITE DDENS ID SECTOR.
01CF      C3 016A      JMP      SELECT      ;SELECT NEW FUNCTION.

```

```

;*****
; FUNCTION 3 - FORMAT STANDARD 3740 - SINGLE SIDED *
;*****

```

```

01D2      3E00         FUN.3:  MVI      A,0      ;3740 SDENS FLAGS.
01D4      32 04DC      STA      F.FLAG      ;STORE FORMAT FLAGS.
01D7      3E00         MVI      A,TRK.0      ;TRACK 0.
01D9      32 04D9      STA      TRK.NO      ;SET TRACK NUMBER.
01DC      CD 031B      ;.REPT: CALL     FMT.SD       ;FORMAT TRACK SDENS.
01DF      C2 028E      JNZ      TRK.ER      ;JUMP ERROR DETECTED.
01E2      CD 0279      CALL     TRK.NX       ;SET FOR NEXT TRACK.
01E5      CA 01DC      JZ       ;.REPT       ;FORMAT NEXT TRACK.
01E8      C3 016A      JMP      SELECT      ;SELECT NEW FUNCTION.

```

```

;*****
; FUNCTION 4 - READ SYSTEM TRACKS *
;*****

```

```

01EB      21 090E      FUN.4:  LXI      H,MSG.RS  ;READ DRIVE MSG.
01EE      CD 0485      CALL     SEL.DR       ;SELECT READ SYS DRV.
01F1      3E52         MVI      A,'R'        ;READ TRANSFER CODE.
01F3      32 04D4      STA      TF.DIR      ;SET TRANSFER DIRC.
01F6      32 04DD      STA      SYS.RF       ;SET SYSTEM READ FLAG.
01F9      CD 0355      CALL     TRNSFR       ;READ SYSTEM TRACKS.
01FC      3A 04DF      LDA      FD.NBR      ;GET FORMAT DRV NMBR.
01FF      4F           MOV      C,A          ;PUT INTO C REG.
0200      1E01         MVI      E,NO.LOG     ;INSURE NO LOGON.
0202      CD 02BC      CALL     BS.DSK       ;BIOS SELECT DISK.
0205      C3 016A      JMP      SELECT      ;RESELECT FUNCTION.

```

```

;*****

```

```

;*****
; FUNCTION 2 - FORMAT IN SINGLE DENSITY *
;*****

```

```

0208 3A 0547 FUN.2: LDA SD.FLG ;LOAD SDENS FLAGS.
020B 32 04DC STA F.FLAG ;STORE FORMAT FLAGS.
020E 3E00 MVI A,TRK.0 ;TRACK 0.
0210 32 04D9 STA TRK.NO ;SET TRACK NUMBER.
0213 CD 031B CALL FMT.SD ;FORMAT TRACK SDENS.
0216 C2 028E JNZ TRK.ER ;JUMP ERROR DETECTED.
0219 3E01 MVI A,TRK.1 ;TRACK 1
021B 32 04D9 STA TRK.NO ;SET TRACK NUMBER.
021E CD 0321 CALL FMT.DD ;FORMAT TRACK DDENS.
0221 C2 028E JNZ TRK.ER ;JUMP ERROR DETECTED.
0224 3E02 MVI A,TRK.2 ;TRACK 2.
0226 32 04D9 STA TRK.NO ;SET TRACK NUMBER.
0229 CD 031B ..REPT: CALL FMT.SD ;FORMAT TRACK SDENS.
022C C2 028E JNZ TRK.ER ;JUMP ERROR DETECTED.
022F CD 0279 CALL TRK.NX ;SET FOR NEXT TRACK.
0232 CA 0229 JZ ..REPT ;FORMAT NEXT TRACK.
0235 CD 02E2 ..ID: CALL WSD.ID ;WRITE SDENS ID SECTOR.
0238 C3 016A JMP SELECT ;SELECT NEW FUNCTION.

```

```

;*****
; NON DOCUMENTED FUNCTION - FORMAT JADE SYSTEM TRACKS *
;*****
; USED FOR SPECIAL PURPOSE - NOT NEEDED BY END USER *
;*****

```

```

023B 3A 0547 FMT.ST: LDA SD.FLG ;LOAD SDENS FLAGS.
023E 32 04DC STA F.FLAG ;STORE FORMAT FLAGS.
0241 3E00 MVI A,TRK.0 ;TRACK 0.
0243 32 04D9 STA TRK.NO ;SET TRACK NUMBER.
0246 CD 031B CALL FMT.SD ;FORMAT TRACK SDENS.
0249 C2 028E JNZ TRK.ER ;JUMP ERROR DETECTED.
024C 3E01 MVI A,TRK.1 ;TRACK 1
024E 32 04D9 STA TRK.NO ;SET TRACK NUMBER.
0251 CD 0321 CALL FMT.DD ;FORMAT TRACK DDENS.
0254 C2 028E JNZ TRK.ER ;JUMP ERROR DETECTED.
0257 CD 02E2 CALL WSD.ID ;WRITE SDENS ID SECTOR.
025A C3 016A JMP SELECT ;SELECT NEW FUNCTION.

```

```

;*****

```

```

;*****
; FUNCTION 5 - WRITE SYSTEM TRACKS *
;*****

```

```

025D 3A 04DD FUN.5: LDA SYS.RF ;LOAD SYSTEM READ FLAG.
0260 FE52 CPI 'R' ;TEST IF READ CODE.
0262 C2 0270 JNZ ..NSYS ;JUMP IF NO SYSTEM.
0265 3E57 MVI A,'W' ;WRITE TRANSFER CODE.
0267 32 04D4 STA TF.DIR ;SET TRANSFER DIRC.
026A CD 0355 CALL TRNSFR ;WRITE SYSTEM TRACKS.
026D C3 016A JMP SELECT ;WRITE ANOTHER DISK.
0270 11 095E ..NSYS: LXI D,MSG.NR ;NO SYSTEM LOADED MSG.
0273 CD 0297 CALL MSG.OT ;ISSUE THIS MESSAGE.
0276 C3 016A JMP SELECT ;SELECT NEW FUNCTION.

```

```

;*****
; NEXT TRACK SELECT ROUTINE *
;*****

```

```

0279 3A 04DA TRK.NX: LDA TRK.MX ;LOAD MAX TRACK NMBR.
027C 47 MOV B,A ;SAVE IN REG B.
027D 3A 04D9 LDA TRK.NO ;GET THIS TRACK NO.
0280 B8 CMP B ;CHECK FOR LAST TRACK.
0281 CA 028A JZ ..DONE ;JUMP IF LAST TRACK.
0284 3C INR A ;GET NEXT TRACK.
0285 32 04D9 STA TRK.NO ;STORE NEXT TRACK.
0288 AF XRA A ;SET ZERO FLAG.
0289 C9 RET ;RETURN TO CALLER.
028A 3EFF ..DONE: MVI A,ONES ;SET ALL ONES.
028C A7 ANA A ;SET FLAG NOT ZERO.
028D C9 RET ;LAST TRACK EXIT.

```

```

;*****
; FORMAT TRACK ERROR *
;*****

```

```

028E 11 0945 TRK.ER: LXI D,MSG.FE ;FORMAT ERROR MSG ADDR.
0291 CD 0297 CALL MSG.OT ;DISPLAY MESSAGE.
0294 C3 016A JMP SELECT ;SELECT NEW FUNCTION.

```

```

;*****

```

```

;*****
; MESSAGE DISPLAY ROUTINE
;*****

```

```

0297 0E09 MSG.OT: MVI C,BC.PTX ;PRINT TEXT VECTOR.
0299 C3 0005 JMP BDOS ;CONTINUE IN BDOS.

```

```

;*****
; CONSOLE INPUT ROUTINE
;*****

```

```

029C 11 0500 CNS.IN: LXI D,RC.BUF ;KEYBOARD BUFFER ADDR.
029F 0E0A MVI C,BC.RCB ;BDOS CONSOLE BUF READ.
02A1 C3 0005 JMP BDOS ;CONTINUE IN BDOS.

```

```

;*****
; BIOS VECTOR DEFINITIONS
;*****

```

```

02A4 C3 0000 BS.WRM: JMP 0 ;RELOAD CCP/BDOS.
02A7 C3 0000 JMP 0 ;GET CONSOLE STATUS.
02AA C3 0000 JMP 0 ;CONSOLE CHAR INPUT.
02AD C3 0000 JMP 0 ;CONSOLE CHAR OUTPUT.
02B0 C3 0000 JMP 0 ;PRINTER OUTPUT.
02B3 C3 0000 JMP 0 ;PUNCH CHARACTER OUT.
02B6 C3 0000 JMP 0 ;READER INPUT.
02B9 C3 0000 JMP 0 ;HOME SELECTED DRIVE.
02BC C3 0000 BS.DSK: JMP 0 ;SELECT DISK DRIVE.
02BF C3 0000 BS.TRK: JMP 0 ;SET TRACK NUMBER.
02C2 C3 0000 BS.SEC: JMP 0 ;SET SECTOR NUMBER.
02C5 C3 0000 BS.DMA: JMP 0 ;SET TRANSFER ADDR.
02C8 C3 0000 BS.RDS: JMP 0 ;READ DISK SECTOR.
02CB C3 0000 BS.WRS: JMP 0 ;WRITE DISK SECTOR.
02CE C3 0000 JMP 0 ;LIST DEV STATUS.
02D1 C3 0000 JMP 0 ;SECTOR TRANSLATE.
02D4 C3 0000 BS.FMT: JMP 0 ;FORMAT DISK TRACK.
0033 BS.VSZ == .-BS.WRM ;CALCULATE SIZE.

```

```

;*****
; BLOCK MOVE SUBROUTINE
;*****

```

```

02D7 7E B.MOVE: MOV A,M ;GET BYTE
02D8 23 INX H ;INC SOURCE.
02D9 12 STAX D ;STORE BYTE.
02DA 13 INX D ;INC DESTINATION.
02DB 0B DCX B ;ONE LESS TO DO.
02DC 78 MOV A,B ;GET B REG.
02DD B1 ORA C ;OR IN C REG.
02DE C2 02D7 JNZ B.MOVE ;REPEAT FOR LENGTH.
02E1 C9 RET ;RETURN CALLER.

```

```

;*****

```

; WRITE ID SECTOR *

***** (SET TRANSFER ADDRESS) *****

02E2	01 0516	WSD.ID:	LXI	B,IDS.SS	;ID SECTOR ADDRESS.
02E5	3A 04D8		LDA	TS.FLG	;GET TWO SIDES FLG.
02E8	A7		ANA	A	;TEST.
02E9	CA 02FF		JZ	WRT.ID	;JUMP ONE SIDED.
02EC	01 0596		LXI	B,IDS.DS	;DOUBLE SIDED.
02EF	C3 02FF		JMP	WRT.ID	;JUMP TWO SIDED.
02F2	01 0616	WDD.ID:	LXI	B,IDS.SD	;ID SECTOR ADDRESS.
02F5	3A 04D8		LDA	TS.FLG	;GET TWO SIDES FLG.
02F8	A7		ANA	A	;TEST.
02F9	CA 02FF		JZ	WRT.ID	;JUMP ONE SIDED.
02FC	01 0696		LXI	B,IDS.DD	;DOUBLE SIDED.
02FF	CD 02C5	WRT.ID:	CALL	BS.DMA	;BIOS TRANSFER ADDR.

***** (SET TRACK AND SECTOR NUMBERS) *****

0302	0E00		MVI	C,TRK.0	;TRACK 0 SET.
0304	CD 02BF		CALL	BS.TRK	;BIOS SET TRACK.
0307	0E01		MVI	C,ID.SEC	;ID SECTOR VALUE.
0309	CD 02C2		CALL	BS.SEC	;BIOS SET SECTOR.

***** (PERFORM WRITE SECTOR) *****

030C	CD 02CB		CALL	BS.WRS	;BIOS WRITE SECTOR.
030F	B7		ORA	A	;SET CONDITION CODES.
0310	C8		RZ		;RETURN USER GOOD.
0311	11 092B		LXI	D,MSG.NC	;TRANSFER INCOMPLETE.
0314	CD 0297		CALL	MSG.OT	;ISSUE MESSAGE.
0317	3EFF		MVI	A,ONES	;SET ACUMULATOR.
0319	A7		ANA	A	;SET FLAGS NOT ZERO.
031A	C9		RET		;ERROR RETURN.

```

;*****
; FORMAT TRACK DRIVER *
;*****

```

```

;***** ( DENSITY ENTRIES )*****

```

```

031B 01 0A00 FMT.SD: LXI B,FT3740 ;LOAD INJECTION ADDR.
031E C3 0324 JMP ST.DMA ;GO SET DMA ADDR.
0321 01 0B00 FMT.DD: LXI B,FTJ50D ;LOAD INJECTION ADDR.

```

```

;***** ( SET INJECTION MODULE ADDRESS )*****

```

```

0324 CD 02C5 ST.DMA: CALL BS.DMA ;SET TRANSFER ADDRESS.

```

```

;***** ( SET TRACK NUMBER AND DCM FLAGS )*****

```

```

0327 3A 04D9 LDA TRK.NO ;LOAD TRACK NMBR.
032A 4F MOV C,A ;PUT INTO C REGISTER.
032B CD 02BF CALL BS.TRK ;SET TRACK NMBR.

032E 3A 04DC LDA F.FLAG ;LOAD DCM FLAG.
0331 4F MOV C,A ;DCM FLAGS.
0332 CD 02C2 CALL BS.SEC ;SET DCM FLAGS.

```

```

;***** ( PERFORM FORMAT TRACK )*****

```

```

0335 CD 02D4 CALL BS.FMT ;BIOS WRITE TRACK.
0338 32 04D7 STA FT.STS ;FORMAT STATUS.
033B E6FE ANI FT.ERC ;TEST FOR ERRORS.
033D C0 RNZ ;ERROR EXIT.

```

```

;***** ( SET CONTROLS FOR SIDE/SIDES )*****

```

```

033E 3A 04D7 LDA FT.STS ;GET STATUS.
0341 E601 ANI FT.TSM ;TEST TWO SIDES FLAG.
0343 32 04D8 STA TS.FLG ;STORE FLAG.
0346 C2 034E JNZ ..TWSID ;TWO SIDES IS A 1.
0349 3E4C MVI A,77-1 ;SINGLE SIDED MAX.
034B C3 0350 JMP ..EXIT ;EXIT.
034E 3E99 ..TWSID: MVI A,2*77-1 ;DOUBLE SIDED MAX.
0350 32 04DA ..EXIT: STA TRK.MX ;SET MAX TRACK.
0353 AF XRA A ;SET ZERO FLAG.
0354 C9 RET ;RETURN TO CALLER.

```

```

;*****

```

```

;*****
; SYSTEM TRACKS TRANSFER FUNCTION - ENTRY POINT *
;*****

;***** ( TRANSFER INITIALIZE )*****

0355 21 03A9 TRNSFR: LXI H,ST.LST ;ADDR OF TRANSFER LIST.
0358 22 04D2 SHLD TF.PTR ;SET TRANSFER POINTER.

;***** ( SET BIOS TRACK NUMBER )*****

035B CD 0391 ..REPT: CALL ..PLST ;POP BYTE FROM LIST.
035E FEFF CPI EOL ;TEST FOR END OF LIST.
0360 C8 RZ ;EXIT TRANSFER.
0361 CD 02BF CALL BS.TRK ;BIOS SET TRACK.

;***** ( SET BIOS SECTOR NUMBER )*****

0364 CD 0391 CALL ..PLST ;POP BYTE FROM LIST.
0367 CD 02C2 CALL BS.SEC ;BIOS SET SECTOR.

;***** ( SET BIOS TRANSFER ADDRESS )*****

036A CD 0391 CALL ..PLST ;POP BYTE FROM LIST.
036D CD 039B CALL ..ADDR ;CALCULATE ADDRESS.
0370 CD 02C5 CALL BS.DMA ;BIOS TRANSFER ADDR.

;***** ( SECTOR TRANSFER OPERATION )*****

0373 21 0386 LXI H,..RETN ;LOAD RETURN ADDRESS.
0376 E5 PUSH H ;PUSH ONTO STACK.
0377 3A 04D4 LDA TF.DIR ;LOAD TRNSFR DIRECTION.
037A FE57 CPI 'W' ;SEE IF WRITE FUNCTION.
037C CA 02CB JZ BS.WRS ;BIOS WRITE SECTOR.
037F FE52 CPI 'R' ;SEE IF READ FUNCTION.
0381 CA 02C8 JZ BS.RDS ;BIOS READ SECTOR.
0384 3EFF MVI A,ONES ;ERROR CODE NOT R/W.
0386 B7 ..RETN: ORA A ;SET CONDITION CODES.
0387 CA 035B JZ ..REPT ;DO SOME MORE.

;***** ( ENCOUNTERED DIFFICULTY )*****

038A 11 092B LXI D,MSG.NC ;MESSAGE ADDRESS.
038D CD 0297 CALL MSG.OT ;SEND MESSAGE.
0390 C9 RET ;GO HOME.

;*****

```


FORMAT - JADE DOUBLE D

SYSTEM TRACKS TRANSFER SUBROUTINE

```

;*****
; POP LIST SUBROUTINE
;*****

```

```

0391 2A 04D2  ..PLST: LHL  TF.PTR  ;LOAD LIST POINTER.
0394 4E      MOV   C,M      ;GET ITEM NUMBER.
0395 23      INX   H        ;INCREMENT POINTER.
0396 22 04D2  SHLD  TF.PTR  ;STORE LIST POINTER.
0399 79      MOV   A,C      ;MOVE C TO ACUM.
039A C9      RET                ;RETURN TO CALLER.

```

```

;*****
; GET MEMORY ADDRESS SUBROUTINE
;*****

```

```

039B A7      ..ADDR: ANA   A        ;CLEAR CARRY BIT.
039C 1F      RAR                ;DIVIDE BY 2.
039D 47      MOV   B,A      ;HI ORDER TO B REG.
039E 3E00   MVI   A,0      ;CLEAR ACUMULATOR.
03A0 1F      RAR                ;CARRY BIT TO MSB.
03A1 4F      MOV   C,A      ;LO ORDER TO C REG.
03A2 2A 04D0  LHL  TF.INX  ;LOAD TRANSFER INDEX.
03A5 09      DAD   B        ;ADD IN OFFSET.
03A6 44      MOV   B,H      ;HALF MOV BC,HL.
03A7 4D      MOV   C,L      ;THE OTHER (HA!)
03A8 C9      RET                ;RETURN TO CALLER.

```

```

;*****

```

```
*****  
; THE FOLLOWING IS A LIST OF SYSTEM TRACK SECTORS *  
; USED BY THE TRNSFR SUBROUTINE. THERE ARE THREE *  
; ENTRIES PER SECTOR. 1ST IS TRACK NUMBER. 2ND IS *  
; SECTOR NUMBER. 3RD IS MEMORY LOAD OFFSET. *  
*****  
; SECTORS 2 THRU 26 ARE TRANSFERED ON TRACK 0. SECTOR *  
; 1 IS NOT TRANSFERED, THIS IS THE IDENTITY SECTOR. *  
; TRACK 0 SECTOR ARE LOCATED IN SEQUENTIAL ORDER, SO *  
; THIS LIST IS STAGGERED. SECTORS 1 THRU 48 ARE *  
; TRANSFERED ON TRACK 1. *  
*****
```

```
0000 TK0 == 0 ;DEFINE TRACK 0.  
0001 TK1 == 1 ;DEFINE TRACK 1.  
00FF EOL == OFFH ;DEFINE END OF LIST.
```

```
*****
```

```
03A9 000404000808 ST.LST: .BYTE TK0,04,04,TK0,08,08,TK0,12,12,TK0,16,16  
03B5 001414001818 .BYTE TK0,20,20,TK0,24,24,TK0,02,02,TK0,06,06  
03C1 000A0A000E0E .BYTE TK0,10,10,TK0,14,14,TK0,18,18,TK0,22,22  
03CD 001A1A000505 .BYTE TK0,26,26,TK0,05,05,TK0,09,09,TK0,13,13  
03D9 001111001515 .BYTE TK0,17,17,TK0,21,21,TK0,25,25,TK0,03,03  
03E5 000707000B0B .BYTE TK0,07,07,TK0,11,11,TK0,15,15,TK0,19,19  
03F1 001717 .BYTE TK0,23,23  
  
03F4 01011B01021C .BYTE TK1,01,27,TK1,02,28,TK1,03,29,TK1,04,30  
0400 01051F010620 .BYTE TK1,05,31,TK1,06,32,TK1,07,33,TK1,08,34  
040C 010923010A24 .BYTE TK1,09,35,TK1,10,36,TK1,11,37,TK1,12,38  
0418 010D27010E28 .BYTE TK1,13,39,TK1,14,40,TK1,15,41,TK1,16,42  
0424 01112B01122C .BYTE TK1,17,43,TK1,18,44,TK1,19,45,TK1,20,46  
0430 01152F011630 .BYTE TK1,21,47,TK1,22,48,TK1,23,49,TK1,24,50  
043C 011933011A34 .BYTE TK1,25,51,TK1,26,52,TK1,27,53,TK1,28,54  
0448 011D37011E38 .BYTE TK1,29,55,TK1,30,56,TK1,31,57,TK1,32,58  
0454 01213B01223C .BYTE TK1,33,59,TK1,34,60,TK1,35,61,TK1,36,62  
0460 01253F012640 .BYTE TK1,37,63,TK1,38,64,TK1,39,65,TK1,40,66  
046C 012943012A44 .BYTE TK1,41,67,TK1,42,68,TK1,43,69,TK1,44,70  
0478 012D47012E48 .BYTE TK1,45,71,TK1,46,72,TK1,47,73,TK1,48,74  
  
0484 FF .BYTE EOL ;END OF LIST.
```

```
*****
```

```

;*****
; SELECT DRIVE THRU BIOS
;*****

```

```

;***** ( DISPLAY MESSAGE AND WAIT FOR RESPONSE )*****

```

```

0485 22 04D5 SEL.DR: SHLD MSG.SV ;SAVE MESSAGE ADDRESS.
0488 2A 04D5 ..REPT: LHLD MSG.SV ;LOAD MESSAGE ADDRESS.
048B EB XCHG ;PUT ADDRESS IN DE.
048C CD 0297 CALL MSG.OT ;ISSUE MESSAGE.
048F CD 029C CALL CNS.IN ;CONSOLE INPUT.
0492 3A 0501 LDA RC.NBR ;LOAD NMBR OF CHARS.
0495 FE01 CPI 1 ;SEE IF ONE CHARACTER.
0497 C2 0488 JNZ ..REPT ;IF NOT 1 CHAR TOO BAD.

```

```

;***** ( SEE IF DRIVE LETTER GOOD )*****

```

```

049A 3A 0502 LDA RC.TXT ;LOAD LETTER.
049D D641 SUI 'A' ;GET NUMBER.
049F DA 04B1 JC ..ILLG ;ILLEGAL, REPEAT.
04A2 FE04 CPI 04H ;DRIVE A THRU D?
04A4 DA 04BA JC ..NMBR ;GOOD NUMBER.
04A7 D620 SUI 'A'-'A' ;OFFSET LOWER CASE.
04A9 DA 04B1 JC ..ILLG ;ILLEGAL, REPEAT.
04AC FE04 CPI 04H ;LOWER A THRU D?
04AE DA 04BA JC ..NMBR ;LEGAL DRIVE.

```

```

;***** ( EXIT TO RESELECT FUNCTION )*****

```

```

04B1 11 07A2 ..ILLG: LXI D,MSG.SE ;'SELECT ERROR'
04B4 CD 0297 CALL MSG.OT ;ISSUE MESSAGE.
04B7 C3 0488 JMP ..REPT ;REPEAT SELECTION.

```

```

;***** ( VALID DRIVE NUMBER )*****

```

```

04BA 32 04DE ..NMBR: STA SV.NBR ;SAVE DRIVE NUMBER.
04BD 4F MOV C,A ;DRIVE NMBR TO C.
04BE 1E01 MVI E,NO.LOG ;LOG ON VECTOR.
04C0 CD 02BC CALL BS.DSK ;BIOS SELECT DISK.
04C3 7C MOV A,H ;CHECK RETURN ADDR.
04C4 B5 ORA L ;SET FLAGS Z/NZ.
04C5 CA 04B1 JZ ..ILLG ;BIOS SAID NOGO IF 0.
04C8 3A 04DE LDA SV.NBR ;GET NUMBER.
04CB C9 RET ;RETURN CALLER.

```

```

;*****

```

```
*****  
; SOFTWARE TRAPS - DDT EXIT *  
*****  
04CC    FF          RST.7:  RST      7          ;EXIT FORMAT.  
04CD    C3 016A    JMP      SELECT      ;RETURN FOR SELECTION.  
  
*****  
; WORKING VARIABLES *  
*****  
04D0    0F80      TF.INX:  .WORD   0F80H    ;TRANSFER INDEX.  
04D2    0000      TF.PTR:  .WORD   0          ;LIST ADDRESS POINTER.  
04D4    00        TF.DIR:  .BYTE   0          ;TRANSFER DIRECTION.  
04D5    0000      MSG.SV:  .WORD   0          ;MESSAGE SAVE ADDRESS.  
04D7    00        FT.STS:  .BYTE   0          ;FORMAT STATUS SAVE.  
04D8    00        TS.FLG:  .BYTE   0          ;TWO SIDED DRIVE FLAG.  
04D9    00        TRK.NG:  .BYTE   0          ;TRACK NUMBER HOLD.  
04DA    00        TRK.MX:  .BYTE   0          ;LAST TRACK LIMIT.  
04DB    00        SEC.NG:  .BYTE   0          ;SECTOR NUMBER HOLD.  
04DC    00        F.FLAG:  .BYTE   0          ;FORMAT FLAG (DCM).  
04DD    00        SYS.RF:  .BYTE   0          ;SYSTEM TRACK READ FLAG.  
04DE    00        SV.NBR:  .BYTE   0          ;SEL.DV TEMP STORAGE.  
04DF    00        FD.NBR:  .BYTE   0          ;FORMAT DRIVE NUMBER.  
  
04E0          STACK:  .BLKW   16          ;PROGRAM STACK AREA.  
0500          SP.TOP  ==      .          ;TOP OF STACK.  
  
*****  
; CONSOLE INPUT BUFFER AREA *  
*****  
0014          CB.SIZ  ==      20          ;CONSOLE BUFFER SIZE SET.  
  
0500    14        RC.BUF:  .BYTE   CB.SIZ  ;DECLARE BUFFER SIZE.  
0501    00        RC.NBR:  .BYTE   0          ;INPUT STRING SIZE.  
0502          RC.TXT:  .BLKB   CB.SIZ  ;RESERVE CONSOLE BUFFER AREA.  
  
*****
```

; JADE SINGLE DENSITY - IDENTITY SECTORS *

0516 4A6164652044 IDS.SS: .ASCII "JADE DD S SIDED S DENSITY FORMAT "

0536 .LOC IDS.SS+20H ;LOCATE CP/M 2.2 DPB.
0536 001A .WORD 26 ;SECTORS PER TRACK.
0538 03 .BYTE 3 ;BLOCK SHIFT FACTOR.
0539 07 .BYTE 7 ;BLOCK MASK.
053A 00 .BYTE 0 ;EXM.
053B 00F2 .WORD 26*75/8-1 ;DISK SIZE - 1.
053D 003F .WORD 63 ;DIRECTORY MAXIMUM.
053F C0 .BYTE 11000000B ;ALLOC 0.
0540 00 .BYTE 0 ;ALLOC 1.
0541 0010 .WORD 16 ;CHECK SIZE.
0543 0002 .WORD 2 ;TRACK OFFSET.

0546 .LOC IDS.SS+30H ;LOCATE DCM BLOCK.
0546 00 .BYTE 0 ;NOT USED.
0547 02 SD.FLG: .BYTE 00000010B ;DISKETTE FLAGS.

0596 .LOC IDS.SS+SEC.SZ ;EXTEND FULL SECTOR.

0596 4A6164652044 IDS.DS: .ASCII "JADE DD D SIDED S DENSITY FORMAT "

05B6 .LOC IDS.DS+20H ;LOCATE CP/M 2.2 DPB.
05B6 001A .WORD 26 ;SECTORS PER TRACK.
05B8 04 .BYTE 4 ;BLOCK SHIFT FACTOR.
05B9 0F .BYTE 15 ;BLOCK MASK.
05BA 01 .BYTE 1 ;EXM.
05BB 00F6 .WORD 26*152/16-1 ;DISK SIZE - 1.
05BD 003F .WORD 63 ;DIRECTORY MAXIMUM.
05BF 80 .BYTE 10000000B ;ALLOC 0.
05C0 00 .BYTE 0 ;ALLOC 1.
05C1 0010 .WORD 16 ;CHECK SIZE.
05C3 0002 .WORD 2 ;TRACK OFFSET.

05C6 .LOC IDS.DS+30H ;LOCATE DCM BLOCK.
05C6 00 .BYTE 0 ;NOT USED.
05C7 0A .BYTE 00001010B ;DISKETTE FLAGS.

0616 .LOC IDS.DS+SEC.SZ ;EXTEND FULL SECTOR.

; JADE DOUBLE DENSITY - IDENTITY SECTORS *

0616 4A6164652044 IDS.SD: .ASCII "JADE DD S SIDED D DENSITY FORMAT "

0636 .LOC IDS.SD+20H ;LOCATE CP/M 2.2 DPB.
0636 0032 .WORD 50 ;SECTORS PER TRACK.
0638 04 .BYTE 4 ;BLOCK SHIFT FACTOR.
0639 0F .BYTE 00001111B ;BLOCK MASK.
063A 01 .BYTE 1 ;EXM.
063B 00E9 .WORD 50*75/16-1 ;DISK SIZE - 1.
063D 003F .WORD 63 ;DIRECTORY MAXIMUM.
063F 80 .BYTE 10000000B ;ALLOC 0.
0640 00 .BYTE 0 ;ALLOC 1.
0641 0010 .WORD 16 ;CHECK SIZE.
0643 0002 .WORD 2 ;TRACK OFFSET.

0646 .LOC IDS.SD+30H ;LOCATE DCM BLOCK.
0646 00 .BYTE 0 ;NOT USED.
0647 06 DD.FLG: .BYTE 00000110B ;DISKETTE FLAGS.

0696 .LOC IDS.SD+SEC.SZ ;EXTEND TO FULL SIZE

0696 4A6164652044 IDS.DD: .ASCII "JADE DD D SIDED D DENSITY FORMAT "

06B6 .LOC IDS.DD+20H ;LOCATE CP/M 2.2 DPB.
06B6 0032 .WORD 50 ;SECTORS PER TRACK.
06B8 05 .BYTE 5 ;BLOCK SHIFT FACTOR.
06B9 1F .BYTE 31 ;BLOCK MASK.
06BA 03 .BYTE 3 ;EXM.
06BB 00EC .WORD 50*152/32-1 ;DISK SIZE - 1.
06BD 007F .WORD 127 ;DIRECTORY MAXIMUM.
06BF 80 .BYTE 10000000B ;ALLOC 0.
06C0 00 .BYTE 0 ;ALLOC 1.
06C1 0020 .WORD 32 ;CHECK SIZE.
06C3 0002 .WORD 2 ;TRACK OFFSET.

06C6 .LOC IDS.DD+30H ;LOCATE DCM BLOCK.
06C6 00 .BYTE 0 ;NOT USED.
06C7 0E .BYTE 00001110B ;DISKETTE FLAGS.

0716 .LOC IDS.DD+SEC.SZ ;EXTEND TO FULL SIZE

```
0716          ;*****  
MSG.BG:          ;CONSOLE SIGN ON          *  
          ;*****  
  
0716  ODOA      .ASCII [CR][LF]  
0718  ODOA2D2D2D2D .ASCII [CR][LF]'-----'  
073A  ODOA464F524D .ASCII [CR][LF]'FORMAT UTILITY 2 - JADE DOUBLE D'  
075C  ODOA2D2D2D2D .ASCII [CR][LF]'-----'  
077E  ODOA24     .ASCII [CR][LF][EOM]  
  
          ;*****  
0781  MSG.FD:    ;FORMAT ON DRIVE SELECT    *  
          ;*****  
  
0781  ODOA53656C65 .ASCII [CR][LF]'SELECT DRIVE TO BE FORMATTED: '[EOM]  
  
          ;*****  
07A2  MSG.SE:    ;SELECTION ERROR                *  
          ;*****  
  
07A2  ODOA4E4F5420 .ASCII [CR][LF]'NOT A VALID SELECTION '[EOM]  
  
          ;*****  
07BB  MSG.FL:    ;DISPLAY FUNCTION SELECTIONS        *  
          ;*****  
  
07BB  ODOA0DOA     .ASCII [CR][LF][CR][LF]  
07BF  ODOA2D2D2D2D .ASCII [CR][LF]'-----'  
07E1  ODOA20202020 .ASCII [CR][LF]'          FUNCTIONS LIST          '  
0803  ODOA2D2D2D2D .ASCII [CR][LF]'-----'  
0825  ODOA        .ASCII [CR][LF]  
0827  ODOA20312E20 .ASCII [CR][LF]' 1. FORMAT  DOUBLE DENSITY 8"  '  
0847  ODOA20322E20 .ASCII [CR][LF]' 2. FORMAT  SINGLE DENSITY 8"  '  
0867  ODOA20332E20 .ASCII [CR][LF]' 3. FORMAT  STANDARD 3740 8"  '  
0887  ODOA20342E20 .ASCII [CR][LF]' 4. READ    SYSTEM TRACKS IMAGE  '  
08AA  ODOA20352E20 .ASCII [CR][LF]' 5. WRITE   SYSTEM TRACKS IMAGE  '  
08CD  ODOA        .ASCII [CR][LF]  
08CF  ODOA2D2D2D2D .ASCII [CR][LF]'-----'  
08F1  ODOA24     .ASCII [CR][LF][EOM]  
  
          ;*****  
08F4  MSG.SF:    ;SELECT FUNCTION                *  
          ;*****  
  
08F4  ODOA454E5445 .ASCII [CR][LF]'ENTER FUNCTION NUMBER: '[EOM]  
  
          ;*****
```

```
090E          ;*****  
MSG.RS:      ;READ SYSTEM ON DRIVE      *  
          ;*****  
090E  0D0A    .ASCII [CR][LF]  
0910  0D0A52454144 .ASCII [CR][LF]'READ SYSTEM FROM DRIVE: '[EOM]  
  
          ;*****  
092B          MSG.NC:      ;TRANSFER INCOMPLETE      *  
          ;*****  
092B  0D0A    .ASCII [CR][LF]  
092D  0D0A5452414E .ASCII [CR][LF]'TRANSFER INCOMPLETE'  
0942  0D0A24    .ASCII [CR][LF][EOM]  
  
          ;*****  
0945          MSG.FE:      ;FORMAT ERROR      *  
          ;*****  
0945  0D0A    .ASCII [CR][LF]  
0947  0D0A464F524D .ASCII [CR][LF]'FORMAT TRACK ERROR'  
095B  0D0A24    .ASCII [CR][LF][EOM]  
  
          ;*****  
095E          MSG.NR:      ;SYSTEM NOT LOADED      *  
          ;*****  
095E  0D0A    .ASCII [CR][LF]  
0960  0D0A53595354 .ASCII [CR][LF]'SYSTEM TRACKS NOT LOADED'  
097A  0D0A24    .ASCII [CR][LF][EOM]  
  
          ;*****
```


FORMAT - JADE DOUBLE D

INJECTION MODULE - MACRO DEFINITIONS

```

;*****
; FORMAT - TITLE BLOCK AND PAGE ALIGNMENT *
;*****

```

```

        .DEFINE FORMAT [NAME] = [
NAME     ==      (.!OFFH)+1      ;SET NEXT PAGE BOUNDRY.
        .LOC      NAME           ;SET LOC TO NEXT PAGE.
OFFSET   =      FMT.EA-NAME     ;DETERMINE ADDR OFFSET.
        .Z80                     ;NOW USE Z80 CODE.
        .ASCII   'FORMAT!'      ;INCLUDE HEADER!

```

```

;*****
; DENSITY - DECLARE TYPE *
;*****

```

```

        .DEFINE DENSITY [TYPE] = [
        .IFIDN [TYPE][SINGLE], [
        .ASCII 'S'
        .EXIT]
        .IFIDN [TYPE][DOUBLE], [
        .ASCII 'D'
        .EXIT]
        .ERROR 'INVALID DENSITY']

```

```

;*****
; SECTORS - SPECIFY SEQUENCE AND NUMBER OF SECTORS *
;*****

```

```

        .DEFINE SECTORS [LIST,NMBR] = [
        LXI     H,LIST+OFFSET    ;SECTOR SEQUENCE ADDR.
        MVI     E,NMBR          ;NUMBER OF SECTORS.

```

```

;*****
; BLOCK - GENERATE A BLOCK OF CONSTANTS *
;*****

```

```

        .DEFINE BLOCK [COUNT,BYTE,%REPT] = [
        NMBR = COUNT             ;SET EQUAL FOR NOW.
        MVI     B,NMBR          ;LOAD NMBR OF BYTES.
%REPT:  IN      XP.DSH          ;WAIT FOR DATA REQ.
        MVI     A,BYTE         ;LOAD BYTE VALUE.
        XRA     C              ;INVERT (1791-01).
        OUT     WD.DTA         ;WRITE DATA FORT.
        DJNZ   %REPT]         ;REPEAT FOR COUNT.

```

```

;*****

```

```
*****  
; REPEAT - REPEAT FORMAT SECTION FOR EACH SECTOR *  
*****  
  
        .DEFINE REPEAT [LOCATION] = [  
DCR     E           ;DEC NMBR SECTORS LEFT.  
JNZ     LOCATION+OFFSET]  
  
*****  
; ENDING - RECORD NMBR OF TRAILING BYTES WRITTEN *  
*****  
  
        .DEFINE ENDING [BYTE,%REPT] = [  
LXI     H,0         ;COUNT OF ZERO.  
%REPT: IN     XP.DSH   ;WAIT FOR REQ.  
MVI     A,BYTE     ;LOAD CONSTANT.  
XRA     C           ;INVERT (1791-01).  
OUT     WD.DTA     ;WRITE TO PORT.  
INX     H           ;INCREMENT COUNT.  
JMP     %REPT+OFFSET ;CONTINUE.]  
  
*****
```

```
*****  
; WRITE - WRITE SPECIFIC FORMAT BYTES *  
*****  
      .DEFINE WRITE [TYPE,VALU] = [  
  
***** ( ID ADDRESS MARK )*****  
      .IFIDN [TYPE][ID.MARK], [  
      IN      XP.DSH          ;WAIT FOR DATA REQ.  
      MVI     A,OFEH          ;ID ADDR MARK.  
      XRA     C               ;INVERT (1791-01).  
      OUT     WD.DTA          ;WRITE DATA PORT.  
      .EXIT]                 ;TERMINATE MACRO  
  
***** ( INDEX MARK )*****  
      .IFIDN [TYPE][INDEX.MARK], [  
      IN      XP.DSH          ;WAIT FOR DATA REQ.  
      MVI     A,OFCH          ;INDEX MARK.  
      XRA     C               ;INVERT (1791-01).  
      OUT     WD.DTA          ;WRITE DATA PORT.  
      .EXIT]                 ;TERMINATE MACRO  
  
***** ( DATA ADDRESS MARK )*****  
      .IFIDN [TYPE][DATA.MARK], [  
      IN      XP.DSH          ;WAIT FOR DATA REQ.  
      MVI     A,OFBH          ;DATA ADDR MARK.  
      XRA     C               ;INVERT (1791-01).  
      OUT     WD.DTA          ;WRITE DATA PORT.  
      .EXIT]                 ;TERMINATE MACRO  
  
***** ( CRC )*****  
      .IFIDN [TYPE][CRC], [  
      IN      XP.DSH          ;WAIT FOR DATA REQ.  
      MVI     A,OF7H          ;GENERATE CRC.  
      XRA     C               ;INVERT (1791-01).  
      OUT     WD.DTA          ;WRITE DATA PORT.  
      .EXIT]                 ;TERMINATE MACRO  
  
***** ( EXPLICIT BYTE VALUE )*****  
      .IFIDN [TYPE][BYTE], [  
      IN      XP.DSH          ;WAIT FOR DATA REQ.  
      MVI     A,VALU          ;EXPLICIT VALUE.  
      XRA     C               ;INVERT (1791-01).  
      OUT     WD.DTA          ;WRITE DATA PORT.  
      .EXIT]                 ;TERMINATE MACRO  
  
***** ( TRACK NUMBER )*****  
      .IFIDN [TYPE][TRACK.NO], [  
      IN      XP.DSH          ;WAIT FOR REQUEST.
```

```
IN      WD.TRK      ;GET TRACK NMBR.  
OUT     WD.DTA      ;WRITE DATA PORT.  
.EXIT]
```

***** (SECTOR NUMBER)*****

```
.IFIDN [TYPE][SECTOR.NO], [  
IN      XP.DSH      ;WAIT FOR REQUEST.  
MOV     A,M         ;SET SECTOR NMBR.  
XRA     C           ;INVERT (1791-01).  
OUT     WD.DTA      ;WRITE DATA PORT.  
INX     H           ;INC SEC-NMBR PNTR.  
.EXIT              ;TERMINATE MACRO]
```

***** (SIDE NUMBER)*****

```
.IFIDN [TYPE][SIDE.NO], [  
IN      XP.DSH      ;WAIT FOR REQUEST.  
MVI     A,0         ;SET SIDE NUMBER.  
XRA     C           ;INVERT (1791-01).  
OUT     WD.DTA      ;WRITE DATA PORT.  
.EXIT              ;TERMINATE MACRO]
```

***** (SECTOR SIZE CODE)*****

```
.IFIDN [TYPE][SECTOR.SIZE], [  
SEC.CD = OFFH      ;DECLARE BLANK.  
.IFIDN [VALU][128], [SEC.CD = 000H]  
.IFIDN [VALU][256], [SEC.CD = 001H]  
.IFIDN [VALU][512], [SEC.CD = 002H]  
.IFIDN [VALU][1024], [SEC.CD = 003H]  
.IFE     (SEC.CD-OFFH), [  
.ERROR  'INVALID SECTOR SIZE']  
IN      XP.DSH      ;WAIT FOR DATA REQ.  
MVI     A,SEC.CD    ;LOAD SIZE CODE.  
XRA     C           ;INVERT (1791-01).  
OUT     WD.DTA      ;WRITE DATA PORT.  
.EXIT              ;TERMINATE MACRO]
```

***** (ILLEGAL EXPANSION)*****

```
.ERROR 'ILLEGAL EXPANSION']
```

TDL Z80 CP/M DISK ASSEMBLER VERSION 2.21
 FORMAT - JADE DOUBLE D
 *INJECTION MODULE FT3740

0A00	464F524D4154		FORMAT	FT3740
0A07	53		DENSITY	SINGLE
0A08	21 17B5		SECTORS	SS3740,26
0A0D	0628	BG3740:	BLOCK	40, ONES
0A18	0606		BLOCK	6, ZEROS
0A23	DB80		WRITE	INDEX.MARK
0A2A	061A		BLOCK	26, ONES
0A35	0606	RP3740:	BLOCK	6, ZEROS
0A40	DB80		WRITE	ID.MARK
0A47	DB80		WRITE	TRACK.NO
0A4D	DB80		WRITE	SIDE.NO
0A54	DB80		WRITE	SECTOR.NO
0A5B	DB80		WRITE	SECTOR.SIZE,128
0A62	DB80		WRITE	CRC
0A69	060B		BLOCK	11, ONES
0A74	0606		BLOCK	6, ZEROS
0A7F	DB80		WRITE	DATA.MARK
0A86	0680		BLOCK	128, 0E5H
0A91	DB80		WRITE	CRC
0A98	061B		BLOCK	27, ONES
0AA3	1D		REPEAT	RP3740
0AA7	21 0000		ENDING	ONES
0AB5	010203040506	SS3740:	.BYTE	1, 2, 3, 4, 5, 6, 7, 8, 9,10
0ABF	0B0C0D0E0F10		.BYTE	11,12,13,14,15,16,17,18,19,20
0AC9	15161718191A		.BYTE	21,22,23,24,25,26

OB00	464F524D4154		FORMAT	FTJ50D
OB07	44		DENSITY	DOUBLE
OB08	21 17AE		SECTORS	SSJ50D,50
OB0D	0650	BGJ50D:	BLOCK	80,04EH
OB18	0608	RPJ50D:	BLOCK	8,ZEROS
OB23	0603		BLOCK	3,0F5H
OB2E	DB80		WRITE	ID.MARK
OB35	DB80		WRITE	TRACK.NO
OB3B	DB80		WRITE	SIDE.NO
OB42	DB80		WRITE	SECTOR.NO
OB49	DB80		WRITE	SECTOR.SIZE,128
OB50	DB80		WRITE	CRC
OB57	0616		BLOCK	22,04EH
OB62	060C		BLOCK	12,ZEROS
OB6D	0603		BLOCK	3,0F5H
OB78	DB80		WRITE	DATA.MARK
OB7F	0680		BLOCK	128,0E5H
OB8A	DB80		WRITE	CRC
OB91	0611		BLOCK	17,04EH
OB9C	1D		REPEAT	RPJ50D
OBA0	21 0000		ENDING	ONES
OBAE	010B151F29	SSJ50D:	.BYTE	1,11,21,31,41
OBB3	020C16202A		.BYTE	2,12,22,32,42
OBB8	030D17212B		.BYTE	3,13,23,33,43
OBBD	040E18222C		.BYTE	4,14,24,34,44
OBC2	050F19232D		.BYTE	5,15,25,35,45
OBC7	06101A242E		.BYTE	6,16,26,36,46
OBCC	07111B252F		.BYTE	7,17,27,37,47
OBD1	08121C2630		.BYTE	8,18,28,38,48
OBD6	09131D2731		.BYTE	9,19,29,39,49
OBD8	0A141E2832		.BYTE	10,20,30,40,50
			.END	

FORMAT - JADE DOUBLE D

+++++ SYMBOL TABLE +++++

BC.PTX	0009	BC.RCB	000A	BDS	0005	BEGIN	0100
BG3740	0A0D	BGJ50D	0B0D	BS.DMA	02C5	BS.DSK	02BC
BS.FMT	02D4	BS.PTR	0001	BS.RDS	02C8	BS.SEC	02C2
BS.TRK	02BF	BS.VSZ	0033	BS.WRM	02A4	BS.WRS	02CB
B.MOVE	02D7	CB.SIZ	0014	CNS.IN	029C	CR	000D
DD.FLG	0647	EOL	00FF	EOM	0024	FD.NBR	04DF
FMT.DD	0321	FMT.EA	1700	FMT.SD	031B	FMT.ST	023B
FT3740	0A00	FTJ50D	0B00	FT.ERC	00FE	FT.STS	04D7
FT.TSM	0001	FUN.1	01AA	FUN.2	0208	FUN.3	01D2
FUN.4	01EB	FUN.5	025D	F.FLAG	04DC	IDS.DD	0696
IDS.DS	0596	IDS.SD	0616	IDS.SS	0516	ID.SEC	0001
INIT	0146	LF	000A	LIST	0164	MSG.BG	0716
MSG.FD	0781	MSG.FE	0945	MSG.FL	07BB	MSG.NC	092B
MSG.NR	095E	MSG.OT	0297	MSG.RS	090E	MSG.SE	07A2
MSG.SF	08F4	MSG.SV	04D5	NMBR	0011	NG.LOG	0001
OFFSET	0C00	ONES	00FF	RC.BUF	0500	RC.NBR	0501
RC.TXT	0502	REBOOT	0000	RP3740	0A35	RPJ50D	0B18
RST.7	04CC	SD.FLG	0547	SEC.CD	0000	SEC.NO	04DB
SEC.SZ	0080	SELECT	016A	SEL.DR	0485	SP.TOP	0500
SS3740	0AB5	SSJ50D	0BAE	STACK	04E0	ST.DMA	0324
ST.LST	03A9	SV.NBR	04DE	SYS.RF	04DD	TF.DIR	04D4
TF.INX	04D0	TF.PTR	04D2	TK0	0000	TK1	0001
TPA	0100	TRK.0	0000	TRK.1	0001	TRK.2	0002
TRK.ER	028E	TRK.MX	04DA	TRK.NO	04D9	TRK.NX	0279
TRANSFR	0355	TS.FLG	04D8	WDD.ID	02F2	WD.DTA	0007
WD.TRK	0005	WRT.ID	02FF	WSD.ID	02E2	XP.DSH	0080
ZEROS	0000						

DDT

DDT VERS 2.2

-IFORMAT.COM

-R

NEXT PC

0C00 0100

-DA00, AFF

0A00 46 4F 52 4D 41 54 21 53 21 B5 17 1E 1A 06 28 DB FORMAT!S!.....(.
0A10 80 3E FF A9 D3 07 10 F7 06 06 DB 80 3E 00 A9 D3 .>.....>...
0A20 07 10 F7 DB 80 3E FC A9 D3 07 06 1A DB 80 3E FF>.....>.
0A30 A9 D3 07 10 F7 06 06 DB 80 3E 00 A9 D3 07 10 F7>.....
0A40 DB 80 3E FE A9 D3 07 DB 80 DB 05 D3 07 DB 80 3E ..>.....>...
0A50 00 A9 D3 07 DB 80 7E A9 D3 07 23 DB 80 3E 00 A9^...#...>..
0A60 D3 07 DB 80 3E F7 A9 D3 07 06 0B DB 80 3E FF A9>.....>..
0A70 D3 07 10 F7 06 06 DB 80 3E 00 A9 D3 07 10 F7 DB>.....
0A80 80 3E FB A9 D3 07 06 80 DB 80 3E E5 A9 D3 07 10 .>.....>.....
0A90 F7 DB 80 3E F7 A9 D3 07 06 1B DB 80 3E FF A9 D3 ...>.....>...
0AA0 07 10 F7 1D C2 35 17 21 00 00 DB 80 3E FF A9 D35.!.....>...
0AB0 07 23 C3 AA 17 01 02 03 04 05 06 07 08 09 0A 0B .#.....
0AC0 0C 0D 0E 0F 10 11 12 13 14 15 16 17 18 19 1A 00
0AD0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0AE0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0AF0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
-DB00, BFF
0B00 46 4F 52 4D 41 54 21 44 21 AE 17 1E 32 06 50 DB FORMAT!D!...2.P.
0B10 80 3E 4E A9 D3 07 10 F7 06 08 DB 80 3E 00 A9 D3 .>N.....>...
0B20 07 10 F7 06 03 DB 80 3E F5 A9 D3 07 10 F7 DB 80>.....>...
0B30 3E FE A9 D3 07 DB 80 DB 05 D3 07 DB 80 3E 00 A9 >.....>.....>..
0B40 D3 07 DB 80 7E A9 D3 07 23 DB 80 3E 00 A9 D3 07^...#...>.....
0B50 DB 80 3E F7 A9 D3 07 06 16 DB 80 3E 4E A9 D3 07 ..>.....>N...
0B60 10 F7 06 0C DB 80 3E 00 A9 D3 07 10 F7 06 03 DB>.....
0B70 80 3E F5 A9 D3 07 10 F7 DB 80 3E FB A9 D3 07 06 .>.....>.....
0B80 80 DB 80 3E E5 A9 D3 07 10 F7 DB 80 3E F7 A9 D3 ...>.....>...
0B90 07 06 11 DB 80 3E 4E A9 D3 07 10 F7 1D C2 18 17>N.....
0BA0 21 00 00 DB 80 3E FF A9 D3 07 23 C3 A3 17 01 0B !.....>.....#.....
0BB0 15 1F 29 02 0C 16 20 2A 03 0D 17 21 2B 04 0E 18 ..)....*...!+...
0BC0 22 2C 05 0F 19 23 2D 06 10 1A 24 2E 07 11 1B 25 ",...#-...\$....%
0BD0 2F 08 12 1C 26 30 09 13 1D 27 31 0A 14 1E 28 32 /...&0...^1...(2
0BE0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0BF0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
-

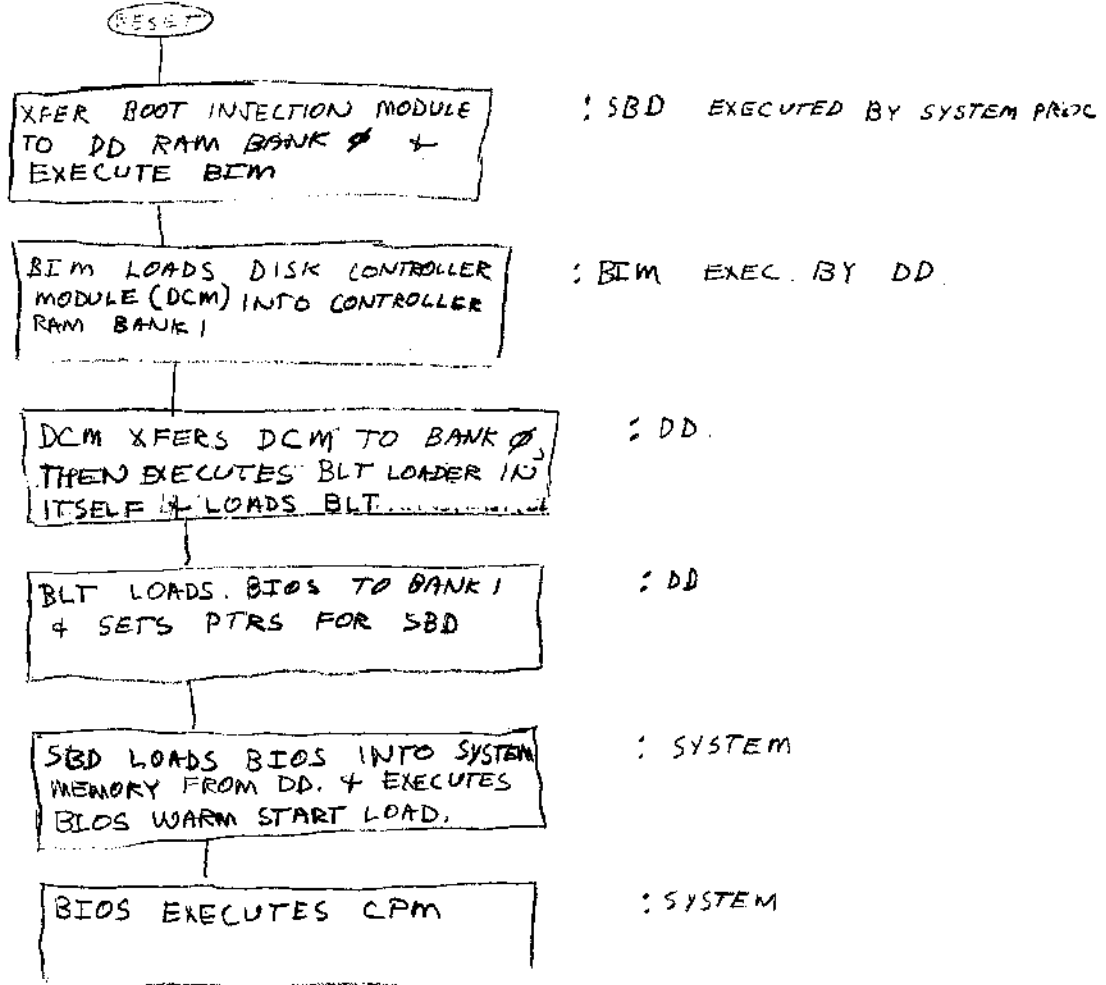
JADE

BOOT UP

- 1) PGM TO INJECT DCM INTO DD
- 2) PGM TO ~~WRITE~~ FORMAT DISK
- 3) " TO WRITE SYSTEM ON DISK
- 4) " TO COPY CP/M TO DISK
- 5) MODIFY CPM BIOS
- 6) BURN PROM(S) FOR BOOT

< HDD DEBUG

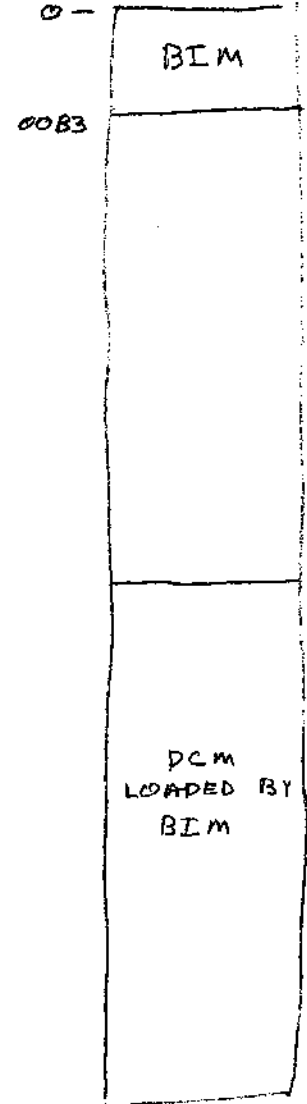
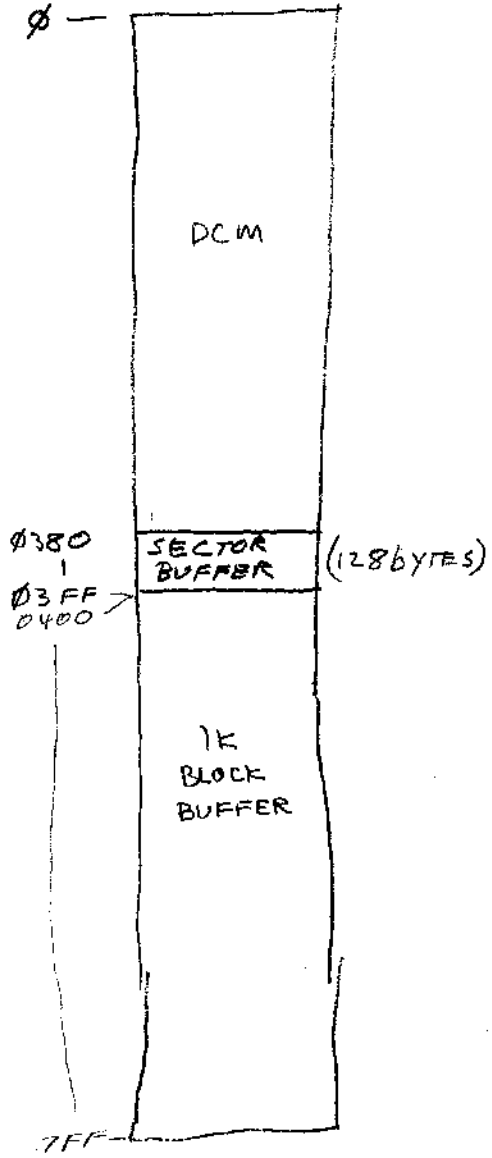
JADE DD COLD START



DD MEM. MAP

AFTER BOOT

BEFORE BOOT



BIM LOADED BY SYSTEM