DIAGNOSTIC PROGRAM
FLOPPY DISK SYSTEMS

DATARAM CORPORATION

DIAGNOSTIC PROGRAM
FLOPPY DISK SYSTEMS

Proprietary rights of DATARAM CORPORATION are involved in this subject matter and all manufacturing, reproduction, use and sales rights pertaining to such subject matter are expressly reserved. It is submitted in confidence for a specified purpose and the recipient, by accepting this material, agrees that this material will not be used, copied or reproduced in whole or in part, nor its contents revealed in any manner or to any person except to meet the purpose for which it was delivered.

TABLE OF CONTENTS

1.0 FLOPPY DIAGNOSTIC

The floppy diagnostic is designed to offer the user an easy and reliable means to test and verify the operation of any Dataram RX02 floppy system: FD-311, FD-311P, FD-511 and FD-511P. After running the diagnostic, any problem with a floppy disk drive or the floppy controller card should be known. The diagnostic also enables the user to format disks (single or double density) and clean the heads (with the aid of Ideck's Shugart-approved head cleaning kit).

1.1 Bringing It Up

1.1.1 Requirements

The minimal system to run this diagnostic consists of:

1) DEC PDP-11 or LSI-11 series processor
2) 28K words of memory
3) A Console terminal interface board and a console terminal
4) A LTC type of system clock (to control timeout intervals)
5) A F03/A,B,C,E or F33/A,B controller card with single or double sided floppy drive(s) as appropriate
6) A RX02 bootstrap device (the floppy subsystem can be used if it is working)

NOTE 1: If using more than one floppy controller, each must be assigned a unique vector and address; only one device can be the boot device.

NOTE 2: See section 1.3 for a description of miscellaneous commands available for the control of tests and error reports.

1.1.2 Booting

The program provides its own run time system and cannot be loaded under RT-11, RSX, or any other operating system. Booting requires loading sectors one and three on track one (the bootstrap routine) into memory and executing the routine.

1.1.3 Starting

Insure that the computer is in the "RUN" mode and turn on the LTC. Boot the diagnostic floppy disk in drive 0. If the boot disk was read (i.e., heads loaded for 5-10 seconds with audible or visible track seeks), but nothing is displayed on the console device, the difficulty is probably in the setting of the console device registers, which should begin at 777560.

When booted the screen will display:

1) FD diagnostic, version 2.7A -
   The diagnostic has been successfully booted. The boot disk may now be replaced by a scratch test diskette and another scratch test diskette may also be inserted into drive 1 to allow it to be fully exercised. The lower track limit is set at 20 initially, which provides sufficient room for both the diagnostic and the error log (adequate space for the logging of extensive testing of reasonably good units). If the boot disk is removed, and logging is not desired, the lower limit of testing may reset to 0. The scratch test diskettes must be good write enabled diskettes (with foil tape covering the write protect hole), and should be formatted by selecting and performing para. 1.2.3 Test 12 "Format Test". (Select command "0" and then test "12" in para 1.2.2.1 top level menu).

2) Terminal line length (72)? -
   This sets the maximum column position, beyond which the display will be continued to the next line. For most terminals, this is sufficient. The diagnostic does not produce a longer message without a <cr>. (Any value in parenthesis and after a query is a default - see section 1.3.3).

3) Is line frequency 60 Hz (yes)? -
   If answered no, the diagnostic assumes 50 Hz in determining timeouts.

4) Controller address ok (0177170)? -
   This selects the address that the program will use whenever it "talks" to the floppy subsystem. The default (0177170 octal) is the standard address for RX01 and RX02 floppy subsystems. If a floppy subsystem with another address is to be tested, input its address now. Note: If more than 28K of memory is present, the processor correctly maps this address to the real I/O page.

5) Interrupt vector ok (0264)? -
   This sets the vector that the program will expect the floppy subsystem to use whenever it interrupts. The default (0264 octal) is the standard vector for all RX01 and RX02-like floppy subsystems. If you are testing a device set at another vector, input that vector address now. The program only uses interrupts during the interrupt test.

When this last question is answered, the diagnostic examines the controller to determine whether an F03 controller is present. The RX02 bit (bit 14) is checked - this indicates that the controller is an RX02-like device and can write and read in double density mode. A check is made for the presence of double sided drives. Lastly, the drives are checked to determine which ones are on line and ready. The diagnostic has now been initialized, and the system is ready for option selection. In the event that the system fails to boot, refer to para. 1.4 for a simple ODT (Octal Debugging Technique) test to determine basic system failure.

1.2 Running

The default flow of control of the diagnostic is illustrated below.

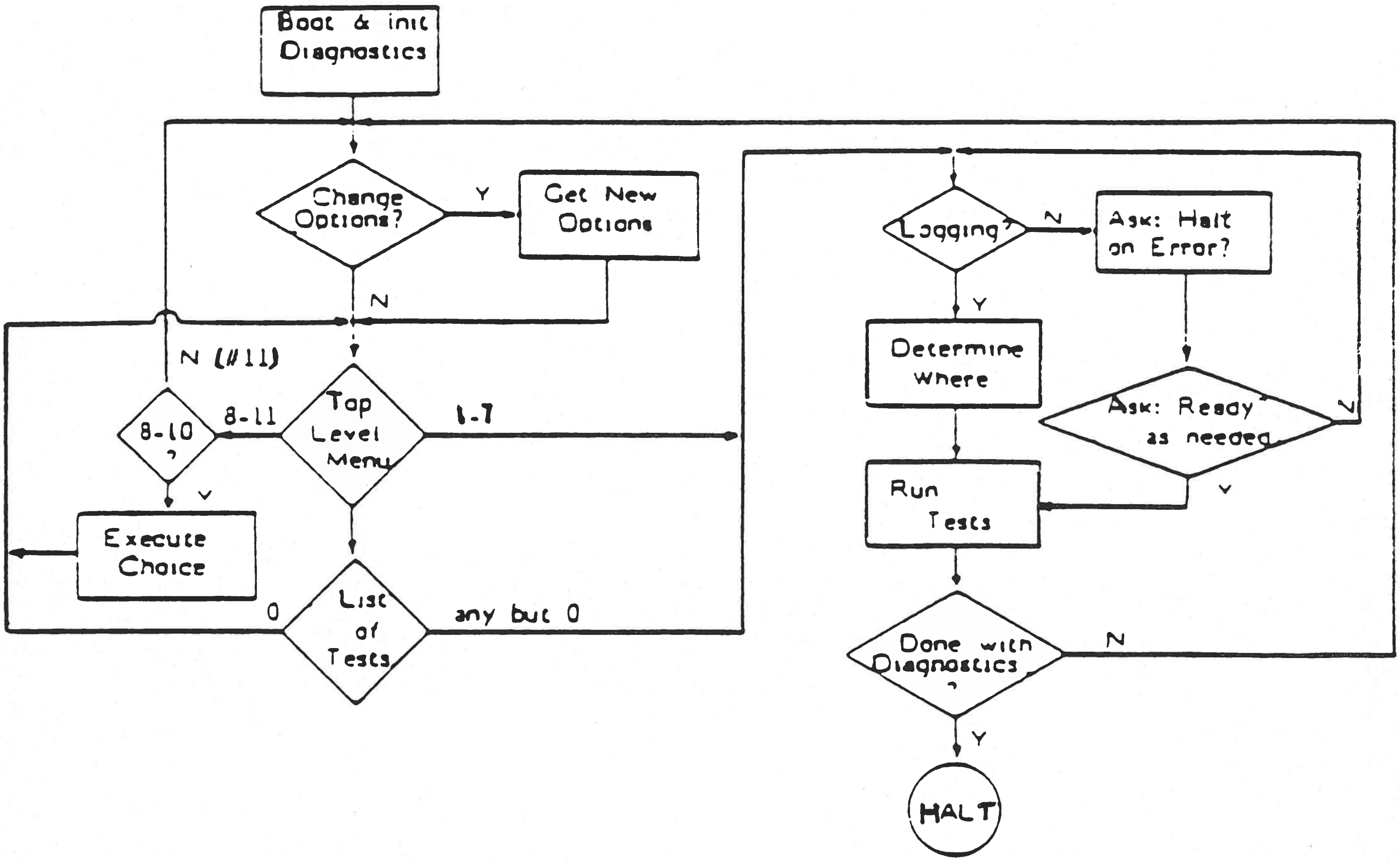

Figure 1-1

1.2.1 Diagnostic Options

Option changing permits one to optimize the diagnostic for the unit under test and the problem that is being investigated. As such, each time any of the following parameters are changed, the new value becomes the default on the next occasion that options are queried or changed. Selections following are initial, and are reset to those shown after each boot.

1) Test unit: 0, 1, or both (now 1) -
If more than one dirve is available to the chosen controller, one can choose which drive is to be exercised: drive 0, drive 1 or both (drive 0 and drive 1).

NOTE: This next question will display only when testing double sided drives.

2) Test side: 0,1, or both (now 0) -
If the chosen controller responded like an F03/B or F03/E (or F33/B) double sided controller, which side to test can be chosen: side 0 (the default side for single sided drives), side 1 or both (side 0 and side 1). Only side 0 of a single sided disk can be accessed in a double sided drive.

3) Test density: 1 single or 2 double (now 2)? -
This sets the density at which disk reads or writes, and DMA transfers will occur. Clearly if diskettes of the wrong density are used, many density errors will occur.

4) Print track number each time sector one is accessed (no)? -
This causes the program to print the track number any time the sector 1 is accessed on any track. This is helpful both in keeping a close tab on head position and in providing more rapid feedback about the progress of the test. The option will, however, slow down the execution of some tests.

5) Track range on test floppy: Low is 20 -
This sets the lower limit of access. Tests which use one track use this value. Thus to run the Quick W/R test on track 29, set this to 29. Selecting a larger value (65 to 70) limits the test range to the innermost track, which is the most difficult part of the diskette to decode (due to high bit density and phase shift).

6) Track range on test floppy: High is 76 -
Similar to item 5. Band testing is implemented by setting the high value ten greater than the low. The high value may be equal to but not less than the low.

1.2.2 Selection of Diagnostic Test

The diagnostic tests are divided into two levels: a top level menu (Figure 1-2) which is a collection of tests selected from the bottom level menu (Figure 1-3).

To execute a top level diagnostic test, select one of the following tests (1-8,9). Two questions will now be asked:

1) Error logging to floppy (no)? -
Errors and status reports can be recorded on a floppy disk. This is useful for examining a test run which had many errors, or for extended testing, when the pass/fail messages will scroll off a video screen. The logging diskette must be a write-enabled double density formatted diskette (no initialization required). The logging section asks for a controller address and unit. The default address is 0177170, but if a functional controller is present at another address, it can be used, independent of the controller under test. The default logging unit is zero - all logging is to side zero of the logging diskette. The log area is above the area occupied by the diagnostic but below track twenty. If the logging is to be a diskette also under test, the lower limit on the track range should be set at 20 or above.

2) Halt on error (no)? -
Whenever an error occurs, information about it is displayed on the terminal, and the selected test continues. Halting on error allows one to study each error as it occurs. After examining the error message, typing a

carriage return (<cr>) continues the test. The question is not asked if errors are being logged.

The specified tests begin upon answering the above question. Pass or fail messages are produced as appropriate. When the selected test completes, 'Done with diagnostic no ?' appears. Typing<cr> will lead to options and then to the main loop, while 'n' will stop the program with 'Main returns<cr> HALT, WILL NOT RESTART'. If the program produces some other equally serious message (or this message at any other times), then a fatal error has occurred.

1.2.2.1 Top Level Menu

The menu is shown in Figure 1-2, with the test actually invoked shown for items 1 to 7.

	Command	Runs
0)	List of tests	(Test list chosen by user)
1)	All tests	1 - 18
2)	Quick tests	5, 10, 11
3)	Read only tests	1, 2, 6, 7, 11, 16
4)	Random exerciser	17
5)	Drive test	3, 4, 6, 7, 8, 14, 15, 16
6)	Board test	1, 2, 5, 10, 11, 13
7)	Overnight test	3, 4, 6, 8, 17
8)	Clear errors	
9)	Print error log	
10)	Clean heads	
11)	Change options	

FIGURE 1-2

The commands are:

0) Bring up the list of tests (default).

1) Run all the tests. All the standard test combination are in numerical order.

2) Quickly (less than 20 seconds) determine whether or not the machine works at all.

3) Run all the read-only tests which do not expect to find any special pattern written on the diskette (included are tests which only exercise controller buffer memory).

4) Call the random exerciser with a large run count.

5) Run those tests which principally depend on drive characteristics.

6) Run those tests which principally depend on controller card characteristics.

7) Use the best extended-period tests. These repeatedly test basic functionality of the system, and should not generate more than one error on a good machine with approved diskettes for each full night of testing.

8) Clear all run and error counts. Any testing which follows begins with a clean slate; the result of an extended series of tests is clearer.

9) Retrieve and print the error log. The diagnostic asks whether or not to paginate output (i.e., print it in pages of 23 lines or just print it). If the output of the log is not paged, the rate of feed can be controlled by the use of ^S and ^Q (XOFF & XON). Note that once a new log is started, the old (if any) becomes unrecoverable to the diagnostic.

10) Runs the head cleaning utility. With an approved cleaning disk, one may clean the read/write heads of a disk drive. The routine will ask that such a cleaning of disk (Innovative Computer Products prepared, be inserted into the appropriate drive (per options), and will load the heads on all tracks. No errors are reported, and the heads should be thoroughly clean when done. For most installations, cleaning once a month is sufficient. By running the heads on all tracks, wear on the diskette is minimized, thus extending its useful life. When done, the user is prompted to replace the diskette with one containing real media.

11) Explicitly change the test option.

1.2.2.2 Bottom Level Menu

To execute a test (or tests) from this menu, you must specify a sequence of the tests you want. This should be a line of comma separated test numbers and/or an inclusive range of test numbers separated by dashes. (Example: 10, 12, 3-6, 8 will execute test 10, 12, 3, 4, 5, 6, 8 in that order).

After your selection has been made, the following questions will be asked:

1) Number of times to execute test or sequence (1)? -
The selected test(s) will execute this number of times, and in the order specified. If testing of either or both sides or both units (or both, which causes each named test to be run four times) is opted, a message delineating the actual parameters for each instance of a test are printed.

2) Error logging to floppy (no)? -
Errors and status reports can be recorded on a floppy disk. This is useful for examining a test run which had many errors or for extended testing, where the pass/fail messages will scroll off a video screen. The logging diskette must be write-enabled and double density formatted (no initialization is required). The logging section asks for a controller address and unit. The default address is 0177170, but if a functional controller is present at another address, it can be used, independent of the controller under test. The default logging unit is zero - all logging is to side zero of the logging diskette. The log is above the area occupied by the diagnostic but below track twenty. If the logging is to a diskette that is also being used for testing the lower limits on the track range should be set at 20 or above.

3) Halt on error (no)? -
Whenever an error occurs, information about it is displayed on the terminal, and the selected test continues. Halting on error allows one to study each error as it occurs. After examining the error message, typing a carriage return (<cr>) continues the test. The question is not asked if errors are being logged.

4) Ready (yes)? -
Typing a <cr> will start the test (or sequence); otherwise, the above questions are repeated. The query is suppressed after the first instance in a run if there is to be more than one pass or if a menu of tests was chosen; else it appears before each individual test is run.

The specified tests are now run. Pass or fail messages are produced as appropriate. Whn done testing, 'Done with diagnostic (no)?' appears. Typing <cr> will lead to options and then to the main loop, while 'n' will stop the program with 'Main returns <cr> HALT, WILL NOT RESTART'. If the program producrs some other equally serious messages (or this message at any other time), then a fatal error has occurred.

1.2.3 List of Tests

1) DMA test
2) Padding test
3) Basic write test
4) Basic read test
5) Quick W/R test
6) Seek test
7) Self test
8) Sector uniqueness
9) Uniqueness verify
10) Interrupt test
11) Initialize test
12) Format test
13) Write protect
14) Deleted data test
15) Overlap test
16) Readability
17) Random exercises
18) One track

FIGURE 1-3

The complete list of tests (of which the menu items 1 to 7 are composed), with a description of purpose, method, and possible errors is listed here. Also included are pass or fail messages and approximate running time (where appropriate, the times are for a track range equal to floppy capacity). Run times depend in part on head positioning time (somewhat faster for double sided drives) and on the speed of the processor. All should be within 30% of those stated. In all of the following, singly or doubly quoted messages are approximately what the diagnostic actually prints.

The tests included are:

1) DMA test -

   A test which exercises the ability of the controller to transfer information between the computer's memory and the controller's buffer. It is accomplished by transferring selected dtat to and from the controller with a check for identity on the data. In between each transfer the diagnostic buffer is filled with a counting pattern. Success is indicated by passed DMA test. Running time is approximately thirty seconds.

   Possible errors -

   a) DMA test: transfer problem on subpass XX - XX is the word count for this transfer, which failed.

   b) DMA test: YY words wrong out of XX words moved - again, XX is the word count; the transfer succeeded but the controller changed the data in YY words. The buffers can be inspected if desired.

   c) DMA test: ZZ failed transfers (XX trys) - ZZ is the aggregate sum of all the instances of item a, while XX is 64 or 128, depending on the density selected. It is printed at the end of the test if errors were encountered.

d) DMA test: total of ZZ data miscompare errors - ZZ is the aggregate sum of all the YY's generated in item b. Like item c, this is printed at the end of the test.

2) Padding test -

A test which checks to see that on partial transfers the controller zeros out the balance of its buffer. The test is run for all possible partial transfers. As above, between steps internal buffers are filled with a counting pattern (equal to the index of the word). Success is indicated by passed Padding test. Running test is approximately thirty seconds.

Possible errors -

a) Padding test: error at word XX, whose value is YY (0 correct) - on a partial transfer, not every word above the word count was zeroed out. The index XX of the first error is printed, along with the value YY generated by the controller (in octal).

b) (XX more errors on subpass YY) - after a failure in a given subpass YY, the additional errors are counted and printed (XX).

c) Padding test: net conflicts this subpass = XX - the sum of words transferred incorrectly with those not zeroed. Where greater than item b, the DMA test should be run (compare-type errors are tallied but not reported).

d) Padding test: total errors reported = XX - the sum of item c messages.

3) Basic Write test -

A test which writes standard patterns over the range (per options) on the diskette. Standard worst case patterns for various links in the floppy subsystem are used. No verification of writing occurs at this time. The test is designed to operate with the Basic Read test.

The hardest place to write is above track 65. Passing is indicated by passed Basic Write test (sectors written = XX), where XX is the number of sectors in the track range (tracks multiplied by sectors per track). Failure causes the message failed Basic Write test: errors = YY out of XX trys, where XX is as above and YY is the sum of sectors not written. An internal map is maintained on these failures, so that the Basic Read test can skip reading incorrectly written sectors. Execution time is approximately seven minutes.

If a successful write is obtained, this diskette can be loaded in another machine, and the Basic Read test run with the same track range. In this way, a thorough read test of another unit can be made without depending on any of its own write electronics.

4) Basic Read test -

A test which reads over a selected area of a floppy disk and checks for the presence of standard data patterns. It should generally be run in conjunction with the Basic Write test. Execution time is approximately seven minutes. Passing is indicated by passed Basic Read test (sectors read = XX) - XX is the number of sectors actually read, exclusive of item a below.

Possible errors -

a) Basic Read test: skipped reading XX sectors not written correctly - XX is the sum of skipped sectors from the internal map maintained by the write test. It is not an error but rather a statement of a limit.

b) Failed Basic Read test: XX read errors of YY ok sectors YY is the number of sectors the test was asked to read less the number marked internally as not correctly written. XX is the number of errors encountered in reading correctly written sectors, whether from nonrecovery or miscompare.

5) Quick W/R test -

A short version of the complete write and read tests which uses only one track (the lower limit). Basic functionality of the floppy system is established here. Passing is indicated by three messages, one each from the write test, the read test, and this test. Running time is approximately fifteen seconds. The messages produced are from tests 3 and 4.

6) Seek test -

A test which issues seek commands to the controller in various manners so as to exercise the head positioning mechanism. Since the drive does its own seeking, a seek is defined as a read sector. Therefore, for the test to be meaningful, the test unit must contain a good diskette (all sectors in the track range readable).

The subpasses include:

a) Seeking back and forth (bouncing) over an increasing or decreasing range with one limit (upper or lower track) fixed, for a total of four variations.

b) Seeking as above, but with both limits changing (either approaching each other or reading).

c) Seeking ahead a few more tracks than back, thus moving the head over the entire diskette (at various paces).

d) Seeking to the current track (which should require no head movement).

When run with the track range set at the limits of the floppy, over one thousand seeks are requested. Not every subpass is run if the difference in low versus high range is too small - the subpasses so affected and the cutoff are indicated below. Success is indicated by passed Seek test. Execution time is approximately six minutes.

Possible errors -

a) Seek test: fail increasing interval low to fixed high - the seek pattern is a seek to the upper limit of the track range followed by a seek to a variable (and decreasing towards the lower limit of range) track.

b) Seek test: fail forward two, back one - starting at the low limit, the seek pattern is forward two tracks, back one. If the range difference is six or less, the subpass is skipped.

c) Seek test: fail increasing interval fixed low to high - the pattern is as in item a, except the lower limit is fixed, rather than the upper.

d) Seek test: fail ends to middle - from the upper and lower range limits, each seek is towards the opposite extreme, each extreme is bumped towards the middle after each use.

e) Seek test: fail decreasing interval high to fixed low - the pattern is as in item a, except the seeks are progressively closer together, and the lower limit is held fixed.

f) Seek test: fail decreasing interval fixed high to low - the pattern is as in item e, except the upper limit is fixed instead of the lower.

g) Seek test: fail seek to current track - the pattern is a series of seeks to one track with no other positioning.

h) Seek test: fail middle to ends - as in item d, but the seeks start at the middle of the range and work outwards towards range limits. The subpass is not run for range differences of ten or less.

i) Seek test: fail forward three, back two - as in item b, but with three and two instead of two and one. Skipped if the range difference is ten or less.

j) Seek test: failed all subpasses - all subpasses actually run were failed.

k) Failed Seek test - if any but not all of items a - i are printed, then so is this.

NOTE: In items a - i above, only the first message carries the label Seek test: - the balance (if any) are printed without the test name.

7) Self test -

A test which invokes the microcode self test routine, which in turn reads the entire diskette and is passed or failed based on the setting of the ERROR Bit in the controller register. The test requires good media to pass; where this in question, the Readability test may be more appropriate. The self test is the most rapid way to check readability of properly formatted media. Note that on double sided drives, the controller will read both sides of a double sided disk independent of the setting of the Side Bit. Therefore, setting the side to test as both in options will cause precisely the same event to happen twice. If the controller does not accept the self test Function Code (to set up the Self test), a message to the effect is generated. Passing is indicated by passed Self test (media good). Running time is approximately 25 seconds.

8) Sector uniqueness -

An exercise which tests the ability of the controller to distinguish all sectors uniquely. This is implemented as follows:

a) For each sector of each track in the test range, write its track and sector address twice (using the first four byes, respectively).

b) Between writing tracks, verify that the just-written track contains the correct uniqueness marks (to insure the accuracy of the write). Sector misalignment is detected here.

c) After all sectors in all tracks in the test range have been written, each sector is read back to verify the uniqueness of what was written. Since sector alignment was previously verified, only track misalignment errors should be discovered here.

Once a floppy has been written (and before another writing test is run), re-running this test essentially becomes an instance of test 9: failed writes will not be detected since the marks are already present. Success is indicated by passes Sector uniqueness. Execution time is approximately fifteen minutes.

Possible errors -

a) Sector uniqueness: bad write @ track XX, sector YY The controller flagged an error on write at the indicated location. The message is printed just after the write fails. No further attempts are made to use the sector.

b) Sector uniqueness: mark not readable @ track XX, sector YY - the controller flagged an error (where indicated) for failing to read back the just-written track while looking for marks. No further attempt is made to read the sector.

c) Sector uniqueness:

```
Wrong mark:       want      got      (got)
track             X1        X2       (X3)
sector            Y1        Y2       (Y3)
```

While reading the just-written track the mark at the indicated sector (which is the same as the want mark: X1, Y1) the correct value was not found. The two got columns are the value s actually found, where any value retrieved which is not a possible track or sector address is printed as -1. (The second got column is for the second pair). No further attempt is made to read the sector.

d) Sector uniqueness: XX bad sector writes - the sum of all message a reports.

e) Sector uniqueness: XX unreadable sectors - summary of all b messages (not counting bad sectors noted in item a, which were not tried).

f) Sector uniqueness: XX wrong marks - summary of all messages from c, less any from a or b.

After any or all of the above messages have been printed, test 9 is invoked to read all that was written (i.e., check track alignment). The pass or fail message comes after test 9 returns.

9) Uniqueness verify -

A test which performs the verify portion of the sector uniqueness test. There are two purposes to the test: first, it serves as the check portion of the test 8; second, after a floppy has been written with the sector uniqueness pattern, it can be removed and placed in another machine in order to test the machine's ability to read sectors uniquely, even if it cannot so write them. If any sectors were skipped or marked as bad in the writing portion, then that diskette is not suitable for use with this test and/or in another machine.

Success is indicated by passed uniqueness verify. Possibly this will be accompanied by a message about the number of sectors skipped because of being marked bad in the other half. Should every uniqueness mark be labelled as unreadable, this test is considered failed. Execution time is approximately five minutes.

Other errors -

a) Uniqueness verify: mark unreadable @ track XX, sector YY - an actual read at the indicated location failed. If the message occurs immediately after test 8 has been run, there is a more serious error, as the sector was readable before but is not now.

b) Uniqueness verify: got wrong mark

```
want              got    (got)
track X1           X2    (X3)
sector Y1          Y2    (Y3)
```

While reading the mark at the indicated location (which is the same as the want mark: track X1, sector Y1) the correct value was not found. The two got columns are the values actually found, where any value retrieved which is not a possible track or sector address is printed as -1.

c) Uniqueness verify: XX bad reads - summary of all messages from a.

d) Uniqueness verify: XX bad marks - summary of all messages from b.

10) Interrupt test -

A test which determines whether the controller correctly asserts an interrupt on DONE, and at the correct vector. Interrupts which occur at any other time than during this (short) test are counted as errors. Running time is approximately one second. Success is indicated by passed Interrupt test.

Possible errors -

a) Interrupt test: did not interrupt on fill buffer - as indicated, the controller executed a fill buffer command but did not interrupt after setting the DONE bit of the control register.

b) Interrupt test: did not interrupt on empty buffer - as in item a.

c) Interrupt test: did not interrupt on read sector - as in item a.

d) Interrupt test: did not interrupt on write sector - as in item a.

e) Failed Interrupt test - at least one of the above occured.

f) Interrupt test: complete failure to interrupt - something yet more drastic has occured. All of items a - d should have preceded this message. To be sure of the result, the diagnostic should be rebooted and the interrupt test run first.

11) Initialize test -

A test which issues an INIT to the controller and checks that the controller is in its initial state. This is defined as follows: Initialize Done is set in the data/status register; side, Unit and Density Bits all set to zero in the control register; RX02, DONE, and possibly ERROR Bits set (the last is flagged) in same. At the end of this test, the side, Unit and Density Bits are reset according to the current options. Success is indicated by passes Initialize test. Running time is approximately one second.

Possible errors -

a) Initialize test: bad initialize (should be 04040, is XX) - the control register came back incorrectly set. The value listed are in octal.

b) Initialize test: did not set initialize Done Bit - the ID Bit in the data/status register was not set after completion of the initialize. This condition is not reported if item a is reported.

c) Initialize test: initialized with Error Bit set - some error was detected by the controller on initialize. The error code is printed.

d) Failed Initialize test - if any of itmes a - c are printed, then so is this.

12) Format test -

A test which determines the ability of the controller to write headers and data fields. Writing headers is optional, but if elected, then the data fields are also written, as the density mark is written at that time. Verification of correctness consists of reading every sector via Self test. Optionally, one may choose to read every sector checking to see that the data there is zero. This is a stronger passing condition, but takes considerably longer to execute (about four minutes instead of one and one half minutes). If Self test is run and fails, test for readability. Passing is indicated by passing the Self test (or all sectors zeroed) after formatting and passed Format test.

Non-readable sectors (Preamble/Header not found) probably indicate a failure in the writing of headers. The presence of such errors might call for the Readability test. Note that as with the Self test, the setting of the Side Bit is immaterial to the controller during formatting, so setting the side to test as both in options is redundant.

It is inadvisable to format a diskette manufactured for single density at double density, as the oxide is probably not of adequate quality to retain the greater bit density. Double density disks may be formatted at either density without difficulty. Trying to format the second side of a single sided disk is also a poor idea. Lastly, formatting the headers of a write-protected disk causes head motion but no writing.

Possible errors -

a) Format test: disk headers not formatted - an error was returned by the controller while trying to format the diskette headers. It should not be used until reformatted.

b) Format test: disk data fields not written - an error was returned by the controller while trying to write the data fields and density marks. As above, the diskette should not be used until correctly formatted.

c) Format test: XX bytes not zeroed @ track YY, sector ZZ - if the longer verify is elected, and a sector which is not all zeroes is found, the message is printed as shown.

d) Format test: XX total nonzero sectors - the sum of all item c reports.

13) Write protect -

A test which checks the write protect mechanism of the drive. This is accomplished as follows:

a) The diagnostic asks for a write-protected disk to be installed in the appropriate drive (as chosen in options), and waits for a carriage return before continuing with the test.

b) The data on the test sector is read into an internal buffer (failure on this or any other transfer terminates the test).

c) The diagnostic creates some data (which should vary on each call so that the test is not passed on a fluke, and which is different from what is presently in the test sector), and then tries to write the data on the test sector.

d) If the write protect error code is returned, and the data on the diskette is as it was in step b, the test is considered passed. Otherwise,

e) The data in the test sector is examined. The nature of what is found is returned as an error message (see error messages e, f, g below)

f) The diagnostic asks for a write-enabled disk, and waits for a carriage return to by typed.

For units with a write-protect switch, setting the switch to protect is equivalent to installing protected disk. Complete failure (i.e., executing normal write with no error code) might mean the photodetector is covered with dirt or a stray write-enable tab. Success is indicated by passed Write protect. Running time is approximately ten seconds.

Possible errors -

a) Write protect: failed initial sector read - a terminal error, as the diagnostic verifies the data independent of the setting of the error code.

b) Write protect: transfer failure before write - the controller did not accept transfer of the test data. Since the contents of the controller's buffer are now what was read in step a, continuing the test is meaningless: there is no way to discern whether or not it actually writes.

c) Write protect: error code not set - the controller did not set the code. If a write-enabled disk is installed, this is to be expected.

d) Write protect: unable to recover test sector after write - the result of the attempted write cannot be learned. Since there then can be no check on the data, this ends the test.

e) Write protect: controller executed normal write - the data read back after the attempted write is the data of the attempted write. As is step b, this is the correct action of a write-enabled disk is installed.

f) Write protect: controller did not write - the data in the test sector was not changed by the attempted write. The message is only printed if no error code was signalled (as in item b above).

g) Write protect: controller wrote garbage - the data in the test sector is neither what was there before the test nor what was in the attempted write.

h) Failed Write protect - if any of items a - g are printed, then so is this.

14) Deleted data test -

A test which determines the ability of the controller to write and read, and remove deleted data marks from a diskette.

The test is implemented as follows:

a) Write a track with deleted data marks

b) Reads this track, checking that the Deleted Data Mark bit is set.

c) Rewrite the track without marks.

d) Reread the track to see that no mark remains.

Note that most operating systems do not make use of this feature. Failure comes in two forms: a mark written could not be recovered, or a mark written and then erased is still present. The former are tallied and placed in the fail message (failed Deleted data test: missed rereading XX marks), while the latter are enumerated in each sector read. Success is indicated by passed Deleted data test. Running time is approximately thirty seconds.

15) Overlap test

A test which coarsely examines the possibility of sector overlap on write at the beginning and end of each track. This is a special case of the Basic Write/Basic Read test and is useful when errors consistently appear at the first or last sectors of various tracks. It is not expected that the test will be frequently useful. Success is indicated by passed Overlap test. Execution time is approximately one minute.

Possible errors -

a) Overlap test: bad write, track XX - either the low or the high sector on the indicated track was not written properly (signalled by the controller). The track is skipped in subsequent testing.

b) Overlap test: read error, track XX, low sector - the indicated data could not be recovered.

c) Overlap test: read error, track XX, high sector - the indicated data could not be recovered.

d) Overlap test: bad data check, track XX, low sector - the indicated location did not contain the data originally written there.

e) Overlap test: bad data check, track XX, high sector - As in item d. Note that the presence of item d or e on one properly-written track is precisely the condition the test is looking for.

f) Overlap test: XX write errors - the sum of all item a notes.

g) Overlap test: XX read errors - the sum of all reports from b and c.

h) Overlap test: XX apparently overlapping tracks - the union of items d and e (i.e., at most one error per track).

i) Failed Overlap test - if any of items a - h are printed, then so is this.

16) Readability -

A test which just tries to read all sectors on every track in the track limits. No attempt is made to transfer the data out of the controller (and so of course no check of the data is made). Only errors flagged by the controller are reported. The Self test does more or less the same thing, but on error it aborts with no indication of the location of the error. The test reports where sectors are unreadable (and why), along with not giving up before completion of the appointed number of reads. Success is indicated by (passed Readability). Execution time is approximately one minute.

Possible errors -

a) Readability test: bad read @ track XX, sector YY, where XX and YY are the track and sector numbers respectively. On the next line, the error code and meaning are printed.

b) Failed Readability test: XX read errors out of YY trys is printed. where XX is the number of individual sector messages printed and YY is the number of sectors on which a read was attempted.

17) Random exercise -

A test which picks random tracks within user limits, random sectors on those tracks and writes (and reads) random data. Between write and read, the controller's buffer is cleared.

The test is designed to simulate irregular (random) real use. After about 300 sectors have been written/read back, the test is exited. Success is indicated by passed Random exercise. Execution time is approximately five minutes.

Possible errors -

a) Random exercise: hardware error (message) on pass XX - some failure (indicated by the message) was encountered. A standard error report is also generated at this time.

b) Random exercise: YY data miscompare errors - despite success in every transfer, the data came back changed. The user may examine the data, though interpretation may not be immediately apparent. In any case, the test can be continued.

c) Random exercise: XX total hardware errors - the sum of all XX's in items a above - this is printed at the end of the test.

d) Random exercise: YY total data miscompare errors - the sum of all YY's in item b above - this is printed at the end of the test.

18) One track -

A test which reads or writes one track continuously. It first asks whether to read or write, and then what track to start at. Each time sector one is accessed, a message about the track number is printed (which message can be turned off or on). The test is useful primarily in conjunction with an oscilloscope to verify timing and head positioning. No errors are reported. The test does not in and of itself ever terminate, but rather can be exited by ESCape or e. Typing single command characters (which are not printed on the screen) modifies behavior as follows:

a) d or D - shift down one track (i.e., smaller track value), subject to range restriction (beeps if movement is not permissable).

b) e or E - exit test.

c) m or M - move to a new track (track is prompted for, and must be in range).

d) p or P - print track value each time sector one is accessed.

e) r or R - print the current track and track range.

f) s or S - silence track announcement at first sector.

g) u or U - shift up one track (larger track value), subject to range restriction (beeps if movement is not permissable).

h) ' ' or <cr> - print current track value.

1.2.4 Choosing Tasks

Items in the top level menu which are not just sequences of tests are executed immediately. Otherwise, the list of tests shown in figure 1-2 are run. At the level of individual tests, a sequence can be specified. The line should be a comma-separated list of test numbers. Several sequential tests can be indicated by putting a dash between the end valued. Embedded spaces are ignored. Example: 10, 12, 3-5, 7 will run tests 10, 12, 3, 4, 5, and 7 in that order. These are the only two methods of directing the diagnostic to do anything.

1.3 MISCELLANEOUS

1.3.1 Other Commands

Any of these can be typed at any time, with effects as described.

1) ESC - escape from current activity and restart
This will halt whatever test or other action is occuring and brings the user back to the "change options" level of the diagnostic. It is an asynchronous operation.

2) Delete - remove last-typed character from input line
This delete the last character typed. It displays the deleted character(s) as RT-11 does: the first occurrence of a delete causes display of a "/" and the character printed just before, with subsequent deletes displaying the characters typed (in reverse order) up to the beginning of the line. When another characrer is typed, another "/" is displayed with the new character (i.e., CRDLA/AL/S is input as CRDS). If nothing is printed after typing a delete, either output has been inhibited by ^S or there are no input characters on the current line.

3) ^U - remove all of current input line
(Entered by typing the "CTRL" (control) and "u" (or "U") key at the same time). This deletes the current input line. Only a new line is displayed.

4) ^S - Stop output
This halts all transmissions to the console terminal. It can be used to allow one to read information even if it is being transmitted quickly and might otherwise scroll off the (video) screen.

5) ^Q - Restart output
If output has been stopped with a ^S and a character is ready to be sent to the console, the program will wait for a ^Q to be typed (and thus further testing is suspended).

6) ^O - Ignore output
This causes the diagnostic to discard all output. The diagnostic continues to run, but note that 'all output' includes all prompting. It is useful when too much output is being produced, especially if a printer is used as the console device. Errors (should they be occuring fast and furious) are still correctly recorded in the summary. Typing any other nonspecial character cancels this. However, the character typed will be used as input by the program except for the following.

7) ^P - Resume outputting the output
Print all further output on the console device (only effective after ^O).

8) ^C - Summarize errors
A summary of the errors noted so far can be generated by this (unless already in the process of generating one). ^C stops (on a restartable basis) whatever is in progress, and asks how the errors should be broken down. The errors can be subdivided by occurence on unit or side, by above or below track 65 (higher tracks are harder to read or write), or simply a sum of all errors. Tests which do not read or write the diskette do not incur errors on the above/below counts. Hence, the sum of these two may be less than the sum of all errors. In any case, the error count, run count, and test name (for all tests) are printed out. Next, some global state information is printed out. Lastly, one can get a listing of the errors signalled by the controller, with the same choices of breakdown (the default is whatever mode was chosen above). The error summary can be obtained at any time. Note that since the summary interrupts anything, if a prompt had been printed just prior to initiation of the above, then the prompt is lost. Summaries are not logged.

1.3.2 Error Reporting and Handling

Error reports as generated by the diagnostic have a standard form, of which Figure 1-4 is a sample.

```
Error in pass 1 of test 3 (Basic Write test) at 1:17:29

Error code 0120:  A preamble could not be found

Test unit one, side zero, double density, at track XX, sector YY

Command:  0425 (Write a sector to floppy)
Control Data
before 04620   040
after 04460    040
```

FIGURE 1-4

The report in Figure 1-4 is interpreted:

1) Pass in the pass (set at initiation of the test run) running when the error occured. Test and name mean the one actually running. (Tests which call other tests update the test number). The time is the interval in hours, minutes, and seconds since either the diagnostic was booted or errors were last cleared.

2) The error code (if any) is printed and interpreted.

3) Where the error occured (track and sector are printed only if the test reads or writes the floppy).

4) The command is the last control word sent, with expansion of the Function Code.

5) These are the sampled values of the floppy registers before and after the command which caused some sort of exception.

In general, an error report is produced as each low level error is detected (depending on the nature of the test).

The diagnostic measures the duration of activity by the controller while the latter executes its function. If, after a period considered to be very long compared to the normal time needed to complete the operation in question, the controller has not yet signalled DONE, then a timeout fault has occurred. It is interrupted by asserting the Initialize Bit, and the message 'Timeout failure on DONE Bit' is printed. The operation in question is not retried. The TRANS Bit is watched in a similar manner.

The diagnostic checks for consistency between sent and returned values of unit, side, and density (where choice exists). Disagreement provokes a line to the effect, with what each is set at. Execution continues.

Spurious interrupts (i.e., those at the vector but while the interrupt was not enabled) cause the message '>>Controller interrupt<<'. As above, an error is noted, and execution continues.

A summary of errors can be obtained by typing ^C (section 1.3.1). In general, errors are not retried. Where a write failure occurs and some special pattern was to be written, no read attempt should be made on the sector (the mechanism which insures this can be foiled with effort). Interpretation of errors is always a context sensitive notion. Repeatable errors to a specific sector on a specific diskette (but no other) are clearly not grounds for considering the subsystem to be defective. Sporadic errors, i.e., nonrepeatable errors scattered over both time and an apparently good diskette (both of which are most readily learned by looking at a log of individual error report), are more indicative of a marginal device.

1.3.3 Details

Except as noted, input and output are in decimal. On output, a leading zero is used to signal octal information. Input of an octal value does not require a leading zero. Supplying a value outside of some acceptable range causes the diagnostic to print what is considers to be an acceptable range and then asks for another try. For yes or no questions, any of Yes, YES, yes, Y or y are taken as yes, with a similar group for no meaning no. Values in parentheses are defaults. That is, if a (cr) is typed, the value shown is taken as the intended response to the query. The diagnostic interleaves accesses to sectors, that is, does not read or write in numerical order. This only becomes apparent when several errors occur on one track.

1.4 If The System Does Not Boot

In the event the system fails to boot, the operator can revert to ODT (Octal Debugging Technique) to determine faults in the CPU system. The following simple procedure utilized ODT for some basic system diagnostics.

a) Operate RUN/HALT switch to HALT.

b) Depress the INIT switch.

c) One or more 173000 <CR> <LF> "@" should appear and the LED indicators on both drives should flash once.

IF NOT

1) Check that power is applied to the computer.

2) Check for correct location of boards in the computer. See appropriate unit instruction manual.

3) Check to see that the Serial Line Interface Board and the Terminal are set to the same baud rate. See DEC Microcomputer Interface Manual.

4) Ascertain that the cable from the floppy drives to the controller board is connected.

5) Ascertain that the Processor is set to start at address 173000. See Microcomputer Manual.

6) Check that the Memory is set up properly. See appropriate Memory Configuration Sheet.

7) Has the bootstrap jumper been installed on the F03 or F33?

8) Is more than one bootstrap enabled?

9) Does the system have sufficient memory for the diagnostic program to execute? It requires at least 24K words.

d) Perform the following Memory/Register tests using ODT

1. Set the RUN/HALT switch to HALT.

2. Set the LTC switch to OFF.

3. Depress the INIT Switch.

4. Enter the following program:

Memory Location	Memory Contents	Load Memory
4/	XX XX XX	6 <LF>
6/	XX XX XX	0 <LF>
10/	XX XX XX	10011 <LF>
12/	XX XX XX	20021 <LF>
14/	XX XX XX	1775 <LF>
16/	XX XX XX	0 <CR>
@ R0/	XX XX XX	0 <LF>
R1/	XX XX XX	20 <CR>
@ R7/	XX XX XX	10 <CR>
@		P (Program starts
@	00 00 10	(Program stops)
R1/	16 00 00	

NOTE: If the interactive program halts at Location 16, a memory failure has occurred. Check memory board for correct switch settings for starting address and board capacity).

e) For further use of ODT (Octal Debugging Technique) for running and debugging programs, reference the DEC "Microcomputer and Memories" manual.

DATARAM CORPORATION

Princeton Road
Cranbury, New Jersey 08512
Tel: 609-799-0071 TWX: 510-685-2542