42 5T 5!

Chmielewski ub 2030 - 2800

Schreuer

TABLE 0F C0NTENTS

MICRO COMPUTER - GENEVA
726 East State
Geneva, Illinois 60134

#40079103

INDUSTRIAL MICRO SYSTEMS
628 N. Eckhoff street
Orange, California 92668

TABLE OF CONTENTS

* CP/M is a trademark of Digital Research

CP/M IMPLEMENTATION
(Series 8000 and Series 5000)

The Industrial Micro Systems implementation of CP/M for the Series 5000 and Series 8000 include some extended accessing capabilities for single and double density, single and double sided diskettes. Accessing capabilities differ depending on the configuration of the system. Two versions of the system are currently available, one version designed to reside on a double-density, single- or double-sided diskette, and a second for single-density, single-sided diskettes. A Single-density, double-sided format is not supported because it essentially has the same storage capacity as the double-density, single-sided format. The Series 8000 CP/M supplied on single-density, single-sided diskettes can only access single-density formats and contain special scaled-down utility programs. The remainder of this section only applies to the double-density systems.

The Series 8000 double-density CP/M is designed to reside on either single- or double-sided diskettes. A double-sided sign-on message is printed after the cold boot when the system is loaded from a double-sided diskette. With Series 8000's supplied with single-sided drives, diskette density is automatically determined the first time the diskette is accessed, therefor after booting from a double-density diskette, a single-density diskette may be placed in drive B, then PIP may be used to transfer files from B to A, or vise versa. Drive B may also be logged in and programs in single-density may be executed directly. Series 8000's supplied with double-sided drives automatically sense the use of double-sided diskettes to extend the storage capability acordingly. In effect, this means that both density's can be accessed transparent to BDOS and all programs written under BDOS can take advantage of either density. Caution should be exercised with the use of programs that call BIOS directly for disk operations. Information about the BIOS jump table as well as the BIOS source listing are contained in this document and will aid in tailoring these programs to run on the system. The storage capacity for all three formats are as follows

Diskette format	Formatted storage	Maximum directory entries
single-density	241 kilobytes	64
double-density	482 kilobytes	128
double-sided	964 kilobytes	255

Double-density diskettes are handled by BIOS with the use of a buffer for reading and writing. Before a diskette is removed, care should be taken to insure that records written to the buffer are updated on the diskette. Any reading from a diskette or a warm boot will insure that the write buffer is flushed. Most transient programs return to CP/M using the warm boot entry.

All versions of the system contain modifications to BDOS for the handling of errors as well as extensive read/write error messages. When a permanent disk error is encountered, BIOS will print it's error message followed by a BDOS prompt. Typing a return will ignore the error and continue, while typing a CONTROL-C will warm boot the system and leave drive A selected. If an attempt is made to access a drive which is either not present on the system or is present but not ready with a diskette inserted, a "drive not ready" prompt is typed on the console. Typing a Control-c C will warm boot the system and leave drive A selected. Typing anything else will perform a reaccess after the diskette has been inserted.

A patch is provided in the BIOS to "clean up" the way BDOS handles rubouts and CONTROL-U if a high speed CRT is used as the console. The code recognized by CP/M for rubouts is 7F (hex) . It is optionally assembled with the BIOS in a second level generation of the system. The patch is made in the input line function of BDOS and works with the CCP as well as any transient program calling BDOS for this function. Note that the standard CP/M editor calls this function at the command level, but does not call BDOS for input line in the insert mode. This patch is used on all standard distributions of the system.

CP/M SYSTEM START-UP PROCEDURES

The initial boot-up of any Series 8000 or Series 5000 consists of the following: First connect any 9600 baud CRT terminal, which requires no special hand-shaking, to the channel one serial port. The system initially is set up with the channel one serial port set at 9600 baud and defined as the CP/M console. The channel two serial port is initially set at 300 baud and defined as the CP/M list device. To change the baud rate of either port, refer to the processor section (340 8080 cpu or 440/450 Z80/IO) of this manuel. Turn power on both terminal and computer. Next depress the reset switch located on the front panel and the cold boot sign-on message "should appear on the console. If the message doesn't appear then before going any further perform the following checks. Check the terminal cable from the CRT to the computer for shorts or opens then consult the BIOS section of this document for the word format used on serial ports and verify that your terminal is set correctly If the message appears at the console then insert the supplied system diskette into drive A (left hand drive, diskette label facing right) and type return at the console. Disk activity should begin immediately and the operating system will print it's sign-on message, followed by CP/M's prompt "A>" which means the system is ready to accept commands. If for some reason this doesn't happen then read on please. If the cold boot loader comes back with a "boot error", then verify that the diskette is inserted correctly into the drive. Try the boot once or twice more. Drive A with the system diskette should have it's activity light on. If errors continue, it's possible that the diskette has bad sectors on the operating system tracks. Try a back-up system disk, and if one is not availible then consult your dealer. If absolutly nothing happens when a return is typed, including no drive activity, then it's possible that the terminal cable is open in one direction, check this out. If your system is equipped with a Z80/IO board combination, then be sure the phantom line is enabled on the memory board addressed at 0. When none of the above steps lead to the ultimate problem, contact your dealer.

SYSTEM UTILITY PROGRAMS

This section explains the use of utility programs supplied with the system in addition to the Digital Research transient programs which are covered in the CP/M system features and utilities provided by Digital Research. Most of these additional programs are unique to this system and will not run on other manufacturer's implementation of CP/M. They deal with the low level drivers of the system (BIOS) and are dependent on the system hardware, i.e. disk controllers, CPUs, and I/O boards. They are affective for system level housekeeping, diskette back-ups, or media and memory validations. Following is an explaination of these programs.

FORMAT.COM - Diskette formatting utility

This program is used to setup the sector format of a diskette to be used with the supplied CP/M system. It should be used on new diskettes to insure it's compatability with the system and may be used to reformat crashed disks which read or write with . errors on the system. This programs destroys all data on the disk so proper care should be taken in recovering the data on crashed disks. The program requires no parameters at the command level when invoked and is entirely self-prompting. There are two versions of FORMAT, one distributed with the Series 8000 (8" disks) and one for the Series 5000 (5" disks) . The Series 8000 version formats 77 tracks in either single density (FM, 26 sectors by 128 bytes/sector) or double density (MFM, 26 sectors by 256 bytes/sector) . The Series 5000 version formats 40 tracks in double density (MFM, 16 sectors by 256 bytes/sector. Both versions first prompt for a drive name (A through D) The Series 8000 version if used would issue a density prompt here. Both versions issue a last chance prompt, then proceed formatting. If the system is equipped with double sided capability, FORMAT will automatically sense the presence of a double sided diskette and format the second side right after the first, still requiring only one pass on the disk. After the function is complete, FORMAT will again issue the drive name prompt and return may be typed to reboot CP/M. FORMAT will discontinue formatting on any type of error and reboot CP/M, always remaining on drive A after rebooting.

DSTAT.COM - Drive status utility .

After a. drive has been accessed for the first time, DSTAT may be run to review the format of the diskette inserted in it. No parameters are needed to run DSTAT. The drive name (A through D) is printed along with the associated format, "double-density", "double-sided", or "single-density". The information is derived from the current value of the density table located in BIOS.

COPY.COM - Diskette Copy utility

COPY is used to make a track for track direct COPY of one diskette to another. The diskettes must be of the same format or COPY will print a density error on the console and reboot CP/M. COPY accepts three copying commands, ALL, DATA, or SYSTEM, where "COPY ALL" copies the entire disk, tracks 0-76 for the Series 8000 and tracks 0-39 for the Series 5000. "COPY DATA" copies the data and directory from tracks 2-76 for the Series 8000 and tracks 2-39 for the Series 5000. "COPY SYSTEM" will transfer the first two tracks on the diskette which contain the CP/M operating system. Source and Destination drives (A through D) are specified after the COPY command in the format; Destination=Source. COPY defaults to a read after write verification unless the command line is followed by a "/" character to switch this mode off.COPY is initiated at the command level of CP/M with the form:

```
        COPY <command> Destination=Source </>
```

COPY is loaded and it reprints the source and destination drive, then a last chance prompt which accepts a CONTROL-C to reboot CP/M. At this time the diskettes involved in the COPY may be placed into the source and destination drives and return should be used to start the copy. Any pair of valid format diskettes may be placed into the drives even if a drive was previously set up for a different format. Copy calls BIOS direct for track copying and any returned errors from BIOS cause the message "IGNORE? " to be printed on the console. If a "Y" is typed, COPY will continue on with the copy. If any other key is typed, COPY will do an immediate reboot of CP/M. After the transfer is complete, a prompt to reboot CP/M is issued with the option of repeating the exact type of copy again, even for a different format.

MEMTEST.COM - Memory test utility

This utility will perform a one pass test on system memory beginning at the end of the program in the TPA (approximately 300H) to the beginning of BIOS. No parameters are needed when the program is invoked at the command level of CP/M. Once loaded, MEMTEST will begin testing and after one pass, a reboot of the system is performed. Errors are display with the memory address first, followed by the byte MEMTEST wrote at that location and the byte which was read back. In the case of multiple memory errors, CONTROL-S may be used to stop the display and examine the error.

DSKTEST.COM - Drive/Media Verification Utility

DSKTEST is used to verify the readability of every sector on the diskette. DSKTEST does not write and is otherwise non-destructive to the diskette. It is entirely self prompting and requires only the name "DSKTEST" to be typed at the command level of CP/M. There is also an optional long seek testing mode on the Series 8000 in which a seek to track 76 is performed before tracks 0-42 are read, and a seek to track 0 is performed before tracks 43-76 are read. DSKTEST . calls BIOS directly for all disk control so if an unreadable sector is found, the BIOS error message is printed first, then DSKTEST will print the physical track and sector location the error was reported. DSKTEST has a built-in sector reading interlace of 2 (or every other sector) so the entire track can be read in two revolutions of the diskette.

D.COM - list directory utility

D is a wide-screen, alphabetizing directory program used to make large diskette directories easier to read than CP/M's built in "DIR" command. D supports wildcards in the command line and defaults to listing every entry. The number of files found for a particulay look-up is printed on the console as well as the free disk space in kilobytes. The default screen width is 5 entries wide or a width may be specified by adding "/*" at the end of the command line where x is the number of entries per line.

IMSGEN.COM - Sysgen for the IMS bios

IMSGEN is functionally identical to Digital Research's Sysgen with the added capabilities for handling the double-density formats. With this program, systems may be brought into memory from either double-density format and then written out to any format. Single-density diskettes are a special case in that only a partial system will fit on the operating system tracks. This may be done to allow a single-density diskette to be placed in the operating system drive A, but only warm boots are allowed after this. Cold boots must all ways be done with a double-density diskette.

THE BIOS

This section deals with the current versions of the BIOS distributed with the Series 8000 and the Series 5000. A basic understanding of the responsibility of the BIOS, as outlined by Digital Research's CP/M alteration guide, is assumed. The NEC UPD765 document will aid in understanding the absolute disk I/O routines, and the write-ups on the IMS model 340 (8080 CPU) or model 440 (I/O board) will aid in understanding console and list related routines. The following information deals with the standard configuration of the system when delivered by IMS to it's dealer. Dealers may have, previous to the end user's delivery, altered the BIOS i.e. custom printer, and/or console drivers so information about possible changes should be checked with your dealer.

The IMS model 340 (8080 CPU) has two serial ports incorporated on the board. The base address of ports is fixed at 10 hex. Channel one serial is assigned to the console device, which is assumed to be a 9600 baud CRT terminal. Channel two serial is assigned to the list device, which operates at 300 baud. No special device handshaking is used with either serial port. The IMS model 450 (Z80 CPU) is used in conjuction with the model 440 (I/O board) which has two serial ports that are software compatible with the model 340. In addition, this board has a 24 bit parallel device (8255) assigned to ports directly above the serial ports. The standard system does not use this parallel device, but a conditional equate in the BIOS will assemble a driver for a parallel Centronics printer as the list device. The model 440 can be assigned different base addresses in 16 address increments. For the standard system, its assigned 10 hex as the base address to be compatible with the model 340 serial ports. In effect, this means the system software is independent of whether the 8080 cpu or Z80 CPU is used. Both serial ports have software selectable baud rates through the use of a programmable interval timer (8253) They are initially set for 9600 and 300 baud by the cold boot ROM, but can be reprogrammed to other baud rates in the BOOT routine of the BIOS. The serial word format can be changed the same way, but both uarts are initialized for an 8-bit word with two stop bits and no parity.

The model 401 and 430 (8" and 5" disk controllers) are set-up with their base address etched into the board at 80 hex. This can be moved in 16 address boundries but this is not recommended to avoid incompability with disk controller dependent utility programs such as FORMAT and the cold boot loader.

The cold boot of the system is handled by a ROM located on the model 340 8080 or in the case of the model 440 Z80, on the 450 I/O board. In either case, there are only two versions of this ROM for booting any format of the system, one for the Series 8000 and one for the Series 5000. The boot program loads the entire system into memory and jumps into the cold boot entry point in BIOS. The location of the system in memory is determined by an origin word located in the first sector on the diskette. A sign-on message and prompt are issued on the console before booting begins.

Currently there are four TRUE/FALSE equates for different printer drivers supplied with the standard system. Only one of these equates may be set true for proper assembly. The four equates cover the following:

```
        •PARPRN         ; parallel printer driver
```

This driver sets up the parallel device on the I/O board for handshaking required to drive most Centronics printers.

```
        SERPRN          ; standard serial driver
```

This equate is the one set true in the standard distribution of the BIOS. It will assemble a driver for the channel two serial port with no special handshaking.

```
        TTY40           ; Teletype model 40 driver
```

This driver controls the Teletype motor via the channel two serial request to send line and requires buffer overflow handshaking through the clear to send status input bit. The channel two baud rate is kicked up to 9600 baud in the BOOT routine of the BIOS.

DAISY ; Daisy wheel driver

This driver supports buffer overflow status through the clear to send status input bit of channel two for NEC Spinwriter, QUME, or Multiterm type daisy wheel printers. It assumes a high true on the clear to send bit and may need changing if this bit is low true. The channel two baud rate is changed to 1200 baud in the BOOT routine of the BIOS.

There are additional TRUE/FALSE equates possible:

REMEX ; if Remex drives are used

This conditional when set true will assemble the neccessary code into the BOOT routine to kick the previously 6 ms step rate up to 4 ms. Shugarts will not step reliably at this rate so this should only be used when Remex drives only are on the system. This code will be used with systems supplied on double-sided diskettes.

DELETE ; if clean CRT rubouts are desired

This conditional when set true will assemble code neccessary to do rubouts by backing the cursor up and erasing the character underneth it. Displayed control characters as well as tabs may also be rubbed out without loosing screen control. The patch is made in BDOS's line input function and is effective with any transient programs that call BDOS for this function. The high speed CRT terminal that is used need only support backspace (08 hex) and a destructive forward space (20 hex). As an extra added attraction, the delete line editing function control-U, will back-up and erase the line. All systems are supplied with this code.

THE SERIES 8000 BIOS:

Single- and double-density accessing, as well as double-sided is performed by maintaining a drive density table. This table is four bytes long and represents each of the four possible drives respectively. Each byte may take on one of four values only:

81 hex = double-sided
01 hex = double-density
00 hex = single-density
FF hex = not mounted (first access has not been made)

All the CP/M disk parameters for these formats as well as disk controller values are kept in BIOS as data which is patched into BDOS and BIOS "on the fly" for accessing across formats. This table must begin at a relative address to the beginning of the BIOS. The offset is 39 hex, and may always be found by adding 39 hex to the beginning of BIOS.

example:

```
        LHLD    1           -           ; get bios page
        MVI     L,39H                   ; add in offset
```

Registers HL now points to the current density of drive A. The system programmer may use this to write density independent BIOS calls. This sequence works because the beginning of BIOS will always be on an even page boundry, so register H contains that page.

Single-density diskettes may be used as the common mode for recovering files written on other CP/M systems. The BIOS expects the single-density diskette's to be formatted with 128 bytes per sector, 26 sectors per track, sequentially numbered 1 through 26 with 77 tracks per diskette (0 through 76). When BIOS is in the single-density mode, reading and writing-is done immediately from and to the disk, into and out of the preset DMA address. The standard sector interlace table in BDOS is used.

BIOS expects double-density, single-sided diskettes to be formatted with 256 byte sector, 26 sectors per track, sequentially numbered 1 through 26 with 77 tracks per diskette (0 through 76) . When BIOS is in the double-density mode, reading and writing is done from an internally maintained 256 byte buffer. Each read call from BDOS causes half of. this buffer (128 bytes) to be transferred to the current DMA address. Each write call brings 128 bytes from the DMA address into this buffer. Double-sided diskettes are handled with this same buffering but considers the track on side two of the diskette to be just an extension of the same track on side one. The second side is formatted the same way as the first. Each 256 byte physical sector on the diskette represents two 128 byte logical records for BDOS. The double-density mode does not use a sector interlace table, instead the SETSEC subroutine in BIOS contains math to determine the interlace.

The first 15 jumps in the BIOS jump table conform to Digital Research's definition as layed out in the CP/M system alteration guide for either single- or double-density and double-sided modes. In addition, four jumps have been added to the end of the table for easier direct handling of double-density diskettes and multiple sector reads and writes. These jumps do not perform the internal buffer of double-density sectors needed with BDOS. Some IMS system utility programs use these jumps, i.e. full track reads and writes. The IMS BIOS jumps perform like this.

THE SERIES 5000 BIOS:

Double-density, single-sided diskettes are the only format in the single-sided version of the system. With double-sided systems, single-sided may be accessed also. Extra jumps exist in the BIOS for multiple sector reads and writes like the Series 8000. Also the CONST and CONIN routines are written to except input from either serial port at any time.

SECOND LEVEL SYSTEM GENERATION

This section is designed as a guide for generating your own custom BIOS for use on the IMS systems. A basic understanding of the operation of the BIOS is layed out in Digital Research's CP/M system alteration guide and should be reviewed before continuing. A second level system generation is necessary if you wish to alter the BIOS in any way, even for changing memory sizes. A back-up diskette of some kind should be used to write the newly constructed system to. After which the altered system can be tested then ultimately written to all diskettes containing systems. Before altering the BIOS, please consult the section in this document on the BIOS. It may be helpful in customizing your own version, as well as for tailoring user written programs that call BIOS directly for disk operations. This section is not necessary for simply duplicating your current system on another diskette. Tools provided for this are "COPY SYSTEM" and "IMSGEN" which are covered in the previous system utility section. A second boot loader known as SBOOT is not needed with the IMS system because the cold boot loader handles the entire initial load of the system. Memory size independence is achived with the use of a system origin word located on the first sector of the diskette, normally where the SBOOT resides. This word is determined when the BIOS is assembled and placed on the first sector when BIOS hex file is overlayed on CP/M and written to a diskette using IMSGEN.

The files needed in a second level generation are:

```
DBIOS.ASM or MDDBIOS.ASM (Series 5000)
CPM.CCM
ED.COM
ASM.COM
DDT.COM
IMSGEN.COM
```

STEP #1: RE-ASSEMBLY OF THE BIOS.

Using CP/M editor "ED", IMS "polyvue/80" or other text editor, edit the MSIZE equate in BIOS to reflect to exact new memory size. This equate is the rounded memory size in kilobytes. Also note the conditional equates for different list device drivers which can be set or new custom routines can be added.

Now using "ASM", re-assemble BIOS to produce a .HEX file on the disk. Do not attemp to use load once the .HEX file is made.

STEP #2: RELOCATING CP/M SYSTEM:

Use CPM.COM to generate a relocated CP/M system for new memory size. The format of the command is as follows:

```
A>CPM 62 *
      ^  ^
      I  +---- instructs CPM.COM to leave system in TPA area

      +--- this is new memory size rounded in kilobytes
             minus 2 (CP/M is pushed down 2k for BIOS)
             eg. 64k target system.
```

Follow the instructions given by CPM.COM for saving memory image on disk in a file CPMxx.COM where xx will be the target memory size minus 2k.

STEP #3: OVERLAYING THE CUSTOM BIOS ON CP/M SYSTEM:

The next task is to overlay our custom BIOS on top of the new CP/M system. This is done with the aid of "DDT". Issue the command:

```
A>DDT CPMxx.COM
  Where "xx" is the target system size minus 2
```

DDT will load the new CP/M system into user TPA area and issue its own subsystem prompt "-". Type in the next instruction to DDT:

```
 -IDBIOS.HEX  (or MDDBIOS.HEX for the Series 5000)
```

This initializes DDT's file control block for new BIOS file. The next part of this step is to read in the re-assembled custom BIOS with the proper offset to place it into memory over CP/M. Offsets differ with every memory configuration. Some standard offsets are listed next but any offset may be computed by subtracting the absolute address of BIOS from lE80(hex). This computation must either be done in hex or converted to hex for use in DDT.

COMMON OFFSETS:

Target memory size	Offset in hex
32k	A880
36k	9880
40k	8880
44k	7880
48k	6880
52k	5880
56k	4880
60k	3880
64k	2880

Once the offset is known, issue this next DDT command to read DBIOS.HEX (or MDDBIOS.HEX) into memory.

```
-Rxxxx  .        where "xxxx" is the offset required.
```

If DDT comes back with a NEXT value of 2100 then the wrong offset has been used. The NEXT value should be in the range of 2300 to 26FF. If the value is greater than 2700 then the offset may be wrong or the new BIOS may be too large for the allocated space. Solve problems of this kind before proceeding any further. At this point the new CP/M system has been constructed in memory and is ready to be written out to your new diskette so be sure you have a new diskette formated or one ready to accept the system and leave the DDT program by typing:

```
-GO               this performs a CP/M warm boot.
```

STEP #4: WRITING NEW SYSTEM TO DISKETTE:

After leaving the DDT program, no other transient program except "IMSGEN" may be loaded due to the newly constructed system still contained in memory. Type the next command in:

```
A>IMSGEN
```

This loads the IMS system generation program which then issues the prompt:

```
IMS Sysgen vl.0 Type source drive name (or return to skip)
```

Only type a return here because new system is already in memory. IMSGEN will then issue the prompt:

```
Type destination drive name (or return to reboot)
```

Type in the drive name (a-d) where the new system is to be written. After writing the system IMSGEN will re-prompt with the same message as before to allow you to write the system to more than one diskette. After all diskettes have been written, do a reset on the front panel and insert new diskette into drive A, and verify a properly written system by trying a cold boot from the diskette.

```
                ;
                ;               dbios.asm       79oct01
                ;               (double-density, single- and double-sided 8" bios)
                ;
                ; Basic input output system for:
                ; Industrial Micro Systems Model 400 floppy disk controller board
                ;
                ; This module contains all the input/output functions
                ; pertaining to the CP/M disk operating system 4-drive version 1.4
                ;
                ; Distrabution Information: This version of bios is constructed to reside
                ; on a double density (256 bytes/sector), single or double-sided formated
                ; disk with sectors sequentially numbered from 1 to 26 on a side. Any
                ; format disk may be accessed automatic through this bios.
                ;
                ;
                ;
                        TITLE   'IMS double-density 8" CP/M bios'
                        PAGE    56
                ;
                ;
0040 =          MSIZE   EQU     64                      ; memory size
FFFF =          TRUE    EQU     -1                      ; define true
0000 =          FALSE   EQU     NOT TRUE                ; define false
0005 =          RTRY    EQU     5                       ; retry count
000D =          CR      EQU     0DH                     ; define return
000A =          LF      EQU     0AH                     ; define line feed
                ;
                ; conditional assembly equates:
                ; select only one of next four conditionals
0000 =          PARPRN  EQU     FALSE                   ; if parallel printer
FFFF =          SERPRN  EQU     TRUE                    ; if serial printer
0000 =          TTY40   EQU     FALSE                   ; if teletype model 40
0000 =          DAISY   EQU     FALSE                   ; if daisy wheel printer
                ;
0000 =          REMEX   EQU     FALSE                   ; if remex drives
FFFF =          DELETE  EQU     TRUE                    ; if crt rubouts
                ;
                ; i/o definitions
0010 =          IOB     EQU     10H                     ; i/o base address
0010 =          CCOM    EQU     IOB+0                   ; console command (uart 0)
0010 =          CSTAT   EQU     IOB+0                   ; console status
0011 =          CDATA   EQU     IOB+01H                 ; console data
0012 =          LCOM    EQU     IOB+02H                 ; list command (uart 1)
0012 =          LSTAT   EQU     IOB+02H                 ; list status
0013 =          LDATA   EQU     IOB+03H                 ; list data
0014 =          CNT0    EQU     IOB+04H                 ; counter 0 (uart 0)
0015 =          CNT1    EQU     IOB+05H                 ; counter 1 (uart 1)
0016 =          CNT2    EQU     IOB+06H                 ; counter 2 (RTC)
0017 =          SETCNT  EQU     IOB+07H                 ; set counters
0018 =          MASK    EQU     IOB+08H                 ; mask and RTS
                ;
                ; these additional ports pertain to the 440 I/O board
0019 =          RTCRES  EQU     IOB+09H                 ; RTC reset
001C =          PORTA   EQU     IOB+0CH                 ; parallel port A
001D =          PORTB   EQU     IOB+0DH                 ; parallel port B
001E =          PORTC   EQU     IOB+0EH                 ; parallel port C
```

6402 - UART -

8253 TIMER

TIMER

8255 - PP

```
#02          PCNTB.   EQU     IOB+OFH              ; parallel control
                      ;
                      ; status bit definitions
   1 =       RDA      EQU     00000001B            ; recieve data ready.
0002 =       TBE      EQU     00000010B            ; transmitter buffer empty.
0080 =       CTS      EQU     10000000B            ; clear to send bit.
                      ;
                      ; I/O control values
0036 =       SETO     EQU     36H                  ; set up counter 0
0076 =       SET1     EQU     76H                  ; set up counter
00B6 =       SET2     EQU     0B6H                 ; set up counter 2
00A2 =       PMODE    EQU     OA2H                 ; parallel device mode select
0080 =       LPBSY    EQU     080H                 ; parallel printer busy
                      ;
                      ; FDC and DMA port definitions
0080 =       DSKB     EQU     080H                 ; disk base address
0082 =       CH1DMA   EQU     DSKB+2               ; channel 1 dma
0083 =       CH1TC    EQU     DSKB+3               ; channel 1 terminal count
0088 =       DMAST    EQU     DSKB+8               ; dma status and commands
008A =       DSEL     EQU     DSKB+0AH             ; drive select port
008D =       ICS      EQU     DSK3+0DH             ; on-board interrupt command/status
008E =       FDCMSR   EQU     DSKB+0EH             ; fdc main status register
008F =       DDATA    EQU     DSKB+0FH             ; disk data
                      ;
                      ; uPD765 floppy disk controller instruction set
0003 =       SCYCMD   EQU     03H                  ; specify drive parameters
0005 =       SWRCMD   EQU     05H                  ; single density write data
0006 =       SRDCMD   EQU     06H                  ; single density Read data
  +5 =       DWRCMD   EQU     45H                  ; double —
  46 =       DRDCMD   EQU     46H                  ; double -
00C5 =       D2WCMD   EQU     OC5H                 ; dbl density, dbl sided write
00C6 =       D2RCMD   EQU     OC6H                 ; dbl density, dbl sided read
0004 =       SDSCMD   EQU     04H                  ; sense drive status
0007 =       RECCMD   EQU     07H                  ; recalibrate
0008 =       SISCMD   EQU     08H                  ; sense interrupt
000A =       RIDCMD   EQU     OAH                  ; read sector ID
000F =       SKCMD    EQU     0FH                  ;seekcommand
                      ;
                      ; Specifications for Remex drive
0024 =       HLDT     EQU     (18)*2               ; 36 ms head load time
00C0 =       SRT      EQU     (16-4)*16            ; 4 ms step rate
000F =       HUT      EQU     15                   ; 240 ms head unload time
0000 =       ND       EQU     0                    ; non-dma mode
                      ;
                      ; DMA controller codes
0080 =       DMARD    EQU     80H                  ; read dma
0040 =       DMAWR    EQU     40H                  ; write dma
0042 =       CH1ENA   EQU     42H                  ; enable channel 1
                      ;
                      ; CP/M module and size equates
FE80 =       RWBUFF   EQU     (MSIZE*1024)-180H    ; buffer locations
B800 =       CBASE    EQU     (MSIZE-18)*1024      ; move cp/m down 2k
El00 =       CPMB     EQU     CBASE+2900H          ; start of CP/M
E900 =       BDOSB    EQU     CBASE+3100H          ; bdos base address
 906 =       BDOS     EQU     CBASE+3106H          ; bdos entry point
E90F =       TRANS    EQU     BDOSB+OFH            ; fly-patch location
```

```
E93A =          BDOSTBL EQU     BDOSB+3AH               ; fly-patch location
1500 =          CPML    EQU     1500H                   ; length of cp/m
[illegible] =   NSECTS  EQU     CPML/128                ; number of sectors
                ;
                ;
                ; set up CP/M origin word before system
                ; when the bios hex file is loaded over cp/m in a second
                ; level generation, this word is loaded into the record devoted
                ; to the system bootstrap sector for the IMS cold boot rom
E080                    ORG     CPMB-128
E080 OOB8               DW      CBASE
                ;
                ;
E969                    ORG     BDOSB+69H
E969 26FD               DW      PERMERR
                ;
                ;
                ; these are permanent patches in bdos for crt rubouts
                        IF      DELETE
EA8B                    ORG     BDOSB+18BH
                ;
EA8B CD02F8             CALL    RUBOUT
EA8E C373EA             JMP     BDOSB+173H
                ;
EAB9                    ORG     BDOSB+1B9H
                ;
EAB9 CD29F8             CALL    DELBUF
                        END IF
                ;
                ;
F172                    ORG     BDOSB+872H
F172 CDF4EB             CALL    BDOSB+2F4H
F175 00                 NOP
F176 00                 NOP
F177 00                 NOP
F178 79                 MOV     A,C
F179 21F8F5             LXI     H,BDOSB+0CF8H
                ;
                ;
                ; I/O jump vector
                ; This is where CP/M calls whenever it needs to do any input/output operations
                ; user programs may use these entry points also, but note that the location
                ; of this jump vector is dependent on memory size.
                ; Additional vectors have been added to complament the use of
                ; the Industrial Micro Systems floppy disk controller board IMS400
                ;
F600            BIOS:   ORG     CPMB+CPML               ; first address in bios
                ;
F600 C3F9F6             JMP     BOOT                    ; from cold start loader
F603 C332F7     WBOOTE: JMP     WBOOT                   ; from warm boot (location 0)
F606 C3CAF7             JMP     CONST                   ; check console keyboard status
F609 C3D3F7             JMP     CONIN                   ; read console character
F60C C3ECF7             JMP     CONOT                   ; write console character
F60F C372F8             JMP     LIST                    ; write listing character
F612 C37DF8             JMP     PUNCH                   ; write punch
F615 C37EF8             JMP     READER                  : read reader
```

```
F618 C3D6FC          JMP     RECAL                   ; move disk to track zero
F    C381F8          JMP     SELDSK                  ; select disk drive
FC37DF9              JMP     SETTRK                  ; seek to track in reg A
F621 C392F9          JMP     SETSEC                  ; set sector number
F624 C3D9F9          JMP     SETDMA                  ; set disk DMA address
F627 C3E5F9          JMP     READBDF                 ; read selected sector
F62A C3ACFC          JMP     WRITEBUF                ; write selected sector
               ; these jumps added for absolute reading and writing
F62D C3DFF9          JMP     SETTC                   ; set terminal count
F630 C340FA          JMP     READ                    ; absolute read
F633 C332FA          JMP     WRITE                   ; absolute write
F636 C3C2F9          JMP     SECTOR                  ; absolute set sector
               ; note: when the absolute sector set jmp is called for a double sided
               ;      disk, use 1 through 52 as values, where the last 26 are
               :      auto selected for second side of disk
               .
               ; density table : stores modes of four drives
               ; 00=single density, 01=double density, 81=double sided, FF=undetermined
               ; this table must originate +39(HEX) relative to start of bios
               DENTBL:
F639 FF              DB      0FFH                    ; fdd 0
F63A FF              DB      0FFH                    ; fdd 1
F63B FF              DB      0FFH                    ; fdd 2
F63C FF              DB      0FFH                    ; fdd 3
               ;
               ;
               ; variable storage area
               ;
   ) 00        DENS    DB      0                     ; current density mode
F63E 0000      DMAADD  DW      0                     ; set initial DMA address (16 bits)
F640 0000      TRMCUT  DW      0                     ; terminal count storage (14 bits)
F642 FF        RECFL   DB      0FFH                  ; recalibrate flag (0 or 1)
F643 00        RTCNT   DB      0                     ; retry counter location
F644 00        UDSKNO  DB      0                     ; user disk number
F645 00        USECT   DB      0                     ; user sector number
F646 00        TEMP    DB      0                     ; temporary storage
F647 00        WRST    DB      0                     ; write buffer status flags
F648 00        RDST    DB      0                     ; read buffer  "  "
F649 0000      RWBPT   DW      0                     ; buffer pointer
F64B 00        RWFLAG  DB      0                     ; source of error byte
               ;
               ;
               ; read / write table:
               ; This table defines sector formatting to the uPD7&5
               ; and is sent to that device every read or write
               ;
               RWTBL:
               ;
F64C 00        DMACMD  DB      0                     ; dma command byte
F64D 00        FDCCMD  DB      0                     ; fdc current command
F64E 00        DSKN0   DB      0                     ; currently selected disk
F64F 00        TRKN0   DB      0                     ; currently selected track
F650 00        HEAD    DB      0                     ; head address (0 or 1)
F651 01        SECT    DB      1                     ; current record
F652 01        N       DB      1                     ; bytes per record code byte
F653 1A        EOT     DB      26                    ; end of track
```

```
CP/M MACRO ASSEM 2.0    #005    IMS double-density 8" CP/M bios

F655 0E         GPL     DB      OEH                     ; gap length
'(    F         DTL     DB      255                     ; data length
                ;
'656 00         WRCMD   DB      0                       ; current write command
F657 00         RDCMD   DB      O                       ; current read command
                ;
                ;
                DDVAL:                                  ; double-density values
                ; read/write values
?658 01                 DB      1
?659 1A                 DB      26
F65A 0E                 DB      0EH
F65B FF                 DB      255
F65C 45                 DB      DWRCMD                  ; write command
F65D 46                 DB      DRDCMD                  ; read command
F65E 347F040FF2         DB      52,127,4,15,242,192,2
F665 C392F9             JMP     SETSEC
                ;
                D2VAL:                                  ; double—density, double-sided
                ;
F668 01                 DB      1
F669 1A                 DB      26
F66A 0E                 DB      0EH
F66B FF                 DB      255
F66C C5                 DB      D2WCMD
F66D C6                 DB      D2RCMD
F66E68FF051FF2          DB      104,255,5,31,242,192,2
     C392F9             JMP     SETSEC
                ;
                SDVAL:                                  ; single-density values
                ;
F678 00                 DB      0
F679 1A                 DB      26
F67A 07                 DB      07H
F67B 80                 DB      128
F67C 05                 DB      SWRCMD
F67D 06                 DB      SRDCMD
F67E 1A3F0307F2         DB      26,63,3,7,242,192,2
F685 211AE9             LXI     H,BDOSB+1AH
                ;
                ;
                ; read/write result phase storage table
                ; This table is used to store the resulting status bytes
                ; from a disk read or write operation
                ;
                RWSTBL:
                ;
F688 00                 DB      0                       ; status reg-0
F689 00                 DB      0                       ; status reg-1
F68A 00                 DB      0                       ; status reg-2
F68B 00                 DB      0                       ; track
F68C 00                 DB      0                       ; head
F68D 00                 DB      0                       ; record
F68E 00                 DB      0                       ; bytes/sector code
                ;
                ;
```

```
                 ; Track table
                 ; stores head position of four drives
                 TRKTBL:
 8F 00                  DB      0                       ; fdd 0
 90 00                  DB      0                       ; fdd 1
 91 00                  DB      0                       ; fdd 2
 92 00                  DB      0                       ; fdd 3
                 ;
                 ;
                 ; Subroutines
                 ;
                 ; Check FDC ready bit 5 and fall into output ready routine when true
                 CMDRDY:
F693 F5                 PUSH    PSW                     ; save command
F694 DB8D        MON:   IN      ICS                     ; check for motor on
F696 17                 RAL
F697 DAAAF6             JC      BUSY                    ; yes, branch to FDC busy
F69A C5                 PUSH    3                       ; no, time out 1 second
F69B 010080             LXI     B,8000H
F69E E3          DELAY: XTHL                            ; some delay
F69F E3                 XTHL                            ; some more delay
F6A0 0B                 DCX     B                       ; cut counter
F6A1 78                 MOV     A,B                     ; check for zero
F6A2 B1                 ORA     C
F6A3 C29EF6             JNZ     DELAY                   ; loop if not
F6A6 C1                 POP     B
F6A7 C394F6             JMP     MON                     ; check motor status again
     DB8E        BUSY:  IN      FDCMSR                  ; get status
F6AC E620               ANI     20H                     ; mask ready bit
F6AE C2AAF6             JNZ     BUSY                    ; loop if busy
F631 F1                 POP     PSW                     ; restore command
                 ;
                 ; Check bits 7-6 for output ready condition
                 OUTRDY:
F6B2 F5                 PUSH    PSW                     ; save value
                 OUTRDY1:
F6B3 DB8E               IN      FDCMSR                  ; get status
F6B5 17                 RAL                             ; mask bits 7-6
F6B6 D2B3F6             JNC     OUTRDY1                 ; loop if not ready
F6B9 17                 RAL
F6BA DAD8F6             JC      FDCERR
F6BD F1                 POP     PSW                     ; restore output value
F6BE D38F               OUT     DDATA                   ; send it
F6C0 C9                 RET                             ; after sending
                 ;
                 ; Check bits 7-6 for input ready condition
                 INRDY:
F6C1 DB8E               IN      FDCMSR                  ; get status
F6C3 17                 RAL                             ; mask bits 7-6
F6C4 D2C1F6             JNC     INRDY                   ; loop if not ready
F6C7 17                 RAL
F6C8 D2D8F6             JNC     FDCERR
F6CB DB8F               IN      DDATA                   ; read value
F6CD C9                 RET                             ; with value in reg A
                 ;
                 ; check interrupt status of FDC
```

```
                   LNTRDY:
 F6CE  DB8D                IN        ICS                      ; read board status
   F6D01F                  RAR                                ; shift bit-0 into carry
   F6D1D2CEF6              JNC       INTRDY                   ; loop until interrupt
F6D4 AF                    XRA       A                        ; zero acc
 F6D5  D388                OUT       DMAST                    ; reset dma channels
 F6D7  C9                  RET
                   ;
                   FDCERR:
 F6D8  CDEBF6              CALL      TYPE
 F6DB  ODOA464443          DB        CR, LF, 'FDC  ERROR  $'
 F6E8  C3C5FB              JMP       ERRPRMPT
                   ;
                   ;messagetype , - utility
                   TYPE:
 F6EB  E3                  XTHL                               ; msg defined in memory after call
 F6EC  7E                  MOV       A,M                      ; get next char
 F6ED  23                  INX       H                        ; bump pointer
 F6EE  FE24                CPI       '$'                      ; stop on this
 F6FO  E3                  XTHL                               ; restore stack
 F6F1  C8                  RZ                                 ; return to next instruction
 F6F2  4F                  MOV       C,A                      ; char to reg C
 F6F3  CDECF7              CALL      CONOT                    ; print char
 F6F6  C3EBF6              JMP       TYPE                     ; loop
                   ;
                   ;
                   ;  BOOT
                   ;  This section is executed whenever the cold boot loader
                   ;  brings the entire system initialy from disk to ram
                   ;
F6F9  318000       BOOT:    LXI      SP,80H                   ; set stack pointer
                            *
                            IF       REMEX
                            MVI      A,SCYCMD                 ; get specify command
                            CALL     CMDRDY                   ; send it to fdc
                            MVI      A,SRT+HUT                ; step rate and head unload times
                            CALL     OUTRDY                   ; send them
                            MVI      A,HLDT+ND                ; head load time and non—dma
                            CALL     OUTRDY                   ; send them
                            ENDIF
                            *
                            IF       TTY40                    ; if teletype model 40
                            MVI      A,SET1 UART1             ; re-initialize port b to 9600
                            OUT      SETCNT
                            MVI      A,0DH
                            OUT      CNT1
                            MVI      A,0
                            OUT      CNT1
                            MVI      A,11H                    ; turn off request to send
                            OUT      MASK
                            ENDIF
                            *
                            »        PARPRN                   ; if parallel printer
                      IMVI           A.PMODE                  ; get mode control word
                      •OUT           PCNTR
                      •MVI           A 1                      ; set data strobe
```

```
                         OUT     PCNTR
                         MVI     A,8                    ; set-up IP
                         OUT     PCNTR
                         MVI     A,9
                         OUT     PCNTR
                         MVI     A,0CH                  ; disable interrupt
                         OUT     PCNTR
                         ENDIF
                         *
                         IF      DAISY                  ; if daisy wheel printer
                         MVI     A,SET1                 ; then set printer port for 1200 baud
                         OUT     SETCNT
                         MVI     A,68H
                         OUT     CNT1
                         MVI     A,0
                         OUT     CNT1
                         ENDIF
                         *
F6FC 2139F6              LXI     H.DENTBL               ; point to A in density table
F6FF 7E                  MOV     A,M
F700 CD97F8              CALL    CKDEN                  ; initialize density
F703 CD2EFD              CALL    SMSG                   ; print sign-on message
                 ;
                 GOCPM:
F706 213AF6              LXI     H,DENTBL+1             ; set B,C,&D to not mounted
F709 36FF                MVI     M,0FFH
F70B 23                  INX     H
F70C 36FF                MVI     M.0FFH
F70E 23                  INX     H
F70F 36FF                MVI     M.0FFH
F711 3EC3                MVI     A,0C3H                 ; put jump to warm boot
F713 320000              STA     0                      ; at address 0
F716 2103F6              LXI     H,WBOOTE
F719 220100              SHLD    1
F71C 320500              STA     5
F71F 2106E9              LXI     H,BDOS                 ; put jump to BDOS
F722 220600              SHLD    6                      ; at loc. 5,6&7
F725 218000              LXI     H,80H                  ; set default DMA address
F728 223EF6              SHLD    DMAADD
F72B 3A44F6              LDA     UDSKNO                 ; get current disk # to
F72E 4F                  MOV     C,A                    ; pass to CCP in reg C
F72F C300E1              JMP     CPMB                   ; start up CCP
                 ;
                 ;
                 ; WARM-BOOT:     read all of CPM back in
                 ; except bios, then re-enter CCP.
                 ;
F732 318000      WBOOT:  LXI     SP,80H                 ; set stack pointer
F735 3EFF                MVI     A,0FFH                 ; set drive A to not mounted
F737 3239F6              STA     DENTBL
F73A 3A44F6              LDA     UDSKNO                 ; load and save current disk #
F73D 3246F6              STA     TEMP
F740 0E00                MVI     C,0                    ; force drive A
F742 CD81F8              CALL    SELDSK
F745 CDD6FC              CALL    RECAL                  ; home using recal
F748 3E02                MVI     A,2
```

```
 F74A 3251F6             STA     SECT                    ; start at sector two
 F74D 2100E1             LXI     H,CPMB
    0 223EF6             SHLD    DMAADD                  ; set dma address
 F733 3A3DF6             LDA     DENS
 F756 B7                 ORA     A
 F757 CA7AF7             JZ      SDWBOOT                 ; decide what kind of warm boot
 F75A 21FF14             LXI     H,(NSECTS*128)-1        ; terminal count for ccp & bdos
 F75D 2240F6             SHLD    TRMCNT
 F760 CD40FA             CALL    READ                    ; bring it in
 F763 B7                 ORA     A                       ; set flags
 F764 C2B1F7             JNZ     BOOTERR                 ; error if non-zero
 F767 3A3DF6             LDA     DENS                    ; get current density
 F76A 17                 RAL                             ; check for two sides
 F76B DA74F7             JC      DSWB                    ; branch if true
 F76E CDBEF8             CALL    GODD                    ; force double-density on cp/m
 F771 C3A7F7             JMP     WBEND
 F774 CDB2F8     DSWB:   CALL    GOD2                    ; force double-sided on cp/m
 F777 C3A7F7             JMP     WBEND
                 SDWBOOT:
 F77A 217F0C             LXI     H, (25*128)-!           ; terminal count for track zero
 F77D 2240F6             SHLD    TRMCNT
 F780 CD40FA             CALL    READ                    ; bring it in
 F783 B7                 ORA     A                       ; set flags
 F784 C2B1F7             JNZ     BOOTERR                 ; error if non-zero
 F787 0E01               MVI     C,l
 F789 CD7DF9             CALL    SETTRK                  ; seek to track one
 F78C 3E01               MVI     A,l
 F778E 3251F6            STA     SECT                    ; set sector one
   F91 21 80 ED          LXI     H,CPMB+(25*128)         ; compute next dma address
 F794 223EF6             SHLD    DMAADD                  ; set it
 F797 217F08             LXI     H,((NSECTS-25)*128)-1   ; remaining system terminal count
 F79A 2240F6             SHLD    TRMCNT                  ; set it
 F79D CD40FA             CALL    READ                    ; bring it in
 F7A0 B7                 ORA     A                       ; set flags
 F7A1 C2B1F7             JNZ     BOOTERR                 ; error if non-zero
 F7A4 CDCAF8             CALL    GOSD                    ; force single-density on cp/m
                 WBEND:
 F7A7 3A46F6             LDA     TEMP                    ; restore disk #
 F7AA 4F                 MOV     C,A
 F7AB CD81F8             CALL    SELDSK                  ; reselect it
 F7AE C306F7             JMP     GOCPM                   ; go back to CP/M
                 ;
                 BOOTERR:
 F7B1 CDEBF6             CALL    TYPE
 F7B4 5741524D20         DB      'WARM BOOT ERROR$'
 F7C4 CDD3F7             CALL    CONIN                   ; read a char from console
 F7C7 C332F7             JMP     WBOOT                   ; try another warm boot
                 ;
                 ;
                 ; check console input status..
                 ;
 F7CA DB1JD      CONST:  IN      CSTAT                   ; read console status
 F7CC E60T               ANI     RDA                     ; look at kbd ready bit
 F7CE 3EO(?              MVI     A,0                     ; set A=0 for return
   DO C8                 RZ                              ; and return when not ready
 .7D1 2F                 CMA                             ; else set A=FF
```

```
F7D2 C9                  RET                          ; if console is ready
                ;
                ;
                ; read a character from console.
                ;
                CONIN:
                         *
                         IF      DELETE               ; if nice crt rubouts desired
F7D3 3A01F8              LDA     FSTPOS
F7D6 B7                  ORA     A
F7D7 C2E0F7              JNZ     CST
F7DA 3A05EB              LDA     BDOSB+205H
F7DD 3201F8              STA     FSTPOS
                         ENDIF
                         *
                         IF      TTY40
                         MVI     A,11H
                         OUT     MASK
                         ENDIF
                         *
F7E0 DB10       CST:     IN      CSTAT                ; read console status
F7E2 E601                ANI     RDA                  ; check kbd ready bit
F7E4 CAE0F7              JZ      CST                  ; and loop if not ready
F7E7 DB11                IN      CDATA                ; read character
F7E9 E67F                ANI     7FH                  ; mask to 7-bit Ascii
F7EB C9                  RET                          ; with char in A
                ;
                ; write a character to the console device.
                ;
                CONOT:
                         IF      DELETE               ; for crt rubouts
F7EC 3E0A                MVI     A,0AH
F7EE B9                  CMP     C
F7EF C2F6F7              JNZ     CONOT1
F7F2 AF                  XRA     A
F7F3 3201F8              STA     FSTPOS
                CONOT1:
                         ENDIF
F7F6 DB10       OST:     IN      CSTAT                ; read console status
F7F8 E602                ANI     TBE                  ; check transmitter ready bit
F7FA CAF6F7              JZ      OST                  ; and loop if busy
F7FD 79                  MOV     A,C                  ; get character to output
F7FE D311                OUT     CDATA                ; send it
F800 C9                  RET                          ; with outputted char in A&C
                ;
                         IF      DELETE
                ;
                ; this routine added to clean up bios rubouts
F801 00         FSTPOS   DB      0
                ;
                RUBOUT:
F802 E5                  PUSH    H
F803 C5                  PUSH    3
F804 CD4DF8              CALL    FNDPOS
F807 79                  MOV     A,C
F808 C1                  POP     3
```

```
F809  C5                    PUSH    B
F80A  05                    DCR     3
      C216F8                JNZ     RUBO
F80E  2101F8                LXI     H,FSTPOS
F811  4E                    MOV     C,M
F812  91                    SUB     C
F813  C31CF8                JMP     RUB1
F816  F5          RUBO:     PUSH    PSW
F817  CD4DF8                CALL    FNDPOS
F81A  Fl                    POP     PSW
F81B  91                    SUB     C
F81C  2105EB      RUB1:     LXI     H,BDOSB+205H
F81F  71                    MOV     M,C
F820  47                    MOV     B,A
F821  CD39F8                CALL    BCKUP
F824  Cl                    POP     B
F825  El                    POP     H
F826  05                    DCR     B
F827  2B                    DCX     H
F828  C9                    RET
                  ;
                  DELBUF:                                      ; control-U
F829  78                    MOV     A,B
F82A  B7                    ORA     A
F82B  C8                    R2
F82C  CD4DF8                CALL    FNDPOS
F82F  3A01F8                LDA     FSTPOS
F832  3205EB                STA     BDOSB+205H
F835  47                    MOV     B,A
F836  79                    MOV     A,C
F837  90                    SUB     B
F838  47                    MOV     B,A
                  BCKUP:
F839  OE08                  MVI     C,08H
F83B  CDECF7                CALL    CONOT
F83E  OE20                  MVI     C,20H
F840  CDECF7                CALL    CONOT
F843  0E08                  MVI     C,08H
F845  CDECF7                CALL    CONOT
F848  05                    DCR     B
F849  C8                    RZ
F84A  C339F8                JMP     BCKUP
                  ;
                  ENDPOS:
F84D  3A01F8                LDA     FSTPOS
F850  4F                    MOV     C,A
F851  2AOAE3                LHLD    BDOSB+20AH
F854  23                    LNX     H
F855  23                    INX     H
                  FNDP1:
F856  7E                    MOV     A,M
F857  FE09                  CPI     09H
F859  C265F8                JNZ     NOTAB
F85C  3EF8                  MVI     A,0F8H
F85E  Al                    ANA     C
F85F  C508                  ADI     01000B
```

```
F861 4F                  MOV     C,A
F862C36CF8               JMP     POSOUT
 F8653E20       NOTAB:    CPI     20H
F867D26BF8               JNC     CHAR
F86A 0C                  INR     C
F86B OC         CHAR:     INR     C
F86C  23        POSOUT:   INX     H
F86D  05                 DCR     B
F86E  C8                 RZ
F86F  C356F8             JMP     FNDP1
                         ENDIF
                ;
                ;
                ; Write a character on the listing device
                ;  this code is set to conditionally assemble printer drivers
                ; depending upon the true/false equates at the top
                ;
                LIST:
                          *
                          IF      PARPRN                  ; if parallel printer
                LST:     IN      PORTC                   ; get printer status
                          ANI     LPBSY
                          JZ      LST                     ; loop until ready
                          MOV     A,C
                          OUT     PORTA                   ; load port A with char
                          XRA     A
                          OUT     PCNTR                   ; raise strobe
                          INR     A
                          OUT     PCNTR                   ; lower strobe
                          RET
                          ENDIF
                          *
                          IF      SERPRN                  ; if normal serial mode
 F872 DB12      LST   :   IN      LSTAT                   ; get lister status
 F874  E602               ANI     TBE                     ; look at ready bit
 F876  CA72F8             JZ      LST                     ; and loop if not ready
 F879  79                 MOV     A,C                     ; get character
 F87A  D313               OUT     LDATA                   ; send it out
 F87C  C9                 RET                             ; return from list
                          ENDIF
                          *
                          IF      TTY40                   ; if teletype model 40
                          MVI     A,l
                          OUT     MASK                    ; raise request to send
                LST:      IN      LSTAT
                          ANI     CTS+TBE        ; check clear to send as well
                          XRI     TBE                     ; look for bit 7 low and 0 high
                          JNZ     LST
                          MOV     A,C
                          OUT     LDATA                   ; output char
                          RET
                          ENDIF
                          *
                          IE      DAISY                   ; if daisy wheel printer
                LST:      IN      LSTAT
                          ANI     CTS+TBE                 ; check clear to send
```

```
                        XRI     CTS+TBE                 ; look for bits 0&7 high
                        JNZ     LST
                        MOV     A,C
                        OUT     LDATA
                        RET
                        ENDIF
                ;
                ;
                ; Normally used to punch paper tape, but is not used now.
 F87D C9        PUNCH:  RET
                5
                ; Normally used to read from paper tape device,
                ; but is set up now to read console in standard system.
 F87E C3D3F7    READER: JMP     CONIN                   ; read from console.
                ;
                ;
                ; select disk number according to register c.
                ; value is buffered and the actual hardware select
                ; is not done until one of these disk accesses is made
                ; (read, write, recal, seek)
                SELDSK:
 F881 C5                PUSH    B
 F882 CD4AFC            CALL    WRSTAT                  ; flush write buffer
 F885 CDE8FB            CALL    FLGSOFF                 ; flags are off
 F888 Cl                POP     B
 F889 3E03              MVI     A,3                     ; mask to bits 0-1
 F88B A1                ANA     C
 F88C 3244F6            STA     UDSKNO                  ; store new drive
   3F 4F                MOV     C,A                     ; find density of new drive
 F890 0600              MVI     3,0
 F892 2139F6            LXI     H,DENTBL                ; get start of table
 F895 09                DAD     3                       ; add drive value
 F896 7E                MOV     A,M                     ; density in acc
                ; density check routine
                ; compare current density setting with setting of newly selected drive
                ; and perform switch on the fly if needed
                CKDEN:
 F897 F5                PUSH    PSW                     ; save density
 F898 010000            LXI     B,0
 F89B CDD4FB            CALL    SIDSEC                  ; force side zero
 F89E Fl                POP     PSW                     ; restore density
 F89F FEFF              CPI     OFFH
 F8A1 CAF3F8            JZ      FINDEN                  ; find density
 F8A4 213DF6            LXI     H, DENS                 ; point to current density
 F8A7 BE                CMP     M                       ; save as new?
 F8A8 C8                RZ                              ; yes, return
 F8A9 77                MOV     M,A                     ; update current density flag
 F8AA B7                ORA     A                       ; find which one
 F8AB CACAF8            JZ      GOSD                    ; if single density
 F8AE 17                RAL                             ; check double sided
 F8AF D2BEF8            JNC     GOOD                    ; no
                GOD2:
 F8B2 21FFOO            LXI     H, 255                  ; physical sector terminal count
 F8B5 2240F6            SHLD    TRMCNT
 F8B8 2168F6            LXI     H,D2VAL                 ; point to d squared table
 F8BB C3D3F8            JMP     DEN0
```

```
                GODD:
F8BE 21FF00             LXI     H,255           ; set double-density terminal count
F8C1 2240F6             SHLD    TRMCNT
F8C4 2158F6             LXI     H,DDVAL         ; point to dblden table
F8C7 C3D3F8             JMP     DEN0            ; patch bdos
                GOSD:
F8CA 217F00             LXI     H,127           ; set single-density terminal count
F8CD 2240F6             SHLD    TRMCNT
F8D0 2178F6             LXI     H,SDVAL         ; point to sinden table
                DEN0:
F8D3 1152F6             LXI     D,N             ; update read/write table
F8D6 0606               MVI     B,6             ; byte transfer count
F8D8 7E         DEN1:   MOV     A,M
F8D9 12                 STAX    D
F8DA 23                 INX     H
F8DB 13                 INX     D
F8DC 05                 DCR     B
F8DD C2D8F8             JNZ     DEN1
F8E0 113AE9             LXI     D,BDOSTBL       ; update bdos file parameter block
F8E3 0607               MVI     B,7             ; byte transfer count
F8E5 7E         DEN2:   MOV     A,M
F8E6 12                 STAX    D
F8E7 23                 INX     H
F8E8 13                 INX     D
F8E9 05                 DCR     B
F8EA C2E5F8             JNZ     DEN2
F8ED 110FE9             LXI     D,TRANS
F8F0 0603               MVI     B,3
F8F2 7E         DEN3:   MOV     A,M             ; set sector translation routine
F8F3 12                 STAX    D
F8F4 23                 INX     H
F8F5 13                 INX     D
F8F6 05                 DCR     B
F8F7 C2F2F8             JNZ     DEN3
F8FA C9                 RET

                5
                ;
                FINDEN:
F8FB E5                 PUSH    H
F8FC CD52F9             CALL    DSKSEL          ; select it
F8FF CDB2F8             CALL    GOD2            ; force double-sided
F902 3E04               MVI     A,SDSCMD        ; get drive status
F904 CD93F6             CALL    CMDRDY
F907 3A4EF6             LDA     DSKN0           ; from current disk
F90A CDB2F6             CALL    OUTRDY
F90D CDC1F6             CALL    INRDY
F910 E608               ANI     08H             ; mask two-side flag
F912 EE08               XRI     08H             ; if true
F914 0E81               MVI     C,81H
F916 CA31F9             JZ      DENFOOND        ; then call it double—sided
F919 3EC9               MVI     A,RET
F91B 329FFA             STA     ETRAP           ; defete error routines
F91E CDCAF8             CALL    GOSD            ; force single-density
F921 CD40F9             CALL    READID          ; try a read
F924 0E00               MVI     C,0             ; prep for leaving
F926 CA31F9             JZ      DENFOUND        ; if good read
```

```
 F929 CDBEF8             CALL    GODD                    ; force double-density
 F92C CD40F9             CALL    READ ID                 ; try a read
 F92F 0E01               MVI     C,l
                 DENFOUND:
 F931 El                 POP     H
 F932 3EC8               MVI     A,RZ
 F934 329FFA             STA     ETRAP                   ; enable error routines
 F937 79                 MOV     A,C
 F938 77                 MOV     M,A                     ; update table
 F939 323DF6             STA     DENS
 F93C C8                 RZ                              ; if valid value
 F93D 36FF               MVI     M,0FFH
 F93F C9                 RET
                 ;
                 ;
                 ; read sector ID field
                 ;
                 READ ID:
 F940 3A57F6             LDA     RDCMD                   ; get current read command
 F943 E6EO               ANI     0E0H                    ; mask instruction out
 F945 F60A               ORI     RIDCMD                  ; make read ID command
 F947 CD93F6             CALL    CMDRDY                  ; send it
 F94A 214EF6             LXI     H,DSKN0                 ; point into read/write table
 F94D 0601               MVI     B,l
 F94F C380FA             JMP     RWTOUT                  ; use read routines for status
                 ;
                 ;
                 ;
                 DSKSEL:
F952 3A44F6              LDA     UDSKNO                  ; get current drive
 F955 C5                 PUSH    3
 F956 47                 MOV     B, A                    ; put it back
 F957 3A4EF6             LDA     DSKN0                   ; get current disk #
 F95A E603               ANI     3                       ; mask it also
 F95C B8                 CMP     B                       ; old = new ?
 F95D Cl                 POP     B
 F95E C8                 RZ                              ; if so, forget it
 F95F 218FF6             LXI     H,TRKTBL                ; point HL at track table
 F962 E5                 PUSH    H                       ; save it
 F963 1600               MVI     D,0                     ; zero D for dad
 F965 5F                 MOV     E,A                     ; old disk number offset in table
 F966 19                 DAD     D                       ; point to it's track
 F967 3A4FF6             LDA     TRKN0                   ; get old track number
. F96A 77                MOV     M,A                     ; store it
 F96B El                 POP     H                       ; point HL again
 F96C 3A44F6             LDA     UDSKNO                  ; get new disk #
 F96F E603               ANI     3                       ; mask it
 F971 324EF6             STA     DSKN0                   ; store new disk
 F974 D38A               OUT     DSEL                    ; select on fdc board
 F976 5F                 MOV     E,A                     ; point at new track #
 F977 19                 DAD     D
 F978 7E                 MOV     A,M                     ; get it
 F979 324FF6             STA     TRKN0                   ; make it current
 -97C C9                 RET                             ; return from disk select
                 ;
                 ;
                 ; SET TRACK
                 ; set track number to whatever is in register c
                 ; also perform move to the correct track (seek).
                 ;
                 SETTRK:
 F97D CD52F9             CALL    DSKSEL
 F980 3A4FF6             LDA     TRKNO                 ; get present track #
 F983 B9                 CMP     C                     ; old = new?
 F984 C8                 RZ                            ; yes, so split
 F985 CD4AFC             CALL    WRSTAT                ; check write buffer
 F988 CDE8FB             CALL    FLGSOFF               ; reset all flags
 F98B 79                 MOV     A,C                   ; no, so get new track #
 F98C 324FF6             STA     TRKNO                 ; update track # in table
 F98F C3F2FC             JMP     SEEK                  ; move to new track #
                 ;
                 ;
                 ; Set disk sector number
                 ;
                 SETSEC:
 F992 3A3DF6             LDA     DENS                  ; check for single-density
 F995 B7                 ORA     A
 F996 C29EF9             JNZ     SEC1                  ; if not single-density
                 SEC:
 F999 79                 MOV     A,C
 F99A 3251F6             STA     SECT                  ; and store in absolute sector if true
 F99D C9                 RET
                 SEC1:
 F99E 1E1A               MVI     E,26                  ; max sector allowed
  F9A0 17                RAL                           ; check double sided
  F9A1 D2A6F9            JNC     SEC2                  ; no
 F9A4 1E34               MVI     E,52                  ; yes, interlace spec change
 F9A6 79         SEC2 :  MOV     A,C
 F9A7 B7                 ORA     A                     ; clear carry
 F9A8 1F                 RAR                           ; divide by two
 F9A9 F5                 PUSH    PSW                   ; save odd even status
 F9AA 4F                 MOV     C,A
 F9AB AF                 XRA     A                     ; start lace at zero
 F9AC 0D         FLACE : DCR     C                     ; sector minus one
 F9AD FABAF9             JM      FLACE2                ; done when less than zero
 F9B0 C609               ADI     9                     ; add lacing
 F9B2 BB                 CMP     E                     ; out of bounds?
 F9B3 DAACF9             JC      FLACE                 ; no, loop
 F9B6 93                 SUB     E                     ; adjust to max sectors
 F9B7 C3ACF9             JMP     FLACE                 ; loop
 F9BA 4F         FLACE2: MOV     C,A                   ; store laced sector in C
 F9BB Fl                 POP     PSW                   ; restore status
 F9BC 79                 MOV     A,C                   ; restore laced sector
 F9BD 17                 RAL                           ; mult by two with odd/even
 F9BE 3245F6             STA     USECT                 ; or buffered user sector if false
 F9C1 C9                 RET                           ; from set sector routine
                 ;
                 ; this routine added for direct sector accessing
                 SECTOR:
 F9C2 3A3DF6             LDA     DENS                  ; get current density
  9C5 17                 RAL                           ; check double-sided
  9C6 D299F9             JNC     SEC                   ; no, set sector directly
```

```
F9C979              MOV     A,C             ; get sector into ace
      600           MVI     B,0             ; assume head zero first
EAB                 CPI     27
F9CE DAD4FB         JC      SIDSEC          ; select side 0
F9D1 D61A           SUI     26              ; adjust for second side
F9D3 4F             MOV     C,A             ; update C with sector
F9D4 0601           MVI     B,l             ; set for head one
F9D6 C3D4FB         JMP     SIDSEC          ; select them
            ;
            ; Set disk dma address
F9D9 60     SETDMA: MOV     H,B             ; move BC to HL
F9DA 69             MOV     L,C
F9DB 223 EF6        SHLD    DMAADD          ; store new value
F9DE C9             RET                     ; from setdma
            ;
            ; set terminal count
F9DF 60     SETTC:  MOV     H,B             ; mov BC to HL
F9EO 69             MOV     L,C
F9E1 2240F6         SHLD    TRMCNT          ; store new value
F9E4 C9             RET                     ; from settc
            ;
            :
            ; double-density CP/M read routine which does internal buffering
            READBUF:
F9E5 CD52F9         CALL    DSKSEL          ; check for new disk
F9E8 3A3DF6         LDA     DENS            ; get current density
F9EB B7             ORA     A               ; check for single
     CA40FA         JZ      READ            ; and go directly to read if true
F9F0CDF0FB          CALL    BUFSTAT         ; check buffer status
F9F2 2148F6         LXI     H,RDST          ; point HL at flag
F9F5 47             MOV     B,A             ; save buffer half requested
F9F6 A6             ANA     M               ; is this half valid
F9F7 C21FFA         JNZ     RDBGD           ; branch if read buffer good
F9FA C5             PUSH    B               ; save request
F9FB CD4AFC         CALL    WRSTAT          ; check if write buffer needs flush
F9FE Cl             POP     B               ; restore request
F9FF CO             RNZ                     ; return if disk error
FA00 2148F6         LXI     H,RDST          ; point HL again
FA03 78             MOV     A,B             ; get back request
FA04 A6             ANA     M               ; is half good now
FA05 C21FFA         JNZ     RDBGD           ; yes, branch
FA08 2A3EF6         LHLD    DMAADD          ; no, read new sector
FA0B E5             PUSH    H               ; save user's dma address
FA0C 2180FE         LXI     H,RWBUFF        ; set our own
FA0F 223 EF6        SHLD    DMAADD
FA12 CD40FA         CALL    READ            ; fill disk buffer
FA15El              POP     H
FA16 223 EF6        SHLD    DMAADD          ; restore user's dma address
FA19 CO             RNZ                     ; bad read
FA1A 3E03           MVI     A,3
FA1C 3248F6         STA     RDST            ; read buffer good now
            RDBGD:
FA1F 2A3EF6         LHLD    DMAADD          ; get user's dma address
  22 EB             XCHG                    ;putintoDE
FA23 2A49F6         LHLD    RWBPT           ; get our buffer pointer
            DOWN:
```

```
FA26 7E                 MOV     A,M                     ; move 128 bytes to user
FA27  12                STAX    D                       ; beginning at dma address
  FA28  23              INX     H
FA29  13                INX     D
FA2A 3E7F               MVI     A,7FH
FA2C A5                 ANA     L
FA2D  C226FA            JNZ     DOWN
FA30 AF                 XRA     A                       ; set for good read
FA31 C9                 RET                             ; from read buffer
               ;
               ;
               ; write the sector based on the read/write table
               ;  transfer begins at dma address
               ;
               WRITE:
FA32 3EFF               MVI     A,0FFH
FA34 324BF6             STA     RWFLAG                  ; set for write error msg
FA37  3A56F6            LDA     WRCMD                   ; get current command
FA3A  67                MOV     H,A                     ; into H
FA3B  2E80              MVI     L,DMARD                 ; get dma command into L
FA3D  C34AFA            JMP     RDWR                    ; branch to read/write
               ;
               ;
               ; Read the sector based on the read/write table
               ; and transfer beginning at dma address
               ;
               READ:
FA40  AF                XRA     A
  FA41  324BF6          STA     RWFLAG                  ; set for read error msg
  FA44  3A57F6          LDA     RDCMD                   ; get current command
FA47  67                MOV     H, A                    ; into H
FA48  2E40              MVI     L,DMAWR                 ; get dma command into L
               ;
               ;
               ; perform read or write command
               RDWR:
FA4A  224CF6            SHLD    RWTBL                   ; store at beginning of table
FA4D  3EFF              MVI     A,0FFH                  ; reset recal flag
FA4F  3242F6            STA     RECFL                   ; for track retry
FA52  3E05              MVI     A,RTRY                  ; get retry max
FA54  3243F6            STA     RTCNT                   ; store into counter
               RETRY:
FA57  214CF6            LXI     H, RWTBL                ; point to read/write table
FASA  7E                MOV     A,M                     ; get dma mode
FA5B  23                INX     H                       ; bump pointer
FA5C  E5                PUSH    H                       ; save pointer
FA5D  F5                PUSH    PSW                     ; save dma mode
FA5E  AF                XRA     A                       ; reset low/high flip flop
FA5F  D388              OUT     DMAST
FA61  2A40F6            LHLD    TRMCNT
FA64 7D                 MOV     A,L                     ; send terminal count
FA65  D383              OUT     CH1TC                   ; low order first
FA67  Fl                POP     PSW
^4.68  B4               ORA     H                       ; add in dma mode
  169  D383             OUT     CH1TC                   ; high order second
FA6B  2A3EF6            LHLD    DMAADD
```

```
FA6E 7D                 MOV     A,L                   ; send dma address
     D382               OUT     CH1DMA                ; low order first
     7C                 MOV     A,H
FA72 D382               OUT     CH1DMA                ; high order second
FA74 3E42               MVI     A,CH1ENA
FA76 D388               OUT     DM AST                ; enable dma device
FA78 El                 POP     H                     ; restore pointer
FA79 7E                 MOV     A,M                   ; get command
FA7A 23                 INX     H                     ; bump pointer
FA7B CL93F6             CALL    CMDRDY                ; send command
FA7E 0608               MVI     B,8
               RWTOUT:                                ; send rest of table
FA80 7E                 MOV     A,M
FA81 CDB2F6             CALL    OUTRDY                ; send byte to fdc
FA84 23                 INX     H                     ; bump pointer
FA85 05                 OCR     B
FA86 C280FA             JNZ     RWTOUT                ; loop for 8 bytes
FA89 CDCEF6             CALL    INTRDY                ; wait for interrupt
FA8C 2188F6             LXI     H,RWSTBL               point to result table
FA8F E5                 PUSH    H                     ; save pointer
FA90 0607               MVI     B,7
               RWTIN:                                 ; read status back
FA92 CDC1F6             CALL    INRLY                 ; get a byte from fdc
FA95 77                 MOV     M,A                   ; store it
FA96 23                 INX     H                     ; bump pointer
FA97 05                 OCR     B
FA98 C292FA             JNZ     RWTIN                 ; loop for 7 bytes
FA9B El                 POP     H                     ; restore pointer
FA9C 7E                 MOV     A,M                   ; check for errors
7A9D E6CO               ANI     0C0H                  ; are either bits 6 or 7 set?
FA9F C8        ETRAP:   RZ                            ; no, return with zero set
FAA0 1157FA             LXI     D,RETRY               ; yes', fall into error handler
FAA3 D5                 PUSH    D                     ; prep stack for return
               ;
               :
               ; follow through with error handling routines
               RWERR:
FAA4 7E                 MOV     A,M                   ; get back status-0
FAA5 E618               ANI     18H                   ; check equip & ready stat
FAA7 C2BOFB             JNZ     DRVNRDY               ; branch to drive not ready
FAAA 23                 INX     H
FAAB 23                 INX     H
FAAC 7E                 MOV     A,M                   ; get status-2
FAAD E610               ANI     10H                   ; check for wrong cylinder
FAAF CACAFA             JZ      RWERR1                ; do common error handling
               ; check to see if a recalibrate has been done
FAB2 3A42F6             IDA     RECFL                 ; get recal flag
FAB5 3C                 INR     A                     ; see if it's true
FAB6 C2CAFA             JNZ     RWERR1                ; Sorry, only 1 retry, skip recal
FAB9 3242F6             STA     RECFL                 ; set flag false
               ; do error recalibration on selected drive
FABC 3A4FF6             LDA     TRKNO
FABF F5                 PUSH    PSW                   ; save requested track
FAC0 CDD6FC             CALL    REGAL                 ; issue recal on drive
FAC3 Fl                 POP     PSW
FAC4 324FF6             STA     TRKNO                 ; restore requested track
```

```
 FAC7 C3F2FC            JMP     SEEK                ; with caller's return first on stack
                   ;
                   ; do common error routines
                   RWERR1:
 FACA  3A43F6           LDA     RTCNT               ; get retry counter
 FACD  3D               DCR     A                   ; do we have another chance
 FACE  3243F6           STA     RTCNT               ; store new value
 FAD1  F0               RP
 FAD2  Dl               POP     D                   ; bring return off stack
                   KWERR1$2*
 FAD3  2189F6           LXI     H,RWSTBL+1
 FAD6  3A4BF6           LDA     RWFLAG              ; who had the error?
 FAD9  B7               ORA     A
 FADA  CAECFA           JZ      RDERR               ; if read
 FADD  CDEBF6           CALL    TYPE                ; must have been write
 FAE0  0D0A575249       DB      CR, LF, 'WRITE  $'
 FAE9  C3F7FA           JMP     RWERR2
 FAEC  CDEBF6     RDERR:  CALL    TYPE
 FAEF  0D0A524541       DB      CR, LF, 'READ   $'
 FAF7  7E        RWERR2:  MOV     A,M               ; get back st-1
 FAF8  E680             ANI     80H
 FAFA  CA11FB           JZ      ERR3
 FAFD  CDEBF6           CALL    TYPE                ; if end of cylinder error
 FB00  454E44204F       DB      'END  OF  CYLINDER  $'
 FB11  7E          ERR3:  MOV     A,M               ; get back st-1
 FB12  E620             ANI     20H
 FB14  CA40FB           JZ      ERR4
 FB17 23                INX     H                   ; if crc error
   FB18  7E             MOV     A,M                 ; check for data or ID field
FB19  23                DCX     H
 FB1A  E620             ANI     20H
 FB1C  CA2FFB           JZ      IDCRC               ; if crc in ID field
 FB1F  CDEBF6           CALL    TYPE                ; if crc in data field
 FB22  4441544120       DB      'DATA  CRC  $'
 FB2C  C340FB           JMP     ERR4
 FB2F  CDEBF6     IDCRC:  CALL    TYPE              ; if crc in id field
 FB32  4944204649       DB      'ID  FIELD  CRC  $'
 FB40  7E          ERR4:  MOV     A,M               ; get back st-1
 FB41  E604             ANI     04H
 FB43  CA5BFB           JZ      ERR5
 FB46  CDEBF6           CALL    TYPE                ; if no data error
 FB49  534543544F       DB      'SECTOR  NOT  FOUND  $'
 FB5B  7E          ERR5:  MOV     A,M               ; get st-1
 FB5C  E602             ANI     02H
 F35E  CA6DFB           JZ      ERR6
 FB61  CDEBF6           CALL    TYPE                ; if write protect error
 FB64  50524F5445       DB      'PROTECT  $'
 FB6D  7E          ERR6:  MOV     A,M               ; get st-1
 FB6E  E601             ANI     01H
 FB70  CAA4FB           JZ      ERR7
 FB73  23               INX     H                   ; if missing address mark
 FB74  7E               MOV     A,M                 ; get  st-2
 FB75  E601             ANI     01H                 ; check for data or ID field
 FB77  CA86F3           JZ      IDAM                ; if ID error
   FB7A  CDEBF6         CALL    TYPE                ; if data mark error
FB7D  4441544120        DB      'DATA  $'
```

```
 FB83 C393F3             JMP     IDERR
 EB86 CDEBF6    IDAM:    CALL    TYPE
      4944204649         DB      'ID FIELD $'
EB89CDEBF6      IDERR:   CALL    TYPE
 FB96 4144445245         DB      'ADDRESS MARK $'
 FBA4 CDEBF6    ERR7 :   CALL    TYPE
 FBA7 4552524F52         DB      'ERROR$'
 FBAD AF                 XRA     A
 FBAE 3C                 INR     A                    ; set for permanent error
 FBAF C9                 RET                          ; from read/write routine
                ;
                ;
                ; Drive not ready error branches here
                ;
                DRVNRDY:
 FBB0 CDEBF6             CALL    TYPE
 FBB3 ODOA445249         DB      CR,LF,'DRIVE NOT READY $'
                ERRPRMPT:
 FBC5 CDD3F7             CALL    CON IN               ; wait for response
 FBC8 FED3               CPI     'C'-40H              ; if control-c
 FBCA CA26FD             JZ      PERMERR              ; branch out of error
 FBCD CDEBF6             CALL    TYPE                 ; do cr & If
 FBDO ODOA24             DB      CR,LF,'$'
 FBD3 C9                 RET
                ;
                9
                ;
                ; select side and sector routine
                ; reg b contains side (0 or 1), reg c contains sector (1 to 26)
                SIDSEC:
FBD4 E5                  PUSH    H
 FBD5 2151F6             LXI     H, SECT              ; point at sector in table
 FBD8 71                 MOV     M,C                  ; set sector in rwtbl
 FBD9 2B                 DCX     H                    ; point to head in table
 FBDA 70                 MOV     M,B                  ; store head
 F3DB 78                 MOV     A,B
 FBDC 07                 RLC
 FBDD 07                 RLC                          ; shift into bit 2
 FBDE 47                 MOV     B,A
 FBDF 2B                 DCX     H
 FBE0 2B                 DCX     H                    ; point to dskno
 FBE1 3E03               MVI     A,3
 FBE3 A6                 ANA     M                    ; get only disk number
 FBE4 BO                 ORA     B                  . ; combine with head
 FBE5 77                 MOV     M,A                  ; store new value
 F3E6 El                 POP     H
 FBE7 C9                 RET
                ;
                ; turn flags off
                FLGSOFF:
 FBE8 AF                 XRA     A
 FBE9 3247F6             STA     WRST                 ; reset write status flag
 F3EC 3248F6             STA     RDST                 ; reset read status flag
 FBEF C9                 RET                          ; from flagsoff
                ;
                ;
                ; check buffer status for read or write
```

```
                 BUFSTAT:
      45F6                LDA     USECT            ; which user's sector
 F3 4F                    MOV     C,A              ; store in C
 F4 0600                  MVI     B,0              ; assume head zero for now
 F6 3A3DF6                LDA     DENS             ; get current density
 F9 17                    RAL                      ; check double sided
 FA D208FC                JNC     GETSEC           ; if not
 FD 79                    MOV     A,C              ; get back usect
 FE FE34                  CPI     52               ; greater than side zero
C00 DA08FC                JC      GETSEC           ; no, branch
C03 D634                  SUI     52               ; adjust
C05 4F                    MOV     C,A              ; update C
C06 0601                  MVI     B,1              ; set head one
                 GETSEC:
C08 79                    MOV     A,C              ; get physical sector now
C09 3246F6                STA     TEMP             ; save it for now
C0C E6FE                  ANI     0FEH             ; make it even
C0E 0F                    RRC                      ; divide by two
C0F 3C                    INR     A                ; find physical sector
FC10 4F                   MOV     C,A              ; save in c
FC11 2151F6               LXI     H,SECT           ; point at current sector
FC14 BE                   CMP     M                ; are they the same?
FC15 C21EFC               JNZ     BDWRBF           ; no, bad buffer
FC18 2B                   DCX     H                ; check head address
FC19 78                   MOV     A,B
FC1A BE                   CMP     M                ; are they the same?
FC1B CA2EFC               JZ      BUFST1           ; yes, buffer valid
                 BDWRBF:
FC1E C5                   PUSH    B
FC1F CD4AFC               CALL    WRSTAT           ; user has moved sectors
FC22 C1                   POP     B
FC23 CA28FC               JZ      GDRDWR           ; good clean up
FC26 E1                   POP     H                ; bad return
FC27 C9                   RET
                 GDRDWR:
FC28 CDD4FB               CALL    SIDSEC           ; set sector and side
FC2B CDE8FB               CALL    FLGSOFF          ; reset flags
                 BUFST1:
FC2E 0600                 MVI     B,0              ; zero B for dad
FC30 3A46F6               LDA     TEMP             ; get back usect
FC33 0F                   RRC                      ; odd or even?
FC34 DA3EFC               JC      SETB2            ; if odd
FC37 3E01                 MVI     A,1              ; first half flag
FC39 0E00                 MVI     C,0              ; prep C for dad
FC3B C342FC               JMP     SETBPT
FC3E 3E02        SETB2:   MVI     A,2              ; second half flag
FC40 0E80                 MVI     C,80H            ; prep C for dad
FC42 2180FE      SETBPT:  LXI     H,RWBUFF         ; point to beginning of buffer
FC45 09                   DAD     B                ; add our address
FC46 2249F6               SHLD    RWBPT            ; store away
FC49 C9                   RET                      ; from bufstat routine
                 ;
                 ;
                 WRSTAT:                           ; check write buffer status
FC4A 3A47F6               LDA     WRST             ; get flag
FC4D B7                   ORA     A                ; if flag is zero
```

```
 C4E C8            R2                      ; do nothing
           CLWRB:
      FA3EF6       LHLD    DMAADD          ; save user's dma address
 FC52 E5           PUSH    H               ; on the stack
 FC53 FE03         CPI     3               ; check for both halves valid
 FC55 CA96FC       JZ      WRBGD           ; yes, simple flush
 FC58 0F           RRC                     ; test bit 0
 FC59 DA7AFC       JC      FIX2            ; update second part ram buffer
 FC5C 2180FE       LXI     H,RWBUFF        ; update first part ram buffer
 FC5F 223EF6       SHLD    DMAADD          ; set dma address
 FC62 217F00       LXI     H,127           ; terminal count for half sector
 FC65 2240F6       SHLD    TRMCNT          ; set it
 FC68 CD40FA       CALL    READ            ; bring it in
 FC6B C29FFC       JNZ     CLWRBO          ; error if non-zero
 FC6E 21FF00       LXI     H,255           ; terminal count for full sector
 FC71 2240F6       SHLD    TRMCNT          ; set it
 FC74 CD32FA       CALL    WRITE           ; update disk sector
 FC77 C39FFC       JMP     CLWRBO          ; exit wrstat
           FIX2:
 FC7A 2100FF       LXI     H,RWBUFF+80H    ; update second part of ram buffer
 FC7D 223EF6       SHLD    DMAADD          ; set dma address
 FC80 CD40FA       CALL    READ            ; bring in old sector
 FC83 C29FFC       JNZ     CLWRBO          ; error if non-zero
 FC86 2100FF       LXI     H.RWBUFF+8OH    ; point HL
 FC89 1180FF       LXI     D,RWBUFF+100H   ; point DE
           FIX20:
 FC8C 1A           LDAX    D               ; transfer old half in
    FC8D 77        MOV     M,A
 FC8E 23           INX     H
 FC8F 13           INX     D
 FC90 3E7F         MVI     A,7FH
 FC92 A5           ANA     L
 FC93 C28CFC       JNZ     FIX20           ; loop for 128 bytes
           WRBGD:
 FC96 2130FE       LXI     H,RWBUFF        ; get buffer address
 FC99 223EF6       SHLD    DMAADD          ; set dma address
 FC9C CD32FA       CALL    WRITE           ; update it              -
           CLWRB0:
 FC9F 2147F6       LXI     H,WRST          ; point to flags
 FCA2 3600         MVI     M,0             ; reset write status flag
 FCA4 23           INX     H
 FCA5 3603         MVI     M,3             ; set read status flag
 FCA7 El           POP     H               ; restore user's dma address
 FCA8 223EF6       SHLD    DMAADD
 FCAB C9           RET                     ; from write status
           ;
           ;
           ; write routine for CP/M (buffered writing)
           WRITEBUF:
 FCAC CD52F9       CALL    DSKSEL          ; new disk
 FCAF 3A3DF6       LDA     DENS            ; get current density
 FCB2 B7           0BA     A               ; check for single density
 FCB3 CA32FA       JZ      WRITE           ; and go directly to write if true
   FCB6 CDFOFB     CALL    BUFSTAT         ; check buffer status
   FCB9 2147F6     LXI     H.WRST          ; point to write flag
 FCBC 47           MOV     B,A             ; save request
```

```
FCBD B6                 ORA     M                     ; update write flag
FCBE 77                 MOV     M,A
 'BF 23                 INX     H                     ; point to read flag
  CO 78                 MOV     A,B                   ; get back request
TCC1 B6                 ORA     M                     ; update read flag
FCC2 77                 MOV     M,A
FCC3 2A3EF6             LHLD    DMAADD                ; get user's dma address
FCC6 EB                 XCHG                          ; into DE
FCC7 2A49F6             LHLD    RWBPT                 ; get our buffer pointer
               UP:
FCCA 1A                 LDAX    D                     ; get record from user
FCCB 77                 MOV     M,A
FCCC 23                 INX     H
FCCD 13                 INX     D
FCCE 3E7F               MVI     A,7FH
FCDO A5                 ANA     L
FCD1 C2CAFC             JNZ     UP                    ; loop for 128 bytes
FCD4 AF                 XRA     A                     ; set for good return
FCD5 C9                 RET                           : from write buffer routine
               ;
               ;
               ;
               ; Recalibrate selected drive
               ;
               RECAL:
               ;
FCD6 CD52F9             CALL    DSRSEL                ; check for new disk
FCD9 CD4AFC             CALL    WRSTAT                ; check write buffer
 "DC CDE8FB             CALL    FLGSOFF               ; reset flags
 .DF 3E07               MVI     A,RECCMD              ; get command
FCE1 CD93F6             CALL    CMDRDY                ; send it
FCE4 3A4EF6             LDA     DSOO                  ; get current drive from table
FCE7 CDB2F6             CALL    OUTRDY                ; send it
FCEA CD06FD             CALL    SIS                   ; wait for completion
FCED AF                 XRA     A
FCEE 324FF6             STA     TROO
FCF1 C9                 RET                           ; from recal
               ;
               ;
               ; Perform seek on the current drive to the current sector
               ; specified by the read/write table.
               ;
               SEEK:
               ;
FCF2 3E0F               MVI     A,SKCMD               ; get seek command
FCF4 CD93F6             CALL    CMDRDY                ; send it
FCF7 3A4EF6             LDA     DSKNO                 ; get current disk
FCFA CDB2F6             CALL    ODTRDY                ; send it
FCFD 3A4FF6             LDA     TRKNO                 ; get current track #
FD0O CDB2F6             CALL    OUTRDY                ; send new track #
FD03 C306FD             JMP     SIS
               ;
               ;
               ; sense interrupt status from fdc
               SIS:
 J06 CDCEF6             CALL    INTRDY                ; wait for interrupt
 FD09 3E08      SIS1 :   MVI     A.SISCMD               ; command
 FD0B  CD93F6            CALL    CMDRDY                 ; send it
  ~»0E  CDC1F6           CALL    INRDY                  ; read st-0
   11 E6C0               ANI     OCOH                   ; mask error bits
-2D13 FE80               GPI     80H
 FD15 C8                 RZ                             ; return if no interrupts pending
 FD16 47                 MOV     B,A                    ; save it
 FD17 CDC1F6             CALL    INRDY                  ; read pen
 FD1A 78                 MOV     A,B
 FD1B  E6CO              ANI     OCOH                   ; check for errors
 FD1D  C8                RZ                             ; none, good
 FD1E  FECO              CPI     OCOH                   ; ready change status
 FD20  CA09FD            JZ      SIS1                   ; do another sis
 FD23  C330FB            JMP     DRVNRDY                : drive must not be there
                ;
                ;
                ; permanent disk errors 1branch here
                PERMERR::
 FD26  OE00              MVI     C,0                    ; force disk zero
 FD28  CD81F8            CALL    SELDSK                 ; for warm boot
 FD2B  C332F7            JMP     WBOOT                  ; do it
                ;
                ;
                SMSG:
 FD2E  CDEBF6            CALL    TYPE
 FD31  ODOAOA496E        DB      CR,LF,LLF,'Industrial Micro Systems',CR,LF
 FD4E  3634              DB      MSIZE/10-+'0 1MSIZE MOD 10 + '0'
 FD50  4B2043502F        DB      'K CP/M /V1.4 of 79oct01',CR,LF
  •^68  446F75626C       DB      'Double-*density',CR,LF,'$'
   79 3A3DF6             LDA     DENS                   → Frode/[illegible]
 TD7C 17                 RAL
 FD7D  D292FD            JNC     SMSG1
 FD80  CDEBF6            CALL    TYPE
 FD83  446F75626C        DB      'Double-:sided',CR,LF,'$'
                SMSG1:
 FD92  CDEBF6            CALL    TYPE
 FD95  5379737465        DB      1System 8000 Version',CR,LF,'$'
 FDAB  C9                RET
                ;
                ;
 FDAC                    END
```

```
F839 BCKTJP      E900 BDOSB       E906 BDOS        E93A BDOSTBL     FC1E BDWRBF
F600 BIOS        F6F9 BOOT        F7B1 BOOTERR     FC2E 3UFST1      FBFO BUFSTAT
   A BUSY        B800 CBASE       0010 CCOM        0011 CDATA       0082 CH1DMA
   2 CH1ENA      0083 CH1TC       F86B CHAR        F897 CKDEN       FC4F CLWRB
FC9F CLWRBO      F693 CMDRDY      0014 CNTO        0015 CNT1        0016 CNT2
F7TD3 CONIN      F7EC CONOT       F7F6 CONOT1      F7CA CONST       El00 CPMB
1500 CPML        OOOD CR          0010 CSTAT       F7EO CST         0080 CTS
OOC6 D2RCMD      F668 D2VAL       OOC5 D2WCMD      0000 DAISY       008F DDATA
F658 DDVAL       F69E DELAY       F829 DELBUF      FFFF DELETE      F8D3 DENO
F8D8 DEN1        F8E5 DEN2        F8F2 DEN3        F931 DENFOUND    F63D DENS
F639 DENTBL      F63E DMAADD      F64C DMACMD      0080 DMARD       0088 DMAST
0040 DMAWR       FA26 DOWN        0046 DRDCMD      FBBO DRVNRDY     008A DSEL
0080 DSKB        F64E DSKNO       F952 DSKSEL      F774 DSWB        F655 DTL
0045 DWRCMD      F653 EOT         FB11 ERR3        FB40 ERR4        FB5B ERRS
FB6D ERR6        FBA4 ERR7        FBC5 ERRPRMPT    FA9F ETRAP       0000 FALSE
F64D FDCCMD      F6D8 FDCERR      008E FDCMSR      F8FB FINDEN      FC7A FIX2
FC8C FIX20       F9AC FLACE       F9BA FLACE2      FBE8 FLGSOFF     F856 FNDP1
F84D FNDPOS      F801 FSTPOS      FC28 GDRDWR      FC08 GETSEC      F706 GOCPM
F8B2 GOD2        F8BE GOOD        F8CA GOSD        F654 GPL         F650 HEAD
0024 HLDT        000F HUT         008D ICS         FB86 IDAM        FB2F IDCRC
FB93 IDERR       F6C1 INRDY       F6CE INTRDY      0010 IOB         0012 LCOM
0013 LDATA       000A LF          F872 LIST        0080 LPBSY       0012 LSTAT
F872 LST         0018 MASK        F694 MON         0040 MSIZE       0000 ND
F652 N           F865 NOTAB       002A N SECTS     F7F6 OST         F6B2 OUTRDY
F6B3 OUTRDY1     0000 PARPRN      00 IF PCNTR      FD26 PERMERR     OOA2 PMODE
001C PORTA       00 ID PORT3      00 IE PORTC      F86C POSOUT      F87D PUNCH
0001 RDA         FA1F RDBGD       F657 RDCMD       FAEC RDERR       F648 RDST
FA4A RDWR        F9E5 READBUF     FA40 READ        F87E READER      F940 READ ID
  16 REGAL       0007 RECCMD      F642 RECFL       0000 REMEX       FAS7 RETRY
  JA RIDCMD      F643 RTCNT       0019 RTCRES      0005 RTRY        F816 RUBO
F81C RUB1        F802 RUB OUT     F649 RWBPT       FE80 RWBUFF      FAA4 RWERR
FACA RWERR1      FAD3 RWERR12     FAF7 RWERR2      F64B RWFLAG      F688 RWST3L
F64C RWT3L       FA92 RWTIN       FA80 RWTOUT      0003 SCYCMD      0004 3DSCMD
F678 SDVAL       F77A SDWBOOT     F99E SEC1        F999 SEC         F9A6 SEC2
F651 SECT        F9C2 SECTOR      FCF2 SEEK        F881 SELDSK      FFFF SERPRN
0036 SETO        0076 SET1        OOB6 SET2        FC3E SETB2       FC42 SETBPT
0017 SETCNT      F9D9 SETDMA      F992 SETSEC      F9DF SETTC       F97D SETTRK
FBD4 SIDSEC      FD06 SIS         FD09 SIS1        0008 SISCMD      OOOF SKCMD
FD2E SMSG        FD92 SMSG1       0006 SRDCMD      OOCO SRT         0005 SWRCMD
0002 TBE         F646 TEMP        E90F TRANS       F64F TRKNO       F68F TRKTBL
F640 TRMCNT      FFFF TRUE        0000 TTY40       F6EB TYPE        F644
FCCA UP          F645 USECT       F7A7 WBEND       F603 WBOOTE      F732 WBOOT
FC96 WRBGD       F656 WRCMD       FCAC WRITEBUF    FA32 WRITE       F647 WRST
FC4A WRSTAT
```