MAR 86 REMARK
PAINT.ASM PART I

FDC 880H SOFTWARE MANUAL

7210 CLAIREMONT MESA BLVD. • SAN DIEGO, CA 92111 • (619) 560-1272

C.D.R. Systems
INCORPORATED

C.D.R. Systems
INCORPORATED

7210 Clairemont Mesa Blvd. • San Diego, California 92111 • Telephone: (714) 560-1272

FDC 880H

SOFTWARE MANUAL

BASIC INSTALLATION AND STARTUP

The standard version of the FDC-88ØH is configured to use ports 73 thru 77 (3B-3F Hex). The FDC-88ØH may be installed in either the cassette position (P51Ø) or the Serial I/O position (P511).

The signal cable for the floppy disc system plugs into the FDC-88ØH connectors at J3 for 8" drives and at J4 for 5.25" drives. (Refer to the hardware manual for additional guidance on Prom installation, Cable installation, and Board installation procedures).

Note that CDR82 or your present 444-66 zero origin prom may be used at location U517. Either prom will provide the ram at location zero for your computer. The supplied jumper wire is to be used between the center pin of JJ5Ø3 and pin 17 of P5Ø9. (JJ5Ø3 is NOT present in later versions of the H/Z/89/9Ø and the jumper wire may be discarded).

The present H17 boot prom is to be left in place at U52Ø and the present Monitor prom at U518 is also left in place.

The I/O decoder prom at U55Ø must be replaced with the CDR86 decoder prom. This allows the FDC-88ØH to be accessed at port 73-77 (3B-3F HEX).

The supplied prom CDR83D, must be used at location U519. This allows retention of all HDOS operation with hard sectored 5.25" discs as supplied by Heath/Zenith, as well as all CP/M operation on soft sectored discs when accessed thru your CDR Systems supplied BIOS. Note that those drives connected to the H88-4 controller become additional drives (Hard Sectored Discs only) when accessed as logical drives E:,F:, and G: with your CDR Systems BIOS.

CONFIGURATION FOR USE OF THE FDC-880H WITH H88-4 CONTROLLER CARD

The FDC-880H can be run in conjunction with the H88-4 controller card that comes with the Z89 or H89. The FDC-880H can be plugged into the cassette I/O slot (P510) , or the serial I/O slot (P511) , with the H88-4 in it's own current slot (P512) . The C.D.R. Systems BIOS V.2.01 and subsequent updates, are set to run up to 4 drives on the FDC-880H and 3 drives on the H88-4 Controller. Therefore up to 7 drives can be run simultaneously through the BIOS.

The System can be booted up on any of the 4 drives connected to the FDC-880H. When the system boots up, the drive that is booted is accessed as logical drive A: . The other drives connected to the FDC-880H are logically oriented according to the physical location of the booted drive. The drives connected to the H88-4 controller are always accessed in a fixed manner. They are logically designated as drives E: , F: , and G: .

EXAMPLE DRIVE CONFIGURATIONS

THROUGH FDC-880H CONTROLLER
PHYSICAL DRIVE NUMBERS

	Ø	1	2	3
BOOTØ	A:*	B:	C:	D:
BOOT1	D:	A:*	B:	C:
BOOT2	C:	D:	A:*	B:
BOOT3	B:	C:	D:	A:*

THROUGH H88-4 CONTROLLER
PHYSICAL DRIVE NUMBERS

3	2	1
E:	F:	G:
E:	F:	G:
F:	F:	G:
E:	F:	G:

If BOOT1 then physical drive 2 = B:, drive 3 = C:, drive Ø = D:

The FDC-880H uses soft sectored formatting on diskettes with single density in FM and double density in MFM recording modes. The BIOS is set for various industry standard diskette formats, including formats compatable with Heath/Zenith 8 in. diskette formats.

The hard sectored diskettes that are used with the H88-4 controller card can be reformatted with a soft sectored double density format through the FDC-880H controller card. This allows for the same Hard sectored diskette to run at double density through the FDC-880H controller card.

FIRST TIME THROUGH

Assuming the FDC-88ØH has been installed and your floppy drives are connected and powered, (see FDC-88ØH Hardware Operation Manual), it's only neccessary to insert the soft sector formatted CP/M diskette (set with FDC-88ØH BIOS), and type the letter B . The Heath boot message will appear. Type a number Ø thru 3 , and the system will attempt to boot from the designated drive. There will be a 5 to 1Ø second pause while the FDC-88ØH examines your drive to determine if it contains a readable diskette and what type of diskette is present. (5.25" or 8", single or double density, single or double sided.) If the drive does not contain a readable diskette the boot prom will search the connected drives for a readable diskette.

The drive that CP/M is booted up on will be labeled logical drive A: in the system. This drive may well be physical drive 3 in a 4 drive system, (set as drives Ø-3) . In this case drive 3 will be logical drive A: , drive Ø will be logical drive B: , drive 1 will be logical drive C: , and drive 2 will be logical drive D: . Thus you may boot your system from any drive connected to the system.

EXAMPLE: booting a 48K CP/M on the fourth physical drive. What is typed on the keyboard is shown in brackets [...]

```
|--------------------------------|--------------------------------|
| For use with CDR81 type prom   | For use with CDR83 type prom   |
|                                |                                |
|      H: [B]oot[3]              |      H: [G]o [1ØØØØ]           |
|                                |    BOOT[3]                     |
|    48K CP/M 2.2                |                                |
|                                |    48K CP/M 2.2                |
|    A>                          |                                |
|                                |    A>                          |
|--------------------------------|--------------------------------|
```

The first access of a drive will cause the FDC-88ØH to pause to determine what type of diskette is in the drive. The subsequent accesses to that drive (if diskette type has not changed) will not cause the FDC-88ØH to pause.

HERE'S A VERY HOT TIP
* Never ever turn off your computer with a disk in a drive *

HERE'S ANOTHER TIP
* Always power up all connected drives prior to using your system. If a drive must be turned off for maintenance, etc., disconnect the signal cable from the drive. No fatal error will be caused, but CP/M will not properly access the remaining drives if a dead (i.e. non pulled up) terminator is on the signal lines.

ADDITIONAL SOFTWARE SUPPLIED WITH BIOS

There is additional software that we supply for your CP/M 2.2 . On your diskette there are three programs called PUTCPM32.COM , PUTCPM48.COM , and PUTCPM64.COM . These programs are for the easy transfer of your CP/M to another diskette. Each holds a sized image of your CP/M and the C.D.R. Systems BIOS. By executing one of these programs, your CP/M is placed on a specified physical drive.

EXAMPLE: Placing CP/M set for 48K of memory on physical drive Ø
What is typed on the keyboard is shown in brackets [...]

```
A>[PUTCPM48]

        PUTCPM48 ROUTINE V.1.7
TRANSFERS 43K IMAGE OF CP/M 2.2 TO
A SPECIFIED (PHYSICAL) DRIVE NUMBER


ENTER PHYSICAL DRIVE NUMBER  FROM Ø TO 3
>[Ø]DRIVE ØØ TYPE CØ

END OF PUTSYS ROUTINE

A>
```

There is also a disk formatter routine supplied which alows you to format various types of diskettes. It is Menu driven, and runs in Heath screen mode. By executing the format routine and following the menu, most types of diskettes can be formatted.

C.D.R. Systems is continually improving the current software that it supplies, and adding to the current programs that come with the FDC-88ØH. A configure routine for easy modifications of BIOS parameters, a duplication routine for quick back-up of disks, and an update on the current C.D.R. Systems BIOS will be available to all owners of the FDC-88ØH. The abillity to run HDOS with your FDC-88ØH controller will be available shortly, and other support programs will be comming in the future. All current owners will receive updates with at most a small handling fee.

AIDES AND ASSISTANCE SUPPLIED IN THE BOOT PROM

A disk diagnostic program is included in the boot prom and may be accessed by going to location 3ØØØ3. (18Ø3 hex) in the CDR 81 type prom. If the CDR 83 type prom is used then go to location 1ØØØ3 (8Ø3 hex).

A menu will be displayed. The menu allows six routines to be run. A seventh routine allows rebooting or booting from the diagnostic.

EXAMPLE: accessing boot prom diagnostics
What is typed on the keyboard is shown in brackets [...]

```
  E: [G]o [3ØØØ3]

DIAGNOSTIC
Ø=BT 1=DR 2=RD 3=RR 4=WT 5=WR
>
```

Ø = BOOT CP/M SYSTEM
Reads the getsys sector and sets parameters for Booting the CP/M Operating System.

1 = TEST FOR ACTIVE DRIVES
Displays the drive number and the type of disk in hex code.

```
Bit 7 = Sides:  Ø=Double Sided, 1=Single Sided
Bit 6 = Density:  Ø=Double Density, 1=Single Density
Bit 5 = Type of Drive:  Ø= 8 inch, 1= 5.25 inch
Bits 4-2: are currently unused
Bit 1 = 5.25" New Extended Format, 1=Extended
Bit Ø = Sector Size:  Ø=Normal, 1=Extended
```

EXAMPLE:

```
>1
DV ØØ TP CØ  (drive Ø,single side, single density, 8")
DV Ø1 TP FF  (drive 1, inactive drive)
DV Ø2 TP Ø1  (drive 2,two sides,double density,8",Extended)
DV Ø3 TP AØ  (drive 3,single side, double density, 5.25")
```

AIDES AND ASSISTANCE SUPPLIED IN THE BOOT PROM (CONTINUED)

2 = READ SEQUENTIALLY FROM A SPECIFIED DRIVE
Reads each sector of a drive, Displays errors,
Prints a ´P´ at the end of each full pass.

3 = READ RANDOMLY FROM A SPECIFIED DRIVE
Reads random tracks and sectors from a drive,
displays errors

4 = WRITE SEQUENTIALLY RANDOM DATA TO A SPECIFIED DRIVE
Writes random byte patterns into each sector,
Reads the sector back into memory and
Compares the two sectors, displaying errors.
Special error code: FD = Read data doesn´t match
Prints a ´P´ at the end of each full pass.
WARNING This routine overwrites all data on the
specified diskette.

5 = WRITE RANDOMLY RANDOM DATA ONTO A SPECIFIED DRIVE
Same as test 4, but with random tracks and sectors.
WARNING This routine overwrites all data on the
specified diskette.

PRESSING THE ESCAPE KEY EXITS FROM THE ABOVE TESTS

THE ERROR MESSAGES SHOWN BY THESE TESTS ARE EXPLAINED ON
PAGE A21 OF THE BIOS SOURCE LISTING.

```
***************************************************************************
***************************************************************************
**                                                                       **
**                                                                       **
**           C.D.R. SYSTEMS INC. I/O CONFIGURATION PROGRAM               **
**                                                                       **
**                   BY  M.L.DAWSON & M.D.BROOKS                         **
**                                                                       **
**                                                                       **
**                                                                       **
**                                                                       **
**      VERSION 1.Ø                                      3/Ø1/8Ø         **
**                                                                       **
***************************************************************************
***************************************************************************
```

INTRODUCTION

This configuration program is provided to aid users in customizing parts of the C.D.R Systems BIOS. The program tries to be as user friendly as possible. It is menu driven and allows only those changes that are known to the BIOS. The user need only make a single letter selection to make the desired changes with option to undo the changes.

The following sections describe what to expect from the program and what steps to make.

INVOKING THE PROGRAM

The program, CDRMOD, is invoked from CP/M by typing the command:

```
CDRMOD
```

From here, the user makes menu selections. As a note, the user may exit (abort) the program by entering the CTRL-C code at any time and yet maintain system integrity. Also, any selection not in the menu is ignored by the program.

MAIN MENU

CDRMOD first comes up with the main menu as follows:

```
        C.D.R. Systems Inc. I/O Configuration Program

Version 1.0       by M.L.Dawson and M.D.Brooks        1/20/81

CP/M CONFIGURATION

        A       TERMINAL AND PRINTER
        B       DEFAULT I/O

        X       MAKE PERMANENT CHANGE AND QUIT
        Y       MAKE TEMPORARY CHANGE AND QUIT
        Z       MAKE NO CHANGF AND QUIT

SELECTION:
```

To go further, select one of the letters preceeding a selection message.

The following is what happens when you select one of the above letters:

A .... the TERMINAL AND PRINTER sub-menu replaces the main menu.
B .... the DEFAULT I/O sub-menu replaces the main menu.
X .... the screen is cleared and you are prompted as to whether you want to update the "PUTCPM" files. In any case, a permanent change is made to the BIOS and you are returned back to CP/M. (See section under PERMANENT I/O CHANGES.)
Y .... a temporary change is made to the BIOS and you are returned back to CP/M.
Z .... no changes are made and you are returned back to CP/M.

TERMINAL AND PRINTER

The following is the sub-menu that replaces the main menu after selecting the letter "A":

```
TERMINAL AND PRINTER

        A       CRT: BAUD RATE = 9600  PORT = ØE8H (350Q)
        B       TTY: BAUD RATE = 300   PORT = ØDØH (320Q)
        C       LPT: BAUD RATE = 9600  PORT = ØEØH (340Q)
        D       UR1: BAUD RATE = 300   PORT = ØD8H (330Q)
        E       UP1: BAUD RATE = 300   PORT = ØD8H (330Q)

        F       NUMBER OF NULLS AFTER CR ON CRT: Ø
        G       NUMBER OF NULLS AFTER CR ON TTY: Ø
        H       NUMBER OF NULLS AFTER CR ON LPT: Ø

        I       PRINTER READY SIGNAL POLARITY <HIGH OR LOW>: HIGH
        J       PRINTER READY SIGNAL TYPE <DTR OR RTS>: RTS

        K       SET DEFAULT CONFIGURATION

        Y       MAKE CHANGE AND RETURN TO MAIN MENU.
        Z       MAKE NO CHANGE AND RETURN TO MAIN MENU.

SELECTION:
```

Here, several different changes can be made. Entering one of the letters "A" thru "E" will select a "baud rate" menu followed by a "port selection" menu. Entering one of the letters "F thru H" accesses an even different menu to select the number of nulls to follow a carriage-return.

Selecting the letters "I" or "J" causes the menu to refresh with the conditions toggled. Selecting the letter "K" will present the menu in the original form that you received it on disk.

Entering the letter "Y" makes the changes actually take place as well as return to the main menu. So to make no changes and also exit this menu, enter the letter "Z".

The following is the menu that replaces the "TERMINAL AND PRINTER" menu for one of the letter selections of "A" thru "E":

```
BAUD RATE SELECTION

        A       75
        B       110
        C       134
        D       150
        E       300
        F       600
        G       1200
        H       2400
        I       4800
        J       9600
        K       19200
        Z       NO CHANGE

CRT: BAUD RATE =
```

Selecting one of the letters changes the baud rate to one corresponding to the selected letter. Of course, no change has to be made as the letter "Z" may be selected and you skip to the "PORT SELECTION" menu. The baud rate selection could also be made on one of the following devices depending on which letter selection was made in the "TERMINAL AND PRINTER" menu.

```
      TTY: BAUD RATE =
      LPT: BAUD RATE =
      UR1: BAUD RATE =
      UP1: BAUD RATE =
```

The "PORT SELECTION" menu replaces the "BAUD RATE SELECTION" menu next and again selecting one of the letters will change the device port to the one corresponding to the letter. Also, no change has to be made, just select the letter "Z" which returns to the "TERMINAL AND PRINTER" menu.

```
PORT SELECTION

        A       0E8H (350Q)
        B       0E0H (340Q)
        C       0D8H (330Q)
        D       0D0H (320Q)
        Z       NO CHANGE

CRT: PORT =
```

The port selection could also be made on one of the following devices depending on which letter selection was made in the "TERMINAL AND PRINTER" menu.

```
       TTY: PORT =
       LPT: PORT =
       UR1: PORT =
       UP1: PORT =
```

The following menu replaces the "TERMINAL AND PRINTER" menu when one of the letters "F", "G" or "H" is selected:

```
NUMBER OF NULLS

        A       Ø
        B       1
        C       2
        D       3
        E       4
        F       5
        G       6
        H       7
        I       8
        J       9
        K       1Ø
        L       11
        M       12
        N       13
        O       14
        P       15
        Z       NO CHANGE

AFTER CR ON CRT:
```

Selecting a letter coresponding to one of the values (1 thru 15) will change the number of nulls following a carriage-return to provide delay after the carriage return. Selecting the letter "Z" returns to the "TERMINAL AND PRINTER" menu without making any changes to the null count.

The device that the change applies to is determined by the letter selection from the "TERMINAL AND PRINTER" menu. the other devices are "LPT" and "TTY".

DEFAULT I/Ø

The following is the sub-menu that replaces the main menu after selecting the letter "B":

```
DEFAULT I/O

        A       CON: = CRT:     < TTY: CRT: BAT: UC1: >
        B       RDR: = UR1:     < TTY: PTR: UR1: UR2: >
        C       PUN: = UP1:     < TTY: PTP: UP1: UP2: >
        D       LST: = LPT:     < TTY: CRT: LPT: UL1: >

        E       SET DEFAULT

        Y       MAKE CHANGE AND RETURN TO MAIN MENU.
        Z       MAKE NO CHANGE AND RETURN TO MAIN MENU.

SELECTION:
```

Selecting one of the letters, "A" thru "D", causes the above menu to refresh with a change in the logical assignment. As an example, if the letter "D" is selected then the new presentation would look like:

```
        D       LST: = UL1:     < TTY: CRT: LPT: UL1 >
                       ---
```

Selecting the letter "D" again and it would look like:

```
        D       LST: = TTY:     < TTY: CRT: LPT: UL1 >
                       ---
```

The part of the presentation that is updated is the part that is underlined in the example. (The underline is not in the normal menu.)

Selecting the letter "E", refreshes the menu with the original default for this menu. The above menu is that original default.

Selecting the letter "Y" makes the new changes and returns to the main menu. Selecting the letter "Z" makes no changes but also returns to the main menu.

PERMANENT I/O CHANGES

This is the section that replaces the main menu after selecting the letter "X". The screen is cleared and the following is the prompt that is presented:

```
   DO YOU WANT TO MAKE CHANGES TO "PUTCPM" ROUTINES? (Y OR N):
```

Selecting the letter "N" causes the changes to become permanent to the CP/M operating system but not to the PUTCPM "COM" files. Then this program says that it is completed and returns to CP/M.

Selecting the letter "Y" causes the changes to become permanent to the CP/M operation system and also to the PUTCPM "COM" files. AS the PUTCPM files are changed their names are presented as follows:

```
   PUTCPM32COM ....
   PUTCPM48COM ....
   PUTCPM64COM ....

   PUTCPM MODIFICATIONS COMPLETED
```

Following the presented names are messages that report progress or errors.

EXAMPLE:

```
        PUTCPM32COM .... NOT ON DRIVE A:
        PUTCPM48COM .... MODIFIED
        PUTCPM64COM .... MODIFIED
```

Next, the actual bios is modified on the System file on drive "A". This fact is indicated by the following presentation that is part of this section on "PERMANENT I/O CHANGES".

```
   MODIFYING BIOS ON DRIVE A:
   DRIVE Ø1  TYPE CØ

   END OF I/OMOD ROUTINE

A>
```

C.D.R. SYSTEMS INC. FORMATTER ROUTINE VERSION 2.2 2/4/82
Formats a Disk to a Specified Type and Drive Number

This is version 2.2 of the C.D.R. Systems Disk Formatter Due to the numerous format types that are being used under CP/M in Heath/Zenith computers, this formatter allows for the formatting of disks in three general catagories:

- 8 inch drives
- 5.25 inch drives
- alternate 5.25 inch drives

Within the three general catagories there are FOUR disk type formats that can be selected:

- single density
- double density
- single density extended
- double density extended

Once the disk type has been selected, the formatter allows for the selection of the number of sides of a disk that are to be formatted at this type. WARNING: Single sided drives with the formatter will accept double sided format commands, but the resulting disk will be unreadable, and will need reformatting.

In the 5.25 inch Catagories there are drives with different track per side capacities. This formatter asks for one of four standard 5.25 inch tracks per side counts to be entered:

- 35 tracks per side
- 4Ø tracks per side
- 77 tracks per side
- 8Ø tracks per side

The formatter will also ask for the LOGICAL drive number that holds the disk to be formatted. As each track is formatted either an ansterisk ´*´ for a good track, or a dash ´-´ for a bad track is sent to the console. When the format is completed, if errors have occured, an error message is sent to the screen. While the disk is being formatted, the format process can be aborted by the CP/M warm start code. (^C control-C)

This version is an upgrade from the earlier version 1.X formatters, in that it interfaces with the CP/M BIOS to locate the LOGICAL drive number, and tests each formatted track for errors.

The following is a list of the formatted capacities currently available with this formatter. The capacities do not reflect the actual amount of CP/M file space available on a disk, due to the number of system tracks used by CP/M as reserved for operating system area. (2 tracks for 8", 3 tracks for 5.25")

FORMATTED CAPACITIES PER TRACK

	8 inch disk type		bytes/sec	secs/track	bytes/side	
*#	SINGLE DENSITY	(B)	128	26	256 k	
*#	DOUBLE DENSITY	(B)	256	26	512 k	
!	SINGLE EXTENDED		256	15	295 k	
*#	DOUBLE EXTENDED	(B)	1024	8	630 k	
	standard 5.25 in.		bytes/sec	secs/track	bytes/side	bytes/side
*	SINGLE DENSITY		128	18	40tk 92 k	80tk 184 k
*	DOUBLE DENSITY	(B)	256	18	184 k	368 k
!#	SINGLE EXTENDED		256	10	102 k	204 k
!*#	DOUBLE EXTENDED		512	10	204 k	409 k
	alternate 5.25 in.		bytes/sec	secs/track	bytes/side	bytes/side
!	SINGLE DENSITY		128	16	40tk 81 k	80tk 163 k
!#	DOUBLE DENSITY		256	16	163 k	327 k
!	SINGLE EXTENDED		512	5	102 k	204 k
!	DOUBLE EXTENDED		1024	5	204 k	409 k

* = currently supported by C.D.R. SYSTEMS INC. BIOS
= currently supported by HEATH/ZENITH
! = will be supported by C.D.R. SYSTEMS INC. BIOS (UPDATE)
(B) = Currently bootable disk type through the CDR BOOT PROM

```
ADDR  CODE       STMT SOURCE STATEMENT

                 0002 ;
                 0003 ;
                 0004 ;         COPYRIGHT       1981,   C.D.R. SYSTEMS INC.
                 0005 ;                                 SAN DIEGO, CA.
                 0006 ;
                 0007 ;         VERSION 2.7C OF THE  FDC-880H FLOPPY DISK
                 0008 ;         CONTROLLER INTERFACE BIOS FOR CP/M 2.2
                 0009 ;
                 0010 ;         THIS BIOS IS SET TO RUN WITH THE  CDR81D, OR
                 0011 ;         CDR83D VERSION 7 BOOT PROM.  WITH THE BIOS SOURCE
                 0012 ;         FOR THE FDC-880H TO RUN THROUGH PORTS 03BH - 03FH
                 0013 ;         TO USE THESE PORTS THE CDR86 PROM MUST BE INSTALLED
                 0014 ;
                 0015 ;                 INITIAL VERSION    12/10/80
                 0016 ;                 UPDATE 12/15/80 DOUBLE DENSITY TABLES
                 0017 ;                 UPDATE 2/17/81 5.25 CAPABILLITY ADDED
                 0018 ;                 UPDATE 3/11/81 AUTO DISK TYPE TESTER
                 0019 ;                 UPDATE 4/30/81 UPDATE BASED ON FIELD TESTS
                 0020 ;                 UPDATE 5/5/81  EXTENDED 8" ADDED (NEEDS UPDATE)
                 0021 ;                 UPDATE 6/6/81  SINGLE SIDE ON DOUBLE S. DRIVE
                 0022 ;                 UPDATE 7/3/81  EXTENDED 5.25 IN. ADDITION
                 0023 ;                 UPDATE 8/27/81 HEATH 5.25" CONTROLLER INTERFACE
                 0024 ;                                ADDED TO BIOS, EXTENDED 8" UPDATED
                 0025 ;                 UPDATE 10/20/81 CLEAN-UP BASED ON FIELD TESTS
                 0026 ;                 UPDATE 1/11/82 CONFIGURE INTERFACE (I/OMOD.COM)
                 0027 ;                 UPDATE 1/21/82 CHANGES IN EXTENDED INTERNAL SECTOR
                 0028 ;                         SKEW FACTOR (SEQUENTIAL LOCATIONS)
                 0029 ;                         AS SUGGESTED BY CDI
                 0030 ;                         NOTE: FILES ON EARLIER EXTENDED FORMATS
                 0031 ;                         WILL NEED TO BE PASSED TO NON EXTENDED
                 0032 ;                         FORMATS FOR UPGRADING TO THIS VERSION
                 0033 ;                         OF THE BIOS. (CLOSER TO HEATH 8" EXTENDED)
                 0034 ;                 UPDATE 2/05/82 EXTENDED TYPE MODIFICATIONS
                 0035 ;                 UPDATE 4/2/82 CLEAN-UP OF DISK ERROR RETURN IN
                 0036 ;                         BIOS V.2.27, ALSO CONDITIONAL ASSEMBLY
                 0037 ;                         PARAMETERS SET FOR NEEDED CHANGES.
                 0038 ;                 UPDATE 5/28/82 HARD SECTOR INTERFACE IMPROVEMENTS
                 0039 ;                         AS SUGGESTED BY PAUL A. BECK
                 0040 ;                 UPDATE 6/10/82 H/Z37 DISK TYPES ADDED TO ASSEMBLY
                 0041 ;                         PARAMETERS, BUT SET AS NOT ASSEMBLED
                 0042 ;                 UPDATE 8/3/82 TESTING FOR 8" SINGLE DENSITY DISK TYPES
                 0043 ;                         MADE FASTER, BOOT MESSAGE PLACED AT BACK OF BIOS
                 0044 ;                 UPDATE 9/15/82 CDR DOUBLE AND SINGLE DENSITY DISK TYPES
                 0045 ;                         REPLACED WITH HEATH/ZENITH DISK TYPES IN THE
                 0046 ;                         ASSEMBLY OF THE BIOS. (THESE TYPES NON-BOOTABLE)
                 0047 ;                         ON 5.25", MUST BOOT WITH CDR EXTENDED DISK TYPE
                 0048 ;
                 0049 ;  BELOW IS A TABLE OF SOFT SECTORED DISK TYPES THE BIOS IS NOW SET FOR:
                 0050 ;!-----------------------!-----------------------!-----------------------!
                 0051 ;! 8 INCH          SIDES ! 40 TRACK 5.25   SIDES ! 80 TRACK 5.25   SIDES !
                 0052 ;!-----------------------!-----------------------!-----------------------!
                 0053 ;! SINGLE DENSITY*  ONE  ! SINGLE DENS. H/Z  ONE !                       !
                 0054 ;! SINGLE DENSITY*  TWO  !                       ! SINGLE DENS. H/Z  TWO !
                 0055 ;! DOUBLE DENSITY*  ONE  ! DOUBLE DENS. H/Z  ONE !                       !
                 0056 ;! DOUBLE DENSITY*  TWO  !                       ! DOUBLE DENS. H/Z  TWO !
                 0057 ;! EXTENDED DENS.*  ONE  ! EXTENDED  CDR*    ONE !                       !
                 0058 ;! EXTENDED DENS.*  TWO  !                       ! EXTENDED* CDR     TWO !
                 0059 ;!-----------------------!-----------------------!-----------------------!
```

```
ADDR  CODE       STMT SOURCE STATEMENT

                 0060 :   * = BOOTABLE DISK TYPE  (SOURCE LISTING ON DISK HOLDS UPDATES ON BIOS)
                 0061 :
                 0062 :          THIS VERSION OF THE BIOS WAS ASSEMBLED WITH 11 TYPES
                 0063 :          OF DISKS, THAT THE BIOS AUTOMATICALLY TESTS FOR,
                 0064 :          WHICH ARE TO BE USED WITH THE FDC-880H. WHEN CHANGING
                 0065 :          DISK TYPES IN A DRIVE THAT HAS BEEN PREVIOUSLY
                 0066 :          ACCESSED THROUGH THE BIOS, A WARM BOOT (^C) LETS
                 0067 :          THE BIOS ACCEPT THE NEW TYPE.
                 0068 :
                 0069 :          NOTE THAT THE H88-4/H17 CONTROLLER CAN BE USED
                 0070 :          WITH THIS BIOS. THE DRIVES ATTACHED THROUGH IT
                 0071 :          ARE PRE-SET AS DRIVES  E:, F:, AND G:.  THE SYSTEM
                 0072 :          BOOTS UP USING ONE OF THE DRIVES CONNECTED TO THE
                 0073 :          FDC-880H.
                 0074 :
                 0075 :          THIS SOURCE WAS ASSEMBLED USING THE
                 0076 :          Z80 DEVELOPMENT PACKAGE FROM
                 0077 :          LIFEBOAT ASSOCIATES
                 0078 :          2248 BROADWAY, NEWYORK NY 10024 (212)580-0082
                 0079 :
                 0080 :          FOR QUESTIONS AND FURTHER UPDATES, CONTACT
                 0081 :                MARC D. BROOKS
                 0082 :                C.D.R. SYSTEMS INC.
                 0083 :                7210 CLAIREMONT MESA BLVD.
                 0084 :                SAN DIEGO, CA. 92111
                 0085 :                PH. (714)560-1272
                 0086 :
                 0087 :          ASSEMBLY TIME CONDITIONAL PARAMETERS
                 0088 :
>FFFF            0089 TRUE    EQU     0FFFFH  : CONDITIONAL ASSEMBLY TRUE
>0000            0090 FALSE   EQU     0       : CONDITIONAL ASSEMBLY FALSE
                 0091 :
>FFFF            ~~0092~~ A8INCH  EQU     TRUE    : ASSEMBLE 8 INCH PARAMETERS
>FFFF            ~~0093~~ AHZ8    EQU     TRUE    : ASSEMBLE H47 8 INCH
>0000            ~~0094~~ ACDR8   EQU     .NOT.AHZ8 : ASSEMBLE CDR 8 INCH
>FFFF            0095 A5INCH  EQU     TRUE    : ASSEMBLE 5 INCH PARAMETERS
>FFFF            0096 ASDE5   EQU     TRUE    : EXTENDED SINGLE DENSITY 5 INCH
>FFFF            0097 AHZ5    EQU     TRUE    : ASSEMBLE HEATH/ZENITH 5 INCH
>0000            0098 ACDR5   EQU     .NOT.AHZ5 : ASSEMBLE CDR 5 INCH
>0000            0099 AHZ5E   EQU     FALSE   : ASSEMBLE HEATH/ZENITH EXTENDED 5 INCH
>FFFF            0100 ACDR5E  EQU     .NOT.AHZ5E : ASSEMBLE CDR EXTENDED 5 INCH
>0000            0101 A40DS   EQU     FALSE   : ASSEMBLE 40 TRACK DOUBLE SIDED
>FFFF            0102 A80DS   EQU     .NOT.A40DS : ASSEMBLE 80 TRACK DOUBLE SIDED
>FFFF            0103 AHSTYP  EQU     TRUE    : ASSEMBLE HARD SECTORED PARAMETERS
>0000            0104 ANDTYP  EQU     FALSE   : DON'T ASSEMBLE HARD SECTORED PARAMETERS
>0000            0105 ADPEMOD EQU     FALSE   : ASSEMBLE MODS TO ACCEPT TRANSLATION TABLES
                 0106 :
                 0107 :          CP/M ALLOCATED SCRATCHPAD AREAS FOR THE CBIOS
                 0108 :
>0000            0109 RAMST   EQU     0       : START OF USEABLE RAM
>0004            0110 DEFALT  EQU     RAMST+4 : DRIVE BOOT DEFAULT LOC.
>0040            0111 TADDR   EQU     RAMST+40H : DATA TRANSFER ADDRESS
>0043            0112 UNIT    EQU     RAMST+43H : UNIT NUMBER
>0045            0113 TRACK   EQU     RAMST+45H : TRACK NUMBER
>0044            0114 SECTR   EQU     RAMST+44H : SECTOR NUMBER
>0042            0115 ERRST   EQU     RAMST+42H : ERROR STATUS STORAGE
>0047            0116 IDSV    EQU     RAMST+47H : IDENTIFICATION SAVE AREA
>0048            0117 SIDEID  EQU     RAMST+48H : READ ADDR. SIDE NUMBER
```

```
     (C) C.D.R. SYSTEMS INC.  BIOS   SD SYSTEMS Z80 ASSEMBLER PAGE 0007
ADDR  CODE        STMT SOURCE STATEMENT

>004A             0118 SECSIZ  EQU     RAMST+4AH : SECTOR SIZE FROM ADDR. READ
>004D             0119 SAVESP  EQU     RAMST+4DH : STACK POINTER SAVE
>004F             0120 SIDE    EQU     RAMST+4FH : DISK DENSITY/SIDE TYPE STORAGE
                  0121 :
                  0122 :       H88/89 CONSOLE I/O VARIABLES
                  0123 :               0E8H
>00E9             0124 DIOB    EQU     0E9H    : INITIAL I/O BYTE >  0ECH
>00F2             0125 MEMCNT  EQU     0F2H    : MEMORY 0 ORIGIN FORMATTING
                  0126 :
                  0127 :       CDR SYSTEMS CONTROLLER  I/O PORTS
                  0128 :
>0038             0129 DPLOC   EQU     038H    : DRIVE PORT START LOCATION 078H
>003B             0130 DSELEC  EQU     DPLOC+3 : DRIVE SELECT PORT DPLOC+1
>003C             0131 DSTAT   EQU     DPLOC+4 : DRIVE STATUS PORT DPLOC+2
>003C             0132 DCMD    EQU     DPLOC+4 : DRIVE COMMAND PORT (SAME AS DSTAT)
>003D             0133 DTRACK  EQU     DPLOC+5 : DRIVE TRACK PORT - DPLOC+3
>003E             0134 DSECT   EQU     DPLOC+6 : DRIVE SECTOR PORT - DPLOC+2
>003F             0135 DDATA   EQU     DPLOC+7 : DRIVE DATA PORT - DPLOC+3
                  0136 :
                  0137 :       DISK TYPE BIT PATTERNS FOR BIOS
                  0138 :
>0020             0139 _5INCH  EQU     00100000B       : 5.25 INCH DRIVE TYPE 1 ON BIT 5
>0000             ~~0140~~ _8INCH  EQU     00000000B       : 8 INCH DRIVE TYPE 0 ON BIT 5
>0040             0141 _SDENS  EQU     01000000B       : SINGLE DENSITY DISK TYPE
>0000             0142 _DDENS  EQU     00000000B       : DOUBLE DENSITY DISK TYPE
>0080             0143 _SSIDE  EQU     10000000B       : SINGLE SIDED
>0000             0144 _DSIDE  EQU     00000000B       : DOUBLE SIDED
>0004             0145 _SAMETO EQU     00000100B       : SAME DISK TYPE ON TRACK 0
>0010             0146 _NRESET EQU     00010000B       : RESET CONTROLLER CHIP
>0080             0147 _INTEN  EQU     10000000B       : ENABLE CONTROLLER INTERRUPTS
>0001             0148 _EXTEND EQU     00000001B       : EXTENDED SECTOR SIZE
>0000             0149 _STAND  EQU     00000000B       : STANDARD SECTOR SIZE
>00F4             0150 _H17TYP EQU     11110100B       : H88-4 CONTROLLER DRIVES
>0001             0151 _H17ACC EQU     00000001B       : H88-4 DRIVE ACCESSED BEFORE
>0000             0152 _CYL    EQU     00000000B       : CYLINDER ACCESS ON DOUBLE SIDED DISKS
>0040             0153 _SIDED  EQU     01000000B       : SIDE/SIDE ACCESS ON DOUBLE SIDED DISKS
>0000             0154 _1STEP  EQU     00000000B       : SINGLE STEP TRACK LOCATE
>0010             0155 _2STEP  EQU     00010000B       : DOUBLE STEP TRACK LOCATE (40 IN 80)
>000F             0156 _H30MIL EQU     30/2            : 30 MILLISECOND STEP RATE FOR H17
>000A             0157 _H20MIL EQU     20/2            : 30 MILLISECOND STEP RATE FOR H17
>0007             0158 _H10MIL EQU     15/2            : 30 MILLISECOND STEP RATE FOR H17
                  0159 :
                  0160 :       COMMAND CODE BIT PATTERNS FOR BIOS
                  0161 :
>0008             0162 _RESTOR EQU     00001000B       : RESTORE THE DRIVE COMMAND
>0010             0163 _SEEK   EQU     00010000B       : SEEK FOR TRACK COMMAND
>0020             0164 _STEP   EQU     00100000B       : STEP SAME DIRECTION COMMAND
>0040             0165 _STEPIN EQU     01000000B       : STEP IN A TRACK COMMAND
>0060             0166 _STEPOT EQU     01100000B       : STEP OUT A TRACK COMMAND
>0088             0167 _READSC EQU     10001000B       : READ A SECTOR COMMAND
>00A8             0168 _WRITSC EQU     10101000B       : WRITE A SECTOR COMMAND
>00C0             0169 _READAD EQU     11000000B       : READ AN ADDRESS COMMAND
>00E0             0170 _READTK EQU     11100000B       : READ A TRACK COMMAND
>00D0             0171 _WRITTK EQU     11010000B       : WRITE A TRACK COMMAND
>00D0             0172 _INTERC EQU     11010000B       : FORCED INTERRUPT COMMAND
                  0173 :
                  0174 :       COMMAND CODE PARAMETERS FOR BIOS            54
                  0175 :
```

```
ADDR  CODE        STMT SOURCE STATEMENT

>0008             0176 _HEADLD EQU      00001000B         ; LOAD THE HEAD PARAMETER
>0004             0177 _VERIFY EQU      00000100B         ; VERIFY COMMAND RESULT PARAMETER
>0003             0178 _30MIL  EQU      00000011B         ; 15 TO 30 MILSEC WAIT ON STEP
>0002             0179 _20MIL  EQU      00000010B         ; 10 TO 20 MILSEC WAIT ON STEP
>0001             0180 _12MIL  EQU      00000001B         ; 6 TO 12 MILSEC WAIT ON STEP
>0000             0181 _6MIL   EQU      00000000B         ; 3 TO 6 MILSEC WAIT ON STEP
>0010             0182 _UPDTRK EQU      00010000B         ; UPDATE TRACK REGISTER PARAMETER
>0002             0183 _SIDE1  EQU      00000010B         ; SET SIDE 1 PARAMETER
>0000             0184 _SIDE0  EQU      00000000B         ; SET SIDE 0 PARAMETER
>0004             0185 _CDELAY EQU      00000100B         ; COMMAND DELAY PARAMETER
>0010             0186 _MULTPL EQU      00010000B         ; MULTIPLE SECTOR R/W PARAMETER
                  0187 ;
                  0188 ;         TYPE 1 COMMAND ERROR CODE BIT PATTERNS FOR BIOS
                  0189 ;
>0080             0190 _NOTRDY EQU      10000000B         ; NOT READY FLAG
>0040             0191 _WRPROT EQU      01000000B         ; WRITE PROTECT FLAG
>0020             0192 _HEADDN EQU      00100000B         ; HEAD DOWN (LOADED) FLAG
>0010             0193 _SEKERR EQU      00010000B         ; SEEK ERROR FLAG
>0008             0194 _CRC    EQU      00001000B         ; CRC ERROR FLAG
>0004             0195 _TRACK0 EQU      00000100B         ; ON TRACK ZERO FLAG
>0002             0196 _INDEX  EQU      00000010B         ; INDEX MARKS FOUND FLAG
>0001             0197 _BUSY   EQU      00000001B         ; CONTROLLER BUSY FLAG
                  0198 ;
                  0199 ;         TYPE 2 AND 3 SPECIFIC ERROR CODE BIT PATTERNS FOR BIOS
                  0200 ;
>0020             0201 _RECTYP EQU      00100000B         ; READ RECORD TYPE ERROR FLAG
>0020             0202 _WRFALT EQU      00100000B         ; WRITE FAULT ERROR FLAG
>0010             0203 _RNF    EQU      00010000B         ; RECORD NOT FOUND ERROR FLAG
>0004             0204 _LOSTDA EQU      00000100B         ; LOST DATA ERROR FLAG
>0002             0205 _DRQ    EQU      00000010B         ; NO DATA REQUEST ACKNOWLEDGE FLAG
                  0206 ;
                  0207 ;         CREATED COMMAND ERROR CODE BIT PATTERNS FOR BIOS
                  0208 ;
>00FF             0209 _NODISK EQU      11111111B         ; NO DISK ACCESSED YET FLAG
>00FB             0210 _INACT  EQU      11111011B         ; INACTIVE DRIVE TEST LIMIT
>00F9             0211 _RETEST EQU      11111001B         ; DISK TYPE RETEST FLAG
>000F             0212 _DROPTP EQU      00001111B         ; DROP DISK TYPE BITS
>0070             0213 _DROPDV EQU      01110000B         ; DROP DRIVE NUMBER, SIDE BITS
                  0214 ;
                  0215 ;         CDR SYSTEMS CONTROLLER TIME-OUT AND ACCEPTANCE COUNTS
                  0216 ;                 5CH?
>002E             0217 NORMAL  EQU      2EH     ; NORMAL TIME OUT
>0010             0218 ACCTIM  EQU      16 -32  ; COMMAND ACCEPTANCE WAIT TIME
>00BF             0219 BUSRTY  EQU      0BFH    ; BUSY RETRY TIMEOUT COUNT
>0070             0220 BUSYW   EQU      070H    ; BUSY WAIT BETWEEN TRIES
>00C8             0221 WAITTM  EQU      200 →400; TIME OUT WAIT TIME COUNT (400 MILSECS)
>0070             0222 MIL2    EQU      112 224 ; TWO MILLISECOND WAIT TIME COUNT (112)
                  0223 ;
                  0224 ;         DISK CONTROLLER INTERRUPT ROUTINE LOCATIONS
                  0225 ;
>0028             0226 INTLOC  EQU      28H     ; DISK READ/WRITE INTERRUPT LOCATION
>002B             0227 INTRET  EQU      2BH     ; READ/WRITE INTERRUPT RETURN LOCATION
                  0228 ;
                  0229 ;         SECTOR TRANSLATION SKIP COUNT CALCULATION CONTROL
                  0230 ;
>0001             0231 _CALC   EQU      1       ; USED TO CALCULATE SECTOR INTEAD OF TABLE
                  0232 ;
                  0233 ;         CP/M DISK COUNT - INCLUDING H17 DISKS
```

```
                 0234 ;
 >0004           0235 NDISKS  EQU     4       ; NUMBER OF DRIVES POSSIBLE ON SYSTEM
                 0236         IF      AHSTYP  ; SET FOR HARD SECTORED DISKS
 >0003           0237 H17DKS  EQU     3       ; NUMBER OF H17 DISKS POSSIBLE
                 0238         ENDIF
                 0239         IF      ANOTYP  ; SET FOR NO HARD SECTORED DISKS
                 0240 H17DKS  EQU     0       ; NUMBER OF H17 DISKS POSSIBLE
                 0241         ENDIF
                 0242 ;
                 0243 ;       CP/M BIOS JUMP TABLE
                 0244 ;
'0000 C3EE01'    0245 BIOSST: JP      BOOT    ; COLD START BOOT ENTRY
'0003 C3F801'    0246         JP      WARM    ; WARM START CP/M ENTRY
'0006 C3110C'    0247         JP      CONST   ; CHECK CONSOLE STATUS
'0009 C3360C'    0248         JP      CONIN   ; INPUT CHARACTER FROM CONSOLE
'000C C3440C'    0249         JP      CONOUT  ; OUTPUT CHARACTER TO CONSOLE
'000F C3620C'    0250         JP      LIST    ; OUTPUT CHARACTER TO PRINTER
'0012 C3720C'    0251         JP      PUNCH   ; OUTPUT CHARACTER TO PUNCH
'0015 C3830C'    0252         JP      READER  ; INPUT CHARACTER FROM READER
'0018 C3CE02'    0253         JP      HOME    ; HOME CURRENT DISK DRIVE
'001B C3AA04'    0254         JP      SELDSK  ; SELECT A DISK DRIVE
'001E C3B605'    0255         JP      SETTRK  ; SET TRACK NUMBER
'0021 C3BC05'    0256         JP      SETSEC  ; SET SECTOR NUMBER
'0024 C3C605'    0257         JP      SETDMA  ; SET READ/WRITE ADDRESS
'0027 C30A08'    0258         JP      READ    ; READ A SECTOR
'002A C32908'    0259         JP      WRITE   ; WRITE A SECTOR
'002D C3520C'    0260         JP      LISTST  ; TEST STATUS OF PRINTER
'0030 C3C105'    0261         JP      SECTRN  ; TRANSLATE LOGICAL SECTORS
                 0262 ;
                 0263 ;       SYSTEM DRIVES DEFINITION TABLES
                 0264 ;
'>0033           0265 DPBASE  EQU     $       ; BASE OF DISK PARAMETER BLOCKS
                 0266 ;       FDC-880H DPE TABLES
'0033 01000000   0267 DPE0:   DEFW    _CALC,0 ; TRANSLATE TABLE (CALCULATED)
'0037 00000000   0268         DEFW    0,0     ; SCRATCH AREA
'003B DB12EF00'  0269         DEFW    DIRBUF,DPB1 ; DIR BUFF,PARM BLOCK
'003F B3145B13'  0270         DEFW    CSV0,ALV0 ; CHECK, ALLOC VECTORS
'0043 01000000   0271 DPE1:   DEFW    _CALC,0 ; TRANSLATE TABLE (CALCULATED)
'0047 00000000   0272         DEFW    0,0     ; SCRATCH AREA
'004B DB12EF00'  0273         DEFW    DIRBUF,DPB1 ; DIR BUFF,PARM BLOCK
'004F F314A813'  0274         DEFW    CSV1,ALV1 ; CHECK,ALLOC VECTORS
'0053 01000000   0275 DPE2:   DEFW    _CALC,0 ; TRANSLATE TABLE (CALCULATED)
'0057 00000000   0276         DEFW    0,0     ; SCRATCH AREA
'005B DB128201'  0277         DEFW    DIRBUF,DPB7 ; DIR BUFF, PARM BLOCK
'005F 3315F513'  0278         DEFW    CSV2,ALV2 ; CHECK, ALLOC VECTORS
'0063 01000000   0279 DPE3:   DEFW    _CALC,0 ; TRANSLATE TABLE (CALCULATED)
'0067 00000000   0280         DEFW    0,0     ; SCRATCH AREA
'006B DB128201'  0281         DEFW    DIRBUF,DPB7 ; DIR BUFF, PARM BLOCK
'006F 73154214'  0282         DEFW    CSV3,ALV3 ; CHECK, ALLOC VECTORS
                 0283 ;       H88-4 DPE TABLE (CALCULATED)S
'0073 01000000   0284 DPE4:   DEFW    _CALC,0 ; TRANSLATE TABLE (CALCULATED)
'0077 00000000   0285         DEFW    0,0     ; SCRATCH AREA
'007B DB12D701'  0286         DEFW    DIRBUF,DPBH17 ; DIR BUFF,PARM BLOCK
'007F B3158F14'  0287         DEFW    CSV4,ALV4 ; CHECK,ALLOC VECTORS
'0083 01000000   0288 DPE5:   DEFW    _CALC,0 ; TRANSLATE TABLE (CALCULATED)
'0087 00000000   0289         DEFW    0,0     ; SCRATCH AREA
'008B DB12D701'  0290         DEFW    DIRBUF,DPBH17 ; DIR BUFF, PARM BLOCK
'008F C3159B14'  0291         DEFW    CSV5,ALV5 ; CHECK, ALLOC VECTORS
```

```
 ADDR  CODE       STMT SOURCE STATEMENT

'0093  01000000   0292 DPE6:    DEFW    _CALC,0 : TRANSLATE TABLE (CALCULATED)
'0097  00000000   0293          DEFW    0,0      : SCRATCH AREA
'009B  DB12D701'  0294          DEFW    DIRBUF,DPBH17 : DIR BUFF, PARM BLOCK
'009F  D315A714'  0295          DEFW    CSV6,ALV6 : CHECK, ALLOC VECTORS
                  0296 ;
                  0297 ;         UNIT OFFSET CODE FOR CP/M
                  0298 ;
'00A3  00         0299 OFFSET: DEFB    0        : SET FOR NO OFFSET
                  0300 ;
                  0301 ;         ACTIVE PHYSICAL DRIVES TABLE.  LAST THREE ARE H17 DRIVES
                  0302 ;         HOLDS CHARACTERISTIC LIMITATIONS FOR DRIVES
                  0303 ;
'00A4  01         0304 ACTTBL: DEFB    _8INCH+_1STEP+_CYL+_DSIDE+_12MIL : FDC-880H PHYS. DR0
'00A5  01         0305          DEFB    _8INCH+_1STEP+_CYL+_DSIDE+_12MIL : FDC-880H PHYS. DR1
'00A6  23         0306          DEFB    _5INCH+_1STEP+_CYL+_DSIDE+_30MIL : FDC-880H PHYS. DR2
'00A7  23         0307          DEFB    _5INCH+_1STEP+_CYL+_DSIDE+_30MIL : FDC-880H PHYS. DR3
'00A8  03         0308          DEFB    _H17TYP+_H30MIL         : H17 PHYSICAL DRIVE 3
'00A9  03         0309          DEFB    _H17TYP+_H30MIL         : H17 PHYSICAL DRIVE 2
'00AA  03         0310          DEFB    _H17TYP+_H30MIL         : H17 PHYSICAL DRIVE 1
                  0311 ;         ********************************************
                  0312 ;         COLD BOOT PORT INITIALIZATION
                  0313 ;
'00AB  AD00'      0314          DEFW    CBINIT
'00AD  C3EE01'    0315 CBINIT: JP      BOOT
'00B0  00         0316          NOP
'00B1  00         0317          NOP
'00B2  00         0318          NOP
                  0319 ;         ********************************************
                  0320 ;         BIOS PRE-SET I/O PARAMETERS
'00B3  E9         0321 DEFIOB: DEFB    DIOB                :DEFAULT IOBYTE
'00B4  00         0322          DEFB    0                   :CONSOLE TYPE BYTE (RESERVED)
                  0323 ;
                  0324 ; SERIAL DEVICE STRUCTURES       DEFB    PORT$NUMBER
                  0325 ;                                DEFW    CONTROL$WORD
                  0326 ; THE CONTROL$WORD CONTAINS      B15     MAP LOWER TO UPPER CASE
                  0327 ;                                B14-B12 NO. OF NULLS AFTER A CR
                  0328 ;                                B11-B00 BAUD RATE DIVISOR
'00B5  E8         0329 H84PT1: DEFB    H84CRT
'00B6  0C00       0330 CRTBAUD:DEFW    B9600
'00B8  D0         0331 H84PT2: DEFB    H84TTY
'00B9  8001       0332 TTYBAUD:DEFW    B300
'00BB  E0         0333 H84PT3: DEFB    H84LPT
'00BC  6000       0334 LPTBAUD:DEFW    B1200
'00BE  D8         0335 H84PT4: DEFB    H84RDP
'00BF  8001       0336 RDPBAUD:DEFW    B300
'00C1  000000     0337          DEFB    0,0,0               : RESERVED
'00C4  00         0338 DCLPOS: DEFB    0                   :CHECK ON LP OUTPUT STATUS
'00C5  00         0339 TTYCTS: DEFB    0                   :CHARS. TO SEND COUNT FOR TTY
'00C6  00         0340 CRTCTS: DEFB    0                   :                         CRT
'00C7  00         0341 LPTCTS: DEFB    0                   :                         LPT
'00C8  00         0342 MDCTS:  DEFB    0                   :                         MODEM
'00C9  00         0343 DBDCTS: DEFB    0                   :OUTPUT STATE MACHINE FOR DBD
                  0344 ;         FDCH88 NUMBER OF READ/WRITE TRYS
'00CA  05         0345 RWTRYS: DEFB    5        : LSB=TRYS BETWEEN RESTORES
'00CB  0E         0346          DEFB    14       : MSB=TOTAL NUMBER OF TRYS
                  0347 ;         FDC-880H BUSY TIME-OUT DELAY CONSTANT
'00CC  2E         0348 DLYCON: DEFB    NORMAL   : 16TH MILSEC DELAY CONST. (2.048 MHZ)
                  0349 ;
```

```
     (C) C.D.R. SYSTEMS INC.  BIOS   SD SYSTEMS Z80 ASSEMBLER PAGE 0007
 ADDR  CODE       STMT SOURCE STATEMENT

                  0350 :         TEST CLEAR TO SEND FOR PRINTER (AND 20H = DSR)
                  0351 :
'00CD  00         0352 CTSTST: NOP                : SET AS CPL (2FH) IF CTS INVERTED
'00CE  E610       0353         AND     10H      : TEST CLEAR TO SEND
'00D0  C9         0354         RET
                  0355 :
                  0356 :         H17 STORED PARAMETERS
'00D1  FFFFFF     0357 H17TRK: DEFB    0FFH,0FFH,0FFH  : H17 CURRENT TRACK STORAGE
'00D4  FF00       0358 TRKPT:  DEFW    0FFH             : TRACK POINTER
'00D6  00         0359 TSECTR: DEFB    0                : TRANSLATED SECTOR NUMBER
                  0360 :         CURRENT ERROR COUNTS
'00D7  00         0361 SOFTC:  DEFB    0        : R/W SOFT ERROR COUNT FOR SESSION
'00D8  00         0362 HARDC:  DEFB    0        : R/W HARD ERROR COUNT FOR SESSION
                  0363 :         AUTO WARM START CODE 0=NO AUTO WARM, FF=AUTO AVAILABLE
'00D9  00         0364 AUTO:   DEFB    0        : ZERO = AUTO RUN ON WARM START
                  0365 :
                  0366 :         BEGINNING OF DPB TABLES THAT WILL BE TESTED FOR
                  0367 :
'>00DA            0368 DPBTBL  EQU     $        : BEGINNING OF DPB TABLES
                  0369 :
                  ~~0370~~         IF      A8INCH  : FOR 8 INCH CDR TYPE DISKS
                  ~~0371~~ :         8" SINGLE SIDED, SINGLE DENSITY VERSION (STANDARD)
                  ~~0372~~ :
'>00DA            ~~0373~~ DPB0    EQU     $        : DISK PARM BLOCK
'00DA  1A00       ~~0374~~         DEFW    26       : SECTORS PER TRACK
'00DC  03         ~~0375~~         DEFB    3        : BLOCK SHIFT
'00DD  07         ~~0376~~         DEFB    7        : BLOCK MASK
'00DE  00         ~~0377~~         DEFB    0        : EXTENT MASK
'00DF  EF00       ~~0378~~         DEFW    239      : (SIZE SET TO 242 IF NOT BOOTABLE)
'00E1  3F00       ~~0379~~         DEFW    63       : DIRECTORY MASK
'00E3  C0         ~~0380~~ ALS0:   DEFB    192      : ALLOC0
'00E4  00         ~~0381~~         DEFB    0        : ALLOC1
'00E5  1000       ~~0382~~ CSS0:   DEFW    16       : CHECK SIZE
'00E7  0200       ~~0383~~         DEFW    2        : OFFSET
'00E9  C0         ~~0384~~         DEFB    _SSIDE+_SDENS+_8INCH : DTYPE 8", S.D., S.S
'00EA  06         ~~0385~~         DEFB    6        : SECTOR TRANSLATION SKIP COUNT
'00EB  02         ~~0386~~         DEFB    _20MIL   : TRACK STEP TIME CODE
'00EC  1A         ~~0387~~         DEFB    26       : NUMBER OF HOST SECTORS/TRACK
'00ED  01         ~~0388~~         DEFB    1        : COUNT OF 128 BYTE SECTORS/HOSTSEC
'00EE  4C         ~~0389~~         DEFB    4CH      : NUMBER OF TRACKS PER SIDE OF DISK
                  ~~0390~~         ENDIF
                  ~~0391~~ :
                  ~~0392~~         IF      A8INCH  : FOR 8 INCH CDR OR HEATH TYPE DISKS
                  ~~0393~~ :         8" SINGLE SIDED, DOUBLE DENSITY VERSION
                  ~~0394~~ :
'>00EF            ~~0395~~ DPB1    EQU     $        : DISK PARM BLOCK
'00EF  3400       ~~0396~~         DEFW    52       : SECTORS PER TRACK
'00F1  04         ~~0397~~         DEFB    4        : BLOCK SHIFT
'00F2  0F         ~~0398~~         DEFB    15       : BLOCK MASK
'00F3  00         ~~0399~~         DEFB    0        : EXTENT MASK
'00F4  F200       ~~0400~~         DEFW    242      : DISK SIZE - 1
'00F6  7F00       ~~0401~~         DEFW    127      : DIRECTORY MASK
'00F8  C0         ~~0402~~ ALS1:   DEFB    0C0H     : ALLOC0
'00F9  00         ~~0403~~         DEFB    0        : ALLOC1
'00FA  2000       ~~0404~~ CSS1:   DEFW    32       : CHECK SIZE
'00FC  0200       ~~0405~~         DEFW    2        : OFFSET
'00FE  80         ~~0406~~         DEFB    _SSIDE+_DDENS+_8INCH : DTYPE 8", D.D., S.S.
'00FF  09         ~~0407~~         DEFB    9        : SECTOR TRANSLATION SKIP COUNT
```

```
        (C) C.D.R. SYSTEMS INC.  BIOS   SD SYSTEMS Z80 ASSEMBLER PAGE 0008
 ADDR  CODE       STMT SOURCE STATEMENT

 0100  02         0408          DEFB    _20MIL  : TRACK STEP TIME CODE
 0101  1A         0409          DEFB    26      : NUMBER OF HOST SECTORS/TRACK
 0102  02         0410          DEFB    2       : COUNT OF 128 BYTE SECTORS/HOSTSEC
 0103  4C         0411          DEFB    4CH     : NUMBER OF TRACKS PER SIDE OF DISK
                  0412          ENDIF
                  0413 :
                  0414          IF      A8INCH  : FOR/8 INCH CDR OR HEATH TYPE DISKS
                  0415 :        8" SINGLE SIDED, DOUBLE DENSITY EXTENDED
                  0416 :
 0104             0417 DPB2     EQU     $       : DISK PARM BLOCK
 0104  4000       0418          DEFW    64      : SECTORS PER TRACK
 0106  04         0419          DEFB    4       : BLOCK SHIFT
 0107  0F         0420          DEFB    15      : BLOCK MASK
 0108  00         0421          DEFB    0       : EXTENT MASK
 0109  2B01       0422          DEFW    299     : DISK SIZE - 1
 010B  7F00       0423          DEFW    127     : DIRECTORY MASK
 010D  C0         0424 ALS2:    DEFB    0C0H    : ALLOC0
 010E  00         0425          DEFB    0       : ALLOC1
 010F  2000       0426 CSS2:    DEFW    32      : CHECK SIZE
 0111  0200       0427          DEFW    2       : OFFSET
 0113  81         0428          DEFB    _SSIDE+_DDENS+_8INCH+_EXTEND : DTYPE 8",D.D.,S.S.,EX
 0114  01         0429          DEFB    1       : SECTOR TRANSLATION SKIP COUNT
 0115  02         0430          DEFB    _20MIL  : TRACK STEP TIME CODE
 0116  08         0431          DEFB    8       : NUMBER OF HOST SECTORS/TRACK
 0117  08         0432          DEFB    8       : COUNT OF 128 BYTE SECTORS/HOSTSEC
 0118  4C         0433          DEFB    4CH     : NUMBER OF TRACKS PER SIDE OF DISK
                  0434          ENDIF
                  0435 :
                  0436          IF      A8INCH.AND.ACDR8 : FOR 8 INCH CDR TYPE DISKS
                  0437 :        8" DOUBLE SIDED, SINGLE DENSITY
                  0438 :
                  0439 DPB3     EQU     $       : DISK PARM BLOCK
                  0440          DEFW    26      : SECTORS PER TRACK
                  0441          DEFB    4       : BLOCK SHIFT
                  0442          DEFB    15      : BLOCK MASK
                  0443          DEFB    1       : EXTENT MASK
                  0444          DEFW    244     : DISK SIZE - 1 (246 IF NON BOOTABLE)
                  0445          DEFW    127     : DIRECTORY MASK
                  0446 ALS3:    DEFB    0C0H    : ALLOC0
                  0447          DEFB    0       : ALLOC1
                  0448 CSS3:    DEFW    32      : CHECK SIZE
                  0449          DEFW    2       : OFFSET
                  0450          DEFB    _DSIDE+_SDENS+_8INCH : DTYPE 8", D.D., S.S.
                  0451          DEFB    6       : SECTOR TRANSLATION SKIP COUNT
                  0452          DEFB    _12MIL  : TRACK STEP TIME CODE
                  0453          DEFB    26      : NUMBER OF HOST SECTORS/TRACK
                  0454          DEFB    1       : COUNT OF 128 BYTE SECTORS/HOSTSEC
                  0455          DEFB    4CH     : NUMBER OF TRACKS PER SIDE OF DISK
                  0456          ENDIF
                  0457 :
                  0458          IF      A8INCH.AND.ACDR8 : FOR 8 INCH CDR TYPE DISKS
                  0459 :        8" DOUBLE SIDED, DOUBLE DENSITY VERSION
                  0460 :
                  0461 DPB4     EQU     $       : DISK PARM BLOCK
                  0462          DEFW    52      : SECTORS PER TRACK
                  0463          DEFB    4       : BLOCK SHIFT
                  0464          DEFB    15      : BLOCK MASK
                  0465          DEFB    0       : EXTENT MASK
```

```
ADDR  CODE        STMT SOURCE STATEMENT

                  0466          DEFW    493     : DISK SIZE - 1
                  0467          DEFW    127     : DIRECTORY MASK
                  0468 ALS4:    DEFB    0C0H    : ALLOC0
                  0469          DEFB    0       : ALLOC1
                  0470 CSS4:    DEFW    32      : CHECK SIZE
                  0471          DEFW    2       : OFFSET
                  0472          DEFB    _DSIDE+_DDENS+_8INCH : DTYPE 8", D.D., D.S.
                  0473          DEFB    9       : SECTOR TRANSLATION SKIP COUNT
                  0474          DEFB    _6MIL   : TRACK STEP TIME CODE
                  0475          DEFB    26      : NUMBER OF HOST SECTORS/TRACK
                  0476          DEFB    2       : COUNT OF 128 BYTE SECTORS/HOSTSEC
                  0477          DEFB    4CH     : NUMBER OF TRACKS PER SIDE OF DISK
                  0478          ENDIF
                  0479 :
                  0480          IF      A8INCH.AND.ACDR8 : FOR 8 INCH CDR TYPE DISKS
                  0481 :        8" DOUBLE SIDED, DOUBLE DENSITY EXTENDED
                  0482 :
                  0483 DPB5     EQU     $       : DISK PARM BLOCK
                  0484          DEFW    64      : SECTORS PER TRACK
                  0485          DEFB    4       : BLOCK SHIFT
                  0486          DEFB    15      : BLOCK MASK
                  0487          DEFB    0       : EXTENT MASK
                  0488          DEFW    607     : DISK SIZE - 1
                  0489          DEFW    127     : DIRECTORY MASK
                  0490 ALS5:    DEFB    0C0H    : ALLOC0
                  0491          DEFB    0       : ALLOC1
                  0492 CSS5:    DEFW    32      : CHECK SIZE
                  0493          DEFW    2       : OFFSET
                  0494          DEFB    _DSIDE+_DDENS+_8INCH+_EXTEND  : DTYPE 8",D.D.,D.S.,EX
                  0495          DEFB    1       : SECTOR TRANSLATION SKIP COUNT
                  0496          DEFB    _6MIL   : TRACK STEP TIME CODE
                  0497          DEFB    8       : NUMBER OF HOST SECTORS/TRACK
                  0498          DEFB    8       : COUNT OF 128 BYTE SECTORS/HOSTSEC
                  0499          DEFB    4CH     : NUMBER OF TRACKS PER SIDE OF DISK
                  0500          ENDIF
                  0501 :
                  0502          IF      A8INCH.AND.AHZ8 : FOR 8 INCH HEATH/ZENITH TYPE DISKS
                  0503 :        8" DOUBLE SIDED, SINGLE DENSITY
                  0504 :
0119              0505 DPB3     EQU     $       : DISK PARM BLOCK
0119  1A00        0506          DEFW    26      : SECTORS PER TRACK
011B  04          0507          DEFB    4       : BLOCK SHIFT
011C  0F          0508          DEFB    15      : BLOCK MASK
011D  01          0509          DEFB    1       : EXTENT MASK
011E  F400        0510          DEFW    244     : DISK SIZE - 1 (246 IF NON BOOTABLE)
0120  FF00        0511          DEFW    255     : DIRECTORY MASK
0122  F0          0512 ALS3:    DEFB    0F0H    : ALLOC0
0123  00          0513          DEFB    0       : ALLOC1
0124  4000        0514 CSS3:    DEFW    64      : CHECK SIZE
0126  0200        0515          DEFW    2       : OFFSET
0128  40          0516          DEFB    _DSIDE+_SDENS+_8INCH : DTYPE 8", S.D., D.S.
0129  06          0517          DEFB    6       : SECTOR TRANSLATION SKIP COUNT
012A  01          0518          DEFB    _12MIL  : TRACK STEP TIME CODE
012B  1A          0519          DEFB    26      : NUMBER OF HOST SECTORS/TRACK
012C  01          0520          DEFB    1       : COUNT OF 128 BYTE SECTORS/HOSTSEC
012D  4C          0521          DEFB    4CH     : NUMBER OF TRACKS PER SIDE OF DISK
                  0522          ENDIF
                  0523 :
```

```
    (C) C.D.R. SYSTEMS INC.  BIOS   SD SYSTEMS Z80 ASSEMBLER PAGE 0010
 ADDR  CODE        STMT SOURCE STATEMENT

                   ~~0524~~          IF      A8INCH.AND.AHZ8 ; FOR 8 INCH HEATH/ZENITH TYPE DISKS
                   ~~0525~~ ;        8" DOUBLE SIDED, DOUBLE DENSITY VERSION
                   ~~0526~~ ;
'>012E             ~~0527~~ DPB4     EQU     $        ; DISK PARM BLOCK
'012E  3400        ~~0528~~          DEFW    52       ; SECTORS PER TRACK
'0130  04          ~~0529~~          DEFB    4        ; BLOCK SHIFT
'0131  0F          ~~0530~~          DEFB    15       ; BLOCK MASK
'0132  00          ~~0531~~          DEFB    0        ; EXTENT MASK
'0133  ED01        ~~0532~~          DEFW    493      ; DISK SIZE - 1
'0135  FF00        ~~0533~~          DEFW    255      ; DIRECTORY MASK
'0137  F0          ~~0534~~ ALS4:    DEFB    0F0H     ; ALLOC0
'0138  00          ~~0535~~          DEFB    0        ; ALLOC1
'0139  4000        ~~0536~~ CSS4:    DEFW    64       ; CHECK SIZE
'013B  0200        ~~0537~~          DEFW    2        ; OFFSET
'013D  00          ~~0538~~          DEFB    _DSIDE+_DDENS+_8INCH ; DTYPE 8", D.D., D.S.
'013E  09          ~~0539~~          DEFB    9        ; SECTOR TRANSLATION SKIP COUNT
'013F  00          ~~0540~~          DEFB    _6MIL    ; TRACK STEP TIME CODE
'0140  1A          ~~0541~~          DEFB    26       ; NUMBER OF HOST SECTORS/TRACK
'0141  02          ~~0542~~          DEFB    2        ; COUNT OF 128 BYTE SECTORS/HOSTSEC
'0142  4C          ~~0543~~          DEFB    4CH      ; NUMBER OF TRACKS PER SIDE OF DISK
                   ~~0544~~          ENDIF
                   ~~0545~~ ;
                   ~~0546~~          IF      A8INCH.AND.AHZ8 ; FOR 8" HEATH/ZENITH TYPE DISKS
                   ~~0547~~ ;        8" DOUBLE SIDED, DOUBLE DENSITY EXTENDED
                   ~~0548~~ ;
'>0143             ~~0549~~ DPB5     EQU     $        ; DISK PARM BLOCK
'0143  4000        ~~0550~~          DEFW    64       ; SECTORS PER TRACK
'0145  04          ~~0551~~          DEFB    4        ; BLOCK SHIFT
'0146  0F          ~~0552~~          DEFB    15       ; BLOCK MASK
'0147  00          ~~0553~~          DEFB    0        ; EXTENT MASK
'0148  5F02        ~~0554~~          DEFW    607      ; DISK SIZE - 1
'014A  FF00        ~~0555~~          DEFW    255      ; DIRECTORY MASK
'014C  F0          ~~0556~~ ALS5:    DEFB    0F0H     ; ALLOC0
'014D  00          ~~0557~~          DEFB    0        ; ALLOC1
'014E  4000        ~~0558~~ CSS5:    DEFW    64       ; CHECK SIZE
'0150  0200        ~~0559~~          DEFW    2        ; OFFSET
'0152  01          ~~0560~~          DEFB    _DSIDE+_DDENS+_8INCH+_EXTEND  ; DTYPE 8",D.D.,D.S.,EX
'0153  01          ~~0561~~          DEFB    1        ; SECTOR TRANSLATION SKIP COUNT
'0154  00          ~~0562~~          DEFB    _6MIL    ; TRACK STEP TIME CODE
'0155  08          ~~0563~~          DEFB    8        ; NUMBER OF HOST SECTORS/TRACK
'0156  08          ~~0564~~          DEFB    8        ; COUNT OF 128 BYTE SECTORS/HOSTSEC
'0157  4C          ~~0565~~          DEFB    4CH      ; NUMBER OF TRACKS PER SIDE OF DISK
                   ~~0566~~          ENDIF
                   ~~0567~~ ;
                   0568          IF      A5INCH  ; FOR 5 INCH SINGLE DENSITY DISKS
                   0569 ;        5.25" SINGLE SIDED, SINGLE DENSITY VERSION
                   0570 ;
'>0158             0571 DPB6     EQU     $        ; DISK PARM BLOCK
'0158  1200        0572          DEFW    18       ; NUMBER OF SECTORS PER TRACK
'015A  03          0573          DEFB    3        ; BLOCK SHIFT
'015B  07          0574          DEFB    7        ; BLOCK MASK
'015C  00          0575          DEFB    0        ; EXTENT MASK
'015D  5000        0576          DEFW    80       ; DISK SIZE - 1
'015F  3F00        0577          DEFW    63       ; DIRECTORY MASK
'0161  C0          0578 ALS6:    DEFB    192      ; ALLOC 0
'0162  00          0579          DEFB    0        ; ALLOC 1
'0163  1000        0580 CSS6:    DEFW    16       ; CHECK SIZE
'0165  0300        0581          DEFW    3        ; OFFSET
```

```
 ADDR  CODE       STMT SOURCE STATEMENT

'0167  E0         0582           DEFB    _SSIDE+_SDENS+_5INCH ; DTYPE 5.25", S.D., S.S.
'0168  05         0583           DEFB    5        ; SECTOR TRANSLATION SKIP COUNT
'0169  02         0584           DEFB    _20MIL   ; TRACK STEP TIME CODE
'016A  12         0585           DEFB    18       ; NUMBER OF HOST SECTORS/TRACK
'016B  01         0586           DEFB    1        ; COUNT OF 128 BYTE SECTORS/HOSTSEC
'016C  27         0587           DEFB    39       ; NUMBER OF TRACKS PER DISK SIDE
                  0588           ENDIF
                  0589 ;
                  0590           IF      A5INCH.AND.ASDE5  ; 5" EXTENDED SINGLE DENSITY DISKS
                  0591 ;         5.25" SINGLE SIDED, EXTENDED SINGLE DENSITY VERSION
                  0592 ;
'>016D            0593 DPB6B     EQU     $        ; DISK PARM BLOCK
'016D  1400       0594           DEFW    20       ; NUMBER OF SECTORS PER TRACK
'016F  03         0595           DEFB    3        ; BLOCK SHIFT
'0170  07         0596           DEFB    7        ; BLOCK MASK
'0171  00         0597           DEFB    0        ; EXTENT MASK
'0172  5B00       0598           DEFW    91       ; DISK SIZE - 1
'0174  3F00       0599           DEFW    63       ; DIRECTORY MASK
'0176  C0         0600 ALS6B:    DEFB    192      ; ALLOC 0
'0177  00         0601           DEFB    0        ; ALLOC 1
'0178  1000       0602 CSS6B:    DEFW    16       ; CHECK SIZE
'017A  0300       0603           DEFW    3        ; OFFSET
'017C  E5         0604           DEFB    _SSIDE+_SDENS+_5INCH+_EXTEND+_SAMETO ; 5.25",S.D.,S.S.
'017D  01         0605           DEFB    1        ; SECTOR TRANSLATION SKIP COUNT
'017E  02         0606           DEFB    _20MIL   ; TRACK STEP TIME CODE
'017F  0A         0607           DEFB    10       ; NUMBER OF HOST SECTORS/TRACK
'0180  02         0608           DEFB    2        ; COUNT OF 128 BYTE SECTORS/HOSTSEC
'0181  27         0609           DEFB    39       ; NUMBER OF TRACKS PER DISK SIDE
                  0610           ENDIF
                  0611 ;
                  0612           IF      A5INCH.AND.A40DS  ; FOR 5 INCH SINGLE DENSITY DISKS
                  0613 ;         5.25" DOUBLE SIDED, SINGLE DENSITY VERSION
                  0614 ;
                  0615 DPB6C     EQU     $        ; DISK PARM BLOCK
                  0616           DEFW    18       ; NUMBER OF SECTORS PER TRACK
                  0617           DEFB    3        ; BLOCK SHIFT
                  0618           DEFB    7        ; BLOCK MASK
                  0619           DEFB    0        ; EXTENT MASK
                  0620           DEFW    172      ; DISK SIZE - 1
                  0621           DEFW    125      ; DIRECTORY MASK
                  0622 ALS6C:    DEFB    0C0H     ; ALLOC 0
                  0623           DEFB    0        ; ALLOC 1
                  0624 CSS6C:    DEFW    32       ; CHECK SIZE
                  0625           DEFW    3        ; OFFSET
                  0626           DEFB    _DSIDE+_SDENS+_5INCH ; DTYPE 5.25", S.D., S.S.
                  0627           DEFB    5        ; SECTOR TRANSLATION SKIP COUNT
                  0628           DEFB    _20MIL   ; TRACK STEP TIME CODE
                  0629           DEFB    18       ; NUMBER OF HOST SECTORS/TRACK
                  0630           DEFB    1        ; COUNT OF 128 BYTE SECTORS/HOSTSEC
                  0631           DEFB    39       ; NUMBER OF TRACKS PER DISK SIDE
                  0632           ENDIF
                  0633 ;
                  0634           IF      A5INCH.AND.ASDE5.AND.A40DS ; 5" EXTENDED SINGLE DENS.
                  0635 ;         5.25" DOUBLE SIDED, EXTENDED SINGLE DENSITY VERSION
                  0636 ;
                  0637 DPB6D     EQU     $        ; DISK PARM BLOCK
                  0638           DEFW    20       ; NUMBER OF SECTORS PER TRACK
                  0639           DEFB    3        ; BLOCK SHIFT
```

```
ADDR  CODE      STMT SOURCE STATEMENT

                0640            DEFB    7         ; BLOCK MASK
                0641            DEFB    0         ; EXTENT MASK
                0642            DEFW    191       ; DISK SIZE - 1
                0643            DEFW    255       ; DIRECTORY MASK
                0644 ALS6D:     DEFB    240       ; ALLOC 0
                0645            DEFB    0         ; ALLOC 1
                0646 CSS6D:     DEFW    32        ; CHECK SIZE
                0647            DEFW    3         ; OFFSET
                0648            DEFB    _DSIDE+_SDENS+_5INCH+_EXTEND+_SAMETO ; 5.25",S.D.,S.S.
                0649            DEFB    1         ; SECTOR TRANSLATION SKIP COUNT
                0650            DEFB    _20MIL    ; TRACK STEP TIME CODE
                0651            DEFB    10        ; NUMBER OF HOST SECTORS/TRACK
                0652            DEFB    2         ; COUNT OF 128 BYTE SECTORS/HOSTSEC
                0653            DEFB    39        ; NUMBER OF TRACKS PER DISK SIDE
                0654            ENDIF
                0655 ;
                0656            IF      A5INCH.AND.ACDR5 ; FOR 5 INCH CDR TYPE DISKS
                0657 ;          5.25" SINGLE SIDED, DOUBLE DENSITY VERSION (40 TRACKS)
                0658 ;
                0659 DPB7       EQU     $         ; DISK PARM BLOCK
                0660            DEFW    36        ; NUMBER OF SECTORS PER TRACK
                0661            DEFB    3         ; BLOCK SHIFT
                0662            DEFB    7         ; BLOCK MASK
                0663            DEFB    0         ; EXTENT MASK
                0664            DEFW    165       ; DISK SIZE - 1
                0665            DEFW    63        ; DIRECTORY MASK
                0666 ALS7:      DEFB    192       ; ALLOC 0
                0667            DEFB    0         ; ALLOC 1
                0668 CSS7:      DEFW    16        ; CHECK SIZE
                0669            DEFW    3         ; OFFSET
                0670            DEFB    _SSIDE+_DDENS+_5INCH ; DTYPE 5.25", D.D., S.S.
                0671            DEFB    8         ; SECTOR TRANSLATION SKIP COUNT
                0672            DEFB    _30MIL    ; TRACK STEP TIME CODE
                0673            DEFB    18        ; NUMBER OF HOST SECTORS/TRACK
                0674            DEFB    2         ; COUNT OF 128 BYTE SECTORS/HOSTSEC
                0675            DEFB    39        ; NUMBER OF TRACKS PER DISK SIDE
                0676            ENDIF
                0677 ;
                0678            IF      A5INCH.AND.AHZ5 ; FOR 5 INCH HEATH/ZENITH DISKS
                0679 ;          5.25" SINGLE SIDED, DOUBLE DENSITY VERSION (40 TRACKS)
                0680 ;
'0182           0681 DPB7       EQU     $         ; DISK PARM BLOCK
'0182 2000      0682            DEFW    32        ; NUMBER OF SECTORS PER TRACK
'0184 03        0683            DEFB    3         ; BLOCK SHIFT
'0185 07        0684            DEFB    7         ; BLOCK MASK
'0186 00        0685            DEFB    0         ; EXTENT MASK
'0187 9700      0686            DEFW    151       ; DISK SIZE - 1
'0189 7F00      0687            DEFW    127       ; DIRECTORY MASK
'018B F0        0688 ALS7:      DEFB    0F0H      ; ALLOC 0
'018C 00        0689            DEFB    0         ; ALLOC 1
'018D 2000      0690 CSS7:      DEFW    32        ; CHECK SIZE
'018F 0200      0691            DEFW    2         ; OFFSET
'0191 A4        0692            DEFB    _SSIDE+_DDENS+_5INCH+_SAMETO ; DTYPE 5.25",D.D.,S.S.
'0192 01        0693            DEFB    1         ; SECTOR TRANSLATION SKIP COUNT
'0193 03        0694            DEFB    _30MIL    ; TRACK STEP TIME CODE
'0194 10        0695            DEFB    16        ; NUMBER OF HOST SECTORS/TRACK
'0195 02        0696            DEFB    2         ; COUNT OF 128 BYTE SECTORS/HOSTSEC
'0196 27        0697            DEFB    39        ; NUMBER OF TRACKS PER DISK SIDE
```

```
 ADDR  CODE       STMT SOURCE STATEMENT

                  0698            ENDIF
                  0699 ;
                  0700            IF      A5INCH.AND.ACDR5E ; FOR 5 INCH CDR EXTENDED DISKS
                  0701 ;          5.25" SINGLE SIDED, DOUBLE DENSITY EXTENDED VERSION
                  0702 ;
'>0197            0703 DPB8       EQU     $         ; DISK PARM BLOCK
'0197  2800       0704            DEFW    40        ; NUMBER OF SECTORS PER TRACK
'0199  03         0705            DEFB    3         ; BLOCK SHIFT
'019A  07         0706            DEFB    7         ; BLOCK MASK
'019B  00         0707            DEFB    0         ; EXTENT MASK
'019C  B800       0708            DEFW    184       ; DISK SIZE - 1
'019E  3F00       0709            DEFW    63        ; DIRECTORY MASK
'01A0  C0         0710 ALS8:      DEFB    192       ; ALLOC 0
'01A1  00         0711            DEFB    0         ; ALLOC 1
'01A2  1000       0712 CSS8:      DEFW    16        ; CHECK SIZE
'01A4  0300       0713            DEFW    3         ; OFFSET
'01A6  A1         0714            DEFB    _SSIDE+_DDENS+_5INCH+_EXTEND ; 5.25",D.D.,S.S.,EX
'01A7  03         0715            DEFB    3         ; SECTOR TRANSLATION SKIP COUNT
'01A8  03         0716            DEFB    3         ; TRACK STEP TIME CODE
'01A9  0A         0717            DEFB    10        ; NUMBER OF HOST SECTORS/TRACK
'01AA  04         0718            DEFB    4         ; COUNT OF 128 BYTE SECTORS/HOSTSEC
'01AB  27         0719            DEFB    39        ; NUMBER OF TRACKS PER DISK SIDE
                  0720            ENDIF
                  0721 ;
                  0722            IF      A5INCH.AND.AHZ5E ; FOR 5 INCH HEATH/ZENITH EXTENDED
                  0723 ;          5.25" SINGLE SIDED, DOUBLE DENSITY EXTENDED VERSION
                  0724 ;
                  0725 DPB8       EQU     $         ; DISK PARM BLOCK
                  0726            DEFW    40        ; NUMBER OF SECTORS PER TRACK
                  0727            DEFB    3         ; BLOCK SHIFT
                  0728            DEFB    7         ; BLOCK MASK
                  0729            DEFB    0         ; EXTENT MASK
                  0730            DEFW    189       ; DISK SIZE - 1
                  0731            DEFW    127       ; DIRECTORY MASK
                  0732 ALS8:      DEFB    0F0H      ; ALLOC 0
                  0733            DEFB    0         ; ALLOC 1
                  0734 CSS8:      DEFW    32        ; CHECK SIZE
                  0735            DEFW    2         ; OFFSET
                  0736            DEFB    _SSIDE+_DDENS+_5INCH+_EXTEND+_SAMETO ; 5.25",D.D.,S.S.,EX
                  0737            DEFB    1         ; SECTOR TRANSLATION SKIP COUNT
                  0738            DEFB    _30MIL    ; TRACK STEP TIME CODE
                  0739            DEFB    5         ; NUMBER OF HOST SECTORS/TRACK
                  0740            DEFB    8         ; COUNT OF 128 BYTE SECTORS/HOSTSEC
                  0741            DEFB    39        ; NUMBER OF TRACKS PER DISK SIDE
                  0742            ENDIF
                  0743 ;
                  0744            IF      A5INCH.AND.ACDR5.AND.A80DS ; FOR 5 INCH CDR TYPE DISKS
                  0745 ;          5.25" DOUBLE SIDED, DOUBLE DENSITY 80 TRACKS
                  0746 ;
                  0747 DPBA       EQU     $         ; DISK PARM BLOCK
                  0748            DEFW    36        ; NUMBER OF SECTORS PER TRACK
                  0749            DEFB    4         ; BLOCK SHIFT
                  0750            DEFB    15        ; BLOCK MASK
                  0751            DEFB    0         ; EXTENT MASK
                  0752            DEFW    352       ; DISK SIZE - 1
                  0753            DEFW    127       ; DIRECTORY MASK
                  0754 ALSA:      DEFB    0C0H      ; ALLOC 0
                  0755            DEFB    0         ; ALLOC 1
 ADDR  CODE       STMT SOURCE STATEMENT

                  0756 CSSA:    DEFW    32       : CHECK SIZE
                  0757          DEFW    3        ; OFFSET
                  0758          DEFB    _DSIDE+_DDENS+_5INCH : DTYPE 5.25", D.D., D.S.
                  0759          DEFB    8        ; SECTOR TRANSLATION SKIP COUNT
                  0760          DEFB    _20MIL  : TRACK STEP TIME CODE
                  0761          DEFB    18       ; NUMBER OF HOST SECTORS/TRACK
                  0762          DEFB    2        : COUNT OF 128 BYTE SECTORS/HOSTSEC
                  0763          DEFB    79       ; NUMBER OF TRACKS PER DISK SIDE
                  0764          ENDIF
                  0765 ;
                  0766          IF      A5INCH.AND.AHZ5.AND.A8ODS ; 5 INCH HEATH/ZENITH DISKS
                  0767 :        5.25" DOUBLE SIDED, DOUBLE DENSITY 80 TRACKS
                  0768 ;
'01AC             0769 DPBA     EQU     $        ; DISK PARM BLOCK
'01AC  2000       0770          DEFW    32       : NUMBER OF SECTORS PER TRACK
'01AE  04         0771          DEFB    4        : BLOCK SHIFT
'01AF  0F         0772          DEFB    15       : BLOCK MASK
'01B0  00         0773          DEFB    0        ; EXTENT MASK
'01B1  3B01       0774          DEFW    315      : DISK SIZE - 1
'01B3  FF00       0775          DEFW    255      : DIRECTORY MASK
'01B5  F0         0776 ALSA:    DEFB    0F0H     : ALLOC 0
'01B6  00         0777          DEFB    0        : ALLOC 1
'01B7  4000       0778 CSSA:    DEFW    64       : CHECK SIZE
'01B9  0200       0779          DEFW    2        : OFFSET
'01BB  24         0780          DEFB    _DSIDE+_DDENS+_5INCH+_SAMETO : DTYPE 5.25",D.D.,D.S.
'01BC  01         0781          DEFB    1        ; SECTOR TRANSLATION SKIP COUNT
'01BD  00         0782          DEFB    _6MIL   : TRACK STEP TIME CODE
'01BE  10         0783          DEFB    16       : NUMBER OF HOST SECTORS/TRACK
'01BF  02         0784          DEFB    2        : COUNT OF 128 BYTE SECTORS/HOSTSEC
'01C0  4F         0785          DEFB    79       : NUMBER OF TRACKS PER DISK SIDE
                  0786          ENDIF
                  0787 :
                  0788          IF      A5INCH.AND.A8ODS.AND.ACDR5E ; FOR 5 INCH CDR TYPE DISKS
                  0789 :        5.25" DOUBLE SIDED, DOUBLE DENSITY EXTENDED 80 TRACKS
                  0790 ;
'01C1             0791 DPBB     EQU     $        ; DISK PARM BLOCK
'01C1  2800       0792          DEFW    40       ; NUMBER OF SECTORS PER TRACK
'01C3  04         0793          DEFB    4        ; BLOCK SHIFT
'01C4  0F         0794          DEFB    15       ; BLOCK MASK
'01C5  00         0795          DEFB    0        ; EXTENT MASK
'01C6  8701       0796          DEFW    391      ; DISK SIZE - 1
'01C8  7F00       0797          DEFW    127      ; DIRECTORY MASK
'01CA  C0         0798 ALSB:    DEFB    0C0H     : ALLOC 0
'01CB  00         0799          DEFB    0        ; ALLOC 1
'01CC  2000       0800 CSSB:    DEFW    32       : CHECK SIZE
'01CE  0300       0801          DEFW    3        : OFFSET
'01D0  21         0802          DEFB    _DSIDE+_DDENS+_5INCH+_EXTEND : DTYPE 5.25",D.D.,D.S.,EX
'01D1  03         0803          DEFB    3        ; SECTOR TRANSLATION SKIP COUNT
'01D2  02         0804          DEFB    _20MIL  : TRACK STEP TIME CODE
'01D3  0A         0805          DEFB    10       ; NUMBER OF HOST SECTORS/TRACK
'01D4  04         0806          DEFB    4        : COUNT OF 128 BYTE SECTORS/HOSTSEC
'01D5  4F         0807          DEFB    79       : NUMBER OF TRACKS PER DISK SIDE
                  0808          ENDIF
                  0809 ;
                  0810          IF      A5INCH.AND.A8ODS.AND.AHZ5E : FOR 5 INCH HEATH/ZENITH
                  0811 :        5.25" DOUBLE SIDED, DOUBLE DENSITY EXTENDED 80 TRACKS
                  0812 :
                  0813 DPBB     EQU     $        ; DISK PARM BLOCK
```

```
             0814           DEFW    40       : NUMBER OF SECTORS PER TRACK
             0815           DEFB    4        : BLOCK SHIFT
             0816           DEFB    15       : BLOCK MASK
             0817           DEFB    0        : EXTENT MASK
             0818           DEFW    394      : DISK SIZE - 1
             0819           DEFW    255      : DIRECTORY MASK
             0820 ALSB:     DEFB    0F0H     : ALLOC 0
             0821           DEFB    0        : ALLOC 1
             0822 CSSB:     DEFW    64       : CHECK SIZE
             0823           DEFW    2        : OFFSET
             0824           DEFB    _DSIDE+_DDENS+_5INCH+_EXTEND+_SAMETO : 5.25",D.D.,D.S.,E
             0825           DEFB    1        : SECTOR TRANSLATION SKIP COUNT
             0826           DEFB    _12MIL   : TRACK STEP TIME CODE
             0827           DEFB    5        : NUMBER OF HOST SECTORS/TRACK
             0828           DEFB    8        : COUNT OF 128 BYTE SECTORS/HOSTREC
             0829           DEFB    79       : NUMBER OF TRACKS PER DISK SIDE
             0830           ENDIF
             ~~0831 :~~
             0832           IF      A5INCH.AND.ACDRS.AND.A40DS : FOR 5 INCH CDRS TYPE DISKS
             0833 :         5.25" DOUBLE SIDED, DOUBLE DENSITY 40 TRACKS
             0834 :
             0835 DPBA      EQU     $        : DISK PARM BLOCK
             0836           DEFW    36       : NUMBER OF SECTORS PER TRACK
             0837           DEFB    4        : BLOCK SHIFT
             0838           DEFB    15       : BLOCK MASK
             0839           DEFB    1        : EXTENT MASK
             0840           DEFW    172      : DISK SIZE - 1
             0841           DEFW    127      : DIRECTORY MASK
             0842 ALSA:     DEFB    0C0H     : ALLOC 0
             0843           DEFB    0        : ALLOC 1
             0844 CSSA:     DEFW    32       : CHECK SIZE
             0845           DEFW    3        : OFFSET
             0846           DEFB    _DSIDE+_DDENS+_5INCH : DTYPE 5.25", D.D., D.S.
             0847           DEFB    8        : SECTOR TRANSLATION SKIP COUNT
             0848           DEFB    _20MIL   : TRACK STEP TIME CODE
             0849           DEFB    18       : NUMBER OF HOST SECTORS/TRACK
             0850           DEFB    2        : COUNT OF 128 BYTE SECTORS/HOSTREC
             0851           DEFB    39       : NUMBER OF TRACKS PER DISK SIDE
             0852           ENDIF
             0853 :
             0854           IF      A5INCH.AND.AHZ5.AND.A40DS : 5 INCH HEATH/ZENITH DISKS
             0855 :         5.25" DOUBLE SIDED, DOUBLE DENSITY 40 TRACKS
             0856 :
             0857 DPBA      EQU     $        : DISK PARM BLOCK
             0858           DEFW    32       : NUMBER OF SECTORS PER TRACK
             0859           DEFB    4        : BLOCK SHIFT
             0860           DEFB    15       : BLOCK MASK
             0861           DEFB    0        : EXTENT MASK
             0862           DEFW    155      : DISK SIZE - 1
             0863           DEFW    255      : DIRECTORY MASK
             0864 ALSA:     DEFB    0F0H     : ALLOC 0
             0865           DEFB    0        : ALLOC 1
             0866 CSSA:     DEFW    64       : CHECK SIZE
             0867           DEFW    2        : OFFSET
             0868           DEFB    _DSIDE+_DDENS+_5INCH+_SAMETO : DTYPE 5.25",D.D.,D.S.
             0869           DEFB    1        : SECTOR TRANSLATION SKIP COUNT
             0870           DEFB    _20MIL   : TRACK STEP TIME CODE
             0871           DEFB    16       : NUMBER OF HOST SECTORS/TRACK
```

```
ADDR  CODE       STMT SOURCE STATEMENT

                 0872          DEFB    2        : COUNT OF 128 BYTE SECTORS/HOSTSEC
                 0873          DEFB    39       : NUMBER OF TRACKS PER DISK SIDE
                 0874          ENDIF
                 0875 ;
                 0876          IF      A5INCH.AND.A40DS.AND.ACDR5 : FOR 5 INCH CDR DISKS
                 0877 ;        5.25" DOUBLE SIDED, DOUBLE DENSITY EXTENDED 40 TRACKS
                 0878 ;
                 0879 DPBB     EQU     $        : DISK PARM BLOCK
                 0880          DEFW    40       : NUMBER OF SECTORS PER TRACK
                 0881          DEFB    4        : BLOCK SHIFT
                 0882          DEFB    15       : BLOCK MASK
                 0883          DEFB    1        : EXTENT MASK
                 0884          DEFW    191      : DISK SIZE - 1
                 0885          DEFW    127      : DIRECTORY MASK
                 0886 ALSB:    DEFB    0C0H     : ALLOC 0
                 0887          DEFB    0        : ALLOC 1
                 0888 CSSB:    DEFW    32       : CHECK SIZE
                 0889          DEFW    3        : OFFSET
                 0890          DEFB    _DSIDE+_DDENS+_5INCH+_EXTEND : DTYPE 5.25".D.D.,D.S.,EX
                 0891          DEFB    3        : SECTOR TRANSLATION SKIP COUNT
                 0892          DEFB    _12MIL   : TRACK STEP TIME CODE
                 0893          DEFB    10       : NUMBER OF HOST SECTORS/TRACK
                 0894          DEFB    4        : COUNT OF 128 BYTE SECTORS/HOSTSEC
                 0895          DEFB    39       : NUMBER OF TRACKS PER DISK SIDE
                 0896          ENDIF
                 0897 ;
                 0898          IF      A5INCH.AND.A40DS.AND.AHZ5 : FOR 5 INCH HEATH/ZENITH
                 0899 ;        5.25" DOUBLE SIDED, DOUBLE DENSITY EXTENDED 40 TRACKS
                 0900 ;
                 0901 DPBB     EQU     $        : DISK PARM BLOCK
                 0902          DEFW    40       : NUMBER OF SECTORS PER TRACK
                 0903          DEFB    4        : BLOCK SHIFT
                 0904          DEFB    15       : BLOCK MASK
                 0905          DEFB    0        : EXTENT MASK
                 0906          DEFW    194      : DISK SIZE - 1
                 0907          DEFW    255      : DIRECTORY MASK
                 0908 ALSB:    DEFB    0F0H     : ALLOC 0
                 0909          DEFB    0        : ALLOC 1
                 0910 CSSB:    DEFW    64       : CHECK SIZE
                 0911          DEFW    2        : OFFSET
                 0912          DEFB    _DSIDE+_DDENS+_5INCH+_EXTEND+_SAMETO : 5.25".D.D.,D.S.,EX
                 0913          DEFB    1        : SECTOR TRANSLATION SKIP COUNT
                 0914          DEFB    _12MIL   : TRACK STEP TIME CODE
                 0915          DEFB    5        : NUMBER OF HOST SECTORS/TRACK
                 0916          DEFB    8        : COUNT OF 128 BYTE SECTORS/HOSTSEC
                 0917          DEFB    39       : NUMBER OF TRACKS PER DISK SIDE
                 0918          ENDIF
                 0919 ;
                 0920 ;        END OF DPB TABLES  SET FOR TABLE SEARCHES
                 0921 ;
01D6  00         0922          DEFB    0        : END OF FDC-880H TABLES
                 0923 ;
                 0924          IF      AHSTYP  : HARD SECTORED TABLE
                 0925 ;        H17 5.25" SINGLE SIDED, HARD SECTORED FORMAT
                 0926 ;
01D7             0927 DPBH17   EQU     $        : DISK PARM BLOCK
01D7  1400       0928          DEFW    20       : NUMBER OF SECTORS PER TRACK
01D9  03         0929          DEFB    3        : BLOCK SHIFT
```

```
'01DA  07        0930          DEFB    7       : BLOCK MASK
'01DB  00        0931          DEFB    0       : EXTENT MASK
'01DC  5B00      0932          DEFW    91      : DISK SIZE - 1
'01DE  3F00      0933          DEFW    63      : DIRECTORY MASK
'01E0  C0        0934 ALSH17:  DEFB    192     : ALLOC 0
'01E1  00        0935          DEFB    0       : ALLOC 1
'01E2  1000      0936 CSSH17:  DEFW    16      : CHECK SIZE
'01E4  0300      0937          DEFW    3       : OFFSET
'01E6  F4        0938          DEFB    _H17TYP : DTYPE 5.25". H.S., S.S.
'01E7  04        0939          DEFB    4       : SECTOR TRANSLATION SKIP COUNT
'01E8  0F        0940          DEFB    STEPR   : H17 CONTROLLER STEP RATE
'01E9  0A        0941          DEFB    10      : NUMBER OF HOST SECTORS/TRACK
'01EA  02        0942          DEFB    2       : COUNT OF 128 BYTE SECTORS/HOSTSEC
'01EB  27        0943          DEFB    39      : NUMBER OF TRACKS PER DISK SIDE
                 0944          ENDIF
                 0945
                 0946 :
                 0947 :        INCLUDE THE FDC-880H CONTROLLER MODULE
                 0948 :
                 0949          INCLUDE CDRBB.SRC
                +0001 :
                +0002 :        COMMAND SAVE AND COMMAND ERROR MASK AREA
                +0003 :
'01EC  00       +0004 CMDSV:   DEFB    0       : COMMAND STORAGE
'01ED  00       +0005 ERRMSK:  DEFB    0       : ERROR MASK
                +0006 :
                +0007 :        COLD START CP/M BOOT
                +0008 :
'01EE  3AD30E'  +0009 BOOT:    LD      A,(SMSG) : TEST IF BOOTED FROM PROM
'01F1  3C       +0010          INC     A       : RESULT IS ZERO IF PROM BOOT
'01F2  CAD40E'  +0011          JP      Z,PBOOT : BRANCH FOR PROM BOOT IF ZERO RESULT
'01F5  CD670E'  +0012          CALL    CRT1A   : INITIALIZE PORTS
                +0013 :
                +0014 :        WARM START CP/M ENTRY
                +0015 :
                +0016 :
'01F8  FD21FFFF'+0017 WARM:    LD      IY,BIOSST-1 : SET BIOS START LOCATION
'01FC  31FF15'  +0018          LD      SP,BSTACK : SET BOOT STACK
'01FF  AF       +0019          XOR     A       : SET LOGICAL DRIVE 0
'0200  324300   +0020          LD      (UNIT),A
'0203  32D70E'  +0021          LD      (DISKNO),A
'0206  326109'  +0022          LD      (HSTACT),A : SET HOST BUFFER INACTIVE
'0209  325909'  +0023          LD      (UNACNT),A : SET FOR NO UNALLOCATED
'020C  325F09'  +0024          LD      (HSTWRT),A : CLEAR WRITE PENDING FLAG
'020F  CDE102'  +0025          CALL    DSKTYP  : GET THE DRIVE TYPE
'0212  2071     +0026          JR      NZ,BOOTE : BRANCH IF ERROR
'0214  210200   +0027          LD      HL,2    : SET FOR TYPE TEST TRACK
'0217  224500   +0028          LD      (TRACK),HL
'021A  3E03     +0029          LD      A,3     : SET SECTOR
'021C  324400   +0030          LD      (SECTR),A
'021F  CDD105'  +0031          CALL    RDHOST  : READ A SECTOR (NORMAL DRIVE TYPE)
'0222  2061     +0032          JR      NZ,BOOTE : BRANCH IF ERROR
'0224  11D80E'  +0033          LD      DE,HSTBUF : SET LOC. FOR TEMP. STORAGE
'0227  210000'  +0034          LD      HL,BIOSST : WITH FRONT PART OF BIOS
'022A  010004   +0035          LD      BC,400H : SIZE OF AREA TO SAVE
'022D  EDB0     +0036          LDIR
'022F  AF       +0037          XOR     A
'0230  224500   +0038          LD      (TRACK),HL
```

```
 ADDR  CODE       STMT SOURCE STATEMENT

                  +0039 :
                  +0040          IF     ADREMOD : IF TRANSLATION TABLE CAPABILLITY
                  +0041          LD     (TRANCD),A : SET FOR SECTOR TRANSLATION
                  +0042          ENDIF
                  +0043 :
'0233  2100EA'    +0044          LD     HL,BIOSST-1600H : SET START OF CP/M
'0236  22D90E'    +0045          LD     (HSTLOC),HL : PLACE IN HOST READ LOCATION
'0239  CDDD05'    +0046 LOADER:  CALL   RWHOSO : READ A SECTOR (NEW TRANSFER LOC.)
'023C  2047       +0047          JR     NZ,BOOTE : BRANCH IF ERROR
'023E  CD9C02'    +0048          CALL   SECINC  : SET FOR NEXT SECTOR
'0241  FDE5       +0049          PUSH   IY      : GET MAX MEMORY LIMIT
'0243  E1         +0050          POP    HL
'0244  ED5BD90E'  +0051          LD     DE,(HSTLOC)
'0248  ED52       +0052          SBC    HL,DE   : TEST IF CP/M IN MEMORY
'024A  30ED       +0053          JR     NC,LOADER : BRANCH UNTIL READY
'024C  110000'    +0054          LD     DE,BIOSST : SET FRONT PART OF BIOS
'024F  21DB0E'    +0055          LD     HL,HSTBUF : WITH LOC. FOR TEMP. STORAGE
'0252  010004     +0056          LD     BC,400H : SIZE OF STORAGE TO SET
'0255  EDB0       +0057          LDIR
'0257  2106F2'    +0058 BOOTEN:  LD     HL,BIOSST-0DFAH : SET BDOS OFFSET
'025A  220600     +0059          LD     (RAMST+6),HL
'025D  210300'    +0060          LD     HL,BIOSST+3 : SET BIOS START
'0260  220100     +0061          LD     (RAMST+1),HL
'0263  3EC3       +0062          LD     A,0C3H  : SET CP/M RETURN JUMP INSTRUCTIONS
'0265  320000     +0063          LD     (RAMST),A : WARM START BRANCH
'0268  320500     +0064          LD     (RAMST+5),A : BDOS BRANCH
'026B  0E80       +0065          LD     C,80H   : SET DMA ADDRESS
'026D  CDC605'    +0066          CALL   SETDMA
'0270  AF         +0067          XOR    A
'0271  57         +0068          LD     D,A     : SET FOR CP/M ENTRY
'0272  5F         +0069          LD     E,A
'0273  3A0400     +0070          LD     A,(DEFALT) : SET CURRENT DEFAULT
'0276  4F         +0071          LD     C,A
'0277  32D70E'    +0072          LD     (DISKNO),A : SET AS DESIRED DISK
'027A  3AD900'    +0073          LD     A,(AUTO) : TEST IF AUTO WARM START
'027D  B7         +0074          OR     A
'027E  7A         +0075          LD     A,D
'027F  CA00EA'    +0076          JP     Z,BIOSST-1600H : BRANCH TO AUTO CP/M
'0282  C303EA'    +0077          JP     BIOSST-15FDH : BRANCH TO NO AUTO CP/M
'0285  CD1C0E'    +0078 BOOTE:   CALL   ERRSND
'0288  219602'    +0079          LD     HL,BOOTET : SET FOR RETURN TO DIAGNOSTIC
'028B  110030     +0080          LD     DE,3000H
'028E  010800     +0081          LD     BC,8
'0291  EDB0       +0082          LDIR
'0293  C30030     +0083          JP     3000H   : JUMP TO RESTORE DIAG. ROUTINE
'0296  AF         +0084 BOOTET:  XOR    A       : LOAD PROMS, STOP TIMER
'0297  D3F2       +0085          OUT    (H88CTL),A
'0299  C30308     +0086          JP     803H    : BRANCH TO CDR83D DIAGNOSTIC
                  +0087 :
                  +0088 :        INCREMENT SECTOR NUMBER
                  +0089 :
'029C  214400     +0090 SECINC:  LD     HL,SECTR : GET SECTOR LOCATION
'029F  34         +0091          INC    (HL)    : SET FOR NEW SECTOR NUMBER
'02A0  DD7E12     +0092          LD     A,(IX+18) : GET COUNT OF HOST SECTORS/TRACK
'02A3  BE         +0093          CP     (HL)    : TEST IF NEW TRACK
'02A4  F0         +0094          RET    P       : END IF IN TRACK
'02A5  3601       +0095          LD     (HL),1  : RESET SECTOR
'02A7  214400     +0096          LD     HL,SECTR : GET SECTOR NUMBER LOCATION
```

```
 ADDR  CODE        STMT SOURCE STATEMENT

'02AA  3601      +0097          LD     (HL),1  ; SET AS FIRST SECTOR
'02AC  214500    +0098          LD     HL,TRACK
'02AF  34        +0099          INC    (HL)    ; SET FOR NEW TRACK
'02B0  CD5705'   +0100          CALL   BLKSET
                 +0101 ;
                 +0102 ;        SPECIAL NOTE: REMAINDER OF CP/M RESIDES ON THE
                 +0103 ;               LAST TRACK OF SINGLE DENSITY DISKETTES
                 +0104 ;               THIS IS DO TO THE ADDED SIZE OF BIOS
                 +0105 ;
'02B3  DD460F    +0106          LD     B,(IX+15) ; TEST FOR SINGLE DENSITY
'02B6  CB70      +0107          BIT    6,B
'02B8  C8        +0108          RET    Z       ; RETURN IF NOT SINGLE DENSITY
'02B9  3E02      +0109          LD     A,2     ; SET 8" SINGLE DENSITY SKIP TRACK
'02BB  CB68      +0110          BIT    5,B     ; TEST IF 8" DISK
'02BD  2801      +0111          JR     Z,SECIN1 ; BRANCH IF 8" SINGLE DENSITY
'02BF  3C        +0112          INC    A       ; SET FOR 5.25" SKIP TRACK
'02C0  BE        +0113 SECIN1:  CP     (HL)    ; TEST IF AT THE SKIP TRACK
'02C1  C0        +0114          RET    NZ      ; RETURN IF NOT
'02C2  DD7E14    +0115          LD     A,(IX+20) ; GET MAXIMUM TRACK # FOR SIDE
'02C5  CB10      +0116          RL     B       ; TEST IF DOUBLE SIDED
'02C7  3803      +0117          JR     C,SECIN2 ; BRANCH IF NOT
'02C9  CB27      +0118          SLA    A       ; DOUBLE THE TRACK #
'02CB  3C        +0119          INC    A       ; SET FOR NEXT TRACK
'02CC  77        +0120 SECIN2:  LD     (HL),A  ; SET SKIPPED TRACK NUMBER
'02CD  C9        +0121          RET
                 +0122 ;
                 +0123 ;        HOME THE CURRENT DISK DRIVE
                 +0124 ;
'02CE  DDE5      +0125 HOME:    PUSH   IX
'02D0  3A5F09'   +0126          LD     A,(HSTWRT) ; CHECK FOR WRITE PENDING
'02D3  B7        +0127          OR     A
'02D4  2008      +0128          JR     NZ,HOME1 ; RETURN ABNORMAL IF WRITE PENDING
'02D6  AF        +0129          XOR    A
'02D7  326109'   +0130          LD     (HSTACT),A ; SET CLEARED HOST ACTIVE
'02DA  3C        +0131          INC    A       ; SET FOR TRACK 1
'02DB  32D30E'   +0132          LD     (TRKSTR),A ; CLEAR WANTED TRACK
'02DE  DDE1      +0133 HOME1:   POP    IX
'02E0  C9        +0134          RET
                 +0135 ;
                 +0136 ;        TEST FOR THE DISK TYPE USING THE GETTYP ROUTINE
                 +0137 ;        ALLOWS FOR RETEST IF RESET ERROR OR SOFT ERROR
                 +0138 ;
'02E1  0603      +0139 DSKTYP:  LD     B,3     ; SET FOR THREE TRIES AT GETTING THE DISK TYPE
'02E3  C5        +0140 DSKTY1:  PUSH   BC      ; SAVE TRY COUNT
'02E4  CDF002'   +0141          CALL   GETTYP  ; GET THE DRIVE TYPE
'02E7  C1        +0142          POP    BC
'02E8  C8        +0143          RET    Z       ; END IF TYPE FOUND
'02E9  10F8      +0144          DJNZ   DSKTY1  ; RETRY UNTIL NOT FINDABLE
'02EB  04        +0145          INC    B       ; SET ERROR RETURN
'02EC  CD1C0E'   +0146          CALL   ERRSND  ; SEND ERROR MESSAGE
'02EF  C9        +0147          RET
                 +0148 ;
                 +0149 ;        GET THE TYPE OF DRIVE AND SET DPB
                 +0150 ;
'02F0  AF        +0151 GETTYP:  XOR    A       ; SET SIDE 0
'02F1  324F00    +0152          LD     (SIDE),A
'02F4  CD2606'   +0153          CALL   CDTIME  ; CLEAR HEAD DELAY TIMER
'02F7  3C        +0154          INC    A
```

```
ADDR  CODE       STMT SOURCE STATEMENT

'02F8 324500    +0155          LD     (TRACK),A
'02FB 324400    +0156          LD     (SECTR),A
'02FE 32D80E'   +0157          LD     (NEWDRV),A : SET AS NEW DRIVE
'0301 D33E      +0158          OUT    (DSECT),A       : SET SECTOR NUMBER IN CONTROLLER
'0303 3A4300    +0159          LD     A,(UNIT) : SAVE OLD UNIT NUMBER
'0306 329C03'   +0160          LD     (UNITSV),A : SAVE THE UNIT NUMBER
'0309 3AD70E'   +0161          LD     A,(DISKNO) : GET WANTED DRIVE
'030C 324300    +0162          LD     (UNIT),A : SET UNIT
'030F F5        +0163          PUSH   AF
'0310 CD5705'   +0164          CALL   BLKSET  : GET THE CURRENT DPB TABLE
'0313 F1        +0165          POP    AF
'0314 FE04      +0166          CP     NDISK3  : TEST IF FOR HRS-4 CONTROLLER
'0316 F29D03'   +0167          JP     P,GETH17 : BRANCH IF FOR HRS-4
                +0168
'0319 DD7E0F    +0169          LD     A,(IX+15) : GET THE DISK TYPE
'031C F685      +0170          OR     _SSIDE+_SAMETO+_EXTEND  : SET AS SINGLE SIDED
'031E EE05      +0171          XOR    _SAMETO+_EXTEND : CLEAR EXTENT AND TRACK 0 FLAGS
'0320 4F        +0172          LD     C,A       : PLACE FOR RESTORE DRIVE COMMAND
'0321 C5        +0173          PUSH   BC
'0322 CD4304'   +0174          CALL   RESTOR  : HOME THE DRIVE
'0325 C1        +0175          POP    BC
'0326 280B      +0176          JR     Z,GETTY0 : BRANCH IF SAME DRIVE TYPE
'0328 79        +0177          LD     A,C       : SWITCH DRIVE TYPE
'0329 EE20      +0178          XOR    _5INCH
'032B 4F        +0179          LD     C,A
'032C C5        +0180          PUSH   BC
'032D CD4304'   +0181          CALL   RESTOR  : HOME THE DRIVE AT THE DIFFERENT TYPE
'0330 C1        +0182          POP    BC
'0331 2013      +0183          JR     NZ,GETENR : BRANCH IF ERROR
'0333 CDE703'   +0184 GETTY0:  CALL   TYPTES  : TEST AT THE TYPE
'0336 281A      +0185          JR     Z,GETTY2 : BRANCH IF TYPE FOUND
'0338 79        +0186          LD     A,C       : SWITCH THE DISK DENSITY
'0339 EE40      +0187          XOR    _SDENS
'033B 4F        +0188          LD     C,A
'033C CDE703'   +0189          CALL   TYPTES  : TEST AT THE DIFFERENT DENSITY
'033F 2811      +0190          JR     Z,GETTY2 : BRANCH IF TYPE FOUND
'0341 CDE703'   +0191          CALL   TYPTES  : TEST AGAIN AT THE DIFFERENT DENSITY
'0344 280C      +0192          JR     Z,GETTY2 : BRANCH IF TYPE FOUND
'0346 0EE0      +0193 GETENR:  LD     C,_SSIDE+_SDENS+_5INCH : SET SINGLE SIDED S.D. 5"
'0348 CDF303'   +0194          CALL   GMATCH
'034B 3EFF      +0195          LD     A,_NODISK : SET ERROR CODE
'034D CD5606'   +0196          CALL   SETERR  : SET ERROR NZ RETURN
'0350 1840      +0197          JR     GETEND
                +0198
'0352 CBB9      +0199 GETTY2:  RES    7,C       : SET TEST FOR DOUBLE SIDED
'0354 3E02      +0200          LD     A,2       : SET SIDE 2
'0356 324F00    +0201          LD     (SIDE),A
'0359 CDE703'   +0202          CALL   TYPTES
'035C 2007      +0203          JR     NZ,GETTY3 : BRANCH IF NOT
'035E 3A4800    +0204          LD     A,(SIDEID) : GET SIDE IN READ ID
'0361 FE01      +0205          CP     1         : TEST IF SIDE 1
'0363 2819      +0206          JR     Z,GETTY5 : BRANCH IF DOUBLE SIDED
                +0207
'0365 CBF9      +0208 GETTY3:  SET    7,C       : SET AS SINGLE SIDED
'0367 AF        +0209          XOR    A         : SET SIDE 0
'0368 324F00    +0210          LD     (SIDE),A
'036B CDF303'   +0211          CALL   GMATCH  : RESTORE TO CURRENT TYPE
'036E CB6F      +0212          BIT    5,A       : TEST IF 5.25 INCH
```

```
 ADDR  CODE       STMT SOURCE STATEMENT

'0370  2007      +0213          JR     NZ,GETTY4 : BRANCH IF 5.25 INCH TYPE (NO RESET)
'0372  CDCF03'   +0214          CALL   RESET   : RESET THE CONTROLLER
'0375  3E01      +0215          LD     A,1     : RESET THE ACTUAL TRACK #
'0377  D33D      +0216          OUT    (DTRACK),A
'0379  CD7D04'   +0217 GETTY4:  CALL   GETID1  : RESTORE THE SECTOR ID
'037C  20E7      +0218          JR     NZ,GETTY3 : BRANCH UNTIL REACH ORIGINAL
                 +0219
'037E  3A4A00    +0220 GETTY5:  LD     A,(SECSIZ) : GET SIZE OF RECORD
'0381  CD9C04'   +0221          CALL   BITSHF  : SET AS SECS/HOST SECTOR
'0384  DDBE13    +0222          CP     (IX+19) : TEST WITH STANDARD LENGTH
'0387  2809      +0223          JR     Z,GETEND : END IF STANDARD
'0389  CB19      +0224          RR     C        : GET SECTOR SIZE BIT
'038B  3F        +0225          CCF             : CHANGE SECTOR SIZE
'038C  CB11      +0226          RL     C        : RESTORE SECTOR SIZE BIT
'038E  CDE307'   +0227          CALL   GMATCH  : MATCH TYPE
                 +0228
'0391  AF        +0229          XOR    A        : SET NORMAL RETURN
                 +0230
'0392  3A9C03'   +0231 GETEND:  LD     A,(UNITSV) : GET ORIGINAL UNIT NUMBER
'0395  324300    +0232          LD     (UNIT),A
'0398  CD1B06'   +0233          CALL   HDTIME  : SET HEAD DELAY CONSTANTS
'039B  C9        +0234          RET
                 +0235 :
'039C  00        +0236 UNITSV:  DEFB   0        : UNIT NUMBER TEMPORARY SAVE
                 +0237 :
                 +0238 :        H17 GETTYP ROUTINE
                 +0239 :
>039D            +0240 GETH17   EQU    $
                 +0241
                 +0242          IF     AHSTYP  : FOR HARD SECTORED TYPE
'039D  CD3705'   +0243          CALL   DSLCHK  : CHECK IF H88-4 MUST BE DESELECTED
'03A0  3A4300    +0244          LD     A,(UNIT) : GET THE DRIVE NUMBER
'03A3  CD870A'   +0245          CALL   H17DRV  : TRANSLATE FOR DRIVE TYPE
'03A6  F610      +0246          OR     DFMO    : SET TO TURN DRIVE ON
'03A8  D37F      +0247          OUT    (DPDC),A
'03AA  3E1E      +0248          LD     A,30     : WAIT FOR DRIVE TO COME ON
'03AC  CD090B'   +0249          CALL   DLY
'03AF  DB7F      +0250          IN     A,(DPDC) : GET DRIVE STATUS
'03B1  E601      +0251          AND    1        : SET FOR SECTOR HOLE TEST
'03B3  47        +0252          LD     B,A
'03B4  0E4A      +0253          LD     C,74     : SET (7,4)*40 MIL SEC. COUNT
'03B6  3E04      +0254 GET17B:  LD     A,4      : SET FOR 4 OUT OF 40 COUNT
'03B8  CD090B'   +0255          CALL   DLY      : WAIT FOR 4 MILSECS
'03BB  DB7F      +0256          IN     A,(DPDC) : GET DRIVE STATUS
'03BD  E601      +0257          AND    1        : TEST IF SECTOR HOLE STATUS CHANGED
'03BF  B8        +0258          CP     B        : FROM INITIAL STATUS TO CURRENT
'03C0  200A      +0259          JR     NZ,GET17C : BRANCH IF NEW STATUS (DISK PRESENT)
'03C2  0D        +0260          DEC    C        : TEST IF 40 MILSECS ARE UP
'03C3  20F1      +0261          JR     NZ,GET17B : BRANCH IF NOT
'03C5  79        +0262          LD     A,C      : NO SECTOR HOLE CHANGE (DISK NOT PRESENT)
'03C6  3D        +0263          DEC    A
                 +0264          ENDIF
                 +0265
'03C7  324200    +0266          LD     (ERRST),A : SET ERROR STATUS
'03CA  18C6      +0267          JR     GETEND  : BRANCH FOR ERROR
'03CC  AF        +0268 GET17C:  XOR    A        : SET FOR GOOD RETURN
'03CD  18C3      +0269          JR     GETEND
                 +0270 :
```

```
 ADDR  CODE       STMT SOURCE STATEMENT

                  +0271 ;        RESET THE CONTROLLER
                  +0272 ;
'03CF  F5         +0273 RESET:   PUSH    AF
'03D0  3A3505'    +0274          LD      A,(CURSEL) ; GET CURRENT DRIVE SELECT STATUS
'03D3  F5         +0275          PUSH    AF       ; SAVE SELECT
'03D4  CBA7       +0276          RES     4,A      ; SET AB RESET
'03D6  D33B       +0277          OUT     (DSELEC),A
'03D8  3E10       +0278          LD      A,ACCTIM ; WAIT FOR CONTROLLER AFFECT
'03DA  CDE307'    +0279          CALL    MSECS
'03DD  F1         +0280          POP     AF
'03DE  D33B       +0281          OUT     (DSELEC),A
'03E0  DB3F       +0282          IN      A,(DDATA) ; CLEAR CONTROLLER INTERRUPTS
'03E2  CD2E07'    +0283          CALL    FDCINT  ; INTERRUPT CONTROLLER
'03E5  F1         +0284          POP     AF
'03E6  C9         +0285          RET
                  +0286 ;
                  +0287 ;        TEST THE DESIRED DRIVE AT THE CURRENT TYPE
                  +0288 ;        C = ACTIVE CODE, IX = DPB PTR., RETURNS NZ = INACTIVE
'03E7  C5         +0289 TYPTES:  PUSH    BC
'03E8  CD2E07'    +0290          CALL    FDCINT  ; INTERRUPT CONTROLLER COMMAND
'03EB  CDF303'    +0291          CALL    GMATCH  ; SET DPB TABLE
'03EE  CD6704'    +0292          CALL    GETID   ; TRY TO READ AN ADDRESS
'03F1  C1         +0293          POP     BC
'03F2  C9         +0294          RET
                  +0295 ;
                  +0296 ;        MATCH THE DPB PTR. FOR THE LOGICAL DRIVE'S DPE WITH
                  +0297 ;        THE PHYSICAL ACTIVE DRIVE CODE.  C = ACTIVE CODE
                  +0298 ;
'03F3  C5         +0299 GMATCH:  PUSH    BC
'03F4  CD5705'    +0300          CALL    DLXGET  ; GET THE LOGICAL DRIVE PTR.
'03F7  DD7E0F     +0301          LD      A,(IX+15) ; GET THE DPB CODE
'03FA  F604       +0302          OR      _SAMETO ; DROP SAME TRACK 0 TEST BIT
'03FC  EE04       +0303          XOR     _SAMETO
'03FE  B9         +0304          CP      C        ; TEST WITH ACTIVE CODE
'03FF  281B       +0305          JR      Z,GMATC3 ; BRANCH IF ALREADY MATCHED
'0401  0616       +0306          LD      B,22     ; SET COUNT OF TRIES
'0403  CD1E04'    +0307 GMATC1:  CALL    CHANGE  ; SET FOR NEW DPB
'0406  DD7E0F     +0308          LD      A,(IX+15) ; GET THE DPB CODE
'0409  F604       +0309          OR      _SAMETO ; DROP SAME TRACK 0 TEST BIT
'040B  EE04       +0310          XOR     _SAMETO
'040D  B9         +0311          CP      C        ; TEST WITH THE ACTIVE CODE
'040E  280A       +0312          JR      Z,GMATC2 ; BRANCH IF MATCH
'0410  05         +0313          DEC     B        ; TEST IF ALL FORMATS TRIED
'0411  20F0       +0314          JR      NZ,GMATC1 ; BRANCH IF NOT
'0413  DD21E201'  +0315          LD      IX,DPB7 ; SET ERROR DEFAULT DPB
'0417  CD1E04'    +0316          CALL    CHANGE
'041A  AF         +0317 GMATC2:  XOR     A        ; SET NON ZERO RETURN
'041B  3C         +0318          INC     A
'041C  C1         +0319 GMATC3:  POP     BC
'041D  C9         +0320          RET
                  +0321 ;
                  +0322 ;        CHANGE THE CURRENT DPB POINTER
                  +0323 ;        RETURNS WITH DPE POINTING TO NEW DPB, IX = NEW DPB
                  +0324 ;
'041E  C5         +0325 CHANGE:  PUSH    BC
'041F  E5         +0326          PUSH    HL
'0420  CDCC04'    +0327          CALL    SELDSN  ; GET THE DPE PTR.
'0423  E5         +0328          PUSH    HL       ; SAVE DPE
```

```
 ADDR  CODE       STMT SOURCE STATEMENT

'0424  CD3504'   +0329            CALL    NEWDPB  : GET THE NEW DPB LOC
'0427  DDE1      +0330            POP     IX        : SET DPE FOR NEXT DPB
'0429  DD740B    +0331            LD      (IX+0BH),H
'042C  DD750A    +0332            LD      (IX+0AH),L
'042F  E5        +0333            PUSH    HL        : PLACE NEW DPB IN IX
'0430  DDE1      +0334            POP     IX
'0432  E1        +0335            POP     HL
'0433  C1        +0336            POP     BC
'0434  C9        +0337            RET
                 +0338 ;
                 +0339 ;          SET A NEW DPB POINTER
                 +0340 :          ENTERED WITH IX=CURRENT DPB POINTER
                 +0341 ;          RETURNS WITH HL=NEXT DPB POINTER
                 +0342 :
'0435  DDE5      +0343 NEWDPB: PUSH    IX        : GET THE DPB PTR
'0437  C1        +0344            POP     BC
'0438  211500    +0345            LD      HL,21   : SET FOR NEXT DPB
'043B  09        +0346            ADD     HL,BC
'043C  7E        +0347            LD      A,(HL)  : TEST IF AT END OF TABLES
'043D  B7        +0348            OR      A
'043E  C0        +0349            RET     NZ        : BRANCH IF NOT
'043F  21DA00'   +0350            LD      HL,DPBTBL : SET FOR START OF DPBS
'0442  C9        +0351            RET
                 +0352 ;
                 +0353 ;          RESTORE CURRENT DRIVE
                 +0354 :          ENTER C = DISK TYPE, RETURNS Z = GOOD RESTOR
                 +0355 ;
'0443  CDE303'   +0356 RESTOR: CALL    GMATCH  : SET THE DRIVE TYPE
'0446  3E10      +0357            LD      A,_NRESET : SET FOR RESET OF DRIVE
'0448  323405'   +0358            LD      (CURENT),A : CHANGE CURRENT DRIVE FOR RETEST
'044B  C5        +0359            PUSH    BC
'044C  CDEE04'   +0360            CALL    SETDRV  : CHECK AND SET FOR THE CURRENT DRIVE
'044F  C1        +0361            POP     BC
'0450  3E80      +0362            LD      A,_NOTRDY : SET DRIVE ERROR MASK
'0452  32ED01'   +0363            LD      (ERRMSK),A
'0455  3E08      +0364            LD      A,_RESTOR : GET THE RESTORE COMMAND
'0457  CD2F06'   +0365            CALL    SCMND   : SEND COMMAND
'045A  C0        +0366            RET     NZ
'045B  AF        +0367            XOR     A         : SET ACTUAL TRACK 0
'045C  D33D      +0368            OUT     (DTRACK),A
'045E  3C        +0369            INC     A         : SET TO GO TO TRACK 1
'045F  D33F      +0370            OUT     (DDATA),A
'0461  3E1B      +0371            LD      A,_SEEK+_HEADLD+_30MIL : SET FOR READ TRACK 1
'0463  CD2F06'   +0372            CALL    SCMND   : NON VERIFY TRACK 1
'0466  C9        +0373            RET
                 +0374 ;
                 +0375 :          GET CURRENT TRACK ID. FROM DRIVE  RET NZ=ERROR
                 +0376 ;
'0467  3E90      +0377 GETID:  LD      A,_NOTRDY+_RNF
'0469  32ED01'   +0378            LD      (ERRMSK),A
'046C  AF        +0379            XOR     A         : CLEAR ERROR STATUS LOCATION
'046D  324200    +0380            LD      (ERRST),A
'0470  3C        +0381            INC     A
'0471  D33D      +0382            OUT     (DTRACK),A : SET CURRENT TRACK
'0473  D33F      +0383            OUT     (DDATA),A
'0475  3E1C      +0384            LD      A,_SEEK+_HEADLD+_VERIFY : VERIFY THE TRACK
'0477  C5        +0385            PUSH    BC
'0478  CD2F06'   +0386            CALL    SCMND   : VERIFY FOR TRACK 1
```

```
 ADDR  CODE        STMT SOURCE STATEMENT

'047B  C1        +0387          POP     BC
'047C  C0        +0388          RET     NZ       ; END IF TRACK NOT VERIFIED
'047D  0605      +0389 GETID1:  LD      B,5      ; SET # OF READ ID PASSES
'047F  C5        +0390 GETID2:  PUSH    BC
'0480  21CC00'   +0391          LD      HL,DLYCON : GET THE NORMAL TIMEOUT
'0483  46        +0392          LD      B,(HL)
'0484  CB18      +0393          RR      B        : CUT WAIT TIME IN HALF
'0486  70        +0394          LD      (HL),B : SET FOR A QUICK TIMEOUT
'0487  CB10      +0395          RL      B        : RESET WAIT TIME
'0489  C5        +0396          PUSH    BC       : SAVE WAIT TIME
'048A  CD7906'   +0397          CALL    READID  : GET THE ID
'048D  C1        +0398          POP     BC
'048E  21CC00'   +0399          LD      HL,DLYCON ; RESTOR THE NORMAL TIMEOUT
'0491  70        +0400          LD      (HL),B
'0492  C1        +0401          POP     BC       : RESTORE # OF READ ID PASSES
'0493  C8        +0402          RET     Z        : END IF GOOD RETURN
'0494  DB3E      +0403          IN      A,(DSECT) : GET THE TRACK NUMBER
'0496  D33D      +0404          OUT     (DTRACK),A : SET AS TRACK NUMBER
'0498  10E5      +0405          DJNZ    GETID2
'049A  05        +0406          DEC     B
'049B  C9        +0407          RET
                 +0408 :
                 +0409 :        SHIFT THE BITS IN THE  A  REGISTER
                 +0410 :        A = 0,1,2, OR 3 .   RETURNS A = 1,2,4, OR 8
                 +0411 :
'049C  C5        +0412 BITSHF:  PUSH    BC
'049D  4F        +0413          LD      C,A      : SET SHIFT COUNT
'049E  B7        +0414          OR      A        : TEST IF NO SHIFT
'049F  3E01      +0415          LD      A,1      : SET BIT TO SHIFT
'04A1  2805      +0416          JR      Z,BITSH2 : BRANCH FOR NO SHIFT
'04A3  CB07      +0417 BITSH1:  RLC     A        : SHIFT THE BIT
'04A5  0D        +0418          DEC     C        : SET NEW SHIFT COUNT
'04A6  20FB      +0419          JR      NZ,BITSH1
'04A8  C1        +0420 BITSH2:  POP     BC
'04A9  C9        +0421          RET
                 +0422 :
                 +0423 :        SELECT A DISK DRIVE TO BE THE CURRENT ONE
                 +0424 :
'04AA  79        +0425 SELDSK:  LD      A,C      : SET UNIT NUMBER
'04AB  32D70E'   +0426          LD      (DISKNO),A : SET THE CP/M DISK NUMBER
'04AE  3A5F09'   +0427          LD      A,(HSTWRT) : TEST IF WRITE PENDING
'04B1  B7        +0428          OR      A
'04B2  2018      +0429          JR      NZ,SELDSN ; BRANCH FOR PENDING
'04B4  CB43      +0430          BIT     0,E      : HAS DRIVE BEEN ACCESSED BEFORE
'04B6  2014      +0431          JR      NZ,SELDSN : BRANCH IF NOT
'04B8  ED734D00  +0432          LD      (SAVESP),SP : SAVE CURRENT STACK LOCATION
'04BC  31FF15'   +0433          LD      SP,BSTACK
'04BF  DDE5      +0434          PUSH    IX
'04C1  CDE102'   +0435          CALL    DSKTYP  ; GET THE DRIVE TYPE
'04C4  DDE1      +0436          POP     IX
'04C6  ED7B4D00  +0437          LD      SP,(SAVESP) : RESTORE THE STACK
'04CA  2012      +0438          JR      NZ,SELDSE : BRANCH IF DISK TYPE NOT FOUND
                 +0439 :
                 +0440 :        SELECT THE CURRENT DISK DRIVE'S DPE TABLE PTR
                 +0441 :        RETURNS HL=DPE LOC, NC=ERROR
                 +0442 :
'04CC  210000    +0443 SELDSN:  LD      HL,0
'04CF  3AD70E'   +0444          LD      A,(DISKNO)
```

```
'04D2  FE04     +0445           CP     NDISKS
'04D4  3808     +0446           JR     C,SELDS1 ; BRANCH IF FDC-880H DRIVE
'04D6  D604     +0447           SUB    NDISKS  : DROP FDC-880H SECTION
'04D8  FE03     +0448           CP     H17DKS  ; TEST IF WITHIN H17 RANGE
'04DA  3AD70E'  +0449           LD     A,(DISKNO) ; RESTORE DPE COUNT
'04DD  D0       +0450           RET    NC       : END IF NOT
'04DE  6F       +0451 SELDS1:  LD     L,A      ; SAVE PHYSICAL DRIVE #
'04DF  29       +0452           ADD    HL,HL
'04E0  29       +0453           ADD    HL,HL
'04E1  29       +0454           ADD    HL,HL
'04E2  29       +0455           ADD    HL,HL
'04E3  113300'  +0456           LD     DE,DPBASE
'04E6  19       +0457           ADD    HL,DE
'04E7  C9       +0458           RET
                +0459 :
'04E8  AF       +0460 SELDSE:  XOR    A        : SET NO ERROR RETURN
'04E9  320400   +0461           LD     (DEFALT),A : RESET DEFAULT TO (A:)
'04EC  67       +0462           LD     H,A
'04ED  6F       +0463           LD     L,A
'04EE  C9       +0464           RET
                +0465 :
                +0466 :         SET AND SELECT THE CURRENT DRIVE
                +0467 :         RETURNS IX=DPB PTR., NZ=NEW DRIVE SELECTED
                +0468 :
'04EF  CD5705'  +0469 SETDRV:  CALL   BLKGET  ; SET POINTER FOR DPB IN IX REG.
'04F2  CD3705'  +0470           CALL   DSLCHK  ; CHECK IF H88-4 MUST BE DESELECTED
'04F5  213405'  +0471           LD     HL,CURENT : GET THE CURRENT LOGICAL DRIVE
'04F8  3A4300   +0472           LD     A,(UNIT) : GET THE WANTED DRIVE NUMBER
'04FB  BE       +0473           CP     (HL)     : TEST IF THE SAME DRIVE
'04FC  C8       +0474           RET    Z        : END IF THE SAME DRIVE
'04FD  77       +0475           LD     (HL),A   ; SAVE AS NEW CURRENT DRIVE
'04FE  47       +0476           LD     B,A      : SET UNIT NUMBER
'04FF  3AA300'  +0477           LD     A,(OFFSET) ; GET THE OFFSET
'0502  80       +0478           ADD    A,B      : SET PHYSICAL UNIT
'0503  FE04     +0479           CP     NDISKS  ; TEST IF WRAP AROUND
'0505  3802     +0480           JR     C,SETDR2 : BRANCH IF NOT
'0507  D604     +0481           SUB    NDISKS  : WRAP AROUND TO NEXT DRIVE
'0509  4F       +0482 SETDR2:  LD     C,A      : SAVE WANTED PHYSICAL DRIVE
'050A  B7       +0483           OR     A        : TEST IF DRIVE 0
'050B  3E01     +0484           LD     A,1      : SET DRIVE 0 CODE =1
'050D  2804     +0485           JR     Z,SETDR4 : BRANCH IF DRIVE 0
'050F  07       +0486 SETDR3:  RLCA            : ROTATE IN THE DRIVE CODE
'0510  0D       +0487           DEC    C        : TEST IF DRVE NUMBER FOUND
'0511  20FC     +0488           JR     NZ,SETDR3 : BRANCH IF NOT
'0513  4F       +0489 SETDR4:  LD     C,A      ; SAVE CREATED DRIVE CODE
'0514  3A7505'  +0490           LD     A,(CURSEL) : GET CURRENT DRIVE SELECT STATUS
'0517  E670     +0491           AND    _DROPDV : CLEAR THE DRIVE NUMBER
'0519  B1       +0492           OR     C        : MERGE IN THE NEW DRIVE CODE
'051A  D33B     +0493           OUT    (DSELEC),A : SET NEW SELECT IN CONTROLLER
'051C  327505'  +0494           LD     (CURSEL),A : SET NEW DRIVE SELECT STATUS
'051F  DD7E0F   +0495           LD     A,(IX+15) ; GET THE CURRENT DISK TYPE
'0522  E640     +0496           AND    _SSIDE+_5INCH : SET FOR TEST FOR RESET
'0524  217605'  +0497           LD     HL,OLDTYP : GET THE OLD DRIVE TYPE
'0527  BE       +0498           CP     (HL)     : TEST IF MATCH
'0528  2804     +0499           JR     Z,SETDR5 : BRANCH IF THE SAME
'052A  77       +0500           LD     (HL),A   : SET RESET OLD DRIVE TYPE
'052B  CDCF03'  +0501           CALL   RESET    ; RESET CONTROLLER IF NOT
'052E  AF       +0502 SETDR5:  XOR    A        : SET FOR NON ZERO RETURN
```

```
 ADDR  CODE       STMT SOURCE STATEMENT

'052F  3C         +0503            INC     A
'0530  32D80E'    +0504            LD      (NEWDRV),A : SET NEW DRIVE CODE
'0533  C9         +0505            RET
                  +0506 ;
'0534  FF         +0507 CURENT:  DEFB    _NODISK : CURRENT DRIVE NUMBER
'0535  10         +0508 CURSEL:  DEFB    _NRESET : CURRENT DRIVE SELECT STATUS
'0536  10         +0509 OLDTYP:  DEFB    _NRESET : OLD DRIVE TYPE (BIT5=0=8,=1=5)
                  +0510 ;
                  +0511 ;          DESELECT THE CONTROLLER BOARD CHECK ROUTINE
                  +0512 ;          ENTERED: (UNIT)=THE CURRENT UNIT SELECTED, (OLDDSK)=OLD
                  +0513 ;                  UNIT SELECTED.
                  +0514 ;          IF THE CONTROLLER BOARD IS CHANGED IN GOING TO THE NEW UNIT,
                  +0515 ;          THEN THE APPROPRIATE DESELECT IS EXECUTED FOR THE OTHER BOARD
                  +0516 ;          RETURNS: THE  A,HL  REGISTERS AFFECTED.
                  +0517 ;
'0537  3A4300     +0518 DSLCHK:  LD      A,(UNIT) : GET THE CURRENT UNIT NUMBER
'053A  FE04       +0519            CP      NDISKS  : TEST FOR WHICH CONTROLLER BOARD
'053C  215605'    +0520            LD      HL,OLDDSK : GET OLD DISK TYPE
'053F  7E         +0521            LD      A,(HL)
'0540  FA4B05'    +0522            JP      M,DSLCH1 : BRANCH IF FDC-380H CONTROLLER BOARD
'0543  FE04       +0523            CP      NDISKS  : TEST IF BOARD CHANGED FROM FDC-380H
'0545  F0         +0524            RET     P         : END IF NOT
'0546  CDB609'    +0525            CALL    DSLFDC  : ELSE DESELECT THE FDC-380H
'0549  1806       +0526            JR      DSLCH2  : AND SET NEW CONTROLLER CODE
'054B  FE04       +0527 DSLCH1:  CP      NDISKS  : TEST IF BOARD CHANGED FROM HBB-4
'054D  F8         +0528            RET     M         : END IF NOT
'054E  CDCA09'    +0529            CALL    DSLH17  : ELSE DESELECT THE HBB-4
'0551  3A4300     +0530 DSLCH2:  LD      A,(UNIT) : GET THE NEW TYPE
'0554  77         +0531            LD      (HL),A  : SET THE NEW TYPE
'0555  C9         +0532            RET
                  +0533 ;
'0556  00         +0534 OLDDSK:  DEFB    0         : CONTROLLER TYPE STORAGE
                  +0535 ;
                  +0536 ;          SET POINTER TO HOST DRIVE'S DPB
                  +0537 ;          RETURNS IX=DPB TABLE POINTER
                  +0538 ;
'0557  3A4500     +0539 BLKSET:  LD      A,(TRACK) : GET HOST TRACK NUMBER
'055A  32AA05'    +0540            LD      (TESTRK),A : SET TRACK LOCATION TO TEST FOR
'055D  3A4300     +0541            LD      A,(UNIT) : GET HOST DRIVE NUMBER
'0560  C5         +0542 BLKST1:  PUSH    BC        : SAVE REGISTERS
'0561  D5         +0543            PUSH    DE
'0562  E5         +0544            PUSH    HL
'0563  4F         +0545            LD      C,A       : GET UNIT NUMBER FOR DPH
'0564  210000     +0546            LD      HL,0
'0567  CDDE04'    +0547            CALL    SELDS1  : SET HL WITH DPH POINTER
'056A  010A00     +0548            LD      BC,0AH  : INDEX INTO DPH FOR DPB PTR.
'056D  09         +0549            ADD     HL,BC
'056E  4E         +0550            LD      C,(HL)  : GET DPB PTR LSB
'056F  23         +0551            INC     HL
'0570  46         +0552            LD      B,(HL)  : GET DPB PTR MSB
'0571  C5         +0553            PUSH    BC        : SET DPB PTR. IN IX
'0572  DDE1       +0554            POP     IX
'0574  3AAA05'    +0555            LD      A,(TESTRK) : GET THE CURRENT TRACK #
'0577  B7         +0556            OR      A         : TEST IF TRACK 0
'0578  201B       +0557            JR      NZ,BLKSE2 : BRANCH IF NOT
'057A  DD4E0F     +0558            LD      C,(IX+15) : GET THE DTYPE CODE
'057D  CB51       +0559            BIT     2,C       : TEST IF NO CHANGE IN TYPE ON TRACK 0
'057F  2014       +0560            JR      NZ,BLKSE2 : BRANCH IF NO CHANGE
```

```
ADDR  CODE         STMT SOURCE STATEMENT

'0581  79          +0561          LD     A,C       : SET DISK TYPE FOR CHANGES
'0582  E6FC        +0562          AND    0FCH      : DROP EXTENDED, AND TRACK 0 EXT.
'0584  F6C0        +0563          OR     _SSIDE+_SDENS : SET AS SINGLE SIDED, AND S. D.
'0586  4F          +0564          LD     C,A       : SET NEW DISK TYPE
'0587  C5          +0565 BLKSE1:  PUSH   BC        : SAVE NEW DISK TYPE CODE
'0588  CD3504'     +0566          CALL   NEWDPB    : CHANGE CURRENT DISK TYPE
'058B  C1          +0567          POP    BC        : RESTORE WANTED DISK TYPE
'058C  E5          +0568          PUSH   HL        : SET NEW DPB POINTER
'058D  DDE1        +0569          POP    IX
'058F  DD7E0F      +0570          LD     A,(IX+15) : GET DISK TYPE FOR COMPARE
'0592  B9          +0571          CP     C         : TEST IF FOUND DISK TYPE
'0593  20F2        +0572          JR     NZ,BLKSE1 : BRANCH UNTIL FOUND
'0595  DD7E0F      +0573 BLKSE2:  LD     A,(IX+15) : GET THE NEW DRIVE TYPE
'0598  E670        +0574          AND    _DROPDV : SET FOR NEW CURENT CODE
'059A  47          +0575          LD     B,A       : SAVE NEW TYPE
'059B  217505'     +0576          LD     HL,CURSEL : GET THE CURRENT SELECT
'059E  7E          +0577          LD     A,(HL)
'059F  E60F        +0578          AND    _DROPTP : DROP THE OLD TYPE
'05A1  B0          +0579          OR     B         : MERGE IN THE NEW TYPE
'05A2  F610        +0580          OR     _NRESET : SET FOR NON RESET MODE
'05A4  77          +0581          LD     (HL),A : SAVE RESULT
'05A5  E1          +0582          POP    HL        : RETURN NORMAL
'05A6  D1          +0583          POP    DE
'05A7  C1          +0584          POP    BC
'05A8  AF          +0585          XOR    A
'05A9  C9          +0586          RET
                   +0587 :
'05AA  00          +0588 TESTRK:  DEFB   0         : CURRENT TRACK NUMBER
                   +0589 :
                   +0590 :        SET POINTER TO WANTED DRIVE'S DPB
                   +0591 :        RETURNS IX=DPB TABLE POINTER
'05AB  3AD70E'     +0592 NEWSET:  LD     A,(TRKSTR) : GET THE WANTED TRACK NUMBER
'05AE  32AA05'     +0593          LD     (TESTRK),A
'05B1  3AD70E'     +0594          LD     A,(DISKNO) : GET THE WANTED DRIVE NUMBER
'05B4  18A4        +0595          JR     BLKST1  : CONTINUE WITH [illegible] SET
                   +0596 :
                   +0597 :        SET THE TRACK NUMBER TO SEEK TO
                   +0598 :
'05B6  60          +0599 SETTRK:  LD     H,B       : GET TRACK NUMBER
'05B7  69          +0600          LD     L,C
'05B8  22D70E'     +0601          LD     (TRKSTR),HL : SAVE AS CP/M TRACK #
'05BB  C9          +0602          RET
                   +0603 :
                   +0604 :        SET SECTOR NUMBER TO ACCESS
                   +0605 :
'05BC  79          +0606 SETSEC:  LD     A,C       : GET SECTOR NUMBER
'05BD  32D50E'     +0607          LD     (SECTOR),A : SAVE AS CP/M SECTOR #
'05C0  C9          +0608          RET
                   +0609 :
                   +0610 :        TRANSLATE THE SECTOR GIVEN BY BC SETTING FOR SECTOR #1
                   +0611 :        GIVEN BY DE
                   +0612 :
'05C1              +0613 SECTRN   EQU    $
                   +0614          IF     ADREMOD : IF TRANSLATION TABLE CAPABILITY
                   +0615          EX     DE,HL     : SET TRANSLATION TABLE
                   +0616          LD     A,H       : GET SECTOR TABLE MSB
                   +0617          ADD    HL,BC     : ADD SECTOR AS TABLE OFFSET
                   +0618          OR     A         : TEST IF DDR TRANSLATION TYPE
```

```
 ADDR  CODE       STMT SOURCE STATEMENT

                  +0619           LD     L,(HL)  : GET CODE FROM TRANSALTION TABLE
                  +0620           ENDIF
'05C1  2600       +0621           LD     H,0
                  +0622           IF     ADPEMOD : IF TRANSLATION TABLE CAPABILITY
                  +0623           RET    NZ      : END IF NOT CDR TYPE
                  +0624           LD     (TRANCD),A : SET 0 FOR SECTOR TRANSLATION
                  +0625           ENDIF
'05C3  0C         +0626           INC    C       : SET SECTOR FOR SECTOR TRANSLATION
'05C4  69         +0627           LD     L,C     : SET OFFSET SECTOR NUMBER
'05C5  C9         +0628           RET
                  +0629 :
                  +0630 :         SET READ/WRITE MEMORY ADDRESS
                  +0631 :
'05C6  60         +0632 SETDMA:   LD     H,B     : TRANSFER MEMORY ADDRESS
'05C7  69         +0633           LD     L,C
'05C8  224000     +0634           LD     (TADDR),HL : STORE MEMORY ADDRESS
'05CB  C9         +0635           RET
                  +0636 :
                  +0637 :         WRITE TO A SECTOR ON THE CURRENT DRIVE
                  +0638 :         (UNIT)=HOST DISK #, (TRACK)=HOST TRACK #,
                  +0639 :         (SECTR)=HOST SECTOR #,  WRITES FROM (HSTLOC) BUFFER
                  +0640 :
'05CC  21DE06'    +0641 WTHOST:   LD     HL,WTSEC : SET WRITE IN READ/WRITE ROUTINE
'05CF  1803       +0642           JR     RWHOST  : BRANCH TO WRITE A SECTOR
                  +0643 :
                  +0644 :         READ A SECTOR FROM THE CURRENT DRIVE
                  +0645 :         (UNIT)=HOST DISK #, (TRACK)=HOST TRACK #,
                  +0646 :         (SECTR)=HOST SECTOR #,  READS INTO (HSTLOC) BUFFER
                  +0647 :
'05D1  21B806'    +0648 RDHOST:   LD     HL,RDSEC : SET READ A SECTOR ROUTINE
                  +0649 :
                  +0650 :         HOST READ/WRITE ROUTINE
                  +0651 :
'05D4  22EA05'    +0652 RWHOST:   LD     (RWHOS2+1),HL : SET COMMAND TYPE (READ OR WRITE)
'05D7  21D90E'    +0653           LD     HL,HSTBUF : SET HOST READ/WRITE BUFFER
'05DA  22D90E'    +0654           LD     (HSTLOC),HL
'05DD  ED4BCA00'  +0655 RWHOS0:   LD     BC,(RWTRYS) : TRY READ/WRITE (NN) TIMES
'05E1  CD2606'    +0656           CALL   CDTIME  : CLEAR HEAD DELAY TIMES
'05E4  C5         +0657 RWHOS1:   PUSH   BC
'05E5  AF         +0658           XOR    A       : SET SIDE 0 FOR INITIAL ACCESS
'05E6  324F00     +0659           LD     (SIDE),A
'05E9  CDB806'    +0660 RWHOS2:   CALL   RDSEC   : CALL TO READ/WRITE A HOST SECTOR
'05EC  C1         +0661           POP    BC
'05ED  2825       +0662           JR     Z,RWXIT : RETURN IF GOOD READ
'05EF  21D700'    +0663           LD     HL,SOFTC : INCREMENT SOFT ERROR COUNT
'05F2  34         +0664           INC    (HL)
'05F3  0D         +0665           DEC    C       : SET FOR TRACK TEST
'05F4  2012       +0666           JR     NZ,RWHOS3 : BRANCH IF NOT LSB TRYS
'05F6  C5         +0667           PUSH   BC
'05F7  CDCF03'    +0668           CALL   RESET   : RESTORE THE DRIVE
'05FA  3E08       +0669           LD     A,_RESTOR : GET THE RESTORE COMMAND
'05FC  CD2F06'    +0670           CALL   SCMND   : SEND COMMAND
'05FF  AF         +0671           XOR    A       : SET ACTUAL TRACK 0
'0600  D33D       +0672           OUT    (DTRACK),A
'0602  C1         +0673           POP    BC
'0603  21CA00'    +0674           LD     HL,RWTRYS : RESET TEST COUNT
'0606  4E         +0675           LD     C,(HL)
'0607  04         +0676           INC    B       : READ TEST WON'T END ON RESTOR
```

```
 ADDR  CODE        STMT SOURCE STATEMENT

'0608  10DA       +0677 RWHOS3: DJNZ    RWHOS1  : BRANCH FOR MSB TRIES
                  +0678 :
                  +0679 :       READ/WRITE ERROR EXIT
                  +0680 :
'060A  CD1C0E'    +0681         CALL    ERRSND  : SEND ERROR MESSAGE
'060D  04         +0682         INC     B       : SET NON ZERO RETURN
'060E  210800'    +0683         LD      HL,HARDC : INCREMENT HARD ERROR COUNT
'0611  34         +0684         INC     (HL)
'0612  1803       +0685         JR      RWXITE  : BRANCH FOR ERROR
                  +0686 :
                  +0687 :       READ/WRITE EXIT
                  +0688 :
'0614  22D90E'    +0689 RWXIT:  LD      (HSTLOC),HL : SET NEXT HOST LOCATION
'0617  CD1B06'    +0690 RWXITE: CALL    HDTIME  : SET HEAD DELAY TIMER
'061A  C9         +0691         RET
                  +0692 :
                  +0693 :       SET HEAD DELAY TIMES FOR KEEPING HEAD LOADED
                  +0694 :
'061B  F3         +0695 HDTIME: DI
'061C  E5         +0696         PUSH    HL
'061D  210F06     +0697         LD      HL,DELAYS : SET DELAY TIMER
'0620  226209'    +0698         LD      (DLYMO),HL
'0623  E1         +0699         POP     HL
'0624  FB         +0700         EI
'0625  C9         +0701         RET
                  +0702 :
                  +0703 :       CLEAR HEAD DELAY TIMER
                  +0704 :
'0626  E5         +0705 CDTIME: PUSH    HL
'0627  210000     +0706         LD      HL,0
'062A  226209'    +0707         LD      (DLYMO),HL
'062D  E1         +0708         POP     HL
'062E  C9         +0709         RET
                  +0710 :
                  +0711 :       SEND A NON INTERRUPT COMMAND
                  +0712 :       A = COMMAND TO SEND
                  +0713 :
'062F  32ED01'    +0714 SCMND:  LD      (CMDSV),A : SAVE COMMAND
'0632  3A3505'    +0715         LD      A,(CURSEL) : GET CURRENT DRIVE SELECT STATUS
'0635  D33B       +0716         OUT     (DSELED),A : SEND NEW CONTROLS
'0637  CDD307'    +0717         CALL    BUSYTS  : WAIT TILL CONTROLLER NOT BUSY
'063A  C0         +0718         RET     NZ      : END IF BUSY ERROR TIMEOUT
'063B  3AED01'    +0719         LD      A,(CMDSV) : RESTORE COMMAND
'063E  DDB611     +0720         OR      (IX+17) : MERGE IN STEP TIME FOR DISK TYPE
'0641  D330       +0721         OUT     (DCMD),A : SEND COMMAND
                  +0722 :
                  +0723 :       CONTINUE WITH END OF COMMAND TEST
                  +0724 :
                  +0725 :       COMMAND END CONTROL OF GOOD OR BAD ACK
                  +0726 :
'0643  CDD307'    +0727 CEND:   CALL    BUSYTS  : WAIT FOR CONTROLLER NOT BUSY
'0646  C0         +0728         RET     NZ      : END IF BUSY TIMEOUT ERROR
'0647  D5         +0729         PUSH    DE
'0648  57         +0730         LD      D,A     : SAVE STATUS
'0649  3AED01'    +0731         LD      A,(ERRMSK) : GET MASK
'064C  A2         +0732         AND     D       : MASK OUT UNIMPORTANT BITS
'064D  D1         +0733         POP     DE
'064E  324200     +0734         LD      (ERRST),A : SET IN STATUS
```

```
ADDR  CODE      STMT SOURCE STATEMENT

'0651  C8       +0735          RET    Z       : NORMAL RETURN IF GOOD
'0652  CD5606'  +0736          CALL   SETERR  ; SET ERROR CODE
'0655  C9       +0737          RET
                +0738 :
                +0739 :        SET ERROR CODE AND INTERRUPT CONTROLLER COMMAND
                +0740 :        ENTER WITH A = ERROR CODE. EXIT (ERRST) SET. NON ZERO
                +0741 :
'0656  324200   +0742 SETERR:  LD     (ERRST),A         : SET ERROR MESSAGE
'0659  CD2E07'  +0743          CALL   FDCINT            : INTERRUPT CONTROLLER
'065C  B7       +0744          OR     A                 : SET NON ZERO RETURN
'065D  C9       +0745          RET
                +0746 :
                +0747 :        READ DATA AND DISSABLE CONTROLLER INTERRUPTS
                +0748 :
'065E  EDB2     +0749          INIR              : READ THE DATA UP TO 1024 BYTES
'0660  EDB2     +0750          INIR
'0662  EDB2     +0751          INIR              : READ THE DATA UP TO 512 BYTES
'0664  EDB2     +0752 DISINR:  INIR              : READ THE DATA UP TO 256 BYTES
'0666  5F       +0753          LD     E,A     : GET FIRST CHARACTER
'0667  1808     +0754          JR     DISINT  : BRANCH TO RETURN TO INTERRUPTS
                +0755 :
'0669  EDB3     +0756          OTIR              : WRITE THE DATA UP TO 1024 BYTES
'066B  EDB3     +0757          OTIR
'066D  EDB3     +0758          OTIR              : WRITE THE DATA UP TO 512 BYTES
'066F  EDB3     +0759 DISINW:  OTIR              : WRITE THE DATA UP TO 256 BYTES
                +0760 :
'0671  3A3505'  +0761 DISINT:  LD     A,(CURSEL) : GET CURRENT DRIVE SELECT STATUS
'0674  D33B     +0762          OUT    (DSELED),A : SET DRIVE SELECT PORT WITH RESULT
'0676  FB       +0763          EI                : ENABLE NORMAL INTERRUPTS
'0677  ED4D     +0764          RETI              : RETURN FROM R/W INTERRUPT
                +0765 :
                +0766 :        READ PRESENT DISK ADDRESS
                +0767 :
'0679  CD0107'  +0768 READID:  CALL   SETRD   : INITIALIZE INTERRUPT FOR READING
'067C  214700   +0769          LD     HL,IDSV : GET LOCATION TO PLACE ID
'067F  23       +0770          INC    HL      ; SKIP TRACK BYTE
'0680  0605     +0771          LD     B,5     : GET DATA COUNT FOR READING
'0682  3E9C     +0772          LD     A,_NOTRDY+_RNF+_CRC+_LOSTDA ; SET ERROR MASK
'0684  32ED01'  +0773          LD     (ERRMSK),A
'0687  3A4F00   +0774          LD     A,(SIDE) : GET THE SIDE
'068A  F6C4     +0775          OR     _READAD+_CDELAY : MERGE WITH READ ADDR. COMMAND
'068C  CD9706'  +0776          CALL   SENDRW  : EXECUTE COMMAND
'068F  C0       +0777          RET    NZ      : END IF ERROR
'0690  7B       +0778          LD     A,E     : GET TRACK #
'0691  324700   +0779          LD     (IDSV),A : SET TRACK #
'0694  D33D     +0780          OUT    (DTRACK),A : SEND IT TO THE CONTROLLER
'0696  C9       +0781          RET
                +0782 :
                +0783 :        SEND THE READ OR WRITE COMMAND INTERRUPT ROUTINE
                +0784 :        A=COMMAND. B=COUNT OF DATA. IY=DPB TABLE.
                +0785 :        E=FIRST BYTE IF WRITE COMMAND. HL=TRANSFER LOC.
                +0786 :
'0697  32ED01'  +0787 SENDRW:  LD     (CMDSV),A : SAVE COMMAND
'069A  3A3505'  +0788          LD     A,(CURSEL) : GET CURRENT DRIVE SELECT STATUS
'069D  F680     +0789          OR     _INTEN  : ENABLE CONTROLLER INTERRUPTS
'069F  D33B     +0790          OUT    (DSELED),A : SEND NEW CONTROLLER STATUS
'06A1  CDD307'  +0791          CALL   BUSYTS  : WAIT FOR CONTROLLER NOT BUSY
'06A4  C0       +0792          RET    NZ      : END IF ERROR TIME-OUT
```

```
'06A5  0E3F     +0793         LD     C,DDATA ; GET IN/OUT PORT LOC
'06A7  CD080C'  +0794         CALL   DISABL  ; DISABLE OTHER INTERRUPTS
'06AA  3AED01'  +0795         LD     A,(CMDSV) ; RESTORE COMMAND
'06AD  D330     +0796         OUT    (DCMD),A ; SEND COMMAND
'06AF  7B       +0797         LD     A,E      ; SET FIRST BYTE FOR FAST WRITE
'06B0  76       +0798         HALT            ; WAIT FOR INTERRUPT
'06B1  CD000D'  +0799         CALL   ENABLE  ; REENABLE OTHER INTERRUPTS
'06B4  CD4306'  +0800         CALL   CEND    ; TEST READ/WRITE RETURN STATUS
'06B7  C9       +0801         RET             ; RETURN WHEN DONE
                +0802 ;
                +0803 ;       READ A SECTOR FROM THE CURRENT DRIVE
                +0804 ;       RETURNS NZ=ERROR
                +0805 ;
'06B8  3A4700   +0806 RDSEC:  LD     A,(UNIT) ; GET UNIT FOR H17 TEST
                +0807
                +0808         IF     AHSTYP  ; FOR HARD SECTORED TYPE
'06BB  FE04     +0809         CP     NDISKS
'06BD  F2D509'  +0810         JP     P,RD17  ; BRANCH FOR H17 TYPE
                +0811         ENDIF
                +0812
'06C0  CDEF04'  +0813         CALL   SETDRV  ; SET FOR CURRENT DRIVE ACCESS
'06C3  C47D04'  +0814         CALL   NZ,GETID1 ; IF FIRST DISK ACCESS THEN READ ID
'06C6  C0       +0815         RET    NZ      ; END IF BAD READ ID
'06C7  CD0107'  +0816         CALL   SETRD   ; SET R/W INTERRUPT FOR READING
'06CA  CD0D07'  +0817         CALL   SETSL   ; SET SECTOR SIZE AND TRANSFER LOC.
'06CD  15       +0818         DEC    D       ; SET FOR ONE LESS BYTE TO READ
'06CE  23       +0819         INC    HL      ; SET NEXT LOCATION
'06CF  0E88     +0820         LD     C,_READSC ; SET READ COMMAND AND ERROR MASK
'06D1  06BE     +0821         LD     B,_NOTRDY+_RECTYP+_RNF+_CRC+_LOSTDA+_DRQ
'06D3  CD3A07'  +0822         CALL   DRVSND  ; EXECUTE READ COMMAND
'06D6  C0       +0823         RET    NZ      ; END IF BAD READ
'06D7  E5       +0824         PUSH   HL      ; SAVE END OF READ LOCATION
'06D8  2AD90E'  +0825         LD     HL,(HSTLOC) ; GET LOC. FOR FIRST BYTE
'06DB  73       +0826         LD     (HL),E  ; PLACE FIRST BYTE IN SECTOR
'06DC  E1       +0827         POP    HL      ; SAVE END OF READ LOCATION
'06DD  C9       +0828         RET
                +0829 ;
                +0830 ;       WRITE A SECTOR ONTO THE CURRENT DRIVE
                +0831 ;       RETURNS NZ=ERROR
                +0832 ;
'06DE  3A4700   +0833 WTSEC:  LD     A,(UNIT) ; GET UNIT FOR H17 TEST
                +0834
                +0835         IF     AHSTYP  ; FOR HARD SECTORED TYPE
'06E1  FE04     +0836         CP     NDISKS
'06E3  F2190A'  +0837         JP     P,WR17  ; BRANCH FOR H17 TYPE
                +0838         ENDIF
                +0839
'06E6  CDEF04'  +0840         CALL   SETDRV  ; SET FOR CURRENT DRIVE ACCESS
'06E9  C47D04'  +0841         CALL   NZ,GETID1 ; IF FIRST DISK ACCESS THEN READ ID
'06EC  C0       +0842         RET    NZ      ; END IF BAD ID
'06ED  210508'  +0843         LD     HL,WRITTB ; SET WRITE INTERRUPT TABLE
'06F0  CD0407'  +0844         CALL   SETRW   ; PLACE IN INTERRUPT LOC.
'06F3  CD0D07'  +0845         CALL   SETSL   ; SET SECTOR SIZE AND TRANSFER LOC.
'06F6  15       +0846         DEC    D       ; SET FOR ONE LESS BYTE TO WRITE
'06F7  5E       +0847         LD     E,(HL)  ; GET FIRST BYTE FOR INITIAL WRITE
'06F8  23       +0848         INC    HL      ; SET NEXT LOCATION
'06F9  0EA8     +0849         LD     C,_WRITSC ; SET WRITE COMMAND AND ERROR MASK
'06FB  06FE     +0850         LD     B,_NOTRDY+_WRPROT+_WRFALT+_RNF+_CRC+_LOSTDA+_DRQ
```

```
ADDR  CODE      STMT SOURCE STATEMENT

'06FD CD3A07'  +0851            CALL   DRVSND  : EXECUTE WRITE COMMAND
'0700 C9       +0852            RET
               +0853 :
               +0854 :          SET THE CONTROLLER INTERRUPT FOR READING
               +0855 :
'0701 210008'  +0856 SETRD:     LD     HL,READTB : GET THE READ INT. TABLE PTR.
               +0857 :
               +0858 :          SET INTERRUPT TABLE (R/W)
               +0859 :
'0704 112B00   +0860 SETRW:     LD     DE,INTLOC : GET THE READ/WRITE INT. LOC.
'0707 010500   +0861            LD     BC,5    : GET SIZE OF WRITE INT. TABLE
'070A EDB0     +0862            LDIR           : SET R/W INT. FOR WRITING
'070C C9       +0863            RET
               +0864 :
               +0865 :          GET THE SIZE OF A SECTOR AND IT'S TRANSFER LOCATION
               +0866 :          RETURNS A=(0=256 OR GREATER, 1=128), HL=TRANSFER LOC.
               +0867 :
'070D DD7E13   +0868 SETSL:     LD     A,(IX+19) : GET COUNT OF 128 BYTE SECS/HOSTSEC
'0710 CB0F     +0869            RRC    A       : SET FOR 128 OR 256 BYTE SECTORS
'0712 3815     +0870            JR     C,SETS2 : BRANCH IF 128 BYTES PER SECTOR
'0714 CB3F     +0871            SRL    A       : TEST IF 512 BYTES
'0716 3811     +0872            JR     C,SETS2 : BRANCH IF NOT
'0718 2A2B00   +0873            LD     HL,(INTRET) : GET INTERRUPT RETURN LOCATION
'071B 2B       +0874            DEC    HL      : SET FOR 512 BYTE INTERRUPT INTERRUPT
'071C 2B       +0875            DEC    HL
'071D CB3F     +0876            SRL    A       : TEST IF 1024 BYTES
'071F 3805     +0877            JR     C,SETS1 : BRANCH IF NOT
'0721 2B       +0878            DEC    HL      : SET FOR 1024 BPS
'0722 2B       +0879            DEC    HL
'0723 2B       +0880            DEC    HL
'0724 2B       +0881            DEC    HL
'0725 AF       +0882            XOR    A       : CLEAR 1024 CODE
'0726 222B00   +0883 SETS1:     LD     (INTRET),HL : RESTORE INTERRUPT RETURN
'0729 57       +0884 SETS2:     LD     D,A     : SET RESULT AS BYTE COUNT
'072A 2A2B0E'  +0885            LD     HL,(HSTLOC) : GET TRANSFER LOC
'072D C9       +0886            RET
               +0887 :
               +0888 :          CONTROLLER INTERRUPT
               +0889 :
'072E F5       +0890 FDCINT:    PUSH   AF
'072F 3ED0     +0891            LD     A,INTERR : FORCE INTERRUPT OF DRIVE
'0731 D330     +0892            OUT    (DCMD),A
'0733 3E20     +0893            LD     A,ADDTIM*2
'0735 CD930B'  +0894            CALL   UDLY    : DELAY FOR INTERRUPT ACK
'0738 F1       +0895            POP    AF
'0739 C9       +0896            RET
               +0897 :
               +0898 :          SEND TO THE DRIVE THE READ/WRITE COMMAND
               +0899 :          ENTERED WITH B=ERROR MASK, C=COMMAND, D=BYTECOUNT,
               +0900 :          E=FIRST DATA BYTE IF WRITE COMMAND, HL=TRANSFER ADDR.
               +0901 :
'073A 3A4400   +0902 DRVSND:    LD     A,(SECTR) : GET SECTOR #
'073D CD5507'  +0903            CALL   TRANSL  : TRANSLATE PHYSICAL SECTOR IF NEEDED
'0740 D33E     +0904            OUT    (DSECT),A : SET CONTROLLER WITH SECTOR
'0742 CD8507'  +0905            CALL   SEEK    : SEEK AND SELECT TRACK AND SECTOR
'0745 C0       +0906            RET    NZ      : END IF TRACK NOT FOUND
'0746 3A4F00   +0907            LD     A,(SIDE) : GET SIDE OF DISK
'0749 B1       +0908            OR     C       : MERGE WITH COMMAND
```

```
 ADDR  CODE       STMT SOURCE STATEMENT

'074A  4F        +0909             LD      C,A      ; RESTORE COMMAND
'074B  78        +0910             LD      A,B      ; GET RDWT ERROR MASK
'074C  32ED01'   +0911             LD      (ERRMSK),A ; SET FOR ERROR RETURN
'074F  42        +0912             LD      B,D      ; GET THE BYTE COUNT
'0750  79        +0913             LD      A,C      ; SET COMMAND FOR RETURN
'0751  CD9706'   +0914             CALL    SENDRW   ; EXECUTE COMMAND INT. ROUTINE
'0754  C9        +0915             RET              ; RETURN NORMAL IF GOOD
                 +0916 ;
                 +0917 ;           TRANSLATION ROUTINE FOR HOST SECTORS
                 +0918 ;           USES SECTOR TRANSLATION SKIP COUNT FROM DPB TABLE
                 +0919 ;           A = HOST SECTOR #, RETURNS A = TRANSLATED NUMBER
                 +0920 ;
'0755  C5        +0921 TRANSL:     PUSH    BC
'0756  E5        +0922             PUSH    HL
'0757  4F        +0923             LD      C,A      ; SAVE SECTOR #
                 +0924             IF      ADPEMOD ; IF TRANSLATION TABLE CAPABILLITY
                 +0925             LD      A,(TRANCD) ; TEST IF TRANSLATION NEEDED
                 +0926             OR      A
                 +0927             LD      A,C      ; SET OLD SECTOR NUMBER
                 +0928             JR      NZ,TRANS2 ; BRANCH IF NO TRANSLATION
                 +0929             ENDIF
'0758  DD7E0F    +0930             LD      A,(IX+15) ; GET THE DISK TYPE
'075B  CB57      +0931             BIT     2,A      ; TEST IF SYSTEM TRACKS SKEWED
'075D  280A      +0932             JR      Z,TRANS0 ; BRANCH IF SYSTEM TRACKS SKEWED
'075F  3A4500    +0933             LD      A,(TRACK) ; GET THE TRACK NUMBER
'0762  DDBE0D    +0934             CP      (IX+13) ; TEST WITH DIRECTORY OFFSET
'0765  79        +0935             LD      A,C      ; SET OLD SECTOR NUMBER
'0766  FA8207'   +0936             JP      M,TRANS2 ; BRANCH IF IN SYSTEM TRACKS
'0769  3E01      +0937 TRANS0:     LD      A,1      ; SET NEW SECTOR NUMBER
'076B  0D        +0938 TRANS1:     DEC     C        ; TEST IF NUMBER FOUND
'076C  2814      +0939             JR      Z,TRANS2
'076E  DD8610    +0940             ADD     A,(IX+16) ; ADD SECTOR SKIP COUNT
'0771  DDBE12    +0941             CP      (IX+18) ; TEST IF EXCEEDED MAX
'0774  38F5      +0942             JR      C,TRANS1 ; BRANCH IF NOT
'0776  28F3      +0943             JR      Z,TRANS1
'0778  DD9612    +0944             SUB     (IX+18) ; SET WITHIN SECTOR LIMITS
'077B  FE01      +0945             CP      1        ; TEST IF SHIFT NEEDED
'077D  20EC      +0946             JR      NZ,TRANS1 ; BRANCH IF NOT
'077F  3C        +0947             INC     A
'0780  18E9      +0948             JR      TRANS1
'0782  E1        +0949 TRANS2:     POP     HL
'0783  C1        +0950             POP     BC
'0784  C9        +0951             RET
                 +0952 ;
                 +0953 ;
                 +0954             IF      ADPEMOD ; IF TRANSLATION TABLE CAPABILLITY
                 +0955 TRANCD:     DEFB    0        ; SECTOR TRANSLATION TYPE STORAGE
                 +0956             ENDIF
                 +0957
                 +0958 ;
                 +0959 ;           SEEK FOR A TRACK AND SET SECTOR ON CURRENT DRIVE
                 +0960 ;
'0785  C5        +0961 SEEK:       PUSH    BC       ; SAVE REGISTERS
'0786  D5        +0962             PUSH    DE
'0787  E5        +0963             PUSH    HL
                 +0964 ;           SET THE SIDE FOR THE CURRENT DRIVE (WITHIN SEEK)
                 +0965 ;           IX = DPB TABLE PTR., RETURNS (SIDE) SET.
                 +0966 ;           C = MODIFIED TRACK #, USES B
 ADDR  CODE       STMT SOURCE STATEMENT

'0788  0600      +0967          LD     B,0      ; SET SIDE # FOR SIDE 0
'078A  3A4500    +0968          LD     A,(TRACK) ; GET THE WANTED TRACK #
'078D  DD4E0F    +0969          LD     C,(IX+15) ; GET DISK TYPE
'0790  CB79      +0970          BIT    7,C      ; TEST IF DOUBLE SIDED
'0792  2006      +0971          JR     NZ,SEEK1 ; BRANCH IF NOT
'0794  CB3F      +0972          SRL    A        ; SET DOUBLE SIDED TRACK NUMBER
'0796  3002      +0973          JR     NC,SEEK1 ; BRANCH IF SIDE 0
'0798  0602      +0974          LD     B,2      ; SET FOR SIDE 1
'079A  4F        +0975 SEEK1:   LD     C,A      ; SAVE TRACK #
'079B  78        +0976          LD     A,B      ; SET SIDE BIT FOR DISK SIDE TYPE
'079C  324F00    +0977          LD     (SIDE),A ; SET FOR R/W COMMANDS SIDE SET
'079F  3ADB0E'   +0978          LD     A,(NEWDRV) ; TEST FOR NEW DRIVE
'07A2  B7        +0979          OR     A
'07A3  2005      +0980          JR     NZ,SEEK2 ; BRANCH IF NEW DRIVE
'07A5  DB3D      +0981          IN     A,(DTRACK) ; GET CURRENT TRACK #
'07A7  B9        +0982          CP     C        ; TEST IF WANTED TRACK SAME AS CURRENT
'07A8  2821      +0983          JR     Z,SEEK4 ; BRANCH IF THE SAME
'07AA  3E90      +0984 SEEK2:   LD     A,_NOTRDY+_SEKERR ; SET SEEK ERROR CODE
'07AC  32ED01'   +0985          LD     (ERRMSK),A ; SET MASK
'07AF  79        +0986 SEEK3:   LD     A,C      ; SET TRACK #
'07B0  D33F      +0987          OUT    (DDATA),A ; SEND FOR TRACK #
'07B2  C5        +0988          PUSH   BC       ; SAVE COUNT AND TRACK #
'07B3  3E1C      +0989          LD     A,_SEEK+_HEADLD+_VERIFY ; SET SEEK COMMAND
'07B5  CD2F06'   +0990          CALL   SCMND    ; SEND COMMAND
'07B8  C1        +0991          POP    BC       ; RESTORE THE COUNT AND TRACK #
'07B9  2810      +0992          JR     Z,SEEK4 ; BRANCH IF VALID SEEK
'07BB  F5        +0993          PUSH   AF       ; SAVE ERROR
'07BC  C5        +0994          PUSH   BC       ; SAVE COUNT AND TRACK #
'07BD  3E08      +0995          LD     A,_RESTOR ; RESTORE THE DRIVE
'07BF  CD2F06'   +0996          CALL   SCMND
'07C2  C1        +0997          POP    BC       ; RESTORE COUNT AND TRACK
'07C3  F1        +0998          POP    AF       ; RESTORE ERROR
'07C4  3E10      +0999          LD     A,_SEKERR ; RETURN WITH ERROR FOR SEEK
'07C6  CD5606'   +1000          CALL   SETERR   ; SET ERROR CODE (SEEK ERROR)
'07C9  1804      +1001          JR     SEEK5    ; BRANCH FOR ERROR
'07CB  AF        +1002 SEEK4:   XOR    A        ; SET AS GOOD RETURN
'07CC  32DB0E'   +1003          LD     (NEWDRV),A ; CLEAR NEW DRIVE CODE
'07CF  E1        +1004 SEEK5:   POP    HL       ; RESTORE REGISTERS
'07D0  D1        +1005          POP    DE
'07D1  C1        +1006          POP    BC
'07D2  C9        +1007          RET
                 +1008 ;
                 +1009 ;        TEST IF CONTROLLER BUSY (USED IN NON INTERRUPTS)
                 +1010 ;        RETURNS Z = NOT BUSY, NZ = NOT READY OR TIME-OUT ERROR
                 +1011 ;                A = CONTROLLER STATUS
                 +1012 ;
                 +1013
'07D3  3E10      +1014 BUSYTS:  LD     A,ACCTIM ; WAIT FOR COMMAND ACCEPTANCE
'07D5  CD930B'   +1015          CALL   UDLY
'07D8  C5        +1016          PUSH   BC
'07D9  06BF      +1017          LD     B,BUSRTY ; SET BUSY TIMEOUT RETRY COUNT
'07DB  DB3C      +1018 BUSYT1:  IN     A,(DSTAT) ; GET THE CONTROLLER STATUS
'07DD  CB7F      +1019          BIT    7,A      ; TEST IF DRIVE READY
'07DF  2008      +1020          JR     NZ,BUSYT2 ; BRANCH IF DRIVE NOT READY
'07E1  CB47      +1021          BIT    0,A      ; TEST IF CONTROLLER BUSY
'07E3  280C      +1022          JR     Z,BUSYT3 ; BRANCH IF NORMAL RETURN
'07E5  3E70      +1023          LD     A,BUSYW ; SET WAIT FOR NEXT TEST
'07E7  CDF307'   +1024          CALL   MSECS
```

```
 ADDR  CODE      STMT SOURCE STATEMENT

'07EA  10EF      +1025          DJNZ    BUSYT1  : BRANCH FOR RETRY UNTIL TIMEOUT
'07EC  3EFF      +1026 BUSYT2:  LD      A,_NODISK : SET ERROR INACTIVE CODE
'07EE  CD5606'   +1027          CALL    SETERR
'07F1  C1        +1028 BUSYT3:  POP     BC
'07F2  C9        +1029          RET
                 +1030 ;
                 +1031 ;        WAIT A NUMBER OF MILLISECONDS
                 +1032 ;
'07F3  C5        +1033 MSECS:   PUSH    BC
'07F4  47        +1034          LD      B,A      : SAVE THE ENTERED COUNT
'07F5  3A0000'   +1035 MSEC0:   LD      A,(DLYCON) : GET THE DELAY CONSTANT
'07F8  CD9308'   +1036          CALL    UDLY     : WAIT WITH DELAY CONSTANT
'07FB  05        +1037          DEC     B
'07FC  20F7      +1038          JR      NZ,MSEC0 : WAIT WITH THE MILSEC COUNT
'07FE  C1        +1039          POP     BC
'07FF  C9        +1040          RET
                 +1041 :
                 +1042 :        READ AND WRITE INTERRUPT TABLES. PLACED AT
                 +1043 :        CONTROLLER INTERRUPT LOC.
                 +1044 :        NOTE: THE WRITE ROUTINE MUST SEND THE FIRST BYTE
                 +1045 :              AT A FASTER THAN NORMAL RATE. THIS BYTE IS
                 +1046 :              SET IN THE E REGISTER PRIOR TO INTERRUPT.
                 +1047 :
'0800  DB3F      +1048 READTB:  IN      A,(DDATA) : READ INTERRUPT TABLE
'0802  C36406'   +1049          JP      DISINR
                 +1050 :
'0805  D33F      +1051 WRITTB:  OUT     (DDATA),A : WRITE INTERRUPT TABLE
'0807  C36F06'   +1052          JP      DISINW  : BASED ON E REGISTER
                  0950 :
                  0951 :        INCLUDE THE CP/M 2.2 BLOCKING ROUTINES
                  0952 :
                  0953          INCLUDE CDRBC.SRC
                 +0001 :
                 +0002 :        INTERNAL BLOCKING-DEBLOCKING ROUTINES FOR THE
                 +0003 :        CDR SYSTEMS CONTROLLER BIOS
                 +0004 :
'080A  ED734D00 +0005 READ:     LD      (SAVESP),SP : SET FOR BIOS STACK
'080E  31FF15'   +0006          LD      SP,BSTACK : GET BIOS STACK PTR.
'0811  DDE5      +0007          PUSH    IX
'0813  CD4B05'   +0008          CALL    NEWSET  : SET NEW DPB TABLE
'0816  AF        -0009          XOR     A        : CLEAR UNALLOCATED
'0817  325909'   +0010          LD      (UNACNT),A
'081A  3E01      +0011          LD      A,1      : SET READ OPERATION CODE
'081C  325609'   +0012          LD      (READOP),A
'081F  325709'   +0013          LD      (RSFLAG),A : SET MUST READ CODE
'0822  3E02      +0014          LD      A,WRUAL : SET A UNALLOCATED
'0824  325809'   +0015          LD      (WRTYPE),A
'0827  187A      +0016          JR      RWOPER  : BRANCH FOR READ WRITE OP.
                 +0017 :
                 +0018 :        WRITE ENTRY POINT
                 +0019 :
'0829  ED734D00 +0020 WRITE:    LD      (SAVESP),SP : SET FOR BIOS STACK
'082D  31FF15'   +0021          LD      SP,BSTACK : GET BIOS STACK PTR.
'0830  DDE5      +0022          PUSH    IX
'0832  CD4B05'   +0023          CALL    NEWSET  : SET NEW DPB TABLE
'0835  AF        +0024          XOR     A        : SET READ OPERATION CODE FALSE
'0836  325609'   +0025          LD      (READOP),A
'0839  79        +0026          LD      A,C      ; GET WRITE TYPE FROM BDOS
```

```
 ADDR  CODE      STMT SOURCE STATEMENT

'083A  325809'  +0027          LD     (WRTYPE),A
'083D  FE02     +0028          CP     WRUAL   : TEST IF WRITE UNALLOCATED
'083F  201F     +0029          JR     NZ,CHKUNA : CHECK FOR UNALLOCATED
                +0030 :
                +0031 :        WRITE TO UNALLOCATED, SET PARAMETERS
'0841  DD4602   +0032          LD     B,(IX+2) : GET BLOCK SIZE CODE
'0844  3E01     +0033          LD     A,1      : SET COUNT FOR SECTORS/BLOCK
'0846  CB27     +0034 WRITE1:  SLA    A        : MULTIPLY SECTOR COUNT BY 2
'0848  05       +0035          DEC    B        : TEST IF SECTOR COUNT SET
'0849  20FB     +0036          JR     NZ,WRITE1 : BRANCH UNTIL TRUE SECTOR COUNT
'084B  325909'  +0037          LD     (UNACNT),A : SET UNALLOCATED SECTORS
'084E  3A070E'  +0038          LD     A,(DISKNO) : SET DISK TO SEEK
'0851  325A09'  +0039          LD     (UNADSK),A
'0854  2A030E'  +0040          LD     HL,(TRKSTR) : SET TRACK TO SEEK
'0857  225B09'  +0041          LD     (UNATRK),HL
'085A  3A050E'  +0042          LD     A,(SECTOR) : SET SECTOR TO SEEK
'085D  325D09'  +0043          LD     (UNASEC),A
                +0044 :
                +0045 :        CHECK FOR WRITE TO UNALLOCATED SECTOR
'0860  3A5909'  +0046 CHKUNA:  LD     A,(UNACNT) : TEST FOR UNALLOCATED SECTORS
'0863  B7       +0047          OR     A
'0864  2835     +0048          JR     Z,ALLOC : BRANCH IF NOT
                +0049 :
                +0050 :        MORE UNALLOCATED SECTORS REMAIN
'0866  3D       +0051          DEC    A        : DROP AN UNALLOCATED SECTOR
'0867  325909'  +0052          LD     (UNACNT),A
'086A  3A070E'  +0053          LD     A,(DISKNO) : TEST IF SAME DISK
'086D  215A09'  +0054          LD     HL,UNADSK
'0870  BE       +0055          CP     (HL)
'0871  2028     +0056          JR     NZ,ALLOC : BRANCH IF NOT
                +0057 :
                +0058 :        THE DISKS ARE THE SAME
'0873  215B09'  +0059          LD     HL,UNATRK : GET TRACK NUMBER
'0876  CD4A09'  +0060          CALL   STRKCP  : COMPARE SEEK TRACK WITH UNALLOCATED
'0879  2020     +0061          JR     NZ,ALLOC : BRANCH IF NOT
                +0062 :
                +0063 :        THE TRACKS ARE THE SAME
'087B  3A050E'  +0064          LD     A,(SECTOR)       : TEST IF SAME SECTOR
'087E  215D09'  +0065          LD     HL,UNASEC
'0881  BE       +0066          CP     (HL)
'0882  2017     +0067          JR     NZ,ALLOC         : BRANCH IF NOT
                +0068 :
                +0069 :        THE SECTORS ARE THE SAME. SET NEXT SECTOR
'0884  34       +0070          INC    (HL)    : SET FOR NEXT SECTOR
'0885  DD4600   +0071          LD     B,(IX)  : GET MAX SECTORS/TRACK
'0888  7E       +0072          LD     A,(HL)  : GET SECTOR #
'0889  B8       +0073          CP     B       : TEST WITH MAX SECTORS
'088A  3809     +0074          JR     C,NOOVF : SKIP IF NO OVERFLOW
                +0075 :
                +0076 :        OVERFLOW TO NEXT TRACK
'088C  3600     +0077          LD     (HL),0  : CLEAR UNALLOCATED SECTOR
'088E  2A5B09'  +0078          LD     HL,(UNATRK) : SET NEW UNALLOCATED TRACK
'0891  23       +0079          INC    HL
'0892  225B09'  +0080          LD     (UNATRK),HL
                +0081 :
                +0082 :        MATCH FOUND. SET AS UNNECESSARY READ
'0895  AF       +0083 NOOVF:   XOR    A       : CLEAR READ DATA FLAG
'0896  325709'  +0084          LD     (RSFLAG),A
```

```
 ADDR  CODE       STMT SOURCE STATEMENT

'0899  1808     +0085               JR      RWOPER  : BRANCH TO PERFORM WRITE
                +0086 ;
                +0087 ;             NOT AN UNALLOCATED RECORD, USE PRE-READ
'089B  AF       +0088 ALLOC:        XOR     A         : CLEAR UNALLOCATED RECORD COUNT
'089C  325909'  +0089               LD      (UNACNT),A
'089F  3C       +0090               INC     A         : SET READ DATA FLAG
'08A0  325709'  +0091               LD      (RSFLAG),A
                +0092 ;
                +0093 ;             COMMON BLOCKING CODE FOR READ AND WRITE OPERATIONS
                +0094 ;
'08A3  AF       +0095 RWOPER:       XOR     A         : CLEAR ERROR FLAG
'08A4  324200   +0096               LD      (ERRST),A
'08A7  4F       +0097               LD      C,A       : CLEAR HOST SECTOR CODE
'08A8  DD7E13   +0098               LD      A,(IX+19) : GET NUMBER OF SECTORS/HOST SCT.
'08AB  47       +0099               LD      B,A       : STORE IT
'08AC  3AD50E'  +0100               LD      A,(SECTOR) : GET THE DESIRED SECTOR
'08AF  F5       +0101               PUSH    AF        : SAVE THE DESIRED SECTOR
'08B0  0C       +0102 RWOP0:        INC     C         : SECTOR PHYSICAL SECTOR COUNT
'08B1  90       +0103               SUB     B         : IS WANTED SECTOR IN HOST SEC.
'08B2  2807     +0104               JR      Z,RWOP1 : BRANCH IF IT IS
'08B4  F2B008'  +0105               JP      P,RWOP0 : BRANCH IF NOT
'08B7  79       +0106 RWOP1:        LD      A,C       : GET THE PHYSICAL SECTOR COUNT
'08B8  325E09'  +0107               LD      (SECHST),A
'08BB  F1       +0108               POP     AF        : RESTORE THE DESIRED SECTOR
'08BC  05       +0109               DEC     B         : SET SECTOR #
'08BD  3D       +0110               DEC     A
'08BE  A0       +0111               AND     B         : FOR COUNT WITHIN THE SECTOR
'08BF  326009'  +0112               LD      (HOFST),A : SET WITHIN SECTOR COUNT
                +0113 ;
                +0114 ;             TEST IF ACTIVE HOST SECTOR
'08C2  216109'  +0115               LD      HL,HSTACT : GET HOST ACTIVE FLAG
'08C5  7E       +0116               LD      A,(HL)
'08C6  3601     +0117               LD      (HL),1  : SET FOR ACTIVE
'08C8  B7       +0118               OR      A         : TEST IF ALREADY ACTIVE
'08C9  2801     +0119               JR      Z,FILHST : BRANCH IF NOT ACTIVE
                +0120 ;
                +0121 ;             HOST BUFFER ACTIVE, TEST IF SAME AS SEEK
'08CB  3AD70E'  +0122               LD      A,(DISKNO) : GET WANTED DRIVE
'08CE  214300   +0123               LD      HL,UNIT : GET HOST DISK
'08D1  96       +0124               SUB     (HL)      : TEST IF SAME
'08D2  2011     +0125               JR      NZ,NOMACH : BRANCH IF NO MATCH
                +0126 ;
                +0127 ;             SAME DISK, TEST IF SAME TRACK
'08D4  214500   +0128               LD      HL-TRACK : GET HOST TRACK LOC.
'08D7  CD4A09'  +0129               CALL    STRKCP  : TEST IF SAME TRACKS
'08DA  2009     +0130               JR      NZ,NOMACH : BRANCH IF NO MATCH
                +0131 ;
                +0132 ;             SAME TRACK, TEST IF SAME SECTOR
'08DC  3A5E09'  +0133               LD      A,(SECHST) : GET WANTED SECTOR
'08DF  214400   +0134               LD      HL,SECTR : GET HOST SECTOR PTR.
'08E2  BE       +0135               CP      (HL)      : TEST IF SAME
'08E3  2824     +0136               JR      Z,MATCH : BRANCH IF SAME
                +0137 ;
                +0138 ;             SECTOR IS NOT THE SAME AS WANTED, BUT SAME DRIVE
'08E5  3A5F09'  +0139 NOMACH:       LD      A,(HSTWRT) : TEST IF HOST WRITTEN
'08E8  B7       +0140               OR      A
'08E9  C4CC05'  +0141               CALL    NZ,WTHOST : CLEAR HOST BUFFER
                +0142 ;
 ADDR  CODE       STMT SOURCE STATEMENT

                  +0143 ;        MAY HAVE TO FILL THE HOST BUFFER
'08EC  3AD70E'    +0144 FILHST:  LD      A,(DISKNO) ; GET WANTED DRIVE #
'08EF  324300     +0145          LD      (UNIT),A ; SET AS HOST DRIVE
'08F2  2AD30E'    +0146          LD      HL,(TRKSTR) ; SET WANTED TRACK #
'08F5  224500     +0147          LD      (TRACK),HL
'08F8  3A5E09'    +0148          LD      A,(SECHST) ; SET WANTED SECTOR
'08FB  324400     +0149          LD      (SECTR),A
'08FE  3A5709'    +0150          LD      A,(RSFLAG) ; TEST IF NEED TO READ
'0901  B7         +0151          OR      A
'0902  C4D105'    +0152          CALL    NZ,RDHOST ; YES IF NOT 0
'0905  AF         +0153          XOR     A       ; CLEAR ACCUMULATOR
'0906  325F09'    +0154          LD      (HSTWRT),A ; CLEAR HOST WRITE PENDING
                  +0155 ;
                  +0156 ;        COPY DATA TO OR FROM BUFFER
'0909  3A6009'    +0157 MATCH:   LD      A,(HOFST) ; GET HOST OFFSET INDEX COUNT
'090C  018000     +0158          LD      BC,128  ; SET INDEX SIZE
'090F  21DB0E'    +0159          LD      HL,HSTBUF ; GET HOST BUFFER LOCATION
'0912  3D         +0160 MATCH1:  DEC     A       ; TEST IF ACTUAL TRANSFER LOC. FOUND
'0913  FA1909'    +0161          JP      M,MATCH2 ; BRANCH IF FOUND
'0916  09         +0162          ADD     HL,BC   ; ADD SECTOR INDEX
'0917  18F9       +0163          JR      MATCH1  ; TEST UNTIL LOC. FOUND
'0919  EB         +0164 MATCH2:  EX      DE,HL   ; SET HOST LOC. IN DE
'091A  2A4000     +0165          LD      HL,(TADDR) ; SET TRANSFER ADDRESS
'091D  EB         -0166          EX      DE,HL   ; SET TRANSFER FOR READING
'091E  3A5609'    +0167          LD      A,(READOP) ; TEST IF TO READ OR WRITE
'0921  B7         +0168          OR      A
'0922  2006       +0169          JR      NZ,RWMOVE ; BRANCH IF TO TADDR
                  +0170 ;
                  +0171 ;        READ FROM HOST OPERATION, SET PARAMETERS
'0924  3E01       +0172          LD      A,1     ; SET HOST WRITE
'0926  325F09'    +0173          LD      (HSTWRT),A
'0929  EB         +0174          EX      DE,HL   ; SET FOR HOST TO MEMORY
                  +0175 ;
                  +0176 ;        MOVE BC BYTES TO PTR. DE FROM PTR. HL
'092A  EDB0       +0177 RWMOVE:  LDIR
'092C  EB         +0178          EX      DE,HL
                  +0179 ;
                  +0180 ;        DATA HAS BEEN MOVED
'092D  3A5809'    +0181          LD      A,(WRTYPE) ; TEST IF WRITE TO DIR.
'0930  FE01       +0182          CP      WRDIR
'0932  3A4200     +0183          LD      A,(ERRST) ; GET ERROR STATUS
'0935  200C       +0184          JR      NZ,RWEND ; END IF NO DIRECTORY WRITE
                  +0185 ;
                  +0186 ;        CLEAR HOST BUFFER FOR DIRECTORY WRITE
'0937  B7         +0187          OR      A       ; TEST IF ERRORS OCCURED
'0938  2009       +0188          JR      NZ,RWEND ; END IF ERRORS
'093A  325F09'    +0189          LD      (HSTWRT),A ; SET BUFFER WRITTEN FLAG
'093D  CDCC05'    +0190          CALL    WTHOST  ; WRITE FROM BUFFER
'0940  3A4200     +0191          LD      A,(ERRST) ; GET ERROR FLAG
                  +0192 ;
                  +0193 ;        READ/WRITE END
                  +0194 ;
'0943  DDE1       +0195 RWEND:   POP     IX
'0945  ED7B4D00   +0196          LD      SP,(SAVESP) ; RESTORE BDOS STACK
'0949  C9         +0197          RET
                  +0198 ;
                  +0199 ;
                  +0200 ;        SUBROUTINE FOR 16 BIT COMPARE OF TRKSTR WITH HL
```

```
ADDR  CODE       STMT SOURCE STATEMENT

                 +0201 ;
'094A  EB        +0202 STRKCP: EX      DE,HL
'094B  21D30E'   +0203         LD      HL,TRKSTR ; GET WANTED TRACK
'094E  1A        +0204         LD      A,(DE)   ; GET LSB
'094F  BE        +0205         CP      (HL)     ; TEST IF LSB MATCH
'0950  C0        +0206         RET     NZ       ; END IF NOT
'0951  13        +0207         INC     DE       ; SET FOR MSB
'0952  23        +0208         INC     HL
'0953  1A        +0209         LD      A,(DE)   ; GET MSB
'0954  BE        +0210         CP      (HL)     ; TEST IF MSB MATCH
'0955  C9        +0211         RET
                 +0212 ;
                 +0213 ;       BLOCKING VARIABLES
                 +0214 ;
>0002            +0215 WRUAL   EQU     2          ; WRITE TO UNALLOCATED
>0001            +0216 WRDIR   EQU     1          ; WRITE TO DIRECTORY
>0000            +0217 WRALL   EQU     0          ;WRITE TO ALLOCATED
                 +0218 ;
'0956  00        +0219 READOP: DEFB    0          ; READ OPERATION FLAG
'0957  00        +0220 RSFLAG: DEFB    0          ; READ DATA FLAG
'0958  00        +0221 WRTYPE: DEFB    0          ; UNALLOCATED FLAG
'0959  00        +0222 UNACNT: DEFB    0          ; UNALLOCATED SECTOR COUNT
'095A  00        +0223 UNADSK: DEFB    0          ; UNALLOCATED DISK #
'095B  0000      +0224 UNATRK: DEFW    0          ; UNALLOCATED TRACK #
'095D  00        +0225 UNASEC: DEFB    0          ; UNALLOCATED SECTOR #
'095E  00        +0226 SECHST: DEFB    0          ; WANTED SECTOR # IN HOST SIZE
'095F  00        +0227 HSTWRT: DEFB    0          ; TEST IF HOST WRITTEN FLAG
'0960  00        +0228 HOFST:  DEFB    0          ; 128 BYTE HOST BUFFFER OFFSET
'0961  00        +0229 HSTACT: DEFB    0          ; HOST ACTIVE FLAG
                 +0230 ;
                  0954 ;
                  0955 ;       INCLUDE THE H89-4 CONTROLLER MODULE
                  0956 ;
                  0957         INCLUDE CDRPD.SRC
                 +0001 ;
                 +0002 ;       H17 DISK AND I/O ROUTINES FOR HEATH BIOS
                 +0003 ;       WRITTEN BY ZENITH/HEATH
                 +0004 ;       REWRITTEN IN Z80 SOURCE AND MODIFIED FOR USE WITH FDC-880H
                 +0005 ;
                 -0006 ;       NOTE:   SPECIAL PORTS (3B-3F HEX) USED BY THE FDC-880H
                 +0007 ;               IN THIS SOURCE VERSION. THE FDC-880H CAN BE
                 +0008 ;               SET TO A PORT THAT IS MORE CONVENIENT FOR YOUR
                 +0009 ;               SYSTEM CONTACT C.D.R. SYSTEMS FOR MORE DETAILS
                 +0010 ;
>0003            +0011 IOBYTE  EQU     RAMST+3          ; I/O DEVICE ASSIGNMENT BYTE
                 +0012 ;
                 +0013 ;DEFAULT PORT ASSIGNMENTS
                 +0014 ;
>00FA            +0015 H85CRT  EQU     372Q
>00D0            +0016 H84TTY  EQU     0D0H             ; DEFAULTED IF 0D0H USED
>00E8            +0017 H84CRT  EQU     0E8H
>00E0            +0018 H84LPT  EQU     0E0H
>00D8            +0019 H84RDP  EQU     0D8H
                 +0020 ;
                 +0021 ;BAUD RATE DIVISORS FOR 8250'S
                 +0022 ;
>0600            +0023 B75     EQU     1536
>0417            +0024 B110    EQU     1047
```

```
ADDR  CODE       STMT SOURCE STATEMENT

>0359            +0025 B134    EQU     857
>0180            +0026 B300    EQU     384
>00C0            +0027 B600    EQU     192
>0060            +0028 B1200   EQU     96
>0030            +0029 B2400   EQU     48
>0018            +0030 B4800   EQU     24
>000C            +0031 B9600   EQU     12
>0006            +0032 B19200  EQU     6
                 +0033 ;
                 +0034 ;ASCII VALUES
                 +0035 ;
>0000            +0036 NULL    EQU     00H
>0003            +0037 CTLC    EQU     03H
>0007            +0038 BELL    EQU     07H
>000D            +0039 CR      EQU     0DH
>000A            +0040 LF      EQU     0AH
>001B            +0041 ESC     EQU     1BH
>0045            +0042 CLR     EQU     'E'
>000D            +0043 PADCH   EQU     CR                      ;CHAR FOR NULL PADS, MUST [illegible]
                 +0044 ;
                 +0045 ;DEFAULT I/O BYTE
                 +0046 ;          CON: = CRT:
                 +0047 ;          RDR: = UR1:
                 +0048 ;          PUN: = UP1:
                 +0049 ;          LST: = UL1:
                 +0050 ;
                 +0051
>0000            +0052 TTY     EQU     0
                 +0053
>0001            +0054 CRT     EQU     1
>0001            +0055 PTR     EQU     1
>0001            +0056 PTP     EQU     1
                 +0057
>0002            +0058 BAT     EQU     2
>0002            +0059 UR1     EQU     2
>0002            +0060 UP1     EQU     2
>0002            -0061 LPT     EQU     2
                 +0062
>0003            +0063 UC1     EQU     3
>0003            +0064 UR2     EQU     3
>0003            +0065 UP2     EQU     3
>0003            +0066 UL1     EQU     3
                 +0067 ;
                 +0068 ; H17 DISK RELATED EQUATES
                 +0069 ;
                 +0070
>00F2            +0071 H89CTL  EQU     0F2H                    ;H89 CONTROL PORT
>00F0            +0072 H8CTL   EQU     0F0H                    ;H8 CONTROL PORT
>0002            +0073 CLKF    EQU     002H                    ; TURN ON 2MS CLOCK
>0020            +0074 ZORG    EQU     020H                    ; ZERO ORIGIN
>00D0            +0075 H8CLKF  EQU     0D0H                    ; TURN ON H8 2MS CLOCK
>0090            +0076 H8ZORG  EQU     090H                    ; TURN OFF H8 2MS CLOCK
                 +0077
>007C            +0078 UPDP    EQU     07CH                    ;DISK DATA PORT
>007D            +0079 UPFC    EQU     07DH                    ;FILL CHARACTER
>007D            +0080 UPST    EQU     07DH                    ;STATUS FLAGS
>007E            +0081 UPSC    EQU     07EH                    ;SYNC CHARACTER (OUTPUT)
>007E            +0082 UPSR    EQU     07EH                    ;SYNC RESET (INPUT)
```

```
ADDR  CODE       STMT SOURCE STATEMENT

>007F            +0083 DPDC     EQU      07FH              :DISK CONTROL PORT
                 +0084 ;
                 +0085 ;        H17 DISK RELATED EQUATES CONTINUED
                 +0086 :
>0002            +0087 U0       EQU      02H               :H17     UNIT 0
>0004            +0088 U1       EQU      04H               :        UNIT 1
>0008            +0089 U2       EQU      08H               :        UNIT 2
>0010            +0090 DFMO     EQU      10H               :MOTOR ON (ALL DRIVES)
>0020            +0091 DFDI     EQU      20H               :DIRECTION (0 = OUT)
>0040            +0092 DFST     EQU      40H               :STEP COMMAND (ACTIVE HIGH)
                 +0093
>0002            +0094 DFTC     EQU      02H               :TRACK 0 DETECT
>0004            +0095 DFWP     EQU      04H               :WRITE PROTECT
>0008            +0096 DFSD     EQU      08H               :SYNC DETECT
                 +0097
>00FD            +0098 DSYN     EQU      0FDH              :PREFIX SYNC CHARACTER
                 +0099
>0014            +0100 LPSA     EQU      20                :# OF TRIES FOR SECTOR
>0005            -0101 STSA     EQU      7/2+1             :MS/2 TO WAIT FOR INDEX HOLE
>0007            +0102 STSB     EQU      12/2+1            :MS/2 TO WAIT PAST INDEX HOLE
>0014            +0103 WHDA     EQU      20                :UDLY COUNT FOR HOLE DEBOUNCE
>0014            +0104 WHNA     EQU      20                :UDLY COUNT FOR HOLE DEBOUNCE
>0050            +0105 WSCA     EQU      64*25/20          :LOOP COUNT FOR 25 CHARACTERS
>0014            +0106 WRITA    EQU      20                :GUARDBAND COUNT FOR WRITE
>000A            +0107 WRITB    EQU      10                :# OF ZERO CHARS. AFTER HOLE
>0010            +0108 WRITC    EQU      128/8             :TWO CHAR. DELAY BEFORE WRITING
>0030            +0109 READA    EQU      48                :DELAY BEFORE HUNT MODE
>00FA            +0110 SPD      EQU      250               :250 * 4MS = 1 S
>0014            +0111 HLTG     EQU      20                : 20 * 4MS = 80 MS
>0006            +0112 HST      EQU      24/4              :HEAD SETTLE TIME 24 MS
>000F            +0113 STEPR    EQU      30/2              :STEP RATE MS/2
>060F            +0114 DELAYS   EQU      6*256+15          :HEAD LOAD, MOTOR ON TIMES
                 +0115
>0008            +0116 CLKVEC   EQU      0008H             :CLOCK INTERRUPT VECTOR
>000B            +0117 TICCNT   EQU      000BH             :TWO BYTE TICK COUNTER
>000D            +0118 CTLPRT   EQU      000DH             :CURRENT CONTENTS OF H89 CONTROL LATCH
>000E            +0119 H8FLAG   EQU      000EH             :CONTENTS = 0 FOR H/Z89, = [illegible] FOR H8
>000F            +0120 DEVCTL   EQU      000FH             :CURRENT CONTENTS OF H17 CONTROL LATCH
                 +0121
>0001            +0122 D$E$TRK  EQU      001H              :BAD TRACK ERROR
>0002            +0123 D$E$HSY  EQU      002H              :HEADER SYNC ERROR
>0004            +0124 D$E$HCK  EQU      004H              :HEADER CHECKSUM
>0008            +0125 D$E$CHK  EQU      008H              :CHECKSUM ERROR
>0010            +0126 D$E$RNF  EQU      010H              :RECORD NOT FOUND
>0020            +0127 D$E$MDS  EQU      020H              :MISSING DATA SYNC
>0040            +0128 D$E$WRP  EQU      040H              :WRITE PROTECT ERROR
                 +0129 :
                 +0130 :   2 MS CLOCK INTERRUPT SERVICE ROUTINE
                 +0131 :
'0962  00        +0132 DLYMO:   DEFB     0
'0963  00        +0133 DLYH:    DEFB     0
'0964  00        +0134 DLYW:    DEFB     0
'0965  220D0C'   +0135 CLOCK:   LD       (HSAV),HL         :SAVE AF,HL
'0968  E1        +0136          POP      HL                :GET THE RETURN ADDRESS
'0969  220F0C'   +0137          LD       (RETSAV),HL       :SAVE IT, BUT NOT ON USER STACK
'096C  F5        +0138          PUSH     AF                :SAVE AF, HL
'096D  210D00    +0139          LD       HL,CTLPRT         :GET CURRENT CONTROL PORT VALUE
'0970  7E        +0140          LD       A,(HL)
```

```
'0971  D3F2     +0141          OUT    (H89CTL),A       ;OUTPUT AGAIN, RESET INT REQ
'0973  2A0B00   +0142 CLK0:    LD     HL,(TICCNT)      ;GET THE TICK COUNTER
'0976  23       +0143          INC    HL               ;INCREMENT IT
'0977  220B00   +0144          LD     (TICCNT),HL
'097A  7D       +0145          LD     A,L              ;IS IT MULTIPLE OF 1/2 SECOND?
'097B  B7       +0146          OR     A
'097C  201F     +0147          JR     NZ,CLKRET        ;  IF NOT
'097E  216209'  +0148          LD     HL,DLYMO         ;POINTER TO MOTOR DELAY TIMER
'0981  7E       +0149          LD     A,(HL)
'0982  B7       +0150          OR     A                ;IF ALREADY ZERO
'0983  2818     +0151          JR     Z,CLKRET         ;  THEN DON'T DECREMENT
                +0152
'0985  35       +0153          DEC    (HL)             ;DECREMENT TIMER
'0986  200A     +0154          JR     NZ,CLK1          ;  IF NO TIME-OUT, CHECK HEADS
'0988  3A0F00   +0155          LD     A,(DEVCTL)       ; GET CONTROL PORT VALUE
'098B  E6EF     +0156          AND    0FFH-DFMO        ;TURN OFF MOTOR
'098D  320F00   +0157          LD     (DEVCTL),A
'0990  D37F     +0158          OUT    (DPDC),A
'0992  23       +0159 CLK1:    INC    HL               ;POINT TO THE HEAD DELAY
'0993  7E       +0160          LD     A,(HL)
'0994  B7       +0161          OR     A                ;IF ALREADY ZERO
'0995  2806     +0162          JR     Z,CLKRET         ;  THEN DON'T DECREMENT
'0997  35       +0163          DEC    (HL)             ;DECREMENT TIMER
'0998  2003     +0164          JR     NZ,CLKRET        ;  SKIP IF NOT TIMED OUT
'099A  CDB609'  +0165          CALL   DSLFDC           ; DESELECT THE FDC-880H
'099D  3A0B00   +0166 CLKRET:  LD     A,(TICCNT)
'09A0  CB1F     +0167          RR     A                ; IS IT EVEN, TICKING AT 4MS
'09A2  3808     +0168          JR     C,CLKR2
'09A4  216409'  +0169          LD     HL,DLYW          ;CHECK WAIT TIMER
'09A7  7E       +0170          LD     A,(HL)           ;  DECREMENT IT IF IT IS NOT
'09A8  B7       +0171          OR     A                ;  ALREADY ZERO
'09A9  2801     +0172          JR     Z,CLKR2
'09AB  35       +0173          DEC    (HL)
'09AC  F1       +0174 CLKR2:   POP    AF               ;RESTORE THE MACHINE STATE
'09AD  2A0F00'  +0175          LD     HL,(RETSAV)
'09B0  E5       +0176          PUSH   HL
'09B1  2A0D00'  +0177          LD     HL,(HSAV)
'09B4  FB       +0178          EI
'09B5  C9       +0179          RET
                +0180 ;
                +0181 ;        DESELECT THE FDC-880H
                +0182 ;
'09B6  3A3505'  +0183 DSLFDC:  LD     A,(CURSEL)       ; GET THE CURRENT DRIVE SELECT
'09B9  E670     +0184          AND    _DROPDV          ; DROP THE DRIVE NUMBERS
'09BB  D33B     +0185          OUT    (DSELEC),A       ; SEND TO CONTROLLER
'09BD  CDCA09'  +0186          CALL   DSLH17           ; CLEAR H17 CONTROLLER
'09C0  3E60     +0187          LD     A,_STEPOT        ; SEND STEP OUT COMMAND (DEANO MOD)
'09C2  D33C     +0188          OUT    (DCMD),A         ; (FOR TWO CONTROLLERS ON SAME CABLE)
'09C4  3E0F     +0189          LD     A,15             ; WAIT FOR COMMAND ACCEPTANCE
'09C6  CDF307'  +0190          CALL   MSECS
'09C9  C9       +0191          RET
                +0192 ;
                +0193 ;        DESELECT THE H17 DRIVE
                +0194 ;
'09CA  3A0F00   +0195 DSLH17:  LD     A,(DEVCTL)                ; CLEAR DEVICE CONTROL
'09CD  E6F1     +0196          AND    0FFH-U0-U1-U2
'09CF  320F00   +0197          LD     (DEVCTL),A
'09D2  D37F     +0198          OUT    (DPDC),A
```

```
 ADDR  CODE        STMT SOURCE STATEMENT

'09D4  C9         +0199               RET
                  +0200 ;
                  +0201
                  +0202               IF      AHSTYP   ; ASSEMBLE IF H17 TYPE NEEDED
                  +0203
                  +0204 ;
                  +0205 ; RD17 - READ A SELECTED SECTOR
                  +0206 ;
                  +0207
'09D5  CD5705'    +0208 RD17:         CALL    BLKSET            ; SET PONTER TO DPB
'09D8  CD3705'    +0209               CALL    DSLCHK            ; CHECK IF FDC-880H DESELECTED
'09DB  3E88       +0210               LD      A,_READSC         ; SET CODE AS READ
'09DD  32EC01'    +0211               LD      (CMDSV),A
'09E0  CD640A'    +0212               CALL    SDP               ;SET PARAMETERS FOR THIS OPERATION
'09E3  CDC10A'    +0213               CALL    SDT               ;SEEK THE DESIRED TRACK
'09E6  CD130B'    +0214               CALL    LPS               ;FIND THE PROPER SECTOR
'09E9  381C       +0215               JR      C,RW17E           ;COULDN'T FIND IT
'09EB  0600       +0216               LD      B,0               ;READ 256 BYTES
'09ED  2AD90E'    +0217               LD      HL,(HSTLOC)       ;POINTER TO BUFFER
'09F0  CDA50B'    +0218               CALL    WSC               ;WAIT FOR SYNC
'09F3  3E20       +0219               LD      A,D$E$MDS         ;MISSING DATA SYNC ERROR
'09F5  3810       +0220               JR      C,RW17E           ;MISSING SYNC BYTE
                  +0221
'09F7  CD970B'    +0222 RD172:        CALL    RDB               ;READ A BYTE FROM THE DISK
'09FA  77         +0223               LD      (HL),A            ;PUT IT IN MEMORY
'09FB  23         +0224               INC     HL                ;INCREMENT POINTER
                  +0225
'09FC  05         +0226               DEC     B                 ;COUNT BYTE AS READ
'09FD  20F8       +0227               JR      NZ,RD172                  ;MORE TO READ
                  +0228
'09FF  42         +0229               LD      B,D
'0A00  CD970B'    +0230               CALL    RDB               ;CHECK CHECKSUM
'0A03  B8         +0231               CP      B
'0A04  C8         +0232               RET     Z                 ;EVERYTHING IS OKAY
'0A05  3E08       +0233               LD      A,D$E$CHK         ;SIGNAL CHECKSUM ERROR
                  +0234
'0A07  FB         +0235 RW17E:        EI                        ; REENABLE INTERRUPTS
'0A08  324200     +0236               LD      (ERRST),A         ; SET ERROR TYPE
'0A0B  E67F       +0237               AND     07FH
'0A0D  FE01       +0238               CP      D$E$TRK           ; TEST IF TRACK TYPE ERROR
'0A0F  2007       +0239               JR      NZ,RW17E2         ; RETURN IF NOT
'0A11  CDE10A'    +0240               CALL    STZ               ; SET TRACK 0
'0A14  3A4200     +0241               LD      A,(ERRST)
'0A17  B7         +0242               OR      A                 ; SET AS ERROR
'0A18  C9         +0243 RW17E2: RET                             ; END OF H17 READ/WRITE
                  +0244 ;
                  +0245 ; WR17 - WRITE A SECTOR
                  +0246 ;
'0A19  CD5705'    +0247 WR17:         CALL    BLKSET            ; SET DPB POINTER
'0A1C  CD3705'    +0248               CALL    DSLCHK            ; CHECK IF FDC-880H DESELECTED
'0A1F  3EA8       +0249               LD      A,_WRITSC         ;SET CODE AS WRITE
'0A21  32EC01'    +0250               LD      (CMDSV),A
'0A24  CD640A'    +0251               CALL    SDP               ;SET DISK PARAMETERS
'0A27  DB7F       +0252 WR171:        IN      A,(DPDC)          ;SEE IF WRITE PROTECTED
'0A29  E604       +0253               AND     DFWP
'0A2B  3E40       +0254               LD      A,D$E$WRP         ;POSSIBLE WRITE PROTECT ERROR
'0A2D  20D8       +0255               JR      NZ,RW17E          ;YES, IT IS A WRITE PROTECT ERROR
'0A2F  CDC10A'    +0256               CALL    SDT               ;GET CORRECT TRACK
```

```
 ADDR  CODE       STMT SOURCE STATEMENT

'0A32  CD130B'   +0257            CALL    LPS              ;FIND THE PROPER SECTOR
'0A35  38D0      +0258            JR      C,RW17E          ;COULDN'T FIND IT
'0A37  0600      +0259            LD      B,0              ;256 BYTES/SECTOR
'0A39  2AD90E'   +0260            LD      HL,(HSTLOC)      ;POINTER TO SOURCE OF DATA
'0A3C  3E14      +0261            LD      A,WRITA
'0A3E  3D        +0262 WR172:     DEC     A
'0A3F  20FD      +0263            JR      NZ,WR172
'0A41  0E0A      +0264            LD      C,WRITB
'0A43  3E10      +0265            LD      A,WRITC
'0A45  CDC30B'   +0266            CALL    WSP              ;WRITE THE SYNC PATTERN
'0A48  7E        +0267 WR173:     LD      A,(HL)
'0A49  CDD60B'   +0268            CALL    WNB              ;WRITE THIS DATA BYTE
'0A4C  23        +0269            INC     HL
'0A4D  05        +0270            DEC     B
'0A4E  20F8      +0271            JR      NZ,WR173         ;LOOP TO WRITE ALL 256 BYTES
'0A50  7A        +0272            LD      A,D              ;WRITE CHECKSUM
'0A51  CDD60B'   +0273            CALL    WNB
'0A54  CDD60B'   +0274            CALL    WNB              ;CONTINUE TUNNEL ERASE
'0A57  CDD60B'   +0275            CALL    WNB              ;FOR 3 CHARACTER TIMES
'0A5A  CDD60B'   +0276            CALL    WNB
'0A5D  3A0F00    +0277            LD      A,(DEVCTL)       ;OFF WRITE GATE
'0A60  D37F      +0278            OUT     (DPDC),A
'0A62  AF        +0279            XOR     A
'0A63  C9        +0280            RET
                 +0281 ;
                 +0282 ; SDP - SET DEVICE PARAMETERS
                 +0283 ;          SET RETRY COUNT, SET MOTOR ON AND SELECT DRIVE
'0A64  FB        +0284 SDP:       EI                       ; ENABLE INTERRUPTS
'0A65  210000    +0285            LD      HL,0
'0A68  226709'   +0286            LD      (DLYMO),HL
'0A6B  3A4300    +0287            LD      A,(UNIT)         ; GET WANTED DRIVE
'0A6E  CDB70A'   +0288            CALL    H17DRV           ; GET PHYSICAL DRIVE CODE
'0A71  F610      +0289            OR      DFMO             ;TURN ON THE MOTOR
'0A73  D37F      +0290            OUT     (DPDC),A
'0A75  47        +0291            LD      B,A
'0A76  210F00    +0292            LD      HL,DEVCTL        ;WHAT WAS ITS STATE?
'0A79  7E        +0293            LD      A,(HL)
'0A7A  E610      +0294            AND     DFMO             ;WAS THE MOTOR ON?
'0A7C  2011      +0295            JR      NZ,SPD2          ;YES, DON'T HAVE TO WAIT FOR IT
'0A7E  DD7E01    +0296            LD      A,(IX+1)         ; GET THE STEP RATE
'0A81  CB17      +0297            RL      A
'0A83  3EFA      +0298            LD      A,SPD            ;UP TO SPEED IN SPD * 4 MS
'0A85  3012      +0299            JR      NC,SPD3
'0A87  CB1F      +0300            RR      A                ;NEW DRIVES UP IN 1/4 TIME
'0A89  CB1F      +0301            RR      A
'0A8B  E63F      +0302            AND     03FH
'0A8D  180A      +0303            JR      SPD3
'0A8F  7E        +0304 SPD2:      LD      A,(HL)
'0A90  E60E      +0305            AND     U0+U1+U2         ;CHECK THE AVAILABLE UNITS
'0A92  A0        +0306            AND     B                ;WAS THIS UNIT SELECTED?
'0A93  3E00      +0307            LD      A,0
'0A95  2002      +0308            JR      NZ,SPD3          ;THIS HEAD WAS ALREADY LOADED
'0A97  3E14      +0309            LD      A,HLTG           ;MUST WAIT FOR HEAD LOAD TIMING
'0A99  326409'   +0310 SPD3:      LD      (DLYW),A
'0A9C  78        +0311            LD      A,B
'0A9D  320F00    +0312            LD      (DEVCTL),A
'0AA0  3A4300    +0313            LD      A,(UNIT)         ; GET THE CURRENT DRIVE
'0AA3  D604      +0314            SUB     4                ; SET H17 OFFSET
```

```
ADDR  CODE      STMT SOURCE STATEMENT

'0AA5  21D100'  +0315          LD     HL,H17TRK         ; GET H17 TRACK TABLE
'0AA8  1600     +0316          LD     D,0
'0AAA  5F       +0317          LD     E,A
'0AAB  19       +0318          ADD    HL,DE             ; GET POINTER TO CURRENT TRACK
'0AAC  22D400'  +0319          LD     (TRKPT),HL        ; SET AS TRACK POINTER
'0AAF  7E       +0320          LD     A,(HL)
'0AB0  CB17     +0321          RL     A                 ;IF MSB IS 0
'0AB2  D0       +0322          RET    NC                ;  THEN TRKPT POINTS TO TRACK
'0AB3  CDE10A'  +0323          CALL   STZ               ;  ELSE HEAD POSITION UNKNOWN
'0AB6  C9       +0324          RET                      ;     AND IS ZEROED
                +0325 :
                +0326 :        CALCULATE H17 DRIVE CODE
                +0327 :
'0AB7  D603     +0328 H17DRV:  SUB    3                 ; DROP EARLY DRIVES
'0AB9  CD9C04'  +0329          CALL   BITSHF            ; CALCULATE H17 DRIVE CODE
'0ABC  C9       +0330          RET
                +0331 :
                +0332 : SDT - SEEK DESIRED TRACK
                +0333 :        SEEK TO TRACK UPDATING *TRKPT
                +0334 :
'0ABD  34       +0335 SDT0:    INC    (HL)
'0ABE  CDEE0A'  +0336          CALL   MAI
                +0337
'0AC1  2AD400'  +0338 SDT:     LD     HL,(TRKPT)
'0AC4  3A4500   +0339          LD     A,(TRACK)
'0AC7  BE       +0340          CP     (HL)
'0AC8  2808     +0341          JR     Z,SDT1            ;AT DESIRED TRACK
'0ACA  30F1     +0342          JR     NC,SDT0           ;  MUST MOVE ARM IN
                +0343                                   ;  ELSE MUST MOVE ARM OUT
'0ACC  35       +0344          DEC    (HL)
'0ACD  CDF20A'  +0345          CALL   MAO
                +0346
'0AD0  18EF     +0347          JR     SDT
                +0348
                +0349
'0AD2  3A6409'  +0350 SDT1:    LD     A,(DLYW)          ;DELAY FOR HEAD SETTLE TIME
'0AD5  FE06     +0351          CP     HST               ;IS WAIT > HEAD SETTLE
'0AD7  D0       +0352          RET    NC                ;IF SO, RETURN
'0AD8  3E06     +0353          LD     A,HST             ;  ELSE DELAY FOR HEAD SETTLE
'0ADA  326409'  -0354          LD     (DLYW),A
'0ADD  C9       +0355          RET
                +0356
                +0357 :
                +0358 ; STZ - SEEK TRACK 0
                +0359 :        CALLED DURING ERROR RECOVERY AND
                +0360 :        TO INITIALLY POSITION HEADS
                +0361
'0ADE  CDF20A'  +0362 STZ0:    CALL   MAO               ;MOVE THE ARM OUT
'0AE1  DB7F     +0363 STZ:     IN     A,(DPDC)          ;CHECK THE TRACK ZERO SENSOR
'0AE3  E602     +0364          AND    DFT0
'0AE5  28F7     +0365          JR     Z,STZ0            ;IF NOT SET, THEN STEP OUT
'0AE7  2AD400'  +0366          LD     HL,(TRKPT)        ;ZERO TRACK NUMBER FOR DRIVE
'0AEA  3600     +0367          LD     (HL),0
'0AEC  18E4     +0368          JR     SDT1              ;HEAD DELAY FOR GOING TO 0
                +0369
                +0370 :
                +0371 : MAI - MOVE ARM IN
                +0372 : MAO - MOVE ARM OUT
```

```
 ADDR  CODE       STMT SOURCE STATEMENT

                  +0373 ;
                  +0374
'0AEE  3E20       +0375 MAI:     LD      A,DFDI            ;SET DIRECTION
'0AF0  1801       +0376          JR      MA01
'0AF2  AF         +0377 MA0:     XOR     A                 ;SET DIRECTION
'0AF3  E5         +0378 MA01:    PUSH    HL
'0AF4  67         +0379          LD      H,A
'0AF5  3A0F00     +0380          LD      A,(DEVCTL)        ;GET CURRENT DISK PORT VALUE
'0AF8  B4         +0381          OR      H                 ;OR IN DIRECTION
'0AF9  D37F       +0382          OUT     (DPDC),A          ; SEND IT TO DISK
'0AFB  F640       +0383          OR      DFST              ;OR IN STEP
'0AFD  D37F       +0384          OUT     (DPDC),A          ; SEND IT TO DISK
'0AFF  EE40       +0385          XOR     DFST              ;CLEAR STEP
'0B01  D37F       +0386          OUT     (DPDC),A          ; SEND IT TO DISK
'0B03  DD7E11     +0387          LD      A,(IX+17)
'0B06  E67F       +0388          AND     07FH
'0B08  E1         +0389          POP     HL
                  +0390 ;        CALL    DLY               ;IMPLICIT CALL DLY AND RET
                  +0391 ;        RET
                  +0392
                  +0393          ENDIF
                  +0394
                  +0395 ;
                  +0396 ; DLY - DELAY A * 2 MS
                  +0397 ;
                  +0398
'0B09  E5         +0399 DLY:     PUSH    HL
'0B0A  210B00     +0400          LD      HL,TICCNT         ;TICK COUNT PTR, INC. EVERY 2MS
'0B0D  86         +0401          ADD     A,(HL)            ;VALUE OF TICCNT AFTER DELAY
'0B0E  BE         +0402 DLY1:    CP      (HL)              ;WAIT FOR TICCNT TO CATCH UP
'0B0F  20FD       +0403          JR      NZ,DLY1
'0B11  E1         +0404          POP     HL
'0B12  C9         +0405          RET
                  +0406
                  +0407          IF      AHSTYP  ; ASSEMBLE IF H17 TYPE NEEDED
                  +0408
                  +0409 ;
                  +0410 ; LPS - LOCATE PROPER SECTOR
                  +0411 ;
                  +0412
'0B13  3A4400     +0413 LPS:     LD      A,(SECTR)         ; GET SECTOR NUMBER
'0B16  CD5E07'    +0414          CALL    TRANSL            ; TRANSLATE THE SECTOR
'0B19  3D         +0415          DEC     A                 ; SET AS 0-9 SECTOR COUNT
'0B1A  320A00'    +0416          LD      (TSECTR),A        ; SET TRANSLATED SECTOR
'0B1D  1803       +0417          JR      LPSS              ; BRANCH TO TEST READY READ
'0B1F  CD5D0B'    +0418 LPS0:    CALL    STS               ;SKIP THIS SECTOR
                  +0419
'0B22  3A6409'    +0420 LPSS:    LD      A,(DLYW)          ;READY TO READ YET?
'0B25  B7         +0421          OR      A
'0B26  20F7       +0422          JR      NZ,LPS0           ;IF NOT, WAIT A SECTOR TIME
                  +0423
'0B28  0614       +0424          LD      B,LPSA
                  +0425
'0B2A  F3         +0426 LPS1:    DI
'0B2B  CDA50B'    +0427          CALL    WSC               ;WAIT FOR A SYNC CHARACTER
'0B2E  3E02       +0428          LD      A,D$E$HSY         ;FLAG HEADER SYNC ERROR
'0B30  3821       +0429          JR      C,LPS2            ;COULDN'T FIND ONE
                  +0430
 ADDR  CODE       STMT SOURCE STATEMENT

'0B32  CD970B'   +0431              CALL    RDB              :READ THE VOLUME NUMBER
'0B35  CD970B'   +0432              CALL    RDB              :READ THE TRACK NUMBER
'0B38  214500    +0433              LD      HL,TRACK
'0B3B  BE        +0434              CP      (HL)
'0B3C  3E01      +0435              LD      A.D$E$TRK        :BAD TRACK ERROR
'0B3E  2013      +0436              JR      NZ,LPS2          :WRONG TRACK
                 +0437
'0B40  CD970B'   +0438              CALL    RDB              :READ THE SECTOR NUMBER
'0B43  21D600'   +0439              LD      HL,TSECTR        : POINT TO TRANSLATED SECTOR
'0B46  BE        +0440              CP      (HL)
'0B47  3E10      +0441              LD      A.D$E$RNF        :RECORD NOT FOUND ERROR
'0B49  2008      +0442              JR      NZ,LPS2          :WRONG SECTOR
                 +0443
'0B4B  62        +0444              LD      H,D
'0B4C  CD970B'   +0445              CALL    RDB              :DO CHECKSUM ON HEADER
'0B4F  BC        +0446              CP      H
'0B50  C8        +0447              RET     Z                        :OKAY
'0B51  3E04      +0448              LD      A.D$E$HCK        :HEADER CHECKSUM IS WRONG
                 +0449
'0B53  F5        +0450 LPS2:        PUSH    AF
'0B54  CD5D0B'   +0451              CALL    STS              :SKIP THIS SECTOR
'0B57  F1        +0452              POP     AF
'0B58  05        +0453              DEC     B                :ANOTHER TIME PASSES QUICKLY
'0B59  20CF      +0454              JR      NZ,LPS1
                 +0455
'0B5B  37        +0456              SCF                      :ENOUGH ALREADY
'0B5C  C9        +0457              RET
                 +0458
                 +0459 :
                 +0460 : STS - SKIP THIS SECTOR
                 +0461 :            EXIT AT BEGINNING OF NEXT SECTOR
                 +0462 :            1. IF HEAD NOT OVER HOLE, WAIT 8 MS FOR HOLE CHECKING.
                 +0463 :               IF NO HOLE IN THIS TIME, THEN IN REGULAR GAP,
                 +0464 :               WAIT FOR THE NEXT HOLE AND EXIT.
                 +0465 :            2. IF HEAD IS OVER HOLE, OR IS SO DURING THE 8 MS,
                 +0466 :               THEN WAIT FOR HOLE TO PASS. WAIT 10 MS IN CASE OF
                 +0467 :               INDEX HOLE, THEN WAIT FOR THE NEXT HOLE AND EXIT.
                 +0468 :
                 +0469
'0B5D  FB        +0470 STS:         EI
'0B5E  C5        +0471              PUSH    BC
'0B5F  DB7F      +0472              IN      A,(DPDC)         :CHECK THE DISK PORT
'0B61  CB1F      +0473              RR      A                :FOR SECTOR HOLES
'0B63  3812      +0474              JR      C,STS2           :CURRENTLY OVER A HOLE
                 +0475
                 +0476 ;            NO HOLE YET, WAIT 8 MS MIN (10 MS MAX) FOR HOLE
                 +0477
'0B65  210800    +0478              LD      HL,TICCNT
'0B68  46        +0479              LD      B,(HL)
'0B69  DB7F      +0480 STS1:        IN      A,(DPDC)
'0B6B  CB1F      +0481              RR      A
'0B6D  3808      +0482              JR      C,STS2           :FOUND A HOLE
                 +0483
'0B6F  3E05      +0484              LD      A,STSA
'0B71  80        +0485              ADD     A,B
'0B72  BE        +0486              CP      (HL)
'0B73  20F4      +0487              JR      NZ,STS1          :8 MS STILL NOT UP
'0B75  1808      +0488              JR      STS3             :FOUND A SECTOR GAP
```

```
 ADDR  CODE       STMT SOURCE STATEMENT

                  +0489
                  +0490 ;         HAVE HOLE. SKIP IT AND WAIT 12 MS
                  +0491
'0B77  CD8B0B'    +0492 STS2:     CALL    WNH                ;WAIT FOR NO HOLE
'0B7A  3E07       +0493           LD      A,STSB
'0B7C  CD090B'    +0494           CALL    DLY
'0B7F  C1         +0495 STS3:     POP     BC
'0B80  F3         +0496           DI
                  +0497
                  +0498 ; WHD - WAIT HOLE DETECT
                  +0499 ;
'0B81  DB7F       +0500 WHD:      IN      A,(DPDC)           ;WATCH THE DISK CONTROL PORT
'0B83  CB1F       +0501           RR      A                  ;UNTIL A HOLE IS FOUND
'0B85  30FA       +0502           JR      NC,WHD             ; STILL NO HOLE
'0B87  3E14       +0503           LD      A,WHDA             ;SET UP LOOP DELAY COUNT
'0B89  1808       +0504           JR      UDLY
                  +0505 ;
                  +0506 ; WNH - WAIT FOR NO HOLE
                  +0507 ;
'0B8B  DB7F       +0508 WNH:      IN      A,(DPDC)           ;WATCH THE DISK CONTROL PORT
'0B8D  CB1F       +0509           RR      A                  ;UNTIL CURRENT HOLE IS PAST
'0B8F  38FA       +0510           JR      C,WNH
'0B91  3E14       +0511           LD      A,WHNA             ;SET UP LOOP DELAY COUNT
                  +0512
                  +0513           ENDIF
                  +0514
                  +0515 ;
                  +0516 ; UDLY - MICROSECOND DELAY
                  +0517 ;         CALLED WITH INTERRUPTS DISABLED TO WAIT
                  +0518 ;         A * ( 15 / 2.048 ) MICROSECONDS ON 8080
                  +0519 ;         A * ( 14 / 2.048 ) MICROSECONDS ON Z80
'0B93  3D         +0520 UDLY:     DEC     A
'0B94  20FD       +0521           JR      NZ,UDLY
'0B96  C9         +0522           RET
                  +0523
                  +0524           IF      AHSTYP  ; ASSEMBLE IF AHS TYPE NEEDED
                  +0525
                  +0526 ;
                  +0527 ; RDB - READ BYTE FROM DISK
                  +0528 ;
'0B97  DB7D       +0529 RDB:      IN      A,(UPST)           ;IS A BYTE READY?
'0B99  CB1F       +0530           RR      A
'0B9B  30FA       +0531           JR      NC,RDB             ;  WAIT UNTIL READY
'0B9D  DB7C       +0532           IN      A,(UPDP)           ;GET THE BYTE
'0B9F  5F         +0533           LD      E,A                ;SAVE IT IN E
'0BA0  AA         +0534           XOR     D                  ;UPDATE CRC
'0BA1  07         +0535           RLCA
'0BA2  57         +0536           LD      D,A
'0BA3  7B         +0537           LD      A,E                ;RESTORE BYTE READ TO A
'0BA4  C9         +0538           RET
                  +0539 ;
                  +0540 ; WSC - WAIT SYNC CHARACTER
                  +0541 ;         WSC WAITS FOR APPEARANCE OF SYNC CHARACTER. DISK
                  +0542 ;         SHOULD BE SELECTED. MOVING. AND HEAD SHOULD BE OVER
                  +0543 ;         PRE-SYNC ZERO BAND
                  +0544 ;
                  +0545 ;         IF SYNC IS NOT FOUND IN 25 CHARACTER TIMES. ERROR
                  +0546 ;
 ADDR  CODE       STMT SOURCE STATEMENT

'0BA5  3E30     +0547 WSC:    LD      A,READA          ;DELAY PAST GARBAGE BYTE
'0BA7  3D       +0548 WSC0:   DEC     A
'0BA8  20FD     +0549         JR      NZ,WSC0
'0BAA  3EFD     +0550         LD      A,DSYN           ;SET UP SYNC CHARACTER
'0BAC  D37E     +0551         OUT     (UPSC),A
'0BAE  DB7E     +0552         IN      A,(UPSR)
'0BB0  3E50     +0553         LD      A,WSCA           ;COUNT OF LOOPS IN 25 CHARS.
'0BB2  57       +0554         LD      D,A
'0BB3  DB7F     +0555 WSC1:   IN      A,(DPDC)
'0BB5  E608     +0556         AND     DFSD             ;CHECK FOR SYNC
'0BB7  2005     +0557         JR      NZ,WSC2          ;GOT IT
'0BB9  15       +0558         DEC     D
'0BBA  20F7     +0559         JR      NZ,WSC1          ;TRY UNTIL TIME-OUT
'0BBC  37       +0560         SCF                      ;COULDN'T FIND SYNC
'0BBD  C9       +0561         RET
                +0562 ;
                +0563 ;       FOUND SYNC CHARACTER
'0BBE  DB7C     +0564 WSC2:   IN      A,(UPDR)         ;GOBBLE THE SYNC CHARACTER
'0BC0  1600     +0565         LD      D,0              ;CLEAR CHECKSUM
'0BC2  C9       +0566         RET
                +0567 ;
                +0568 ; WSP - WRITE SYNC PATTERN
                +0569 ;       WSP WRITES ZEROS SYNC PATTERN, FOLLOWED BY SYNC CHAR.
                +0570 ;
                +0571 ;       ENTRY   A        INITIAL DELAY COUNTER
                +0572 ;               C        NUMBER OF ZERO BYTES TO WRITE
                +0573
'0BC3  3D       +0574 WSP:    DEC     A                ;DELAY
'0BC4  20FD     +0575         JR      NZ,WSP
                +0576
                +0577 ;       DELAY IS UP- TURN ON WRITE GATE
                +0578
'0BC6  3A0F00   +0579         LD      A,(DEVCTL)
'0BC9  3C       +0580         INC     A                ;SET WRITE GATE ON
'0BCA  D37F     +0581         OUT     (DPDC),A
                +0582
                +0583 ;       WRITE # OF ZEROS SPECIFIED IN C
                +0584
'0BCC  AF       +0585 WSP1:   XOR     A
'0BCD  CDD60B'  +0586         CALL    WNB              ;WRITE A ZERO
'0BD0  0D       +0587         DEC     C                ;COUNT IT
'0BD1  20F9     +0588         JR      NZ,WSP1
                +0589
'0BD3  3EFD     +0590         LD      A,DSYN           ;WRITE A SYNC CHARACTER
'0BD5  57       +0591         LD      D,A              ; PRE-CLEAR CHECKSUM
                +0592 ;       JR      WNB              ;IMPLICIT CALL, RETURN WNB
                +0593
                +0594 ;
                +0595 ; WNB - WRITE NEXT BYTE
                +0596 ;       WRITE BYTE TO DISK PRESUMING WRITE GATE ALREADY ON
                +0597 ;
                +0598
'0BD6  5F       +0599 WNB:    LD      E,A              ;SAVE CHAR. TO BE WRITTEN
'0BD7  DB7D     +0600 WNB1:   IN      A,(UPST)         ;IS USRT READY FOR A CHAR.
'0BD9  A7       +0601         AND     A                ;SET FLAGS
'0BDA  F2D70B'  +0602         JP      P,WNB1           ;NOT READY, WAIT SOME MORE
                +0603
'0BDD  7B       +0604         LD      A,E              ;GET CHARACTER
```

```
      (C) C.D.R. SYSTEMS INC.  BIOS   SD SYSTEMS Z80 ASSEMBLER PAGE 0050
 ADDR  CODE       STMT SOURCE STATEMENT

'0BDE  D37C      +0605           OUT     (UPDP),A          :WRITE IT TO DISK
'0BE0  AA        +0606           XOR     D                 :UPDATE CRC
'0BE1  07        +0607           RLCA
'0BE2  57        +0608           LD      D,A
'0BE3  C9        +0609           RET
                 +0610
                 +0611           ENDIF
                 +0612
                 +0613 :
                 +0614 :         INITIALIZE CLOCK INTERRUPT
                 +0615 :
                 +0616
'0BE4  3EC3      +0617 CLKINT:   LD      A,0C3H            :  SET CLOCK INTERRUPT
'0BE6  320800    +0618           LD      (CLKVEC),A
'0BE9  216509'   +0619           LD      HL,CLOCK
'0BEC  220900    +0620           LD      (CLKVEC+1),HL
'0BEF  212200    +0621           LD      HL,ZORG+CLKE      : SET GENERAL PURPOSE PORT
'0BF2  220D00    +0622           LD      (CTLPRT),HL
'0BF5  2100FB    +0623           LD      HL,0FB00H
'0BF8  220F00    +0624           LD      (CLKVEC+7),HL
'0BFB  3EC9      +0625           LD      A,0C9H
'0BFD  321100    +0626           LD      (CLKVEC+9),A
                 +0627 :         CALL    ENABLE
                 +0628 :         RET
                 +0629 :
                 +0630
                 +0631 :
                 +0632 :         ENABLE OTHER INTERRUPTS
                 +0633 :
'0C00  3A0D00    +0634 ENABLE:   LD      A,(CTLPRT) : GET THE INTERRUPT CONTROL
'0C03  F622      +0635           OR      ZORG+CLKE : SET THE CLOCK INTERRUPT ON
'0C05  D3F2      +0636           OUT     (H89CTL),A
'0C07  C9        +0637           RET
                 +0638 :
                 +0639 :         DISABLE OTHER INTERRUPTS
                 +0640 :
'0C08  3E20      +0641 DISABL:   LD      A,ZORG  : SHUT ALL INTERRUPTS DOWN
'0C0A  D3F2      +0642           OUT     (H89CTL),A
'0C0C  C9        +0643           RET
                 +0644 :
'0C0D            +0645 HSAV:     DEFS    2                 :SAVED HL DURING CLOCK INT.
'0C0F            +0646 RETSAV:   DEFS    2                 :SAVED RETURN ADDR.
                 +0647 :
                  0958 :
                  0959 :         INCLUDE THE CP/M 2.2 I/O DRIVERS AND STORAGE AREAS
                  0960 :
                  0961           INCLUDE CDRBE.SRC
                 +0001 :
                 +0002 :         LOGICAL DEVICE ROUTINES
                 +0003 :
                 +0004 :         THESE ROUTINES HANDLE THE LOGICAL TO PHYSICAL
                 +0005 :         DEVICE MAPPING ESTABLISHED BY THE CP/M IOBYTE
                 +0006 :
                 +0007 :
                 +0008 : CONSOLE STATUS
                 +0009 :
'0C11  CD190C'   +0010 CONST:    CALL    CONS              :GET STATUS OF SPECIFIC DEVICE
'0C14  B7        +0011           OR      A
```

```
 ADDR  CODE      STMT SOURCE STATEMENT

'0C15  C9        +0012          RET     Z                   ;IF NOT READY RETURN 0 IN A
'0C16  3EFF      +0013          LD      A,0FFH              ; ELSE RETURN FF
'0C18  C9        +0014          RET
                 +0015 ;
'0C19  3A0300    +0016 CONS:    LD      A,(IOBYTE)          ;USE BITS 1-0 FOR CONSOLE
'0C1C  CD920C'   +0017          CALL    INDXIT
'0C1F  C20C'     +0018          DEFW    TTYSTAT
'0C21  A10C'     +0019          DEFW    CRTSTAT
'0C23  270C'     +0020          DEFW    RDRST               ;2: BATCH MODE (USE READER)
'0C25  A10C'     +0021          DEFW    CRTSTAT
                 +0022 ;
                 +0023 ; READER STATUS
                 +0024 ;
'0C27  3A0300    +0025 RDRST:   LD      A,(IOBYTE)
'0C2A  0F        +0026          RRCA
'0C2B  CD930C'   +0027          CALL    ROTXIT
'0C2E  C20C'     +0028          DEFW    TTYSTAT
'0C30  BD0D'     +0029          DEFW    BUSY                ;UNIMPLEMENTED INPUTS
'0C32  0F0D'     +0030          DEFW    MDSTAT
'0C34  A10C'     +0031          DEFW    CRTSTAT
                 +0032 ;
                 +0033 ; CONSOLE INPUT
                 +0034 ;
'0C36  3A0300    +0035 CONIN:   LD      A,(IOBYTE)
'0C39  CD920C'   +0036          CALL    INDXIT
'0C3C  B90C'     +0037          DEFW    TTYIN               ;0: TTY
'0C3E  A70C'     +0038          DEFW    CRTIN               ;1: CRT
'0C40  830C'     +0039          DEFW    READER              ;2: BAT (READER INPUT)
'0C42  A70C'     +0040          DEFW    CRTIN               ;UC1: CRT INPUT, LST: OUTPUT
                 +0041 ;
                 +0042 ; CONSOLE OUT
                 +0043 ;
'0C44  3A0300    +0044 CONOUT:  LD      A,(IOBYTE)
'0C47  CD920C'   +0045          CALL    INDXIT
'0C4A  C80C'     +0046          DEFW    TTYOUT              ;0: TTY
'0C4C  B00C'     +0047          DEFW    CRTOUT              ;1: CRT
'0C4E  620C'     +0048          DEFW    LIST                ;2: BAT (OUTPUT TO LST)
'0C50  620C'     +0049          DEFW    LIST                ;UC1: CRT INPUT, LST: OUTPUT
                 +0050 ;
                 +0051 ; LISTST - LIST STATUS CHECK
                 +0052 ;
'0C52  3A0300    +0053 LISTST:  LD      A,(IOBYTE)          ;GET THE CURRENT IOBYTE
'0C55  07        +0054          RLCA                        ;SHIFT INTO POSITION
'0C56  07        +0055          RLCA
'0C57  CD920C'   +0056          CALL    INDXIT
'0C5A  260D'     +0057          DEFW    TTYOS               ;0: TTY
'0C5C  2E0D'     +0058          DEFW    CRTOS               ;1: CRT
'0C5E  4F0D'     +0059          DEFW    LPTOS               ;2: LPT
'0C60  7E0D'     +0060          DEFW    DBDOS               ;3: DIABLO
                 +0061 ;
                 +0062 ; LIST OUT
                 +0063 ;
'0C62  3A0300    +0064 LIST:    LD      A,(IOBYTE)
'0C65  07        +0065          RLCA                        ;BITS 7-6 TO 0-1
'0C66  07        +0066          RLCA
'0C67  CD920C'   +0067          CALL    INDXIT
'0C6A  C80C'     +0068          DEFW    TTYOUT              ;0: TTY
'0C6C  B00C'     +0069          DEFW    CRTOUT              ;1: CRT
```

```
 ADDR  CODE        STMT SOURCE STATEMENT

'0C6E  D10C'      +0070           DEFW    LPTOUT           ;2: LPT
'0C70  EA0C'      +0071           DEFW    DBD              ;3: DIABLO
                  +0072 ;
                  +0073 ; PUNCH OUT
                  +0074 ;
'0C72  3A0300     +0075 PUNCH:    LD      A,(IOBYTE)       ;BITS 4-5 TO 1-2
'0C75  0F         +0076           RRCA
'0C76  0F         +0077           RRCA
'0C77  0F         +0078           RRCA
'0C78  CD930C'    +0079           CALL    GOTOIT
'0C7B  C30C'      +0080           DEFW    TTYOUT           ;0: TTY
'0C7D  C00D'      +0081           DEFW    DMYOUT
'0C7F  150D'      +0082           DEFW    MDOUT            ;2: UP1 MODEM PORT OUTPUT
'0C81  B00C'      +0083           DEFW    CRTOUT
                  +0084 ;
                  +0085 ; READER IN
                  +0086 ;
'0C83  3A0300     +0087 READER:   LD      A,(IOBYTE)       ;BITS 3-2 TO 2-1
'0C86  0F         +0088           RRCA
'0C87  CD930C'    +0089           CALL    GOTOIT
'0C8A  B90C'      +0090           DEFW    TTYIN            ;0: TTY
'0C8C  BE0D'      +0091           DEFW    DMYIN
'0C8E  090D'      +0092           DEFW    MDIN             ;2: UP1 MODEM PORT INPUT
'0C90  A70C'      +0093           DEFW    CRTIN
                  +0094 ;
                  +0095 ; DISPATCH SUBROUTINE - INDEXED TABLE JUMP
                  +0096 ;
'0C92  07         +0097 INDXIT:   RLCA
'0C93  E606       +0098 GOTOIT:   AND     06H              ;MASK BITS
'0C95  E3         +0099           EX      (SP),HL          ;SAVE HL, GET TABLE ADDRESS
'0C96  85         +0100           ADD     A,L
'0C97  6F         +0101           LD      L,A
'0C98  3001       +0102           JR      NC,GOTOI1
'0C9A  24         +0103           INC     H
'0C9B  7E         +0104 GOTOI1:   LD      A,(HL)
'0C9C  23         +0105           INC     HL
'0C9D  66         +0106           LD      H,(HL)
'0C9E  6F         +0107           LD      L,A
'0C9F  E3         +0108           EX      (SP),HL          ;PUSH ROUTINE ADDRESS, OLD HL
'0CA0  C9         +0109           RET                      ;DISPATCH
                  +0110
                  +0111 ;
                  +0112 ;         PHYSICAL DEVICE ROUTINES
                  +0113 ;
                  +0114 ; ACCESSED VIA THE LOGICAL DEVICE ROUTINES ABOVE
                  +0115 ;
                  +0116 ;
                  +0117 ;"CRT" PHYSICAL STATUS ROUTINE
                  +0118 ;         USES H84PT1
'0CA1             +0119 CRTSTAT:
'0CA1  21B500'    +0120           LD      HL,H84PT1        ;POINTER TO BASE PORT
'0CA4  C3C10D'    +0121           JP      US               ;GET STATUS
                  +0122 ;
                  +0123 ;"CRT" PHYSICAL INPUT ROUTINE
                  +0124 ;
'0CA7  21B500'    +0125 CRTIN:    LD      HL,H84PT1
'0CAA  CDE20D'    +0126           CALL    UI               ;GET CHAR FROM 2650
'0CAD  E67F       +0127           AND     7FH              ;MASK PARITY
```

```
 ADDR  CODE       STMT SOURCE STATEMENT

'0CAF  C9         +0128          RET
                  +0129 ;
                  +0130 ;"CRT" PHYSICAL OUTPUT ROUTINES
                  +0131 ;
'0CB0  CD2E0D'    +0132 CRTOUT: CALL    CRTOS
'0CB3  B7         +0133          OR      A
'0CB4  28FA       +0134          JR      Z,CRTOUT
'0CB6  C3CA0D'    +0135          JP      UO              ; OUTPUT CHARACTER IN C
                  +0136 ;
                  +0137 ; TTY INPUT
                  +0138 ;
'0CB9  21B800'    +0139 TTYIN:  LD      HL,HBAPT2
'0CBC  CDE20D'    +0140          CALL    UI
'0CBF  E67F       +0141          AND     7FH             ;MASK PARITY
'0CC1  C9         +0142          RET
                  +0143 ;
                  +0144 ; TTY STATUS
                  +0145 ;
'0CC2  21B800'    +0146 TTYSTAT:LD      HL,HBAPT2
'0CC5  C3010D'    +0147          JP      UR              ;GET STATUS
                  +0148 ;
                  +0149 ; TTY OUTPUT
                  +0150 ;
'0CC8  CD260D'    +0151 TTYOUT: CALL    TTYOS
'0CCB  B7         +0152          OR      A
'0CCC  28FA       +0153          JR      Z,TTYOUT
'0CCE  C3CA0D'    +0154          JP      UO              ;OUTPUT CHARACTER IN C
                  +0155 ;
                  +0156 ; LINE PRINTER OUT
                  +0157 ;
'0CD1  3A0400'    +0158 LPTOUT: LD      A,(DCLPOS)      ;IF NO CHAR. [illegible] OUTPUT STATUS
'0CD4  B7         +0159          OR      A
'0CD5  200A       +0160          JR      NZ,LPTOU2               ; THEN SKIP THE TEST
'0CD7  CD4E0D'    +0161 LPTOU1: CALL    LPTOS           ;  IF SO, WAIT FOR READY
'0CDA  B7         +0162          OR      A
'0CDB  28FA       +0163          JR      Z,LPTOU1
'0CDD  21BE00'    +0164 LPTOU2: LD      HL,HBAPT3       ;POINTER TO DEVICE STRUCTURE
'0CE0  110700'    +0165          LD      DE,LPTOTS       ;AND ONE TO CHAR TO SEND
'0CE3  AF         +0166          XOR     A               ;CHECK LP OUTPUT STATUS
'0CE4  320400'    +0167          LD      (DCLPOS),A      ;  NEXT TIME
'0CE7  C3CA0D'    +0168          JP      UO
                  +0169 ;
                  +0170 ; DIABLO ETX/ACK PROTOCOL DRIVER
                  +0171 ;
'>0CEA            +0172 DBD:    EQU     $
'0CEA  CD7E0D'    +0173          CALL    DBDOS
'0CED  B7         +0174          OR      A
'0CEE  28FA       +0175          JR      Z,DBD
'0CF0  CDCA0D'    +0176          CALL    UO              ;SEND CHAR. IN C TO PRINTER
'0CF3  21080D'    +0177          LD      HL,HSCNT        ;UPDATE HANDSHAKE COUNT
'0CF6  35         +0178          DEC     (HL)
'0CF7  FE1B       +0179          CP      01BH            ;ESC?
'0CF9  7E         +0180          LD      A,(HL)
'0CFA  2006       +0181          JR      NZ,DBD1         ;WAS NOT AN ESCAPE
'0CFC  FE02       +0182          CP      2               ;LAST CHAR WAS ESCAPE,
'0CFE  D0         +0183          RET     NC              ; TWO CHARS MUST FOLLOW
'0CFF  3602       +0184          LD      (HL),2          ;  WITHOUT INTERVENING ETX
'0D01  C9         +0185          RET
```

```
ADDR  CODE       STMT SOURCE STATEMENT

'0D02 B7        +0186 DBD1:   OR      A                 ;TIME TO HANDSHAKE?
'0D03 C0        +0187         RET     NZ
'0D04 3E01      +0188         LD      A,1               ; SET DBDOS FOR HANDSHAKE
'0D06 12        +0189         LD      (DE),A
'0D07 C9        +0190         RET
'0D08 20        +0191 HSCNT:  DEFB    32
                +0192 ;
                +0193 ; MDIN - MODEM INPUT ROUTINE
                +0194 ;
'0D09 21BE00'   +0195 MDIN:   LD      HL,H84PT4
'0D0C C3E20D'   +0196         JP      UI
                +0197 ;
                +0198 ; MDSTAT - MODEM INPUT STATUS
                +0199 ;
'0D0F 21BE00'   +0200 MDSTAT: LD      HL,H84PT4
'0D12 C3C10D'   +0201         JP      US
                +0202 ;
                +0203 ; MDOUT - MODEM OUTPUT
                +0204 ;
'0D15 CD1E0D'   +0205 MDOUT:  CALL    MDOS
'0D18 B7        +0206         OR      A
'0D19 28FA      +0207         JR      Z,MDOUT
'0D1B C3CA0D'   +0208         JP      UO
                +0209 ;
                +0210 ; MDOS, TTYOS, AND CRTOS - MODEM, TTY, AND CRT OUTPUT STATUS
                +0211 ;        RETURNS 00 FOR BUSY
                +0212 ;                FF FOR READY TO ACCEPT ANOTHER CHARACTER
'0D1E 21BE00'   +0213 MDOS:   LD      HL,H84PT4
'0D21 11CF00'   +0214         LD      DE,MDCTS
'0D24 180E      +0215         JR      CRTOS1
'0D26 21B800'   +0216 TTYOS:  LD      HL,H84PT2
'0D29 11C500'   +0217         LD      DE,TTYCTS
'0D2C 1806      +0218         JR      CRTOS1
'0D2E 21B500'   +0219 CRTOS:  LD      HL,H84PT1
'0D31 11C600'   +0220         LD      DE,CRTCTS
'0D34 7E        +0221 CRTOS1: LD      A,(HL)            ;CAN UART TAKE A CHAR
'0D35 C605      +0222         ADD     A,5
'0D37 CDEC0D'   +0223         CALL    PIN
'0D3A E620      +0224         AND     020H              ;IF NOT
'0D3C 280F      +0225         JR      Z,CRTOSB                    ;  THEN RETURN BUSY
'0D3E 1A        +0226         LD      A,(DE)            ;ANY NULLS TO BE SENT
'0D3F B7        +0227         OR      A
'0D40 2002      +0228         JR      NZ,CRTOS2                   ;IF SO, GO SEND ONE
'0D42 3D        +0229         DEC     A                 ;ELSE, SET READY
'0D43 C9        +0230         RET
'0D44 3D        +0231 CRTOS2: DEC     A                 ;COUNT THIS NULL AS SENT
'0D45 12        +0232         LD      (DE),A
'0D46 C5        +0233         PUSH    BC
'0D47 0E00      +0234         LD      C,NULL            ;SEND A NULL
'0D49 CDCA0D'   +0235         CALL    UO
'0D4C C1        +0236         POP     BC
'0D4D AF        +0237 CRTOSB: XOR     A                 ;RETURN STILL BUSY
'0D4E C9        +0238         RET
                +0239 ;
                +0240 ; LPTOS - LINE PRINTER OUTPUT STATUS
                +0241 ;        WITH HARDWARE HANDSHAKE
                +0242 ;
'0D4F 21BB00'   +0243 LPTOS:  LD      HL,H84PT3
```

```
 ADDR  CODE      STMT SOURCE STATEMENT

'0D52  11C700'  +0244          LD    DE,LPTCTS
'0D55  7E       +0245          LD    A,(HL)          ; CAN UART TAKE A CHAR?
'0D56  C605     +0246          ADD   A,5
'0D58  CDEDOD'  +0247          CALL  PIN
'0D5B  E620     +0248          AND   20H
'0D5D  281D     +0249          JR    Z,LPTOSB                 ;THE UART STILL BUSY
                +0250 ;
                +0251 ;        CTS HANDSHAKE USED FOR H14/WH24
                +0252 ;        IF PRINTER DOES NOT USE CTS HANDSHAKE TO SHOW "BUSY"
                +0253 ;                DELETE THE NEXT 5 LINES
                +0254 ;
'0D5F  7E       +0255          LD    A,(HL)          ;GET THE BASE ADDRESS
'0D60  C606     +0256          ADD   A,6             ;ADD OFFSET FOR 8250 MODEM STAT
'0D62  CDEDOD'  +0257          CALL  PIN
'0D65  CDCD00'  +0258          CALL  CTSTST          ; TEST CLEAR TO SEND BIT
'0D68  2012     +0259          JR    NZ,LPTOSB       ;THE PRINTER IS BUSY
'0D6A  1A       +0260          LD    A,(DE)          ;ANY NULLS TO SEND?
'0D6B  B7       +0261          OR    A
'0D6C  2005     +0262          JR    NZ,LPTOS1              ;YES, NULLS REQUIRED
'0D6E  3D       +0263          DEC   A               ;NO, RETURN A = 0FFH (READY)
'0D6F  320400'  +0264          LD    (DOLPOS),A      ;FLAG DON'T CHECK LP STATUS
'0D72  C9       +0265          RET
'0D73  3D       +0266 LPTOS1:  DEC   A               ;COUNT THIS NULL AS SENT
'0D74  12       +0267          LD    (DE),A
'0D75  C5       +0268          PUSH  BC              ;SAVE THE ORIGINAL CHARACTER
'0D76  0E00     +0269          LD    C,NULL
'0D78  CDCA0D'  +0270          CALL  UO
'0D7B  C1       +0271          POP   BC
'0D7C  AF       -0272 LPTOSB:  XOR   A               ;INDICATE BUSY
'0D7D  C9       +0273          RET
                -0274 ;
                +0275 ; DBDOS - DIABLO OUTPUT STATUS
                +0276 ;        IF CTS == 0 THEN OKAY TO SEND CHARACTERS
                +0277 ;           CTS == 1 THEN SEND ETX, SET CTS TO 2
                +0278 ;           CTS == 2 THEN WAIT FOR ACK, THEN SET CTS TO 0
                +0279 ;
'0D7E  215B00'  +0280 DBDOS:   LD    HL,H84PT3
'0D81  11C900'  +0281          LD    DE,DBDCTS
'0D84  1A       +0282          LD    A,(DE)          ;FIND OUT THE STATE OF OUTPUT
'0D85  FE02     +0283          CP    2               ;IF NOT 2,
'0D87  201A     +0284          JR    NZ,DBDOS1              ;  THEN GO DO OUTPUT
                +0285 ; MUST RECEIVE AN ACK FROM THE PRINTER
'0D89  7E       +0286          LD    A,(HL)          ;CHECK UART FOR INCOMING
'0D8A  C605     +0287          ADD   A,5
'0D8C  CDEDOD'  +0288          CALL  PIN
'0D8F  E601     +0289          AND   1
'0D91  2829     +0290          JR    Z,DBDOSB        ;NO CHAR, FROM PRINTER YET
'0D93  7E       +0291          LD    A,(HL)
'0D94  CDEDOD'  +0292          CALL  PIN             ;GET THE CHARACTER
'0D97  E67F     +0293          AND   07FH            ;STRIP OFF PARITY
'0D99  D606     +0294          SUB   6               ;COMPARE IT TO ACK
'0D9B  201F     +0295          JR    NZ,DBDOSB       ;NOT AN ACK, SO STILL BUSY
'0D9D  12       +0296          LD    (DE),A          ;WAS AN ACK, CAN SEND MORE CHARS
'0D9E  3E20     +0297          LD    A,32            ;RESET THE HANDSHAKE COUNT
'0DA0  32080D'  +0298          LD    (HSCNT),A
'0DA3  7E       +0299 DBDOS1:  LD    A,(HL)          ;CAN UART SEND ANOTHER CHAR?
'0DA4  C605     +0300          ADD   A,5
'0DA6  CDEDOD'  +0301          CALL  PIN
```

```
 ADDR  CODE       STMT SOURCE STATEMENT

'0DA9  E620      +0302            AND     020H
'0DAB  280F      +0303            JR      Z,DBDOSB          ;UART NOT READY FOR A CHAR.
'0DAD  1A        +0304            LD      A,(DE)            ;IS IT TIME TO SEND ETX?
'0DAE  B7        +0305            OR      A
'0DAF  2002      +0306            JR      NZ,DBDOS2                 ;YES, GO SEND ETX
'0DB1  3D        +0307            DEC     A                 ;NO, INDICATE READY (A = 0FFH)
'0DB2  C9        +0308            RET
'0DB3  3C        +0309 DBDOS2:    INC     A                 ; FLAG TO WAIT FOR ACK
'0DB4  12        +0310            LD      (DE),A
'0DB5  C5        +0311            PUSH    BC
'0DB6  0E03      +0312            LD      C,3      ; 'C' MOD 32 (SEND THE ETX)
'0DB8  CDCA0D'   +0313            CALL    UO
'0DBB  C1        +0314            POP     BC
'0DBC            +0315 BUSY:
'0DBC  AF        +0316 DBDOSB:    XOR     A
'0DBD  C9        +0317            RET
                 +0318 ;
                 +0319 ; DUMMY INPUT AND OUTPUT ROUTINES
                 +0320 ;
'0DBE  3E1A      +0321 DMYIN:     LD      A,'Z'-40H         ;UNIMPLEMENTED INPUTS GET CTL-Z
'0DC0  C9        +0322 DMYOUT:    RET                       ;DUMMY OUTPUTS DO NOTHING
                 +0323 ;
                 +0324 ; 8250 I/O ROUTINES
                 +0325 ;
                 +0326 ; US - GET UART (INPUT) STATUS
'0DC1  7E        +0327 US:        LD      A,(HL)
'0DC2  C605      +0328            ADD     A,5               ;OFFSET TO THE STATUS REGISTER
'0DC4  CDEC0D'   +0329            CALL    PIN
'0DC7  E601      +0330            AND     1                 ;MASK THE DATA AVAILABLE BIT
'0DC9  C9        +0331            RET
                 +0332 ;
                 +0333 ; UO - OUTPUT TO UART
                 +0334 ;
'0DCA  7E        +0335 UO:        LD      A,(HL)
                 +0336 ;          JR      POUT
                 +0337 ;
                 +0338 ; POUT - OUTPUT BYTE IN C TO PORT IN A
                 +0339 ;
'0DCB  32D00D'   +0340 POUT:      LD      (POUT1+1),A
                 +0341 ;          PUSH    HL                ; * MAP TO UPPER CASE CODES
                 +0342 ;          INC     HL                ; * DROPPED TO SAVE SPACE
                 +0343 ;          INC     HL                ;POINT TO FLAG BYTE
                 +0344 ;          LD      A,(HL)            ; * INFREQUENTLY USED
                 +0345 ;          RLA
                 +0346 ;          POP     HL
'0DCE  79        +0347            LD      A,C
                 +0348 ;          CALL    C,MUC             ;MAP TO UPPER CASE
'0DCF  D300      +0349 POUT1:     OUT     (00H),A           ;SELF-MODIFYING CODE
'0DD1  FE0D      +0350 POUT2:     CP      PADCH             ;CHAR NEEDS PADDING? (CR)
'0DD3  C0        +0351            RET     NZ                        ;NO
'0DD4  E5        +0352            PUSH    HL                ;FIND NUMBER OF NULLS NEEDED
'0DD5  23        +0353            INC     HL
'0DD6  23        +0354            INC     HL
'0DD7  7E        +0355            LD      A,(HL)            ;GET COUNT FROM DATA STRUCTURE
'0DD8  E1        +0356            POP     HL
'0DD9  1F        +0357            RRA                       ;SHIFT INTO LEAST SIG 3 BITS
'0DDA  1F        +0358            RRA
'0DDB  1F        +0359            RRA
```

```
 ADDR  CODE       STMT SOURCE STATEMENT

'0DDC  1F        +0360           RRA
'0DDD  E607      +0361           AND     07H              ;MASK ONLY COUNT
'0DDF  C8        +0362           RET     Z                ;RETURN IF NO NULLS REQUIRED
'0DE0  12        +0363           LD      (DE),A           ;SAVE COUNT OF NULLS IN XXXCTR
'0DE1  C9        +0364           RET
                 +0365 ;
                 +0366 ; UI - INPUT FROM UART
                 +0367 ;
'0DE2  7E        +0368 UI:       LD      A,(HL)
'0DE3  C605      +0369           ADD     A,5
'0DE5  CDEC0D'   +0370           CALL    PIN
'0DE8  1F        +0371           RRA
'0DE9  30F7      +0372           JR      NC,UI
'0DEB  7E        +0373           LD      A,(HL)
                 +0374 ;         JR      PIN
                 +0375 ;
                 +0376 ; PIN - INPUT BYTE FROM PORT IN A
                 +0377 ;
'0DEC  32F00D'   +0378 PIN:      LD      (PIN1+1),A
'0DEF  DB00      +0379 PIN1:     IN      A,(00H)          ;SELF-MODIFYING CODE
'0DF1  C9        +0380           RET
                 +0381 ;
                 +0382 ; MUC - MAP CHARACTER IN A TO UPPER CASE ( NOT USED )
                 +0383 ;
                 +0384 ;MUC:     CP      'A'              ;IF LESS THAN LOWER CASE A
                 +0385 ;         RET     C                ;  THEN ALREADY UPPER CASE
                 +0386 ;         CP      'Z'+1            ;IF GREATER THAN LOWER CASE Z
                 +0387 ;         RET     NC               ; THEN NOT A LOWER CASE LETTER
                 +0388 ;         SUB     'A'-'A'          ;CONVERT TO UPPER CASE
                 +0389 ;         RET
                 +0390 ;
                 +0391 ; PMSG - PRINT THE MESSAGE AT HL UNTIL NULL
                 +0392 ;
'0DF2  7E        +0393 PMSG:     LD      A,(HL)           ;GET A CHAR
'0DF3  B7        +0394           OR      A                ;CHECK FOR NULL
'0DF4  C8        +0395           RET     Z                ;END OF MESSAGE
'0DF5  4F        +0396           LD      C,A              ; ELSE
'0DF6  E5        +0397           PUSH    HL               ;SAVE THE POINTER
'0DF7  CD440D'   +0398           CALL    CONOUT           ; PRINT THIS CHARACTER
'0DFA  E1        +0399           POP     HL
'0DFB  23        +0400           INC     HL               ; POINT TO NEXT
'0DFC  18F4      +0401           JR      PMSG             ;REPEAT
                 +0402 ;
                 +0403 ;         SEND A SPACE, THEN A HEX BYTE
                 +0404 ;
'0DFE  F5        +0405 SHOUT:    PUSH    AF       ; SAVE HEX BYTE
'0DFF  0E20      +0406           LD      C,20H    ; SEND BYTE
'0E01  CD440D'   +0407           CALL    CONOUT
'0E04  F1        +0408           POP     AF
                 +0409 ;         CONTINUE WITH HOUT ROUTINE
                 +0410 ;
                 +0411 ; HOUT - HEX OUTPUT ROUTINE
                 +0412 ;         TYPE CONTENTS OF A IN HEX ON CONSOLE
                 +0413 ;
'0E05  F5        +0414 HOUT:     PUSH    AF               ;SAVE CONTENTS OF A
'0E06  0F        +0415           RRCA
'0E07  0F        +0416           RRCA
'0E08  0F        +0417           RRCA
```

```
 ADDR  CODE       STMT SOURCE STATEMENT

'0E09  0F         +0418          RRCA
'0E0A  CD0E0E'    +0419          CALL   NIBBLE          ;PUT OUT HIGH ORDER NIBBLE
'0E0D  F1         +0420          POP    AF              ;FALL THROUGH FOR LOW NIBBLE
'0E0E  E60F       +0421 NIBBLE:  AND    0FH             ;MASK
'0E10  FE0A       +0422          CP     10              ;> 10 ?
'0E12  3802       +0423          JR     C,NIBBL1        ;IF 0-9
'0E14  C607       +0424          ADD    A,7             ;  ELSE CONVERT TO A-F
'0E16  C630       +0425 NIBBL1:  ADD    A,30H           ;BINARY TO ASCII
'0E18  4F         +0426          LD     C,A             ;TYPE IT ON THE CONSOLE
'0E19  C3440D'    +0427          JP     CONOUT
                  +0428 ;
                  +0429 ;        ERROR REPORT ROUTINE
                  +0430 ;
'0E1C  F5         +0431 ERRSND:  PUSH   AF
'0E1D  C5         +0432          PUSH   BC
'0E1E  D5         +0433          PUSH   DE
'0E1F  E5         +0434          PUSH   HL
'0E20  21470E'    +0435          LD     HL,ERRMS1 ; SEND ERROR MESSAGE
'0E23  CDF20D'    +0436          CALL   PMSG
'0E26  0604       +0437          LD     B,4      ; SET COUNT OF SEND BUFFER
'0E28  214300     +0438          LD     HL,ERRST ; START OF SEND BUFFER
'0E2B  7E         +0439 ERRSN1:  LD     A,(HL)   ; GET A BYTE
'0E2C  E5         +0440          PUSH   HL
'0E2D  C5         +0441          PUSH   BC
'0E2E  CDFE0D'    +0442          CALL   SHOUT    ; SEND SPACE AND BYTE TO CONSOLE
'0E31  C1         +0443          POP    BC
'0E32  E1         +0444          POP    HL
'0E33  23         +0445          INC    HL       ; SET FOR NEXT BYTE
'0E34  10F5       +0446          DJNZ   ERRSN1   ; BRANCH UNTIL BUFFER SENT
'0E36  DD7E0F     +0447          LD     A,(IY-15) ; GET THE DISK TYPE
'0E39  CDFE0D'    +0448          CALL   SHOUT
'0E3C  3A8D01'    +0449          LD     A,(CMDSV)
'0E3F  CDFE0D'    +0450          CALL   SHOUT
'0E42  E1         +0451          POP    HL
'0E43  D1         +0452          POP    DE
'0E44  C1         +0453          POP    BC
'0E45  F1         +0454          POP    AF
'0E46  C9         +0455          RET
                  +0456 ;
                  +0457 ; ERROR MESAGES
                  +0458 ;
'0E47  0D0A       +0459 ERRMS1:  DEFB   CR,LF
'0E49  45525220   +0460          DEFM   'ERR DR SC TK TP CD'
       44522053
       4320544B
       20545020
       4344
'0E5B  0D0A00     +0461          DEFB   CR,LF,0
                  +0462 ;
                  +0463 ;  INITIALIZE THE 8250S
                  +0464 ;
'0E5E  2AB600'    +0465 CBT1:    LD     HL,(CRTBAUD)    ;PICK UP BAUD RATE
'0E61  3AB500'    +0466          LD     A,(H84PT1)      ;  AND THE PORT NUMBER
'0E64  CD920E'    +0467          CALL   IN8250          ;INITIALIZE THIS PORT
'0E67  2AB900'    +0468 CBT1A:   LD     HL,(TTYBAUD)
'0E6A  3AB800'    +0469          LD     A,(H84PT2)
'0E6D  CD920E'    +0470          CALL   IN8250
'0E70  2ABC00'    +0471          LD     HL,(LPTBAUD)
```

```
 ADDR  CODE       STMT SOURCE STATEMENT

'0E73  3ABB00'   +0472          LD     A,(H84PT3)
'0E76  CD920E'   +0473          CALL   IN8250
'0E79  2ABF00'   +0474          LD     HL,(RDPBAUD)
'0E7C  3ABE00'   +0475          LD     A,(H84PT4)
'0E7F  CD920E'   +0476          CALL   IN8250
'0E82  3EFF      +0477          LD     A,_NODISK        ; RESET CURRENT DRIVE CODE
'0E84  323405'   +0478          LD     (CURENT),A
'0E87  3AB300'   +0479          LD     A,(DEFIOB)       ;SET THE DEFAULT IOBYTE
'0E8A  320300    +0480          LD     (IOBYTE),A
'0E8D  CDE40B'   +0481          CALL   CLKINT           ; INITIALIZE CLOCK INTERRUPT
'0E90  FB        +0482          EI
'0E91  C9        +0483          RET
                 +0484 ;
                 +0485 ; IN8250 - INITIALIZE AN 8250
                 +0486 ;    HL CONTAINS BAUD RATE DIVISOR (WORD)
                 +0487 ;    A  HAS BASE PORT NUMBER
                 +0488 ;
'0E92  47        +0489 IN8250:  LD     B,A              ;SAVE BASE PORT NUMBER IN B
'0E93  EB        +0490          EX     DE,HL            ;MOVE BAUD RATE DIVISOR TO DE
'0E94  21D10E'   +0491          LD     HL,OUTH84+1      ;POINT TO PORT IN OUT INST.
'0E97  3E03      +0492          LD     A,3              ;BAUDR ACCESS ON BASE+3 PORT
'0E99  80        +0493          ADD    A,B              ;GET ACTUAL PORT
'0E9A  4F        +0494          LD     C,A              ;SAVE IN C FOR LATER
'0E9B  77        +0495          LD     (HL),A           ;AND MODIFY OUTPUT INSTRUCTION
'0E9C  3E83      +0496          LD     A,83H            ;SET DIVISOR LATCH ACCESS BIT
'0E9E  CDD00E'   +0497          CALL   OUTH84           ;TO A "1"
'0EA1  34        +0498          INC    (HL)             ;POINT TO MODEM CNTL. REGISTER
'0EA2  3E0F      +0499          LD     A,0FH            ;SET DSR & CTS HIGH FOR DHABLE
'0EA4  CDD00E'   +0500          CALL   OUTH84           ; AND OTHER TERMS THAT NEED IT
'0EA7  70        +0501          LD     (HL),B           ;SET PORT TO LEAST SIG BYTE
'0EA8  7B        +0502          LD     A,E
'0EA9  CDD00E'   +0503          CALL   OUTH84
'0EAC  7A        +0504          LD     A,D              ;NOW DO MOST SIG BYTE
'0EAD  E60F      +0505          AND    0FH              ;AND OFF CONTROL FLAGS
'0EAF  34        +0506          INC    (HL)             ;ON NEXT PORT
'0EB0  CDD00E'   +0507          CALL   OUTH84
'0EB3  71        +0508          LD     (HL),C           ;RESET PORT TO DIVISOR ACCESS
'0EB4  FE04      +0509          CP     4                ;IF SET FOR GREATER THAN 110
'0EB6  3E03      +0510          LD     A,3              ; THEN NO PAR, 8 BITS 1 STOP
'0EB8  3802      +0511          JR     C,IN821
'0EBA  F604      +0512          OR     4                ; ELSE 2 STOP FOR <= 110
'0EBC  CDD00E'   +0513 IN821:   CALL   OUTH84
'0EBF  35        +0514          DEC    (HL)             ;NOW SET PORT FOR INT. CONTL.
'0EC0  35        +0515          DEC    (HL)
'0EC1  AF        +0516          XOR    A                ;DISABLE ALL DEVICE INTERRUPTS
'0EC2  CDD00E'   +0517          CALL   OUTH84           ;DISABLE INTS
                 +0518 ;
                 +0519 ; DELAY FOR APPROXIMATELY TWO CHARACTER TIMES
                 +0520 ;
'0EC5  EB        +0521          EX     DE,HL            ;PUT BAUD RATE DIVISOR IN HL
'0EC6  29        +0522          ADD    HL,HL            ;MULTIPLY BY 16 TO GET DELAY
'0EC7  29        +0523          ADD    HL,HL
'0EC8  29        +0524          ADD    HL,HL
'0EC9  29        +0525          ADD    HL,HL
'0ECA  2B        +0526 LOOP1:   DEC    HL
'0ECB  7D        +0527          LD     A,L
'0ECC  B4        +0528          OR     H
'0ECD  20FB      +0529          JR     NZ,LOOP1
```

```
 ADDR  CODE      STMT SOURCE STATEMENT

'0ECF  C9        +0530          RET
                 +0531 ;
                 +0532 ; SELF MODIFYING OUT INSTRUCTION USED BY IN8250
                 +0533
'0ED0  D300      +0534 OUTH84:  OUT     (00H),A         ;PORT IS MODIFIED
'0ED2  C9        +0535          RET
                 +0536 ;
                 +0537 ;        CPM ENTRY MESSAGE AND COLD BOOT PARAMETER
                 +0538 ;        NOTE:   THIS AREA IS OVERLAYED AFTER THE INITIAL COLD
                 +0539 ;                BOOT BY CP/M PARAMETER STORAGE AREA (SMSG CHANGED)
                 +0540 ;                IF SMSG IS 0FFH THEN PROM COLD BOOT IS PERFORMED
                 +0541 ;
'0ED3  FF        +0542 SMSG:    DEFB    0FFH     ; COLD BOOT FROM PROM PARAMETER
                 +0543 ;
'0ED4  320400    +0544 PBOOT:   LD      (DEFALT),A ; SET CURRENT DEFAULT TO DRIVE A:
'0ED7  31FF15'   +0545          LD      SP,BSTACK ; SET STACK POINTER
'0EDA  DB3B      +0546          IN      A,(DSELEC) ; GET DRIVE SELECT INFORMATION
'0EDC  2F        +0547          CPL              ; SET AS ACTUAL COUNT PLUS TYPE
'0EDD  E60F      +0548          AND     _DROPTR ; DROP THE TYPE CODE
'0EDF  CD080F'   +0549          CALL    BITRSH  ; RESHIFT BIT PATTERN TO FORM A NUMBER
'0EE2  32A700'   +0550          LD      (OFFSET),A ; STORE THE DRIVE OFFSET
'0EE5  CD470E'   +0551          CALL    CRT1A   ; INITIALIZE I/O PORTS AND CLOCK
'0EE8  21300F'   +0552          LD      HL,CBMSG1 ; SEND FIRST PART OF C.B. MESSAGE
'0EEB  CDF20D'   +0553          CALL    PMSG
'0EEE  210016'   +0554          LD      HL,BIOSST+1600H ; GET END OF CP/M MEMORY SPACE
'0EF1  7C        +0555          LD      A,H      ; GET MSB FOR 64K TEST
'0EF2  CB1F      +0556          RR      A        ; SET AS DECIMAL CODE
'0EF4  CB1F      +0557          RR      A
'0EF6  E63F      +0558          AND     03FH     ; TEST IF 64K SIZE
'0EF8  2002      +0559          JR      NZ,PBOOT1 ; BRANCH IF NOT
'0EFA  3E40      +0560          LD      A,64     ; SET FOR A 64K MESSAGE
'0EFC  CD140F'   +0561 PBOOT1:  CALL    DECOUT   ; OUTPUT THE MEMORY SIZE
'0EFF  21330F'   +0562          LD      HL,CBMSG2 ; SEND REST COLD BOOT MESSAGE
'0F02  CDF20D'   +0563          CALL    PMSG
'0F05  C35702'   +0564          JP      BOOTEN   ; END BOOT WITH INITIALIZATION
                 +0565 ;
                 +0566 ;        RESHIFT THE BITS IN THE  A  REGISTER
                 +0567 ;        A = 1,2,4, OR 8 .  RETURNS A = 0,1,2, OR 3
                 +0568 ;
'0F08  C5        +0569 BITRSH:  PUSH    BC
'0F09  4F        +0570          LD      C,A      ; SET SHIFT CODE
'0F0A  AF        +0571          XOR     A        ; SET THE COUNT TO 0
'0F0B  CB19      +0572 BITRS1:  RR      C        ; SHIFT RIGHT THE CODE
'0F0D  3803      +0573          JR      C,BITRS2 ; BRANCH UNTIL NUMBER FOUND
'0F0F  3C        +0574          INC     A        ; INCREMENT THE COUNT
'0F10  18F9      +0575          JR      BITRS1
'0F12  C1        +0576 BITRS2:  POP     BC
'0F13  C9        +0577          RET
                 +0578 ;
                 +0579 ;        OUTPUT A TWO DIGIT DECIMAL NUMBER FROM THE A REGISTER
                 +0580 ;        ENTERED A=HEX NUMBER.  RETURNS WITH REGISTERS MODIFIED
                 +0581 ;
'0F14  0E00      +0582 DECOUT:  LD      C,0      ; CLEAR RESULT BYTE
'0F16  D60A      +0583 DECOU1:  SUB     10       ; SUBTRACT 10 FROM THE A REGISTER
'0F18  DA1E0F'   +0584          JP      C,DECOU2 ; BRANCH IF ALL OF RESULT BYTE COUNTED
'0F1B  0C        +0585          INC     C        ; INCREASE RESULT BYTE
'0F1C  18F8      +0586          JR      DECOU1   ; UNTIL DECIMAL MSB FOUND
'0F1E  C60A      +0587 DECOU2:  ADD     A,10     ; RESTORE DECIMAL LSB FROM NEGATIVE
```

```
      (C) C.D.R. SYSTEMS INC.  BIOS   SD SYSTEMS Z80 ASSEMBLER PAGE 0061
 ADDR  CODE        STMT SOURCE STATEMENT

'0F20  F5         +0588             PUSH    AF      ; SAVE THE DECIMAL LSB
'0F21  79         +0589             LD      A,C     ; GET THE DECIMAL MSB
'0F22  C630       +0590             ADD     A,'0'   ; SET AS ASCII DECIMAL CHARACTER
'0F24  4F         +0591             LD      C,A     ; PLACE CHARACTER FOR OUTPUT
'0F25  CD440C'    +0592             CALL    CONOUT  ; OUTPUT THE FIRST DECIMAL CHARACTER
'0F28  F1         +0593             POP     AF      ; RESTORE THE DECIMAL LSB
'0F29  C630       +0594             ADD     A,'0'   ; SET AS ASCII DECIMAL CHARACTER
'0F2B  4F         +0595             LD      C,A     ; PLACE CHARACTER FOR OUTPUT
'0F2C  CD440C'    +0596             CALL    CONOUT  ; OUTPUT THE SECOND DECIMAL CHARACTER
'0F2F  C9         +0597             RET
                  +0598 ;
                  +0599 ;           COLD BOOT MESSAGE
                  +0600 ;
'0F30  1B4500     +0601 CBMSG1: DEFB    ESC,CLR,0
'0F33  48204350 +0602 CBMSG2: DEFM    'H CP/M 2.2'
       2F4D2032
       2E32
'0F3D  0D0A       +0603             DEFB    CR,LF
'0F3F  432E442E +0604             DEFM    'C.D.R. SYSTEMS BIOS V.2.7C'
       522E2053
       59535445
       4D532042
       494F5320
       562E322E
       3743
'0F59  0D0A00     +0605             DEFB    CR,LF,0
                  +0606
                  +0607 ;
                  +0608 ;           PROM COLD BOOT OVERLAYED AREA START
                  +0609 ;
'>0ED3            +0610             ORG     SMSG    ; PARAMETER STORAGE START POINT
                  +0611 ;
                  +0612 ;           RESERVED UNINITIALIZED DATA AREA
                  +0613 ;           NOTE: THIS AREA OVERWRITES THE INITIAL CP/M ENTRY MESSAGE
                  +0614 ;
 >0ED3            +0615 TRKSTR: DEFS    2           ; TWO BYTES FOR EXPANSION
 >0ED5            +0616 SECTOR: DEFS    2           ; TWO BYTES FOR EXPANSION
 >0ED7            +0617 DISKNO: DEFS    1           ; DISK NUMBER 0-15
 >0ED8            +0618 NEWDRV: DEFS    1           ; NEW DISK NUMBER TEST
                  +0619
                  +0620 ;
                  +0621 ;           HOST DISK BUFFER
                  +0622 ;
 >0ED9            +0623 HSTLOC: DEFS    2           ; HOST BUFFER POINTER
 >0EDB            +0624 HSTBUF: DEFS    1024        ; MAXIMUM HOST BUFFER AREA
                  +0625 ;
                  +0626 ;           SCRATCH AREA FOR BDOS USE
                  +0627 ;
'>12DB            +0628 BEGDAT: EQU     $           ; BEGINNING OF DATA AREA
 >12DB            +0629 DIRBUF: DEFS    128         ; SCRATCH DIRECTORY AREA
 >135B            +0630 ALV0:   DEFS    77          ; ALLOCATION VECTOR 0
 >13A8            +0631 ALV1:   DEFS    77          ; ALLOCATION VECTOR 1
 >13F5            +0632 ALV2:   DEFS    77          ; ALLOCATION VECTOR 2
 >1442            +0633 ALV3:   DEFS    77          ; ALLOCATION VECTOR 3
 >148F            +0634 ALV4:   DEFS    12          ; ALLOCATION VECTOR 4 (H17)
 >149B            +0635 ALV5:   DEFS    12          ; ALLOCATION VECTOR 5 (H17)
 >14A7            +0636 ALV6:   DEFS    12          ; ALLOCATION VECTOR 6 (H17)
 >14B3            +0637 CSV0:   DEFS    64          ; CHECK VECTOR 0
 ADDR  CODE        STMT SOURCE STATEMENT

 >14F3            +0638 CSV1:   DEFS    64       ; CHECK VECTOR 1
 >1533            +0639 CSV2:   DEFS    64       : CHECK VECTOR 2
 >1573            +0640 CSV3:   DEFS    64       ; CHECK VECTOR 3
 >15B3            +0641 CSV4:   DEFS    16       : CHECK VECTOR 4 (H17)
 >15C3            +0642 CSV5:   DEFS    16       : CHECK VECTOR 5 (H17)
 >15D3            +0643 CSV6:   DEFS    16       : CHECK VECTOR 6 (H17)
                  +0644 :
'>15E3            +0645 ENDDAT: EQU     $        ; END OF DATA AREA
 >0308            +0646 DATSIZ: EQU     $-BEGDAT : SIZE OF DATA AREA
                  +0647 :
                  +0648 ;       STACK FOR BIOS USE  ( AT END OF MEMORY BLOCK )
                  +0649 :
'>15FF            +0650 BSTACK: EQU     BIOSST+15FFH
                   0962 :
                   0963         END

ERRORS=0000
```

```
ADDR  CODE        STMT SOURCE STATEMENT

                  0058 ;
                  0059 ;        CP/M CBIOS MODULE  ERROR MESSAGE DESCRIPTION
                  0060 ;
                  0061 ;        IF A HARD ERROR OCCURS WHILE READING OR WRITING, AN
                  0062 ;        ERROR MESSAGE WILL BE PRINTED.
                  0063 ;        THE ERROR MESSAGE IS PRINTED IN THE FOLLOWING FORMAT:
                  0064 ;
                  0065 ;        ERR DR SC TK TP CD
                  0066 ;         XX XX XX XX XX XX
                  0067 ;
                  0068 ;        ERROR-CODE,DRIVE,SECTOR,TRACK,DISK-TYPE,COMMAND-TYPE
                  0069 ;
                  0070 ;        THE ERROR CODES REPESENT THE FOLOWING ERRORS
                  0071 ;        BIT 7 = 1  DRIVE NOT READY
                  0072 ;        BIT 6 = 1  WRITE PROTECT
                  0073 ;        BIT 5 = 1  RECORD TYPE FOR READING, OR WRITE FAULT
                  0074 ;        BIT 4 = 1  SEEK ERROR, OR RECORD NOT FOUND
                  0075 ;        BIT 3 = CRC ERROR
                  0076 ;        BIT 2 = LOST DATA
                  0077 ;        BIT 1 = INDEX OR DATA REQUEST
                  0078 ;        BIT 0 = BUSY FLAG FROM DRIVE
                  0079 ;
                  0080 ;        AN  FF  CODE IS FORMED BY THIS DIAGNOSTIC SOFTWARE
                  0081 ;        IF A DRIVE IS NOT CONNECTED FOR THE SPECIFIED #,
                  0082 ;        OR A DISK IS NOT IN THE DRIVE.
                  0083 ;
                  0084 ;        DISK TYPE DESCRIBES THE TYPE OF DISK IN THE
                  0085 ;        CURRENTLY ACCESSED DRIVE, IN HEX CODE.
                  0086 ;        BIT 7 = SIDES, 0=DOUBLE SIDED, 1=SINGLE SIDED
                  0087 ;        BIT 6 = DENSITY, 0=DOUBLE, 1=SINGLE
                  0088 ;        BIT 5 = TYPE OF DISK, 0= 8 INCH, 1= 5.25 INCH
                  0089 ;        BITS 4 - 2 ARE UNUSED
                  0090 ;        BIT 1 = 5.25 EXTENDED, 0=NORMAL, 1=EXTENDED
                  0091 ;        BIT 0 = SECTOR SIZE, 0=NORMAL, 1=EXTENDED
                  0092 ;
                  0093 ;        EXAMPLE:  . . . TYPE . . .
                  0094 ;                         A0
                  0095 ;                        (10100000 IN BITS)
                  0096 ;        DRIVE 00 IS SINGLE SIDED,DOUBLE DENSITY,5.25 INCH
                  0097 ;        SPECIAL CODES:
                  0098 ;                FF = INACTIVE DRIVE
                  0099 ;                FE = ACTIVE DRIVE, NON STANDARD DISK
                  0100 ;
                  0101 ;        THE COMMAND TYPES PERFORM THE FOLLOWING FUNCTIONS
                  0102 ;        0A = RESTORE THE CURRENT DRIVE
                  0103 ;        12 = SEEK NON VERIFY
                  0104 ;        1E = NORMAL SEEK TO A TRACK COMMAND
                  0105 ;        C4 = READ A SECTOR ADDRESS
                  0106 ;        88 = READ A SECTOR COMMAND
                  0107 ;        A8 = WRITE A SECTOR COMMAND
                  0108 ;        F4 = WRITE A TRACK COMMAND
                  0109 ;        E4 = READ A TRACK INTO MEMORY COMMAND
                  0110 ;
                  0111 ;        FOR MORE INFORMATION ABOUT COMMAND AND ERROR CODES
                  0112 ;        REFERENCE THE FD 179X-02 FLOPPY DISK FORMATTER/
                  0113 ;        CONTROLLER MANUAL BY WESTERN DIGITAL.
```