MYARC, Inc.

Hard and Floppy Disk Controller

Users Manual

Fourth Edition

MYARC, Inc.
Martinsville, NJ

NOTICE FROM THE CONVERSION AUTHOR

The following text was digitally scanned from the original document and composed as a new text file, preserving all information as found in the original, except for the correction of obvious errors (language and factual). Locations in the text where factual errors were corrected or where information was added

MYARC HFDC

are indicated with a red bar on the text margin. Language error corrections remain unmarked.

Table of contents

The MYARC Hard and Floppy Disk Controller

The MYARC Hard and Floppy Disk Controller (HFDC) is a powerful and versatile disk controller that will allow you to use up to three hard drives at a time with up to 134 megabytes on each drive. It will also allow you to control up to 4 floppy disk drives, which may be 40 or 80 track drives, single or double sided, and either 5.25 or 3.5 inch size or any combination of these sizes. The floppy drives may be any capacity up to 720 kilobytes (1.44 megabytes with the 36 sector per track option).

Note – Actually you can control up to 8 floppy disk drives by adding a floppy disk controller to the HFDC (see page 9).

What you need

You will need the following equipment with the HFDC controller card.

- GENEVE 9640 computer or TI-99/4A console
- TI Peripheral Expansion Box or equivalent
- MYARC or TI 32K or larger memory expansion card (if used with a TI-99/4A)
- One or more floppy disk drives
- Floppy drive cable(s). If you are going to have both internal and external floppy drives, you will need a special cable with a female 34-pin connector in the middle of the cable and one or more 34-pin edge connectors on each end for the internal and external drives. The female pin connector attaches to the connector on the card located inside the PE Box (marked J1), just as it does for internal drives with a floppy drive controller. The cable must be routed outside the box for external drives. When using the TI Expansion Box, if possible, use card slot 6 or 7 to give room to pass the cable outside.
- Hard disk drive
- Enclosure for the hard drive.
- One 34-pin cable with edge connectors on each end for the hard drive
- One 20-pin cable with an edge connector on one end and a slotted female pin connector on the other.
- Power supply for the hard disk drive. Use the following guidelines if you are providing the power supply separately. Note that these are general guidelines only, and may vary with your particular drive. These figures are for dedicated power supplies only, not power supplies that are powering other devices as well. It is also possible to have too much power for a low-powered drive. Switching power supplies of the type used with IBM PCs may not provide reliable voltages when the power draw is far below capacity. Linear, rather than switching power supplies are the most reliable for hard drive applications.

Type	Min. Amps on 12V supply
5.25" full height	5
5.25" half height	2.5
3.5" half height	1.5

Caution: The peripheral expansion box does not provide adequate power for most hard drives.

Selecting a hard drive

The hard drive or drives you select may be up to 134 megabytes in size and should be ST506/412 compatible. The drives are the same type as those used with IBM PCs, XTs and ATs and their clones.

Run Length Limited (RLL) drives **can not** be used with the MYARC HFDC.

Setting up the HFDC

Before you connect anything, examine your hard drive. Write down any information contained on the drive, in particular, serial number, model number, and any indication of the number of heads, and number of cylinders. If you have a choice, make sure the drive is configured as drive 1. Next, examine the edge connectors to locate pin 1 on each connector. Most drives will have a slot in each edge connector near the low numbered end of the connector, but look for pin numbers to be sure. The pins proceed from pin 1 to pin 19 or 33 (the odd numbers) on one side and pin 2 to pin 20 or 34 (the even numbers) on the other. Pins 1 and 2 are on the same end on opposite faces.

Setting the DIP switches on the HFDC

There are two sets of DIP switches contained on the HFDC printed circuit board. One set tells the HFDC what type of floppy drive is connected and the other tells the hardware/software where to reside on the CRU bus (i.e., the Input/Output bus of the 9900 series microprocessors.)

The switches must be set for your particular configuration. To set the switches, you must remove the clamshell case from the printed circuit board. Examine the bottom edge of the case where the card's edge connector plugs into the PE Box. This edge has four clips on one half of the case that snap into place on the other half and hold the case closed. To disengage these clips, gently insert a screwdriver or fingernail between the two halves at one end and depress the nearest clip on its flat surface. If done correctly, the card will start to separate slightly without forcing. Depress each successive clip in turn, and the two halves will part. Make note of how the two halves fit together on the opposite (top) edge of the card, and remove the case completely. You will now have access to the DIP switches.

MYARC HFDC

The set of switches controlling the CRU bus is located in the center of the HFDC board. There are 4 DIP switches which are either "ON" or not. They are preset at the factory such that the HFDC will take the "logical" place of either the MYARC, TI or Corcomp floppy disk controllers. In this mode, you will have to remove any other disk controller in the PE Box before you power-up the system (if you do not, serious damage could occur). If, however, you want to operate the HFDC in the PE Box along with your standard floppy disk controller, you may do so in one of two possible configurations.

The normal address for the HFDC card is at CRU >1100, and setting switch 1 off and all others to the ON position provides this address. Other addresses are available by changing the DIP switches, as follows:

Switch CRU address

	1000	1100	1200	1300	1400	1500	1600	1700	1800	1900	1A00	1B00	1C00	1D00	1E00	1F00
1	on	off	on	off	on	off	on	off	on	off	on	off	on	off	on	off
2	on	on	off	off	on	on	off	off	on	on	off	off	on	on	off	off
3	on	on	on	on	off	off	off	off	on	on	on	on	off	off	off	off
4	on	on	on	on	on	on	on	on	off	off	off	off	off	off	off	off

If you are going to use a different address for the card, you must not conflict with another card at the same address. The MYARC RAMdisk, for example, occupies address >1000. Other RAMdisks may be adjustable to other locations. If you set the card to CRU >1000, any floppy disk drives it controls must be numbered above 4 or they will prevent access to the drives controlled by the controller at >1100.

The other set of switches control the type of floppy disk drives you have connected to the HFDC. There are 8 switches which control this and they are located on the left side of the printed circuit board.

Switches 1 and 2 are for the first drive, switches 3 and 4 are for the second drive, and so on. Each set of 2 switches sets any one of four possible states to the HFDC. The states are as follows:

high sw (2,4,6,8)	low sw (1,3,5,7)	Sectors per track	Nr of Track	Step Rate	Capacity
0	0	9/16/18	40	16 msec	360K
0	1	9/16/18	40	8 msec	360K
1	0	9/16/18	80	2 msec	720K
		(and for possible future expansion:)			
1	1	36	80	2 msec	1.44M

All switches are preset to 00 at the factory.

Setting up the floppy drives

Each drive must be individually "jumpered" or switched to the proper select number. Many floppy disk drives have drive selects labeled 0 to 3. If so, jumper the **0** location for DSK1, the **1** location for DSK2 and so on.

Caution: Never jumper more than one drive to the same select location. Disk drive damage may occur.

The last setting to make is resistor packs. Each floppy drive contains a resistor pack. However, they should be removed from all floppy drives except the one located furthest from J1 in terms of cable distance. In other words, if you have one floppy drive, leave the resistor pack alone. If you have two drives, remove the resistor pack from the drive closest (in terms of cable distance) to the HFDC, and so on.

Once you have properly connected each floppy drive to the HFDC, jumpered each drive select and removed all resistor packs but one, all connections to the floppy drives are complete.

Note – By adding a floppy controller in addition to the HFDC, you can control up to 8 floppy drives. With two controllers the HFDC controls drives 5 through 8. Set up the physical drives that will be used as drives 5 through 8 as if they are drives 1 through 4, cabling them to the HFDC. The HFDC will address them as drives 5 through 8. **Never connect floppy drives to the 34-pin card-edge connector of the HFDC as damage to the controller can occur.**

Connecting to hard disk drives

Because most hard drives will be located external to the PE Box, all connectors to these drives extend outside the PE Box.

Connecting hard disk drives to the HFDC is similar to connecting floppy drives. You must properly "jumper" each hard drive to the proper number. In other words, WDS1 should be jumpered in the **0** select location, WDS2 should be jumpered in the **1** location and WDS3 should be jumpered in the **2** location.

You must make sure that only the last drive in the chain contains a resistor pack (i.e. the furthest from the 34 pin card edge connector in terms of cable distance).

Connecting the cables

Locate your hard drive in a level, well-ventilated location near the expansion box and within reach of the cables. If you are mounting the drive in a box or on a base, check the pin numbers on the connectors before attaching the drive as explained above.

To connect the drive, make sure the power supply is unplugged from the line current (110 VAC) and plug the four-wire connector from the power supply to the four-pin socket on the hard drive. The connectors will only fit together one way.

Next, attach the 34-pin and 20-pin cables to the card and the drive, as follows. On the card, pin 1 is at the lower end for both cables when the card is upright in expansion box. Plug the 20-pin cable's slotted connector onto the card first. There are three identical 20-pin connectors on the card marked J3 through J5. The connector for drive 1 (J3) is the one closest to the expansion box; the outside connector (J5) is for drive 3. The connector will only fit one way because of the slot. Take careful note which edge of the ribbon cable connects to pin 1 and connect the same edge of the cable to pin 1 of the hard drive.

Next, attach the 34-pin hard drive cable to the edge connector of the card (marked J2). The cable edge connector will fit either way on the card, so check again which edge of the cable is attached to pin 1 and make sure it is consistent on both ends. If you are using multiple hard drives, different 20-pin cables are used for each drive, but the 34-pin cable is shared by putting multiple edge connectors on the cable. (The cable must **not** be shared with floppy drives.)

If not previously done, secure the hard drive to its base or enclosure. Next insert the HFDC in the expansion box. Use card slot 6 or 7 if possible, and make sure that the card is properly seated in the card slot in the PE box.

Finally, attach the floppy drive cable to the card and to the floppy disk drives. Make a final check that all cables are connected properly, and the card is seated properly. Then close the expansion box.

Caution: Do not connect J1 to a hard disk drive, or J2 to a floppy disk drive. Damage to either or both the drive and HFDC will result.

Turning the system on

If you are using your hard drive with the MYARC GENEVE 9640, turn on the hard drives, the expansion box, and the peripherals simultaneously.

To power up the TI-99/4A system, first turn on the expansion box. Next, turn on the hard drive itself. Wait 15 seconds for the drive to come up to full speed before turning on anything else. If you are using more than one hard drive, turn them on one at a time. The drive draws the greatest power while it is coming up to speed. Next turn on any external disk drives, the monitor, and any other external devices. Finally, turn on the 99/4A console.

It is important to keep in mind that when the hard drive has come up to speed, it is enabled for writing at any moment. Therefore, since the logic levels of the HDFC are not at a determined state during power up or power down, you will probably want to power the expansion box and drive from a switchable plug strip.

Turning the system off

To power the system down, first turn off the hard drives, then the PEB, then, if being used on the TI, turn the 99/4A off, and last the peripherals. Or alternatively, turn off the master switch on the power strip.

Setting the time and date

With the 9640, this step is unnecessary as the time and date are automatically read from the 9640's own clock chip.

With the 99/4A the time and date are set in the HFDC clock in either of two ways. One way is to load MDM5, and it will prompt you to enter the time and date, if necessary, when you enter the program. The second way is by entering BASIC or any version of Extended BASIC and entering CALL DT. You are then prompted through the steps necessary to set the clock.

Loading the Myarc Disk Manager V

The MYARC Disk Manager V can be loaded from DSK1 directly from console BASIC without any cartridge by typing the following:

```
CALL MDM
```

You may use the same procedure from Extended BASIC or MYARC Extended BASIC II. Alternatively, you may load from the Editor/Assembler Option 5, TI-Writer Option 3, the FUNNELWEB LOAD option, or any equivalent program image loader. For all of these, type the filename DSK1.MDM5.

The MYARC Disk Manager V also can be loaded from Extended Basic by running the program LOAD, which loads and runs the file MDMXB. If LOAD and MDMXB are placed in drive 1, MDM V will automatically load when Extended BASIC is entered.

MDM5 can be loaded also from Editor/Assembler, Option 3 or any equivalent method using the file MDM.

9640 users can install a loader as a cartridge in the GPL mode. The MDMV file is a cartridge-type file which will load from the GPL menu by setting the F4 to speed 5 and entering DSK1.MDMV.

GRAM KRACKER or other GROM device owners can use this file too. It loads in the GRAM KRACKER at >gE000.

Formatting your hard drive

Your hard drive must be formatted before it can be used, just as floppy disks must be formatted. Formatting erases completely anything already on the disk, so never use this routine if the disk is already formatted and contains material you wish to keep.

To format the hard drive, load MYARC Disk Manager V as explained above. If loading from floppy drive, keep the disk in the drive, because the Format routine is on a separate file that is only loaded when it is needed. From the first menu you will see the following choices:

Catalog	Directory	Find
Quit	Setup	Utility

Select U for *Utility*. This selection will present you with the following choices:

Clone	DSK1	Exit
Format	RAMdisk*	Test

Note: *Only if a MYARC Ramdisk is installed

Select F for *Format*, and you will see three choices:

Exit	Floppy	Hard

Press H for *Hard*, and you are asked for the hard drive number. Enter the appropriate number (usually 1), and you will see the hard disk format screen. The screen asks you for the necessary information to format the drive.

WDS*.

Sectors per track?	32
Heads?	
Cylinders?	
Reserved dir/file sectors?	
Write Pre-compensation?	
Reduced write current?	
Sector verification retry?	8
Interlace (1-Sec/Trk)?	3
Step rate (0-7)?	1

The top left of the screen will display (WDS*.). You may input a hard drive volume name at the blinking cursor. The volume name is limited to a maximum length of 10 characters.

Most drives in the 5 to 20 megabyte range will use 32 sectors per track and 4 heads. The number of cylinders (i.e. tracks) varies with the size of the drive. The following figures are common:

	5 meg	10 meg	20 meg
Sectors per track	32	32	32
Heads	4	4	4
Cylinders	153	306	615

After the "Sectors per track?", "Heads?" and "Cylinders?" prompts are entered, "Reserved dir/file sectors?" will appear. This is the number of sectors reserved at the beginning of the drive for directory and file headers. If all such headers are placed in this section of the drive, the search speed for a particular directory or file is greatly enhanced.

The data for "Write pre-compensation?" is automatically filled in after you enter

the number of cylinders. You can accept that data, or if you know that your drive can handle zero compensation, you can change the data to 0.

The next line is "Reduced write current?" which will show the same data as the preceding line. Again, you can reduce this value to 0 if you know your drive accepts it.

The next line, "Sector verification retry?" allows you to set the number of retries that will be attempted when a sector does not verify correctly. The default is 8. The lower the number, the more bad sectors the format routine will find and, therefore, it will be less likely that a bad sector will be found during normal file handling.

Next is "Interlace (1-Sec/Trk)?" and the default is 3. This value will be correct for most drives. A 28 msec drive may be able to handle an interlace of 2. Older, very slow drives, may require a setting as high as 6 or 7.

The last line is "Step rate (0-7)?" with a default of 1. A higher value may be required for 65 msec or slower drives. 28 msec drives will generally handle a setting of 0. Even two drives of the same make and model can vary in the optimum settings.

Try formatting at several interlace and step rate settings and time how long it takes to format. Use the settings that produce the shortest format time in order to get the best performance from your drive.

When you have entered the right settings, press enter on the last setting and you will be prompted to input the passcode **FORMAT**, press Enter again and MDM5 will look for the format routine in DSK1, if found, MDM5 will proceed to format the hard drive. Depending on the size of the drive, it may take anywhere from a few seconds to a few minutes.

Formatting your floppy disks

Each floppy disk you use must also be formatted before it can be used. If you have been using floppy disks with the TI-99/4A system previously, there is little or no change to what you are familiar with, particularly if you previously used a MYARC controller. The maximum capacity of disks may be more than you have used before with the 99/4A, but this is dependent on your drives as much as on the controller. You can format a floppy disk up to the maximum capacity of the drive or you can format it at a smaller capacity. Even if your drives will handle 18 sector, 80-track, double-sided disks, you will sometimes want to format disks as 9 sector, 40-track, single-sided drives or any standard setting in between. Most commercial software will come on 5.25 inch, single-sided disks, and you will sometimes want to match that format. Even if you are using 80-track drives, you will be able to read the simplest formats. The converse may not be true however. 40-track disks that you format on an 80-track drive may not be readable on a 40-track drive. If you are going to provide diskettes to 40-track-only systems, it is a good idea to keep at least one 40-track drive in your own system.

When you format a floppy disk with the MDM, you select Utility and Format

from the menus just as you did for the hard drive. When you select the Floppy drive option, you will be asked in what format you want to format the floppy disk. You must select the number of sectors per track- 9, 16, or 18 (also 36, optional with the 1.44 megabyte upgrade), the number of tracks (40 or 80), the interlace setting, and the number of sides. You may also enter a diskname (i.e. volume name) at the same time.

Formatting with more than four drives

The format options for drives 1 through 4 are in some cases more limited when you use a floppy controller with the MYARC HFDC. If you are using a MYARC floppy controller with an 80-track EPROM for drives 1 through 4, you will have all the normal options. With a MYARC controller without the 80-track option, you will be able to format disks in 9 or 16 sectors per track. 18 sectors is not supported in drives 1-4. You can, however, format in 9, 16, or 18 sectors per track in drives 5-8. With a Corcomp controller, you can format only in 9 or 18 sectors per track in drives 1-4.

Using your drive

Once you have formatted your hard drive, you will be able to use it just as you would a floppy drive. You can SAVE and LOAD BASIC programs and text files, you can INPUT and PRINT to or otherwise access data files, and you can load assembly programs. As long as you have not set up subdirectories (more on this later), the only difference between accessing the hard drive and accessing floppies is that hard drives are addressed as WDS where floppy drives are addressed as DSK.

For many disk functions you will need to use the MYARC Disk Manager V software furnished with the HFDC. Whereas a BASIC program can be directly SAVEd to your hard disk, protecting and unprotecting files, erasing files, copying files and disks, and other disk utility functions require special software such as MDM5.

Organizing your hard disk: Subdirectories

Although your hard disk can hold thousands of files, it only can hold up to a maximum of 127 files at its top level. Saving and using more files than that requires you to set up subdirectories, each of which can hold up to 127 files. Using subdirectories is something like using separate floppy disks, except that the disks are always loaded, ready to use, and access is very fast.

The top level, also known as the root directory, can hold up to 114 subdirectories in addition to its maximum of 127 files. Each subdirectory can in turn hold another 127 files and 114 subdirectories of its own. You can set up as many levels of subdirectories within subdirectories as you wish, subject only to some practical limits as to how many characters it takes to address them. You could, for example set up a directory at the root (top) level titled WP (for *Word Processing*), put your word processing and related programs in that directory, and add subdirectories for text files. These could be titled, for example, BUS for *business*, PER for *personal*, and DOCS, and the subdirectory names would help you organize the text files you include in each. The directory structures you actually set up will depend on your own particular needs and interests.

Addressing a program or file in a subdirectory is usually accomplished by stringing together WDS1, each directory or subdirectory name in between, and finishing with the filename itself. Each element is separated by a period, for example, WDS1.WP.BUS.MYARCLTR. The string of names up to but not including the filename itself is called the **pathname** of the file. If you want to call a file within a program, you address the file with its complete pathname. The total name can be up to 40 characters long.

An alternate way of using a pathname is to include the volume name of the hard drive instead of its number, just as you can do with floppy drives. WDS.HD1.WP.BUS.MYARCLTR is an acceptable alternate way of addressing the file first named above, if the hard disk is named HD1.

Some programs that were not written with hard disks in mind will only allow 10 characters after the WDS1. You can usually continue to use those programs as long as you keep within the 10 characters. For this reason, it is a good idea to keep your directory and subdirectory names very short. Programs that only let you change the drive number, but not the DSK prefix require some extra efforts. Anticipating the need, MYARC has provided several ways to make a program think it is addressing a floppy drive when it is really accessing a hard drive. The section on **DSK1 and DSK Emulation** (page 27) describes these features.

Directories on floppy disks

The HFDC supports limited subdirectories on floppy disk. With today's high capacity floppy drives, this feature can be very useful. You can create up to three subdirectories on each drive in addition to the root (top level) directory. The four directories in all will support up to 508 (127 · 4) files.

Using the MYARC Disk Manager V

The **MYARC Disk Manager V** functions are reached through single character key presses corresponding to the first letter of the function. The current menu is always displayed on screen. When you load the Manager, you will see the following menu displayed:

Catalog	Directory	Find
Quit	Setup	Utility

Catalog

Catalog lets you see the contents of your floppy disks and hard disk directories, and by using its commands, lets you perform file management functions such as copying, moving, and de1eting files.

You can print the current catalog to printer at any time by pressing the CTRL key and P at the same time. To do this you must have your print device name correctly defined by using the *Setup* function (see page 21.) The print device choice does not appear on the menus.

If you press CTRL key and P at the root directory screen, the full directory structure (called the directory tree), will be printed out to the printer. When you select *Catalog*, you are given the choice of *Floppy* or *Hard disk* or to Exit back to the main menu screen by pressing the appropriate letter. When you select *Floppy* or *Hard*, you in turn are asked for the drive number. When you enter the drive number, the drive is accessed and you see a header with information about the drive displayed at the top of the screen.

Hard disk catalogs

With a freshly initialized 20 meg drive, you will see something like the following:

```
WDS1.
Files     0        Used     68        Free     78650
```

If you have directories set up on hard disk, you will see the volume name followed on subsequent lines by the names of each subdirectory that appears at root (top) level. The number of subdirectories and files within each of those directories, and the dates each directory was created are listed:

Name	Dir/Sub	Files	Created
HD1	3	12	03/15/88
DB	3	8	03/31/88
DSK1	0	47	03/22/88
WP	3	22	04/01/88

The first number is the number of subdirectories in the directory, and the

second is the number of files. The menu at the bottom of the screen looks like this:

Back	Continue	Directory
Exit	Pathname	Root

If there are more than 15 directories, go forward and backward through the pages using Page Up or Page Down on the GENEVE or FCTN-E and FCTN-X or FCTN-4 and FCTN-6 on the 99/4A.

If you press C for Continue at this point, you will next see a display of the files present in the top level. If you press the down arrow key (FCTN-X), the first subdirectory will be highlighted instead. When you press D for Directory (or Enter), the highlighted directory will be added to the pathname at the top of the screen, and the next level of subdirectories for the directory you selected will be displayed. You can proceed down through levels with the arrow keys and D or Enter keys until you reach the subdirectory you want to catalog. Whenever you reach the subdirectory you want cataloged, just press C for Continue and the files will be displayed.

Remember that the D or Enter key displays directories and the C key displays files.

If you go a level too far down, you can *Back* up a level with the B key. R for *Root* takes you back to the top level at any time. *Exit* takes you back to the main screen as does FCTN-5.

If you know the full pathname to the directory you want displayed, you can bypass stepping through the levels by pressing P for Pathname. When you press P, the cursor goes to the pathname at the top of the screen and lets you type in the pathname instead of building it through the menus.

When you are typing the Pathname or any other text entries in MDM5, there are several active FCTN keys to aid you in editing:

FCTN-1	Deletes the character under the cursor.
FCTN-2	Activates insert mode. The next character or characters you type are inserted and characters under and to the right of the cursor are moved to the right
FCTN-3	Deletes the line or field you are typing.
FCTN-5	Escapes to main command menu.
FCTN-8	Replaces what you are typing with what was previously there (also known as an Oops Key).
FCTN-9	Exits the current function and returns to the menu.

Once you select C to display the files in a directory, you will see a screen set up as follows:

WDS1.EA.

| Files 15 | | | Used 10284 | | Free 68434 | |
| Emulate: MYDSK1 | | | | Vol: HD1 | | |
C	Filename	E	Size	Type	Length	P
	ASSM1		88	PROGRAM	8192	P
	ASSM2		20	PROGRAM	4614	P
	BSCSUP		15	DIS/FIX	80	P
	CHARAI		5	PROGRAM	1024	
	DEBUG		14	DIS/FIX	80	P
	EDIT1		25	PROGRAM	5894	P
	MDM5		11	PROGRAM	2228	
	MDM6		34	PROGRAM	8192	
	MDM7		34	PROGRAM	8192	
	MDM8		21	PROGRAM	4676	
	MDM9		26	PROGRAM	5956	
	MYDSK1	1	96	EMULATE		
	SAVE		13	DIS/FIX	80	
	TEXT		17	DIS/VAR	80	

Exit   Root   Utility

The top line of the screen lists the current pathname, identifying the directory that is being cataloged. The second line begins with the number of Files in the current directory. *Used* and *Free* refer to the sectors of the hard disk as a whole. The third line shows the filename for any DSK1 emulation file currently in effect (see **DSK1 and DSK emulation**, page 27) and the hard disk volume name. The files display appears in alphanumeric order.

If there are more than 15 files, go forward and backward through the pages using Page Up and/or Page Down on the GENEVE, or FCTN-E and FCTN-X or FCTN-4 and FCTN-6 on the 4A.

The first column, labeled *C*, is for you to enter Commands when you select the *Utility* function. After the *Filename* column, the *E* column has a special meaning used in DSK1 Emulation (see **DSK1 and DSK emulation**, page 27). *Size* lists the number of 256-byte sectors that the file takes up on the disk. The *Type* and *Length* columns show the specific type of format in which the file has been stored on disk, and either the record size or the length in bytes.

Many programs, both assembly and BASIC, are stored in PROGRAM image form. PROGRAM files show the length of the file in bytes. Text files, data files, and some types of programs are stored in either DISPLAY or INTERNAL forms. The files can be FIXED in length for faster processing or VARIABLE in length for more efficient storage on disk. The file types can be DIS/VAR, DIS/FIX, INT/VAR, or INT/FIX. For FIXED length record files the length of the record is shown after the INT/FIX or DIS/FIX. For VARIABLE length files, the maximum record length follows the INT/VAR or DIS/VAR. Text files are normally DIS/VAR 80.

There is also a special new type of file used: in DSK1 emulation that is

explained in the section on **DSK1 and DSK emulation**, page 27. These files are marked EMULATE. If you catalog a disk with a different disk manager, EMULATE files will show up as PROGRAM files.

The right-hand column is marked *P* for Protected or left blank. Protected files cannot be deleted with the ERASE command in BASIC or by accidental overwriting or Deletion with the MYARC Disk Manager V.

The menu at the bottom of the screen gives you a choice of *Exit*, *Root*, or *Utility*. Root takes you back to the top level directory display. Utility gives you a choice of functions to perform such as copying, protecting, etc. These functions are detailed in the following pages.

Floppy disk catalogs

Floppy disk catalog menus and functions are nearly the same as for hard disk. The catalog display differs somewhat from the hard disk catalog:

```
DSK1.SUBDIR          VOL:              DSDD/18
    Files    24      Used    2530      Free    350
```

The difference in the floppy catalog header is the addition of the disk volume name to the line of the display. The previous **MYARC Disk Manager III** for floppies located the volume name just after the disk number. With the addition of directories and the use of pathnames on floppy disk, the volume name has been separated so it would not be confused as part of the pathname.

The other feature on the top line is the disk format. The first pair of letters represents the number of sides (SS = single sided, DS = double sided). The second pair represents the density (SD = single density, DD = double density). Finally, the number after the slash represents the number of sectors per track – 9, 16, or 18 (also 36 is available with the 1.44 megabyte upgrade).

If there are subdirectories (up to three are allowed) on the floppy, the directory listing appears below the header. The menu is simpler than the hard disk menu; the choices are *Continue*, *Exit*, or the Enter key. The *Back* and *Pathname* commands are not used since there is only one subdirectory level. As with the hard disk catalog, you can select a subdirectory by moving the highlighted bar and pressing Enter or select the root directory by pressing C for Continue.

Utility

When you press U for the *Utility* command, the menu at the bottom of screen changes:

```
    Backup      Copy        Delete      Exit        Move
    Protect     Rename      See         Unprotect   Xecute
```

To use one of the menu functions select the desired file by moving the highlighted area with the arrow keys on the GENEVE, use FCTN or CTRL-E and

FCTN or CTRL-X on the 99/4A.

Backup lets you copy all files on a disk by simultaneously entering a C in the command column of the catalog for every file on a disk or directory. If you want to copy nearly all, but not all files on a disk, you can use the B command and then erase the C from the files you do not want to copy.

Compare the Backup function with the *Clone* command that you can reach from the *Utility* function on the main menu. The *Clone* command produces an exact copy of a floppy disk (but not hard disk). If you want to copy all files from a hard disk directory or to copy all files from a floppy disk but change disk formats, you must use the *Backup* function, not the *Clone* function.

Another time to use *Backup* with floppies is when files have been broken into pieces or fractured. When a disk has had files erased from it, new files may be written into the old spaces. If the new files are larger than the old spaces, the files will end up fractured into non-consecutive pieces. You will not normally be aware that this has happened, but it can cause the disk heads to travel a lot more when accessing a file. If you recopy a disk with the Backup command to produce unfractured files throughout, smoother disk access will result.

Copy will duplicate one or more files from one drive or subdirectory to another. Move the highlighted bar to each of the files you want to copy and type C for each. When you have marked all the files you want to copy and any other functions you wish to perform, press X for Xecute. You will then be given the choice of Hard, Floppy, or Exit. Select the type of drive you want to copy to. You will then be taken through the same process to select the destination drive and directory that you did in selecting the drive to catalog. When you have designated the drive or directory, the copy will proceed. When it is finished, it will return you again to the main menu.

Delete erases a file. Press D for each file you wish to delete and press *Xecute* to carry out the deletion. The file itself is not actually erased, only the pointers to the file in its directory are erased. The file remains on the disk until you write to the disk again. With floppy disks, files can be recovered immediately after deletion in many cases.

Exit leaves the catalog screen and returns you to the main menu.

If you accidentally Exit from the Utility screen by pressing E for *Exit* instead of X for *Xecute*, you may return to the level above the Utility screen by pressing FCTN-8.

Move is similar to *Copy*, except that it deletes the original file after completing the copy function. Type M for all files you want to move. If anything interferes with copying, such as a write-protect tab on the destination disk, the deletion is not performed. This provides you against accidental erasure of the only copy of a file. Despite this protection, it is good practice to not use the *Move* function on an only copy of a file.

Protect prevents a file's accidental erasure or overwrite. To protect a file, select the file with the highlighted bar and press P. The actual protection only

takes place when you subsequently press *Xecute*. Be sure to finish with that step as with other functions.

Rename will change the name of a file. Move the highlighted bar to the file you wish to rename, and press R. A blinking cursor will appear at the beginning of the present filename. Type in the new name. Unwanted characters are erased by pressing FCTN-1. When you are through, press Enter. The actual rename does not take place until you *Xecute*, so be sure to finish with that step as with other functions.

See lets you read a text (DIS/VAR 80) file. When you press S for a highlighted text file, the file will proceed to scroll up the screen until it is finished. You can stop the scroll or start it up again by pressing any key. When the display has finished, press Enter to return to the Edit screen. You can cancel the See operation by pressing FCTN-4.

Unprotect removes a file's protection, as indicated in the catalog display by a P in the right-hand column. Protected files must be *Unprotected* before they can be *Deleted*, updated with a newer version, or *Moved*. A protected file can be *Renamed*.

Volume lets you change the volume name of the floppy or hard disk that you are cataloging. While *Volume* does not appear on the menu it can be invoked by pressing the V key.

1 and **0** to activate or deactivate EMULATE files, (available only on the hard drive) is controlled from the *Catalog* utility screen by moving the highlighted bar to the EMULATE file you want to activate and pressing the 1 (one) key. It will at the same time deactivate any other active file anywhere on the hard disk. The 0 (zero) key will deactivate a highlighted active EMULATE file. An active file is designated by a 1 in the E column. See the section **DSK1 and DSK emulation** (page 27) for more details.

Setup

The Setup function lets you configure your choice of colors, types of floppy drives, and printer defaults. When you select S from the main screen, the *Setup* screen appears. The menu is as follows:

Background color	Change	Exit
Foreground color	Save	

To set your preferred Foreground and Background colors, typing F or B will toggle through all 16 color possibilities for each.

When you select C for Change, the cursor appears at the top of the screen to designate the number of the *Main Drive*. By entering your choices, you will proceed across and down the screen, selecting your choice of *Copy Drive #*, selecting single or multiple drive operation, and telling the program which type of drives you have for up to eight floppy drives. If you are running only one hard drive and/or one floppy drive, answering Y to "Single Drive" prompts will

set the single drive defaults. Thereafter you will not be prompted for drive number when you select Hard or Floppy drive functions. If you should later add additional hard and/or floppy drives, just return to the *Setup* screen and change the option.

The next selection is used to designate your preferred *Output Device*, which will normally be a parallel or serial printer. Refer to your printer manual for the correct information to go on this line. For parallel printers, PIO is all that is usually required, but those using a serial RS-232 interface need more information. If you use "PIO." for your printer setting, make sure you put a period at the end of it. Without it you will get a **Device error** when you try to access it.

If you want to *Save* your settings permanently, press S. It will ask you whether to save the settings to floppy or hard drive. Whichever you select, it will next ask for drive number and proceed to list the header information and any top level directories on the hard or floppy drive you designate.

After you enter the Output Device, the program asks you for the "*Pathname for reloading of MDM5:*". Here you enter the pathname, beginning with the device name, of the location of the MDM files. Normally, you will make these available on hard drive. If you keep them at root directory level, all you need enter here is WDS1.

If you want to *Save* your settings permanently, press S. It will ask you whether to save the settings to floppy or hard drive. Whichever you select, it will next ask for drive number and proceed to list the header information and any top level directories on the hard or floppy drive you designate.

When you *Save* the *Setup* on the 9640 system, the data is saved to both the MDM5 and MDMV files. Make sure that both files are located at the indicated pathname and the loader pathname (i.e. if the MDM programs are loaded at a different pathname on a hard drive than the loaders MDM, MDM5, and MDMV etc., you must have the loaders at, and save the setup to the same location as the programs as well as saving the setup to the location of the loaders only.) MDMV is not required on a TI-99/4A.

At the bottom of the screen appear the familiar choices:

Back	Continue	Directory
Exit	Pathname	Root

As when listing a catalog, you then type in the pathname to MDM5 or build it through the menus. When you have the correct pathname to your MDM5 program shown at the top of the screen, press *Continue* and your Settings will be saved.

Utilities

Pressing U for **Utility** from the Main Menu gives you a different set of choices than pressing *Utility* from a *Catalog* menu.

As shown with the TI 99/4A:

Clone	DSK1	Exit
Format	RAMdisk	Test

As shown with the GENEVE 9640:

Clone	DSK1	Exit
Format	Test	

Clone is for floppy disks only. It lets you produce an exact sector for sector copy of another disk. If you want to make an exact copy of a disk, press C for Clone and MDM5 will prompt you through the process. It will allow you to designate a range of sectors rather than the full disk, if you wish. This choice can be useful if you **know** that only the first part of the disk has been filled, but use it with caution. The last sector used on a disk can easily be higher than the number of sectors used, particularly if any files have been erased previously. As a rule, always pick a high sector at least 34 more than the number of sectors used, even if you know there have been no erasures.

Use extreme care in selecting the *Clone* function. The disk that is copied to will lose all files on it, and they cannot be recovered. Do not use *Clone* as a substitute for files *Backup*, explained above, which will restore fractured files. If a cloned disk has fractured files, its clone will also have fractured files.

DSK1 is used in setting up files used in one type of DSK1 emulation on a hard drive. See **DSK1 and DSK emulation**, page 27.

Format was explained earlier in this manual as the one of the first steps in getting started with a new hard disk system. This command leads to both floppy and hard disk formatting.

RAMdisk is only shown if using the HFDC with the TI-99/4A and allows you to format a MYARC 128K, 256K or 512K RAMdisk card. When you select this command, your system is checked to see if a MYARC RAMdisk is present and if it is initialized. If it is present and initialized, its current partitioning and drive emulation is displayed. Whether or not it is initialized, you can change the partitioning of the card into RAMdisk and print buffer by selecting C for *Change* and typing in the appropriate values (see your RAMdisk manual for specifics). **Caution**: Changing the partitioning erases the contents of the drive. To change the drive emulation, select D for *Disk* and you can toggle through the possible choices. E will *Exit* the RAMdisk function.

Test. When you select *Test* from the *Utility* menu, you will be asked Hard, Floppy, or Exit. When you select Hard, the non-destructive test begins immediately and continues until the entire drive has been checked. **Do not interrupt** the procedure once it has begun or you may lose data on the drive. Any bad sectors found will be "mapped out" of the drive. If a bad sector is found in a file, delete the file, then run the test utility program again.

If you select *Floppy*, you will be prompted for drive number and then asked if you want to log errors to printer and if you want to loop test. If you answer Y to

the loop test prompt, the test will repeat multiple times until you stop it. After you have made your selections, you will then be asked which type of test to perform.

The types of tests are as follows:

> **Non-destructive quick test**. This test is a read only test and will not destroy data on the disk. You are given the chance to log errors to printer, if you wish. Your print device must be correctly identified by the *Setup* process first.

> **Destructive quick test**. This test must be selected with **caution** as it will **destroy all data** on the disk you are testing. The destructive disk test will also ask if you wish to log errors to printer.

> **Comprehensive disk test**. This is the most complex test to select and run. It will require about 20 minutes to complete all five tests executed on a floppy disk of normal size. The time will vary according to disk type and size. Like the quick destructive test, it will **destroy all data on the disk being tested**. Use extreme care in selecting the disk to be tested.

Find

Find is a useful function for hard drive primarily. Use it when you want to know in what directory or directories a particular file is located. When you select F for Find, you are asked to enter the filename to locate. It then asks you whether to look in hard disks or floppy and which drive number. After you enter the information, it proceeds to search the entire drive and list to the screen the complete pathname for each directory in which it is located.

Directory

The directory selection takes you to a set of utilities to use in managing directories and subdirectories on hard disk or floppy.

Create	Delete	Exit
Rename	Utility	

Create lets you set up new subdirectories on both hard and floppy disks. When you press C for Create, you are asked to choose between hard disk and floppy, as usual. You are then prompted for the drive number. You then go through the same process as in selecting a directory to catalog until you reach the level at which you want to create a new subdirectory. When you reach the right level, press C for *Continue*, and you are then prompted for the new directory name.

Delete erases subdirectories. When you select it, you are taken through the directory screens until you select the subdirectory to be deleted by pressing the Enter key and the C (for *Continue*) key. You cannot delete a subdirectory if it contains any files or subdirectories within it.

Rename allows you to rename a subdirectory. Progress through the directory

screens as you do for a *Catalog* until you reach the directory you want to rename. Make sure the subdirectory to be renamed is displayed in the pathname at the top of the screen by pressing D or Enter on the highlighted subdirectory name, and then press C to continue. The cursor will appear at the first letter of the subdirectory in the pathname and you can type in the new name and enter it.

Utility takes you to a set of special functions concerned with backing up and restoring your hard disk contents. The program is external and must load separately from drive #1. If you are normally loading MDM5 from floppy disk, make sure the disk is in the appropriate drive before selecting any of the functions from the Utility menu below.

<div align="center">

Backup   Catalog

Exit   Restore

</div>

Backup

Backup allows you to backup part or all of your hard drive contents. It also lets you choose between backing up all files or only those that have been changed or added since your last *Backup*.

You may choose to backup an individual directory or directories, or you may backup the complete contents of your hard disk without regard to directory structure. If you choose the latter and backup the complete contents, you save the directories as well as the files and therefore, you can later restore the complete contents **and disk organization** exactly as before *Backup*. If instead, you backup individual directories, only files are saved, one directory at a time. For a complete backup in that case, every directory and subdirectory **must** be backed up separately and later, when restoring, every directory and subdirectory must be *Restored* separately.

When you select *Directory* and then *Utility* from the Directory screen, you are then asked to select *Complete*, *Directory*, or *Exit*. As described in the preceding paragraph, "Complete" saves the entire contents of the hard disk and "Directory" saves all files of individual directories only.

If you press C for *Complete* and then press B for *Backup*, you will be asked for active drive number and backup name. Enter drive number and any name up to 10 characters. Press Enter and you are next asked whether you want to Backup all files or only files added or changed since last *Backup*. Next you are asked whether to *Print Catalog* or *Skip Printout*. Whichever you select, you are prompted to enter the drive numbers of all floppy drives that you will use in backing up the hard disk. Enter the numbers of the active drives without spaces between them, for example, "134" designates drives 1, 3, and 4 to be used for backup purposes. You are next prompted to put blank disks in each drive to be used. When you press Enter, the backup proceeds. When the first disks are full, you are prompted to load a new set of blank disks. When the backup is complete, you are prompted to put the first disk back in, and the total number of disks are written to it for future reference.

If you press D for *Directory* and then B for *Backup* you are asked for the hard drive number to backup. After you select the drive number, the top directory screen is displayed. Pressing C for *Continue* at this point will lead to backing up the files at root directory level, however you may select to back up directory or subdirectory files at lower levels by using the arrow keys and D or Enter to narrow the area to be considered. When you finally press *Continue*, all files for the directory pathname displayed will be the files that will be backed up. After C is pressed you are next prompted for an active drive number and backup name. Enter the drive number and any name up to 10 characters. You will then be prompted through the remaining steps in a sequence similar to that detailed in the previous paragraph.

Restore

The **Restore** function lets you replace the complete contents and directory structure to the hard disk, or files to individual directories, whichever you have previously backed up on to diskettes.

To restore the complete contents and directory structure, select *Directory* and then *Utility* from the Directory Screen, then select *Restore* and you will be prompted through the complete restoration.

To restore part or all of the hard disk's previous contents, you can select the level of directory or subdirectory at which to restore the files. You do not necessarily have to restore saved files into the same directory structure as you used previously. You can, if you wish, *Restore* a *Saved* subdirectory to a different directory. Any subdirectory you restore to must exist on the current hard disk. You cannot restore to a non-existent subdirectory. Therefore, if you backed up at a subdirectory level, you will generally want to create that subdirectory again with the *Create* function if necessary, before restoring its contents. Once you have selected the proper directory level to restore, press *Continue* and you will be prompted through the restoration.

Note – For both *Complete* and *Directory* restoring, when you are asked for the date and filename, you must enter them exactly as they were saved to the Backup disks. Catalog them first, if necessary.

Catalog

The Catalog selection from the Directory Utility menu is strictly for cataloging Backup disks. It cannot be used to catalog normal disks nor can the normal catalog function be used to catalog backup disks.

DSK1 and DSK emulation

Many existing programs for the 99/4A were written without hard disks in mind. Programs that look for data or program routines on disk while they are running will often only accept a DSK1 or other DSK# address. Others will only look for a disk with a particular volume name such as MULTIPLAN, which looks for additional routines on a disk with the name DSK.TIMP. MYARC has anticipated the problems that this can cause for hard disk users and has provided solutions to most applications. These take the form of DSK1 and DSK emulation in hard drive directories and a special type of file that is accessed as DSK1.

DSK1 Directory Emulation. The HFDC includes built-in software that causes programs that seek DSK1 to look to the hard drive for a directory with the name DSK1 before it will look to the physical DSK1 drive. To take advantage of this, all you have to do is to create a directory with the name DSK1 at root directory level. You can then put all the files you want (within the 127 file limit) in this directory, and they will automatically be accessed whenever a program looks for DSK1. Just be careful you do not try to put several files with the same name within the directory such as LOAD or UTIL1. If the program you are using reads and writes to DSK1, it will read and write instead to the simulated DSK1. If there is no DSK1 directory or the particular filename cannot be found in the DSK1 directory, the physical drive will be accessed next as usual.

In order to boot the 9640 from hard disk, you must set up files and directories correctly. The file LOAD/SYS must be in the first level directory named DSK1. The file SYSTEM/SYS must be in the Root directory.

DSK Directory Emulation. To handle files in a way as done by MULTIPLAN, which looks for files on a disk with a particular volume name, create a directory at root level named DSK. Then create a subdirectory of DSK with the same name as the necessary volume name. Add all the necessary files to that subdirectory. For example, the MULTIPLAN program files should be in the subdirectory with the pathname DSK.TIMP. The HFDC will automatically direct all file accesses that look for DSK.TIMP.filename to the hard drive. If there is no directory name DSK or the filename cannot be found, the access will next go to the physical floppy drive, as usual.

DSK1 File Emulation. A third method of emulation has been included, which is a special kind of archived file that software will recognize as a physical floppy DSK1. A limitation is that to use this type of file, the disk controller must be at CRU base >1100 (the normal base.) You can have as many files of this type on the hard drive that you want, but only one can be active at a time. The files do not have to be at root directory level, but can be anywhere on the disk. The special file can have a number of files archived and usable within it, such as all the program files from a particular floppy disk.

To create a DSK1 file, prepare a fresh floppy disk with the files that you want included in the emulation. The entire disk will be included in the file normally. Next select *Utility* from the main MDM5 menu and *DSK1* from the *Utility* menu. It will ask you for the hard drive number, and you then set up your pathname

to store the file in as you do any time you save a file. The next prompt asks you what name to give the file. When you have typed in the name and entered it, the program asks you what floppy drive to access to create the new file. Enter the drive number and it will proceed just as if you were cloning a disk. You will be asked what sector range to include, and the same restrictions and cautions apply that pertain to cloning a disk. Make sure that you do not select too low a sector number, because the highest sector used can be higher than the number of sectors used.

Once the file has been created, it will appear in a disk catalog as an **EMULATE** file. More than one EMULATE file can be included in a catalog, but only one can be active at a time in the whole system. From the *Utility* Menu of the *Catalog* screen, you activate an EMULATE file by pressing the 1 (one) key. The 0 (zero) key deactivates a file, or pressing a new 1 key will deactivate whatever was previously selected. The current active file (if any) is designated by a number 1 appearing in the E column.

The program and data files in an active EMULATE file can be accessed just as if they were on physical floppy DSK1.

HFDC disk operations with TI BASIC

TI BASIC can be used with the HFDC. Since the HFDC is designed to be totally compatible with the floppy disk system and it is fully integrated with the TI operating system, all TI BASIC commands are the same in the HFDC hard disk system as they are in the floppy disk system.

MYARC's unique HFDC hard disk system does not require any working space in main memory. In contrast to the floppy disk system which uses 2088 bytes of RAM to accommodate three simultaneously opened files, the HFDC can support up to eleven open hard disk files without using main memory.

Please note that in the floppy disk system, the CALL FILES statement can be used to change the allowable number of "open" floppy files. In the Winchester environment, the allowable number of open files, both hard and floppy, is permanently preset to 11. A CALL FILES(4) statement, for example, will limit the number of open floppy files to 4, but 7 more can still be opened on hard disk to bring the total to 11.

TI BASIC statements

The following sections detail the use of the HFDC with TI BASIC. They cover the "Saving and Loading of Programs", "File Processing", and "Additional TI-BASIC commands supported by MYARC's HFDC".

Saving and loading programs

The HFDC stores and retrieves TI BASIC programs quickly and efficiently through the SAVE (store) and OLD (retrieve) commands. Depending on the length of the program, this can be anywhere from four to ten times faster than the floppy disk system.

The SAVE command is used to store a program to the disk. The SAVE format is:

> SAVE *device.file-pathname*

The device is a Winchester Disk (WDS) and the program file pathname includes the directories and subdirectories under which the program file is to be located.

In order to SAVE the program to the correct file, the system must know the type of device, its number and the file pathname. Alternatively, the system must know the type of device, the Main Directory and the file pathname.

As an example, to store a program called CALCULATE, we will assume the following:

- The program file CALCULATE is in the subdirectory DEMO.
- The volume name of the disk is MYARC.

The two ways to format the SAVE command are:

```
1. SAVE WDS1.DEMO.CALCULATE
2. SAVE WDS.MYARC.DEMO.CALCULATE
```

Note: Periods must separate device, directory and file name.

In the first alternative, the computer searches for the first hard disk (#1), finds the subdirectory DEMO, and saves the program CALCULATE under it. In the second a1ternative, the computer searches for a hard disk with a vo1ume name of MYARC which it knows will be on a Winchester Disk (WDS), finds the appropriate subdirectory, DEMO, and saves the program file under it.

The OLD command is used to retrieve a program from the disk. The OLD format is the same as the SAVE format:

OLD *device.file-pathname*

When the OLD command is entered, the designated program is read into the computer's memory. The program can then be RUN, LISTed or EDITed.

File processing

In addition to the CALL FILES statement mentioned previously, there are seven main TI BASIC statements used to access files on the disk. They are OPEN, CLOSE, INPUT, PRINT, EOF, RESTORE and DELETE. Each are discussed below.

The only TI BASIC statement that can be used to access a directory is OPEN. Directory access is limited to reading the subdirectories and files of the particular directory.

1. OPEN – The OPEN statement prepares a TI BASIC program to use data files stored on the disk. It links a file number used in a program to a file on the disk and describes a file's attributes so the program can create or process the file. If the file a1ready exists, the description given in the program must match the actua1 file characteristics.

   The OPEN statement format is:

   a. **For a directory**:

      OPEN #*filenum*:"*device.directory-pathname.*", INPUT, RELATIVE, INTERNAL

      The device and directory pathname must be included in the OPEN statement for a directory. The open mode is always INPUT since a directory can only be read in TI BASIC.

   b. **For a file**:

      OPEN #*filenum*:"*device.file-pathname*" [,*file-organization*] [,*file-type*] [,*open-mode*] [,*record-type*]

      The file number and device file pathname must be included in the OPEN statement for a file. The other information, enclosed in brackets, may be in any order and are optional entries. If the entries are not made, the computer assigns certain defaults described be1ow.

      • *filenum*. The file number is assigned to a file by the OPEN statement.

Assign a unique number (1-255) to each file. The file number is entered as a number sign (#) followed by an integer from 1 to 255. The only restriction is that the number assigned cannot be the same number as another open file.

- "*device.file-pathname*" – For any file located on a Winchester Disk, the device is always "WDS." See the above explanation in SAVE for the two acceptable "*device.file-pathname*" entries.

- *file-organization*. TI BASIC file organization is the same on the HFDC as it is on the floppy disk system. Files can be organized either sequentially, where its records can be read or written one after the other, or randomly, where they can be read or written in any order including sequentially. A sequential file is indicated by using the SEQUENTIAL option. A random file is designated by using RELATIVE. The user may optionally specify the initial number of records on a file by following the word SEQUENTIAL or RELATIVE with a numeric expression (see "Maximizing System Performance" page 40, for additional discussion.) If no file organization is specified, the default is SEQUENTIAL.

- *file-type*. Files can be stored on the Winchester Disk in two ways. They can be stored as ASCII characters or as machine-readable binary code. If the information will be printed or displayed for people to use, the files should be stored in ASCII format. If the information will only be processed, not displayed or printed, the files should be stored in the compact and more quickly processed binary form.

  To specify a binary file, use INTERNAL. To indicate an ASCII file, either use DISPLAY or leave *file-type* blank. The system defaults to ASCII file types.

  Note: ~~The length of an ASCII type file is limited to 150 bytes.~~

- *open-mode*. There are four open modes in TI BASIC – UPDATE, INPUT, OUTPUT, and APPEND. The system assumes a default value of UPDATE if no mode is specified:

    UPDATE, the file may be both read and written to.

    INPUT, the file may only be read.

    OUTPUT, the file may only be written to.

    APPEND, the file may only be added to.

  A "protected" file may only be read (INPUT). A file can only be APPENDed if it contains VARIABLE length records.

  Note: If an unprotected file already exists on a disk, specifying an *open-mode* of OUTPUT to the same file name writes over the existing file with the new file. This can be prevented by reading all the existing records which moves you to the end of the file. Alternatively, use the RESTORE statement (page 36) with the proper record.

- *record-type*. File records may be all the same length (FIXED) or may vary in length (VARIABLE). Fixed records have a maximum file length of 255 bytes and are processed faster than variable records. Variable records also have a maximum length of 255 bytes.

  File lengths are specified by following FIXED or VARIABLE with a numeric expression. If no record length is specified, the default is 80 bytes.

  Please note:

  RELATIVE files must have FIXED length records.

  SEQUENTIAL files may be either FIXED or VARIABLE. If no record type is specified, the default is VARIABLE for a SEQUENTIAL file.

  A FIXED length file may be reopened for either SEQUENTIAL OR RELATIVE access regardless of previous file organization assignments.

The following are examples of OPEN statements:

```
OPEN #1:"WDS1.DEMO.SALES"
```
Creates or reopens a file on Winchester Drive 1 named SALES. The SALES file is located in the subdirectory DEMO. The file is SEQUENTIAL, kept in DISPLAY format, in UPDATE mode with VARIABLE length records with a maximum length of 80 bytes. (These are all the defaults.)

```
OPEN #10:"WDS.MAIN.DEMO.SALES", RELATIVE 100, FIXED 80
```
Creates or reopens a file named SALES in the subdirectory DEMO. DEMO is in the Main Directory MAIN which is on a Winchester Disk (WDS). The system knows which Winchester Disk number from the Main Directory name.

The SALES file is RELATIVE with 100 records made available for the file. Record size is FIXED at a length of 80 bytes.

2. CLOSE – The CLOSE statement discontinues the association between a file and a program. After the CLOSE statement is performed, the file is not available until it is reopened again using an OPEN statement. Files may also be deleted by adding ":DELETE" to the end of the CLOSE statement.

The CLOSE statement format is:

```
CLOSE #file-number [:DELETE]
```

The *file-number* is the same number used in the OPEN statement. It is important to CLOSE a file otherwise data may be lost. Editing a program automatically closes any open files; but if a program ends due to a BREAK statement, an error, or pressing FCTN-4 (CLEAR), files may not be closed even if there is a CLOSE statement in the program. To prevent data loss in these cases, enter NEW to close the file properly. Alternatively enter BYE to close the file and to leave TI BASIC.

Note: Data may be lost if FCTN-= (QUIT) is pressed instead of entering BYE. When processing files, always use BYE to leave TI BASIC.

3. INPUT – The INPUT statement, used with the HFDC system, allows data to be read from the files. It can only be used with files opened in INPUT or UPDATE mode. The INPUT statement is the only allowable TI BASIC file access statement that can be used for directories. When INPUTing a directory, the user will see an actual listing of the directory contents.

   The INPUT statements have these formats:

   a) **For a directory**:

      INPUT #*file-number*: *variable-list*

   b) **For a file**:

      INPUT #*file-number* [,REC r*ecord-number*]:*variable-list*

   The *file-number* and a *variable-list* must be included in the INPUT statement. The *record-number* may optionally be included when reading random access files.

   • *file-number* – The file number is the number assigned to a file by the OPEN statement. The file number is entered as a number sign (#) followed by an integer from 1 – 255 and is the number of the file that is open.

   • *record-number* – The record number refers to the file record to be read. The record number can only be specified for RELATIVE files since SEQUENTIAL files are read in sequential order.

   • *variable-list* – The *variable-list* is a list of variables into which the data from the file is to be read. It consists of a sequence of numeric variables or string variables, separated by commas.

The following shows examples of INPUT statements:

```
INPUT #5:X$              Puts the next value available in the file
                         opened as #5 into X$.

INPUT #30:X,A,LL$        Puts the next three va1ues available in
```

	the file opened as #30 into X, A and LL$.
`INPUT #15,REC 32:RESULTS`	Puts the first value of record number 32 in the file opened as #15 into RESULTS.
`INPUT #2:A,B,C,`	Puts into A, B, and C the next three values from the file opened as #2. The comma after C creates a pending input condition.

When the next INPUT statement using this file is performed, one of the following actions occurs:

- If the next INPUT statement has no REC clause, the computer uses the data beginning where the previous INPUT statement stopped.

- If the next INPUT statement includes a REC clause, the computer terminates the pending input conditions and reads the specified record.

4. PRINT – The PRINT statement allows data to be written onto files on the disk. It can only be used with files opened in OUTPUT, UPDATE or APPEND mode.

   The PRINT statement has this format:

   PRINT #*file-number* [,REC *record-number*] [:*print-list*]

   - *file-number* – The file number is the number assigned to a file by the OPEN statement. The file number is entered as a number sign (#) followed by an integer from 1 – 255 and is the number of the file that has been opened. The *file-number* must be included in the PRINT statement.

   - *record-number* – The record-number refers to the file record to be written on. The record-numbers can only be specified for RELATIVE files since SEQUENTIAL files are written on in sequential order.

   - *print-list* – The print list is the list of values to be put on the file. It consists of a string of numeric variables or constants separated by commas, colons, and semicolons.

The following shows PRINT statement examples:

`PRINT #5:XS`	Puts the value in X$ into the next position

of the file opened as #5.

`PRINT #10:A,B;"RESULTS"`	Puts the values of A, B, and "RESULTS" into the next record of the file opened as #10.
`PRINT #15,REC 20:"TAX"`	Puts the string constant "TAX" into record number 20 of the file opened as #15.
`PRINT #20: A,B,C,`	Puts the values of A, B, C into the next three positions in the file opened as #20. The comma after C creates a pending print condition.

When the next PRINT statement is performed, one of the following actions occur:

- If the next PRINT statement has no REC clause, then the computer places the data immediately following the previous data.

- If the next PRINT statement has a REC clause, then the computer writes the pending print record onto the file at the position indicated counter and performs the new PRINT REC statement as usual.

5. EOF – End-of-file (EOF) indicates whether there is another record to be read from a file.

   The EOF format is:

   EOF(*file-number*)

   The *file-number* is the same as the number of the OPEN file.

   The EOF function assumes that the next record is going to be read sequentially even if the file is RELATIVE.

   The EOF function will return a value which indicates where you are in the file. If you are at the end of a file, the function returns a value of 1. If you are not at the end of the file, the value is 0. If the diskette is full and you are at the end of the file, the value is -1.

The following examples illustrate how the EOF function can be used:

`PRINT EOF(2)`	Prints a value of 0, 1, or -1 depending on where you are in the file opened as #2.

```
IF EOF(30) <> 0 THEN 870
```
If you are at the end of the file opened as #30, then control goes to line 870.

~~The way to keep track of the last record in a relative file is to create a "dummy" record as the first record in the file. Since the first record contains the number of records in the file, it must be updated each time the file length is changed.~~

6. RESTORE – The RESTORE statement positions the file at a specified file record.

   The RESTORE statement format is:

   > RESTORE #*file-number* [,REC *record-number*]

   The *file-number* must be used with the RESTORE statement while *record-number* is optional.

   - *file-number* – This is the number assigned to the file by the OPEN statement. The file-number is entered as a # followed by the number of the open file.

   - *record-number* – The record-number refers to the record where the file is to be positioned.

The following examples show how the RESTORE statement is used:

```
RESTORE #5
```
The next record read from or written to the file opened as #5 will be the first record on file.

```
RESTORE #14, REC 12
```
The next record read from or written to the file opened as #14 will be the thirteenth record on the file (remember that the first record is number zero.)

   With RELATIVE files, RESTORE is only used to position the file to use the EOF function. This is because the record that will be read or written to can be specified in the INPUT or PRINT statements.

7. DELETE – The DELETE statement erases files.

   The DELETE statement format is:

   > DELETE #*file-number*

   The *file-number* must be included in the DELETE statement. This number is assigned to the file by the OPEN statement.

   Note: Protected files cannot be deleted.

Interfacing with the clock

With TI BASIC, TI Extended BASIC and MYARC Extended BASIC II you may set and read the following clock parameters:

Seconds
Minutes
Hours
Date
Month
Year

Four BASIC statements are used to interface with the clock:

OPEN
PRINT
INPUT
CLOSE

1. OPEN – The OPEN statement format is

    OPEN #X:"TIME", INTERNAL, FIXED
    where X is a unique number from 1 – 255 not assigned to another open file.

2. PRINT – The PRINT statement is used to initialize (set) the clock. The format of the PRINT statement is:

    PRINT #X:SEC$,MIN$,HR$,DAY$,MON$,YR$

    File number is the same number as specified in the OPEN statement.

    Each string is a string of length 2.

    SEC$ = seconds, MIN$ = minutes, HR$ = hours, MON$ = month, and YR$ = year.

    All six parameters must be included in the PRINT statement.

    Hours is based on the 24 hour clock and the c1ock automatically adjusts for lengths of months and leap year.

3. INPUT – The INPUT statement is used to read the clock in a running program. The format of the INPUT statement is:

    INPUT #X:SEC$,MIN$,HR$,DAY$,MON$,YR$

    Definitions of the file number and the six string variables are the same as given in the PRINT statement.

4. CLOSE – The format of the CLOSE statement statement is:

    CLOSE #X

    where X is the same number as assigned in the OPEN statement.

Additional commands supported by HFDC

The HFDC supports four additional commands available from TI BASIC.

- CALL ILR – This command emulates the CALL INIT command available with the Editor Assembler operating system. This routine may be used from the command level, or from a running BASIC program. Care must be exercised when using this command. When executed, all programs and data will be cleared from the expanded memory. Example as executed from BASIC:

    ```
    100 CALL CLEAR
    110 CALL ILR
    120 END
    ```

- CALL LR(*device.file-pathname*) – This command emulates the call command used in the Editor/Assembler System. Example as run from BASIC:

    ```
    100 CALL ILR
    110 CALL LR("DSK1.OBJECTFILE")
    ```

- CALL LLR(*program-name*) – This command is the same as the CALL LINK command used with the Editor/Assembler. Example:

    ```
    100 CALL ILR
    110 CALL LR("DSK1.OBJECTFILE")
    120 CALL LLR("START")
    130 END
    ```

- CALL DIR(*pathname*) – This command allows the user to catalog any directory on any disk drive connected to the HFDC. This command may be executed from command level without overwriting the program in memory. The pathname must specify a legitimate directory on a given drive number.

    Examples of legitimate pathnames are as follows:

    DSK1

    DSK2.DIR1

    WDS1.A.B.C.123

    In order to be compatible with the *MYARC DDCC-1 Floppy Disk Controller*, CALL DIR(X) will catalog the main directory of a floppy disk drive where *X* represents the floppy drive number connected to the HFDC.

Error codes in TI BASIC for the HFDC

The normal TI-99/4A error codes are given in the TI-99/4A User's Reference Guide. TI BASIC error codes for the floppy disk system can be found in TI's *Disk Memory System Reference Book*. The following codes relate specifically to TI BASIC error codes when using the HFDC hard disk system.

Note: All Winchester disk operating systems errors are "hard coded" and

will be indicated in plain text on the screen. See "Hard disk utility error messages", page 44.

TI BASIC errors on the HFDC have the same two-digit format and same meaning with minor variations. The first digit indicates the command or statement involved in the error and the second digit indicates the type of error.

First digit	Command or statement
0	OPEN
1	CLOSE
2	INPUT
3	PRINT
4	RESTORE
5	OLD
6	SAVE
7	DELETE
9	EOF

Second digit	Type of error
0	DEVICE NOT FOUND – The disk drive specified could not be found.
1	WRITE-PROTECTED – The file is write-protected. Unprotect the file via the *Modify File Protection* Utility in the HFDC File Management application.
2	BAD OPEN ATTRIBUTE – One or more OPEN options were illegal or did not match the directory or file attributes.
3	ILLEGAL OPERATION – Should not be generated by the disk system. It could be caused, however, by attempting to perform some illegal file operation, such as requesting INPUT from the Thermal Printer.
4	OUT OF SPACE – There are three conditions that would cause this error: the Winchester Disk is full, the user is attempting to create more than 127 files under one directory, or the user is trying to open more files than the 11 allowed by the system. (**Note**: A file is open when it is being deleted, saved, loaded, etc. If there are 11 open files and delete is attempted on a 12th file this error code will appear.)
5	END OF FILE – There was an attempt to read past the end

of the file.

6   DEVICE ERROR – May occur if:
   - The disk is not initialized. Initialize the disk.
   - The system had its power disconnected during a previous print.
   - The disk is damaged.
   - The disk unit is not functioning properly.

7   FILE ERROR – The indicated file or disk does not exist or a directory in the file pathname does not exist.

Maximizing HFDC system performance

For system developers, here are some suggestions for maximizing HFDC system performance:

File hierarchy implementation

The file system's depth (number of directory levels) and/or breadth (number of directories and/or files at a particular level) have no effect on the speed at which the system stores and retrieves data. The only time file system organization affects speed is when initially opening a file. In this case breadth is, on average, faster than depth.

Create data files to be their approximate final size

If, for storing data, the fastest possible performance is desired from the disk system, all files should be created to the approximate final size. This will generally make the file more contiguous which reduces the amount of time and distance the disk heads need to travel to store data. This will also allow all the disk sectors for data to be allocated to the file when opening the file, not when storing data. This further speeds processing time.

The following example illustrates how to create the file:

```
OPEN #1:"WDS1.DIR.FILE",RELATIVE 1000,FIXED 255,UPDATE
```

Use of the HFDC system with other TI cartridges

The HFDC System should perform satisfactorily with TI cartridges that can accept user-entered pathnames. When using **DSK1 File Emulation** the HFDC will even function properly with cartridges that have hard-wired device names – for example *Terminal Emulator II*.

Troubleshooting procedures

Whenever the Winchester Disk system is completely powered up, three things will happen. First, when the switch on the Winchester Disk Drive is pressed **on**, the disk drive will activate. Later, when the 99/4A console is switched **on**, the red indicator at the left hand front corner of the drive will go off and the MYARC HFDC card light will briefly illuminate. If any of these do not happen, follow the troubleshooting procedures outlined below.

First step for hardware malfunctions

1. Ensure all cables are properly connected and that the Peripheral Expansion System, the HFDC, the MYARC 9640 or the 99/4A console, the Disk Controller Card, the floppy disk drive and the monitor are all properly plugged in and all units are turned **on**. For the Winchester Disk, make sure the power switch is **on** and the electrical power cords are connected to both the disk drive and the power source. The power cord should have a snug fit and be firmly in place in both the unit and the power source. Check that all components are installed according to the directions outlined in "Connecting the Cables". Also determine if the Peripheral Expansion System is plugged in and that the console is connected to it.

2. Power DOWN the entire system by switching off the components in this exact order:

   • first, the 99/4A console.

   • second, the monitor.

   • third, the floppy disk drive(s) and other peripherals.

   • fourth, the Winchester Disk Drive Cabinet.

   • finally, the TI Peripheral Expansion System.

   Note that this is the reverse order of powering **up** the system. When ready to power UP, start with the TI Peripheral Expansion System, then the Winchester Disk Drive, etc.

   The 99/4A console is always the last to be powered up.

3. Power UP the monitor and 99/4A console only to check to see that they are working properly. If this checks out then, power **down** again.

4. Unplug the Peripheral Expansion System. Check that the Peripheral Expansion fuse is not blown by looking at the fuse to see if the wire is broken. If the fuse is blown, follow "Changing the Fuse" instructions in the Peripheral Expansion Manual. After replacing the fuse, plug the system back into the 115 VAC outlet.

5. Determine which problem occurred during your initial power **up** sequence and reference the appropriate corrective action outlined here.

41

Red light on the Winchester disk drive failed to go on

1. Make sure the entire system is **off**. Follow the power **down** sequence outlined above.

2. Re-position the HFDC Card by following the HFDC Card insertion instructions outlined in "Connecting the Cables".

3. Check that all connections are properly made, this includes the Peripheral Expansion System to the console and to the Winchester Disk.

4. Reference the instructions outlined in Section 1 and **repeat** the procedure to connect the HFDC Card to the Winchester and to connect the Winchester to 115 VAC power.

5. Power **up** the system again. If the red light still fails to illuminate, refer to the section on Maintenance, Warranty and Service Information.

Light on HFDC card failed to go on when the console was turned on

1. Make sure the entire system is **off**. Follow the power **down** sequence outlined above.

2. Remember, the HFDC Card light goes on and off very quickly when the console is turned on so it is easy to miss.

3. Re-position the HFDC Card according to the instructions outlined in "Connecting the Cables" and power **up** the system.

4. If you still do not see the light flash on, then you need to check if another accessory card is causing a general system problem. Remove all accessory cards from the Peripheral Expansion System and, one at a time, insert a card, power **up** the system, and determine if that card is functional. It it is, power **down** and try another accessory card. If it still seems that the problem is with MYARC's HFDC Card, refer to the section "Service Information", page 70.

Winchester disk and fan failed to activate

1. Make sure the entire system is **off**. Follow the power **down** sequence outlined above.

2. Check for a blown fuse in your building circuits or in the Winchester Disk Drive Cabinet. **Unplug the Winchester disk drive from the 115 VAC outlet.**

Red light on HFDC card stays on permanently and the console display does not appear

1. Turn the entire system **off**. Follow the power **down** sequence outlined above.

2. Re-position the HFDC Card by following the HFDC Card insertion

instructions outlined in "Connecting the Cables", wait two minutes before re-powering.

3. If this same problem recurs, refer to the section on *Service Information*, page 70.

Disk does not initialize properly

1. Follow the instructions for hardware malfunctions outlined above in "First Step for Hardware Malfunctions." Power **up** the system again and try to initialize the disk by following the instructions in "Formatting your hard drive". Remember to use the utility diskette labeled "MYARC DISK MANAGER V".

2. If the disk fails to initialize, perform the disk diagnostic for a "Destructive Test". The procedure for this is detailed under DESTRUCTIVE QUICK TEST, page 24. The test will identify which hardware malfunction is causing the problem. After performing the destructive test, call MYARC for directions on how to proceed.

EPROM version number

When using a 99/4A and first loading the MYARC Disk Manager V, you will see the MYARC logo, the version number of MDM V, and the prompts for setting the date and time. If you are loading on a 9640 or after you have the date and time you will advance to the main menu. If you press V at this point (i.e., main menu), the version number of the EPROM on the HFDC will be displayed.

If you need contact MYARC about a problem with the HFDC please have the EPROM version number available. We will need the version number to analyze your problem and will ask you for it.

Hard disk utility – Error messages

Below is a list of error messages that may appear while using MYARC's *Disk Management System Utilities*. These messages relate to hard disk operations only.

	Message	Cause	Corrective action
1.	INVALID / DISK / DIRECTORY / FILE NAME	Periods or blank spaces embedded within a name	Delete periods and/or blank spaces in disk, directory, or file name.
2.	PATHNAME ERROR	A directory included in the pathname does not exist	Find out what directories exist by invoking CATALOG and working down the directory structure or press P and type in pathname if known.
		Pathname contains embedded blank spaces, periods, typing errors, or ends with a period	Delete blank spaces or periods in disk, directory, or file names. Correct typing error where appropriate.
		Pathname exceeds 40 characters or a Directory or file name was entered containing more than 10 characters	Rename directories and files so the pathname does not exceed 40 characters. Check pathname for typing errors.
3.	FILE / DIRECTORY DOES NOT EXIST	Last name in pathname does not exist but all previous directories do	Find out what files exist by invoking CATALOG or CREATE a new directory or OPEN a new file.
4.	DISK ERROR	An invalid disk number or disk name was accessed	Obtain an up-to-date listing of disk contents by invoking CATALOG Utility.
		The disk was not properly initialized	Re-initialize the disk
		The disk may have developed bad media	Perform DISK DIAGNOSTICS to determine if this is the case.
5.	FILE PROTECTED	*Delete file* was attempted on a protected file	Unprotect file.
6.	(DIRECTORY) OR (FILE) ALREADY EXISTS	An attempt was made to create or rename a directory or file to a name	Create or rename file or directory with a different name.

MYARC HFDC

that already exists

Assembly programming support

This document is intended to give a programmer the information necessary to perform all data storage and retrieval techniques on both Winchester and floppy disk drives connected to the Hard and Floppy Disk Controller (hereafter referred to as the HFDC). The document is divided into three distinct sections. The first section details the hardware of the HFDC, the second contains the software interface to the HFDC and the last section describes the internal data structures of both the floppy and hard disk formats.

Hardware description

Pinouts

There are six connectors located on the HFDC card. They allow for the control and proper functioning of up to three hard disk drives, four floppy drives, and one streaming tape drive.

Connector J1, a 34-pin header connector, is used to connect the HFDC to 1 to 4 floppy disk drives. The pinout of J1 is as follows:

2 -	(not used)	20 -	Step
4 -	(not used)	22 -	Write data
6 -	Drive select 4	24 -	Write gate
8 -	Index	26 -	Track 0
10 -	Drive select 1	28 -	Write protected
12 -	Drive select 2	30 -	Read data
14 -	Drive select 3	32 -	Side select
16 -	Motor on	34 -	(not used)
18 -	Direction select		

All odd-numbered pins are connected to ground.

Connectors J2 to J5 are used to connect to 1 to 3 hard disk drives. Connector J2, a 34-pin card edge connector, is the drive control connector and must be connected to each hard disk drive. The maximum length of these cables should be less than 10 feet. The pinout of J2 is as follows:

2 -	Reduced write current	20 -	Index
4 -	Head select 2	22 -	Ready
6 -	Write gate	24 -	Step
8 -	Seek complete	26 -	Drive select 1
10 -	Track 0	28 -	Drive select 2
12 -	Write fault	30 -	Drive select 3

14 - Head select 0	32 - (not used)
16 - (not used)	34 - Direction select
18 - Head select 1	

All odd numbered pins are connected to ground.

Connectors J3 to J5, 20-pin header connectors, are used to send data back and forth to the hard disk drives. J3 connects to drive 1, J4 connects to drive 2 and J5 connects to drive 3. The maximum length of these cables should be less than 10 feet. The pinout is as follows:

Pins 1 to 10 are not used

11 - Ground	12 - Ground
13 - MFM+ write data	14 - MFM- write data
15 - Ground	16 - Ground
17 - MFM+ read data	18 - MFM- read data
19 - Ground	20 - Ground

Socket U22 is provided to interface with a digital streaming tape with the industry standard QIC-02 interface. The maximum cable length should be no more than 10 feet. The pinout of U22 is as follows:

1 - (not used)	2 - Acknowledge
3 - Ready	4 - Exception
5 - Direction	6 - Online
7 - Request	8 - Reset
9 - Transfer	10 - Ground
11 - Ground	12 - Data bus 0
13 - Data bus 1	14 - Data bus 2
15 - Data bus 3	16 - Data bus 4
17 - Data bus 5	18 - Data bus 6
19 - Data bus 7	20 - +5 volts

CRU definition

The HFDC can reside at any of the 16 possible CRU addresses allowable in the

TI-99/4A, i.e. >1000 to >1F00. It is set at the factory to >1100, but can be changed by setting the 4-switch DIP on the HFDC card. Switch 4 is the most significant address select and switch 1 is the least in that order. When the switch is **on**, the address select is a **0**. If the switch is **off**, the select is **1**. The following CRU output definition applies to any of the 16 possible CRU selections:

CRU relative address	Definition	Active
00	Activate card and map ROM into >4000	high
02	Reset controller chip	low
04	Floppy motor on / Clock divider bit 0 (CD0)	
06	ROM page select bit 1 (MSB) / CD1	
08	ROM page select bit 0 (LSB) / CRU select	
12-1A	RAM page select at >5400 (LSB at 12)	
1C-24	RAM page select at >5800 (LSB at 1C)	
26-2E	RAM page select at >5C00 (LSB at 26)	

The clock divider bits CD0, CD1 are used for configuring the data separator for different densities.

The following is the CRU input definition:

CRU select=0	Definition	Active
00	Disk controller interrupt	high
02	DMA in progress	high
04	Motor status	low
06	Wait for HD1 ready	high
08	Tape	
0A	Tape	
0C	Tape	
0E	Tape	

CRU select=1	8 Switch DIP
00	Switch 4 status
02	Switch 3 status
04	Switch 2 status
06	Switch 1 status
08	Switch 8 status

0A	Switch 7 status
0C	Switch 6 status
0E	Switch 5 status

Memory map definition

Memory map as seen by the 99/4A peripheral expansion box:

Address	Definition
4000 – 4FBF	One of four possible ROM pages
4FC0 – 4FCF	Tape select
4FD0 – 4FDF	Disk controller select
4FE0 – 4FFF	Time chip select
5000 – 53FF	Static RAM fixed page >08
5400 – 57FF	Static RAM, any of 32 pages
5800 – 5BFF	Static RAM, any of 32 pages
5C00 – 5FFF	Static RAM, any of 32 pages

Controller chips

The HFDC uses the SMC 9234, 9226, 9223, 9216 as the 4-chip set that controls the reading and writing of data to and from hard and floppy disks. The chip set can support hard disk data rates of 5 megabits per second, and floppy disk data rates of up to 500 kilobits per second. Reset to the 9234 controller chip is both under software and hardware control.

The QIC-02 tape interface is contained mainly in the PAL16R4 on the HFDC board. Reset to the tape drive is provided by the main system bus reset line.

The real time clock is a national semiconductor MM58274 and is used for providing time and date both to external software and for time stamping of the files.

In addition, the HFDC contains a special integrated circuit designed and developed by MYARC. This chip allows for the separation of data buses on the card such that programs can run from the DSR ROM while the controller is reading data into the static RAM on the HFDC. Up to 32K of static RAM is supported by the HFDC. In addition, this chip allows for DMA transfers of data to and from RAM. This means that data can be read from or written to memory while the processor is performing other functions.

For additional information on these chips, consult the following manuals:

- Standard Microsystems Corporation Data Catalog, 1986

- National Semiconductor Corp. Logic Databook Volume I, 1984

MYARC HFDC

- Teac MT-2ST/45D Streaming Cassette Magnetic Tape Unit Specs.

Software interface specifications

The information contained in these sections give a complete specification of the interface between the HFDC and 99/4A standard input/output. Since the routines for the floppy and hard disk interface are compatible, the routine for the floppy and the hard disk are contained in the same section, with the extension for the hard disk noted.

In the following sections describing the opcodes, all opcodes beginning with 1 are to floppy disk while all opcodes beginning with 2 are to hard disk. There is, however, an exception, that being when the hard drive is emulating floppy drive 1, an opcode going to floppy drive 1 may in fact go to the hard drive. Consult earlier sections in the manual to get a better understanding of the emulation types available.

Sector Read/Write – Subprograms >10 AND >20

The input parameters for this routine are:

>834A	Returned sector number (>10 only)	
>834C	Unit number	Read / Write
>834E	Buffer start address	
>8350	Sector number	
>8352	MSB sector number (>20 only)	

Sector number – is the number of the sector to be written or read. Sectors are addressed as logical sectors starting with zero. In the case of sector I/O to the floppy, only the word at >8350 is used to define the desired sector. Also in the case of the floppy, the sector desired is returned in memory location >834A. It is not returned in the case of I/O to the Winchester. In the case of the hard drive, the word at >8352 must also be defined as the most significant word of a 32 bit sector definition.

Unit number – indicates the drive number for which the operation has to be performed. Also, if the most significant bit is set, the buffer is assumed to be in CPU RAM, while if the MSB is zero, the buffer is assumed to be in VDP RAM. Drive numbers start with 1.

Read / Write – indicates the direction of data flow.

```
     0  =Write
not 0  =Read
```

Buffer start address – points to the start of the buffer for data transfer. The number of bytes will always be 256.

Error codes are returned in CPU location >8350.

Note: Standard DSRs return errors in the status byte and DSRLNK does not

MYARC HFDC

take >8350 into account for subprograms (DATA >000A).

Floppy Disk Formatting – Subprogram >11

The input parameters for this routine are as follows:

>834A	Returned number of sectors per disk	
>834C	Unit number	Number of tracks
>834E	Buffer start address	
>8350	Density	Number of sides

Returned number of sectors per disk – is the number of sectors just formatted by the format routine

Unit number – indicates the drive number for which the operation is to be performed. Drive numbers start at 1.

Number of tracks – indicates the number of tracks to be formatted. This entry must either be 40 or 80.

Buffer start address – This is included for completeness in documentation, but is not used by the HFDC.

Density – This byte conveys two pieces of information. The leftmost 6 bits contain (right justified) the interleave information used by the format routine. The interleave can be any number from 1 to the number of sectors per track minus one. If this field is left blank, the default of 4 for single density and 5 for double density will be used. The right most two bits indicate the density. They have the following meaning:

Value	Meaning
0, 1	Single density FM, 125 Kbits/sec
2	Double density MFM, 250 Kbits/sec
3	Double density MFM, 500 Kbits/sec

Number of sides – this byte indicates the number of sides the floppy is to be formatted. If the value is 0 or 1, the floppy is formatted as single sided. If the value is 2, the floppy is formatted as double sided.

Error codes are returned in >8350.

Modify File Protection – Subprograms >12 and >22

The input parameters for this routine are as follows:

>834C	Unit number	Protect code
>834E	Pointer to file name	

Unit number – indicates the drive number for which the operation has to be performed. also, if the most significant bit is set, the file name is assumed to be in CPU RAM, while if the MSB is zero, the file name is assumed to be in VDP RAM. Drive numbers start with 1.

Protect code – The protect bit for the given file will be set or reset according to the information given in >834D. If this byte is 0, then the file protect bit is set and all types of file operations are allowable. If it is not 0, then the bit is set and the file operations of SAVE, OPEN for OUTPUT, APPEND, and UPDATE are disallowed.

Pointer to file name – points to where a 10 character file name exists in either VDP or CPU RAM. The file name must be left justified and padded with blanks to the right. No blanks or decimal points can appear in the file name.

Error codes are returned in >8350

File Rename Routine – Subprograms >13 AND >23

The input parameters for this routine are as follows:

>834C	Unit number	Unused
>834E	Pointer to **new** file name	
>8350	Pointer to **old** file name	

Unit number – indicates the drive number for which the operation has to be performed. Also, if the most significant bit is set, the file names are assumed to be in CPU RAM, while if the MSB is zero, the file names are assumed to be in VDP RAM. Drive numbers start with 1.

Pointer to file names – points to where a 10 character file name exists in either VDP or CPU RAM. The file name must be left justified and padded with blanks to the right. No blanks or decimal points can appear in the file name. If the old file name exists and the new one does not, then it will be renamed to the new file name.

Error codes are returned in location >8350.

Access Direct Input File – Subprograms >14 AND >24

The input parameters for this routine are as follows:

>834C	Unit number	Access code

>834E	Pointer to file name	
>8350	Additional info	Unused

Unit number – indicates the drive number for which the operation has to be performed. Also, if the most significant bit is set, the file name and the data buffer are assumed to be in CPU RAM, while if the MSB is zero, the file name and data buffer are assumed to be in VDP RAM. Drive numbers start with 1.

Pointer to file name – points to where a 10 character file name exists in either VDP or CPU RAM. The file name must be left justified and padded with blanks to the right. No blanks or decimal points can appear in the file name.

Access code – Since this subprogram combines multiple functions, an access code is used to indicate which function is to be performed. The following codes are used:

0: Transfer file parameters. This will transfer level 2 parameters to the additional information area (six or 10 bytes). It also passes the number of 256 byte sectors allocated for the file.

N: With N not equal to 0, indicates the number of sectors to be read from the given file, starting at the sector number indicated in the additional info block. After the read has been completed, this entry contains the actual number of sectors read. If all sectors have been read, this entry will be 0.

Additional information – points to a 10 or 14 byte location in CPU RAM location >83xx for direct disk access:

>83XX	Buffer start address	
>83XX+2	Number of first sector	
>83XX+4	Status flags	Number of records / sector
>83XX+6	EOF offset	Logical record size
>83XX+8	Number of level 3 records allocated	
>83XX+1	MSB number of first sector	MSB level 3 records
>83XX+1	Extended record length	

Buffer start address – points to the start of the buffer for data transfer. The buffer has to be able to store at least n · 256 bytes in which n is the access code.

Number of first sector – indicates the sector number at which the read should begin. If the access code = 0 (parameter passing), the total number of sectors allocated for the file will be returned.

The next six bytes are explained in the **Internal Data Structures** part of the

manual.

MSB # of first sector and MSB level 3 records – these are used in the case of the Winchester since the number of sectors can be larger than 65536.

Extended record length – included for completeness and compatibility with the GENEVE 9640. If the record length is greater than 255 bytes, the logical record length will return as 0 and the extended record length will be used to pass the actual record length. This is true for both hard disk and floppy.

Error codes are returned in >8350.

Access Direct Output File – Subprograms >15 AND >25

The input parameters for this routine are as follows:

>834C	Unit number	Access code
>834E	Pointer to file name	
>8350	Additional info	Unused

Unit number – indicates the drive number for which the operation has to be performed. Also, if the most significant bit is set, the file name and the data buffer are assumed to be in CPU RAM, while if the MSB is zero, the file name and data buffer are assumed to be in VDP RAM. Drive numbers start with 1.

Pointer to file name – points to where a 10 character file name exists in either VDP or CPU RAM. The file name must be left justified and padded with blanks to the right. No blanks or decimal points can appear in the file name.

Access code – Since this subprogram combines multiple functions, an access code is used to indicate which function is to be performed. The following codes are used:

> **0**: Create file and copy level 3 parameters from the additional information area.

> **N**: With n not equal to 0, indicates the number of sectors to be written to the given file, starting at the sector number indicated in the additional info block.

Buffer Allocation Routine – Subprogram >16

The argument for this subprogram is the number of file buffers to be allocated. This argument is given in >834C.

The effect of this routine is that an attempt is made to allocate enough VDP space for disk usage to facilitate the simultaneous opening of the given number of files. This number has to be between 1 and 9.

Error codes will be returned in >8350.

Set Current Pathname – Subprogram >27

The input parameters for this routine are as follows:

>834C	Unit number	Unused
>834E	Pointer to file name	
>8350	Error code	Unused

Unit number – indicates the drive number for which the operation has to be performed. Also, if the most significant bit is set, the path name is assumed to be in CPU RAM, while if the MSB is zero, the path name is assumed to be in VDP RAM. Drive numbers start with 1. A1so, the drive number is checked against the drive number in the path name for consistency.

Pointer to path name – points to the length of a path name followed immediately by the path name. The path name must start with WDSx, where the x can be either 1, 2 or 3 for that respective Winchester drive. The path name must be that of a directory, it must end with a period, and it must be less than or equal to 39 characters in length.

The HFDC supports a different current path name setting for each hard drive. On power-up, the default setting is to the main directory. All the other >20 opcodes work on the current directory.

Create Subdirectory – Subprogram >28

The input parameters for this routine are as follows:

>834C	Unit number	Unused
>834E	Pointer to directory name	
>8350	Error code	Unused

Unit number – indicates the drive number for which the operation has to be performed. Also, if the most significant bit is set, the directory name is assumed to be in CPU RAM, while if the MSB is zero, the directory name is assumed to be in VDP RAM. Drive numbers start with 1.

Pointer to directory name – points to where a 10 character directory name exists in either VDP or CPU RAM. The directory name must be left justified and padded with blanks to the right. No blanks or decimal points can appear in the file name.

Error codes are returned in >8350.

Delete Subdirectory – Subprogram >29

The input parameters for this routine are as follows:

MYARC HFDC

>834C	Unit number	Unused
>834E	Pointer to directory name	
>8350	Error code	Unused

Unit number – indicates the drive number for which the operation has to be performed. Also, if the most significant bit is set, the directory name is assumed to be in CPU RAM, while if the MSB is zero, the directory name is assumed to be in VDP RAM. Drive numbers start with 1.

Pointer to directory name – points to where a 10 character directory name exists in either VDP or CPU RAM. The directory name must be left justified and padded with blanks to the right. No blanks or decimal points can appear in the file name.

Error codes are returned in >8350.

If the directory to be deleted contains any files or subdirectories, the operation will fail.

Subdirectory Rename Routine – Subprograms >13 AND >23

The input parameters for this routine are as follows:

>834C	Unit #	Unused
>834E	Pointer to **new** directory name	
>8350	Pointer to **old** directory name	

Unit number – indicates the drive number for which the operation has to be performed. Also, if the most significant bit is set, the file names are assumed to be in CPU RAM, while if the MSB is zero, the file names are assumed to be in VDP RAM. Drive numbers start with 1.

Pointer to directory names – points to where a 10 character file name exists in either VDP or CPU RAM. The file name must be left justified and padded with blanks to the right. No blanks or decimal points can appear in the file name. If the old directory name exists and the new name does not, then it will be renamed to the new directory name.

Error codes are returned in location >8350.

Upper Level File Management

Besides accessing sectors and files at the low level as described above, one may also access files through the standard Peripheral Access Block (PAB). The following shows the standard PAB configuration:

0	I/O opcode	Flag / Status
2	Data buffer address	
4	Logical record length	Character count
6	Record number	
8	Screen offset	Pathname length
10	Pathname	

The PAB must be located in VDP RAM and when control is passed to the DSR ROM, >8354 must contain the length of the device name, and >8356 must point to the "." immediately following the device. For a more detailed discussion, please refer to the **TI Editor/Assembler manual** and the **TI-99/4A Device Service Routine Specification**.

The HFDC supports two extensions to the standard PAB. The first extension is to support I/O to CPU RAM rather than VDP RAM. In other words, the data buffer points to a CPU RAM location rather than a VDP RAM location. In order to perform CPU I/O, merely set bit 6 (where bit 7 is the MSB and bit 0 the LSB) of the **opcode** byte. Everything else will remain the same. The second extension to the PAB is to support total CPU RAM use of PAB and **data buffer**. The DSR ROM will assume that the PAB is in CPU RAM when the MSB of >8354 is a **1**. The configuration of the PAB in CPU RAM is a subset of the 9640's PAB.

Configuration of the CPU RAM Peripheral Access Block is as follows:

0	I/O opcode	Flags
2	Error code	(not used)
4	Buffer address	
6	Record number	
8	00	Logical record length
10	VDP / CPU flag	(not used)
12	00	Character count
14	Status byte	Pathname length
16	Pathname	

The content of the CPU-only PAB is the same as the VDP-only PAB except for the VDP/CPU flag such that if **0**, I/O is to VDP RAM, and if **not zero**, I/O is to CPU RAM.

Internal disk data structures

This section describes the arrangement of data on both hard and floppy disks as supported by the HFDC card. Great effort was afforded to make the internal format of the hard disk as compatible as possible to the floppy disk, whi1e still maintaining a method to access hundreds of thousands of additional sectors. Additionally, back and forward pointers have been maintained such that files and directory paths can be reconstructed even in the event a link is broken.

Floppy Disk Internal Data Structures

The diskette format used by the 99/4A and 9640 computers are as follows:

Sector 0 contains the Volume Information Block (VIB). This block contains general information about the diskette like:

Number of total sectors and available sectors

Number of sectors per track

Names and locations of 3 subdirectories

Bit Maps (i.e. used/unused sectors)

Protection

Number of tracks per side

Sector 1 contains pointers to file descriptor records (file headers).

Sector 2 and up contains file descriptor records and data blocks. The file descriptor records contain general information about the file such as file name, file data type and location of data on the disk.

Volume Information Block

This block contains general information about the diskette.

Offset	Field
0 2 4 6 8	Disk volume name
10	Total number of sectors
12	Number of sectors per track / "D"
14	"S" / "K"
16	(Reserved) / Number of tracks per side
18	Number of sides / Density
20 22 24 26 28	Directory 1 name
30	Pointer to File Descriptor Index Record of directory 1
32 34 36 38 40	Directory 2 name
42	Pointer to File Descriptor Index Record of directory 2
44 46 48 50 52	Directory 3 name
54	Pointer to File Descriptor Index Record of directory 3
56 58 ... 252 254	Allocation bit map

Disk volume name – contains the volume name of the diskette. The volume

name can be any combination of up to 10 ASCII characters, except for the space or period characters and the null character. The name is space filled to the right in the case of less than 10 characters.

Total number of sectors – gives the total number of sectors on the diskette.

Number of sectors per track – indicates the number of sectors per track.

DSK – used by some disk managers to determine if the disk has been initialized.

Reserved – used for protection.

Number of tracks per side – gives the number of tracks on a side of a floppy diskette.

Number of sides – indicates the number of sides the floppy diskette has been formatted.

Density – indicates the density of the floppy. Single density is denoted by 0 or 1. Double density is denoted by 2. Quad density is denoted by a 3.

Directory names – These are the three possible directory names for the floppy diskette. The format of these names are the same as the volume name.

Pointers to File Descriptor Index Records – These point to file descriptor index records for the associated directory. If the pointer is 0, the directory does not exist.

Allocation bit map – this map is used to allocate or deallocate sectors on the floppy diskette. Each bit represents one sector except in the case of a quad density floppy where a bit represents two sectors.

File Descriptor Index Record

The File Descriptor Index Record contains up to 127 one-word pointers. Each non-zero pointer points to a File Descriptor Record. These pointers are **alphabetically sorted** according to the filename in the associated File Descriptor Record. The pointer list starts at the beginning of this block, and ends with a zero entry.

0	Sector number of FDR of first file
2	Sector number of FDR of second file
4	Sector number of FDR of third file
…	…
252	Sector number of FDR of 127th file
254	0

The file descriptor index for the main directory is located at sector one of the

floppy disk, while the index for the subdirectories are pointed to by their respective pointers.

File Descriptor Records

The file descriptor record (FDR), as mentioned previously, contains general information about the associated file. All the information the system needs to know to access and update the file has to be contained within the FDR.

The physical layout of a FDR is as follows:

Offset		
0		
2		
4	File name	
6		
8		
10	Extended record length	
12	File status flags	Number of records per sector
14	Number of sectors currently allocated	
16	End-of-file offset	Logical record length
18	Number of level-3 records allocated (little-endian)	
20	Date and time of creation	
22		
24	Date and time of last update	
26		
28		
30		
...	Data chain pointer blocks	
252		
254		

File name – contains the file name of the file. The file name can be any combination of up to 10 ASCII characters, except for the space or period characters and the null character. The name is space filled to the right in the case of less than 10 characters.

Extended record length – contains the record length if it is a data file and has a record length of greater than 255 bytes.

File status flags – the flags are as follows with bit 0 being the least significant bit:

7	6	5	4	3	2	1	0

Record type	-	(Emulate)	(Backup)	Protect	-	Bin./ASCII	Prg/data

      0   Program/data file indicator

          0   Data file

          1   Program file

      1   Binary/ASCII data

          0   ASCII data (DISPLAY file)

          1   Binary data (INTERNAL or program file)

      2   Reserved for future data type expansion

      3   PROTECT flag

          0   Not Protected

          1   Protected

      4   Backup flag (only used for HD)

      5   Emulate flag (only used for HD)

      6   Reserved for future expansion

      7   FIXED/VARIABLE flag

          0   Fixed length records

          1   Variable length records

Number of records per sector – contains the number of records that will fit into a sector.

Number of sectors currently allocated – contains the number of sectors allocated to this file.

End-of-file offset – contains the offset into the highest sector used in the case of program and variable length files. When this field is not 0, the byte length of a program file calculates as (n-1)·256 + eof, where n is the number of sectors allocated to this file, and eof is the value of this field. When 0, the length is n · 256.

Logical record length – contains the logical record length of a data file. If this entry is zero and it is a data file, then the record length is given in the extended record length at bytes 10 and 11.

Number of level 3 records allocated – In the case of fixed length record files, this contains the highest record actually written to. In the case of variable length record files it contains the highest sector actually written to. **Note**: The bytes in this field are in **reverse order.**

Date and time of creation – gives the date and time the file was first created on the disk.

Date and time of last update – gives the date and time the file was last written to.

Data chain pointer blocks – Each data chain pointer block is comprised of

three bytes which comprise a cluster. The three bytes are broken down into 2 three-nibble b1ocks. The first indicates the first actual sector in the cluster. The second indicates the highest logical record offset in the cluster of contiguous sectors. The following diagram shows how each three byte cluster relates to the address of the first sector and the highest logical sector offset in the cluster.

Byte 1	N2	N1		Byte 2	M1	N3		Byte 3	M3	M2
First sector					N3	N2	N1			
Number of last sector					M3	M2	M1			
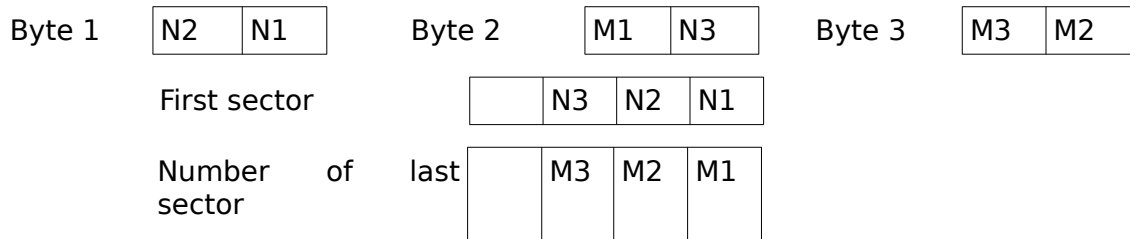
If the number of the last sector is 6 (number counting from 0) this means that the last sector of this cluster contributes the 7th sector of the complete file. Hence, the number of the last sector of the last cluster is also the number of sectors of the whole file plus 1.

Hard Disk Internal Data Structures

The hard disk format used by the 99/4A and 9640 computers are as follows:

> **Sector 0** contains the Volume Information Block (VIB). This block contains general information about the hard disk such as:

>> Volume Name
>> Number of allocation units
>> Number of sectors per track
>> Number of tracks per head
>> Pointers to subdirectories (max. 114)
>> Other physical disk information

> **Sectors 1** to **31** contain the bit map (i.e. used / unused sectors).

> **Sectors 64** and up contain file descriptor records, data sectors, file descriptor index records and directory descriptor records.

Volume Information Block

This block contains general information about the hard disk. The information is configured as below:

0	
2	
4	Disk volume name
6	
8	

10	Total number of allocation units

12	Number of sectors / track	"W"
14	"I"	"N"

16	Hard disk parameters
18 20	Date and time of creation

22	Number of files	Number of subdirectories

24	Pointer to File Descriptor Index Record
26	Pointer to DSK1 emulation file
28 30 … 25 2 25 4	Pointers to subdirectories

Disk volume name – contains the volume name of the hard disk. The volume name can be any combination of up to 10 ASCII characters, except for the space or period characters and the null character. The name is space filled to the right in the case of less than 10 characters.

Total number of allocation units – contains the total number of allocation units on the disk. This number may represent the total number of sectors on the disk, or some power of 2 thereof. Due to the limited size of the bit map (31 sectors @ 256 bytes @ 8 bits), the total number cannot exceed the value >F800, which means 63488 AUs.

Number of sectors per track – indicates the number of sectors per track.

WIN – used by some disk managers to determine if the disk has been initialized.

In place of the letters W, I, N, the following values can be stored:

    W – Number of AUs allocated for file/dir. headers at the beginning of the disk

    I – Step speed parameter used by the hard disk controller

    N – First cylinder where reduced write current is applied

Hard disk parameters – The meaning of this word is as follows:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
(#Sectors / AU) – 1				Num. of Heads – 1				Buf	Write pre-compensation						

AU = Allocation unit; **Buf** = Buffered head stepping (0=no, 1=yes)

Date and time of creation – contains the date and time the disk was initialized.

Number of files – indicates the number of files in this directory.

Number of subdirectories – contains the number of subdirectories of this directory.

Pointer to file descriptor index record – This pointer indicates the AU where the file descriptor index record for this directory is located on the disk.

Pointer to DSK1 Emulation file – points to the AU on the hard disk that is the file whose contents are sector-by-sector a direct copy of a floppy disk. If this entry is 0, then no emulation will occur.

Pointers to subdirectories – Up to 114 AU pointers to directory descriptor records can be supported by the VIB. The total number is given by the number of subdirectories in byte 23. The pointers are arranged in **alphabetic order**.

File Descriptor Index Records

The file descriptor index record contains up to 127 one word pointers. Each non-zero pointer points to a file descriptor record. These pointers are alphabetically sorted according to the filename in the associated file descriptor record. The pointer list starts at the beginning of this block, and the 128th entry always points back to its associated Directory Descriptor Record.

0	AU number of FDR of first file
2	AU number of FDR of second file
4	AU number of FDR of third file
...	...
252	AU number of FDR of 127th file
254	AU number of DDR of this FDIR

The AU number of a File Descriptor Index Record is given in bytes 24 and 25 of its associated Directory Descriptor Record.

Note that for floppy disks, sector 0 plays the role of all directory descriptor records, thus the last word in the FDIR is always 0 for floppy disks.

Directory Descriptor Records

The format of a Directory Descriptor Record (DDR) is the exact same as that of the VIB with the following two exceptions:

The word **WIN** is replaced by **DIR**.

The **Pointer to DSK1 Emulation File** is replaced by a pointer to AU of the **Parent DDR**.

File Descriptor Records

The File Descriptor Record (FDR), as mentioned previously, contains general information about the associated file. All the information the system needs to know to access and update the file has to be contained within the FDR.

The physical layout of an FDR is shown on the following page.

Offset	Field
0 / 2 / 4 / 6 / 8	File name
10	Extended record length
12	File status flags | Number of recs / sector
14	Number of sectors currently allocated
16	End-of-file offset | Logical record length
18	Number of level-3 records allocated (little-endian)
20 / 22	Date and time of creation
24 / 26	Date and time of last update
28	"F" | "I"
30	Pointer to previous FDR in chain
32	Pointer to next FDR in chain
34	Number of AUs allocated for this FDR
36	Pointer to File Descriptor Index Record
38	Extended Information
40 / 42 / ... / 252 / 254	Data chain pointer blocks

File name — contains the file name of the file. The file name can be any combination of up to 10 ASCII characters, except for the space or period characters and the null character. The name is space-filled to the right in the case of less than 10 characters.

Extended record length – contains the record length if it is a data file and

has a record length of greater than 255 bytes.

File status flags – the flags are as follows with bit 0 being the least significant bit:

7	6	5	4	3	2	1	0
Record type	-	Emulate	Backup	Protect	-	Bin./ASCII	Prg/data

    0   Program/data file indicator

        0   Data file

        1   Program file

    1   Binary/ASCII data

        0   ASCII data (DISPLAY file)

        1   Binary data (INTERNAL or program file)

    2   Reserved for future data type expansion

    3   PROTECT flag

        0   Not Protected

        1   Protected

    4   BACKUP f1ag

        0   Has not been modified since last backup

        1   Has been modified since last backup

    5   DSK1 file emulation flag

        0   File is not a DSK1 file emu1ation

        1   File is a DSK1 file emulation

    6   Reserved for future expansion

    7   Record type

        0   FIXED length records

        1   VARIABLE length records

Number of records per sector – contains the number of records that will fit into a sector.

Number of sectors currently allocated – contains the number of sectors allocated to this file. The most significant nibble is located in **Extended Information**.

End-of-file offset – contains the offset into the highest sector used in the case of program and variable length files. See the floppy FDR for more explanations.

Logical record length – contains the logical record length of a data file. If this entry is zero and it is a data file, then the record length is given in the extended record length at bytes 10 and 11.

Number of level-3 records allocated – In the case of FIXED length record files, this contains the highest record actually written to. In the case of VARIABLE length record files it contains the highest sector actually written to.

The most significant nibble of this field is located in the **Extended Information** word. Note: the bytes in this field are in reverse order.

Date and time of creation – gives the date and time the file was first created on the disk.

Date and time of last update – gives the date and time the file was last written to.

FI – the ASCII characters "FI" are located in bytes 28 and 29 of each FDR.

Pointer to previous FDR in chain (AU) – In case that a single FDR cannot hold all cluster entries (for heavily fractured files), a chain of FDRs is created. If this FDR is the first one in the chain, the value is zero. Otherwise, the value points to the AU of its predecessor.

Pointer to next FDR in chain (AU) – points to the AU of the successor FDR, or contains zero if this FDR is the last in the chain or if it is the only FDR of this file.

Number of AUs allocated to this FDR – gives the number of AUs that have been allocated to this FDR. This is useful in determining quickly if this is the desired FDR, or a predecessor or successor is.

Pointer to File Descriptor Index Record – points to the FDIR which points to this file.

Extended information – This word is divided into 4 nibbles. The leftmost nibble is the most significant nibble (MSN) of the number of sectors allocated to the file. The next nibble is the MSN of the number of sectors actually used in a variable record file. The next nibble is the sector number within an AU pointing to the predecessor FDR. The last nibble is the sector within an AU pointing to the successor FDR.

A	B	C	D
Number of allocated sectors / 65536	Number of sectors of a VARIABLE file / 65536	Sector number in AU pointing to previous FDR	Sector number of AU pointing to next FDR

Data chain pointer blocks – This section of the FDR contain data block pointer clusters. Each cluster is a two-word entity. The first word points to the first AU in the data block and the second word points to the last AU in the data block. The number of contiguous AUs allocated by the cluster is the difference plus 1.

Service information

Since the HFDC is a software system operating in electronic combination with complex equipment, malfunctions may be attributable to a variety of causes. For this reason, it is critical that you call MYARC or your dealer to obtain **return authorization** for in-warranty repairs or replacement. We strongly recommend that requests for out-of-warranty repairs are also called in **before** return shipment.

This may ultimately save you time and money by possibly avoiding unnecessary shipping and service costs.

1. Before calling, refer to, and follow all instructions in both the "Troubleshooting" and the "Hardware" and "Error messages", pages 41-44. The listed corrective actions will frequently eliminate your problems.

2. If your troubleshooting does not resolve the problem, confirm with MYARC that the unit requires servicing. If the unit is in-warranty, you must have our verbal or written Return Authorization. MYARC will instruct you on which unit(s) need to be returned. Generally we will request that you send back the entire unit including the Winchester Disk Drive Cabinet, the HFDC Card and the power cord. For a **Return Authorization Number** write or call us at:

   MYARC, INC.
   P.O. Box 140
   Basking Ridge, NJ 07920
   (201) 805 - 0007

   Please note this is not a toll-free number and we cannot accept collect calls.

   Business hours: 9 AM – 5 PM, EST or EDT (as appropriate), Monday to Friday, except for legal holidays.

3. To protect against possible loss:

   • **Insure the unit** for its full value prior to shipping.

   • Pack the HFDC in its **original shipping carton**. Our carton is custom engineered to protect the units from shipping and handling shocks.

4. Enclose inside the carton your name and address, your daytime phone number, and the Return Authorization Number provided to you. Please also write the Return Authorization Number on the outside of the carton on the label. Be sure to enclose a description providing details of the malfunction. Return the unit, shipping and insurance **prepaid** to our manufacturing location as follows:

   MYARC, INC.
   50 Darren Woods Drive
   Martinsville, NJ 08836

Warranty

ONE-YEAR LIMITED WARRANTY – This MYARC HFDC warranty extends to the original consumer purchaser of the MYARC HFDC.

WARRANTY DURATION – The MYARC HFDC is warranted for a period of one (1) year from the date of the original purchase by the consumer.

WARRANTY COVERAGE – This MYARC HFDC is warranted against defective materials or workmanship. This warranty is **void** if the computer has been damaged by **accident**, **unreasonable use**, **neglect**, **improper service** or other causes not arising out of defects in materials or workmanship. This warranty is also void if the **seals** to the hard disk drive are **broken** or otherwise tampered with.

WARRANTY DISCLAIMERS – Any implied warranties arising out of this sale, including but not limited to the implied warranties of merchantability and fitness for a particular purpose, are limited in duration to the above one (1) year period. MYARC shall not be liable for loss of use of the hardware or other incidental or consequential costs, expenses, or damages incurred by the consumer or any other user. Some states do not allow the exclusion or limitation of implied warranties or consequential damages, so the above limitations or exclusions may not apply to you in those states.

LEGAL REMEDIES – This warranty gives you specific legal rights, and you may also have other rights that vary from state to state.

WARRANTY PERFORMANCE – During the first 180 days of the above one-year warranty period, your MYARC HFDC will be repaired or replaced with a new or reconditioned unit of the same or equivalent model (at MYARC's option) when return is authorized by MYARC and the unit is properly packed in the original MYARC shipping carton and returned by prepaid shipment to MYARC at the address shown below. The repaired or replacement unit will be warranted for 180 days from date of repair or replacement. Other than the shipping requirement, no charge will be made for the repair or replacement of in-warranty units during this initial 180-day period. Repaired or replacement units will be returned shipping prepaid within the Continental U.S.

After the first 180 days of the one-year warranty period, MYARC will supply, at no cost, new or rebuilt replacements for any defective parts. After the first 180 days of the one-year period you must pay for all labor and shipping charges.

SHIPPING INSTRUCTIONS: If you believe that your unit requires servicing, please contact MYARC before you return your unit. We will try to analyze and may be able to solve your problem without need of returning the unit. Please obtain a Return Authorization number from us before you ship the card back.

MYARC recommends that you insure the package for value, prior to shipment.

MYARC's address:

MYARC, INC., 50 Darren Woods Drive, Martinsville, NJ 08836