

Diagnostics II User's Manual

**Copyright © 1981 by SuperSoft Inc.
All rights reserved worldwide.
No unauthorized duplication permitted.**

Diagnostics II

Table of Contents

CHAPTER 1	1
Introduction	3
List of Files Supplied	4
How to Use Diagnostics II	8
CHAPTER 2	11
The Memory Test	14
How to Run the Memory Test	15
Standard Output	16
Error Messages	17
Command Line Parameters	19
CHAPTER 3	17
The CPU Test	17
How to Run the CPU Test	17
Standard Output	19
Command Line Parameter	19
Comments	21
CHAPTER 4	21
The Disk Drive Test	23
How to Run the Disk Drive Test	23
Standard Output	24
Error Messages	25
Command Line Parameters	25
CHAPTER 5	27
The Disk Drive Alignment Program	27
How to Use the Disk Drive Alignment Program	30
CHAPTER 6	31
The CRT Terminal Test	31
How to Run the CRT Terminal Test	32
Standard Output	36
Comments	37
Command Line Parameter	37
CHAPTER 7	32
The Printer Test	32
How to Run the Printer Test	36
Command Line Parameters	37
CHAPTER 8	37
The Quick Test	37
How to Run the Quick Test	

SOFTWARE NON-DISCLOSURE AGREEMENT

As with all SuperSoft software, acceptance of this or other products, both in machine and human readable form, implies agreement with the principles and concepts below.

1. All software is sold on an individual CPU basis. Usage on secondary machines without permission constitutes a violation of this agreement.
2. Five (5) backup copies may be made by the user. These are for the protection of user's investment only.
3. The ideas, concepts and machine/human interface of software are all considered the property of SuperSoft Inc. and its authors.
4. The user agrees to non-disclosure of the following:
 - Underlying concepts
 - Documentation
 - Code and code fragments (in both source and object)
 - User Interface
 - Any and all aspects of software which SuperSoft developed.
5. All software is non-transferable and may not be re-sold without permission.
6. User modification of software completely removes SuperSoft Inc. from any liability regarding the operation or reliability of the software.

STATEMENT OF WARRANTY

SuperSoft disclaims all warranties with regard to the software contained on diskette, tape, or in printed form, including all warranties of merchantability and fitness; and any stated express warranties are in lieu of all obligations or liability on the part of SuperSoft for damages, including, but not limited to special, indirect or consequential damages arising out of or in connection with the use or performance of the software licensed.

Title and ownership shall at all times remain with SuperSoft and its authors.

Trademark Acknowledgments

CP/M is a trademark of Digital Research Incorporated.

Intel 8080 and **Intel 8085** are trademarks of Intel Corporation.

Zilog Z80 is a trademark of Zilog Corporation.

Diablo is a trademark of Diablo Systems Incorporated.

Qume is a trademark of Qume Corporation.

Spinwriter is a trademark of NEC Information Systems, Incorporated.

CHAPTER 1

Introduction

Diagnostics II is a set of programs, running under CP/M, designed to test each major component of a CP/M-based, 8080/8085/Z80 microcomputer system. Since the earliest days of computers, programs have been written to test whether the various components of a computer system were functioning correctly. Computer system manufacturers have usually assumed the responsibility for providing such programs. However, many microcomputer system manufacturers have been reluctant to supply test or diagnostic programs with their hardware—for two reasons: first, they are often not the ones called upon to repair that hardware; and second, the low cost of such systems often does not justify the large expense of developing diagnostic software, especially when neither hardware components nor operating systems are standardized.

Many manufacturers have, for years, built microcomputer systems around the 8080-series CPU's (that is, the Intel 8080 and 8085, and the Zilog Z80). In the last few years, one microcomputer operating system, CP/M, has come into such widespread use that it is now recognized as the *de facto* standard among microcomputer operating systems. At the present time, 8080/8085/Z80 microcomputer systems, running under CP/M, are extremely common. Such systems are composed, in mix-and-match fashion, of a large variety of different hardware components. Since Diagnostics II runs under CP/M, we need not provide individualized tests for each possible hardware system component: we can, instead, interface with all of them through the standard system calls. (However, because of this approach, any hardware features not supported under CP/M cannot be tested.) The huge market these systems represent allows us to offer Diagnostics II at an affordable price.

Diagnostics II contains programs to test your system's memory, CPU, disk drives, CRT terminal, and printer. It also contains a program to aid you in aligning disk drives. The table on page 3 lists the names of all the files supplied on your Diagnostics II distribution disk with a brief description of their contents.

All the tests prompt you to enter any necessary information and require no special knowledge to use. Obviously, the more you do know about microcomputer systems, the more useful the error reports will be to you. However, even if you are a novice at computers, you should be able to determine, at the very least, which system component or components are malfunctioning. Once you have isolated a problem to a specific piece of equipment, that unit can be removed from service and repaired or replaced. Note that these tests assume that your system is operable enough to load and execute programs. If not, we recommend that you have your entire system examined by qualified service personnel.

You can also use Diagnostics II to verify that your computer system is in good working order. Even in the absence of any apparent malfunctions, we recommend that you run all the tests at least once a week. We suggest that you run the Quick Test, which takes only about four minutes (or less, depending on your system), every time you power up your system. If, by following this procedure, you can detect a problem before any data is damaged or programming time wasted, you will be saving time in the long run.

If questions about how to run any of the tests in Diagnostics II arise while you are using the package on your system, you can refer to the **HELP** file included on your disk. This file contains excerpts from this manual describing the various tests and how to use them, which you can view on your screen. For information on how to use **HELP**, refer to the sheet describing our Online User Help System that accompanied this manual.

PLEASE NOTE: Frequently, throughout this manual, numerical values will be given in terms of the hexadecimal or base 16 number system. All hexadecimal numbers will be followed by the letter "H" to distinguish them from the more familiar decimal numbers. Since each digit in a hex number can take any value from 0 thru 15 (decimal), the usual decimal digits "0" thru "9" cannot represent the full range of possible values. Hence, in the hexadecimal number system, the letters "A" thru "F" are used as digits representing the decimal values 10 thru 15. Therefore, the hex number 10FBH, in decimal notation, is equal to $(11 \times 16^0) + (15 \times 16^1) + (0 \times 16^2) + (1 \times 16^3) = 11 + 240 + 0 + 4,096 = 4,347$.

List of Files Supplied

Each Diagnostics II discette contains the following files:

MTEST.COM	Memory Test (for the region between 1500H and FFFFH)
MTEST2.COM	Memory Test (for the region between 0100H and 14FFH, occupied by MTEST.COM)
CPU.COM	CPU Test
DISK.COM	Disk Drive Test
ALIGN.COM	Disk Drive Alignment Program
TERM.COM	CRT Terminal Test
TERM.DEF	CRT Data File used by TERM.COM
SOROC.DEF	Version of TERM.DEF for Soroc CRT's
HAZEL.DEF	Version of TERM.DEF for Hazeltine CRT's
ADDS.DEF	Version of TERM.DEF for Adds CRT's
PRINTER.COM	Printer Test (includes specialized test for Diablo, Qume, and Spinwriter)
QRUN.COM	Quick Memory Test (automatically invokes QDISK.COM)
QDISK.COM	Quick Disk Drive Test (automatically invokes CPU.COM)
TESTALL.SUB	A sample CP/M SUBMIT file
HELP.COM	Online Help System Program
HELP	Diagnostics II Help file

How to Use Diagnostics II

FOR YOUR OWN PROTECTION, make a copy of your Diagnostics II distribution disk when you first receive it. Store the original in a safe place as a master disk, and for all subsequent work use only the copy. DO NOT alter your master disk in any way. Doing so will completely void any responsibility on the part of SuperSoft for the operation of the software.

Sequence of Tests

The recommended test sequence is:

1. Memory Test (MTEST.COM, MTEST2.COM)
2. CPU Test (CPU.COM)
3. Disk Drive Test (DISK.COM)
4. CRT Terminal Test (TERM.COM)
5. Printer Test (PRINTER.COM)

The Memory Test should be performed first. If your memory is faulty, the results of all the other tests are suspect, since there is no way to be certain that the code executed to perform those tests resided correctly in memory.

Once you have verified that your memory is functioning correctly, then you should test your CPU to be certain that it is executing all of its instructions correctly. There are many instructions in the 8080, 8085, and Z80 instruction sets that are seldom used. If your CPU fails to execute one of those less frequently used instructions properly, it may still execute most programs correctly. Thus, even careful checking of program output against anticipated results might not reveal a flaw in your CPU that would cause it to fail miserably when executing certain programs. Therefore, unless you have verified that your CPU is also functioning correctly, the results of the remaining tests may not be reliable.

If the tests of both your memory and CPU reveal no malfunctions, then you can test the rest of your system and be confident that the test results are valid. You may run the remaining tests in any order.

Invoking a Test

Each of the tests provided is in the form of an executable command file (CP/M ".COM" file). Hence, you can *invoke*, or begin to run, any one of the tests simply by entering its filename (without the extension ".COM") from the command or system level. Under CP/M, entering the name of an executable command file from the system level causes the operating system to load that file into memory and to begin executing the program it contains.

For example, if you were to type "MTEST", the name of the file containing the Memory Test program, immediately after (i.e., with no intervening spaces) the CP/M system prompt (">") and hit the Return or Enter key, the Memory Test program would be loaded into memory and begin executing. At that point, you should see the following lines on your screen:

A>MTEST

DIAGNOSTICS II—MEMORY TEST

COPYRIGHT © 1981 SUPERSOFT INC.

ANY MEMORY FROM 1500H TO FFFFH MAY BE TESTED. TEST WILL BE
REPEATED UNTIL SPECIFIED NUMBER OF ITERATIONS HAVE BEEN
COMPLETED. STRIKING ANY KEY WILL ABORT THE TEST.

STRIKING SPACE BAR OR RETURN KEY CAUSES START AND END
ADDRESSES TO DEFAULT TO 1500H AND TOP OF TPA RESPECTIVELY.

STRIKING ANY KEY WILL CAUSE ERROR TOTALS UP TO THAT POINT
TO BE DISPLAYED BY BIT POSITION FOR 1K REGIONS.

ALL NUMBERS INPUT AND OUTPUT ARE IN HEX.

TYPE L TO LOG OUTPUT TO DISK IN DRIVE A:

In the first line, "A" indicates that you are currently using your system's disk drive A (any one of the letters "B" thru "P" may appear here instead, indicating one of the other disk drives that may be attached to your system), and ">" is, of course, the CP/M system prompt, which indicates that you are at the command or system level. The underlined letters that follow are your keyboard input, which is echoed on your terminal's screen. The lines that follow, except the last, are the sign-on message of the Memory Test program. The last line is your first prompt from the Memory Test program.

Almost all the tests, when invoked by entering the program name only from the command level, will prompt you in a similar fashion to select program options or to input the information required to run the test. (If you hit the Return or Enter key after a prompt, the programs will usually interpret that response as specifying a reasonable default condition for that item.) As soon as you have responded to all of a particular program's prompts, execution of the test itself will begin. The CPU Test requires only that you decide whether or not to include the parameter "LOG" in the command line invoking it. The Quick Test, alone, requires no input from you. Both begin executing their respective tests as soon as they are invoked.

Using Command Line Parameters

Rather than providing the information required to run a test in response to a series of program prompts, you may instead supply all the required information to each program (except the disk drive alignment program, ALIGN.COM) in the form of a list of parameters following the program name on the command line. In the case of the CPU Test, this is the only way possible. The Quick Test, since it requires no input from you, also requires no parameters for execution.

The format for these command line parameters varies depending on the test. For some tests you must follow a strict format in listing parameters on the command line; other tests allow a free format. The proper format for each individual test will be described under that test.

Using CP/M SUBMIT Files to Run Diagnostics II

It would be convenient if you could run the entire sequence of tests in Diagnostics II on your system by entering a single command. You can do so by using a CP/M SUBMIT file and the SUBMIT command. Consult your CP/M documentation or one of the several CP/M manuals available for details on how to use these facilities. Obviously, you will be making your life easier if you specify how you want each "SUBMIT-ed" test performed through the use of command line parameters.

A sample SUBMIT file for all the tests in Diagnostics II (except the Quick Test) is included on your Diagnostics II distribution disk. This file, TESTALL.SUB, is listed below:

```
; TESTALL.SUB—Suggested CP/M SUBMIT file
;           to test your entire system
;
;
MTEST 1 0 0 Q N 1
;
MTEST2 1 0 0 Q N 1
;
CPU LOG
;
; The command line for the Disk Test has been
; commented out to prevent accidental loss
; of data. Please consult your user's manual
; before attempting to use this test.
;
; DISK T-B L=A M=1,77 S=1,26 F=00FF N=150
;
TERM 1
;
PRINTER P=0 L=80 C=2
```

The command line for the Disk Drive Test is incorporated only as a comment to further emphasize that you must use that test on a *blank formatted disk only*, or valuable data may be lost. You will probably need to modify the command line parameters specified in TESTALL.SUB to tailor it to your system.

The Disk Log Option

The Memory Test (MTEST.COM and MTEST2.COM), the CPU Test (CPU.COM), and the Disk Drive Test (DISK.COM) all provide a disk log option which allows you to store the test output in a disk file named DIAG.LOG. Test output still appears on your screen. The CRT Terminal Test and the Printer Test do not provide this option, because each of those tests is of no value unless you are closely observing your terminal's screen or your printer respectively, while the test is executing. The Quick Test, because it is only a brief preliminary check of your system, also does not provide this option.

The disk log option allows you to retain a copy of the error reports generated by a test for future reference. While this option is useful when you are running tests on your system in person, it is of greatest value when used in conjunction with CP/M's SUBMIT facilities for unattended testing of your system. For instance, if you had set your system to perform a series of time-consuming tasks overnight, you might have greater confidence in the results when you saw them the following morning, if you had also set your system to run the appropriate tests in Diagnostics II at intervals during the night, leaving any error reports in a file for you to examine.

In the case of the Memory Test and the CPU Test, if you select the disk log option, test output will be logged only to the disk in drive A. In the case of the Disk Drive Test, however, you can choose to log test output to a disk in any one of the drives attached to your system.

Each time you run a test using the disk log option, an entry is made into the file DIAG.LOG. If the file DIAG.LOG is not found on the disk, it will be created. If there is insufficient space remaining on that disk for a test's results, however, errors will occur. Hence, you must use your judgement in specifying this option for tests that are to be run unattended.

A test run using the disk log option may "hang" (abruptly cease executing) if the file DIAG.LOG becomes malformed to the point that it is no longer accessible. When this happens, frequently your entire keyboard will "go dead"—hitting any key produces no response. These events occur because CP/M has gone into an error state to prevent disk I/O operations from being performed on corrupted data. If this does happen, you should reset your system, reboot CP/M, erase DIAG.LOG and run the test again. If this problem persists, you should use a different disk for the file DIAG.LOG.

How this Manual is Organized

Each of the remaining chapters in this manual describes an individual test program. The tests are presented in the same order that we recommend they usually be executed in testing your system. Each chapter is divided into two sections: the first describes the test and its features; the second provides step-by-step instructions for its use.

No matter what kind of use you make of computer systems, we are certain that you will find Diagnostics II to be an indispensable tool. We would value any comments you may have about either the programs or their documentation.

CHAPTER 2

The Memory Test

Two versions of the Memory Test are provided in Diagnostics II. They are identical in all respects, except that the first version, `MTEST.COM`, can be used to test any region in RAM other than the one it itself occupies, while the second version, `MTEST2.COM`, can only be used to test the region `MTEST.COM` occupied. How to specify, for each version, the memory region to be tested will be discussed in detail later in this chapter.

The Memory Test, in either version, is not just one test, but actually a battery of four different tests: a quick test, a walking bit test, a burn-in test, and a speed test. You must select which of these four tests you wish to perform, each time you run the Memory Test. These tests can only be performed one at a time. To perform all four tests in sequence, you must re-enter all parameters (except the one enabling the disk log option) and select the next test—after the previous one is completed. Any errors detected by each of the first three tests are reported in a common format to be described later. The last test (the speed test) does not report any errors.

The **quick test** performs three checks on each individual bit* in the region to be tested. First, to check whether or not each bit can be set to zero (0), it writes `00H` into every byte in the region. It then reads each of those bytes and tests the value read so as to reveal each bit that could not be set to zero. Next, to check whether or not each bit can be set to one (1), it writes `FFH` into every byte in the region. It then reads each of those bytes and tests the value read so as to reveal each bit that could not be set to one. Finally, to check whether or not each bit can be set, at random, to either zero (0) or one (1), it writes random numbers into every byte in the region. It then reads each of those bytes and compares the value read with the value written so as to reveal each bit that is in error. By means of the quick test, you can discover every non-functional bit in memory and almost all other memory defects. The time required to perform this test is directly proportional to the number of bytes tested.

The **walking bit test** will reveal memory defects that the quick test cannot. In particular, it will reveal defects that cause addressing errors or that cause a change in the contents of one memory location to affect the contents of another. This test is made up of two parts—each one the complement of the other.

At the beginning of the first part, every bit in the region to be tested is set to zero (0). Next, the first bit in that region is set to one (1) and every other bit is tested to determine whether or not it remained zero. Any non-zero bits (except the first) are reported as errors. Then, the first bit is reset to zero and the second bit is set to one. Every other bit in the region (including the first) is tested to determine whether or not it remained zero. As before, any non-zero bits (except, in this case, the second) are reported as errors. This same process is repeated, in turn, for every bit in the region. In effect, a bit with a value of one is “walked” from one end of the region to the other; while, after each “step,” the rest of bits in that region are tested to determine whether or not changing the value of any other bit in the region would alter their values.

*What is a bit? In microcomputers based on the 8080-series CPU's, memory is organized into *bytes*, with each byte containing eight *bits* of information. Each bit can contain either the value 0 or the value 1. As in Morse code with its dots and dashes, any kind of information can be represented in terms of these 1's and 0's. Bits and bytes are grouped together to form numbers or codes representing all the kinds of information that can be stored and manipulated within a computer system.

The second part of this test is identical to the first; except that, at the beginning of the second part, every bit in the region to be tested is set to one, and in the process that follows, the "walking bit" is set to zero. In this case, any zero-valued bits (other than the "walking bit") are reported as errors.

The time required to perform this test is proportional to the square of the number of bytes tested. Depending on the speed at which your CPU runs, to perform the walking bit test on a 16K byte region in memory could take up to 13 hours. To perform the same test on a 32K byte region could take *four times* longer. It will take less time to test several smaller regions in memory than it would to test one large region containing the same total number of bytes. However, to detect the greatest number of possible errors (especially addressing errors), the region tested must be as large as possible. Obviously, you must use your own judgement in deciding how large a region to test.

The burn-in test is designed to detect "after image" errors in dynamic RAM's. A routine only five bytes in length is loaded into the first five bytes of the memory region to be tested and then executed. This routine is the most rudimentary kind of loop, which starting with an initial register value of zero, decrements that register by one each time through the loop, and tests it until its value again equals zero, at which point the routine returns control to the burn-in test program. This routine makes no references to memory outside of these five bytes and does not alter the five bytes during execution of the loop. Therefore, executing the commands in this short a loop this many times (65,536 times!) results in the most intense possible use of the five bytes of memory in which they are stored. The bit values contained in these bytes, which are refreshed every cycle and never altered, are thus thoroughly "burned" into those bytes.

After control is returned to the test program, those five bytes are complemented, read, and compared with the values originally written to them so as to reveal any bits which could not be complemented. Those bits that retained "after images" of their previous values are reported as errors. The burn-in test program then loads the same short routine into the next five bytes of the region to be tested. This entire process is repeated for each successive five byte segment of memory, in turn, until the whole region has been tested. The time required to perform this test is directly proportional to the number of bytes tested.

The speed test checks whether or not your memory and CPU are, as a unit, running at their rated speed. This test reports no errors. The bell on your terminal will ring once at the beginning and once at the end of each iteration of this test. The interval between rings is an index of the speed of your memory and CPU. (That is, the shorter the interval between rings, the faster your CPU.) For example, if your CPU is rated at 2Mhz, the interval between rings will be one minute. The table below relates the interval between rings to the rated speed of your CPU.

Speed (CPU Clock Rate)	Interval Between Rings (seconds)
2 Mhz	60
4 Mhz	30
5 Mhz	20

Specifying Memory Regions to be Tested

As mentioned earlier, the Memory Test is provided in two versions, **MTEST.COM** and **MTEST2.COM**, each of which may be used to test memory regions that fall within a different range of memory addresses. **MTEST.COM** may be used to test any region within the range 1500H to FFFFH. **MTEST2.COM** may be used to test any region within the range 0100H to 14FFFH (the memory locations occupied by **MTEST.COM**). To specify, for either version, the memory region to be tested, you must specify the start address and the end address of the region, either as parameters on the command line invoking the Memory Test program or in response to the prompts displayed on your screen by the program as it executes.

If your system has more than 64K of RAM organized in selectable memory banks, these same address limitations apply when either version of the Memory Test is used to test those banks. Both versions allow you to choose whether or not you wish to test one of those banks, and to specify the memory location or output port and the control byte required to select it. This input may also be either in the form of command line parameters or responses to program prompts.

Please note that to give you the greatest possible flexibility in testing your system's memory, the Memory Test does *not* check the start address and the end address you have specified to determine whether or not testing such a region would cause the CP/M system routines to be overwritten. This is only likely to be a problem when running **MTEST.COM**. If this does occur, the Memory Test will either fail to execute or behave quite strangely. To recover from such a situation, you must reset your system, re-boot CP/M, and run the Memory Test again with a safer set of start and end addresses. (Hitting Return or Enter in response to the Memory Test program's prompt for the end address of the region to be tested causes that end address to default to the top of the Transient Program Area [TPA], thus avoiding any damage to CP/M.)

However, it is just as important to test the memory occupied by the CP/M system routines as it is to test the rest of your memory. To do so, you must first reconfigure those routines for a memory size smaller than the actual size of your system's memory using CP/M's **MOVCPM** facility. Then you may test that portion of your system's memory which is now above and beyond that occupied by CP/M. For example, if you have a 32K hardware system, you could reconfigure CP/M to fit a 16K memory size and then test the memory from 4000H to 8000H (1K or one kilobyte of memory contains 0400H addressable memory locations).

How to Run the Memory Test

Throughout the following section, suggested user input is underlined, while program output is not. Both are in upper case and indented to distinguish them from the rest of the text. Program output is exactly as it should appear on your terminal.

1. To invoke the Memory Test, enter "MTEST" or "MTEST2" immediately after (i.e., with no intervening spaces) the CP/M system prompt (">").
2. If you entered "MTEST", you should see the following lines on your screen:

```
A>>MTEST
DIAGNOSTICS II—MEMORY TEST
COPYRIGHT © 1981 SUPERSOFT INC.
ANY MEMORY FROM 1500H TO FFFFH MAY BE TESTED. TEST WILL BE
REPEATED UNTIL SPECIFIED NUMBER OF ITERATIONS HAVE BEEN
COMPLETED. STRIKING ANY KEY WILL ABORT THE TEST.

STRIKING SPACE BAR OR RETURN KEY CAUSES START AND END
ADDRESSES TO DEFAULT TO 1500H AND TOP OF TPA RESPECTIVELY.

STRIKING ANY KEY WILL CAUSE ERROR TOTALS UP TO THAT POINT TO
BE DISPLAYED BY BIT POSITION FOR 1K REGIONS.

ALL NUMBERS INPUT AND OUTPUT ARE IN HEX.
TYPE L TO LOG OUTPUT TO DISK IN DRIVE A:
```

If you entered "MTEST2", you should see the following lines:

```
A>>MTEST2
DIAGNOSTICS II—MEMORY TEST
COPYRIGHT © 1981 SUPERSOFT INC.
ANY MEMORY FROM 0100H TO 14FFH MAY BE TESTED. TEST WILL BE
REPEATED UNTIL SPECIFIED NUMBER OF ITERATIONS HAVE BEEN
COMPLETED. STRIKING ANY KEY WILL ABORT THE TEST.

STRIKING SPACE BAR OR RETURN KEY CAUSES START AND END
ADDRESSES TO DEFAULT TO 0100H AND 14FFH RESPECTIVELY.

STRIKING ANY KEY WILL CAUSE ERROR TOTALS UP TO THAT POINT
TO BE DISPLAYED BY BIT POSITION FOR 1K REGIONS.

ALL NUMBERS INPUT AND OUTPUT ARE IN HEX.
TYPE L TO LOG OUTPUT TO DISK IN DRIVE A:
```

In either case, the first line you see contains your command invoking the Memory Test program (here shown underlined to distinguish your input from the program's output). All the remaining lines, except the last, are the Memory Test's sign-on message. The last line is the program's first prompt.

3. Type an "L" immediately following that first prompt if you want the test output to be placed in a file on the disk in your system's drive A, as well as to be displayed on your terminal's screen. If you want the output only to appear on your screen, hit the Return or Enter key.

If you type an "L", you will see:

```
TYPE L TO LOG OUTPUT TO DISK IN DRIVE A:L
OUTPUT WILL BE LOGGED TO A:DIAG.LOG.
TYPE START ADDRESS (HEX):
```

Whenever you select this Disk Log option, the test output will always be placed in a file of this same name on whatever disk is in drive A. (If there is not already a file with that name on that disk, it will be created.) The last line you see is your next program prompt.

If you hit Return or Enter, you will see:

```
TYPE L TO LOG OUTPUT TO DISK IN DRIVE A:
TYPE START ADDRESS (HEX):
```

In this case, the test will send its output only to your terminal. As above, the last line you see is your next program prompt.

4. Next, type the four hexadecimal digits you wish the program to use as the start address for the region to be tested immediately after the prompt line shown above. Only the characters "0" thru "9" and "A" thru "F" are recognized as valid. If you type any other character, the program will prompt you again for the entire address string. Hence, if you make a mistake in typing these digits, you cannot delete or backspace over the mistake. You should, instead, type an invalid character following the mistake. This will cause the program to reprompt you and thus allow you to type in the correct digits. If the error is in the last digit, you have no choice but to terminate the test and start again. As soon as you have typed in the last digit, the program will enter the entire string as the start address of the region to be tested. Hitting the space bar or the Return or Enter key will cause the start address to default to a pre-determined value. In the case of MTEST.COM, this value is 1500H. For MTEST2.COM, it is 0100H. Either default value will be printed on the line following the prompt. As soon as you have made your choice, you will be prompted for the end address of the region to be tested. If you were running MTEST.COM and had chosen the default value, you should see the following lines on your screen:

```
TYPE START ADDRESS (HEX):
1500
TYPE END ADDRESS (HEX):
```

5. You should follow the exact same procedure in responding to the prompt for the end address as you did for the start address. In this case, the defaults for MTEST.COM and MTEST2.COM are the top of the TPA and 14FFH respectively. If you were running MTEST2.COM and chose the default, you should see the following lines on your screen:

```
TYPE END ADDRESS (HEX):
14FF

PLEASE SELECT TEST:
Q — QUICK TEST
W — WALKING BIT TEST
B — BURN-IN TEST
S — SPEED TEST
```

6. The last five lines above prompt you to choose the test you wish to perform on your memory. To select a test, you should type the first letter of its name. If you chose the Quick Test, you would see the following on your screen:

```
PLEASE SELECT TEST:
Q - QUICK TEST
W - WALKING BIT TEST
B - BURN--IN TEST
S - SPEED TEST
Q
TYPE B TO TEST SELECTABLE
MEMORY BANK:
```

The last two lines are the next program prompt.

7. If your system's memory has the bank select feature, you may type the letter "B" immediately after this prompt to test any one of those banks. If you chose to do so, you will then be prompted for the information required to select one of those banks and your screen should appear as follows:

```
TYPE B TO TEST SELECTABLE
MEMORY BANK:B
POKE WHICH PORT?
```

In response to this prompt, you should specify the hex value corresponding to either the number of the I/O port or the address of the memory location that controls your memory's bank select feature. If the value specified is less than or equal to 0100H (256), the program assumes that it is the number of a port. Otherwise, the program assumes that it is the address of a memory location. Since this value is hardware dependent, you should consult the user's manual for your system to determine what it should be. As soon as you have typed in the appropriate value, you will see the next prompt:

WHAT VALUE?

This prompts you to input the appropriate hex value for the byte which, if placed either at the I/O port or the memory location previously specified, will cause the memory bank you wish to test to be selected. As this value is also hardware dependent, you should again consult the user's manual for your system to determine what it should be. As soon as you have typed in the appropriate value, the program will present you with its final prompt.

If your system's memory does not have the bank select feature or you do not wish to test any of the alternate memory banks, hit the "Return" or "Enter" key. You will then see the program's final prompt.

8. The Memory Test's final prompt is:

TYPE NUMBER OF ITERATIONS (DEFAULT = 1):

If you want to execute the test you selected only once on the memory region you specified, select the default condition of only one iteration by hitting the "Return" or "Enter" key. Otherwise, type the hex value corresponding to the number of iterations desired immediately after this prompt. You may choose any number of iterations from 0001H (1) to FFFFH (65,535). A small number of iterations will suffice to discover all possible memory defects within the region tested. Once you have responded to this prompt, the Memory Test program will begin to execute the test you specified. It will re-execute this test either until the specified number of iterations have been completed or until any key on your keyboard is hit. Hitting any key immediately aborts the current test.

Standard Output

The memory test will produce the following output:

1. First, you will see a memory map of the tested area printed on your screen. For example, if you are testing the region from 2000H to 3000H, you will see:

```

0000H      2000H      4000H                      FFFFH
OK          8K        16K          //          64K
+ ..... + ..... + ..... // ..... +
                11111

```

A "1" means that memory is present, a "0" indicates that no memory exists.

2. If the region under test is good, the following output will be seen at the end of each pass:

PASS COMPLETE XXXX ; COUNT OF ERROR BYTES = 0

Where XXXX is the number of passes that have occurred. This means that no errors have been found.

3. For each error that is detected, the following is displayed.

```

LOCATION      DATA WRITTEN  DATA READ
xxxxH        xxH          xxH

```

Here, as throughout the memory test, all numeric output is in hexadecimal. The error report will continue as long as errors are discovered. This means that transient errors will produce only sporadic output whereas dead memory will produce constant output.

4. When you stop the test by pressing a key, you will be given a summary of all errors bit by bit. The cumulative error report is issued so that you can let the test run for a long time, say overnight, without having to watch the display. For example:

ADDRESS — BITS

```

          7       6       5       4       3       2       1       0
F000     0017    0000    0000    0000    0012    0023    0000    0000
F100     0000    0000    0000    0000    0000    0000    0000    0000

```

Here, we see that 17H errors occurred in bit 7, 12H in bit 3, and 23H in bit 2 of in the region F000H to F0FFH. The second K, F100H to F1FFH had no errors.

Since memory is often organized by bits, this is useful in determining the exact memory chip that failed. (Consult your memory board hardware manual for information.)

Remember, each "PASS . . ." displays a cumulative error count. Even if you do not see errors on the screen the count will tell you that they are there. Upon key press, they will be reported by bits.

Note that if a memory region which includes non-existent locations is tested, the memory will test as bad. On the S100 bus, the contents of these locations should appear as FFHs no matter what is written into them.

5. If you run the memory speed test, you will hear the bell on your terminal sound at a constant interval. This interval is defined in the table below:

Speed (CPU Clock Rate)	Interval Between Rings (seconds)
2 Mhz	60
4 Mhz	30
5 Mhz	20

If you do not get this timing, then either something is wrong, or your memory/CPU is not running up to specifications.

Error Messages

As described in the previous section, there are two types of error reports. The first is the report that occurs while the test is running. It takes the following form:

LOCATION	DATA WRITTEN	DATA READ
xxxxH	xxH	xxH
xxxyH	xyH	xyH
.	.	.
.	.	.
.	.	.

Where the location refers to the actual memory address where the failure happened, data written is the byte that the test tried to write to that location, and data read is the byte that the test read back.

The second error report is the summary. As described earlier, this report prints cumulative totals of all errors by bit position for 1K regions. It takes the following form:

ADDRESS — BITS

	7	6	5	4	3	2	1	0
X000	yyyy	yyyy	yyyy	yyyy	yyyy	yyyy	yyyy	yyyy
X100	yyyy	yyyy	yyyy	yyyy	yyyy	yyyy	yyyy	yyyy
X200	yyyy	yyyy	yyyy	yyyy	yyyy	yyyy	yyyy	yyyy
.								
.								
.								

The address column indicates the 1K memory region tested. In the columns under each bit number you will see a four digit hexadecimal number representing the total number of errors that occurred in that bit position within the 1K region indicated.

If you have found errors in your memory, then you should have the faulty board serviced. The above summary will pinpoint the error for you.

Command Line Parameters

All options allowed by the memory test may be specified in the command line. There are two forms, bank select form and non-bank select form. For both forms, the *exact ordering* of the parameters is necessary. First, with bank select:

A > MTEST L S E T B P V R

Where:

- L: Disk log option, an "L" here specifies that the output be directed to disk. A "N" suppresses the disk log option.
- S: The start address of the test in hexadecimal. The default can be used by entering "0".
- E: The end address of the test in hexadecimal. The default can be used by entering "0".
- T: This field specifies the type of test. Enter a "Q" for quick, "W" for walking bit, "B" for burn-in, and "S" for speed.
- B: Bank select. If a "B" is present then bank select is enabled, type an "N" otherwise.
- P: Port or memory location. When bank select is enabled this field should have the value, in hexadecimal, of the port or memory location that is used to control bank select.
- V: Bank control byte. When bank select is enabled this field should have the control byte that is recognized by your memory system.
- R: This is the repetition number in hexadecimal.

Example:

A > MTEST L 2000 3000 Q B 23 55 1

Or, if no bank select option is used the following command line convention holds:

A > MTEST L S E T N R

Where:

- L: Disk log option. An "L" here specifies that the output be directed to disk. A "N" suppresses the disk log option.
- S: The start address of the test in hexadecimal. The default can be used by entering "0".
- E: The end address of the test in hexadecimal. The default can be used by entering "0".
- T: This field specifies the type of test. Enter a "Q" for quick, "W" for walking bit, "B" for burn-in, and "S" for speed.
- N: No bank select used.
- R: This is the repetition number in hexadecimal.

Example:

A > MTEST C 3000 30FF W N 3

CHAPTER 3

The CPU Test

The CPU Test, CPU.COM, interprets a program that is designed to execute all single instruction sequences and many multiple instruction sequences. After each instruction sequence, the program tests all of the CPU registers to see that the proper registers changed correctly, *and* only those registers changed. This will detect, for instance, if storing into the A register affects the B register.

Test features:

- Determines whether your CPU is a Z80 or an 8080/8085.
- Checks the speed at which your CPU runs.
- Tests each of the instructions that your CPU is able to execute, checks all results, and reports any errors.

How to Run the CPU Test

1. In response to the system prompt, enter "CPU". The CPU test accepts the following command line parameter: "LOG". If you wish to direct the output from the test to a disk file then enter the word "LOG" on the command line; otherwise leave it blank. You will see:

```
Diagnostics II  CPU Test
Copyright © 1981 SuperSoft Inc.
```

2. After the copyright notice the following line is displayed:

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

Each character represents a short preliminary test of the CPU. If this sequence is not printed, the CPU is faulty, perhaps too faulty to give reliable results on the rest of the test. Included in the preliminary tests are tests of the jump instructions and various other instructions that are crucial to the performance of the test. Also included is a checksum of the test itself. This is to determine if the test is residing correctly in memory.

Standard Output

1. The first message displayed indicates the CPU type of your computer. The message will be either "CPU IS Z80" or "CPU IS 8080/8085". If this message is wrong, then your CPU is faulty.
2. Next, a timing test is performed to determine whether or not your CPU is running at its rated speed. The message "BEGIN TIMING TEST" will be displayed and the bell on your terminal (if it has one) will ring. A short time later, usually less than one minute, the bell on your terminal will ring again and the message "END TIMING TEST" will be displayed. You should time the interval between these rings. The following table summarizes the relationship between the speed of your CPU and the interval between rings.

Speed (CPU Clock Rate)	Interval Between Rings (seconds)
2 Mhz	60
4 Mhz	30 *
5 Mhz	20

* 60 secs for 4MHz Micro Decision

If the timing as you measured it is radically different from what would be expected for your CPU, then your system may be malfunctioning. (If you have added wait states, have DMA, run multi-user, or run an interrupt driven system, the timing test may not give the results described above.)

3. Next, each instruction will be tested.
4. At the completion of a successful pass you will see:

CPU TESTS OK

The program will then terminate, returning to the CP/M system level.

5. If errors are detected, they will be reported in the following format—as they are discovered:

```
CPU FAILED TEST:
ERROR COUNT XXXXH
INSTRUCTION SEQUENCE WAS XXXXXXH
REGISTER U CONTAINS XXH
BUT SHOULD CONTAIN XXH
REGISTER VALUE BEFORE INSTRUCTION
SEQUENCE WAS XXH
TEST NUMBER XXXXH
```

All numbers printed by the CPU test (represented by the X's above) are hexadecimal. The instruction sequence given is the first one that caused your CPU to fail. Hitting the space bar will cause the test to continue. It must be taken into consideration when examining later error reports that the bad register value is not replaced by the good register value. Thus, some later error reports may be spurious.

Note: all register values are compared byte by byte and the register names are printed according to the following table:

Register name		Register		
High byte	Low byte			
B	C	BC		
D	E	DE		
A	F	PSW		
H	L	HL		
S	P	SP		
ZB	ZC	BC	(Z80 alternate register)	
ZD	ZE	DE	"	"
ZA	ZF	PSW	"	"
ZH	ZL	HL	"	"
ZI	ZX	IX	• (Z80 only)	
ZJ	ZY	IY	"	"
	ZR	RF	"	"
	ZV	IV	"	"

6. The total number of errors is reported at the end of the test:

XXXXH ERRORS

Command Line Parameters

The command line takes the following form:

A> CPU LOG

Where

LOG: If "LOG" is present then the output will be logged to the disk in drive A, otherwise output will be sent only to the console.

Comments

Most likely, if the CPU fails to execute an instruction sequence correctly, it will continue to do so. To examine more precisely what your CPU is doing, write a short assembly language program that includes the instruction sequence given in the first error message, and execute it, perhaps using DDT.

Of course, such an examination would only be for verification, as a bad CPU cannot be repaired.

It appears that some 8080 and 8085 CPU's store different values in the flag byte during some arithmetic instructions. If the CPU test fails and if you are satisfied with the performance of your CPU you can "customize" this test for your CPU using DDT (See the CP/M manual for a description of how to use DDT). To do this, you should follow the procedure outlined below:

1. From the command level you should enter the following command line:

A> SAVE 0 GO.COM

This creates a file of zero length with filename GO.COM on the disk in your system's drive A. You will make use of this file later in this process.

2. Again from the command level, you should enter the command line shown below:

A> DDT CPU.COM

The following message will then be displayed on your screen:

```
DDT VERS 2.0
NEXT PC
4C00 0100
—
```

3. The dash on the last line above is DDT's prompt. Immediately after that dash, you should enter the following DDT command:

— S3B6B

DDT will respond by displaying the following line on your screen:

3B6B 01

4. On this same line, you should type "00" and hit Return. DDT will respond by displaying another line on your screen. This should appear as follows:

```
3B6B 01 00
3B6C 01
```

5. On this new line, you should again type "00" and hit Return. DDT will again respond by displaying another line on your screen. These two lines are shown below as they should appear on your screen:

```
3B6C 01 00
3B6D 58
```

6. On this last line, you should type "." and hit Return. DDT will respond by displaying its prompt, "-", on the next line. These two lines should appear as follows:

```
3B6D 58 .
-
```

7. After DDT's prompt, you should type "G" and hit Return. Up to this point, your screen should be identical to what is shown below:

```
A > SAVE 0 GO.COM
A > DDT CPU.COM
DDT VERS 2.0
NEXT PC
4C00 0100
- S3B6B
3B6B 01 00
3B6C 01 00
3B6D 58 .
- G
```

8. Now, from the command level, you should invoke GO.COM, the file of zero length you created earlier, by typing "GO" and hitting Return. Doing this causes whatever currently is residing in the TPA (Transient Program Area) to be executed as if it were a standard CP/M command file. Execution begins at *tbase*. In this case, the modified CPU Test program is executed. Hence, after entering "GO", should see several screenfuls of debug information in rapid succession.
9. When all this activity on your screen comes to a halt and you again see the CP/M system prompt, you should enter the following command line:

```
A > SAVE 75 CPU1.COM
```

If everything has gone smoothly, you will have succeeded in creating a version of the CPU Test, with filename CPU1.COM, tailored to suit your particular 8080 or 8085 CPU. You may now erase GO.COM.

CHAPTER 4

The Disk Drive Test

The Disk Drive Test, DISK.COM, may be used to detect any malfunctions in the disk drives attached to your system. It may also be used to verify that a drive is functioning properly.

Disk drive problems are among the most annoying of problems. They also tend to cause the most damage—in the form of irrecoverable data loss. We recommend that any malfunctions in your disk drives be serviced immediately.

Test features:

- Tests read/write functions.
- Tests seek accuracy.
- Tests data integrity after transmission.
- Allows user defined or random seek patterns.
- Allows memory locations or port contents to be dumped if errors occur.
- Allows user defined fill patterns.
- Allows user defined test sectors.

How to Run the Disk Drive Test

1. ABOVE ALL REMEMBER: you must use a *blank formatted discette* for this test. If you use one which has anything important on it, then that information will be destroyed. *The disk test will destroy any and all data that was previously on the disk!*

2. Enter "DISK" while at system level. You will see the following:

Diagnostics II — Disk Drive Test.
Copyright © 1981
SuperSoft Inc.

Enter drive name for disk log (CR=disk log off):

If you wish to have the disk test log its output to a disk file then you should respond with a drive name, "A" thru "P". If you want the output to appear only on the console, then simply hit carriage return. *It is important to remember that the log file cannot be on the disk being tested!*

3. Next you will see:

Select drive to be tested:

Respond with a drive name, "A" thru "P". Remember, the discette that you use in the disk test should not contain information, because the disk test will destroy the contents of that disk.

4. Next, you will be prompted for the range of tracks that you wish to test:

Enter Min and Max track to be tested: *

Here, you should enter two decimal numbers from 1 to 32,767, separated by a colon. The default is 1:10 if a carriage return is entered. This number is dependent on the exact nature and size of your disk system. The default will work on any drive.

* Use the default settings for single-sided Micro Decisions. For double-sided systems, enter 0-79 for min/max track and 1-40 for min/max sector.

5. Next:

Enter min and max sector to be tested:

As before, enter two decimal numbers in the range 1 to 32,767, separated by a colon. The default here is 1:26 if you strike a carriage return. This number, like the one before, is dependent on your specific hardware. The default will work on any disk system.

6. You will then be prompted for the seek pattern:

Enter seek pattern:

This allows the user to specify a sequence of sectors to tested. The information is entered as a string of number pairs separated by commas. Numbers within pairs are separated by colons. The first number in the pair is the track; the second is the sector. You must enter the seek pattern in decimal. For example, if you wish to seek to track 1, sector 2, then track 2, sector 4, you would enter the information thus:

1:2,2:4

You may *not* use spaces in this string. The maximum number of track/sector combinations is 128.

You may default this to a random seek by simply hitting a carriage return.

7. Next:

Enter memory locations and IO ports to dump
on error:

If you would like to see the contents of specific memory locations or ports, then you may enter them here. A port is assumed if the number entered is less than 256, otherwise a memory location is assumed. These numbers must be entered in hexadecimal. The ports or locations must be separated by commas; no spaces are allowed. The limit of memory locations and ports combined is 128.

For example, if you wish to see the contents of memory locations 9FFF, A000 and port 37, then you should enter:

9FFF,A000,37

8. Then, you will be asked for the sector fill pattern:

Enter sector fill pattern:

This refers to the exact byte or byte sequence which will be written into each sector. The pattern is entered in hexadecimal notation with no spaces allowed. The maximum number of entries is 126; the first two bytes in each sector are used by the disk program. If you enter less than 126 bytes, then the pattern will be repeated until the sector is full. For example, if you wish to write FF,00,FF,01 throughout each sector then enter that information thus:

FF00FF01

You may default this to a random pattern by simply hitting carriage return.

9. The next prompt is for the number of times you wish the test to be performed:

Enter the number of iterations (default = 1):

As soon as you have entered this number, the test will begin. If you hit carriage return, the test will be performed only once.

You must always use a discette that has been initialized, otherwise false disk errors will result! Opening the drive door, pulling the plug, and so on will also produce false disk errors.

Standard Output

Normal output will occur in the following order:

1. The first event to occur is the labeling of the sectors. This is done so that the program can confirm that the proper sector is being read. You will see the following message on the console telling you this:

Labeling all sectors

Only those sectors specified will be labeled. You will see a "." for each track labeled.

2. Once the labeling is complete an error total will be displayed if errors occur. It takes the following form:

Totals: X hard errors, X seek errors, X data transfer errors

The nature of these errors are explained later in this chapter.

3. Next, the test will begin, using either the specifications you gave, or the defaults. At the end of each pass, you will see the following message:

XX Passes remaining

This keeps you up to date on the progress of the test. This message is displayed only for user defined seek patterns. If you use the random seek default no message is displayed.

4. Once the repetition count has been reached, the test will terminate. Again, a summary of errors will be printed. It will take the same form as in item 2 above.

Error Messages

There are three kinds of errors which the disk test will report. They are explained below:

- **Hard disk error.** This error may be caused by the following:

- write protection
- open drive doors
- loss of power
- extensive drive malfunction
- an unformatted discette

The error report takes the form:

Hard disk error on write. track X, sector Y
(read)

Where X and Y are the track and sector numbers respectively.

- **Seek error.** This error may be caused by the following:

- faulty drive
- bad discette
- faulty memory
- faulty controller
- line voltage transients
- sluggish stepper motor

Frankly, some drives will just have trouble making some long seeks. If this is the case, you should consult the manufacturer for instructions. This error takes the form:

Seek error: sought track X sector Y, found track Q sector W

Where X,Y,Q, and W stand for the track and sector numbers.

A data transfer error, described below, in the first two bytes of a sector may cause a false seek error. However, there would still be a malfunction in your drive.

- **Data transfer error.** This error reports instances of data improperly written, stored, or read. The principal causes are:

- faulty or old discettes
- dirty heads on drive
- bad memory
- line transients
- physical shock during read/write

The form of the report is:

Data transfer error of X bytes, track Y sector Z

Where X,Y, and Z stand for the number of bytes in error, the track number and the sector number respectively.

Command Line Parameters

All information needed to operate the disk test may be specified on the command line. The form is free format; the following convention is used:

A >DISK L=x M=x,y S=x,y T=x P=x D=x F=x N=x

Where

- L: Specifies the drive for disk log output. Enter a drive name "A" thru "P". If no log is desired then omit this parameter altogether.
- M: Set minimum and maximum tracks to be tested. With x referring to minimum and y to maximum.
- S: Sets minimum and maximum sectors to be tested. With x referring to minimum and y to maximum.
- T: Specifies the disk drive to be tested. Enter a drive name, "A" thru "P".
- P: Sets the seek pattern. If not present then random seek occurs. Enter the pattern same as described earlier: <track>:<sector>,<track>:<sector> ...
- D: Sets the memory locations or ports to be displayed on error. Enter addresses separated by commas.
- F: Sets the sector fill pattern. Entered same as described earlier, hexadecimal digits separated by commas.
- N: Sets the number of repetitions.

Examples:

A >DISK L=A M=1,6 S=1,4 T=B P=1:2,1:3 D=4404 F=FF00FF N=1

A >DISK M=1,10 T=B N=1

CHAPTER 5

The Disk Drive Alignment Program

The Disk Drive Alignment Program, `ALIGN.COM`, has been included in Diagnostics II to aid you in aligning the disk drives attached to your system. This program is essentially identical to the Disk Drive Test program except that it performs read operations only. You may specify the ranges of tracks and sectors to be read—as well as the order in which the sectors are to be read. This program will not help you align your drives unless you also have available to you the special alignment disk for your system. This disk is usually available from the manufacturer of your disk system.

In general terms, the procedure for aligning a disk drive is as follows: first, after powering up your computer system, insert the special alignment disk in the drive to be aligned; then, connect the test leads of an oscilloscope to the appropriate output lines of that drive; next, using `ALIGN.COM`, specify the range of tracks to be sought, the range of sectors to be read, and the order in which the sectors are to be read. Finally, while this program is executing those read operations, observe your oscilloscope's display and adjust the calibration on your drive until the signal displayed is at maximum amplitude. Once this adjustment has been made for each track on the disk, your drive will be in alignment.

The procedure described above obviously glosses over important details. You should not attempt to use this program to align your drives until you have thoroughly studied the documentation provided with your drives and with the appropriate alignment disk.

How to Use the Disk Drive Alignment Program

1. Insert the special alignment disk in the drive to be aligned.
2. Type `"ALIGN"` while at system level. You will see the following:

```
Diagnostics II
Disk Drive Alignment Program
Copyright © 1981
SuperSoft Inc.
```

Select drive to be aligned:

Respond with a drive name, "A" thru "P".

3. Next, you will be prompted for the range of tracks to be sought:

Enter Min and Max track to be sought:

Here, you should enter two decimal numbers between 1 and 32,767, separated by a colon. The default is 1:10 if you hit Return. This number is dependent on the exact nature and size of your disk system. The default will work on any drive.

4. Next:

Enter min and max sector to be read:

As before, enter two decimal numbers in the range 1 to 32,767, separated by a colon. The default here is 1:26 if you hit Return. This number, like the one before, is dependent on your specific hardware. The default will work on any disk system.

5. You will then be prompted for the read pattern:

Enter read pattern:

This allows you to specify the sequence in which sectors are to be read. The information is entered as a string of number pairs separated by commas. Numbers within pairs are separated by colons. The first number in the pair is the track; the second is the sector. You must enter the read pattern in decimal. For example, if you wish to read to track 1, sector 2, then track 2, sector 4, you would enter the information thus:

1:2,2:4

You may *not* use spaces in this string. The maximum number of track/sector combinations is 128. You may default this to a random read pattern by simply hitting Return.

6. Next:

Enter memory locations & IO ports to dump on error:

If you would like to see the contents of specific memory locations or ports, then you may enter them here. A port is assumed if the number entered is less than 256, otherwise a memory location is assumed. These numbers must be entered in hexadecimal. The ports or locations must be separated by commas; no spaces are allowed. The limit of memory locations and ports combined is 128.

For example, if you wish to see the contents of memory locations 9FFF, A000 and port 37, then you should enter:

9FFF,A000,37

7. The next prompt is for the number of iterations:

Enter the number of iterations (default = 1):

As soon as you responded to this last prompt, the Disk Drive Alignment program will begin to execute the specified read operations. If you hit Return in response to this prompt, the specified sequence of operations will be performed only once.

CHAPTER 6

The CRT Terminal Test

The CRT Terminal Test requires extensive operator interaction and attention. Various functions will be tested. While the tests are taking place, you must watch your screen closely to note any errors that occur.

Test Features:

- Tests all printing ASCII characters in all fields.
- Tests clear screen terminal command.
- Tests cursor addressing terminal commands.
- Tests foreground and background mode.
- Tests clear foreground terminal commands.
- Tests clear to end of line terminal command.

How to Run the CRT Terminal Test

1. There are three files supplied on your discette:

SOROC.DEF
HAZEL.DEF
ADDS.DEF

If you have a Soroc or ADM-3, or compatible, you should use SOROC.DEF. If you have a Hazeltine 1500 or compatible, then use HAZEL.DEF. If you have an ADDS Regency or compatible, then use ADDS.DEF. Otherwise you must use the program to create your own TERM.DEF file. In other words, we have supplied versions that will test the above mentioned terminals without user definition of terminal command codes. If you do not have one of the above then you will have to patch your terminal command codes using the prompts in the program.

2. TERM.COM will read the terminal information from TERM.DEF. If you have one of the predefined terminals, then you should issue the following command:

A > PIP TERM.DEF=SOROC.DEF { or Yourterm.DEF }

If you do not have one of the predefined terminals, you will have to create a TERM.DEF file for your terminal. To do so, you should follow the procedure described below.

3. From the system level, enter "TERM" immediately after the system prompt. You will see the following prompt:

Diagnostics II—CRT Terminal Test V1.1
Copyright © 1981
SuperSoft Inc.

Enter N to create new terminal definition,
or any other character for standard file:

If you have one of the predefined terminals and have already PIPed it into a copy of "TERM.DEF" then you should answer with a carriage return, otherwise type an "N". Typing an "N" will allow you to create a TERM.DEF file for your terminal.

4. If you are lucky enough to have a predefined terminal, then skip ahead to section 22. All entries described below are entered without spaces.
5. You will see the following:

All responses will be hexadecimal unless otherwise noted
Enter HEX sequence for cursor home:

Enter both the lead in code, if any, and the actual command for homing the cursor. These hexadecimal numbers should be entered without spaces and with leading zeros. For example, a Soroc 120 has the following cursor home command: 1E.

6. Next:

Enter HEX sequence for begin protected field:

This refers to the command to start protected field write. The same conventions hold as for above. For a Soroc 120 the entry is: 1B29. Here the lead code is ESC (1B) and the actual command is 29.

7. Next:

Enter HEX sequence for end protected field:

As above, except to end protected field write. For Soroc 120: 1B28.

8. Next:

Enter HEX sequence for enable protected fields:

Some terminals do not require this. If yours does not then enter "00"; otherwise enter the necessary information. For Soroc 120: 1B26.

9. Next:

Enter HEX sequence for disable protected fields:

Some terminals do not require this. If yours does not then enter "00"; otherwise enter the necessary information. For Soroc 120: 1B27.

10. Next:

Enter HEX sequence for clear screen:

For Soroc 120: 1B2A.

11. Next:

Enter HEX sequence for clear to end of line:

For Soroc 120: 1B54.

12. Next:

Enter HEX sequence for clear unprotected fields:

As before, this may also be called clear "background" on some terminals. For Soroc 120: 1B2B.

13. Next:

Enter HEX sequence for load cursor position:

For Soroc 120: 1B30.

14. Next:

Enter DECIMAL value for col 0:

This refers to the cursor addressing scheme that your terminal uses. That is, usually terminal cursor addressing is done as follows: send the load cursor command, send column coordinate, send row coordinate. We need to know the base numbers for the upper left hand corner of your screen. This number is 32 for a Soroc 120.

15. Next:

Enter DECIMAL value for top line:

As before, we need to know the base for the upper left hand corner. For Soroc 120 it is again 32.

In the last three paragraphs, you should have defined the cursor commands to write into the first position in the upper left hand corner. That is, load cursor command, give column, give row.

16. Next

Are the cursor coords in col,row order?

In other words, does your terminal expect the coordinates to be sent in x,y (i.e., col,row) order? If so, type "Y" for yes. Otherwise, type "N" for no. For Soroc 120: N.

17. Next

Does it use ASCII cursor coordinates?

The proper response is "Y" for a VT-100, for most others it is "N".

18. Next:

Enter HEX sequence to separate ASCII coordinates:

For a VT-100 this is "1", most other terminals need not worry about this, just enter "00".

19. Next:

Enter DECIMAL number of nulls to pad commands:

Most terminals require one to two nulls.

20. Then:

Enter DECIMAL max number of lines:

Enter DECIMAL max number of columns:

Both of these together define the size of your screen. In general, you enter 24 and 79. The 79 is used instead of 80 because the columns are generally counted starting at 0.

21. Next:

Test lower case too?

If you have lower case on your terminal then enter "Y", else "N".

22. Finally you will be asked for the number of times you wish the test to be performed.

Enter the number of iterations (default = 1):

23. As soon as you have entered this number, the test will begin. If you hit Return, the test will be performed only once. As the test runs, you must watch the display, looking for any discrepancies.

Standard Output

A sequence of screens will appear, each testing some function of the terminal. Each screen will hold for about 5 seconds or until a key is struck.

1. First, all ASCII printing characters will be displayed in each character position in a "barber pole" pattern.
2. Next, a cursor addressing test is run. In this test a row of "X" is printed on the terminal screen from the bottom up. On some terminals, the bottom row may have the last "X" cause an unwanted line feed. Do not worry about this. After the "X"s have been printed in reverse scroll fashion, the message

SuperSoft Associates Terminal Test Program 0123456789

will be printed with alternating character positions being printed each time. This should create the effect of four cursors acting simultaneously.

3. The last line of the terminal will print a message for user input and will test the backspace five times. The cursor will move out and then back repeatedly.

4. At this point the foreground/background test begins. First, the screen is cleared using repeated **CLEARLINE** commands instead of clear screen and the cursor is addressed in the home position. If your screen does not clear properly, then **CLEARLINE** is not functioning properly.
5. The screen will be filled with alternating foreground and background messages.
6. The cursor will jump back to the top of screen and ask for input. After input, foreground will clear.
7. Then, the user will again be asked for input. After input, the background will clear and the test is terminated.

Comments

If your terminal does not seem to work right, the first thing to do is to make sure that you have the control codes correctly entered into the **TERM.DEF** data file. Some terminals send the cursor address sequence in the order X-Y, while others send it Y-X. If the cursor addressing does not work, try reversing the X and Y coordinates when you enter them. Remember, some terminals may not have all the above mentioned functions.

Command Line Parameters

The terminal test supports the following command line convention:

A> TERM N

Where

N: Sets the number of iterations

Using the command line parameter defaults the terminal test file to the current **TERM.DEF**. If one does not exist, an error will occur and the test will abort.

CHAPTER 7

The Printer Test

The Printer Test, `PRINTER.COM`, exercises all the features of your printer that can be exercised under CP/M. It requires that you closely observe the operation of your printer while it is being put through its paces and note any inadequacies in its performance.

Test features:

- Tests each printing ASCII character in each carriage position.
- Tests both UPPER and lower case.
- Tests every head and carriage motion for Spinwriters, Qumes, and Diablos (Serial interface only).

Including:

- Forward and reverse printing
- Interleaved overprinting
- Forward and reverse overprinting
- Overprinting with horizontal tabbing
- Boldface printing
- Superscript and subscript printing
- Scale page horizontally and vertically

How to Run the Printer Test

1. Enter "PRINTER" while at system level.
2. You will see the following:

```
Diagnostics II—Printer Exerciser  
SuperSoft Inc.  
Copyright © 1981
```

```
Enter number of iterations (default = 1)
```

Enter the number of times you want the test to run.

3. Then:

```
Enter 0 for output to printer, 1 for output to punch:
```

At this point you must tell the test where to send characters. Most users actually have the printer connected to the CP/M logical device called PRN (for printer). However, some users may have the printer connected to the logical device called PUN (for punch). If you are not sure, then try printer first. If that does not work then try punch.

4. Next:

Enter width of line in columns:

This refers to number of characters across that your printer will accommodate. That is, if you have an 80 column printer then enter 80.

5. Next:

Select test pattern:

1. 96 character ASCII
2. 64 character ASCII
3. user defined test pattern

Select one:

Usually you will want to type a "1" here, selecting the full ASCII character set. If you have only upper case on your printer, then you will normally select number "2", the 64 upper case character set. Occasionally, pattern dependent problems can occur. If this happens, then you may enter the specific test sequence. If you type a "3" you will then be prompted for the test pattern. This pattern can have a maximum length of 132 characters. The program will only allow you to enter a sequence as long as the number you entered for the columns. You may enter less. In that case, the sequence is repeated as necessary.

6. Next you will be prompted with the following:

Select printer type: (S)pinwriter, (D)iablo, (Q)ume, (O)ther:

You must tell the test what kind of printer you have. If you have one of the specified daisy wheel printers, then select the appropriate letter, either an "S", "D", or "Q", otherwise type an "O".

7. Next, you will need to enter the number of nulls after each carriage motion:

Enter number of nulls after carriage motion:

Usually two is sufficient for most printers.

8. If you have selected one of the daisy wheel printers listed above, then you will receive the following prompts, otherwise you should skip ahead to item 16.

Indicate (Y/N) which tests are to be run:

Normal print forward and backward?

This test simply prints the test pattern forward and backward. It is a preliminary test of your printer and its internal buffering logic.

9. Next:

Interleaved overprinting?

This test will overprint each character in the test pattern by first printing a character, then backing up and overprinting it before going on to the next character. It does this in both the forward and backward directions. This is useful to test head alignment.

10. Then:

Print same line forward and backward?

This will print the test pattern once forward then, without a carriage return or linefeed, backward. This also tests head alignment.

11. Next:

Overprint with absolute horizontal tabbing?

This test will print the pattern once, then overprint this line by the following method. First overprint the character in position 1. Next, the character in position 132. Next, the one in position 2, then 131, and so on. This will cause the head to fly back and forth quite quickly! Your printer should be able to perform this overprint just as well as the other examples. This is perhaps the single most rigorous test for a daisy wheel printer. It may be necessary to increase the number of nulls after carriage motion for this test to execute correctly.

12. Then:

Print bold with 1/120" offset?

This will print an excellent bold face of the test pattern. It tests the incremental spacing features of your printer.

13. Next:

Print normal, super, and subscript?

This will print a line, then backup and print a word in normal, then superscript, then subscript, etc. This test checks the carriage motions.

14. Then:

Scale page in inches horizontally?

This will print vertical bars on the page, each one inch apart horizontally. This is a measure of the accuracy of the head controller.

15. Next:

Scale page in inches vertically?

This prints horizontal lines on the page, one inch apart vertically on the page. This is a measure of the accuracy of the carriage controller.

16. Then:**Barber pole test?**

This test will print the ASCII character set in a barber pole fashion. That is, each line will be moved one character over from the preceeding, with the first character wrapped around the edge. For example:

```
ABCDEFGH  
BCDEFGA  
CDEFGAB  
DEFGABC  
EFGABCD  
FGABCDE  
GABCDEF
```

This tests each character in each character position. Once you have responded to this last prompt, the test will begin.

Command Line Parameters

All of the information necessary to run the printer test may be entered on the command line. The following convention holds:

A> PRINTER P=x L=n T=n,n . . . C=n P=n N=n O=n

Where

P: Specifies printer type. Enter "S" for Spinwriter, "D" for Diablo, "Q" for Qume, "O" for everything else.

L: Sets the line length of the printer.

T: Sets the tests performed, used only if P is not set to "O".

The tests are entered by their number:

- 1: normal forward and backward
- 2: interleaved overprinting
- 3: same line forward & backward
- 4: horizontal tabbing
- 5: bold face
- 6: superscript & subscript
- 7: scale horizontal
- 8: scale vertical
- 9: barber pole

The numbers are entered in sequence separated by commas. (see example below)

C: Selects the character set. "1" if 96 character ASCII, "2" for 64 character. If you wish to enter your own, then enter it between quotes at this point. For example:

C="abcd" or C=1

R: Sets the number of repetitions.

N: Sets the number of nulls after carriage motion.

O: Determines whether output is directed to printer or to punch. "1" for Punch, "0" for printer.

Examples:

A> PRINTER P=S L=80 T=1,2,3 C=1 R=2 N=1 O=0

A> PRINTER P=O L=64 C="THIS IS A TEST" R=1 N=0 O=1

CHAPTER 8

The Quick Test

The Quick Test, QRUN.COM, is an excellent way to detect major errors in a very short amount of time. The entire Quick Test will take only about four minutes (at 2mhz) to run! It is great as the first program run at the start of each session.

The Quick Test is really three programs that chain together. They are "QRUN", "QDISK" and the standard CPU test. You will need all of these on the same disk if you are to use the Quick test.

Test Features:

- Performs a quick test of the entire user memory (TPA).
- Performs a quick non-destructive test of the disk drive currently in use.
- Performs the standard CPU Test.

How to Run the Quick Test

1. While at system level enter "QRUN". This will start execution of the quick memory test. You will see the following message on your screen:

```
SuperSoft Inc. Diagnostics II
Quick Test Copyright © 1981
```

```
Memory test
```

3. Once the memory test has completed, you will see the message:

```
Memory test complete
```

If errors are found, they are reported in the following fashion:

LOCATION	DATA WRITTEN	DATA READ
XXXXH	YYH	YYH

This format is the same used for the standard memory test. If errors do occur, then you should run the standard memory test to receive a more thorough diagnosis.

4. Once the memory test is completed the Quick Disk Test, QDISK.COM, will be loaded. You will see the following messages, displayed a few seconds apart:

Disk test
Read/write test
Random seek test

5. Once the Quick Disk Test is completed you will see the following message:

X read/write errors detected
X seek errors detected

Where X is the number of errors. If errors occur, then you should run the standard disk test in order to receive a more thorough diagnosis.

6. Then the standard CPU Test, CPU.COM, will be loaded and executed. Its operation and output were discussed in chapter 3.

R E A D E R C O M M E N T C A R D

We appreciate your comments and suggestions for ways to improve our documentation. Please use the form below, and attach additional sheets if you like.

Include page numbers wherever possible.

Points that were not clear enough:

Important details that were not discussed or were hard to find:

Other problems with the organization of the manual:

Positive aspects of the manual:

Thank you very much for taking the time to complete this form. Fold it in thirds, staple or tape it closed, and drop it in the mail. No postage is needed if mailed within the U.S.



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES



POSTAGE WILL BE PAID BY ADDRESSEE

Morrow Designs
P.O. Box 5755
San Leandro, CA 94577



Attn: Customer Service