C.R.P.
SOFTWARE, INC.

100 QUARTZ WAY • SYRACUSE, NEW YORK 13219

(315) 488-0485

DISK WIZARD II

USER'S MANUAL

FOR THE ATARI 400/800/XL COMPUTERS

DISK WIZARD II

USER'S MANUAL

PROGRAM
WRITTEN BY:

P. A. CAMPANARO

MANUAL
WRITTEN BY:

A. L. ARBITAL
AND
R. W. PAKAN

Copyright © 1984 by C. A. P. Software, Inc.
Syracuse, New York 13219

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise without written permission from C.A.P. Software, Inc.. While every precaution has been taken in the preparation of this manual, C.A.P. Software assumes no responsibility for errors or omissions. Neither is any liability assumed for damages resulting from the use of the information contained herein.

TABLE OF CONTENTS

I. INTRODUCTION 1

II. BACKUP OF ORIGINAL 2

III. KEYBOARD PROTOCOL 2

IV. GETTING STARTED WITH DISK WIZARD II 3

V. BOOTING UP DISK WIZARD II 4

VI. DISK BACKUP 5
- CHANGE OPTIONS 6
- BACKUP DISK 8
- MAP DISK 10
- RETURN TO MAIN MENU 12

VII. DISK EDIT 14
- DISPLAY/MODIFY SECTOR 14
- SCAN SECTORS 19
- DISPLAY/PRINT DIRECTORY 23
- RECOVER/TRACE/REPAIR FILE 25
- FORMAT DISK 29
- DECIMAL/HEX CONVERSIONS 30
- CHANGE OPTIONS 30
- RETURN TO MAIN MENU 30

VIII. DISK SPEED 31
- DISPLAY DISK SPEED 31
- WRITE SECTORS 32
- CHANGE OPTIONS 35
- RETURN TO MAIN MENU 35

IX. DISASSEMBLER 36
- DISASSEMBLE BY SECTOR NUMBER 36
- DISASSEMBLE BY FILE NAME 40
- CHANGE OPTIONS 42
- RETURN TO MAIN MENU 42

X. TECHNICAL INFORMATION 43

XI. BACKUP PHILOSOPHY 47

XII. DISK PROTECTION METHODS 48

XIII. LIMITED WARRANTY AND DISCLAIMER 51

APPENDIX A 52

APPENDIX B 53

APPENDIX C 54

I. INTRODUCTION

Thank you for your purchase of C.A.P. Software's DISK WIZARD II. Our goal in designing the programs on this disk has been to make them as user friendly as possible.

IMPORTANT - fill out the owner registration card included with this manual. C.A.P. Software cannot provide support or future updates to nonregistered owners.

TRADEMARK ACKNOWLEDGEMENTS

ATARI,ATARI 400, ATARI 800, ATARI 1200XL and ATARI 810 are registered trademarks of ATARI Inc.,
a Warner Communications Co.

II. BACKUP OF ORIGINAL

We strongly suggest that before proceeding any further, you make a backup copy of your DISK WIZARD II disk. Please refer to Appendix A for the Backup Procedure.

III. KEYBOARD PROTOCOL

The keyboard entries follow the format shown below:

- When a prompt is followed by two values separated by a slash, the entry can be either of the two values (e.g.,FORMAT BACKUP DISK(Y/N)). The entry may be either a Y or an N. Press [RETURN] after the value has been selected.

- When a prompt is followed by two values separated by a dash, the entry can be any value between the first and second value, inclusive (e.g.,ENTER START SECTOR(1-720)). The entry may be any integer between 1 and 720 including 1 and 720. Press [RETURN] after a value has been selected.

- When a prompt consists of a word or words with the first letter in inverse video, the entry is the first letter of that word (enter non-inverse letters only).

- If at any time an incorrect keyboard entry is made, the computer will "Beep" and wait for the correct entry. If you're pressing the correct keys and are still getting a "Beep" check to make sure the keyboard isn't set for inverse video or lower case.

- If you type the wrong character, hit DELETE BACK S to correct.

- Pressing the escape key will return you to the main menu of the program you are currently running.

IV. GETTING STARTED WITH DISK WIZARD II

DISK WIZARD II requires a minimum of 40K of RAM and supports 1 to 4 drives.

DW II contains four major programs:

1. DISK BACKUP: Makes a sector by sector copy of any disk. It also contains a mapping function that provides a summary of the status of all sectors on any disk.

2. DISK EDIT: Enables the user to: review and/or modify data on a sector, scan sectors, display/print the disk directory, recover deleted files, trace and repair damaged files, format disks with lockout of bad sectors, and perform hex/decimal conversions

3. DISASSEMBLER: Allows disassembly of machine language directly from disk by file name or sector number. ATARI memory location mnemonics may be incorporated in the disassembly.

4. DISK SPEED: Provides display of exact disk drive speed. Also allows for creation of bad sectors on an ATARI 810 disk drive.

A detailed description of each of the programs is presented in sections VI, VII, VIII, and IX of this manual.

V. BOOTING UP DISK WIZARD II

The following procedure should be used to load the DISK WIZARD II disk into your ATARI computer:

1. Turn on your TV or Monitor.

2. Remove any cartridges that are installed in your computer.

3. Turn on the disk drive and wait for the busy light to go out. Insert the DISK WIZARD II disk into disk drive 1 and turn on your computer. DISK WIZARD II is now being loaded. When the program load is complete you will see the DISK WIZARD II MAIN MENU on the screen as shown below:

```
C. A. P. SOFTWARE
PRESENTS
DISK WIZARD II

DISK BACKUP
DISK EDIT
DISK SPEED
DISASSEMBLER

SELECT AND START
S/N 000000
REL. 2.X (C) 1984
```

At this point you are ready to select one of the four DISK WIZARD II programs. Press the SELECT button until the desired program title is highlighted. Press the START button to run the selected program.

VI. DISK BACKUP

When the DISK BACKUP program is selected, you will see the following display:

```
DISK BACKUP

OPTIONS
SOURCE DRIVE - 1      DEST. DRIVE - 1
DENSITY - SINGLE      FORMAT - YES
EMP SECT WRT - YES    FAST WRITE - NO
START SECTOR - 1      END SECTOR - 720

DISK BACKUP MENU

1 BACKUP DISK
2 MAP DISK
3 CHANGE OPTIONS
4 RETURN TO MAIN MENU

ENTER NUMBER (1-4)
```

This program is used to make a sector by sector copy of a disk or to produce a DISK MAP of the sectors on a disk. An option table is located at the top of the display. These options may be changed to meet your specific requirements. Although they are not displayed this way, for ease of understanding, the menu selections will be discussed in the following order:

3. CHANGE OPTIONS

1. BACKUP DISK

2. MAP DISK

4. RETURN TO MAIN MENU

Before continuing, let's define the following two terms:

SOURCE DISK: The disk which contains the program you wish to copy.

BACKUP DISK: A blank disk that will contain a copy of the source disk program after running DISK BACKUP.

3. CHANGE OPTIONS

If you wish to change any of the options in the option table, type 3 [RETURN]. A prompt for each option will then be displayed one at a time. Each prompt must be answered before continuing on.

OPTIONS

ENTER SOURCE DRIVE (1-4) - Enter the number of the disk drive that will contain the source disk. Pressing just [RETURN] will cause the program to default to the value presently shown in the option table.

ENTER DESTINATION DRIVE (1-4) - Enter the number of the disk drive that will contain the destination disk. Pressing just [RETURN] will cause the program to default to the value presently shown in the option table.

ENTER DENSITY?(S/D) - Type S [RETURN] if your disk drive(s) is a single density drive. Press D [RETURN] if you have a double density drive(s) with a double density disk installed. Pressing just [RETURN] will cause the program to default to the value presently shown in the option table.

FORMAT BACKUP DISK?(Y/N) - TYPE Y [RETURN] if you wish to format the backup disk. A blank disk must be formatted before any data is written onto it. (CAUTION-Formatting a disk destroys any data previously saved on that disk.) Type N [RETURN] if you do not wish to format the backup disk. You will not want to format the disk if you are doing a partial copy and just wish to transfer selected sectors to a disk that contains other data. Pressing just [RETURN] will cause the program to default to the value shown in the option table.

WRITE EMPTY SECTORS?(Y/N) - If N [RETURN] is depressed, any empty sectors that are encountered on the source disk will not be written on the backup disk. An empty sector is a sector that contains all zero data. This option saves time if there are many empty sectors on the source disk. The rationale behind not writing the empty sectors is as follows: When a disk is formatted on an Atari Disk Drive, all 720 sectors will be written with all zero data. Thus, if we read a source disk and encounter a sector with all zero data, there is no need to write this sector on the backup disk. DO NOT use this feature (i.e. do not type N [RETURN]) if the backup disk is not being formatted because there is no guarantee that a given sector will contain all zero data. If the disk used as a backup once had data on it, the sectors will not be all zeroes. Empty sector numbers are identified on the screen as they are read from the source disk and will also

be identified on the disk map (see MAP DISK).

If Y [RETURN] is pressed then all empty sectors on the source disk will be written on the backup disk.

Pressing just [RETURN] will cause the program to default to the value presently shown in the option table.

****IMPORTANT NOTE****

If you are using a PERCOM Disk Drive (or any other Drive that formats with non zero data), type Y [RETURN] to write empty sectors on the backup disk.

****

FAST WRITE?(Y/N) - Normally data is written onto the disk then read back by the drive to verify proper data transfer.

If Y [RETURN] is pressed, data written onto the disk is not read back for verification. Selecting this option increases the copying speed at the expense of not verifying the data written on the disk.

Pressing N [RETURN] allows the normal read after write verification of data transferred onto the disk.

Pressing just [RETURN] will cause the program to default to the value presently shown in the option table.

PARTIAL COPY?(Y/N) - If N [RETURN] is pressed in response to the PARTIAL COPY prompt, the entire disk will be copied (START SECTOR=1 and END SECTOR=720). The program will then display the original DISK BACKUP MENU.

If just [RETURN] is pressed the program will default to the START and END sector numbers shown in the option table. The program will then display the original DISK BACKUP MENU.

Pressing Y [RETURN] allows you to make a partial copy of a disk. A disk contains 720 sectors, starting with sector number 1 and ending with sector number 720. If it is known that the program on the source disk you wish to copy has less than 720 sectors of data or for some other reason you wish to transfer a certain group of sectors, then the partial copy option can be employed. After you type Y [RETURN], the following prompt will appear on the display:

ENTER START SECTOR NUMBER(1-720)

Enter the number of the first sector you wish to copy then press [RETURN]. The following prompt will then be displayed:

ENTER END SECTOR NUMBER(X-720)

<<Where X is the starting sector number you entered>>

Enter the number of the last sector you wish to copy then press [RETURN]. For example, if you know a program occupies sectors 1 through 200 on a particular disk, setting the START SECTOR NUMBER=1 and the END SECTOR

NUMBER=200, the program can be copied without taking the time to copy sectors 201-720.

After changing the desired options, you are now back to the original DISK BACKUP MENU. You may now do one of the following:

1. TYPE 1 [RETURN] - This will start the Backup procedure, described in detail in the section DISK BACKUP PROCEDURE.

2. TYPE 2 [RETURN] - This allows you to generate a map of the disk. Refer to the MAP DISK section.

3. TYPE 3 [RETURN] - This allows you to change the options in the OPTION TABLE again.

4. Type 4 [RETURN] - This will return you to the DISK WIZARD II MAIN MENU where you may re-select the program that you wish to run.

DISK BACKUP PROCEDURE

1. BACKUP DISK

Typing 1 [RETURN] allows you to start the copying process.

NOTE

If at any time during the copying process you wish to stop, press the ESCAPE KEY [ESC]. The computer will halt the copying process and return you to the DISK BACKUP MENU.

****

If you are using two disk drives please read this section first and then refer to the additional information at the end of this section.

COPYING USING ONE DISK DRIVE

The following prompt will be displayed on the screen:

INSERT SOURCE DISK IN DRIVE X
PRESS RETURN TO CONTINUE

At this point, remove the DISK WIZARD II disk from the disk drive and insert the disk with the program you wish to copy (SOURCE DISK).

NOTE

It is a good idea to make sure the source disk has the write protect tab installed to prevent any inadvertent damage.

****

Press [RETURN] and the source disk will be read sector by sector. As each sector is read from the disk, its number and status will be displayed on the screen.

A sample display is shown below:

DISK BACKUP

```
[OPTIONS]
SOURCE DRIVE - 1      DEST. DRIVE - 1
DENSITY - SINGLE      FORMAT - YES
EMP SECT WRT - YES    FAST WRITE - NO
START SECTOR - 1      END SECTOR - 720
```

READING -- 19

SECTOR 18

STATUS -- EMPTY

A sector can be one of the following four types:

1. DATA Sector - The sector contains all good data.
2. EMPTY Sector - The sector contains all zero data.
3. CRC Error - The sector can be located by the disk drive but the data that is read from that sector fails the CRC check (similar to a checksum failure).
4. BAD Sector - Either the disk drive cannot locate the sector or it cannot read the data from the sector.

Sector data will continue to be read into computer memory until memory is full or the End Sector is read.

INFORMATION NOTE

If the Backup program encounters any sectors with the same exact data as the previously read sector, then a "duplicate sector" flag is set internal to the program and the latter of these sectors is not stored. This feature saves memory space and therefore more sectors can be read in a single "pass".

****

The following prompt will now be displayed:

INSERT BACKUP DISK IN DRIVE X
PRESS RETURN TO CONTINUE

When the busy light goes out on the disk drive unit, remove the source disk and insert a blank disk (BACKUP DISK) and press [RETURN]. If the FORMAT BACKUP DISK option is enabled the backup disk will now be formatted. After the format operation is complete, the backup procedure will continue. Sector data will now be written on the backup disk. As each sector is written on the disk the sector number will be displayed on the screen. This will continue until all sectors previously read into memory are written onto the backup disk. If the backup process is not complete (there are more sectors to be copied) the prompt INSERT SOURCE DISK IN DRIVE X , PRESS RETURN will again be displayed. Continue swapping source and backup disks as indicated by the display prompts until copying is complete. Copying is complete when you see the SECTOR COUNTS display. For an explanation of this display and the disk map display, refer to the MAP DISK section.

If the backup copy of the program does not run, please refer to section XII of the manual, DISK PROTECTION METHODS.

COPYING USING TWO DISK DRIVES

If you are using two disk drives the following prompt is displayed:

INSERT SOURCE DISK IN DRIVE X
INSERT BACKUP DISK IN DRIVE X
PRESS RETURN TO CONTINUE

At this point, remove the DISK WIZARD II disk from drive 1 and insert the source disk into the appropriate drive.

Insert a backup disk into the appropriate drive and press [RETURN].

The backup process will now commence and the program will automatically switch between the source and destination drives. At the completion of the disk backup, the SECTOR COUNTS will be displayed as discussed in the MAP DISK section.

2. MAP DISK

Choosing this menu selection allows you to make a map of any disk. A disk map is a display showing the status of each sector that is read by the computer.

Type 2 [RETURN] and the following prompt will be displayed:

INSERT SOURCE DISK IN DRIVE X
PRESS RETURN TO CONTINUE

Insert the disk that you wish to map into the drive indicated by the display, press [RETURN].

The computer will now read the selected sectors (START SECTOR to END SECTOR in the option table) from the source disk. When the read is complete, the SECTOR COUNTS display will appear on the screen as follows:

DISK BACKUP

OPTIONS

SOURCE DRIVE - 1	DEST. DRIVE - 1
DENSITY - SINGLE	FORMAT - YES
EMP SECT WRT - YES	FAST WRITE - NO
START SECTOR - 5	END SECTOR - 720

SECTOR COUNTS

NUMBER OF DATA SECTORS	- 587
NUMBER OF EMPTY SECTORS	- 112
NUMBER OF BAD SECTORS	- 14
NUMBER OF CRC ERRORS	- 3
LAST DATA SECTOR	- 622

MAP,PRINT

The SECTOR COUNTS display will include the following information about the disk just mapped or copied:

The number of data sectors
The number of empty sectors
The number of bad sectors
The number or CRC error sectors
The sector number of the last sector containing non-zero data

The following prompt is displayed at the bottom of the screen:

MAP,PRINT

Typing P [RETURN] will print the information shown on the screen.

Typing M [RETURN] will display the DISK MAP (first half) as shown in the example display below:

```
                DISK BACKUP
                  DISK MAP
  1   ....*   **EEE   EEEEE   EEEBB   20
 21   BBBBB   *BCBB   BCB*C   B****   40
 41   *****   *****   *****   *****   60
 61   *****   *****   *****   *****   80
 81   *****   *****   *****   *****   100
101   *****   *****   *****   *****   120
121   *****   *****   *****   *****   140
141   *****   *****   *****   *****   160
161   *****   *****   *****   *****   180
181   *****   *****   *****   *****   200
201   *****   *****   *****   *****   220
221   *****   *****   *****   *****   240
241   *****   *****   *****   *****   260
261   *****   *****   *****   *****   280
281   *****   *****   *****   *****   300
301   *****   *****   *****   *****   320
321   *****   *****   *****   *****   340
341   *****   *****   *****   *****   360

HALF,PRINT
```

This display shows the status of sectors 1 - 360. The following symbols are used to indicate the sectors' status:

- * - DATA SECTOR
- E - EMPTY SECTOR
- C - CRC ERROR
- B - BAD SECTOR
- . - SECTOR NOT READ

The prompt HALF,PRINT is displayed at the bottom of the screen.

Typing H [RETURN] will display the second half of the DISK MAP (Sectors 361 - 720). Typing H [RETURN] again will redisplay the first half of this map.

Typing P [RETURN] will print the entire DISK MAP (Sectors 1-720).

Typing [RETURN] will return you to the SECTOR COUNTS display. Type [RETURN] again to return to the DISK BACKUP MENU.

4. RETURN TO MAIN MENU

This selection is used to exit from the DISK BACKUP program and return directly to the DISK WIZARD II MAIN

MENU unless a disk backup was performed. Since the disk backup process uses all of memory to store sector data, DISK WIZARD II must be reloaded into memory if you wish to run any of the other program selections. If this is the case, you will see the message: INSERT DISK WIZARD II IN DRIVE 1 PRESS RETURN. Following the instructions on the screen will reboot the computer to bring you back to the DISK WIZARD II MAIN MENU.

VII. DISK EDIT

Upon selecting the Disk Edit program, you will see the following display:

```
                    DISK EDIT

                  OPTIONS
DRIVE - 1                DENSITY - SINGLE

                DISK EDIT MENU

1 DISPLAY/MODIFY        5 FORMAT DISK
  SECTOR

2 SCAN SECTORS          6 DECIMAL/HEX
                          CONVERSIONS

3 DISPLAY/PRINT         7 CHANGE OPTIONS
  DIRECTORY

4 RECOVER/TRACE/        8 RETURN TO
  REPAIR FILE             MAIN MENU

ENTER NUMBER (1-8)
```

The following sections on the features of Disk Edit assumes knowledge on your part of the layout of a DOS structured disk. Further information on this subject can be found in SECTION X of this manual. Also, several examples are given that require a test disk. Please refer to Appendix B for instructions on creating your test disk.

1. DISPLAY/MODIFY SECTOR

This menu selection allows you to display and/or modify any sector on a disk. To select this section type 1 [RETURN]. The following prompt will appear on the screen:

ENTER SECTOR NUMBER (1-720)

Enter the number of the sector that you wish to inspect. When you enter the number and press [RETURN], the sector will be read and displayed on the screen in hex format. Following is an example of the sector display:

SECTOR 4

	08	19	2a	3b	4c	5d	6e	7f	ASCII
0	AA	08	14	0B	BE	0A	CB	09	.....
8	00	0B	A6	0B	07	85	44	AD	.....D.
10	0A	07	8D	D6	12	AD	0C	07	.....
18	85	43	AD	0D	07	85	44	AD	.C....D.
20	0A	07	8D	0C	13	A2	07	8E	.....
28	0D	13	0E	0C	13	B0	0D	A9	.....
30	00	9D	11	13	9D	29	13	9D	.....)...
38	31	13	F0	36	A0	05	A9	00	1..6....
40	91	43	E8	8E	01	03	A9	53	.C.....S
48	8D	02	03	20	53	E4	A0	02	... S...
50	AD	EA	02	29	20	D0	01	88	...) ...
58	98	AE	0D	13	9D	11	13	A5	.....
60	43	9D	29	13	A5	44	9D	31	C.)..D.1
68	13	20	70	08	88	F0	03	20	. P....
70	70	08	CA	10	B2	AC	09	07	P.....
78	A2	00	A9	00	88	00	05	7D	.....

FN = 0 NEXT SECT = 5 BYTES = 125

ATASCII,ENTER,MOD,NXT,+,-,PRT,WRT

The sector display is divided into two parts. The left half is an 8x16 matrix containing the hex bytes that are stored on the disk. The right half displays the ASCII(ATASCII) equivalent of these hex bytes. If a dot is displayed in the ASCII(or ATASCII) table, it means that the corresponding data byte is non printing.

The inverse video bands across the top and down the left side of the display are used as an address matrix to help you locate bytes within the sector. To determine the address of any byte in the sector, take the corresponding left column address and if it ends in zero, use the left most digit of the top row address (i.e.; 0,1,2,3,4,5,6,7). If the address ends in an eight, use the right most digit of the top row address (i.e.; 8,9,a,b,c,d,e,f). The address of each byte in the addressing matrix is shown below:

	08	19	2a	3b	4c	5d	6e	7f
0	00	01	02	03	04	05	06	07
8	08	09	0A	0B	0C	0D	0E	0F
10	10	11	12	13	14	15	16	17
18	18	19	1A	1B	1C	1D	1E	1F
20	20	21	22	23	24	25	26	27
28	28	29	2A	2B	2C	2D	2E	2F
30	30	31	32	33	34	35	36	37
38	38	39	3A	3B	3C	3D	3E	3F
40	40	41	42	43	44	45	46	47
48	48	49	4A	4B	4C	4D	4E	4F
50	50	51	52	53	54	55	56	57
58	58	59	5A	5B	5C	5D	5E	5F
60	60	61	62	63	64	65	66	67
68	68	69	6A	6B	6C	6D	6E	6F
70	70	71	72	73	74	75	76	77
78	78	79	7A	7B	7C	7D	7E	7F

Below the Sector Matrix you will see a line containing the DOS link bytes FN, NEXT SECT and BYTES. FN is the file number, NEXT SECT is the next sector pointer, and BYTES is the number of bytes used in the sector.

NOTE

If the sector being displayed on the screen is non-DOS format, then the values for the above symbols will be meaningless.

****

At the very bottom of the display you will see the following prompt:

ATASCII,ENTER,MOD,NXT,+,-,PRT,WRT

You may enter either A,E,M,N,+,-,P, or W corresponding to ATASCII, ENTER, MODIFY, NEXT, PLUS ONE, MINUS ONE, PRINT or WRITE respectively.

A. TYPING A [RETURN] will cause the right half of the screen to display the ATASCII equivalent of the hex bytes in that sector. Typing A [RETURN] again will revert the display back to ASCII format.

B. TYPING E [RETURN] allows you to enter the number of another sector that you wish to display. You will then see the following prompt:

ENTER SECTOR NUMBER (1-720)

Enter the number of the sector you wish to display then press [RETURN].

C. TYPING M [RETURN] allows you to change (modify) the data in the display. You will see the following prompt at the bottom of the screen:

ENTER HEX ADDRESS (0-7F)

Type in the address of the byte that you wish to change and [RETURN]. You will notice that this byte is highlighted and will now see the prompt:

ENTER NEW HEX DATA (0-FF)

NOTE

If the highlighted byte is not the one that you thought you had selected, then press [RETURN] and recalculate the address of the desired byte.

****

Type in the new data (in hex format) for this address and [RETURN]. The new data will be displayed in the selected address and the next sequential address will be highlighted. This is done to ease entry of strings of data. If you do not wish to enter data at the newly highlighted address then just press [RETURN]. You will again see the prompt:

ENTER HEX ADDRESS (0-7F)

If you wish to change another address just enter it as you did before.

If you do not wish to enter any more data in the display, then just press [RETURN]. If you wish to write the new data on the disk, use the write command.

D. TYPING N [RETURN] will enable you to display the next sector in the file. The next sector number is displayed in the file links at the bottom of the screen.

E. TYPING + [RETURN] will display the contents of the next consecutive sector on the disk. For example, if sector 50 is currently being displayed, typing + [RETURN] will cause sector 51 to be displayed and so on.

F. TYPING - [RETURN] will display the contents of the sector preceeding the sector displayed on the screen. As in the previous example typing - [RETURN] would display sector number 49.

G. TYPING P [RETURN] prints the sector currently displayed on the screen. ASCII will be printed even if ATASCII is displayed on the screen.

H. TYPING W [RETURN] enables you to write the sector information back on the disk. You will see the following prompt:

WRITE - ENTER SECTOR NUMBER (1-720)

Enter the number of the sector you wish to write to. It can be the sector you originally read from or a different one. After entering the number and pressing [RETURN], the computer will write the sector then reread and redisplay the sector.

USING DOUBLE DENSITY

The following paragraphs explain the differences in using the DISPLAY/MODIFY SECTOR section for a Double Density disk. If you haven't already done so, we suggest that you read the DISPLAY/MODIFY SECTOR section in its entirety before continuing.

The sector display for double density is broken up into two separate 128 byte matrix displays. When a sector number is selected, the screen will display the first 128 bytes(0-7F) of the sector and the following prompt :

ATASC,ENT,HALF,MOD,NXT,+,-,PRT,WRT

This is essentially the same as in single density mode with the addition of a HALF command. Typing H [RETURN] allows you to toggle between the two halves of the double density display.

To modify a double density sector, use the MOD command as in single density. If you wish to change both halves of the sector, MOD the first half; display the second half of the sector; MOD that half and then use the WRT command to write the new data to the disk.

INFORMATION NOTE

In double density mode, the addresses of the DOS control bytes are FD-FF.

****

There is one more important point about disks that are written in Double Density:

SECTORS 1, 2 AND 3 OF ALL DISKS ARE WRITTEN IN SINGLE DENSITY. IF YOU ARE IN DOUBLE DENSITY MODE, THE PROGRAM WILL AUTOMATICALLY SWITCH TO SINGLE DENSITY MODE FOR THESE SECTORS.

EXAMPLE 1

Let's try an example using the test disk you created in Appendix B. Return to the Disk Edit Main Menu and insert your test disk into drive 1. Type 1 [RETURN] to display/modify sectors. Now enter the sector number you want to look at. For this example enter sector 100 then press [RETURN]. To modify some data on this sector, type M [RETURN] (modify). Enter a hex address of 7D then press [RETURN]. Notice how the byte at address 7D is highlighted. Now enter data of 00 then press [RETURN]. Byte 7D now equals 00 and the next byte at 7E is highlighted. Enter data 08 then press [RETURN]. The next sector number is now 08 and the file number is equal to zero. 7D/7E are the control bytes used by DOS to tell it what the file number and next sector are. The byte at address 7F is now highlighted. Enter hex data of 05 then press [RETURN]. The byte count has changed since byte 7F is the control byte that contains the number of bytes used in that sector.

Experiment on your own and try entering data at several different addresses, using the same sector. When you're done experimenting, press [RETURN] until you see the following prompt:

ATASCII,ENTER,MOD,NXT,+,-,PRT,WRT

Type W [RETURN]. Enter 100 [RETURN]. The new data will now be written on the test disk. The sector will be reread and displayed again along with the original prompt.

Now type A [RETURN]. The right half of the screen now displays the byte equivalent in ATASCII instead of ASCII. Type A [RETURN] again and ASCII will be displayed. This comes in handy when looking for ASCII code in a program.

If you have a printer turn it on and print the sector. Type P [RETURN]. The message "Printing Sector" is displayed at the bottom of the screen while the sector is being printed.

Type E [RETURN]. This allows you to enter a new sector number. You may enter any number between 1 and 720, but for this example let's enter 20 [RETURN]. Sector number 20 will now be read in and displayed.

Type N [RETURN] to display the next sector. The next sector, as defined by the control bytes, will now be read in and displayed. Remember this only works for a DOS formatted file. Try this a few times to trace through the file.

Note what sector number you have displayed. Now, type + [RETURN] to display the next consecutive sector. If you had typed - [RETURN] instead of + [RETURN] the screen would have displayed the preceeding sector. Type [RETURN] to get back to the DISK EDIT MENU.

2. SCAN SECTORS

Typing 2 [RETURN] allows you to scan through any disk, sector by sector, searching for a group of bytes that comprise a certain instruction or piece of data. The following display will appear:

```
DISK EDIT

OPTIONS
DRIVE - 1                DENSITY - SINGLE

SCAN SECTORS

1 BYTE SCAN

2 STRING SCAN

ENTER NUMBER (1-2)
```

This search can be done in either of two ways.

The first way is a BYTE SCAN (this is selected by typing 1 [RETURN]). When BYTE SCAN is selected the screen display will look as follows:

```
DISK EDIT

OPTIONS
DRIVE - 1                DENSITY - SINGLE

SCAN SECTORS

__ __ __ __ __ __ __ __

__ __ __ __ __ __ __ __

__ __ __ __ __ __ __ __

__ __ __ __ __ __ __ __

ENTER BYTE (0-FF)
```

As you see there are 32 blanks on the screen. You may enter up to 32 bytes for any scan.

To enter data just type in each byte followed by a [RETURN]. Any single digit byte entered will have a 0 put before it (i.e.; 4 will become 04). You will notice that the previously entered byte appears in the top display. Repeat this procedure until all desired bytes are entered. At this time just press [RETURN].

The screen will now display the series of bytes that were entered and the prompt:

ENTER START SECTOR (1-720)

Enter the number of the sector that you wish to start the search at. After entering the START SECTOR, you will be

asked to enter the END SECTOR for the search. When the start and end sectors are both entered the following prompt will appear:

ENTER NUMBER OF BYTES (125/128) [or (253/256) for double density]

DOS files contain 125 data bytes (253 data bytes for double density). The last three bytes are control bytes. This bytes per sector question is necessary for the simple fact that if the string you are searching for spans across two sectors, you must tell the computer to ignore the control bytes and continue the data byte scan in the next sector. For all DOS files you should input 125 (or 253) [RETURN] for this question. If, however, you are doing a byte scan in a program that is not DOS formatted, the sector contains 128 data bytes (256 for double density) and all must be checked in the search. For this case you should type 128 (or 256) [RETURN].

Upon entering the number of data bytes per sector the computer will begin to scan the selected sectors for the desired series of bytes.

If a match is found, then the screen will display the sector in which the match was found and the first byte of the match will be highlighted. Keep in mind that if the series of bytes is not all contained in one sector then the computer will only display the sector which contains the first byte of the series. The display will appear as in the example shown below:

```
SECTOR 4

   08 19 2a 3b 4c 5d 6e 7f        ASCII
 0 AA 08 14 0B BE 0A CB 09        ........
 8 00 0B A6 0B 07 85 44 AD        ......D.
10 0A 07 8D D6 12 AD 0C 07        ........
18 85 43 AD 0D 07 85 44 AD        .C....D.
20 0A 07 8D 0C 13 A2 07 8E        ........
28 0D 13 0E 0C 13 B0 0D A9        ........
30 00 9D 11 13 9D 29 13 9D        .....)..
38 31 13 F0 36 A0 05 A9 00        1..6....
40 91 43 E8 8E 01 03 A9 53        .C.....S
48 8D 02 03 20 53 E4 A0 02        ... S...
50 AD EA 02 29 20 D0 01 88        ...) ...
58 98 AE 0D 13 9D 11 13 A5        ........
60 43 9D 29 13 A5 44 9D 31        C.)..D.1
68 13 20 70 08 88 F0 03 20        . P....
70 70 08 CA 10 B2 AC 09 07        P.......
78 A2 00 A9 00 88 00 05 7D        ........

FN = 0   NEXT SECT = 5   BYTES = 125

ATASCII,CONTINUE,PRINT
```

By typing A [RETURN] you may display the ATASCII equivalent on the right side of the display.

Typing P [RETURN] will print the sector currently being displayed.

Typing C [RETURN] will cause the scan operation to continue. Each time a match is found, it will be displayed as shown in the above display. When the last sector is scanned, the screen will display the SCAN COMPLETE message and will contain the number of matches that were found. Pressing [RETURN] will bring you back to the original SCAN SECTORS display. If you no longer wish to scan sectors just press [RETURN] again and you will see the DISK EDIT MENU displayed.

The second way to scan a sector is the STRING SCAN. The computer searches through selected sectors to find a string of numbers or letters. If STRING SCAN is selected the screen will look as follows:

```
                    DISK EDIT
                 OPTIONS
DRIVE - 1                DENSITY - SINGLE

                 SCAN SECTORS


ENTER STRING:
```

Type in the string that you want to locate and then press [RETURN].

As in BYTE SCAN you will be asked for START SECTOR, END SECTOR and NUMBER OF BYTES/SECTOR. After answering all of these questions, which are described in detail above, the computer will begin to search for the deisired string.

If an exact match is found the screen will display the sector in which the string was found. The display has the same format as the BYTE SCAN display with the same ATASCII, CONTINUE and PRINT options.

EXAMPLE 2

Here is an example using the scan feature of DISK WIZARD II. Ensure your test disk is installed in Drive 1. Return to the DISK EDIT MENU and type 2 [RETURN] for SCAN SECTORS. Now type 1 [RETURN] to do a byte scan. Let's just do a simple one, a scan for the letter A(or 41 in ASCII). Type in 41 then [RETURN]. Since this is the only byte we want to scan for, type [RETURN] again. Now type 4 [RETURN] to start the scan at sector 4. We'll scan 21 sectors, so now type in 24 [RETURN]. We now need to enter the number of bytes per sector and since this is a DOS formatted file, type 125 [RETURN]. The computer will now search until it finds our first 41 or A. Notice how it is

highlighted. Now type C [RETURN]. The computer will now continue the scan until it finds the next match. Keep typing C [RETURN] until you have finished scanning sector 24 (last sector). The screen will display the SCAN COMPLETE message showing the number of matches found.

Now let's try a quick string scan. Press [RETURN] to get back to the SCAN SECTOR display. Type 2 [RETURN] to do a STRING SCAN. Say you are trying to locate where in the DOS program the "WHICH DRIVE TO FORMAT?" question is. Here's how you do it. Type in WHICH DRIVE TO FORMAT? then press [RETURN]. Type 4 [RETURN] for start sector and type 100 [RETURN] for end sector. Type 125 [RETURN] to enter the number of bytes per sector. The computer is now searching for the string. It will take a few moments. After the string is found, it will be displayed on the screen. If no match was found, ensure the string was entered correctly. Remember, the string may cross a sector boundary so only the first portion may be displayed. Type [RETURN] twice to get back to the DISK EDIT MENU.

3. DISPLAY/PRINT DIRECTORY

Typing 3 [RETURN] allows you to display and/or print the directory sectors.

DISPLAY DIRECTORY - To display the first directory sector, 361, type 1 [RETURN]. An example display follows:

```
                    DISK EDIT
                 OPTIONS
DRIVE - 1                DENSITY - SINGLE

              DIRECTORY SECTOR 361

FN   FILENAME                START  LENGTH

 0 * DOS      SYS              4      39
 1 * DUP      SYS             43      42
 2   FILE1          (D)       85      33
 3   FILE2    EXT            118      46
 4   TEST     BAS            164      11
 5   AUTORUN  SYS            175      52
 6   PROGRAM  BAS            227      17
 7   COMMAND  ASM            244       2

          FREE SECTORS - 498

DISPLAY NEXT DIRECTORY SECTOR? (Y/N)
```

The left hand column shows the file number.

The next column contains the file name of the directory entry. If an * appears to the left of the file name, it means that the file is locked. If a (D) appears to the right of the file name, it means that that file has been deleted. A deleted file may be recovered, refer to the next section in DISK EDIT.

The next column contains the number of the sector where the program of that file name starts. A program or file always begins at byte 00 of the starting sector.

The last column contains the sector length, which is the total number of sectors contained in that file.

The second line from the bottom shows the number of free sectors. A free sector is a sector still available for storing data.

At the bottom of the display will be the following prompt:

DISPLAY NEXT DIRECTORY SECTOR? (Y/N)

If you want to look at the next consecutive directory sector press Y [RETURN]. The next directory sector will be displayed. All eight (8) directory sectors can be displayed in this manner. Typing N [RETURN] or just [RETURN] will return you to the DISPLAY/PRINT DIRECTORY display.

PRINT DIRECTORY - To print the directory type 2 [RETURN]. The following prompt will be displayed:

ENTER TITLE FOR PRINTOUT

Enter the title you want to appear at the top of the directory printout (up to 32 characters in length). Press [RETURN].

The directory will now be printed out (empty directory entries will not be printed out). An example is shown below.

SAMPLE DIRECTORY

FN		FILENAME		START	LENGTH
0	*	DOS SYS		4	39
1	*	DUP SYS		43	42
2		FILE1	(D)	85	33
3		FILE2 EXT		118	46
4		TEST BAS		164	11
5		AUTORUN SYS		175	52
6		PROGRAM BAS		227	17
7		COMMAND ASM		244	2

FREE SECTORS - 498

4. RECOVER/TRACE/REPAIR FILE

This section of Disk Edit is used to recover, trace or repair a deleted file. After typing 4 [RETURN], the following will be displayed on the screen:

DISK EDIT

OPTIONS

DRIVE - 1 DENSITY - SINGLE

RECOVER/TRACE/REPAIR FILE

1 RECOVER DELETED FILE

2 TRACE FILE LINKS

3 REPAIR FILE LINKS

ENTER NUMBER (1-3)

RECOVER

To recover a deleted file type 1 [RETURN]. The following is displayed:

DISK EDIT

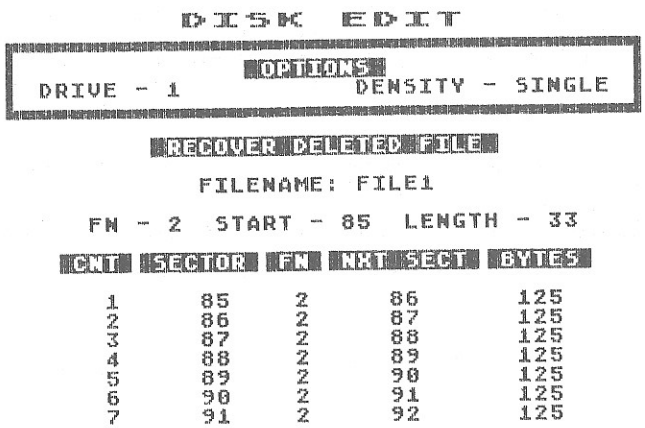
OPTIONS

DRIVE - 1 DENSITY - SINGLE

RECOVER DELETED FILE

ENTER FILENAME:

Type in the file name including any extension (i.e.; DUP.SYS). DO NOT INCLUDE THE DISK NUMBER OR USE A WILDCARD (i.e.; D1:DUP.*). The title and extension may contain up to 12 characters, type [RETURN]. The computer will now search for the deleted file. If the file is found with the delete bit set in the directory, the file links will be automatically verified. The contents of the file will be displayed as shown in the following example:



The sector links are verified to ensure that the entire file is complete and no data has been lost.

The file name is shown, along with the File Number (FN), the starting sector number (START) and the number of sectors contained in the sector (LENGTH).

NOTE

If you wish to stop the display to view sector information, press the space bar. Press the space bar again to restart the display.

****

If there were no errors encountered in verifying the file links, the following prompt is displayed:

RECOVER FILE? (Y/N)

Type Y [RETURN] to recover the file. When the screen displays FILE RECOVERY COMPLETE, the file has been recovered and can now be used. Typing N [RETURN] returns you back to the RECOVER/TRACE/REPAIR FILE MENU.

TRACE

To TRACE FILE LINKS, type 2 [RETURN]. Type in the file name of the file you wish to trace. The format of the file name is the same as that for recovering a file. Type [RETURN], the file links will now be traced and the sector information will be displayed in the same format as the example in the previous RECOVER FILE section. Again, the space bar may be used to start and stop the display. If the entire file is intact, the screen will display the

following:

TRACE COMPLETE WITH NO ERRORS

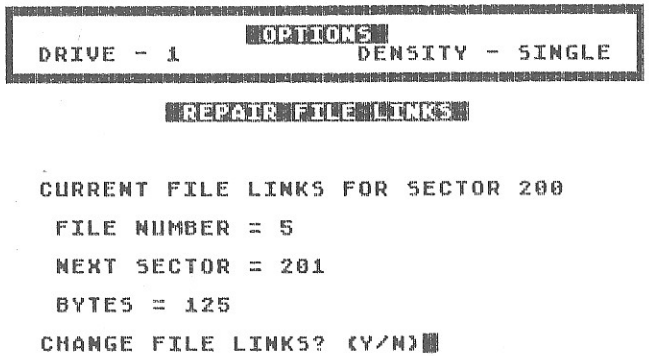
If an error is found, the error will be displayed, following the sector in which the error was found. Use this information to repair the file links (see next section). Type [RETURN] to return to the RECOVER/TRACE/REPAIR FILE MENU.

REPAIR

To REPAIR FILE LINKS type 3 [RETURN]. The following prompt is displayed:

ENTER SECTOR NUMBER (1-720)

Enter the sector number that needs to have the file links repaired then press [RETURN]. The current file link information for that sector will now be displayed. An example follows:



If you wish to change the file links, type Y [RETURN]. Enter the new file link data as required by the prompts. Pressing just [RETURN] will retain the file links data displayed on the screen. After all the data has been entered, the following prompt will be displayed:

WRITE DATA ON DISK? (Y/N)

Type Y [RETURN] to write the displayed file link data on the disk. After the operation is complete, the screen will again display the prompt:

CHANGE FILE LINKS? (Y/N)

If the file link data is now correct type N [RETURN] to return to the RECOVER/TRACE/REPAIR FILE Menu. If the file link data is still incorrect, then typing Y [RETURN] will enable you once again to enter data.

EXAMPLE 3

Ensure the test disk is installed in Drive 1. Return to the DISK EDIT MENU and type 3 [RETURN]. Now type 1 [RETURN] to display the directory. Notice the (D) on the same line as the TEST2 file. This file has been deleted and cannot be used. This portion of the example will show how to recover a deleted file, in this case the file TEST2 which was deleted when you made the test disk.

Return again to the DISK EDIT MENU. Type 4 [RETURN] to select the RECOVER/TRACE/REPAIR FILE portion of DISK EDIT, then type 1 [RETURN] to select RECOVER file. Enter the name of the file to be recovered by typing TEST2, then press [RETURN]. The program now searches the directory to make sure the file exists. The file links are then verified to ensure the entire file is complete. Type Y [RETURN] in response to the "RECOVER FILE?" question. The deleted file TEST2 has now been recovered. You may wish to look at the directory, as we did in the beginning of this example, to verify the file TEST2 is no longer deleted.

In the second part of this example we'll trace and repair a bad file. The bad file TEST1 should have been created when you wrote new data on sector 100 in EXAMPLE 1. Please make sure this was done before continuing.

Return to the DISK EDIT MENU and type 4 [RETURN]. Now type 2 [RETURN] to trace the file. Enter the file name by typing TEST1 then press [RETURN]. The program will now search the directory and trace the file links. A file number error (the first error encountered) should be displayed at sector 100. Note that the data for file number, next sector and number of bytes per sector is incorrect. These were the control bytes that were changed in EXAMPLE 1.

To repair the file, the following must be done. Looking at the display, it should be obvious that the file number should equal 2, the next sector should be 101 ad the bytes per sector should be 125. This data agrees with the previous sector data patterns. Press [RETURN] then type 3 [RETURN] to repair the file. We know the error is in sector 100, so enter the sector number by typing 100 [RETURN]. Type Y [RETURN] in response to the "CHANGE FILE LINKS?" question. Enter the file number by typing 2 [RETURN] , the next sector number by typing 101 [RETURN] and the nuber of bytes per sector by typing 125 [RETURN]. Type Y [RETURN] in response to the "WRITE DATA ON DISK?" question. After the new data has been written type N [RETURN]. The file TEST1 has now been repaired.

To verify that the file was indeed repaired, type 2

[RETURN] to again trace the file links. Type in TEST1 then press [RETURN]. If our new file link data is correct, the trace will complete with no errors.

Press [RETURN] twice to get back to the DISK EDIT MENU.

5. FORMAT DISK

Typing 5 [RETURN] allows you to format disks. You will see the following prompt displayed on the screen:

INSERT DISK TO BE FORMATTED INTO DRIVE X

CONTINUE WITH DISK FORMAT? (Y/N)

Insert the disk you wish to format into the appropriate drive and type Y [RETURN]. The disk drive will now format the disk. The screen will display DISK FORMAT COMPLETE when the format operation is finished. At this time you may either format another disk by typing Y [RETURN] or return to the DISK EDIT Menu by typing N [RETURN].

The disk format program in DISK EDIT has an advantage over using the DOS format option. Some diskettes cannot be formatted because they contain defects, or because they have become damaged. If a disk has a bad spot, and fails during format, the screen will display the sectors that could not be formatted. The format program now takes these sectors and "locks them out" in the VTOC (volume table of contents). Also a "file" is created in the directory with the name BADSECT.FMT. The length of this file is the total number of bad sectors on the disk. This will enable you to see if there are any bad sectors on the disk later on by simply obtaining a directory of the disk.

In order for a disk to be useable by DOS, however, sectors 1-3 and 360-368 are required to be good. These sectors are always used by DOS. If any of these sectors are bad, the screen will display the following message:

DISK IS UNUSABLE BY DOS

If the format was successful, the disk can now be used to store any DOS Format files. The reason this works is because the bad sector numbers have been set or "locked out" in the VTOC and DOS thinks that these bad sectors are already used, so no attempt is made to store data on them.

-HELPFUL HINT -

If you need all sectors to be good try formatting the disk several times until no bad sectors are displayed. Often two or three attempts will yield a successful format.

6. DECIMAL/HEX CONVERSION

This portion of the program is used to convert from hex to decimal and decimal to hex.

When you select item 6 from the DISK EDIT Menu the screen will display the following:

1-DECIMAL TO HEX
2-HEX TO DECIMAL

ENTER NUMBER (1-2)

If you wish to convert from decimal to hex just type 1 [RETURN]. You will see the following prompt:

ENTER DECIMAL NUMBER (0-65535)

You simply type in the decimal number you want converted to hex and [RETURN]. If you want to convert from hex to decimal then type 2 [RETURN]. The prompt:

ENTER HEX NUMBER (0-FFFF)

will appear on the screen. The question CONVERT ANOTHER HEX (OR DECIMAL) NUMBER? (Y/N) will be displayed at the bottom of the screen after a number has been converted. Type Y [RETURN] to convert another number or N [RETURN] to return to the DECIMAL/HEX CONVERSION Menu. If no more numbers are to be converted then press [RETURN]. This will bring you back to the DISK EDIT Menu.

7. CHANGE OPTIONS

When this section is selected by typing 7 [RETURN] you will be asked the following two questions:

ENTER DRIVE NUMBER (1-4) - Enter the number of the disk drive which contains the disk you wish to edit. If you do not wish to change this option simply press [RETURN]. The other question that will appear on the screen is:

ENTER DENSITY (S/D) - If you are using a single density disk drive just type S [RETURN]. If you are using a double density disk drive then enter D [RETURN]. You will now find yourself back to the DISK EDIT Menu with the options changed to the number or density you wanted.

8. RETURN TO MAIN MENU

Type 8 [RETURN] to exit from the DISK EDIT program and return to the DISK WIZARD MAIN Menu.

VIII. DISK SPEED

When the DISK SPEED program is loaded you will see the following display:

DISK SPEED

OPTIONS
DRIVE - 1

DISK SPEED MENU

1 DISPLAY DISK SPEED
2 WRITE SECTOR(S)
3 CHANGE OPTIONS
4 RETURN TO MAIN MENU

ENTER NUMBER (1-4)

This program is used to verify your disk drive speed or to selectively write bad sectors onto a disk. Each selection in the DISK SPEED MENU is explained below.

1. DISPLAY DISK SPEED

Typing 1 [RETURN] selects the DISPLAY DISK SPEED display with the following prompt:

INSERT FORMATTED DISK IN DRIVE X
PRESS RETURN TO CONTINUE

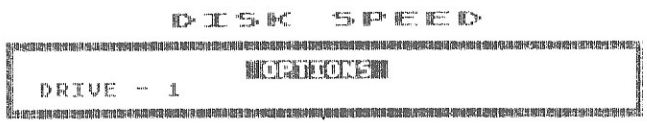
NOTE

If you attempt to display the disk speed with no disk or a non-formatted disk installed, the program will ignore this error and will continue trying to read the speed of the disk drive. Press the [ESC] button if this error is made.

****

Insert any formatted disk into the appropriate disk drive as indicated by the display prompt. Press [RETURN].

After a few seconds, the screen will display the disk drive RPM as shown in the following example:



The program will update the display every few seconds with the latest value of disk RPM. If the RPM of the disk is less than 100 RPM or greater than 368 RPM, the RPM display may be erratic. Refer to the next section of this manual (WRITE SECTORS) for information on how to adjust the ATARI 810 disk drive speed.

The correct speed for an 810 Drive is between 285 and 290 with nominal being 287. Percom Drives run at approximately 295 RPM.

Pressing the [ESC] button terminates the speed readout display and returns you to the DISK SPEED MENU.

2. WRITE SECTORS

This section of the DISK SPEED program is used to create bad sectors on a disk. To create a bad sector(s) you must first slow down the speed of the ATARI 810 disk drive. We recommend you do this only with the ATARI 810 disk drive.

Bad sectors can be written on a disk by following the step by step instructions given below:

---WARNING---

Before you attempt to adjust your drive, we must caution that the operation described here may void any warranty you have for your ATARI drive. Even if the drive is out of warranty, ATARI does not recommend that users attempt to adjust the speed of their drives.

Neither the authors, nor C.A.P. Software, Inc. can assume any responsibility for damage caused to your drive while attempting to make a speed adjustment. We do know that hundreds of ATARI owners are already adjusting the speed of their drives with no negative results.

--*-*-*-*-*

i. Boot-up DISK WIZARD II (refer to section V of the manual) and select the DISK SPEED program. Press START.

ii. When the DISK SPEED MENU is displayed, type 1 [RETURN] to DISPLAY DISK SPEED. Remove the DISK WIZARD II disk from the disk drive.

iii. Be sure you have a clean working environment so that dust, hair, etc. will not get into the disk drive.

iv. You will need a pen knife, a small to medium size phillips head screwdriver and a small slot screwdriver.

v. Using the pen knife or similar instrument, lift off the four plastic stick-on screw hole covers on the top of the drive.

vi. Using the phillips screwdriver, remove the four screws that secure the drive cover to the base of the drive.

vii. Carefully lift the cover off the drive and set it aside.

viii. With the drive facing you, locate the drive speed potentiometer. We have found two different sizes and locations of the speed potentiometer. One type is a white or blue nylon wheel potentiometer with a slot in it. It is located in the back of the drive to the left side. The other type is a small green rectangular potentiometer with a very small slotted screw located about 2" from the back of the unit and just left of the center.

ix. Turn on the disk drive and insert any formatted disk, as indicated by the display prompt. Press [RETURN].

x. The disk drive will now spin up and display the disk speed (should be about 285-290 RPM).

xi. If you have the large nylon wheel type potentiometer turn it clockwise to slow the drive down. Just turn it a small amount and wait for the RPM display to update, to insure you are indeed slowing it down. If you have the small rectangular type potentiometer turn it counterclockwise to slow the drive down. It will take several turns to make a difference. In either case watch the RPM readout to verify that you are slowing it down (The direction of rotation may vary depending on the particular drive that you are using).

xii. Continue turning the potentiometer in the appropriate direction to slow the drive further until you hear the familiar "SNARK" sound or until your disk speed is below 223 RPM.

NOTE

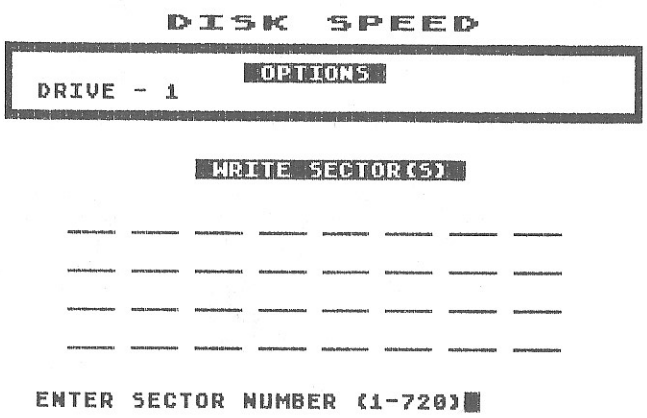
Because several changes have been made to the Atari 810 disk drive over the years, you may not be able to adjust your particular drive to a low enough speed to write bad sectors. If this is the case then you may wish to purchase one of the hardware kits available from other sources to modify your Atari 810 Disk drive for slow speed operation.

****

xiii. Now increase the speed (turn potentiometer in the other direction) just enough to stop the "SNARK" sound. This will take a little experimentation, but once you've got it, it's easy.

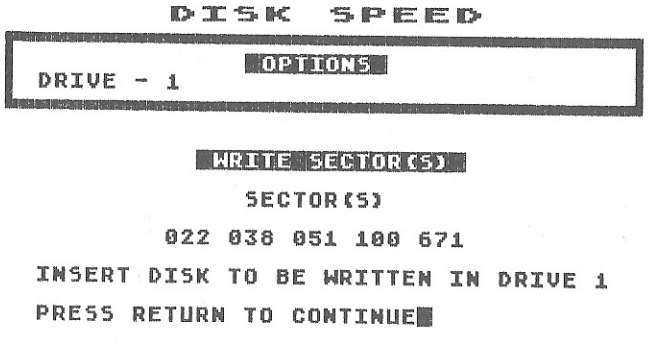
xiv. You are now ready to write bad sectors. Press [ESC] to terminate the DISK SPEED DISPLAY. Remove the formatted disk used to measure speed from the disk drive.

xv. Type 2 [RETURN] to display the WRITE SECTOR DISPLAY as shown below:



xvi. You may now enter up to 32 sector numbers to be written on the disk as bad sectors. As you may remember, the DISK BACKUP program produces a MAP of the bad sectors found on the original program disk. These are the bad sectors which must be recopied onto the backup disk to make it an exact copy of the original.

After you have entered the last sector number press [RETURN]. The following display now appears:



xvii. As indicated by the display prompt, insert the disk into the drive you wish to write the bad sectors onto (write protect tab removed). Press [RETURN]. The bad sectors will now be written on the disk.

xviii. After the bad sectors are written, return to the DISK SPEED DISPLAY and display the speed of the drive. Re-adjust the speed of the potentiometer to obtain the normal drive speed of 285-290 RPM.

You have now created a backup disk with bad sectors that should run on your computer. If you have any problems running the backup copy then refer to section XII (DISK PROTECTION METHODS) of this manual.

3. CHANGE OPTIONS

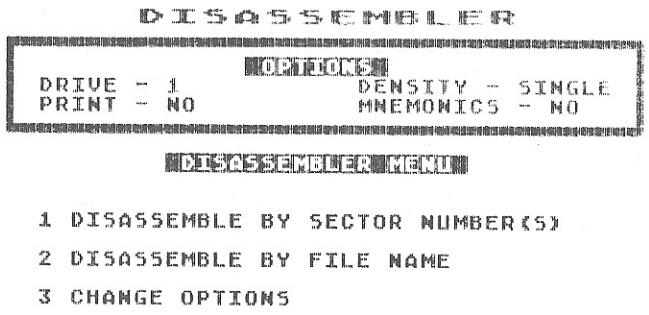
Typing 3 [RETURN] allows you to change the options in the OPTION DISPLAY. Enter the number of the disk drive that you wish to verify the speed of, or to be used to write the bad sectors onto the backup disk.

4. RETURN TO MAIN MENU

TYPING 4 [RETURN] returns you to the DISK WIZARD II MAIN MENU where you may re-select the program you wish to run.

IX. DISASSEMBLER

When the DISASSEMBLER program is selected, you will see the following display:



This program is used to disassemble machine language from disk by file name or sector number(s). These sections of the manual explain how to use the DISASSEMBLER program on your DISK WIZARD II disk. It is beyond the scope of this manual to explain 6502 Assembly Language programming. Refer to any of the books on the market dealing with 6502 Assembly Language

Each selection in the menu will be described below including the options seen in the option table at the top of the DISASSEMBLER MENU.

NOTE

Typing [ESC] any time during this program will return you to the DISASSEMBLER MENU. If any problems are encountered during keyboard entries please refer to the KEYBOARD PROTOCOL section of this manual.

****

1. DISASSEMBLE BY SECTOR NUMBER

Typing 1 [RETURN] allows you to disassemble a disk by sector number. You may disassemble as many sectors as you wish, from 1 to 720. If just [RETURN] is pressed in

response to any prompt (except the printout title), the program will return to the DISASSEMBLER MENU. The first prompt to be displayed is:

ENTER START SECTOR (1-720)

This is the sector number you wish to start the disassembler listing from. Enter the number then press [RETURN]. After the start sector number has been entered you will see the prompt:

ENTER END SECTOR (X-720)

(where X is the start sector you just entered). This is the last sector which will be disassembled. Enter the number then press [RETURN]. If, for example, you wanted to disassemble only one sector, say sector 1, enter the START SECTOR equal to 1 and END SECTOR equal to 1. After the START and END SECTOR has been entered, the following is displayed:

ENTER NUMBER OF BYTES/SECTOR (125/128)

Enter the number of bytes per sector for the sectors that you wish to disassemble. Enter 125 if you are working with sectors from a DOS file or 128 for non-DOS files. Press [RETURN]. Refer to the TECHNICAL INFORMATION section of this manual for more detailed information on the number of bytes contained in a sector.

After the BYTES/SECTOR has been entered the following prompt is displayed:

ENTER ORIGIN (0-FFFF)

This is the assumed start address of the code that is being disassembled. How is this determined? If it is a DOS file, the first 6 bytes of the file usually contain this information in the following format (numbers used are for example only):

FFFF-BINARY FILE INDENT
0007-START ADDRESS -0700 (lo,hi)
8507-END ADDRESS-0785 (lo,hi)

This header tells us that the origin for the disassembly should be 0700. However, the six byte header must be discarded because it isn't part of the actual code. To do this, just enter a START BYTE of 6 to the next prompt. If you are disassembling the entire file, use the DISASSEMBLE BY FILE NAME option. Refer to the next section of this manual for further information on binary load files.

If you're working with a non-DOS format disk, you're pretty much on your own. You'll have to disassemble the boot code on sector 1 on up and figure out where the

program is loaded. The first six bytes on sector 1 contain the following information:

flags	byte 1
# of sectors	byte 2
memory address to start load	byte 3 byte 4
init address	byte 5 byte 6

Byte 1 - usually equals zero.

Byte 2 - contains the number of disk sectors to be read as part of the boot process.

Bytes 3 & 4 - contain the address (lo,hi) at which to start loading the first byte of sector one.

Bytes 5 & 6 - contain the address (lo,hi) to which the booter will transfer control after the boot process is complete.

After the ORIGIN has been entered, the following prompt is displayed:

ENTER START BYTE (0-7C)

Enter the number of the byte in the Start Sector which contains the first instruction to be disassembled, then press [RETURN].

If the listing is to be printed (PRINT DISASSEMBLY option is YES) the following prompt will also be displayed as shown below:

ENTER TITLE FOR PRINTOUT

Type in the title you want to be printed at the top of each page of the disassembler printout. The title may be any group of alphanumeric characters up to 32 characters in length. Press [RETURN]. If you do not want a title, just press [RETURN].

At this point, the program will read the sector data from disk and begin to display and/or print the disassembled program.

NOTE

If at any time you wish to stop the display of the DISASSEMBLED program, press the space bar . Pressing the space bar again will continue the display listing.

****

The disassembled listing of the sector data will

appear on the display as in the example below (note: a mnemonic disassembly is shown):

DISASSEMBLER

```
ADDR  B1 B2 B3      INSTRUCTION  S 2

0774  A0 40         LDY  #$40
0776  90 04         BCC   $077C
0778  A9 57         LDA  #$57
077A  A0 80         LDY  #$80
077C  8D 02 03      STA   DCOMND
077F  8C 03 03      STY   DSTATS
0782  A9 31         LDA  #$31
0784  A0 0F         LDY  #$0F
0786  8D 00 03      STA   DDEVIC
0789  8C 06 03      STY   DTIMLO
078C  A9 03         LDA  #$03
078E  8D FF 12      STA   $12FF
0791  A9 00         LDA  #$00
0793  A0 80         LDY  #$80
0795  CA            DEX
0796  F0 04         BEQ   $079C
0798  A9 01         LDA  #$01
079A  A0 00         LDY  #$00
079C  8D 09 03      STA   DBYTHI
```

The program address is shown in the left column. The program addresses are not always sequential because of the variable length instructions which may be from one to three bytes long. Each byte requires one memory location.

The beginning of a sector disassembly may contain question marks even though there is valid data. This is due to phase error caused by the variable length of the instructions. The first byte of the first instruction may be on the previous sector thus causing the disassembler to misinterpret the code.

The next three columns contain the actual machine code (object program). Again these instructions may be from one to three bytes in length. B1, B2 and B3 at the top of the screen signify Byte 1, Byte 2 and Byte 3 of the instructions respectively.

The last column of information contains the disassembled machine language (source program). Note that the sector number containing the displayed disassembled code is shown in the upper right-hand corner of the screen.

The DISASSEMBLER program will continue to disassemble the sector data until the last required sector is disassembled (END SECTOR NUMBER). The following message is then displayed:

DISASSEMBLY COMPLETE
PRESS RETURN TO CONTINUE

Pressing [RETURN] will bring you back to the DISASSEMBLER MENU.

2. DISASSEMBLE BY FILE NAME

This option allows automatic disassembly of compound binary files. The following discussion, in conjunction with the DOS Manual, should help you to understand the structure of a binary file.

A binary file consists of 1 to N segments. Each segment is preceeded by an optional(except for the first segment) FF FF pair of bytes followed by the start address of the segment [in lo, hi format] and the end address of the segment. For example:

FIRST SEGMENT

```
BINARY FILE IDENT - FF FF
                    00 07  START ADDRESS=700
                    20 07  END ADDRESS=720
```

[MEANS 21 BYTES; LOAD THESE BYTES AT ADDRESSES 700 TO 720]

SECOND SEGMENT

```
OPTIONAL - FF FF
                    01 20  START ADDRESS=2001
                    34 20  END ADDRESS=2034
```

[MEANS 34 BYTES; LOAD THESE BYTES AT ADDRESSES 2001 TO 2034]

In order to disassemble this type of file with a standard disassembler, you would have to determine the length and start address of each segment and then disassemble it by sector numbers. The DISK WIZARD II DISASSEMBLER, however, will automatically handle each segment and disassemble it to the correct address.

There is one problem, however, in that there may be some extraneous bytes at the end of the file. For example, suppose the last segment is 23 bytes long and that the last three bytes, of this segment, are on the last sector. Ideally, the word count of the last sector would be three. Depending on the method used to generate the binary file, the word count may be greater than three. This causes the disassembler to interpret the suceeding bytes as a start address, end address header. If these bytes are such that the end address is less than the start address, then the disassembly will terminate with an END ADDR < START ADDR error. If the spurious END ADDRESS is greater than or equal to the START ADDRESS, then the disassembly will continue. If an end of file is reached before the segment is finished ,then the disassembly will terminate with an END OF FILE ENCOUNTERED error. Neither case is cause for alarm; simply ignore the invalid segment.

After selecting the DISASSEMBLE BY FILE NAME option, the following prompt will be displayed:

ENTER FILE NAME _____

Type in the name of the file that you wish to disassemble. The file name may be any group of alphanumeric characters up to 12 characters in length, the first character being a letter. The file name must be an exact match including extension, if any, of the file you wish to disassemble (e.g. TEST.ASM).

If the file name is not found in the directory, the computer will "BEEP" and display:

ERROR-FILE NAME NOT IN DIRECTORY

PRESS RETURN TO CONTINUE

Pressing [RETURN] allows you to go back and enter another file name.

If the file is in the directory, the first two bytes are verified to be FF (binary file ident). If they are not FF, then the following is displayed:

ERROR-NOT A BINARY FILE

PRESS RETURN TO CONTINUE

Pressing [RETURN] allows you to go back and enter another file name.

If the listing is to be printed (PRINT DISASSEMBLY option is YES) the following prompt will also be displayed:

ENTER TITLE FOR PRINTOUT

_____

Type in the title you want to be printed at the top of each page of the disassembler printout. The title may be any group of alphanumeric characters up to 32 characters in length. Press [RETURN]

If no title is desired, just press [RETURN].

The format of the disassembler display will be the same as in the previous section, DISASSEMBLE BY SECTOR NUMBER, except that each segment of the file will be preceeded by a START ADDR and END ADDR header.

The DISASSEMBLER program will continue until all the data in the file is disassembled at which point the

following message will appear:

DISASSEMBLY COMPLETE
PRESS RETURN TO CONTINUE

Press [RETURN] to go back to the DISASSEMBLER MENU.

3. CHANGE OPTIONS

Typing 3 [RETURN] allows you to change the options shown in the option table of the DISASSEMBLER program. A prompt for each option will be displayed one at a time. Each prompt must be answered before continuing on. Each option is explained below.

ENTER DRIVE NUMBER (1-4) - Enter the number of the disk drive that will be used to read in the program to be disassembled. Press [RETURN]. If just [RETURN] is pressed, the program will default to the value presently shown in the option table.

ENTER DENSITY (S/D) - Type S if you are disassembling from a single density disk, D if you are doing the same from a double density disk. Press [RETURN]. If just [RETURN] is pressed, the program will default to the value presently shown in the option table.

PRINT DISASSEMBLY? (Y/N) - Type Y [RETURN] to print the disassembled listing on your printer. If N [RETURN] is typed the listing will not be printed. If just [RETURN] is pressed, the program will default to the value presently shown in the option table.

MNEMONIC DISASSEMBLY? (Y/N) - Type Y [RETURN] if you wish to have the ATARI memory location mnemonics included in the disassembled listing. This is an extremely useful feature of your DISK WIZARD II DISASSEMBLER. If, for example, a machine instruction loaded a 0 into memory location 82, the disassembled listing with mnemonics would display LMRGIN for the memory location in the object code column instead of just 82. APPENDIX C contains a list of all the mneumonics and their equivalent address. If N [RETURN] is typed no mnemonics will be included in the disassembled listing. If just [RETURN] is pressed, the program will default to the value presently shown in the option table.

After the last option question is answered, the program will return to the DISASSEMBLER MENU.

4. RETURN TO MAIN MENU

Typing 4 [RETURN] returns you to the DISK WIZARD MAIN MENU where you may re-select the program you wish to run.

X. TECHNICAL INFORMATION

This section provides the user with some basic information pertaining to disk layout and DOS usage of the disk.

There are a total of 720 sectors and 40 tracks on a diskette. A sector is the basic unit of storage on the disk drive. It contains 128 bytes of data (256 in double density). There are 18 sectors per track. A track is simply a complete circle on the disk, thus the disk is laid out in 40 concentric circles; each of which is divided into 18 parts. When a diskette is formatted, the drive positions the read/write head over each track and writes an identification pattern and a known data pattern onto each sector. From that point on, reading and writing to the disk drive consists of read/write commands along with a sector number.

At this level of communication to the disk drive, it would be very cumbersome to save and read data to the disk. We would have to keep track of which sectors we wrote to and then reread them to retrieve the data from the disk at a later date. It would be nice to have a program to do all the "bookkeeping" for us. This is the prime function of a Disk Operating System (DOS). By using DOS, we can save to and read from the disk by filename and not have to worry about specific sector allocation. In order to perform it's task, DOS requires some disk space to keep track of sector usage and file start locations. Sectors 1 - 3 are used for the initial bootstrap from disk. Sector 360 is the Volume Table of Contents (VTOC). This is simply a map of 720 bits (representing 720 sectors); if the bit is reset, the sector is in use. If the bit is set, the sector is available. Sectors 361 - 368 are used to store the directory information for each file. Each directory sector can contain information for eight files (thus eight directory sectors give us a capacity of 64 files). The directory entry for a file contains the filename, the sector that the file starts on, the total number of sectors in the file, and a status byte that indicates whether the file is locked, in use, or deleted.

Since the directory entry only tells us where the first sector in the file is located, we need to be able to find the rest of the sectors in the file. DOS does this by reserving the last three bytes in each sector for its own use. These bytes are called the file links and contain a file number (equivalent to its position in the directory - i.e.; entry 6 would have a file number of 6), a next sector pointer which points to the next sector in the file, and a byte count value which indicates how many bytes in the sector are used. The end of the file is indicated by a next sector value of zero.

Let's briefly examine how DOS would read, write, or delete a file:

READ

First, the directory is searched for the file name. If the file name is found, the file number is set equal to the file's position in the directory. The first sector in the file is the start sector number in the directory. This sector is read in and the file number in the link bytes of the sector is checked against the original file number. If the file numbers don't match, then an error is reported. If the next sector number in the link bytes isn't zero, then that sector is read into memory at an address equal to the address of where the previous sector was read in plus the word count of the previous sector (obtained from the file links). This process is repeated until a next sector value of zero is encountered.

WRITE

If the file already exists, it is deleted (see DELETE writeup). The file is then written to the disk. The sector numbers to use are determined by searching the VTOC for sectors that are unused. The bits corresponding to the sectors that were used to write the file are then reset in the VTOC to indicate they are now in use.

DELETE

The file name is located in the directory and the status byte is checked to see if the file is locked. If it is, an error is reported. If the file isn't locked, it is then read and every sector that it occupies is set in the VTOC to indicate that the sector is available for use. Finallly the delete bit is set in the status byte in the file's directory entry.

USING BOTH SIDES OF A SINGLE SIDED DISK

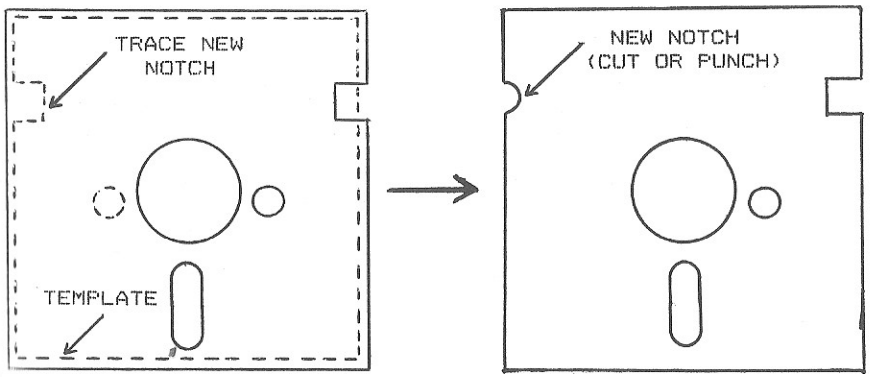
If you would like to double the storage capacity of your single sided disks follow the directions below. There is no need to spend the extra money on double sided disks because with a little modification, you can use both sides of a single sided disk.

You will need the following items to double side your disks.

1. A pencil
2. A disk to use as a template

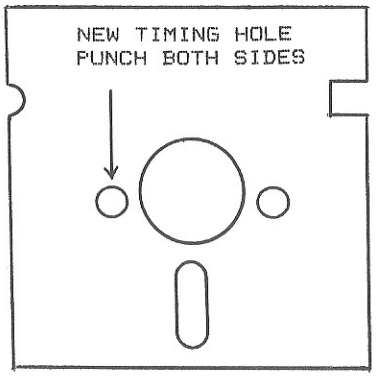
3. A hole punch

If you have an Atari 810 Disk Drive all you need do is punch out another write protect notch on the other side of the disk sleeve. Refer to the illustration below:



Take the disk you are using as a template and lay it on top of the disk you are going to double side with the notch positioned as shown in the illustration. Using the pencil, trace the outline of the notch onto the bottom disk. Now using the paper punch, punch a hole in the disk where you marked it. Insert the punch in far enough to punch a half circle. That's all there is to it, you can now use the second side to store your program.

Other drives (including Percom) require that you punch one other hole. These drives use the small timing hole located near the large center hole. Using the hole punch and a lot of care punch a hole in BOTH sides of the sleeve as shown below:



Inserting a small piece of paper between the sleeve and the disk surface will help protect the recording surface. Your disk will now work on both sides.

IMPORTANT NOTE

If you have REV A of the operating system in your computer (generally can be found in computers built before Oct.1980), you will notice that the two color display of Disk Wizard II will occasionally go all red or all grey when you do a printout or read/write to the disk. This is due to a bug in the old operating system that disabled the Display List Interrupt during SIO. This does not affect the actual operation of the program. You can have your computer updated at an authorized ATARI service center if you so desire.

XI. BACKUP PHILOSOPHY

The DISK BACKUP program on DISK WIZARD II is solely for the purpose of making a backup copy of your original disk for your own use. This backup copy is necessary because the original disk may eventually wear out or can be accidentally damaged.

In all fairness to the software manufacturers and dealers we urge you not to use this program for anything but the stated purpose.

XII. DISK PROTECTION METHODS

You will notice on some of the program disks that you make a backup copy of; the program will load but will not run. This is due to the fact that the programmer of the original disk installed a protection device so that any duplicate copy of the original will not run.

Even though the Back-up copy does not run, it may be used to recover a partially damaged original disk. When you made a backup of the disk using DISK WIZARD II, a disk map was produced showing the location of all bad sectors. If your original copy eventually fails to run, you may remap it to determine which previously good sectors are now bad. The backup copy can then be used to restore these sectors on the original.

****

There are three commonly used protection methods for software, on the market today. These methods will be explained below.

First some background information on disk structure. When an 810 disk drive formats a diskette, it defines 18 sectors per track (40 tracks total). A sector consists of timing and identification marks, 128 data bytes, and Cyclic Redundancy Check data (CRC - similar in concept to a checksum)

In simple terms, once a diskette has been formatted we can access a particular sector by telling the disk drive the sector number (1-720) and the command (read or write sector). The disk drive then:

1. Determines which track the sector is located on (based on 18 sectors per track).

2. Positions the drive head to that track.

3. Reads the identification data in the sectors on that track until it finds the desired sector.

4. Performs the requested command (read/write sector).

Lets examine what can go wrong in the above scenario.

1. If the drive positions the head to the track and then can't find a sector with the correct number, it assumes the head is mispositioned over the incorrect

track. The head is stepped back to the home position and repositioned over the track (the "snark" noise you hear when reading "bad" sectors). If the second attempt to locate the sector fails, the drive returns a bad sector status to the computer.

2. Suppose the sector is located but the CRC check fails (indicates that the sector data is bad - caused by a bad spot on the disk). In this case the drive will return a bad status to the computer but no "snark" will occur because the sector was located on the diskette. This type of failure can be intentionally created by writing bad CRC data on the diskette (not possible with the 810).

Now lets look at the three common protection methods and how to recognize them.

1. BAD SECTOR

Recognizable by the fact that the original diskette causes the drive to "snark" during boot. The original diskette has a special formatting that results in some sectors not being created. A read of these non existant "bad" sectors causes the drive to "snark" and return a bad status to the computer. The original program expects the bad status when it accesses these sectors. This type of disk can be backed up by DISK WIZARD II. Bad sectors are simulated by writing them at a reduced disk speed. When the drive is returned to normal speed, the slow speed data on the diskette can't be read; thus simulating a non existant sector.

2. DOUBLE SECTORING

Recognizable by seeing one bad sector per track during DISK BACKUP (usually the last sector per track - 18, 36, 54, etc.). Also, the original disk will NOT cause the drive to "snark" during boot. In double sectoring, the diskette is formatted such that two sectors on a track contain the same number (usually the next to last sector on the track - two sector 17's, two 35's, etc.). The original program upon booting reads the two sector 17's and never reads the non existant sector 18. When you attempt to back up this diskette, a read of sector 18 causes the drive to "snark" because the drive can't find it. The drive returns a bad status to the computer and DISK BACKUP displays the sector as bad. Obviously, writing a bad sector 18 on the backup diskette isn't going to work because the program is looking for a second sector 17. You can't write a second sector 17 because the 810 can't format a diskette in such a fashion.

3. CUSTOM FORMAT

Recognizable by finding all the bad sectors on one track and that track also contains some readable sectors. Also, the drive does not snark during boot of the original program. As in double sectoring, the "bad" sectors don't exist on the diskette. The good sectors on the track in question are arranged in non standard order and there will be several copies of a particular sector on the track. When the original program boots in, it accesses the multiple copy sector several times and determines how much time it takes. Since we don't have to wait for the diskette to make a complete revolution each time the sector is read (as we would on a normal disk), the elapsed time will be shorter than it would for a normal diskette. A backup of this type of disk won't work because the time test in the program will fail.

Some diskettes may be protected by a combination of the above methods.

In summary, you can only backup diskettes protected by method one. To backup the other two types requires either drive modification (to allow non standard formatting) or modification of the original program to bypass the protection checks (requires extensive assembly language experience and a lot of time).

XIII. LIMITED WARRANTY AND DISCLAIMER

C.A.P. SOFTWARE, Inc. warrants to the original consumer/ purchaser that this C.A.P. SOFTWARE program diskette shall be free from defects in material or workmanship for a period of 90 days from the date of purchase. If a defect is found during this warranty period, C.A.P. SOFTWARE, Inc. will replace the diskette, provided the diskette and proof of purchase is delivered or mailed, postage prepaid, to C.A.P. SOFTWARE, Inc.

This warranty shall not apply if the diskette has been misused, shows signs of excessive wear, or has been partially or totally erased or damaged in any other way by the purchaser. Any applicable implied warranties, including warranties of merchantability and fitness, are limited to 90 days from the date of purchase. Consequential or incidental damages resulting from a breach of any applicable, express or implied warranties are hereby excluded.

C.A.P. Software, Inc. shall have no liability or resposibility to the original consumer/purchaser or any other person or entity with respect to any claim, loss, liability or damage caused directly or indirectly by computer programs sold by C.A.P. Software, Inc. This disclaimer includes, but is not limited to, any interruption of services, loss of business or anticipatory profits, and/or consequential damages resultin from the purchase, use or operation of C.A.P. Software computer programs.

Every effort has been made to ensure that this manual accurately documents this product. However, because of ongoing improvements and updating of computer software and hardware, C.A.P. Software cannot guarantee the accuracy of printed material after the date of publication and shall not accept responsibility for errors or emissions.

APPENDIX A

DISK WIZARD II BACKUP PROCEDURE

To make a backup copy of DISK WIZARD II just follow these step by step instructions:

1. Ensure that there are no cartridges installed in your computer.

2. Turn Disk Drive 1 power on. When the busy light goes out, open the disk drive door and insert the DISK WIZARD II disk.

3. Turn on your computer and TV screen and wait until the DISK WIZARD II MAIN MENU appears.

4. Press the console START button and observe that the DISK BACKUP MENU is displayed.

5. Enter 1 [RETURN].

6. The display will read INSERT SOURCE DISK IN DRIVE 1, PRESS RETURN.

7. Since the source disk is already installed, just press [RETURN].

8. DISK BACKUP will commence reading the disk data into memory. The sector number currently being read will be displayed on the screen.

9. When the disk read is complete the prompt: INSERT BACKUP DISK, PRESS RETURN will appear.

10. Remove the DISK WIZARD II disk and insert a blank disk (with write protect tab removed). Press [RETURN].

11. FORMATTING BACKUP DISK message will appear. After a few moments, writing of the disk data will begin.

12. After the disk write is complete the SECTOR COUNTS display will appear on the screen.

The backup process is now complete. Remove the backup disk and install a write protect tab on it. You now have a backup copy of DISK WIZARD II in case the original is lost or damaged.

APPENDIX B

CREATING THE TEST DISK

The following step by step instructions create a test disk which will be used throughout the manual for various examples. The DOS files are written on the disk followed by four test files. One of the files will then be deleted.

1. Remove the basic cartridge from your computer and install any disk with DOS on it. Turn on your computer and wait until you see the DOS Menu.

2. Remove the disk containing DOS from your disk drive and install any blank disk (write protect tab removed) into drive 1, this will be your test disk.

3. Type I [RETURN]. Type 1 [RETURN] in response to the "WHICH DRIVE TO FORMAT?" question. Now type Y [RETURN] to format the test disk.

4. After the format operation is complete, type H [RETURN]. Type 1 [RETURN] in response to the "DRIVE TO WRITE DOS FILES TO?" question. Type Y [RETURN] to write the DOS files.

5. After the DOS files write is complete, type K [RETURN].

6. Type TEST1,2000,3000 then press [RETURN].

7. After the write operation is complete, type K [RETURN] again.

8. This time type TEST2,2000,3000 then press [RETURN].

9. Repeat steps 7 and 8 for the following entries:

TEST3,2000,3000
TEST4,2000,3000

10. After all the files have been saved, type A [RETURN], then press [RETURN] again. The directory of the test disk should look as follows:

DOS	SYS	039
DUP	SYS	042
TEST1		033
TEST2		033
TEST3		033
TEST4		033

11. The TEST2 file will now be deleted. Type D [RETURN]. Type TEST2 then press [RETURN]. Type Y [RETURN] in response to the "D1:TEST2.?" question. The test disk for DISK WIZARD II is now complete.

APPENDIX C

This appendix contains a list of all the ATARI memory mnemonics included in the disassembler. Along with each mnemonic name is it's equivalent address and length (number of bytes). An explanation of the meaning and use of these locations is beyond the scope of this manual. This information can be obtained in any of the ATARI Memory Map books that are available on the market.

NAME	ADDR	LENGTH
CASINI	$0002	2
RAMLO	$0004	2
TRAMSZ	$0006	1
TSTDAT	$0007	1
WARMST	$0008	1
BOOT	$0009	1
DOSVEC	$000A	2
DOSINI	$000C	2
APPMHI	$000E	2
POKMSK	$0010	1
BRKKEY	$0011	1
RTCLOK	$0012	1
RTCLOK	$0013	1
RTCLOK	$0014	1
BUFADR	$0015	2
ICCOMT	$0017	1
DSKFMS	$0018	2
DSKUTL	$001A	2
PTIMOT	$001C	1
PBPNT	$001D	1
PBUFSZ	$001E	1
PTEMP	$001F	1
ICHIDZ	$0020	1
ICDNOZ	$0021	1
ICCOMZ	$0022	1
ICSTAZ	$0023	1
ICBALZ	$0024	1
ICBAHZ	$0025	1
ICPTLZ	$0026	1
ICPTHZ	$0027	1
ICBLLZ	$0028	1
ICBLHZ	$0029	1
ICAX1Z	$002A	1
ICAX2Z	$002B	1
ICAX3Z	$002C	1
ICAX4Z	$002D	1
ICAX5Z	$002E	1
ICAX6Z	$002F	1
STATUS	$0030	1

NAME	ADDR	LENGTH
CHKSUM	$0031	1
BUFRLO	$0032	1
BUFRHI	$0033	1
BFENLO	$0034	1
BFENHI	$0035	1
CRETRY	$0036	1
DRETRY	$0037	1
BUFRFL	$0038	1
RECVDN	$0039	1
XMTDON	$003A	1
CHKSNT	$003B	1
NOCKSM	$003C	1
BPTR	$003D	1
FTYPE	$003E	1
FEOF	$003F	1
FREQ	$0040	1
SOUNDR	$0041	1
CRITIC	$0042	1
ZBUFP	$0043	2
ZDRVA	$0045	2
ZSBA	$0047	2
ERRNO	$0049	1
CKEY	$004A	1
CASSBT	$004B	1
DSTAT	$004C	1
ATRACT	$004D	1
DRKMSK	$004E	1
COLRSH	$004F	1
TMPCHR	$0050	1
HOLD1	$0051	1
LMARGN	$0052	1
RMARGN	$0053	1
ROWCRS	$0054	1
COLCRS	$0055	2
DINDEX	$0057	1
SAVMSC	$0058	2
OLDROW	$005A	1
OLDCOL	$005B	2
OLDCHR	$005D	1

NAME	ADDR	LENGTH
OLDADR	$005E	2
NEWROW	$0060	1
NEWCOL	$0061	2
LOGCOL	$0063	1
ADRESS	$0064	2
MLTTMP	$0066	2
SAVADR	$0068	2
RAMTOP	$006A	1
BUFCNT	$006B	1
BUFSTR	$006C	2
BITMSK	$006E	1
SHFAMT	$006F	1
ROWAC	$0070	2
COLAC	$0072	2
ENDPT	$0074	2
DELTAR	$0076	1
DELTAC	$0077	2
ROWINC	$0079	1
COLINC	$007A	1
SWPFLG	$007B	1
HOLDCH	$007C	1
INSDAT	$007D	1
COUNTR	$007E	2
LOMEM	$0080	2
VNTP	$0082	2
VNTD	$0084	2
VVTP	$0086	2
STMTAB	$0088	2
STMCUR	$008A	2
STARP	$008C	2
RUNSTK	$008E	2
MEMTOP	$0090	2
STOPLN	$00BA	2
ERRSAV	$00C3	1
FR0	$00D4	1
FR0M	$00D5	5
FRE	$00DA	6
FR1	$00E0	1
FR1M	$00E1	5
FR2	$00E6	6
FRX	$00EC	1
CIX	$00F2	1
INBUFF	$00F3	2
RADFLG	$00FB	1
FLPTR	$00FC	2
FPTR2	$00FE	2
VDSLST	$0200	2
VPRCED	$0202	2
VINTER	$0204	2
VBREAK	$0206	2
VKEYBD	$0208	2
VSERIN	$020A	2

NAME	ADDR	LENGTH
VSEROR	$020C	2
VSEROC	$020E	2
VTIMR1	$0210	2
VTIMR2	$0212	2
VTIMR3	$0214	2
VIMIRQ	$0216	2
CDTMV1	$0218	2
CDTMV2	$021A	2
CDTMV3	$021C	2
CDTMV4	$021E	2
CDTMV5	$0220	2
VVBLKI	$0222	2
VVBLKD	$0224	2
CDTMA1	$0226	2
CDTMA2	$0228	2
CDTMF3	$022A	1
SRTIMR	$022B	1
CDTMF4	$022C	1
INTEMP	$022D	1
CDTMF5	$022E	1
SDMCTL	$022F	1
SDLSTL	$0230	1
SDLSTH	$0231	1
SSKCTL	$0232	1
LPENH	$0234	1
LPENV	$0235	1
CDEVIC	$023A	1
CCOMND	$023B	1
CAUX1	$023C	1
CAUX2	$023D	1
TEMP	$023E	1
ERRFLG	$023F	1
DFLAGS	$0240	1
DBSECT	$0241	1
BOOTAD	$0242	2
COLDST	$0244	1
DSKTIM	$0246	1
LINBUF	$0247	40
GPRIOR	$026F	1
PADDL0	$0270	1
PADDL1	$0271	1
PADDL2	$0272	1
PADDL3	$0273	1
PADDL4	$0274	1
PADDL5	$0275	1
PADDL6	$0276	1
PADDL7	$0277	1
STICK0	$0278	1
STICK1	$0279	1
STICK2	$027A	1
STICK3	$027B	1
PTRIG0	$027C	1

NAME	ADDR	LENGTH
PTRIG1	$027D	1
PTRIG2	$027E	1
PTRIG3	$027F	1
PTRIG4	$0280	1
PTRIG5	$0281	1
PTRIG6	$0282	1
PTRIG7	$0283	1
STRIG0	$0284	1
STRIG1	$0285	1
STRIG2	$0286	1
STRIG3	$0287	1
CSTAT	$0288	1
WMODE	$0289	1
BLIM	$028A	1
TXTROW	$0290	1
TXTCOL	$0291	2
TINDEX	$0293	1
TXTMSC	$0294	2
TXTOLD	$0296	6
TMPX1	$029C	1
HOLD3	$029D	1
SUBTMP	$029E	1
HOLD2	$029F	1
DMASK	$02A0	1
TMPLBT	$02A1	1
ESCFLG	$02A2	1
TABMAP	$02A3	15
LOGMAP	$02B2	4
INVFLG	$02B6	1
FILFLG	$02B7	1
TMPROW	$02B8	1
TMPCOL	$02B9	2
SCRFLG	$02BB	1
HOLD4	$02BC	1
HOLD5	$02BD	1
SHFLOK	$02BE	1
BOTSCR	$02BF	1
PCOLR0	$02C0	1
PCOLR1	$02C1	1
PCOLR2	$02C2	1
PCOLR3	$02C3	1
COLOR0	$02C4	1
COLOR1	$02C5	1
COLOR2	$02C6	1
COLOR3	$02C7	1
COLOR4	$02C8	1
RAMSIZ	$02E4	1
MEMTOP	$02E5	2
MEMLO	$02E7	2
DVSTAT	$02EA	4
CBAUDL	$02EE	1
CBAUDH	$02EF	1

NAME	ADDR	LENGTH
CRSINH	$02F0	1
KEYDEL	$02F1	1
CH1	$02F2	1
CHACT	$02F3	1
CHBAS	$02F4	1
CHAR	$02FA	1
ATACHR	$02FB	1
CH	$02FC	1
FILDAT	$02FD	1
DSPFLG	$02FE	1
SSFLAG	$02FF	1
DDEVIC	$0300	1
DUNIT	$0301	1
DCOMND	$0302	1
DSTATS	$0303	1
DBUFLO	$0304	1
DBUFHI	$0305	1
DTIMLO	$0306	1
DBYTLO	$0308	1
DBYTHI	$0309	1
DAUX1	$030A	1
DAUX2	$030B	1
TIMER1	$030C	2
ADDCOR	$030E	1
CASFLG	$030F	1
TIMER2	$0310	2
TEMP1	$0312	2
TEMP2	$0314	1
TEMP3	$0315	1
SAVIO	$0316	1
TIMFLG	$0317	1
STACKP	$0318	1
TSTAT	$0319	1
HATABS	$031A	38
ICHID0	$0340	1
ICDNO0	$0341	1
ICCOM0	$0342	1
ICSTA0	$0343	1
ICBAL0	$0344	1
ICBAH0	$0345	1
ICPTL0	$0346	1
ICPTH0	$0347	1
ICBLL0	$0348	1
ICBLH0	$0349	1
ICAX10	$034A	1
ICAX20	$034B	1
ICAX30	$034C	1
ICAX40	$034D	1
ICAX50	$034E	1
ICAX60	$034F	1
ICHID1	$0350	1
ICDNO1	$0351	1

NAME	ADDR	LENGTH
ICCOM1	$0352	1
ICSTA1	$0353	1
ICBAL1	$0354	1
ICBAH1	$0355	1
ICPTL1	$0356	1
ICPTH1	$0357	1
ICBLL1	$0358	1
ICBLH1	$0359	1
ICAX11	$035A	1
ICAX21	$035B	1
ICAX31	$035C	1
ICAX41	$035D	1
ICAX51	$035E	1
ICAX61	$035F	1
ICHID2	$0360	1
ICDNO2	$0361	1
ICCOM2	$0362	1
ICSTA2	$0363	1
ICBAL2	$0364	1
ICBAH2	$0365	1
ICPTL2	$0366	1
ICPTH2	$0367	1
ICBLL2	$0368	1
ICBLH2	$0369	1
ICAX12	$036A	1
ICAX22	$036B	1
ICAX32	$036C	1
ICAX42	$036D	1
ICAX52	$036E	1
ICAX62	$036F	1
ICHID3	$0370	1
ICDNO3	$0371	1
ICCOM3	$0372	1
ICSTA3	$0373	1
ICBAL3	$0374	1
ICBAH3	$0375	1
ICPTL3	$0376	1
ICPTH3	$0377	1
ICBLL3	$0378	1
ICBLH3	$0379	1
ICAX13	$037A	1
ICAX23	$037B	1
ICAX33	$037C	1
ICAX43	$037D	1
ICAX53	$037E	1
ICAX63	$037F	1
ICHID4	$0380	1
ICDNO4	$0381	1
ICCOM4	$0382	1
ICSTA4	$0383	1
ICBAL4	$0384	1
ICBAH4	$0385	1

NAME	ADDR	LENGTH
ICPTL4	$0386	1
ICPTH4	$0387	1
ICBLL4	$0388	1
ICBLH4	$0389	1
ICAX14	$038A	1
ICAX24	$038B	1
ICAX34	$038C	1
ICAX44	$038D	1
ICAX54	$038E	1
ICAX64	$038F	1
ICHID5	$0390	1
ICDNO5	$0391	1
ICCOM5	$0392	1
ICSTA5	$0393	1
ICBAL5	$0394	1
ICBAH5	$0395	1
ICPTL5	$0396	1
ICPTH5	$0397	1
ICBLL5	$0398	1
ICBLH5	$0399	1
ICAX15	$039A	1
ICAX25	$039B	1
ICAX35	$039C	1
ICAX45	$039D	1
ICAX55	$039E	1
ICAX65	$039F	1
ICHID6	$03A0	1
ICDNO6	$03A1	1
ICCOM6	$03A2	1
ICSTA6	$03A3	1
ICBAL6	$03A4	1
ICBAH6	$03A5	1
ICPTL6	$03A6	1
ICPTH6	$03A7	1
ICBLL6	$03A8	1
ICBLH6	$03A9	1
ICAX16	$03AA	1
ICAX26	$03AB	1
ICAX36	$03AC	1
ICAX46	$03AD	1
ICAX56	$03AE	1
ICAX66	$03AF	1
ICHID7	$03B0	1
ICDNO7	$03B1	1
ICCOM7	$03B2	1
ICSTA7	$03B3	1
ICBAL7	$03B4	1
ICBAH7	$03B5	1
ICPTL7	$03B6	1
ICPTH7	$03B7	1
ICBLL7	$03B8	1
ICBLH7	$03B9	1

NAME	ADDR	LENGTH
ICAX17	$03BA	1
ICAX27	$03BB	1
ICAX37	$03BC	1
ICAX47	$03BD	1
ICAX57	$03BE	1
ICAX67	$03BF	1
PRNBUF	$03C0	40
CASBUF	$03FD	131
LBPR1	$057E	1
LBPR2	$057F	1
LBUFF	$0580	128
CARTCS	$BFFA	2
CART	$BFFC	1
CARTFG	$BFFD	1
CARTAD	$BFFE	2
HPOSP0	$D000	1
HPOSP1	$D001	1
HPOSP2	$D002	1
HPOSP3	$D003	1
HPOSM0	$D004	1
HPOSM1	$D005	1
HPOSM2	$D006	1
HPOSM3	$D007	1
SIZEP0	$D008	1
SIZEP1	$D009	1
SIZEP2	$D00A	1
SIZEP3	$D00B	1
SIZEM	$D00C	1
GRAFP0	$D00D	1
GRAFP1	$D00E	1
GRAFP2	$D00F	1
GRAFP3	$D010	1
GRAFM	$D011	1
COLPM0	$D012	1
COLPM1	$D013	1
COLPM2	$D014	1
COLPM3	$D015	1
COLPF0	$D016	1
COLPF1	$D017	1
COLPF2	$D018	1
COLPF3	$D019	1
COLBK	$D01A	1
PRIOR	$D01B	1
VDELAY	$D01C	1
GRACTL	$D01D	1
HITCLR	$D01E	1
CONSOL	$D01F	1
POT0	$D200	1
POT1	$D201	1
POT2	$D202	1
POT3	$D203	1

NAME	ADDR	LENGTH
POT4	$D204	1
POT5	$D205	1
POT6	$D206	1
POT7	$D207	1
ALLPOT	$D208	1
KBCODE	$D209	1
RANDOM	$D20A	1
POTGO	$D20B	1
SERIN	$D20D	1
IRQST	$D20E	1
SKSTAT	$D20F	1
PORTA	$D300	1
PORTB	$D301	1
PACTL	$D302	1
PBCTL	$D303	1
DMACTL	$D400	1
CHACTL	$D401	1
DLISTL	$D402	1
DLISTH	$D403	1
HSCROL	$D404	1
VSCROL	$D405	1
PMBASE	$D407	2
CHBASE	$D409	1
WSYNC	$D40A	1
VCOUNT	$D40B	1
PENH	$D40C	1
PENV	$D40D	1
NMIEN	$D40E	1
NMIRES	$D40F	1
AFP	$D800	1
FASC	$D8E6	1
IFP	$D9AA	1
FPI	$D9D2	1
ZFR0	$DA44	1
ZF1	$DA46	1
FSUB	$DA60	1
FADD	$DA66	1
FMUL	$DADB	1
FDIV	$DB28	1
PLYEVL	$DD40	1
FLDOR	$DD89	1
FLDOP	$DD8D	1
FLD1R	$DD98	1
FLD1P	$DD9C	1
FSTOR	$DDA7	1
FSTOP	$DDAB	1
FMOVE	$DDB6	1
EXP	$DDC0	1
EXP10	$DDCC	1
LOG	$DECD	1
LOG10	$DED1	1

NAME	ADDR	LENGTH
DISKIV	$E450	1
DSKINV	$E453	1
CIOV	$E456	1
SIOV	$E459	1
SETVBV	$E45C	1
SYSVBV	$E45F	1
XITVBV	$E462	1
SIOINV	$E465	1
SENDEV	$E468	1
INTINV	$E46B	1
CIOINV	$E46E	1
BLKBDV	$E471	1
WARMSV	$E474	1
COLDSV	$E477	1
RBLOKV	$E47A	1
CSOPIV	$E47D	1