Owner's Manual

Model 2422

Multimode Floppy Disk Controller

REV A

California Computer Systems

CCS MODEL 2422

FLOPPY DISK CONTROLLER

OWNER'S MANUAL

CALIFORNIA COMPUTER SYSTEMS
250 CARIBBEAN DRIVE
SUNNYVALE, CA 94086

MANUAL NO. 89000-02422

TABLE OF CONTENTS

TABLE OF CONTENTS CONTINUED

FIGURES

TABLES

CHAPTER 1

INTRODUCTION

California Computer System's 2422 Floppy Disk Controller is an S-100 compatible board which can control four drives in any combination of 5.25" and 8" two-sided or single-sided drives. It can read and write both single-density (FM) and double-density (MFM) soft-sectored diskettes. An on-board 2K ROM contains monitor firmware designed for systems using CCS's 2810 Z-80 CPU and bootstrap loader for loading in CP/M from disk.

1.1 SYSTEM AND DRIVE BUS COMPATIBILITY

The 2422 is compatible with the IEEE proposed standards for the S-100 bus, thus making it bus compatible with most of the S-100 systems currently on the market. The drive busses are designed primarily to be plug-compatible with Shugart 800/850 and 400/450 drives and any drives using the same drive bus. See Appendix A for the signals used by the system and disk drive busses.

1.2 DISKETTE COMPATIBILITY

Western Digital's FD1793 disk controller chip used by the 2422 reads and writes diskettes which conform to the IBM 3740 format for single-density diskettes or the IBM System 34 format for double-density diskettes and which contain 128, 512, 256, or 1024 bytes per sector. Because of its format requirements, it cannot read diskettes formatted by the 1771 disk controller chip, although the 1771 can read diskettes formatted by the 1793.

1.3 USER OPTIONS

The 2422 incorporates a number of optional features that are jumper-selected or configured. These are described fully in Chapter Two. In addition, CCS offers a separately available address decoding ROM that can be inserted in U21 to allow memory mapping of the 2422 registers to addresses FFF8h-FFFDh.

1.4 THE BOOT/MONITOR ROM

The on-board ROM contains a bootstrap loader for loading CP/M into memory from disk and the MOSS 2.2 Disk Monitor. Both the bootstrap loader and the MOSS monitor take advantage of the Z-80's large instruction set and thus cannot cannot be used with an 8080 CPU. The MOSS monitor is designed to work specifically with CCS's Model 2810 Z-80 CPU; it uses the 2810's serial port for console interface. You can configure the 2422 so that it either boots in CP/M automatically or brings up the monitor when you turn your system on or reset it. When the ROM is enabled and addressed, the ROM LED lights and the 2422 generates the PHANTOM signal which allows memory overlay of all devices which share the ROM's memory space and recognize the PHANTOM signal. The ROM can be software-disabled once CP/M is loaded, the loader program, CCBOOT, supplied with CCS's CP/M package doing so automatically.

Those of you who want to alter the firmware or design your own will find programming information on the 2422 in Appendix B.

CHAPTER 2

SETUP AND INSTALLATION

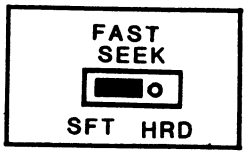
2.1 2422 BOARD SETUP

Your 2422 Floppy Disk Controller is designed to be flexible enough to fit different systems and different customer needs. This flexibility is achieved in part by on-board jumpers which allow you to selectively enable/ disable features or configure them to fit your needs. Each jumper-enabled or configured feature is discussed in a section below. In Appendix D, you will find a board layout on which the jumpers are clearly indicated, labeled, and referenced to the appropriate section below.

2.1.1 FAST-SEEK SELECT JUMPER

If you are not using a voice coil drive, remove the FAST SEEK jumper plug entirely. If you are using a voice coil drive, the FAST SEEK jumper allows you to either software-enable or hardware-enable the fast seek mode. Placing the jumper plug at the SFT position as shown in Figure 2-1 allows you to enable fast seeks by writing a 0 to bit 4 of Control Register 2. Placing the jumper plug at the HRD postion permanently enables fast seeks. If you are planning to use the CCS firmware/software supplied with the 2422, fast seeks will be enabled only if you set the jumper to the HRD position, since the CCS software does not enable the fast seek mode.

FIGURE 2-1



2.1.2 BANK-BYTE SELECT JUMPERS

The 2422 Disk Controller can be hardware assigned to one of eight banks, or levels of 64K. To do so, place the jumper plug on the BANK BYTE jumper header which corresponds to the bank level to which you want this board assigned. For example, putting the plug on header D0 assigns this board to bank 0. Once you have assigned this board to a bank, you can in turn select that bank and enable the board by outputting to port 40 a data byte with a logic 1 in the bit position corresponding to the bank level. If you plan to use the version of CP/M supplied by CCS, assign the board to bank 0. Its loader program, CCBOOT, disables the monitor and bootstrap loader firmware after it loaded into memory from disk by outputting a hex 01 to port 40h. If you have your 2422 assigned to any other bank than bank 0, your board will be disabled.

2.1.3 MEMORY-MAP ENABLE JUMPER

As mentioned in the introduction, CCS makes available to its 2422 users a control ROM which allows the registers on the 2422 to be memory mapped. If you plan to use the memory map option, you can enable/disable memory mapping through the M MAP jumper. Position OFF of the M MAP jumper disables memory mapping; the opposite position enables it. The CCS firmware does not make use of memory mapping.

2.1.4 PARTIAL ROM ENABLE JUMPER

By setting the PR EN jumper to ON, you allow the portion of the ROM containing the basic I/O routines and the primitive disk routines used by the monitor to be available after CP/M is loaded in. This portion of the ROM, located at F600h-F7FFh, contains essentially the same basic I/O routines as CCS's customized BIOS, CCBIOS, on the distribution diskette. If you are planning to tailor the CCBIOS to your system, you may wish to have your customized BIOS call some of the routines located in the ROM. This will give you the greater reliability of ROM memory and save some disk space. If you do not wish the basic I/O portion of the ROM to remain in memory after CP/M is loaded in, set the PR EN jumper to OFF.

2.1.5 ROM WAIT STATE ENABLE JUMPER

The on-board ROM has the relatively slow memory access time of 450 nsecs. A CPU running at 4 MHz will not provide the access time needed by the ROM. The access time of the 1793 registers when they are memory mapped is also slow, about 350 nsecs. The ROM WAIT jumper, when set to ON, allows you to place the CPU in one Wait state per cycle in which either the ROM or the 1793 is selected. Some CPUs, such as CCS's Model 2810 Z-80 CPU, use pin 98 of the bus to indicate whether the CPU is operating at 2 or 4 MHz. If your CPU does so, setting the jumper to 4 MHZ allows the 2422 to place the CPU in a Wait state only when the CPU is operating at 4 MHz. OFF completely disables the ROM Wait circuitry.

2.1.6 BANK ENABLE JUMPER

The BANK EN jumper allows you three options in using the bank-select system to enable the board. Position ON makes the bank-select system fully operable, so that to enable the board, you must software-select the bank this board resides in by outputting the correct data byte to port 40h. Postion RST disqualifies the bank select system on power-on and system reset, allowing the board to be enabled after you turn on or reset your system without its bank being software-selected first. Otherwise, the bank-select system functions as usual. Position OFF completely disqualifies the bank-select system, permanently enabling the board. Note that in the last two cases, the Bank LED will be lit when the board is enabled through disqualifying the bank-select system. If you are using the version of CP/M supplied by CCS, set BANK EN to RST or OFF, since the board must be enabled on reset. If you are operating with one disk controller in your system, the OFF position is probably the wiser choice since it eliminates the possibility of accidently disabling the board.

2.1.7 BOOT ENABLE JUMPER

The BOOT EN jumper allows you to choose between three methods of enabling/ disabling the bootstrap loader and monitor firmware. If you set the BOOT EN jumper to OFF, neither the bootstrap loader or the monitor firmware can be accessed. If you have set the PR EN jumper to OFF, the entire ROM will then be disabled. If you set the BOOT EN jump to position A, the

bootstrap loader and monitor are enabled when your system is turned on or reset and disabled when any data byte is output to port 40h. Because port 40h is the Bank Select Port as well, you must also have the BANK EN jumper set to RST or OFF so that the board is enabled on reset or power on. If you are planning to use the CCS version of CP/M supplied with your board, set the BOOT EN jumper to position A. This allows the bootstrap loader and monitor to be disabled automatically when CP/M is loaded. Position B of the BOOT EN jumper allows the bootstrap loader and monitor to be enabled/disabled entirely through software control. Writing a 0 to bit 1 of Control Register 2 enables them; a 1 disables them.

2.1.8 AUTO BOOT ENABLE JUMPER

If you are using the ROM-resident firmware, this jumper allows you to choose whether or not CP/M will be loaded or the monitor entered on power-on and reset. If you set this jumper to OFF, the monitor will be entered on power-on and reset. CP/M can then be loaded in under monitor control by use of the Boot command. For those who plan to use the 2422 in a system with a 2810 Z-80 CPU and wish to use the initialization firmware provided for the on-board port, the AUTO BOOT jumper should be set to OFF. This allows you to synchronize the baud rate of the 2810's port to the baud rate of your console device by hitting the carriage return key two to three times. This brings up the monitor, allowing CP/M to be booted in at will. If you set the AUTO BOOT jumper to ON, CP/M will be booted in on power-on and reset. Since only the bootstrap loader portion of the ROM will be accessed, this setting frees the user of the constraint of using the 2422 in a system with a 2810 Z-80 CPU. However, the user must then provide his own console initialization routine in the BIOS.

2.1.9 AUTO WAIT SELECT JUMPER

The disk drive cannot read or write data on the disk as fast as the CPU can send or receive data. Thus there are times when the CPU must wait for the data register in the disk controller chip to become ready to receive a data byte from or transmit a data byte to the CPU. The state of the DRQ (Data Request) line from the 1793 indicates whether or not the data register is ready for data transfer. If it is low, the data register is not ready. If you set the AUTO WAIT jumper to DATA, you can force the CPU into a Wait state every time the CPU tries to read or

write to the data register when the DRQ line is low. By setting it to STAT1, you force the CPU into a Wait state every time it tries to read Board Status Register 1 when the DRQ line is low. In both cases, the CPU will remain in a Wait state until the DRQ line goes high again. Both forms of the Auto Wait are also enabled/ disabled through software control. Writing a 1 to bit 7 of Control Register 1 enables Auto Waits; a 0 to bit 7 disables them.

2.1.10 INTERRUPT JUMPERS

The interrupt jumpers allow you to tie DRQ and/or INTRQ to either the Interrupt line, pINT, the Nonmaskable Interrupt line, NMI, or any of the 8 Vectored Interrupt lines, VI0-VI7. INTRQ, when active, indicates that a command has been completed and that the 1793 is awaiting a new command. DRQ, when active, indicates that the data buffer either has a byte to be read or is empty and requires a new byte to transmit, depending on the nature of the disk operation in progress. Either or both of these lines can be used to generate interrupts and thus request servicing from the processor. To generate a Vectored Interrupt 2 by the active INTRQ, for example, run a bus wire from the INTRQ pad to the VI2 pad and solder it in.

2.2 SYSTEM SETUP FOR FIRMWARE COMPATIBILITY

In order for the bootstrap loader and monitor firmware to work as described in Chapter 3, you must have a power-on jump circuit somewhere in your Z-80 system set to force the CPU to jump to location F000h when you turn your system on or reset it. Any RAM sharing the ROM's memory space must be disabled while the firmware is being accessed. If your RAM board accepts the PHANTOM signal output by the 2422 when the ROM is selected, the RAM will automatically be disabled. On CCS memory boards, this entails jumper-enabling the PHANTOM signal. If your RAM board uses the same bank select system as the 2422, you also can configure your board so that the memory block sharing memory space with the 2422 ROM is assigned to bank 0 and disabled on power-on or reset. When the Loader program from disk is loaded, it outputs a hex 01 to port 40h. This disables the bootstrap and monitor firmware as it enables the RAM. Please note that if you use this method you must have at least 256 bytes of low RAM memory enabled on reset; to be safe it would be wise to enable all RAM except that which directly conflicts with the ROM. For example, if you own CCS's Model 2065 64K Dynamic RAM board, you

would assign the board to bank 0 and configure it to be bank-disabled on reset, with the first 16K blocks bank-independent and the last 16K block bank-dependent. Note that if you want the basic I/O portion of the ROM enabled after CP/M is loaded, you will have to use the PHANTOM line to disable the RAM sharing its memory space. Also if you are running a 61K or larger CP/M system and are using CCS's 2810 CPU board, you must disable the 2810's ROM. Once the 2422's ROM is disabled, the board no longer generates the PHANTOM signal and thus cannot disable the 2810's ROM.

The monitor and basic I/O routines require some additional set up. They are designed to work in a system with a Model 2810 Z-80 CPU configured as follows: SER EN and JMP EN set ON, MON EN set OFF, and SER ADDRESS SELECT set to 20h. You must also set the 2/4 MHZ switch to 4 MHz if you plan to be reading or writing double-density diskettes; the firmware design will not allow double-density diskettes to be read or written when the the CPU is operating at 2 MHz. In addition, your terminal must be set as described in section 2.2.2 of the 2810 Owner's Manual.

2.3 INSTALLATION

Because we can not anticipate what drive or drives you will be using the 2422 board with, we can not give specific installation instructions. However, there are some general instructions we can give. If you plan to use more than one drive, you must make sure that the common lines are terminated in the last drive on the cable only. This may mean removing jumper plugs or resistor packs: see your manual. You must also enable the appropriate Drive Select line to each drive, usually accomplished by moving a jumper plug. Some signals, such as TWO-SIDED, may also require some user-configuration to be enabled. If your drive allows, you can minimize diskette wear by configuring your drive so that you can select it without loading the Read/Write head.

The cable assemblies needed to connect the 2422 with your drives are not not supplied with the 2422. For the 5.25" drives and the 8" drives you need 34 and 50 conducter flat-ribbon cables, respectively. The connectors you need are as follows:

Mating Connectors for the 2422:

5.25" drives (J1) = Ansley #609-3430 or equivalent
8" drives (J2) = Ansley #609-5030 or equivalent

Back Panel Connectors:

5.25" drives = Ansley #609-3416 or equivalent
8" drives = Ansley #609-5016 or equivalent

Mating Connectors for Back Panel:

5.25" drives = Ansley #609-3430 or equivalent
8" drives = Ansley #609-5030 or equivalent

Mating Connectors to the Drive P. C. Board:

5.25" drives = Ansley #609-5015M or equivalent
8" drives = Ansley #609-3415M or equivalent

If you assemble your own cables, be sure that the pin 1 strip of the cable (usually marked by an outside colored stripe) matches pin 1 of all the connectors. Owners of Shugart, Memorex, and other bus compatible drives can simply install the assembled cables and connect them, being careful to match pin 1s. Owners of the Per Sci drives will have to do some rewiring, since the Per Sci drive bus differs from the 2422. Figure A-2 in Appendix A shows the pinouts for J1 and J2.

CHAPTER 3

THE 2422 ROM-RESIDENT FIRMWARE

3.1 COLD-START ENTRY

The firmware cold-start entry point is F000h. If you set a power-on jump circuit to this address, the CPU will jump to the cold-start entry point when your system is turned on or reset. The cold-start initialization routine loads the low RAM locations called to by the Z-80 restart commands with jump vectors to the restart error message. It then finds the highest continuous active RAM address and locates the monitor stack and work space below it. Next it checks the state of the Auto Boot bit in the board's Status Register 1; if the Auto boot bit is 0 the initialization routine passes control to the bootstrap loader, which then loads in CP/M as described section 3.4 below. The monitor work space is overwritten as CP/M is loaded in. If the Auto Boot bit is 1, the initialization routine continues, initializing the 2810 Z-80 CPU's serial port. When it has synchronized the serial port's baud rate to the console's baud rate, it turns control over to the monitor executive.

3.2 PAGE 0 RAM USED BY FIRMWARE

The following locations in page 0 of system memory are used by the the firmware. Except where noted, these locations should be reserved.

Address	Contents
0000h-0002h	These locations contain the warm-start vector for the monitor. When CP/M is loaded, they are overwritten by the warm-start vector for CP/M.
0003h	This location contains the Intel Standard IOBYTE loaded during cold-start initialization and used for monitor I/O (see section 3.5.2).
0008h-000Ah 0010h-0012h 0018h-001Ah 0020h-0022h 0028h-002Ah 0030h-0032h 0038h-003Ah	During cold-start initialization these locations called by the Z-80 restart commands are loaded with jump vectors to the restart error message (see section 3.5.5.) They can be overwritten by valid restart routines. In addition, locations 0008h-000Ah are used for software breakpoint processing by the monitor GO command.
0040h-0053h	These locations contain disk parameters used by the bootstrap loader and the monitor. They are described in more detail in section 3.3.3. Locations 0040-004Fh are defined by CP/M as user scratchpad locations; 0050-0053h are unused by present versions of CP/M but are held reserved.
0080h-017Fh	Temporary buffer used by the bootstrap loader and CP/M to store the Loader program from disk.

TABLE 3-1 LOW RAM LOCATIONS USED BY FIRMWARE

3.3 THE FIRMWARE DISK ROUTINES

The primitive disk routines used by the monitor and the bootstrap loader are designed to read or write disks which conform to the IBM 3740 and System 34 standards for soft-sectored diskette format. Although strictly speaking these standard apply to 8" diskettes only, they can be adapted for 5.25" diskettes. Since the firmware routines are designed for diskettes conforming to the IBM format standards, it might be helpful if we discuss diskette format in general and the IBM standards in particular.

3.3.1 DISKETTE FORMAT

Track numbering on a diskette begins at its circumference with Track 00 and proceeds toward the center; thus the innermost track on an 8" diskette with the standard 77 tracks is track 76. Each track on side 0 of a two-sided diskette has an associated track on side 1; these track-pairs are often called cylinders. Unlike track numbering, sector numbering starts with 1, the number given to the first sector immediately following the index pulse. The number of sectors on a track is dependent on disk size, data density, and number of bytes per sector.

The IBM 3740 standard for single-density diskettes allows sector sizes of 128, 256, and 512 bytes; the System 34 standard for double-density diskettes allow sectors sizes of 256, 512, and 1024 bytes. (The 1793 can format single-density diskettes in 1024-byte sectors and double-density diskettes in 128-byte sectors as well, but those additional sector sizes have no practical advantage.) Before each sector is an unique address or ID field identifying the track number, diskette side, sector number, and number of sectors per track. In addition, the ID fields and data fields must be separated by gaps and sync fields of a minimum length per sector. Figure B-1 of Appendix B illustrates the IBM 3740 format standard for single-density 8" diskettes. The 1793 adds an additional constraint in diskette format: it expects gaps to consist of minimum number of FFh bytes, followed by several bytes of 00h. Diskettes formatted by a 1771 disk controller chip do not meet the 1793's requirements. Thus the 1793 cannot read such diskettes. (The 1771 can, however, read disks formatted by the 1793.)

3.3.2 DESCRIPTION OF THE DISK ROUTINES

The firmware contains two routines for sector reads and writes: DREAD and DWRITE. The bootstrap loader calls DREAD for reading the first two sectors of Track 00; the monitor Read and Write commands use both routines. DREAD and DWRITE both transfer one sector at a time and automatically determine disk size, sector size, and density format if the disk has not been accessed before. They conform to the CP/M calling conventions and return a 0 in the A register if the disk operation was successful and a non-zero if it was not successful after ten tries. Both routines reside in the upper 1/2K of ROM which can remain enabled after CP/M is loaded in. Thus they can be called to from a user's BIOS. The entry point for DREAD is F6EAh; for DWRITE, F6EBh.

3.3.3 DISK PARAMETERS FOR DISK OPERATIONS

DREAD and DWRITE use locations 0040h-0053h to store the disk parameters they need. Below are the definitions and addresses of some of the more important disk parameters:

Address	Name	Description
0040h	DISKNO	Stores the number of the currently-selected drive: 0, 1, 2, or 3.
0041h	TRACK	Stores the number of the current track.
0042h	SECTOR	Stores the number of the current sector.
0043h	SIDE	Stores the byte written to Control Register 2 to select disk side. (90h = side 0; D0h = side 1)
0045h	TWOSID	Stores 0 if the disk in the currently-selected drive is one-sided; 1 if it is two-sided.
004Ah	CUNIT	Stores the byte last written to Control Register 1, giving information on the currently-selected drive unit.
004Ch	HSTBUF	Stores the starting address in memory for disk transfers to and from memory.
004Eh-0053h	IDSV	Stores the ID field information from the diskette in the current drive.

TABLE 3-2 DISK PARAMETERS

3.4 THE BOOTSTRAP LOADER

The bootstrap loader, when entered at F55Eh, reads in at locations 80h through 17F the contents of the first two sectors of track 00, side 0 of the disk in drive A and then transfers control to them. These sectors should contain a loader program for loading the system tracks (tracks 00 and 01 in an 8" diskette; tracks 00, 01, and 03 in a 5¼" diskette) into memory and transferring control to CP/M. In addition, Track 00 of this disk must be formatted in 128-byte single-density sectors. If the bootstrap loader encounters an error, it jumps to the Disk Error routine in the monitor portion of the ROM. If you were booting CP/M in from the monitor so that the 2810 CPU's serial port had been initialized, you will receive the Disk Error message as described in section 3.5.5 and control will be returned to the monitor. If you were booting in CP/M directly on system power-on or reset, your system will "hang." When it is finished reading in the Loader program, the bootstrap loader leaves some disk parameters in memory:

NAME	VALUE	
DISKNO	0	
SIDE	0	
TRACK	00	
SECTOR	3	
CUNIT	21	for a single-density 5.25" diskette
	31	for a single-density 8" diskette
	61	for a double-density 5.25" diskette
	71	for a double-density 8" diskette
IDSV + 3	00	if diskette sector size is 128
	01	if diskette sector size is 256
	02	if diskette sector size is 512
	03	if diskette sector size is 1024

TABLE 3-3 PARAMETER VALUES AFTER BOOT

After it is loaded, the Loader program in the disk supplied by CCS outputs hex 01 to port 40h. If the BOOT EN jumper has been set to postion A, this simultaneously disables the bootstrap and monitor firmware and enables any RAM assigned to bank 0 and with a bank select port of 40h.

3.5 THE MONITOR

CCS's MOSS 2.2 Disk Monitor is designed to allow you to control a system using a 2810 Z-80 CPU from the console keyboard. It allows you to display a block of memory in hex and ASCII, to move, change, and verify memory, and to transfer control to another program in memory with breakpoints set. You can also output or input a data byte to or from any I/O port and command the monitor to read and write floppy disks.

For the MOSS 2.2 Monitor to work exactly as described below, your 2422 Disk Controller board and 2810 Z-80 CPU must be configured as described in sections 2.1 and 2.2.

3.5.1 THE MONITOR'S MEMORY SPACE

In addition to the memory the ROM occupies and the page 0 addresses specified in section 3.2, the monitor requires some high RAM locations for the system stack and temporary storage area. The monitor scans the available memory until it finds the highest active RAM address and then counts down 56 bytes to store the breakpoints, registers, and register restore routine. It locates the system stack below that: you should reserve at least 88 bytes of high RAM memory for the monitor's use.

3.5.2 THE IOBYTE AND THE BASIC I/O ROUTINES

The monitor's basic I/O routines are essentially the same as those used by CCBIOS. They are designed for a system using CCS's 2810 Z-80 CPU, configured as described in section 2.2. As with the primitive disk routines, they reside in the last 1/2K of the ROM, allowing them to be available after CP/M is loaded, should you wish them to be. Section 3.5.2.3 below contains information on tailoring this portion of the ROM to fit your system's needs, if your are using a system with a different CPU or wish to provide driver routines for other peripherals. Please note that the method of initializing the console interface's baud rate described in the sections on bringing up the monitor and in the Y command is highly dependent on the hardware configuration of the 2810's serial port. Thus we cannot guarantee that the description of the monitor's operation in these sections will be valid if you alter the firmware to work with a different console interface.

3.5.2.1 The IOBYTE

The basic I/O routines in this portion of the ROM implement the IOBYTE function, as developed in the Intel MDS system and as used by CP/M. The IOBYTE function allows the user to assign a physical device to one or more of four logical I/O categories: Console, List, Punch, and Reader.. The current physical to logical device assignments are stored in the IOBYTE at location 0003h. The IOBYTE can be altered through the MOSS monitor Assign command or the CP/M STAT command. When an I/O routine, such as Console Input, is called, the routine loads the IOBYTE, using it to determine the currently assigned physical device, and then jumps to the driver routine called by the physical device assignment. For the allowable physical assignments in each logical category, see the Assign Command, section 3.5.4.1. In each logical category, the firmware provides only the Teletype assignment with driver routines. These routines are designed to drive the serial port on the 2810 CPU. Please note that the physical assignment names do not have to accurately describe the actual peripheral used; the actual physical device driven by the teletype assignment routines could easily be a CRT. For all physical device assignments other than the teletype, the I/O routines jump to location F462h, the location of the monitor I/O error message, resulting in the I/O Error message being displayed and control returned to the monitor as described in section 3.5.5.

3.5.2.2 The Basic I/O Routines

The user may call the following basic I/O routines from his own programs while in the monitor or from his own customized BIOS if the PR EN jumper is set ON.

Name	Address	Description
CI	F646	Console Input
*CONI	F68F	Console Input, strips ASCII parity bit
*CO	F600	Console Output
*CSTS	F623	Console Status Input
*LO	F610	List Output
*LSTAT	F669	List Status Input
*RI	F656	Paper Tape Reader Input
*PO	F67C	Papar Tape Punch Output
PRTWA	F698	Prints ASCII string on console which is terminated by bit 7 set in last character.
PRTWD	F695	Same as above, but does carriage return, line feed first.
CRLF	F6A9	Generates carriage return, line feed sequence to start new line on console

TABLE 3-4 THE BASIC I/O ROUTINES

The starred routines are CP/M compatible routines, basically the the same as the following routines used in the CCBIOS: CONIN, CONOUT, CONST, LIST, LISTST, READER, and PUNCH. They perform the basic IOBYTE handling as described above. Again, actual driver routines exist only for the teletype assignment for each logical category. These driver routines conform to the CP/M calling conventions, passing the data in the C register for any output and in the A register for any input. PRTWA, PRTWD, and CRLF are not routines used by a CP/M BIOS, however they are useful routines which are available as long as the Basic I/O portion of the ROM is accessible. CI is an alternative console input routine which does not strip the parity bit.

3.5.2.3 Customizing the Basic I/O Routines

As mentioned before, the monitor's basic I/O routines are designed to drive the console interface on the 2810 Z-80 CPU. Should you wish to add drivers for other peripheral devices or to use another console interface, you will have to alter the ROM firmware. There are two ways to do so. You can reprogram the ROM so that the jump instruction associated with a particular physical device assignment forces the CPU to jump to your

peripheral's driver routine. For example, to add a line printer to your system, you would change the jump instructions at locations F61D and F676 so that they contained the beginning address of your printer driver routines. Or you can, if you have CCS's 2810 Z-80 CPU and peripheral cards, use memory overlay techniques. Since the 2422 board generates, but does not receive, the PHANTOM signal, its ROM has to be moved to the CPU board. There the jump vectors called by the physical assignments can be overlaid with new jump vectors by the peripheral board's ROM.

3.5.3 BRINGING UP THE MONITOR

To enter the monitor, turn your system on or reset it. This results automatically in a cold-start entry into the monitor. Set your terminal to the baud rate at which you wish to operate. You have a choice of any baud rate between 2 and 56K baud. To allow the 2810 CPU's serial port to be initialized to the baud rate, hit the carriage return key until the monitor responds with

MOSS VERS 2.2

The maximum number of carriage returns needed before the monitor responds is three. When the monitor prompt appears, you may start entering commands.

3.5.4 MONITOR COMMANDS

The MOSS Monitor commands must conform to a specific format. The general form is

-CE1 E2 E3

where - is the prompt, C is the command character and E1-E3 are the address and data entries, if any. The essential parts of a command are as follows:

THE COMMAND CHARACTER: The monitor is controlled by one-character commands entered from the keyboard in response to the monitor prompt, a dash (-). No space is allowed between the prompt and the command character.

ADDRESS AND DATA ENTRIES: The general form for an address is a four digit hex number; for data, a two digit hex number. Leading zeros need not be entered; the monitor will supply them. No

space is allowed between the command character and the first address or data entry. Subsequent entries must be separated by a delimiter. The monitor looks at only the last four address characters or last two data characters before a delimiter. So if you make a mistake while typing an entry, keep typing until the last two or four characters are correct.

DELIMITERS: The MOSS Monitor recognizes three delimiters: a carriage return [CR], a space, or a comma. A carriage return indicates to the monitor that the current command is complete and should be executed. Either a space or a comma can mark the end of an address or data entry. In our command examples we will generally use a space as a delimiter, unless a comma makes the command form clearer. Please note, however, that you can use the space and the comma interchangeably. In certain commands a space or a comma can also be interchanged with a carriage return. These are commands for which the Monitor expects a fixed number of entries (and hence delimiters) following the command character.

SAMPLE COMMAND

The following commands to display the block of memory 0FFBh to 100Ah are all equivalent. Although the spacing is not free-form, some variety in the command form is allowed. Note that the display command requires two and only two address parameters, so that the last delimiter can be a comma or a space as well as a carriage return.

```
-D0FFB 100A[CR]
-DFFB,100A,
-DFFB,100A[CR]
-DFFB 100A[space]
-D0EF0FFB,100A[space]
```

3.5.5 ERROR MESSAGES

The MOSS monitor detects four types of error conditions and responds with a different error message for each. They are as follows:

COMMAND ERROR: Should you make an invalid entry, the command will be aborted, a warm boot of the system will occur, and the error message

????

will be printed, followed by the monitor prompt.

I/O ASSIGNMENT ERROR: As described in section 3.3, the Assign command allows you to assign a physical device to a logical peripheral category. When an I/O routine involving the logical category is called, the CPU will jump to the driver routine indicated by the physical assignment. If there is no driver routine, it will jump instead to the I/O Assignment Error routine. This routine sets the IOBYTE to its default value, outputs the error message

I/O ERR

and does a warm boot of the system.

RESTART ERROR: During cold-start initialization, jump-vectors to a restart error message are loaded in the memory locations called by the Z-80 restart instructions. This is done to prevent a jump to a restart address without code. A restart error causes the message

RST ERR

to be displayed and a warm boot of the system to occur.

DISK ERROR: The monitor, when executing the Read, Write, or Boot commands, will output the following error message and status information if it is unable to execute the command:

DSK ERR U XX T XX S XX C XX E XX

The first three hex bytes identify which physical record the monitor was unable to read or write. U gives the unit or drive number (0-3), T the track number and S the sector number of the record where the error occurred. C and E give the operation status at the time of the error. They reflect the contents of two of the 1793's internal registers: C shows the last command loaded in the Command register; E gives the contents of the Status register.

3.5.6 COMMAND DESCRIPTION

3.5.6.1 Assign (A)

The Assign command allows you to change the physical-to-logical device assignments and thus choose the peripherals you wish to work with while in the monitor. The IOBYTE function as developed by Intel for the MDS systems divides peripherals into four logical categories: Console, typically a teletype or a CRT; Reader, a paper tape reading device; Punch, a paper tape punching device; and List, a hard-copy printing device. Each of the four logical categories may have one of four physical devices assigned to them. The possible physical-to-logical assignments are as follows:

```
(C) Console
    (T) Teletype
    (C) CRT
     B) Batch Mode (input from logical reader device;
                    output to logical list device)
    (1) User Console #1

(R) Reader
    (T) Teletype
    (P) Paper tape reader
    (1) User reader #1
    (2) User reader #2

(P) Punch
    (T) Teletype
    (P) High speed paper tape punch
    (1) User punch #1
    (2) User punch #2

(L) List
    (T) Teletype
    (L) High speed line printer (CRT in CP/M)
    (1) User list #1 (High speed line printer in CP/M)
    (2) User list #2 (User list # 1 in CP/M)
```

To assign a physical device to a logical device category, enter

```
-AX
```

where X equals either C,R,P, or L, the logical device codes. If you enter a character other than these four, the computer will return with ???? and another prompt. If you enter a valid logical device code, the computer will return immediately with a

prompt for the physical device code. Enter

-Y

where Y equals the physical device code. Should you enter a delimiter only or a nonvalid device code, the device assignment will remain the same.

EXAMPLE:

Entering

-AR-P

assigns a high speed paper tape reader to the Reader logical device category.

Since the firmware contains driver routines only for the teletype assignment, you should receive the I/O error message if you attempt I/O operations with any other physical device without having altered the firmware first.

3.5.6.2 Boot (B)

The Boot command allows you to load in CP/M from disk under console control. Entering

-B

causes the bootstrap loader to load CP/M in from the disk in drive A and control to be transferred from the monitor to CP/M. When CP/M is loaded, the CP/M sign on message will appear, followed by the CP/M prompt. Should the bootstrap loader be unable to read in the first two sectors on Track 00, it will respond with the Disk Error message.

3.5.6.3 Display (D)

This command allows you to display the contents of a specified block of memory. The general form for the command is

-DA1 A2

where A1 and A2 are the first and last bytes, respectively, of the memory block.

The resulting display divides the memory into 16 bytes per line. Each line begins with the starting address of the 16 byte block, followed by the hex contents and their ASCII equivalents. The contents of addresses with the same last hex digit are aligned in vertical columns. Periods represent data for which there are no ASCII equivalents. As the display fills the screen, it automatically scrolls up. To freeze the display, type a control-S. To start it again, hit any key on the keyboard. Should you wish to escape from the display mode, hitting any key on the keyboard will abort the routine and return control to the monitor.

EXAMPLE

Entering

DF453 F4C8

results in the following display:

```
F453                  E1 08 D9 D1 C1 F1 E1 F9 00 21 00 00 C3     a.YQAqay.!..C
F460  00 00 AF 32 03 00 21 6C F4 C3 B5 F6 49 2F 4F 20   ../2..!ltC5vI/O
F470  45 52 D2 44 53 4B 20 45 52 52 3A 20 55 AD 20 54   ERRDSK ERR: U- T
F480  AD 20 53 AD 20 43 AD 20 45 AD 0D 8A 3F 3F 3F BF   - S- C- E-..????
F490  4D 4F 53 53 20 56 45 52 53 20 32 2E 32 0D 8A 3E   MOSS VERS 2.2..>
F4A0  0F D3 24 11 40 00 62 6A DB 26 A3 28 FB DB 26 23   .S$.@.bj[&#([[&#
F4B0  A3 A3 C2 AD F4 E5 29 5C 19 19 E5 29 29 DB 20 2B   ##B-te)\..e))[ +
F4C0  7D B4 C2 BD F4 E1 3E 83 D3                        }4B=ta>.S
```

3.5.6.4 Fill (F)

The fill command allows you to fill a block of memory with a specified constant. The general command form is

-FA1 A2 C

where A1 and A2 are the addresses of the first and last bytes of the memory block and C is the constant in hexidecimal.

EXAMPLE

Entering

-F10AA 10BB 1

fills the memory block 10AAh to 10BBh with the constant 1.

3.5.6.5 Goto (G)

The G command allows you to transfer control from the monitor to another program. It allows you to specify the entry address and to set up to two breakpoints for returning control to the monitor. When the monitor encounters a breakpoint, it saves the contents of the Z-80 registers in the system's temporary storage and outputs to the console device an asterisk followed by the address after the break. It then returns the prompt. You can use the Examine Register command (X) at this time to examine or change the saved registers.

The general form for the G command is

-GA B1 B2

where A is the entry address, and B1 and B2 are the addresses of the breakpoints. There are many allowed variations on this command, however, which makes it a powerful and convenient command. You have the option of establishing 0, 1, or 2 breakpoints: simply enter a [CR] when you have established the number of breakpoints you wish. If you enter the maximum, two, a delimiter (a comma or space) is all that is necessary to begin command execution.

You may also begin execution of the program at the PC address saved in the register storage area. Thus you can return control to the address where the program stopped when it encountered a breakpoint, or to the address you have loaded in the saved PC register through the Examine Register command. Note that since all breakpoints are cleared when any breakpoint is encountered, you must specify any desired breakpoints in the command if you use it this way. The form of the command for transferring program control to the address in the PC register is

-G[CR] (no breakpoints)
or
-G,B1,B2 (breakpoints set)

There are two more points regarding breakpoints that ought to be mentioned. Because breakpoints are generated by the monitor inserting a RST 8 instruction (CF) into the program at the breakpoint location, breakpoints can be set only in programs residing in RAM. Further, a breakpoint must be inserted at an op code location. If it is inserted in an operand or data field, it will not be executed.

3.5.6.6 Hex Number Addition (H)

This command provides an easy way to add or subtract hex addresses. Entering

-HA1 A2

where A1 and A2 are the hex addresses results in the output

AS AD

where AS=A1+A2 and AD=A1-A2. Note that if the sum is greater than FFFF, the carried one is lost. If A2 is greater than A1, A2 will be subtracted from A1 + 10000h.

3.5.6.7 Input (I)

This general purpose input command allows you to read a data byte from any input port. To do so, enter

-IA

where A is the port address in hex. The monitor will respond by printing the data byte in binary.

3.5.6.8 Move (M)

The M command moves a block of data to a specified address. The general form for the command is

-MA1 A2 AD

where A1 and A2 are the addresses of the first and last bytes of the memory block and AD is the destination address.

When using this command, be careful not to locate the destination address within the source block. Since the block is moved byte by byte, starting with the byte with the lowest address, the data being transferred will write over the original contents of the section of the source block that follows the destination address.

3.5.6.9 Output (O)

This general purpose output command allows you to output a data byte to any output port. Enter

-OA D

where A is the port address and D is the data in hex.

Please note that if the BOOT EN jumper is set to position A and you output to port 40h, you will disable the monitor portion of the ROM. The results of doing so are unpredictable.

3.5.6.10 Parameters (P)

The P command allows you to specify three parameters concerning the diskette selected for disk operations: the number of the unit it is in (U); the number of sectors it has per track; (S); and whether it is a one-sided or two-sided diskette. These parameters must be set before you attempt a disk read or write; however, they do not need to be reset until the parameters are no longer valid. The form of the command is:

-PU S D

U is a number 0 through 3, where 0 selects drive A, 1 selects drive B, etc. If you try to assign a number greater than 3, the monitor will return with ???? and the prompt. S is the number of sectors per track in hex. It is dependent on the number of bytes per sector, the diskette size and the density format. The following table shows the allowable number of sectors per track for a diskette of a given size and format:

BYTES PER SECTOR	8" DISKETTES		5.25" DISKETTES	
	SINGLE-DENSITY	DOUBLE-DENSITY	SINGLE-DENSITY	DOUBLE-DENSITY
128	1Ah (26d)	--	12h (18d)	--
256	Fh (15d)	1Ah (26d)	Ah (10d)	12h (18d)
512	8	Fh (15d)	5	Ah (10d)
1024	--	8	--	5

TABLE 3-5 DISKETTE FORMAT AND SECTORS PER TRACK

Note the firmware does not support 1024-byte sectors in double-density and 128-bytes in double-density. The last parameter, D, is 0 for a one-sided diskette; 1 for a two-sided diskette.

3.5.6.11 Parameters 2 (Q)

The Q command allows you to set the starting track, side, and sector number for disk reads or writes. Enter

-QT D S

where T is the beginning track number, D is the disk side, and S is the beginning sector number. If you plan to be transferring contiguous data to or from the disk, these parameters need to be set only prior to the first disk access. They must be reset for noncontiguous memory or sectors. T is the track number in hexidecimal. In practice, T will probably be a number between 0 and 4Ch (76d), inclusive, although the monitor will accept any value up to FFh. D is either a 0 or 1, depending on which side of the disk you wish the read or write to be performed on. S is the sector number in hexidecimal. It will always be a number between 1 and 1Ah, inclusive. Should you assign a track number or sector number greater than the number of tracks or sectors on the disk, you will get the Disk Error message when you use the Read or Write commands.

3.5.6.12 Read (R)

The R command allows you to transfer data from a disk into a specified area of memory. The R command sets the memory parameters; the disk parameters must have already been set by the P and Q commands. Enter

-RA1 A2

where A1 is the start address in memory and A2 is the end address. The R command does only complete sector transfers. Thus if the end address is reached before a sector is completely transferred into memory, the data will overflow the specified memory area. If the diskette is single-sided and the last sector in a track is reached before the read into memory is complete, the drive head steps in to the next track and the sector pointer is reset to 1. The number of sectors per track set by the P command determines whether or not the end of the track is reached. In the case of track overflow on side 0 of a double-sided diskette, the read continues on the same track on

side 1. A track overflow on side 1 causes the head to step in and read the next track on side 0.

Please remember that reading double-density diskettes requires a 4 MHz CPU.

3.5.6.13 Substitute (S)

The substitute command allows you to examine the contents of a specific memory location and alter them if you desire. Begin the S command by entering

-SA,

where A is the address of the memory location you wish to examine. The computer will immediately respond with the data contents followed by a prompt:

-SA,D-

If you wish to leave the data unaltered, simply enter a delimiter. If the delimiter is a space or a comma, the computer will respond with the contents of the next consecutive memory location and another prompt. If it is a carriage return, the command is terminated and control is returned to the monitor. Should you wish to alter the data, enter the desired data followed by a delimiter: a carriage return if you want to terminate the command or a space or a comma if you wish to review the next memory location. You also have the option of reviewing the previous memory location by hitting the line feed key. You can continue examining and altering memory byte by byte in this way as long as you wish. To make it easier for you to keep track of where you are, on every 8-byte boundary (that is, an address ending with either 0 or 8, the monitor will do a line feed and print the address along with the data.

3.5.6.14 Test (T)

The test command provides a quick way to test RAM memory for hard data bit failures without destroying the contents of the RAM. To test a block of memory for bit failures, enter

-TA1 A2

where A1 and A2 are the addresses of the first and last bytes in the block, respectively. The monitor will respond by printing the address of any byte in error, followed by an 8-bit

representation of the byte in which a one indicates an erroneous bit. For example, should bit 4 of location A3F8h be in error, the monitor outputs the following display

A3F8 00001000

If you wish to freeze the display type a Control-S. To start it again, hit any key. Hitting any key while the command is executing returns you to the monitor.

3.5.6.15 Verify (V)

You can use the V command to compare two blocks of memory and verify that they are the same. Type

-VA1 A2 AV

where A1 and A2 are the addresses of the first and last byte in the source block and AV is the starting address of the block to be verified. Should the two blocks match, the monitor will return with the prompt. Should the contents of two bytes sharing the same relative address differ, the monitor will display the source address and byte, followed by a dash and the corresponding byte in the block being verified. During the execution of the command, the display can be frozen or control returned to the monitor as described in previous section.

3.5.6.16 Write (W)

The W command allows you to transfer a specified block of memory to a disk. The W command sets the memory parameters; the disk parameters must have been already set by the P and Q commands. (Mind your P's and Q's before doing Reads and Writes). Enter

-WA1 A2

where A1 is the start address of the memory block and A2 is the end address. The write routine checks to see if the end address in memory has been reached only after it has completed a sector write. If the end address is reached before a sector write is completed, the routine will continue to pull data from memory until the sector is filled. During disk writes, track overflow is handled as described in the Read command. Please note the writing of double-density diskettes requires a 4 MHz CPU.

3.5.6.17 Examine (X)

The X command is a very useful command when used in conjunction with the G command's breakpoint facilities. Entering

-X[CR, space or comma]

causes the Z-80 registers currently stored in the system stack area to be displayed for examination. These registers are the main and alternate accumulator and general purpose registers, the Interrupt register (I), the Program Counter register (P), the Stack Pointer register (S), the two Index Registers (X and Y) and the Refresh register (R). In addition, the contents of the memory locations addressed by the main and alternate H and L registers are also displayed (M and M'). The registers are displayed in the following four-row format

```
A-xx B-xx C-xx D-xx E-xx F-xx H-xx L-xx
M-xx P-xxxx S-xxxx I-xx
A'-xx B'-xx C'-xx D'-xx E'-xx F'-xx H'-xx L'-xx
M'-xx X-xxxx Y-xxxx R-xx
```

where xx equals a two digit hex byte and xxxx equals a four digit hex address.

To examine or alter the contents of one register, enter

-Xr[CR, space or comma]
or
-X'r[CR, space or comma]

where r is a main register and r' is an alternate register. (Note that if you wish to examine the X, Y, or R registers, you must preface the register character with the prime mark.) The monitor will return with the contents of the register and a prompt:

-Xr,Dh-

As in the substitute memory command, you have the option of altering the memory (entering the desired contents followed by a delimiter) or leaving the contents unchanged (entering a delimiter). A carriage return terminates the command; a space or a comma causes the contents of the next register to be displayed. Note that altering the contents of the H and L registers changes the contents of the registers themselves; if you wish to alter the contents of the memory location pointed to by them, alter the M register.

3.5.6.18 Initialize Baud Rate (Y)

To change the baud rate of your system without a system reset, use the Y command. Enter

-Y (no delimiter)

and then set the baud rate of your terminal to any baud rate between 2 and 56K baud. Hit the carriage return key until the monitor returns with the prompt. The maximum number of carriage returns required is three.

3.5.6.19 Zleep (Z)

You can use the Z command to prevent unauthorized use of your system. Entering

-Z (no delimiter)

locks up the system so it will not respond to anything other than the ASCII bell character (control G). Entering two consecutive bell characters will unlock the system, returning control to the monitor without altering anything.

CHAPTER 4

THEORY OF OPERATION

This chapter is organized into three parts: The 2422 program accessible registers, the system bus interface, and the disk drive interface. We do not discuss the operation of the 1793; such a discussion is beyond the scope of this manual. Instead we concentrate on our unique circuitry external to the 1793. We have, however, included its data sheet in Appendix C for those of you who need information on its operation. If you consult it, please keep in mind that the data sheet covers the entire 1790 family; certain portions may not be applicable to the 1793.

In this chapter, active-low signals are indicated with an asterisk following the signal name.

4.1 THE REGISTERS

The 1793 contains five addressable registers: the Command register (write only), the Status register (read only), the Track register, the Sector register, and the Data register. On the 2422, these registers are addressed as four I/O ports, 30-33h, the Command and Status registers sharing the same address. Programming information on these registers can be found in the 1793 data sheet in Appendix C. In addition, the 2422 contains four registers external to the 1793: Status registers 1 and 2 (read only) and Control registers 1 and 2 (write only). These registers are addressed as two I/O ports, 34h and 04h, the status registers being selected during Read cycles and the control registers during Write cycles. The status registers consist of two 8-bit buffers, U25 and U26. When enabled by being addressed during a Read cycle, these chips gate selected signals from the drive busses, the system bus, and the control registers onto the data bus to be read by the CPU. Control registers 1 and 2, when

addressed during a write cycle, latch the command bits on the data bus and output high or low signals to the disk drive busses, the CPU and drive interface circuitry, and the 1793. They are cleared by pRESET* or EXT CLR*. Control Register 1 consists of a 7-bit latch, U13, which latches data bits D0-D6, and an independent flip-flop, U34b, which latches D7, the Auto Wait bit. The Auto Wait bit is latched by a separate flip-flop so that it can be reset not only by pRESET* and EXT CLR*, but also by the INTRQ signal from the 1793 (see section 4.2.8, "Auto Wait"). Control Register 2 consists of a 3-bit latch, U12. For the bit definitions of the external control/status registers, see Appendix B.

4.2 THE SYSTEM INTERFACE

4.2.1 THE BANK SELECT CIRCUITRY

The 2422 registers and the on-board ROM cannot be selected unless the internal signal BANK SELECT* is active low. This signal is the Q* output of the flip-flop U30b; the complementary Q output is used to light the Bank LED. The conditions under which BANK SELECT* is active low depend on the setting of the BANK EN jumper. If the BANK EN jumper has been set to OFF, disabling the bank select circuitry, the Preset input to flip-flop U30b is jumpered to ground, forcing BANK SELECT* permanently low, thus circumventing the Bank Select circuitry. If the jumper is set to position ON, the Clear input to the flip-flop is jumpered to the pRESET* and EXT CLR* signals from the system bus. If either goes low, as they both would during power-on or system reset, the flip-flop is cleared, and BANK SELECT* is forced inactive high. After both pRESET* and EXT CLR* release the Clear input, the BANK SELECT* line can be set low if the flip-flop is clocked while its D input is high. The flip-flop is clocked when pWR* goes high at the end of an I/O write cycle to port 40h. The state of the D input is determined by the Bank Select Byte being written to port 40h at this time. Only if the Bank Select Byte has a 1 in the bit position that is jumpered on BANK BYTE jumpers will the D input be high, resulting in the active BANK SELECT*. Finally, if the BANK EN jumper has been set to RST, the flip-flop's Preset input has been jumpered to pRESET* and EXT CLR*. During power-on or reset, then, BANK SELECT* is forced active low. In this case, BANK SELECT* will go inactive high only if the flip-flop is clocked when its D input is low; in other words, if the user selects another bank for operation.

4.2.2 SELECTING THE 2422 REGISTERS

The decoding of the port addresses is accomplished primarily by U22, an address-decoding ROM. When it is enabled by either the active sOUT or sINP, it decodes the register address on the low-byte address lines into one of four outputs. One output goes low for address 40h and is used for clocking the bank select flip-flop, as described in the previous section. Another output goes low for addresses in the 30-33h range. It is ORed with BANK SELECT*; when both signals are low, the result is the active signal CONTROLLER SELECT*. This signal is tied to the 1793's Chip Select input, enabling the 1793 when it is active. Selection of the individual registers within the 1793 is performed by address lines A0 and A1, tied to the 1793's two address inputs.

The two remaining outputs of U22 are used to select the external registers. One goes low for either address 04h or 34h. When it is ORed with the active BANK SELECT*, the result is the active 04/34 SELECT* line. This line enables a 2- to 4-line decoder, U45a. The final output of U22, which goes low for address 34h, is input to this decoder, along with the WR line (high whenever MWRITE or pWR* is active). U45a decodes these two inputs into the four enable lines to the external registers: CNTRL1*, STAT1*, CNTRL2*, STAT2*.

4.2.2.1 Memory-mapping of the Registers

As mentioned before, the 2422 has optional memory-mapped I/O capabilities. U21, when installed, maps the all 2422 registers, expect for the Bank Select register, to the last six bytes but one of a 64K bank; that is, locations FFF8-FFFD. When U21 is enabled by an output of address-decoding ROM U23 going low in response to an FF on the high-order address line, U21 decodes a low-byte address in the F8-FD range into three outputs which correspond to the 30-33, 04/34, and 34 outputs of U22 and are tied to them. Thus if U21 receives an address in the range of F8-FB, for example, it pulls U22's 30-33 output low, resulting in CONTROLLER SELECT* as described above. Table 4-1 shows the registers' memory locations and the corresponding port addresses.

RAM Location	Port Address
FFF8	30
FFF9	31
FFFA	32
FFFB	33
FFFC	34
FFFD	04

TABLE 4-1 MEMORY-MAPPED I/O ADDRESSES

4.2.3 SELECTING THE ROM

The ROM has two enable inputs, both of which must be active low for the ROM to be enabled. One enable input is pulled low whenever pWR* and MWRITE are both inactive. The other enable input is pulled low by ROM SELECT*, the output of the ROM Select circuitry which is active whenever one of the enabled portions of the ROM is addressed while BANK SELECT* is active. Once the ROM is enabled, address lines A0-A10, input directly to the ROM itself, select one location in the ROM's 2K of memory.

The ROM Select circuitry is designed to distinguish the Basic I/O portion of the ROM so that it can be enabled independently of the monitor/bootstrap portion of the ROM. To do so, U23, an address decoding ROM, decodes a high-byte address byte in the range of F0-F7 into two outputs when it is enabled by sINP, sOUT, and sINTA being inactive while BANK SELECT* is active. One goes low for a high byte address of F6 or F7, indicating an address in the range of the Basic I/O portion of the ROM; the other goes low for any address in the ROM's range. Since the bootstrap loader and monitor are needed only before CP/M is loaded, the latter output of the decoding ROM is qualified by the signal BOOT ENABLE*. Only if BOOT ENABLE* is low can the output pull ROM SELECT* low.

The state of the BOOT ENABLE* line can be controlled one of three ways, depending on the setting of the BOOT EN jumper. If the BOOT EN jumper is set to OFF, BOOT ENABLE* is set permanently high. If the jumper is set to position A, the BOOT ENABLE line is jumpered to the Q output of flip-flop U30a. This flip-flop is cleared by PRESET* or EXT CLR*, thus forcing the BOOT ENABLE* line low during system power-on or reset and enabling the ROM. The flip-flop can then be clocked by an I/O write to port 40h. Since the D input to the flip-flop is tied high, BOOT ENABLE* goes high when the flip-flop is clocked. Because the bank the board resides in is also selected by an output to port 40h, the BANK SELECT* line must be either set permanently low or set low

on reset if this method of enabling/disabling the bootstrap loader is to work. If the BOOT EN jumper is set to position B, BOOT ENABLE* is jumpered to the BOOT* signal from Control Register 2. Thus the state of BOOT ENABLE* is entirely software controlled: writing a 0 to bit 7 of Control Register 2 pulls BOOT ENABLE* low; a 1 pulls it high.

Once CP/M is loaded and BOOT ENABLE* is high, disabling the monitor and bootstrap loader portions of the ROM, the basic I/O portion can still be accessed if the PR EN jumper is set ON. This allows the F6-F7 output of the address decoding ROM to pull ROM SELECT* low when it goes low and thus enable the ROM.

4.2.4 THE BOARD SELECT LINE AND LED

CONTROLLER SELECT*, 04/34 SELECT*, and ROM SELECT* are NANDed together by U32a, the output of which is BOARD SELECT. If any of these three lines is low, BOARD SELECT is pulled high, lighting the Board Select LED. BOARD SELECT, when inactive, disables the data bus buffers.

4.2.5 PHANTOM* AND FF DETECT

The FF Detect circuitry is used to detect unused locations in the on-board ROM so that when an unused location is addressed PHANTOM* is forced high, allowing another device to respond to the address. When an empty location in the ROM is addressed, the ROM outputs an FFh, or all ones. Only if an unused location is addressed will this be the case. An 8-input NAND gate, U40, monitors the internal data lines for this condition. As long as a non-FF byte is being transferred, the NAND gate's output is high. This high is in turn NANDed with pDBIN and BOARD SELECT. If both signals are high, the output of the NAND gate, PHANTOM*, is low and enables the Data In buffer. When the internal data lines contain an FF, the FF-detect NAND gate goes low, disabling PHANTOM* and the Data In buffer (input to the CPU). Thus another device can respond to the memory read without interference from the ROM.

4.2.6 THE DATA BUS

During Write cycles, the 2422's internal bi-directional data bus is driven by U38, an 8-bit buffer. When enabled by either MWRITE or pWR* being active when BOARD SELECT is active, this chip gates the data bits on the Data Out bus (output from the CPU) onto the 2422's internal data bus. When the chip is disabled, its outputs are in a high impedance state. The Data In bus is driven by U39, another 8-bit buffer. When enabled by BOARD SELECT, pDBIN, and the output from the FF Detect circuitry being high, this chip gates the data bits on the 2422's internal data bus onto the Data In bus. When disabled, its outputs are also in a high impedance state.

4.2.7 ROM WAIT CIRCUITRY

The purpose of the ROM Wait circuitry is to increase the memory access time allowed to the ROM and to the registers in the disk controller when they are memory mapped. One Wait state per memory cycle in which either the ROM or the registers are addressed is sufficient for this purpose. If the ROM WAIT jumper is set to ON, pREADY is forced low only when either ROM SELECT* or CONTROLLER SELECT* is low when pSYNC is high. pSYNC is used to ensure that that pREADY is pulled low in every cycle in which the ROM or disk controller chip is selected and that it remains low only long enough to generate one Wait state. If the ROM WAIT jumper is set to 4 MHZ, the state of pREADY is further qualified by the 2*/4 MHZ signal. Only when the 2*/4 MHZ signal is high, indicating the CPU is operating at 4 MHz, can pREADY go low.

4.2.8 AUTO WAIT CIRCUITRY

The Auto Wait circuitry is designed to force the CPU into as many Wait states as needed when the disk controller is not ready for transfer of data. It uses the state of the DRQ (Data Request) line from the 1793 to determine the data register's readiness for data transfer. As mentioned before, the user has a choice of two types of Auto Waits: one type generates Wait states when Status register 1 is read when DRQ is low; the other generates Wait states when the data register is selected when DRQ is low. Both types of Auto Waits are enabled by writing a 1 to bit 7 of Control Register 1. Addressing Control Register 1 clocks the Auto Wait flip-flop, U43b. The D input of the flip-flop is tied to data line DO7. If bit 7 is high, the Q

output of the flip-flop will be pulled high; the Q* low. Only if the outputs of the flip-flop are in these states will either type Auto Wait be enabled. The Auto Wait flip-flop is cleared by pRESET*, EXT CLR*, or INTRQ.

4.2.8.1 Status Register 1 Wait

The Q* signal from flip-flop U43b is ORed with DRQ. If Q* is high, the output of the OR gate will always be high, regardless of the state of DRQ, thus disqualifying DRQ and disabling the Auto Wait cicuitry. If it is low, a low on DRQ pulls the OR gate's output low. This low is then ORed with STAT1*, which, if active, pulls the OR gate's output low. If the AUTO WAIT jumper has been set to STAT, this low pulls pREADY low.

4.2.8.2 Data Register Wait

The Q signal from flip-flop U43b is ANDed with the inverted DRQ. If both signals are high, the resulting high output from the AND gate pulls the Clear input to flip-flop U43a high, allowing the flip-flop to be clocked and its outputs change state. The flip-flop is clocked by the output of U45b, which is used as a a 2- to 1-line decoder. U45b is enabled by the active CONTROLLER SELECT* and decodes address bits A0 and A1. When enabled, its output goes low when A0 and A1 are high, indicating the data register is being selected. This low is inverted and clocks the flip-flop. Since the flip-flop's D input is tied high, Q* will go low. This low, if the AUTO WAIT jumper is set to DATA, pulls pREADY low.

4.3 THE DISK DRIVE INTERFACE

4.3.1 THE CLOCK SIGNAL

The 1793 Disk Controller chip needs a 2 MHz signal at its CLK input when it is operating with 8" drives and a 1 MHz CLK input when operating with 5.25" drives. All timing on the 2422 board is controlled by a 16 MHz crystal. IC U15, a binary counter, divides the 16 Mhz signal by 2, 4, 8 and 16. The 1 and 2 MHz signals from the divide-by-16 and -8 outputs are input to U16a, a 4-to-1-line multiplexer, the output of which is tied to the CLK input of the 1793. The Select input controlling the output of this multiplexer is the MAXI*/MINI signal from Control Register 1. When the signal is low, selecting the 8" drive, the output of U16a is the 2 MHz clock. When the signal is high, selecting a 5.25" drive, the output of U16a is the 1 MHz clock.

4.3.2 THE READ CLOCK SIGNAL

The 1793 can separate the data bits from the mingled clock and data bit stream from the disk drive. To do so, however, it needs a Read Clock signal, RCLK, which provides the data and clock "windows" required to separate the data bits from the clock bits. RCLK must be phased so it frames a data or a clock pulse during one phase of its cycle. To do so, RCLK's nominal cycle should equal the Read Data cycle time: 2 usecs for an 8" double density disk, 4 usecs for an 8" single density disk or a 5.25" double density disk, and 8 usecs for a 5.25" single density disk.

To acheive a RCLK of the correct frequency, the 8 MHz, 4 MHz, and 2 MHz signals from the binary counter U15 are multiplexed by U16b, a 4-to-1-line multiplexer. MAXI* and DDEN* (Double Density) control the select lines of the multiplexer. Thus the multiplexer outputs the following clock rates for the following states of MAXI* and DDEN*:

MAXI*	DDEN*	SIGNAL RATE
0	0	8 MHz
0	1	4 MHz
1	0	4 MHz
1	1	2 MHz

The above rates are 16X the desired RCLK frequency for each combination of drive size and format density. The output of the multiplexer is used to clock an 8-bit parallel-out serial shift register, U17. The eight outputs of this shift register go high successively as the shift register is clocked; the time it takes for the eight output to go high, then, is equal to the length of one phase of RCLK.

The shift register is used in combination with a couple of flip-flops and NAND gates to detect approximately when pulses in the read data stream occur. The two flip-flops are triggered by the pulses in the Read data stream and are set by the count-3 and count-6 outputs from the shift register. This enables the circuitry to detect whether a pulse occurs before count 3, between and including counts 3 and 5, or after count 5. If the pulse occurs before count 3, the circuitry is set to clock the Read Clock flip-flop, U18b, on count 7. The Q output of this flip-flop is the RCLK signal to the 1793. If the pulse occurs on or between counts 3 and 5, the Read Clock flip-flop is clocked on count 8. Another flip-flop, clocked and cleared by the same signals used by the shift-register and set by the count 8 output of the shift register, allows the circuitry to clock the Read Clock flip-flop on count 9, if the pulse occurs after count 5. The delay between the pulse being received and the Read Clock flip-flop being clocked ensures that the pulse will fall well within the window provided by RCLK. As the Read Clock flip-flop is clocked, the shift register is cleared. It then counts to eight to create an opposite phase of the desired length and on the eighth count clocks the Read Clock flip-flop. Since the Q* output of the Read Clock flip-flop is its D input, the state of RCLK will then change again. This process continues, creating an RCLK signal of the needed rate and phasing. Since the Read Data pulses should occur within 16-count intervals (or some multiple of 16), pulses which occur before count 3 or after count 6 will tend to move toward the middle counts, since they clock the Read Clock flip-flop on counts 7 and 9, not 8. The result is an RCLK signal synchrononized to the Read Data pulses so that each pulse occurs in the middle of the same phase of RCLK.

4.3.3 RAW READ SIGNAL

The 1793 recommends that the Read Data pulses be approximately 250 nsecs in width so that they fall entirely within the window provided by RCLK. The 2422 employs a monostable multivibrator, U3a, to ensure that the pulses are approximately 250 nsecs in length. U3a is clocked by the rising edge of each pulse in the inverted READ DATA stream and is set generate a negative-going pulse of 250 nsecs each time it is

clocked. The output of this chip forms the Read Data input, RAW READ*, to the 1793.

4.3.4 WRITE PRECOMPENSATION

On a double-density formatted diskette, certain bit patterns may cause a bit to shift from its nominal write position and appear at the read data separator early or late enough not to fall within its window when the diskette is being read. Write precompensation rectifies this problem during disk writes by shifting such a bit from its nominal position in the opposite direction to its known read shift. The 1793 is smart enough to recognize the bit patterns that cause a bit to shift and puts out the signals EARLY and LATE to indicate that the bit being output should be write precompensated either early or late. Since write precompensation is usually necessary only for data written on tracks on the inner half of the disk, the 1793 also puts out the signal TG43 to indicate that the head is positioned over a track greater than 43. The 2422, when operating in the double density mode, uses these signals to write bits needing precompensation 160 nsecs early or late.

The 160 nsec interval is provided by a monostable multivibrator, U29a. The positive-going data and clock pulses from the 1793 are inverted, and the trailing edge of a pulse triggers the monostable multivibrator. It then puts out a series of positive-going pulses of 160 nsecs until it is retriggered by a new Write Data pulse.

The direction of the shift is provided by a shift register, U19. The active low clock or data pulse from the 1793 which triggers the multivibrator also pulls low the load input to the shift register, loading in the values on its parallel inputs. The shift register is then clocked by the 160 nsec pulses from the multivibrator. When the shift register is clocked, it outputs the value on its G input and shifts the values on its inputs down one. The inputs of primary interest are the EARLY*, LATE*, and NO PRECOMP* signals. The EARLY* and LATE* signals are the EARLY and LATE signals from the 1793 qualified by both TG43 and DDEN. Only if TG43 and DDEN are both active can either the EARLY* or LATE* signals be active. NO PRECOMP* is active whenever both EARLY* and LATE* are inactive. These signals, EARLY*, NO PRECOMP*, and LATE*, are the G, F, and E inputs to the register, respectively. As the register is clocked successively, they are each output in turn. A low output from the shift register clocks a second monostable vibrator, U29b, the output of which is the Write Data stream. The 200 nsec low-going pulse which results from the vibrator being clocked is the clock or data pulse to be written

to the disk. Thus if EARLY* is low, the shift register output goes low, clocking U29b, the first time the register is clocked--in other words, just after it has been loaded. If NO PRECOMP* is low, the output of the register does not go low until the register is clocked a second time, or 160 nsecs later. If LATE* is low, the shift register must be clocked three times after it has been loaded before its output goes low. Thus bits that are to be written early or late are shifted 160 nsecs in either direction from the NO PRECOMP, or nominal, position.

4.3.5 HEAD LOAD TIMING

After the 1793 has given a Head Load Command, it pulls the HLD output high and waits to start read or write operations until it receives an high signal on its Head Load Timing input, indicating that the head is engaged and operable. The 2422 ensures that HLT goes active after a sufficient delay from HLD. The rising edge of HLD clocks U3b, a monostable multivibrator, which outputs a negative-going pulse of about 50 msecs, the HLT signal. When this signal becomes high again, the 1793 assumes that the head is engaged.

APPENDIX A

THE 2422 DISK CONTROLLER BUSSES

A.1 THE 2422 SYSTEM BUS

The following are definitions of the system bus signals used by the 2422. With the exception of 2*/4 MHZ, the signals used conform to the IEEE proposed standards for the S-100 bus. Active low signals are indicated by an asterisk following the signal name.

A.1.1 ADDRESS AND DATA LINES

A0-A15 The 16-bit parallel address lines.

DI0-DI7 The 8-bit parallel data input lines to the CPU.

DO0-DO7 The 8-bit parallel data output lines from the CPU.

A.1.2 CPU STATUS SIGNALS

sINTA The Interrupt Acknowledge signal indicates the CPU has accepted an interrupt.

sOUT The Output signal indicates the CPU is executing an output instruction.

sINP The Input signal indicates the CPU is executing an input instruction.

2*/4 MHZ When high, this signal indicates the CPU is operating at 4 MHz. When low, it indicates the CPU is operating at 2 MHz.

A.1.3 CONTROL INPUTS

pSYNC The Sync signal indicates the presence of status bits on the Data Out bus.

pDBIN The Data Bus In signal indicates that the CPU is conditioned to read data bits on the Data In bus.

pWR* The Write signal indicates the presence of valid data on the output bus.

pRESET* The system Reset signal resets the CPU and bus slaves. It is generated during power-on and often by a front panel switch.

EXT CLR* When active, the External Clear signal resets the bus slaves. It is also generated during power-on and often by a front panel switch.

MWRITE The Memory Write signal indicates that the current data on the data out bus is to be written into the memory location specified by the address bus. Often generated by front panel devices, it usually is used for front panel memory deposit.

A.1.4 CONTROL OUTPUTS

pRDY The Ready signal allows an addressed bus slave to hold the CPU in a Wait state until the slave is ready for data transfer.

PHANTOM* The Phantom signal controls memory overlay. On the 2422 board, it is used to allow the on-board ROM to take precedence over memory devices sharing the same memory space.

pINT* (jumper-enabled) The Interrupt signal allows external devices to request service from the CPU.

NMI* (jumper-enabled) The Nonmaskable Interrupt signal allows external devices to assert an interrupt request that cannot be masked off by the CPU.

VI0*-VI7* (jumper-enabled) The Vectored Interrupt lines are used to allow interrupt arbitration between eight levels of interrupt request priorites. They are ususally input to an interrupt arbitrating device which then asserts pINT* to the CPU and outputs the appropriate vectoring data.

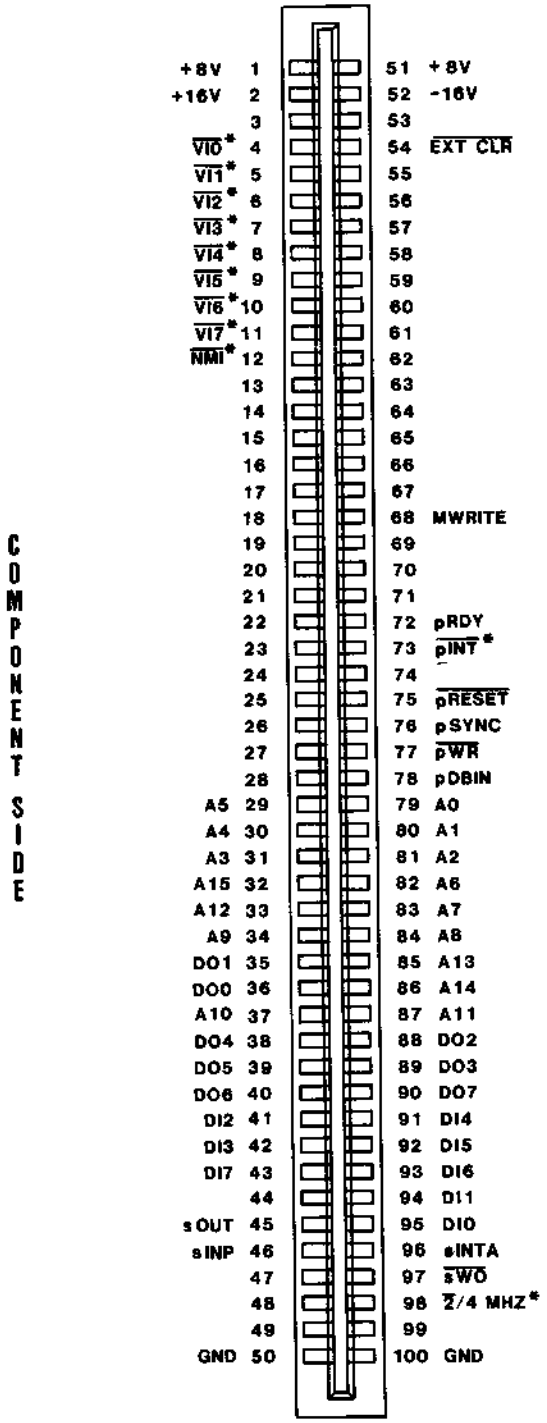
A.1.5 THE POWER LINES

+8 VOLTS The unregulated +8 volts power line.

+16 VOLTS The unregulated +16 volts power line.

-16 VOLTS The unregulated -16 volts power line.

FIGURE A-1 2422 BUS CONNECTOR PINOUTS



TOP VIEW

*Jumper-enabled signals

A.2 THE 2422 DISK DRIVE BUSSES

The following signals are used by the disk controller board to interface to both size drives.

OUTPUTS:

DS1*-DS4* When a Drive Select line is active low, the corresponding drive is enabled. The other drives will ignore all signals until selected.

MOTOR ON* When active low, the Motor On signal turns on the motor to all drives accepting the signal.

STEP* Each negative going pulse of this signal steps the read/write head forward or backward one pulse. For MFM the pulse width is 2us; for FM it is 4us. The stepping rate for multiple steps is determined by the Step Command.

DIRC The Direction signal determines the direction the read/write head steps. If it is low when STEP* goes active, the head steps in one track toward the center. If it is high when STEP* goes active, the head steps out one track toward the perimeter, or track 00.

WRITE GATE* When the Write Gate is active low, current flows into the read/write head, enabling diskette write operations.

WRITE DATA* The Write Data signal is the combined clock and data pulses that are written on the diskette. The pulse width is approximately 200 nsecs.

SIDE SELECT* This signal indicates which side of a two-sided disk should be used for reading or writing. A high selects side 0; a low, side 1.

INPUTS:

INDEX* The Index Pulse goes low for a minimum of 10us when the drive detects the index hole.

TRK 00* When low, this signal indicates that the read/write head is positioned over Track 00.

WPRT* The Write Protect signal goes low if the currently selected drive contains a write-protected diskette. It is sampled whenever the 1793 receives a write command and terminates that command if it is active low. (On some drives, write-protection detect circuitry is optional.)

WRITE DATA* This signal is the intermingled clock and data pulses received directly from the drive. Each recorded flux transistion results in a negative pulse.

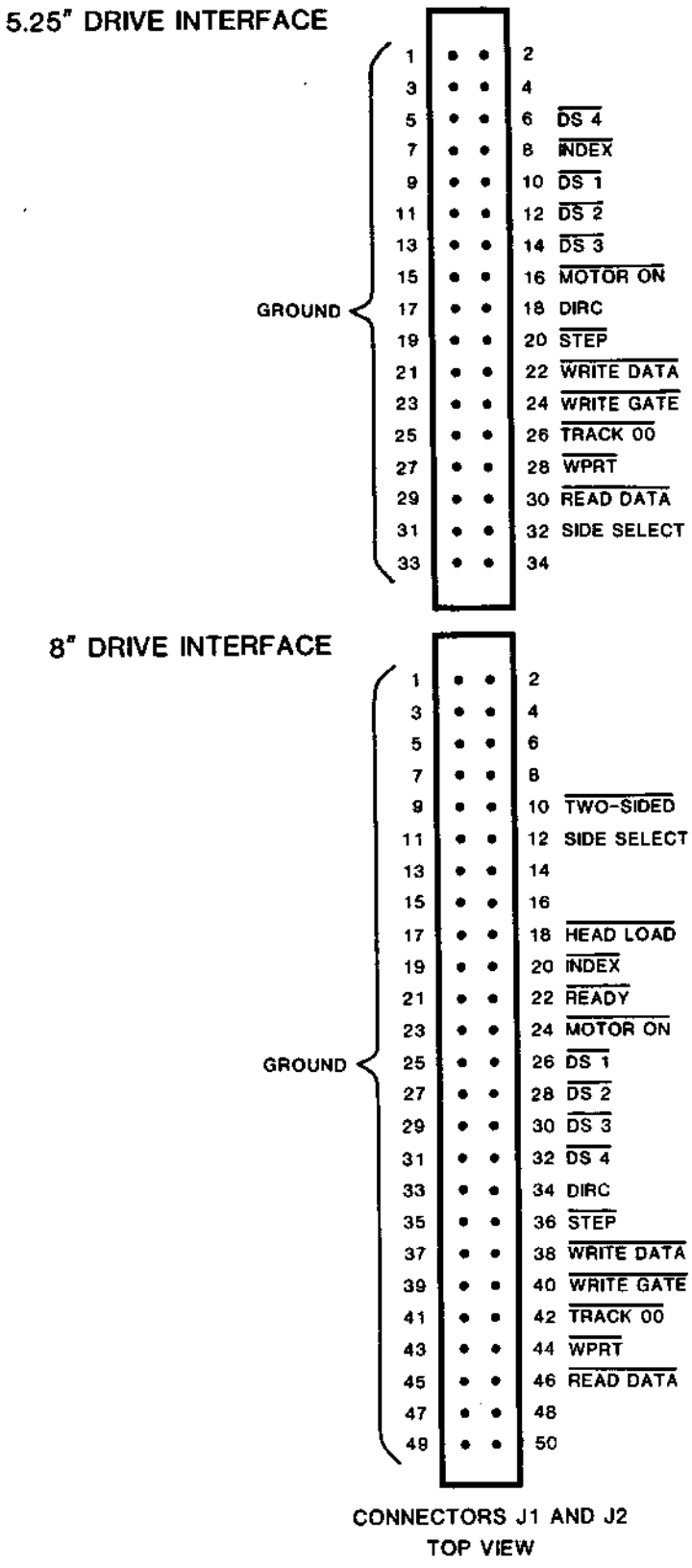
The following signals are available on the 8" drive bus only.

READY* This signal indicates that the disk drive is ready for operation.

TWO-SIDED* This signal goes active low when a two-sided diskette is in the currently selected drive.

HLD* When low, this signal tells the drive to load the read/write head against the diskette.

FIGURE A-2 2422 DISK DRIVE BUSSES PINOUTS



CONNECTORS J1 AND J2
TOP VIEW

APPENDIX B

2422 PROGRAMMING INFORMATION

B.1 THE 2422 ADDRESSABLE REGISTERS

The 2422 Floppy Disk Controller contains 9 accessible registers for controlling disk operations. They are addressed as six I/O ports or, if the memory map decoding ROM has been installed, six memory locations. Five of these registers are internal to the FD1791: the Status register (read-only), the Command register (write-only), the Track register, the Sector register, and the Data register. Four registers are external: Control registers 1 and 2 (write-only) and Status Registers 1 and 2 (read-only). In addition, the 2422 contains a write-only register for bank selection. The registers are addressed as follows:

Address		Register	
I/O	Memory*	Read	Write
30	FFF8	Status	Command
31	FFF9	Track	Track
32	FFFA	Sector	Sector
33	FFFB	Data	Data
34	FFFC	Status 1	Control 1
04	FFFD	Status 2	Control 2
40	----	----	Bank Select

* Memory Map address decoding ROM must be installed.

TABLE B-1 REGISTER ADDRESSES

The FD1793 Data Sheet included with this manual gives bit descriptions for each of the 1793's internal registers. Descriptions of the external registers follow.

B.1.1 CONTROL REGISTER 1

Summary:

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
AUTO WAIT	**DDEN**	**MOTOR ON**	**MINI**	**DS1**	**DS2**	**DS3**	**DS4**

TABLE B-2 CONTROL REGISTER 1

All the bits are reset by power-on, reset, or external clear.

Bit Definitions:

BIT 7 Auto Wait. A 1 written to bit 7 enables Auto Waits. A 0 disables them. Auto Waits are disabled after reset or after INTRQ, indicating the 1793 has finished executing a command, goes active.

BIT 6 Double Density. A 1 written to bit 6 conditions the 2422 for reading and writing double-density formatted diskettes. A 0 in bit 6 conditions the 2422 for single-density diskettes. Bit 6 is set to 0 on reset.

BIT 5 Motor On. Bit 5 controls the state of the MOTOR ON* signal. A 0 written to bit 5 forces MOTOR ON* low, turning on the motors of all drives accepting the signal. A 1 written to bit 5 forces MOTOR ON* high, turning off the drives' motors. MOTOR ON* is set high on reset.

BIT 4 Mini. A 0 written to bit 4 conditions the 2422 for operation with 5.25" (mini) drives. A 1 conditions the 2422 for operation with 8" drives. 8" drive operation is selected on reset.

BITS 3-0 Drive Select 1-4. These bits control the state of the Drive Select lines to the individual drives. A 1 written to one of the Drive Select bits activates the Drive Select line to the corresponding drive, selecting the drive for disk operations. Only one drive should be selected at a time. The Drive Select bits are set to 0 on reset.

B.1.2 STATUS REGISTER 1

Summary:

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
DRQ	**AUTO BOOT**	**HLD**	**DS1**	**DS2**	**DS3**	**DS4**	**INTRQ**

TABLE B-3 STATUS REGISTER 1

Bit Definitions:

BIT 7 DRQ. Bit 7 reflects the state of the DRQ (Data Request) signal from the 1793. During disk writes, a 1 in bit 7 indicates that the 1793's data register is empty and can accept a new byte to be written to disk. During disk reads, it indicates the 1793's data register holds a data byte to be read by the CPU. A 0 in bit 7 indicates the data register is not ready for data transfer with the CPU.

BIT 6 Auto Boot. Bit 6 is used by the CCS firmware during cold-start initialization to determine whether CP/M or the monitor is to be entered. It reflects the state of the Auto Boot jumper. If the AUTO BOOT jumper is set ON, bit 6 is set to 0, causing the cold-start initialization routine to turn control over to the bootstrap loader. If the AUTO BOOT jumper is set OFF, bit 6 is set to 1, causing the cold-start initialization routine to turn control over to the monitor executive.

BIT 5 Head Load. Bit 5 reflects the state of the HLD* signal from the 1793. A 1 in bit 5 indicates that the read/write head of the currently-selected drive is loaded.

BITS 4-1 Drive Select 1-4. A 1 in one of the Drive Select bits indicates that the corresponding drive has been selected for disk operations.

BIT 0 Interrupt Request. Bit 0 reflects the state of the INTRQ signal from the 1793. This signal goes high when the 1793 has finished executing the current command in the command register and is awaiting a new command.

B.1.3 CONTROL REGISTER 2

Summary:

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
BOOT	**SIDE SELECT**	don't care	**FAST SEEK**	don't care	don't care	don't care	don't care

TABLE B-4 CONTROL REGISTER 2

All bits are reset by power-on, reset, or external clear.

Bit Definitions:

BIT 7 Boot. If the BOOT EN jumper has been set to position B, bit 7 enables/disables the monitor/bootstrap loader firmware. A 0 written to bit 7 enables the firmware; a 1 disables it. This bit is set to 0 on reset.

BIT 6 Side Select. This bit controls the state of the SIDE SELECT signal to the currently-selected two-sided drive. A 1 written to bit 6 selects side 0 of a two-sided diskette for a read or write. A 0 written to bit 6 selects side 1 of a two-sided diskette. Side 0 is selected on reset.

BIT 4 Fast Seek. If the FAST SEEK jumper is set to SFT, bit 4 enables/disables the fast seek mode for voice-coil drives. A 0 written to bit 4 enables the fast seek mode; a 1 disables it. The fast seek mode is disabled on reset.

B.1.4 STATUS REGISTER 2

Summary:

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
DRQ	**TWO-SIDED**	**DDEN**	**INDEX**	**2/4 MHZ**	**WPRT**	**MINI**	**TRACK 00**

TABLE B-5 STATUS REGISTER 2

Bit Definitions:

BIT 7 DRQ. Bit 7 reflects the state of the DRQ signal from the 1793. During disk writes, a 1 in bit 7 indicates that the 1793's data register is empty and requires a new byte. During disk reads, a 1 in bit 7 indicates that the 1793's data register holds a data byte to be read by the CPU. A 0 in bit 7 indicates that the 1793's register is not ready for data transfer.

BIT 6 Two-sided. Bit 6 reflects the state of the signal TWO-SIDED* from the currently-selected two-sided drive. A 0 in bit 6 indicates a two-sided disk is in the drive.

BIT 5 Double-density. A 1 in bit 5 indicates that the 2422 has been conditioned to read or write double-density formatted diskettes. A 0 indicates the 2422 has been conditioned for single-density diskettes.

BIT 4 Index. Bit 4 reflects the state of the INDEX* signal from the currently- selected drive. It is set to 0 for a minimum of 10 usecs when the drive detects the index hole on the diskette.

BIT 3 2/4 MHZ. Bit 3 reflects the state of the signal on pin 98 of the system bus. In many systems this signal indicates the operating frequency of the processor. For such a system, a 1 in bit 3 indicates a 4 MHz operating frequency; a 0 indicates a 2 MHz operating frequency.

BIT 2 Write Protect. Bit 2 reflects the state of the WPRT* signal from the currently-selected drive. (On some drives write protect detection circuitry is an optional feature. See your manual.) A 0 in bit 2 indicates a write-protected diskette is in the currently selected drive.

BIT 1 Mini. A 1 in bit 1 indicates that the 2422 is conditioned for operation with a 5.25" drive. A 0 indicates that the 2422 is conditioned for an 8" drive.

BIT 0 Track 00. The CCS software uses this bit to determine whether the currently selected drive is a 5.25" or 8" drive. When the head is positioned over Track 00, bit 0 will be low for a 5.25" drive and high for an 8" drive.

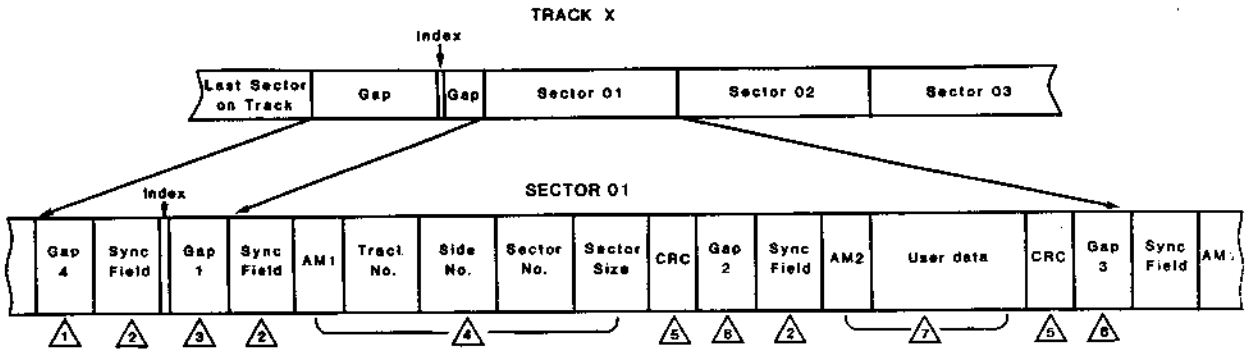
B.1.5 BANK SELECT PORT

To select the bank the 2422 is assigned to, output a data byte to port 40h with a 1 in the bit position corresponding to the bank level. Depending on the setting of the BANK EN jumper, this register is either cleared to 0 (Bank Disabled) or preset to 1 (Bank Enabled) on system power-on or reset.

If the BOOT EN jumper has been set to position A, writing any byte to port 40h disables the bootstrap loader and monitor firmware.

B.2 DISKETTE FORMAT

Figure B-1 below is an illustration of the IBM 3740 format for an 8" single-density diskette. The format differs slightly for a double-density diskette; see Table B-7 below and the 1793 data sheet for differences. There is no IBM standard for 5.25" diskettes; the 2422 software is designed to read and write 5.25" diskettes of a format adapted from the IBM standards for 8" diskettes. For the actual 5.25" and 8" single- and double-density formats used by the utility program CCSINIT in initializing diskettes, see Tables B-6 and B-7 below.



1. Pre-index gap. The 1793 expects all FF's.
2. 6 bytes of 00 in FM, 12 bytes of 00 in MFM.
3. Post-index gap. The 1793 expects all FF's.
4. ID FIELD

AM1 (Address Mark 1) = Hex FE. Identifies ID field.

Track No. = A value usually between hex 00 and 4C, inclusive. (0 and 76 decimal.)

Side No. = Hex 00 for one-sided diskettes and side 0 of two-sided diskettes.
Hex 01 for side 1 of two-sided diskettes.

Sector No. = Sector number in hex.

Sector Size = Hex 00 for 128 bytes per sector.
Hex 01 for 256 bytes per sector.
Hex 02 for 512 bytes per sector.
Hex 03 for 1024 bytes per sector.

5. Cyclic Redundancy Check bytes. CRC bytes are generated during disk writes. Used during disk reads to verify data is read correctly. CRC includes all data in ID and data fields starting with address mark.
6. Post-ID gap. The 1793 expects all FF's.
7. DATA FIELD

AM2 = hex FB. Identifies data field.
User data = 128, 256, 512, or 1024 bytes.

8. Post-data gap. The 1793 expects all FF's.

FIGURE B-1 IBM 3740 FORMAT

B.2.1 FORMATTING A SINGLE-DENSITY DISKETTE

Table B-6 below shows IBM-compatible formats for single-density 5.25" and 8" diskettes. These formats are both used by the CCSINIT utility program; the 8" diskette format conforms to the format specified by the 1793 data sheet.

	NUMBER OF BYTES 5.25"	NUMBER OF BYTES 8"	HEX VALUE OF BYTE WRITTEN
	16	40	FF (Gap 4)
	-	6	00 (Sync Field--8" only)
	-	1	FC (Index Mark--8" only)
	-	26	FF (Gap 1--8" only)
Write bracketed field once for every sector	6	6	00 (Sync Field)
	1	1	FE (ID Address Mark)
	1	1	Track Number
	1	1	Side Number (00 or 01)
	1	1	Sector Number
	1	1	Sector Size 00 = 128 bytes 01 = 256 bytes 02 = 512 bytes 03 = 1024 bytes
	1†	1†	F7 (CRC request)
	11	11	FF (Gap 2)
	6	6	00 (Sync Field)
	1	1	FB (Data Address Mark)
	$128*2^n$	$128*2^n$	Data (n=sector size indicator; data fill=E5)
	1†	1†	F7 (CRC request)
	11	27	FF (Gap 3)
	m	m	FF (m=variable number of bytes; continue writing until 1793 interrupts out.)

†CRC request is one byte; two CRC bytes actually written to disk.

TABLE B-6

B.2.2 FORMATTING A DOUBLE-DENSITY DISKETTE

Table B-7 below shows IBM-compatible formats for double-density 5.25" and 8" diskettes. Both of these formats are used by the utility program CCSINIT; the 8" diskette format conforms to the format specified by the 1793 data sheet.

	NUMBER OF BYTES 5.25"	NUMBER OF BYTES 8"	HEX VALUE OF BYTE WRITTEN
	32	80	4E (Gap 4)
	-	12	00 (Sync Field-8" only)
	-	3	F6 (8" only)
	-	1	FC (Index Mark--8" only)
	-	50	4E (Gap 1--8" only)
Write bracketed field once for every sector	8	12	00 (Sync Field)
	3	3	F5
	1	1	FE (ID Address Mark)
	1	1	Track No.
	1	1	Side No. (00 or 01)
	1	1	Sector No.
	1	1	Sector Size 00 = 128 bytes 01 = 256 bytes 02 = 512 bytes 03 = 1024 bytes
	1†	1†	F7 (CRC Request)
	22	22	4E (Gap 2)
	12	12	00 (Sync Field)
	3	3	F5
	1	1	FB (Data Address Mark)
	$128*2^n$	$128*2^n$	Data (n=sector size indicator; data fill=E5††)
	1†	1†	F7 (CRC request)
	22	54	4E (Gap 3)
	m	m	4E (m = variable number of bytes; continue writing until 1793 interrupts out.)

†CRC request is one byte; two CRC bytes actually written to disk.

††CP/M requires an E5h fill character, while the IBM-format specifies 40h as the fill character.

TABLE B-7

APPENDIX C

1793 DATA SHEET

WESTERN DIGITAL
CORPORATION

FD 179X-02 Floppy Disk Formatter/Controller Family

MAY 1980

FEATURES

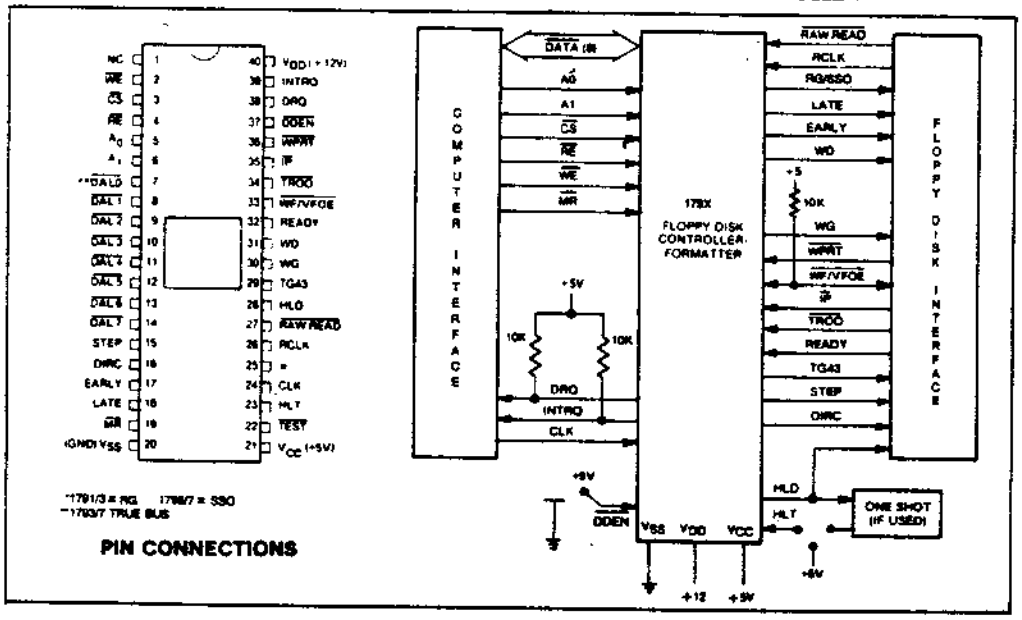
- TWO VFO CONTROL SIGNALS
- SOFT SECTOR FORMAT COMPATIBILITY
- AUTOMATIC TRACK SEEK WITH VERIFICATION
- ACCOMMODATES SINGLE AND DOUBLE DENSITY FORMATS
  - IBM 3740 Single Density (FM)
  - IBM System 34 Double Density (MFM)
- READ MODE
  - Single/Multiple Sector Read with Automatic Search or Entire Track Read
  - Selectable 128 Byte or Variable length Sector
- WRITE MODE
  - Single/Multiple Sector Write with Automatic Sector Search
  - Entire Track Write for Diskette Formatting
- SYSTEM COMPATIBILITY
  - Double Buffering of Data 8 Bit Bi-Directional Bus for Data, Control and Status
  - DMA or Programmed Data Transfers
  - All Inputs and Outputs are TTL Compatible
  - On-Chip Track and Sector Registers/Comprehensive Status Information
- PROGRAMMABLE CONTROLS
  - Selectable Track to Track Stepping Time
  - Side Select Compare
- WRITE PRECOMPENSATION
- WINDOW EXTENSION
- INCORPORATES ENCODING/DECODING AND ADDRESS MARK CIRCUITRY
- FD1792/4 IS SINGLE DENSITY ONLY
- FD1795/7 HAS A SIDE SELECT OUTPUT

179X-02 FAMILY CHARACTERISTICS

FEATURES	1791	1793	1795	1797
Single Density (FM)	X	X	X	X
Double Density (MFM)	X	X	X	X
True Data Bus		X		X
Inverted Data Bus	X		X	
Write Precomp	X	X	X	X
Side Selection Output			X	X

APPLICATIONS

FLOPPY DISK DRIVE INTERFACE
SINGLE OR MULTIPLE DRIVE CONTROLLER/ FORMATTER
NEW MINI-FLOPPY CONTROLLER



FD179X SYSTEM BLOCK DIAGRAM

*1791/3 = RG 1795/7 = SSO
**1793/7 TRUE BUS

PIN CONNECTIONS

GENERAL DESCRIPTION

The FD179X are MOS LSI devices which perform the functions of a Floppy Disk Formatter/Controller in a single chip implementation. The FD179X, which can be considered the end result of both the FD1771 and FD1781 designs, is IBM 3740 compatible in single density mode (FM) and System 34 compatible in Double Density Mode (MFM). The FD179X contains all the features of its predecessor the FD1771, plus the added features necessary to read/write and format a double density diskette. These include address mark detection, FM and MFM encode and decode logic, window extension, and write precompensation. In order to maintain compatibility, the FD1771, FD1781, and FD179X designs were made as close as possible with the computer interface, instruction set, and I/O registers being identical. Also, head load control is identical. In each case, the actual pin assignments vary by only a few pins from any one to another.

The processor interface consists of an 8-bit bidirectional bus for data, status, and control word transfers. The FD179X is set up to operate on a multiplexed bus with other bus-oriented devices.

The FD179X is fabricated in N-channel Silicon Gate MOS technology and is TTL compatible on all inputs and outputs. The 1793 is identical to the 1791 except the DAL lines are TRUE for systems that utilize true data busses.

The 1795/7 has a side select output for controlling double sided drives, and the 1792 and 1794 are "Single Density Only" versions of the 1791 and 1793. On these devices, DDEN must be left open.

PIN OUTS

PIN NUMBER	PIN NAME	SYMBOL	FUNCTION
1	NO CONNECTION	NC	Pin 1 is internally connected to a back bias generator and must be left open by the user.
19	MASTER RESET	$\overline{MR}$	A logic low on this input resets the device and loads HEX 03 into the command register. The Not Ready (Status Bit 7) is reset during $\overline{MR}$ ACTIVE. When $\overline{MR}$ is brought to a logic high a RESTORE Command is executed, regardless of the state of the Ready signal from the drive. Also, HEX 01 is loaded into sector register.
20	POWER SUPPLIES	V_{SS}	Ground
21		V_{CC}	+5V ±5%
40		V_{DD}	+12V ±5%
COMPUTER INTERFACE:			
2	WRITE ENABLE	$\overline{WE}$	A logic low on this input gates data on the DAL into the selected register when $\overline{CS}$ is low.
3	CHIP SELECT	$\overline{CS}$	A logic low on this input selects the chip and enables computer communication with the device.
4	READ ENABLE	$\overline{RE}$	A logic low on this input controls the placement of data from a selected register on the DAL when $\overline{CS}$ is low.
5,6	REGISTER SELECT LINES	A0, A1	These inputs select the register to receive/transfer data on the DAL lines under $\overline{RE}$ and $\overline{WE}$ control: A1 A0 $\overline{RE}$ $\overline{WE}$ 0 0 Status Reg Command Reg 0 1 Track Reg Track Reg 1 0 Sector Reg Sector Reg 1 1 Data Reg Data Reg
7-14	DATA ACCESS LINES	$\overline{DAL0}$-$\overline{DAL7}$	Eight bit inverted Bidirectional bus used for transfer of data, control, and status. This bus is receiver enabled by $\overline{WE}$ or transmitter enabled by $\overline{RE}$.
24	CLOCK	CLK	This input requires a free-running square wave clock for internal timing reference, 2 MHz for 8" drives, 1 MHz for mini-drives.

PIN NUMBER	PIN NAME	SYMBOL	FUNCTION
38	DATA REQUEST	DRQ	This open drain output indicates that the DR contains assembled data in Read operations, or the DR is empty in Write operations. This signal is reset when serviced by the computer through reading or loading the DR in Read or Write operations, respectively. Use 10K pull-up resistor to +5.
39	INTERRUPT REQUEST	INTRQ	This open drain output is set at the completion of any command and is reset when the STATUS register is read or the command register is written to. Use 10K pull-up resistor to +5.
FLOPPY DISK INTERFACE:			
15	STEP	STEP	The step output contains a pulse for each step.
16	DIRECTION	DIRC	Direction Output is active high when stepping in, active low when stepping out.
17	EARLY	EARLY	Indicates that the WRITE DATA pulse occurring while Early is active (high) should be shifted early for write precompensation.
18	LATE	LATE	Indicates that the write data pulse occurring while Late is active (high) should be shifted late for write precompensation.
22	$\overline{\text{TEST}}$	$\overline{\text{TEST}}$	This input is used for testing purposes only and should be tied to +5V or left open by the user unless interfacing to voice coil actuated motors.
23	HEAD LOAD TIMING	HLT	When a logic high is found on the HLT input the head is assumed to be engaged.
25	READ GATE (1791/3)	RG	A high level on this output indicates to the data separator circuitry that a field of zeros (or ones) has been encountered, and is used for synchronization.
25	SIDE SELECT OUTPUT (1795, 1797)	SSO	The logic level of the Side Select Output is directly controlled by the 'S' flag in Type II or III commands. When S = 1, SSO is set to a logic 1. When S = 0, SSO is set to a logic 0. The Side Select Output is only updated at the beginning of a Type II or III command. It is forced to a logic 0 upon a MASTER RESET condition.
26	READ CLOCK	RCLK	A nominal square-wave clock signal derived from the data stream must be provided to this input. Phasing (i.e. RCLK transitions) relative to RAW READ is important but polarity (RCLK high or low) is not.
27	$\overline{\text{RAW READ}}$	$\overline{\text{RAW READ}}$	The data input signal directly from the drive. This input shall be a negative pulse for each recorded flux transition.
28	HEAD LOAD	HLD	The HLD output controls the loading of the Read-Write head against the media.
29	TRACK GREATER THAN 43	TG43	This output informs the drive that the Read/Write head is positioned between tracks 44-76. This output is valid only during Read and Write Commands.
30	WRITE GATE	WG	This output is made valid before writing is to be performed on the diskette.

PIN NUMBER	PIN NAME	SYMBOL	FUNCTION
31	WRITE DATA	WD	A 250 ns (MFM) or 500 ns (FM) pulse per flux transition. WD contains the unique Address marks as well as data and clock in both FM and MFM formats.
32	READY	READY	This input indicates disk readiness and is sampled for a logic high before Read or Write commands are performed. If Ready is low the Read or Write operation is not performed and an interrupt is generated. Type I operations are performed regardless of the state of Ready. The Ready input appears in inverted format as Status Register bit 7.
33	$\overline{\text{WRITE FAULT}}$ $\overline{\text{VFO ENABLE}}$	$\overline{\text{WF/VFOE}}$	This is a bi-directional signal used to signify writing faults at the drive, and to enable the external PLO data separator. When WG = 1, Pin 33 functions as a WF input. If WF = 0, any write command will immediately be terminated. When WG = 0, Pin 33 functions as a VFOE output. VFOE will go low during a read operation after the head has loaded and settled (HLT = 1). On the 1795/7, it will remain low until the last bit of the second CRC byte in the ID field. VFOE will then go high until 8 bytes (MFM) or 4 bytes (FM) before the Address Mark. It will then go active until the last bit of the second CRC byte of the Data Field. On the 1791/3, VFOE will remain low until the end of the Data Field.
34	$\overline{\text{TRACK 00}}$	$\overline{\text{TR00}}$	This input informs the FD179X that the Read/Write head is positioned over Track 00.
35	$\overline{\text{INDEX PULSE}}$	$\overline{\text{IP}}$	This input informs the FD179X when the index hole is encountered on the diskette.
36	$\overline{\text{WRITE PROTECT}}$	$\overline{\text{WPRT}}$	This input is sampled whenever a Write Command is received. A logic low terminates the command and sets the Write Protect Status bit.
37	$\overline{\text{DOUBLE DENSITY}}$	$\overline{\text{DDEN}}$	This pin selects either single or double density operation. When $\overline{\text{DDEN}}$ = 0, double density is selected. When $\overline{\text{DDEN}}$ = 1, single density is selected. This line must be left open on the 1792/4

ORGANIZATION

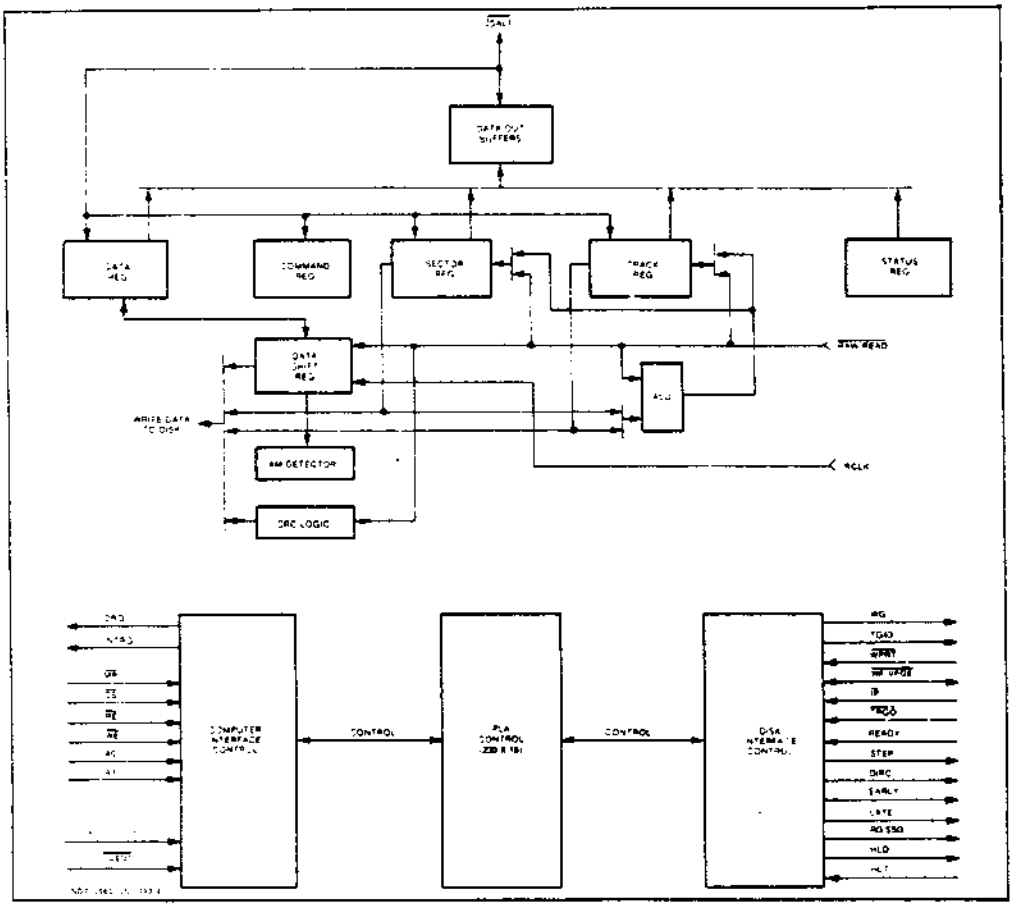
The Floppy Disk Formatter block diagram is illustrated on page 5. The primary sections include the parallel processor interface and the Floppy Disk interface.

Data Shift Register—This 8-bit register assembles serial data from the Read Data input ($\overline{\text{RAW READ}}$) during Read operations and transfers serial data to the Write Data output during Write operations.

Data Register—This 8-bit register is used as a holding register during Disk Read and Write operations. In Disk Read operations the assembled data byte is transferred in parallel to the Data Register from the Data Shift Register. In Disk Write operations information is transferred in parallel from the Data Register to the Data Shift Register.

When executing the Seek command the Data Register holds the address of the desired Track position. This register is loaded from the DAL and gated onto the DAL under processor control.

Track Register—This 8-bit register holds the track number of the current Read/Write head position. It is incremented by one every time the head is stepped in (towards track 76) and decremented by one when the head is stepped out (towards track 00). The contents of the register are compared with the recorded track number in the ID field during disk Read, Write, and Verify operations. The Track Register can be loaded from or transferred to the DAL. This Register should not be loaded when the device is busy.



FD179X BLOCK DIAGRAM

Sector Register (SR)—This 8-bit register holds the address of the desired sector position. The contents of the register are compared with the recorded sector number in the ID field during disk Read or Write operations. The Sector Register contents can be loaded from or transferred to the DAL. This register should not be loaded when the device is busy.

Command Register (CR)—This 8-bit register holds the command presently being executed. This register should not be loaded when the device is busy unless the new command is a force interrupt. The command register can be loaded from the DAL, but not read onto the DAL.

Status Register (STR)—This 8-bit register holds device Status information. The meaning of the Status bits is a function of the type of command previously executed. This register can be read onto the DAL, but not loaded from the DAL.

CRC Logic—This logic is used to check or to generate the 16-bit Cyclic Redundancy Check (CRC). The polynomial is: $G(x) = x^{16} + x^{12} + x^5 + 1$.

The CRC includes all information starting with the address mark and up to the CRC characters. The CRC register is preset to ones prior to data being shifted through the circuit.

Arithmetic/Logic Unit (ALU)—The ALU is a serial comparator, incrementer, and decrementer and is used for register modification and comparisons with the disk recorded ID field.

Timing and Control—All computer and Floppy Disk Interface controls are generated through this logic. The internal device timing is generated from an external crystal clock.

The FD1791/3 has two different modes of operation according to the state of $\overline{DDEN}$. When $\overline{DDEN}$ = 0 double density (MFM) is assumed. When $\overline{DDEN}$ = 1, single density (FM) is assumed.

AM Detector—The address mark detector detects ID, data and index address marks during read and write operations.

PROCESSOR INTERFACE

The interface to the processor is accomplished through the eight Data Access Lines ($\overline{DAL}$) and associated control signals. The $\overline{DAL}$ are used to transfer Data, Status, and Control words out of, or into the FD179X. The $\overline{DAL}$ are three state buffers that are enabled as output drivers when Chip Select ($\overline{CS}$) and Read Enable ($\overline{RE}$) are active (low logic state) or act as input receivers when $\overline{CS}$ and Write Enable ($\overline{WE}$) are active.

When transfer of data with the Floppy Disk Controller is required by the host processor, the device address is decoded and $\overline{CS}$ is made low. The address bits A1 and A0, combined with the signals $\overline{RE}$ during a Read operation or $\overline{WE}$ during a Write operation are interpreted as selecting the following registers:

A1-A0	READ ($\overline{RE}$)	WRITE ($\overline{WE}$)
0 0	Status Register	Command Register
0 1	Track Register	Track Register
1 0	Sector Register	Sector Register
1 1	Data Register	Data Register

During Direct Memory Access (DMA) types of data transfers between the Data Register of the FD179X and the processor, the Data Request (DRQ) output is used in Data Transfer control. This signal also appears as status bit 1 during Read and Write operations.

On Disk Read operations the Data Request is activated (set high) when an assembled serial input byte is transferred in parallel to the Data Register. This bit is cleared when the Data Register is read by the processor. If the Data Register is read after one or more characters are lost, by having new data transferred into the register prior to processor readout, the Lost Data bit is set in the Status Register. The Read operation continues until the end of sector is reached.

On Disk Write operations the data Request is activated when the Data Register transfers its contents to the Data Shift Register, and requires a new data byte. It is reset when the Data Register is loaded with new data by the processor. If new data is not loaded at the time the next serial byte is required by the Floppy Disk, a byte of zeroes is written on the diskette and the Lost Data bit is set in the Status Register.

At the completion of every command an INTRQ is generated. INTRQ is reset by either reading the status register or by loading the command register with a new command. In addition, INTRQ is generated if a Force Interrupt command condition is met.

FLOPPY DISK INTERFACE

The 179X has two modes of operation according to the state of $\overline{DDEN}$ (Pin 37). When $\overline{DDEN}$ = 1, single density is selected. In either case, the CLK input (Pin 24) is at 2 MHz. However, when interfacing with the mini-floppy, the CLK input is set at 1 MHz for both single density and double density. When the clock is at 2 MHz, the stepping rates of 3, 6, 10, and 15 ms are obtainable. When CLK equals 1 MHz these times are doubled.

HEAD POSITIONING

Five commands cause positioning of the Read-Write head (see Command Section). The period of each positioning step is specified by the r field in bits 1 and 0 of the command word. After the last directional step an additional 15 milliseconds of head settling time takes place if the Verify flag is set in Type I commands. Note that this time doubles to 30 ms for a 1 MHz clock. If $\overline{TEST}$ = 0, there is zero settling time. There is also a 15 ms head settling time if the E flag is set in any Type II or III command.

The rates (shown in Table 1) can be applied to a Step-Direction Motor through the device interface.

Step—A 2 μs (MFM) or 4 μs (FM) pulse is provided as an output to the drive. For every step pulse issued, the drive moves one track location in a direction determined by the direction output.

Direction (DIRC)—The Direction signal is active high when stepping in and low when stepping out. The Direction signal is valid 12 μs before the first stepping pulse is generated.

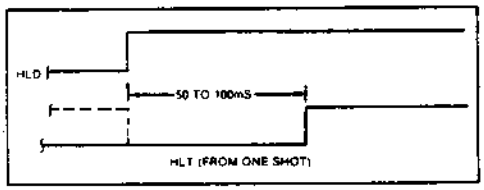
When a Seek, Step or Restore command is executed an optional verification of Read-Write head position can be performed by setting bit 2 (V = 1) in the command word to a logic 1. The verification operation begins at the end of the 15 millisecond settling time after the head is loaded against the media. The track number from the first encountered ID Field is compared against the contents of the Track Register. If the track numbers compare and the ID Field Cyclic Redundancy Check (CRC) is correct, the verify operation is complete and an INTRQ is generated with no errors. The FD179X must find an ID field with correct track number and correct CRC within 5 revolutions of the media; otherwise the seek error is set and an INTRQ is generated.

Table 1. STEPPING RATES

CLK	2 MHz	2 MHz	1 MHz	1 MHz	2 MHz	1 MHz
$\overline{DDEN}$	0	1	0	1	x	x
R1 R0	$\overline{TEST}$=1	$\overline{TEST}$=1	$\overline{TEST}$=1	$\overline{TEST}$=1	$\overline{TEST}$=0	$\overline{TEST}$=0
0 0	3 ms	3 ms	6 ms	6 ms	184μs	368μs
0 1	6 ms	6 ms	12 ms	12 ms	190μs	380μs
1 0	10 ms	10 ms	20 ms	20 ms	198μs	396μs
1 1	15 ms	15 ms	30 ms	30 ms	208μs	416μs

The Head Load (HLD) output controls the movement of the read/write head against the media. HLD is activated at the beginning of a Type I command if the h flag is set (h = 1), at the end of the Type I command if the verify flag (V = 1), or upon receipt of any Type II or III command. Once HLD is active it remains active until either a Type I command is received with (h = 0 and V = 0); or if the FD179X is in an idle state (non-busy) and 15 index pulses have occurred.

Head Load Timing (HLT) is an input to the FD179X which is used for the head engage time. When HLT = 1, the FD179X assumes the head is completely engaged. The head engage time is typically 30 to 100 ms depending on drive. The low to high transition on HLD is typically used to fire a one shot. The output of the one shot is then used for HLT and supplied as an input to the FD179X.



HEAD LOAD TIMING

When both HLD and HLT are true, the FD179X will then read from or write to the media. The "and" of HLD and HLT appears as a status bit in Type I status.

In summary for the Type I commands: if h = 0 and V = 0, HLD is reset. If h = 1 and V = 0, HLD is set at the beginning of the command and HLT is not sampled nor is there an internal 15 ms delay. If h = 0 and V = 1, HLD is set near the end of the command, an internal 15 ms occurs, and the FD179X waits for HLT to be true. If h = 1 and V = 1, HLD is set at the beginning of the command. Near the end of the command, after all the steps have been issued, an internal 15 ms delay occurs and the FD179X then waits for HLT to occur.

For Type II and III commands with E flag off, HLD is made active and HLT is sampled until true. With E flag on, HLD is made active, an internal 15 ms delay occurs and then HLT is sampled until true.

DISK READ OPERATIONS

Sector lengths of 128, 256, 512 or 1024 are obtainable in either FM or MFM formats. For FM, $\overline{DDEN}$ should be placed to logical "1." For MFM formats, $\overline{DDEN}$ should be placed to a logical "0." Sector lengths are determined at format time by a special byte in the "ID" field. If this Sector length byte in the ID field is zero, then the sector length is 128 bytes. If 01 then 256 bytes. If 02, then 512 bytes. If 03, then the sector length is 1024 bytes. The number of sectors per track as far as the FD179X is concerned can be from 1 to 255 sectors. The number of tracks as far as the FD179X is concerned is from 0 to 255 tracks. For IBM 3740 compatibility, sector lengths are 128 bytes with 26 sectors per track. For System 34 compatibility (MFM), sector lengths are 256 bytes/sector with 26 sectors/track; or lengths of 1024 bytes/sector with 8 sectors/track. (See Sector Length Table.)

For read operations, the FD179X requires $\overline{RAW\ READ}$ Data (Pin 27) signal which is a 250 ns pulse per flux transition and a Read clock (RCLK) signal to indicate flux transition spacings. The RCLK (Pin 26) signal is provided by some drives but if not it may be derived externally by Phase lock loops, one shots, or counter techniques. In addition, a Read Gate Signal is provided as an output (Pin 25) which can be used to inform phase lock loops when to acquire synchronization. When reading from the media in FM, RG is made true when 2 bytes of zeroes are detected. The FD179X must find an address mark within the next 10 bytes; otherwise RG is reset and the search for 2 bytes of zeroes begins all over again. If an address mark is found within 10 bytes, RG remains true as long as the FD179X is deriving any useful information from the data stream. Similarly for MFM, RG is made active when 4 bytes of "00" or "FF" are detected. The FD179X must find an address mark within the next 16 bytes, otherwise RG is reset and search resumes.

During read operations (WG = 0), the $\overline{VFOE}$ (Pin 33) is provided for phase lock loop synchronization. $\overline{VFOE}$ will go active when:

a) Both HLT and HLD are True
b) Settling Time, if programmed, has expired
c) The 179X is inspecting data off the disk

If $\overline{WF}/\overline{VFOE}$ is not used, leave open or tie to a 10K resistor to +5.

DISK WRITE OPERATION

When writing is to take place on the diskette the Write Gate (WG) output is activated, allowing current to flow into the Read/Write head. As a precaution to erroneous writing the first data byte must be loaded into the Data Register in response to a Data Request from the FD179X before the Write Gate signal can be activated.

Writing is inhibited when the $\overline{Write\ Protect}$ input is a logic low, in which case any Write command is immediately terminated, an interrupt is generated and the Write Protect status bit is set. The Write Fault input, when activated, signifies a writing fault condition detected in disk drive electronics such as failure to detect write current flow when the Write Gate is activated. On detection of this fault the FD179X terminates the current command, and sets the $\overline{Write\ Fault}$ bit (bit 5) in the Status Word. The $\overline{Write\ Fault}$ input should be made inactive when the Write Gate output becomes inactive.

For write operations, the FD179X provides Write Gate (Pin 30) and Write Data (Pin 31) outputs. Write data consists of a series of 500 ns pulses in FM ($\overline{DDEN}$ = 1) and 250 ns pulses in MFM ($\overline{DDEN}$ = 0). Write Data provides the unique address marks in both formats.

Also during write, two additional signals are provided for write precompensation. These are EARLY (Pin 17) and LATE (Pin 18). EARLY is active true when the WD pulse appearing on (Pin 30) is to be written early. LATE is active true when the WD pulse is to be written LATE. If both EARLY and LATE are low when the WD pulse is present, the WD pulse is to be written at nominal. Since write precompensation values vary from disk manufacturer to disk manufacturer, the actual value is determined by several one shots or delay lines which are located external to the FD179X. The write precompensation signals EARLY and LATE are valid for the duration of WD in both FM and MFM formats.

Whenever a Read or Write command (Type II or III) is received the FD179X samples the Ready input. If this input is logic low the command is not executed and an interrupt is generated. All Type I commands are performed regardless of the state of the Ready input. Also, whenever a Type II or III command is received, the TG43 signal output is updated.

COMMAND DESCRIPTION

The FD179X will accept eleven commands. Command words should only be loaded in the Command Register when the Busy status bit is off (Status bit 0). The one exception is the Force Interrupt command. Whenever a command is being executed, the Busy status bit is set. When a command is completed, an interrupt is generated and the Busy status bit is reset. The Status Register indicates whether the completed command encountered an error or was fault free. For ease of discussion, commands are divided into four types. Commands and types are summarized in Table 2.

Table 2. COMMAND SUMMARY

		BITS							
TYPE	COMMAND	7	6	5	4	3	2	1	0
I	Restore	0	0	0	0	h	V	r_1	r_0
I	Seek	0	0	0	1	h	V	r_1	r_0
I	Step	0	0	1	u	h	V	r_1	r_0
I	Step In	0	1	0	u	h	V	r_1	r_0
I	Step Out	0	1	1	u	h	V	r_1	r_0
II	Read Sector	1	0	0	m	F_2	E	F_1	0
II	Write Sector	1	0	1	m	F_2	E	F_1	a_0
III	Read Address	1	1	0	0	0	E	0	0
III	Read Track	1	1	1	0	0	E	0	0
III	Write Track	1	1	1	1	0	E	0	0
IV	Force Interrupt	1	1	0	1	I_3	I_2	I_1	I_0

Note: Bits shown in TRUE form.

Table 3. FLAG SUMMARY

TYPE I COMMANDS

h = Head Load Flag (Bit 3)
- h = 1, Load head at beginning
- h = 0, Unload head at beginning

V = Verify flag (Bit 2)
- V = 1, Verify on destination track
- V = 0, No verify

r_1-r_0 = Stepping motor rate (Bits 1-0)
- Refer to Table 1 for rate summary

u = Update flag (Bit 4)
- u = 1, Update Track register
- u = 0, No update

Table 4. FLAG SUMMARY

TYPE II & III COMMANDS

m = Multiple Record flag (Bit 4)
- m = 0, Single Record
- m = 1, Multiple Records

a_0 = Data Address Mark (Bit 0)
- a_0 = 0, F8 (Data Mark)
- a_0 = 1, F8 (Deleted Data Mark)

E = 15 ms Delay (2MHz)
- E = 1, 15 ms delay
- E = 0, no 15 ms delay

(F_2) S = Side Select Flag (1791/3 only)
- S = 0, Compare for Side 0
- S = 1, Compare for Side 1

(F_1) C = Side Compare Flag (1791/3 only)
- C = 0, disable side select compare
- C = 1, enable side select compare

(F_1) S = Side Select Flag
- (Bit 1, 1795/7 only)
- S = 0 Update SSO to 0
- S = 1 Update SSO to 1

(F_2) b = Sector Length Flag
- (Bit 3, 1975/7 only)

	Sector Length Field			
	00	01	10	11
b = 0	256	512	1024	128
b = 1	128	256	512	1024

Table 5. FLAG SUMMARY

TYPE IV COMMAND

I_3-I_0 = Interrupt Condition flags (Bits 3-0)
- I_0 = 1, Not-Ready to Ready Transition
- I_1 = 1, Ready to Not-Ready Transition
- I_2 = 1, Index Pulse
- I_3 = 1, Immediate Interrupt
- I_3 -I_0 = 0, Terminate with no Interrupt

TYPE I COMMANDS

The Type I Commands include the Restore, Seek, Step, Step-In, and Step-Out commands. Each of the Type I Commands contains a rate field (r_1r_0), which determines the stepping motor rate as defined in Table 1.

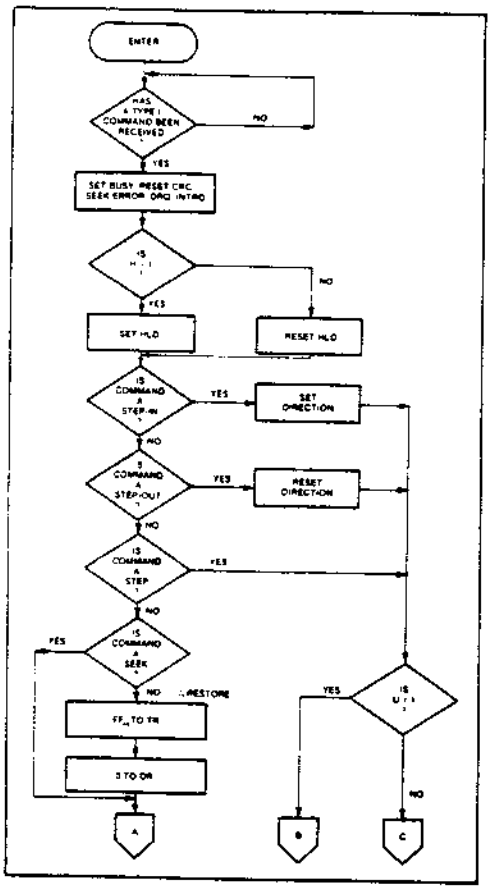
The Type I Commands contain a head load flag (h) which determines if the head is to be loaded at the beginning of the command. If h = 1, the head is loaded at the beginning of the command (HLD output is made active). If h = 0, HLD is deactivated. Once the head is loaded, the head will remain engaged until the FD179X receives a command that specifically disengages the head. If the FD179X is idle (busy = 0) for 15 revolutions of the disk, the head will be automatically disengaged (HLD made inactive).

The Type I Commands also contain a verification (V) flag which determines if a verification operation is to take place on the destination track. If V = 1, a verification is performed, if V = 0, no verification is performed.

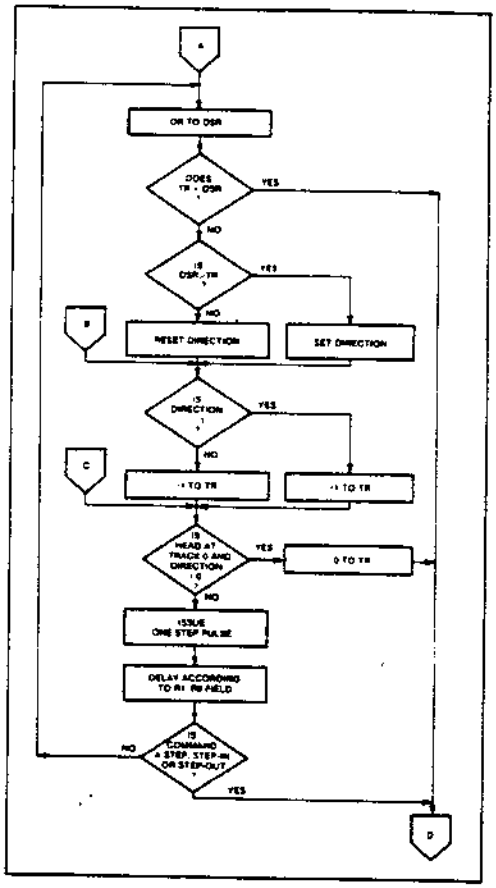
During verification, the head is loaded and after an internal 15 ms delay, the HLT input is sampled. When HLT is active (logic true), the first encountered ID field is read off the disk. The track address of the ID field is then compared to the Track Register; if there is a match and a valid ID CRC, the verification is complete, an interrupt is generated and the Busy status bit is reset. If there is not a match but there is valid ID CRC, an interrupt is generated, and Seek Error Status bit (Status bit 4) is set and the Busy status bit is reset. If there is a match but not a valid CRC, the CRC error status bit is set (Status bit 3), and the next encountered ID field is read from the disk for the verification operation. If an ID field with a valid CRC cannot be found after four revolutions of the disk, the FD179X terminates the operation and sends an interrupt, (INTRQ).

The Step, Step-In, and Step-Out commands contain an Update flag (U). When U = 1, the track register is updated by one for each step. When U = 0, the track register is not updated.

On the 1795/7 devices, the SSO output is not affected during Type 1 commands, and an internal side compare does not take place when the (V) Verify Flag is on.



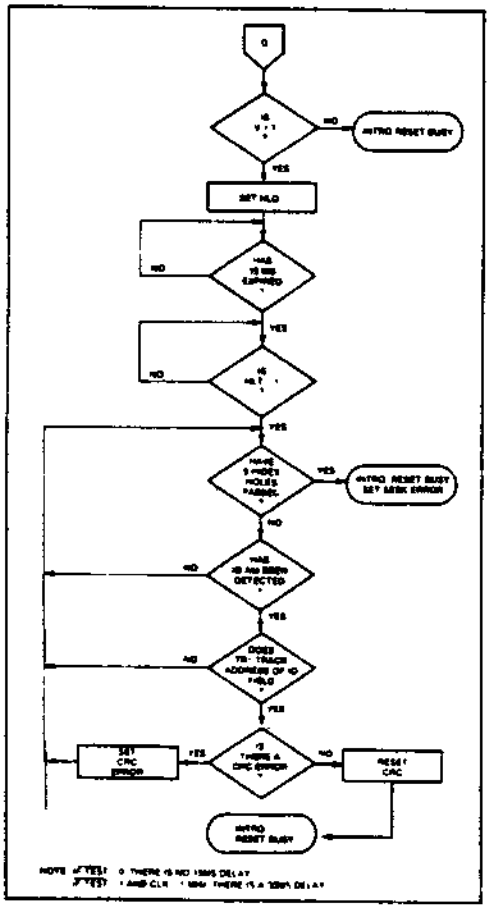
TYPE I COMMAND FLOW



TYPE I COMMAND FLOW

RESTORE (SEEK TRACK 0)

Upon receipt of this command the Track 00 ($\overline{TR00}$) input is sampled. If $\overline{TR00}$ is active low indicating the Read-Write head is positioned over track 0, the Track Register is loaded with zeroes and an interrupt is generated. If $\overline{TR00}$ is not active low, stepping pulses (pins 15 to 16) at a rate specified by the rate field are issued until the $\overline{TR00}$ input is activated. At this time the Track Register is loaded with zeroes and an interrupt is generated. If the $\overline{TR00}$ input does not go active low after 255 stepping pulses, the FD179X terminates operation, interrupts, and sets the Seek error status bit. A verification operation takes place if the V flag is set. The h bit allows the head to be loaded at the start of command. Note that the Restore command is executed when $\overline{MR}$ goes from an active to an inactive state.



TYPE I COMMAND FLOW

SEEK

This command assumes that the Track Register contains the track number of the current position of the Read-Write head and the Data Register contains the desired track number. The FD179X will update the Track register and issue stepping pulses in the appropriate direction until the contents of the Track register are equal to the contents of the Data Register (the desired track location). A verification operation takes place if the V flag is on. The h bit allows the head to be loaded at the start of the command. An interrupt is generated at the completion of the command.

STEP

Upon receipt of this command, the FD179X issues one stepping pulse to the disk drive. The stepping motor direction is the same as in the previous step command. After a delay determined by the rate field, a verification takes place if the V flag is on. If the u flag is on, the Track Register is updated. The h bit allows the head to be loaded at the start of the command. An interrupt is generated at the completion of the command.

STEP-IN

Upon receipt of this command, the FD179X issues one stepping pulse in the direction towards track 76. If the u flag is on, the Track Register is incremented by one. After a delay determined by the rate field, a verification takes place if the V flag is on. The h bit allows the head to be loaded at the start of the command. An interrupt is generated at the completion of the command.

STEP-OUT

Upon receipt of this command, the FD179X issues one stepping pulse in the direction towards track 0. If the u flag is on, the Track Register is decremented by one. After a delay determined by the rate field, a verification takes place if the V flag is on. The h bit allows the head to be loaded at the start of the command. An interrupt is generated at the completion of the command.

TYPE II COMMANDS

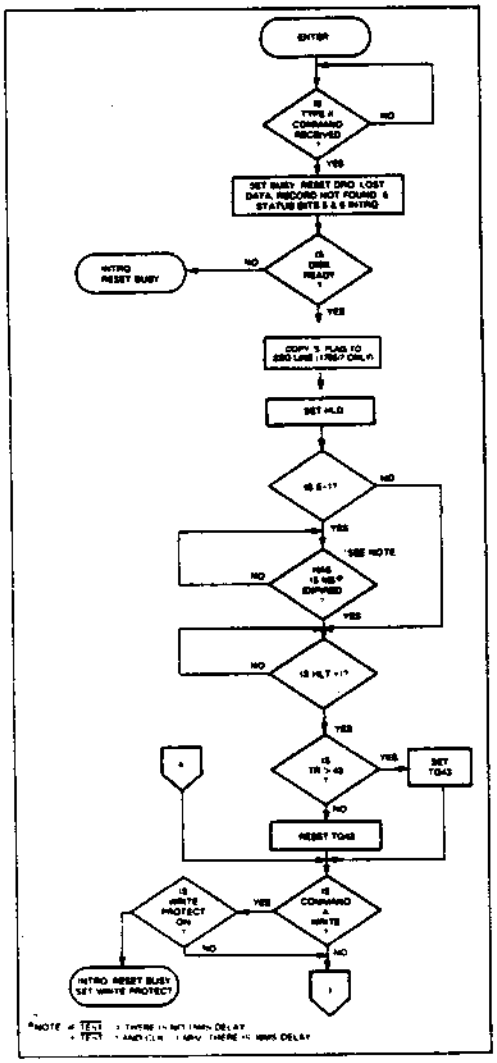
The Type II Commands are the Read Sector and Write Sector commands. Prior to loading the Type II Command into the Command Register, the computer must load the Sector Register with the desired sector number. Upon receipt of the Type II command, the busy status Bit is set. If the E flag = 1 (this is the normal case) HLD is made active and HLT is sampled after a 15 msec delay. If the E flag is 0, the head is loaded and HLT sampled with no 15 msec delay. The ID field and Data Field format are shown on page 13.

When an ID field is located on the disk, the FD179X compares the Track Number on the ID field with the Track Register. If there is not a match, the next en-

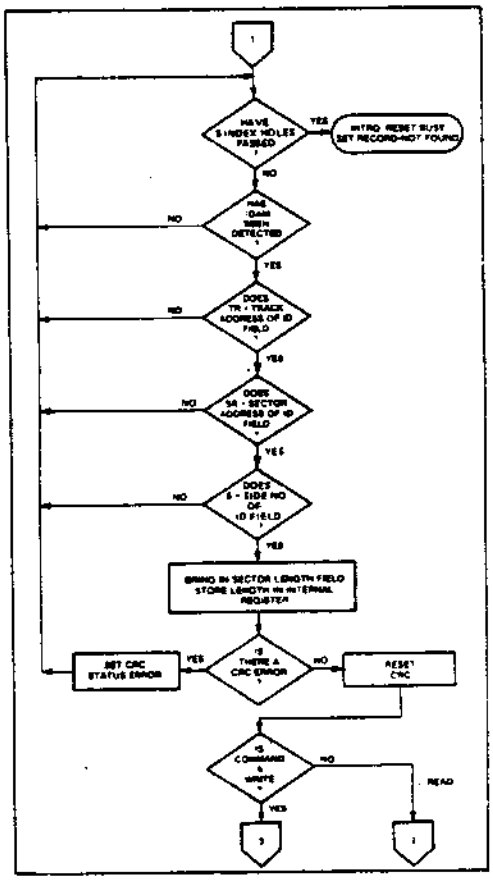
countered ID field is read and a comparison is again made. If there was a match, the Sector Number of the ID field is compared with the Sector Register. If there is not a Sector match, the next encountered ID field is read off the disk and comparisons again made. If the ID field CRC is correct, the data field is then located and will be either written into, or read from depending upon the command. The FD179X must find an ID field with a Track number, Sector number, side number, and CRC within four revolutions of the disk; otherwise, the Record not found status bit is set (Status bit 3) and the command is terminated with an interrupt.

Sector Length Table	
Sector Length Field (hex)	Number of Bytes in Sector (decimal)
00	128
01	256
02	512
03	1024

Each of the Type II Commands contains an (m) flag which determines if multiple records (sectors) are to be read or written, depending upon the command. If m = 0, a single sector is read or written and an interrupt is generated at the completion of the command. If m = 1, multiple records are read or written with the sector register internally updated so that an address verification can occur on the next record. The FD179X will continue to read or write multiple records and update the sector register until the sector regis-



TYPE II COMMAND



TYPE II COMMAND

ter exceeds the number of sectors on the track or until the Force Interrupt command is loaded into the Command Register, which terminates the command and generates an interrupt.

If the Sector Register exceeds the number of sectors on the track, the Record-Not-Found status bit will be set.

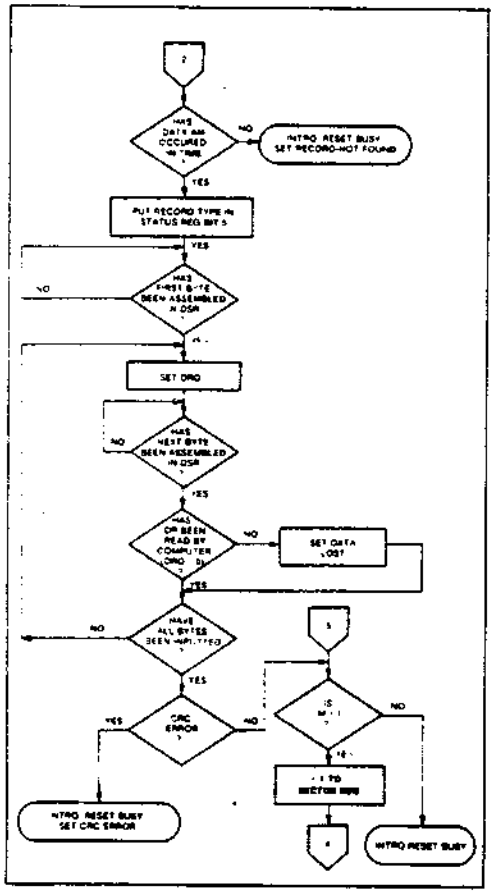
The Type II commands also contain side select compare flags. When C = 0, no side comparison is made. When C = 1, the LSB of the side number is read off the ID Field of the disk and compared with the contents of the (S) flag. If the S flag compares with the side number recorded in the ID field, the 179X continues with the ID search. If a comparison is not made within 5 index pulses, the interrupt line is made active and the Record-Not-Found status bit is set.

The 1795/7 READ SECTOR and WRITE SECTOR commands include a 'b' flag. The 'b' flag, in conjunction with the sector length byte of the ID Field, allows different byte lengths to be implemented in each sector. For IBM compatability, the 'b' flag should be set to a one. The 's' flag allows direct control over the SSO Line (Pin 25) and is set or reset at the beginning of the command, dependent upon the value of this flag.

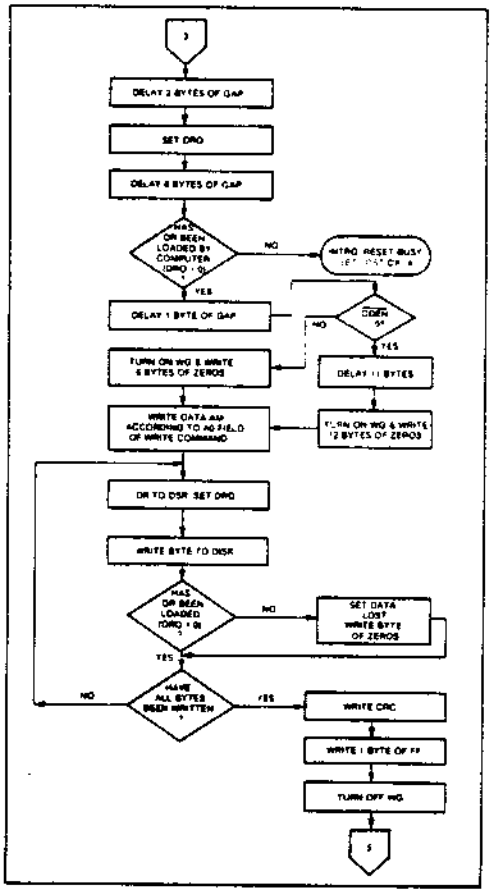
READ SECTOR

Upon receipt of the Read Sector command, the head is loaded, the Busy status bit set, and when an ID field is encountered that has the correct track number, correct sector number, correct side number, and correct CRC, the data field is presented to the computer. The Data Address Mark of the data field must be found within 30 bytes in single density and 43 bytes in double density of the last ID field CRC byte; if not, the Record Not Found status bit is set and the operation is terminated.

When the first character or byte of the data field has been shifted through the DSR, it is transferred to the DR, and DRQ is generated. When the next byte is accumulated in the DSR, it is transferred to the DR and another DRQ is generated. If the Computer has not read the previous contents of the DR before a new character is transferred that character is lost and



TYPE II COMMAND



TYPE II COMMAND

the Lost Data Status bit is set. This sequence continues until the complete data field has been inputted to the computer. If there is a CRC error at the end of the data field, the CRC error status bit is set, and the command is terminated (even if it is a multiple record command).

At the end of the Read operation, the type of Data Address Mark encountered in the data field is recorded in the Status Register (Bit 5) as shown below:

STATUS BIT 5	
1	Deleted Data Mark
0	Data Mark

WRITE SECTOR

Upon receipt of the Write Sector command, the head is loaded (HLD active) and the Busy status bit is set. When an ID field is encountered that has the correct track number, correct sector number, correct side number, and correct CRC, a DRQ is generated. The FD179X counts off 11 bytes in single density and 22 bytes in double density from the CRC field and the Write Gate (WG) output is made active if the DRQ is serviced (i.e., the DR has been loaded by the computer). If DRQ has not been serviced, the command is terminated and the Lost Data status bit is set. If the DRQ has been serviced, the WG is made active and six bytes of zeros in single density and 12 bytes in double density are then written on the disk. At this time the Data Address Mark is then written on the disk as determined by the a_0 field of the command as shown below:

a_0	Data Address Mark (Bit 0)
1	Deleted Data Mark
0	Data Mark

The FD179X then writes the data field and generates DRQ's to the computer. If the DRQ is not serviced in time for continuous writing the Lost Data Status Bit is set and a byte of zeros is written on the disk. The command is not terminated. After the last data byte has been written on the disk, the two-byte CRC is computed internally and written on the disk followed by one byte of logic ones in FM or in MFM. The WG output is then deactivated.

TYPE III COMMANDS

READ ADDRESS

Upon receipt of the Read Address command, the head is loaded and the Busy Status Bit is set. The next encountered ID field is then read in from the disk, and the six data bytes of the ID field are assembled and transferred to the DR, and a DRQ is generated for each byte. The six bytes of the ID field are shown below:

TRACK ADDR	SIDE NUMBER	SECTOR ADDRESS	SECTOR LENGTH	CRC 1	CRC 2
1	2	3	4	5	6

Although the CRC characters are transferred to the computer, the FD179X checks for validity and the CRC error status bit is set if there is a CRC error. The Track Address of the ID field is written into the sector register. At the end of the operation an interrupt is generated and the Busy Status is reset.

READ TRACK

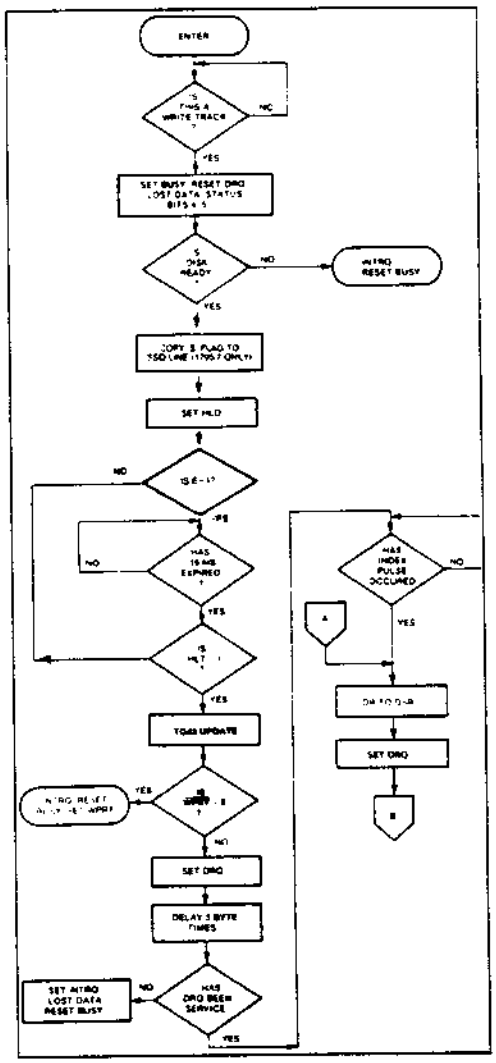
Upon receipt of the Read Track command, the head is loaded and the Busy Status bit is set. Reading starts with the leading edge of the first encountered index pulse and continues until the next index pulse. As each byte is assembled it is transferred to the Data Register and the Data Request is generated for each byte. No CRC checking is performed. Gaps are included in the input data stream. The accumulation of bytes is synchronized to each Address Mark encountered. Upon completion of the command, the interrupt is activated. RG is not activated during the Read Track Command. An internal side compare is not performed during a Read Track.

WRITE TRACK

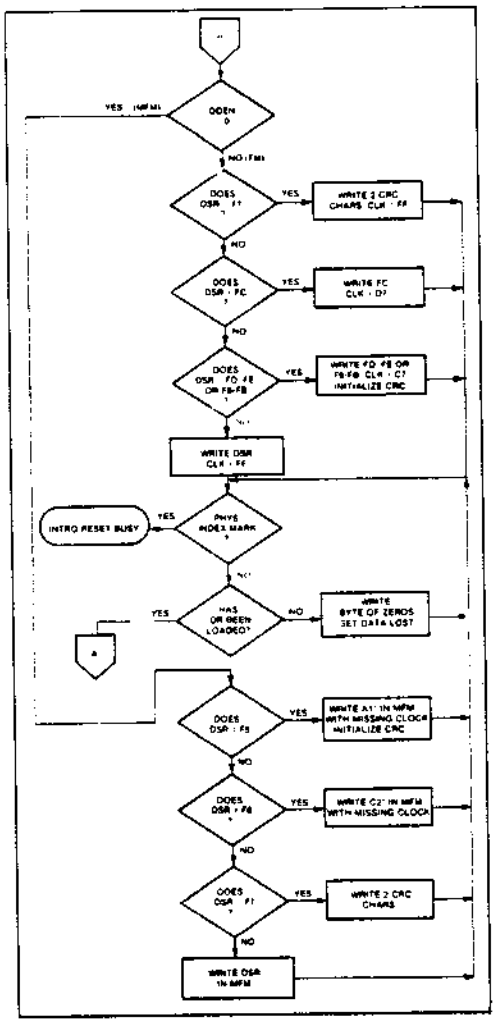
Upon receipt of the Write Track command, the head is loaded and the Busy Status bit is set. Writing starts with the leading edge of the first encountered index pulse and continues until the next index pulse, at which time the interrupt is activated. The Data Request is activated immediately upon receiving the command, but writing will not start until after the first byte has been loaded into the Data Register. If the DR has not been loaded by the time the index pulse is encountered the operation is terminated making the device Not Busy, the Lost Data Status Bit is set, and the Interrupt is activated. If a byte is not present in the DR when needed, a byte of zeros is substituted. Address Marks and CRC characters are written on the disk by detecting certain data byte patterns in the outgoing data stream as shown in the table below. The CRC generator is initialized when any data byte from F8 to FE is about to be transferred from the DR to the DSR in FM or by receipt of F5 in MFM.

GAP III	ID AM	TRACK NUMBER	SIDE NUMBER	SECTOR NUMBER	SECTOR LENGTH	CRC 1	CRC 2	GAP II	DATA AM	DATA FIELD	CRC 1	CRC 2
ID FIELD										DATA FIELD		

In MFM only, IDAM and DATA AM are preceded by three bytes of A1 with clock transition between bits 4 and 5 missing.



TYPE III COMMAND WRITE TRACK



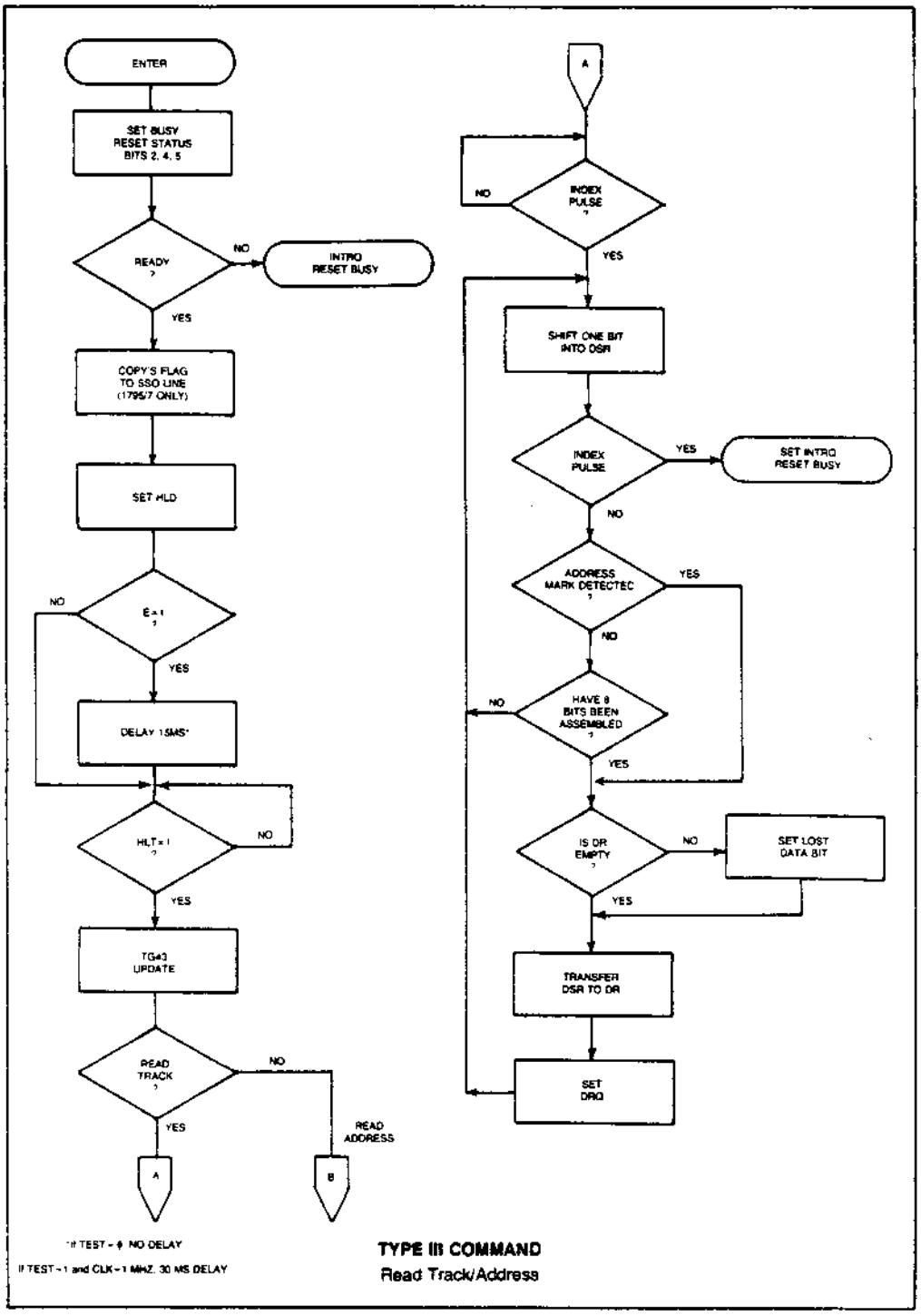
TYPE III COMMAND WRITE TRACK

CONTROL BYTES FOR INITIALIZATION

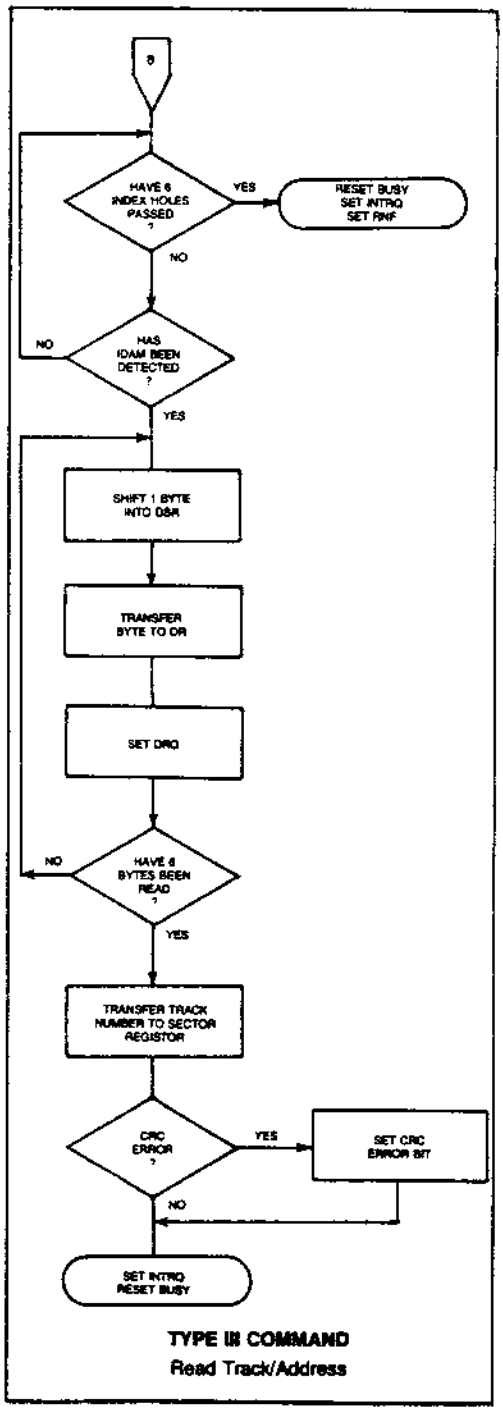
DATA PATTERN IN DR (HEX)	FD179X INTERPRETATION IN FM ($\overline{DDEN}$ = 1)	FD1791/3 INTERPRETATION IN MFM ($\overline{DDEN}$ = 0)
00 thru F4	Write 00 thru F4 with CLK = FF	Write 00 thru F4, in MFM
F5	Not Allowed	Write A1* in MFM, Preset CRC
F6	Not Allowed	Write C2** in MFM
F7	Generate 2 CRC bytes	Generate 2 CRC bytes
F8 thru FB	Write F8 thru FB, Clk = C7, Preset CRC	Write F8 thru FB, in MFM
FC	Write FC with Clk = D7	Write FC in MFM
FD	Write FD with Clk = FF	Write FD in MFM
FE	Write FE, Clk = C7, Preset CRC	Write FE in MFM
FF	Write FF with Clk = FF	Write FF in MFM

*Missing clock transition between bits 4 and 5

**Missing clock transition between bits 3 & 4



TYPE III COMMAND
Read Track/Address



TYPE III COMMAND
Read Track/Address

TYPE IV COMMAND

FORCE INTERRUPT

This command can be loaded into the command register at any time. If there is a current command under execution (Busy Status Bit set), the command will be terminated and an interrupt will be generated when the condition specified in the I_0 through I_3 field is detected. The interrupt conditions are shown below:

I_0 = Not-Ready-To-Ready Transition
I_1 = Ready-To-Not-Ready Transition
I_2 = Every Index Pulse
I_3 = Immediate Interrupt (requires reset, see Note)

NOTE: If $I_0 - I_3 = 0$, there is no interrupt generated but the current command is terminated and busy is reset. *This is the only command that will enable the immediate interrupt to clear on a subsequent Load Command Register or Read Status Register.*

STATUS DESCRIPTION

Upon receipt of any command, except the Force Interrupt command, the Busy Status bit is set and the rest of the status bits are updated or cleared for the new command. If the Force Interrupt Command is received when there is a current command under execution, the Busy status bit is reset, and the rest of the status bits are unchanged. If the Force Interrupt command is received when there is not a current command under execution, the Busy Status bit is reset and the rest of the status bits are updated or cleared. In this case, Status reflects the Type I commands.

The format of the Status Register is shown below:

(BITS)							
7	6	5	4	3	2	1	0
S7	S6	S5	S4	S3	S2	S1	S0

Status varies according to the type of command executed as shown in Table 6.

FORMATTING THE DISK

(Refer to section on Type III commands for flow diagrams)

Formatting the disk is a relatively simple task when operating programmed I/O or when operating under Formatting the disk is accomplished by positioning the R/W head over the desired track number and issuing the Write Track command. Upon receipt of the Write Track command, the FD179X raises the Data Request signal. At this point in time, the user loads the data register with desired data to be written on the disk. For every byte of information to be written on the disk, a data request is generated. This sequence continues from one index mark to the next index mark. Normally, whatever data pattern appears in the data register is written on the disk with a normal clock pattern. However, if the FD179X detects a data pattern of F5 thru FE in the data register, this is interpreted as data address marks with missing clocks or CRC generation. For instance, in FM an FE pattern will be interpreted as an ID address mark (DATA-FE, CLK-C7) and the CRC will be initialized. An F7 pattern will generate two CRC characters in FM or MFM. As a consequence, the patterns F5 thru FE must not appear in the gaps, data fields, or ID fields. Also, CRC's must be generated by an F7 pattern.

Disks may be formatted in IBM 3740 or System 34 formats with sector lengths of 128, 256, 512, or 1024 bytes.

IBM 3740 FORMAT—128 BYTES/SECTOR

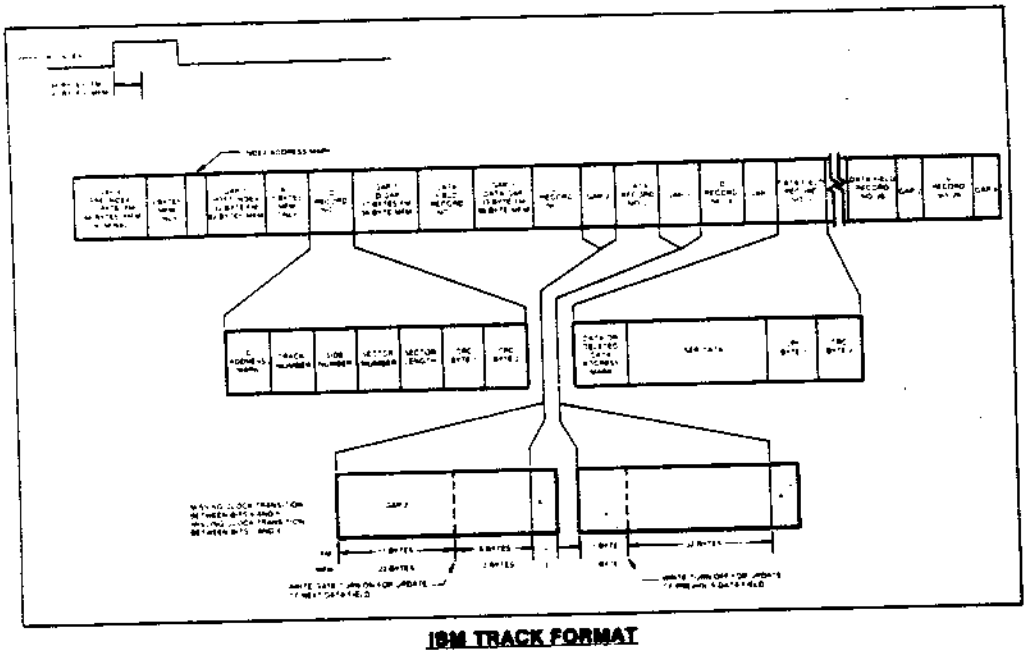
Shown below is the IBM single-density format with 128 bytes/sector. In order to format a diskette, the user must issue the Write Track command, and load the data register with the following values. For every byte to be written, there is one data request.

NUMBER OF BYTES	HEX VALUE OF BYTE WRITTEN
40	FF (or 00)[1]
6	00
1	FC (Index Mark)
26	FF (or 00)
6	00
1	FE (ID Address Mark)
1	Track Number
1	Side Number (00 or 01)
1	Sector Number (1 thru 1A)
1	00
1	F7 (2 CRC's written)
11	FF (or 00)
6	00
1	FB (Data Address Mark)
128	Data (IBM uses E5)
1	F7 (2 CRC's written)
27	FF (or 00)
247**	FF (or 00)

*Write bracketed field 26 times

**Continue writing until FD179X interrupts out. Approx. 247 bytes.

1-Optional '00' on 1795/7 only.



IBM TRACK FORMAT

IBM SYSTEM 34 FORMAT-256 BYTES/SECTOR

Shown below is the IBM dual-density format with 256 bytes/sector. In order to format a diskette the user must issue the Write Track command and load the data register with the following values. For every byte to be written, there is one data request.

NUMBER OF BYTES	HEX VALUE OF BYTE WRITTEN
80	4E
12	00
3	F6
1	FC (Index Mark)
50*	4E
12	00
3	F5
1	FE (ID Address Mark)
1	Track Number (0 thru 4C)
1	Side Number (0 or 1)
1	Sector Number (1 thru 1A)
1	01
1	F7 (2 CRCs written)
22	4E
12	00
3	F5
1	FB (Data Address Mark)
256	DATA
1	F7 (2 CRCs written)
54	4E
598**	4E

* Write bracketed field 26 times
**Continue writing until FD179X interrupts out. Approx. 598 bytes.

1. NON-IBM FORMATS

Variations in the IBM format are possible to a limited extent if the following requirements are met: sector size must be a choice of 128, 256, 512, or 1024 bytes; gap size must be according to the following table. Note that the Index Mark is not required by the 179X. The minimum gap sizes shown are that which is required by the 179X, with PLL lock-up time, motor speed variation, etc., adding additional bytes.

	FM	MFM
Gap I	16 bytes FF	32 bytes 4E
Gap II	11 bytes FF	22 bytes 4E
*	6 bytes 00	12 bytes 00 3 bytes A1
Gap III	10 bytes FF	24 bytes 4E 3 bytes A1
**	4 bytes 00	8 bytes 00
Gap IV	16 bytes FF	16 bytes 4E

*Byte counts must be exact.
**Byte counts are minimum, except exactly 3 bytes of A1 must be written.

ELECTRICAL CHARACTERISTICS

MAXIMUM RATINGS

V_{DD} With Respect to V_{SS} (Ground) +15 to −0.3V

Max. Voltage to Any Input With Respect to V_{SS} +15 to −0.3V

Operating Temperature 0°C to 70°C

Storage Temperature −55°C to +125°C

V_{DD} = ID ma Nominal V_{CC} = 35 ma Nominal

OPERATING CHARACTERISTICS (DC)

TA = 0°C to 70°C, V_{DD} = + 12V ± .6V, V_{SS} = OV, V_{CC} = + 5V ± .25V

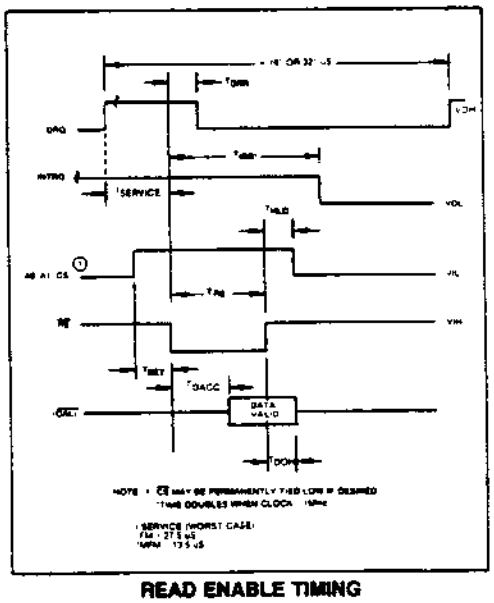
SYMBOL	CHARACTERISTIC	MIN.	MAX.	UNITS	CONDITIONS
I_{IL}	Input Leakage		10	μA	V_{IN} = V_{DD}
I_{OL}	Output Leakage		10	μA	V_{OUT} = V_{DD}
V_{IH}	Input High Voltage	2.6		V	
V_{IL}	Input Low Voltage		0.8	V	
V_{OH}	Output High Voltage	2.8		V	I_O = −100 μA
V_{OL}	Output Low Voltage		0.45	V	I_O = 1.6 mA
P_D	Power Dissipation		0.5	W	

TIMING CHARACTERISTICS

T_A = 0°C to 70°C, V_{DD} = + 12V ± .6V, V_{SS} = 0V, V_{CC} = +5V ± .25V

READ ENABLE TIMING

SYMBOL	CHARACTERISTIC	MIN.	TYP.	MAX.	UNITS	CONDITIONS
TSET	Setup ADDR & CS to $\overline{RE}$	50			nsec	
THLD	Hold ADDR & CS from $\overline{RE}$	10			nsec	
TRE	$\overline{RE}$ Pulse Width	400			nsec	C_L = 50 pf
TDRR	DRQ Reset from $\overline{RE}$		400	500	nsec	
TIRR	INTRQ Reset from $\overline{RE}$		500	3000	nsec	See Note 5
TDACC	Data Access from $\overline{RE}$			350	nsec	C_L = 50 pf
TDOH	Data Hold From $\overline{RE}$	50		150	nsec	C_L = 50 pf



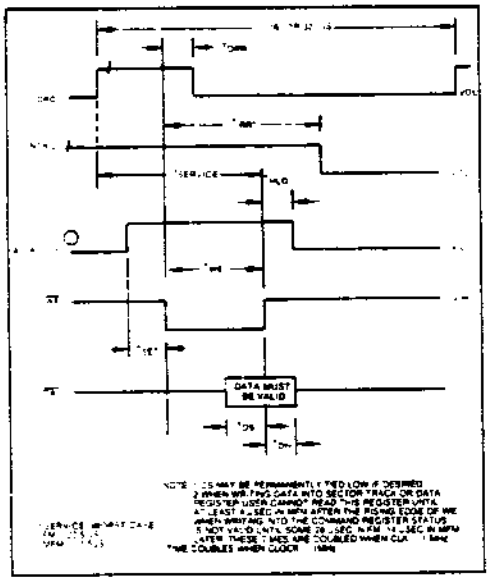
READ ENABLE TIMING

WRITE ENABLE TIMING

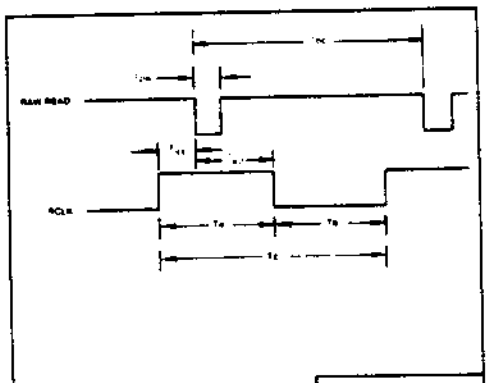
SYMBOL	CHARACTERISTIC	MIN.	TYP.	MAX.	UNITS	CONDITIONS
TSET	Setup ADDR & CS to $\overline{WE}$	50			nsec	
THLD	Hold ADDR & CS from $\overline{WE}$	10			nsec	
TWE	$\overline{WE}$ Pulse Width	350			nsec	
TDRR	DRQ Reset from $\overline{WE}$		400	500	nsec	
TIRR	INTRQ Reset from $\overline{WE}$		500	3000	nsec	See Note 5
TDS	Data Setup to $\overline{WE}$	250			nsec	
TDH	Data Hold from $\overline{WE}$	70			nsec	

INPUT DATA TIMING:

SYMBOL	CHARACTERISTIC	MIN.	TYP.	MAX.	UNITS	CONDITIONS
Tpw	$\overline{Raw\ Read}$ Pulse Width	100	200		nsec	See Note 1
tbc	$\overline{Raw\ Read}$ Cycle Time		1500		nsec	1800 ns @ 70°C
Tc	RCLK Cycle Time		1500		nsec	1800 ns @ 70°C
T_{x1}	RCLK hold to $\overline{Raw\ Read}$	40			nsec	See Note 1
T_{x2}	$\overline{Raw\ Read}$ hold to RCLK	40			nsec	



WRITE ENABLE TIMING

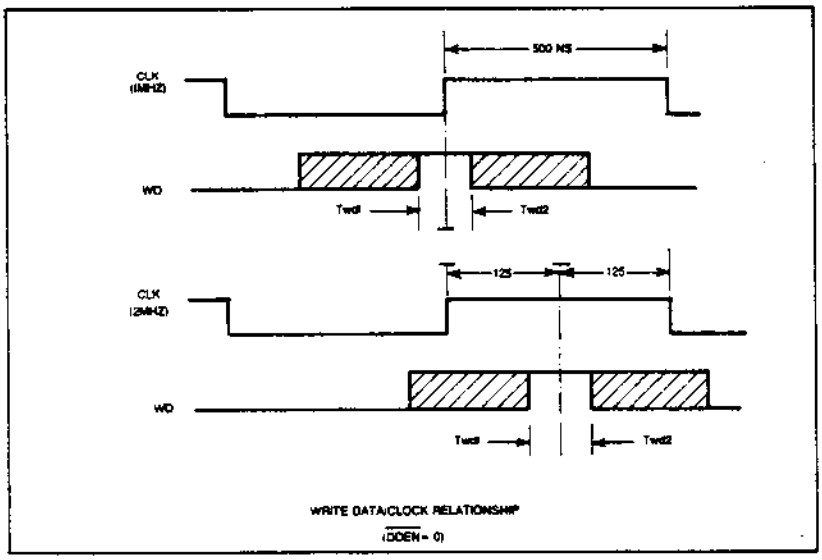


DISKETTE	MODE	$\overline{DDEN}$	CLK	NOMINAL T_1	NOMINAL T_2	NOMINAL T_3
8"	MFM	0	2 MHz	1 µs	1 µs	2 µs
8"	FM	1	2 MHz	2 µs	2µs	4 µs
5"	MFM	0	1 MHz	2 µs	2µs	4 µs
5"	FM	1	1 MHz	4 µs	4 µs	8 µs

INPUT DATA TIMING

WRITE DATA TIMING: (ALL TIMES DOUBLE WHEN CLK = 1 MHz)

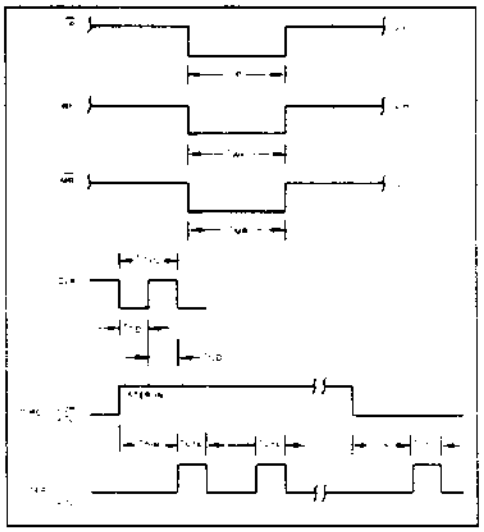
SYMBOL	CHARACTERISTICS	MIN.	TYP.	MAX.	UNITS	CONDITIONS
Twp	Write Data Pulse Width	450	500	550	nsec	FM
		150	200	250	nsec	MFM
Twg	Write Gate to Write Data		2		µsec	FM
			1		µsec	MFM
Tbc	Write data cycle Time		2,3, or 4		µsec	±CLK Error
Ts	Early (Late) to Write Data	125			nsec	MFM
Th	Early (Late) From Write Data	125			nsec	MFM
Twf	Write Gate off from WD		2		µsec	FM
			1		µsec	MFM
Twd1	WD Valid to Clk	100			nsec	CLK=1 MHZ
		50			nsec	CLK=2 MHZ
Twd2	WD Valid after CLK	100			nsec	CLK=1 MHZ
		30			nsec	CLK=2 MHZ



WRITE DATA TIMING

MISCELLANEOUS TIMING:

SYMBOL	CHARACTERISTIC	MIN.	TYP.	MAX.	UNITS	CONDITIONS
TCD_1	Clock Duty (low)	230	250	20000	nsec	
TCD_2	Clock Duty (high)	200	250	20000	nsec	
TSTP	Step Pulse Output	2 or 4			μsec	See Note 5
TDIR	Dir Setup to Step		12		μsec	± CLK ERROR
TMR	Master Reset Pulse Width	50			μsec	
TIP	Index Pulse Width	10			μsec	See Note 5
TWF	Write Fault Pulse Width	10			μsec	



MISCELLANEOUS TIMING

NOTES:

1. Pulse width on RAW READ (Pin 27) is normally 100-300 ns. However, pulse may be any width if pulse is entirely within window. If pulse occurs in both windows, then pulse width must be less than 300 ns for MFM at CLK = 2 MHz and 600 ns for FM at 2 MHz. Times double for 1 MHz.
2. A PPL Data Separator is recommended for 8" MFM.
3. tbc should be 2 μs, nominal in MFM and 4 μs nominal in FM. Times double when CLK = 1 MHz.
4. RCLK may be high or low during $\overline{\text{RAW READ}}$ (Polarity is unimportant).
5. Times double when clock = 1 MHz.

Table 6. STATUS REGISTER SUMMARY

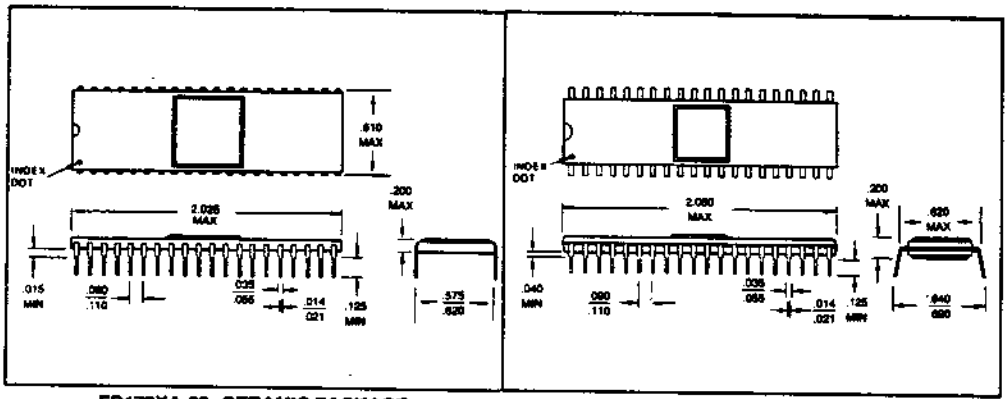
BIT	ALL TYPE I COMMANDS	READ ADDRESS	READ SECTOR	READ TRACK	WRITE SECTOR	WRITE TRACK
S7	NOT READY	NOT READY	NOT READY	NOT READY	NOT READY	NOT READY
S6	WRITE PROTECT	0	0	0	WRITE PROTECT	WRITE PROTECT
S5	HEAD LOADED	0	RECORD TYPE	0	WRITE FAULT	WRITE FAULT
S4	SEEK ERROR	RNF	RNF	0	RNF	0
S3	CRC ERROR	CRC ERROR	CRC ERROR	0	CRC ERROR	0
S2	TRACK 0	LOST DATA	LOST DATA	LOST DATA	LOST DATA	LOST DATA
S1	INDEX	DRQ	DRQ	DRQ	DRQ	DRQ
S0	BUSY	BUSY	BUSY	BUSY	BUSY	BUSY

STATUS FOR TYPE I COMMANDS

BIT NAME	MEANING
S7 NOT READY	This bit when set indicates the drive is not ready. When reset it indicates that the drive is ready. This bit is an inverted copy of the Ready input and logically 'ored' with MR.
S6 PROTECTED	When set, indicates Write Protect is activated. This bit is an inverted copy of $\overline{WRPT}$ input.
S5 HEAD LOADED	When set, it indicates the head is loaded and engaged. This bit is a logical "and" of HLD and HLT signals.
S4 SEEK ERROR	When set, the desired track was not verified. This bit is reset to 0 when updated.
S3 CRC ERROR	CRC encountered in ID field.
S2 TRACK 00	When set, indicates Read/Write head is positioned to Track 0. This bit is an inverted copy of the $\overline{TR00}$ input.
S1 INDEX	When set, indicates index mark detected from drive. This bit is an inverted copy of the $\overline{IP}$ input.
S0 BUSY	When set command is in progress. When reset no command is in progress.

STATUS FOR TYPE II AND III COMMANDS

BIT NAME	MEANING
S7 NOT READY	This bit when set indicates the drive is not ready. When reset, it indicates that the drive is ready. This bit is an inverted copy of the Ready input and 'ored' with MR. The Type II and III Commands will not execute unless the drive is ready.
S6 WRITE PROTECT	On Read Record: Not Used. On Read Track: Not Used. On any Write: It indicates a Write Protect. This bit is reset when updated.
S5 RECORD TYPE/ WRITE FAULT	On Read Record: It indicates the record-type code from data field address mark. 1 = Deleted Data Mark. 0 = Data Mark. On any Write: It indicates a Write Fault. This bit is reset when updated.
S4 RECORD NOT FOUND (RNF)	When set, it indicates that the desired track, sector, or side were not found. This bit is reset when updated.
S3 CRC ERROR	If S4 is set, an error is found in one or more ID fields; otherwise it indicates error in data field. This bit is reset when updated.
S2 LOST DATA	When set, it indicates the computer did not respond to DRQ in one byte time. This bit is reset to zero when updated.
S1 DATA REQUEST	This bit is a copy of the DRQ output. When set, it indicates the DR is full on a Read Operation or the DR is empty on a Write operation. This bit is reset to zero when updated.
S0 BUSY	When set, command is under execution. When reset, no command is under execution.



FD179XA-02 CERAMIC PACKAGE

FD179XB-02 PLASTIC PACKAGE

This is a preliminary specification with tentative device parameters and may be subject to change after final product characterization is completed.

WESTERN DIGITAL CORPORATION | 3128 REDHILL AVENUE, BOX 2180
NEWPORT BEACH, CA 92663 (714) 557-3550,TWX 910-595-1139

APPENDIX D

TECHNICAL INFORMATION

TABLE D-1 2422 SPECIFICATIONS

BOARD MEASUREMENTS:

Board: 10" L x 5" W
Connector: 6.35" L x .3" W (2.125" from right of board)
0.125" pin spacing
Component Height: less than .5"
Weight: approximately 11 ounces

POWER

Supply: Unregulated +8, +16, -16 volts
Maximum power draw: .800 amps at +8 volts
Power Dissipation: less than 8 watts

ENVIRONMENTAL REQUIREMENTS

Temperature: 0 to 70 degrees Celsius
Humidity: 0 to 90% noncondensing

PARTS LIST

QTY	REF NO.	DESCRIPTION	CCS PART NO.*
Capacitors			
2	C1,C8	56pf 500v 10% Mica	42215-55605
1	C2	.47uf 50v 20% Monolythic	42034-24746
4	C3,4,9,10	4.7 35v 20% Tantalum	42804-54756
1	C5	.01uf 50v 20% Ceramic Disk	42142-21036
1	C6	20pf 500v 10% Mica	42215-52005
1	C7	10pf 500v 10% Mica	42215-51005
4	C11,12,13,14	.1uf 50v 20% Monolythic	42034-21046
ICs			
1	U1	7805, +5v Regulator	32000-07805
1	U2	78L12, +12v Regulator	32000-17812
2	U3,29	74LS123	30000-00132
4	U4,18,30,43	74LS74	30000-00074
1	U5	74LS03	30000-00003
2	U6,27	74LS04	30000-00004
2	U7,33	74LS00	30000-00000
1	U8	FD1793	31900-01793
1	U9	7407	30200-07407
1	U10	75468	30300-00468
1	U11	75368	30200-74368
1	U12	74LS175	30000-00175
1	U13	74LS273	30000-00273
1	U14	74LS14	30000-00014
1	U15	74LS197	30000-00197
1	U16	74LS153	30000-00153
1	U17	74LS164	30000-00164
1	U19	74LS165	30000-00165
2	U20,32	74LS10	30000-00010
2	U22,23	5623 (74S287) 256 x 4 ROM	30900-05623
1	U24	2716, 2K x 8 EPROM	31900-02716
4	U25,26,38,39	74LS244	30000-00244
1	U28	7404	30200-07404
2	U31,34	74LS32	30000-00032
3	U35,36,37	74LS367	30000-00367
1	U40	74LS30	30000-00030
2	U41,42	75451	30300-00451
1	U44	74LS08	30000-00008
1	U45	74LS139	30000-00139

* Use CCS part numbers when ordering spares or replacements

PARTS LIST CONTINUED

QTY	REF NO.	DESCRIPTION	CCS PART NO.*
3	R1,2,3,	220 ohm 1/4W 5%	40002-02215
1	R4	7.5K 1/4W 5%	40002-07525
1	R5	1K 1/4W 5%	40002-01025
1	R6	220K 1/4W 5%	40002-02245
2	R7,8	470 ohm 1/4W 5%	40002-04715
1	R9	11K 1/4W 5%	40002-01135
1	R10	4.7K 1/4W 5%	40002-04725
1	Z1	150 ohm x 7 20% SIP Network	40930-71516
4	Z2,3,4,5	2.7K x 7 20% SIP Network	40930-72726
Sockets			
14	XU3,10-12,16, 19,21-23,29, 35-37,45	16-Pin Sockets	58102-00160
20	XU4-7,9,14,15, 17,18,20,27, 28,30-34,40 43,44	14-Pin Sockets	58102-00140
1	XU8	40-Pin Socket	58102-00400
5	XU13,25,26,38, 39	20-Pin Sockets	58102-00200
1	XU24	24-Pin Socket	58102-00240
2	XU41,42	8-Pin Socket	58102-00080
Miscellaneous			
3	CR1-3	LEDs, Rectangular Red	37400-00001
1	J1	Header, Right Angle 2 x 17-Pin	56005-02017
1	J2	Header, Right Angle 2 x 25-Pin	56005-02025
5	W1,10,11,15, 16	Header Strip, 1 x 3-Pin	56004-01003
8	W2-9	Header Strip, 1 x 2-Pin	56004-01002
3	W12,13,14	Header Strip, 1 x 4-Pin	56004-01004
1	Y1	16 MHz Crystal .01%	48231-60003
1	-	Heatsink, 220, .5"	60022-00001
9	-	Berg jumper plugs	56200-00001
1	-	Screw, 6-32 x 5/16"	71006-32051
1	-	Nut, Hex Kep 6-32	73006-32001
2	-	PCB Extractor, Non-locking	60010-00001
2	-	Roll Pin Extractor Mounting	60010-00000
1	-	PC Board, Rev A	02422-00002
1	-	Manual, Rev A	89000-02422

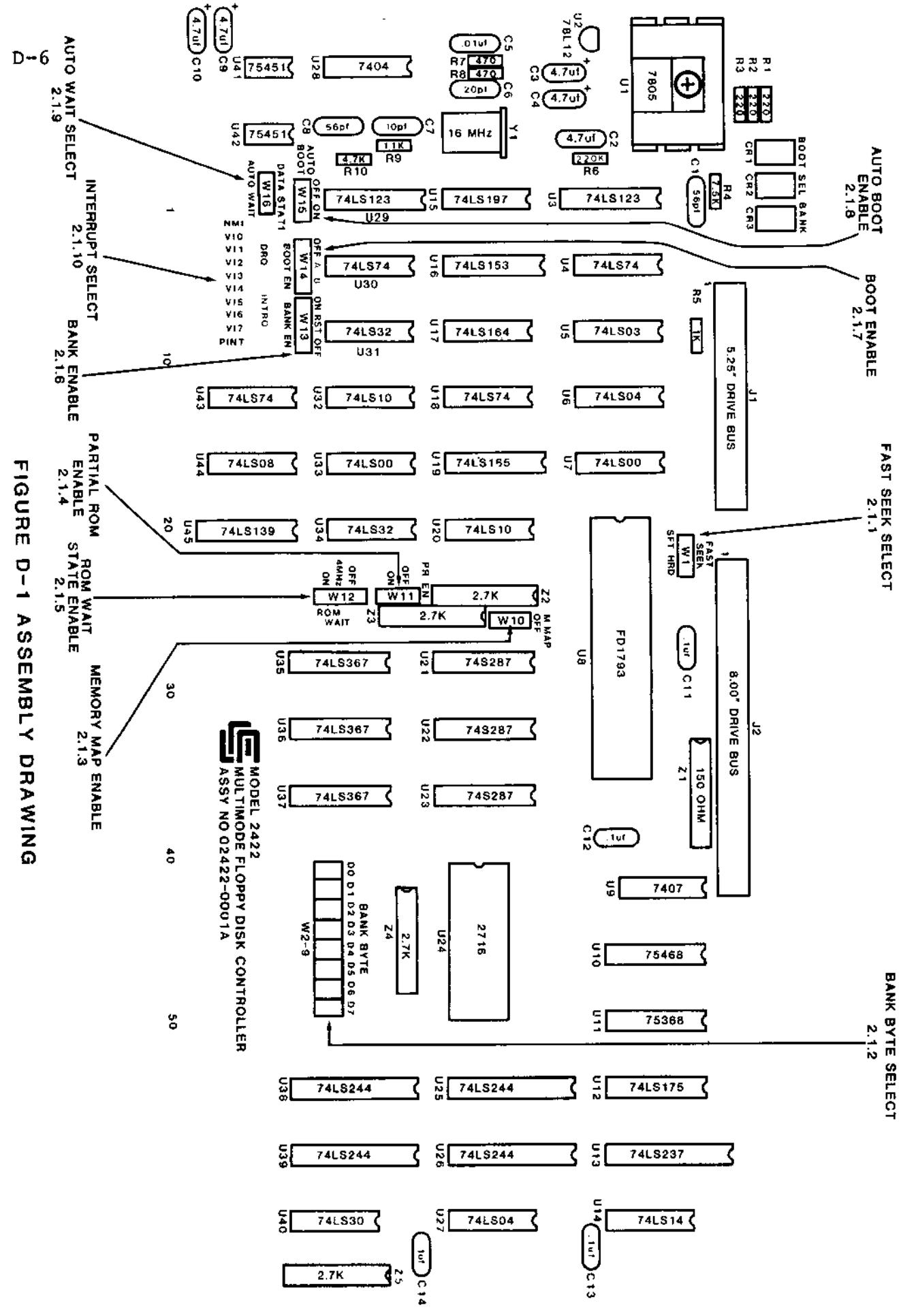
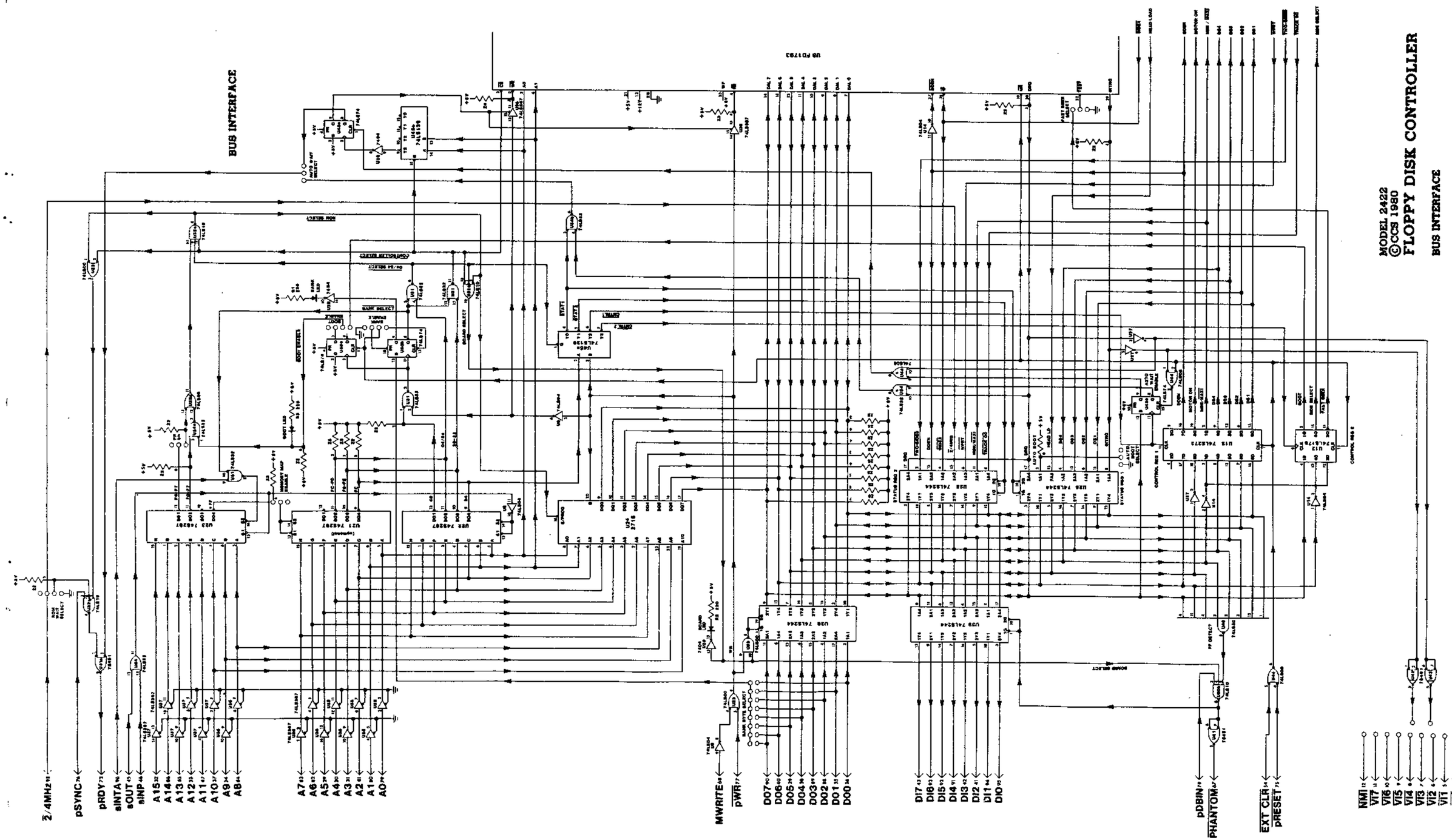
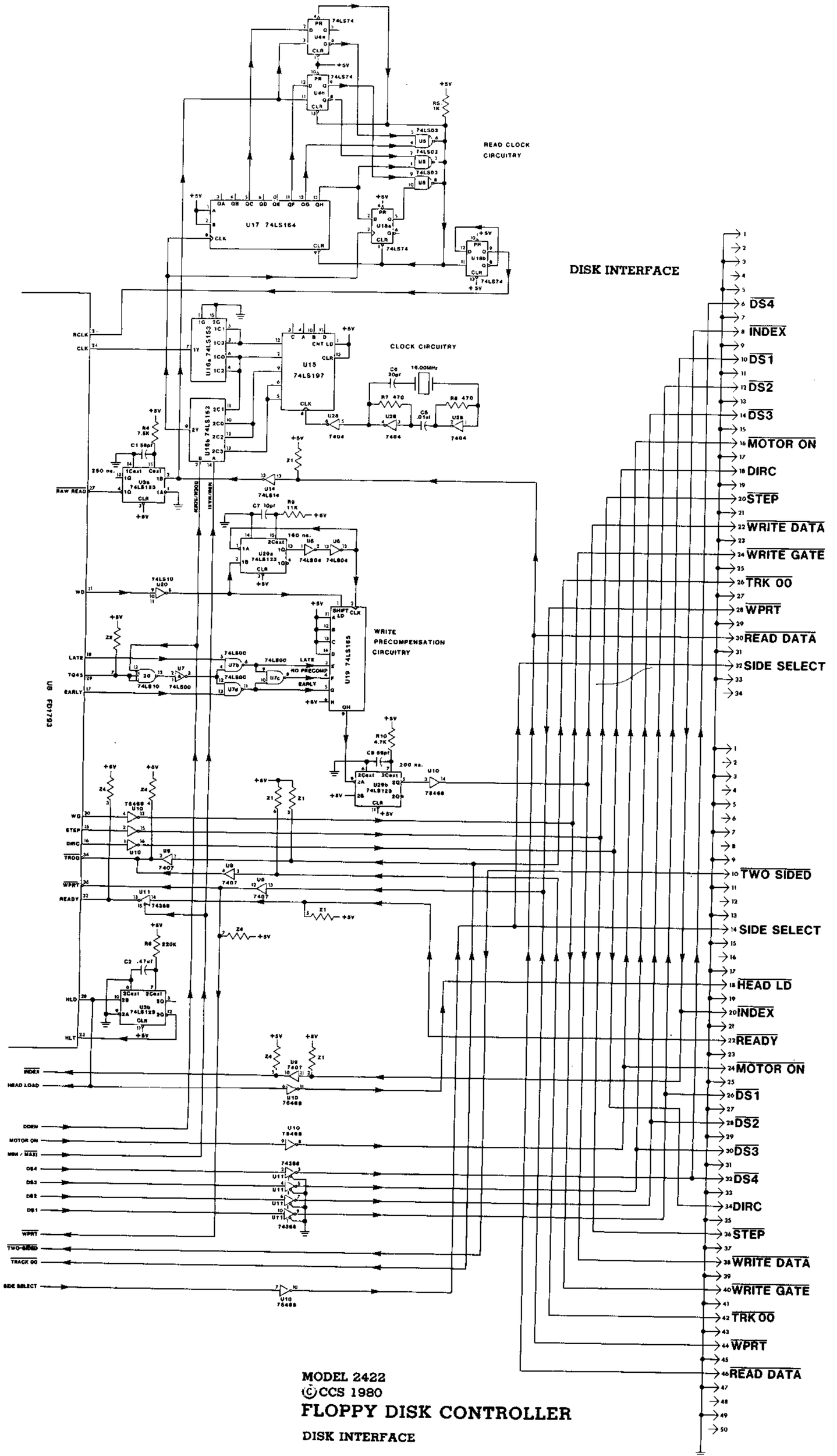


FIGURE D-1 ASSEMBLY DRAWING


BUS INTERFACE
MODEL 2422
©CCS 1980
FLOPPY DISK CONTROLLER
BUS INTERFACE
2/4MHz
pSYNC
pRDY
sINTA
sOUT
sINP
A15
A14
A13
A12
A11
A10
A9
A8
A7
A6
A5
A4
A3
A2
A1
A0
MWRITE
pWR
D07
D06
D05
D04
D03
D02
D01
D00
D17
D16
D15
D14
D13
D12
D11
D10
pDBIN
PHANTOM
EXT CLR
pRESET
NMI
VI7
VI6
VI5
VI4
VI3
VI2
VI1
VI0
pINT



MODEL 2422

FLOPPY DISK CONTROLLER

DISK INTERFACE

APPENDIX E

FIRMWARE LISTING

```
CP/M MACRO ASSEM 2.0    #001    DISK MOSS 2.2 MONITOR

                        ;
                                TITLE   'DISK MOSS 2.2 MONITOR'
                                MACLIB  Z80
                                PAGE    68
                        ;
                        ; DISK MOSS MONITOR (VERSION 2.2)
                        ;
                        ; 14 JUNE 1980
                        ; ALL RIGHTS RESERVED BY ROBERT B. MASON
                        ;
F000                    MOSS:   ORG     0F000H
F000 =                  ROM:    EQU     0F000H  ;ROM START ADDRESS
0000 =                  WSVEC:  EQU     0       ;VECTOR FOR WARM RESTART
0002 =                  NBKPTS: EQU     2       ;NUMBER OF BREAKPOINTS
0013 =                  CTRLS:  EQU     13H     ;ASCII DC3
000D =                  CR:     EQU     0DH     ;ASCII CARRIAGE RETURN
000A =                  LF:     EQU     0AH     ;ASCII LINE FEED
000C =                  FMFD:   EQU     0CH     ;ASCII FORM FEED
0007 =                  BELL:   EQU     7       ;ASCII CNTRL CHAR TO RING THE BELL
0003 =                  IOBYTE: EQU     3       ;ADDRESS OF I/O CONTROL BYTE
0020 =                  SDATA:  EQU     20H     ;SERIAL DATA PORT BASE ADDRESS
0021 =                  SINTEN: EQU     SDATA+1 ;SERIAL INTERRUPT ENABLE REGISTER
0022 =                  SIDENT: EQU     SDATA+2 ;SERIAL INTERRUPT IDENTIFICATION REGISTER
0023 =                  SLCTRL: EQU     SDATA+3 ;SERIAL LINE CONTROL REGISTER
0024 =                  SMDMCT: EQU     SDATA+4 ;SERIAL MODEM CONTROL REGISTER
0025 =                  SLSTAT: EQU     SDATA+5 ;SERIAL LINE STATUS REGISTER
0026 =                  SMDMST: EQU     SDATA+6 ;SERIAL MODEM STATUS REGISTER
                        ;
0006 =                  SPSV:   EQU     6       ;STACK POINTER SAVE LOCATION
                        ;
                        ; REGISTER STORAGE DISPLACEMENTS FROM
                        ; NORMAL SYSTEM STACK LOCATION.
                        ;
0015 =                  ALOC:   EQU     15H
0013 =                  BLOC:   EQU     13H
0012 =                  CLOC:   EQU     12H
0011 =                  DLOC:   EQU     11H
0010 =                  ELOC:   EQU     10H
0014 =                  FLOC:   EQU     14H
0031 =                  HLOC:   EQU     31H
0030 =                  LLOC:   EQU     30H
0034 =                  PLOC:   EQU     34H
0017 =                  SLOC:   EQU     17H
0035 =                  TLOC:   EQU     35H
0025 =                  TLOCX:  EQU     25H
0020 =                  LLOCX:  EQU     20H
                        ;
0009 =                  APLOC:  EQU     9
000B =                  BPLOC:  EQU     11
000A =                  CPLOC:  EQU     10
000D =                  DPLOC:  EQU     13
000C =                  EPLOC:  EQU     12
0008 =                  FPLOC:  EQU     8
000F =                  HPLOC:  EQU     15
000E =                  LPLOC:  EQU     14
0007 =                  XLOC:   EQU     7
0005 =                  YLOC:   EQU     5
0002 =                  RLOC:   EQU     2
0003 =                  ILOC:   EQU     3
                        ;
                        ; DISK CONTROLLER UNIQUE EQUATES
                        ;
0030 =                  DSTAT   EQU     30H     ;DISK STATUS PORT
0030 =                  DCMMD   EQU     DSTAT   ;DISK COMMAND PORT
0031 =                  DTRCK   EQU     DSTAT+1 ;DISK TRACK PORT
0032 =                  DSCTR   EQU     DSTAT+2 ;DISK SECTOR PORT
```

```
CP/M MACRO ASSEM 2.0    #002    DISK MOSS 2.2 MONITOR

 0033 =          DDATA   EQU     DSTAT+3 ;DISK DATA PORT
 0034 =          DFLAG   EQU     DSTAT+4 ;DISK FLAG PORT
 0034 =          DCNTL   EQU     DSTAT+4 ;DISK CONTROL PORT
                 ;
                 ;
 0040 =          DISKNO: EQU     40H     ;ACTIVE DISK NUMBER
 0041 =          TRACK:  EQU     DISKNO+1
 0042 =          SECTOR: EQU     TRACK+1
 0043 =          SIDE:   EQU     SECTOR+1  ;SIDE SELECT HOLD AREA
 0044 =          SPT:    EQU     SIDE+1  ;SECTORS PER TRACK HOLD
 0045 =          TWOSID: EQU     SPT+1   ;SINGLE/DOUBLE SIDED SWITCH HOLD
 0046 =          STPRAT: EQU     46H     ;STEP RATE SAVE AREA
 0047 =          STATUS: EQU     47H
 0048 =          CMND:   EQU     STATUS+1
 0049 =          LUNIT:  EQU     49H     ;LAST USED DRIVE
 004A =          CUNIT:  EQU     LUNIT+1 ;CURRENT DRIVE
 004B =          RWFLG:  EQU     4BH
 004C =          HSTBUF: EQU     4CH     ;HOST BUFFER ADDRESS
 004E =          IDSV:   EQU     4EH     ;DISK ID SAVE AREA
 0080 =          TBUF:   EQU     80H
                 ;
                 ;
                 ; JUMP TARGETS FOR BASIC INPUT/OUTPUT
                 ;
 F000 C35BF0     CBOOT:  JMP     INIT    ;COLD START
 F003 C346F6     CONIN:  JMP     CI      ;CONSOLE INPUT
 F006 C356F6     READER: JMP     RI      ;READER INPUT
 F009 C300F6     CONOUT: JMP     CO      ;CONSOLE OUTPUT
 F00C C37CF6     PUNCH:  JMP     PO      ;PUNCH OUTPUT
 F00F C310F6     LIST:   JMP     LO      ;LIST OUTPUT
 F012 C323F6     CONST:  JMP     CSTS    ;CONSOLE STATUS
 F015 C36AF1             JMP     IOCHK   ;PUT IOBYTE INTO (A)
 F018 C365F1             JMP     IOSET   ;(C) HAS A NEW IOBYTE
 F01B C38AF0             JMP     MEMCK   ;MEMORY LIMIT CHECK
 F01E C394F6             JMP     RTS     ;IODEF- DEFINE USER I/O ENTRY POINTS
 F021 C394F6             JMP     RTS     ;SPCL- I/O CONTROL
 F024 C3CFF3             JMP     REST    ;BREAKPOINT ENTRY POINT
                 ;
                 ; TBL CONTAINS THE ADDRESSES OF THE ACTION ROUTINES
                 ;   THE EXECUTIVE USES IT TO LOOK UP THE DESIRED ADDRESS.
                 ;
 F027 F8F0       TBL:    DW      ASGN
 F029 5EF5               DW      BOOT
 F02B 09F1               DW      QPRT
 F02D ACF1               DW      DISP
 F02F 09F1               DW      QPRT
 F031 3CF1               DW      FILL
 F033 FDF1               DW      GOTO
 F035 D0F5               DW      HEXN
 F037 4DF2               DW      INPT
 F039 09F1               DW      QPRT
 F03B 09F1               DW      QPRT
 F03D 09F1               DW      QPRT
 F03F 5DF2               DW      MOVE
 F041 09F1               DW      QPRT
 F043 55F2               DW      OUPT
 F045 A7F5               DW      PARM
 F047 BDF5               DW      QPARM
 F049 F6F4               DW      READ
 F04B 67F2               DW      SUBS
 F04D 8FF2               DW      MTEST
 F04F 09F1               DW      QPRT
 F051 91F1               DW      COMP
 F053 F7F4               DW      WRITE
 F055 ECF2               DW      XMNE
 F057 9FF4               DW      I8250
 F059 82F1               DW      BYE
                 ;
```

```
CP/M MACRO ASSEM 2.0    #003    DISK MOSS 2.2 MONITOR

                ; THE COLD INITIALIZATION CODE
                ;
F05B F3         INIT:   DI              ;DISABLE INTERRUPTS
F05C 313F00             LXI     SP,3FH  ;USE STACK TO INITIALIZE RESTARTS
F05F 2100C3             LXI     H,JMP*256       ;  WITH RESTART ERROR VECTORS
F062 11B2F6             LXI     D,RSTER
F065 0610               MVI     B,16    ;16 TIMES (64 BYTES)
F067 D5         INIT1:  PUSH    D
F068 E5                 PUSH    H
                        DJNZ    INIT1
F069+10FC
F06B 3195F0             LXI     SP,FAKE-2       ;SET UP TEMPORARY STACK
F06E 3E00               MVI     A,0     ; SKIP THE NEXT INST
F06F                    ORG     $-1     ;SAVE A BYTE HERE
                ;
                ; MEMSIZ CALCULATES THE TOP OF CONTIGUOUS RAM.  IT SEARCHES
                ;       FROM THE BOTTOM UP UNTIL A NON-RAM LOCATION IS
                ;       FOUND.  IT THEN TAKES OFF FOR MONITOR WORK SPACE
                ;       NEEDS AND RETURNS THE VALUE IN (H,L).
                ;
F06F C5         MEMSIZ: PUSH    B       ;MONITOR START LOCATION
F070 0100F0             LXI     B,ROM
F073 21FFFF             LXI     H,-1    ;START OF MEMORY ADDRESS SPACE
F076 24         MEMSZ1: INR     H
F077 7E                 MOV     A,M
F078 2F                 CMA
F079 77                 MOV     M,A
F07A BE                 CMP     M
F07B 2F                 CMA
F07C 77                 MOV     M,A
                        JRNZ    MEMSZ2
F07D+2004
F07F 7C                 MOV     A,H     ;SEE IF ON MONITOR BORDER
F080 B8                 CMP     B
                        JRNZ    MEMSZ1
F081+20F3
F083 25         MEMSZ2: DCR     H       ;TAKE OFF WORKSPACE
F084 01DEFF             LXI     B,EXIT-ENDX-3*NBKPTS+1
F087 09                 DAD     B
F088 C1                 POP     B       ;(B,C) IS UNPREDICTABLE DURING INIT
F089 C9                 RET
                ;
                ; ROUTINE MEMCHK FINDS THE CURRENT TOP OF CONTIGUOUS MEMORY
                ;       (LESS THE MONITOR WORKSPACE) AND RETURNS THE VALUE.
                ;
F08A E5         MEMCK:  PUSH    H       ;SAVE (H,L)
F08B CD6FF0             CALL    MEMSIZ  ;GET THE RAM SIZE
F08E 7D                 MOV     A,L
F08F D63C               SUI     60      ;TAKE OFF WORK SPACE
                        JRNC    MEMCK0
F091+3001
F093 25                 DCR     H
F094 44         MEMCK0: MOV     B,H
F095 E1                 POP     H
F096 C9                 RET
                ;
F097 99F0       FAKE:   DW      FAKE+2
F099 F9                 SPHL
F09A 1145F4             LXI     D,EXIT
F09D EB                 XCHG
F09E 011D00             LXI     B,ENDX-EXIT
                        LDIR
F0A1+EDB0
F0A3 010600             LXI     B,3*NBKPTS
F0A6 D5                 PUSH    D
F0A7 E1                 POP     H
F0A8 2B                 DCX     H
                        LDIR
```

```
CP/M MACRO ASSEM 2.0    #004    DISK MOSS 2.2 MONITOR

 F0A9+EDB0
 F0AB 21E8FF            LXI     H,-24
 F0AE 39                DAD     SP
 F0AF E5                PUSH    H
 F0B0 23                INX     H       ;ADJUST USER STACK LOCATION
 F0B1 23                INX     H
 F0B2 220600            SHLD    SPSV    ;SAVE THE STACK INITIAL VALUE
 F0B5 160A              MVI     D,10    ;INITIALIZE REGISTER STORAGE AREA
 F0B7 C5        INIT2:  PUSH    B
 F0B8 15                DCR     D       ;LOOP CONTROL
                        JRNZ    INIT2
 F0B9+20FC
                ; INSERT I/O INIT CODE HERE
 F0BB CD59F5            CALL    DINIT   ;SEE IF AUTO BOOT WANTED
 F0BE CD9FF4            CALL    I8250   ;INITIALIZE THE 8250
 F0C1 CD94F6            CALL    RTS
 F0C4 2190F4            LXI     H,LOGMSG ;LOG ONTO THE SYSTEM
 F0C7 CD95F6            CALL    PRTWD
                        JMPR    WINIT   ;GO TO MONITOR EXECUTIVE
 F0CA+1843
                ;
                ;  ROUTINE EXF READS ONE PARAMETER.  IT EXPECTS THE FIRST
                ;       CHARACTER OF THE PARAMETER TO BE IN THE A REGISTER
                ;       ON ENTRY.
                ;
 F0CC 0601      EXF:    MVI     B,1     ;SET UP FOR ONE PARAMETER
 F0CE 210000            LXI     H,0
                        JMPR    EX1     ;FIRST CHARACTER IN A ALREADY
 F0D1+180C
                ;
                ;  ROUTINE EXPR READS PARAMETERS FROM THE CONSOLE
                ;       AND DEVELOPS A 16 BIT HEXADECIMAL FOR EACH ONE.
                ;       THE NUMBER OF PARAMETERS WANTED IS IN THE B REG
                ;       ON ENTRY.  A CARRIAGE RETURN WILL TERMINATE THE
                ;       ENTRY SEQUENCE; A BLANK OR A COMMA WILL END THE
                ;       CURRENT PARAMETER ENTRY.  EACH PARAMETER ONLY
                ;       TAKES THE LAST 4 DIGITS TYPED IN; ANY EXCESS IS
                ;       DISCARDED.  A NON-HEX DIGIT WILL TERMINATE THE
                ;       ENTRY SEQUENCE AND CAUSE A WARM BOOT OF THE MON.
                ;
                AS3:    DJNZ    AS2     ;PART OF THE ASSIGN CODE
 F0D3+1079
                EX3:    JRNZ    QPRT    ;NON-ZERO IS ERROR
 F0D5+2032
 F0D7 05        EXPR1:  DCR     B       ;MORE PARAMETERS?
 F0D8 C8                RZ              ;NO, RETURN
 F0D9 210000    EXPR:   LXI     H,0     ;INITIALIZE PARAMETER
 F0DC CD7BF3    EX0:    CALL    ECHO    ;GET NEXT NUMBER
 F0DF 4F        EX1:    MOV     C,A     ;SAVE CHAR FOR LATER USE
 F0E0 CDB0F3            CALL    NIBBLE
                        JRC     EX2     ;NOT A NUMBER, JUMP
 F0E3+3808
 F0E5 29                DAD     H       ;MULTIPLY BY 16
 F0E6 29                DAD     H
 F0E7 29                DAD     H
 F0E8 29                DAD     H
 F0E9 B5                ORA     L       ;ADD ON NEW DIGIT
 F0EA 6F                MOV     L,A
                        JMPR    EX0     ;GO GET NEXT DIGIT
 F0EB+18EF
 F0ED E3        EX2:    XTHL            ;PUT UNDER RETURN ADDRESS ON STACK
 F0EE E5                PUSH    H       ;RESTORE RETURN ADDRESS
 F0EF 79                MOV     A,C     ;REGET THE LAST CHARACTER
 F0F0 CDC3F3            CALL    P2C     ;TEST FOR DELIMITER
                        JRNC    EX3     ;JUMP IF NOT CARRIAGE RETURN
 F0F3+30E0
                        DJNZ    QPRT    ;CARRET WITH MORE PARAM MEANS ERROR
 F0F5+1012
```

```
CP/M MACRO ASSEM 2.0    #005    DISK MOSS 2.2 MONITOR

 F0F7 C9                RET
                ;
                ; MAIN ACTION ROUTINES
                ;
                ; LOGICAL ASSIGNMENT OF PERIPHERALS
                ;
                ;THIS ROUTINE CONTROLS THE ASSIGNMENT OF PHYSICAL
                ;       PERIPHERALS TO THE FOUR LOGICAL DEVICE TYPES. IT
                ;       ALTERS IOBYTE (MEMORY LOCATION 0003) TO MATCH THE
                ;       CURRENT ASSIGNMENT.  THE FOUR LOGICAL DEVICES ARE
                ;       CONSOLE, READER, LIST, AND PUNCH.  IN ALL CASES,
                ;       THE TTY DEVICE IS SET UP AS THE DEFAULT DEVICE.
                ;
 F0F8 CD7BF3    ASGN:   CALL    ECHO    ;GET THE LOGICAL DEVICE DESIRED
 F0FB 216EF1            LXI     H,ALT   ;START OF CONVERSION TABLE
 F0FE 110500            LXI     D,APT-ALT       ;DISTANCE BETWEEN LOGICAL CHOICES
 F101 0604              MVI     B,4     ;NUMBER OF LOGICAL CHOICES
 F103 BE        ASO:    CMP     M       ;IS THIS ONE IT?
                        JRZ     AS1     ;YES, JUMP

 F104+2842
 F106 19                DAD     D       ;NO, GO TO NEXT LOGICAL ENTRY
                        DJNZ    ASO

 F107+10FA
 F109 218CF4    QPRT:   LXI     H,QMSG  ;GET ADDRESS OF QUESTION MARK MSG
 F10C CD98F6            CALL    PRTWA   ;PRINT IT
                ;
                ; THE WARM START CODE
                ;
 F10F 2A0600    WINIT:  LHLD    SPSV    ;RESET THE STACK
 F112 F9                SPHL
 F113 210FF1    WINITA: LXI     H,WINIT ;RESET RETURN AND WARM START VECTOR
 F116 E5                PUSH    H
 F117 220100            SHLD    WSVEC+1
 F11A 3EC3              MVI     A,0C3H
 F11C 320000            STA     WSVEC
 F11F CDA9F6            CALL    CRLF    ;START A NEW LINE
 F122 CD78F3            CALL    DECHO   ;GET THE COMMAND
 F125 D641              SUI     'A'     ;GET RID OF ASCII ZONE
                        JRC     QPRT    ;BAD COMMAND

 F127+38E0
 F129 FE1A              CPI     'Z'-'A'+1       ;CHECK UPPER LIMIT
                        JRNC    QPRT    ;BAD COMMAND

 F12B+30DC
 F12D 87                ADD     A       ;DOUBLE IT FOR TABLE OFFSET
 F12E 5F                MOV     E,A     ;SET UP FOR DOUBLE ADD
 F12F 1600              MVI     D,0
 F131 0602              MVI     B,2     ;SET UP FOR TWO PARAMETERS
 F133 2127F0            LXI     H,TBL   ;GET ACTION ROUTINE ADDRESS
 F136 19                DAD     D
 F137 7E                MOV     A,M     ;LOAD H,L INDIRECT
 F138 23                INX     H
 F139 66                MOV     H,M
 F13A 6F                MOV     L,A
 F13B E9                PCHL            ;GO TO ACTION ROUTINE
                ;
                ; FILL ACTION ROUTINE
                ;
                ;       THIS ROUTINE FILLS A BLOCK OF MEMORY WITH A USER-
                ;       DETERMINED CONSTANT.  IT EXPECTS THREE PARAMETERS
                ;       TO BE ENTERED IN THE FOLLOWING ORDER:
                ;
                ;       START ADDRESS
                ;       FINISH ADDRESS
                ;       FILL VALUE
                ;
 F13C CD86F3    FILL:   CALL    EXPR3   ;GET THREE PARAMETERS
 F13F 71        FIO:    MOV     M,C     ;PUT DOWN THE FILL VALUE
```

```
CP/M MACRO ASSEM 2.0    #006    DISK MOSS 2.2 MONITOR

F140 CD8FF3             CALL    HILO    ;INCREMENT AND CHECK THE POINTER
                        JRNC    FIO     ;NOT DONE YET, JUMP
F143+30FA
F145 D1                 POP     D       ;RESTORE STACK POINTER IN CASE
                        JMPR    WINIT   ; STACK WAS OVERWRITTEN
F146+18C7
                ;
F148 50         AS1:    MOV     D,B     ;SAVE THE COUNTER RESIDUE
F149 0604               MVI     B,4     ;LOOP CONTROL
F14B CD78F3             CALL    DECHO   ;GET THE NEW ASSIGNMENT
F14E 23         AS2:    INX     H       ;INCREMENT POINTER
F14F BE                 CMP     M       ;SEE IF THIS IS IT
                        JRNZ    AS3
F150+2081
F152 68                 MOV     L,B     ;SAVE THE RESIDUE TO FORM ASGT
F153 2D                 DCR     L       ;ADJUST VALUE
F154 42                 MOV     B,D     ;REGET THE LOGICAL RESIDUE
F155 2603               MVI     H,3     ;SET UP THE IOBYTE MASK
F157 05                 DCR     B       ;ADJUST THIS ONE ALSO
                        JRZ     AS5     ;NO SHIFT NEEDED
F158+2804
F15A 29         AS4:    DAD     H       ;SHIFT THE MASKS INTO POSITION
F15B 29                 DAD     H
                        DJNZ    AS4     ;NOT DONE YET, JUMP
F15C+10FC
F15E 3A0300     AS5:    LDA     IOBYTE
F161 B4                 ORA     H       ;MASK THE DESIRED ASSIGNMENT IN
F162 AC                 XRA     H       ;LOGICAL ASGT BITS NOW OFF
F163 B5                 ORA     L       ;PUT IN NEW VALUE
F164 4F                 MOV     C,A
F165 79         IOSET:  MOV     A,C
F166 320300             STA     IOBYTE  ;SAVE NEW ASSIGNMENTS
F169 C9                 RET
F16A 3A0300     IOCHK:  LDA     IOBYTE
F16D C9                 RET
                ;
F16E 4C         ALT:    DB      'L'     ;LOGICAL LIST DEVICE TABLE
F16F 32                 DB      '2'     ;USER DEVICE #2
F170 31                 DB      '1'     ;USER DEVICE #1
F171 4C                 DB      'L'     ;LIST TO HIGH SPEED PRINTER
F172 54                 DB      'T'     ;LIST TO TTY
F173 50         APT:    DB      'P'     ;LOGIPAL PUNCH DEVICE TABLE
F174 32                 DB      '2'     ;USER DEVICE #2
F175 31                 DB      '1'     ;USER DEVICE #1
F176 50                 DB      'P'     ;PUNCH TO HIGH SPEED PUNCH
F177 54                 DB      'T'     ;PUNCH TO TTY
F178 52         ART:    DB      'R'     ;LOGIPAL READER DEVICE TABLE
F179 32                 DB      '2'     ;USER DEVICE #2
F17A 31                 DB      '1'     ;USER DEVICE #1
F17B 50                 DB      'P'     ;READER TO HIGH SPEED READER
F17C 54                 DB      'T'     ;READER TO TTY
F17D 43         ACT:    DB      'C'     ;LOGIPAL CONSOLE DEVICE TABLE
F17E 31                 DB      '1'     ;USER DEVICE #1
F17F 42                 DB      'B'     ;CONSOLE TO BATCH (PRINTER OR PTR)
F180 43                 DB      'C'     ;CONSOLE TO CRT
F181 54                 DB      'T'     ;CONSOLE TO TTY
                ;
                ; THE BYE ROUTINE IS USED TO PREVENT UNAUTHORIZED USAGE
                ;       OF THE SYSTEM.  THE SYSTEM LOCKS UP AND WILL NOT
                ;       RESPOND TO ANYTHING OTHER THAN TWO ASCII BELL
                ;       CHARACTERS.  WHEN IT SEES THEM CONSECUTIVELY,
                ;       CONTROL IS RETURNED TO THE MONITOR WITHOUT ALTERING
                ;       ANYTHING.
                ;
F182 0602       BYE:    MVI     B,2     ;SET UP FOR TWO CHARACTERS
F184 CD8FF6     BYE1:   CALL    CONI    ;GO READ THE CONSOLE
F187 FE07               CPI     BELL    ;SEE IF AN ASCII BELL
                        JRNZ    BYE     ;NO, START OVER AGAIN
```

```
CP/M MACRO ASSEM 2.0    #007    DISK MOSS 2.2 MONITOR

 F189+20F7
 F18B CD7EF3            CALL    ECH1    ;ECHO THE BELL
                        DJNZ    BYE1    ;NOT YET, GET NEXT ONE
 F18E+10F4
 F190 C9                RET             ;RETURN TO MONITOR
                ;
                ;  COMPARE ROUTINE
                ;
                ;THIS ROUTINE COMPARES TWO BLOCKS OF MEMORY AGAINST EACH
                ;       OTHER.  IF ADIFFERENCE IN THE RELATIVE ADDRESSES
                ;       IS DETECTED, THE ADDRESS OF THE FIRST BLOCK IS
                ;       DISPLAYED, ALONG WITH ITS CONTENTS AND THE CONTENTS
                ;       OF THE OTHER BLOCK'S SAME RELATIVE ADDRESS.
                ;
 F191 CD86F3    COMP:   CALL    EXPR3   ;GO GET THREE PARAMETERS
 F194 0A        CMPA:   LDAX    B       ;GET SOURCE 2 DATA
 F195 C5                PUSH    B       ;SAVE SOURCE 2 POINTER
 F196 46                MOV     B,M     ;READ SOURCE 1 DATA
 F197 B8                CMP     B       ;COMPARE DATA
                        JRZ     CMPB    ;JUMP IF OK
 F198+280C
 F19A F5                PUSH    PSW     ;SAVE SOURCE 2 DATA
 F19B CDFBF5            CALL    LADRB   ;WRITE THE ADDRESS
 F19E 78                MOV     A,B     ;GET SOURCE 1 DATA
 F19F CDF4F5            CALL    DASH1   ;FORMAT
 F1A2 F1                POP     PSW     ;REGET SOURCE 2 DATA
 F1A3 CDE6F5            CALL    HEX1    ;OUTPUT IT
 F1A6 C1        CMPB:   POP     B
 F1A7 CD9BF3            CALL    HILOXB  ;INCREMENT SOURCE 1 POINTER AND SEE IF DONE
                        JMPR    CMPA    ;JUMP IF NOT DONE YET
 F1AA+18E8
                ;
                ; DISPLAY ACTION ROUTINE
                ;
                ;       THIS ROUTINE DISPLAYS A BLOCK OF MEMORY ON THE
                ;       CURRENT CONSOLE DEVICE (CONSOLE DUMP).  THE USER
                ;       MUST SPECIFY THE START AND FINISH ADDRESSES.
                ;       THE DISPLAY IS ORGANIZED TO DISPLAY UP TO 16 BYTES
                ;       PER DISPLAY LINE, WITH ALL COLUMNS ALIGNED SO
                ;       EACH COLUMN HAS THE SAME LAST HEX DIGIT IN ITS ADDRESS.
                ;
 F1AC CDA4F6    DISP:   CALL    EXLF    ;GO GET BLOCK LIMITS
 F1AF CDFBF5    DIS1:   CALL    LADRB   ;DISPLAY THE START ADDRESS
 F1B2 7D                MOV     A,L     ;SEE IF ON 16 BYTE BOUNDARY
 F1B3 CDF0F1            CALL    TRPLSP  ;SKIP OVER TO RIGHT COLUMN
 F1B6 E5                PUSH    H       ;SAVE (H,L)
 F1B7 7E        DIS2:   MOV     A,M     ;GET THE CONTENTS
 F1B8 CDE6F5            CALL    HEX1    ;OUTPUT IT
 F1BB CD8FF3            CALL    HILO    ;INCREMENT, CHECK POINTER
                        JRC     DIS7    ;DONE IF CARRY SET
 F1BE+382A
 F1C0 CDFEF5            CALL    BLK     ;MAKE COLUMNS
 F1C3 7D                MOV     A,L     ;READY FOR NEW LINE?
 F1C4 E60F              ANI     0FH
                        JRNZ    DIS2
 F1C6+20EF
 F1C8 E1        DIS3:   POP     H       ;REGET LINE START ADDRESS
 F1C9 7D                MOV     A,L     ;SKIP OVER TO RIGHT SPACE
 F1CA E60F              ANI     0FH
 F1CC CDF5F1            CALL    TRPL2
 F1CF 7E        DIS4:   MOV     A,M     ;GET MEMORY VALUE
 F1D0 E67F              ANI     7FH     ;STRIP OFF PARITY BIT
 F1D2 4F                MOV     C,A     ;SET UP FOR OUTPUT
 F1D3 FE20              CPI     ' '     ;SEE IF PRINTABLE IN ASCII
                        JRC     DIS5    ;JUMP IF SO
 F1D5+3804
 F1D7 FE7E              CPI     7EH
                        JRC     DIS6
```

```
CP/M MACRO ASSEM 2.0     #008     DISK MOSS 2.2 MONITOR

 F1D9+3802
 F1DB 0E2E       DIS5:   MVI     C,'.'   ;ELSE, PRINT A DOT
 F1DD CD09F0     DIS6:   CALL    CONOUT
 F1E0 CD9CF3             CALL    HILOX   ;INCREMENT (H,L) AND SEE IF DONE
 F1E3 7D                 MOV     A,L     ;NOT DONE, READY FOR NEW LINE?
 F1E4 E60F               ANI     0FH
                         JRNZ    DIS4    ;JUMP IF NOT
 F1E6+20E7
                         JMPR    DIS1    ;DO THE NEXT LINE
 F1E8+18C5
 F1EA 93         DIS7:   SUB     E       ;SKIP OVER TO START ASCII PRINTOUT
 F1EB CDF0F1             CALL    TRPLSP
                         JMPR    DIS3    ;GO PRINT THE ASCII
 F1EE+18D8
                 ;
 F1F0 E60F       TRPLSP: ANI     0FH     ;ISOLATE THE LOW FOUR BITS
 F1F2 47                 MOV     B,A     ;PREPARE TO SPACE OVER TO RIGHT COLUMN
 F1F3 87                 ADD     A       ;TRIPLE THE COUNT
 F1F4 80                 ADD     B
 F1F5 47         TRPL2:  MOV     B,A     ;PUT BACK INTO B
 F1F6 04                 INR     B       ;ADJUST COUNTER
 F1F7 CDFEF5     TRPL1:  CALL    BLK     ;DO THE SPACING
                         DJNZ    TRPL1   ;NO, DO ANOTHER COLUMN
 F1FA+10FB
 F1FC C9                 RET
                 ;
                 ; GO TO ACTION ROUTINE
                 ;
                 ;
                 ; GOTO COMMAND TRANSFERS CONTROL TO A SPECIFIED ADDRESS.
                 ;  IT ALLOWS THE SELECTIVE SETTING OF UP TO TWO BREAKPOINTS
                 ;    AS WELL AS ALLOWING ANY CONSOLE INPUT TO BREAKPOINT
                 ;    THE RUN, AS LONG AS INTERRUPT 1 IS ACTIVE.
                 ;
 F1FD CDC0F3     GOTO:   CALL    PCHK    ;SEE IF OLD ADDRESS WANTED
                         JRC     GO3     ;  YES, JUMP
 F200+3837
                         JRZ     GO0     ;  YES, BUT SET SOME BREAKPOINTS
 F202+2810
 F204 CDCCF0             CALL    EXF     ;GET NEW GOTO ADDRESS
 F207 D1                 POP     D
 F208 213400             LXI     H,PLOC  ;PUT ADDRESS IN PC LOCATION
 F20B 39                 DAD     SP
 F20C 72                 MOV     M,D     ;LOW BYTE
 F20D 2B                 DCX     H
 F20E 73                 MOV     M,E     ;HIGH BYTE
 F20F 79                 MOV     A,C
 F210 FE0D               CPI     CR      ;SEE IF A CR WAS LAST ENTERED
                         JRZ     GO3
 F212+2825
 F214 0602       GO0:    MVI     B,NBKPTS
 F216 213500             LXI     H,TLOC  ;POINT TO TRAP STORAGE
 F219 39                 DAD     SP
 F21A C5         GO1:    PUSH    B       ;SAVE NUMBER OF BREAKPOINTS
 F21B E5                 PUSH    H       ;SAVE STORAGE POINTER
 F21C 0602               MVI     B,2     ;SET UP TO GET A TRAP ADDRESS
 F21E CDD7F0             CALL    EXPR1   ;GET A TRAP ADDRESS
 F221 D1                 POP     D       ;GET THE TRAP ADDRESS INTO (D,E)
 F222 E1                 POP     H       ;REGET THE STORAGE ADDRESS
 F223 7A                 MOV     A,D     ;INSURE THE TRAP ADDRESS ISN'T ZERO
 F224 B3                 ORA     E
                         JRZ     GO2     ;JUMP IF SO
 F225+280A
 F227 73                 MOV     M,E     ;SAVE THE BREAKPOINT ADDRESS
 F228 23                 INX     H
 F229 72                 MOV     M,D
 F22A 23                 INX     H
 F22B 1A                 LDAX    D       ;SAVE THE INSTRUCTION FROM THE BP ADDRESS
```

```
CP/M MACRO ASSEM 2.0    #009    DISK MOSS 2.2 MONITOR

 F22C 77                MOV     M,A
 F22D 23                INX     H
 F22E 3ECF              MVI     A,RST OR 8      ;INSERT THE BREAKPOINT
 F230 12                STAX    D
 F231 79        GO2:    MOV     A,C     ;REGET THE DELIMITER TO SEE
 F232 FE0D              CPI     CR      ;  IF WE ARE DONE SETTING BREAKPOINTS
 F234 C1                POP     B       ;  UNLOAD THE STACK FIRST
                        JRZ     GO3     ;YES, JUMP

 F235+2802
                        DJNZ    GO1     ;JUMP IF NOT AT BP LIMIT

 F237+10E1
 F239 CDA9F6    GO3:    CALL    CRLF
 F23C E1                POP     H       ;GET RID OF STACK JUNK
 F23D 2143F4            LXI     H,RS9
 F240 E5                PUSH    H
 F241 21CFF3            LXI     H,REST
 F244 220900            SHLD    9       ;SET BREAKPOINT JUMP VECTOR ADDRESS
 F247 211800            LXI     H,24    ;FIND REGISTER SET ROUTINE ADDRESS
 F24A 39                DAD     SP
 F24B D1                POP     D       ;ADJUST THE STACK
 F24C E9                PCHL            ;GO TO THE DESIRED PLACE

                ;
                ; GENERAL PURPOSE INPUT/OUTPUT ROUTINES
                ;
                ;THESE ROUTINES ALLOW BYTE-BY-BYTE INPUT OR OUTPUT FROM
                ;       THE CURRENT CONSOLE DEVICE.  THEY ARE INVOKED BY
                ;       THE MONITOR "I" OR "O" COMMAND, THEN ANSWERING THE
                ;       QUESTIONS WHICH APPEAR ON THE CONSOLE.
                ;
 F24D CDD7F0    INPT:   CALL    EXPR1   ;GET INPUT PORT NUMBER
 F250 C1                POP     B       ;GET PORT # INTO C REGISTER
                        INP     E       ;READ VALUE INTO E REGISTER

 F251+ED58
                        JMPR    BITS2   ;GO DO A BINARY PRINT OF THE VALUE

 F253+1851
                ;
 F255 CDD9F0    OUPT:   CALL    EXPR    ;GET THE ADDRESS AND DATA FOR OUPTUT
 F258 D1                POP     D       ;DATA VALUE INTO E
 F259 C1                POP     B       ;PORT INTO C
                        OUTP    E       ;DO THE OUTPUT

 F25A+ED59
 F25C C9                RET

                ;
                ;  MOVE ROUTINE
                ;
                ;       THIS ROUTINE EXPECTS THREE PARAMETERS, ENTERED
                ;       SOURCE FIRST BYTE ADDRESS
                ;       SOURCE LAST BYTE ADDRESS
                ;       DESTINATION FIRST BYTE ADDRESS
                ;
 F25D CD86F3    MOVE:   CALL    EXPR3   ;GET THREE PARAMETERS
 F260 7E        MOV1:   MOV     A,M     ;GET NEXT BYTE
 F261 02                STAX    B       ;MOVE IT
 F262 CD9BF3            CALL    HILOXB  ;GO INCREMENT, CHECK SOURCE POINTER
                        JMPR    MOV1    ;NOT THERE YET, GO DO IT AGAIN

 F265+18F9
                ;
                ; SUBSTITUTE ACTION ROUTINE
                ;
                ;THIS ROUTINE ALLOWS THE USER TO INSPECT ANY MEMORY LOCATION
                ;       AND ALTER THE CONTENTS, IF DESIRED AND IF THE ADDRESS
                ;       IS IN RAM.  THE CONTENTS MAY BE LEFT UNALTERED
                ;       BY ENTERING A SPACE, COMMA, OR A CARRIAGE RETURN.  IF
                ;       A CARRIAGE RETURN IS ENTERED, THE ROUTINE IS TERMINATED.
                ;       IF A SPACE OR COMMA IS ENTERED, THE ROUTINE
                ;       PROCEEDS TO THE NEXT LOCATION AND PRESENTS THE USER
                ;       WITH AN OPPORTUNITY TO ALTER IT.
```

```
CP/M MACRO ASSEM 2.0    #010    DISK MOSS 2.2 MONITOR

                ;
F267 CDD7F0     SUBS:   CALL    EXPR1   ;GO GET ONE PARAMETER
F26A E1                 POP     H       ;GET THE START ADDRESS
F26B 7E         SUB1:   MOV     A,M     ;GET THE CONTENTS OF THE ADDRESS
F26C CDF4F5             CALL    DASH1   ;DISPLAY IT ON CONSOLE AND A DASH
F26F CDC0F3             CALL    PCHK    ;GET, CHECK CHARACTER
F272 D8                 RC              ;DONE IF CARRIAGE RETURN
                        JRZ     SUB2    ;NO CHANGE IF BLANK OR ,
F273+280F
F275 FE0A               CPI     LF      ;SEE IF PREVIOUS BYTE WANTED
                        JRZ     SUB3    ;YES, DO IT
F277+280D
F279 E5                 PUSH    H       ;SAVE MEMORY POINTER
F27A CDCCF0             CALL    EXF     ;GO GET REST OF NEW VALUE
F27D D1                 POP     D       ;NEW VALUE TO E REGISTER
F27E E1                 POP     H       ;RESTORE MEMORY POINTER
F27F 73                 MOV     M,E     ;PUT DOWN NEW VALUE
F280 79                 MOV     A,C     ;GET THE DELIMITER
F281 FE0D               CPI     CR      ;SEE IF DONE (CARRIAGE RETURN)
F283 C8                 RZ              ;YES, RETURN TO MONITOR
F284 23         SUB2:   INX     H       ;NO, INCREMENT MEMORY POINTER
F285 23                 INX     H       ;ALLOW A FALL-THROUGH ON THE NEXT INSTRUCTION
F286 2B         SUB3:   DCX     H       ;ADJUST (H,L) AS APPROPRIATE
F287 7D                 MOV     A,L     ;GET LO ADDRESS BYTE
F288 E607               ANI     7       ;SEE IF ON A BOUNDARY
F28A CCFBF5             CZ      LADRB   ;CALL IF ON THE BOUNDARY
                        JMPR    SUB1    ;GO DO THE NEXT LOCATION
F28D+18DC

                ;
                ; MTEST ROUTINE TESTS A SPECIFIED BLOCK OF MEMORY TO
                ;   SEE IF ANY HARD DATA BIT FAILURES EXIST.  IT IS
                ;   NOT AN EXHAUSTIVE TEST, BUT JUST A QUICK INDICATION
                ;   OF THE MEMORY'S OPERATIVENESS.
                ;
F28F CDA4F6     MTEST:  CALL    EXLF
F292 7E         MTEST1: MOV     A,M     ;READ A BYTE
F293 F5                 PUSH    PSW     ;SAVE IT
F294 2F                 CMA             ;COMPLEMENT IT
F295 77                 MOV     M,A     ;WRITE IT
F296 AE                 XRA     M       ;RESULT SHOULD BE ZERO
F297 C4A1F2             CNZ     BITS    ;LOG ERROR IF NOT
F29A F1         MTEST2: POP     PSW     ;RESTORE ORIGINAL BYTE
F29B 77                 MOV     M,A
F29C CD9CF3             CALL    HILOX   ;POINT TO NEXT AND SEE IF DONE
                        JMPR    MTEST1  ;NO, CONTINUE
F29F+18F1
                ;
F2A1 D5         BITS:   PUSH    D       ;SAVE (D,E)
F2A2 5F                 MOV     E,A     ;SAVE ERROR PATTERN IN E
F2A3 CDFBF5             CALL    LADRB   ;FIRST PRINT THE ADDRESS
F2A6 0608       BITS2:  MVI     B,8     ;LOOP CONTROL FOR 8 BITS
F2A8 7B         BITS1:  MOV     A,E     ;GET NEXT BIT
F2A9 07                 RLC             ;  INTO CARRY
F2AA 5F                 MOV     E,A     ;SAVE REST
F2AB 3E18               MVI     A,'0'/2 ;BUILD ASCII 1 OR 0
F2AD 17                 RAL             ;  CARRY DETERMINES WHICH
F2AE 4F                 MOV     C,A     ;NOW, OUPTUT IT
F2AF CD09F0             CALL    CONOUT
                        DJNZ    BITS1   ;DO IT AGAIN
F2B2+10F4
F2B4 D1                 POP     D
F2B5 C9                 RET
                ;
                ; EXAMINE REGISTERS COMMAND INSPECTS THE VALUES OF THE
                ;   THE REGISTERS STORED BY THE LAST ENCOUNTERED BREAKPOINT.
                ;   THE VALUES MAY BE MODIFIED IF DESIRED.
                ;
F2B6 23         XAA:    INX     H       ;SKIP OVER TO NEXT ENTRY
```

```
CP/M MACRO ASSEM 2.0    #011    DISK MOSS 2.2 MONITOR

F2B7 23                 INX     H
F2B8 34         XA:     INR     M       ;SEE IF AT END OF TABLE
F2B9 C8                 RZ              ;COULDN'T FIND MATCH, QUIT
F2BA F2C1F2             JP      XAB     ;SORT OUT BIT 7 OF TABLE
F2BD F680               ORI     80H     ;SET IT ON TEST VALUE
                        JMPR    XAC

F2BF+1802
F2C1 E67F       XAB:    ANI     7FH     ;RESET BIT 7
F2C3 35         XAC:    DCR     M       ;TO BE PULLED OUT IN ROM
F2C4 BE                 CMP     M       ;SEE IF THIS IS IT
                        JRNZ    XAA     ;NO, GO TRY AGAIN

F2C5+20EF
F2C7 CDFEF5             CALL    BLK     ;YES, PREPARE TO SHOW CURRENT VALUE
F2CA CD15F3             CALL    PRTVAL  ;GO PRINT THE VALUE
F2CD CDF7F5             CALL    DASH    ;PROMPT A NEW VALUE
F2D0 CDC0F3             CALL    PCHK    ;GET THE INPUT
F2D3 D8                 RC              ;DONE IF CARRIAGE RETURN
                        JRZ     XF      ;JUMP IF NO CHANGE DESIRED

F2D4+2812
F2D6 E5                 PUSH    H       ;TO BE CHANGED, SAVE POINTER
F2D7 CDCCF0             CALL    EXF     ;GET THE NEW VALUE
F2DA E1                 POP     H       ;   INTO (H,L)
F2DB 7D                 MOV     A,L     ;GET THE NEW LOW BYTE
F2DC 13                 INX     D       ;ADJUST POINTER
F2DD 12                 STAX    D       ;PUT IT DOWN
F2DE E3                 XTHL            ;RECOVER THE TABLE POINTER
F2DF 7E                 MOV     A,M     ;GET THE ATTRIBUTES
F2E0 E3                 XTHL            ;SET THE STACK STRAIGHT
F2E1 07                 RLC             ;SEE IF 8 BIT REGISTER
                        JRNC    XE      ;JUMP IF SO

F2E2+3003
F2E4 13                 INX     D       ;REGISTER PAIR, DO OTHER 8 BITS
F2E5 7C                 MOV     A,H
F2E6 12                 STAX    D
F2E7 E1         XE:     POP     H       ;RESTORE THE TABLE POINTER
F2E8 79         XF:     MOV     A,C     ;SEE IF IT WAS A CR
F2E9 FE0D               CPI     CR
F2EB C8                 RZ              ;DONE IF SO
F2EC 213DF3     XMNE:   LXI     H,ACTBL ;GET ADDRESS OF REGISTER LOOK-UP TABLE
F2EF CDC0F3     XMNE1:  CALL    PCHK    ;FIND OUT WHAT ACTION IS WANTED
                        JRC     XG      ;SHOW ALL IF CARRIAGE RETURN

F2F2+380B
                        JRZ     XMNE1   ;IGNORE BLANKS OR COMMAS

F2F4+28F9
F2F6 FE27               CPI     ''''    ;SEE IF PRIMES WANTED
                        JRNZ    XA      ;NO, MUST BE SINGLE REGISTER

F2F8+20BE
F2FA 2155F3             LXI     H,PRMTB ;YES, SET TABLE ADDRESS
                        JMPR    XMNE1   ;  AND FIND OUT WHICH ONE

F2FD+18F0

                ;
F2FF 7E         XG:     MOV     A,M
F300 4F                 MOV     C,A
F301 3C                 INR     A       ;SEE IF AT END OF TABLE
F302 C8                 RZ              ;DONE IF SO
F303 FCA9F6             CM      CRLF    ;START A NEW LINE IF BIT 7 IS SET
F306 CD09F0             CALL    CONOUT
F309 CDF7F5             CALL    DASH    ;PROMPT FOR A NEW VALUE
F30C CD15F3             CALL    PRTVAL  ;GO PRINT THE VALUE
F30F CDFEF5             CALL    BLK     ;FORMATTER
F312 23                 INX     H       ;POINT TO NEXT ENTRY
                        JMPR    XG      ;DO THE NEXT VALUE

F313+18EA

                ;
F315 23         PRTVAL: INX     H       ;POINT TO NEXT ENTRY
F316 7E                 MOV     A,M     ;GET OFFSET AND ATTRIBUTES BYTE
F317 E63F               ANI     3FH     ;ISOLATE THE OFFSET
F319 C602               ADI     2       ;ALLOW FOR RETURN ADDRESS
```

```
CP/M MACRO ASSEM 2.0    #012    DISK MOSS 2.2 MONITOR

 F31B EB                XCHG            ;SWAP POINTERS
 F31C 6F                MOV     L,A     ;BUILD THE ADDRESS OF THE REG CONTENTS
 F31D 2600              MVI     H,0
 F31F 39                DAD     SP
 F320 EB                XCHG            ;RE-SWAP THE POINTERS
 F321 7E                MOV     A,M     ;NOW FIND OUT ATTRIBUTES
 F322 0601              MVI     B,1     ;SET UP FOR SINGLE REG VALUE
 F324 07                RLC
                        JRNC    PV1     ;JUMP IF SINGLE REGISTER VALUE WANTED
 F325+300E
 F327 04                INR     B       ;SET UP FOR REGISTER PAIR
 F328 07                RLC
                        JRNC    PV1     ;JUMP IF REGISTER PAIR IS NEXT
 F329+300A
 F32B E5                PUSH    H       ;SPECIAL CASE FOR MEMORY REGISTER
 F32C 1A                LDAX    D       ;BUILD ADDRESS IN (H,L)
 F32D 67                MOV     H,A
 F32E 1B                DCX     D
 F32F 1A                LDAX    D
 F330 6F                MOV     L,A
 F331 7E                MOV     A,M     ;GET THE MEMORY VALUE
 F332 E1                POP     H       ;RESTORE (H,L)
                        DJNZ    PV2     ;ALWAYS JUMP
 F333+1001
 F335 1A        PV1:    LDAX    D       ;GET THE REGISTER CONTENTS
 F336 CDE6F5    PV2:    CALL    HEX1    ;OUTPUT THE VALUE
 F339 1B                DCX     D       ;ADJUST THE MEMORY POINTER
                        DJNZ    PV1
 F33A+10F9
 F33C C9                RET

                ;
 F33D C115      ACTBL:  DB      80H+'A',ALOC
 F33F 4213              DB      'B',BLOC
 F341 4312              DB      'C',CLOC
 F343 4411              DB      'D',DLOC
 F345 4510              DB      'E',ELOC
 F347 4614              DB      'F',FLOC
 F349 4831              DB      'H',HLOC
 F34B 4C30              DB      'L',LLOC
 F34D CDF1              DB      80H+'M',HLOC+0C0H
 F34F 50B4              DB      'P',PLOC+80H
 F351 5397              DB      'S',SLOC+80H
 F353 4903              DB      'I',ILOC
                ;
                ; REST OF Z-80 REGISTER OFFSETS
                ;
 F355 C109      PRMTB:  DB      80H+'A',APLOC
 F357 420B              DB      'B',BPLOC
 F359 430A              DB      'C',CPLOC
 F35B 440D              DB      'D',DPLOC
 F35D 450C              DB      'E',EPLOC
 F35F 4608              DB      'F',FPLOC
 F361 480F              DB      'H',HPLOC
 F363 4C0E              DB      'L',LPLOC
 F365 CDCF              DB      80H+'M',HPLOC+0C0H
 F367 5887              DB      'X',XLOC+80H
 F369 5985              DB      'Y',YLOC+80H
 F36B 5202              DB      'R',RLOC
 F36D FF                DB      0FFH
                ;
                ; GENERAL PURPOSE ROUTINES
                ;
                ;
                ; ROUTINE CONV CONVERTS THE LOW ORDER NIBBLE OF THE
                ;       ACCUMULATOR TO ITS ASCII EQUIVELANT.  IT
                ;       PUTS THE RESULT INTO C FOR LATER OUTPUT.
                ;
 F36E E60F      CONV:   ANI     0FH     ;STRIP OFF BITS 4-7
```

```
CP/M MACRO ASSEM 2.0    #013    DISK MOSS 2.2 MONITOR

F370 C690               ADI     90H     ;PUT ON THE ASCII ZONE
F372 27                 DAA
F373 CE40               ACI     40H
F375 27                 DAA
F376 4F                 MOV     C,A     ;PUT IN OUTPUT PASS REGISTER
F377 C9                 RET

                ;
                ; ROUTINE ECHO READS A BYTE FROM A HALF-DUPLEX CONSOLE
                ;       DEVICE, THEN ECHOES THE CHARACTER BACK TO THE
                ;       CONSOLE.
                ;
F378 CDF7F5     DECHO:  CALL    DASH    ;PRINT A DASH
F37B CD8FF6     ECHO:   CALL    CONI    ;CONSOLE READ, WRITE ROUTINE
F37E C5         ECH1:   PUSH    B       ;  SAVE (B,C)
F37F 4F                 MOV     C,A     ;  PASS CHARACTER IN C REGISTER
F380 CD09F0             CALL    CONOUT  ;  OUTPUT IT
F383 79                 MOV     A,C     ;  PUT CHARACTER BACK INTO A
F384 C1                 POP     B       ;  RESTORE (B,C)
F385 C9                 RET

                ;
                ; ROUTINE EXPR3 GETS THREE PARAMETERS, DOES A CR, LF AND
                ;     THEN LOADS (B,C), (D,E), AND (H,L) WITH THE PARAMETERS.
                ;
F386 04         EXPR3:  INR     B       ;2 IS ALREADY IN THE B REGISTER
F387 CDD9F0             CALL    EXPR    ;GET THE PARAMETERS
F38A C1                 POP     B       ;PUT PARAMETERS INTO REGISTERS
F38B D1                 POP     D
F38C C3AAF6             JMP     CRLFA   ;GO DO THE CARRIAGE RETURN SEQUENCE

                ;
                ; ROUTINE HILO INCREMENTS (H,L).  IT THEN CHECKS FOR (AND
                ;       DISALLOWS) A WRAP-AROUND SITUATION.  IF IT OCCURS,
                ;       THE CARRY BIT WILL BE SET ON RETURN.  IF NO WRAP-
                ;       AROUND OCCURRED, (H,L) IS COMPARED TO (D,E) AND
                ;       THE FLAG BITS SET ACCORDINGLY.
                ;
F38F 23         HILO:   INX     H       ;INCREMENT (H,L)
F390 7C                 MOV     A,H     ;TEST IF ZERO
F391 B5                 ORA     L       ;  IN (H,L)
F392 37                 STC             ;SET CARRY FOR (H,L)=0
F393 C8                 RZ              ;RETURN IF (H,L) = 0
F394 7B                 MOV     A,E     ;COMPARE (H,L) TO (D,E)
F395 95                 SUB     L
F396 7A                 MOV     A,D
F397 9C                 SBB     H
F398 C9                 RET             ;RETURN WITH FLAGS SET

                ;
                ; ROUTINE HILOX INCREMENTS (H,L), COMPARES IT TO (D,E) AND
                ;   IF EQUAL, RETURNS CONTROL TO THE MONITOR EXECUTIVE.
                ;   OTHERWISE, CONTROL RETURNS TO THE CALLING ROUTINE.
                ;
F399 D1         HILOD:  POP     D       ;GET RID OF RETURN ADDRESS
F39A C9                 RET             ;RETURN TO MONITOR
F39B 03         HILOXB: INX     B       ;INCREMENT (B,C)
F39C CD8FF3     HILOX:  CALL    HILO    ;INC AND CHECK (H,L)
                        JRC     HILOD   ;DONE IF CARRY SET

F39F+38F8
F3A1 CD12F0             CALL    CONST   ;SEE IF CONSOLE BREAK PENDING
F3A4 B7                 ORA     A
F3A5 C8                 RZ              ;NONE, RETURN TO CONTINUE
F3A6 CD8FF6             CALL    CONI    ;SEE IF WAIT OR BREAK
F3A9 FE13               CPI     CTRLS
                        JRNZ    HILOD   ;JUMP IF BREAK

F3AB+20EC
F3AD C38FF6             JMP     CONI    ;WAIT FOR ANY INPUT

                ;
                ; ROUTINE NIBBLE CONVERTS THE ASCII CHARACTERS 0-9 AND
                ;       A-F TO THEIR EQUIVELANT HEXADECIMAL VALUE.  IF
                ;       THE CHARACTER IS NOT IN RANGE, THE CARRY BIT IS SET TO
```

```
CP/M MACRO ASSEM 2.0     #014     DISK MOSS 2.2 MONITOR

                ;       FLAG THE ERROR.
                ;
F3B0 D630       NIBBLE: SUI     '0'     ;ASCII TO HEX CONVERSION
F3B2 D8                 RC              ;  DONE IF OUT OF RANGE
F3B3 FE17               CPI     'G'-'0' ;CHECK UPPER END
F3B5 3F                 CMC             ;  TOGGLE THE CARRY BIT
F3B6 D8                 RC              ;  DONE IF OUT OF RANGE
F3B7 FE0A               CPI     '9'-'0'+1  ;SEE IF NUMERIC
F3B9 3F                 CMC             ;  TOGGLE THE CARRY BIT
F3BA D0                 RNC             ;  DONE IF SO
F3BB D607               SUI     'A'-'9'-1  ;SUBTRACT THE ALPHA BIAS
F3BD FE0A               CPI     10      ;  SET CARRY FOR INVALID CHAR
F3BF C9                 RET
                ;
                ; ROUTINE PCHK READS A CHARACTER FROM THE CONSOLE, THEN
                ;       CHECKS IT FOR A DELIMITER. IF IT IS NOT
                ;       A DELIMITER, A NON-ZERO CONDITION IS RETURNED.
                ;       IF IT IS A DELIMITER, A ZERO CONDITION IS RETURNED.
                ;       FURTHER, IF THE DELIMITER IS A CARRIAGE RETURN,
                ;       THE CARRY BIT IS SET.  A BLANK OR A COMMA RESET THE
                ;       CARRY BIT.
                ;
F3C0 CD7BF3     PCHK:   CALL    ECHO    ;GET, TEST FOR DELIMITER
F3C3 FE20       P2C:    CPI     ' '     ;  BLANK?
F3C5 C8                 RZ              ;  YES, DONE
F3C6 FE2C               CPI     ','     ;  NO, COMMA?
F3C8 C8                 RZ              ;  YES, DONE
F3C9 FE0D               CPI     CR      ;  NO, CARRIAGE RETURN?
F3CB 37                 STC             ;  SHOW IT IN CARRY BIT
F3CC C8                 RZ              ;  DONE IF CR
F3CD 3F                 CMC             ;CLEAR CARRY FOR NO DELIMITER
F3CE C9                 RET
                ;
                ;
                ; ROUTINE REST TRAPS ALL OF THE REGISTER CONTENTS WHENEVER A
                ;       RESTART 1 INSTRUCTION IS EXECUTED.  THE TRAPPED CONTENTS
                ;       ARE STORED IN THE SYSTEM STACK AREA FOR LATER ACCESS AND
                ;       USE BY THE GOTO AND THE EXAMINE REGISTERS COMMANDS.
                ;
                ; INSERT INTERRUPT DISABLER SOFTWARE AT START OF REST:
F3CF E5         REST:   PUSH    H       ;SAVE ALL THE REGISTERS
F3D0 D5                 PUSH    D
F3D1 C5                 PUSH    B
F3D2 F5                 PUSH    PSW
F3D3 CD6FF0             CALL    MEMSIZ  ;GET THE MONITOR'S STACK LOCATION
F3D6 EB                 XCHG
F3D7 210A00             LXI     H,10    ;GO UP 10 BYTES IN THE STACK
F3DA 39                 DAD     SP      ;  TO SKIP OVER TEMP REGISTER SAVE
F3DB 0604               MVI     B,4     ;PICK OFF THE REGISTER VALUES
F3DD EB                 XCHG
F3DE 2B         RS1:    DCX     H
F3DF 72                 MOV     M,D     ;SAVE IN WORK AREA
F3E0 2B                 DCX     H
F3E1 73                 MOV     M,E
F3E2 D1                 POP     D
                        DJNZ    RS1
F3E3+10F9
F3E5 C1                 POP     B       ;GET THE BREAKPOINT LOCATION
F3E6 0B                 DCX     B
F3E7 F9                 SPHL            ;SET THE MONITOR STACK
F3E8 212500             LXI     H,TLOCX ;SET UP TO RESTORE BREAKPOINTS
F3EB 39                 DAD     SP
F3EC D5                 PUSH    D
F3ED 1602               MVI     D,NBKPTS ;LOOP CONTROL FOR N BREAKPOINTS
F3EF 7E         RS2:    MOV     A,M
F3F0 91                 SUB     C       ;SEE IF A SOFTWARE TRAP
F3F1 23                 INX     H
F3F2 7E                 MOV     A,M
```

```
CP/M MACRO ASSEM 2.0    #015    DISK MOSS 2.2 MONITOR

 F3F3 98                SBB     B       ;MAYBE, TRY REST OF ADDRESS
                        JRZ     RS5     ;FOUND ONE, JUMP TO RESET IT
 F3F4+2806
 F3F6 23        RS3:    INX     H       ;NOT FOUND, TRY NEXT ONE
 F3F7 23                INX     H
 F3F8 15                DCR     D
                        JRNZ    RS2
 F3F9+20F4
 F3FB 03        RS4:    INX     B       ;NONE FOUND
 F3FC 212000    RS5:    LXI     H,LLOCX
 F3FF D1                POP     D
 F400 39                DAD     SP
 F401 73                MOV     M,E     ;STORE USER (H,L)
 F402 23                INX     H
 F403 72                MOV     M,D
 F404 C5                PUSH    B       ;SAVE (B,C)
 F405 0E2A              MVI     C,'*'   ;TYPE THE BREAK INDICATION
 F407 CD09F0            CALL    CONOUT
 F40A D1                POP     D       ;REGET THE BREAKPOINT LOCATION
 F40B 3EF4              MVI     A,RS9/256
 F40D BA                CMP     D       ;SEE IF A RET BREAKPOINT
                        JRZ     RS6
 F40E+2809
 F410 23                INX     H
 F411 23                INX     H
 F412 73                MOV     M,E     ;RESTORE USER PROGRAM COUNTER
 F413 23                INX     H
 F414 72                MOV     M,D
 F415 EB                XCHG            ;PRINT THE BREAKPOINT LOCATION
 F416 CDE1F5            CALL    LADR
 F419 212500    RS6:    LXI     H,TLOCX
 F41C 39                DAD     SP
 F41D 010002            LXI     B,NBKPTS*256
 F420 5E        RS7:    MOV     E,M     ;RESTORE BREAKPOINTED LOCATIONS
 F421 71                MOV     M,C     ;RESET SYSTEM BP SAVE AREA
 F422 23                INX     H
 F423 56                MOV     D,M
 F424 71                MOV     M,C
 F425 23                INX     H
 F426 7B                MOV     A,E
 F427 B2                ORA     D
                        JRZ     RS8     ;DO NOTHING IF ZERO
 F428+2802
 F42A 7E                MOV     A,M
 F42B 12                STAX    D
 F42C 23        RS8:    INX     H       ;SAME THING FOR OTHER
                        DJNZ    RS7     ; BREAKPOINT
 F42D+10F1
                        EXAF            ;NOW SAVE THE Z-80 UNIQUES
 F42F+08
                        EXX
 F430+D9
 F431 E5        E5              PUSH    H
 F432 D5                PUSH    D
 F433 C5                PUSH    B
 F434 F5                PUSH    PSW
                        PUSHIX
 F435+DDE5
                        PUSHIY
 F437+FDE5
                        LDAI
 F439+ED57
 F43B 47                MOV     B,A
                        LDAR
 F43C+ED5F
 F43E 4F                MOV     C,A
 F43F C5                PUSH    B
 F440 C313F1            JMP     WINITA  ;RETURN TO MONITOR
```

```
CP/M MACRO ASSEM 2.0    #016    DISK MOSS 2.2 MONITOR

F443 E5          RS9:   PUSH    H        ;RET BREAKPOINT ENCOUNTERED, ADJUST THE STACK
F444 CF                 RST     1        ;DO THE BREAKPOINT
                 ;
F445 C1          EXIT:  POP     B
F446 79                 MOV     A,C
                        STAR
F447+ED4F
F449 78                 MOV     A,B
                        STAI
F44A+ED47
                        POPIX
F44C+DDE1
                        POPIY
F44E+FDE1
F450 F1                 POP     PSW
F451 C1                 POP     B
F452 D1                 POP     D
F453 E1                 POP     H
                        EXAF
F454+08
                        EXX
F455+D9
F456 D1                 POP     D
F457 C1                 POP     B
F458 F1                 POP     PSW
F459 E1                 POP     H
F45A F9                 SPHL
F45B 00                 DB      0        ;PLACE FOR EI
F45C 210000             LXI     H,0
F45F C30000             JMP     0
F462 =           ENDX:  EQU     $
                 ;
                 ; ERROR HANDLERS
                 ;
                 ;       THREE TYPES OF ERRORS ARE DETECTED:  A RESTART
                 ;       ERROR; AN I/O ASSIGNMENT ERROR; AND CERTAIN PROGRAM
                 ;       ERRORS (DETERMINED BY THE PARTICULAR ROUTINE WHERE
                 ;       THE ERROR CONDITION WAS ENCOUNTERED.)  EACH CAUSES
                 ;       A UNIQUE MESSAGE TO BE PRINTED, THEN DOES A WARM
                 ;       INITIALIZATION OF THE MONITOR.  THE I/O ERROR
                 ;       CAUSES THE I/O ASSIGNMENTS TO BE RESET TO DEFAULT ASSIGNMENTS
                 ;
F462 AF          IOER:  XRA     A        ;SET IOBYTE TO DEFAULT VALUE
F463 320300             STA     IOBYTE
F466 216CF4             LXI     H,IOMSG  ;GET ADDRESS OF I/O ERROR MSG
F469 C3B5F6             JMP     COMERR   ;GO PROCESS IT
                 ;
F46C 492F4F2045IOMSG:   DB      'I/O ER','R'+80H
F473 44534B2045DERMSG:  DB      'DSK ERR: U','-'+80H
F47E 2054AD             DB      ' T','-'+80H
F481 2053AD             DB      ' S','-'+80H
F484 2043AD             DB      ' C','-'+80H
F487 2045AD             DB      ' E','-'+80H
F48A 0D8A               DB      CR,LF+80H
F48C 3F3F3FBF    QMSG:  DB      '???','?'+80H
F490 4D4F535320LOGMSG:  DB      'MOSS VERS 2.2'
F49D 0D8A               DB      CR,LF+80H
                 ;
                 ;
                 ; INITIALIZATION CODE FOR THE 8250 ASYNCHRONOUS COMMUNICATION
                 ;   ELEMENT.  THIS CODE WILL INITIALIZE THE BAUD RATE OF THE
                 ;   8250, AS WELL AS THE WORD FORMAT.  8 DATA BITS, 1 STOP BIT,
                 ;   AND NO PARITY ARE SELECTED.  EITHER 2 OR 3 CARRIAGE RETURNS
                 ;   MUST BE ENTERED TO ESTABLISH THE CORRECT BAUD RATE.
                 ;
F49F 3E0F        I8250: MVI     A,0FH    ;SET UP THE 8250
F4A1 D324               OUT     SMDMCT
F4A3 114000             LXI     D,40H    ;SET UP TO TIME THE START BIT
```

```
CP/M MACRO ASSEM 2.0    #017    DISK MOSS 2.2 MONITOR

 F4A6 62                MOV     H,D     ;MAKE (H,L)=0
 F4A7 6A                MOV     L,D
 F4A8 DB26      I8250A: IN      SMDMST  ;WAIT FOR START BIT
 F4AA A3                ANA     E
                        JRZ     I8250A

 F4AB+28FB
 F4AD DB26      I8250B: IN      SMDMST  ;NOW, TIME THE START BIT DURATION
 F4AF 23                INX     H
 F4B0 A3                ANA     E
 F4B1 A3                ANA     E
 F4B2 C2ADF4            JNZ     I8250B
 F4B5 E5                PUSH    H       ;SAVE COUNT IN CASE OF 4 MHZ
 F4B6 29                DAD     H       ;PREPARE THE 2 MHZ DIVISOR
 F4B7 5C                MOV     E,H     ;SET UP THE FUDGE FACTOR
 F4B8 19                DAD     D       ;APPLY THE FUDGE FACTOR
 F4B9 19                DAD     D
 F4BA E5                PUSH    H       ;SAVE FOR LATER USE
 F4BB 29                DAD     H       ;WAIT FOR 8 BIT TIMES
 F4BC 29                DAD     H
 F4BD DB20      I8250C: IN      SDATA   ;WASTE SOME TIME
 F4BF 2B                DCX     H
 F4C0 7D                MOV     A,L
 F4C1 B4                ORA     H
 F4C2 C2BDF4            JNZ     I8250C
 F4C5 E1                POP     H       ;REGET 2 MHZ DIVISOR
 F4C6 3E83      I8250D: MVI     A,83H   ;SET DIVISOR REGISTER ACCESS
 F4C8 D323              OUT     SLCTRL
 F4CA 7C                MOV     A,H
 F4CB D321              OUT     SINTEN
 F4CD 7D                MOV     A,L     ;SET THE DIVISOR
 F4CE D320              OUT     SDATA
 F4D0 3E03              MVI     A,3     ;SET DATA REGISTER ACCESS
 F4D2 D323              OUT     SLCTRL
 F4D4 AF                XRA     A       ;DISABLE INTERRUPTS
 F4D5 D321              OUT     SINTEN
 F4D7 D325              OUT     SLSTAT  ;AND RESET ERROR FLAGS
 F4D9 CDCEF6            CALL    TTYIN   ;GET A CHARACTER
 F4DC E67F              ANI     7FH     ;STRIP OFF ANY PARITY BIT
 F4DE FE0D              CPI     0DH     ;SEE IF IT IS A CARRIAGE RETURN
 F4E0 E1                POP     H       ;SET THE STACK STRAIGHT
 F4E1 C8                RZ              ;DONE IF CARRIAGE RETURN RECEIVED
 F4E2 5D                MOV     E,L     ;ELSE, MUST BE 4 MHZ SYSTEM
 F4E3 54                MOV     D,H     ; SO, COUNT=COUNT*5/4
 F4E4 CDEEF4            CALL    DIV2
 F4E7 CDEEF4            CALL    DIV2
 F4EA 19                DAD     D
 F4EB E5                PUSH    H
                        JMPR    I8250D  ;GO SET THE NEW DIVISOR
 F4EC+18D8
                ;
                ;
 F4EE B7        DIV2:   ORA     A       ;CLEAR THE CARRY BIT
 F4EF 7C                MOV     A,H     ;DO A 16-BIT RIGHT SHIFT
 F4F0 1F                RAR
 F4F1 67                MOV     H,A
 F4F2 7D                MOV     A,L
 F4F3 1F                RAR
 F4F4 6F                MOV     L,A
 F4F5 C9                RET
                ;
                ;
 F4F6 3E01      READ:   MVI     A,1     ;SET THE READ/WRITE FLAG
 F4F7                   ORG     $-1     ;SAVE A BYTE HERE
 F4F7 AF        WRITE:  XRA     A       ;RESET THE READ/WRITE FLAG
 F4F8 324B00            STA     RWFLG   ;SAVE THE FLAG
 F4FB 218000            LXI     H,80H
 F4FE 224900            SHLD    LUNIT   ;FORCE A READ ADDRESS COMMAND
 F501 CDA4F6            CALL    EXLF    ;GET THE START, STOP ADDRESS
```

```
CP/M MACRO ASSEM 2.0    #018    DISK MOSS 2.2 MONITOR

 F504 D5                PUSH    D       ;SAVE THE LIMIT
 F505 3A4B00    RW1:    LDA     RWFLG
 F508 B7                ORA     A       ;SEE IF READ OR WRITE
                        JRNZ    RW2     ;JUMP IF READ

 F509+2008
 F50B 224C00            SHLD    HSTBUF  ;SET THE WRITE SOURCE BUF
 F50E CDEBF6            CALL    DWRITE  ;ELSE, DO THE WRITE
                        JMPR    RW3

 F511+1803
 F513 CDE7F6    RW2:    CALL    DREADH  ;DO THE READ
 F516 D1        RW3:    POP     D
                        JRNZ    DERROR  ;JUMP IF ERROR

 F517+2067
 F519 3A4400            LDA     SPT     ;GET THE SECTORS PER TRACK
 F51C 47                MOV     B,A     ;SAVE IT
 F51D DB31              IN      DTRCK   ;SEE IF ON TRACK 00
 F51F B7                ORA     A
                        JRNZ    RW4     ;JUMP IF NOT

 F520+200B
 F522 061A              MVI     B,26    ;ELSE, SET THE SECTORS PER TRK 00
 F524 3A4A00            LDA     CUNIT
 F527 E610              ANI     10H
                        JRNZ    RW4

 F529+2002
 F52B 0612              MVI     B,18    ;MINI DRIVES
 F52D E5        RW4:    PUSH    H       ;SAVE THE DMA ADDRESS
 F52E 214200            LXI     H,SECTOR  ;SET UP MEMORY POINTER
 F531 7E                MOV     A,M     ;GET NUMBER OF SECTORS
 F532 B8                CMP     B       ;SEE IF TRACK OVERFLOW
                        JRC     RW5     ;JUMP IF NOT

 F533+381B
 F535 3A4500            LDA     TWOSID  ;SEE IF DOUBLE-SIDED
 F538 B7                ORA     A
                        JRZ     RW7     ;JUMP IF NOT

 F539+280B
 F53B 3A4300            LDA     SIDE    ;YES, SEE IF NEXT SIDE OR TRACK NEEDED
 F53E FED0              CPI     0D0H
                        JRNZ    RW7     ;NEXT TRACK, JUMP

 F540+2004
 F542 3E90              MVI     A,90H   ;ELSE, SET NEXT SIDE
                        JMPR    RW8

 F544+1805
 F546 3ED0      RW7:    MVI     A,0D0H
 F548 2B                DCX     H       ;ELSE, UPDATE THE TRACK
 F549 34                INR     M
 F54A 23                INX     H
 F54B 324300    RW8:    STA     SIDE
 F54E 3600              MVI     M,0     ;  AND THE SECTOR POINTER
 F550 34        RW5:    INR     M
 F551 E1                POP     H       ;RESTORE THE DMA ADDRESS
 F552 2B                DCX     H
 F553 CD9CF3            CALL    HILOX   ;SEE IF DONE
 F556 D5                PUSH    D       ;CONTINUE IF CONTROL RETURNED
                        JMPR    RW1

 F557+18AC
                ;
                ; ROUTINE DINIT CHECKS THE 2422'S AUTO-BOOT CONTROL BIT
                ;       DURING INITIALIZATION.  IT THEN TRANSFERS
                ;       CONTROL TO EITHER THE MONITOR OR THE BOOTSTRAP,
                ;       AS APPROPRIATE.
                ;
 F559 DB34      DINIT:  IN      DCNTL   ;SEE IF AUTO-BOOT WANTED
 F55B E640              ANI     40H
 F55D C0                RNZ             ;NO, RETURN TO MONITOR INITIALIZATION
                ;
                ; ROUTINE BOOT LOADS IN THE FIRST TWO SECTORS OF
                ;       DRIVE 00 INTO LOCATIONS 80H-17FH, THEN
                ;       TRANSFERS PROGRAM CONTROL TO LOCATION 80H.
```

```
CP/M MACRO ASSEM 2.0    #019    DISK MOSS 2.2 MONITOR

                ;       IT EXPECTS THE DOS LOADER TO BE ON THESE
                ;       TWO SECTORS.
                ;
F55E 210000     BOOT:   LXI     H,0     ;SET UP THE DISK PARMS
F561 224000             SHLD    DISKNO
F564 2101D0             LXI     H,0D001H  ;SIDE 0, SECTOR 1
F567 224200             SHLD    SECTOR
F56A 218000             LXI     H,TBUF
F56D 224900             SHLD    LUNIT   ;FORCE A DISK DETERMINATION
F570 CDE7F6             CALL    DREADH  ;GO GET A SECTOR
                        JRNZ    DERROR  ;QUIT IF AN ERROR ENCOUNTERED

F573+200B
F575 3E02               MVI     A,2     ;GET SECTOR 2, ALSO
F577 324200             STA     SECTOR
F57A CDE7F6             CALL    DREADH
F57D CA8000             JZ      TBUF    ;GO TO THE LOADER
                ;
F580 2173F4     DERROR: LXI     H,DERMSG  ;ADDRESS OF DISK ERROR MESSAGE
F583 CD95F6             CALL    PRTWD   ;START THE MESSAGE
F586 3A4000             LDA     DISKNO  ;DO THE UNIT ASSIGNMENT
F589 CDA1F5             CALL    DERR1
F58C 3A4100             LDA     TRACK   ;AND THE TRACK
F58F CDA1F5             CALL    DERR1
F592 3A4200             LDA     SECTOR  ;AND THE SECTOR
F595 CDA1F5             CALL    DERR1
F598 3A4800             LDA     CMND    ;AND THE COMMAND
F59B CDA1F5             CALL    DERR1
F59E 3A4700             LDA     STATUS  ;AND THE STATUS
F5A1 CDE6F5     DERR1:  CALL    HEX1    ;OUTPUT IT IN HEX
F5A4 C398F6             JMP     PRTWA   ;CONTINUE THE MESSAGE
                ;
                ; SET DISK PARAMETERS ROUTINE EXPECTS THREE PARAMETERS
                ;       TO BE ENTERED FROM THE CONSOLE.  THESE PARAMETERS
                ;       ARE:  UNIT NUMBER (0-3); SECTORS PER TRACK;
                ;       AND DOUBLE-SIDED SWITCH (0 OR NON-0).
                ;       ONLY THE UNIT NUMBER IS CHECKED FOR ERRORS.
                ;
                ;       THIS ROUTINE MUST BE CALLED BEFORE USE OF EITHER
                ;       THE DISK READ OR WRITE ROUTINE.
                ;
F5A7 CD86F3     PARM:   CALL    EXPR3   ;GET THE THREE PARAMETERS
F5AA 7D                 MOV     A,L     ;ERROR CHECK THE UNIT ASSIGNMENT
F5AB B7                 ORA     A
F5AC FA09F1             JM      QPRT
F5AF FE04               CPI     4
F5B1 D209F1             JNC     QPRT
F5B4 324000             STA     DISKNO  ;SET THE UNIT SELECT
F5B7 6B                 MOV     L,E     ;MOVE THE SECTORS PER TRACK OVER
F5B8 61                 MOV     H,C     ;  AND THE TWO-SIDED SWITCH
F5B9 224400             SHLD    SPT     ;STORE THEM
F5BC C9                 RET
                ;
                ; ROUTINE QPARM ALSO SETS CERTAIN DISK PARAMETERS.  IN THIS
                ;       CASE, THE DESIRED START TRACK, SIDE, AND SECTOR ARE
                ;       SET.  THESE PARAMETERS NEED ONLY BE SET PRIOR TO THE
                ;       FIRST DISK ACCESS, OR WHEN A NON-CONTIGUOUS DISK ACCESS
                ;       IS DESIRED.  IF THE PARAMETERS ARE NOT RESET BETWEEN
                ;       DISK ACCESSES, THE DATA TRANSFER WILL OCCUR TO/FROM
                ;       THE NEXT LOGICALLY SEQUENTIAL DISK LOCATIONS.
                ;
F5BD CD86F3     QPARM:  CALL    EXPR3   ;GET THE THREE PARAMETERS
F5C0 61                 MOV     H,C     ;MOVE OVER THE START SECTOR
F5C1 224100             SHLD    TRACK   ;STORE THE TRACK AND SECTOR
F5C4 7B                 MOV     A,E     ;GET THE SIDE INDICATOR
F5C5 B7                 ORA     A       ;SEE IF SINGLE-SIDED
F5C6 3ED0               MVI     A,0D0H  ;SIDE 0 SELECT BITS
                        JRZ     QPARM1  ;JUMP IF SO

F5C8+2802
```

```
CP/M MACRO ASSEM 2.0     #020     DISK MOSS 2.2 MONITOR

F5CA 3E90                MVI      A,90H    ;ELSE, SET THE SIDE 1 CONTROL BIT
F5CC 324300      QPARM1: STA      SIDE     ;SAVE IT
F5CF C9                  RET
                 ;
                 ; HEXN ROUTINE
                 ;
                 ;THIS ROUTINE ADDS AND SUBTRACTS TWO HEXADECIMAL 16 BIT
                 ;       UNSIGNED NUMBERS AND DISPLAYS THE RESULTS ON THE
                 ;       CONSOLE.
                 ;
F5D0 CDA4F6      HEXN:   CALL     EXLF     ;GET THE TWO NUMBERS
F5D3 E5                  PUSH     H        ;SAVE IT FOR THE SUBTRACT
F5D4 19                  DAD      D        ;ADD THEM
F5D5 CDFBF5              CALL     LADRB    ;OUTPUT THEM
F5D8 E1                  POP      H        ;REGET THE FIRST NUMBER
F5D9 B7                  ORA      A        ;CLEAR THE CARRY BIT
                         DSBC     D        ;DO THE SUBTRACT
F5DA+ED52
                         JMPR     LADR     ;GO OUTPUT THE RESULT
F5DC+1803
                 ;
                 ; ROUTINE LADR PRINTS THE CONTENTS OF (H,L) ON THE
                 ;       CURRENT CONSOLE, EITHER AT THE START OF A NEW
                 ;       LINE (EP = LADRA) OR AT THE CURRENT LOCATION (EP
                 ;       = LADR).
                 ;
F5DE CDA9F6      LADRA:  CALL     CRLF     ;START A NEW LINE
F5E1 7C          LADR:   MOV      A,H      ;GET HIGH TWO DIGITS
F5E2 CDE6F5              CALL     HEX1     ;PRINT THEM
F5E5 7D                  MOV      A,L      ;GET LOW TWO DIGITS
F5E6 F5          HEX1:   PUSH     PSW      ;SAVE THE LOW DIGIT
F5E7 0F                  RRC               ;PUT HIGH NIBBLE INTO BITS 0-3
F5E8 0F                  RRC
F5E9 0F                  RRC
F5EA 0F                  RRC
F5EB CDEFF5              CALL     HEX2     ;GO PRINT SINGLE DIGIT
F5EE F1                  POP      PSW      ;REGET THE LOW DIGIT
F5EF CD6EF3      HEX2:   CALL     CONV     ;GO INSERT ASCII ZONE
                         JMPR     CO       ;DO THE CHARACTER OUTPUT
F5F2+180C
                 ;
                 ; ROUTINE DASH TYPES A DASH ON THE CURRENT CONSOLE DEVICE.
                 ;
F5F4 CDE6F5      DASH1:  CALL     HEX1     ;FIRST, PRINT ACCUM AS TWO HEX DIGITS
F5F7 0E2D        DASH:   MVI      C,'-'    ;GET AN ASCII DASH
                         JMPR     CO       ;GO TYPE IT
F5F9+1805
                 ;
                 ; IOBYTE HANDLERS
                 ;
F5FB                     ORG      MOSS+5FBH
F5FB CDDEF5      LADRB:  CALL     LADRA    ;OUTPUT (H,L) AS 4 ASCII DIGITS
                 ;
F5FE 0E20        BLK:    MVI      C,' '    ;OUPTUT A BLANK
                 ;
F600 3A0300      CO:     LDA      IOBYTE
F603 E603                ANI      3        ;ISOLATE CONSOLE ASGT
F605 CADEF6              JZ       TTYOUT   ;TTY DEVICE ACTIVE
F608 FE02                CPI      2
F60A FA62F4              JM       CRTOUT   ;CRT ACTIVE
F60D C262F4              JNZ      CUSO1    ;USER CONSOLE 1 ACTIVE
                 ;
F610 3A0300      LO:     LDA      IOBYTE
F613 E6C0                ANI      0C0H     ;ISOLATE LIST ASGT
F615 CADEF6              JZ       TTYOUT   ;TTY DEVICE ACTIVE
F618 FE80                CPI      80H
F61A FA62F4              JM       CRTOUT   ;CRT ACTIVE
F61D CA62F4              JZ       LPRT     ;LINE PRINTER ACTIVE
```

```
CP/M MACRO ASSEM 2.0    #021    DISK MOSS 2.2 MONITOR

F620 C362F4             JMP     LUSE1   ;USER PRINTER 1 ACTIVE

F623 3A0300     CSTS:   LDA     IOBYTE
F626 E603               ANI     3       ;ISOLATE CONSOLE ASGT
F628 CAC6F6             JZ      TTST    ;TTY ACTIVE
F62B FE02               CPI     2
F62D FA62F4             JM      CRTST   ;CRT ACTIVE
F630 C262F4             JNZ     CUST1   ;USER CONSOLE 1 ACTIVE

F633 3A0300     BATST:  LDA     IOBYTE
F636 E60C               ANI     0CH     ;ISOLATE BATCH ASGT
F638 CAC6F6             JZ      TTST    ;TTY ACTIVE
F63B FE08               CPI     8
F63D FA62F4             JM      PTRST   ;PAPER TAPE READER ACTIVE
F640 CA62F4             JZ      RUST1   ;USER READER 1 ACTIVE
F643 C362F4             JMP     RUST2   ;USER READER 2 ACTIVE

F646 3A0300     CI:     LDA     IOBYTE
F649 E603               ANI     3       ;ISOLATE CONSOLE ASGT
F64B CACEF6             JZ      TTYIN   ;TTY DEVICE ACTIVE
F64E FE02               CPI     2
F650 FA62F4             JM      CRTIN   ;CRT ACTIVE
F653 C262F4             JNZ     CUSI1   ;USER CONSOLE 1 ACTIVE

F656 3A0300     RI:     LDA     IOBYTE
F659 E60C               ANI     0CH     ;ISOLATE BATCH ASGT
F65B CACEF6             JZ      TTYRDR  ;TTY ACTIVE
F65E FE08               CPI     8
F660 FA62F4             JM      PTRIN   ;PAPER TAPE READER ACTIVE
F663 CA62F4             JZ      RUSI1   ;USER READER 1 ACTIVE
F666 C362F4             JMP     RUSI2   ;USER READER 2 ACTIVE

F669 3A0300     LSTAT:  LDA     IOBYTE
F66C E6C0               ANI     0C0H    ;ISOLATE THE LIST DEVICE ASSIGNMENT
F66E CAD6F6             JZ      TTOST
F671 FE80               CPI     80H
F673 FA62F4             JM      CRTOST
F676 CA62F4             JZ      LPRST
F679 C362F4             JMP     LUST1

F67C 3A0300     PO:     LDA     IOBYTE
F67F E630               ANI     30H     ;ISOLATE PUNCH ASGT
F681 CADEF6             JZ      TTPNCH  ;TTY ACTIVE
F684 FE20               CPI     20H
F686 FA62F4             JM      HSP     ;HIGH SPEED PUNCH ACTIVE
F689 CA62F4             JZ      PUSO1   ;USER PUNCH 1 ACTIVE
F68C C362F4             JMP     PUSO2   ;USER PUNCH 2 ACTIVE

                ;
                ; ROUTINE CONI READS THE CONSOLE AND STRIPS OFF THE ASCII
                ;       PARITY BIT.
                ;
F68F CD46F6     CONI:   CALL    CI      ;GET THE NEXT CHARACTER
F692 E67F               ANI     7FH     ;STRIP OFF THE PARITY BIT
F694 C9         RTS:    RET

                ;
                ; ROUTINE PRTWD PRINTS AN ASCII STRING ONTO THE CONSOLE.
                ;       THE STRING MUST BE TERMINATED BY BIT 7 SET IN THE
                ;       LAST CHARACTER OF THE STRING.  THE STRING WILL START
                ;       A NEW LINE (EP = PRTWD) OR CONTINUE ON THE SAME
                ;       LINE (EP = PRTWA)
                ;
F695 CDA9F6     PRTWD:  CALL    CRLF    ;START A NEW LINE
F698 C5         PRTWA:  PUSH    B       ;SAVE (B,C)
F699 4E         PRTA:   MOV     C,M     ;GET NEXT CHARACTER FROM MEMORY
F69A CD00F6             CALL    CO      ;OUTPUT IT
F69D 23                 INX     H       ;INCREMENT MEMORY POINTER
F69E 79                 MOV     A,C
F69F 07                 RLC             ;TEST FOR BIT 7 DELIMITER
```

```
CP/M MACRO ASSEM 2.0     #022     DISK MOSS 2.2 MONITOR

                         JRNC     PRTA      ;NO DELIMITER, GO DO NEXT CHARACTER
F6A0+30F7
F6A2 C1          PRTB:   POP      B         ;RESTORE (B,C)
F6A3 C9                  RET

                 ;
                 ; ROUTINE EXLF READS TWO PARAMETERS, PUTS THEM INTO THE
                 ;       D,E AND H,L REGISTERS, THEN DOES A CARRIAGE RETURN,
                 ;       LINE FEED SEQUENCE.
                 ;
F6A4 CDD9F0      EXLF:   CALL     EXPR      ;GO GET TWO PARAMETERS
F6A7 D1                  POP      D
F6A8 E1                  POP      H

                 ;
                 ; ROUTINE CRLF GENERATES A CARRIAGE RETURN, LINE FEED
                 ;       SEQUENCE ON THE CURRENT CONSOLE TO START A NEW LINE
                 ;       IT INCLUDES TWO NULL CHARACTERS FOR TTY TYPE
                 ;       DEVICES FOR THE HEAD MOVEMENT TIME.
                 ;
F6A9 E5          CRLF:   PUSH     H         ;SAVE THE CONTENTS OF (H,L)
F6AA 21C2F6      CRLFA:  LXI      H,CRMSG   ;ADDRESS OF CR,LF MESSAGE
F6AD CD98F6              CALL     PRTWA     ;  OUTPUT IT
F6B0 E1                  POP      H         ;RESTORE (H,L)
F6B1 C9                  RET

                 ;
F6B2 21BBF6      RSTER:  LXI      H,RSTMSG  ;GET ADDRESS OF RESTART ERROR MSG
F6B5 CD95F6      COMERR: CALL     PRTWD     ;PRINT IT ON NEW LINE
F6B8 C30000              JMP      WSVEC     ;GO TO WARM BOOT

                 ;
F6BB 5253542045  RSTMSG: DB       'RST ER','R'+80H
F6C2 0D0A0080    CRMSG:  DB       CR,LF,0,80H

                 ;
                 ; I/O DRIVERS FOR THE 8250 ASYNC COMM ELEMENT
                 ;
F6C6 DB25        TTST:   IN       SLSTAT    ;GET 8250 LINE STATUS
F6C8 E601                ANI      1         ;SEE IF RECEIVE DATA AVAILABLE
F6CA C8                  RZ                 ;RETURN IF NOT
F6CB C6FE                ADI      0FEH      ;FLAG THAT DATA IS AVAILABLE
F6CD C9                  RET

                 ;
F6CE DB25        TTYIN:  IN       SLSTAT    ;GET 8250 LINE STATUS
F6D0 1F                  RAR                ;MOVE RX DATA READY BIT INTO CARRY
                         JRNC     TTYIN     ;LOOP UNTIL DATA IS IN
F6D1+30FB
F6D3 DB20                IN       SDATA     ;READ THE DATA
F6D5 C9                  RET

                 ;
F6D6 DB25        TTOST:  IN       SLSTAT    ;GET 8250 LINE STATUS
F6D8 E620                ANI      20H       ;ISOLATE TX BUFFER EMPTY BIT
F6DA C8                  RZ                 ;RETURN IF NOT EMPTY
F6DB C6BF                ADI      0BFH      ;FLAG THE EMPTY STATE
F6DD C9                  RET

                 ;
F6DE CDD6F6      TTYOUT: CALL     TTOST     ;GET 8250 LINE STATUS
                         JRZ      TTYOUT    ;WAIT UNTIL ONE OF THE REGISTERS EMPTIES
F6E1+28FB
F6E3 79                  MOV      A,C       ;MOVE THE DATA OVER
F6E4 D320                OUT      SDATA     ;OUTPUT THE DATA
F6E6 C9                  RET

                 ;
                 ; EQUATES FOR ADDITIONAL CONSOLE DEVICES
                 ;
F462 =           CRTIN:  EQU      IOER
F462 =           CRTOUT: EQU      IOER
F462 =           CRTST:  EQU      IOER
F462 =           CRTOST: EQU      IOER      ;UNASSIGNED CRT OUTPUT STATUS
F462 =           CUSI1:  EQU      IOER      ;UNASSIGNED USER CONSOLE (INPUT)
F462 =           CUSO1:  EQU      IOER      ;UNASSIGNED USER CONSOLE (OUPTUT)
F462 =           CUST1:  EQU      IOER
```

```
CP/M MACRO ASSEM 2.0    #023    DISK MOSS 2.2 MONITOR

                        ;
                        ;  EQUATES FOR ADDITIONAL PAPER TAPE PUNCH DEVICES
                        ;
F6DE =          TTPNCH: EQU     TTYOUT  ;UNASSIGNED TELETYPE PUNCH
F462 =          HSP:    EQU     IOER    ;UNASSIGNED HIGH SPEED PUNCH
F462 =          HSPST:  EQU     IOER    ;UNASSIGNED HIGH SPEED PUNCH STATUS
F462 =          PUSO1:  EQU     IOER    ;UNASSIGNED USER PUNCH 1
F462 =          PUSO2:  EQU     IOER    ;UNASSIGNED USER PUNCH 2
                        ;
                        ;  EQUATES FOR ADDITIONAL LIST DEVICES
                        ;
F462 =          LPRT:   EQU     IOER    ;UNASSIGNED LINE PRINTER
F462 =          LPRST:  EQU     IOER    ;UNASSIGNED LINE PRINTER STATUS
F462 =          LUSE1:  EQU     IOER    ;LIST DEVICE 1
F462 =          LUST1:  EQU     IOER    ;UNASSIGNED LIST DEVICE 1 STATUS
                        ;
                        ;  EQUATES FOR ADDITIONAL PAPER TAPE READER DEVICES
                        ;
F6CE =          TTYRDR: EQU     TTYIN   ;UNASSIGNED TELETYPE PAPER TAPE READER
F462 =          PTRIN:  EQU     IOER    ;UNASSIGNED HIGH SPEED PAPER TAPE READER
F462 =          PTRST:  EQU     IOER    ;UNASSIGNED HS PTR STATUS
F462 =          RUSI1:  EQU     IOER    ;UNASSIGNED PAPER TAPE READER 1
F462 =          RUST1:  EQU     IOER    ;UNASSIGNED PAPER TAPE READER 1 (STATUS)
F462 =          RUSI2:  EQU     IOER    ;UNASSIGNED PAPER TAPE READER 2
F462 =          RUST2:  EQU     IOER    ;UNASSIGNED PAPER TAPE READER 2 (STATUS)
                        ;
                        ; THE FOLLOWING ROUTINES DO THE PRIMITIVE DISK ACCESSES.
                        ;       IN ALL CASES, ONE SECTOR OF DATA IS TRANSFERRED.
                        ;       IF THE DISK HAS NOT BEEN PREVIOUSLY ACCESSED,
                        ;       THESE ROUTINES WILL AUTOMATICALLY DETERMINE THE
                        ;       DISK TYPE (8" OR 5"), SINGLE OR DOUBLE DENSITY,
                        ;       AND SECTOR SIZE.
                        ;
                        ;       BEFORE THE DESIRED DATA IS TRANSFERRED, THE DESIRED
                        ;       TRACK IS SEEKED OUT, THE DESIRED SECTOR AND SIDE IS
                        ;       SET, THEN THE ACTUAL DATA TRANSFER.
                        ;
                        ;       UP TO TEN TRIES WILL BE ATTEMPTED BEFORE THE DATA
                        ;       TRANSFER IS ABORTED.  ON RETURN TO THE CALLING
                        ;       ROUTINE, THE A REGISTER WILL CONTAIN A ZERO IF THE
                        ;       OPERATION WAS SUCCESSFUL, OR NON-ZERO IF NOT
                        ;       SUCCESSFUL.  THE FLAG REGISTER WILL NOT NECESSARILY
                        ;       CORRESPOND WITH THE A REGISTER CONTENT.
                        ;
                        ;       THESE ROUTINES ARE CP/M COMPATABLE, AND MAY BE USED
                        ;       AS PART OF THE BIOS.
                        ;
F6E7 224C00     DREADH: SHLD    HSTBUF  ;SAVE THE DMA ADDRESS
F6EA 3E01       DREAD:  MVI     A,1     ;SET READ FLAG
F6EB                    ORG     $-1     ;SAVE A BYTE HERE
F6EB AF         DWRITE: XRA     A       ;SET WRITE FLAG
F6EC 324B00             STA     RWFLG   ;SAVE IT FOR LATER USE
F6EF 060A               MVI     B,10    ;NUMBER OF RETRIES
F6F1 C5         AGN:    PUSH    B
F6F2 CD3BF7             CALL    SEEK
F6F5 CCFDF6             CZ      RDWR
F6F8 C1         READ3:  POP     B
F6F9 C8                 RZ
                        DJNZ    AGN
F6FA+10F5
F6FC C9                 RET
                        ;
F6FD 5F         RDWR:   MOV     E,A     ;SAVE COMMAND
F6FE 3A4B00             LDA     RWFLG
F701 B7                 ORA     A
F702 7B                 MOV     A,E     ;REGET THE COMMAND
```

```
CP/M MACRO ASSEM 2.0    #024    DISK MOSS 2.2 MONITOR

                        JRZ     WRDAT   ;WRITE IF ZERO

F703+2810
F705 324800     RDAT:   STA     CMND
F708 D330               OUT     DCMMD   ;DISK COMMAND PORT
                READ1:  INIR
F70A+EDB2
F70C 15                 DCR     D
                        JRNZ    READ1

F70D+20FB
F70F CD2EF7             CALL    EOJ
F712 E69C               ANI     9CH     ;ISOLATE READ ERROR BITS
F714 C9                 RET
                ;
F715 F620       WRDAT:  ORI     20H     ;ADD WRITE COMMAND
F717 324800             STA     CMND
F71A D330               OUT     DCMMD   ;DISK COMMAND PORT
                WRT1:   OUTIR           ;DO THE OUTPUT
F71C+EDB3
F71E 15                 DCR     D       ;IN CASE > 256 BYTES
                        JRNZ    WRT1

F71F+20FB
                        JMPR    EOJ

F721+180B
                ;
F723 0608       EOJB:   MVI     B,8     ;BASIS OF RESTORE COMMAND
F725 3A4600     EOJA:   LDA     STPRAT  ;GET THE STEP RATE BITS
F728 B0                 ORA     B       ;ADD ON THE COMMAND
F729 324800             STA     CMND
F72C D330               OUT     DCMMD   ;DO THE COMMAND
F72E DB34       EOJ:    IN      DFLAG   ;DISK FLAG PORT
F730 1F                 RAR
                        JRNC    EOJ

F731+30FB
F733 DB30       EOJ1:   IN      DSTAT   ;GET THE DISK STATUS
F735 324700             STA     STATUS
F738 E6FC               ANI     0FCH
F73A C9                 RET
                ;
F73B CD8EF7     SEEK:   CALL    IDRD    ;INSURE HEADER HAS BEEN READ
F73E C423F7             CNZ     EOJB    ;RESTORE THE DRIVE IF ERROR
F741 F8                 RM              ;DONE IF NO DRIVE
F742 3A4200     SEEK1:  LDA     SECTOR  ;SET THE SECTOR
F745 D332               OUT     DSCTR   ;DISK SECTOR PORT
F747 DB31               IN      DTRCK   ;DISK TRACK PORT
F749 4F                 MOV     C,A     ;SAVE IT
F74A 3A4100             LDA     TRACK   ;GET DESIRED TRACK
F74D B9                 CMP     C
                        JRZ     RDWRT   ;JUMP IF NO SEEK NEEDED

F74E+280C
F750 D333               OUT     DDATA   ;SET THE SEEK TRACK
F752 061C               MVI     B,1CH   ;BUILD THE SEEK COMMAND
F754 CD25F7             CALL    EOJA    ;DO THE SEEK
F757 E698               ANI     98H     ;SEEK ERROR MASK
F759 C0                 RNZ             ;DONE IF SEEK ERROR
F75A DB31               IN      DTRCK   ;CHECK FOR TRACK 00
F75C B7         RDWRT:  ORA     A
F75D 214000             LXI     H,40H   ;BUILD SECTOR BYTE COUNT
                        JRZ     RDWRT0  ;JUMP IF TRACK 00

F760+2803
F762 3A5100             LDA     IDSV+3  ;GET SECTOR SIZE
F765 29         RDWRT0: DAD     H       ;DOUBLE (H,L)
F766 3D                 DCR     A       ;LOOP CONTROL
F767 F265F7             JP      RDWRT0
F76A E5                 PUSH    H
F76B 0E80               MVI     C,80H   ;AUTO-WAIT BIT
F76D CDC3F7             CALL    SETUP
F770 DB34               IN      DFLAG   ;DISK FLAG PORT
F772 E620               ANI     20H     ;SEE IF HEAD IS LOADED
```

```
CP/M MACRO ASSEM 2.0    #025    DISK MOSS 2.2 MONITOR

 F774 3E04              MVI     A,4
                        JRZ     RDWRT1  ;JUMP IF NOT
 F776+2801
 F778 AF                XRA     A       ;ELSE, RESET THE HEAD LOAD FLAG
 F779 C688      RDWRT1: ADI     88H     ;BUILD A READ SECTOR COMMAND
 F77B 2A4C00            LHLD    HSTBUF  ;GET THE DMA ADDRESS
 F77E D1                POP     D       ;GET THE BYTE COUNT
 F77F 43                MOV     B,E     ;SET UP FOR Z-80 I/O
 F780 15                DCR     D       ;SEE IF 128 BYTE SECTOR
 F781 14                INR     D
                        JRNZ    RDWRT2  ;JUMP IF NOT
 F782+2001
 F784 14                INR     D
 F785 0E33      RDWRT2: MVI     C,DDATA
 F787 BF                CMP     A       ;CLEAR THE FLAGS
 F788 C9                RET
                ;
 F789 0658      IDRD5:  MVI     B,58H   ;BUILD A STEP-IN COMMAND
 F78B CD25F7            CALL    EOJA
 F78E 2A4900    IDRD:   LHLD    LUNIT
 F791 7C                MOV     A,H     ;GET THE CUNIT VALUE
 F792 BD                CMP     L       ;SEE IF SAME AS LUNIT
 F793 C8                RZ              ;RETURN IF SO
 F794 0E80      IDRD1:  MVI     C,80H   ;SET THE AUTO-WAIT BIT
 F796 CDC3F7            CALL    SETUP
 F799 CD33F7            CALL    EOJ1    ;INSURE A DRIVE IS THERE
 F79C F8                RM              ;ERROR IF NOT
 F79D E5                PUSH    H       ;SAVE POINTER
 F79E 214E00            LXI     H,IDSV  ;SET UP TO READ ADDRESS
 F7A1 013306            LXI     B,600H+DDATA
 F7A4 1601              MVI     D,1
 F7A6 3EC4              MVI     A,0C4H  ;READ ADDRESS COMMAND
 F7A8 CD05F7            CALL    RDAT
 F7AB E1                POP     H       ;RESTORE POINTER
                        JRZ     IDRD2   ;JUMP IF GOOD READ
 F7AC+2808
 F7AE 3E40              MVI     A,40H   ;SEE IF DDEN IS SET
 F7B0 BE                CMP     M
 F7B1 D8                RC              ;TAKE THE ERROR IF SO
 F7B2 B6                ORA     M       ;ELSE, TRY DDEN
 F7B3 77                MOV     M,A
                        JMPR    IDRD
 F7B4+18D8
                ;
 F7B6 DB32      IDRD2:  IN      DSCTR   ;GET THE TRACK NUMBER
 F7B8 D331              OUT     DTRCK   ;SET THE TRACK REGISTER
 F7BA B7                ORA     A       ;INSURE NOT ON TRACK 0
                        JRZ     IDRD5   ;JUMP IF NOT OKAY
 F7BB+28CC
 F7BD 7E                MOV     A,M     ;REGET SELBITS
 F7BE 324900            STA     LUNIT   ;UPDATE LAST USED UNIT
 F7C1 AF                XRA     A       ;RESET ERROR FLAGS
 F7C2 C9                RET
                ;
                ;SET UP DRIVE NUMBER
 F7C3 214A00    SETUP:  LXI     H,CUNIT ;SEE IF DRIVE HAS BEEN ACTIVE
 F7C6 7E                MOV     A,M     ;GET THE SELBITS
 F7C7 B7                ORA     A       ;SEE IF SET UP YET
                        JRNZ    SU0     ;YES, SKIP INIT CODE
 F7C8+2025
                ;
 F7CA 3A4000    SETIT:  LDA     DISKNO  ;GET THE DESIRED DRIVE
 F7CD 47                MOV     B,A     ;SAVE IN WORK REGISTER
 F7CE 04                INR     B       ;PREPARE TO CONVERT TO SELBITS
 F7CF AF                XRA     A       ;ZERO TO A
 F7D0 37                STC             ;DRIVE SELECT BIT
 F7D1 17        SET1:   RAL             ;SHIFT BIT INTO POSITION
                        DJNZ    SET1    ;LOOP TIL BIT IS IN POSITION
```

```
CP/M MACRO ASSEM 2.0     #026     DISK MOSS 2.2 MONITOR

 F7D2+10FD
 F7D4 F620               ORI      20H       ;ADD ON MOTOR ON BIT
 F7D6 77                 MOV      M,A       ;SAVE IT
 F7D7 D334               OUT      DCNTL     ;SELECT THE DRIVE
 F7D9 114600             LXI      D,STPRAT  ;SET INITIAL STEP RATE
 F7DC 3E03               MVI      A,3       ; TO SLOWEST POSSIBLE
 F7DE 12                 STAX     D
 F7DF CD23F7             CALL     EOJB      ;RESTORE THE DRIVE
 F7E2 F8                 RM                 ;DONE IF DRIVE NOT READY
 F7E3 DB04               IN       4         ;READ THE MINI TRK00 BIT
 F7E5 1F                 RAR                ;ISOLATE IT
                         JRNC     SU0       ;JUMP IF MINI DRIVE

 F7E6+3007
 F7E8 3E10               MVI      A,10H     ;ELSE, ADD ON MAXI BIT
 F7EA B6                 ORA      M
 F7EB 77                 MOV      M,A
 F7EC 3E02               MVI      A,2       ;SET MAXI STEP RATE
 F7EE 12                 STAX     D
 F7EF DB31       SU0:    IN       DTRCK     ;ELSE, SEE IF TRACK ZERO
 F7F1 B7                 ORA      A
 F7F2 7E                 MOV      A,M       ;REGET THE SELBITS
                         JRNZ     SU1

 F7F3+2002
 F7F5 E6BF               ANI      0BFH      ;INSURE DDEN IS RESET
 F7F7 B1         SU1:    ORA      C         ;ADD ON AUTOWAIT BIT
 F7F8 D334               OUT      DCNTL     ;OUTPUT THE SELBITS
 F7FA 3A4300             LDA      SIDE      ;SET THE SIDE SELECT
 F7FD D304               OUT      4
 F7FF C9                 RET
```

APPENDIX F

LIMITED WARRANTY

California Computer Systems (CCS) warrants to the original purchaser of its products that its CCS assembled and tested products will be free from materials defects for a period of one (1) year, and be free from defects of workmanship for a period of ninety (90) days.

The responsibility of CCS hereunder, and the sole and exclusive remedy of the original purchaser for a breach of any warranty hereunder, is limited to the correction or replacement by CCS at CCS's option, at CCS's service facility, of any product or part which has been returned to CCS and in which there is a defect covered by this warranty; provided, however, that in the case of CCS assembled and tested products, CCS will correct any defect in materials and workmanship free of charge if the product is returned to CCS within ninety (90) days of original purchase from CCS; and CCS will correct defects in materials in its products and restore the product to an operational status for a labor charge of $25.00, provided that the product is returned to CCS within one (1) year in the case of CCS assembled and tested products. All such returned products shall be shipped prepaid and insured by original purchaser to:

Warranty Service Department
California Computer Systems
250 Caribbean Drive
Sunnyvale, California
94086

CCS shall have the right of final determination as to the existence and cause of a defect, and CCS shall have the sole right to decide whether the product should be repaired or replaced.

This warranty shall not apply to any product or any part thereof which has been subject to

(1) accident, neglect, negligence, abuse or misuse;

(2) any maintenance, overhaul, installation, storage, operation, or use, which is improper; or

(3) any alteration, modification, or repair by anyone other than CCS or its authorized representative.

THIS WARRANTY IS EXPRESSLY IN LIEU OF ALL OTHER WARRANTIES EXPRESSED OR IMPLIED OR STATUTORY INCLUDING THE WARRANTIES OF DESIGN, MERCHANTABILITY, OR FITNESS OR SUITABILITY FOR USE OR INTENDED PURPOSE AND OF ALL OTHER OBLIGATIONS OR LIABILITIES OF CCS. To any extent that this warranty cannot exclude or disclaim implied warranties, such warranties are limited to the duration

of this express warranty or to any shorter time permitted by law.

CCS expressly disclaims any and all liability arising from the use and/or operation of its products sold in any and all applications not specifically recommended, tested, or certified by CCS, in writing. With respect to applications not specifically recommended, tested, or certified by CCS, the original purchaser acknowledges that he has examined the products to which this warranty attaches, and their specifications and descriptions, and is familiar with the operational characteristics thereof. The original purchaser has not relied upon the judgement or any representations of CCS as to the suitability of any CCS product and acknowledges that CCS has no knowledge of the intended use of its products. CCS EXPRESSLY DISCLAIMS ANY LIABILITY ARISING FROM THE USE AND/OR OPERATION OF ITS PRODUCTS, AND SHALL NOT BE LIABLE FOR ANY CONSEQUENTIAL OR INCIDENTAL OR COLLATERAL DAMAGES OR INJURY TO PERSONS OR PROPERTY.

CCS's obligations under this warranty are conditioned on the original purchaser's maintenance of explicit records which will accurately reflect operating conditions and maintenance preformed on CCS's products and establish the nature of any unsatisfactory condition of CCS's products. CCS, at its request, shall be given access to such records for substantiating warranty claims. No action may be brought for breach of any express or implied warranty after one (1) year from the expiration of this express warranty's applicable warranty period. CCS assumes no liability for any events which may arise from the use of technical information on the application of its products supplied by CCS. CCS makes no warranty whatsoever in respect to accessories or parts not supplied by CCS, or to the extent that any defect is attributable to any part not supplied by CCS.

CCS neither assumes nor authorizes any person other than a duly authorized officer or representative to assume for CCS any other liability or extension or alteration of this warranty in connection with the sale or any shipment of CCS's products. Any such assumption of liability or modification of warranty must be in writing and signed by such duly authorized officer or representative to be enforceable. These warranties apply to the orginal purchaser only, and do not run to successors, assigns, or subsequent purchasers or owners; AS TO ALL PERSONS OR ENTITIES OTHER THAN THE ORIGINAL PURCHASER, CCS MAKES NO WARRANTIES WHATSOEVER, EXPRESS OR IMPLIED OR STATUTORY. The term "original purchaser" as used in this warranty shall be deemed to mean only that person to whom its product is originally sold by CCS.

Unless otherwise agreed, in writing, and except as may be necessary to comply with this warranty, CCS reserves the right to make changes in its products without any obligation to

incorporate such changes in any product manufactured theretofore.

This warranty is limited to the terms stated herein. CCS disclaims all liability for incidental or consequential damages. Some states do not allow limitations on how long an implied warranty lasts and some do not allow the exclusion or limitation of incidental or consequential damages so the above limitations and exclusions may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.