# MESA TWO DIAGNOSTICS
# General Instructions

Pub # 5-1000

June 1974



MARTIN, WOLFE INC.

# CONTENTS

| Section | Title | Page |
|---|---|---|

| Section | Title | Page |
| --- | --- | --- |

# SECTION 1

## 1.0 INTRODUCTION

The MESA TWO diagnostics are a series of tests designed to aid the analysis of a system problem. All diagnostics are contained on a single diagnostic disk pack. In addition to diagnostics a diagnostic test pack contains utilities that provide certain capabilities (e.g., copying disk packs, etc.).

Associated with each diagnostic disk pack are release and version numbers. These numbers identify the generation of tests on the pack. New releases are issued periodically as new diagnostics become available. A new release is version one of that release. New versions of a release generally indicate a fix has been made to an existing diagnostic.

## 1.1 GENERAL

Each disk pack contains a loader, a set of tests, and a data and save area.

### LOADER

The diagnostic loader is located on the upper surface, sector zero, cylinder zero of the diagnostic disk pack. The loader is placed in memory by issuing "reset-program load." The loader reads the test number from the console switches and loads the appropriate diagnostic into the first 4K of memory (4K = 4096 words). When the diagnostic has been loaded, control is transferred to it via the entry point.

### DIAGNOSTIC TESTS

Each diagnostic is located on a single cylinder of the disk pack. Test zero (the instruction sheet) is located on cylinder zero, test one is located on cylinder one, etc. Cylinders 0–77 (octal) are reserved for diagnostic tests.

### DATA AND SAVE AREA

All cylinders above 77 (octal) to the last cylinder on the disk packs are a data and save area. This area contains special versions of diagnostics to be run consecutively and data areas used by some diagnostics. The exact allocation of this area may change in each new release. This are cannot be considered a scratch area.

## 1.2 LOADING AND EXECUTION

A diagnostic is loaded and begins execution by issuing "reset-program load" with the proper console switches set.

### 1.2.1 Loading A Diagnostic

Each diagnostic is loaded by the standard disk bootstrap. Switch zero must be up to indicate a high speed device and switches eleven and twelve must be up to indicate booting from the disk drive. The desired diagnostic is selected by properly setting switches four through nine inclusive. Switches four through nine indicate the desired test number. For example, the switch setting 100030 (octal) boots test number zero while the switch setting 100730 (octal) boots test number seven.

The loader halts when a bad status is encountered loading itself into core, a test is not available, a test cannot be read, or option switches are required by the test. The exact address of each halt is documented in the "Instruction Sheet," test zero. If a test is not available (i.e., switches contain an illegal test number), the message "TEST NOT AVAILABLE" is output to the main terminal before halting. If option switches are required by the diagnostic, the procedure described under "1.2.2 Test Option Switches" is followed.

### 1.2.2 Test Option Switches

Certain tests allow option console switch settings which alter the flow of the test, change the devices being tested, or control other functions of the diagnostic.

Once the diagnostic has been placed into memory, the loader checks if option switches are used by the diagnostic. If the test does not use option switches, control is passed via the entry point and the diagnostic begins execution. If the test uses option switches, the loader outputs "SET OPTION SWITCHES - PRESS CONTINUE" on the main terminal and halts. The user should reset the switches and press CONTINUE to start the diagnostic.

### 1.2.3 Test Execution

Once the diagnostic is loaded, and option switches set if required, the loader passes control to the diagnostic. Each diagnostic executes in a different manner. The specific design and execution of a diagnostic is explained separately for each test in Section 3 — TEST DESCRIPTION.

Some diagnostics test several pieces of equipment simultaneously; other diagnostics are designed to test only a specific piece of equipment. A diagnostic which detects a hardware problem may only indicate the symptom and not the cause of the error. For instance the central processing unit test may indicate an instruction failure when the real problem is a faulty memory.

Certain tests run continuously until manually halted or an error is detected. These tests, the CPU Logic Test for example, output the word "PASS" after a specific number of good passes through the diagnostic. Other tests are run via input from the main terminal by the user.

### 1.2.4 Test Data Input

Certain tests require data input from the main terminal. This input varies and is described for each test in Section 3.

The main terminal of a system is either the 50/51 terminal of a 50/51 or 50/51 - 10/11 system or MUX unit zero on a MUX system. Any data input on a 10/11 terminal or any unit other than unit zero of a MUX system is ignored. The exception to this rule is the actual terminal diagnostic.

### 1.2.5 Test Halts

Most diagnostics halt upon encountering an error, unexpected condition, or invalid interrupt. Some diagnostics output error messages. All halts for a diagnostic are documented in Section 3.

# SECTION 2

## 2.0 HARDWARE CONFIGURATION

The basic hardware configuration is a central processing unit, video terminal, line printer, and disk drive.

## 2.1 CENTRAL PROCESSING UNITS

All central processing units are supported by the MESA TWO system diagnostics.

## 2.2 DISK DRIVES

The system may contain from one to four disk drives. All disk drives are controlled by a single disk controller. Each disk drive contains a fixed spindle and a removable spindle. A page of data contains 256 words (4096 bits). There are 12 pages per track, and two tracks per cylinder.

The double-density drive has 406 (decimal) cylinders. These cylinders are more closely packed (approximately half as far apart) than the cylinders on the single density drive.

A single density disk contains approximately half the information contained on a double-density drive. The single density drive has 204 (decimal) cylinders.

## 2.3 VIDEO TERMINALS

Each system contains one or more video terminals for data input. Most single terminal systems use a 50/51 basic I/O board as the interface to the central processor. A multiple terminal system may be either a 50/51 - 10/11 two terminal system, or a QUAD MUX (multiplexor) system.

### 2.3.1 50/51 - 10/11 Terminal System

A 50/51 - 10/11 terminal system can support two terminals. Terminal one uses the 50/51 basic I/O board as the interface to the central processor, and terminal two uses the 10/11 basic I/O board as the interface to the central processor.

### 2.3.2 QUAD MUX Terminal System

A single QUAD MUX multiplexor board can support from one to four terminals. Each terminal has a distinct line number — zero through three.

### 2.3.3 Main System Terminal

The main system terminal is always the 50/51 terminal of a 50/51 or 50/51 - 10/11 system or MUX terminal zero of a QUAD MUX system.

The MESA TWO system diagnostics, except for the actual terminal diagnostic, communicate with the user through the main system terminal only. All test data input must be entered from the main terminal. All test data results and error messages are output to the main terminal.

## 2.4 PRINTERS

The MESA TWO system supports five printers:

1. 200 line per minute printer
2. 60 line per minute printer (63 ASCII character set)
3. 60 line per minute printer (64 ASCII character set)
4. 125 line per minute printer
5. 600 line per minute printer

The 60, 125, and 200 line per minute printers use the same printer controller as the interface to the central processing unit. The 600 line per minute printer uses a different printer controller board.

## 2.5 OPTIONAL EQUIPMENT

Some systems have additional equipment to support optional system facilities. The point of sale terminals and binary synchronous communications are examples of system options available.

The optional equipment includes:

1. Real-Time Clock
2. Binary Synchronous Adaptor
3. Asynchronous Adaptor

# SECTION 3

## 3.0 TEST DESCRIPTION

This section contains a description of each diagnostic and utility provided on the MESA TWO system diagnostic.

Each diagnostic is booted via appropriate switch settings. Each diagnostic occupies a single cylinder of the diagnostic disk pads. The test number and the cylinder number correspond.

The test entry point is the start of the diagnostic. It may be possible to restart a test at the entry point. However, a test should always be reloaded from the diagnostic disk pack if possible. Restarting from an entry point is unpredictable and should be avoided.

All numbers in this section are octal digits unless otherwise specified.

## 3.1 INSTRUCTION SHEET

Test Number: 0
Bootstrap Switch Settings: 100030
Entry Point: 400
Option Switches: None

### 3.1.1 Test Description

This test is an instruction sheet of detailed information on the diagnostic disc pack. The information includes:

1. The release and version number of the diagnostic pack.
2. A list of all diagnostics on the pack and the appropriate bootstrap switch settings.
3. Halt addresses executed by the loader.

### 3.1.2 Execution Procedure

This instruction sheet is booted into memory by issuing "reset – program load" with boot address 100030. When the instruction sheet is in memory, information is displayed on the main terminal.

In the upper right hand corner of the display is the release and version number of the diagnostic disc pack. In the lower left hand corner is the message "TAB FOR MORE DATA." In the lower right hand corner the message "END OF DATA" appears if this is the last page of data to display. The rest of the screen contains test and halt information.

Typing TAB automatically produces the next page of information. If the last page of information is displayed, typing TAB redisplays the first page.

### 3.1.3 Halts

None

### 3.1.4 Error Messages

None

## 3.2 DISK DIAGNOSTIC

Test Number: 1
Bootstrap Switch Settings: 100130
Entry Point: 2
Option Switches: None

### 3.2.1 Test Description

The disk diagnostic tests the disk controller and all units attached to the system.

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 terminal 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-P, Control-E, Control-T, and Control-N.

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

| | |
|---|---|
| RUBOUT<br>Or<br>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-P | Return the program to the PACK request (Step 4 below), and echoes back as ↑P. |
| Control-E | Returns the program to the EMSK request (Step 5 below) and echoes back as ↑E. |
| Control-T | Returns the program to the TEST request (Step 6 below) and echoes back as ↑T. |
| Control-N | Stops execution of the test in progress, starts execution of the next test in the sequence specified by the last TEST request, and echoes back as ↑N. |

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 five digits are input for a numeric value than the program recognizes 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 outputs the error message:

??

and repeat the request. Similarly, for input requests in which the operator may input a variable number of parameters, the program outputs 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. In the examples, underlining is used to indicate operator inputs and the symbol ↓ is used to indicate a RETURN.

1) Make sure each disk drive to be tested is loaded with a cartridge. Load the diagnostic from the diagnostic disk pack. Once loaded, the program automatically starts execution at Step 2 below. The user, however, may restart the program at memory location 2 or 3. Restarting on Location 2, proceeds from Step 2 while restarting at Location 3 proceeds from Step 4.

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

```
WBUF = a
RBUF = b
BUFL = c
```

where a is the starting address of the write buffer, b is the starting address of the read buffer, and c is the buffer length. Following this step the program enters its normal execution sequence.

```
Example:  WBUF = 6521
          RBUF = 7141
          BUFL =  420
```

The write buffer starts at $6521_8$, the read buffer at $7141_8$, and the length of each buffer is $420_8$.

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:

| $Sectors/Track_{10}$ | w | s |
|---|---|---|
| 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 of the range $2\text{-}74_8$. If omitted the default value of $30_8$ will be assumed. i, if required, should be entered as a number in the range $0\text{-}17_8$. If omitted a default value of 5 will be assumed.

Example: FMPR=300,20,,34↓

Specifies the words per sector, sectors per track, and coupler device addresses as $300_8$, $20_8$, and $34_8$, 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:

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

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 c specifies the cylinders to be used by the diagnostic and is entered as either c or $c_1\text{-}c_2$, where c, $c_1$, and $c_2$ are cylinder numbers, and $c_1$ is less than or equal to $c_2$. In the first form cylinders 0 through, and including, c will be used while in the second form cylinders c, through, and including $c_2$ 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 character 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 $0\text{-}318_8$

Unit 0, Pack 1, Cylinders $10_8\text{-}25_8$

Unit 0, Pack 1, Cylinders $200_8\text{-}247_8$

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

EMSK=

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

$$\{m\}\{,\{\ell\}\}$$

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:

| Bit | Value | Interpretation |
|---|---|---|
| 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 $177777_8$ is used.

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

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

Example: EMSK=,3 ↓

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

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
$m_1$*($m_2$*t)
$t_1$,m*$t_2$
$t_1$,$m_2$*$t_2$,$m_2$*($m_3$*$t_3$)
$t_1$,$m_1$*($t_2$,$m_2$*$t_4$,$m_3$*($t_5$,$t_6$)),$t_7$
etc.

If m is entered as 0 then the following t or t' is repeated indefinitely. Parenthetical nesting of sequences may be used four levels deep.

Each t can either be omitted or entered as either $n_1$ - $n_2$ where n, $n_1$ and $n_2$ are test numbers and $n_1$ is less than $n_2$. The form n specifies the single test n while the form $n_1$-$n_2$ specifies tests $n_1$, through, and including, $n_2$.

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

| Test Number$_8$ | Test |
|---|---|
| 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 |
| 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-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.

## INTERACTIVE TESTS

Four tests which require operator interaction are provided as follows:

**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 halts 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 types:

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 types:

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 repeats 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 types:

STNR u,p

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

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 repeats 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 types:

LDFE u,p

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

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 repeats this sequence for each pack specification entered in response to the PACK request.

## 3.2.2 Execution Procedure

**DISK DIAGNOSTIC**

Boot Address — 100130

| Screen Display | Description |
|---|---|
| WBUF = xxxx<br>RBUF = xxxx<br>BUFL = xxx | Size and location of I/O buffers. |
| FM, P, R = 400, 14 ↓ | FM, P, R = w, s, b, d, i<br>w = words/sector<br>s = sectors/track<br>b = bits/word<br>d = device address of coupler<br>i = interrupt mask bit |
| PACK = 0/0/625, 0/1/625 ↓<br>Do not use with diagnostic pack | PACK = U/P/C   U = 0-3 unit #<br>P = 0 removable pack<br>P = 1 fixed pack<br>($625_8$ = 405 cylinders — CDC)<br>($313_8$ = 202 cylinders — Caelus) |
| EMSK = ↓ | Not normally used. |
| TEST = ↓<br>(↓ runs all tests except 0) | Runs all tests starting with #1.<br>0-24 can be entered individually and each test will be run separately. See Xebec Diagnostic List for test identification. |

Errors are displayed on the CRT as they occur. The last four numbers in each line describe the error condition as outlined under the CONTROLLER STATUS section.

Control P – returns control to PACK
Control T – returns control to TEST
Control N – returns control to next test in the sequence.

Refer to Xebec manual for further information.

**DISK DIAGNOSTIC USER SPECIFIED TEST**

PACK = U ↓

OP = o, w, d, e, m ↓

| | | |
|---|---|---|
| o = | op code | 0-7* |
| w = | word count | 0-(BUFL) |
| d = | data type | 0-5** |
| e = | expected status | 0-7777 |
| m = | error mask | 0-7777 |

AD = u, p, c, t, s ↓

| | | | | |
|---|---|---|---|---|
| u = | unit | 0-3 | | |
| p = | pack | 0-1 | 0 – removable | 1 – fixed |
| c = | cylinder | 0-7777 | | |
| h = | head | 0-1 | 0 – upper | 1 – lower |
| s = | sector | 0-(SPTK-1) | | |

OP = ↓

*0 = No Operation
1 = Write Preamble and a Sector
2 = Check Preamble and Write a Sector
3 = Check Preamble and Read a Sector
4 = Read Diagnostic Mode
5 = Check Preamble and Write a Sector, but Ignore Write Protect Bit in Preamble
6 = Write Diagnostic Mode
7 = Ignore Preamble and Read 1 Sector

**0 = all 0's
1 = all 1's
2 = worst case pattern (061430)
3 = ascending count
4 = random data based on sector address
5 = random data

### 3.2.3 Halts

None

### 3.2.4 Error Messages

Printing of error messages is completely controlled by the error word mask specified by the operator. 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.

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:

ssss – Current test state of the test in progress.

| | |
|---|---|
| OP=o,www | o is the disk operation that was performed and wwww 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, 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,s   EC=nnnn
aaaa  eeeeee   rrrrrr
  .      .       .
  .      .       .
  .      .       .
```

and is interpreted as follows:

| | |
|---|---|
| ssss | current 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 | is the disk unit number, p is the pack number, ccc is the cylinder number, t is the track number, ss is the sector number. |

| | |
|---|---|
| EC=nnnn | number of data words in error (i.e., the number of detail lines to follow). |
| aaaa | address, relative to WBUF and RBUF, of the data word in error. |
| eeeeee | expected data word from WBUF. |
| rrrrrr | received data word from RUBF. |

Whenever detail lines are printing, the operator may delete remaining detail lines by entering any key from the terminal 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:

| | |
|---|---|
| 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. |

## 3.3 CPU LOGIC TEST

Test Number: 2
Bootstrap Switch Settings: 100230
Entry Point: 400
Option Switches: None

### 3.3.1 Test Description

The CPU Logic Test checks the logic operation of the central processor. The arithmetic and logic commands (COM, NEG, MOV, etc.) are executed and the outcome verified. The first command tested is a "HALT." After this halt, an invalid result causes a halt. After each 2048 (decimal) successful passes through the program, the word "PASS" is output to the main terminal.

The CPU Logic Test is designed to be a pass/no-pass test of the central processor. A halt (other than this initial halt) is considered a no-pass condition and can be caused by a faulty memory or central processor.

### 3.3.2 Execution Procedura

The CPU Logic Test is booted into memory by issuing "reset-program load" with boot address 100230. Once the logic test is loaded, execution begins immediately. The first instruction tested is the "HALT" instruction and the program should halt at 405. The user must press CONTINUE to resume the test. The "HALT" instruction is never tested again after the first loop. The word "PASS" is output to the main terminal after 2048 (decimal) successful passes. A halt occurs (other than the initial halt) if an error is encountered. The test loops infinitely until stopped manually if no error is detected.

### 3.3.3 Halts

405 – the first instruction tested is a "HALT." This halt should occur immediately after loading. Press CONTINUE to resume the diagnostic.

2731 – arithmetic or logic command failed. This is a no-pass condition. This generally indicates either the central processor or memory is faulty.

### 3.3.4 Error Messages

None

## 3.4 CPU EXERCISER

Test Number: 3
Bootstrap Switch Settings: 100330
Entry Point: 2
Test Option Switches: 4 – test real-time clock interrupts

### 3.4.1 Test Description

The exerciser tests the reliable operation of the central processor and the real-time clocks. Raising switch four on the CPU tests the real-time clocks. The exerciser tests the CPU instruction set and the auto-incrementing and auto-decrementing memory cell locations. The exerciser is a viable memory diagnostic.

The exerciser uses a buffer area to execute and test instructions. After the first pass through the program, this buffer area is expanded from approximately 600 words to the entire available memory. All passes after pass one require a significantly longer time to run. Each successful pass through the program displays the word "PASS" on the main terminal; a failure results in a halt.

The real-time clock option tests only the interrupt capabilities of the real-time clock and not the time accuracy of the interrupts.

### 3.4.2 Execution Procedure

The exerciser is loaded into memory by issuing "reset-program load" with the correct switch setting. When the diagnostic is loaded, the loader allows the user to set the option switches.

The diagnostic executes pass one and sizes the memory. The buffer area is increased to the maximum memory size. The switches are read at the end of each instruction pass. If the device switch has been set during the pass, the device is activated and the interrupt system enabled. If the device has been reset during the pass, the device is deactivated.

The diagnostic displays "PASS" at the end of each successful pass and halts if an error is detected. Diagnostic operation continues until manually stopped.

### 3.4.3 Halts

Effective error detection is accomplished by dispersing halts throughout the test program. This technique makes it necessary to group halt addresses. These addresses may change as new versions of the exerciser are issued.

When the real-time clock is activated and the ION lamp is off, an error occurred at the interrupt service level.

| Halt | Description |
|---|---|
| 621 | Unknown interrupt. The interrupting device code was not the activated device. |
| 1074 | Busy or done flag of the real-time clock is not properly set at interrupt. |
| 1313 - 1436 | Failure of LDA or STA command. May indicate a memory problem. |

| Halt | Description |
|---|---|
| 1437 - 1462 | JSR command failed. |
| 1463 - 1501 | JMP command failed. |
| 1502 - 1540 | Positive displacement test using LDA command failed. |
| 1541 - 1602 | Negative displacement test using LDA command failed. |
| 1603 - 1645 | Positive displacement using JSR command failed. |
| 1646 - 1701 | Negative displacement using JSR command failed. |
| 1702 - 1724 | PC location pointer failed. |
| 1725 - 1762 | ISZ skipped when quantity was not zero. |
| 1763 - 2002 | ISZ failed to skip when quantity was zero. |
| 2003 - 2141 | ISZ instruction failed. |
| 2142 - 2160 | DSZ skipped when quantity was not zero. |
| 2161 - 2200 | DSZ did not skip when quantity was zero. |
| 2201 - 2353 | DSZ instruction failed. |
| 2354 - 2550 | Indirect addressing did not function properly. |
| 2551 - 2606 | JSR instruction failed. |
| 2607 - 2712 | Indirect addressing (2 levels) failed. |
| 2713 - 3024 | Indirect addressing failed. |
| 3025 - 3346 | Auto Increment/Decrement failed. |

### 3.4.4 Error Messages

None

## 3.5 PRINTER DIAGNOSTIC

Test Number: 4
Bootstrap Switch Settings: 100430
Entry Point: 400
Option Switches: None

### 3.5.1 Test Description

The printer diagnostic tests all printer types and models available on the MESA system. The diagnostic contains three tests and a special test, zero, which automatically cycles a specific number of times through the tests.

Test one outputs a "barber pole" and checks the busy/done flag setting of the controller. Test two outputs a single line of the characters entered and skips the number of lines entered. Test two also validates the interrupt capabilities of the printer. Test three prints a count from one to 132 to validate printing of each print position.

### 3.5.2 Execution Procedure

The disk diagnostic is loaded by issuing "reset-program load" with the correct switch settings. When the diagnostic is loaded, it outputs "PRINTER TYPE =" on the screen. Enter the correct printer type (1 thru 5) as outlined below:

1 = 200 line per minute printer

2 = 60 line per minute printer (63 ASCII character set)

3 = 60 line per minute printer (64 ASCII character set)

4 = 125 line per minute printer

5 = 600 line per minute printer

Next "TEST =" appears on the screen; enter the appropriate test number. For tests one and two, "START CHARACTER =" appears – enter the character to be output. For test two, "SKIPS BETWEEN LINES =" appears; enter the number of skips between each line. Type ESC to stop execution and return to the "TEST =" question.

### 3.5.3 Halts

All errors cause an error message to be output to the main terminal. After the error message has been displayed, the diagnostic halts at 404. A power failure causes a halt at 403.

### 3.5.4 Error Messages

All errors encountered during the diagnostic run result in an error message displayed on the main terminal as follows:

BUSY FLAG NOT TURNING ON

This error message is displayed if the busy flag does not turn on immediately after issuing an I/O command to output a character to the printer.

## DONE FLAG NOT TURNING OFF

The done flag did not turn off immediately after issuing an I/O command to output a character to the printer.

## DONE FLAG NOT TURNING ON

The done flag did not turn on to accept the character output on the last I/O command.

## WRONG DEVICE CODE ON PRINTER INTERRUPT

The device interrupting on the I/O command is not the legal printer device code. Either the wrong device is interrupting, or the printer controller is returning an illegal device code.

## PRINTER NOT INTERRUPTING

An I/O command was issued to the printer, but the printer did not properly interrupt within approximately one second from the I/O command output.

## ILLEGAL INTERRUPT

An interrupt occurred when no interrupt was expected. It was not a power fail interrupt. A mask out of -1 was issued. Either the mask out instruction did not function, the controller did not properly accept the mask out bit, or the controller issued an invalid interrupt.

## 3.6 MEMORY ADDRESSING DIAGNOSTIC

Test Number: 5
Bootstrap Switch Settings: 100530
Entry Point: 2
Option Switches: None

### 3.6.1 Test Description

The Memory Address Test is used to detect memory addressing errors. The test writes the cell address into the cell and rereads and validates the result. An invalid result causes the program to halt. After each 140 successful passes through the diagnostic, the word "PASS" is printed at the main terminal.

Data is also written out of memory bounds to validate addressing memory out of bounds does not affect real memory.

### 3.6.2 Execution Procedure

The memory address test is loaded into core by issuing "reset-program load" with the correct switch settings. When the diagnostic is loaded, it sizes core to detect and output the highest core address to the main terminal. If this address is not the last core location of the machine, an entire bank of core may be failing.

Once core has been sized, the diagnostic writes the cell address into the cell and rereads and validates the result. After each 140 successful passes through the diagnostic, the word "PASS" is output to the main terminal. An invalid result causes a halt.

### 3.6.3 Halts

Two halts may occur in the memory diagnostic test:

216 MEMORY CELL ADDRESS FAILURE. The address written into memory cell in AC3 could not be properly read. The contents read from the cell are in AC2.

230 UPPER MEMORY BOUND FAILURE. Data was intentionally written into a memory cell address larger than available. The data was effectively written. Either the sizing scheme failed to find the true last cell or addressing out of memory affects actual memory.

Either halt can be diagnosed by the following register values:

AC2 = the value read from the memory cell.

AC3 = address of the cell failure.

For halt 216 if the carry flag is on when the halt occurred, the memory failed following readout of the location just stored. This condition suggests faulty currents at the failing location. If the carry flag is off when the halt is executed, the location failed after one successful read. This condition suggests another address may affect the failing address.

### 3.6.4 Error Messages

None

## 3.7 CHECKERBOARD MEMORY TEST

Test Number: 6
Bootstrap Switch Settings: 100630
Entry Point: 2

The test may not be re-entered via the entry point since it moves itself randomly in core.

Option Switches:

0 on = 1024 read/write disturbs
15 on = enable halt on error

### 3.7.1 Test Description

The checkerboard memory test checks the proper operation of the memory sense amps, inhibit drivers, and currents. Setting switch zero activates a read/write disturb cycle and may aid in catching marginal failures. If an error is detected, the main terminal bell sounds.

### 3.7.2 Execution Procedure

Checkerboard is loaded into memory by using "reset-program load" with the correct switch setting. When the diagnostic is loaded, it sizes core and outputs "LAST LOCATION TESTED =" and the octal number of the last memory location. If this number is not the last core cell, an entire memory block is failing.

For each successful pass through core, the diagnostic outputs the word "PASS" to the main terminal. Depending on the core size, each pass requires approximately two minutes.

A memory error sounds the main terminal bell if switch 15 is off (down) or causes a program halt if switch 15 is on (up).

### 3.7.3 Halts

The test halt is enabled by setting switch 15 to on. A halt occurs if a memory error is detected. The halt address changes as the program relocates itself in core. The error is diagnosed by accumulators AC0 through AC3 which contain:

AC0 = correct memory pattern

AC1 = the error word

AC2 = the error address

AC3 = address of program

### 3.7.4 Error Messages

None

## 3.8 CPU ARITHMETIC AND LOGIC TEST

Test Number: 7

Bootstrap Switch Settings: 100730

Entry Point: 2

This program may not be reentered via the entry point since it relocates itself randomly in core.

Option Switches:

0 = proceed from error

1 = inhibit error printout

2 = print failure rate

### 3.8.1 Test Description

The arithmetic and logic test exercises the arithmetic and logic instructions of the central processing unit. The program adjusts parameters to the size of memory and relocates itself randomly in core.

### 3.8.2 Execution Procedure

The arithmetic and logic test is loaded into memory by issuing "reset-program load" with the correct switch settings. When the program is loaded, it sizes memory and outputs the message, "LAST LOCATION IN MEMORY IS XXXXX." If this value does not equal the last location, an entire bank of core may be failing.

The program begins execution and tests the arithmetic and logic instructions of the central processing unit. At the end of each iteration, the word "PASS" is output to the main terminal. The test relocates to a new memory location. The program continues operation until manually stopped.

Because this program relocates itself, it is not a viable test to use when setting a troubleshoot loop. This test is intended to be a pass/no pass test and should be used within these constraints.

### 3.8.3 Halts

| Halt | Description |
|---|---|
| 144 | Program halted in the middle of relocating in memory. |

### 3.8.4 Error Messages

When an error is detected by the program the following will be printed:

| | |
|---|---|
| ABS PC | The memory location of the error subroutine call. |
| LIST PC | Where to look in the listing to find the failing routine. |
| ORIGINAL AC0, AC1, AC2 | The accumulator values as determined via the random number generator. |
| RESULT AC0, AC1, AC2 | The result in the accumulators prior to the error call |

After the printout the program will iterate the failing routine with the same numbers as printed. Console switch 2 may be used to determine the rate of failure. Switch 0 will cause the program to proceed to the next test.

## 3.9 50/51 TERMINAL DIAGNOSTIC

Test Number: 10
Bootstrap Switch Settings: 101030
Entry Point: 400
Option Switches: None

### 3.9.1 Test Description

The 50/51 terminal diagnostic detects faults in a 50/51 terminal. The diagnostic consists of four tests run consecutively. The diagnostic halts if an error is detected.

Tests two and three are used to check the functioning of all memory bit positions. Test two outputs all question marks (? = 77 octal) and test three outputs all at signs (@ = 100 octal).

#### TEST ONE

**Edge test.** Output question marks (?) along the extreme upper, lower, left, and right print positions on the screen.

#### TEST TWO

**Question mark memory test.** Output question marks to each memory position of the terminal.

#### TEST THREE

**At sign test.** Output at signs (@) to each memory position of the terminal.

#### TEST FOUR

**Echo test.** Echo characters entered on the keyboard back to the terminal.

### 3.9.2 Execution Procedure

The diagnostic is loaded into memory by issuing "reset-program load" with the correct switch setting. Once the diagnostic is loaded, test one is immediately executed. The diagnostic consists of four tests run consecutively. Press ESC to advance to the next test. After test four, press ESC to reset to test one. Any detected error causes a halt.

When tests one, two, and three are complete, the cursor is positioned at the lower-right corner of the screen. This last position remains blank. An additional character in this position causes the first line to scroll.

### 3.9.3 Halts

All detected errors cause halts.

| Halt | Description |
|---|---|
| 402 | No interrupt error. An I/O command was issued to the terminal, but the terminal did not interrupt within approximately one second. |
| 403 | Invalid device code. The diagnostic was sending data but the interrupting device code was not 51 or the diagnostic was in the echo test waiting for a character from the keyboard but the interrupting device code was not 50. |

### 3.9.4 Error Messages

None

## 3.10 MUX TERMINAL DIAGNOSTIC

Test Number: 11
Bootstrap Switch Settings: 101130
Entry Point: 400
Option Switches: None

### 3.10.1 Test Description

The MUX terminal diagnostic detects faults in a QUAD-MUX board or attached terminals. All terminals attached to the multiplexor board (1–4) are tested. The diagnostic runs four tests on all terminals simultaneously.

Tests two and three are used to check the functioning of all memory bit positions. Test two outputs all question marks (? = 77 octal) and test three outputs all at signs (@ = 100 octal).

#### TEST ONE

**Edge test.** Output question marks (?) along the extreme upper, lower, left, and right print positions on the screen.

#### TEST TWO

**Question mark memory test.** Output question marks to each memory position of the terminal.

#### TEST THREE

**At sign test.** Output at signs (@) to each memory position of the terminal.

#### TEST FOUR

**Echo test.** Echo characters entered on any keyboard to all terminals attached to the multiplexor. This test can be used to detect a malfunctioning terminal or port in the multiplexor.

### 3.10.2 Execution Procedure

The diagnostic is loaded into memory by issuing "reset-program load" with the correct switch setting. Once the diagnostic is loaded, test one is immediately executed. The diagnostic consists of four tests run consecutively. Press ESC to advance to the next test. After test four, press ESC to reset to test one. Any detected error causes a halt.

When tests one, two, and three are complete, the cursor is positioned at the lower-right corner of the screen. This last position remains blank. An additional character in this position causes the first line to scroll.

### 3.10.3 Halts

All detected errors cause halts.

| Halt | Description |
|---|---|
| 402 | No interrupt error. An I/O command was issued to a terminal, but the terminal did not interrupt within approximately one second. |
| 403 | Invalid device code. The multiplexor interrupted with the wrong device code. |
| 404 | Bad MUX interrupt. The diagnostic expected a transmit interrupt but a receive interrupt occurred or a receive interrupt was expected and a transmit interrupt occurred. |
| 405 | Wrong terminal interrupt. The diagnostic was sending to one port but the transmit interrupt was received from another port. For example, output to line one but the interrupt occurs on line zero. |

### 3.10.4 Error Messages

None

## 3.11 10/11 TERMINAL DIAGNOSTIC

Test Number: 12
Bootstrap Switch Settings: 101230
Entry Point: 400
Option Switches: None

### 3.11.1 Test Description

The 10/11 terminal diagnostic detects faults in a 10/11 terminal. The diagnostic consists of four tests run consecutively. The diagnostic halts if an error is detected.

Tests two and three are used to check the functioning of all memory bit positions. Test two outputs all question marks (? = 77 octal) and test three outputs all at signs (@ = 100 octal).

#### TEST ONE

**Edge test.** Output question marks (?) along the extreme upper, lower, left, and right print positions on the screen.

#### TEST TWO

**Question mark memory test.** Output question marks to each memory position of the terminal.

#### TEST THREE

**At sign test.** Output at signs (@) to each memory position of the terminal.

#### TEST FOUR

**Echo test.** Echo characters entered on the keyboard back to the terminal.

### 3.11.2 Execution Procedure

The diagnostic is loaded into memory by issuing "reset-program load" with the correct switch setting. Once the diagnostic is loaded, test one is immediately executed. The diagnostic consists of four tests run consecutively. Press ESC to advance to the next test. After test four, press ESC to reset to test one. Any detected error causes a halt.

When tests one, two, and three are complete, the cursor is positioned at the lower-right corner of the screen. This last position remains blank. An additional character in this position causes the first line to scroll.

### 3.11.3 Halts

All detected errors cause halts.

| Halt | Description |
|---|---|
| 402 | No interrupt error. An I/O command was issued to the terminal, but the terminal did not interrupt within approximately one second. |
| 403 | Invalid device code. The diagnostic was sending data but the interrupting device code was not 11 or the diagnostic was in the echo test waiting for a character from the keyboard but the interrupting device code was not 10. |

### 3.11.4 Error Messages

None

## 3.12 GENERAL COPY UTILITY

Test Number: 13
Bootstrap Switch Settings: 101330
Entry Point: 400
Option Switches: None

### 3.12.1 Test Description

The general copy utility provides the capability to copy any spindle and unit to another spindle and unit. When the copy is complete, a verification run can be initiated to reread and validate that both copies are duplicates.

This utility can copy both single and double-density disk packs. The density of the disk pack is detected by the program issuing a seek to the last cylinder.

### 3.12.2 Execution Procedure

The copy utility is loaded into memory by issuing "reset-program load" with the correct switch settings.

When the utility is loaded into core, the message:

GENERAL COPY UTILITY
MOUNT PACK(S), TYPE ESCAPE.

appears on the main terminal. Mount the correct pack(s) and press ESC when the drive(s) is ready.

The copy utility outputs the following questions:

FROM UNIT (0-3):
FROM PLATTER (R OR F):
TO UNIT (0-3):
TO PLATTER (R OR F):

to determine the direction of the copy. If ESC is entered to any of these questions the utility restarts at the "FROM UNIT:" question.

When all four questions are answered, the copy utility outputs a message stating the direction of the copy. The user must enter a "Y" to start the copy process or an "N" to return to the "FROM UNIT:" question.

If "Y" is entered, the utility issues a seek to each spindle of the copy to determine the density. The message "SINGLE DENSITY DRIVE(S)" or "DOUBLE DENSITY DRIVE(S)" appears on the main terminal. The utility begins the copy and outputs "COPY IN PROGRESS" on the main terminal. Once the copy is started, it continues to completion or an unrecoverable error is encountered.

At the successful completion of the copy the messages "COPY COMPLETE" and "VERIFY COPY (Y OR N)?" appear on the main terminal. The copies can be verified by rereading and comparing each copy or the verification can be passed by entering "N". If the verification is requested, the message "VERIFICATION IN PROGRESS" is output to the main terminal. Upon completion of a successful verification the messages "VERIFICATION COMPLETE – GOOD COPY" and "TYPE ESCAPE TO RESTART PROGRAM" appears on the main terminal.

### 3.12.3 Halts

| Halt | Description |
|---|---|
| 3 | Power failure. A power failure occurred during the copy process. The copy must be restarted. |
| 402 | Density test failure. A disk error occurred when issuing a seek to determine the density of the disk drive. The copy must be restarted. |
| 403 | Density error. The "from" spindle and "to" spindle are not of equal density. This error can be caused by a faulty drive or controller. |
| 1367 | Disk read failure. A sector could not be read after five retries. Press CONTINUE for five more retries. |
| 1375 | Disk write failure. A sector could not be written after five retries. Press CONTINUE for five more retries. |

### 3.12.4 Error Messages

#### DISC READ FAILURE. STATUS IN AC0.

This error denotes that a sector of the disk pack could not be read after five retries. Press CONTINUE for five more retries. This message corresponds to halt 1367.

#### DISC WRITE FAILURE. STATUS IN AC0.

This error denotes that a sector could not be written after five retries. Press CONTINUE for five more retries. This message corresponds to halt 1375.

#### READY DRIVE(S). TYPE ESCAPE

All drives necessary for the copy are not ready. Power up and ready drives; press ESC.

#### VERIFICATION ERROR
#### BAD COPY
#### TYPE ESCAPE TO RESTART PROGRAM.

A verification error occurred in matching the copies. The copies are not duplicates. A recopy is started by pressing ESC and re-entering the copy information.

## 3.13 MUX 8 TERMINAL DIAGNOSTIC

Test Number: 14
Bootstrap Switch Settings: 101430
Entry Point: 400
Option Switches: None

### 3.13.1 Test Description

The MUX-8 terminal diagnostic detects faults in a QUAD-MUX board or attached terminals. All terminals attached to the multiplexor board (1–8) are tested. The diagnostic runs four tests on all terminals simultaneously.

Tests two and three are used to check the functioning of all memory bit positions. Test two outputs all question marks (? = 77 octal) and test three outputs all at signs (@ = 100 octal).

#### TEST ONE

**Edge test.** Output question marks (?) along the extreme upper, lower, left, and right print positions on the screen.

#### TEST TWO

**Question mark memory test.** Output question marks to each memory position of the terminal.

#### TEST THREE

**At sign test.** Output at signs (@) to each memory position of the terminal.

#### TEST FOUR

**Echo test.** Echo characters entered on any keyboard to all terminals attached to the multiplexor. This test can be used to detect a malfunctioning terminal or port in the multiplexor.

### 3.13.2 Execution Procedure

The diagnostic is loaded into memory by issuing "reset-program load" with the correct switch setting. Once the diagnostic is loaded, test one is immediately executed. The diagnostic consists of four tests run consecutively. Press ESC to advance to the next test. After test four, press ESC to reset to test one. Any detected error causes a halt.

When tests one, two, and three are complete, the cursor is positioned at the lower-right corner of the screen. This last position remains blank. An additional character in this position causes the first line to scroll.

### 3.13.3 Halts

All detected errors cause halts.

| Halt | Description |
| --- | --- |
| 402 | No interrupt error. An I/O command was issued to a terminal, but the terminal did not interrupt within approximately one second. |
| 403 | Invalid device code. The multiplexor interrupted with the wrong device code. |
| 404 | Bad MUX interrupt. The diagnostic expected a transmit interrupt but a receive interrupt occurred or a receive interrupt was expected and a transmit interrupt occurred. |
| 405 | Wrong terminal interrupt. The diagnostic was sending to one port but the transmit interrupt was received from another port. For example, output to line one but the interrupt occurs on line zero. |

### 3.13.4 Error Messages

None

## 3.14 REAL-TIME CLOCK DIAGNOSTIC (54)

Test Number: 15
Bootstrap Switch Settings: 101530
Entry Point: 2
Option Switches: None

### 3.14.1 Test Description

The Real-Time Clock Diagnostic checks the logic operation of the real-time clock. The device code of the clock is assumed to be 54.

The test runs in two phases. Phase one validates the busy, done, and interrupt properties of the clock. Phase two validates the timing characteristics.

### 3.14.2 Execution Procedure

The Real-Time Clock Test is booted into memory by issuing "reset-program load" with the correct switch setting.

When the diagnostic is loaded, phase one testing begins immediately. The message:

```
EXECUTING PHASE ONE
```

appears on the main terminal. Phase One requires about 10-15 seconds to complete. If an error occurs, an error message appears on the main terminal. If phase one is completed normally, the message:

```
PHASE ONE TEST COMPLETE
SELECT OPTION (1 or 2)
```

appears on the main terminal.

The option is a phase two option. Option one allows the program to select the clock frequency. The four available frequencies, line frequency (assumed to be 60 hertz), 10 hertz, 100 hertz, and 1000 hertz, are cycled through automatically under option one. Each frequency is run for 15 seconds.

Option two allows the choice of frequency. The chosen frequency is run for the full one minute phase two test cycle. If option two is chosen, the messages:

```
SELECT CLOCK FREQUENCY

0 = LINE FREQUENCY (ASSUMED TO BE 60 HERTZ)
1 = 10 HERTZ
2 = 100 HERTZ
3 = 1000 HERTZ

ENTER CHOICE:
```

appear on the main terminal. The clock frequency is selected by entering 0–3.

Once option one or option two is properly selected, the message:

TYPE ESCAPE TO START 5 SECOND TIMER TEST.

appears on the main terminal. This allows the user to synchronize with an external clock. When ESC is entered, an audible tone is output. This tone is repeated every five seconds. The accuracy of the time must be checked against an external clock. If the tone does not occur every five seconds, either the timer accuracy of the real-time clock is faulty or the terminal is failing. This five second tone test will continue for 60 seconds. All frequencies of the clock should be checked. If it is necessary to terminate phase two prematurely, type ESC to complete phase two after the present five second interval. If an error other than a time accuracy occurs during phase two, an error message appears on the main terminal.

When phase two is complete, the message:

PHASE TWO TEST COMPLETE,TYPE ESCAPE TO RESTART TEST.

appears on the main terminal. Type ESC to restart the real-time clock test at phase one.

### 3.14.3 Halts

| Halt Address | Description |
|---|---|
| 3 | Power Failure. A power failure occurred during the diagnostic run. |
| 401 | Program Error. A program error occurred during execution. This halt and register values should be reported to the diagnostic group for correction. |

### 3.14.4 Error Messages

Any errors detected during the test result in an error message displayed on the main terminal. The error message includes four numbers, N1 through N4, which are the four registers AC0 through AC3.

PHASE ONE ERRORS:

If the error occurs during phase one, the message:

ERROR IN PHASE ONE.
N1 N2 N3 N4

appears on the main terminal.

For phase one, number N4 is the key to the error.

| Value of N4 | Description |
|---|---|
| 1253 | Busy flag of real-time clock did not go to zero on an I/O reset command. |
| 1257 | Done flag of real-time clock did not go to zero on an I/O reset command. |
| 1266 | Clock did not interrupt at 60 hertz. |

| Value of N4 | Description |
|---|---|
| 1274 | NIOC did not clear clock busy flag. |
| 1302 | I/O reset did not clear clock busy flag. |
| 1312 | Busy flag did not turn on. |
| 1317<br>1335 | Busy flag of clock turned on while doing I/O to other devices. |
| 1351 | Busy flag clock turned on without a start pulse. |
| 1362 | Busy flag of clock cleared without a clear pulse. |
| 1373 | Busy flag of clock cleared during clear pulse to other device. |
| 1402 | Done not turning on. |
| 1407 | Clock busy is gated to SELB at times other than clock select. |
| 1416 | Clock done is gated to SELD at times other than clock select. |
| 1426 | Done flag of clock is not zero after I/O reset. |
| 1436 | Done flag of clock is not zero after clear pulse. |
| 1446 | Done flag of clock is not zero after a start pulse. |
| 1455 | Clock interrupted without a start pulse. |
| 1472 | Clock interrupted after a mask out (MSKO) to inhibit interrupts. |
| 1506 | Clock did not interrupt. |
| 1522 | Clock responded to other device mask out bit. |
| 1535 | Clear pulse to clock did not inhibit interrupt. |
| 1552 | I/O reset did not clear the mask out bit on the clock. |
| 1563 | Clock did not interrupt at 1000 hertz. |
| 1574 | Clock did not interrupt at 100 hertz. |
| 1605 | Clock did not interrupt at 10 hertz. |

PHASE TWO ERRORS:

If the error occurs during phase two, the message:

ERROR IN PHASE TWO
EM
N1 N2 N3 N4

appears on the terminal. EM is an appropriate error message. Numbers N1 through N4 are the register values.

WRONG DEVICE CODE

The clock interrupted with the wrong device code.

N1 = device code on interrupt.
N2 = expected clock device code.

BUSY NOT ZERO AT INTERRUPT

The clock interrupted but the busy flag was not zero.

DONE NOT ONE AT INTERRUPT

The clock interrupted but the done flag was not one.

BUSY NOT TURNING ON

The busy flag of the clock did not turn on when a start pulse was issued to the clock.

DONE NOT TURNING OFF

The done flag of the clock did not turn off when a start pulse was issued to the clock.

CLOCK NOT INTERRUPTING

The clock did not interrupt after a start pulse was issued.

## 3.15 TELECOMMUNICATION FAULT ISOLATION

Prior to isolating an apparent system fault, check the following items if the POWER lamp is not illuminated:

1. Loose power cord connection.
2. Improper AC voltage level selected.
3. Blown AC fuse. Replace with AGC 0.5 ampere fuse if defective.

If POWER lamp is illuminated:

1. Verify that all switches are set for normal operation.
2. Check for loose connectors at data terminal equipment, auxiliary equipment (if used), or telephone line equipment.

If this preliminary check of the modem and its connections fails to correct the fault, a systematic check of the system should be conducted. A typical M.W.I. digital data communications network consists of three major components:

1. Data terminal equipment (local and remote).
2. Modems (local and remote).
3. Telephone lines.

Each of these components is isolated and checked for normal operation. During the following procedure an operator should be present at both local and remote sites.

### 3.15.1 Synchronous Line Adaptor Diagnostic

Test Number: 15
Bootstrap Switch Setting: 101630
Entry Point: 400
Option Switches: None

#### 3.15.1.1 Test Description

The synchronous line adaptor diagnostic checks the logic operation of the 4074 SLA board. The SLA must be connected to a modem supporting DTE loopback.

The test runs until manually stopped. The message 'PASS' is output to the main terminal after a specific number of successful passes through the diagnostic.

#### 3.15.1.2 Execution Procedure

The synchronous line adaptor diagnostic is loaded into memory by issuing "reset-program load" with the correct switch settings.

When the diagnostic is loaded, the messages:

SYNCHRONOUS LINE ADAPTOR DIAGNOSTIC
PUT MODEM IN DTE LOOPBACK, TYPE ESCAPE.

appear on the main terminal. Set modem switch to 'DTE' loopback and press ESC on the keyboard; the diagnostic immediately begins execution. After a specific number of successful passes, the message 'PASS' appears on the main terminal. If an error occurs, an error message is output to the main terminal. The diagnostic continues to run until mannually stopped.

Upon completion of the test and the computer halted, set the modem switch from 'DTE' loopback to the normal position.

### 3.15.1.3 Halts

| Halt | Description |
|---|---|
| 3 | Power failure. A power failure occurred during the diagnostic run. |
| 403 | Program error. A program error occurred during execution. This halt and register values should be reported to the diagnostic group for correction. |

### 3.15.1.4 Error Messages

Errors detected during the test display an error message on the main terminal. The message appears in the following form:

ERROR:
ERROR MESSAGE.
N1 N2 N3 N4
TYPE ESCAPE TO RESTART TEST.

The numbers N1 through N4 are the four registers AC0 through AC3. These numbers are documented with the error message if their value is significant.

TI INTERRUPT WHEN RI EXPECTED, OR RI INTERRUPT WHEN TI EXPECTED.

N1 = expected interrupt
N2 = actual interrupt

BAD LINE NUMBER.

N2 = SLA I/O command. Bits 2-7 are the line number.

SLA NOT INTERRUPTING FOR RECEIVE CHARACTER.

SLA DONE NOT TURNING ON.

WRONG DEVICE CODE.

N1 = Actual Device Code
N2 = Expected Device Code

SLA BUSY NOT TURNING OFF.

TI INTERRUPT WHEN NOT EXPECTED.

CHARACTER CHANGED DURING TRANSMISSION.

N1 = expected character
N2 = actual character

SLA NOT INTERRUPTING FOR TRANSMIT.

RI INTERRUPT WHEN NOT EXPECTED.

ALL DONE FLAGS NOT ZERO AFTER I/O RESET.

ALL BUSY FLAGS NOT ZERO AFTER I/O RESET.

INTERRUPT ENABLE NOT ZERO AFTER MSKO INSTRUCTION.

SLA DONE NOT ZERO AFTER I/O RESET.

SLA BUSY NOT ZERO AFTER I/O RESET.

INTERRUPT ENABLE NOT ON AFTER CPU START PULSE.

A DIAC RETURNED DATA FROM SLA WITHOUT START PULSE.

A DIB RETURNED DATA FROM SLA WITHOUT START PULSE.

SLA BUSY NOT TURNING ON.

SLA BUSY GATED WHEN NOT SELECTED.

DOB MODE 0 WITH SLA SELECT NOT CLEARING SLA BUSY.

DOB MODE 0 WITHOUT SLA SELECT CLEARED SLA BUSY.

DOB MODE 1 AFFECTING SLA BUSY FLAG.

DOB MODE 2 AFFECTING SLA BUSY FLAG.

DOB MODE 3 AFFECTING SLA BUSY FLAG.

SLA RESPONDING TO WRONG DEVICE CODE.

N2 = SLA I/O command. Bits 10–15 are the device code.

SLA DONE GATED WHEN NOT SELECTED.

MSKO NOT INHIBITING SLA INTERRUPT.

SLA DONE NOT TURNING OFF.

IMPROPER TIME DIFFERENCE BETWEEN FIRST TWO CHARACTERS OUTPUT TO SLA.

TIME BETWEEN TI'S NOT EQUAL AFTER FIRST ONE.

TI NOT TURNING ON.

DEVICE CODE ON I/O BUS AFTER I/O RESET.

SLA RESPONDING TO WRONG MASK OUT BIT.

SLA NOT RESPONDING TO PROPER MASK OUT BIT.

SLA MASK BIT NOT CLEARING ON I/O RESET.

SLA NOT INTERRUPTING AT FIXED RATE.

A DIB DID NOT RETURN CARRIER DETECT FROM MODEM.

### 3.15.2 Local Modem

If the local data terminal equipment is operating properly, perform a "back-to-back" test on the local and remote modems. This test is performed by using the internal pseudo-random test data generator as follows:

1. Set the MODE switch to LINE LOOPBACK.

2. Set the DATA switch to TEST.

3. POWER, TEST, and SIGNAL DETECT lamps should be illuminated.

NOTE: Other lamps, on panel with nine indicators, also are illuminated. These can be ignored for purposes of this test.

4. The TEST lamp should not blink off during this test. If it does blink the modem is not operating properly and should be replaced.

### 3.15.3 Telephone Lines (4-wire dedicated or dial-up networks)

Set the local modem MODE switch to NORMAL; set remote modem MODE switch to LINE LOOPBACK. This places the telephone lines in the test system along with the local modem and isolates the remote modem and data terminal equipment. Set the local modem DATA switch to TEST. The pseudo-random test pattern is transmitted over the telephone lines and looped back to the receiver of the local modem. Sensed errors cause the local modem TEST lamp to blink off. More than 10 blinks within seven minutes constitute an unacceptable error rate. Notify the telephone company that the telephone lines may require service (leased lines) or hang up and re-dial (DDD network).

The telephone company test center will instruct you to enter the data mode via the data key (exclusion key) on your data station and depress the TEST switch. Instructions will be given prior to entering the data mode to return to the talk mode and release the TEST switch after a certain period of time. Upon completion of the test, the results can be discussed and subsequent action, if necessary, can be determined.

### 3.15.4 Remote Modem

The remote modem usually is tested as part of the procedure for the local modem. However, it may be necessary to test the remote modem only (e.g., after replacement of a defective modem). Use the procedure in paragraph 3.15.2 to perform the "back-to-back" test on the remote modem.

### 3.15.5 Remote Data Terminal Equipment

Set the remote modem MODE switch to DTE LOOPBACK. With the data terminal equipment isolated from the remote modem, test the remote data terminal equipment specified by the DTE manufacturer.

### 3.15.6 Two-Wire Half Duplex Network Test

Prior to performing this test, perform "back-to-back" tests on the local and remote modems. Set the local modem MODE switch to XMIT ONLY; set remote modem switch to RCV ONLY. Set both modem DATA switches to TEST. If the receiving modem TEST indicator blinks off, it indicates faulty telephone lines.

WARNING: Modems with automatic answer option require one of the following steps for 2-wire DDD network test.

1. Set DATA switch to TEST and pick up telephone handset (preferred).

2. Or remove dust cover and set strap F to 1.

Either 1 or 2 above connects the modem through the data coupler to the telephone line. After both sites are connected to the telephone line in this manner, the operators can proceed with the normal 2-wire test procedure.

If the test is satisfactory, reverse the data flow by setting the remote modem MODE switch to XMIT ONLY and the local modem MODE switch to RCV ONLY. Set both modem DATA switches to TEST. If the receiving modem TEST lamp blinks off, the telephone lines are probably defective.

NOTE: Sensed errors cause the TEST lamp to blink off. More than 10 blinks within seven minutes constitute an unacceptable error rate.

### 3.15.7 References

ICC Modem 20 LSI Instruction Manual
Bell System Technical Reference #41801 and #41802
Data General Test and Diagnostic #096-000061-03
Binary Tape #095-000094-04

## 3.16 POWER SHUT-DOWN TEST

Test Number: 17
Bootstrap Switch Setting: 101730
Entry Point: 2
Option Switches: None

### 3.16.1 Test Description

The Power Shut-Down Test tests the power monitor and autorestart option. It may be used with or without the power monitor option.

Memory retention is checked via a checksum and memory address routine.

### 3.16.2 Execution Procedure

The Power Shut-Down Test is booted into memory by issuing "reset-program load" with the correct switch setting.

The main terminal, either 50/51 or mux unit zero, must be plugged into a separate power source. When the power monitor option is tested, all power on the computer baseplate will be removed. If the main terminal is plugged into the baseplate for this test, it is impossible to get error messages due to the warmup time of the CRT. When the Power Shut-Down Test is complete, the main terminal should be reconnected to the appropriate connector at the computer baseplate.

The first test run is a memory size routine. When the size of core is determined, the messages:

POWER SHUT DOWN TEST

LAST CORE LOCATION = XXXXXX

TURN THE COMPUTER OFF. TURN THE COMPUTER ON

AND RESTART AT LOCATION 40

appear on the main terminal. The number XXXXXX should be the last core address. If the number XXXXXX is not the last core location, a hardware component (CPU, memory, etc.) is failing.

**Operation without the Power Monitor**

After the memory size message is displayed, turn the computer off, turn the computer on, and restart at 40 (octal). After approximately three seconds the main terminal bell should ring twice. If an error is detected, an error message appears on the main terminal. This procedure should be repeated several times.

**Operation with the Power Monitor Option**

The procedure outlined under "Operation without the Power Monitor" should be tried several times. After several restarts, lock the computer console and turn off the main power switch on the computer baseplate. Restore the power by turning on the power switch. The computer should restart automatically. After approximately three seconds, the main terminal bell should ring twice. If an error is detected, an error message appears on the main terminal. This procedure should be repeated several times.

### 3.16.3 Halts

Most halts are associated with error messages.

| Halt | Meaning |
|---|---|
| 115 | No interrupt detected on power shut down. |
| 122 | Program sum check error. |
| 217 | Invalid interrupt. Register ACO contains the device code of the device requesting an interrupt. |
| 255 | The machine was restarted without a power fail interrupt. |

### 3.16.4 Error Messages

An error message is displayed on the main terminal whenever a problem is detected.

**Messages**

THE PROCESSOR DID NOT RUN FOR 2 MS AFTER THE POWER FAIL FLAG.

NO INTERRUPT DETECTED ON POWER SHUT DOWN

Either the power fail flag did not set or it did not cause an interrupt.

COMPUTER WAS RESTARTED WITHOUT A POWER FAIL INTERRUPT

A restart at location zero occurred.

PROGRAM SUM CHECK ERROR

Memory has been modified by the power up-down sequence.

C (ADDRESS) = XXXXXX

Memory has been modified by the power up-down sequence. The address and new contents are displayed.

AN INTERRUPT BUT NO POWER FAIL DONE FLAG

POWER TURN ON FAILED TO CLEAR ACCUMULATORS

## 3.17 REAL-TIME CLOCK DIAGNOSTIC (44)

Test Number: 20
Bootstrap Switch Settings: 102030
Entry Point: 2
Option Switches: None

### 3.17.1 Test Description

The Real-Time Clock Diagnostic checks the logic operation of the real-time clock. The device code of the clock is assumed to be 44.

The test runs in two phases. Phase one validates the busy, done, and interrupt properties of the clock. Phase two validates the timing characteristics.

### 3.17.2 Execution Procedure

The Real-Time Clock Test is booted into memory by issuing "reset-program load" with the correct switch setting.

When the diagnostic is loaded, phase one testing begins immediately. The message:

EXECUTING PHASE ONE

appears on the main terminal. Phase One requires about 10-15 seconds to complete. If an error occurs, an error message appears on the main terminal. If phase one is completed normally, the message:

PHASE ONE TEST COMPLETE
SELECT OPTION (1 or 2)

appears on the main terminal.

The option is a phase two option. Option one allows the program to select the clock frequency. The four available frequencies, line frequency (assumed to be 60 hertz), 10 hertz, 100 hertz, and 1000 hertz, are cycled through automatically under option one. Each frequency is run for 15 seconds.

Option two allows the choice of frequency. The chosen frequency is run for the full one minute phase two test cycle. If option two is chosen, the messages:

SELECT CLOCK FREQUENCY

0 = LINE FREQUENCY (ASSUMED TO BE 60 HERTZ)
1 = 10 HERTZ
2 = 100 HERTZ
3 = 1000 HERTZ

ENTER CHOICE:

appear on the main terminal. The clock frequency is selected by entering 0–3.

Once option one or option two is properly selected, the message:

TYPE ESCAPE TO START 5 SECOND TIMER TEST.

appears on the main terminal. This allows the user to synchronize with an external clock. When ESC is entered, an audible tone is output. This tone is repeated every five seconds. The accuracy of the time must be checked against an external clock. If the tone does not occur every five seconds, either the timer accuracy of the real-time clock is faulty or the terminal is failing. This five second tone test will continue for 60 seconds. All frequencies of the clock should be checked. If it is necessary to terminate phase two prematurely, type ESC to complete phase two after the present five second interval. If an error other than a time accuracy occurs during phase two, an error message appears on the main terminal.

When phase two is complete, the message:

PHASE TWO TEST COMPLETE, TYPE ESCAPE TO RESTART TEST.

appears on the main terminal. Type ESC to restart the real-time clock test at phase one.

### 3.17.3 Halts

| Halt Address | Description |
|---|---|
| 3 | Power Failure. A power failure occurred during the diagnostic run. |
| 401 | Program Error. A program error occurred during execution. This halt and register values should be reported to the diagnostic group for correction. |

### 3.17.4 Error Messages

Any errors detected during the test result in an error message displayed on the main terminal. The error message includes four numbers, N1 through N4, which are the four registers AC0 through AC3.

PHASE ONE ERRORS:

If the error occurs during phase one, the message:

ERROR IN PHASE ONE.
N1 N2 N3 N4

appears on the main terminal.

For phase one, number N4 is the key to the error.

| Value of N4 | Description |
|---|---|
| 1253 | Busy flag of real-time clock did not go to zero on an I/O reset command. |
| 1257 | Done flag of real-time clock did not go to zero on an I/O reset command. |
| 1266 | Clock did not interrupt at 60 hertz. |
| 1274 | NIOC did not clear clock busy flag. |
| 1302 | I/O reset did not clear clock busy flag. |
| 1312 | Busy flag did not turn on. |

| Value of N4 | Description |
|---|---|
| 1317<br>1335 | Busy flag of clock turned on while doing I/O to other devices. |
| 1351 | Busy flag clock turned on without a start pulse. |
| 1362 | Busy flag of clock cleared without a clear pulse. |
| 1373 | Busy flag of clock cleared during clear pulse to other device. |
| 1402 | Done not turning on. |
| 1407 | Clock busy is gated to SELB at times other than clock select. |
| 1416 | Clock done is gated to SELD at times other than clock select. |
| 1426 | Done flag of clock is not zero after I/O reset. |
| 1436 | Done flag of clock is not zero after clear pulse. |
| 1446 | Done flag of clock is not zero after a start pulse. |
| 1455 | Clock interrupted without a start pulse. |
| 1472 | Clock interrupted after a mask out (MSKO) to inhibit interrupts. |
| 1506 | Clock did not interrupt. |
| 1522 | Clock responded to other device mask out bit. |
| 1535 | Clear pulse to clock did not inhibit interrupt. |
| 1552 | I/O reset did not clear the mask out bit on the clock. |
| 1563 | Clock did not interrupt at 1000 hertz. |
| 1574 | Clock did not interrupt at 100 hertz. |
| 1605 | Clock did not interrupt at 10 hertz. |

PHASE TWO ERRORS:

If the error occurs during phase two, the message:

ERROR IN PHASE TWO
EM
N1 N2 N3 N4

appears on the terminal. EM is an appropriate error message. Numbers N1 through N4 are the register values.

WRONG DEVICE CODE

The clock interrupted with the wrong device code.

N1 = device code on interrupt.
N2 = expected clock device code.

BUSY NOT ZERO AT INTERRUPT

The clock interrupted but the busy flag was not zero.

DONE NOT ONE AT INTERRUPT

The clock interrupted but the done flag was not one.

BUSY NOT TURNING ON

The busy flag of the clock did not turn on when a start pulse was issued to the clock.

DONE NOT TURNING OFF

The done flag of the clock did not turn off when a start pulse was issued to the clock.

CLOCK NOT INTERRUPTING

The clock did not interrupt after a start pulse was issued.

## 3.18 Telecommunication Diagnostic – Remote (Called Site)

Test Number: 21
Bootstrap Switch Settings: 102130
Entry Point: 2
Option Switches: None

### 3.18.1 Test Description

The Telecommunication Remote diagnostic communicates bisynchronously with another central processor over switched telephone lines. The remote diagnostic is similar to the central (calling) telecommunication diagnostic, except that it operates only with an auto-answer data coupler and expects to receive first and then transmit. The remote diagnostic operates at a remote site and must be called by a system using the central (calling) telecommunication diagnostic.

This diagnostic is the final step in the testing of a Mesa/Two communications capability. Prior to running this test the following components should be verified (see Telecommunication Fault Isolation, page 3-35).

| | | |
|---|---|---|
| SLA Board | – | can be tested with SLA diagnostics |
| Modem | – | can be run in self-test mode |
| RTC Board | – | can be tested with RTC diagnostic |
| Auto Answer Data Coupler (1001A or 1001F) | – | can be tested by telephone company |

Once the diagnostic has been loaded into core and the options and transmit record count selected, the test periodically scans the data coupler for an incoming call signal (ring indicator). When the ring indicator is received, "data terminal ready" is set, the call is answered, and the communication between processors is initiated.

The diagnostic operates in two phases. First, the calling system transmits data to the called system. The remote diagnostic expects a certain data pattern - once the received records are verified they are discarded. Second, the called system transmits data back to the caller. The records contain a specific data pattern internally generated by the remote diagnostic.

The processors communicate under the rules of "Bisynchronous Communication" as outlined in IBM #GA-24-3089 and Martin, Wolfe Telecommunication Information Memorandum of October 30, 1974.

### 3.18.2 Execution Procedure

The Telecommunication Remote diagnostic is booted into memory by issuing "reset - program load" with the correct switch setting.

When the diagnostic is loaded, it requests a response to three options:

1. Re-enable on completion.
2. Beep terminal during telecommunication link.
3. Number of records to transmit.

Option one allows the diagnostic to automatically reset to answer another call after the existing communication has been completed. Option two causes the diagnostic to periodically beep the main terminal during the actual telecommunication link.

Option three allocates the number of records, 0 - 60,000, to transmit to the calling system. Approximately 200 records per minute can be transmitted. The total amount of time, in minutes, required for a complete communication is approximately the sum of the records sent and received divided by 200. The number of records to receive is controlled by the calling system.

The records to transmit are generated internally. Each record is one 80-character card image containing numbers from 1 to 60,000. All records received from the calling system are verified for correct content then discarded.

All option entries must be terminated by the TAB character. An option may be re-entered by pressing ESC.

After completing the entries, the main terminal screen is set up to display the options selected, records to transmit, current communication status, and the current count of records, blocks, enquires (ENQ), and negative acknowledgements (NAK); received and transmitted.

The communication status and the record, block, enquiry, and NAK counts are continually updated as changes occur.

The total number of enquires and NAKS (transmitted and received), gives an indication of the quality of the transmission. This figure should not be greater than 7% of the total number of blocks transmitted and received. A figure higher than 7% indicates a poor phone line or equipment problems.

The state of the diagnostic is displayed on the line labeled "communication status". The status may be:

1. Communication Enabled
2. Line Bid
3. Receiving Data
4. Transmitting Data
5. Communication Complete - Re-enabled

6. Communication Complete - Press ESC to Restart

7. Communication Aborted by Operator. Press ESC to Restart. ABORT CODE = code - address

8. Communication Aborted - Re-enabled
ABORT CODE = code - address

9. Communication Failure. PE CODE = code - address

"Communication Enabled" means the diagnostic has been properly loaded with options and record count selected; the diagnostic is scanning the line for a ring indicator.

"Line Bid" means the ring indicator has been detected, the phone answered, "data terminal ready" has been set, and the diagnostic is sampling the phone line for a "line bid" from the calling system.

"Receiving Data" means the diagnostic is receiving blocks of data from the calling system.

"Transmitting Data" means the diagnostic is transmitting blocks of data to the calling system.

"Communication Complete - Re-enabled" means the communication completed normally and the re-enable option (option one) was selected. The line is being scanned for the ring indicator. The options and transmit record count are the same as selected for the first call.

"Communication Complete - Press ESC to Restart" means the communication completed normally and the re-enable option (option one) was not selected. Pressing ESC allows the user to re-enter the options and transmit record count.

"Communication Aborted by Operator" means the operator has requested an abort. An abort may be initiated by the operator at any time by pressing ESC. If a communication is actually running when an abort is requested, five to 10 seconds are required to service the abort since the diagnostic must wait for the present block of data or acknowledgement before transmitting the disconnect sequence to the calling system. The abort code for an operator requested abort is seven. Once the abort has been serviced and the abort message output for "Communication Status," the user may restart the diagnostic by pressing ESC and re-entering the options and transmit record count.

"Communication Aborted - Re-enable" means the communication link was dropped. An abort (other than an operator requested abort) may occur for several reasons as explained under "Error Messages - Abort". An abort is generally caused by the phone line dropping, the calling system failing, or a real-time clock failure.

"Communication Failure" means a serious problem was detected by the diagnostic. A communication failure is generally caused by a hardware problem. After communication failure is output to the terminal, the diagnostic exits to the "system error" routine, outputs a system error message, and continually beeps the terminal.

### 3.18.3 Halts

A power failure will cause the diagnostic to halt. No other halts are coded.

### 3.18.4 Error Messages

Three types of errors can occur:

1. Abort
2. Communication Failure (PE)
3. System Error

An abort is usually caused by the inability to continue the communication. This condition can be caused by a dropped line or by a calling system failure. An abort is a recoverable problem and automatically causes a re-enable to answer an in-coming call (except for the operator requested abort). Continuous aborts may indicate a hardware problem. All aborts cause an abort code to be output in the form:

ABORT CODE = code - address

The code number is the key to the problem. The address number is a pointer into the diagnostic and is not usually relevant in debugging an abort. Abort codes are listed at the end of this section.

A communication failure is usually caused by a detected hardware problem with the MODEM, SLA, or data coupler. A communication failure exits through the system error routine and forces a system error to occur. All communication failures cause a PE code to be output in the form:

PE CODE = code - address

The code number is the key to the problem. The address number is a pointer into the diagnostic and is not usually relevant in debugging a communication failure. Communication failure codes are listed at the end of this section.

A system error can be caused by either a communication failure or by a disk I/O problem.

| Abort Code | Description |
|---|---|
| 1 | Calling central system has dropped the communication link during its data transmission. The calling system transmitted either an "EOT" or a "DLE EOT" to disconnect the telecommunication link. |
| 2 | Calling central system has dropped the communication link during called system's data transmission. The calling central system transmitted either an "EOT" or a "DLE EOT" to disconnect the telecommunication link. |
| 3 | Calling central system failed to respond with new data. The calling central system did not send a new data block in 21 seconds. The called remote system disconnected the telecommunication link by transmitting the disconnect sequence, "DLE EOT". |
| 4 | Calling central system failed to respond to a block of data transmitted by called remote system after 15 retries or enquires. Called remote system transmitted the disconnect sequence. |
| 5 | Block transmission could not be completed in 4 seconds. Blocks should take a maximum of 2.5 seconds to transmit. This abort is usually traceable to the real-time clock interrupting too fast or the SLA failing to interrupt. |

| Abort Code | Description |
|---|---|
| 6 | No valid line bid from calling system in 20 seconds after sensing a ring indicator and answering the phone. |
| 7 | Operator abort requested and serviced. |

| Communication Failure (PE) | Definition |
|---|---|
| 1 | Invalid program status word. The communication program status word has an invalid setting. Either core has been destroyed, a software error, or CPU failure occurred. |
| 2 | Data set not ready. During the communication the "data set ready" signal was not returned on the status input from the SLA. |
| 3 | Invalid number conversion. A number set up for transmission has an invalid digit count. This error can only be caused by a software or CPU failure. |

## 3.19 Telecommunication Diagnostic - Central (Calling Site)

Test Number: 22
Bootstrap Switch Settings: 102230
Entry Point: 2
Option Switches: None

### 3.19.1 Test Description

The Telecommunication Central diagnostic communicates bisynchronously with another central processor over switched telephone lines. The central (calling) diagnostic is similar to the remote (called) telecommunication diagnostic, except that it operates only with a data coupler capable of dialing a remote site and expects to transmit first and then receive. The central diagnostic operates to test the telecommunication capability of a remote MESA/TWO system.

Prior to running this test the following components should be verified (see Telecommunication Fault Isolation, page 3-35).

| | | |
|---|---|---|
| SLA Board | – | can be tested with SLA diagnostics |
| Modem | – | can be run in self-test mode |
| RTC Board | – | can be tested with RTC diagnostic |
| Handset Data Coupler (1000A) | – | can be tested by telephone company |

Once the diagnostic has been loaded into core and the options and transmit record count selected, the test is ready for the operator to dial the remote unit. "Dial remote computer" is output under communication status and line bids are output at one second intervals to the SLA.

The diagnostic runs in two phases. First, the central system transmits data to the remote system. The remote diagnostic expects a certain data pattern - once the received records are verified they are discarded. Second, the remote system transmits data back to central. The records contain a specific data pattern internally generated by the remote diagnostic and checked by the central diagnostic.

The processors communicate under the rules of "Bisynchronous Communication" as outlined in IBM #GA-24-3089 and Martin, Wolfe Telecommunication Information Memorandum of October 30, 1974.

### 3.19.2 Execution Procedure

The Telecommunication Central diagnostic is booted into memory by issuing "reset - program load" with the correct switch setting.

When the diagnostic is loaded, it requests a response to two options:

1. Beep terminal during telecommunication link.

2. Number of records to transmit.

Option one causes the diagnostic to periodically beep the main terminal during the actual telecommunication link.

Option two allocates the number of records, 0 - 60,000, to transmit to the called system. Approximately 200 records per minute can be transmitted. The total amount of time, in minutes, required for a complete communication is approximately the sum of the records sent and received divided by 200. The number of records to receive is controlled by the called system.

The records to transmit are generated internally. Each record is one 80-character card image containing numbers from 1 to 60,000. All records received from the called system are verified for correct content then discarded.

All option entries must be terminated by the TAB character. An option may be re-entered by pressing ESC.

After completing the entries, the main terminal screen is set up to display the options selected, records to transmit, current communication status, and the current count of records, blocks, enquires (ENQ), and negative acknowledgements (NAK), received and transmitted.

The communication status and the record, block, enquiry, and NAK counts are continually updated as changes occur.

The total number of enquires and NAKS (transmitted and received), gives an indication of the quality of the transmission. This figure should not be greater than 7% of the total number of blocks transmitted and received. A figure higher than 7% indicates a poor phone line or equipment problems.

The state of the diagnostic is displayed on the line labeled "communication status". The status may be:

1. Dial Remote Computer

2. Receiving Data

3. Transmitting Data

4. Communication Complete - Press ESC to Restart

5. Communication Aborted by Operator. Press ESC to Restart. ABORT CODE = code - address

6. Communication Aborted - Press ESC to restart.
   ABORT CODE = code - address

7. Communication Failure. PE Code = code - address

"Dial Remote Computer" means the diagnostic has been properly loaded with options and record count selected. The diagnostic is transmitting line bids through the SLA at one second intervals. The operator should pick up the handset and dial the remote computer. When the remote system picks up the call, an answer back tone of about 2 - 3 seconds will be transmitted by the remote system. The operator should now connect the two systems via the button on the dialing handset. The diagnostic should output "transmitting data" after its line bid is acknowledged by the remote system.

The central calling diagnostic allows the operator five minutes to establish a connection with the remote system. After five minutes an abort 8 is issued.

"Receiving Data" means the diagnostic is receiving blocks of data from the called system.

"Transmitting Data" means the diagnostic is transmitting blocks of data to the called system.

"Communication Complete - Press ESC to Restart" means the communication completed normally. Pressing ESC allows the user to restart the test by entering the options and transmit record count.

"Communication Aborted by Operator" means the operator has requested an abort. An abort may be initiated by the operator at any time by pressing ESC. If a communication is actually running when an abort is requested, five to 10 seconds are required to service the abort since the diagnostic must wait for the present block of data or acknowledgement before transmitting the disconnect sequence to the called system. The abort code for an operator requested abort is seven. Once the abort has been serviced and the abort message output for "Communication Status," the user may restart the diagnostic by pressing ESC and re-entering the options and transmit record count.

"Communication Aborted - Press ESC to Restart" means the communication link was dropped. An abort (other than an operator requested abort) may occur for several reasons as explained under "Error Messages - Abort". An abort is generally caused by the phone line dropping, the calling system failing, or a real-time clock failure.

"Communication Failure" means a serious problem was detected by the diagnostic. A communication failure is generally caused by a hardware problem. After communication failure is output to the terminal, the diagnostic exits to the "system error" routine, outputs a system error message, and continually beeps the terminal.

### 3.19.3 Halts

A power failure will cause the diagnostic to halt. No other halts are coded.

### 3.19.4 Error Message

Three types of errors can occur:

1. Abort
2. Communication Failure (PE)
3. System Error

An abort is usually caused by the inability to continue the communication. This condition can be caused by a dropped line or by a called system failure. Continuous aborts may indicate a hardware problem. All aborts cause an abort code to be output in the form:

ABORT CODE = code - address

The code number is the key to the problem. The address number is a pointer into the diagnostic and is not usually relevant in debugging an abort. Abort codes are listed at the end of this section.

A communication failure is usually caused by a detected hardware problem with the MODEM, SLA, or data coupler. A communication failure exits through the system error routine and forces a system error to occur. All communication failures cause a PE code to be output in the form:

PE CODE = code - address

The code number is the key to the problem. The address number is a pointer into the diagnostic and is not usually relevant in debugging a communication failure. Communication failure codes are listed at the end of this section.

A system error can be caused by either a communication failure, or by a disk I/O problem.

| Abort Code | Description |
|---|---|
| 1 | Called remote system has dropped the communication link during its data transmission. The called remote system transmitted either an "EOT" or a "DLE EOT" to disconnect the telecommunication link. |
| 2 | Called remote system has dropped the communication link during calling central system's data transmission. The called remote system transmitted either an "EOT" or a "DLE EOT" to disconnect the telecommunication link. |
| 3 | Called remote system failed to respond with new data. The called remote system did not send a new data block in 21 seconds. The calling central system disconnected the telecommunication link by transmitting the disconnect sequence, "DLE EOT". |
| 4 | Called remote system failed to respond to a block of data transmitted by calling central system after 15 retries or enquires. Calling central system transmitted the disconnect sequence. |
| 5 | Block transmission could not be completed in 4 seconds. Blocks should take a maximum of 2.5 seconds to transmit. This abort is usually traceable to the real-time clock interrupting too fast or the SLA failing to interrupt. |
| 6 | No valid line bid from called remote system after calling central system has completed its data transmission. |
| 7 | Operator abort requested and serviced. |
| 8 | Failed to establish connection with called remote system after five minutes. Once the message "dial remote computer" is output under communication status, the operator has five minutes to establish a connection with the remote system. If an acknowledgement to the line bid is not received within five minutes, an abort - 8 is issued. |
| 9 | Twenty successive NAKs received. A block of data sent by the calling central system was NAK'ed twenty times in succession after twenty retransmissions. This abort indicates a very poor line or hardware problems. The calling central system has dropped the telecommunication link by transmitting the disconnect sequence. |

| Abort Code | Description |
|---|---|
| 10 | Twenty successive NAKS transmitted. A block of data was improperly received at the calling central station after twenty successive retransmissions. This abort indicates a very poor line or hardware problem. The calling central system has dropped the telecommunication link by transmitting the disconnect sequence. |

| Communication Failure (PE) | Definition |
|---|---|
| 1 | Invalid program status word. The communication program status word has an invalid setting. Either core has been destroyed, a software error, or CPU failure occurred. |
| 2 | Data set not ready. During the communication the "data set ready" signal was not returned on the status input from the SLA. |
| 3 | Invalid number conversion. A number set up for transmission has an invalid digit count. This error can only be caused by a software or CPU failure. |

## 3.20 Change Pack ID Utility

Test Number: 23
Bootstrap Switch Setting: 102330
Entry Point: 400
Option Switches: None

### 3.20.1 Test Description

The Change Pack ID utility provides the capability of rewriting the system pack ID.

Every MESA/TWO disk pack must have a unique pack ID. A pack ID consists of five elements:

| | |
|---|---|
| SYSTEM | This number identifies the customer. |
| PACK | This value identifies the "type" of pack. This number should always be zero for customer production. |
| DRIVE | In a single drive system this number is always zero. In a two drive system, this number is set to one on all packs that belong in unit one. The value of drive is the unit it belongs in, unit 0, unit 1, unit 2, etc. |
| M or B | This entry specifies whether the pack is a "main" or "backup". |
| SUBID | This number is always zero for a main. It is one or two for a backup. |

### 3.20.2 Execution Procedure

The Change Pack ID Utility is booted into memory by issuing "reset - program load" with the correct switch setting.

When the diagnostic is loaded, the message:

CHANGE PACK ID UTILITY
MOUNT PACK, TYPE ESCAPE

should appear. Mount the MESA/TWO pack to be changed on unit zero and type escape when the drive is ready. It is important to note that changing a pack ID can only be done on the disk pack in unit zero. Therefore, if a two drive system must have drive zero and drive one pack ID's changed, the two packs must be "cycled through" unit zero.

All digits entered and displayed for data values are decimal numbers. An input value is terminated by TAB. If an input is incorrect, typing escape before TAB will permit the data to be re-entered. If the input is invalid (e.g. DRIVE: 95), the program will redisplay the header and allow the data to be re-entered.

The standard procedure for changing a system ID follows. The N character represents decimal numbers and the A character represents alphabetic information (i.e. M or B for MAIN or BACKUP).

After typing escape, the message:

PRESENT ID IS:

| | |
|---|---|
| SYSTEM: | NNNNN |
| PACK: | NNN |
| DRIVE: | N |
| M/B: | A |
| SUB-ID: | N |

should appear. This display is the current pack identification.

Immediately after this display, the message:

ENTER NEW ID:

SYSTEM:

should appear. Enter the new system number terminated by a TAB. The program will request SYSTEM, PACK, DRIVE, M/B, and SUB-ID in order. Always enter pack = 0 for a customer's production pack. For M/B, enter M for main and B for backup. The SUB-ID is zero for main, one for backup #1, two for backup #2, etc.

Once the SUB-ID has been entered, the new ID will be written out and a "read after write" check will re-read the ID and display it on the terminal. The output should appear similar to:

NEW ID WRITTEN

NEW ID IS:

| | |
|---|---|
| SYSTEM: | NNNNN |
| PACK: | NNN |
| DRIVE: | N |
| M/B: | A |
| SUB-ID: | N |

TYPE ESCAPE TO RESTART

Enter escape to cycle through the program again.

Note that the MESA/TWO system READY command ID? will cause a display in the style

sssss-ppp-X-n

where "s" is the system value, "p" is the pack value, "X" is M or B, and "n" is the subid. The "drive" number is implicit.

For example, a one-drive Levitz store has a main and two backups. The pack IDs on the three packs might be

10342-0-M-0
10342-0-B-1
10342-0-B-2

If this store is a two-drive system the six packs would be

| | | | |
|---|---|---|---|
| 10342-0-M-0 | (unit 0) | 10342-0-M-0 | (unit 1) |
| 10342-0-B-1 | (unit 0) | 10342-0-B-1 | (unit 1) |
| 10342-0-B-2 | (unit 0) | 10342-0-B-2 | (unit 1) |

### 3.20.3 Halts

| Halt Address | Definition |
|---|---|
| 3 | A power failure occurred. Remount diagnostic pack and restart entire procedure. |
| 402 | Select or seek failure. The program could not successfully select unit zero or seek to cylinder zero. |
| 403 | Read after write failure. The "read after write" check could not re-read the ID sector. |

### 3.20.4 Error Messages

Listed below are error messages and recovery actions which may occur during ID change.

1. CANNOT READ DISC

   ENTER NEW ID

   The ID sector could not be read after five retries. A new ID can be entered and the sector will be reformatted and written out. Although the sector is rewritten, the pack cannot be booted as a system pack. Entering a new ID under these circumstances should only be done on new raw packs and never on customer packs.

2. READY UNIT, TYPE ESCAPE

   Escape was entered to the "MOUNT PACK" message before unit zero became ready. Wait for unit zero to come to ready, type escape.

3. WRITE FAILURE. PRESS CONTINUE FOR RETRY.

   The new ID could not be written after five retries. Press continue for five more retries.