

THE analyzer

FOR ALL ATARI COMPUTER SYSTEMS



THE
SCANALYZER

BY: GEORGE MORRISON
GUY ALBERTELLI
(C) COPYRIGHT 1984

AN ALPHA SYSTEMS PRODUCT

This manual is solely for the use of the purchaser and no part of this manual may be reproduced in any way without prior written consent from Alpha Systems.

Atari is a registered trademark of Atari, Inc.

NOTICE: Portions of this product were produced using copyrighted software products of MONARCH DATA SYSTEMS, Cochituate, MA, 01778.

Alpha Systems
4435 Maplepark Road
Stow, Ohio 44224

THE SCANALYZER

Congratulations! You now possess the most powerful program analyzer available for Atari computers. This package contains a disk full of high-powered utilities designed to analyze all forms of Atari programs.

The Scanalyzer is an integrated software package that allows users to understand and even modify programs that are normally too protected or complex for that. The separate modules that make up Scanalyzer work together to allow you to get maximum use from your software.

The modules are:

1. BASIC LISTER - creates a listable modifiable BASIC program from one that is protected.
2. DIRECTORY FINDER - finds and displays hidden disk directories and allows the copying of files to a disk with a normal directory.
3. CARTRIDGE READER - reads any 4, 8, or 16K cartridge and saves its data to a disk as a binary load file.
4. DISK SCANNER - finds and identifies all the popular forms of disk protection, including forced CRC errors and duplicate sectors. Also allows you to display and modify the contents of the disk, shows detailed directory and

VTOC information, allows tracing of sector links, and several other functions.

5. DATA ANALYZER - converts the 6502 machine language into readable assembler source code that is compatible with the Atari Assembler Editor. It can analyze binary load files (like those created by the Cartridge Reader), boot sectors, selected arbitrary sectors or programs in memory.

6. DISK BACK-UP - a fast action sector copier that gives you two ways to create standard "bad" sectors.

7. INTRO DISPLAY - for those who like fancy introduction screens, but don't like to sit through it every time you use the disk.

GENERAL LOADING INSTRUCTIONS

1. Turn on disk drive.
2. Insert Scanalyzer disk in drive #1.
3. (For XL computers, use the translator disk first).
4. Turn computer on.
5. The Scanalyzer main menu will automatically appear.

IMPORTANT: Always try to operate Scanalyzer on a back-up copy of your software. Modifying the original software could be dangerous if you are not sure what you are doing.

BASIC LISTER

BASIC LISTER will list even those "unlistable" BASIC programs. It unprotects them and restores the variable table if needed. It sends output to the screen, printer, or to another disk file. It can also help you to recover partially destroyed BASIC programs.

Before using BASIC LISTER

Do NOT use BASIC LISTER unless you know the program was written in BASIC. Fortunately, this is very easy to determine.

To tell if a program was written in BASIC:

1. Boot in any of your own disks with DOS on it and the BASIC cartridge in (your Atari Master Disk is fine).
2. When the "READY" prompt appears, type "DOS" and hit RETURN.
3. Read the directory (option A and hit RETURN twice) and write down the names of the files. If they are scrambled, or no directory displays, then try the Directory Finder program of Scanalyzer.
4. Return to BASIC (option B).
5. Now try to LOAD the programs one at a time using the

BASIC LOAD command. Example: "LOAD "D:program"
If you get an error (mainly error 21) it means that the file was not written in BASIC, but if it loads and comes back with "READY", then the program is in BASIC and can be listed with the BASIC LISTER.

NOTE: After loading the program, try typing "LIST". If your machine locks up, or you get some garbage, then you know you have a protected BASIC program that BASIC LISTER will work fine on.

USING BASIC LISTER

1. Select BASIC LISTER from the main menu.
2. Place the disk with the program you want listed in your disk drive (which ever one you choose).
3. Enter the number of the disk drive you wish to use.
4. If you forget the file name, request a directory.
5. Enter the name of the file you want to list. Next the variable table is displayed. Sometimes the variable table is wiped out by the programmer and is unreadable or control characters.
6. If the variables look normal (for example: A, XA, Q\$, FN\$), then enter "Y" when asked if you want to use the variables.

If they look like total garbage or are unlistable, type "N" (when in doubt, always use "N"). If you select "N", the program will create its own variable names for you. The program will name all numeric variables as N1 through

N128, and string variables as S1\$ through S128\$.

Next the program will read through the file to find all string variables so that it can handle them properly on the second pass. It then displays the variable names and allows you to change them if you wish.

When it is done, it will display the following options on the screen:

Send output to:

- A Printer
- B Screen
- C File

For the printer or the screen, just hit the appropriate key and press return.

Sending output to a file

Saving the program to a file is very easy if you have two disk drives (just save the file to drive 2 by specifying D2:program name). If you have only one disk drive, then you must have room for the output file on the same disk that you are reading from. To insure plenty of room, copy the file that you want to decode onto a blank, formatted disk, so there will be enough space to save the output file onto the same disk. Also, be sure to give the output file a different name than the input file.

NOTE: The file will be saved in LIST format so you must use the ENTER command to load it in.

Programs listed to the screen will scroll by very fast. To stop the program, press the CONTROL key and "I" (at the same time). Press them again to re-start. In most protected programs, you will see some garbage at the end, like ERROR in statement pointer, or invalid EOF. This is the part of the program that did the protection, so remove it, it is no longer needed.

PROBLEMS

1. If the program stops working during execution, there are two possibilities:

A) You see an error message displayed

In this case you are given the option to stop or continue. Normally you would stop, since this signals the end of the program. However, in the case of a partially damaged file, you may want to continue.

Continuing: If you tell the BASIC LISTER to continue, it will scan whatever is left of the file. It has to find the next valid BASIC line, and this can take a while if a large part of the file was damaged. If it finds more valid lines, it will continue to list the program (to the screen, printer, or file) by itself. It will stop and display a message if it reaches the end of the file before it finds another valid line.

B) The system locks up.

This happens when there is a problem with the variables. The variable table may have looked complete, but probably wasn't. In this case, reboot the Scanalyzer and answer 'N' when you are asked if you want to use the

programs variables (this will make Scanalyzer create its own variable table).

2. After sending output to a file, if you find that it does not run properly, there are two possible reasons why:

A. The mostly likely reason is that the program is not a "stand-alone" program. Many programs are loaded in by other programs which set up certain routines or try to call other programs or files which may not be on your new disk. Copy all other files from the original onto your copy disk and then boot in the copy as usual.

B. Another possibility is that the program is searching for some protection that was on the original disk. If this is the case, you must either try to find and remove this check so the program will run normally, or recreate the protection on the copied disk.

DIRECTORY FINDER

DIRECTORY FINDER is a utility that will search a disk to find the disk directory. Many times a disk will have its directory hidden to prevent people from copying or modifying the programs. This program will usually find and display the hidden directory and allow you to copy the files off of it, and onto a disk with a normal directory for easy use.

IMPORTANT:

A large number of commercial programs have no directory at all because they use boot sectors. Boot sectors load directly with no need for files or a directory. The Directory Finder can search these disks, but, of course, it will not find a directory, since none is present.

If the disk you are searching contains two directories (one real and one as a decoy), this program is able to find both. If you have a reason to believe there is a hidden directory on your disk, you should run this program even if the disk appears to have a normal directory, because, in some cases, it may also contain a hidden one with additional files.

RUNNING THE PROGRAM

This program will start by asking you which disk drive you wish to use. Of course, if you only have one, type a one and press RETURN. The program will then prompt you to insert your disk and begin its search. If you wish to stop the search at any time, hit the "S" key and it will stop after the next read.

If no hidden directory is present on the disk, a message will be displayed and you will be given the option to run the

program again or return to the main Scanalyzer menu.

If the directory is found, it will be displayed. Next, the following options will appear:

1. Rerun the program
2. Return to the main menu
3. Copy file to a normal disk
4. Display the hidden directory
5. Display the normal directory
6. Continue search for another directory

Option 3 will enable you to move files from the disk with a hidden directory to a normally formatted disk. When selected, it will ask you the name of the file you want to move and then read it (or as much of the file that will fit into your computer's memory) before prompting you to insert your destination disk (the disk you wish to copy the file to). You may have to switch the disks a few times to get the whole file copied.

Before using this feature, be sure you have a normally formatted disk with enough space to hold the file or files you wish to copy.

The files can be read and used normally once they are copied to your other disk.

Option 6 tells the program to continue searching even though one directory has been found. It is a good idea to continue if the only directory found was the real directory (the one stored in sector 361). This program can find any number of hidden directories if you tell it to continue its search after each one.

DISK SCANNER

Disk Scanner allows you to scan the disk and identify and display the protection techniques used on it. Disk Scanner also allows you to display and edit directory, VTOC, and sector data on the disk. There are also functions to allow you to trace the sectors of a file and display all the data contained in duplicate sectors. Further explanation of these terms and disk protection techniques is provided below.

DISK PROTECTION

One of the functions of Disk Scanner is its unique ability to detect the protection techniques being used on a disk. Several forms of disk protection are dealt with in other parts of Scanalyzer (i.e., unlistable BASIC programs and hidden disk directories). This section deals with a form of disk protection known as custom formatting.

You probably know that Atari disks are formatted in a standard way. The standard is 40 tracks with 18 sectors and 128 bytes per track. This standard layout is automatically created by the microprocessor in your disk drive each time you format a disk. You cannot change this layout from the computer using a normal disk drive because the format is totally controlled by the disk drive itself.

Most companies now use a variation on the standard format to protect their disks from being copied. Here is how it works:

The original disks are formatted in a special way that can be detected by a standard disk drive. They could be missing sector 20, for example. Then program can then check to be sure that sector 20 is missing from the disk before it runs the rest of the program. Basically, it tries to read sector 20 and if it gets an error, then it runs normally. If it finds a sector 20 on the disk, then the program knows it is on a normally formatted copy and the program could stop running, or even

try to format the copied disk.

Almost all disk protection is a variation on the techniques mentioned above. Below is a list of the protection techniques and an explanation of what they are:

1. Bad or Missing Sectors: These are sectors that either cannot be read or are completely missing from the disk. These bad sectors can be created with a standard disk drive. See the "Disk Back-up" option of Scanalyzer for details.

2. Forced CRC Errors: Every time you write a sector, your disk drive calculates some totals that are written to a special area of the disk. When the sector is read, the disk recalculates the totals and compares them to what is stored on the disk. The disk can tell that it is a good sector if the totals match. You can protect a disk by forcing the totals to not match on certain sectors.

3. Bad Data Marks: Each sector contains data marks. Data marks indicate what kind of data will be in the sector (i.e. double density, etc.). Using invalid data marks is another protection method since the computer can check for them.

4. Duplicate Sectors: In these cases, the disk contains two or more sectors with the same sector number, but the sectors contain different data. The computer can read the sector several times to see if it gets the same data. If different data is obtained, then it knows it is on the original disk. The program will find the same data every time if it is on a copy disk because a normally formatted disk no two sectors have the same number.

The protection methods mentioned above can be combined together for added protection. Also, the data contained in sectors with forced CRC errors, bad data marks or duplicates

can be needed by the program. This program not only identifies the protection, but shows you the data contained in the sectors. If you wish to learn more about these protection methods and how they are used, you can buy the book "Atari Software Protection Techniques" which is available from Alpha Systems.

Of the methods above, only bad sectors can be easily created on a standard disk drive. (See Disk Back-up section). Special hardware (i.e., Happy Enhancement from Happy Computer or the Archiver Chip from Spartan Software) is needed to easily create the other methods.

USING THE DISK SCANNER

When the Disk Scanner is selected from the main menu, the following options will appear:

1. ANALYZE & EDIT
2. FAST SCAN
3. RETURN TO MAIN MENU

Type the number you want and press RETURN.

1. ANALYZE AND EDIT. First you will be asked which disk drive you wish to use. Type a "1" if one is all you have. Next, you will be asked for a sector number to start with. Type any sector number from 1 to 720 then hit RETURN and that sector will be displayed with many options at the bottom. The one of the most interest is the HELP option. Type "H" from this screen and a HELP screen will be displayed which will explain the numerous other functions available in this mode. This is one of the more powerful functions of the Scanalyzer so read the HELP screen thoroughly.

2. FAST SCAN. Fast Scan will scan through any number of sectors on the disk and identify the form of protection they use. This part of the program only tells the form of protection (i.e. bad data marks, duplicate sectors, etc.). Write down the

sector numbers that are protected. Then enter the Scan & Edit option to display the data contained in the sectors and for further protection information.

CARTRIDGE READER

Cartridge Reader will read a 4, 8, or 16K cartridge and save its data to disk as a binary load file. This file can then be used with DATA ANALYZER to create an assembler source code listing compatible with Atari's Assembler Editor Cartridge. This allows you to list and analyze the programs stored on the cartridge.

USING THE CARTRIDGE READER

When Cartridge Reader is selected from the main menu, the screen will instruct you to insert a cartridge. Of course, this is very easy with the 600XL or 800XL, but special preparations must be made for the Atari 400 or 800.

To use this function with the 400 or 800, you must prepare your computer. First, open the door where you usually insert a cartridge. When you look inside, you will see a small slot on the right side. This slot contains a small switch that turns off your computer when the door is open. Your cartridge door must be open to insert the cartridge when the program tells you to. To keep your door open and the computer on, you must put something in the slot that contains the switch. With your computer turned off, find the switch and put a small object (i.e., a dull pencil point or pen cap) in the slot to keep the switch turned on. Another way is to lodge a folded piece of paper in the slot.

After you have done this, try turning on your computer with the cartridge door still open. If you have done it right, your computer will come on with no problems.

NOTE: If you have problems finding the tiny slot, watch closely as you close the door of your computer and you will see it on the far right.

INSERTING THE CARTRIDGE

You must gently, but firmly, insert the cartridge in the left slot (the only slot on XL computers) when told to do so by the program. Sometimes your computer may lock-up when you are inserting the cartridge. If this happens, just turn off your computer and try again. Try tilting the cartridge so that the right side goes in first.

When you have successfully inserted the cartridge, the screen will prompt you for a file name. (If the cartridge was a 16K, a 16 will appear on the right side of the screen.) At this point, remove the Scanalyzer disk, and place one of your own formatted disks in drive 1. Then enter the file name and hit RETURN.

The cartridge program will be saved to disk as that file name.

USING THE FILE

The file created by Cartridge Reader is a standard binary load file. As mentioned earlier, this file can be analyzed and converted to assembler language by the 'Data Analyzer' option of Scanalyzer. See instructions for that option.

RUNNING THE FILE

The purpose of the Cartridge Reader is to allow you to read and analyze cartridge software, but you will find that many of the early cartridges released before 1983 will run normally when loaded from DOS. Just load DOS and use option 'L' to load and run the file (be sure no cartridges are installed). If you try to run several of the newer cartridges, your computer will lock-up or behave strangely. This is because of a protection technique being used in most new cartridges. This is how it works:

When the cartridge program begins running, it attempts to save data on top of its own programs. This destroys the program and causes your computer to lock-up or fail to operate properly. However, when a real cartridge is installed, the area of memory where the program is stored is protected so the program runs normally because it can't overlay itself.

This simple method keeps cartridge software from running without the real cartridge installed.

There are two ways to get around this problem. The first is to find the places where the protection occurs and then remove them. Scanalyzer is a tremendous help in doing this, however, you must still have a good knowledge of assembler to do it. Scanalyzer's main purpose is to help you read and understand the programs. There is an easy way to make runnable back-up copies of the protected cartridges if that is what you want. Basically, you must trick your computer into thinking an actual cartridge is present. To do this requires a special piece of hardware. The trick is to insert a dummy cartridge in the cartridge slot after the file is loaded. That way, when the program checks that area of memory or tries to destroy itself, it senses an actual cartridge and runs properly. A regular cartridge won't work because its programming will be stored on top of the disk loaded program. You need a special cartridge that will disable the area of RAM where a real cartridge is stored.

For this purpose, a special cartridge known as the 'Impersonator' is available from Alpha Systems for only \$29.95 plus \$2.00 shipping and handling charge. This will permit you to make a perfectly runnable back-up copy of all your 4, 8, and 16K cartridges. This special cartridge permits the files saved with Scanalyzer to run perfectly. Of course, these back-ups are legal only if used to back-up your own cartridges.

See the back page for the address to order from.

DATA ANALYZER

The Data Analyzer is an invaluable tool to help you understand or debug programs. It will convert a virtually unreadable program into readable assembler language. It also shows you the names (which were assigned by Atari) for special locations in your computer. The assembler language is listed to the screen and can also be put on a disk file which is compatible with Atari's assembler Editor cartridge. This means that you can not only list the program in a readable fashion, but also modify and debug it easily with the Assembler Editor cartridge. The program will accept input to disassemble from four different media:

- A. Programs in memory (RAM).
- B. Programs on disk in binary files (those loadable by the DOS 'L' command).
- C. Programs from the boot sectors of any bootable disk.
- D. Programs from arbitrary sectors (any sectors you wish).

The output of the program will always go to the screen, but can also go to either a printer or a disk file (in the Assembler Editor cartridge format).

Formats for Entering Addresses

Addresses may be entered in either decimal or hexadecimal. To enter a decimal address, type the number, then hit RETURN. To enter a hexadecimal address, enter a dollar sign (\$) first, then the address. You must use either two or four hexadecimal digits. (Example: \$1B3C)

Formats for Disk File Specifications

You can enter file names in several ways. If drive 1 is used, "Dn:" is not necessary. For example, the following three specifications will be accepted as valid and all refer to the same file:

MY.SCR

D:MY.SCR

D1:MY.SCR

If another drive is used, then the file specification must start with "Dn:".

Program Initialization

The initialization of the disassembler is a three phase process. The first is initialization of the program itself, the second is specification of the input source, the third is the specification of the output destination.

Input Specification

The first data that the disassembler will request is the source of the input. the screen will appear as follows:

REPLY WITH DESIRED CODE

A INPUT FROM RAM

B INPUT FROM BINARY SAVE FILE

C INPUT FROM BOOT SECTORS

D INPUT FROM ARB. SECTORS

At this this point one of the four possible codes (A, B, C, or D) should be selected. If any other character is entered, the question is repeated. After the desired code is entered, further questions will be asked, depending on the input function requested.

Input from RAM

If input from RAM is selected, the program needs to know what address to start disassembling at. The program will ask:

ENTER STARTING ADDR:

The starting address may be any address in the machine, including any location now in ROM. The format used to input the address has already been described.

Input from Binary Save File

If input from a binary save (also known as a binary load file) is picked, the next question is for the name of the file to be disassembled. Before replying, make sure that the diskette containing the file is in the disk drive. The question looks like:

INPUT DISK FILE NAME:

Your reply may be any DOS file name as previously described. The binary file is the read to provide the disassembler with a map of all the sectors it occupies. The Disassembler also verifies that the file has the proper format of a binary file. If it does not, then one of the following messages will appear:

ILLEGAL BINARY FILE

LOGIC ERROR AT 11430

The first error indicates that the file does not have the proper binary header. The second indicates that the header is correct, but the internal format is invalid. In either case, the program will return to the main menu.

If the binary file contains more than 30 segments (resets of the location counter in assembler), the following message will appear:

FILE INCOMPLETE ARRAYS TOO SMALL

At this point, the Disassembler will continue with just the first 30 segments.

The Disassembler will then proceed to list the starting and ending addresses of the segments found in the binary file.

BINARY FILE SEGMENTS

Here is a brief explanation of what binary file segments are. Most binary load files are split up into several sections, called segments, that start at different places in memory. A segment can be a subroutine for a specific function, or may just be an area to store data. The Data Analyzer automatically finds and displays where these segments begin and end so you can easily check the area that you want without wasting your time reading unused locations. The screen will display the segments as shown below:

START	END
decimal address[hexaddress]	decaddr[hexaddr]
decaddr[hexaddr]	decaddr[hexaddr]

The address is shown in both hex and decimal format

The program will then request the starting address with the prompt:

ENTER STARTING ADDR:

The starting address may be entered as previously described, but it must lie within one of the segments starting

and ending addresses as listed.

Input from Boot Sectors

If input from the boot sectors of a bootable disk is requested, the following message will be displayed:

INSERT DISK WITH BOOT FILE AND HIT RETURN

The Disassembler will then automatically disassemble the boot sectors on that disk. The starting address is already contained in the boot sector itself, as the third and fourth bytes. The second byte determines the number of sectors that will be disassembled.

Input from Arbitrary Sectors

If input from arbitrary sectors is requested, the Disassembler assumes the sectors contain 128 bytes each of data read continuously into memory. The program will ask for the sector number of the first sector:

ENTER THE FIRST SECTOR NUMBER:

The next question the program will ask is the address that the first sector should occupy in memory:

ENTER THE ADDRESS OF THE FIRST SECTOR:

It too can be entered as previously specified. Next, enter the number of sectors you want disassembled:

ENTER NUMBER OF SECTORS:

The last question asked is for the address to start disassembling at:

ENTER STARTING ADDR:

Output Specifications

Next, you must decide where you want the output to go. The screen will say:

REPLY WITH DESIRED CODE

A DISK OUTPUT

B PRINTER OUTPUT

C NONE (screen only)

At this point, one of the three possible codes (A, B, or C) should be selected. If any other character is entered, the question will be repeated. After the desired code is entered, further questions will be asked depending on the output function requested.

Output to Disk

since output to disk has been chosen, the disassembler will create source code to a disk file in the format compatible with the Atari Assembler Editor cartridge. The Disassembler will request a DOS file specification as follows:

ENTER DISK FILE SPECIFICATION:

The file name may be as specified previously. At this point the file will be opened by the Disassembler.

NOTE: It is tricky to disassemble a file on disk (either binary, boot, or arbitrary) and put the output on a disk file with only 1 disk drive (unless, of course, it is to the same diskette).

NOTE: This program inserts the standard Atari labels in the program as well as two special labels. They are 'SIOV' for address \$E459 and 'DSKIN' for address \$E453. These labels must be defined to the Assembler/Editor if you want to reassemble the file. Use the standard LABEL function.

Output to Printer

The output will go to the printer and the screen.

No Output

If this option is selected, then the Disassembler will output onto the TV screen only.

NOTE: The TV screen will receive output in all three cases.

Program Operation

The basic operation of the Disassembler is to read bytes from the input source and format those bytes into 6502 assembler source code, and send it to the output device you select. Other aspects that must be understood are the process of label assignments and the method of restoring the program.

Label Assignment

The Data Analyzer supports certain symbols that are defined in page (0) of the 6502 memory. Those symbols will be put in the reconstructed source when detected. References to the byte past the symbol will be defined as 'symbol+1'. Two other symbols will be generated, They are 'SIOV' (for address \$E459) and DSKIN (for address \$E453) These two symbols will also cause the computer to beep since their use tends to indicate interesting places in the code (most software protection schemes use these addresses).

Stopping and Starting the Program

You can stop the program at any time by pressing the OPTION or START key. If output option A or B has been selected, then the following message appears:

REPLY Y TO CLOSE OUTPUT

If the reply is 'Y', then the current output file is closed. If not, then it is left open for the next disassembly request. In either case, the next input source is requested.

Restarting at Another Point

By pressing the SELECT key you can stop the program and restart it somewhere else. The following options occur depending on the input source selection that was made.

Input Source	Action
A	Process as OPTION or START
B address	Display segment list and ask for starting address
C	Process as OPTION or START
D	Ask for the starting address

DISK BACK-UP

Disk Back-Up is a bonus program for all Scanalyzer purchasers. It is recommended that you back-up programs before modifying them with Scanalyzer. Disk Back-Up contains a sector copier (which copies all normal sectors from the disk) and supplies two ways to create bad sectors.

NOTE: Disk Back-Up should be used only to copy your own disks for safekeeping. It is illegal to sell or give away copies of commercial software.

There are several things to know before using Disk Back-Up. Any time you press the wrong key and end up on the wrong screen, just press RETURN to get back to the menu. The term "source disk" refers to the original disk that you wish to copy, and "destination disk" refers to the disk that you will put the back-up on. Since this program can be used with more than one disk drive, it is necessary to specify which drive or drives you wish to use. If you have only one drive, always answer 1 when asked for source or destination drives.

Selecting Disk Back-Up from the main menu will give you a menu like this:

1. Copy whole disk
2. Return to menu
3. Write bad sectors - option 1
4. Adjust disk speed
5. Disk directory
6. Format disk

7. Write bad sectors - option 2

To back-up a disk, write protect your source disk (if it has a notch on the side, cover it). Prepare a blank formatted disk to use as a destination disk. Type option 1, COPY WHOLE DISK, and follow the instructions on the screen.

As the disk is coping, you may encounter bad sectors. When your drive attempts to read these sectors, you will hear a loud grinding sound and see a message. This is the sound of your drive trying to realign itself. Each bad sector will be shown on the screen, write them down because you will need them later. Complete the copy by inserting the source and destination disks as instructed. The more memory you have, the fewer times you will have to swap the disks.

If there were no bad sectors encountered during the copy, the disk is ready to run. However, if bad sectors were encountered, follow the write bad sectors procedure.

NOTE: If you wish to try the disk before putting the bad sectors on it, be sure to write protect it (some programs attempt to format copies!).

Writing Bad Sectors

There are several ways to put bad sectors on a disk. This program contains two methods which can be done with standard hardware.

Bad Sector option 1 requires that you slow your disk speed, but it is much quicker than option 2. Some types of disk drives cannot be slowed down enough to write bad sectors, so if yours is one of those, you must use option 2.

Using Option 1

The first step in using option 1 is to adjust your drive speed down to 220 +/- 10 RPMs. Before changing your drive speed, go to option 3 - ADJUST DRIVE SPEED. This option will help you get your speed properly adjusted. Try to write the bad sectors at the slowest possible speed without getting I/O errors. To get the best bad sectors, your disk should just

barely be able to write.

Adjusting Your Drive Speed

To adjust your drive speed, it is necessary to remove the top cover and adjust one screw. To remove the cover, pry off the four little tabs on the top of the drive with any sharp instrument. Then, with a standard phillips head screwdriver, loosen the four screws (under the tabs) that hold the cover down and gently lift off the cover. There are two basic types of Atari 810 disk drives. The newer drives have a circuit board across the top (see diagram "A"). The older drives have no circuit board across the top and have a large white plastic screw in the back left hand corner of the drive (see diagram "B"). This large white screw can be turned by hand to adjust your speed. It is very sensitive, so a quarter turn may be all you need.

The newer drives are a bit trickier to adjust. To find the speed adjustment, look for a small green box with a tiny silver screw in it. It is located to the rear and a little left of center in the drive. It is very small, but it can be adjusted using a micro-screwdriver. The speed adjustment on the newer drives is not very precise, it may take as many as eight complete revolutions to properly adjust your speed.

The next step is to write the sectors. Use option 1 to do this. Enter the sectors you wish to create, then press RETURN. After all your bad sectors are complete, return to option 3 to adjust your drive speed back to normal.

Using Option 2

This option provides an alternate method of writing bad sectors. I still recommend option 1 as the faster and easier method. However, if your drive was purchased after January, 1983, or you own a non-Atari disk drive, you may need to use option 2.

To use this method, you must attach two long pieces of tape (folded over onto themselves) firmly to the top of the disk you wish to write bad sectors on (see diagram 1). Then, insert the disk in your drive so that the tape sticks out when the door is closed. (Be sure the tape is long enough to get a

DIAGRAM 1

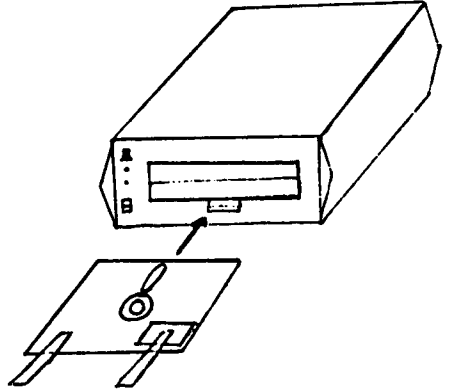
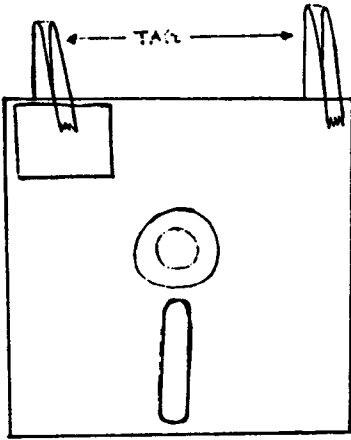


DIAGRAM A

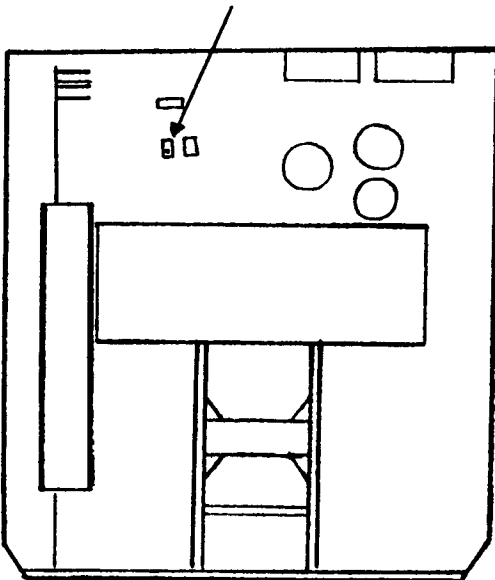
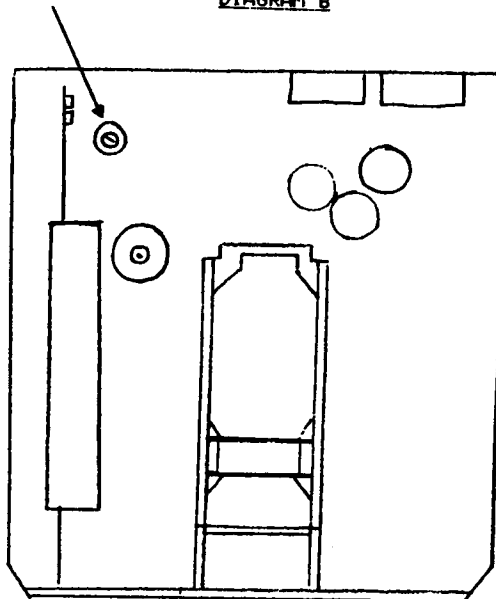


DIAGRAM B



grip on when the disk door is closed.) Next, enter the destination disk drive number (the disk you wish to write the bad sectors on) and the sector number. Be sure everything is set and type RETURN. The screen will now prompt you to shake the tape. You can gently move the tape back and forth and alternately push on one piece while pulling on the other. The computer will beep and signal you when the bad sector is written. Then stop pulling while it rechecks the sector. Note that until you get good at it, it can take 10 or more minutes to write a single bad sector. So keep at it and wait for the computer to signal you that it is done. If you wish to abort the process, hit any key.

INTRODUCTION DISPLAY

Introduction Display is there for those of you who like fancy program introductions, but hate to wait through them every time you load the disk. Just select Introduction Display from the main menu and sit back and relax and watch the show.

LIMITED WARRANTY

Alpha Systems warrants to the original purchaser of this computer software product that the recording medium on which the software programs are recorded will be free from defects in materials and workmanship for ninety days from the date of purchase. Defective media returned by the purchaser during that ninety day period will be replaced without charge, provided that the returned media have not been subjected to misuse, damage or excessive wear.

Following the initial ninety day warranty period, defective media will be replaced for a replacement fee of \$6.50.

Defective media should be returned to:

ALPHA SYSTEMS
4435 Maplepark Road
Stow, Ohio 44224

in protective packaging accompanied by: (1) a brief statement describing the defect; (2) a \$6.50 check (if beyond the ninety day warranty period; (3) your return address; and (4) the problem disk.

WHAT IS NOT COVERED BY THIS WARRANTY

THIS WARRANTY DOES NOT APPLY TO THE SOFTWARE PROGRAMS THEMSELVES. THE PROGRAMS ARE PROVIDED "AS IS".

THIS WARRANTY IS IN LIEU OF ALL OTHER WARRANTIES, WHETHER ORAL OR WRITTEN, EXPRESS OR IMPLIED. ANY IMPLIED WARRANTIES, INCLUDING IMPUTED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, ARE LIMITED IN DURATION TO NINETY DAYS FROM THE DATE OF PURCHASE. ALPHA SYSTEMS SHALL NOT BE LIABLE FOR INCIDENTAL OR CONSEQUENTIAL DAMAGE FOR BREACH OF ANY EXPRESS OR IMPLIED WARRANTY.

The provisions of the foregoing warranty are subject the laws of the state in which the disk is purchased. Such laws may broaden the warranty protection available to the purchaser of the disk.

