

XEBEC SYSTEMS, INC.

XDF-50

DISK FORMATTER DIAGNOSTIC

FOR

DATA GENERAL NOVA SERIES COMPUTERS

Approved By: CJ Hurd

Date: February 8, 1974

P/N: 310073

Revision: B

PROGRAM DESCRIPTION

The disk diagnostic program enables the user to 1) exercise all positioning and data transfer functions provided by the formatter, 2) check all status and error flags returned by the formatter, 3) verify proper operation of all units connected to the formatter, 4) verify proper operation of the coupler functions, and 5) check that part of the computer interrupt system used by the formatter. Some of the tests included in the diagnostic are designed to isolate errors which depend on the data content or length of a data transfer. Another set of tests are devoted to forcing error conditions and checking that they are properly reported by the controller. Tests are also provided to check for proper sector formatting and unit to unit compatibility.

Error reporting is accomplished via the console teletype. In general, errors are reported when either status information or data input from the formatter differ from that expected by the program. The operator has the ability to inhibit error message outputs as required. In addition, the operator specifies the order in which tests are performed and the sequence in which certain tests select units.

The program operates in a minimum memory configuration of 4K but will automatically utilize extra memory to test DMA addressing. The upper 200₈ words of memory are reserved by the program for loaders and are not overwritten.

OPERATING INSTRUCTIONS

The diagnostic executes in a sequence of well defined steps that are described below. All communications between the operator and the program are via the console teletype and, in some cases, the computer switch register. All replies to specific input requests are entered from the keyboard and are terminated by the RETURN key. The program ignores any characters other than the 64 printing characters (including space), RUBOUT, RETURN, and the "break" characters Control-F, Control-P, Control-E, Control-T, and Control-N.

Three of the recognized keyboard characters are input line-editing characters that may be used by the operator to correct errors made while typing. The editing characters function as follows:

RUBOUT Or SHIFT-L	Deletes one character to the left for each input and echoes back as \.
SHIFT-O	Deletes the entire input line to the left and echoes back as ←.

The break characters are used to alter the execution sequence of the diagnostic as follows:

Control-F	Echoes back as ↑ F and then returns the program to the FMPR request (Step 3 below).
Control-P	Echoes back as ↑ P and then returns the program to the PACK request (Step 4 below)
Control-E	Echoes back as ↑ E and then returns the program to the EMSK request (Step 5 below)
Control-T	Echoes back as ↑ T and then returns the program to the TEST request (Step 6 below)
Control-N	Echoes back as ↑ N, stops execution of the test in progress, and then starts execution of the next test in the sequence specified by the last TEST request.

Not all of the break characters are recognized at each step in the program's execution sequence, therefore, those break characters that are recognized are specifically indicated in the steps described below.

All numeric values input to or output from the program, unless otherwise specified, are in octal. If more than 5 digits are input for a numeric value then the program will recognize only the low ordered 16 binary bits of the value specified.

If any operator reply to an input request cannot be recognized by the program or is outside of the allowed set of values, then the program will output the error message ?? and repeat the request. Similarly, for input requests in which the operator may input a variable number of parameters, the program will output the error message:

TOO MANY

and repeat the request if the number of parameters exceeds the maximum allowed.

In the steps below, the symbols { and } are used to enclose operator inputs that are optional. Examples that are given use underlining to indicate operator inputs and the symbol ↵ to indicate a RETURN.

- 1) Make sure the console teletype is on the LINE position and that each disk drive to be tested is loaded with a cartridge. Load the diagnostic using the Data General absolute binary loader. Once loaded, the program will automatically start execution at Step 2,

below. The user, however, may restart the program at memory location 2 or 3. Restarting on Location 2, will proceed from Step 2 while restarting at Location 3 will proceed at the last parameter request (Step 3, 4, 5, or 6) output by the program.

- 2) The program now types out the size and locations of the I/O buffers as:

```

WBUF  = a
RBUF  = b
BUFL  = c
ENDM  = d

```

where a is the starting address of the write buffer, b is the starting address of the read buffer, c is the buffer length, and d is the address of the last memory location to be used by the program. Following this step the program enters its normal execution sequence.

```

Example:  WBUF  =6210
          RBUF  =6630
          BUFL  =420
          ENDM  =17577

```

The write buffer starts at 6210_8 , the read buffer at 6630_8 , the length of each buffer is 420_8 , and the last memory location to be used is 17577_8 .

- 3) The program requests the disk formatter configuration by typing:

```
FMPR =
```

The operator replies by entering from 2 to 5 numeric parameters in the form:

```
w, s {, { b } {, { d } { { i } } } }
```

where the individual parameters are defined as:

```

w Number of words per sector
s Number of sectors per track (revolution)
b Number of bits per word
d Device address of the formatter coupler
i Number of the formatter coupler interrupt mask bit

```

The parameters w and s must be specified and are normally entered as one of the following pairs:

<u>Sectors/Track</u>	<u>w</u>	<u>s</u>
10		
12	400	14
16	300	20
24	200	30
32	100	40

The parameters b, d, and i are optional. b should be either omitted or entered as 6, its default value, specifying 16 bits per word. d, if required, should be entered as an even number in the range 2-74₈. If omitted the default value of 30₈ will be assumed. i, if required, should be entered as a number in the range 0-17₈. If omitted a default value of 5 will be assumed. The program recognizes the break character Control-F during this step.

Example: FMPR= 300, 20, , 34 2

Specifies the word per sector, sectors per track, and coupler device address as 300₈, 20₈, 34₈, respectively. 16 bits per word and an interrupt mask bit of 5 are both assumed.

- 4) The program requests the identity and sequence of the packs to be tested by typing:

PACK=

The operator replies by entering from 1 to 8 pack identifiers, p, in the form:

$p_1 \{, p_2 \{, p_3 \{, \dots p_n \} \} \}$

where each p has the form:

u/r/e

and where the individual parameters in p are defined as:

u	Unit Number
r	Pack Number
e	Cylinders to be used

u specifies the unit number as a single digit in the range 0-3. For units having removable packs, r specifies the removable pack as the digit 0 and the fixed pack, if present, as the digit 1. For units having only a single fixed pack, r specifies this pack as the digit 0. The parameter e specifies the cylinders to be used by the diagnostic and is entered as either e or e_1-e_2 where e, e_1 , and e_2 are cylinder numbers, and e_1 is less than or equal to e_2 . In the first form cylinders 0 through, and including, e will be used while in the second form cylinders e_1 through, and including, e_2 will be used.

All of the pack identifiers are treated independently so that the user may specify different packs on the same unit and/or different cylinders on the same pack.

The diagnostic will use only those packs and cylinders specified.

The program recognizes the break characters Control-F and Control-P during this step.

Example: PACK=0/0/313, 0/1/10-25, 0/1/200-247

Specifies the following packs:

Unit 0, Pack 0, Cylinders 0-313₈

Unit 0, Pack 1, Cylinders 10₈ - 25₈

Unit 0, Pack 1, Cylinders 200₈ -247₈

- 5) The program requests the parameters to be used to control error message printing by typing:

EMSK=

The operator replies by entering up to 2 parameters in the form:

$\{m \{, \{l\}\}\}$

where the individual parameters are defined as:

m Error word mask
 l Verify detail line limit

m is a 16 bit numeric value which is used to mask an error word that is generated by the program during each I/O or buffer compare operation. Each bit in the error word is assigned to a status indicator detected by the program and is set to a 1 if the detected status differs from the expected status. At the completion of the operation m is ANDed with the error word and, if the result is non-zero, an error message is printed describing the error.

The bits are assigned to the error word, and thus to m, as follows:

<u>Bit</u>	<u>Value</u>	<u>Interpretation</u>
2	20000	Verify Error
3	10000	Interrupt Error
4	4000	Not Ready Error
5	2000	Write Protect Error
6	1000	Cylinder Address Error
7	400	Preamble Check Error
8	200	Time Out Error
9	100	Format Error
10	40	CRC Error
11	20	Rate Error
12	10	Bad Sector Flag Error
13	4	Word Count Error
14	2	Done/Busy Error
15	1	Seek Error

Any bits other than those described above are ignored by the program.

If m is omitted a value of 177777_8 will be used.

λ is a number that specifies the maximum number of detail lines to be output following a verify error message. If specified as 0, then no detail lines will be printed, if omitted, then all of the detail lines will be printed.

The program recognizes the break characters Control-F, Control-P, and Control-E during this step.

Example: $EMSK=, \underline{3}$

 Sets the error print mask to 177777_8 , all errors, and the verify detail line limit to 3.

- 6) The program requests the tests to be performed by typing:

TEST=

The operator replies by entering a sequence of test specifications, t , which specify the order in which the tests are to be executed. The sequence of test specifications permits the inclusion of nested test execution loops and is entered in any one of the following forms:

t
 $m*t$
 (t')
 $m*(t')$
 t'_1, t'_2

where m is a number in the range $0-7777_8$ and t' is any one of the above forms. The simple forms t and (t') specify just t and t' , respectively.

The forms $m*t$ and $m*(t')$, however, specify t and t' , respectively, repeated m times while the form t'_1, t'_2 specifies the sequence t'_1 followed by t'_2 .

As a result any of the following input sequences could be entered.

t
 m*t
 m₁*(m₂*t)
 t₁, m*t₂
 t₁, m₂*t₂, m₂*(m₃*t₃)
 t₁, m₁*(t₂, m₂*t₄, m₃*(t₅, t₆)), t₇
 etc.

If m is entered as 0 then the following t or t' is repeated indefinitely.

Parenthetical nesting of sequences may be used 4 levels deep.

Each t can either be omitted or entered as either n or n₁ - n₂ where

n, n₁ and n₂ are test numbers and n₁ is less than n₂. The form n specifies the single test n while the form n₁ - n₂ specifies tests n₁, through, and including, n₂.

If t is omitted the default set of tests, 1-24, is specified. The allowed test numbers are as follows:

<u>Test Number s</u>	<u>Test</u>
0	Check Sectors
1	Format Sectors
2	Verify Preambles
3	Verify Sector Addressing
4	Seek and Restore
5	Write Range
6	Read Range
7	CRC Generation
10	Memory Addressing
11	Data Patterns
12	Random Data
13	Random Parameters
14	Preamble Sector Compare Error
15	Preamble Cylinder Compare Error
16	CRC Compare Error
17	Bad Sector Flag Error
20	Write Protect Flag Error
21	Time Out Error
22	Cylinder Addressing Error
23	Word Count Error
24	Interrupt System

<u>Test Number</u>	<u>Test</u>
25	Transfer Rate Error
26	Write Protect Switch Error
27	Not Ready Error
30	Format Error
31	Automatic Program Load
32	Miscellaneous Functions

The program recognizes the break characters Control-F, Control-P, Control-E, and Control-T during this step.

Example: TEST= 3* (1-2, 2*4, 2* (15,14)) , , 0

Specifies the following sequence of tests:

1, 2, 4, 4, 15, 14, 15, 14, 1, 2, 4, 4, 15, 14, 15, 14

1, 2, 4, 4, 15, 14, 15, 14, 1, 2, 3, 4, 5, 6; 7, 10,

11, 12, 13, 14, 15, 16, 17, 20, 21, 23, 24, and 0.

- 7) The program Proceeds to execute the tests specified in Step 6 on the packs specified in Step 4. When the specified test sequence is complete the program returns to Step 6.

The program recognizes all of the allowed break characters during this step.

ERROR MESSAGES

The printing of error messages is completely controlled by the operator via the error word mask. Three types of error messages are output by the program, one as a result of status errors returned by the formatter (hardware errors), another resulting from errors in the data input from the formatter (verify errors), and a third resulting from errors in the interrupt system.

The printing of hardware error messages is controlled by bits 4-15 of the error word mask. Bits 4-13 and 15 of the error word correspond to actual bits returned in the formatter status register, while bit 14 of the error word is set when an error is found in the sequencing of the coupler BUSY/DONE flags. A further description of the status register bits and the BUSY/DONE sequencing can be found in the XDF-50 I/O Specification Manual.

Following each operation performed by the diagnostic, the value of the error word is compared with the value that the program expects to receive at that point. If one or more bits in the register is other than expected and if the corresponding bit is set in the error word mask then a hardware error message is printed.

Hardware error messages have the format:

H ssss OP=o, www AD=u, p, ccc, t, ss ST=eeee, rrrr

and are interpreted as follows:

H ssss ssss is the current test state of the test in progress.

OP=o, www o is the disk operation that was performed and www is the word count specified for that operation. (The buffer address was WBUF for write operations and RBUF for read operations.)

AD=u, p, ccc,t,ss

u is the disk unit number, p is the pack number, ccc is the cylinder number, t is the track number, and ss is the sector number.

ST=eeee, rrrr

eeee is the expected value and rrrr the received value of bits 4-15 of the error word.

The printing of verify error messages is controlled by bit 2 of the error word mask and by the verify detail line limit. Verify error messages occur when data read in from the controller differs from that which is expected by the program. Each verify error message consists of one line to notify the operator that a verify error occurred followed by a detail line for each data word in error. Verify error messages are printed only if bit 2 of the error word mask is set. The number of detail lines printed will always be less than or equal to the verify line limit.

The format of these messages is:

```
V ssss OP=o, www AD=u, p, ccc, t, ss EC=nmmn
aaaa eeeee rrrrr
. . .
. . .
. . .
```

and is interpreted as follows:

V ssss	ssss is the current test state of the test in progress.
OP= o, www	o is the last disk operation that was performed and www is the word count specified for that operation. (The buffer address was RBUF.)
AD= u, p, ccc,t,ss	u is the disk unit number, p is the pack number, ccc is the cylinder number, t is the track number, and ss is the sector number.
EC= nnnn	nnnn is the number of data words in error (i. e., the number of detail lines to follow).
aaaa	is the address, relative to WBUF and RBUF, of the data word in error.
eeeeee	is the expected data word from WBUF.
rrrrrr	is the received data word from RBUF.

Whenever detail lines are printing, the operator may delete remaining detail lines by entering any key from the teletype keyboard. In this case the program will output * and continue with the test in progress.

The printing of interrupt error messages is controlled by bit 3 of the error word mask. These error messages occur during the interrupt system test (Test 24) and indicate some form of interrupt failure. Interrupt error messages are printed only if bit 3 of the error word mask is set.

The format of these messages is :

I s or I s, aa

and is interpreted as follows:

I s s is the state of the interrupt system test and indicates the type of error.

aa (when present) specifies the address of a device which caused an interrupt.

The interpretation of s is as follows:

<u>s</u>	<u>Interpretation</u>
0	The disk DONE flag was not cleared by a clear (NIOC) function.
1	An interrupt occurred with all mask bits set and the disk DONE flag reset (from the previous NIOC).
2	An interrupt occurred with all mask bits set, except for that of the disk controller, and with the disk DONE flag reset (from the previous NIOC).
3	An interrupt occurred with all mask bits set and the disk DONE flag set (from a NOP operation).
4	An interrupt did not occur with all mask bits set, except for that of the disk controller, and with the disk DONE flag set.
5	An IORST instruction failed to clear the disk DONE flag.

INTERACTIVE TESTS

Four tests which require operator interaction are provided. These are:

Test 25 Transfer Rate Error

This test verifies that the formatter detects transfer rate errors, during both read and write operations, by initiating each operation and then halting the computer. The program will halt, and the operator should press CONTINUE, twice for each pack specification entered in response to the PACK request.

Test 26 Write Protect Switch Error Test

This test verifies that the formatter detects the write protect switch and prevents writing when it is set. The program will type out:

STWP u, p

The operator should set the write protect switch for the unit and pack specified by u and p, and then press the teletype SPACE bar to continue. After the test is complete the program will type:

CLWP u, p

The operator should turn off the write protect switch for the selected pack and then press the SPACE bar to continue.

The program will repeat this sequence for each pack specification entered in response to the PACK request.

Test 27

Not Ready Test

This test verifies that the formatter detects a unit not ready condition.

The program will type out:

STNR u, p

The operator should unload the unit specified by u, and then press the teletype SPACE bar to continue. After the test is complete the program will type:

CLNR u, p

The operator should reload the selected unit, wait until it is ready and then press the SPACE bar to continue.

The program will repeat this sequence for each pack specification entered in response to the PACK request.

Test 30

Format Error Test

This test verifies that the formatter detects sector format errors.

To execute this test a pack must be available that has fewer words per sector (more sectors per track) than the number for which the formatter is configured.

The program will type out:

LDFE u, p

Test 30

Format Error Test - Continued

The operator should load the test pack into the unit specified by u, and then press the teletype SPACE bar. After the test is complete the program will type:

ULFE u, p

The operator should remove the test pack from the specified unit, replace it with a properly formatted pack, and then press the SPACE bar to continue.

The program will repeat this sequence for each pack specification entered in response to the PACK request.

Should the user desire to bypass the testing of a pack in Tests 26-30, a CONTROL-I should be entered for the setup request for that unit rather than the SPACE bar. The program will respond with an * and proceed to the next pack specified.