

JADE COMPUTER PRODUCTS  
CP/M 2.2 - DOUBLE D

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.
BIM.ASM	BOOTSTRAP INJECTION MODULE - PART 2 OF A SYSTEM BOOTSTRAP PROM. IN TDL MACRO ASSEMBLER SOURCE.
BIM.HEX	BOOTSTRAP INJECTION MODULE - THIS FILES IN INTEL HEX FORMAT.
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 CBIOS RESIDES AT 1000H TO 13FFH. USE DDT TO PUT THE CBIOS IMAGE AT THAT ADDRESS.
BLT.ASM	BIOS LOADER TRANSIENT - SOURCE CODE FOR "COLD START LOADER" AS GENERATED BY MOVCPM.COM.
BOOT.COM	BOOT - PRESENTED AS A COM FILE, IF EXECUTED BY A SINGLE DENSITY CP/M SYSTEM IN WHICH THE DOUBLE D ALSO IS PRESENT, THIS WILL BOOT THE DOUBLE D SYSTEM.
CBIOS.ASM	BASIC I/O SYSTEM - SKELETAL BIOS PROVIDED BY DIGITAL RESEARCH.
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.
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.
SBD.ASM	SYSTEM BOOT DRIVER - PART 1 OF SYSTEM PROM BOOT. ASSEMBLED SOURCE LISTING IS IN SOFTWARE MANUAL.
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 COMPATABLE. 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
01	N.A.	1000H	IDT
02	1380H (DD)	1080H	BLT
03		1100H	
04	4A00H+B	1180H	BIOS
05	4A80H+B	1200H	BIOS
06	4B00H+B	1280H	BIOS
07	4B80H+B	1300H	BIOS
08	4C00H+B	1380H	BIOS
09	4C80H+B	1400H	BIOS
10	4D00H+B	1480H	BIOS
11	4D80H+B	1500H	BIOS
12		1580H	RSV
13	1000H (DD)	1600H	DCM
14	1080H (DD)	1680H	DCM
15	1100H (DD)	1700H	DCM
16	1180H (DD)	1780H	DCM
17	1200H (DD)	1800H	DCM
18	1280H (DD)	1880H	DCM
19	1300H (DD)	1900H	DCM
20	1380H (DD)	1980H	DCM
21	1400H (DD)	1A00H	RSV
22	1480H (DD)	1A80H	RSV
23	1500H (DD)	1B00H	RSV
24	1580H (DD)	1B80H	RSV
25	1600H (DD)	1C00H	RSV
26	1680H (DD)	1C80H	RSV

THE FOLLOWING TABLE SHOWS THE LAYOUT OF SYSTEM TRACK 1. THIS TRACK IS FORMATTED IN DOUBLE DENSITY WITH 48 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

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 SIMILIAR TO THE SYSGEN.COM DESCRIBED IN THE CP/M 2.0 MANUAL SET. NOTICE THE DIFFERENCE IS THE BIOS MODULE. EACH OF THESE PROGRAMS PROVIDE A FILE LOAD FEATURE. AN EXAMPLE, IF YOU HAD A FILE CPM32.COM AND YOU WISHED TO WRITE THIS FILE TO THE SYSTEM TRACKS.

```
A>DDT CPM32.COM
(DDT DISPLAYS MESSAGE)
-GO
A>SYSGEN
(AND THEN WRITE YOUR FILE)
```

```
A>SYSGEN CPM32.COM
(YOU GET A FILE LOADED MESSAGE)
NOW WRITE YOUR FILE.
```

IN CHANGING THE CP/M SYSTEM SIZE, YOU MUST REWRITE BLT/CCP/BDOS USING SYSGEN.COM AND REWRITE BIOS USING BIOSGEN.COM. DCMGEN IS USED TO MODIFY DCM FOR SUCH PARAMETERS AND DISK DRIVE STEP TIMING. CONSULT YOUR DCM LISTING FOR THE TIMING PARAMETERS.

UTILITY	MODULE	TRACK	SECTORS	SYSYSTEM 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

NOTES FOR RELEASE #1 DISKETTES. THE DDBIOS IS ASSEMBLED FOR A REV B DOUBLE D CONTROLLER BOARD. DDBIOS SOURCE SHOULD BE MODIFIED, REASSEMBLED, AND WRITTEN TO SYSTEM TRACK 0 FOR REV C BOARDS.



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.

THE FOLLOWING JUMPERS SHOULD BE USED TO CONFIGURE A SHUGART 800 DISK DRIVE.

EACH DRIVE:      A,    B,    <sup>Y</sup>C,    <sup>Z</sup>Y,    T2,    <sup>HL</sup>DS,    <sup>801</sup>800

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

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

THE L JUMPER IS SET DEPENDING ON THE -5V SUPPLY.  
CONSULT YOU SAs00 MANUAL.

USE ONLY THESE JUMPERS, NO OTHERS.



DIGITAL RESEARCH CP/M R 22 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.

• •

)

)

)

```
*****
;
; DISK CONTROLLER MODULE (DCM)
;
*****
;
; PROPERTY OF: JADE COMPUTER PRODUCTS INC.
; 4901 W. ROSECRANS BLVD.
; HAWTHORNE, CALIFORNIA
; 90250, U.S.A.
;
*****
;
; VERSION: 2.20 - 8" DRIVES
;
*****
;
; WRITTEN BY: STAN KRUMME
;
*****
; THE DISK CONTROLLER MODULE (DCM) RESIDES 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 6 WORD TIMING BLOCK WHICH
; SHOULD BE PATCHED TO MATCH THE USERS DISK DRIVES.
; THIS HAS NORMALLY BEEN SET FOR SHUGART SA800/801.
*****
```

\*\*\*\*\*

; 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	;1791 COMMAND REGISTER.
0004	WD.STS	==	004H	;1791 STATUS REGISTER.
0005	WD.TRK	==	005H	;1791 TRACK REGISTER.
0006	WD.SEC	==	006H	;1791 SECTOR REGISTER.
0007	WD.DTA	==	007H	;1791 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.DDS	==	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.
0000	BC.PCZ	==	0	;PRECOMP - OFF.

\*\*\*\*\*

\*\*\*\*\*

; 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 INDICATOR.
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 1791-01 DISK CONTROLLER.

0010	DC.HDL	==	00010000B	;SEEK/LOAD RW HEAD.
0018	DC.HDU	==	00011000B	;SEEK/UNLD RW HEAD.
0080	DC.RDS	==	10000000B	;READ SECTOR.
00A0	DC.WRS	==	10100000B	;WRITE SECTOR.
00F0	DC.WRT	==	11110000B	;WRITE TRACK FORMAT.
00C0	DC.RDA	==	11000000B	;READ TRACK ADDRESS.
00D0	DC.STS	==	11010000B	;SET TYPE 1 STATUS
00E8	DC.IFI	==	11011000B	;FORCED INTERRUPT.

\*\*\*\*\*

; THE FOLLOWING LIST CONTAINS ALL THE  
; MASKS USED TO TEST THE 1791-01 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.

\*\*\*\*\*

\*\*\*\*\* (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.

\*\*\*\*\*

```
*****  
; PAUSE IS USED AFTER THE 179X-01 IS ISSUED A COMMAND  
; AND BEFORE THE 179X-01 STATUS REGISTER IS READ. *  
; THIS ALLOWS THE 179X-01 TO 'DIGEST' THE COMMAND! *  
*****
```

```
.DEFINE PAUSE = [  
XTHL  
XTHL  
XTHL  
XTHL]
```

```
*****  
; WAIT IS A RESTART CALL 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  
.PHEX  
.XLINK
```

```
***** (TIMING CONSTANTS) *****
```

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

```
***** (RETRY VALUES) *****
```

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

```
*****
```

```

;*****
; 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          ;BEGINNING.
1000      C3 0000      JMP          0          ;NOT IMPLEMENTED YET.
1003      C3 1780      JMP          INIT.B+BANK.L ;INIT - BOOTSTRAPED.
1006      1041      HR.VEC: .WORD      X.CUTE      ;BUS 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 RESTART LOCATIONS ARE SPARES AT THIS *
; TIME. THEY MAY BE USED IN FUTURE UPGRADES. *
;*****
    
```

```

1018          .LOC      RST.3
1020          .LOC      RST.4
1028          .LOC      RST.5
1030          .LOC      RST.6
    
```

```

;*****
; 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 REG 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          XP.IRR      ;RESET INTERRUPT REG FF
103A      D1        POP          D          ;PURGE INTERRUPTED ADDR
103B      2A 1006      LHL         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      ..MASK      ;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,..CMDT     ;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.

```

```

;*****

```

```

1053          ..CMDT ==          .          ;COMMAND TABLE.
1053    10AF       ..CM0A: .WORD  $.LGON    ;LOG-ON DRIVE.
1055    107C       ..CM1A: .WORD  $.READ    ;READ SECTOR.
1057    108B       ..CM2A: .WORD  $.WRIT    ;WRITE SECTOR.
1059    109A       ..CM3A: .WORD  $.FORM    ;FORMAT TRACK.
105B    10CC       ..CM4A: .WORD  $.ADDR    ;READ ADDRESS.
105D    10CF       ..CM5A: .WORD  $.SFAR    ;SPARE.
105F    10CF       ..CM6A: .WORD  $.SFAR    ;SPARE.
1061    10D2       ..CM7A: .WORD  $.IDLE    ;BACKGROUND.
0007          ..MASK ==          007H      ;COMMAND MASK.

```

```

;*****

```

```

;*****
; THE FOLLOWING SECTION IS THE NON-MASKABLE INTERRUPT*
; ROUTINE. UPON 1791-01 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 1791 STATUS.
1068  A9        XRA      C              ;INVERT (1791).
1069  32 130D    STA      SV.STS      ;SAVE STATUS.
106C  F0E3      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 10DD   $.READ: CALL   SELECT   ;SELECT DRIVE ROUTINE.
107F   C2 103F           JNZ     FETCH   ;DRIVE NOT READY EXIT.
1082   CD 1180           CALL    SEEK    ;SEEK TRACK, SET CTLS.
1085   CD 1265           CALL    RD.SEC  ;READ DISK SECTOR.
1088   C3 103F           JMP     FETCH   ;GET NEXT COMMAND.
    
```

```

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

```

108B   CD 10DD   $.WRIT: CALL   SELECT   ;SELECT DRIVE ROUTINE.
108E   C2 103F           JNZ     FETCH   ;DRIVE NOT READY EXIT.
1091   CD 1180           CALL    SEEK    ;SEEK TRACK, SET CTLS.
1094   CD 1274           CALL    WR.SEC  ;WRITE DISK SECTOR.
1097   C3 103F           JMP     FETCH   ;GET NEXT COMMAND.
    
```

```

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

```

109A   CD 10DD   $.FORM: CALL   SELECT   ;SELECT DRIVE NUMBER.
109D   C2 103F           JNZ     FETCH   ;DRIVE NOT READY EXIT.
10A0   3A 1373           LDA     CB.SEC  ;LOAD FORMAT FLAGS.
10A3   DD7705           MOV     DV.FLG(X),A ;RESET DRIVE FLAGS.
10A6   CD 1180           CALL    SEEK    ;SEEK TRACK, SET CTLS.
10A9   CD 1203           CALL    WR.TRK  ;WRITE DISK TRACK.
10AC   C3 103F           JMP     FETCH   ;GET NEXT COMMAND.
    
```

```

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

```

10AF   CD 10DD   $.LGON: CALL   SELECT   ;SELECT DRIVE NUMBER.
10B2   C2 103F           JNZ     FETCH   ;DRIVE NOT READY EXIT.
10B5   AF             XRA     A       ;ZERO REGISTER A.
10B6   32 1372           STA     CB.TRK  ;SET TRACK AT 0.
10B9   3C             INR     A       ;NOW A REG IS 1.
10BA   32 1373           STA     CB.SEC  ;SET SECTOR TO ID.
10BD   CD 1180           CALL    SEEK    ;SEEK TRACK, SET CTLS.
10C0   C2 103F           JNZ     FETCH   ;POSSIBLE ERROR EXIT.
10C3   CD 1265           CALL    RD.SEC  ;READ ID SECTOR.
10C6   CD 114B           CALL    LOG.ON  ;LOG ON DISK DRIVE.
10C9   C3 103F           JMP     FETCH   ;GET NEXT COMMAND.
    
```

```

;*****
    
```

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

```
100C C3 103F $.ADDR: JMP FETCH ;NOT IMPLEMENTED.
```

```
*****  
; $.SPAR IS A SPARE COMMAND CONTROLLER. *  
*****
```

```
100F C3 103F $.SPAR: JMP FETCH ;GET NEXT COMMAND.
```

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

```
10D2 DB00 $.IDLE: IN BL.STS ;INPUT BOARD STATUS.  
10D4 E608 ANI BS.INT ;CHECK HOST INTERRUPT.  
10D6 28FA JRZ $.IDLE ;REPEAT IDLE CHECK.  
10D8 DB20 IN XP.IRR ;RESET INTERRUPT REQ.  
10DA C3 103F JMP FETCH ;GET NEXT COMMAND.
```

```
*****
```

```

;*****
; THE FOLLOWING SUBROUTINE IS RESPONSIBLE FOR
; SELECTING THE COMMAND REQUESTED DRIVE NMBR.
; BEFORE DRIVE SELECTION, THE DRIVE MOTOR STATE
; IS TESTED, AND IF NEEDED THEY ARE TURNED ON.
; INDEX REGISTER X IS SET TO POINT TO THE NEW
; SELECTED DRIVE IN THE DRIVE TABLE.
;*****

```

\*\*\*\*\* (MOTOR CHECK ROUTINE) \*\*\*\*\*

```

10DD DB00 SELECT: IN BL.STS ;BOARD LEVEL STATUS.
10DF E620 ANI BS.MOF ;CHECK MOTOR STATE.
10E1 2807 JRZ ..CKDV ;IF ON, CHECK DRIVE.
10E3 DB40 IN XP.MTX ;ISSUE MOTOR START.
10E5 ED5B 1314 LDED TM.MTO ;MOTOR STARTUP DELAY.
10E9 CF WAIT ;PROGRAMMABLE DELAY.

```

\*\*\*\*\* (SELECT DRIVE ROUTINE) \*\*\*\*\*

```

10EA DB40 ..CKDV: IN XP.MTX ;EXTEND MOTOR TIMER.
10EC 3A 1371 LDA CB.DRV ;LOAD DRIVE NUMBER.
10EF DBBE00 CMP DV.NBR(X) ;CURRENTLY SELECTED?
10F2 2831 JRZ ..DVOK ;GO IF DRIVE SELECTED.
10F4 CD 124B CALL EX.HDU ;UNLOAD R/W HEAD.
10F7 3A 130C LDA SV.DRV ;LOAD PORT IMAGE.
10FA E6FB ANI #BC.DSE ;DRIVE SELECT DSBLD.
10FC D300 OUT BL.CTL ;ISSUE Deselect.
10FE 3A 1371 LDA CB.DRV ;LOAD DESIRED DRIVE.
1101 E603 ANI BC.DSN ;MASK DRIVE NUMBER.
1103 D300 OUT BL.CTL ;OUTPUT DRIVE NMBR.
1105 F604 ORI BC.DSE ;SET DRV ENABLE BIT.
1107 D300 OUT BL.CTL ;ENABLE NEW DRIVE.
1109 32 130C STA SV.DRV ;SAVE DRIVE SELECT.

```

\*\*\*\*\* (SET TABLE POINTER) \*\*\*\*\*

```

110C E603 ANI BC.DSN ;GET DRIVE NUMBER.
110E DD21 1322 LXI X,DV.TBL ;DRIVE TABLE ADDR.
1112 11 0007 LXI D,DV.DES ;DRIVE ENTRY SIZE.
1115 3D ..NEXT: DCR A ;DECREMENT DRV NO.
1116 FA 1110 JM ..SLED ;IF S=1 EXIT.
1119 DD19 DADX D ;POINT NEXT DRIVE.
111B 18F8 JMFR ..NEXT ;TRY THIS DRIVE.

```

\*\*\*\*\* (EXIT) \*\*\*\*\*

```

111D CD 1258 ..SLED: CALL EX.STS ;CHECK DRIVE STATUS.
1120 E680 ANI DM.DNR ;CHECK DRIVE NOT RDY.
1122 32 1377 STA CB.STS ;SET COMMAND STATUS.
1125 C9 ..DVOK: RET ;RETURN TO CALLER.

```

\*\*\*\*\*

```
*****  
; 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) *****
```

```
1126 2EFF HOME.D: MVI L,255 ;SET STEP COUNTER.  
1128 CD 1258 ..STEP: CALL EX.STS ;CHECK DISK STATUS.  
112B E604 ANI DM.TKO ;INSPECT TRACK 0 FLG.  
112D 200C JRNZ ..EXIT ;IF SET, GO ..EXIT.  
112F 2D DCR L ;DECREMENT STEP COUNT.  
1130 2815 JRZ ..ERROR ;ERROR IF 255 STEPS.  
1132 DE08 IN XP.STP ;ISSUE STEP PULSE.  
1134 ED5B 1310 LDED TM.STP ;LOAD STEP DELAY.  
1138 CF WAIT ;PROGRAMMABLE DELAY.  
113F 18ED JMPR ..STEP ;TRY STEPPING AGAIN.
```

```
***** (DRIVE IS RESTORED) *****
```

```
113B ED5B 1312 ..EXIT: LDED TM.ALS ;TIME AFTER LAST STEP.  
113F CF WAIT ;PROGRAMMABLE DELAY.  
1140 3E00 MVI A,000H ;SET COMPLETE FLAG.  
1142 DD7701 MOV DV.TRK(X),A ;SET TRACK VALUE.  
1145 A7 ANA A ;SET RETURN FLAGS.  
1146 C9 RET ;RETURN TO CALLER.
```

```
***** (TRACK 0 NOT FOUND) *****
```

```
1147 3EFF ..ERROR: MVI A,OFFH ;SET FAILURE FLAG.  
1149 A7 ANA A ;SET RETURN FLAGS.  
114A 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) \*\*\*\*\*

```

114E 11 131A LOG.ON: LXI D, JADEID ;ID ADDRESS LOADED.
114E 21 1380 LXI H, ID.LBL ;SECTOR ID ADDRESS.
1151 0608 MVI B, ID.SZE ;ID LABEL SIZE.
1153 1A ..CKJI: LDAX D ;GET CHARACTER.
1154 BE CMP M ;CHECK AGAINST DISK.
1155 2017 JRNZ ..3740 ;IF DIFFERENT: 3740.
1157 10FA DJNZ ..CKJI ;REPEAT OPERATION.
    
```

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

```

1159 3A 13A0 LDA ID.SPT ;SECTORS PER TRACK.
115C DD7703 MOV DV.SPT(X),A ;STORE IN DRIVE TBL.
115F 3A 13B0 LDA ID.STG ;STAGGER NUMBER.
1162 DD7703 MOV DV.SPT(X),A ;STORE IN DRIVE TBL.
1165 3A 13B1 LDA ID.FLG ;SIDE AND DENSITIES.
1168 DD7705 MOV DV.FLG(X),A ;STORE IN DRIVE TBL.
116B C3 103F JMP FETCH ;GET NEXT COMMAND.
    
```

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

```

116E 3E1A ..3740: MVI A, 26 ;SECTORS PER TRACK.
1170 DD7703 MOV DV.SPT(X),A ;STORE IN DRIVE TBL.
1173 3E01 MVI A, 1 ;STAGGER NUMBER.
1175 DD7703 MOV DV.SPT(X),A ;STORE IN DRIVE TBL.
1178 3E00 MVI A, ID.FLD ;SIDE AND DENSITIES.
117A DD7705 MOV DV.FLG(X),A ;STORE IN DRIVE TBL.
117D C3 103F JMP FETCH ;GET NEXT COMMAND.
    
```

\*\*\*\*\*

```

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

```

```

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

```

```

1180 CD 1258      SEEK:  CALL    EX.STS      ;GET DRIVE STATUS.
1183 E620        ANI     DM.HDL      ;CHECK HEAD LOAD.
1185 2008        JRNZ    ..HLDD      ;BYPASS IF LOADED.
1187 CD 123E     CALL    EX.HDL      ;LOAD R/W HEAD.
118A ED5B 130E   LDED    TM.HLD      ;HEAD LOAD DELAY.
118E CF         WAIT    ;PROGRAMMABLE DELAY.

```

```

;***** (CHECK TRACK NUMBER) *****

```

```

118F 3A 1372     ..HLDD: LDA     CB.TRK      ;REQUESTED TRACK NO.
1192 DD9601      SUB     DV.TRK(X)   ;TRACK OFFSET REQ.
1195 2006        JRNZ    ..DSET      ;IF OFFSET, SET DIRC.
1197 DD7E06      MOV     A,DV.CTL(X)  ;GET PREVIOUS CTLS.
119A C3 122B     JNP     ..EXIT      ;GOTO SET CONTROLS.

```

```

;***** (SET DIRECTION AND STEP COUNTER) *****

```

```

119D F5         ..DSET: PUSH   PSW        ;SAVE REG A AND FLGS.
119E ED5B 1318  LDED    TM.SAW     ;STEP AFTER WRITE.
11A2 CF         WAIT    ;PROGRAMMABLE DELAY.
11A3 F1         POP     PSW        ;RESTORE A AND FLGS.
11A4 380A       JRC     ..SOUT     ;IF CARRY STEP OUT.
11A6 6F         ..SIN: MOV     L,A      ;MOVE OFFSET TO L.
11A7 3A 130C   LDA     SV.DRV     ;DRIVE SELECT BITS.
11AA F620       ORI     BC.DAS     ;SET DIRECTION IN.
11AC D300       OUT    BL.CTL     ;OUTPUT CONTROL.
11AE 180B       JMFR   ..STEP     ;GOTO STEP ROUTINE.
11B0 ED44       ..SOUT: NEG     ;COMPLEMENT OFFSET.
11B2 FA 1238   JM     ..HOME     ;BETTER HOME DRV.
11B5 6F         MOV     L,A      ;MOVE OFFSET TO L.
11B6 3A 130C   LDA     SV.DRV     ;DRIVE SELECT BITS.
11B9 D300       OUT    BL.CTL     ;SET DIRECTION OUT.

```

```

;***** (ISSUE STEPS) *****

```

```

11B8 DB08       ..STEP: IN     XP.STP   ;ISSUE STEP PULSE.
11BD ED5B 1310  LDED    TM.STP   ;STEP DELAY TIME.
11C1 CF         WAIT    ;PROGRAMMABLE DELAY.
11C2 2D         DCR     L          ;DECREMENT STEPS.
11C3 20F6       JRNZ    ..STEP     ;REPEAT OPERATION.
11C5 ED5B 1312  LDED    TM.ALS   ;MORE AFTER LAST STP.
11C9 CF         WAIT    ;PROGRAMMABLE DELAY.

```



\*\*\*\*\* (TRACK SECTION DISTRIBUTE) \*\*\*\*\*

```

110A    3A 1372          LDA    CB,TRK          ;CURRENT TRACK NMBR.
110D    FE01            CPI    1              ;COMPARE AGAINST 1.
110F    380C            JRC    ..TRK0         ;IF CARRY, TRACK 0.
11D1    2811            JRZ    ..TRK1         ;IF ZERO, TRACK 1.
11D3    FE1A            CPI    26             ;CHECK OUTER THIRD.
11D5    381E            JRC    ..OUTS        ;GOTO OUTS SETTING.
11D7    FE34            CPI    52             ;CHECK MIDDLE THIRD.
11D9    382B            JRC    ..MIDS        ;GOTO MIDS SETTING.
11DB    303A            JRNC   ..INSD        ;GOTO INSD SETTING.
    
```

\*\*\*\*\* (TRACK ZERO) \*\*\*\*\*

```

11DD    3A 130C    ..TRK0: LDA    SV,DRV          ;GET DRIVE NUMBER.
11E0    F6C0            ORI    BC,SDS!BC,PCL    ;SDEN AND PC-LOW.
11E2    1844            JMPR   ..CTLS          ;JUMP CONTROLS SET.
    
```

\*\*\*\*\* (TRACK ONE) \*\*\*\*\*

```

11E4    DDCB054E    ..TRK1: BIT    DF,T1D,DV,FLG(X) ;TRK 1 DENSITY FLG.
11E8    3A 130C          LDA    SV,DRV          ;GET DRIVE NUMBER.
11EB    2004          JRNZ   ..T1DD         ;GO TO SET DDEN.
11ED    F6C0            ORI    BC,SDS!BC,PCL    ;SDEN AND PC-LOW.
11EF    1837            JMPR   ..CTLS          ;JUMP CONTROLS SET.
11F1    F6D0    ..T1DD: ORI    BC,DDS!BC,PCL    ;DDEN AND PC-LOW.
11F3    1833            JMPR   ..CTLS          ;JUMP CONTROLS SET.
    
```

\*\*\*\*\* (OUTSIDE TRACKS) \*\*\*\*\*

```

11F5    DDCB0556    ..OUTS: BIT    DF,DTD,DV,FLG(X) ;DATA DENSITY FLG.
11F9    3A 130C          LDA    SV,DRV          ;GET DRIVE NUMBER.
11FC    2004          JRNZ   ..OTDD         ;GO TO SET DDEN.
11FE    F6C0            ORI    BC,SDS!BC,PCL    ;SDEN AND PC-LOW.
1200    1826            JMPR   ..CTLS          ;JUMP CONTROLS SET.
1202    F6D0    ..OTDD: ORI    BC,DDS!BC,PCL    ;DDEN AND PC-LOW.
1204    1822            JMPR   ..CTLS          ;JUMP CONTROLS SET.
    
```

\*\*\*\*\* (MIDDLE TRACKS) \*\*\*\*\*

```

1206    DDCB0556    ..MIDS: BIT    DF,DTD,DV,FLG(X) ;DATA DENSITY FLG.
120A    3A 130C          LDA    SV,DRV          ;GET DRIVE NUMBER.
120D    2004          JRNZ   ..MDDD         ;GO TO SET DDEN.
120F    F6C0            ORI    BC,SDS!BC,PCL    ;SDEN AND PC-LOW.
1211    1815            JMPR   ..CTLS          ;JUMP CONTROLS SET.
1213    F690    ..MDDD: ORI    BC,DDS!BC,PCM    ;DDEN AND PC-MED.
1215    1811            JMPR   ..CTLS          ;JUMP CONTROLS SET.
    
```

\*\*\*\*\* (INSIDE TRACKS) \*\*\*\*\*

```
1217 DDCB0556 ..INSD: BIT DF.DTD,DV.FLG(X) ;DATA DENSITY FLG.
1218 3A 130C LDA SV.DRV ;GET DRIVE NUMBER.
121E 2004 JRNZ ..INDD ;GO TO SET IDEN.
1220 F6C0 ORI BC.SDS!BC.PCL ;SDEN AND PC-LOW.
1222 1804 JMFR ..CTLS ;JUMP CONTROLS SET.
1224 F650 ..INDD: ORI BC.DDS!BC.PCH ;IDEN AND PC-HI.
1226 1800 JMFR ..CTLS ;JUMP CONTROLS SET.
```

\*\*\*\*\* (SET CONTROLS) \*\*\*\*\*

```
1228 DD7706 ..CTLS: MOV DV.CTL(X),A ;SAVE CONTROLS.
122B D300 ..EXIT: OUT BL.CTL ;OUTPUT CONTROLS.
```

\*\*\*\*\* ( SET TRACK VALUES AND EXIT ) \*\*\*\*\*

```
122D 3A 1372 ..TKVL: LDA CB.TRK ;REQUESTED TRACK NO.
1230 DD7701 MOV DV.TRK(X),A ;SET DRIVE TABLE.
1233 A9 XRA C ;INVERT (1791-01).
1234 D305 OUT WD.TRK ;SET TRACK REGISTER.
1236 AF XRA A ;SET ZERO FLAG.
1237 C9 RET ;RETURN TO CALLER.
```

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

```
1238 CD 1126 ..HOME: CALL HOME.D ;HOME SELECTED DRIVE.
123B C3 118F JMP ..HLDD ;NOW SEEK TRACK.
```

\*\*\*\*\*

```

;*****
; THE FOLLOWING TWO SUBROUTINES PROVIDE THE R/W HEAD *
; LOAD AND UNLOAD FUNCTIONS. AS THE 179X-01 DOES NOT *
; OFFER THIS EXPLICIT COMMAND, THE SEEK COMMAND *
; (TYPE-1) IS USED WITH THE LOAD BIT SET TO EITHER *
; LOAD OR UNLOAD. THE DESTINATION TRACK IS SET EQUAL *
; TO THE CONTENTS OF THE TRACK REGISTER TO BYPASS THE *
; 179X-01 STEP FUNCTION. REFER TO THE FLOW-CHART FOR *
; 179X-01 TYPE 1 COMMANDS. *
;*****
    
```

\*\*\*\*\* (LOAD R/W HEAD) \*\*\*\*\*

```

123E   FDE1   EX.HDL: POP     Y           ;RETURN ADDR IN REG Y.
1240   DB05   IN      WD,TRK   ;READ PRESENT TRACK.
1242   D307   OUT     WD,DTA   ;SET DESTINATION TRK.
1244   3E10   MVI    A,DC.HDL  ;LOAD HEAD LOAD CMND.
1246   A9     XRA    C         ;INVERT (1791-01).
1247   D304   OUT     WD,CMD   ;ISSUE COMMAND.
1249   18FE   JMPR   .         ;WAIT FOR INTERRUPT.
    
```

\*\*\*\*\* (UNLOAD R/W HEAD) \*\*\*\*\*

```

124B   FDE1   EX.HDU: POP     Y           ;RETURN ADDR IN REG Y.
124D   DB05   IN      WD,TRK   ;READ PRESENT TRACK.
124F   D307   OUT     WD,DTA   ;SET DESTINATION TRK.
1251   3E18   MVI    A,DC.HDU  ;LOAD UNLD HEAD CMND.
1253   A9     XRA    C         ;INVERT (1791-01).
1254   D304   OUT     WD,CMD   ;ISSUE COMMAND.
1256   18FE   JMPR   .         ;WAIT FOR INTERRUPT.
    
```

```

;*****
; THE FOLLOWING SUBROUTINE IS USED TO UPDATE THE *
; 179X-01 STATUS PORT TO REFLECT CURRENT TYPE-1 *
; STATUS CODES. NOTE THIS IS A TYPE-4 COMMAND *
; WITH NO INTERRUPT CONDITIONS SET. *
;*****
    
```

```

1258   3ED0   EX.STS: MVI    A,DC.STS  ;LOAD SET-STATUS CMND.
125A   A9     XRA    C         ;INVERT (1791-01).
125B   D304   OUT     WD,CMD   ;ISSUE COMMAND.
125D   E3     PAUSE   ;PAUSE FOR 179X-01.
1261   DB04   IN      WD,STS   ;INPUT STATUS PORT.
1263   A9     XRA    C         ;INVERT (1791-01).
1264   C9     RET     ;RETURN TO USER.
    
```

\*\*\*\*\*

```

;*****
; RD.SEC IS THE SUBROUTINE THAT INTERACTS WITH THE *
; 179X-01 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) \*\*\*\*\*

```

1265 AF RD.SEC: XRA A ;ZERO A REGISTER.
1266 32 130B STA ERR.CT ;ZERO ERROR COUNT.
1269 3A 1373 LDA CB.SEC ;LOAD SECTOR NMBR.
126C A9 XRA C ;INVERT (1791-01).
126D D306 OUT WD.SEC ;SET SECTOR REGISTER.
126F FD21 1288 ..RTRY: LXI Y,..NMI ;LOAD NMI VECTOR.
1273 2A 1309 1325 LHL D BUF.ST ;BUFFER START.
1276 3E80 MVI A,DC.RDS ;READ SECTOR COMMAND.
1278 A9 XRA C ;INVERT (1791-01).
1279 D304 OUT WD.CMD ;ISSUE READ COMMAND
127B E3 PAUSE ;ALLOW 1791 TO SETTLE
    
```

\*\*\*\*\* (DATA TRANSFER LOOP) \*\*\*\*\*

```

127F DB80 ..REPT: IN XP.DSH ;HOLD FOR DATA
1281 D807 IN WD.DTA ;INPUT DATA.
1283 A9 XRA C ;INVERT (1791-01).
1284 77 MOV M,A ;PUT INTO BUFFER
1285 23 INX H ;BUMP BUFF POINTER
1286 18F7 JMPR ..REPT ;GO FOR ANOTHER
    
```

\*\*\*\*\* (CHECK STATUS) \*\*\*\*\*

```

1288 E69D ..NMI: ANI DM.RER ;TEST FOR ERRORS.
128A 32 1377 STA CB.STS ;SAVE READ STATUS.
128D C8 RZ ;RETURN COMPLETE.
128E CD 12DC CALL CHK.RT ;CHECK ABOUT RETRYS.
1291 28DC JRZ ..RTRY ;PERFORM RETRY.
1293 C9 RET ;ERROR RETURN.
    
```

\*\*\*\*\*

```

;*****
; WR.SEC IS THE SUBROUTINE THAT INTERACTS WITH THE .#
; 179X-01 DURING WRITE SETOR OPERATIONS. THIS SECTION*
; INITIATES THE DISK TRANSFER, SERVICES THE CONTROLLER*
; CHIP, AND TERMINATES THE OPERATION. ERROR DETECTION*
; IS IMPLEMENTED. *
;*****
    
```

\*\*\*\*\* (INITIALIZE WRITE OPERATION) \*\*\*\*\*

```

1294 AF WR.SEC: XRA A ;ZERO REGISTER.
1295 32 130B STA ERR.CT ;SET ERROR COUNTER.
1298 3A 1373 LDA CB.SEC ;LOAD SECTOR NMBR.
129B A9 XRA C ;INVERT (1791-01).
129C D306 OUT WD.SEC ;SET SECTOR REGISTER.
129E FD21 12B7 ..RTRY: LXI Y,..NMI ;SET NMI RETURN.
12A2 2A 1309 132F LHLD BUF.ST ;BUFFER START.
12A5 3EAD MVI A,DC.WRS ;LOAD WRITE SECTOR CMD.
12A7 A9 XRA C ;INVERT (1791-01).
12A8 D304 OUT WD.CMD ;ISSUE COMMAND.
12AA E3 PAUSE ;ALLOW 1791 TO SETTLE
    
```

\*\*\*\*\* (DATA TRANSFER LOOP) \*\*\*\*\*

```

12AE DB80 ..REPT: IN XP.DSH ;HOLD FOR DATA REQ.
12B0 7E MOV A,M ;GET DATA BYTE.
12B1 A9 XRA C ;INVERT (1791-01).
12B2 D307 OUT WD.DTA ;OUTPUT DATA BYTE.
12B4 23 INX H ;INCREMENT BUFF POINTER
12B5 10F7 JMPR ..REPT ;REPEAT SEQUECE
    
```

\*\*\*\*\* (CHECK STATUS) \*\*\*\*\*

```

12B7 E6FD ..NMI: ANI DM.WER ;TEST FOR WRITE ERRORS.
12B9 32 1377 STA CB.STS ;STORE WRITE STATUS.
12BC C8 RZ ;RETURN COMPLETE.
12BD CD 12DC CALL CHK.RT ;CHECK ABOUT RETRYS.
12C0 28DC JRZ ..RTRY ;PERFORM RETRY.
12C2 C9 RET ;ERROR RETURN.
    
```

\*\*\*\*\*

```
*****  
; WR.TRK IS THE SUBROUTINE WHICH INITIATES A FORMAT *  
; TRACK COMMAND (WRITE-TRACK 179X-01 TYPE 3). THE *  
; FORMATTING BYTE STREAM IS PROVIDED BY A PROGRAM *  
; WHICH MUST BE PRESENT IN THE FORMAT BUFFER. *  
*****
```

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

```
1203 FD21 1203 WR.TRK: LXI Y,..NMI ;LOAD NMI VECTOR.  
1207 3EFO MVI A,DC.WRT ;WRITE TRACK CMND.  
1209 A9 XRA C ;INVERT (1791-01).  
120A D304 OUT WD.CMD ;ISSUE COMMAND.  
120C E3 PAUSE ;ALLOW 179X TO SETTLE.  
120E C3 1708 JMP FMT.PS ;FORMAT PROG START.
```

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

```
12D3 E6E4 ..NMI: ANI DM.FER ;TEST FOR ERRORS.  
12D5 32 1377 STA CB.STS ;STORE FORMAT STATUS.  
12D8 22 137A SHLD CW.LNG ;DISPLAY TRAIL BYTES.  
12DB 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) \*\*\*\*\*

```

12BC    E680    CHK.RT: ANI        DM.DNR        ;TEST NOT READY BIT.
12DE    2028    JRNZ        ..EXIT        ;CAN NOT RECOVER.
12E0    3A 1376 LDA        CB.MOD        ;GET COMMAND MODE.
12E3    E680    ANI        CM.NRT        ;NO RETRYS CHECK.
12E5    2021    JRNZ        ..EXIT        ;SHOULD NOT RECOVER.
12E7    DB40    IN          XP.MTX        ;MOTOR TIME EXTEND.
    
```

\*\*\*\*\* (RECORD RETRY) \*\*\*\*\*

```

12E9    3A 130B LDA        ERR.CT        ;GET ERROR COUNT.
12EC    3C      INR        A            ;INCREMENT.
12ED    32 130B STA        ERR.CT        ;STORE NEW COUNT.
12F0    FE05    CPI        RTY.SK        ;SHOULD TRY SEEK?
12F2    2008    JRNZ        ..CKLS        ;IF NOT, CHECK LAST.
    
```

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

```

12F4    CD 1126 CALL       HOME.D        ;HOME SELECTED DRIVE.
12F7    200F    JRNZ        ..EXIT        ;ERROR EXIT.
12F9    CD 1180 CALL       SEEK          ;SEEK DESIRED TRACK.
    
```

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

```

12FC    FE09    ..CKLS: CPI        RTY.LS        ;WAS THIS THE LAST.
12FE    2807    JRZ          ..STNZ        ;ERROR LAST RETRY.
1300    ED5B 1316 LDED       TM.PLD        ;PHASE LOCK DELAY.
1304    CF      WAIT        ;PROGRAMMABLE DELAY.
1305    AF      XRA        A            ;CLEAR FOR RETRY.
1306    C9      RET         ;TRY AGAIN EXIT.
    
```

\*\*\*\*\* (ERROR EXIT) \*\*\*\*\*

```

1307    3C      ..STNZ: INR        A            ;SET NOT ZERO.
1308    C9      ..EXIT: RET         ;ERROR EXIT.
    
```

\*\*\*\*\*

\*\*\*\*\* (BUFFER POINTERS) \*\*\*\*\*

1309 1380 BUF.ST: .WORD BUF.BG ;BUFFER START ADDRESS

\*\*\*\*\*

130B 00 ERR.CT: .BYTE 0 ;RETRY ERROR COUNTER.

\*\*\*\*\* (HOLDING AREAS) \*\*\*\*\*

130C 00 SV.DRV: .BYTE 0 ;BL.CTL DRIVE BITS.

130D 00 SV.STS: .BYTE 000H ;1791 STATUS SAVE

\*\*\*\*\* (TIMING VALUES, 0.1 MS) \*\*\*\*\*

130E 015E TM.HLD: .WORD 350 ;HEAD ENGAGE TIME.

1310 0050 TM.STP: .WORD 80 ;STEPPER INTERVAL.

1312 0050 TM.ALS: .WORD 80 ;AFTER LAST STEP.

1314 0001 TM.MTD: .WORD 1 ;MOTOR START UP.

1316 04B0 TM.PLD: .WORD 1200 ;PHASE LOCK RECOVERY.

1318 000A TM.SAW: .WORD 10 ;STEP AFTER WRITING.

; \* ENTRY \* SA 800 FDD100 SA 850

; 8-C

; TM.HLD: .WORD 350 250 350

; TM.STP: .WORD 80 60 30

; TM.ALS: .WORD 80 80 120

; TM.MTD: .WORD 1 1 1

; TM.PLD: .WORD 1200 1200 1200

; TM.SAW: .WORD 10 10 10

\*\*\*\*\* (DISKETTE FORMAT LABEL) \*\*\*\*\*

131A 4A&164&52044 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.

13A0 ID.SPT == ID.BLK+0000H ;SECTORS/TRACK.

13B0 ID.STG == ID.BLK+0010H ;SECTOR STAGGER.

13B1 ID.FLG == ID.BLK+0011H ;DISKETTE FLAGS.

0000 ID.FLD == 00000000B ;3740 FLAGS.

\*\*\*\*\*



\*\*\*\*\* (DRIVE TABLE ENTRIES) \*\*\*\*\*

0000 DV.NBR == 0 ;CURRENT DRIVE NUMBER.  
0001 DV.TRK == 1 ;CURRENT TRACK NUMBER.  
0002 DV.SEC == 2 ;CURRENT SECTOR NUMBER  
0003 DV.SPT == 3 ;SECTORS PER TRACK.  
0004 DV.STG == 4 ;SECTOR STAGGER NUMBER.  
0005 DV.FLG == 5 ;SIDE AND DENSITY FLAGS  
0006 DV.CTL == 6 ;LAST CONTROLS USED.

\*\*\*\*\* (DRIVE TABLE AREA) \*\*\*\*\*

1322 DV.TBL == . ;DRIVE TABLE ADDRESS.  
  
1322 00FF00000002 DT.DE0: .BYTE 0,255,0,0,0,DF.DFL,0C4H ;DRIVE 0.  
1329 01FF00000002 DT.DE1: .BYTE 1,255,0,0,0,DF.DFL,0C5H ;DRIVE 1.  
1330 02FF00000002 .BYTE 2,255,0,0,0,DF.DFL,0C6H ;DRIVE 2.  
1337 03FF00000002 .BYTE 3,255,0,0,0,DF.DFL,0C7H ;DRIVE 3.  
133E 04FF00000000 DT.DE4: .BYTE 4,255,0,0,0,0,0 ;DUMMY.  
  
0007 DV.DES == DT.DE1-DT.DE0 ;ENTRY SIZE.

\*\*\*\*\* (FLAG BIT DEFINITIONS) \*\*\*\*\*

0001 DF.T1D == 1 ;TRACK 1 DENSITY (1 = DOUBLE).  
0002 DF.DTD == 2 ;DATA TRACKS DENSITY (1 = DD).  
0003 DF.TSD == 3 ;TWO SIDED ( 1 = TWO SIDES).  
0002 DF.DFL == 1<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      ;TRACK NUMBER.  
1373    00      CB.SEC: .BYTE  0      ;SECTOR NUMBER.  
1374    00      CB.PFG: .BYTE  0      ;FORMAT FLAGS.  
1375    00      CB.SPR: .BYTE  0      ;UNASSIGNED.  
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.BT1 == 0000001B  ;NOT ASSIGNED.  
0001      CS.BT0 == 0000000B  ;NOT ASSIGNED.
```

```
*****
```

## DISK CONTROLLER MODULE (DCM)

## BOOTSTRAPED INITIALIZE

```

;*****
; 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          LXI      D,BANK.0        ;SET DESTINATION.
1386          LXI      H,BANK.1        ;SET SOURCE ADDR.
1389      ED80          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 FNTR.
1391      ED56          IMI           ;INTERRUPT MODE 1.

```

```

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

```

```

1393      OE00          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      OEFF          MVI      C,OFFH      ;SW CLOSED - 1791.

```

```

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

```

```

139D      DD21 133E      LD.BLT: LXI      X,DT.DED     ;INIT DRIVE TBL.
13A1          MVI      A,2          ;BIOS LOADER SECTOR.
13A3          STA      CB,SEC       ;SET SECTOR VALUE.
13A6          IN       XP,MTX       ;MOTOR TIME EXTEND.
13A8          LXI      H,BUF,BG     ;SET RETURN ADDR.
13AB          PUSH     H            ;PUSH INTO STACK.
13AC          JMP      RD,SEC       ;GET BIOS LOADER.

```

```

;*****
.END

```

## DISK CONTROLLER MODULE (DCM)

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

BANK.0	1000	BANK.1	1400	BANK.L	0400	BASE	1000
BC.DAS	0020	BC.LBE	0010	BC.DDS	0010	BC.DSA	0001
BC.DSB	0002	BC.DSE	0004	BC.DSN	0003	BC.EIA	0008
BC.FCA	0040	BC.FCB	0080	BC.FCH	0040	BC.FCL	00C0
BC.FCM	0080	BC.FCZ	0000	BC.SDS	0000	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.USO	0001
BS.US1	0002	BUF.BG	1380	BUF.ST	1309	CB.CMD	1370
CB.DRV	1371	CB.FFG	1374	CB.MOD	1376	CB.SEC	1373
CB.SPR	1375	CB.STS	1377	CB.TRK	1372	CHK.RT	12DC
CMD.BK	1370	CM.NRT	0080	CS.BTO	0001	CS.BT1	0002
CS.BT5	0020	CS.CRC	0008	CS.DNR	0080	CS.LDE	0004
CS.RNF	0010	CS.WRF	0040	CW.LAD	1378	CW.LNG	137A
DC.HDL	0010	DC.HDU	0018	DC.IFI	0008	DC.RDA	00C0
DC.RDS	0080	DC.STS	00D0	DC.WRS	00A0	DC.WRT	00F0
DF.D-L	0002	DF.DTD	0002	DF.T1D	0001	DF.TSD	0003
DM.LNR	0080	DM.FER	00E4	DM.HDL	0020	DM.LBE	0004
DM.RER	009D	DM.TK0	0004	DM.WER	00FD	DT.DEO	1322
DT.DEL	1329	DT.DED	133E	DV.CTL	0006	DV.DES	0007
DV.FLG	0005	DV.NBR	0000	DV.SEC	0002	DV.SPT	0003
DV.STG	0004	DV.TBL	1322	DV.TRK	0001	ERR.CT	130B
EX.HDL	123E	EX.HDU	124B	EX.STS	1258	FETCH	103F
FMT.BG	1700	FMT.PS	1708	HOME.D	1126	HR.INT	1038
HR.VEC	1006	ID.BLK	13A0	ID.FLD	0000	ID.FLG	13B1
ID.LBL	1380	ID.SPT	13A0	ID.STG	13B0	ID.SZE	0008
INIT.B	1380	IO.BLK	1370	JADEID	131A	LD.BLT	139D
LOG.ON	114B	NM.INT	1066	RD.SEC	1265	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	1180	SELECT	10DD	SV.DRV	130C
SV.STS	130D	TICK.E	1074	TICK.R	1070	TMR.FC	0019
TMR.NC	001C	TM.ALS	1312	TM.HLD	130E	TM.MTQ	1314
TM.FLD	1316	TM.SAW	1318	TM.STP	1310	TP.STK	1370
WD.CMD	0004	WD.DTA	0007	WD.INT	1066	WD.SEC	0006
WD.STS	0004	WD.TRK	0005	WR.SEC	1294	WR.TRK	12C3
XP.DSH	0080	XP.IRR	0020	XP.MTQ	0010	XP.MTX	0040
XP.STP	0008	X.CUTE	1041	\$.ADDR	10CC	\$.FORM	107A
\$.IDLE	10D2	\$.LGON	10AF	\$.READ	107C	\$.SPAR	10CF
\$.WRIT	108B						

```
*****  
;  
; PROGRAM ID: BIOS - CPM 2.2  
;  
*****  
;  
; PROPERTY OF: JADE COMPUTER PRODUCTS  
; 4901 W. ROSECRANS BLVD.  
; HAWTHORNE, CALIFORNIA  
; 90250, U.S.A.  
;  
*****  
;  
; VERSION: 2.2  
;  
*****  
; THIS BIOS CONTAINS DISK SUBROUTINES WHICH ARE  
; COMPATABLE WITH THE DCM-2.2 MODULE AND THE JADE  
; DOUBLE D DISK CONTROLLER. THIS COMBINATION WILL  
; INTERFACE UP TO FOUR 5" DRIVES IN SINGLE AND DOUBLE  
; DENSITY. THE CONSOLE, READER/PUNCH, AND PRINTER  
; INTERFACES SHOULD BE ALTERED FOR SPECIFIC END USER  
; SYSTEMS AS NEEDED.  
***** SK **
```

BIOS - JADE DOUBLE D CP/M 2.2

EQUATES

```

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

```

```

0014      NK.SYS  ==      20          ;SYSTEM SIZE IN K BYTES.
0400      K.BYTE  ==     1024        ;1K BYTE SIZE.
5000      CPM.SZ  ==     NK.SYS*K.BYTE ;TOP SYSTEM ADDRESS.
6000      CPM.BS  ==     CPM.SZ-(20*K.BYTE) ;CP/M BIAS VALUE.
3400      CCP     ==     CPM.BS+3400H ;ADDRESS OF CCP.
3C00      BDOS   ==     CPM.BS+3C00H ;ADDRESS OF BDOS.
4A00      BIOS   ==     CPM.BS+4A00H ;ADDRESS OF BIOS
6600      BIOS.R ==     1000H-BIOS   ;DDT OFFSET 1000H LOAD.
0100      TPA    ==     0100H       ;ADDRESS OF TPA.
00FF      IOBYTE ==     0FFH        ;INITIAL IOBYTE VALUE.
0000      DF.DRV ==     0           ;INITIAL DEFAULT DRV.
0100      SEC.SZ ==     0080H       ;BYTES PER SECTOR.
0100      FMT.SZ ==     0100H       ;FORMAT BUFF SIZE.
0000      N.DRVS ==     2           ;1-4 DRIVES.

```

```

;*****
; DOUBLE D HARDWARE PARAMETERS. PLEASE NOTE THIS
; SECTION CONTAINS CONDITIONAL STATEMENTS.
;*****

```

```

0043      D.PORT ==     043H        ;DOUBLE D PORT ADDRESS.
0001      TRUE   ==     1           ;TRUE IS A ONE.
0000      FALSE  ==     0           ;FALSE IS A ZERO.
0001      REV.B  ==     TRUE        ;SET TRUE FOR REV B BOARDS.
0000      REV.C  ==     FALSE       ;SET TRUE FOR REV C BOARDS.
          .IFG   REV.B, [
0002      DS.HLT ==     002H        ;STATUS PORT HALT INDICATOR]
          .IFG   REV.C, [
          DS.HLT ==     001H        ;STATUS PORT HALT INDICATOR]

```

```

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

```

```

0001      DC.SIN ==     00000001B   ;SWITCH DD BANK 0 INTO SYSTEM.
0001      DC.MB0 ==     00000001B   ;SELECT DOUBLE D BANK 0.
0003      DC.MB1 ==     00000011B   ;SELECT DOUBLE D BANK 1.
0000      DC.SOT ==     00000000B   ;SWITCH DD MEM OUT OF SYSTEM.
0002      DC.INT ==     00000010B   ;ISSUE DD Z80A INTERRUPT.

```

```

;*****

```

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

```
***** ( COMMAND BLOCK DEFINED )*****
```

```
0370 DD.CBT == 0370H ;COMMAND BYTE (BANK 0).  
0371 DD.DRV == 0371H ;DRIVE NUMBER (BANK 0).  
0372 DD.TRK == 0372H ;TRACK NUMBER (BANK 0).  
0373 DD.SEC == 0373H ;SECTOR NUMBER (BANK 0).  
0377 DD.STS == 0377H ;CMMD STATUS (BANK 0).  
0380 DD.BUF == 0380H ;SECTOR BUFFER (BANK 0).  
0300 DD.FBF == 0300H ;FORMAT BUFFER (BANK 1).  
03A0 DD.DPB == 03A0H ;ID SEC DPB (BANK 0).  
03B1 DD.DFB == 03B1H ;ID SEC FLAGS (BANK 0).
```

```
***** ( DCM COMMANDS )*****
```

```
0000 DC.LOG == 000H ;LOG ON DISKETTE.  
0001 DC.RDS == 001H ;READ SECTOR.  
0002 DC.WRS == 002H ;WRITE SECTOR.  
0003 DC.FMT == 003H ;FORMAT TRACK.
```

```
*****  
; CONSOLE DEFINITIONS *  
*****
```

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

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

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

```
4A00 .FABS  
.PHEX  
.XLINK  
.LUC BIOS
```

```
*****
```

```

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

```

4A00    C3 4A83    JMP     'INIT           ;COLD START ENTRY
4A03    C3 4C8C    JMP     WARM           ;RELOAD CCP/BDOS
4A06    C3 4C80    JMP     CNS.CK        ;GET CONSOLE STATUS
4A09    C3 4C8A    JMP     CNS.IN        ;CONSOLE INPUT
4A0C    C3 4C75    JMP     CNS.OT        ;CONSOLE OUTPUT
4A0F    C3 3C88    JMP     LIST          ;PRINTER OUTPUT
4A12    C3 4C17    JMP     PUNCH         ;PUNCH OUTPUT
4A15    C3 4C84    JMP     READER        ;READER INPUT
4A18    C3 4C3C    JMP     HOME          ;HOME SELECTED DRIVE
4A1B    C3 4C41    JMP     SELDSK        ;SELECT DISK DRIVE
4A1E    C3 4C60    JMP     SETTRK        ;SET TRACK NUMBER
4A21    C3 4C55    JMP     SETSEC        ;SET SECTOR NUMBER
4A24    C3 4C5A    JMP     SETDMA        ;SET TRANSFER ADDRESS
4A27    C3 4B03    JMP     DISKRD        ;PERFORM DISK READ
4A2A    C3 4B23    JMP     DISKWR        ;PERFORM DISK WRITE
4A2D    C3 4CAB    JMP     LIST$T        ;RETURN LIST STAT
4A30    C3 4C70    JMP     SECTRN        ;TRANSLATE SECTOR
4A33    C3 4B50    JMP     FORMAT        ;FORMAT A TRACK
    
```

```

;*****
; RESERVE DRIVE PARAMETER BLOCKS
;*****
    
```

```

000F    DFB.SZ ==      15           ;SIZE IS 15 BYTES.
4A36    D0.DPB: .BLKB  DFB.SZ     ;RESERVES 15 BYTES.
4A43    D1.DPB: .BLKB  DFB.SZ     ;FOR EACH DISK DRIVE.
4A54    D2.DPB: .BLKB  DFB.SZ     ;DRIVE A THRU DRIVE D.
4A63    D3.DPB: .BLKB  DFB.SZ
    
```

```

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

```

4A72    00        BT.CMD: .BYTE  0           ;DCM COMMAND.
4A73    00        BT.DRV: .BYTE  0           ;DRIVE NUMBER.
4A74    00        BT.TRK: .BYTE  0           ;TRACK NUMBER.
4A75    00        BT.SEC: .BYTE  0           ;SECTOR NUMBER.
4A76    00        BT.SP0: .BYTE  0           ;SPARE BYTE 0.
4A77    00        BT.SP1: .BYTE  0           ;SPARE BYTE 1.
4A78    00        BT.MOD: .BYTE  0000000B    ;MODE CONTROLS.
4A79    00        BT.STS: .BYTE  0           ;COMMAND STATUS.
4A7A    0000      BT.LAD: .WORD  0           ;LOAD ADDRESS.
4A7C    0000      BT.LNG: .WORD  0           ;LOAD LENGTH.

4A7E    0000      BT.DMA: .WORD  0           ;SYSTEM TRANSFER ADDR.
4A80    0000      DT.PTR: .WORD  0           ;DRIVE TABLE POINTER.
4A82    00        LOG.RQ: .BYTE  0           ;LOG ON REQUEST REG.
    
```

```

;*****
    
```



```

;*****
; INIT - COLD START ENTRY ** DIRECTORY BUFFER OVERLAY **
;*****

4A83      DIR.BF  ==      .      ;BUFFER BEGINNING.

;*****( GET SYSTEM K SIZE IN ASCII )*****

4A83      31 0080      INIT:   LXI      SP,0080H      ;SET UP STACK AREA.
4A86      21 5000      LXI      H,CPM.SZ      ;GET CPM SYS SIZE.
4A89      7C          MOV      A,H          ;GET NMBR OF PAGES.
4A8A      0F          RRC          ; (/2).
4A8B      0F          RRC          ;CPM SIZE IN K.
4A8C      0600      MVI      B,0          ;ZERO TENS COUNTER.
4A8E      FE0A      C.TENS:  CPI      10      ;COMPARE AGAINST 10.
4A90      DA 4A99      JC       C.ONES      ;MUST BE LESS THAN 10.
4A93      D60A      SUI      10          ;SUBTRACT 10.
4A95      04          INR      B          ;INC TENS COUNTER.
4A96      03 4A8E      JMP      C.TENS      ;COUNT TENS.
4A99      C630      C.ONES:  ADI      '0'      ;ADD ASCII 0 TO REM.
4A9B      32 4AED      STA      SYS.OD      ;STORE ONES DIGIT.
4A9E      78          MOV      A,B          ;GET TENS COUNTER.
4A9F      C630      ADI      '0'      ;ADD ASCII 0 TO TENS.
4AA1      32 4AEC      STA      SYS.TD      ;STORE TENS DIGIT.

;*****( SYSTEM INITIALIZATION )*****

4AA4      03 4AC0      JMP      C.LOGO      ;JUMP OVER AREA.
4AA7      .BLKB      25      ;RESERVE INIT AREA.

;*****( SEND SYSTEM SIGN ON )*****

4AC0      21 4ACF      C.LOGO:  LXI      H,MSG.SO      ;SIGN ON MSG ADDR.
4AC3      CD 4CAE      CALL     MSG.OT      ;ISSUE MESSAGE.
4AC6      2A 4A04      LHL     BIOS+4      ;WARM START ADDRESS.
4AC9      22 4A01      SHLD   BIOS+1      ;RESET COLD START.
4ACC      03 4CBC      JMP      WARM       ;PERFORM WARM BOOT.

4ACF      0D0A0D0A      MSG.SO:  .ASCII  [CR][LF][CR][LF]
4AD3      4A4144452043 .ASCII  'JADE COMPUTER PRODUCTS '[CR][LF]
4AEC      30          SYS.TD:  .BYTE  '0'          ;TENS DIGIT.
4AED      30          SYS.OD:  .BYTE  '0'          ;ONES DIGIT.
4AEE      4B2043502F4D .ASCII  'K CP/M VERS 2.2 '[CR][LF][CR][LF]

;*****( FILL OUT DIRECTORY BUFFER )*****

4B03      .IFG      (DIR.BF+SEC.SZ-.), [
          .LOC      DIR.BF+SEC.SZ      ;FILL OUT SECTOR SIZE]

;*****

```

```

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

```

```

4B03      3E01      DISKRD: MVI      A,DC.SIN      ;SWITCH DD INTO SYSTEM.
4B05      D343      OUT        D,PORT          ;ISSUE DD COMMAND.
4B07      3E01      MVI        A,DC.RDS        ;READ SECTOR COMMAND.
4B09      CD 4B78  CALL       DSK.EX          ;PERFORM OPERATION.
4B0C      C2 4B49  JNZ        DSK.ER        ;ERROR EXIT.
4B0F      2A 4A7E  LHLD       BT.DMA        ;LOAD USER BUF ADDRESS
4B12      EB      XCHG       ;MOVE HL TO DE.
4B13      01 0380  LXI        B,DD.BUF      ;LOAD BUFFER OFFSET.
4B16      2A 0040  LHLD       D.ADDR        ;LOAD DD WINDOW ADDR.
4B19      09      DAD        B          ;HL NOW SECTOR BUFFER.
4B1A      01 0080  LXI        B,SEC.SZ      ;LOAD SECTOR SIZE.
4B1D      CD 4BA7  CALL       BLOCK         ;BLOCK MOVE ROUTINE.
4B20      C3 4B43  JMP        DSK.OK        ;NORMAL RETURN.

```

```

;*****
; WRITE A DISK SECTOR ROUTINE
;*****

```

```

4B23      3E01      DISKWR: MVI      A,DC.SIN      ;SWITCH DD INTO SYSTEM.
4B25      D343      OUT        D,PORT          ;ISSUE HARDWARE CMND.
4B27      01 0080  LXI        B,SEC.SZ      ;LOAD SECTOR SIZE.
4B2A      2A 0040  LHLD       D.ADDR        ;DD SYSTEM ADDRESS.
4B2D      11 0380  LXI        D,DD.BUF      ;DD BUFFER OFFSET.
4B30      19      DAD        D          ;HL NOW DD BUF ADDR.
4B31      EB      XCHG       ;DE NOW DD BUF ADDR.
4B32      2A 4A7E  LHLD       BT.DMA        ;HL NOW USER BUF ADDR.
4B35      CD 4BA7  CALL       BLOCK         ;BLOCK MOVE ROUTINE.
4B38      3E02      MVI        A,DC.WRS      ;LOAD WRITE SEC CMND.
4B3A      CD 4B78  CALL       DSK.EX        ;CALL DISK EXECUTIVE.
4B3D      CA 4B43  JZ         DSK.OK        ;JUMP IF WRITE OK.
4B40      C3 4B49  JMP        DSK.ER        ;ERROR EXIT.

```

```

;*****
; DISK READ/WRITE/FORMAT EXITS
;*****

```

```

4B43      3E00      DSK.OK: MVI      A,DC.SOT      ;SWITCH DD OUT OF SYS.
4B45      D343      OUT        D,PORT          ;ISSUE HARDWARE CMND.
4B47      AF      XRA        A          ;ZERO A REGISTER.
4B48      C9      RET          ;ERROR EXIT.

```

```

4B49      3E00      DSK.ER: MVI      A,DC.SOT      ;SWITCH DD OUT OF SYS.
4B4B      D343      OUT        D,PORT          ;ISSUE HARDWARE CMND.
4B4D      3EFF      MVI        A,OFFH        ;LOAD ERROR FLAGS.
4B4F      C9      RET          ;ERROR EXIT.

```

```

;*****

```

```

;*****
; FORMAT A DISK TRACK ROUTINE
;*****
  
```

```

4B50 3E01 FORMAT: MVI A,DC.SIN ;SWITCH DD INTO SYSTEM.
4B52 D343 OUT D,PORT ;ISSUE HARDWARE CMND.
4B54 3E03 MVI A,DC.MB1 ;SELECT DD BANK 1.
4B56 D343 OUT D,PORT ;ISSUE HARDWARE CMND.
4B58 01 0100 LXI B,FMT.SZ ;FORMAT PROG SIZE.
4B5A 2A 0040 LHLD D,ADDR ;DD SYSTEM ADDRESS.
4B5C 11 0300 LXI D,DD.FBF ;DD FORMAT BUF OFFSET.
4B5E 19 DAD D ;HL NOW DD FBUF ADDR.
4B60 EB XCHG ;DE NOW DD FBUF ADDR.
4B62 2A 4A7E LHLD BT.DMA ;FORMAT PROGRAM ADDR.
4B64 CD 4BA7 CALL BLOCK ;BLOCK MOVE ROUTINE.
4B66 3E01 MVI A,DC.MB0 ;RESELECT DD BANK 0.
4B68 D343 OUT D,PORT ;ISSUE TO DD HARDWARE.
4B6A 3E03 MVI A,DC.FMT ;LOAD FORMAT TRK CMND.
4B6C CD 4B78 CALL DSK.EX ;CALL DISK EXECUTIVE.
4B6E CA 4B43 JZ DSK.OK ;JUMP IF WRITE OK.
4B70 C3 4B49 JMP DSK.ER ;ERROR EXIT.
  
```

```

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

```

4B78 32 4A72 DSK.EX: STA BT.CMD ;STORE DCM COMMAND.
4B7A 01 0007 LXI B,7 ;NMBR BYTE TO MOVE.
4B7C 11 0370 LXI D,DD.CBT ;COMMAND BYTE OFFSET.
4B7E 2A 0040 LHLD D,ADDR ;DD SYS ADDRESS.
4B80 19 DAD D ;HL NOW PTS CMND BLK.
4B82 EB XCHG ;NOW ADDR IN DE.
4B84 21 4A72 LXI H,BT.CMD ;BIOS CMND BLOCK.
4B86 CD 4BA7 CALL BLOCK ;PERFORM BLOCK MOVE.
4B88 3E02 MVI A,DC.INT ;LOAD DD INTERRUPT.
4B8A D343 OUT D,PORT ;ISSUE DD INTERRUPT.
4B8C EB XCHG ;EXCHANGE SRC/DSTN.
4B8E 01 0005 LXI B,5 ;5 RETURN BYTES.
4B90 DB43 DSK.WT: IN D,PORT ;READ DD STATUS.
4B92 E602 ANI DS.HLT ;TEST HALT* FLAG.
4B94 C2 4B94 JNZ DSK.WT ;TEST UNTIL HALTED.
4B96 3E01 MVI A,DC.SIN ;SWITCH DD INTO SYS.
4B98 D343 OUT D,PORT ;ISSUE HARDWARE CMND.
4B9A CD 4BA7 CALL BLOCK ;GET DD STATUS.
4B9C 3A 4A79 LDA BT.STS ;LOAD STATUS BYTE.
4B9E A7 ANA A ;TEST FOR ERRORS.
4BA0 C9 RET ;RETURN TO CALLER.
  
```

```

;*****
  
```

\*\*\*\*\*  
 ; BLOCK - BLOCK MOVE (Z80 LDIR REGISTER USAGE) \*  
 \*\*\*\*\*

4BA7	7E	BLOCK: MOV	A,M	;GET BYTE.
4BA8	12	STAX	D	;STORE BYTE.
4BA9	23	INX	H	;INC SOURCE PNTR
4BAA	13	INX	D	;INC DESTINATION PNTR.
4BAB	0B	DCX	B	;DECREMENT COUNT.
4BAC	78	MOV	A,B	;GET HIGH COUNT.
4BAD	B1	ORA	C	;OR IN LO COUNT.
4BAE	C2 4BA7	JNZ	BLOCK	;ANOTHER BYTE.
4BB1	C9	RET		;RETURN TO USER.

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

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

4BB2	3A 4A82	LOG.ON: LDA	LOG.RQ	;CHECK LOG REQUEST.
4BB5	E601	ANI	001H	;LOG ON BIT TEST.
4BB7	C2 4B43	JNZ	DSK.OK	;RETURN, NO LOG-ON.

\*\*\*\*\* ( READ IDENTITY SECTOR )\*\*\*\*\*

4BBA	22 4A80	SHLD	DT.PTR	;STORE DRV TBL PNTR.
4BBD	3E01	MVI	A,DC.SIN	;SWITCH DD INTO SYS.
4BBF	D343	OUT	D.PORT	;ISSUE HARDWARE CMND.
4BC1	3E00	MVI	A,DC.LOG	;LOAD DCM LOG-ON CMND.
4BC3	CD 4B78	CALL	DSK.EX	;PERFORM DISK OP.
4BC6	CA 4BCF	JZ	LOG.OK	;GO TO LOGON ERROR.
4BC9	21 0000	LXI	H,0	;ERROR, BAD LOG ON.
4BCC	C3 4B49	JMP	DSK.ER	;BIOS EXIT.

\*\*\*\*\* ( CHECK FOR JADE ID )\*\*\*\*\*

4BCF	11 0380	LOG.OK: LXI	D,DD.BUF	;DD BUFFER OFFSET.
4BD2	2A 0040	LHLD	D.ADDR	;DD SYS ADDRESS.
4BD5	19	DAD	D	;HL NOW PNTS BUFFER.
4BD6	11 4D4F	LXI	D,JADEID	;DE PNTS BIOS ID.
4BD9	0608	MVI	B,ID.SZE	;SET LABEL SIZE.
4BDB	1A	LOG.ID: LDAX	D	;GET LABEL CHARACTER.
4BDC	BE	CMP	M	;DOES ID SECTOR MATCH.
4BDD	C2 4C0A	JNZ	LG3740	;ASSUME DISKETTE 3740.
4BE0	05	DCR	B	;DECREMENT COUNT.
4BE1	C2 4BDB	JNZ	LOG.ID	;CHECK IF ANOTHER CHR.

\*\*\*\*\* ( DISKETTE CONTAINS ID )\*\*\*\*\*

4BE4	CD 4C29	CALL	TRNONE	;ASSUME DDENS.
4BE7	CD 4C31	CALL	DPB.ADD	;GET DPB ADDR IN DE.
4BEA	01 03A0	LXI	B,DD.DPB	;DPB ADDR OFFSET.
4BED	2A 0040	LHLD	D.ADDR	;DD SYSTEM ADDRESS.
4BF0	09	DAD	B	;HL NOW AT ID DPB.

```

4BF1 01 000F LXI B,DPB.SZ ;DPB SIZE IN BYTES.
4BF4 CD 4BA7 CALL BLOCK ;MOVE INTO DPB.
4BF7 11 03B1 LXI D,DD.DDF ;ID DTA DNS OFFSET.
4BFA 2A 0040 LHLD D.ADDR ;DD SYSTEM ADDR.
4bfd 19 DAD D ;HL POINTS FLAGS.
4BFE 7E MOV A,M ;LOAD FLAGS.
4BFF E604 ANI 04H ;TEST DATA DENSITY.
4C01 CD 4C1F CZ TR3740 ;IF 0 USE 3740 TRN.
4C04 2A 4A80 LHLD DT.PTR ;RELOAD POINTER.
4C07 C3 4B43 JMP DSK.OK ;EXIT BIOS JUMP.

```

\*\*\*\*\* ( ASSUME 3740 DISKETTE )\*\*\*\*\*

```

4C0A CD 4C1F LG3740: CALL TR3740 ;SET SECTOR TRANSLATE.
4C0D CD 4C31 CALL DPB.AD ;SET REGISTER DE.
4C10 01 000F LXI B,DPB.SZ ;DPB SIZE IN BYTES.
4C13 21 4D2C LXI H,SD.PBK ;ADDRESS OF BLK IMAGE.
4C16 CD 4BA7 CALL BLOCK ;MOVE INTO DPB.
4C19 2A 4A80 LHLD DT.PTR ;RELOAD POINTER.
4C1C C3 4B43 JMP DSK.OK ;EXIT BIOS JUMP.

```

\*\*\*\*\* ( SET 3740 SECTOR TRANSLATION )\*\*\*\*\*

```

4C1F 11 4D12 TR3740: LXI D,SDTRAN ;SECTOR TRAN TBL ADDR.
4C22 2A 4A80 LHLD DT.PTR ;ADDR DISK PARA HDR.
4C25 73 MOV M,E ;LOW ORDER ADDR.
4C26 23 INX H ;POINT NEXT BYTE.
4C27 72 MOV M,D ;HIGH ORDER ADDR.
4C28 C9 RET ;RETURN TO LOG USER.

```

\*\*\*\*\* ( SET NO SECTOR TRANSLATION )\*\*\*\*\*

```

4C29 AF TRNONE: XRA A ;ZERO A REGISTER.
4C2A 2A 4A80 LHLD DT.PTR ;ADDR OF PARA HDR.
4C2D 77 MOV M,A ;ZERO LOW ORDER ADDR.
4C2E 23 INX H ;NEXT BYTE.
4C2F 77 MOV M,A ;ZERO HIGH BYTE.
4C30 C9 RET ;RETURN TO LOG USER.

```

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

```

4C31 2A 4A80 DPB.AD: LHLD DT.PTR ;ADDR DISK PARA HDR.
4C34 11 000A LXI D,10 ;DPB TBL FNTR OFFSET.
4C37 19 DAD D ;NOW AT DPB FNTR.
4C38 5E MOV E,M ;LOW ORDER ADDR.
4C39 23 INX H ;NEXT BYTE.
4C3A 56 MOV D,M ;HIGH ORDER ADDR.
4C3B C9 RET ;RETURN TO LOG USER.

```

\*\*\*\*\*

## BIOS - JADE DOUBLE D CP/M 2.2

## DISK PARAMETER SUBROUTINES

\*\*\*\*\* ( HOME DRIVE )\*\*\*\*\*

```
4C3C 0E00 HOME: MVI C,0 ;C REGISTER TO ZERO.
4C3E 03 4C60 JMP SETTRK ;PERFORM SET TRACK.
```

\*\*\*\*\* ( SELECT DRIVE - LOGON )\*\*\*\*\*

```
4C41 21 0000 SELDSK: LXI H,0 ;ERROR RETURN CODE.
4C44 79 MOV A,C ;PUT DRIVE NMBR IN A.
4C45 32 4A73 STA BT.DRV ;STORE DRIVE NUMBER.
4C48 FE02 CPI N.DRV5 ;CHECK IF LEGAL DRIVE.
4C4A D0 RNC ;NO CARRY IF ILLEGAL.
4C4B 7B MOV A,E ;CHECK IF LOG-ON REQ.
4C4C 32 4A82 STA LOG.RQ ;STORE LOGON REGISTER.
4C4F 3A 4A73 RETDSK: LDA BT.DRV ;GET DRIVE NUMBER.
4C52 6F MOV L,A ;L = DISK NUMBER.
4C53 2600 MVI H,0 ;ZERO H REGISTER.
4C55 29 DAD H ;*2.
4C56 29 DAD H ;*4.
4C57 29 DAD H ;*8.
4C58 29 DAD H ;*16 (SIZE OF HEADER).
4C59 11 4D57 LXI D,DO.DPH ;DRIVE 0 D.P.H.
4C5C 19 DAD D ;HL= DRIVE N DPH.
4C5D 03 4BB2 JMP LOG.ON ;GO CHECK LOG-ON.
```

\*\*\*\*\* ( SET TRACK )\*\*\*\*\*

```
4C60 79 SETTRK: MOV A,C ;MOVE TRACK NUMBER.
4C61 32 4A74 STA BT.TRK ;SAVE TRACK NUMBER.
4C64 09 RET ;RETURN TO CALLER.
```

\*\*\*\*\* ( SET SECTOR )\*\*\*\*\*

```
4C65 79 SETSEC: MOV A,C ;MOVE SECTOR NUMBER.
4C66 32 4A75 STA BT.SEC ;SAVE SECTOR NUMBER.
4C69 09 RET ;RETURN TO CALLER.
```

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

```
4C6A 60 SETDMA: MOV H,B ;HIGH ORDER MOVE.
4C6B 69 MOV L,C ;LOW ORDER MOVE.
4C6C 22 4A7E SHLD BT.DMA ;SAVE TRANSFER ADDRESS.
4C6F 09 RET ;RETURN TO CALLER.
```

\*\*\*\*\* ( SECTOR TRANSLATION ) \*\*\*\*\*

4C70	7A	SECTRN:	MOV	A,D	; TESTING TBL ADDR.
4C71	B3		ORA	E	; ADDR IN REG DE.
4C72	CA 4C7B		JZ	NOTRAN	; IF ZERO, NO TRANS.
4C75	EB		XCHG		; (HL) NOW TRANS TBL.
4C76	09		DAD	B	; (HL) NOW TRANS SECTOR.
4C77	6E		MOV	L,M	; L IS TRANSLATED SEC.
4C78	2600		MVI	H,0	; HIGH ORDER BYTE ZERO.
4C7A	09		RET		; RETURN TO CALLER.
4C7B	21 0001	NOTRAN:	LXI	H,1	; SET HL TO ONE.
4C7E	09		DAD	B	; ADD SEC NMBR TO HL.
4C7F	09		RET		; RETURN TO CALLER.

\*\*\*\*\*

\*\*\*\*\*  
 ; CONSOLE INPUT STATUS \*  
 \*\*\*\*\*

4C80	DB00	CNS.CK:	IN	CNI.SP	; INPUT STATUS
4C82	EE00		XRI	CNI.SI	; POLARITY ADJUST
4C84	E602		ANI	CNI.SB	; MASK STATUS BIT
4C86	08		RZ		; RETURN NOT RDY
4C87	3EFF		MVI	A,OFFH	; SET READY FLAG
4C89	09		RET		; RETURN RDY

\*\*\*\*\*  
 ; CONSOLE INPUT CHARACTER \*  
 \*\*\*\*\*

4C8A	CD 4C80	CNS.IN:	CALL	CNS.CK	; CALL STATUS CHECK.
4C8D	CA 4C8A		JZ	CNS.IN	; REPEAT TILL READY.
4C90	DB01		IN	CNI.DP	; INPUT CHARACTER.
4C92	E67F		ANI	07FH	; 7 BIT ASCII+SIGN.
4C94	09		RET		; RETURN TO CALLER.

\*\*\*\*\*  
 ; CONSOLE OUTPUT \*  
 \*\*\*\*\*

4C95	DB00	CNS.OT:	IN	CNO.SP	; INPUT STATUS
4C97	EE00		XRI	CNO.SI	; ADJUST POLARITY
4C99	E604		ANI	CNO.SB	; MASK STATUS BIT
4C9B	CA 4C95		JZ	CNS.OT	; TRY AGAIN
4C9E	79		MOV	A,C	; GET CHARACTER
4C9F	E67F		ANI	07FH	; 7 BIT ASCII
4CA1	D301		OUT	CNO.DP	; SEND CHARACTER
4CA3	09		RET		

\*\*\*\*\*

```

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

```

4CA4 3E1A READER: MVI A,CNTL.Z ;RETURN END OF FILE.
4CA6 C9 RET ;NOT IMPLEMENTED.

4CA7 C9 PUNCH: RET ;NOT IMPLEMENTED.
  
```

```

;*****
; PRINTER DRIVER AREA *
;*****
  
```

```

4CA8 03 4C95 LIST: JMP CNS.OT ;SEND TO CONSOLE

4CAB 3EFF LISTST: MVI A,OFFH ;READY
4CAD C9 RET
  
```

```

;*****
; SEND MESSAGE SUBROUTINE *
;*****
  
```

```

4CAE F5 MSG.OT: PUSH PSW ;SAVE CALLER FLAGS.
4CAF 4E ..REPT: MOV C,M ;LOAD CHARACTER.
4CB0 0D 4C95 CALL CNS.OT ;CONSOLE OUTPUT.
4CB3 7E MOV A,M ;SAME CHARACTER.
4CB4 23 INX H ;INCREMENT POINTER.
4CB5 E680 ANI 080H ;TEST SIGN BIT.
4CB7 CA 4CAF JZ ..REPT ;ANOTHER CHARACTER.
4CBA F1 POP PSW ;RESTORE FLAGS.
4CBB C9 RET ;RETURN TO CALLER.
  
```

```

;*****
  
```

```

000A LF == 00AH ;ASCII LINE FEED.
000D CR == 00DH ;CARRIAGE RETURN.
001A CNTL.Z == 01AH ;CONTROL-Z (EOF).
  
```

```

;*****
  
```



```

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

```

```

;***** ( SET UP FOR SYSTEM BOOT )*****

```

```

4CBC 31 0080 WARM: LXI SP,0080H ;SET UP STACK.
4CBF 3E00 MVI A,0 ;DRIVE 0 VALUE.
4CC1 32 4A73 STA BT.DRV ;SELECT DISK.
4CC4 01 3400 LXI B,CCP ;CP/M CCP ADDRESS.
4CC7 CD 4C6A CALL SETDMA ;SET DMA ADDR.
4CCA 0E02 MVI C,2 ;CCP 1ST SECTOR.
4CCC CD 4C65 CALL SETSEC ;SET SECTOR NMBR.
4CCF 0E01 MVI C,1 ;CCP/BDOS TRACK.
4CD1 CD 4C60 CALL SETTRK ;SET TRACK NUMBER.

```

```

;***** ( READ CCP/BDOS )*****

```

```

4CD4 CD 4B03 W.READ: CALL DISKRD ;READ ONE SECTOR.
4CD7 A7 ANA A ;SET FLAGS.
4CD8 C2 4CF4 JNZ W.EROR ;EXIT IF ERROR.
4CDB 3A 4A75 LDA BT.SEC ;GET SECTOR NMBR.
4CDE FE2D CPI 45 ;LAST SECTOR CHECK.
4CE0 CA 4CFB JZ W.ZRPG ;GOTO ZERO PAGE SET.
4CE3 3C INR A ;INCREMENT SECTOR.
4CE4 32 4A75 STA BT.SEC ;STORE NEXT SECTOR.
4CE7 11 0080 LXI D,SEC.SZ ;GET SECTOR SIZE.
4CEA 2A 4A7E LHLD BT.DMA ;GET TRANSFER ADDR.
4CED 19 DAD D ;CALCULATE NEW ADDR.
4CEE 22 4A7E SHLD BT.DMA ;SET NEW ADDRESS.
4CF1 C3 4CD4 JMP W.READ ;DO ANOTHER WARM READ.

```

```

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

```

```

4CF4 21 4D43 W.EROR: LXI H,MSG.LE ;GET ERROR MESSAGE.
4CF7 CD 4CAE CALL MSG.OT ;ISSUE MESSAGE.
4CFA 76 HLT ;OR GOTO MONITOR

```

```

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

```

```

4CFB 01 0008 W.ZRPG: LXI B,8 ;BASE IMAGE SIZE.
4CFE 11 0000 LXI D,0 ;BASE ADDRESS SET.
4D01 21 4D3B LXI H,BS.IMG ;BASE IMAGE ADDR.
4D04 CD 4BA7 CALL BLOCK ;BLOCK MOVE ROUTINE.

```

```

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

```

```

4D07 21 0080 LXI H,0080H ;DEFAULT SECTOR BUFF.
4D0A 22 4A7E SHLD BT.DMA ;SET TRANSFER ADDRESS.
4D0D 0E00 MVI C,0 ;DISK NMBR: OLD DISK?
4D0F C3 3400 JMP CCP ;JUMP INTO CCP CP/M.

```

```

;*****

```

```
*****  
; (DISK REQUEST PARAMETERS)  
*****
```

0040 D.ADDR == 0040H ;DD SYSTEM MEM ADDRESS.

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

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

4D12 01070D131905 SDTRAN: .BYTE 01H,07H,0DH,13H,19H,05H  
4D18 0B111703090F .BYTE 0BH,11H,17H,03H,09H,0FH  
4D1E 1502080E141A .BYTE 15H,02H,08H,0EH,14H,1AH  
4D24 060C1218040A .BYTE 06H,0CH,12H,18H,04H,0AH  
4D2A 1016 .BYTE 10H,16H

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

4D2C 001A SD.PBK: .WORD 26 ;SECTORS PER TRACK.  
4D2E 03 .BYTE 3 ;BLOCK SHIFT FACTOR.  
4D2F 07 .BYTE 7 ;BLOCK MASK.  
4D30 00 .BYTE 0 ;NULL MASK.  
4D31 00F2 .WORD 242 ;DISK SIZE - 1.  
4D33 003F .WORD 63 ;DIRECTORY MAX.  
4D35 C0 .BYTE 11000000B ;ALLOC 0.  
4D36 00 .BYTE 0 ;ALLOC 1.  
4D37 0010 .WORD 16 ;CHECK SIZE.  
4D39 0002 .WORD 2 ;TRACK OFFSET.

```
*****  
; ZERO PAGE IMAGE *  
*****
```

4D3B C3 4A03 BS.IMG: JMP BIOS+03H ;WARM BOOT VECTOR.  
4D3E FF00 .BYTE IOBYTE,DF,DRV ;I/O BYTE AND DFLT DRV.  
4D40 C3 3C06 JMP BDOS+06H ;BDOS CALL VECTOR.

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

4D43 0D0A4C4F4144 MSG.LE: .ASCIS [CR][LF]'LOAD ERROR'

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

4D4F 4A6164652044 JADEID: .ASCII 'JADE DD' ;ID LABEL.  
0003 ID.SZE == .-JADEID ;LABEL SIZE.

```
*****
```

\*\*\*\*\*  
; DRIVE PARAMETER HEADER AREA \*  
\*\*\*\*\*

4D57	0000	D0.DPH:	.WORD	0	;SECTOR TRAN TBL.
4D59	0000		.WORD	0	;SCRATCH AREA.
4D5B	0000		.WORD	0	;SCRATCH AREA.
4D5D	0000		.WORD	0	;SCRATCH AREA.
4D5F	4A83		.WORD	DIR.BF	;DIRECTORY BUFFER.
4D61	4A36		.WORD	D0.DPB	;DRIVE PARAM BLK.
4D63	4E1B		.WORD	D0.CHK	;DRIVE CHANGE BLK.
4D65	4D97		.WORD	D0.ALL	;DRIVE ALLOCATION.

\*\*\*\*\*

4D67	0000	D1.DPH:	.WORD	0	
4D69	0000		.WORD	0	
4D6B	0000		.WORD	0	
4D6D	0000		.WORD	0	
4D6F	4A83		.WORD	DIR.BF	
4D71	4A45		.WORD	D1.DPB	
4D73	4E2B		.WORD	D1.CHK	
4D75	4DB8		.WORD	D1.ALL	

\*\*\*\*\*

4D77	0000	D2.DPH:	.WORD	0	
4D79	0000		.WORD	0	
4D7B	0000		.WORD	0	
4D7D	0000		.WORD	0	
4D7F	4A83		.WORD	DIR.BF	
4D81	4A54		.WORD	D2.DPB	
4D83	4E3B		.WORD	D2.CHK	
4D85	4DD9		.WORD	D2.ALL	

\*\*\*\*\*

4D87	0000	D3.DPH:	.WORD	0	
4D89	0000		.WORD	0	
4D8B	0000		.WORD	0	
4D8D	0000		.WORD	0	
4D8F	4A83		.WORD	DIR.BF	
4D91	4A63		.WORD	D3.DPB	
4D93	4E4B		.WORD	D3.CHK	
4D95	4DFA		.WORD	D3.ALL	

\*\*\*\*\*

```
*****  
; DRIVE ALLOCATION AREAS DECLARED  
*****
```

```
4D97      ADDR      =      .      ;AT CURRENT LOCATION.  
0021      ALL.SZ    ==     33      ;ALLOCATION BLOCK SIZE.  
  
4D97      D0.ALL    ==     ADDR     ;ALLOCATION FOR DRIVE 0.  
4DB8      ADDR      =     ADDR+ALL.SZ  
4DB8      D1.ALL    ==     ADDR     ;ALLOCATION FOR DRIVE 1.  
4DD9      ADDR      =     ADDR+ALL.SZ  
4DD9      D2.ALL    ==     ADDR     ;ALLOCATION FOR DRIVE 2.  
4DFA      ADDR      =     ADDR+ALL.SZ  
4DFA      D3.ALL    ==     ADDR     ;ALLOCATION FOR DRIVE 3.  
4E1B      ADDR      =     ADDR+ALL.SZ
```

```
*****  
; CHANGED DISK CHECK AREAS DECLARED  
*****
```

```
0010      CHK.SZ    ==     16       ;CHECK AREA SIZE.  
4E1B      D0.CHK    ==     ADDR     ;CHECK AREA FOR DRIVE 0.  
4E2B      ADDR      =     ADDR+CHK.SZ  
4E2B      D1.CHK    ==     ADDR     ;CHECK AREA FOR DRIVE 1.  
4E3B      ADDR      =     ADDR+CHK.SZ  
4E3B      D2.CHK    ==     ADDR     ;CHECK AREA FOR DRIVE 2.  
4E4B      ADDR      =     ADDR+CHK.SZ  
4E4B      D3.CHK    ==     ADDR     ;CHECK AREA FOR DRIVE 3.  
4E5B      ADDR      =     ADDR+CHK.SZ
```

```
*****  
.END
```

BIOS - JADE DOUBLE D CP/M 2.2

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

ADDR	4E5B	ALL.SZ	0021	BDOS	3C00	BIOS	4A00
BIOS.R	0600	BLOCK	4BA7	BS.IMG	4D3B	BT.CMD	4A72
BT.DMA	4A7E	BT.DRV	4A73	BT.LAD	4A7A	BT.LNG	4A7C
BT.MOD	4A78	BT.SEC	4A75	BT.SPO	4A76	BT.SP1	4A77
BT.STS	4A79	BT.TRK	4A74	CCP	3400	CHK.SZ	0010
CNI.DP	0001	CNI.SB	0002	CNI.SI	0000	CNI.SP	0000
CNO.DP	0001	CNO.SB	0004	CNO.SI	0000	CNO.SP	0000
CNS.CK	4C80	CNS.IN	4C8A	CNS.OT	4C95	CNTL.Z	001A
CPM.BS	0000	CPM.SZ	5000	CR	000D	C.LOGO	4AC0
C.ONES	4A99	C.TENS	4A8E	DO.ALL	4D97	DO.CHK	4E1B
DO.DPB	4A36	DO.DPH	4D57	D1.ALL	4DB8	D1.CHK	4E2B
D1.DPB	4A45	D1.DPH	4D67	D2.ALL	4DD9	D2.CHK	4E3B
D2.DPB	4A54	D2.DPH	4D77	D3.ALL	4DFA	D3.CHK	4E4B
D3.DPB	4A63	D3.DPH	4D87	DC.FMT	0003	DC.INT	0002
DC.LOG	0000	DC.MBO	0001	DC.MB1	0003	DC.RDS	0001
DC.SIN	0001	DC.SOT	0000	DC.WRS	0002	DD.BUF	0380
DD.CHT	0370	DD.DDF	03B1	DD.DPB	03A0	DD.DRV	0371
DD.FBF	0300	DD.SEC	0373	DD.STS	0377	DD.TRK	0372
DF.DRV	0000	DIR.BF	4A83	DISKRD	4B03	DISKWR	4B23
DPB.AD	4C31	DPB.SZ	000F	DSK.ER	4B49	DSK.EX	4B78
DSK.OK	4B43	DSK.WT	4B94	DS.HLT	0002	DT.PTR	4A80
D.ADDR	0040	D.PORT	0043	FALSE	0000	FMT.SZ	0100
FORMAT	4B50	HOME	4C3C	ID.SZE	0008	INIT	4A83
IOBYTE	00FF	JADEID	4D4F	K.BYTE	0400	LF	000A
LG3740	4C0A	LIST	4CA8	LISTST	4CAB	LOG.CK	4BCF
LOG.ID	4BDB	LOG.ON	4BB2	LOG.RQ	4A82	MSG.LE	4D43
MSG.OT	4CAE	MSG.SO	4ACF	NK.SYS	0014	NOTRAN	4C7B
N.DRVS	0002	PUNCH	4CA7	READER	4CA4	RETDSK	4C4F
REV.B	0001	REV.C	0000	SDTRAN	4D12	SD.PBK	4D2C
SECTRN	4C70	SEC.SZ	0080	SELDSK	4C41	SETDMA	4C6A
SETSEC	4C65	SETTRK	4C60	SYS.OD	4AED	SYS.TD	4AEC
TPA	0100	TR3740	4C1F	TRNONE	4C29	TRUE	0001
WARM	4CBC	W.ERROR	4CF4	W.READ	4CD4	W.ZRPG	4CFB

• •

○

○

○

```
*****  
;  
; PROGRAM ID: FORMAT  
;  
*****  
;  
; PROPERTY OF: JADE COMPUTER PRODUCTS  
; 4901 W. ROSECRANS BLVD.  
; HAWTHORNE, CALIFORNIA  
; 90250, U.S.A.  
;  
*****  
;  
; VERSION: 2.2  
;  
*****  
;  
; 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.  
; THE USER CAN FORMAT ONLY THE SYSTEM TRACKS OF OLDER  
; SINGLE DENSITY DISKETTES SUCH THAT THESE DISKETTES  
; CAN CARRY THE OPERATING SYSTEM WITHOUT ERASING THE  
; DATA TRACKS. FORMAT.COM IS 8080/8085/Z80 COMPATABLE  
*****
```

\*\*\*\*\*  
; DRIVER MODULE DEFINITIONS \*  
\*\*\*\*\*

000A	LF	==	00AH	;ASCII LINE FEED.
000B	CR	==	00DH	;CARRIAGE RETURN.
0100	TPA	==	0100H	;TRANSIENT PROGRAM.
0003	CNTL.C	==	3	;REQUEST REBOOT CPM.
0000	TRK.0	==	0	;TRACK 0.
0001	TRK.1	==	1	;TRACK 1.
0002	TRK.2	==	2	;TRACK 2.
004C	TRK.76	==	76	;TRACK 76.
0080	SEC.SZ	==	128	;128 BYTES PER SECTOR.
0001	ID.SEC	==	1	;ID SECTOR NUMBER.
0000	REBOOT	==	0	;REBOOT ADDRESS.

\*\*\*\*\*  
; SYSTEM TRACKS COPY DEFINITIONS \*  
\*\*\*\*\*

1000	TK0.LP	==	1000H	;TRACK 0 TPA LOAD ADDR.
0001	TK0.FS	==	1	;TRACK 0 FIRST SECTOR.
001A	TK0.LS	==	26	;TRACK 0 LAST SECTOR.
001A	TK0.NS	==	TK0.LS-TK0.FS+1	;TRACK 0 NMBER SECTORS.
0F80	TK0.LC	==	TK0.LP-SEC.SZ*TK0.FS	;TRACK 0 LOAD CONSTANT.
0D00	TK0.SZ	==	SEC.SZ*TK0.NS	;TRACK 0 LOAD SIZE.
1D00	TK1.LP	==	TK0.LP+TK0.SZ	;TRACK 1 TPA LOAD ADDR.
0001	TK1.FS	==	1	;TRACK 1 FIRST SECTOR.
0030	TK1.LS	==	48	;TRACK 1 LAST SECTOR.

\*\*\*\*\*  
; 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	ZERUS	==	00000000B	;ALL ZERO BYTE.
00FF	ONES	==	11111111B	;ALL ONES BYTE.

\*\*\*\*\*  
; BDOS CALL - VECTOR NUMBERS \*  
\*\*\*\*\*

0005	BDOS	==	0005H	;SYSTEM CALL ADDR.
0001	CNS.RD	==	1	;READ CONSOLE.
0002	CNS.WR	==	2	;WRITE CONSOLE.



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

001B	BS.DSK	==	01BH	;SELECT DISK OFFSET.
001E	BS.TRK	==	01EH	;SET TRACK OFFSET.
0021	BS.SEC	==	021H	;SET SECTOR OFFSET.
0024	BS.DMA	==	024H	;SET ADDRESS OFFSET.
0027	BS.RDS	==	027H	;READ SECTOR OFFSET.
002A	BS.WRS	==	02AH	;WRITE SECTOR OFFSET.
0033	BS.FMT	==	033H	;FORMAT TRACK OFFSET.
0002	BS.PAG	==	0002H	;BIOS VECTOR POINTER.
0001	LOG.VC	==	1	;DRV SEL DONT LOGON.

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

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

```
*****  
; PROGRAM BEGINS *  
*****
```

```
0100 C3 0138 BEGIN: JMP INIT ;GO TO INITIALIZE.
```

```
*****  
; ASCII IDENTIFICATION INSERT *  
*****
```

```
0103 4A4144452043 .ASCII 'JADE COMPUTER PRODUCTS '  
011A 444F55424C45 .ASCII 'DOUBLE D - FORMAT '  
012C 56455253494F .ASCII 'VERSION 2.2 '
```

```
*****  
; SET STACK POINTER AND ISSUE LOG-ON *  
*****
```

```
0138 31 0100 INIT: LXI SP,TPA ;SET STACK POINTER.  
013B 21 0521 LXI H,MSG.BG ;LOAD MESSAGE ADDR.  
013E CD 0281 CALL MSG.OT ;ISSUE MESSAGE.
```

```
*****  
; FUNCTION SELECTION AND CONTROL DISTRIBUTOR *  
*****
```

```
0141 21 056C FUN.BG: LXI H,MSG.FL ;FUNCTION LIST ADDRESS.  
0144 CD 0281 CALL MSG.OT ;ISSUE MESSAGE.  
0147 CD 028F CALL CNS.IN ;GET CONSOLE CHARACTER.  
014A FE31 CPI "1"  
014C CA 017E JZ FUN.1 ;FMT DOUBLE DENSITY.  
014F FE32 CPI "2"  
0151 CA 01E3 JZ FUN.2 ;FMT SINGLE DENSITY.  
0154 FE33 CPI "3"  
0156 CA 01BB JZ FUN.3 ;FMT 3740.  
0159 FE34 CPI "4"  
015B CA 022B JZ FUN.4 ;FMT SYSTEM TRACKS.  
015E FE35 CPI "5"  
0160 CA 0259 JZ FUN.5 ;READ SYSTEM TRACKS.  
0163 FE36 CPI "6"  
0165 CA 026D JZ FUN.6 ;WRITE SYSTEM TRACKS.  
0168 FE26 CPI "&"  
016A CA 03FD JZ RST.6 ;DEBUG TDL TRAP.  
016D FE27 CPI "~"  
016F CA 03FE JZ RST.7 ;DDT TRAP.  
0172 21 066C LXI H,MSG.SE ;SELECT ERROR MESSAGE.  
0175 32 0670 STA LTR.SE ;STORE BAD SYMBOL.  
0178 CD 0281 CALL MSG.OT ;ISSUE MESSAGE.  
017B C3 0141 JMP FUN.BG ;TRY SELECTION AGAIN.
```

```
*****
```

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

017E	3A 04D2	FUN.1:	LDA	DD.FLG	;LOAD DDENS FLAGS.
0181	32 0406		STA	F.FLAG	;STORE FORMAT FLAGS.
0184	21 068B		LXI	H,MSG.FD	;FORMAT DRIVE MESSAGE.
0187	CD 03AA		CALL	SEL.DR	;SELECT DRIVE LETTER.
018A	DA 0141		JC	FUN.BG	;CARRY: RESELECT.
018D	3E00		MVI	A,TRK.0	;TRACK 0.
018F	32 0404		STA	TRK.NO	;SET TRACK NUMBER.
0192	CD 02E4		CALL	FMT.SD	;FORMAT TRACK SDENS.
0195	DA 017E		JC	FUN.1	;JUMP ERROR DETECTED.
0198	3E01		MVI	A,TRK.1	;TRACK 1 VALUE.
019A	32 0404		STA	TRK.NO	;SET TRACK NUMBER.
019D	CD 02EA	..REPT:	CALL	FMT.DD	;FORMAT TRACK DDENS.
01A0	DA 017E		JC	FUN.1	;JUMP ERROR DETECTED.
01A3	3A 0404		LDA	TRK.NO	;LOAD TRACK NUMBER.
01A6	FE4C		CPI	TRK.76	;WAS IT THE LAST TRK.
01A8	CA 01B2		JZ	..ID	;AFTER LAST TRACK.
01AB	3C		INR	A	;INCREMENT TRACK NO.
01AC	32 0404		STA	TRK.NO	;STORE TRACK NUMBER.
01AF	C3 019D		JMP	..REPT	;FORMAT NEXT TRACK.
01B2	21 04A1	..ID:	LXI	H,IDS.DD	;ID SECTOR: DDENS.
01B5	CD 02C1		CALL	WRT.ID	;WRITE ID SECTOR.
01B8	C3 017E		JMP	FUN.1	;ANOTHER DISK?

```
*****
; FUNCTION 3 - FORMAT STANDARD 3740
*****
```

01BB	3E00	FUN.3:	MVI	A,0	;3740 SDENS FLAGS.
01BD	32 0406		STA	F.FLAG	;STORE FORMAT FLAGS.
01C0	21 068B		LXI	H,MSG.FD	;FORMAT DRIVE MESSAGE.
01C3	CD 03AA		CALL	SEL.DR	;SELECT DRIVE LETTER.
01C6	DA 0141		JC	FUN.BG	;CARRY: RESELECT.
01C9	3E00		MVI	A,TRK.0	;TRACK 0.
01CB	32 0404		STA	TRK.NO	;SET TRACK NUMBER.
01CE	CD 02E4	..REPT:	CALL	FMT.SD	;FORMAT TRACK SDENS.
01D1	DA 01BB		JC	FUN.3	;JUMP ERROR DETECTED.
01D4	3A 0404		LDA	TRK.NO	;LOAD TRACK NUMBER.
01D7	FE4C		CPI	TRK.76	;WAS IT THE LAST TRK.
01D9	CA 01BB		JZ	FUN.3	;AFTER LAST TRACK.
01DC	3C		INR	A	;INCREMENT TRACK NO.
01DD	32 0404		STA	TRK.NO	;STORE TRACK NUMBER.
01E0	C3 01CE		JMP	..REPT	;FORMAT NEXT TRACK.

```
*****
```

FORMAT - JADE DOUBLE D  
FUNCTION CONTROLLERS

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

```

01E3 3A 0452 FUN.2: LDA SD.FLG ;LOAD SDENS FLAGS.
01E6 32 0406 STA F.FLAG ;STORE FORMAT FLAGS.
01E9 21 068B LXI H,MSG.FD ;FORMAT DRIVE MESSAGE.
01EC CD 03AA CALL SEL.DR ;SELECT DRIVE LETTER.
01EF DA 0141 JC FUN.BG ;CARRY: RESELECT.
01F2 3E00 MVI A,TRK.0 ;TRACK 0.
01F4 32 0404 STA TRK.NO ;SET TRACK NUMBER.
01F7 CD 02E4 CALL FMT.SD ;FORMAT TRACK SDENS.
01FA DA 01E3 JC FUN.2 ;JUMP ERROR DETECTED.
01FD 3E01 MVI A,TRK.1 ;TRACK 1
01FF 32 0404 STA TRK.NO ;SET TRACK NUMBER.
0202 CD 02EA CALL FMT.DD ;FORMAT TRACK DDENS.
0205 DA 01E3 JC FUN.2 ;JUMP ERROR DETECTED.
0208 3E02 MVI A,TRK.2 ;TRACK 2.
020A 32 0404 STA TRK.NO ;SET TRACK NUMBER.
020D CD 02E4 ..REPT: CALL FMT.SD ;FORMAT TRACK SDENS.
0210 DA 01E3 JC FUN.2 ;JUMP ERROR DETECTED.
0213 3A 0404 LDA TRK.NO ;LOAD TRACK NUMBER.
0216 FE4C CPI TRK.76 ;WAS IT THE LAST TRK.
0218 CA 0222 JZ ..ID ;AFTER LAST TRACK.
021B 3C INR A ;INCREMENT TRACK NO.
021C 32 0404 STA TRK.NO ;STORE TRACK NUMBER.
021F C3 020D JMP ..REPT ;FORMAT NEXT TRACK.
0222 21 0421 ..ID: LXI H,IDS.SD ;ID SECTOR: SDENS.
0225 CD 02C1 CALL WRT.ID ;WRITE ID SECTOR.
0228 C3 01E3 JMP FUN.2 ;ANOTHER DISK?

```

\*\*\*\*\*  
; FUNCTION 4 - FORMAT JADE SYSTEM TRACKS ONLY \*  
\*\*\*\*\*

```

022B 3A 0452 FUN.4: LDA SD.FLG ;LOAD SDENS FLAGS.
022E 32 0406 STA F.FLAG ;STORE FORMAT FLAGS.
0231 21 068B LXI H,MSG.FD ;FORMAT DRIVE MESSAGE.
0234 CD 03AA CALL SEL.DR ;SELECT DRIVE LETTER.
0237 DA 0141 JC FUN.BG ;CARRY: RESELECT.
023A 3E00 MVI A,TRK.0 ;TRACK 0.
023C 32 0404 STA TRK.NO ;SET TRACK NUMBER.
023F CD 02E4 CALL FMT.SD ;FORMAT TRACK SDENS.
0242 DA 022B JC FUN.4 ;JUMP ERROR DETECTED.
0245 3E01 MVI A,TRK.1 ;TRACK 1
0247 32 0404 STA TRK.NO ;SET TRACK NUMBER.
024A CD 02EA CALL FMT.DD ;FORMAT TRACK DDENS.
024D DA 022B JC FUN.4 ;JUMP ERROR DETECTED.
0250 21 0421 LXI H,IDS.SD ;ID SECTOR: SDENS.
0253 CD 02C1 CALL WRT.ID ;WRITE ID SECTOR.
0256 C3 022B JMP FUN.4 ;ANOTHER DISK?

```

\*\*\*\*\*

```
*****  
; FUNCTION 5 - READ SYSTEM TRACKS  
*****
```

0259	21 06B7	FUN.5:	LXI	H,MSG.RS	;READ SYSTEM MESSAGE.
025C	CD 03AA		CALL	SEL.DR	;SELECT DRIVE LETTER.
025F	DA 0141		JC	FUN.BG	;GOTO RESELECT.
0262	3E27		MVI	A,BS.RDS	;READ SECTOR VECTOR.
0264	32 0401		STA	RW.OPV	;TRANSFER IS READ.
0267	CD 0314		CALL	TRNSFR	;READ SYSTEM TRACKS.
026A	C3 0141		JMP	FUN.BG	;RESELECT FUNCTION.

```
*****  
; FUNCTION 6 - WRITE SYSTEM TRACKS  
*****
```

026D	21 06D3	FUN.6:	LXI	H,MSG.WS	;WRITE SYSTEM MESSAGE.
0270	CD 03AA		CALL	SEL.DR	;SELECT REQUESTED DRV.
0273	DA 0141		JC	FUN.BG	;GOTO RESELECT.
0276	3E2A		MVI	A,BS.WRS	;WRITE SECTOR VECTOR.
0278	32 0401		STA	RW.OPV	;TRANSFER IS WRITE
027B	CD 0314		CALL	TRNSFR	;WRITE SYSTEM TRACKS.
027E	C3 026D		JMP	FUN.6	;WRITE ANOTHER DISK.

```
*****
```

```

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

```

```

0281 F5 MSG.OT: PUSH PSW ;SAVE CALLER FLAGS.
0282 7E ..REPT: MOV A,M ;LOAD CHARACTER.
0283 CD 029C CALL CNS.OT ;CONSOLE OUTPUT.
0284 7E MOV A,M ;SAME CAHRACTER.
0287 23 INX H ;INCREMENT POINTER.
0288 E680 ANI 080H ;TEST SIGN BIT.
028A CA 0282 JZ ..REPT ;ANOTHER CHARACTER.
028D F1 POP PSW ;RESTORE FLAGS.
028E C9 RET ;RETURN TO CALLER.

```

```

;*****
; CONSOLE LINKAGE - BDOS CALLS
;*****

```

```

028F 0E01 CNS.IN: MVI C,CNS.RD ;BDOS CONSOLE READ.
0291 CD 0005 CALL BDOS ;SYSTEM CALL.
0294 E67F ANI 07FH ;7 BIT ASCII.
0296 FE03 CPI CNTL.C ;CHECK CONTROL-C.
0298 CA 0000 JZ REBOOT ;REBOOT SYSTEM.
029B C9 RET ;RETURN CALLER.

```

```

029C E5 CNS.OT: PUSH H ;SAVE TEXT INDEX.
029D 5F MOV E,A ;MOVE CHARACTER.
029E 0E02 MVI C,CNS.WR ;BDOS CONSOLE WRITE.
02A0 CD 0005 CALL BDOS ;SYSTEM CALL.
02A3 E1 POP H ;SAVE TEXT POINTER.
02A4 C9 RET ;RETURN CALLER.

```

```

;*****
; DISPLAY HEXIDECIMAL VALUE
;*****

```

```

02A5 7C HXW.OT: MOV A,H ;LOAD HIGH ORDER.
02A6 CD 02AA CALL HXB.OT ;DO HIGH BYTE.
02A9 7D MOV A,L ;LOAD LOW ORDER.
02AA F5 HXB.OT: PUSH PSW ;SAVE CHARACTER.
02AB 0F RRC ;SHIFT
02AC 0F RRC
02AD 0F RRC
02AE 0F RRC
02AF CD 02B3 CALL HXN.OT ;DO UPPER NIBBLE.
02B2 F1 POP PSW ;GET LOWER NIBBLE.
02B3 E60F ANI 00FH ;MASK NIBBLE.
02B5 FE0A CPI 00AH ;LESS THAN 10?
02B7 DA 02BC JC ..NMBR ;DO NUMBER THING.
02BA C607 ..LTTR: ADI 'A'-'9'-1 ;ADD LETTER OFFSET.
02BC C630 ..NMBR: ADI '0' ;START AT ZERO.
02BE C3 029C JMP CNS.OT ;CONSOLE OUTPUT.

```

```

;*****

```

```

;*****
; WRITE ID SECTOR *
;*****

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

02C1    44          WRT.ID: MOV     B,H           ;MOV HL TO BC.
02C2    4D          MOV     C,L           ;MOV HL TO BC.
02C3    2E24        MVI     L,BS.DMA      ;BIOS DMA VECTOR.
02C5    CD 03F8     CALL    BIOS           ;SET TRANSFER ADDRESS.

;***** ( SET TRACK TO ZERO )*****

02C8    0E00        MVI     C,TRK.0       ;TRACK 0 SET.
02CA    2E1E        MVI     L,BS.TRK      ;BIOS TRACK VECTOR.
02CC    CD 03F8     CALL    BIOS           ;SET TRACK NMBR.

;***** ( SET SECTOR TO ID.SEC )*****

02CF    0E01        MVI     C,ID.SEC      ;ID SECTOR VALUE.
02D1    2E21        MVI     L,BS.SEC      ;BIOS SECTOR VECTOR.
02D3    CD 03F8     CALL    BIOS           ;BIOS TRANSFER CALL.

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

02D6    2E2A        MVI     L,BS.WRS      ;BIOS WRITE VECTOR.
02D8    CD 03F8     CALL    BIOS           ;BIOS TRANSFER CALL.
02DB    B7          ORA     A           ;SET CONDITION CODES.
02DC    C8          RZ           ;RETURN USER GOOD.
02DD    21 06FF     LXI     H,MSG.NC      ;TRANSFER INCOMPLETE.
02E0    CD 0281     CALL    MSG.OT        ;ISSUE MESSAGE.
02E3    C9          RET

;*****

```

FORMAT - JADE DOUBLE D

FORMAT TRACK DRIVER LINKAGE

```

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

```

```

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

```

```

02E4 21 0800 FMT.SD: LXI H,FT3740 ;LOAD INJECTION ADDR.
02E7 C3 02ED JMP ST.DMA ;GO SET DMA ADDR.
02EA 21 0900 FMT.DD: LXI H,FTJ48D ;LOAD INJECTION ADDR.

```

```

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

```

```

02ED 44 ST.DMA: MOV B,H ;MOV HL TO BC.
02EE 4D MOV C,L ;MOV HL TO BC.
02EF 2E24 MVI L,BS.DMA ;BIOS DMA VECTOR.
02F1 CD 03F8 CALL BIOS ;SET TRANSFER ADDRESS.

```

```

;***** ( SET TRACK )*****

```

```

02F4 3A 0404 LDA TRK.NO ;LOAD TRACK NMBR.
02F7 4F MOV C,A ;PUT INTO C REGISTER.
02F8 2E1E MVI L,BS.TRK ;BIOS TRACK VECTOR.
02FA CD 03F8 CALL BIOS ;SET TRACK NMBR.

```

```

;***** ( SET SECTOR TO FLAG )*****

```

```

02FD 3A 0406 LDA F.FLAG ;LOAD DCM FLAG.
0300 4F MOV C,A ;DCM FLAGS.
0301 2E21 MVI L,BS.SEC ;BIOS SECTOR VECTOR.
0303 CD 03F8 CALL BIOS ;BIOS TRANSFER CALL.

```

```

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

```

```

0306 2E33 MVI L,BS.FMT ;BIOS WRITE VECTOR.
0308 CD 03F8 CALL BIOS ;BIOS TRANSFER CALL.
030B B7 ORA A ;SET CONDITION CODES.
030C C8 RZ ;RETURN USER GOOD.
030D 21 06FF LXI H,MSG.NC ;TRANSFER INCOMPLETE.
0310 CD 0281 CALL MSG.OT ;ISSUE MESSAGE.
0313 C9 RET

```

```

;*****

```



```

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

```

```

;***** ( TRANSFER TRACK 0 )*****

```

```

0314    21 0407    TRNSFR: LXI    H,TKO.SL    ;ADDR OF SECTOR LIST.
0317    23                INX    H        ;SKIP ID SECTOR.
0318    22 03FF    SHLD   SYS.WP    ;INIT WORK POINTER.
031B    0E00        MVI    C,TRK.0    ;CBS TRACK NUMBER.
031D    2E1E        MVI    L,BS.TRK   ;BIOS TRACK VECTOR.
031F    CD 03F8    CALL   BIOS      ;BIOS TRANSFER CALL.

```

```

;***** ( SET SECTOR IN BIOS )*****

```

```

0322    2A 03FF    ..TKOS: LHLD   SYS.WP    ;LOAD SECTOR POINTER.
0325    4E                MOV    C,M        ;SECTOR NMBR TO REG C.
0326    2E21        MVI    L,BS.SEC   ;BIOS SETSEC VECTOR.
0328    CD 03F8    CALL   BIOS      ;BIOS TRANSFER CALL.

```

```

;***** ( SET ADDRESS IN BIOS )*****

```

```

032B    2A 03FF    LHLD   SYS.WP    ;LOAD SECTOR POINTER.
032E    A7                ANA    A        ;CLEAR CARRY BIT.
032F    7E                MOV    A,M        ;GET SECTOR NUMBER.
0330    1F                RAR                ;SECTOR NMBR / 2.
0331    57                MOV    D,A        ;HIGH LOAD OFFSET.
0332    3E00        MVI    A,0        ;ZERO A REGISTER.
0334    1F                RAR                ;CARRY TO BIT 7.
0335    5F                MOV    E,A        ;LOW LOAD OFFSET.
0336    21 0F80    LXI    H,TKO.LC   ;TRK 0 LOAD CONSTANT.
0339    19                DAD   D        ;HL = SEC LOAD ADDR.
033A    44                MOV    B,H        ;HL TO BC HIGH ORDER.
033B    4D                MOV    C,L        ;HL TO BC LOW ORDER.
033C    2E24        MVI    L,BS.DMA   ;BIOS ADDR VECTOR.
033E    CD 03F8    CALL   BIOS      ;BIOS TRANSFER CALL.

```

```

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

```

```

0341    3A 0401    LDA    RW.OPV     ;LOAD OPERATION VECTOR.
0344    6F                MOV    L,A        ;TRANSFER TO REG L.
0345    CD 03F8    CALL   BIOS      ;BIOS TRANSFER CALL.
0348    B7                ORA    A        ;SET CONDITION CODES.
0349    C2 03A3    JNZ   ..BLOW     ;SOMETHING WRONG!

```

```

;***** ( INCREMENT NEXT SECTOR )*****

```

```

034C    2A 03FF    LHLD   SYS.WP    ;LOAD SECTOR POINTER.
034F    7E                MOV    A,M        ;LOAD SECTOR NUMBER.
0350    FE16        CPI    016H       ;CHECK LAST STAGGERED.
0352    CA 035C    JZ    ..TRK1     ;IF DONE, DO TRK 1.
0355    23                INX    H        ;INCREMENT POINTER.
0356    22 03FF    SHLD   SYS.WP    ;STORE IT AWAY.
0359    C3 0322    JMP   ..TKOS     ;NEXT TRK 0 SECTOR.

```

\*\*\*\*\* ( TRANSFER TRACK 1 )\*\*\*\*\*

```
035C 21 1D00  ..TRK1: LXI    H,TK1.LP    ;TRK 1 ADDR IN TPA.
035F 22 03FF          SHLD   SYS.WP    ;INIT WORK POINTER.
0362 3E01          MVI    A,TK1.FS    ;FIRST SECTOR NUMBER.
0364 32 0405          STA    SEC.NO    ;SET CURRENT SECTOR.
0367 0E01          MVI    C,TRK.1    ;SET TRACK AT 1.
0369 2E1E          MVI    L,BS.TRK   ;BIOS TRACK VECTOR.
036B CD 03F8          CALL   BIOS      ;BIOS TRANSFER CALL.
```

\*\*\*\*\* ( SET SECTOR AND DMA IN BIOS )\*\*\*\*\*

```
036E 3A 0405  ..TK1S: LDA    SEC.NO    ;LOAD CURRENT SECTOR.
0371 4F          MOV    C,A      ;TRANSFER TO REG C.
0372 2E21          MVI    L,BS.SEC   ;BIOS SECTOR VECTOR.
0374 CD 03F8          CALL   BIOS      ;BIOS TRANSFER CALL.
0377 2A 03FF          LHLD  SYS.WP    ;LOAD WORK POINTER.
037A 44          MOV    B,H      ;MOVE HIGH ORDER.
037B 4D          MOV    C,L      ; AND LOW ORDER.
037C 2E24          MVI    L,BS.DMA  ;BIOS ADDR VECTOR.
037E CD 03F8          CALL   BIOS      ;BIOS TRANSFER CALL.
```

\*\*\*\*\* ( SECTOR TRANSFER OPERATION )\*\*\*\*\*

```
0381 3A 0401          LDA    RW.OPV    ;LOAD OPERATION VECTOR.
0384 6F          MOV    L,A      ;TRANSFER TO REG L.
0385 CD 03F8          CALL   BIOS      ;BIOS TRANSFER CALL.
0388 B7          ORA    A      ;SET CONDITION CODES.
0389 C2 03A3          JNZ   ..BLOW    ;SOMETHING WRONG!
```

\*\*\*\*\* ( INCREMENT NEXT SECTOR )\*\*\*\*\*

```
038C 3A 0405          LDA    SEC.NO    ;GET CURRENT SECTOR.
038F FE30          CPI    TK1.LS   ;WAS THIS THE LAST?
0391 C8          RZ          ;IF DONE, GO HOME.
0392 3C          INR    A      ;INCREMENT COUNT.
0393 32 0405          STA    SEC.NO    ;STORE IT AWAY.
0396 2A 03FF          LHLD  SYS.WP    ;LOAD WORK ADDRESS.
0399 11 0080          LXI    D,SEC.SZ  ;GET SECTOR SIZE.
039C 19          DIAD   D      ;ADD TO ADDRESS.
039D 22 03FF          SHLD  SYS.WP    ;STORE NEW ADDRESS.
03A0 C3 036E          JMP   ..TK1S    ;ANOTHER TRK 1 SECTOR.
```

\*\*\*\*\* ( ENCOUNTERED DIFFICULTY )\*\*\*\*\*

```
03A3 21 06FF  ..BLOW: LXI    H,MSG.NC    ;MESSAGE ADDRESS.
03A6 CD 0281          CALL   MSG.OT    ;SEND MESSAGE.
03A9 C9          RET          ;GO HOME.
```

\*\*\*\*\*

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

```
***** ( DISPLAY MESSAGE AND WAIT FOR DRIVE ) *****
```

```
03AA 22 0402 SEL.DR: SHLD MSG.SV ;SAVE MESSAGE ADDRESS.
03AD 2A 0402 ..REPT: LHLD MSG.SV ;LOAD MESSAGE ADDRESS.
03B0 CD 0281 CALL MSG.OT ;ISSUE MESSAGE.
03B3 CD 028F CALL CNS.IN ;GET DRIVE LETTER.
03B6 FE0D CPI CR ;CHECK FOR RESELECT.
03B8 CA 03DE JZ ..EXIT ;IF CR GOTO EXIT AREA.
```

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

```
03BB 32 072D STA DR.LTR ;DRIVE LETTER.
03BE 32 0670 STA LTR.SE ;ILLG DRV LTR.
03C1 D641 SUI 'A' ;GET NUMBER.
03C3 DA 03D5 JC ..ILLG ;ILLEGAL, REPEAT.
03C6 FE04 CPI 04H ;DRIVE A THRU D?
03C8 DA 03E0 JC ..NMBR ;GOOD NUMBER.
03CB D620 SUI 'A'-'A' ;CHECK LOWER CASE.
03CD DA 03D5 JC ..ILLG ;ILLEGAL, REPEAT.
03D0 D604 SUI 04H ;LOWER A THRU D?
03D2 DA 03E0 JC ..NMBR ;LEGAL DRIVE.
```

```
***** ( EXIT TO RESELECT FUNCTION ) *****
```

```
03D5 21 066C ..ILLG: LXI H,MSG.SE ;'SELECT ERROR'
03D8 CD 0281 CALL MSG.OT ;ISSUE MESSAGE.
03DB C3 03AD JMP ..REPT ;REPEAT SELECTION.
```

```
***** ( EXIT TO RESELECT FUNCTION ) *****
```

```
03DE 37 ..EXIT: STC ;SET CARRY.
03DF C9 RET ;RETURN.
```

```
***** ( VALID DRIVE NUMBER ) *****
```

```
03E0 4F ..NMBR: MOV C,A ;DRIVE NMBR TO C.
03E1 1E01 MVI E,LOG.VC ;LOG ON VECTOR.
03E3 2E1B MVI L,BS.DSK ;BIOS SELECT DISK.
03E5 CD 03FB CALL BIOS ;BIOS TRANSFER CALL.
03E8 21 0716 LXI H,MSG.XX ;'TYPE CR WHEN READY'
03EB CD 0281 CALL MSG.OT ;SEND MESSAGE.
03EE CD 028F CALL CNS.IN ;GET CONSOLE CHAR.
03F1 FE0D CPI CR ;CHECK IF CR.
03F3 C2 03AD JNZ ..REPT ;IF NOT, TRY AGAIN.
03F6 A7 ANA A ;CLEAR CARRY.
03F7 C9 RET ;RETURN CALLER.
```

```
*****
```

FORMAT - JADE DOUBLE D  
SMALL ROUTINES AND DATA AREAS

```

;*****
; BIOS TRANSFER ROUTINE
;*****
    
```

```

03F8      3A 0002      BIOS:   LDA      BS.PAG      ;GET HIGH ADDRESS.
03FB      67              MOV      H,A          ;SET IT IN H REG.
03FC      E9              PCHL           ;JUMP TO BIOS.
    
```

```

;*****
; SOFTWARE TRAPS
;*****
    
```

```

03FD      F7          RST.6:  RST      6          ;EXIT FORMAT.
03FE      FF          RST.7:  RST      7          ;EXIT FORMAT.
    
```

```

;*****
; WORKING VARIABLES
;*****
    
```

```

03FF      0000      SYS.WP:  .WORD    0          ;ADDRESS POINTER.
0401      00          RW.OPV:  .BYTE    0          ;READ OR WRITE VECTOR.
0402      0000      MSG.SV:  .WORD    0          ;MESSAGE SAVE ADDRESS.

0404      00          TRK.NO:  .BYTE    0          ;TRACK NUMBER HOLD.
0405      00          SEC.NO:  .BYTE    0          ;SECTOR NUMBER HOLD.
0406      00          F.FLAG:  .BYTE    0          ;FORMAT FLAG (DCM).
    
```

```

;*****
; 3740 SECTOR TRANSLATION
;*****
    
```

```

0407      01070D131905  TKO.SL:  .BYTE    01H,07H,0DH,13H,19H,05H
040D      0B111703090F      .BYTE    0BH,11H,17H,03H,09H,0FH
0413      1502080E141A      .BYTE    15H,02H,08H,0EH,14H,1AH
0419      060C1218040A      .BYTE    06H,0CH,12H,18H,04H,0AH
041F      1016              .BYTE    10H,16H
    
```

```

;*****
    
```

```

;*****
; JADE SINGLE DENSITY - IDENTITY SECTOR *
;*****

```

```

;***** ( JADE IDENTIFICATION )*****

```

```

0421      4A6164652044  IDS.SD: .ASCII  "JADE DD SINGLE DENSITY FORMAT "

```

```

;***** ( DISK PARAMETER BLOCK - CP/M 2.2 )*****

```

```

0441                .LOC      IDS.SD+20H      ;LOCATE CP/M BLOCK.
0441      001A        .WORD      26            ;SECTORS PER TRACK.
0443      03         .BYTE      3             ;BLOCK SHIFT FACTOR.
0444      07         .BYTE      7             ;BLOCK MASK.
0445      00         .BYTE      0             ;NULL MASK.
0446      00F2       .WORD      242           ;DISK SIZE - 1.
0448      003F       .WORD      63            ;DIRECTORY MAXIMUM.
044A      C0         .BYTE      11000000B     ;ALLOC 0.
044B      00         .BYTE      0             ;ALLOC 1.
044C      0010       .WORD      16            ;CHECK SIZE.
044E      0002       .WORD      2             ;TRACK OFFSET.

```

```

;***** ( DCM DISK PARAMETERS )*****

```

```

0451                .LOC      IDS.SD+30H      ;LOCATE DCM BLOCK.
0451      06         .BYTE      6             ;SECTOR STAGGER (TRNS).
0452      02         SD.FLG: .BYTE 00000010B  ;DISKETTE FLAGS.

04A1                .LOC      IDS.SD+SEC.SZ   ;EXTEND FULL SECTOR.

```

```

;*****

```

FORMAT - JADE DOUBLE D

IDENTITY SECTORS

```

;*****
; JADE DOUBLE DENSITY - IDENTITY SECTOR *
;*****

```

```

;***** ( JADE IDENTIFICATION )*****

```

```

04A1      4A6164652044  IDS.DD: .ASCII  "JADE DD DOUBLE DENSITY FORMAT "

```

```

;***** ( DISK PARAMETER BLOCK - CP/M 2.2 )*****

```

```

04C1          .LOC      IDS.DD+20H      ;LOCATE CP/M BLOCK.
04C1      0030      .WORD      48          ;SECTORS PER TRACK.
04C3      04        .BYTE      4          ;BLOCK SHIFT FACTOR.
04C4      0F        .BYTE      00001111B  ;BLOCK MASK.
04C5      01        .BYTE      1          ;NULL MASK.
04C6      00E0      .WORD      224        ;DISK SIZE - 1.
04C8      003F      .WORD      63         ;DIRECTORY MAXIMUM.
04CA      80        .BYTE      10000000B  ;ALLOC 0.
04CB      00        .BYTE      0          ;ALLOC 1.
04CC      0010      .WORD      16         ;CHECK SIZE.
04CE      0002      .WORD      2          ;TRACK OFFSET.

```

```

;***** ( DCM DISK PARAMETERS )*****

```

```

04D1          .LOC      IDS.DD+30H      ;LOCATE DCM BLOCK.
04D1      06        .BYTE      6          ;SECTOR STAGGER (TRNS).
04D2      06        DD.FLG: .BYTE      00000110B  ;DISKETTE FLAGS.

0521          .LOC      IDS.DD+SEC.SZ   ;EXTEND TO FULL SIZE

```

```

;*****

```

```

;*****
0521 MSG.BG: ;CONSOLE SIGN ON *
;*****

0521 0D0A .ASCII [CR][LF]
0523 0D0A444F5542 .ASCII [CR][LF]'DOUBLE D - FORMAT UTILITY'
053E 0D0A56455253 .ASCII [CR][LF]'VERSION 2.2 '
054C 0D0A .ASCII [CR][LF]
054E 0D0A55736520 .ASCIS [CR][LF]'USE CTL-C FOR SYSTEM REBOOT.'

;*****
056C MSG.FL: ;DISPLAY FUNCTION SELECTIONS *
;*****

056C 0D0A .ASCII [CR][LF]
056E 0D0A2A2A2A2A .ASCII [CR][LF]'***** FUNCTIONS LIST *****'
058F 0D0A .ASCII [CR][LF]
0591 0D0A312E2046 .ASCII [CR][LF]'1. FORMAT DOUBLE DENSITY 8" '
05B0 0D0A322E2046 .ASCII [CR][LF]'2. FORMAT SINGLE DENSITY 8" '
05CF 0D0A332E2046 .ASCII [CR][LF]'3. FORMAT STANDARD 3740 8" '
05EE 0D0A342E2046 .ASCII [CR][LF]'4. FORMAT SYSTEM TRACKS ONLY'
060D 0D0A352E2052 .ASCII [CR][LF]'5. READ SYSTEM TRACKS IMAGE '
062F 0D0A362E2057 .ASCII [CR][LF]'6. WRITE SYSTEM TRACKS IMAGE '
0651 0D0A .ASCII [CR][LF]
0653 0D0A454E5445 .ASCIS [CR][LF]'ENTER FUNCTION NUMBER: '

;*****
066C MSG.SE: ;SELECTION ERROR *
;*****

066C 0D0A0D0A .ASCII [CR][LF][CR][LF]
0670 20 LTR.SE: .BYTE ' '
0671 206973204E4F .ASCIS ' IS NOT A VALID SELECTION '

;*****
```

FORMAT - JADE DOUBLE D

TEXT MESSAGES

```

068B          ;*****
MSG.FD:      ;FORMAT ON DISK DRIVE          *
          ;*****

068B  0D0A0D0A .ASCII [CR][LF][CR][LF]
068F  577269746520 .ASCIS 'WRITE FORMAT ON DRIVE (CR TO RESELECT): '

          ;*****
06B7          MSG.RS:      ;READ SYSTEM ON DRIVE          *
          ;*****

06B7  0D0A0D0A .ASCII [CR][LF][CR][LF]
06BB  524541442053 .ASCIS 'READ SYSTEM FROM DRIVE: '

          ;*****
06D3          MSG.WS:      ;WRITE SYSTEM ON DRIVE          *
          ;*****

06D3  0D0A0D0A .ASCII [CR][LF][CR][LF]
06D7  575249544520 .ASCIS 'WRITE SYSTEM TO DRIVE (CR TO RESELECT): '

          ;*****
06FF          MSG.NC:      ;TRANSFER INCOMPLETE          *
          ;*****

06FF  0D0A0D0A .ASCII [CR][LF][CR][LF]
0703  5452414E5346 .ASCIS 'TRANSFER INCOMPLETE'

          ;*****
0716          MSG.XX:      ;TYPE CR WHEN DRIVE READY          *
          ;*****

0716  0D0A0D0A .ASCII [CR][LF][CR][LF]
071A  547970652043 .ASCII 'TYPE CR WHEN DRIVE '
072D  20 DR.LTR: .BYTE ' '
072E  205245414459 .ASCIS ' READY. '

          ;*****

```



```
*****  
; 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 PORT.  
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,0FEH ;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,0FCH ;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,0FBH ;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,0F7H ;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]
```

```
***** ( 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 NUMBR.  
XRA     C           ;INVERT (1791-01).  
OUT     WD.DTA      ;WRITE DATA PORT.  
INX     H           ;INC SEC-NMBR FNTR.  
.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'
```

\*\*\*\*\*

0800	464F524D4154	FORMAT	FT3740
0807	53	DENSITY	SINGLE
0808	21 17B5	SECTORS	SS3740,26
080D	0628	BG3740:	BLOCK 40, ONES
0818	0606		BLOCK 6, ZEROS
0823	DB80		WRITE INDEX.MARK
082A	061A		BLOCK 26, ONES
0835	0606	RP3740:	BLOCK 6, ZEROS
0840	DB80		WRITE ID.MARK
0847	DB80		WRITE TRACK.NO
084D	DB80		WRITE SIDE.NO
0854	DB80		WRITE SECTOR.NO
085B	DB80		WRITE SECTOR.SIZE,128
0862	DB80		WRITE CRC
0869	060B		BLOCK 11, ONES
0874	0606		BLOCK 6, ZEROS
087F	DB80		WRITE DATA.MARK
0886	0680		BLOCK 128,0E5H
0891	DB80		WRITE CRC
0898	061B		BLOCK 27, ONES
08A3	1D		REPEAT RP3740
08A7	21 0000	ENDING	ONES
08B5	010203040506	SS3740:	.BYTE 1, 2, 3, 4, 5, 6, 7, 8, 9,10
08BF	0B0C0D0E0F10		.BYTE 11,12,13,14,15,16,17,18,19,20
08C9	15161718191A		.BYTE 21,22,23,24,25,26

```
0900 464F524D4154          FORMAT FTJ48D
0907 44                    DENSITY DOUBLE
0908 21 17AE              SECTORS SSJ48D,48

090D 0650                BGJ48D: BLOCK 80,04EH
0918 060C                RPJ48D: BLOCK 12,ZEROS
0923 0603                BLOCK 3,OF5H
092E DB80                WRITE ID.MARK
0935 DB80                WRITE TRACK.NO
093B DB80                WRITE SIDE.NO
0942 DB80                WRITE SECTOR.NO
0949 DB80                WRITE SECTOR.SIZE,128
0950 DB80                WRITE CRC
0957 0616                BLOCK 22,04EH
0962 060C                BLOCK 12,ZEROS
096D 0603                BLOCK 3,OF5H
0978 DB80                WRITE DATA.MARK
097F 0680                BLOCK 128,0E5H
098A DB80                WRITE CRC
0991 0616                BLOCK 22,04EH
099C 1D                  REPEAT RPJ48D

09A0 21 0000              ENDING ONES

09AE 0105090D1115 SSJ48D: .BYTE 1, 5, 9,13,17,21
09B4 191D2125292D        .BYTE 25,29,33,37,41,45
09BA 02060A0E1216        .BYTE 2, 6,10,14,18,22
09C0 1A1E22262A2E        .BYTE 26,30,34,38,42,46
09C6 03070B0F1317        .BYTE 3, 7,11,15,19,23
09CC 1E1F23272B2F        .BYTE 27,31,35,39,43,47
09D2 04080C101418        .BYTE 4, 8,12,16,20,24
09D8 1C2024282C30        .BYTE 28,32,36,40,44,48

.END
```

FORMAT - JADE DOUBLE D

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

BDOS	0005	BEGIN	0100	BG3740	080D	BGJ48D	090D
BIOS	03F8	BS.DMA	0024	BS.DSK	001B	BS.FMT	0033
BS.PAG	0002	BS.RDS	0027	BS.SEC	0021	BS.TRK	001E
BS.WRS	002A	CNS.IN	028F	CNS.OT	029C	CNS.RD	0001
CNS.WR	0002	CNTL.C	0003	CR	000D	DD.FLG	04D2
DR.LTR	072D	FMT.DD	02EA	FMT.EA	1700	FMT.SD	02E4
FT3740	0800	FTJ48D	0900	FUN.1	017E	FUN.2	01E3
FUN.3	01B8	FUN.4	022B	FUN.5	0259	FUN.6	026D
FUN.BG	0141	F.FLAG	0406	HXB.OT	02AA	HXN.OT	02B3
HXW.OT	02A5	IDS.DD	04A1	IDS.SD	0421	ID.SEC	0001
INIT	0138	LF	000A	LOG.VC	0001	LTR.SE	0670
MSG.BG	0521	MSG.FD	068B	MSG.FL	056C	MSG.NC	06FF
MSG.OT	0281	MSG.RS	06B7	MSG.SE	066C	MSG.SV	0402
MSG.WS	06D3	MSG.XX	0716	NMBR	0016	OFFSET	0E00
ONES	00FF	REBOOT	0000	RP3740	0835	RPJ48D	0918
RST.6	03FD	RST.7	03FE	RW.OPV	0401	SD.FLG	0452
SEC.CD	0000	SEC.NO	0405	SEC.SZ	0080	SEL.DR	03AA
SS3740	08B5	SSJ48D	09AE	ST.DMA	02ED	SYS.WP	03FF
TKO.FS	0001	TKO.LC	0F80	TKO.LP	1000	TKO.LS	001A
TKO.NS	001A	TKO.SL	0407	TKO.SZ	0D00	TK1.FS	0001
TK1.LP	1D00	TK1.LS	0030	TPA	0100	TRK.0	0000
TRK.1	0001	TRK.2	0002	TRK.76	004C	TRK.NO	0404
TRNSFR	0314	WD.DTA	0007	WD.TRK	0005	WRT.ID	02C1
XP.DSH	0080	ZEROS	0000				

^CB:  
B>DDT  
DDT VERS 1.4  
-F100,1000,00  
-IFORMAT.HEX  
-R

The following is the hex code generated  
by the FORMAT INJECTION MODULES.

NEXT FC  
09DE 0000  
-D800,8FF

0800 46 4F 52 4D 41 54 21 53 21 B5 17 1E 1A 06 28 DB FORMAT!S!.....(  
0810 80 3E FF A9 D3 07 10 F7 06 06 DB 80 3E 00 A9 D3 .>.....>...  
0820 07 10 F7 DB 80 3E FC A9 D3 07 06 1A DB 80 3E FF .....>.....>.  
0830 A9 D3 07 10 F7 06 06 DB 80 3E 00 A9 D3 07 10 F7 .....>.....  
0840 DB 80 3E FE A9 D3 07 DB 80 DB 05 D3 07 DB 80 3E ..>.....>...  
0850 00 A9 D3 07 DB 80 7E A9 D3 07 23 DB 80 3E 00 A9 .....^...#...>..  
0860 D3 07 DB 80 3E F7 A9 D3 07 06 0B DB 80 3E FF A9 .....>.....>..  
0870 D3 07 10 F7 06 06 DB 80 3E 00 A9 D3 07 10 F7 DB .....>.....  
0880 80 3E FB A9 D3 07 06 80 DB 80 3E E5 A9 D3 07 10 .>.....>.....  
0890 F7 DB 80 3E F7 A9 D3 07 06 1B DB 80 3E FF A9 D3 ...>.....>...  
08A0 07 10 F7 1D C2 35 17 21 00 00 DB 80 3E FF A9 D3 .....5.!.....>...  
08B0 07 23 C3 AA 17 01 02 03 04 05 06 07 08 09 0A 0B .#.....  
08C0 0C 0D 0E 0F 10 11 12 13 14 15 16 17 18 19 1A 00 .....  
08D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....  
08E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....  
08F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....  
-D900,9FF

0900 46 4F 52 4D 41 54 21 44 21 AE 17 1E 30 06 50 DB FORMAT!D!...O.P.  
0910 80 3E 4E A9 D3 07 10 F7 06 0C DB 80 3E 00 A9 D3 .>N.....>...  
0920 07 10 F7 06 03 DB 80 3E F5 A9 D3 07 10 F7 DB 80 .....>.....  
0930 3E FE A9 D3 07 DB 80 DB 05 D3 07 DB 80 3E 00 A9 >.....>...  
0940 D3 07 DB 80 7E A9 D3 07 23 DB 80 3E 00 A9 D3 07 .....^...#...>...  
0950 DB 80 3E F7 A9 D3 07 06 16 DB 80 3E 4E A9 D3 07 ..>.....>N...  
0960 10 F7 06 0C DB 80 3E 00 A9 D3 07 10 F7 06 03 DB .....>.....  
0970 80 3E F5 A9 D3 07 10 F7 DB 80 3E FB A9 D3 07 06 .>.....>.....  
0980 80 DB 80 3E E5 A9 D3 07 10 F7 DB 80 3E F7 A9 D3 ...>.....>...  
0990 07 06 16 DB 80 3E 4E A9 D3 07 10 F7 1D C2 18 17 .....>N.....  
09A0 21 00 00 DB 80 3E FF A9 D3 07 23 C3 A3 17 01 05 !.....>...#.....  
09B0 09 0D 11 15 19 1D 21 25 29 2D 02 06 0A 0E 12 16 .....!%).....  
09C0 1A 1E 22 26 2A 2E 03 07 0B 0F 13 17 1B 1F 23 27 .."&\*.....#'  
09D0 2B 2F 04 08 0C 10 14 18 1C 20 24 28 2C 30 00 00 +/.....\$(,0..  
09E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....  
09F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....  
-



```
*****  
;  
;          PROGRAM ID:      BIOS LOADER TRANSIENT      *  
;  
*****  
;  
;          PROPERTY OF:    JADE COMPUTER PRODUCTS      *  
;                          4901 W. ROSECRA NS BLVD.    *  
;                          HAWTHORNE, CALIFORNIA      *  
;                          90250,  U.S.A.             *  
;  
*****  
;  
;          VERSION:        2.2                          *  
;  
*****  
; THE BIOS LOADER IS READ INTO THE DCM SECTOR BUFFER *  
; AFTER THE DCM PROGRAM HAS INITIALIZE. THIS PROGRAM *  
; THEN READS THE BIOS MODULE (1K) INTO BANK 1. THE  *  
; COMMAND BLOCK IN DCM IS SET TO INDICATE THE BIOS  *  
; MODULE SIZE AND ITS LOAD ADDRESS. THE BIOS LOADER *  
; PROGRAM IS GENERATED BY MOVCPM.COM AS THE COLD  *  
; START LOADER (900-97F HEX).                        *  
***** SK *****
```

\*\*\*\*\*  
; CONTROLLER PORT ASSIGNMENTS  
\*\*\*\*\*

0000 BL.STS == 000H ;BOARD STATUS  
0000 BL.CTL == 000H ;BOARD CONTROLS  
0004 WD.CMD == 004H ;1791 COMMAND REGISTER  
0004 WD.STS == 004H ;1791 STATUS REGISTER  
0006 WD.SEC == 006H ;1791 SECTOR REGISTER  
0007 WD.DTA == 007H ;1791 DATA REGISTER  
0010 XP.MTO == 010H ;MOTOR TIME OUT  
0040 XP.MTX == 040H ;MOTOR TIME EXTEND  
0080 XP.DSH == 080H ;DATA SYNC HOLD

\*\*\*\*\*  
; 1791-01 COMMAND AND MASK.  
\*\*\*\*\*

0080 DC.RDS == 1000000B ;READ SECTOR.  
009C DM.RER == 10011100B ;READ ERROR MASK.

\*\*\*\*\*  
; SYSTEM ASSIGNMENTS  
\*\*\*\*\*

0014 NMBR.K == 20 ;SYSTEM SIZE IN K.  
0400 LNG.1K == 1024 ;TOTAL BYTES IN 1K.  
5000 CPM.SZ == NMBR.K\*LNG.1K ;TOTAL SYSTEM BYTES.  
0600 BIOS.S == LNG.1K\*3/2 ;BIOS ALLOCATED SIZE.  
4A00 BIOS.A == CPM.SZ-BIOS.S ;BIOS LOAD ADDRESS.

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

1000 BANK.0 == 1000H ;LOWER BANK ADDRESS.  
0400 BANK.L == 0400H ;1K BANK LENGTH.  
1400 BANK.1 == BANK.0+BANK.L ;UPPER BANK ADDRESS.  
1370 IO.BLK == BANK.0+0370H ;I/O BLOCK ADDRESS.  
1377 CB.STS == IO.BLK+0007H ;COMMAND STATUS BYTE.  
1378 CW.LAD == IO.BLK+0008H ;BIOS LOAD ADDR LOC.  
137A CW.LNG == IO.BLK+000AH ;BIOS LOAD LENGTH LOC.  
1380 SEC.BF == BANK.0+0380H ;SECTOR BUFFER AREA.

\*\*\*\*\*  
; BIOS PROGRAM LINKAGE.  
\*\*\*\*\*

0004 SEC.BG == 4 ;FIRST BIOS SECTOR.  
0008 SEC.NM == 8 ;NUMBER OF SECTORS.  
000B SEC.EX == SEC.BG+SEC.NM-1 ;LAST BIOS SECTOR.

\*\*\*\*\*

```

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

```

```

          .FABS
          .PHEX
          .XLINK
1380      .LOC      SEC.BF      ;PROGRAM START POINT

```

```

;*****
; PAUSE MACRO.  ALLOWS 1791 TO DIGEST INSTRUCTIONS!
;*****

```

```

.DEFINE PAUSE = [
          XTHL
          XTHL
          XTHL
          XTHL]

```

```

;*****
; INITIALIZE BIOS READ OPERATION
;*****

```

```

1380      21 0400      BEGIN: LXI      H,LNG.1K      ;BIOS LOAD LENGTH.
1383      22 137A      SHLD     CW.LNG      ;LOAD LENGTH SET.
1386      21 4A00      LXI      H,BIOS.A      ;BIOS SYSTEM ADDR.
1389      22 1378      SHLD     CW.LAD      ;LOAD ADDRESS SET.
138C      21 1400      LXI      H,BANK.1      ;BIOS LOAD POINT.

```

```

;*****
; SET-UP FOR EACH READ SECTOR COMMAND
;*****

```

```

138F      FD21 13AB      RD.SEC: LXI      Y,RD.TST      ;SET NMI VECTOR.
1393      3A 13C7      LDA      SECTOR      ;FIRST BIOS SECTOR.
1396      A9          XRA      C      ;INVERT (1791-01).
1397      D306      OUT      WD.SEC      ;SET 1791 SEC REG.
1399      3E80      MVI      A,DC.RDS      ;READ SECTOR CMND.
139B      A9          XRA      C      ;INVERT (1791-01).
139C      D304      OUT      WD.CMD      ;ISSUE 1791 COMMAND.
139E      E3          PAUSE      ;ALLOW 1791 TO SETTLE.

```

```

;*****
; READ SECTOR OPERATION
;*****

```

```

13A2      DB80      RD.BYT: IN      XP.DSH      ;WAIT FOR DATA.
13A4      DB07      IN      WD.DTA      ;INPUT INV DATA.
13A6      A9          XRA      C      ;INVERT (1791-01).
13A7      77          MOV      M,A      ;STORE DCM BYTE.
13A8      23          INX     H      ;INCREMENT POINTER.
13A9      18F7      JMPR     RD.BYT      ;REPEAT OPERATION.

```

```

;*****

```

```

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

```

```

4A00    C3 4A83    JMP     'INIT           ;COLD START ENTRY
4A03    C3 4C8C    JMP     WARM           ;RELOAD CCP/BDOS
4A06    C3 4C80    JMP     CNS.CK        ;GET CONSOLE STATUS
4A09    C3 4C8A    JMP     CNS.IN        ;CONSOLE INPUT
4A0C    C3 4C95    JMP     CNS.OT        ;CONSOLE OUTPUT
4A0F    C3 4C88    JMP     LIST          ;PRINTER OUTPUT
4A12    C3 4C87    JMP     PUNCH         ;PUNCH OUTPUT
4A15    C3 4CA4    JMP     READER        ;READER INPUT
4A18    C3 4D8C    JMP     HOME          ;HOME SELECTED DRIVE
4A1B    C3 4C41    JMP     SELDSK        ;SELECT DISK DRIVE
4A1E    C3 4C60    JMP     SETTRK        ;SET TRACK NUMBER
4A21    C3 4C65    JMP     SETSEC        ;SET SECTOR NUMBER
4A24    C3 4C6A    JMP     SETDMA        ;SET TRANSFER ADDRESS
4A27    C3 4B03    JMP     DISKRD        ;PERFORM DISK READ
4A2A    C3 4B23    JMP     DISKWR        ;PERFORM DISK WRITE
4A2D    C3 4CAB    JMP     LISTST        ;RETURN LIST STAT
4A30    C3 4C70    JMP     SECTRN        ;TRANSLATE SECTOR
4A33    C3 4B50    JMP     FORMAT        ;FORMAT A TRACK

```

```

;*****
; RESERVE DRIVE PARAMETER BLOCKS
;*****

```

```

000F    DPB.SZ ==    15           ;SIZE IS 15 BYTES.
4A36    D0.DPB: .BLKB DPB.SZ      ;RESERVES 15 BYTES.
4A43    D1.DPB: .BLKB DPB.SZ      ;FOR EACH DISK DRIVE.
4A54    D2.DPB: .BLKB DPB.SZ      ;DRIVE A THRU DRIVE D.
4A63    D3.DPB: .BLKB DPB.SZ

```

```

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

```

```

4A72    00    BT.CMD: .BYTE 0           ;DCM COMMAND.
4A73    00    BT.DRV: .BYTE 0           ;DRIVE NUMBER.
4A74    00    BT.TRK: .BYTE 0           ;TRACK NUMBER.
4A75    00    BT.SEC: .BYTE 0           ;SECTOR NUMBER.
4A76    00    BT.SP0: .BYTE 0           ;SPARE BYTE 0.
4A77    00    BT.SP1: .BYTE 0           ;SPARE BYTE 1.
4A78    00    BT.MOD: .BYTE 0000000B ;MODE CONTROLS.
4A79    00    BT.STS: .BYTE 0           ;COMMAND STATUS.
4A7A    0000  BT.LAD: .WORD 0           ;LOAD ADDRESS.
4A7C    0000  BT.LNG: .WORD 0           ;LOAD LENGTH.

4A7E    0000  BT.DMA: .WORD 0           ;SYSTEM TRANSFER ADDR.
4A80    0000  DT.PTR: .WORD 0           ;DRIVE TABLE POINTER.
4A82    00    LOG.RQ: .BYTE 0           ;LOG ON REQUEST REG.

```

```

;*****

```

TDL Z80 CP/M DISK ASSEMBLER VERSION 2.21  
BIOS LOADER TRANSIENT - JADE DD CONTROLLER  
+++++ SYMBOL TABLE +++++

BANK.0	1000	BANK.1	1400	BANK.L	0400	BEGIN	1380
BIOS.A	4A00	BIOS.S	0600	BL.CTL	0000	BL.STS	0000
CB.STS	1377	CPM.SZ	5000	CW.LAD	1378	CW.LNG	137A
DC.RDS	0080	DM.RER	009C	ERRORS	13BC	FINISH	13C5
I0.BLK	1370	LNG.1K	0400	NMBR.K	0014	RD.BYT	13A2
RD.SEC	138F	RD.TST	13AB	SECTOR	13C7	SEC.BF	1380
SEC.BG	0004	SEC.EX	000B	SEC.NM	0008	WD.CMD	0004
WD.DTA	0007	WD.SEC	0006	WD.STS	0004	XP.DSH	0080
XP.MTU	0010	XP.MTX	0040				



```
*****  
;  
; PROGRAM ID: SYSTEM BOOTSTRAP DRIVER *  
;  
*****  
;  
; PROPERTY OF: JADE COMPUTER PRODUCTS *  
; 4901 W. ROSECRAWS BLVD. *  
; HAWTHORNE, CALIFORNIA *  
; 90250, U.S.A. *  
;  
*****  
;  
; VERSION: 2.2 *  
;  
*****  
; THE SYSTEM BOOTSTRAP DRIVER IS ONE OF TWO MODULES *  
; THAT MAKE UP THE SYSTEM RESIDENT BOOTSTRAP. THIS *  
; MODULE IS TO BE EXECUTED BY THE SYSTEM PROCESSOR. *  
; DURING EXECUTION, THIS MODULE PERFORMS A BLOCK MOVE *  
; OF THE SECOND MODULE (BOOT INJECTION MODULE) INTO *  
; THE DOUBLE D CONTROLLER MEMORY. A SUCCESSFUL BOOT *  
; OPERATION BY THE DOUBLE D WILL LEAVE DCM IN BANK 0 *  
; AND BIOS IN BANK 1. THE REMAINDER OF THIS MODULE *  
; THEN MOVES THE BIOS IMAGE TO THE PROPER SYSTEM *  
; ADDRESS AND JUMPS TO THE BIOS COLD START ENTRY. *  
***** SK *****
```

\*\*\*\*\*  
; DOUBLE D HARDWARE PARAMETERS. PLEASE NOTE THIS \*  
; SECTION CONTAINS CONDITIONAL STATEMENTS. \*  
\*\*\*\*\*

0043 D.PORT == 043H ;DOUBLE D PORT ADDRESS.  
0001 TRUE == 1 ;TRUE IS A ONE.  
0000 FALSE == 0 ;FALSE IS A ZERO.  
0001 REV.B == TRUE ;SET TRUE FOR REV B BOARDS.  
0000 REV.C == FALSE ;SET TRUE FOR REV C BOARDS.  
0000 MA10 == FALSE ;TRUE IF MA10 JUMPED (REV-B).

.IFG REV.B, [  
0002 DS.HLT == 002H ;STATUS PORT HALT INDICATOR.  
000C DS.ASW == 00CH ;STATUS PORT ADDR SW MASK.  
E400 D.BASE = 0E400H ;SYSTEM WINDOW BASE ADDRESS]

.IFG MA10, [  
D.BASE == 0E000H ;SYSTEM WINDOW BASE ADDRESS]

.IFG REV.C, [  
DS.HLT == 001H ;STATUS PORT HALT INDICATOR.  
DS.ASW == 00EH ;STATUS PORT ADDR SW MASK.  
D.BASE == 0E000H ;SYSTEM WINDOW BASE ADDRESS]

\*\*\*\*\*  
; BOOTSTRAP INJECTION MODULE PARAMETERS (ALTERABLE) \*  
\*\*\*\*\*

0200 IM.ADR == 0200H ;BOOT INJECTION MODULE ADDRESS.  
00C0 IM.SZ == 00C0H ;BOOT INJECTION MODULE SIZE.

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

0080 BSTACK == 0080H ;BOOTSTRAP TOP OF STACK.  
0040 D.ADDR == 0040H ;DOUBLE D ADDRESS POINTER.  
0377 BL.DCS == 0377H ;DCM DISK CONTROLLER STATUS.  
0378 BL.ADR == 0378H ;DCM LOAD AND JUMP ADDRESS.  
037A BL.BSZ == 037AH ;DCM BLOCK LOAD SIZE.

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

0080 DC.BGN == 080H ;RESET Z80A AND EXECUTE.  
0001 DC.MRQ == 001H ;REQUEST MEMORY WINDOW.  
0000 DC.MRT == 000H ;RELEASE MEMORY WINDOW.  
0001 DC.MB0 == 001H ;SELECT MEMORY BANK 0.  
0003 DC.MB1 == 003H ;SELECT MEMORY BANK 1.  
0002 DC.EXC == 002H ;ISSUE DOUBLE D INTERRUPT.

\*\*\*\*\*



```

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

```

```

                                .I8080      ;USE 8080 INSTRUCTION SUBSET.
                                .PABS       ;ASSEMBLE ABSOLUTE ADDRESS.
                                .PHEX      ;GENERATE INTEL HEX FORMAT.
                                .XLINK     ;SUPRESS LINKAGE OUTPUT.
0100      .LOC      0100H      ;MODULE ADDRESS (ALTERABLE).

```

```

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

```

```

0100      31 0080      BEGIN: LXI      SP,BSTACK      ;SET STACK POINTER.
0103      DB43        IN        D,PORT          ;INPUT STATUS PORT.
0105      E60C        ANI      DS.ASW          ;MASK FOR ADDR SWS.
0107      07          RLC                    ;POSITION BITS.
0108      F6E4        ORI      D,BASE>8      ;OR IN BASE ADDR.
010A      67          MOV      H,A            ;HIGH BYTE VALUE.
010B      2E00        MVI      L,0           ;LOW BYTE VALUE.
010D      22 0040      SHLD     D,ADDR         ;STORE THE ADDRESS

```

```

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

```

```

0110      3E01        INJECT: MVI     A,DC.MBO    ;REQUEST DD MEM BANK 0.
0112      D343        OUT      D,PORT          ;ISSUE COMMAND.
0114      01 00C0      LXI      B,IM.SZE       ;INJECTION SIZE.
0117      EB          XCHG                    ;D.ADDR HL TO DE.

0118      21 0200      LXI      H,IM.ADR      ;INJECTION MODULE.
011B      CD 0159      CALL     BLOCK         ;BLOCK MOVE.

```

```

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

```

```

011E      3E80        MVI      A,DC.BGN       ;BEGIN DD PROCESSOR.
0120      D343        OUT      D,PORT          ;ISSUE COMMAND.
0122      E3          XTHL                    ;ALLOW DOUBLE D TIME
0123      E3          XTHL                    ;TO START UP.

```

```

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

```

```

0124      DB43        WAIT:   IN        D,PORT          ;INPUT DD STATUS.
0126      E602        ANI      DS.HLT         ;TEST HALT* STATUS.
0128      C2 0124      JNZ      WAIT          ;WAIT TILL HALTED.

```

```

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

```

012B 3E01 MVI A,DC.MRQ ;REQUEST MEM (BANK 0).
012D D343 OUT D.PORT ;ISSUE COMMAND.
  
```

```

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

```

012F 2A 0040 LHLD D.ADDR ;CONTROLLER ADDRESS.
0132 11 0377 LXI D,BL.DCS ;ERROR CODE OFFSET.
0135 19 DAD D ;SET HL POINTER.
0136 7E MOV A,M ;GET ERROR CODE.
0137 A7 ANA A ;TEST REGISTER.
0138 C2 0166 JNZ BAD.LD ;BAD LOAD.
  
```

```

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

```

013B 2A 0040 LHLD D.ADDR ;CONTROLLER ADDRESS.
013E 11 0378 LXI D,BL.ADR ;LOAD ADDRESS PNTR.
0141 19 DAD D ;SET HL POINTER.
0142 5E MOV E,M ;LOW ORDER ADDR.
0143 23 INX H ;INCREMENT HL.
0144 56 MOV D,M ;HIGH ORDER ADDR.
0145 23 INX H ;REQUIRES BL.BSZ NEXT.
0146 4E MOV C,M ;LOW ORDER LENGTH.
0147 23 INX H ;INCREMENT HL.
0148 46 MOV B,M ;HIGH ORDER LENGTH.
0149 D5 PUSH D ;USE AS JUMP ADDR.
014A 3E03 MVI A,DC.MB1 ;SWITCH TO MEM BANK 1.
014C D343 OUT D.PORT ;ISSUE COMMAND.
014E 2A 0040 LHLD D.ADDR ;DOUBLE D MEM ADDRESS.
0151 CD 0159 CALL BLOCK ;MOVE BIOS MODULE.
  
```

```

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

```

0154 3E01 MVI A,DC.MB0 ;SWITCH TO BANK.0
0156 D343 OUT D.PORT ;ISSUE COMMAND.
0158 C9 RET ;GOTO BIOS COLD ENTRY.
  
```

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

```
0159 7E          BLOCK: MOV      A,M          ;GET BYTE.  
015A 23          INX      H              ;INC POINTER  
015B EB          XCHG                     ;GET DESTINATION.  
015C 77          MOV      M,A          ;PUT BYTE.  
015D 23          INX      H              ;INC POINTER  
015E EB          XCHG                     ;GET SOURCE.  
015F 0B          DCX      B              ;ONE LESS TO DO.  
0160 78          MOV      A,B          ;GET HI COUNT.  
0161 B1          ORA      C              ;GET LO COUNT.  
0162 C2 0159     JNZ      BLOCK          ;FINISH LOADING.  
0163 C9          RET
```

```
*****  
; ERROR HAS BEEN DETECTED *  
*****
```

```
0166 21 016D     BAD.LD: LXI      H,ER.MSG      ;ERROR MESSAGE  
0167 CD 0196     CALL     MSG.OT      ;DISPLAY IT.  
016C 76          HLT
```

```
016D 0D0A0A535953 ER.MSG: .ASCIS [CR][LF][LF]"SYSTEM BOOT LOAD ERROR"
```

```
*****
```

```

;*****
;  CONSOLE LINKAGE DEFINITIONS
;*****
    
```

```

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

000A      LF      ==      00AH      ;ASCII LINE FEED
000D      CR      ==      00DH      ;CARRIAGE RETURN
    
```

```

;*****
;  CONSOLE OUTPUT
;*****
    
```

```

0186      F5      CNS.OT: PUSH      PSW      ;SAVE CHARACTER
0187      DB00    ..WAIT: IN        CNO.SP   ;INPUT STATUS
0189      EE00                XRI        CNO.SI ;ADJUST POLARITY
018B      E604                ANI        CNO.SB ;MASK STATUS BIT
018D      CA 0187            JZ         ..WAIT ;TRY AGAIN
0190      F1                POP        PSW    ;RESTORE CHARACTER
0191      E67F                ANI        07FH  ;7 BIT ASCII
0193      D301                OUT        CNO.DP ;SEND CHARACTER
0195      C9                RET
    
```

```

;*****
;  DISPLAY MESSAGE SUBROUTINE
;*****
    
```

```

0196      F5      MSG.OT: PUSH      PSW      ;SAVE CALLER FLAGS.
0197      7E      ..REPT: MOV        A,M     ;LOAD CHARACTER.
0198      CD 0186            CALL       CNS.OT ;CONSOLE OUTPUT.
019B      7E                MOV        A,M     ;SAME CAHRACTER.
019C      23                INX        H       ;INCREMENT POINTER.
019D      E680                ANI        080H  ;TEST SIGN BIT.
019F      CA 0197            JZ         ..REPT ;ANOTHER CHARACTER.
01A2      F1                POP        PSW    ;RESTORE FLAGS.
01A3      C9                RET         ;RETURN TO CALLER.
    
```

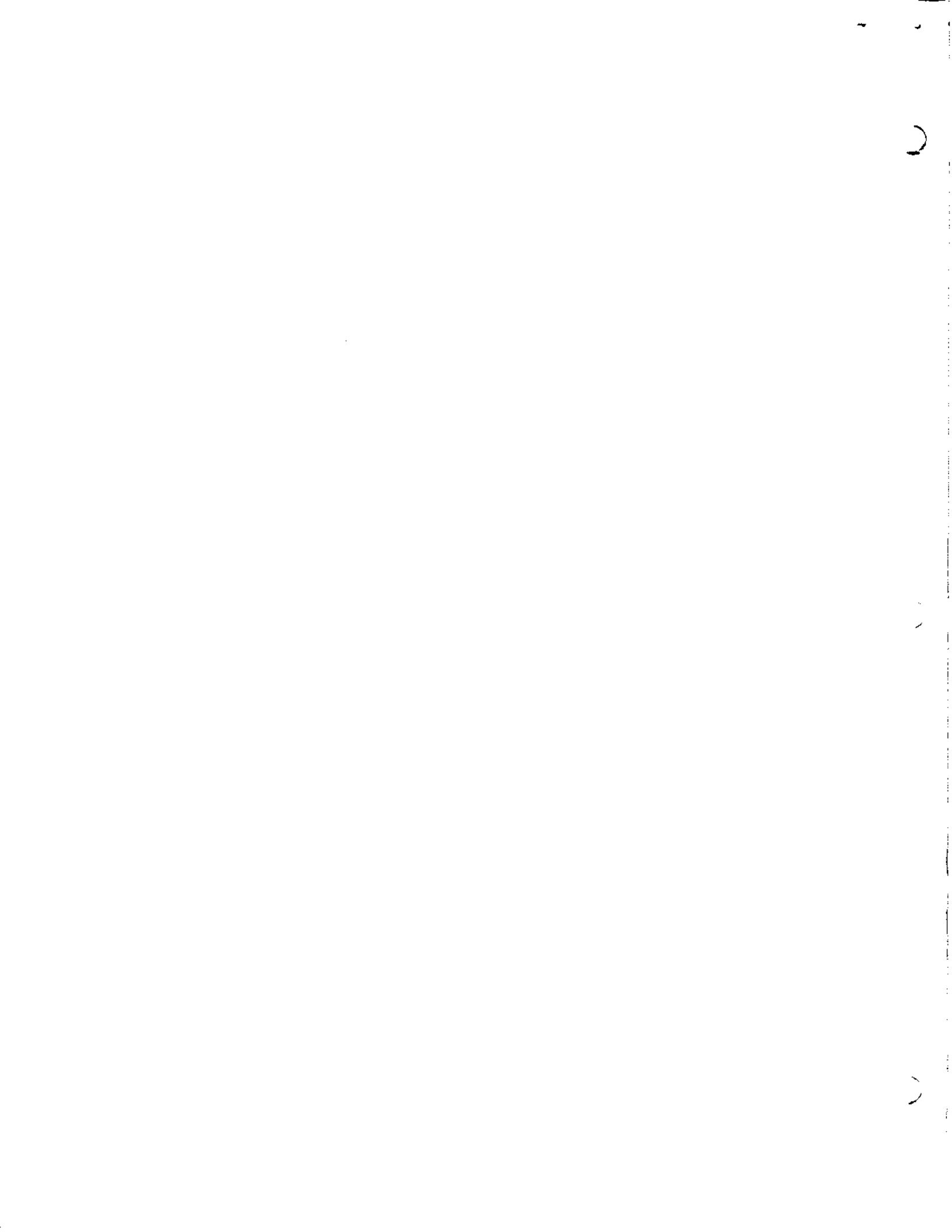
```

;*****
    
```

```

0100                .END      BEGIN
    
```

BAD.LD 0166	BEGIN 0100	BLOCK 0159	BL.ADR 0378
BL.BSZ 037A	BL.DCS 0377	BSTACK 0080	CNO.DP 0001
CNO.SB 0004	CNO.SI 0000	CNO.SP 0000	CNS.OT 0186
CR 000D	DC.BGN 0080	DC.EXC 0002	DC.MB0 0001
DC.MB1 0003	DC.MRQ 0001	DC.MRT 0000	DS.ASW 000C
DS.HLT 0002	D.ADDR 0040	D.BASE E400	D.PORT 0043
ER.MSG 016D	FALSE 0000	IM.ADR 0200	IM.SZE 00C0
INJECT 0110	LF 000A	MA10 0000	MSG.OT 0196
REV.B 0001	REV.C 0000	TRUE 0001	WAIT 0124





```

*****
;
; PROGRAM ID: SYSTEM BOOTSTRAP DRIVER
;
*****
;
; PROPERTY OF: JADE COMPUTER PRODUCTS
; 4901 W. ROSECRANS BLVD.
; HAWTHORNE, CALIFORNIA
; 90250, U.S.A.
;
*****
;
; VERSION: 2.2
;
*****
; THE SYSTEM BOOTSTRAP DRIVER IS ONE OF TWO MODULES
; THAT MAKE UP THE SYSTEM RESIDENT BOOTSTRAP. THIS
; MODULE IS TO BE EXECUTED BY THE SYSTEM PROCESSOR.
; DURING EXECUTION, THIS MODULE PERFORMS A BLOCK MOVE
; OF THE SECOND MODULE (BOOT INJECTION MODULE) INTO
; THE DOUBLE D CONTROLLER MEMORY. A SUCCESSFUL BOOT
; OPERATION BY THE DOUBLE D WILL LEAVE DCM IN BANK 0
; AND BIOS IN BANK 1. THE REMAINDER OF THIS MODULE
; THEN MOVES THE BIOS IMAGE TO THE PROPER SYSTEM
; ADDRESS AND JUMPS TO THE BIOS COLD START ENTRY.
; ***** SK ***
;
; *****
; DOUBLE D HARDWARE PARAMETERS. PLEASE NOTE THIS
; SECTION CONTAINS CONDITIONAL STATEMENTS.
; *****
;
0043 = D$PORT EQU 043H ;DOUBLE D PORT ADDRESS.
0001 = TRUE EQU 1 ;TRUE IS A ONE.
0000 = FALSE EQU 0 ;FALSE IS A ZERO.
0001 = REV$B EQU TRUE ;SET TRUE FOR REV B BOARDS.
0000 = REV$C EQU FALSE ;SET TRUE FOR REV C BOARDS.
0000 = MA10 EQU FALSE ;TRUE IF MA10 JUMPED (REV-B).
;
; IF REV$B
0002 = DS$HLT EQU 002H ;STATUS PORT HALT INDICATOR.
000C = DS$ASW EQU 00CH ;STATUS PORT ADDR SW MASK.
D$BASE SET 0E400H ;SYSTEM WINDOW BASE ADDRESS
ENDIF
;
; IF MA10
D$BASE EQU 0E000H ;SYSTEM WINDOW BASE ADDRESS
ENDIFF
;
; IF REV$C
DS$HLT EQU 001H ;STATUS PORT HALT INDICATOR.
DS$ASW EQU 00EH ;STATUS PORT ADDR SW MASK.
D$BASE EQU 0E000H ;SYSTEM WINDOW BASE ADDRESS
ENDIF
;
; *****
; BOOTSTRAP INJECTION MODULE PARAMETERS
; *****
;
1000 = IM$ADR EQU 1000H ;BOOT INJECTION MODULE ADDRESS.
000C = IM$SIZE EQU 000CH ;BOOT INJECTION MODULE SIZE.
;

```

```

; BOOTSTRAP LINKAGE ADDRESS.
;*****
;
0080 = BSTACK EQU 0080H ;BOOTSTRAP TOP OF STACK.
0040 = D$ADDR EQU 0040H ;DOUBLE D ADDRESS POINTER.
0377 = BL$DCS EQU 0377H ;DCM DISK CONTROLLER STATUS.
0378 = BL$ADR EQU 0378H ;DCM LOAD AND JUMP ADDRESS.
037A = BL$BSZ EQU 037AH ;DCM BLOCK LOAD SIZE.
;
;*****
; 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$EXC EQU 002H ;ISSUE DOUBLE D INTERRUPT.
;
;*****
; ASSEMBLER DIRECTIVES
;*****
;
0100 ORG 0100H ;MODULE ADDRESS (ALTERABLE).
;
;*****
; SET STACK AND CONTROLLER ADDRESS
;*****
;
0100 318000 BEGIN: LXI SP,BSTACK ;SET STACK POINTER.
0103 DB43 IN D$PORT ;INPUT STATUS PORT.
0105 E60C ANI DS$ASW ;MASK FOR ADDR SWS.
0107 07 RLC ;POSITION BITS.
0108 F6E4 ORI D$BASE SHR 8 ;OR IN BASE ADDR.
010A 67 MOV H,A ;HIGH BYTE VALUE.
010B 2E00 MVI L,0 ;LOW BYTE VALUE.
010D 224000 SHLD D$ADDR ;STORE THE ADDRESS
;
;*****
; INJECT BOOT MODULE INTO CONTROLLER
;*****
;
0110 3E01 INJECT: MVI A,DC$MBO ;REQUEST DD MEM BANK 0.
0112 D343 OUT D$PORT ;ISSUE COMMAND.
0114 01C000 LXI B,IM$SIZE ;INJECTION SIZE.
0117 EB XCHG ;D$ADDR HL TO DE.
0118 210010 LXI H,IM$ADR ;INJECTION MODULE.
011B CD5901 CALL BLOCK ;BLOCK MOVE.
;
;*****
; RESET AND START THE DISK PROCESSOR
;*****
;
011E 3E80 MVI A,DC$BGN ;BEGIN DD PROCESSOR.
0120 D343 OUT D$PORT ;ISSUE COMMAND.
0122 E3 XTHL ;ALLOW DOUBLE D TIME
0123 E3 XTHL ;TO START UP.
;
;*****
; WAIT FOR TASK COMPLETION
;*****
;
0124 DB43 WAIT: IN D$PORT ;INPUT DD STATUS.
0126 E602 ANI DS$HLT ;TEST HALT* STATUS.

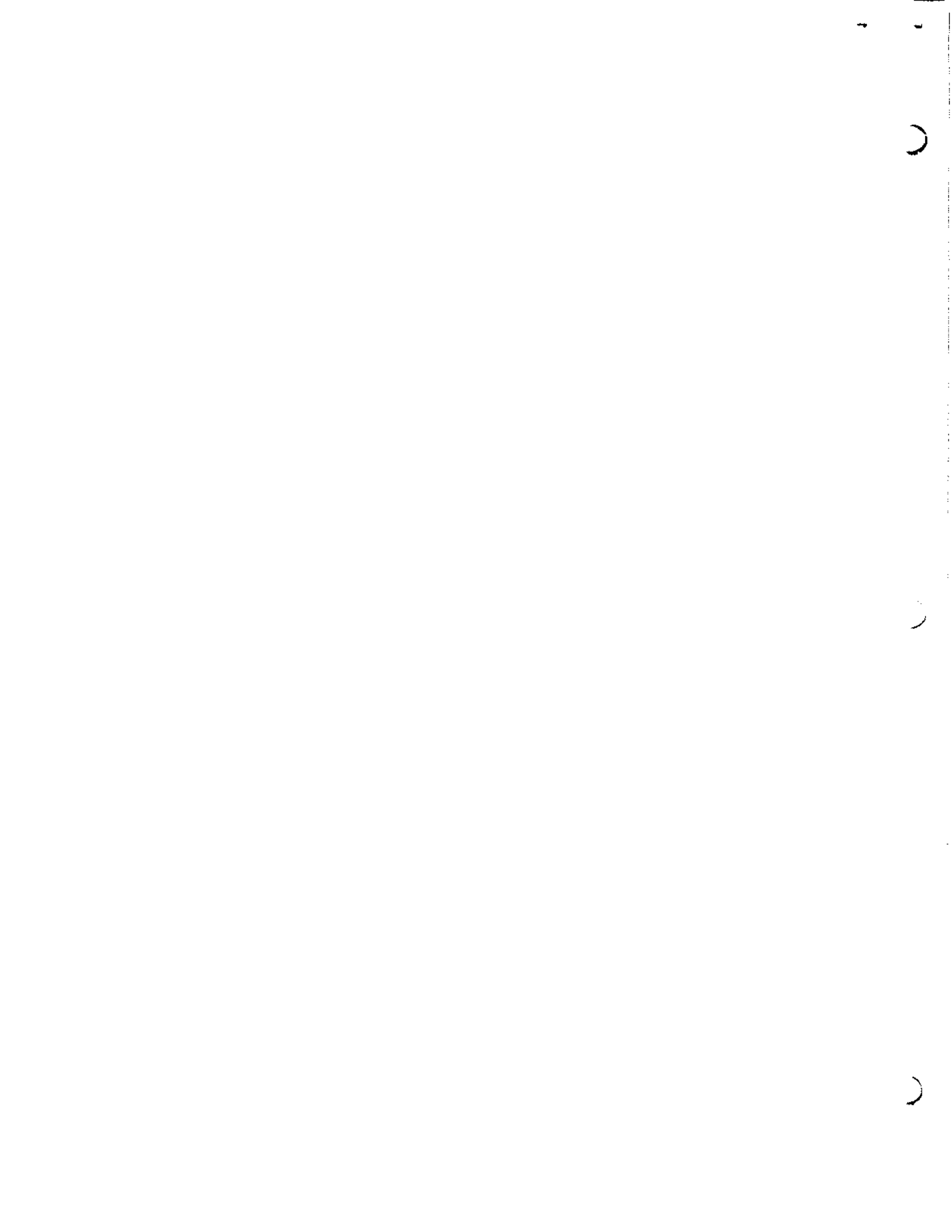
```



```

; *****
; SWITCH CONTROLLER MEMORY INTO SYSTEM BUS *
; *****
;
012B 3E01          MVI    A,DC$MR0      ;REQUEST MEM (BANK 0).
012D D343          OUT    D$PORT      ;ISSUE COMMAND.
;
; *****
; CHECK FOR BOOTSTRAP MALFUNCTION *
; *****
;
012F 2A4000       LHLD   D$ADDR      ;CONTROLLER ADDRESS.
0132 117703       LXI   D,BL$DCS    ;ERROR CODE OFFSET.
0135 19           DAD    D          ;SET HL POINTER.
0136 7E           MOV    A,M        ;GET ERROR CODE.
0137 A7           ANA   A          ;TEST REGISTER.
0138 C26601       JNZ   BAD$LD      ;BAD LOAD.
;
; *****
; PERFORM BLOCK TRANSFER FROM DISK MEMORY *
; *****
;
013B 2A4000       LHLD   D$ADDR      ;CONTROLLER ADDRESS.
013E 117803       LXI   D,BL$ADR    ;LOAD ADDRESS PNTR.
0141 19           DAD    D          ;SET HL POINTER.
0142 5E           MOV    E,M        ;LOW ORDER ADDR.
0143 23           INX   H          ;INCREMENT HL.
0144 56           MOV    D,M        ;HIGH ORDER ADDR.
0145 23           INX   H          ;REQUIRES BL.BSZ NEXT.
0146 4E           MOV    C,M        ;LOW ORDER LENGTH.
0147 23           INX   H          ;INCREMENT HL.
0148 46           MOV    B,M        ;HIGH ORDER LENGTH.
0149 D5           PUSH  D          ;USE AS JUMP ADDR.
014A 3E03         MVI   A,DC$MB1    ;SWITCH TO MEM BANK 1.
014C D343         OUT   D$PORT      ;ISSUE COMMAND.
014E 2A4000       LHLD   D$ADDR      ;DOUBLE D MEM ADDRESS.
0151 CD5901       CALL  BLOCK      ;MOVE BIOS MODULE.
;
; *****
; TRANSFER CONTROL TO OPERATING SYSTEM *
; *****
;
0154 3E01         MVI   A,DC$MBO    ;SWITCH TO BANK 0
0156 D343         OUT   D$PORT      ;ISSUE COMMAND.
0158 C9           RET    ;GOTO BIOS COLD ENTRY.
;
; *****
; BLOCK MOVE SUBROUTINE (Z80 BLOCK MOVE REGISTERS) *
; *****
;
BLOCK: 0159 7E           MOV    A,M        ;GET BYTE.
015A 23           INX   H          ;INC POINTER
015B EB           XCHG   ;GET DESTINATION.
015C 77           MOV    M,A        ;PUT BYTE.
015D 23           INX   H          ;INC POINTER
015E EB           XCHG   ;GET SOURCE.
015F 0B           DCX   B          ;ONE LESS TO DO.
0160 78           MOV    A,B        ;GET HI COUNT.
0161 B1           ORA   C          ;GET LO COUNT.
0162 C25901       JNZ   BLOCK      ;FINISH LOADING.
0165 C9           RET
;
; *****
; ERROR HAS BEEN DETECTED *
; *****

```



```
*****  
;  
; PROGRAM ID: BOOTSTRAP INJECTION MODULE *  
;  
; *****  
;  
; PROPERTY OF: JADE COMPUTER PRODUCTS *  
; 4901 W. ROSECRAWS BLVD. *  
; HAWTHORNE, CALIFORNIA *  
; 90250, U.S.A. *  
;  
; *****  
;  
; VERSION: 2.2 *  
;  
; *****  
; THE BOOTSTRAP INJECTION MODULE IS ONE OF TWO *  
; MODULES THAT TOGETHER MAKE UP THE SYSTEM RESIDENT *  
; BOOT. THIS MODULE IS DOWNLOADED INTO THE DOUBLE D *  
; MEMORY BY THE SYSTEM BOOTSTRAP DRIVER. THE MODULE *  
; THEN READS IN THE DISK CONTROLLER MODULE (DCM) FROM *  
; TRACK 0. THE ORIGIN OF THIS PROGRAM IS FIXED AS IT *  
; IS ASSEMBLED TO EXECUTE INSIDE THE DOUBLE D. THE *  
; BOOT INJECTION MODULE CAN THEREFORE RESIDE IN *  
; THE BOOTSTRAP PROM WITHOUT THE NEED TO REASSEMBLE. *  
; NOTE: STEP TIMING AND MOTOR TURN-ON DELAYS ARE *  
; DEFINED IN THIS MODULE. PATCHING MAY BE REQUIRED. *  
; ***** SK *****
```

```

;*****
; CONTROLLER PORT ASSIGNMENTS *
;*****
    
```

```

0000      BL.STS  ==      000H      ;BOARD STATUS
0000      BL.CTL  ==      000H      ;BOARD CONTROLS
0004      WD.CMD  ==      004H      ;1791 COMMAND REGISTER
0004      WD.STS  ==      004H      ;1791 STATUS REGISTER
0005      WD.TRK  ==      005H      ;1791 TRACK REGISTER
0006      WD.SEC  ==      006H      ;1791 SECTOR REGISTER
0007      WD.DTA  ==      007H      ;1791 DATA REGISTER
0008      XP.STP  ==      008H      ;STEPPER PULSE
0010      XP.MTO  ==      010H      ;MOTOR TIME OUT
0040      XP.MTX  ==      040H      ;MOTOR TIME EXTEND
0080      XP.DSH  ==      080H      ;DATA SYNC HOLD
    
```

```

;*****
; 1791-01 COMMAND CODES *
;*****
    
```

```

0018      DC.HDL  ==      018H      ;LOAD R/W HEAD.
0090      DC.RMS  ==      090H      ;READ MULTI-SECTOR.
00D0      DC.STS  ==      0D0H      ;SET TYPE 1 STATUS.
    
```

```

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

```

0001      BS.US0  ==      001H      ;1791-01 INDICATOR (CLOSED).
0000      BC.DR0  ==      000H      ;DRIVE 0 SELECT.
0004      BC.DSE  ==      004H      ;DRIVE SELECT ENABLE
    
```

```

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

```

009C      DM.RER  ==      10011100B  ;READ ERROR TEST MASK
0004      DM.TK0  ==      00000100B  ;TRACK 0 TEST
    
```

```

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

```

0008      TM.STP  ==      8          ;STEPPER INTERVAL - MS.
0001      TM.DBR  ==      1          ;DELAY BEFORE READ- MS.
0050      NB.TRK  ==      80         ;MAXIMUM NMBR OF STEPS.
    
```

```

;*****
    
```

```
*****  
; INTERNAL MEMORY ASSIGNMENTS *  
*****  
0000 BANK.0 == 0000H ;LOWER BANK ADDRESS.  
0400 BANK.L == 0400H ;1K BANK LENGTH.  
0400 BANK.1 == BANK.0+BANK.L ;UPPER BANK ADDRESS.  
0066 INT.NM == BANK.0+0066H ;NON-MASKABLE INT ADDR.  
0376 BL.ERC == BANK.0+0376H ;ERROR CODE LOCATION.  
0377 BL.DCS == BANK.0+0377H ;DISK CONTROLLER STAT.  
  
*****  
; BOOTSTRAP COMMUNICATION *  
*****  
0001 BE.HOM == 001H ;HOME ERROR.  
0002 BE.RDA == 002H ;READ ERROR A.  
0004 BE.RDB == 004H ;READ ERROR B.  
  
*****  
; DISK CONTROLLER MODULE (DCM) LINKAGE *  
*****  
000D DCM.SS == 13 ;FIRST DCM SECTOR = 13.  
0403 DCM.BG == BANK.1+3 ;DCM COLD START ENTRY.  
0400 DCM.LN == 0400H ;DCM LENGTH  
  
*****  
; ASSEMBLER DIRECTIVES *  
*****  
0000 .PABS ;ABSOLUTE ADDRESSING.  
 .PHEX ;INTEL HEX FILE FORM.  
 .XLINK ;NO LINKAGE OUTPUT.  
 .LOC BANK.0 ;PROGRAM START POINT  
  
*****  
; DELAY MACRO. ALLOWS 1791 TO DIGEST INSTRUCTIONS *  
*****  
 .DEFINE DELAY = [  
 XTHL  
 XTHL  
 XTHL  
 XTHL]  
  
*****
```

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

0000	31 0400	BEGIN:	LXI	SP,BANK.1	;SET UP STACK.
0003	D840		IN	XP.MTX	;TURN ON MOTOR.
0005	0E00		MVI	C,0	;ASSUME 1793.
0007	D800		IN	BL.STS	;INPUT STATUS.
0007	E601		ANI	BS.USO	;TEST USER SW 0.
000B	2002		JRNZ	SELECT	;GOTO SELECT DRV.
000D	0E1F		MVI	C,OFFH	;1791-01 INVERTS.

\*\*\*\*\*  
 ; CLEAR 1791-01 INTERRUPT AND SELECT DRIVE 0 \*  
 \*\*\*\*\*

000F	DD 009C	SELECT:	CALL	STATUS	;179X-01 FORCED CLEAR.
0012	3E04		MVI	A,BC.DSE!BC.DRO	;DRIVE 0, ENABLED.
0014	D300		OUT	BL.CTL	;OUTPUT CONTROLS.

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

0016	2E50	HOME:	MVI	L,NB.TRK	;SET MAX TRACKS..
0018	CD 009C	STEP:	CALL	STATUS	;GET 179X STATUS.
001B	E604		ANI	DM.TKO	;TEST TRACK 0 BIT.
001D	200E		JRNZ	TRACK0	;TRACK 0 EXIT.
001F	2D		DCR	L	;DEC ATTEMPTS.
0020	CA 0091		JZ	ER.HOM	;CANT FIND TRK 0?
0023	D808		IN	XP.STP	;ISSUE STEP PULSE.
0025	11 0008		LXI	D,TM.STP	;STEP INTERVAL TIME.
0028	CD 00A9		CALL	TIMER	;PAUSE FOR PERIOD.
002B	18EB		JMPR	STEP	;TRY ANOTHER TIME.

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

002D	79	TRACK0:	MOV	A,C	;GET TRACK 0 VALUE.
002E	D305		OUT	WD.TRK	;SET TRACK REGISTER.
0030	D307		OUT	WD.DTA	;SEEK SAME TRACK.
0032	FD21 003D		LXI	Y,RD.SET	;SET NMI RETURN ADDR.
0036	3E18		MVI	A,DC.HDL	;HEAD LOAD COMMAND.
0038	A9		XRA	C	;INVERT (1791-01).
0039	D304		OUT	WD.CMD	;ISSUE COMMAND.
003B	18FE		JMPR	.	;WAIT FOR INTERRUPT.

\*\*\*\*\*

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

```
003D 11 0001 RD.SET: LXI D, TM.DBR ;DELAY BEFORE READ.
0040 CD 00A9 CALL TIMER ;CALL MS. TIMER.
0043 11 0400 LXI D, BANK.L ;SET BANK LENGTH
0046 21 0400 LXI H, BANK.1 ;DCM LOAD ADDRESS
0049 FD21 0089 LXI Y, ER.RDA ;READ ERROR TRAP.
004D 3E0D MVI A, DCM.SS ;FIRST SEC OF DCM.
004F A9 XRA C ;INVERT (1791-01)
0050 D306 OUT WD.SEC ;SET 179X SEC REG.
0052 3E90 MVI A, DC.RMS ;READ MULTI-SECTOR.
0054 A9 XRA C ;INVERT (1791-01).
0055 D304 OUT WD.CMD ;ISSUE 179X COMMAND.
0057 E3 DELAY ;ALLOW 179X TO SETTLE.
005B 1813 JMFR R.BYTE ;GOTO READ ROUTINE.
```

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

```
0066 .LOC INT.NM
0066 DB04 WD.INT: IN WD.STS ;GET 1791 STATUS.
0068 A9 XRA C ;INVERT (1791-01).
0069 32 0377 STA BL.DCS ;MAKE STATUS VISIBLE.
006C FDE3 XTIY ;EXCHANGE (SP)<>IY!
006E ED45 RETN ;BRANCH VECTOR ADDR.
```

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

```
0070 DB30 R.BYTE: IN XP.DSH ;WAIT FOR DATA.
0072 DB07 IN WD.DTA ;INPUT INV DATA.
0074 A9 XRA C ;INVERT (1791-01).
0075 77 MOV M, A ;STORE DCM BYTE.
0076 23 INX H ;INCREMENT POINTER.
0077 1B DCX D ;DECREMENT LENGTH.
0078 7A MOV A, D ;GET HIGH REG.
0079 B3 ORA E ;THEN OR-IN LOW REG.
007A 20F4 JRNZ R.BYTE ;READ ANOTHER BYTE.
```

```
*****
```

```

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

```

007C DB04 R.TEST: IN WD.STS ;INPUT READ STATUS.
007E A9 XRA C ;INVERT (1791-01).
007F E69C ANI DM.RER ;TEST FOR ERRORS.
0081 200A JRNZ ER.RDB ;READ ERROR TRAP.
0083 CD 009C CALL STATUS ;TERMINATE READ.
0086 C3 0403 JMP DCM.BG ;TRANSFER TO DCM.
    
```

```

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

```

0089 3E02 ER.RDA: MVI A,BE.RDA ;LOAD READ ERROR CODE.
008B 1806 JMPR ER.HLT ;GO TO ERROR HALT.
008D 3E04 ER.RDB: MVI A,BE.RDB ;LOAD READ ERROR CODE.
008F 1802 JMPR ER.HLT ;GO TO ERROR HALT.
0091 3E01 ER.HOM: MVI A,BE.HOM ;HOME ERROR CODE.
0093 32 0376 ER.HLT: STA BL.ERC ;DISPLAY ERROR CODE.
0096 AF XRA A ;ZERO A REG.
0097 D300 OUT BL.STS ;DESELECT DRIVE.
0099 DB10 IN XP.MTO ;MOTOR OFF!
009B 76 HLT ;TERMINATE.
    
```

```

;*****
; GET UPDATED 1791-01 STATUS
;*****
    
```

```

009C 3E00 STATUS: MVI A,DC.STS ;TYPE 4 - STATUS.
009E A9 XRA C ;INVERT (1791-01).
009F D304 OUT WD.CMD ;ISSUE COMMAND.
00A1 E3 DELAY ;ALLOW 1791 TIME.
00A5 DB04 IN WD.STS ;GET STATUS
00A7 A9 XRA C ;INVERT (1791-01).
00A8 C9 RET ;RETURN TO CALLER.
    
```

```

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

```

00A9 3EF7 TIMER: MVI A,247 ;LOAD INT MS VALUE.
00AB 3D MS.INT: DCR A ;DEC FOR 1 MS.
00AC 20FD JRNZ MS.INT ;REPEAT FOR 1 MS.
00AE 1B DCX D ;TEST FOR ANOTHER MS.
00AF 7A MOV A,D ;CHECK REG D.
00B0 B3 ORA E ;AND REGISTER E.
00B1 20F6 JRNZ TIMER ;DO ANOTHER 1 MS.
00B3 C9 RET ;TIME PERIOD EXPIRED!
    
```

```

;*****
    
```

.END



ADL 330 CP/M DISK ASSEMBLER VERSION 2.21  
 INJECTION MODULE - JADE DOUBLE D DISK CONTROLLER  
 +++++ SYMBOL TABLE +++++

BACK.O 0000	BANK.1 0400	BANK.L 0400	BC.DRO 0000
BL.DSE 0004	BEGIN 0000	BE.HOM 0001	BE.RDA 0002
BE.RDB 0004	BL.CTL 0000	BL.DCS 0377	BL.ERC 0376
BL.STS 0000	BS.USO 0001	DCM.BG 0403	DCM.LN 0400
DCM.SS 000D	DC.HDL 0018	DC.RMS 0090	DC.STS 00D0
DM.RER 009C	DM.TKO 0004	ER.HLT 0093	ER.HOM 0091
ER.RDA 0089	ER.RDB 008D	HOME 0016	INT.NM 0066
MS.INT 00AB	NB.TRK 0050	RD.SET 003D	R.BYTE 0070
R.TEST 007C	SELECT 000F	STATUS 009C	STEP 0018
TIMER 00A9	TM.DBR 0001	TM.STP 0008	TRACK0 002D
WD.CMD 0004	WD.DTA 0007	WD.INT 0066	WD.SEC 0006
WD.STS 0004	WD.TRK 0005	XP.DSH 0080	XP.MTO 0010
XP.MTX 0040	XP.STP 0008		

)

)

)