S/MT Series Diagnostics Manual

NOTE

This equipment generates, uses, and can radiate radio frequency energy. If not installed and used in accordance with the installation instructions, it may interfere with radio communications. The equipment has been tested and found to comply with the limits for a Class A computing device pursuant to Subpart J of Part 15 of FCC Rules, which are designed to provide reasonable protection against such interference when operated in a commercial environment. Operation of this equipment in a residential area is likely to cause interference, which users may be required to correct at their own expense.

Specifications Subject to Change.

Convergent Technologies and NGEN are registered trademarks of Convergent Technologies, Inc.

Art Designer, Chart Designer, ClusterCard, ClusterNet, ClusterShare, Context Manager/VM, Convergent, CT-DBMS, CT-MAIL, CT-Net, CTIX, CTOS, CTOS/VM, DISTRIX, Document Designer, The Operator, AWS, CWS, IWS, S/50, S/120, S/160, S/220, S/320, S/640, S/1280, Multibus, TeleCluster, Voice/Data Services, Voice Processor, and X-Bus are trademarks of Convergent Technologies, Inc.

First Edition (July 1987) 09-00631-01

Copyright © 1987 by Convergent Technologies, Inc., San Jose, CA. Printed in USA.

All rights reserved. No part of this document may be reproduced, stored in a retrieval system, or translated without the prior written consent of Convergent Technologies, Inc. This document contains proprietary information of Convergent Technologies, Inc. and/or its suppliers.

This document and product features and descriptions are subject to change without notice.

CONTENTS

RELATED DOCUMENTATION..... vii

1 OVERVIEW..... 1-1
ABOUT THIS MANUAL..... 1-2
Who Should Use this Manual..... 1-2
Purposes of this Manual..... 1-2
Manual Organization..... 1-3
How to Use this Manual..... 1-4

2 GETTING STARTED..... 2-1
S/MT SERIES DIAGNOSTIC TESTS..... 2-1
CONNECTING A TERMINAL..... 2-1
Starting Up Terminals in Emulate Mode..... 2-4
CONNECTING THE OPTIONAL PARALLEL LINE PRINTER..... 2-5
LOADING THE DIAGNOSTIC TESTS..... 2-7
With CTIX Running..... 2-7
With the Power Off..... 2-8
THE DIAGNOSTIC LOGIN SCREEN..... 2-9
THEORY OF OPERATION..... 2-10

3 THE COMMAND INTERPRETER..... 3-1
DIAGNOSTIC TEST MODES..... 3-1
Attended Mode..... 3-1
Normal Attended Mode..... 3-2
Long Error Mode..... 3-2
Continuous Error Mode..... 3-2
Unattended Mode..... 3-2
TWO-LETTER COMMANDS..... 3-3
Listing the Two-Letter Commands..... 3-3
Executing the Two-Letter Commands..... 3-4
TESTS AND TEST SEQUENCES..... 3-5
SUBTESTS..... 3-6
List Test Sequence Subtests..... 3-7
COMMAND STRUCTURE..... 3-8
SAMPLE COMMANDS..... 3-10
FINAL DETAILS..... 3-12

4 TWO-LETTER COMMANDS..... 4-1
HELP (HE)..... 4-2
DISPLAY MODE TWO-LETTER COMMANDS..... 4-3
Page Mode (PM)..... 4-3
Line Printer Echo (LE)..... 4-4
DISK AND TAPE TWO-LETTER COMMANDS..... 4-5
Repeat Count (RC)..... 4-5
ECC Error Correction Enable/Disable (EC)..... 4-7
Disk Soft Error Count (SE x)..... 4-8
Display Disk Error Statistics (DD).... 4-9
Clear Disk Error Statistics (CD)..... 4-10
Interrupt Mode (IM)..... 4-10
UNIT SELECTION TWO-LETTER COMMANDS..... 4-11
Disk Drive Number (DN)..... 4-11
Tape Drive Number (TN)..... 4-13
Ethernet Combo Board Number (EN)..... 4-15
VME Communication Controller Card Number (MN)..... 4-15
SYSTEM MODE TWO-LETTER COMMANDS..... 4-16
Reboot (RE)..... 4-16
NMI/Parity Enable (PE)..... 4-17
Cache Enable (CA)..... 4-18
System Priority Level (SP x)..... 4-19
DISPLAY STATUS TWO-LETTER COMMANDS..... 4-21
System Configuration (CN)..... 4-21
Display Errors (DE)..... 4-22
Display Test Sequence Subtests (DS)... 4-22
USER SELECTION TWO-LETTER COMMANDS..... 4-23
Change Console (CO)..... 4-23
User-Defined Test Sequence (#=)..... 4-24
Creating the User-Defined Test Sequence..... 4-25
Executing the User-Defined Test Sequence..... 4-25

APPENDIX A: CABLE SPECIFICATIONS..... A-1

LIST OF FIGURES

Figure		Page
2-1.	Connecting a Terminal to Channel 0 (shown attached to an S/640).....	2-3
2-2.	Attaching The Parallel Line Printer (shown attached to an S/222).....	2-6
2-3.	The Diagnostic Screen.....	2-9

LIST OF TABLES

Table		Page
2-1	Diagnostic Terminal Requirements....	2-2
3-1	Diagnostic Test Sequences.....	3-5
3-2	Clock Subtests.....	3-6
3-3	Contents of Test Sequence D.....	3-7
4-1	System Priority Levels (SPL).....	4-20

RELATED DOCUMENTATION

The documents described below provide additional information related to the contents of this manual.

Installation

- S/120 Installation
- S/220 Installation
- S/221 and S/222 Installation
- S/320 and S/640 Installation

Hardware

- S/220 and S/320 Hardware
- S/120 Addendum to the S/220 and S/320 Hardware Manual
- S/MT CPU-02 Technical Reference
- MightyFrame IOP Expansion Board Hardware
- MightyFrame RS-422 Expansion Board Hardware
- MightyFrame 10-Port and 20-POrt I/O Expansion Boards
- MightyFrame VME Expansion

System Administration

- MightyFrame CTIX Administrator's Reference

The S/120 Installation Manual describes procedures for locating, inspecting, and booting the S/120, and for attaching peripherals and installing expansion boards and hard disk drives. It also includes a summary of system status codes and instructions for building required cables.

The S/220 Installation Manual describes procedures for locating, inspecting, and booting an S/220 system, and for attaching peripherals and installing expansion boards and hard disk drives. The manual also includes a summary of system status codes and instructions for building required cables.

The S/221 and S/222 Installation Manual describes procedures for locating, inspecting, and booting S/221 and S/222 systems, and for attaching peripherals and installing expansion boards and hard disk drives. The manual also includes summaries of system status codes and instructions for building required cables.

The S/320 and S/640 Installation Manual describes procedures for installing, powering on, and booting S/320 and S/640 systems, and for connecting peripherals and adding expansion cards and hard disk drives.

The S/320 and S/220 Hardware Manual (Vols. 1 and 2) provides information on the functional description, software interface, and theory of operation for the S/320 and S/220 computer systems.

The S/MT CPU-02 Technical Reference Manual provides an introduction to the CPU-02 board, which is the main processor board in a Series 02 S/MT system. The manual also includes a theory of operation and a description of the board's software interfaces.

The MightyFrame IOP Expansion Board Hardware Manual contains a functional and circuitry description of the Input/Output Processor board, and a description of its theory of operation and software interfaces.

The MightyFrame RS-422 Expansion Board Hardware Manual describes the hardware on the MightyFrame RS-422 Expansion board including descriptions of clock, bus, and channel control. An onboard line printer interface is also described.

The MightyFrame 10-Port and 20-Port RS-232-C I/O Expansion Boards Manual describes the hardware incorporated on the MightyFrame 10-port and 20-port boards at a functional block and component level, including descriptions of interrupt handling, processor interface, and RS-232-C I/O operations.

The MightyFrame VME Expansion Manual contains the functional and circuitry description and theory of operation for the MightyFrame VME Interface board.

The MightyFrame Series Administrator's Reference Manual defines the responsibilities of a MightyFrame system administrator and provides procedures for the administrator to follow. The manual explains the concepts an administrator must understand to maintain a MightyFrame computer system, including user support, CTIX modes, CTIX file systems, peripheral devices, and troubleshooting procedures.

1 OVERVIEW

The diagnostics enable you to comprehensively test all the equipment and capabilities of any S/MT Series computer system. You can execute the diagnostic tests regardless of your system configuration. The diagnostic performs the tests for equipment that is present and ignores the tests for equipment that is absent.

NOTE: There are two different diagnostic tapes available for S/MT systems. Both are documented in this manual. Each contain tests specific to the CPU board (CPU-01 and CPU-02) Except where called out in the text, the tests for the rest of the system hardware are identical.

The diagnostics tapes are not interchangeable. CPU-02 based diagnostics will not run on a CPU-01 system, and vice versa. Be sure you have the right tape for your system.

The installation guide shipped with your system contains information on the different hardware components available with your system. Systems that can be tested with the diagnostics include:

- S/120 (a CPU-01 based system)
- S/220 (a CPU-01 based system)
- S/221 (a CPU-01 based system)
- S/222 (a CPU-01 based system)
- S/320 (a CPU-01 based system)
- S/640 (a CPU-02 based system)

Consult the current sales guide for a complete list and descriptions of all S/MT Series computer systems and their CPU boards.

ABOUT THIS MANUAL

This manual documents S/MT Series diagnostics for release levels 6.0 and up.

WHO SHOULD USE THIS MANUAL

To use diagnostics you should be familiar with computers in general, and you should have some familiarity with your system. Each diagnostic test is described according to what it tests, the language required to execute the test, and what the test output means.

PURPOSES OF THIS MANUAL

This manual has several purposes. It

- helps you ensure proper computer operation
- shows you how to identify problems when they occur
- refers you to related documentation

MANUAL ORGANIZATION

This manual consists of the following chapters that are organized to provide a straightforward guide to the diagnostic tests. The following paragraphs tell you what chapters to consult for the specific task you want to perform.

- This chapter, "Overview," provides a brief introduction to the S/MT Series diagnostics and this manual.

- Chapter 2, "Getting Started," explains how to properly connect the terminal, the optional parallel printer, and load the diagnostic programs from QIC tape.

- Chapter 3, "The Command Interpreter," describes the language required to execute the diagnostic programs.

- Chapter 4, "Two-Letter Commands," presents the individual descriptions of the two-letter commands.

The sections that follow Chapter 4 describe and explain how to execute the numerous subsystem tests. Each section corresponds to a different piece of hardware, and can be reorganized to meet your needs (a bank of blank tabs is provided for this purpose). Future updates to this manual will be published in this format and can be inserted anywhere.

In addition, Appendix A, "Cable Specifications," contains the pin assignments and describes how to make your own cables and loopback connectors required for some diagnostic tests.

As stated above, each set of diagnostic tests is documented by hardware type and organized into sections at the end of this manual. A set of blank tabs is provided to separate the sections.

Before you begin, organize your manual as follows:

1. Determine which CPU board you have in your system and move the corresponding diagnostic test section directly behind the "Test Summaries" tab.

2. Move the disk and tape diagnostic test section behind the CPU board section, separating them with a blank tab.

3. Determine which hardware components you have in your system and move their test summaries behind the disk and tape section, separating each with a blank tab.

4. Place the unused test summaries at the very back of the manual.

As you run each series of tests, refer to each section for a description of what each test does, how to run the test, and what you will see on the screen as the test is running.

Before beginning, read Chapters 1 through 4 to become familiar with the test requirements and command syntax.

2 GETTING STARTED

This chapter introduces the S/MT Series diagnostic tests, shows you how to connect a terminal and an optional parallel line printer to your system, shows you how to load the diagnostic tests, and describes some of the theory behind them.

S/MT SERIES DIAGNOSTIC TESTS

You can use the diagnostic tests to ensure correct system operation and to diagnose computer malfunctions when they occur. There are three levels in the hierarchy of diagnostic tests:

Tests	evaluate specific S/MT Series subsystems
Subtests	individual parts of a test (which combine to form the test)
Test sequences	predefined combinations of tests and subtests

CONNECTING A TERMINAL

Before you start, make sure that you connect only one terminal to your system and that you connect it to Channel 0 on the Main Processor (CPU) board. You must follow these specifications and directions to ensure that the diagnostics will work properly.

You can use two types of terminals:

- any terminal with asynchronous RS-232-C protocol, such as the Freedom 100 or VT-100 terminals

- a PT or GT used in Emulate mode (see "Starting Up Terminals in Emulate Mode" later in this chapter)

Consult the installation manual for the terminal you are using to make sure it is configured to these characteristics (unless altered, PTs and GTs are already configured to the characteristics contained in Table 2-1).

Table 2-1

DIAGNOSTIC TERMINAL REQUIREMENTS

Baud Rate	Data Bits	Stop Bits	Parity
9600	8	1	none

You need a crossed RS-232-C cable with 25-pin connectors on both ends to complete the connection (see Appendix A, "Cable Specifications," for cabling information and pin assignments).

To connect a terminal to Channel 0 on the Main Processor (CPU) board (see Figure 2-1):

1. Insert the 25-pin connector on one end of the cable into Channel 0 on the Main Processor (CPU) board.

2. Insert the 25-pin connector at the other end of the cable into the RS-232-C channel on the terminal.

NOTE: The diagnostics contain a command that lets you change the port after you load the diagnostic program. See the two-letter command **CO** in Chapter 4, "Two-Letter Commands," to issue diagnostic commands from another port.

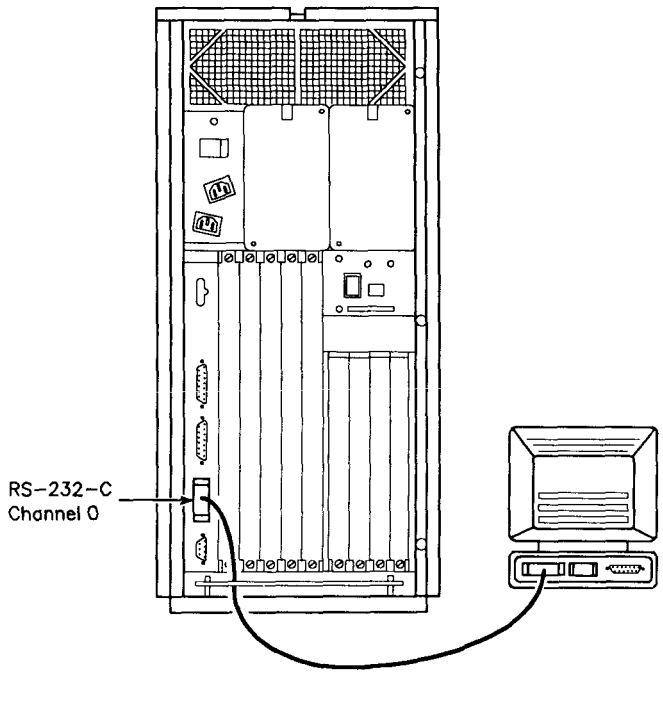


Figure 2-1. Connecting a Terminal to Channel 0

STARTING UP TERMINALS IN EMULATE MODE

If you use a PT or GT, you need to start up that terminal in Emulate mode. When a PT or GT is used in Emulate mode, the system sees the PT or GT as an RS-232-C terminal.

To start up a PT or GT in Emulate mode:

1. Connect the PT or GT to the Main Processor board (see Figure 2-1).

2. Turn the terminal off.

3. Press the space bar while simultaneously turning the terminal on. (The boot ROM prompt does not appear if you release the space bar before the prompt is displayed.)

4. Continue to press the space bar down until the boot ROM prompt is displayed.

   The prompt for a GT with Version 1.0 or 2.0 boot ROM looks like this:

   ```
   V 2.0
   T 222 F 801 R 09 P 00 1
   B,C,E,F,I,P,R,S,T:
   ```

   The prompt for a PT with Version 1.0 boot ROM looks like this:

   ```
   V 1.0
   B,C,E,F,M,R,S,T:
   ```

   The prompt for a PT with Version 2.0 boot ROM looks like this:

   ```
   V 2.0
   B,C,E,F,I,P,R,S,T,W:
   ```

5. Press **e**.

The terminal now starts up in Emulate mode and functions like an RS-232-C terminal. Until you install the diagnostics, the screen will be blank.

NOTE: There are some restrictions for PTs and GTs that contain a 1.0 boot ROM. These restrictions are outlined below.

- You cannot use the Shift key. This means that you cannot use the question mark (?), colon (:), the backslash (\), or any lowercase letters. Use the following keystrokes in place of these.

Replace	With
?	/
:	.
lowercase letters	uppercase letters

- Use the character 8 on the numeric key pad to type a vertical bar (|).

You can still display the command modes and subtests when using a terminal containing a 1.0 boot ROM. Simply type the two letter command (if toggling a command mode) or the test sequence letter (if executing a test sequence). You can also execute tests and subtests by specifying their respective numbers.

CONNECTING THE OPTIONAL PARALLEL LINE PRINTER

An optional parallel line printer can be attached to the main Processor (CPU) board to print the output from the diagnostic tests. This parallel line printer must be connected to the connector marked Parallel Printer on the Main Processor board. (You cannot connect a serial printer to the Main Processor (CPU) board for this purpose.)

You need a parallel printer cable to complete the connection (see Appendix A, "Cable Specifications," for cabling information and pin assignments). Be sure that the end of the cable that attaches to the Main Processor (CPU) board contains a DB25 connector.

To connect the optional parallel line printer (see Figure 2-2):

1. Insert the DB25 end of the Centronics-compatible parallel printer cable into the connector labeled Parallel Printer on the Main Processor board.

2. Insert the connector at the other end of the cable into the connector on the parallel line printer.

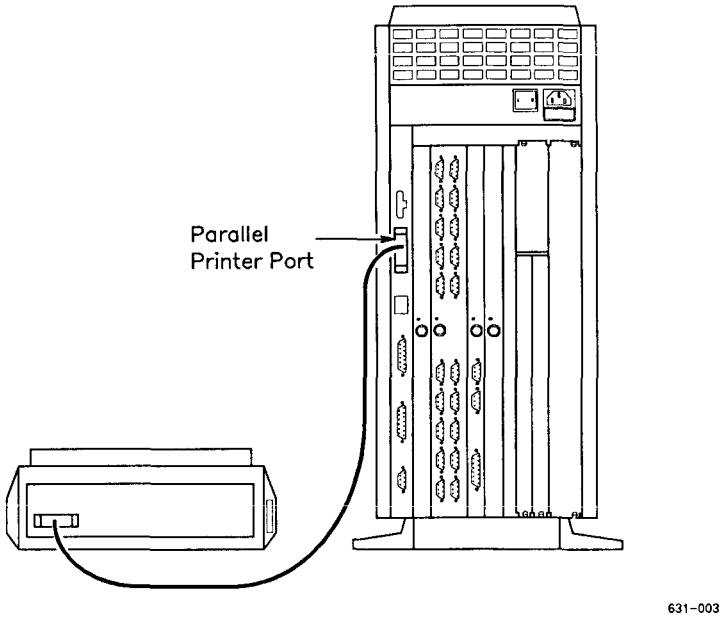


Figure 2-2. Attaching the Parallel Line Printer (shown attached to an S/222)

When the printer is connected to the Main Processor (CPU) board, you can instruct the diagnostic to echo its output to the printer (operator input is not printed):

1. Type **LE**

2. Press **Return**.

This message is displayed to confirm that the line printer is enabled:

Line Printer Enabled

LOADING THE DIAGNOSTIC TESTS

After you connect the terminal, and the line printer if necessary, you can load the diagnostic tests. The diagnostic tests are loaded from QIC tape. You can load the diagnostic tests when the system is running CTIX or when the power is turned off. The following information describes each situation.

NOTE: If you power up your system with the QIC tape in the tape drive and without first connecting a terminal to Channel 0, the diagnostic executes system test sequence d (described in Chapter 3, "The Command Interpreter") until an error occurs. If an error does occur, the error LED on the front status panel lights up. The error message is displayed when you connect a terminal to Channel 0 on the Main Processor board. See the installation manual for your system for a summary of error and status codes. See the hardware manual for your system for a complete description of the boot sequence.

WITH CTIX RUNNING

When CTIX is running, you must terminate all currently running processes before loading the diagnostic tests. See the MightyFrame CTIX Administrator's Reference Manual for instructions on properly shutting down the operating system.

When the prompt

```
OK to stop or reset processor
```

appears on the screen, load the diagnostics as follows:

1. Insert the diagnostic tape into the tape drive.

2. Type **reboot**

3. Press **Return**.

   Alternately, you can press down the Reset switch on the front of the Main Processor (CPU) board.

   Figure 2-3 shows the screen that is displayed when the diagnostics are fully loaded.

WITH THE POWER OFF

To load the diagnostic tests when power is off:

1. Insert the diagnostic tape into the tape drive.

2. Move to the rear of your system and press the power switch to the On position. (If you have one or more V-Boxes attached to your system, press the reboot switch on each immediately after powering them on.)

The system powers up and loads the diagnostic tests into memory. Figure 2-3 shows the screen that is displayed when the tests are fully loaded.

THE DIAGNOSTIC SCREEN

The diagnostic screen (shown in Figure 2-3) contains some important information. It displays the

- system configuration, which lists the equipment acknowledged by the diagnostic
- location in memory occupied by the diagnostic programs
- diagnostic command prompt

This information becomes more significant as you progress with this manual. For now, leave the screen displayed on the terminal while you continue with this chapter.

```
MIGHTYFRAME MC68020 DIAGNOSTIC DEBUGGER V2.0

COPYRIGHT 1983,1984,1985,1987 BY CONVERGENT TECHNOLOGIES

MIGHTYFRAME DIAGNOSTICS-V <6.XX.XX> release (XX)
(XXX XXX X XX:XX:XX PST 19XX - /usr/src/miti/diag)

System configuration:

        X mega bytes of memory
        XMb memory board in slot MX
        MC68881 coprocessor is not present
        VME expansion board is not present
        Ethernet Combo board in slotX is configured for dma slotX
        RS232 channels 2 to 11 22 to 27 in slots:  IO1 IO3
        IOP board is present
        Drive 0 is present [6.0bt]
        Drive 1 is present [85 Hit]
        Drive 2 is present [5.1bt]
Memory tests will begin at 0xXXXXX, end at 0xXXXXXX
Parity interrupts enabled, Disk/QIC Interrupt Mode enabled
The time on the clock calendar is:  XXX XXX XX XX:XX:XX GMT 19XX

command>
```

Figure 2-3. The Diagnostic Screen

Listed below are several points to keep in mind while you execute the diagnostic tests:

- except when each subtest is announced, there is usually no feedback to indicate that a test, subtest, or test sequence is successful. Feedback typically occurs only to report errors.

- unless tests are executed in long or continuous error mode (described in Chapter 3), errors reset the diagnostic and display the system configuration.

- the term page refers to a 4K byte block of memory located from least significant address bits 0x000 to 0xFFF.

- subtest 0 always refers to a predefined group of subtests.

- the data path check, used by several tests, uses the "walking 0 and walking 1" technique. For example, subtest 1 of the RS-232-C test sequence executes data path checks to the baud rate register of each RS-232-C channel that is present.

- the escape character, ESC (Go on a PT/GT), causes the diagnostic to be reset and the system configuration to be displayed. Use this to stop a diagnostic test, to get out of an operator input question if you are confused about what is asked, or if an incorrect response was entered.

  This escape character is more powerful than q, which is not always recognized as a quit character, especially during normal keyboard input.

3 THE COMMAND INTERPRETER

This chapter describes the command syntax required to execute the diagnostic tests. Specifically, it explains test modes, two-letter commands, tests, single-letter test sequences, and subtests.

Before you continue, make sure that you connected a terminal to Channel 0 on the Main Processor (CPU) board and that you loaded the diagnostic software.

DIAGNOSTIC TEST MODES

There are two test modes available in the diagnostic program. They are listed below and described on the following pages:

- attended mode
  - normal mode
  - long error mode
  - continuous error mode
- unattended mode

ATTENDED MODE

Attended mode is set automatically when a terminal is connected to Channel 0 on the Main Processor board. In this mode, you issue commands through the terminal to execute the diagnostic tests and to specify the desired error-reporting mode default stop on error (Long or Continuous). Errors encountered in attended mode flash the error LED on the front status panel. This mode is used throughout this manual.

Normal Attended Mode

In the normal attended mode, an error will cause the diagnostic to rest after printing the error message.

Long Error Mode

This error-reporting mode writes error messages to a buffer in memory. To conserve space, the error messages written to memory are typically shorter than those displayed immediately on the terminal screen. The test continues once the error message has been logged. The two letter command **DE** enables you to view the error log a second time. Long error mode displays error messages under these three circumstances:

- the full sequence of commands is complete
- the log-buffer overflows
- the user quits the test by pressing **ESC** (Go on a PT or GT) or **q**

Continuous Error Mode

This error-reporting mode displays error messages on the terminal screen. It does not write them to a buffer in memory. The test or test sequence continues once the error message is displayed.

UNATTENDED MODE

Unattended mode is entered automatically when your system is turned on with the QIC diagnostic tape in the tape drive and without a properly cabled terminal connected to Channel 0. (The presence of a terminal is detected when Data Set Ready (DSR) is asserted.)

Unattended mode causes the diagnostic to execute test sequence d until an error is encountered. The error flashes the error LED on the status panel, and the diagnostic waits for a terminal to be attached. The corresponding error message is displayed when a terminal is connected to Channel 0 on the Main Processor board.

TWO-LETTER COMMANDS

Two-letter commands perform functions and define the environment for the diagnostic tests. They are introduced below and detailed in Chapter 4, "Two-Letter Commands."

LISTING THE TWO-LETTER COMMANDS

The two-letter commands are used either to perform functions or set the environment for executing diagnostic programs. Two questions marks typed next to the command prompt display the two-letter commands:

1. Type **??**

2. Press **Return.**

   Some of the two-letter commands are displayed below.

   LE Toggles Line Printer Echo

   RE Reboots the system

   CN Displays the current system configuration

   HE Displays the command syntax

EXECUTING THE TWO-LETTER COMMANDS

To execute a two-letter command, you type its two-letter abbreviation next to the command prompt. You can only issue a two-letter command next to the diagnostic prompt (command >). You cannot issue a two-letter command during a diagnostic test. For example, to display the Help screen:

1. Type **HE**

2. Press **Return**.

   The following output is displayed:

```
Commands are as follows:

        [<repeat-count>|]<test-descriptor>[;<test-descriptor>...]
where <test-descriptor> is:
        [<repeat-count>]<command-letter>[L | C]
or        [<repeat-count>]:<test-number>[,subtest-number>][L | C]
or        [two-letter command]

A repeat count of zero indicates the test is repeated until the
operator aborts with the letter q.
The | after the initial repeat count indicates that the whole list of tests
is to be repeated, not just the next test.
A subtest number of zero will performa pre-defined set of subtests.
L and C stand for Long and Continuous modes.
Long mode is similar to 'continue on error' but the errors are saved in
memory and reported at the completion of the test (or group of tests).
Beware that the error log is cleared whenever the parser runsa command
in Long mode.
In Continuous mode, the test will not abort on an error.
Note that multiple commands are separated by ';'.
The following pairs of characters are interchangeable: <:.><?/><|\>
For a list of commands, type:          ?
For information ona specific test, type: ?<command-letter>

command>
```

TESTS AND TEST SEQUENCES

You can also display the test sequences contained in the diagnostic. A single question mark typed next to the command prompt lists the tests and test sequences. Table 3-1 contains some of the test sequences in the diagnostic program.

1. Type ?

2. Press **Return.**

Table 3-1

DIAGNOSTIC TEST SEQUENCES

Test Letter	Test Description
d	System test sequence (default)
a	Map RAM tests
c	Clock tests
h	RS-232-C tests
i	IOP tests
n	Ethernet Combo board tests
l	Line Printer tests
m	Memory tests
p	Page map tests
u	General disk test sequence
w	Disk drive tests
b	MPCC tests

SUBTESTS

You can list the subtests contained in test sequences. The same command also displays the test sequence number. For example, the following command lists the clock subtests and the number of the clock test sequence, some of which are displayed in Table 3-2.

1. Type **?c**

2. Press **Return**.

Table 3-2

CLOCK SUBTESTS

Subtest	Test Description
Clock test (test 4) contains subtests:	
1	Data Path Check to 8259 and 8253
2	Test Timer 0
3	Test Timer 1
4	Test Timer 2
5	Test Clock Calendar
6	Set the Clock Calendar
7	Read date, time from Clock Calendar
8	Special Frequency Calibration Mode

LISTING TEST SEQUENCE SUBTESTS

A special two-letter command lists the test, subtest pairs contained in a selected test sequence. For example, to list the tests that make up system test sequence d:

1. Type **DS**

2. Press **Return.**

   This prompt is displayed:

   Enter Test Sequence command letter:

3. Type **d**

4. Press **Return.**

   The output resembles that contained in Table 3-3.

Table 3-3

CONTENTS OF TEST SEQUENCE D

Test sequence d runs the following test, subtest pairs:

{0,0}
{10,0}
{3,0}
{1,0}
{2,0}
{4,0}
{12,0}
{13,0}
{11,0}
{14,0}
{5,0}
{18,0}

COMMAND STRUCTURE

The basic command structure used to execute the diagnostic tests is:

```
[<repeat-count>|]<test-descriptor>
[;<test-descriptor>...]
```

where <test-descriptor> is:

```
[<repeat-count>]<command-letter>[L|C] or

[<repeat-count>]:<test-number>[<subtest-
   number>][L|C] or

<two-letter command>
```

The parameters contained in brackets ([ ]) are optional. Definitions of repeat-count, command-letter, test-number, and subtest-number are presented below.

Repeat-Count

This optional parameter consists of a positive integer that specifies the number of times the entire test is to be executed. If no repeat count is given in the equation, the test is executed once. If you specify a repeat count, a bar (|) must separate the repeat count from the test descriptor. Do not specify 0 (infinite) as a repeat count.

Command-Letter

This parameter is required when you want to execute an entire test sequence and when you want to refer to that test sequence by its command letter. For example, the letter m identifies the memory test sequence. Several of these sequences (for example, d, s, and #) can only be executed using the single-letter command.

Test-Number

This positive integer is required when you refer to a test by number or when you want to execute subtests in test sequence. When specifying a test number, the number must be preceded by a colon (:) or a period.

Subtest-Number

This parameter is a positive integer (0 to n) that specifies which individual subtest to execute. It is allowed only if the previous parameter is a test number and not a command letter. When both test and subtest numbers are specified, the required delimiter between the integers is a comma (,). If the subtest is not specified, the diagnostic assumes 0.

L|C

This optional parameter specifies the error reporting mode, which can be long (L) or continuous (C).

SAMPLE COMMANDS

Examples of command structures that you can use to execute test sequences, tests, and subtests, are shown below. Since these are examples only, do not execute them.

To execute a test sequence (for example, the memory test sequence), you type its letter next to the command prompt:

Type **m**

Press **Return.**

You can also execute several tests sequentially. For example, to execute the memory tests, the page map (p) tests, and the map RAM (a) tests:

Type **m;p;a**

Press **Return.**

To execute any one, or all three, of the test sequences in continuous error mode:

Type **5|mC;pC;aC**

Press **Return.**

You can also execute each test several times. You type the number of repetitions before the command-letter. For example:

Type **4mC;8pC;2aC**

Press **Return.**

To execute subtest 5 of memory test sequence 0:

Type **:0,5**

Press **Return.**

To execute subtest 5 of memory sequence 10 times in Continuous mode:

Type **10|:0,5C**

Press **Return**.

To execute subtest 5 of memory sequence 0, subtest 3 of the page map test sequence 1 and subtest 4 of the disk drive test sequence 6:

Type **:0,5;:1,3;:6,4**

Press **Return**.

To execute that list of subtests 10 times:

Type **10|:0,5;:1,3;:6,4**

Press **Return**.

To execute that list of subtests 10 times in Continuous mode:

Type **10|:0,5C;:1,3C;:6,4C**

Press **Return**.

NOTE: Long error mode clears the error log before a test or test sequence is executed. Therefore, you cannot execute several test sequences in long mode, like this:

5|mL;pL;aL

Instead, you should use the two-letter command #= to create your own test sequence, which you can then execute in long mode (#L). See Chapter 4, "Two-Letter Commands," to do so.

FINAL DETAILS

Before continuing, here are some important points to remember:

- When you initiate a diagnostic session, the system configuration is displayed on the terminal screen. If the diagnostic does not see equipment that you believe is connected to your system, this can often be the first clue as to why a piece of equipment is not working. It simply may not be properly connected.

  System Configuration (CN) also displays the area in memory occupied by the diagnostic tests.

- The second command you may want to execute is Page Mode (PM).

  PM prevents the contents of the screen from scrolling off the top without permission.

- Control-S (Code-S on PTs and GTs) can be used to stop the output on the screen. Pressing any key will continue the output display.

- In most cases, you can terminate a test sequence or subtest in one of two ways (you do not need the diagnostic prompt):

  Press **q**, or **ESC** (**Go** on a PT or GT)

  The latter command (ESC or Go), provides the most immediate results. When you use either ESC or q to terminate a test, the diagnostic is reset and the system configuration and diagnostic prompt are displayed on the screen.

4 TWO-LETTER COMMANDS

This chapter reviews the language required to list and use the two-letter commands and describes the two-letter commands that perform functions and establish the environment for the diagnostic tests.

NOTE: You can only issue a two-letter command next to the diagnostic prompt (command >). You cannot issue a two-letter command during a diagnostic test.

To display the list of two-letter commands:

1. Type **??**
2. Press **Return.**

To execute a two-letter command, you type its two-letter abbreviation next to the command prompt. For example, to display the system configuration:

1. Type **HE**
2. Press **Return.**

HELP (HE)

This command displays the command syntax and other tips concerning the diagnostic programs.

To display the online help screen:

1. Type **HE**

2. Press **Return.**

Commands are as follows:

```
        [<repeat-count]|]<test-descriptor>[;<test-descriptor>...]
where <test-descriptor> is:
        [<repeat-count>]<command-letter>[L | C]
or       [<repeat-count>]:<test-number>[,subtest-number>][L | C]
or       [two-letter command]
```

A repeat count of zero indicates the test is repeated until the operator aborts with the letter q.
The | after the initial repeat count indicates that the whole list of tests is to be repeated, not just the next test.
A subtest number of zero will perform a pre-defined set of subtests.
L and C stand for Long and Continuous modes.
Long mode is similar to 'continue on error' but the errors are saved in memory and reported at the completion of the test (or group of tests).
Beware that the error log is cleared whenever the parser runs a command
in Long mode.
In Continuous mode, the test will not abort on an error.
Note that multiple commands are separated by ';'.
The following pairs of characters are interchangeable: <:.><?/><|\>
For a list of commands, type: ?
For information on a specific test, type: ?<command-letter>

command>

DISPLAY MODE TWO-LETTER COMMANDS

PAGE MODE (PM)

By default, Page mode is disabled so that output from the diagnostic tests is continuously scrolled through the terminal screen. This continuous scrolling is helpful when you are executing a long series of diagnostic tests. However, in cases where you want to read the output more carefully, such as in reading error messages, continuous scrolling is not helpful.

To enable Page mode:

1. Type **PM**
2. Press **Return**.

   This message is displayed:

   Page mode enabled

When Page mode is enabled, one screenful of information is displayed at a time and the following prompt is displayed along the bottom of each screen. You must press one of the three letters to continue.

<l = next line, e = disable page mode, anything else = next page>

Page Mode is temporarily enabled when, during an attended test, the diagnostic foresees that an error message and subsequent configuration screen could occupy more than one screen. Except for the above situation, Page mode remains enabled until you disable it.

There are two ways to disable Page mode:

- Press **e** when the Page mode message is displayed along the bottom of the screen.
- Type **PM** and press **Return**.

LINE PRINTER ECHO (LE)

Line Printer Echo instructs the diagnostic to echo its output to a line printer. Only the diagnostic output is printed, not your input. The line printer must be connected to the connector marked Parallel Printer on the Main Processor board, not to a printer connector on an expansion board (see Chapter 2 to connect the line printer).

Line Printer Echo is particularly valuable after you execute a test in Long mode. Before you issue the DE command, use LE to echo the error log to the line printer.

To enable line printer (use the same steps to disable Line Printer Echo):

1. Type **LE**
2. Press **Return**.

   This message is displayed:

   ```
   Line Printer Echo Enabled
   ```

   This warning message is displayed when you attempt to enable line printer echo when no line printer is connected to the parallel printer connector:

   ```
   Line Printer Echo Enable
   Warning: printer not present.
   ```

REPEAT COUNT (RC)

Repeat Count lets you specify the number of retries the disk driver executes before it reports a disk drive controller error.

There are two ways to set the number of retries. You can enter a number of your choice or you can cycle through these three predefined settings:

- perform transfer 4 times, recalibrate disk, then execute 5 retries (default setting)
- perform transfer but no retries
- perform transfer, recalibrate disk, and then one retry

The disk driver assumes that what is written to the disk drive is correct, and therefore detects data errors on a read-only basis.

To select the number of retries yourself (for example, 20):

1. Type **RC20** (or RC 20)
2. Press **Return**.

   The following message is displayed. It means that an error is not reported, if one is indeed generated, until 20 transfer operations (either read or write) are attempted.

   Perform transfer (and if necessary) 9 repeats, recal, and 10 retries

Alternately, you can cycle through the three settings listed above:

1. Type **RC**

2. Press **Return.**

   This message is displayed. It instructs the disk driver to read data from the disk drive once before it reports an error. It does not recalibrate the disk and retry the transfer.

   Perform transfer but no retries

3. Type **RC**

4. Press **Return.**

   This message is displayed:

   Perform transfer (and if necessary) recal, and 1 retry

5. Type **RC**

6. Press **Return.**

   The diagnostic is returned to the initial setting to perform the transfer 4 times, recalibrate disk, then execute 5 retries.

ECC ERROR CORRECTION ENABLE/DISABLE (EC)

This two-letter command enables you to change the ability of the ST-506 or SMD disk drive to do ECC correction. This command does not change the format of the disk. The default is disabled, which forces the controller to report the error, creating a more stringent test.

When an soft error occurs and is corrected, it is logged in the disk error log buffer. This buffer contains the number of ECC-corrected errors and the sector location. For ST-506 drives this sector number is the sector location where the error occurred. For Interphase controller SMD drives, this is the starting sector of the current transfer (not necessarily where the error occurred).

To enable ECC correction:

1. Type **EC**

2. Press **Return.**

   The following message is displayed.

   ECC correction is enabled.

DISK SOFT ERROR COUNT (SE x)

This two-letter command enables you to limit the number of soft errors a disk can have before it will stop on error. The default value is 0.

To determine the current number of soft errors allowed:

1. Type **SE?**
2. Press **Return**.

   The following message is displayed.

   x ECC correctable errors allowed.

To set a limit on the number of soft errors:

1. Type **SE x** (where x is a positive integer)
2. Press **Return**.

   The following message is displayed.

   New setting is x.
   x ECC correctable errors allowed.

DISPLAY DISK ERROR STATISTICS (DD)

This two-letter command displays the disk drive error log.

The disk error log is used to keep track of all errors that occur during disk drive tests. The log records the number and sector locations of hard, soft (ECC correctable), and defect errors. This information is displayed at the conclusion of each disk test. DD enables you to look at this information any time.

The disk error log does not automatically clear itself when you run another test on the same drive. Use the CD (Clear Disk Error Statistics) command to clear it. However, if you run a drive test on a different drive, the disk error log will clear itself when the first error is found, and will indicate that with the following message:

> Disk X format has changed since an error was last logged.
> Disk error stats will be cleared.

To display the disk error log:

1. Type **DD**

2. Press **Return**.

   The contents of the disk error log are displayed.

CLEAR DISK ERROR STATISTICS (CD)

This two-letter command clears the disk drive error log (discussed under the DD command).

To clear the disk error log:

1. Type **CD**
2. Press **Return**.

   The disk error log is cleared.

INTERRUPT MODE (IM)

This command toggles disk and tape interrupt mode. During interrupt mode, an operation waits for an interrupt from the 8259 before finishing the operation. During noninterrupt mode, an operation polls the 8259 for an appropriate interrupt before completing the operation. Interrupt mode is enabled when the diagnostic is loaded.

To disable interrupt mode (use the same steps to enable interrupt mode):

1. Type **IM**
2. Press **Return**.

   This message is displayed:

   Disk/QIC Interrupt Mode disabled

UNIT SELECTION TWO-LETTER COMMANDS

DISK DRIVE NUMBER (DN)

This two-letter command enables you to select the hard disk drive or SMD drive from which to execute the disk drive tests. See the installation manual for your system for information on the number and location of the drives in your system. Note that SMD drives are not labeled on the chassis of the X-Box or V-Box.

CAUTION: This command must be used when running test series w or any of the hard disk drive tests (test 6). If you do not, you may destroy the contents of drive D0.

To determine the drive presently selected:

1. Type **DN?**
2. Press **Return**.

There are two ways to select a drive number. The disk drive number is preset for drive D0. You can specify the number of the drive yourself or cycle through all possible drive numbers.

To select a disk drive (for example, the drive number D2):

1. Type **DN2** (or DN 2)

2. Press **Return**.

   This message is displayed:

   Disk Drive 2 is selected.

Alternately, you can issue the two-letter command without specifying a number. In this case, the diagnostic cycles through all possible drive numbers.

1. Type **DN**

2. Press **Return**.

   This message is displayed:

   Disk Drive 1 is selected.

3. Type **DN**

4. Press **Return**.

   This message is displayed:

   Disk Drive 2 is selected.

5. Repeat these steps to select the next disk drive.

TAPE DRIVE NUMBER (TN)

This two-letter command enables you to select the SCSI tape drive (CPU-02 based systems only) or half-inch tape drive from which to execute the tape drive tests.

To determine the tape drive presently selected:

1. Type **TN?**

2. Press **Return.**

Each controller can support up to eight tape drives. The controller on the CPU board is designated as 0. The controller on the first VME card installed on the VME bus (the first one indicated in the EEPROM) is designated as 1, and so on. To determine the tape drive number, use the following formula:

```
TN = (controller# * 8) + unit#
```

To select a tape drive (for example, the tape drive 2):

1. Type **TN2** (or TN 2)

2. Press **Return.**

   This message is displayed:

   ```
   Tape Drive 2 is selected.
   ```

Alternatively, you can issue the two-letter command without specifying a number. In this case, the diagnostic cycles through all possible tape drive numbers.

1. Type **TN**

2. Press **Return**.

   This message is displayed:

   Tape Drive 1 is selected.

3. Type **TN**

4. Press **Return**.

   This message is displayed:

   Tape Drive 2 is selected.

5. Repeat the previous steps to select the next disk drive.

ETHERNET COMBO BOARD NUMBER (EN)

This two-letter command enables you to select the Ethernet Combo board on which you want to run tests.

To select an Ethernet Combo board (for example, board number 2):

1. Type **EN2** (or EN 2)
2. Press **Return**.

   This message is displayed:

   ```
   Ethernet board 2 is selected.
   ```

Note that you cannot use EN? to determine which board is currently selected.

VME COMMUNICATION CONTROLLER CARD NUMBER (MN)

This two-letter command enables you to select the VME Communications Controller card (also called the MPCC card) on which you want to run tests.

To select a VME Communication Controller card (for example, card number 2):

1. Type **MN2** (or MN 2)
2. Press **Return**.

   This message is displayed:

   ```
   MPCC board 2 is selected.
   ```

Note that you cannot use MN? to determine which card is currently selected.

REBOOT (RE)

The **Reboot** command provides a convenient way to reboot your system. Recall that all of the two-letter commands are returned to their preset states when you reboot the diagnostics.

To reboot the diagnostic in this manner:

1. Type **RE**

2. Press **Return**.

   The system and the diagnostic program are rebooted.

You can use the dash (-) option when rebooting the system to skip the initial memory tests run during system boot. To do so,

1. Type **RE-**

2. Press **Return**.

   The system and the diagnostic program are rebooted without running the initial memory tests.

NMI/PARITY ENABLE (PE)

This two-letter command toggles that part of the General Command Register (GCR) that enables the reporting of nonmaskable interrupts (NMIs) and parity errors.

To detect errors as soon as possible, all three interrupts are enabled when the diagnostic is loaded. Interrupts can be disabled when you believe they are generated by a more serious problem.

Interrupts are automatically disabled during the Physical Memory Map tests, and reenabled when those tests are completed. The interrupts must be disabled before executing the desired test or test sequence in question.

To disable NMIs and parity errors:

1. Type **PE**
2. Press **Return**.

   This message is displayed on the screen:

   System interrupts disabled.

Interrupts are automatically enabled when you toggle PE again, when you reset the diagnostic, or by certain tests.

CACHE ENABLE (CA)

The MC68020 contains an on-chip, instruction-only cache memory that improves the performance of the MC68020. Its purpose is to increase execution efficiency by providing a quick-store for instructions.

Initially, the diagnostic enables this cache. This two-letter command lets you access this cache control register and disable the cache. When you disable this cache, you force continuous cache misses and suppress cache fills. This command will not disable any other caches on the CPU board.

To disable the cache:

1. Type **CA**

2. Press **Return**.

   This message is displayed:

   ```
   mc68020 cache disabled
   ```

The processor is now forced to access external memory on every instruction fetch. The cache remains disabled until you either reboot the diagnostic tests or issue the two-letter command again.

SYSTEM PRIORITY LEVEL (SP x)

The System Priority Level is specified by the Processor Status Register in the MC68020. A given device can interrupt the diagnostic at a specific level. The test sets the system priority level to allow that type of interrupt. The system priority levels for the diagnostic tests are listed in Table 4-1. You can only set the interrupt levels to integers 0 through 7.

To change the interrupt level to 2:

1. Type **SP2** or **SP 2**

2. Press **Return**.

The system priority level remains at the respective setting until you change the setting or reboot the diagnostic. When you reboot the diagnostic, the system priority level is returned to the default level of 7.

While the command prompt is displayed on the terminal screen, the diagnostic temporarily lowers the system priority level to 0. The previous system priority level is restored when the diagnostic reads the input line. This allows interrupts to be processed immediately, while the diagnostic waits at the command prompt.

The system priority level is important for the execution of several tests. Tests for which the system priority level is critical set the appropriate level during that test.

To display the system priority level:

1. Type **SP?** (or SP ?)

2. Press **Return**.

Table 4-1

SYSTEM PRIORITY LEVELS (SPL)

Subsystem	SPL
Line printer VME level 1	1
VME level 2	2
QIC Tape Drive SCSI bus RS-422 Expansion board Clocks 0 and 2 VME level 3	3
VME level 4	4
RS-232-C Expansion boards IOP Expansion board VME level 5	5
Clock 1 VME level 6	6
Main system NMI	7

SYSTEM CONFIGURATION (CN)

This two-letter command displays the current system configuration. Hardware not reported by CN may be defective or improperly installed. The information displayed here is the same information that appears on the diagnostic screen when you first load the diagnostics.

To display the system configuration:

1. Type **CN**

2. Press **Return**.

   The system configuration is displayed on the terminal screen. It resembles the following output:

```
System configuration:
     5 mega bytes of memory
     4 Mb memory board in slot M1
     MC68881 coprocessor is present
     VME expansion board is not present.
     RS422 board is present
     Drive 0 is present (52HIT)
     Drive 1 is present (85HIT)
     Parity interrupts enabled
Memory tests will begin at 0x64000, end at 0x700000
```

The system configuration also identifies the disk drives that were present but that contain invalid Volume Home Blocks.

DISPLAY ERRORS (DE)

Display Errors displays the current error log for tests executed in Long mode. Before you execute this command, you can enable the line printer (if you connected one) to generate hard copy of the error log.

To display the error log:

1. Type **DE**

2. Press **Return**.

The error log is displayed on the terminal screen and, if line printer echo is enabled, echoed to the line printer. Unless already enabled, Page mode is temporarily enabled to prevent the error messages from scrolling off the screen. Page mode returns to whatever it was previously set to at the end of the messages.

DISPLAY TEST SEQUENCE SUBTESTS (DS)

A special two-letter command exists to list the pairs of tests and subtests executed as part of selected test sequences.

For example, to list the tests that make up system test sequence d:

1. Type **DS**

2. Press **Return**.

   This prompt is displayed:

   Enter Test Sequence command letter:

3. Type **d**

4. Press **Return.**

   The subtests executed by test sequence d are displayed on the terminal screen.

USER SELECTION TWO-LETTER COMMANDS

CHANGE CONSOLE (CO)

This two-letter command enables you to change the console from which the diagnostic commands are entered. It is particularly valuable when you need to execute the diagnostic tests on an RS-232-C channel other than Channel 0, or to change the baud rate of Channel 0. CO can be used to allow a remote terminal (possibly connected to a modem) to issue diagnostic commands to the system.

To change consoles:

1. Type **CO**

2. Press **Return.**

   These prompts are displayed:

```
Enter channel number for new console:
Echo on tty000?
    0) 300 Baud
    1)1200 Baud
    2)2400 Baud
    3)9600 Baud
Enter baud rate: 0-3:
```

After you respond to the prompts, which are described below, this confirmation is displayed:

```
Switching console to tty020 at 1200 Baud ... echo on tty000.
Welcome!
```

The first prompt requires you to enter the number of the new RS-232-C channel (RS-232-C channels 1 through 41). The second prompt enables you to echo commands from the remote console to the console connected to Channel 0. The third prompt enables you to specify the baud rate. The console channel reverts to Channel 0 when the diagnostic is rebooted.

USER-DEFINED TEST SEQUENCE (#=)

In addition to the numerous predefined test sequences provided by the diagnostic, you can create your own sequence of tests. You may want to execute several memory tests, some nondestructive hard disk drive tests, and some RS-422 Expansion board tests. This command is particularly valuable when you want to execute specific tests or test sequences that are either not provided by the diagnostic or that you want to execute in long error reporting mode.

Creating the User-Defined Test Sequence

To create your own test sequence:

1. Type #=

2. Press **Return**.

   This prompt is displayed:

   ```
   User-defined test sequence definition
   enter test,subtest (return to finish):
   ```

3. Type the **test,subtest** pairs. If just the test number is indicated, subtest 0 is assumed.

4. Press **Return**.

   This prompt is displayed:

   ```
   Enter a one line description of the sequence:
   ```

5. Enter text to describe the sequence.

Executing the User-Defined Test Sequence

The pound sign (#) now becomes a test sequence and can be treated like any other test sequence.

To execute the new test sequence:

1. Type #

2. Press **Return**.

APPENDIX A: CABLE SPECIFICATIONS

This appendix defines pin assignments for the cables, terminators, and loopback plugs required to execute S/MT Series diagnostic programs. Specifically, it includes pin assignments for these items:

- 25-pin to 25-pin RS-232-C crossed cable
- 9-pin to 25-pin RS-232-C crossed cable
- parallel line printer cable
- bidirectional parallel printer port cable
- 9-pin to 9-pin RS-422 cable
- 25-pin RS-232-C loopback plug
- 9-pin RS-232-C loopback plug
- UPS loopback plug
- line printer loopback plug

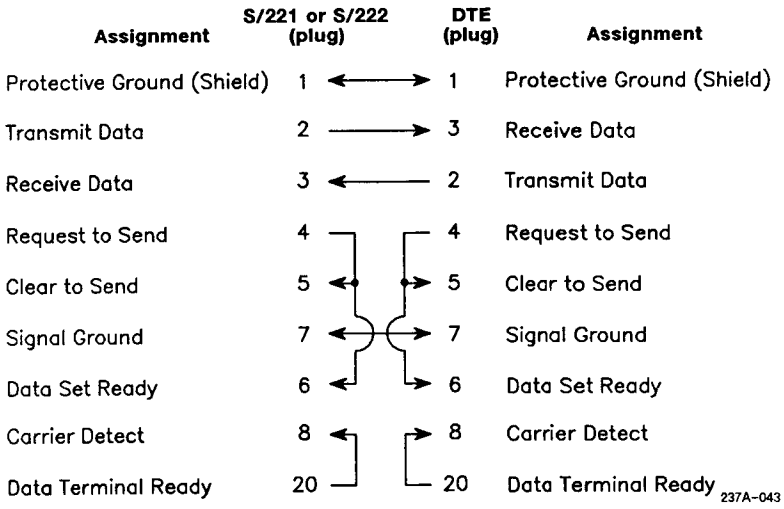
NOTE: To ensure compliance with FCC and VDE emissions regulations, peripherals connected to S/MT Series systems must use Convergent (or equivalent) recommended terminators and cables. The cables incorporate special bulk cable and shielding terminations (that is, 360^{o} shielding from connector shell to connector shell with low en4ergy leakage). Off-the-shelf cables and terminators may not provide the shielding required to meet FCC and VDE emissions requirements. It is the user's responsibility to do whatever testing is required to assure overall system compliance.

RS-232-C 25-PIN TO 25-PIN CROSSED CABLE

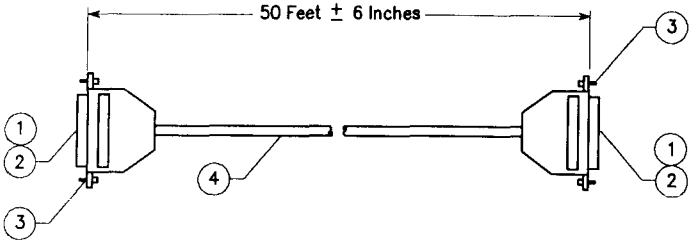
Assembly: Crossed 25-pin to 25-pin cable for RS-232-C communications

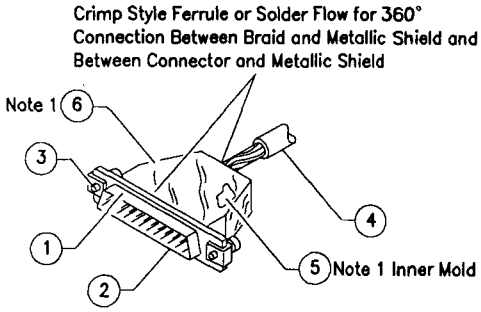
Function: Connects DTE to 25-pin connector on the CPU board

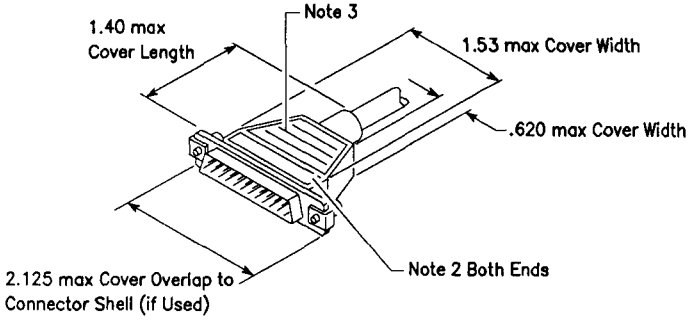
Pin Assignments:



Crossed Cable Construction:







237A-044

Item	Quantity	Description
1	2	25-pin plug assembly (male). Use AMP part number 20764-2 or equivalent.
2	16 or 18	Connector contacts. Use AMP part number 66507-3 or equivalent.
3	2	Connector shell, strain reliefs, and grounding. Use AMP part number 745173-3 or equivalent.
4	25 feet	3-conductor shielded cable. Use Alpha part number 5114 or equivalent.
5	2	Inner crimp ferrule. Use AMP part number 2-745129-1.
6	2	Outer crimp ferrule. Use AMP part number 745130-8.

Notes:

1. Data terminal devices that have metal connectors use Convergent RS-232-C cable, part number 61-00302, or equivalent.

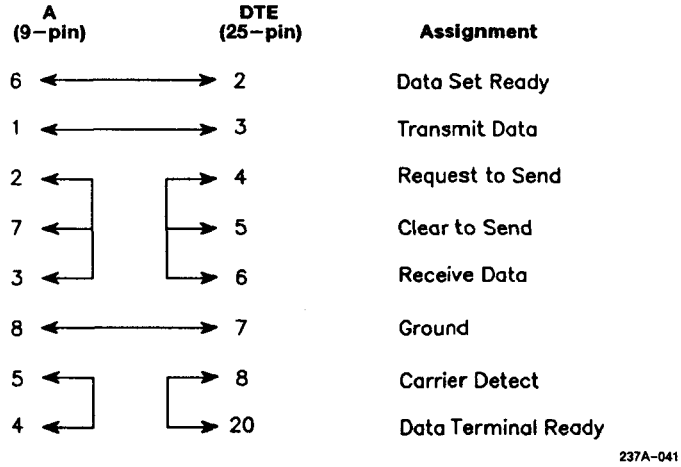
2. If the DTE has a plastic shell connector, pin 1 must be connected at both ends for proper grounding.

RS-232-C 9-PIN TO 25-PIN CROSSED CABLE

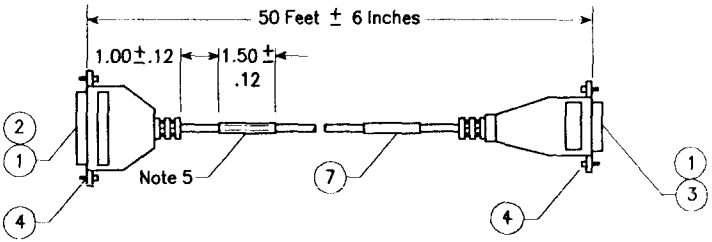
Assembly: Crossed 9-pin to 25-pin cable for RS-232-C communications

Function: Connects DTE to an RS-232-C Expansion board

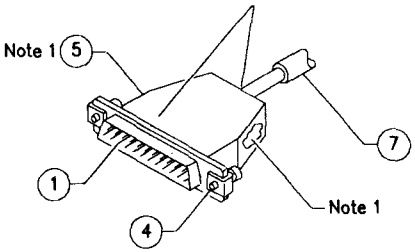
Pin Assignments:

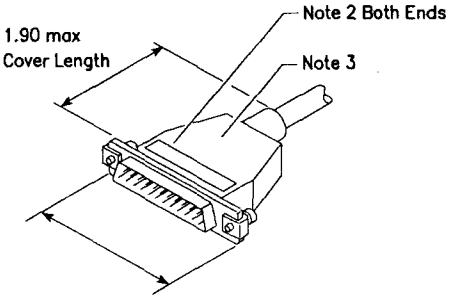


Crossed Cable Construction:



Crimp Style Ferrule or Solder Flow for 360°
Connection Between Braid and Metallic Shield and
Between Connector and Metallic Shield





1.250 max (9pos)/2.125 max (25 pos)
Cover Overlap to Connector Shell (if Used)

237A-042

Item	Quantity	Description
1	16	Contact, crimp pin
2	1	25-position, D-type socket assembly
3	1	9-position, D-type plug assembly
4	4	Screw and retainer (male)
5	AR	Copper foil tape
6	2 feet	Black copper wire, 300V
7	3 feet	3-conductor shielded cable

Notes:

1. Primary conductors must be fully encompassed and strain-relieved by inner mold. Copper foil shield must cover inner mold.

2. Permanently mark part number, dash number, and revision level approximately where shown.

3. Molded cover: material, polyvinyl chloride or equivalent rated UL 94HB or better. Appearance of mold may vary within the noted dimensions. Cable jacket must be fully encompassed and strain relieved by molded cover.

PARALLEL LINE PRINTER CABLE

Assembly: Parallel printer cable

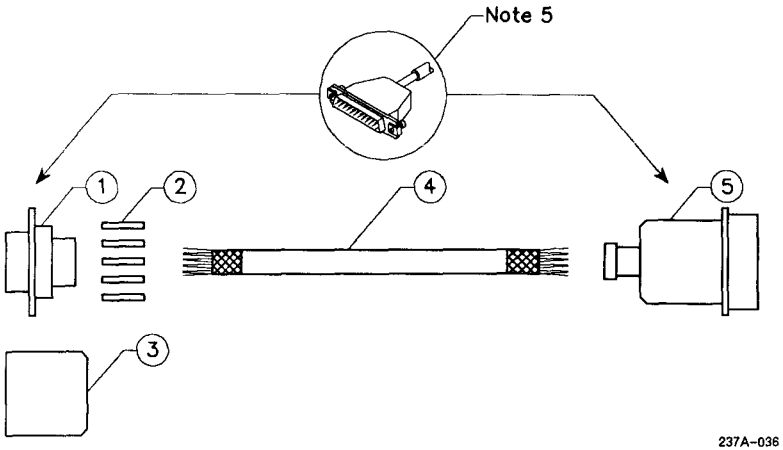
Function: Connects parallel printers to MightyFrame Series systems

Pin Assignments:

A S/221 orS/222 (plug)		DTE Printer (plug)	Assignment
1	→	2	LPT0+
2	→	3	PT1+
3	→	4	LPT2+
4	→	5	LPT3+
5	→	6	LPT4+
6	→	7	LPT5+
7	→	8	LPT6+
8	→	9	LPT7+
9 to 11	↔	21 to 23	GND (spare conductors)
12, 13, 15	↔	24, 25, 26	GND (spare conductors)
14	→	1	LPTSTROBE–
16	←	10	LPTACK–
17	←	11	LPYBUSY+
21	←	12	LPTNOPAPER+
22	←	13	LPTSELECT+
23, 24	↔	19, 20	GND (spare conductors)
25	↔	17	Chassis ground (shield)

237A-035

Parallel Printer Cable Construction:



Item	Quantity	Description
1	1	25-pin D-type plug assembly (male). Use AMP part number 207464-2 or equivalent.
2	25	Connector contacts. Use AMP part number 66507-3 or equivalent.
3	1	Connector shell and strain relief. Use AMP part number 745173-3 or equivalent.
4	10 feet	30-conductor shielded cable. Use Alpha part number 5279/15 or equivalent.

Item	Quantity	Description
5	1	36-pin Blue Ribbon type connector assembly (male). Use AMP part number 57-30360 or equivalent.
6	AR	Copper foil (part number 80-00025).

Notes:

1. The chassis ground must be connected to the shield drain wire at both ends.

2. A twisted pair consists of a signal and a ground.

3. All four unused connectors must be connected to ground at both ends.

4. The vinyl insulation must be stripped back at the 36-pin printer connector so that the metal strain relief clamps down on the conducting shield.

5. Wrap foil around wires. Solder copper foil to the braid and metal connector shell around the entire periphery.

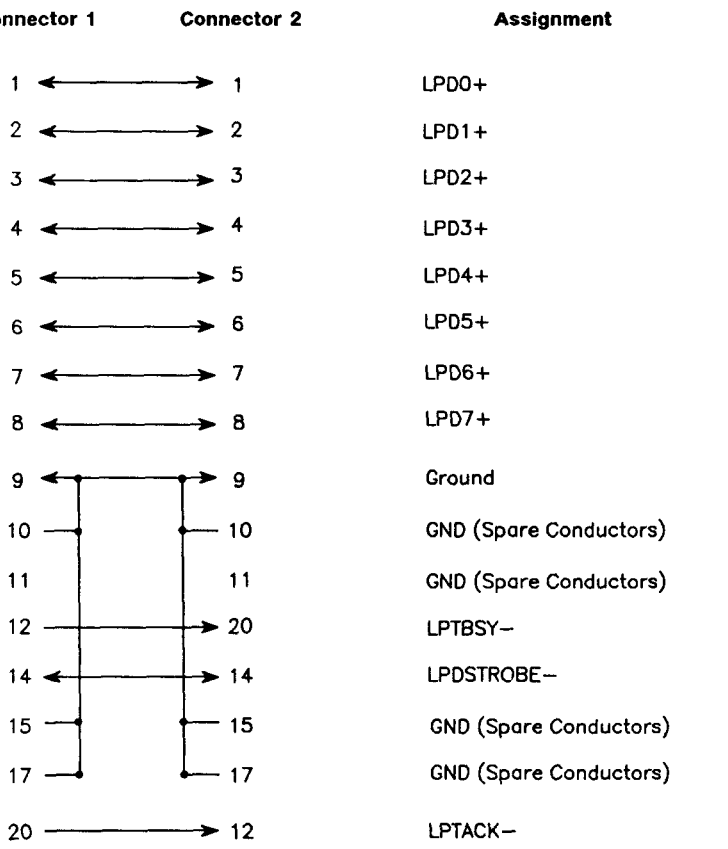
6. The cable must be 10 feet (3 meters) maximum. Longer cables must be approved by the printer manufacturer.

BI-DIRECTIONAL PARALLEL PRINTER PORT CABLE

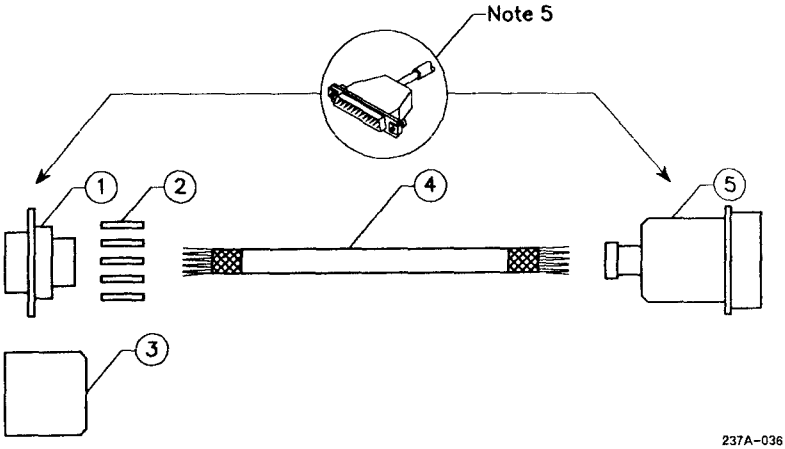
Assembly: Bidirectional parallel printer port cable

Function: Used with bidirectional parallel printer port diagnostic tests in CPU-02 based systems

Pin Assignments:



Bidirectional Parallel Printer Port Cable Construction:



Item	Quantity	Description
1	1	25-pin D-type plug assembly (male). Use AMP part number 207464-2 or equivalent.
2	25	Connector contacts. Use AMP part number 66507-3 or equivalent.
3	1	Connector shell and strain relief. Use AMP part number 745173-3 or equivalent.
4	10 feet	30-conductor shielded cable. Use Alpha part number 5279/15 or equivalent.

Item	Quantity	Description
5	1	36-pin Blue Ribbon type connector assembly (male). Use AMP part number 57-30360 or equivalent.
6	AR	Copper foil (part number 80-00025).

Notes:

1. The chassis ground must be connected to the shield drain wire at both ends.

2. A twisted pair consists of a signal and a ground.

3. All four unused connectors must be connected to ground at both ends.

4. The vinyl insulation must be stripped back at the 36-pin printer connector so that the metal strain relief clamps down on the conducting shield.

5. Wrap foil around wires. Solder copper foil to the braid and metal connector shell around the entire periphery.

6. The cable must be 5 feet (1.5 meters) maximum. Longer cables must be approved by the printer manufacturer.

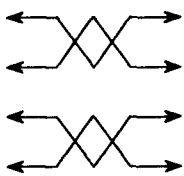
RS-422 CLUSTER COMMUNICATIONS CABLE

Assembly: RS-422 cluster communications

Function: Connects RS-422 terminals to the RS-422 Expansion board

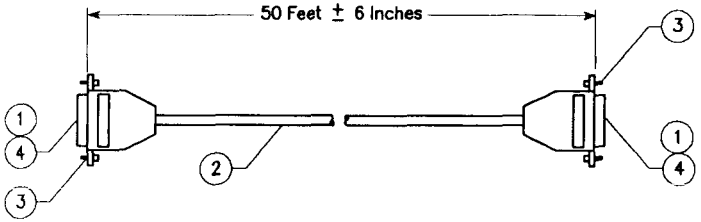
Pin Assignments:

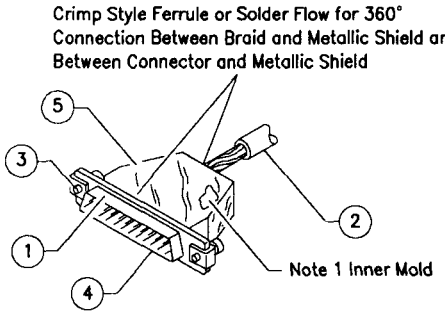
Connector A (plug)	Connector B (plug)	Assignment
1	1	
2	2	
3	3	
4	4	
5	5	
6	6	CLK+
7	7	CLK–
8	8	DATA+
9	9	DATA–

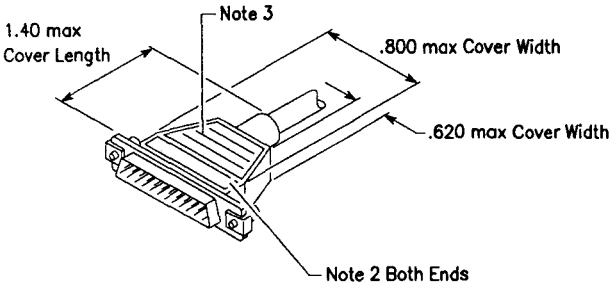


237A-045

Cluster Cable Construction:







237A-046

Item	Quantity	Description
1	2	9-pin D-type plug assembly (male). Use AMP part number 205204-4 or equivalent.
2	25 to 1,200 feet	4-conductor, shielded, twisted-pair cable. Use Belden part number 9829 or equivalent.
3	4	Screw and retainer, male (part number 58-00061)
4	8	Connector contacts. Use AMP part number 66507-3 or equivalent.
5	AR	Copper foil (part number 80-00025).

Notes:

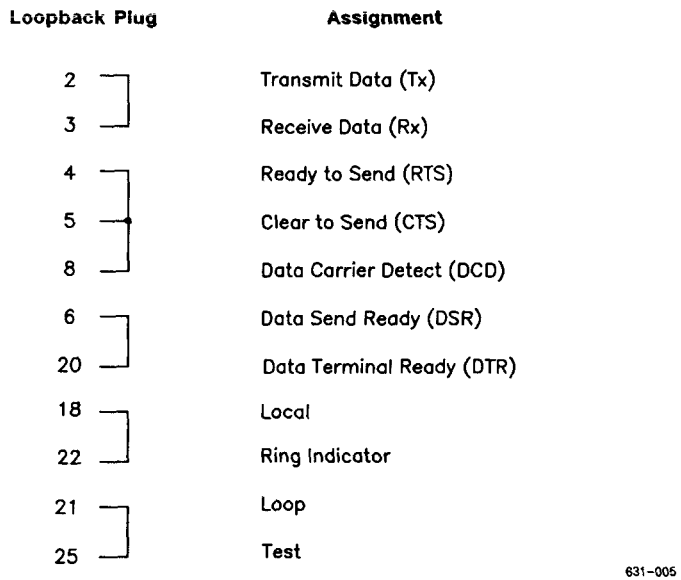
1. Primary conductors must be fully encompassed and strain-relieved by inner mold. Copper foil shield must cover the inner mold.
2. Permanently mark part number, dash number, and revision level where shown.
3. Molded cover: color, grey; material, polyurethane or equivalent rated UL 94HB or better. Cable jacket must be fully encompassed and strain-relieved by molded cover.
4. Pins 6 and 7, and pins 8 and 9, are each twisted pairs.
5. The maximum total length of a cluster communications line is 1,200 feet (365.85 meters). The minimum cable length between terminals is 25 feet (7.62 meters).

RS-232-C 25-PIN LOOPBACK PLUG

Assembly: RS-232-C 25-pin loopback plug

Function: Used with RS-232-C and VME MPCC card diagnostic tests

Pin Assignments:



Construction:

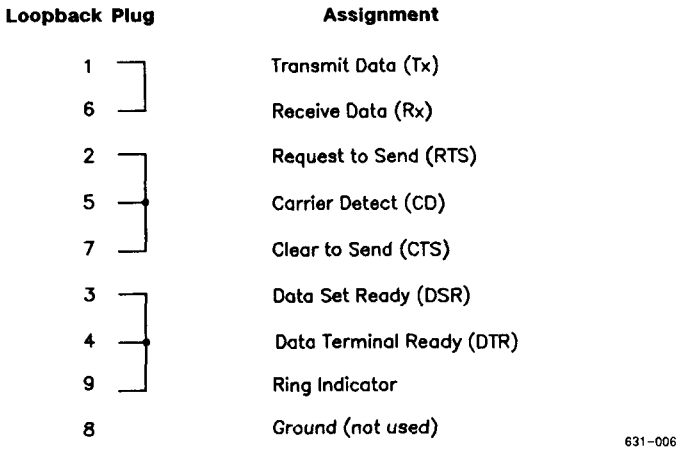
Quantity	Description
11	Contact, crimp pin
1	25-pin D-type connector plug
2	screw and retainer (male)
1 foot	Copper foil tape
1.5 feet	Black copper wire, 26 AWG, 300 Volt

RS-232-C 9-PIN LOOPBACK PLUG

Assembly: RS-232-C 9-pin loopback plug

Function: Used with RS-232-C diagnostic tests

Pin Assignments:



Construction:

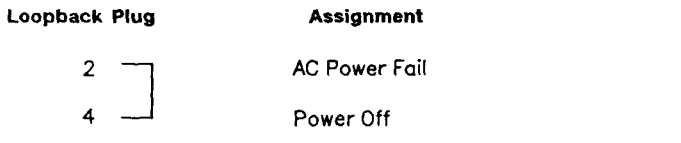
Quantity	Description
8	Contact, crimp pin
1	9-pin D-type connector plug
2	screw and retainer (male)
1 foot	Copper foil tape
2 feet	Black copper wire, 26 AWG, 300 Volt

UNINTERRUPTABLE POWER SUPPLY (UPS) LOOPBACK PLUG

Assembly: UPS loopback plug

Function: Used with UPS diagnostic tests

Pin Assignments:



Construction:

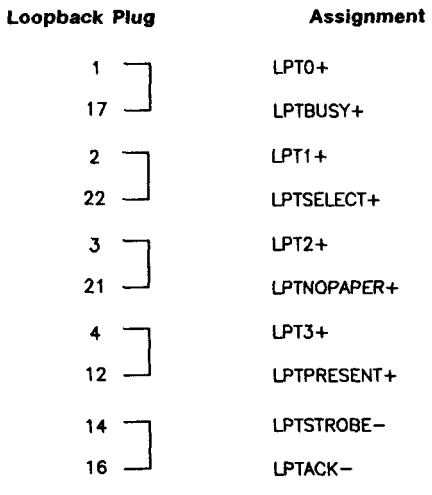
Quantity	Description
2	Contact, crimp pin
1	9-pin D-type connector plug (female)
2	screw and retainer (male)
1 foot	Copper foil tape
0.2 feet	Black copper wire, 26 AWG, 300 Volt

LINE PRINTER LOOPBACK PLUG

Assembly: Line printer loopback plug

Function: Used with line printer diagnostic tests

Pin Assignments:



Construction:

Quantity	Description
10	Contact, crimp pin
1	25-pin D-type connector plug (female)
2	screw and retainer (male)
1 foot	Copper foil tape
1.5 feet	Black copper wire, 26 AWG, 300 Volt

CPU-01 Diagnostic Tests

CPU-01 DIAGNOSTIC TESTS

This chapter describes the CPU-01 Subsystem diagnostic programs, most of which are listed in Table 1.

The CPU-01 (shown in Figure 1) is installed is several systems (including the S/120, S/220, S/221, S/222, and S/320) and can be distinguished from the CPU-02 board by the location of the uninterruptable power supply (UPS) connector.

Table 1

CPU-01 DIAGNOSTIC TESTS

Number	Letter	Description
0	m	Memory
1	p	Page Map
2	o	Physical Memory
3	r	Parity
4	c	Clock
10	a	Map RAM
12	e	Page Protection
13	l	Line Printer

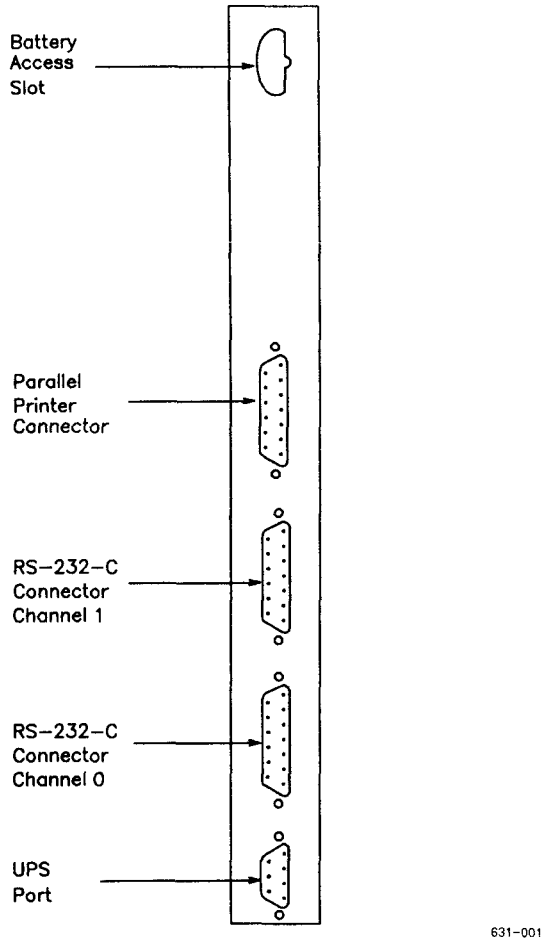


Figure 1. CPU-01 Board

TEST 0: MEMORY TEST (m)

The memory test writes, then reads, data to every location in memory. Error messages are displayed when the diagnostic reads something other than what it wrote. Table 2 contains a summary of memory subtests.

The basic memory test (m) consists of subtests 1 through 13. Subtests 2 through 7 are 32-bit versions while subtests 8 through 13 are 16-bit versions of subtests 2 through 7. Subtests 15 and 20 through 32 require your input and are therefore not included in the test sequence.

Memory is tested in halves. Test sequence m executes subtests 1 through 13 on the top half of memory (the region not occupied by the diagnostics).

Test sequence m then executes subtest 17, which swaps the contents in the top and bottom halves of memory. Test sequence m then executes subtests 1 through 13 again to test the memory not tested during the first pass. Memory is swapped back when one of these events occurs:

- subtest 17 is executed again (in test sequence m or by itself)
- you attempt to execute a test or subtest not contained in the memory test sequence

When memory is swapped, you can only execute another memory subtest. You cannot execute any other test. If you attempt to do so, the diagnostic swaps the memory back before executing the test or subtest.

Table 2

MEMORY SUBTESTS

Subtest	Description
1	Logic control to each 1M byte bank of memory
2	Write/read 0x00000000's (32-bit)
3	Write/read 0xffffffff's (32-bit)
4	Write/read 0xaaaaaaaa's (32-bit)
5	Write/read 0x54545454's (32-bit)
6	Write/read addresses (32-bit)
7	Write/read address complements (32-bit)
8	Write/read 0x0000's (16-bit)
9	Write/read 0xffff's (16-bit)
10	Write/read 0xaaaa's (16-bit)
11	Write/read 0x5454's (16-bit)
12	Write/read addresses (16-bit)
13	Write/read address complements (16-bit)
15	Set memory limits for above subtests
17	Swap top 1/2 of memory with bottom 1/2

Interactive tests:	8-bit	16-bit	32-bit
Write/read address	20	24	28
Read address	21	25	29
Write/read block	22	26	30
Copy block	23	27	31
Compare blocks			32

NOTE: Parity errors are enabled for the memory test sequence. When a parity error occurs, depending on whether it was encountered during the first or second read pass, one of the following two error messages is reported, respectively (where **x** is the byte and **#** is the page number where the parity error occurred):

Address parity error in byte(s)x of physical page #

Refresh parity error in byte(s)x of physical page #

NONINTERACTIVE MEMORY SUBTESTS (SUBTESTS 1-13)

Subtest 1 performs logic control to each 1M byte bank of memory. Subtests 2 through 13 write predefined data patterns to every location in memory. They each consist of two passes.

- write, then immediate read-back pass, to detect pure memory errors
- read-only pass to detect address overlapping and refresh errors

Subtests 2 and 3 delay approximately 15 seconds between passes to detect refresh errors. To prevent memory access, the delay is executed out of the 68020 instruction cache, which is enabled. Subtests 2 through 13 are described in Table 3.

TABLE 3

MEMORY SUBTESTS 2 THROUGH 13

Subtest	Purpose
2,8	Write and read 0's to every location in memory in 32-bit and 16-bit modes, respectively.
3,9	Write and read f's in hexadecimal format to every location in memory in 32-bit and 16-bit modes, respectively.
4,10	Write and read a's in hexadecimal format to every location in memory in 32-bit and 16-bit modes, respectively.
5,11	Write and read 54's in hexadecimal format to every location in memory in 32-bit and 16-bit modes, respectively. (0x54 was chosen because it is almost a complement of 0xaa and because it forces the parity bit in a different direction than the 0xaa pattern.)
6,12	Write and read the addresses of each location in memory in 32-bit and 16-bit modes, respectively.
7,13	Write and read the one's complements of addresses to every location in memory in 32-bit and 16-bit modes, respectively.

To execute subtests 1 through 13:

1. Type **m**

2. Press **Return.**

   The following output is displayed when the tests are successfully executed:

```
**Testing physical memory from 0x7B000 to 0xA00000**
 MEMORY TEST
 MEMORY SUBTEST 1
 MEMORY SUBTEST 2
 MEMORY SUBTEST 3
 MEMORY SUBTEST 4
 MEMORY SUBTEST 5
 MEMORY SUBTEST 6
 MEMORY SUBTEST 7
 MEMORY SUBTEST 8
 MEMORY SUBTEST 9
 MEMORY SUBTEST 10
 MEMORY SUBTEST 11
 MEMORY SUBTEST 12
 MEMORY SUBTEST 13
 MEMORY SUBTEST 17
 MEMORY SUBTEST 1
 MEMORY SUBTEST 2
 MEMORY SUBTEST 3
 MEMORY SUBTEST 4
         .
         .
         .
 MEMORY SUBTEST 13
```

To execute one memory subtest, for example subtest 3:

1. Type **:0,3**

2. Press **Return.**

   The following output is generated when the test is successfully executed:

```
MEMORY TEST
MEMORY SUBTEST 3
```

INTERACTIVE MEMORY SUBTESTS (15, 17, 20-32)

CAUTION: Subtests 15, 17, and 20 through 32 should only be run by qualified technicians who thoroughly understand S/MT Series virtual memory concepts.

Subtests 15, 17, and 20 through 32 provide more control and allow more specific testing than subtests 1 through 13.

- subtest 15 allows you to set the memory limits for subtests 2 through 13.
- subtest 17 swaps the contents in the top and bottom halves of memory.
- subtests 20 through 32 perform limited versions of subtests 2 through 13

Subtests 20 through 32 allow you to define the memory limits and, in some cases, the number of repetitions, data patterns, and destination addresses required by the diagnostic to execute subtests 2 through 13. The purposes of subtests 20 through 32 are displayed in Table 4.

Required Input

Table 5 below summarizes the input you must provide to execute subtests 15 and 20 through 32.

CAUTION: When you respond to the prompts displayed by these subtests, be sure not to specify the area in memory where the diagnostic tests reside. The command to display system configuration (CN) displays the area in memory where the diagnostic tests begin.

Table 4

MEMORY SUBTESTS 15 THROUGH 27

Test Purpose	8-bit	16-bit	32-bit
Write and read address	20	24	28
Read address	21	25	29
Write and read block	22	26	30
Copy block	23	27	31
Compare blocks			32

Table 5

INPUT FOR SUBTESTS 15 THROUGH 27

Type of Input	Subtests
Starting address	15, 20 through 32
Ending address	15, 22, 23, 26, 27, 30, 31
Data pattern	20, 22, 24, 26, 28, 30
Destination address	23, 27, 31
Number of loops	15, 20 through 32

Hex Starting Address. The location in memory where you want the subtest to begin testing. The prompt looks like this:

```
Enter hex starting address                    =>
```

Hex Ending Address. The location in memory where you want the subtest to stop reading, writing, or copying to and from. The prompt looks like this:

```
Enter hex ending address                      =>
```

Hex Data Pattern. The pattern of information that you want the subtest to either write or read (or both) in memory. The prompt looks like this:

```
Enter hex data pattern                    =>
```

Hex Destination Address. The address where you want the subtest to write the data pattern defined above. The prompt looks like this:

```
Enter hex destination address             =>
```

Decimal Number of Loops. The number of times you want to execute the subtest, according to the parameters established above. The prompt looks like this:

```
Decimal number of loops                   =>
```

After you enter the decimal number of loops, the diagnostic displays a confirmation of what you just entered. The confirmation differs according to each test. The prompt provides a convenient way for you to remain in the Interactive Memory Subtest menu. It provides two options.

- execute another interactive subtest
- exit the Interactive Subtest menu

To execute another subtest, type the subtest number next to the prompt. Otherwise, to exit the menu:

Press **Return**

A table presenting the interactive subtests is displayed. To return to the diagnostic prompt:

Type **99** (or press **Return**)

Subtest 15

This subtest lets you set the memory limits for subtests 2 through 13.

To execute subtest 15:

1. Type **:0,15**
2. Press **Return**.

   The following output is generated when the test is successfully executed:

   ```
   MEMORY TEST
   Testing physical memory from 0x87F000 to 0xC00000
   MEMORY SUBTEST 14
   Enter hex starting address                    =>
   ```

3. Type the starting hexadecimal address.
4. Press **Return**.

   This prompt is displayed:

   ```
   Enter hex ending address                      =>
   ```

5. Type the ending hexadecimal address.
6. Press **Return**.

   This prompt is displayed:

   ```
   Enter decimal number of loops                 =>
   ```

7. Enter the number of times you want to execute subtests 2 through 13.

   The number of times that subtest 14 executes subtests 1 through 13 depends on the number of loops you select.

8. Press **Return**.

Subtests 2 through 13 are executed. The following output is displayed as the tests are executed.

```
MEMORY TEST
MEMORY SUBTEST 1
MEMORY SUBTEST 2
MEMORY SUBTEST 3
MEMORY SUBTEST 4
MEMORY SUBTEST 5
MEMORY SUBTEST 6
MEMORY SUBTEST 7
MEMORY SUBTEST 8
MEMORY SUBTEST 9
MEMORY SUBTEST 10
MEMORY SUBTEST 11
MEMORY SUBTEST 12
MEMORY SUBTEST 13
```

Subtest 17

This subtest is included in test sequence m and swaps the contents of the top and bottom halves of memory. When memory is swapped, the region in physical memory formerly occupied by the diagnostic can be tested. Therefore, you can test all regions in memory.

To execute subtest 17 separately (it is automatically executed by test sequence m):

1. Type **:0,17**

2. Press **Return.**

   This prompt is displayed to indicate that the contents in the top and bottom halves were swapped. It also displays the region in memory that the memory subtests can test (0x0 to 0x880000 in this case).

   ```
   MEMORY TEST
   **Test physical memory from 0x0 to 0x880000**
   MEMORY SUBTEST 17
   ```

CAUTION: If you run an interactive memory subtest immediately after running subtest 17, be aware that the interactive memory subtest will prompt you for virtual addresses. Since the swap in subtest 17 is a virtual swap, the address you provide may not be the physical address you expect it to be.

Call your field service technician or customer service for assistance in testing lower physical memory.

Subtests 20, 24, and 28

These subtests write and read data to a location in memory in 8-bit, 16-bit, and 32-bit modes, respectively. To execute one of these three subtests:

1. Type **:0,subtest number**

2. Press **Return**.

   These prompts are displayed. Enter data next to each prompt.

   ```
   MEMORY TEST
   **Testing physical memory from 0x87F000 to
   0xC00000**
   MEMORY SUBTEST 20, 24, or 28
   Write/read an address in ## bit mode
   Enter hex starting address                =>
   Enter hex data pattern                    =>
   Enter decimal number of loops             =>
   ```

3. Press **Return**.

   This output is generated when the tests are successfully executed:

   ```
   Starting address          = 0x000000
   Data Pattern              = 0x0
   Number of loops           = 0
   ```

   The subtest is completed when this prompt is displayed:

   ```
   Enter memory subtest number               =>
   ```

Subtests 21, 25, and 29

These subtests read data from a location in memory in 8-bit, 16-bit, and 32-bit modes, respectively. To execute one of these three subtests:

1. Type **:0,subtest number**

2. Press **Return**.

   These prompts are displayed. Enter data next to each prompt.

   ```
   MEMORY TEST
   **Testing physical memory from 0x87F000 to
   0xC00000**
   MEMORY SUBTEST 21, 25, or 29
   Read from an address in ## bit mode
   Enter hex starting address                =>
   Enter decimal number of loops             =>
   ```

3. Press **Return**.

   This output is generated when the tests are successfully executed:

   ```
   Starting address          = 0x000000
   Number of loops           = 0
   Address 0x000000          = 0x00
   ```

   The subtest is completed when this prompt is displayed:

   ```
   Enter memory subtest number               =>
   ```

Subtests 22, 26, and 30

These subtests write and read a block of data from memory in 8-bit, 16-bit, and 32-bit modes, respectively. To execute one of these three subtests:

1. Type **:0,subtest number**

2. Press **Return.**

   These prompts are displayed. Be sure to respond to each prompt.

```
MEMORY TEST
**Testing physical memory from 0x87F000 to
0xC00000**
MEMORY SUBTEST 22, 26, or 30
Write/read a block in 8 bit mode
Enter hex starting address                =>
Enter hex ending address                  =>
Enter hex data pattern                    =>
Enter decimal number of loops             =>
```

3. Press **Return.**

   This output is generated when the tests are successfully executed:

```
Starting address          = 0x000000
Ending address            = 0x000000
Data Pattern              = 0x0
Number of loops           = 0
```

   The subtest is completed when this prompt is displayed:

```
Enter memory subtest number=>
```

Subtests 23, 27, and 31

These subtests copy a block of data from memory in 8-bit, 16-bit, and 32-bit modes, respectively.

To execute one of these three subtests:

1. Type **:0,subtest number**

2. Press **Return**.

   These prompts are displayed. Enter data next to each prompt.

   ```
   MEMORY TEST
   MEMORY SUBTEST 23, 27, or 31
   Copy a block of data in ## bit mode
   Enter hex starting address                =>
   Enter hex ending address                  =>
   Enter destination address                 =>
   Enter decimal number of loops             =>
   ```

3. Press **Return**.

   This output is generated when the tests are successfully executed:

   ```
   Starting address          = 0x000000
   Ending address            = 0x000000
   Destination Address       = 0x0
   Number of loops           = 0
   ```

   The subtest is completed when this prompt is displayed:

   ```
   Enter memory subtest number          =>
   ```

Subtest 32

This subtest verifies the copy executed by subtests 23, 27, or 31. It compares four bytes of data at one time. To execute subtest 32:

1. Type **:0,32**

2. Press **Return**.

   These prompts are displayed on the terminal screen.

```
MEMORY TEST
MEMORY SUBTEST 32
Compare blocks of data
Enter hex starting address                =>
Enter hex ending address                  =>
Enter address block to compare            =>
Enter decimal number of loops             =>
```

3. Press **Return**.

   This output is generated when the tests are successfully executed:

```
Starting address          = 0x000000
Ending address            = 0x000000
To new location           = 0x000000
```

   The subtest is completed when this prompt is displayed:

```
Enter memory subtest number               =>
```

TEST 1: PAGE MAP TEST (p)

This test verifies the ability of the physical page map to write values to the correct locations in physical memory. It consists of one test which does the following:

1. writes addresses to two locations on two different pages in memory
2. swaps the page table entries for the two entries
3. reads the entries to verify that the values were swapped

To execute the Page Map test:

1. Type **p**
2. Press **Return**.

   This output is displayed on the screen to indicate that the test is being executed:

   ```
   MAP TRANSLATION TEST
   MAP TRANSLATION SUBTEST 1
   ```

   This error message is reported when the values are not swapped successfully:

   Page mapping error: Map number (x), logical address (y), and address read back (z) are x, y, and z.

TEST 2: PHYSICAL MEMORY MAP TESTS (o)

The physical memory map enables the CTIX kernel to track the access history of each physical page. Tracking allows the kernel to identify the least frequently used pages and to select those pages for swapping. The physical access RAM subtests verify that the access history exists and that it works correctly.

1. Tests all combinations of bits in the Physical Memory Map

2. Tests each bit in all map registers

3. Walks 0 and 1 through modified bits of each map register to detect address problems

NOTE: Because these subtests generate NMIs when the physical memory map refers to locations in memory that are unavailable, NMI and Parity errors are disabled. They are enabled when the test is completed regardless of their previous values.

To execute the physical memory map test sequence (subtests 1 through 3):

1. Type **o**

2. Press **Return**.

   Subtests 1 through 3 are confirmed as follows:

   PHYSICAL MEMORY MAP TESTS
   PHYSICAL MEMORY MAP Subtest1
   PHYSICAL MEMORY MAP Subtest2
   PHYSICAL MEMORY MAP Subtest3

SUBTEST 1: TESTS ALL COMBINATIONS OF BITS ON ONE ACCESS REGISTER

Subtest 1 verifies that the circuitry used to determine which register bits are activated is functioning correctly. It does this by testing all 16 combinations of bits on one map register.

To execute subtest 1:

1. Type **:2,1**

2. Press **Return**.

   This output confirms successful test completion:

   ```
   PHYSICAL MEMORY MAP TESTS
   PHYSICAL MEMORY MAP SUBTEST 1
   ```

   This error message is displayed when subtest 1 fails:

   Physical address register at (address = 91000000) is x; expected y; during the read/write/or execute cycle.

SUBTEST 2: TESTS EACH BIT IN ALL ACCESS REGISTERS

Subtest 2 tests all the physical access registers (including those occupied by the diagnostic), except for the three pages that surround the region in memory where the program text for this subtest resides. It differs from subtest 1 in that it tests all locations in memory, but not all combinations.

To execute subtest 2:

1. Type **:2,2**

2. Press **Return**.

   This output confirms successful test completion:

   ```
   PHYSICAL MEMORY MAP TESTS
   PHYSICAL MEMORY MAP SUBTEST2
   ```

   This error message is displayed when subtest 2 fails:

   Page access bits wrong; page number and page bits are x and y.

SUBTEST 3: WALKS 0 AND 1 THROUGH MODIFIED BITS OF EACH ACCESS REGISTER TO DETECT ADDRESS PROBLEMS

Subtest 3 sets all access bits to zero, sets the dirty bit to a 1, then checks that the rest of the bits did not change. This subtest is repeated for each location, and does not test the map area that points to the diagnostic program. This subtest takes approximately 30 minutes to complete.

To execute subtest 3:

1. Type **:2,3**
2. Press **Return**.

   This output confirms successful test completion:

   ```
   PHYSICAL MEMORY MAP TESTS
   PHYSICAL MEMORY MAP SUBTEST3
   ```

   This error message is displayed when subtest 3 fails:

   ```
   Page access bits wrong: address x expected y got z
   ```

TEST 3: PARITY TESTS (r)

This test sequence ensures that the General Status Register (GSR) handles parity errors properly. The three subtests are described on the following pages.

1. Tests Parity Error bits in the GSR

2. Tests all available physical page numbers in the GSR

3. Executes a program with a parity error

NOTE: NMI, bus faults, and parity errors are enabled before these subtests are executed. They remain enabled after the subtests are completed. You must use PE to disable them.

To execute the parity test sequence (subtests 1 through 3):

1. Type **r**

2. Press **Return**.

   This output confirms test execution:

   ```
   PARITY TEST
   PARITY TEST-SUBTEST 1 PE bits in the GSR
   PARITY TEST-SUBTEST 2 physical pages in GSR
   PARITY TEST-SUBTEST 3 execution test
   ```

   The parity subtests do not test the area in memory occupied by the diagnostic program.

SUBTEST 1: TESTS PARITY ERROR BITS IN THE GSR

This subtest tests all combinations of Parity Error bits in the GSR for each available megabyte of physical memory. It ensures that all combinations of the four parity error bits (PE0, PE1, PE2, PE3) in the GSR are working correctly.

Parity errors are forced on PE0, then PE1 and so on to PE3. This is done starting at address 0x000000 for each 1M byte in physical memory. If the memory is not present, then the subtest moves to the next 1M byte of physical memory. Subtest 1 does not destroy memory. Prior to this test, the contents of memory are saved.

To execute subtest 1:

1. Type **:3,1**

2. Press **Return**.

   This output is displayed on the screen upon successful test completion:

   ```
   PARITY TEST
   PARITY TEST-SUBTEST 1 test PE bits in GSR
   ```

   When this subtest fails, an error message reports an address in virtual memory.

SUBTEST 2: TESTS ALL AVAILABLE PHYSICAL PAGE NUMBERS IN THE GSR

This subtest writes even (wrong) parity to the first four bytes of a page in physical memory. Then it writes a single byte of even (wrong) parity to the last byte in a page of physical memory. The subtest then reads all five bytes back to ensure that a Parity Error was reported and that the addresses in physical memory were correctly reported by the GSR.

To execute the subtest 2:

1. Type **:3,2**

2. Press **Return**.

   This output is displayed on the screen upon successful test completion:

   ```
   PARITY TEST
   PARITY TEST-SUBTEST 2 physical page numbers in
   the GSR
   ```

SUBTEST 3: EXECUTES A PROGRAM WITH A PARITY ERROR

This subtest executes a program that contains a parity error. It does so in both Supervisor and User modes. A program is written with even (wrong) parity so that a parity error occurs when the program is executed.

To execute subtest 3:

1. Type **:3,3**

2. Press **Return**.

   If a parity error occurs, the subtest is successful and this output is displayed on the screen:

   ```
   PARITY TEST
   PARITY TEST - SUBTEST 3 EXECUTION TEST
   ```

   If no parity error is reported by the GSR, the test is unsuccessful and one of these two error messages is displayed:

   ```
   No parity interrupt during execution.
   ```

   ```
   A parity error was not properly detected during
   execution test - User/Supervisor mode.
   ```

TEST 4: CLOCK TESTS (c)

The eight clock subtests are listed in Table 6.

Table 6

THE CLOCK SUBTESTS

Subtest	Purpose
1	Data Path Check to 8259 and 8253
2	Test Timer 0
3	Test Timer 1
4	Test Timer 2
5	Test clock calendar
6	Set the clock calendar
7	Read the data and time from the clock calendar
8	Special frequency calibration mode

To execute the clock test sequence (subtests 1 through 8):

1. Type **c**

2. Press **Return**.

   The following output is displayed on the screen to confirm that the subtests are being executed:

```
CLOCK TEST
Data Path Check
Subtest 2 - Timer 0 TEST
Subtest 3 - Timer 1 TEST
9 8 7 6 5 4 3 2 1
Subtest 4 - Timer 2 TEST
Subtest 5 - Clock calendar
Testing calibration for 41 seconds. . .
System clock and clock calendar are in sync
The time from the clock calendar is: Thu Aug 29
16:47:55 GMT 1998
```

SUBTEST 1: DATA PATH CHECK TO 8259 AN 8253

To execute subtest 1:

1. Type **:4,1**
2. Press **Return.**

This output is displayed:

```
CLOCK TEST
Data Path Check
```

SUBTEST 2: TEST TIMER 0

To execute subtest 2:

1. Type **:4,2**
2. Press **Return.**

This output is displayed:

```
CLOCK TEST
Subtest 2 - Timer 0 TEST
```

SUBTEST 3: TEST TIMER 1

To execute subtest 3:

1. Type **:4,3**
2. Press **Return.**

This output is displayed:

```
CLOCK TEST
Subtest 3 - Timer 1 TEST
9 8 7 6 5 4 3 2 1
```

SUBTEST 4: TEST TIMER 2

To execute subtest 4:

1. Type **:4,4**
2. Press **Return**.

This output is displayed:

```
CLOCK TEST
Subtest 4 - Timer 2 TEST
```

SUBTEST 5: TEST CLOCK CALENDAR

To execute subtest 5:

1. Type **:4,5**
2. Press **Return**.

This output is displayed. The number of seconds is determined at random in the diagnostic.

```
CLOCK TEST
Subtest 5 - Clock Calendar
Testing calibration for 41 seconds...
```

SUBTEST 6: SET THE CLOCK CALENDAR

Subtest 6 enables you to set the clock calendar in Greenwich Mean Time (GMT).

To execute subtest 6:

1. Type **:4,6**

2. Press **Return**.

This output is displayed:

```
    CLOCK TEST
Set the clock calendar (in GMT)
Enter the current time: MMDDHHMMYYW (W=0 is Sunday)
                                   :
```

where

MM stands for the month number (01 to 12)
DD is the date (01 through 31)
HH is the hour in military format (00 to 23)
MM is the minute (00 to 59)
YY is the last two digits in the year
W is the day of the week, as follows

- 0 = Sunday
- 1 = Monday
- 2 = Tuesday
- 3 = Wednesday
- 4 = Thursday
- 5 = Friday
- 6 = Saturday

Leading zeros must be included where applicable. For example, to set the date and time for Thursday, August 29, 1998 at 4:47 PM GMT, you would enter

```
08291647984
```

SUBTEST 7: READ THE DATA AND TIME FROM THE CLOCK CALENDAR

To execute subtest 7:

1. Type **:4,7**
2. Press **Return**.

This output is displayed:

```
CLOCK TEST
The time from the clock calendar is: Thu Aug 29 16:47:55 GMT 1998
```

SUBTEST 8: SPECIAL FREQUENCY CALIBRATION MODE

This subtest tests the frequency calibration mode on the 58321 chip. For more information on this mode, see the Oki 58321 specification.

To execute subtest 8:

1. Type **:4,8**
2. Press **Return**.

This output is displayed:

```
CLOCK TEST
Entering Calibration mode.
Issue a read clock command (:4,7) to bring the
calendar chip back to normal state.
```

TEST 10: MAP RAM TEST (a)

The map RAM test sequence verifies that the static RAMs used by the page tables to control virtual memory mapping are functioning correctly. Of the 32 bits used to test the static RAMS, only 14 are significant: the two page protection bits and the 12 page frame number bits. The remaining undefined bits are ignored during this test.

The map RAMS contain no parity; therefore, no parity errors can occur. For this reason, subtest 4 writes and reads 5's to the static RAMs instead of the 54's that the memory subtests write to the dynamic RAMs. The map RAM subtests are listed in Table 7.

Table 7

MAP RAM SUBTESTS

Subtest	Description
1	Write/read 0x00000000's
2	Write/read 0xffffffff's
3	Write/read 0xaaaaaaaa's
4	Write/read 0x55555555's
5	Write/read addresses
6	Write/read address complements

There are two passes to each subtest. The CPU first writes data to an address in memory, then it immediately reads that data back. If the CPU reads exactly what it wrote, then it executes the second read pass.

If, on the second pass, the CPU reads something different from what it wrote, it either logs the error message (Long mode) or terminates (Continuous) the test and displays this error message on the screen:

Memory error at address x. Wrote y; Read Back z

If the first read pass is successful, the second read pass is conducted as follows: the CPU reads the data a second time and if it does not read the same data as was written, an address error is reported. An address error occurs when the CPU does not read (for the second time) what was written to memory. If an error occurs during the second pass, this error is reported:

Suspected address error at x; read y expected z

The difference between the two errors concerns the timing of the read operation. The first error message occurs when the read operation does not function at all. The second error message is reported when the second read operation fails. The second type of failure implies that the data may have been overwritten, and thus an address error is reported.

To execute the entire test sequence:

1. Type **a**

2. Press **Return**.

   The following output is displayed on your screen to confirm that the subtests are being executed:

```
MAP RAM TEST
MAP RAM SUBTEST 1
MAP RAM SUBTEST 2
MAP RAM SUBTEST 3
MAP RAM SUBTEST 4
MAP RAM SUBTEST 5
MAP RAM SUBTEST 6
```

SUBTEST 1: WRITE AND READ 0x00000000's

To execute subtest 1:

1. Type **:10,1**

2. Press **Return**.

   The following output is displayed on your screen to confirm that the subtest is being executed:

```
MAP RAM TEST
MAP RAM SUBTEST 1
```

SUBTEST 2: WRITE AND READ 0xffffffff's

To execute subtest 2:

1. Type **:10,2**

2. Press **Return.**

   The following output is displayed on your screen to confirm that the subtest is being executed:

```
MAP RAM TEST
MAP RAM SUBTEST 2
```

SUBTEST 3: WRITE AND READ 0xaaaaaaaa's

To execute subtest 3:

1. Type **:10,3**

2. Press **Return.**

   The following output is displayed on your screen to confirm that the subtest is being executed:

```
MAP RAM TEST
MAP RAM SUBTEST 3
```

SUBTEST 4: WRITE AND READ 0x55555555's

To execute subtest 4:

1. Type **:10,4**

2. Press **Return**.

   The following output is displayed on your screen to confirm that the subtest is being executed:

```
MAP RAM TEST
MAP RAM SUBTEST 4
```

SUBTEST 5: WRITE AND READ ADDRESSES

To execute subtest 5:

1. Type **:10,5**

2. Press **Return**.

   The following output is displayed on your screen to confirm that the subtest is being executed:

```
MAP RAM TEST
MAP RAM SUBTEST 5
```

SUBTEST 6: WRITE AND READ ADDRESS COMPLEMENTS

To execute subtest 6:

1. Type **:10,6**

2. Press **Return.**

   The following output is displayed on your screen to confirm that the subtest is being executed:

```
MAP RAM TEST
MAP RAM SUBTEST 6
```

TEST 12: PAGE PROTECTION TEST (e)

Both the software and hardware in the S/MT Series system can affect page protection. The software can protect the two Virtual Memory regions and prevent users from accessing the VME region in memory. See the hardware or technical reference manual for your system for information on these locations.

The hardware can prevent access to the Illegal region in memory and protect users from accessing the regions in memory (except the VME region) located above 2G bytes. The two page protection subtests are presented in Table 8.

Table 8

PAGE PROTECTION SUBTESTS

Subtest	Description
1	Checks all combinations of hardware page protection status, User and Supervisor, and read, write, and execute modes
2	In User mode, reads and writes to each page in the Mapped RAM, and each megabyte in Unused and I/O space.

To execute the two page protection subtests:

1. Type **e**

2. Press **Return**.

   This output is displayed on the terminal screen to indicate execution of the page protection test sequence:

```
PAGE PROTECTION TEST
PAGE PROTECTION SUBTEST 1
PAGE PROTECTION SUBTEST 2
Mapped RAM (0x00100000 to 0x18000000)
Mapped RAM (0x7f800000 to 0x80000000)
Unused Region (0x01800000 to 0x7f800000)
IO Space (0x80000000 to 0xffffffff)
```

SUBTEST 1: CHECKS ALL COMBINATIONS OF HARDWARE PAGE PROTECTION STATUS

This subtest selects a single page in Virtual Memory. It reads, writes, and executes each type of page protection in both User and Supervisor modes. The subtest is looking for expected results. That is, it expects to generate a page fault error when it tests an invalid page.

Pages are protected in one of four ways. The four page protection techniques are further determined by User or Supervisor mode. As follows:

- not valid
- kernel read/write
- user read
- user read/write

To execute subtest 1:

1. Type **:12,1**

2. Press **Return**.

   The diagnostic displays this output upon successful completion:

```
PAGE PROTECTION TEST
PAGE PROTECTION SUBTEST 1
```

Subtest 1 can generate these error messages:

No page fault received on write

No page fault received on read

No page fault received on execute

Page fault when none was expected at address xx, with these protection bits, and in user or supervisor mode

Write/Read-back not continued properly

SUBTEST 2: READS AND WRITES TO EACH PAGE IN THE MAPPED RAM AND TO EACH MEGABYTE IN UNUSED I/O SPACE

This subtest reads and writes data in User mode to every page in the two virtual memory regions. Then, in both User and Supervisor modes, it tests the Unused memory region. The subtest is successful when page faults are generated.

Finally, this subtest tests the I/O space by accessing all five I/O regions, in User read/write mode and in increments of 1M byte. User access to all VME regions is denied during this part of subtest 2.

To execute subtest 2:

1. Type **:12,2**

2. Press **Return.**

Failure of subtest 2 can generate one of these error messages:

No page fault during * mode, * access, to addr *.

Unexpected page fault during* mode, * access, to addr *.

No page fault during User mode read access to addr <addr>.

No page fault during Supervisor mode write access to addr <addr>.

Note that mode is User or Supervisor; access is read, write, or execute; and addr is the address of the page that failed the test.

TEST 13: LINE PRINTER TESTS (1)

The line printer test sequence examines the parallel printer and its interface for the printer connectors on the Main Processor board, the IOP Expansion board, and the RS-422 Expansion board. This test sequence consists of the six subtests listed in Table 9.

Table 9

LINE PRINTER SUBTESTS

Subtest	Description
1	Check and display the line printer status register
2	Print a rotating pattern on the printer using interrupts
3	Check the control lines of the printer connector with a loopback plug
4	Same as subtest 1 for the line printer connector on IOP or RS-422 Expansion board
5	Same as subtest 2 for the line printer connector on IOP or RS-422 Expansion board
6	Same as subtest 3 (requires a loopback plug) for the line printer connector on IOP or RS-422 Expansion board

To execute the test sequence:

1. Type **l** (el)

2. Press **Return.**

The line printer test sequence executes the appropriate subtest (2 or 3) depending on whether a printer or loopback plug is detected. If neither a printer nor a loopback plug are detected, a warning message is displayed and the test sequence continues. Pin assignments for the line printer loopback plug are contained in Appendix A.

When either the IOP Expansion board or the RS-422 Expansion board is present, the default test sequence executes the same subtests on the line printer connector present on the expansion board.

SUBTEST 1: CHECK AND DISPLAY THE LINE PRINTER STATUS REGISTER

Subtest 1 checks the status of the data/status register, returns the status, and makes sure that the status is valid (both line printer present and line printer selected, or line printer out of paper).

To execute subtest 1:

1. Type **:13,1**

2. Press **Return.**

   Output like this is displayed as the subtest is executed:

```
LINE PRINTER TEST
LINE PRINTER SUBTEST 1, (onboard) Status test
```

SUBTEST 2: PRINTS A ROTATING PATTERN ON THE PRINTER USING INTERRUPTS, THEN CHECKS THE STATUS OF EACH INTERRUPT

For each repetition, subtest 2 prints a rotating pattern to the printer, which consists of 80-column rotations of the ASCII pattern set.

To execute subtest 2:

1. Type **:13,2**

2. Press **Return**.

   Output like this is displayed as the subtest is executed:

```
LINE PRINTER TEST
LINE PRINTER SUBTEST 2, (onboard) Transfer test
```

SUBTEST 3: CHECK THE CONTROL LINES OF THE PRINTER CONNECTOR WITH A LOOPBACK PLUG

To execute subtest 3:

1. Type **:13,3**

2. Press **Return**.

   Output like this is displayed as the subtest is executed:

```
LINE PRINTER TEST
LINE PRINTER SUBTEST 3, (onboard) Loopback test
```

SUBTEST 4: CHECKS AND DISPLAYS THE LINE PRINTER STATUS REGISTER ON THE IOP OR RS-422 EXPANSION BOARD

To execute subtest 4:

1. Type **:13,4**

2. Press **Return.**

   Output like this is displayed as the subtest is executed. The asterisk refers to either the IOP Expansion board (IOP) or the RS-422 Expansion board (422).

```
LINE PRINTER TEST
LINE PRINTER SUBTEST 4, (*) Status test
```

SUBTEST 5: CHECKS INTERRUPT STATUS

To execute subtest 5:

1. Type **:13,5**

2. Press **Return.**

   Output like this is displayed as the subtest is executed. The asterisk refers to either the IOP Expansion board (IOP) or the RS-422 Expansion board (422).

```
LINE PRINTER TEST
LINE PRINTER SUBTEST 5, (*) Transfer test
```

SUBTEST 6: WITH A LOOPBACK PLUG, CHECKS THE CONTROL LINES OF THE PRINTER CONNECTOR ON THE IOP OR RS-422 EXPANSION BOARD

To execute subtest 6:

1. Type **:13,6**

2. Press **Return**.

   Output like this is displayed as the subtest is executed. The asterisk refers to either the IOP Expansion board (IOP) or the RS-422 Expansion board (422).

```
LINE PRINTER TEST
LINE PRINTER SUBTEST 6, (*) Loopback test
```

CPU-02 Diagnostic Tests

CPU-02 DIAGNOSTIC TESTS

This chapter describes the CPU-02 Subsystem diagnostic programs, most of which are listed in Table 1.

The CPU-02 (shown in Figure 1) is installed is many systems (including the S/640) and can be distinguished from the CPU-01 board by the location of the Uninterruptable Power Supply (UPS) connector.

Table 1

CPU-02 DIAGNOSTIC TESTS

Number	Letter	Description
0	m	Memory
2	o	Physical Memory
4	c	Clock
10	a	Data Cache and Page Table Cache (PTC) RAM
12	e	Page Protection
13	l	Line Printer

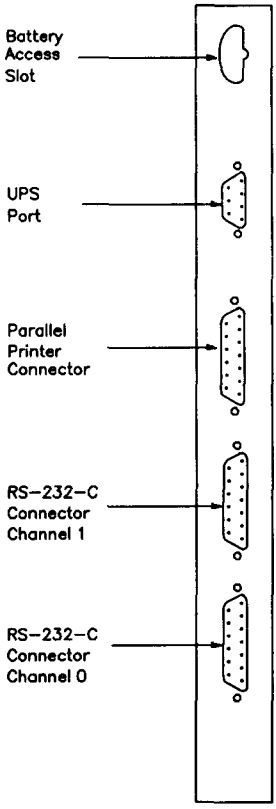


Figure 2. CPU-02 Board

TEST 0: MEMORY TEST (m)

The memory test writes and then reads data to every location in memory. Error messages are displayed when the diagnostic reads something other than what it wrote. Table 2 contains a summary of memory subtests.

The basic memory test (m) consists of subtests 1 through 13. Subtests 2 through 7 are 32-bit subtests. Subtests 15 and 20 through 32 require your input and therefore are not included in the test sequence.

Table 2

MEMORY SUBTESTS

Subtest	Description
1	Logic control to each 2M byte bank of memory
2	Write/read 0x00000000's (32-bit)
3	Write/read 0xffffffff's (32-bit)
4	Write/read 0xaaaaaaaa's (32-bit)
5	Write/read 0x55555555's (32-bit)
6	Write/read addresses (32-bit)
7	Write/read address complements (32-bit)
8	write/read addresses
9	write/read address complements
10	write/read 0x00000000's
11	write/read 0xFFFFFFFF's
12	write/read 0xaaaaaaaa's
13	write/read 0x55555555's
14	Mem test on check bits (addr, ~addr)
15	Set memory limits for above subtests
16	Ecc tests (check bits, syndrome, etc.)

Interactive tests:	8-bit	16-bit	32-bit
Write/read address	20	24	28
Read address	21	25	29
Write/read block	22	26	30
Copy block	23	27	31
Compare blocks			32

NONINTERACTIVE MEMORY SUBTESTS
(SUBTESTS 1-13, 14, 16)

Subtest 1 performs logic control to each 2M byte bank of memory. Subtests 2 through 13 write predefined data patterns to every location in memory. They each consist of two passes.

- write and then immediate read-back pass to detect pure memory errors
- read-only pass to detect address overlapping and refresh errors

Subtests 2 and 3 delay approximately 15 seconds between passes to detect refresh errors. To prevent memory access, the delay is executed out of the 68020 instruction cache which is enabled. Subtests 2 through 7 are described in Table 3.

Subtest 14 performs memory tests on check bits for all existing boards by running the following sequence:

- writing and reading 0s
- writing and reading Fs
- writing and reading a 7-bit address, then the inverse of that address

Subtest 16 verifies that ECC is operational. To do this, the following sequences are run on the upper and lower long words of each bank of memory:

- a walking 1 pattern is run on the readable bits of the Memory Status/Control registers
- a walking EX-OR pattern (reverses a check bit and looks to see that a software error was reported and that ECC corrected the bit in error) is run on each check bit

- a walking EX-OR pattern is run on each memory data bit

- a double-bit error is created by reversing two data bits, reading the data, and then looking to see that an NMI was generated and the correct address of the error is reported in the Memory board Error Address register

TABLE 3

MEMORY SUBTESTS 2 THROUGH 7

Subtest	Purpose
2	Write and read 0's in hexadecimal format to every location in memory in 32-bit.
3	Write and read f's in hexadecimal format to every location in memory in 32-bit
4	Write and read a's in hexadecimal format to every location in memory in 32-bit.
5	Write and read 0x55's in hexadecimal format to every location in memory in 32-bit mode.
6	Write and read the addresses of each location in memory in 32-bit.
7	Write and read the one's complements of addresses to every location in memory in 32-bit.

To execute subtests 1 through 13:

1. Type **m**

2. Press **Return**.

   The following output is displayed when the tests are successfully executed:

```
**Testing physical memory from 0x7B000 to 0xA00000**
 MEMORY TEST
 MEMORY SUBTEST 1
 MEMORY SUBTEST 2
 MEMORY SUBTEST 3
 MEMORY SUBTEST 4
 MEMORY SUBTEST 5
 MEMORY SUBTEST 6
 MEMORY SUBTEST 7
 MEMORY SUBTEST 8
 MEMORY SUBTEST 9
 MEMORY SUBTEST 10
 MEMORY SUBTEST 11
 MEMORY SUBTEST 12
 MEMORY SUBTEST 13
 MEMORY SUBTEST 14
 MEMORY SUBTEST 16
```

To execute one memory subtest, for example subtest 3:

1. Type **:0,3**

2. Press **Return**.

   The following output is generated when the test is successfully executed:

```
MEMORY TEST
MEMORY SUBTEST 3
```

INTERACTIVE MEMORY SUBTESTS (15 and 20-32)

CAUTION: Subtests 15 and 20 through 32 should only be run by qualified technicians who throughly understand S/MT Series virtual memory concepts.

Subtests 15 and 20 through 32 provide more control and allow more specific testing than subtests 1 through 7.

- subtest 15 allows you to set the memory limits for subtests 2 through 13.
- subtests 20 through 32 perform limited versions of subtests 2 thorugh 13

Subtests 20 through 32 allow you to define the memory limits and, in some cases, the number of repetitions, data patterns, and destination addresses required by the diagnostic to execute subtests 2 through 13. The purposes of subtests 20 through 32 are displayed in Table 4.

Required Input

Table 5 below summarizes the input you must provide to execute subtests 15 and 20 through 32.

CAUTION: When you respond to the prompts displayed by these subtests, be sure not to specify the area in memory where the diagnostic tests reside. The command to display system configuration (CN) displays the area in memory where the diagnostic tests begin.

Table 4

MEMORY SUBTESTS 15 THROUGH 32

Test Purpose	8-bit	16-bit	32-bit
Write and read address	20	24	28
Read address	21	25	29
Write and read block	22	26	30
Copy block	23	27	31
Compare blocks			32

Table 5

INPUT FOR SUBTESTS 15 THROUGH 32

Type of Input	Subtests
Starting address	15, 20 through 32
Ending address	15, 22, 23, 26, 27, 30, 31
Data pattern	20, 22, 24, 26, 28, 30
Destination address	23, 27, 31
Number of loops	15, 20 through 32

Hex Starting Address. The location in memory where you want the subtest to begin testing. The prompt looks like this:

```
Enter hex starting address                    =>
```

Hex Ending Address. The location in memory where you want the subtest to stop reading, writing, or copying to and from. The prompt looks like this:

```
Enter hex ending address                      =>
```

Hex Data Pattern. The pattern of information that you want the subtest to either write or read (or both) in memory. The prompt looks like this:

```
Enter hex data pattern                    =>
```

Hex Destination Address. The address where you want the subtest to write the data pattern defined above. The prompt looks like this:

```
Enter hex destination address             =>
```

Decimal Number of Loops. The number of times you want to execute the subtest, according to the parameters extablished above. The prompt looks like this:

```
Decimal number of loops                   =>
```

After you enter the decimal number of loops, the diagnostic displays a confirmation of what you just entered. The confirmation differs according to each test.

The prompt provides a convenient way for you to remain in the Interactive Memory Subtest menu. It provides two options.

- execute another interactive subtest
- exit the Interactive Subtest menu

To execute another subtest, type the subtest number next to the prompt. Otherwise, to exit the menu:

Press **Return**

A table presenting the interactive subtests is displayed. To return to the diagnostic prompt:

Type **99** (or press **Return**)

Subtest 15

This subtest lets you set the memory limits for subtests 2 through 13.

To execute subtest 15:

1. Type **:0,15**

2. Press **Return**.

   The following output is generated when the test is successfully executed:

   ```
   MEMORY TEST
   Testing physical memory from 0x87F000 to 0xC00000
   MEMORY SUBTEST 14
   Enter hex starting address                    =>
   ```

3. Type the starting hexadecimal address.

4. Press **Return**.

   This prompt is displayed:

   ```
   Enter hex ending address                      =>
   ```

5. Type the ending hexadecimal address.

6. Press **Return**.

   This prompt is displayed:

   ```
   Enter decimal number of loops                 =>
   ```

7. Enter the number of times you want to execute subtests 2 through 13.

   The number of times that subtest 14 executes subtests 1 through 13 depends upon the number of loops you select.

8. Press **Return**.

Subtests 2 through 13 are executed. The following output is displayed as the tests are executed.

```
MEMORY TEST
MEMORY SUBTEST 1
MEMORY SUBTEST 2
MEMORY SUBTEST 3
MEMORY SUBTEST 4
MEMORY SUBTEST 5
MEMORY SUBTEST 6
MEMORY SUBTEST 7
MEMORY SUBTEST 8
MEMORY SUBTEST 9
MEMORY SUBTEST 10
MEMORY SUBTEST 11
MEMORY SUBTEST 12
MEMORY SUBTEST 13
```

Subtests 20, 24, and 28

These subtests write and read data to a location in memory in 8-bit, 16-bit, and 32-bit modes, respectively. To execute one of these three subtests:

1. Type **:0,subtest number**

2. Press **Return**.

   These prompts are displayed. Enter data next to each prompt.

```
MEMORY TEST
**Testing physical memory from 0x87F000 to
0xC00000**
MEMORY SUBTEST 20, 24, or 28
Write/read an address in ## bit mode
Enter hex starting address                    =>
Enter hex data pattern                        =>
Enter decimal number of loops                 =>
```

3. Press **Return**.

   This output is generated when the tests are successfully executed:

```
Starting address          = 0x000000
Data Pattern              = 0x0
Number of loops           = 0
```

   The subtest is completed when this prompt is displayed:

```
Enter memory subtest number                   =>
```

Subtests 21, 25, and 29

These subtests read data from a location in memory in 8-bit, 16-bit, and 32-bit modes, respectively. To execute one of these three subtests:

1. Type **:0,subtest number**

2. Press **Return**.

   These prompts are displayed. Enter data next to each prompt.

```
MEMORY TEST
**Testing physical memory from 0x87F000 to
0xC00000**
MEMORY SUBTEST 21, 25, or 29
Read from an address in ## bit mode
Enter hex starting address                    =>
Enter decimal number of loops                 =>
```

3. Press **Return**.

   This output is generated when the tests are successfully executed:

```
Starting address          = 0x000000
Number of loops           = 0
Address 0x000000          = 0x00
```

   The subtest is completed when this prompt is displayed:

```
Enter memory subtest number                   =>
```

Subtests 22, 26, and 30

These subtests write and read a block of data from memory in 8-bit, 16-bit, and 32-bit modes, respectively. To execute one of these three subtests:

1. Type **:0,subtest number**

2. Press **Return**.

   These prompts are displayed. Be sure to respond to each prompt.

```
MEMORY TEST
**Testing physical memory from 0x87F000 to
0xC00000**
MEMORY SUBTEST 22, 26, or 30
Write/read a block in 8 bit mode
Enter hex starting address                    =>
Enter hex ending address                      =>
Enter hex data pattern                        =>
Enter decimal number of loops                 =>
```

3. Press **Return**.

   This output is generated when the tests are successfully executed:

```
Starting address          = 0x000000
Ending address            = 0x000000
Data Pattern              = 0x0
Number of loops           = 0
```

   The subtest is completed when this prompt is displayed:

```
Enter memory subtest number                   =>
```

Subtests 23, 27, and 31

These subtests copy a block of data from memory in 8-bit, 16-bit, and 32-bit modes, respectively.

To execute one of these three subtests:

1. Type **:0,subtest number**

2. Press **Return**.

   These prompts are displayed. Be sure to enter data next to each prompt.

```
MEMORY TEST
MEMORY SUBTEST 23, 27, or 31
Copy a block of data in ## bit mode
Enter hex starting address                =>
Enter hex ending address                  =>
Enter destination address                 =>
Enter decimal number of loops             =>
```

3. Press **Return**.

   This output is generated when the tests are successfully executed:

```
Starting address          = 0x000000
Ending address            = 0x000000
Destination Address       = 0x0
Number of loops           = 0
```

   The subtest is completed when this prompt is displayed:

```
Enter memory subtest number               =>
```

Subtest 32

This subtest verifies the copy executed by subtests 23, 27, or 31. It compares four bytes of data at one time. To execute subtest 32:

1. Type **:0,32**

2. Press **Return.**

   These prompts are displayed on the terminal screen.

```
MEMORY TEST
MEMORY SUBTEST 32
Compare blocks of data
Enter hex starting address                    =>
Enter hex ending address                      =>
Enter address block to compare                =>
Enter decimal number of loops                 =>
```

3. Press **Return.**

   This output is generated when the tests are successfully executed:

```
Starting address          = 0x000000
Ending address            = 0x000000
To new location           = 0x000000
```

   The subtest is completed when this prompt is displayed:

```
Enter memory subtest number                   =>
```

TEST 2: PHYSICAL MEMORY MAP TESTS (o)

The physical memory map enables the CTIX kernel to track the access history of each physical page. Tracking allows the kernel to identify the least frequently used pages and to select those pages for swapping. The physical access RAM subtests verify that the access history exists and that it works correctly by executing the following:

1. Tests all combinations of bits in the Physical Memory Map.

2. Tests each bit in all map registers.

3. Walks 0 and 1 through modified bits of each map register to detect address problems.

NOTE: Because these subtests generate NMIs when the physical memory map refers to locations in memory that are unavailable, NMI errors are disabled. They are enabled when the test is completed.

To execute the physical memory map test sequence (subtests 1 through 3),

1. Type **o**

2. Press **Return.**

   Subtests 1 through 3 are confirmed as follows:

```
PHYSICAL MEMORY MAP TESTS
PHYSICAL MEMORY MAP Subtest1
PHYSICAL MEMORY MAP Subtest2
PHYSICAL MEMORY MAP Subtest3
```

SUBTEST 1: TESTS ALL COMBINATIONS OF BITS ON ONE ACCESS REGISTER

Subtest 1 verifies that the circuitry used to determine which register bits are activated is functioning correctly. It does so by testing all 16 combinations of bits on one map register.

To execute subtest 1:

1. Type **:2,1**

2. Press **Return**.

   This output confirms successful test completion:

   ```
   PHYSICAL MEMORY MAP TESTS
   PHYSICAL MEMORY MAP SUBTEST 1
   ```

   This error message is displayed when subtest 1 fails:

   ```
   Physical access register at (address = AAAAAAAA) is
   x; expected y; during the read/write/or execute cycle.
   ```

SUBTEST 2: TESTS EACH BIT IN ALL ACCESS REGISTERS

Subtest 2 tests all the physical access registers (including those occupied by the diagnostic), except for the three pages that surround the region in memory where the program text for this subtest resides. It differs from subtest 1 in that it tests all locations in memory, but not all combinations.

To execute subtest 2:

1. Type **:2,2**

2. Press **Return**.

   This output confirms successful test completion:

   ```
   PHYSICAL MEMORY MAP TESTS
   PHYSICAL MEMORY MAP SUBTEST2
   ```

   This error message is displayed when subtest 2 fails:

   Page access bits wrong; page number and page bits are x and y.

SUBTEST 3: WALKS 0 AND 1 THROUGH MODIFIED BITS OF EACH ACCESS REGISTER TO DETECT ADDRESS PROBLEMS

Subtest 3 sets all access bits to zero, then sets one bit (the dirty bit) to a 1, then checks that the rest of the bits did not change. This subtest is repeated for each location, and does not test the map area that points to the diagnostic program. This subtest takes approximately 30 minutes to complete.

To execute subtest 3:

1. Type **:2,3**

2. Press **Return**.

   This output confirms successful test completion:

   ```
   PHYSICAL MEMORY MAP TESTS
   PHYSICAL MEMORY MAP SUBTEST3
   ```

   This error message is displayed when subtest 3 fails:

   ```
   Page access bits wrong: address x expected y got z
   ```

TEST 4: CLOCK TESTS (c)

The eight clock subtests are listed in Table 6.

Table 6

THE CLOCK SUBTESTS

Subtest	Purpose
1	Data Path Check to 8259 and 8253
2	Test Timer 0
3	Test Timer 1
4	Test Timer 2
5	Test clock calendar
6	Set the clock calendar
7	Read the data and time from the clock calendar
8	Special frequency calibration mode

To execute the clock test sequence (subtests 1 through 8):

1. Type **c**

2. Press **Return**.

   The following output is displayed on the screen to confirm that the subtests are being executed:

```
CLOCK TEST
Data Path Check
Subtest 2 - Timer 0 TEST
Subtest 3 - Timer 1 TEST
9 8 7 6 5 4 3 2 1
Subtest 4 - Timer 2 TEST
Subtest 5 - Clock calendar
Testing calibration for 41 seconds. . .
System clock and clock calendar are in sync
The time from the clock calendar is: Thu Aug 29
16:47:55 GMT 1998
```

SUBTEST 1: DATA PATH CHECK TO 8259 AN 8253

To execute subtest 1:

1. Type **:4,1**
2. Press **Return**.

This output is displayed:

```
CLOCK TEST
Data Path Check
```

SUBTEST 2: TEST TIMER 0

To execute subtest 2:

1. Type **:4,2**
2. Press **Return**.

This output is displayed:

```
CLOCK TEST
Subtest 2 - Timer 0 TEST
```

SUBTEST 3: TEST TIMER 1

To execute subtest 3:

1. Type **:4,3**
2. Press **Return**.

This output is displayed:

```
CLOCK TEST
Subtest 3 - Timer 1 TEST
9 8 7 6 5 4 3 2 1
```

SUBTEST 4: TEST TIMER 2

To execute subtest 4:

1. Type **:4,4**

2. Press **Return.**

This output is displayed:

```
CLOCK TEST
Subtest 4 - Timer 2 TEST
```

SUBTEST 5: TEST CLOCK CALENDAR

To execute subtest 5:

1. Type **:4,5**

2. Press **Return.**

This output is displayed. The number of seconds is determined at random in the diagnostic.

```
CLOCK TEST
Subtest 5 - Clock Calendar
Testing calibration for 41 seconds...
```

SUBTEST 6: SET THE CLOCK CALENDAR

Subtest 6 enables you to set the clock calendar in Greenwich Mean Time (GMT).

To execute subtest 6:

1. Type **:4,6**

2. Press **Return**.

This output is displayed:

```
    CLOCK TEST
Set the clock calendar (in GMT)
Enter the current time: MMDDHHMMYYW (W=0 is Sunday)
                                   :
```

where

MM stands for the month number (01 to 12)
DD is the date (01 through 31)
HH is the hour in military format (00 to 23)
MM is the minute (00 to 59)
YY is the last two digits in the year
W is the day of the week, as follows

- 0 = Sunday
- 1 = Monday
- 2 = Tuesday
- 3 = Wednesday
- 4 = Thursday
- 5 = Friday
- 6 = Saturday

Leading zeros must be included where applicable. For example, to set the date and time for Thursday, August 29, 1998 at 4:47 PM GMT, you would enter

```
08291647984
```

SUBTEST 7: READ THE DATA AND TIME FROM THE CLOCK CALENDAR

To execute subtest 7:

1. Type **:4,7**
2. Press **Return**.

This output is displayed:

```
CLOCK TEST
The time from the clock calendar is: Thu Aug 29 16:47:55 GMT 1998
```

SUBTEST 8: SPECIAL FREQUENCY CALIBRATION MODE

This subtest tests the frequency calibration mode on the 58321 chip. For more information on this mode, see the Oki 58321 specification.

To execute subtest 8:

1. Type **:4,8**
2. Press **Return**.

This output is displayed:

```
CLOCK TEST
Entering Calibration mode.
Issue a read clock command (:4,7) to bring the
calendar chip back to normal state.
```

TEST 10: DATA CACHE AND PAGE TABLE CACHE RAM (a)

Test 10 verifies operation of the data cache and page table cache RAM by performing memory tests on the areas listed in Table 7.

Table 7

DATA CACHE AND PAGE TABLE CACHE SUBTESTS

Subtest	Description
1	Data cache memory test
2	Data cache tag memory test
3	DMA map memory test
4	Group table memory test
5	Page table cache memory test
6	Page table cache tag memory test
7	Force NMI on PTC fill
8	Page table cache fill test
9	Pointer register tests
10	Indirect Pointer register tests
11	Data cache memory test through virtual space: 0x100000 - 0x120000

For subtests 1 through 6, there are eight passes to each subtest.

- walking 1s
- walking 0's
- all 0's
- all F's
- all 55's
- all A's
- address
- address complement

The CPU first writes data to an address in memory, then it immediately reads that data back. If the CPU reads exactly what it wrote, then it executes the second read pass.

For subtests 1, 2, and 11 there is a delay after the write pass of the pattern. This delay is similar to the delay in memory test subtests 2 and 3, as it is executed out of the 68020 instruction cache to prevent any access to memory.

When running these subtests as part of the test sequence (for example, test sequence a), the delay will be approximately 16 seconds. When each subtest is run individually, you will be prompted to enter in a delay count in hex. The default, 0x1000000, corresponds to approximately 16 seconds and is selected by pressing **Return**. Each count adds approximately 720 nanoseconds of delay time.

To execute the entire test sequence:

1. Type **a**

2. Press **Return**.

   The following output is displayed on your screen to confirm that the subtests are being executed:

```
Data Cache and Page Table Cache Tests
Data cache memory tests.
Data cache tag memory test.
DMA map memory test.
Group table memory test.
Page table cache memory test.
Page table cache tag memory test.
Force NMI on PTC fill.
Page table cache fill test.
Pointer Register tests.
Indirect pointer register tests.
Data cache memory test thru virtual space: 0x100000-
0x120000
```

SUBTEST 1: DATA CACHE MEMORY TEST

This subtest verifies that the data memory cache is operational by running the six memory test sequences. In addition to the six tests there is a "checkerboard" test pattern. This pattern consists of 8 bytes of 0xAs followed by 8 bytes of 0x5s.

To execute subtest 1:

1. Type **:10,1**

2. Press **Return**.

   The following output appears on the screen:

```
Data Cache and Page Table Cache Tests
Data cache memory tests.
Enter delay count: (hex) (default: 0x1000000)
```

SUBTEST 2: DATA CACHE TAG MEMORY TEST

This subtest verifies the operation of the cache tags (a 9-bit pointer for each cache entry which points to the entry's location in physical memory). The test performed is the same memory test sequence described in subtest 1.

To execute subtest 2:

1. Type **:10,2**

2. Press **Return**.

   The following output appears on the screen:

```
Data Cache and Page Table Cache Tests
Data cache tag memory test.
Enter delay count: (hex) (default: 0x1000000)
```

SUBTEST 3: DMA MAP MEMORY TEST

To execute subtest 3:

1. Type **:10,3**

2. Press **Return**.

   The following output appears on the screen:

```
Data Cache and Page Table Cache Tests
DMA map memory test.
```

SUBTEST 4: GROUP TABLE MEMORY TEST

To execute subtest 4:

1. Type **:10,4**
2. Press **Return**.

   The following output appears on the screen:

   ```
   Data Cache and Page Table Cache Tests
   Group table memory test.
   ```

SUBTEST 5: PAGE TABLE CACHE MEMORY TEST

To execute subtest 5:

1. Type **:10,5**
2. Press **Return**.

   The following output appears on the screen:

   ```
   Data Cache and Page Table Cache Tests
   Page table cache memory test.
   ```

SUBTEST 6: PAGE TABLE CACHE TAG MEMORY TEST

To execute subtest 6:

1. Type **:10,6**
2. Press **Return**.

   The following output appears on the screen:

   ```
   Data Cache and Page Table Cache Tests
   Page table cache tag memory test.
   ```

SUBTEST 7: FORCE NMI ON PTC FILL

This subtest imitates a page Table Cache entry, sets up the group table to point to a segment table in memory which is not present, and then accesses an address in that segment and confirms that an NMI occurred and that the appropriate bits in the GSR were latched.

To execute subtest 7:

1. Type **:10,7**

2. Press **Return.**

   The following output appears on the screen:

   ```
   Data Cache and Page Table Cache Tests
   Force NMI on PTC fill.
   ```

SUBTEST 8: PAGE TABLE CACHE FILL TEST

This subtest tests the data path between memory and the Page Table Cache, which is used during a PTC fill. Since subtest 5 tests all of the memory in the PTC, this data path is tested with a walking 1's test across the data bus, and then the address bus.

To execute subtest 8:

1. Type **:10,8**

2. Press **Return.**

   The following output appears on the screen:

   ```
   Data Cache and Page Table Cache Tests
   Page Table Cache fill test.
   ```

SUBTEST 9: POINTER REGISTER TESTS

This subtest tests the Pointer register by directly accessing addresses in the range 0x7A00x000. (The Pointer register is an intermediate latch used during PTC fills. See the CPU-02 Technical Reference Manual for more information.)

To execute subtest 9:

1. Type **:10,9**

2. Press **Return**.

   The following output appears on the screen:

   ```
   Data Cache and Page Table Cache Tests
   Pointer register tests
   ```

SUBTEST 10: INDIRECT POINTER REGISTER TESTS

This subtest tests the Pointer register by forcing PTC misses and ensuring that the translation was correct.

This subtest also requires an overlapped DMA to be active during the test. If an RS-422 Expansion board is installed it will be used as the DMA device. (Channels 0 and 1 must be connected with a RS-422 loopback cable. See the section on RS-422 Expansion board diagnostic tests for more information.) If an RS-422 Expansion board is not installed, the currently selected tape drive will be used as the DMA device. (Be sure that a tape is installed in the tape drive.)

To execute subtest 10:

1. Type **:10,10**

2. Press **Return**.

   The following output appears on the screen:

   ```
   Data Cache and Page Table Cache Tests
   Indirect pointer register tests
   Using xxx as the DMA device
   ```

   where xxx is "422" for the RS-422 Expansion board or "tape" for the currently selected btape drive.

SUBTEST 11: DATA CACHE MEMORY TEST THRU VIRTUAL SPACE: 0x100000 - 0x120000

Subtest 11 runs the same tests as subtest 1. Instead of testing the data cache directly (through the 0x48000000 addresses), however, it is tested indirectly through virtual addresses 0x100000 through 0x120000 with a unity map.

To execute subtest 11:

1. Type **:10,11**

2. Press **Return**.

   The following output appears on the screen:

   ```
   Data Cache and Page Table Cache Tests
   Data cache memory test thru virtual space: 0x100000-
   0x120000
   Enter delay count: (hex) (default: 0x1000000)
   ```

TEST 12: PAGE PROTECTION TEST (e)

Both the software and hardware in the S/MT Series can affect page protection. The software can protect the Virtual Memory region and prevent users from accessing most of the VME region in memory.

The hardware can protect users from accessing the regions in memory (except the VME region) located above 2G bytes. The two page protection subtests are presented in Table 8.

Table 8

PAGE PROTECTION SUBTESTS

Subtest	Description
1	Checks all combinations of hardware page protection status, User and Supervisor, and read, write, and execute modes.
2	In User mode, reads and writes to each page in the Virtual Memory, and each megabyte in I/O space.

To execute the two page protection subtests:

1. Type **e**
2. Press **Return**.

   This output is displayed on the terminal screen to indicate execution of the page protection test sequence:

```
PAGE PROTECTION TEST
PAGE PROTECTION SUBTEST 1
PAGE PROTECTION SUBTEST 2
Testing User Mode Access to 0x40000000 thru 0xFFF00000
Testing User Mode Access to 0x80000 thru 0x3FFFF000
```

SUBTEST 1: CHECKS ALL COMBINATIONS OF HARDWARE PAGE PROTECTION STATUS

This subtest selects a single page in Virtual Memory. It reads, writes, and executes each type of page protection in both User and Supervisor modes. The subtest is looking for expected results. That is, it expects to generate a page fault error when it tests an invalid page.

Pages are protected in one of four ways. The four page protection techniques are further determined by User or Suptervisor mode, as follows:

- not valid
- kernel read/write
- user read
- user read/write

To execute subtest 1:

1. Type **:12,1**
2. Press **Return**.

   The diagnostic displays this output upon successful completion:

   ```
   PAGE PROTECTION TEST
   PAGE PROTECTION SUBTEST 1
   ```

Subtest 1 can generate these error messages:

No page fault received on write

No page fault received on read

No page fault received on execute

Page fault when none was expected at address xx, with these protection bits, and in user or supervisor mode

SUBTEST 2: READS AND WRITES TO EACH PAGE IN THE VIRTUAL MEMORY AND TO EACH MEGABYTE IN UNUSED I/O SPACE

This subtest reads and writes data, in User mode, to every page in the virtual memory region. The subtest is succussful when page faults are generated.

Finally, this subtest tests the I/O space by accessing all five I/O regions, in user read/write mode and in increments of 1M byte. User access to all VME regions is denied during this part of subtest 2.

To execute subtest 2:

1. Type **:12,2**

2. Press **Return.**

Failure of subtest 2 can generate one of these error messages:

User mode read(or write) @ address X succeeded!

Unexpected bus fault on user write to address X.

No protection fault on user write to address X.

TEST 13: LINE PRINTER TESTS (1)

The line printer test sequence examines the parallel printer and its interface for the printer connectors on the Main Processor (CPU) board, the IOP Expansion board, and the RS-422 Expansion board. This test sequence consists of the six subtests listed in Table 9.

Table 9

LINE PRINTER SUBTESTS

Subtest	Description
1	Check and display the line printer status register
2	Print a rotating pattern on the printer using interrupts
3	Check the control lines of the printer connector with a loopback plug
4	Same as subtest 1 for the line printer connector on IOP or RS-422 Expansion board
5	Same as subtest 2 for the line printer connector on IOP or RS-422 Expansion board
6	Same as subtest 3 (requires a loopback plug) for the line printer connector on IOP or RS-422 Expansion board
7	Bidirectional Printer Port Test - Master Mode
8	Bidirectional Printer Port Test - Slave Mode (start this side first)

To execute the test sequence:

1. Type **l** (el)

2. Press **Return.**

The line printer test sequence executes the appropriate subtest (2 or 3) depending upon whether a printer or loopback plug is detected. If neither a printer nor a loopback plug are detected, a warning message is displayed and the test sequence continues. Pin assignments for the line printer loopback plug are contained in Appendix A.

When either the IOP Expansion board or the RS-422 Expansion board is present, the default test sequence executes the same subtests on the line printer connector present on the expansion board.

SUBTEST 1: CHECK AND DISPLAY THE LINE PRINTER STATUS REGISTER

Subtest 1 checks the status of the data/status register, returns the status, and makes sure that the status is valid (both line printer present and line printer selected, or line printer out of paper).

To execute subtest 1:

1. Type **:13,1**

2. Press **Return.**

   Output like this is displayed as the subtest is executed:

```
LINE PRINTER TEST
LINE PRINTER SUBTEST 1, (onboard) Status test
```

SUBTEST 2: PRINTS A ROTATING PATTERN ON THE PRINTER USING INTERRUPTS, THEN CHECKS THE STATUS OF EACH INTERRUPT

For each repetition, subtest 2 prints a rotating pattern to the printer, which consists of 80-column rotations of the ASCII pattern set.

To execute subtest 2:

1. Type **:13,2**

2. Press **Return**.

   Output like this is displayed as the subtest is executed:

```
LINE PRINTER TEST
LINE PRINTER SUBTEST 2, (onboard) Transfer test
```

SUBTEST 3: CHECK THE CONTROL LINES OF THE PRINTER CONNECTOR WITH A LOOPBACK PLUG

To execute subtest 3:

1. Type **:13,3**

2. Press **Return**.

   Output like this is displayed as the subtest is executed:

```
LINE PRINTER TEST
LINE PRINTER SUBTEST 3, (onboard) Loopback test
```

SUBTEST 4: CHECKS AND DISPLAYS THE LINE PRINTER STATUS REGISTER ON THE IOP OR RS-422 EXPANSION BOARD

To execute subtest 4:

1. Type **:13,4**

2. Press **Return**.

   Output like this is displayed as the subtest is executed. The asterisk refers to either the IOP Expansion board (IOP) or the RS-422 Expansion board (422).

```
LINE PRINTER TEST
LINE PRINTER SUBTEST 4, (*) Status test
```

SUBTEST 5: CHECKS INTERRUPT STATUS

To execute subtest 5:

1. Type **:13,5**

2. Press **Return**.

   Output like this is displayed as the subtest is executed. The asterisk refers to either the IOP Expansion board (IOP) or the RS-422 Expansion board (422).

```
LINE PRINTER TEST
LINE PRINTER SUBTEST 5, (*) Transfer test
```

SUBTEST 6: WITH A LOOPBACK PLUG, CHECKS THE CONTROL LINES OF THE PRINTER CONNECTOR ON THE IOP OR RS-422 EXPANSION BOARD

To execute subtest 6:

1. Type **:13,6**

2. Press **Return**.

   Output like this is displayed as the subtest is executed. The asterisk refers to either the IOP Expansion board (IOP) or the RS-422 Expansion board (422).

```
LINE PRINTER TEST
LINE PRINTER SUBTEST 6, (*) Loopback test
```

SUBTESTS 7 AND 8: BIDIRECTIONAL PRINTER PORT TEST

This test requires two systems to be connected via the bidirectional line printer port using a test cable (see Appendix A for cable specifications). One system should be designated as the master, the other as the slave.

Begin with the slave system.

1. Type **:13,8**

2. Press **Return**.

Then start the master system.

1. Type **:13,7**

2. Press **Return**.

The two systems will proceed to talk to each other. First the master will transmit a byte of data and the slave will read it. Then the slave will transmit the one's complement of the data back to the master. This will continue through a walking ls pattern for the data.

Disk and Tape Diagnostic Tests

This chapter describes the hard disk drive and tape drive tests listed in Table 1.

Table 1

STORAGE MEDIA DIAGNOSTIC TESTS

Number	Letter	Description
6	w	Hard Disk Drive
9	u	General Disk Test
16	t	QIC Tape Drive (CPU-01 systems only)
20	j	General Tape Test

CAUTION: Several of the disk drive subtests are destructive, which means that information contained on the disk is destroyed when the respective subtests are executed. Therefore, if important information is stored on the disk, be extremely careful when you execute the disk drive tests. Refer to the MightyFrame Series Administrator's Reference Manual for information on backing up the contents ofa disk drive before you execute a disk test.

Likewise, be careful when you execute the tape tests because they too can destroy information contained on the tape. Usea blank tape when you execute any of the tape test sequences.

DISK DRIVE TESTS (w and u)

The hard disk drive subtests (Test 6), listed in Table 2, test every aspect of the currently selected disk drive. (The disk drive is selected with the DN command. See Chapter 4 for more information on this command.)

The general disk test (test 9) contains all of the subtests found in test 6. The difference between the two is that subtest 9 operates on all drives present (not just the drive selected using DN).

Table 2

HARD DISK DRIVE SUBTESTS (TESTS 6 AND 9)

Subtest	Description
1*	Recalibrate the disk
2	Format the disk
3*	Write test
4	Read test
5*	Random seeks
6	Write multiple blocks
7	Read multiple blocks
8	Spare a sector
9*	Force DMA faults
10	Initialize VHB and BBT
11	Writes VHB and BBT
12	Print VHB and BBT
14	Toggle alien disk mode
15*	Disk DMA register check
20	Print UNIX file system
21	Find unreadable blocks
23	Surface test
25	Sorts BBT
29	Harmonic seek stress test
30	Clone disk

* Test sequences w (for Test 6) and u (for Test 9) run these subtests.

NOTE: The test output for the disk drive subtests are examples only, and do not imply that the default settings will always be the same. In addition, the sample test output contains references to the disk drive number D0, which, in the following examples, contains an Atasi 56M byte drive formatted for Error Checking and Correction (ECC).

BAD BLOCK INFORMATION

Each hard disk drive is shipped with documentation on bad block information. You may need this information while running diagnostic tests. This information is located in different places for each system.

System	Bad Block Information Location
S/120	in bags on the drive tower behind the left side panel
S/220	in a bag attached to each drive
S/221	in bags on the drive tower behind the left side panel
S/222	in bags on the drive tower behind the left side panel
S/320	in bags on the inner surface of the front and rear doors
S/640	in bags on the inner surface of the front and rear doors

DISK DRIVE CONTROLLER REGISTERS

Hard disk drives (ST-506) and SMD drives each contain status and error registers to help you decode disk drive error messages. Each register contains bits whose numbers are reported in several of the disk drive error messages. These bit numbers report specific problems.

Hard Disk (ST-506) Drives

For ST-506 hard disk drives, the error messages include a "Response = #" phrase in which # is a 2 byte hexadecimal number. For the onboard disk (WD2010), the first byte concerns the Status register and the second byte concerns the Error register. Response codes 0x1 through 0x5 are special response codes that are decoded as follows:

- 0x1 Disk was recalibrated but failed to read the Volume Home Block (VHB).
- 0x2 Disk operation completed but timed out waiting for the disk interrupt
- 0x3 Disk operation timed out (was not completed)
- 0x4 Drive failed to recalibrate.
- 0x5 Disk was recalibrated and read the VHB; however, either the VHB or Bad Block Table (BBT) was invalid (bad checksum)

The Status register informs the host of certain events performed by the controller and reports the status of the drive control lines. The Status register contains the 8 bits listed in Table 3.

Table 3

STATUS REGISTER BITS

Bit Number	Code	Purpose
7	BUSY	Busy
6	READY	Ready
5	WF	Write Fault
4	SC	Seek Complete
3	DRQ	Data Request
2	DCORR	Data Was Corrected
1	CIP	Command in Progress
0	ERR	Error

The Error register contains the specific error status codes that pertain to the completion of a command. The bits are described in Table 4 below.

Table 4

ERROR REGISTER BITS

Bit Number	Code	Purpose
7	BAD-BLK	Bad Block Detect
6	DATA-FIELD-ERR	Data Field Error
4	IDNF	ID Not Found
2	CMD-ABRT	Aborted Command
1	TRKO-ERR	TK000 Error
0	ADDR-MRK-NF	Address Not Found

See the Western Digital WD2010-05 Winchester Disk Controller specification for detailed information on these register bits.

SMD Drives

See the S/MT Series VME SMD Controller Card Manual for detailed information on SMD controller status and error registers.

SELECTING THE DISK DRIVE

An S/MT Series computer system can contain several disk drives. Disk drive number D0 is the drive initially selected by the hard disk drive tests.

To determine the currently selected disk drive:

1. Type **DN?** (or DN ?)
2. Press **Return**.

   A message like this is displayed:

   Disk Drive 0 is selected

You can use the two-letter **DN** command to change the disk drive number. For example, to select the disk drive number D2:

1. Type **DN2** (or DN 2)
2. Press **Return**.

   This message appears:

   Disk Drive 2 is selected

   You can also issue this two-letter command repeatedly to cycle through other possible disk drive numbers.

SPECIFYING A REPEAT COUNT

The number of times a disk drive I/O test is retried is preset for 10 transfers. To change the number of retries, refer to RC in Chapter 4, "Two-Letter Commands."

NOTE: The number of retries executed by the driver (if any) is reported at the end of the test.

In addition, the disk driver assumes that what is written to the disk drive by the subtests is correct, and therefore detects data errors on a read-only basis.

EXECUTING THE DISK DRIVE TEST SEQUENCE

The disk drive test sequence (w) consists of the following five subtests which are executed in the order shown below.

- Subtest 1: Recalibrate the disk
- Subtest 15: Disk DMA register integrity check
- Subtest 3: Write test
- Subtest 5: Random seeks
- Subtest 9: Force DMA faults

CAUTION: The diagnostic assumes that it has permission to write to the disk when you issue this command.

To execute the entire disk drive test sequence (see the following pages to execute one subtest at a time):

1. Type **w**
2. Press **Return**.

SUBTEST 1: RECALIBRATE THE DISK

This subtest initializes the disk, brings the head back sector 0 of track 0 (the beginning of the disk) and attempts to read and verify the first blocks on the disk. The first block is the Volume Home Block (VHB), which contains descriptive information such as the number of cylinders, the number of tracks per cylinder, and the number of sectors per track. The second block contains the Bad Block Table (BBT). Note that the BBT may be larger than one block. The VHB indicates the size and location of the BBT.

To execute subtest 1:

1. Type **:6,1**

2. Press **Return**.

   This output confirms test execution:

   ```
   DISK TEST Drive0 [56ATA]
   (Subtest 1) Recal Disk
   ```

   Subtest 1 starts out in Error Checking and Correction mode (ECC). Subtest 1 pulls the head back and reads block 0. If the read operation fails, subtest 1 then reads the same block in Cycle Redundancy Check (CRC) mode. An error message is displayed when subtest 1 fails in CRC mode.

SUBTEST 2: FORMAT DISK

This subtest consists of three steps. First it asks for the information required to create the Volume Home Block and format the disk. Next, it formats the disk. Finally, after the disk is formatted, it surface-tests the disk. The surface test searches for defects, hard errors, and soft errors. At the end of the surface test, new defects are incorporated into the Bad Block Table.

CAUTION: Since Format Disk destroys all information on the disk, you should save the contents of the disk. See the MightyFrame Administrator's Reference Manual to do so.

To format the selected disk:

1. Type **:6,2**
2. Press **Return**.

NOTE: If requested (as shown in the output on the next page), the diagnostic allocates space in slice 0 for a loader and download area. The diagnostic does not put an actual loader or download program on the disk. The CTIX command iv(l) performs this function. See the CTIX Operating System Manual for a description of the iv command.

The last prompt displayed by this subtest provides the opportunity to abort the test. It states:

Enter y to approve parameters, enter n to abort:

The subtest is executed if you type **y** (yes). It is not executed if you type **n** (no).

CAUTION: Be sure the drive you are changing VHB information of is physically installed in your system. For example, if you are changing VHB information for 62CP (a Hitachi drive) be sure the proper Hitachi drive is installed in your system.

The following output is displayed on the terminal screen:

```
        DISK TEST Drive0 [56ATA]
(Subtest 2) Format Disk
Do you want to format Drive0 - erasing contents (Y or N)?
The Predefined Volume Block formats are:
        53CP-MAXTOR
        56CP-ATASI
        62CP-HITACHI
        84CP-HITACHI
        85CP-MAXTOR
        86CP-MICROPOLIS
        87CP-MINISCRIBE
        143CP-MAXTOR
        190CP-MAXTOR
        300CP-Priam807
        301CP-NT8212
        302CP-DX332
        450CP-EAGLE
        600CP-EAGLEXP

Enter format type [?<type> to display VHB format] (press
return to specify your own format)
```

If you specify your own format, the following prompts will appear (if you instead select one of the above format types, these prompts will be skipped):

```
Give # of Cylinders (RETURN = default of 645):
Give # of Tracks per Cylinder (RETURN= default of 7):
Give # of Sectors per Track (RETURN= default of 17):
Cylinders = 645, Tracks = 7, Sectors = 17
Give Pack Name (RETURN = default of 56ATA):
```

The series of prompts continues with these:

```
Is the disk exchangeable (Y or N):
Does the drive use head select bit 3 (Y or N):
Does the drive use reduced write current (Y or N):
Give Step Rate (RETURN = default of 14):
Partition 0 Start Block (RETURN to end) 0-36120:
Give size of BBT in blocks (Default = 1):
Do you want a Loader (Y or N):
Do you want a Dump Area ?(Y or N):
Do you want a Down Load File ?(Y or N):
Give Format Mode 0=ECC 1=CRC (RETURN = Default of
ECC):
Enter y to approve vhb parameters, enter n to abort:
```

SUBTEST 3: WRITE TEST

This subtest ensures that the disk-related circuitry functions properly. It assumes that defects on the disk are known and recorded.

CAUTION: Since Write Disk destroys all information on the disk drive, you should save information already on the disk. Refer to the S/MT Administrator's Reference Manual to do so. In addition, should subtest 3 abort for any reason, immediately execute subtest 11 to rewrite the Volume Home Block and Bad Block Table.

Subtest 3 writes to one track at a time. Then it reads back the data it wrote to the track and verifies the pattern. After the entire disk is tested this way, a read-check operation is performed, which reads the entire disk again, track by track. Error information is displayed on the screen before the diagnostic stops.

Note that any disk errors that are found are not entered into the BBT. Run subtest 23 to enter the bad spots in the BBT.

Subtest 3 can be executed in one of these two modes, which are not related to the test execution modes described in Chapter 3.

- direct mode (as in :6,3) allows you to specify the data pattern, the start block, the number of blocks to transfer, the maximum transfer size, and the number of subtest repetitions.

- test sequence mode (executed within a test sequence, "w" for example) uses predefined settings to execute the test. The default pattern is the "stringent" pattern. This also applies to the user-defined test sequence (#=).

Subtest 3 writes over the entire disk, reading the data after it has been written (in increments of the selected transfer size). After completing the write phase, the data is read back and verified.

To write data to the selected disk:

1. Type **:6,3**

2. Press **Return**

   The following prompts are displayed:

```
        DISK TEST Drive0 [56ATA]
(Subtest 3) Disk Write and Check Read test
Do you want to write to Drive0 [56ATA]?:
Volume Name:56ATA (ecc mode)
Give Data Pattern in hex (RETURN= default of db6,
1=address, 2=cyclic):
Give start Logical Block (RETURN= default of 16):
Give # of Blocks to transfer (RETURN= default of 36104):
Give Max transfer size in Blocks (RETURN= default of 8):
Give # repeat times (RETURN = default of 1):
```

   Data is written to the disk when you respond to the last prompt. This output is displayed when you accept the default settings:

```
Rotating Pattern = 0xDB6, Start Block = 16, # Blocks =
36104,
Increment = 8, Repeats = 1
Use short word alignment?
Drive0 [56ATA]: Initiating Check Read for pass0
Read address is 0x71000, write address is 0x67000
```

SUBTEST 4: READ TEST

This subtest performs the read portion only of Subtest 3 and is meaningless without first executing subtest 3. Subtest 4 is not destructive to the disk. Subtest 4 reads the disk to check for either the default pattern or a hex pattern that you specify. Subtest 4 asks all the same questions that Subtest 3 asks.

To read data already written on the disk by subtest 3:

1. Type **:6,4**

2. Press **Return**.

   These prompts are displayed:

```
        DISK TEST Drive0
(Subtest 4) Disk Read test.
Volume Name: 56ATA (ecc mode)
Give Data Pattern in hex (RETURN= default of db6):
Give start Logical Block (RETURN= default of 16):
Give # of Blocks to transfer (RETURN= default of 36104):
Give Max transfer size in Blocks (RETURN= default of 8:
Give # of Times to Repeat  (RETURN = default of 1):
```

   Data is read from the selected disk when you respond to the last prompt. This output is displayed if you accept the preset settings:

```
Rotating Pattern = 0xDB6, Start Block = 16 # Blocks =
36104
Increment = 8, Repeats = 1
```

SUBTEST 5: RANDOM SEEKS

This test performs random seeks to different sectors on the disk. It reads the sectors, checking to make sure that the seek works consistently and that the head goes to the correct location.

To perform random seeks on the selected disk:

1. Type **:6,5**

2. Press **Return.**

   Subtest 5 allows you to specify the maximum block number and the number of repetitions (random seeks). To do so, you are prompted with these questions:

```
DISK TEST Drive0 [56ATA]
(Subtest 5) Random Seek test.
Volume Name: 56ATA (ecc mode)
Give Maximum Block # (RETURN = default of 36120):
Give # of Times to Repeat (RETURN = default of 1000):
```

   The subtest is executed when you respond to the last prompt. If you accept the default settings, then this output is displayed while the subtest is executed:

```
Maximum Block = 36120, Repeat count = 1000
```

SUBTESTS 6-7: WRITE AND READ MULTIPLE BLOCKS

Subtests 6 and 7 write and read, respectively, multiple sectors to the disk from memory. These subtests allow you to specify the area in memory to write to the disk or, conversely, to read from a location on the disk. The parameters you enter are the starting address in memory, the starting block on disk, and the number of blocks to transfer.

CAUTION: The diagnostic program will be overwritten if you specify the area in memory where the diagnostic program is stored. The default address is the lowest address available. If you do overwrite the diagnostic tests, you must reload them from the QIC tape.

To write multiple sectors to the selected disk:

1. Type **:6,6**

2. Press **Return**.

   These prompts are displayed:

```
        DISK TEST Drive0 [56ATA]
(Subtest 6) Disk Write function.
Do you want to write to Drive0 [56ATA]:
Volume Name: 56ATA (ecc mode)
Give Buffer Address in hex (RETURN= default of 0x60000):
Give start Logical Block (RETURN= default of 2):
Give # of Blocks to transfer (RETURN= default of 36118:
```

   Data is written to mutliple sectors when you respond to the last prompt. If you accept the default settings, then this output is displayed:

```
Buffer Address = 0x60000, Start Block = 2, # Blocks =
36118
```

To read multiple sectors from the selected disk:

1. Type **:6,7**

2. Press **Return**.

These prompts are displayed:

```
DISK TEST DRIVE0 [56ATA]
(Subtest 7) Disk Read function.
Volume Name: 56ATA (ecc mode)
Give Buffer Address in hex (RETURN= default of 0x60000):
Give start Logical Block (RETURN= default of 2):
Give # of Blocks to transfer (RETURN= default of 36118):
```

Multiple sectors are read from the disk when you respond to the last prompt. If you accept the default settings, then this output appears:

```
Buffer Address = 0x60000, Start Block = 2, # Blocks =
36118
```

SUBTEST 8: SPARE A SECTOR

This subtest lets you specify bad blocks in the Bad Block Table. Subtest 8 asks you to specify either the track number (for example, cylinder number and head number), byte number, physical sector, or logical block of the bad block.

To specify a bad block:

1. Type **:6,8**

2. Press **Return**.

   These prompts are displayed:

```
        DISK TEST Drive0 [56ATA]
(Subtest 8) Spare a sector function.
Volume Name: 56ATA (ecc mode)
Specify Spare: [1:RETURN] = (Cylinder, Head and Byte)
                    [2] = Physical Sector
                    [3] = Logical Block
                    [4] = Cylinder, Head and
                          Physical Sector on Track:
```

   If you press Return (Cylinder, Head, and Byte), the following prompts will appear:

```
Give Cylinder Number:
Give Head Number:
Give Byte Number:
```

   If you select 2 (Physical Sector), the following prompt appears:

```
Give Physical Sector:
```

   If you select 3 (Logical Block), the following prompt appears:

```
Give Logical Block:
```

If you select 4 (Cylinder, Head, and Physical sector on Track), the following prompts will appear:

```
Give Cylinder Number:
Give Head Number:
Give Physical Sector on Track:
```

The bad block is entered into the BBT when you respond to the last prompt, and then the BBT is rewritten to the disk. This sample output is displayed while subtest 8 is executed:

```
Cylinder 4, Track 4, Sector 0.
Used Track 32 as alternate.
The BBT contains 2 entries.
Another ? (Y or N)
```

Note that the first two lines of the above prompt will not appear if the sector is already spared.

SUBTEST 9: FORCE DMA FAULTS

This subtest forces parity errors and page faults during a disk transfer. It verifies that the appropriate nonmaskable interrupt (NMI) occurs and that the GSR is functioning correctly. This subtest also tests the DMA address register.

To force DMA faults:

1. Type **:6,9**
2. Press **Return**.

   This following output appears on the screen:

```
DISK TEST Drive0 [56ATA]
(Subtest 9) Force DMA Faults.
```

SUBTEST 10: INITIALIZE VHB AND BBT

This subtest allows you to create the Volume Home Block and the Bad Block Table. This subtest is the same as subtest 2 except that it does not format the disk or perform any surface tests.

Unless the partitions are changed, data previously on the disk is saved. If the partitions are changed, then data on the disk is virtually lost. You can recover the data by restoring the partitions to their original state.

To initialize the Volume Home Block and Bad Block Table:

1. Type **:6,10**

2. Press **Return**.

   The output displayed on the following page is displayed:

```
        DISK TEST Drive0 [56ATA]
(Subtest 10)  Initialize VHB and BBT.
Do you want to change the VHB on Drive0 (Y or N)?
The Predefined VHB formats are:
        53CP-MAXTOR
        56CP-ATASI
        62CP-HITACHI
        84CP-HITACHI
        85CP-MAXTOR
        86CP-MICROPOLIS
        87CP-MINISCRIBE
        143CP-MAXTOR
        190CP-MAXTOR
        300CP-Priam807
        301CP-NT8212
        302CP-DX332
        450CP-EAGLE
        600CP-EAGLEXP
Enter format [?<type> to display VHB format]  (press return to
specify your own format)
```

If you specify your own format, the following prompts will appear (if you instead select one of the above format types, these prompts will be skipped):

```
Give # of Cylinders (RETURN = default of 645):
Give # of Tracks per Cylinder (RETURN= default of 7):
Give # of Sectors per Track (RETURN= default of 17):
Cylinders = 645, Tracks = 7, Sectors = 17
Give Pack Name (RETURN = default of 56ATA):
```

The series of prompts continues with these:

```
Is the disk exchangeable (Y or N):
Does the drive use head select bit3 (Y or N):
Does the drive use reduced write current (Y or N):
Give Step Rate (RETURN = default of 14):
Partition 0 Start Block (RETURN to end) 0-36120:
Give size of BBT in blocks (Default= 1):
Do you want a Loader (Y or N):
Do you want a Dump Area (Y or N):
Do you want a Down Load File (Y or N):
Enter y to approve vhb parameters, entern to abort:
```

The last prompt provides the opportunity to abort the test. If you do not abort, the VHB and BBT are created if they do not already exist. If they already exist, the BBT is rewritten to the disk (unaltered) according to the VHB.

SUBTEST 11: WRITE VHB AND BBT

This subtest restores the Volume Home Block and Bad Block Table when they are lost as a result of executing subtests 3 or 23.

Subtest 3 normally saves the Volume Home Block and Bad Block Table in an internal buffer. However, in cases where subtests 3 or 23 inadvertently write over the Volume Home Block and Bad Block Table, you can execute subtest 11 to rewrite the Volume Home Block and the Bad Block Table.

To restore the Volume Home Block and Bad Block Table:

1. Type **:6,11**

2. Press **Return**.

   This output confirms successful test completion:

   ```
   DISK TEST Drive0 [56ATA]
   (Subtest 11) Write VHB and BBT after aborted Test 3.
   Do you want to write to Drive0 [56ATA]?
   ```

   The subtest is executed when you respond to the second prompt.

SUBTEST 12: PRINT VHB AND BBT

You can display the VHB and the BBT to the screen with this subtest. This information can also be printed at the optionally attached line printer. (See Chapter 2, "Getting Started," to connect the printer and Chapter 4, "Two-Letter Commands," to make it operable.)

To print the VHB and BBT:

1. Type **:6,12**

2. Press **Return**.

   Output such as that displayed below is displayed on the screen (and printed at the line printer if it is attached):

```
        DISK TEST Drive0 [56ATA]
(Subtest 12) Print VHB.
Volume Name: 56ATA
645 Cylinders. 7 Heads per Cylinder.
There are 17 Physical Sectors (of 512 bytes) per Track.
119 Physical Sectors per Cylinder, 76755 Physical Sectors
per Disk.
There are 8 Logical Blocks (of 1024 bytes) per Track.
56 Logical Blocks per Cylinder, 36120 Logical Blocks per
Disk.
The Step Rate supplied to the Controller is 14.
Partition 0: start Block=0, size (in Blocks)=36120
BBT starts at block 1 (size=1 Blocks).
The BBT contains 0 entries.
```

SUBTEST 14: TOGGLE ALIEN DISK MODE

This subtest toggles the Alien mode flag for the disk type specified. When the Alien mode flag is set, subtests 1, 6, 7, and 21 interpret the disk as a nonstandard format disk (for example, no Volume Home Block and different bad block handling).

No other subtests are valid on a disk that is in Alien mode. The Alien mode flags for the various disks are separate and can be set and cleared independently. Subtest 14 asks the same questions asked by subtest 2 so that the disk driver can drive the disk.

To enable Alien disk mode:

1. Type **:6,14**

2. Press **Return**.

   These prompts are displayed:

```
DISK TEST Drive0 [56ATA]
(Subtest 14) Toggle Alien Disk mode.
Enter the format for the alien disk[?<type> to display VHB
format] (press return to specify your own format):
```

   If you specify your own format, the following prompts will appear (if you instead select one of the above format types, these prompts will be skipped):

```
Give # of Cylinders (RETURN = default of 645):
Give # of Tracks per Cylinder (RETURN= default of 7):
Give # of Sectors per Track (RETURN= default of 17):
Cylinders = 645, Tracks = 7, Sectors = 17
Give Pack Name (RETURN= default of 56ATA):
```

The series of prompts continues with these:

```
Is the disk exchangeable (Y or N):
Does the drive use head select bit3 (Y or N):
Does the drive need reduced write current (Y or N):
Give Step Rate (RETURN = default of 14):
```

This statement is displayed when you respond correctly to each prompt:

```
Drive0 [56ATA]: Alien disk mode enabled
```

To disable Alien disk mode:

1. Type **:6,14**
2. Press **Return.**

This output displayed. It confirms that Alien disk mode is disabled:

```
DISK TEST Drive0 [56ATA]
(Subtest 14) Toggle Alien Disk mode.
Drive0 [56ATA]: Alien disk mode disabled
```

SUBTEST 20: PRINT CTIX FILE SYSTEM

This subtest prints the CTIX file system. The output of this command is lengthy. This subtest interprets a given partition of the disk as a CTIX file system and displays a representation of it on the screen.

To print the CTIX file system:

1. Type **:6,20**
2. Press **Return.**

The following prompt appears:

```
Enter partition number to describe (Default = 1)
```

After entering a partition number and pressing **Return**, output similar to the sample below will appear:

```
#inodes=267 fs size=17000
blk=2 (42) inode=1(0) mode=0100000 size=0
blk=2 (42) inode=2(1) mode=040755 size=496 271
inode2.
inode2..
inode3 usr
inode4 bin
```

SUBTEST 21: FIND UNREADABLE BLOCKS

This subtest prompts you to enter a start block number and the number of blocks you want searched. Then it searches the disk for unreadable blocks. Any blocks that cannot be read are displayed along with the failing status. This portion of the disk test is read only. No change is made to the Bad Block Table.

Note that this subtest does not stop when it encounters an error.

To find unreadable blocks:

1. Type **:6,21**

2. Press **Return**.

   These prompts are displayed:

```
DISK TEST Drive0 [56ATA]
Volume Name:
Enter start block (Default= 0):
Enter # of blocks (Default = 36120):
Enter repeat count (Default= 1):
```

SUBTEST 23: SURFACE TEST

Subtest 23 surface tests the disk. It assumes that both the disk-related circuitry and the disk drive mechanics work. Rotating db6 patterns, which may be altered, are written onto the disk one track at a time. The track is then read and the data verified. When all tracks have been tested, the disk goes through a read-check pass. If any errors are encountered, the offending track is then throughly tested sector by sector.

The errors can be a defect, a hard error, or a soft error. The Bad Block Table on the disk is not used to avoid writing to blocks marked as bad. Defects may be found that already exist in the media Bad Block Table, in which case they are printed for the operator. Errors determined to be media defects are added to the Bad Block Table.

To surface test the disk:

1. Type **:6,23**

2. Press **Return.**

   These prompts are displayed:

```
DISK TEST Drive0 [56ATA]
(Subtest 23) Surface test
Do you want to write to Drive0 (56ATA)?
Volume Name:
Give Data Pattern in hex (RETURN= default of DB6 :
Give start Physical Sector (RETURN= default of 51) :
Give # of Sectors to transfer (RETURN= default of 76704) :
Give # Retries required per Sector (RETURN= Default of 10):
Give # of Times to Repeat (RETURN = Default of 1):
```

   If you accept the settings prescribed by the diagnostic, then this output is displayed:

```
Data Pattern = 0XDB6, Start Sector = 51, # Sector = 76704,
Increment = 17, Retries = 10 Repeats = 1
```

SUBTEST 25: SORT BAD BLOCK TABLE

This subtest sorts the BBT numerically by cylinder, then it sorts track and physical sector numerically within the cylinder.

To sort the BBT:

1. Type **:6,25**

2. Press **Return**.

   These prompts are displayed:

   ```
   DISK TEST Drive0
   (Subtest 25) Sort Bad block table.
   Do you want to sort the BBT on Drive0?
   Clear data block buffer on read error? (default= yes)
   Volume Name:
   Saving alternate sector data.
   ```

   If you accept the default settings, then this sample output is displayed while the subtest is executed:

   ```
   The BBT contains 0 entries.
   Restoring alternate sector data.
   ```

   If your drive contains bad blocks, the following message will appear:

   ```
   Added Bad Block: Cylinder XXX, Head X, Sector XX
   Used track XXXX as the alternate
   ```

   The subtest will prompt you for permission to continue on error.

SUBTEST 29: HARMONIC SEEK STRESS TEST

This subtest consists of four activities. First it reads the Volume Home Block. After successfully reading the Volume Home Block on the disk, it performs fixed length seeks and reads from the low disk cylinders to the high disk cylinders.

Next, it executes seeks and reads from the middle cylinders to the outer cylinders. Finally, it performs seeks and reads, increasing in length from 0 cylinders to 1-N cylinders, and then seeks decreasing in length from 1-N cylinders back to 0 cylinders.

To execute the harmonic seek stress subtest:

1. Type **:6,29**

2. Press **Return**.

   This output is displayed on the screen to indicate test execution:

```
DISK TEST Drive0
(Subtest 29) Harmonic Seek stress test.
Volume Name: 56ATA
Seeks: Low to High
Seeks: 0<-Middle->80
Seeks: increasing to 80 then decreasing seeks to0
```

SUBTEST 30: CLONE A DISK

This subtest copies a disk. Both disks must be the same size and both must be connected to the same system. The speed at which this subtest copies the disk is about 5M bytes per minute. The drive number displayed by DN is the source drive, or the drive to be copied from.

CAUTION: Do not copy VHBs between different types of disk drives. Never copy the BBT to another drive.

To clone a disk:

1. Type **:6,30**

2. Press **Return**.

   These prompts are displayed:

   ```
   (Subtest 30) Clone Disk 0.
   Volume Name: 56ATA (ecc mode)
   Copy to disk (RETURN = default of 1:
   Copy to Volume Name: winch
   Give start Logical Block (RETURN= default of 2):
   Give # of Blocks to transfer (RETURN= default of 36118):
   ```

   If you accept the default settings, then this output is displayed. Respond to the prompt:

   ```
   Clone from disk 0 (56ATA) to disk 1 (Winchester)
   Start Block = 2, # Blocks = 36118 (y to confirm):
   ```

   This output confirms test execution:

   ```
   Phase 1: copying disk
   Phase 2: comparing disks
   ```

TEST 16: QUARTER-INCH CARTRIDGE TAPE TEST (t)

Test 16 tests the quarter-inch cartridge (QIC) tape interface and drive mechanism in CPU-01 systems only. It consists of the 10 subtests shown in Table 5. Subtests 4 and 5 consist of numerous options that are detailed in this chapter.

CAUTION: Several of these tests write data to the tape. Therefore, make sure that you use a nonwrite-protected tape that is either blank or contains unwanted information.

The QIC tape test sequence (t) executes subtests 1, 2, 3, 6, and 7 only. To execute these subtests:

1. Type **t**
2. Press **Return**.

NOTE: The output used in this section to illustrate responses to the tape subtest prompts are examples only and do not reflect limitations or default settings.

Table 5

QIC TAPE SUBTESTS

Subtest	Description
1	Reset and initialize tape
2	Perform adverse DMA tests
3	Rewind tape
4	Run interactive diagnostics 0 run utilities 1 rewind tape 2 issue write command 3 issue read command 4 write blocks 5 read blocks 6 display control lines 7 issue write filenames 8 issue read filenames a reset controller b issue retension command c issue read status command d dump blocks with filemark e read blocks up to filemark f display pages of memory g display previous tape status h adverse DMA test
5	Run interactive utilities 0 run diagnostics 1 rewind tape 3 copy disk partition to tape 4 restore disk partition from tape 5 run tape write/read test 6 run write phase only 7 run read phase only 8 copy disk blocks to tape 9 restore disk blocks from tape a reset tape b retension tape c read tape status d set tape data format e erase test
6	Execute write phase of tape test
7	Execute read phase of tape test
8	Set up parameters of tape test
9	Erase tape test
10	Retension tape test

This output is generated when the tests are successfully executed:

```
            Qici - controller reset
            Qici - adverse DMA tests
            Qici - controller reset
PASSED - tape bad parity detection
PASSED - tape DMA to non-valid page test
            Rewinding the tape ... Qici- controller reset
Rewind okay
PASSED - tape DMA address counter test
            Rewinding the tape ... Qici- controller reset
Rewind okay
            Qici Write Test
            Qici - controller reset
Controller reset okay
            Rewinding the tape ... Qici- controller reset
Rewind okay
Writing test pattern
Transfer complete, 40000 blocks w/ filemark
            Rewinding the tape ... Qici- controller reset
Rewind okay
Controller PASSED write test
            Qici Read Test
            Qici - controller reset
Controller reset okay
            Rewinding the tape ... Qici- controller reset
Rewind okay
Verifying Write Test Data on Tape ...
            Rewinding the tape ... Qici- controller reset
Rewind okay
Tape Passed read test
```

SUBTEST 1: RESET AND INITIALIZE TAPE

This subtest performs three steps. First, it executes a data path check to the tape DMA byte Count register. Second, it resets the tape controller. Third, it reads the tape status.

1. Type **:16,1**

2. Press **Return**.

This message appears on the terminal screen upon successful completion of this subtest:

```
Qici - controller reset
```

SUBTEST 2: ADVERSE DMA TESTS

This subtest executes the following three errors:

Parity Error:

An even (wrong) parity word is written to memory. The subtest sets up DMA to read the word from the memory (with the parity error) and write it to the disk. The subtest reports that the tape controller subsystem found the parity error and that the status bit in the Tape Condition register detected the parity error. Correct parity is finally restored to the even parity word.

Page Protection Error:

The subtest sets up DMA to read date from an invalid page. The subtest then checks that a page fault error occurred and that the proper bits were set in the GSR.

Address Error:

The subtest tests all bits in the DMA Address register. To do so, a block of data is set up in memory and then written to tape. Data on the tape is read back into different virtual memory addresses to check that the address map functions properly.

To execute subtest 2:

1. Type **:16,2**
2. Press **Return**.

This output is displayed on the screen:

```
          Qici - adverse DMA tests
          Qici - controller reset
PASSED - tape bad parity detection
PASSED - tape DMA to non-valid page test
          Rewinding the tape ... Qici - controller reset
Rewind okay
PASSED - tape DMA address counter test
```

SUBTEST 3: REWIND TAPE

This subtest rewinds the tape currently in the tape drive. To rewind the tape:

1. Type **:16,3**
2. Press **Return.**

This output is displayed:

```
          Rewinding the tape ... Qici - controller reset
Rewind okay
```

SUBTEST 4: RUN INTERACTIVE DIAGNOSTICS

This subtest contains several options, each of which you can execute separately.

CAUTION: Before proceeding, note that these options provide a great deal of control over the tape drive. Refer to the IEEE Tape Interface Specification for more detailed information on how to use them most effectively.

To display the QIC tape diagnostic options:

1. Type **:16,4**
2. Press **Return.**

This output is displayed:

```
qici diagnostics
qici - diagnostic menu:
   (0)   run utilities
   (1)   rewind tape
   (2)   issue write command
   (3)   issue read command
   (4)   write blocks
   (5)   read blocks
   (6)   display control lines
   (7)   issue write filemark
   (8)   issue read filemark
   (a)   reset controller
   (b)   issue retension command
   (c)   issue read status command
   (d)   dump blocks with filemark
   (e)   read blocks up to filemark
   (f)   display page(s) of memory
   (g)   display previous tape status
   (h)   adverse DMA test (writes to tape)
Q(diag)$
```

In addition, the prompt changes to Q(diag)$ to indicate that you are in the diagnostic menu of the QIC tape test.

To execute any of the tape diagnostic options, you must type its number next to the new prompt (Q(diag)$). For example, to execute option 1:

1. Type **1**
2. Press **Return**.

To exit the diagnostics menu of the QIC tape test:

1. Type **q**
2. Press **Return**.

(0): Run utilities

This option transfers you to another submenu called Utilities which is discussed under Subtest 5.

(1): Rewind Tape

This option rewinds the tape currently in the tape drive.

(2): Issue a Write Command

This option prepares the tape drive for the Write Blocks command (option 4).

(3): Issue a Read Command

This option prepares the tape drive for the Read Blocks command (option 5).

(4): Write Blocks

This option writes unspecified blocks of data from memory to tape. It provides the following output:

```
Qicl write block @ memory address:0x3E6C8

How many 512 byte blocks? 0-100000:
```

(5): Read blocks

This option reads the blocks of data sent to tape from memory in the above option. This option verifies that the transfer was done properly. It provides the following prompt:

```
Qicl read block @ memory address:0x3E6C8
How many 512 byte blocks? 0-100000:
```

(6): Display Control Lines

This option displays the control lines. It presents this output:

```
signals 0xE: not ready, no data error, exception, not ack
```

(7): Issue Write Filemark

This option is executed after option 4 to place a filemark on the tape to indicate the boundary of the blocks written from memory to tape.

(8): Issue Read Filemark

This option can be executed after option 3 to read back the data on the tape in the region indicated by the filemark issued by option 7.

(a): Reset Controller

This option resets the tape controller. It presents this message to confirm its completion:

```
Resetting controller ...
```

(b): Issue Retension Command

This option rewinds the tape, resets the controller, and retensions the tape. It displays this output:

```
     Retension tape test
     Rewinding the tape ... Qici- controller reset
Rewind okay
PASSED Retension test
```

(c): Issue Read Tape Status Command

This option reports the tape status, usually in a format like this:

```
tape status:0x81 Reset Occurred
```

In the above message, here and in the Utilities, the hexadecimal number (0x81) is decoded as per the words that appear after the number (Reset Occurred).

(d): Dump Blocks with Filemark

This option provides a convenient mechanism for options 2, 4, and 7. It writes unspecified blocks of data to the tape, automatically handling the issue write and write filemark commands. This option provides this feedback:

```
How many 512 byte blocks? 0-100000 :
Transfer complete, X blocks with filemark
```

(e): Read Blocks Up to Filemark

This option provides a convenient mechanism for options 2, 4, and 7. It reads unspecified blocks of data from the tape, automatically handling the issue read and read filemark commands.

(f): Display Memory

This option displays the data in memory that is written to tape by option 4. Feedback such as this appears on the terminal screen when you execute this option:

```
qicibfr=0x3E6C8
Input display address [default=0x3E6C8]
0x03E6C8: 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0
0x03E6D0: 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0
0x03E6D8: 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0
0x03E6E0: 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0
0x03E6E8: 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0
0x03E6F0: 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0
0x03E6F8: 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0
0x03E700: 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0
0x03E708: 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0
0x03E710: 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0
0x03E718: 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0
0x03E720: 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0
0x03E728: 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0
0x03E730: 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0
0x03E738: 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0
0x03E740: 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0
```

(g): Display Previous Tape Status

This option displays the tape status prior to the current status. Previous tape status looks something like this:

```
previous tape status
tape status:0x00
```

(h): Adverse DMA Test (Writes to Tape)

This option executes three errors:

- parity error
- page protection error
- address error

SUBTEST 5: RUN INTERACTIVE UTILITIES

This subtest consists of 15 options. To access the interactive tape utilities:

1. Type **:16,5**

2. Press **Return**.

This list of utilities is displayed on your screen:

```
qicl utilities
qicl - utilities menu:
    (0)   run diagnostics
    (1)   rewind tape
    (2)   not available
    (3)   copy disk partition to tape with filemark
    (4)   restore disk partition from tape
    (5)   run Tape write/Read Test
    (6)   run Write Phase only
    (7)   run Read Phase only
    (8)   copy disk blocks to tape
    (9)   restore disk blocks from tape
    (a)   reset tape
    (b)   retension test
    (c)   read tape status
    (d)   Set Tape Data Format
    (e)   erase Test
Q(utl)$
```

To execute any of the tape utilities options, you must type its number next to the QIC tape utility prompt Q(util)$. For example, to execute option 6:

1. Type **6**

2. Press **Return**.

To exit the utilites menu:

1. Type **q**

2. Press **Return**.

(0): Run Diagnostics

This option opens the diagnostics menu.

(1): Rewind Tape

This option rewinds the tape currently in the tape drive.

(3): Copy disk partition to tape with filemark

This option is destructive to the tape because it copies a partition from the specified disk and writes that partition to tape. To allow multiple partitions to be copied to the tape, the tape is not rewound after writing the filemark. The sample test output shown below assumes that D0 and partition 1 were selected:

```
Backup Disk Partition to Tape
Enter disk drive # (default = 0): 0-15 :
Volume Name: 52HIT (ecc mode)
Partition 0: start Block=0 (track 0), size (in Blocks)=32
Partition 1: start Block=32 (track 4), size (in Blocks)=12000
Partition 2: start Block=12032 (track 1504), size (in Blocks)=6000
Partition 3: start Block=18032 (track 2254), size (in
Blocks)=18088
Enter partition number to backup to qici (default= 1): 0-15:
Enter number of blocks to archive (default= 6000): 0-6000:
Insert tape, type Return when ready (q to quit)
Qici - controller reset
starting transfer...
transfer complete, X blocks with filemark
```

(4): Restore Disk Pattern from Tape

This option is also destructive. It writes data from tape to disk. To allow multiple partitions to be copied to the tape, the tape is not rewound after reading the filemark.

The sample test output shown below assumes that D0 and partition 1 were selected:

```
Restore Disk Partition from Tape
Enter disk drive # (default = 0): 0-15 :
Volume Name: 52HIT (ecc mode)
Partition 0: start Block=0 (track 0), size (in Blocks)=32
Partition 1: start Block=32 (track 4), size (in Blocks)=12000
Partition 2: start Block=12032 (track 1504), size (in Blocks)=6000
Partition 3: start Block=18032 (track 2254), size (in
Blocks)=18088
Enter partition number to restore from tape (NO DEFAULT) 0-15:
Enter number of blocks to archive (default= 12000): 0-12000:
Write to disk [y/n]?
Insert tape, type Return when ready (q to quit)
Start block X,Y blocks being retested
```

(5): Run Tape Write/Read

This option writes data from memory to tape and then reads the data back to ensure that it was written correctly. It performs the same functions as QIC tape subtests 6 and 7.

It displays the following output:

```
        Qici Write Test
        Qici - controller reset
Controller reset okay
        Rewinding the tape ... Qici- controller reset
Rewind okay
Writing test pattern
Transfer complete, 40000 blocks w/ filemark
        Rewinding the tape ... Qici- controller reset
Rewind okay
Controller PASSED write test
        Qici Read Test
        Qici - controller reset
Controller reset okay
        Rewinding the tape ... Qici- controller reset
Rewind okay
Verifying Write Test Data on Tape ...
        Rewinding the tape ... Qici- controller reset
Rewind okay
Tape Passed read test
```

(6): Run Write Phase Only

This option is the same as QIC tape subtest 6. It writes 40000 blocks of data from memory to tape. It displays this output:

```
        Qici Write Test
        Qici - controller reset
Controller reset okay
        Rewinding the tape ... Qici- controller reset
Rewind okay
Writing test pattern
Transfer complete, 40000 blocks w/ filemark
        Rewinding the tape ... Qici- controller reset
Rewind okay
Controller PASSED write test
```

(7): Run Read Phase Only

This option is the same as QIC tape subtest 7. It verifies that data written in option 6 above from memory to tape is correct. It displays this output:

```
        Qici Read Test
        Qici - controller reset
Controller reset okay
        Rewinding the tape ... Qici- controller reset
Rewind okay
Verifying Write Test Data on Tape ...
        Rewinding the tape ... Qici- controller reset
Rewind okay
Tape Passed read test
```

(8): Copy Disk Blocks to Tape

This option is similar to Utilities option 3 except that this option lets you copy blocks, instead of partitions, from disk to tape. It displays the following sample output:

```
Backup disk blocks to tape with filemark
Enter disk drive # (default = 0): 0-15:
Volume Name:52HIT (ecc mode)
Partition 0: start Block=0 (track 0), size (in Blocks)=32
Partition 1: start Block=32 (track 0), size (in Blocks)=12000
Partition 2: start Block=12032 (track 0), size (in Blocks)=6000
Partition 3: start Block=18032 (track 0), size (in Blocks)=18088
Enter starting block number (default=0):
Enter number of blocks (default= 36120):
Insert tape, type Return when ready (q to quit)
Starting Transfer...
```

(9): Restore Disk Blocks From Tape

This option is similar to Utilities option 4 except that this option lets you restore blocks, instead of partitions, to disk. It displays this sample output:

```
Restore disk blocks from tape
Enter disk drive # (default = 0): 0-15:
Volume Name:52HIT (ecc mode)
Partition 0: start Block=0 (track 0), size (in Blocks)=32
Partition 1: start Block=32 (track 0), size (in Blocks)=12000
Partition 2: start Block=12032 (track 0), size (in Blocks)=6000
Partition 3: start Block=18032 (track 0), size (in Blocks)=18088
Enter starting block number (default=0):
Enter number of blocks (default= 36120):
Write to disk? (y or n):
Insert tape, type RETURN when ready (q to quit)
Starting transfer...
```

To allow multiple partitions to be copied to the tape, the tape is not rewound after reading the filemark.

(a): Reset tape

This option resets the tape controller. It displays this output:

```
Resetting controller ...
```

(b): Retension Test

This option rewinds the tape, resets the controller, and retensions the tape. It displays this output upon successful completion:

```
Retension tape test
Rewinding the tape ... Qici- controller reset
Rewind okay
PASSED Retension test
```

(c): Read Tape Status

This option displays the tape status. See the same option in the diagnostics section for a complete description.

(d): Set Tape Data Format

This option is identical to Tape Subtest 8, which lets you specify the number of blocks and data pattern for options 6 and 7 above.

(e): Erase Test

This option erases the tape currently in the tape drive.

```
Erase tape test
Rewinding the tape ... Qici- controller reset
Rewind okay
```

SUBTEST 6: WRITE PHASE OF TAPE TEST

This subtest writes 40000 blocks of data (pattern = t=0x29D629D6) from memory to tape. The test is successful when the data is successfully transferred.

NOTE: Subtest 8 lets you specify an alternate number of blocks and an alternate data pattern.

To execute subtest 6:

1. Type **:16,6**
2. Press **Return**.

The following output confirms successful test completion:

```
        Qici Write Test
        Qici - controller reset
Controller reset okay
        Rewinding the tape ... Qici- controller reset
Rewind okay
Writing test pattern
Transfer complete, 40000 blocks w/ filemark
        Rewinding the tape ... Qici- controller reset
Rewind okay
Controller PASSED write test
```

SUBTEST 7: READ PHASE OF TAPE TEST

This subtest verifies that the data written from memory to tape in subtest 6 is correct. To execute subtest 7:

1. Type **:16,7**
2. Press **Return**.

The following output confirms successful completion of subtest 7:

```
        Qici Read Test
        Qici - controller reset
Controller reset okay
        Rewinding the tape ... Qici- controller reset
Rewind okay
Verifying Write Test Data on Tape ...
        Rewinding the tape ... Qici- controller reset
Rewind okay
Tape Passed read test
```

SUBTEST 8: SET UP PARAMETERS FOR TAPE TEST

This subtest lets you specify the number of blocks and the data pattern for subtests 6 and 7.

To select subtest 8:

1. Type **:16,8**

2. Press **Return**.

Subtest 8 prompts you to enter the number of blocks to be written to, and read from, the QIC tape. The default setting is 40000 blocks. You are also prompted to enter the data pattern to be transferred from memory to tape in subtest 6.

```
Qici Default Tape Data Format Setup
Number of 512 Byte Blocks/Filemark [default=40000] 0-100000
Input test pattern [default=0x29D629D6]:
Test will read/write 40000 blocks with pattern 0x29D629D6
```

SUBTEST 9: ERASE TAPE TEST

This subtest erases the contents of the tape in the QIC tape drive.

1. Type **:16,9**

2. Press **Return**.

This output confirms execution of subtest 9:

```
            Erase tape test
            Rewinding the tape ... Qici- controller reset
Rewind okay
Erase status okay
Read test of erased tape
PASSED Erase Tape Test
```

SUBTEST 10: RETENSION TAPE TEST

This subtest retensions the tape, rewinds it, and resets the tape controller. To execute subtest 10:

1. Type **:16,10**

2. Press **Return.**

The following output is displayed on the terminal screen:

```
            Retension tape test
            Rewinding the tape ... Qici- controller reset
Rewind okay
PASSED Retension test
```

TEST 20: GENERAL TAPE DRIVE TESTS (j)

The general tape drive subtests, listed in Table 2, test every aspect of the SCSI and Interphase half-inch tape drives (no other half-inch tape drives are supported) attached to your system. They are described on the following pages. Additionally, the general tape test sequence, consists of subtests 1, 9, 10, and 11 that are listed in Table 6.

Table 6

GENERAL TAPE DRIVE SUBTESTS (TEST 20)

Subtest	Description
1*	Reset and Initialize
2	Set Default Controller Parameters
3	Show Drive status and Parameters
4	Set Tape Drive Parameters
5	Change Tape Speed/Density
6	Rewind the Tape
7	Erase the Tape
8	Seek to Next Filemark
9*	DMA Address Counter Test
10*	Write Blocks of a Data Pattern (Write Test)
11*	Read Blocks of a Data Pattern (Read Test)
12	Copy Disk Partition to Tape
13	Restore Disk partition from tape
14	Copy Disk Blocks to Tape
15	Restore Disk Blocks from Tape
16	Toggle multiple tape drive modes

* Contained in General Tape Test Sequence (j).

SELECTING THE TAPE DRIVE

An S/MT Series computer system can contain several tape drives. The SCSI tape drive in CPU-02 systems is the drive initially selected by the tests. To determine the currently selected tape drive:

1. Type **TN?** (or TN ?)
2. Press **Return**.

   A message like this is displayed:

   Tape Drive 1 is selected.

NOTE: Even though the SCSI tape drive is located at location 0, the diagnostics test sees it and will report it in position 1. This is because the drive is shipped from the factory set for unit number 1.

You can use the two-letter **TN** command to change the tape drive number. To do this, you need to determine the number of the tape drive you want to select.

Each controller can support up to eight tape drives. The controller on the CPU board is designated as 0. Each controller on the VME bus is designated as 1, 2 and so on, depending on the position of the controller in the EEPROM table. (This position usually, but not always, is the same as the slot number where the card is installed. See test 18, subtest 12 for more information.)

To determine the tape drive number, use the following formula:

```
TN = (controller# * 8) + unit#
```

For example, to determine the tape number of the third drive attached to controller 1:

```
TN = (controller# * 8) + unit#
   = (1 * 8) + 3
   = 11
```

To select tape drive number 11:

1. Type **TN11** (or TN 11)

2. Press **Return**.

   This message appears:

   Tape Drive 11 is selected.

   You can also issue this two-letter command repeatedly to cycle through other preset tape drive numbers.

NOTE: It is possible to selecta drive that is not attached to your system. Use the diagnostic screen to determine which controllers and drives are attached to your system.. The screen will indicate which controllers it can find and which drives are online.

EXECUTING THE TAPE DRIVE TEST SEQUENCE

The general tape drive test sequence (j) consists of the following four subtests, which are executed in the order shown below.

- Subtest 1: Reset and Initialize
- Subtest 9: DMA Address Counter Test
- Subtest 10: Write Blocks of a Data Pattern (Write Test)
- Subtest 11: Read Blocks of a Data Pattern (Read Test)

CAUTION: The diagnostic assumes that it has permission to write to the tape when you issue this command.

To execute the entire general tape drive test sequence (see the following pages to execute one subtest at a time):

1. Type **j**
2. Press **Return**.

WARNING MESSAGES

Two warning messages can appear during the diagnostic session.

If the maximum block size is set to a value below the minimum required block size for a subtest, the following warning message will appear:

> This subtest requires a maximum block size of at least XXX bytes.
> Run subtest 4 to set the maximum block size

If the tape drive is currently rewinding and a read or write operation is requested for the drive, the following message is displayed, indicating that the diagnostic is waiting for the tape to rewind before continuing.

> Waiting for drive X to finish rewinding...done

SUBTEST 1: RESET AND INITIALIZE

This subtest will reset and initialize the controller for the selected tape drive and print the controller status on the screen.

To execute subtest 1:

1. Type **:20,1**

2. Press **Return.**

   This output confirms test execution:

   Tape Test (subtest 1) reset and initialize
   Controller X [controllername] Parameters:none

SUBTEST 2: SET DEFAULT CONTROLLER PARAMETERS

This subtest sets predefined parameters proper for the controller selected. Note that this subtest does not apply to the SCSI tape drive or to Interphase controllers.

1. Type **:20,2**

2. Press **Return**

   The following prompts are displayed:

   This subtest does not apply to the [controllername] controller.

SUBTEST 3: SHOW DRIVE STATUS AND PARAMETERS

This subtest prints the current tape drive status and the parameters that have been set for the drive.

To run subtest 3:

1. Type **:20,3**

2. Press **Return**

The following output (shown here for a SCSI tape drive) is displayed:

```
Tape Test (subtest 3) show drive status and
parameters
Drive 1 [SCSI]
Maximum block size is undefined
Parameters have not been set for this session.
```

The following output (shown here for a Cipher F880 tape drive) is displayed:

```
Tape Test (subtest 3) show drive status and
parameters
Drive 8 [Interphase] Status:<online, ready, beginning of
tape,>
Drive 8 [Interphase] Parameters- Density:Default density
Drive used Density select bit
Pins used: <J2-50 is SPD,>
Maximum block size is undefined
Parameters have not been set for this secession
```

The first line indicates the current status of the drive as seen by the controller. The rest of the output is drive parameter information currently stored in the controller's memory. The maximum block size is the maximum number of blocks the diagnostics will use as the block size for this drive.

SUBTEST 4: SET TAPE DRIVE PARAMETERS

This subtest enables the user to set the drive parameters. There are several predefined parameter groups corresponding to different tape drives. If you do not select one of these parameters, you will be prompted for all of the necessary information.

To set tape drive parameters:

1. Type **:20,4**

2. Press **Return**.

   If the SCSI tape drive is selected, the following prompt will appear:

   Tape test (subtest 4) set tape drive parameters
   Enter number of bytes in the maximum block size (hex):

   If a half-inch tape drive is selected, the following prompts will appear:

   Tape test (subtest 4) set tape drive parameters
   The pre-defined tape configurations are:

   F880 - Cipher
   F890 - Cipher
   M990 - Cipher

   Enter a format type, or press Return to enter your own parameters:

   If you press **Return**, you must respond to the following prompts:

   Does this drive use the Density Select Bit?
   Does the drive use pin J1-36 as gap select? (y or n)
   Does the drive use pin J1-36 as high speed? (y or n)
   Does the drive use pin J1-36 as high density? (y or n)
   Does the drive use pin J1-44 as gap select? (y or n)
   Does the drive use pin J2-26 as GCR? (y or n)
   Does the drive use pin J2-50 as high speed? (y or n)
   Does the drive use pin J2-50 as high density? (y or n)
   Enter number of bytes in the maximum block size (hex):

If your drive does not use the Density Select Bit, the following prompts will appear directly after the Density Select Bit prompts:

```
Enter drive density code for 800 bpi (hex):
Enter drive density code for 1600 bpi (hex):
Enter drive density code for 3200 bpi (hex):
Enter drive density code for 6250 bpi (hex):
```

Whether you are using predefined parameters or not, the diagnostic test will prompt you for the desired current tape density. If the selected drive uses the Density Select Bit, the following prompt appears:

```
0 - Low Density
1 - High Density
Enter desired current tape density:
```

Note that the meaning of high and low density is drive dependent.

If the selected drive does not use the Density Select Bit, the following prompt appears:

```
0 - 800 BPI
1 - 1600 BPI
2 - 3200 BPI
3 - 6250 BPI
Select desired current tape density:0-3:
```

These parameters are stored in controller memory and need not be reset at power-on of the system.

SUBTEST 5: CHANGE TAPE SPEED/DENSITY

This subtest enables the user to change the tape speed and density without entering all of the parameters in subtest 4. This test is not supported by SCSI tape drives or Interphase controllers.

To run subtest 5:

1. Type **:20,5**
2. Press **Return**.

   The following output appears:

   ```
   Not currently supported by this controller.
   ```

SUBTEST 6: REWIND THE TAPE

This subtest rewinds the current tape.

To rewind the tape:

1. Type **:20,6**
2. Press **Return**.

   For the SCSI tape drive, the following is displayed:

   ```
   Tape Test (subtest 6) rewind the tape
   Drive 1 [SCSI] status:
   <online, ready,>
   ```

   For a half-inch tape drive, the following is displayed:

   ```
   Tape Test (subtest 6) rewind the tape
   Drive 8 [Interphase] Status: <online, ready, tape
   rewinding,>
   ```

SUBTEST 7: **ERASE THE TAPE**

This subtest erases the entire length of the current tape drive.

To erase the tape:

1. Type **:20,7**

2. Press **Return**.

   For the SCSI tape drive or half-inch tape drive, the following message is displayed:

   Tape Test (subtest 7) erase the tape

SUBTEST 8: **SEEK TO THE NEXT FILEMARK**

This subtest will seek to the next filemark of the current tape drive.

To seek to the next filemark:

1. Type **:20,8**

2. Press **Return**.

   The following message is displayed:

   Tape Test (subtest 8) seek to the next filemark

SUBTEST 9: DMA ADDRESS COUNTER TEST

This subtest exercises the DMA address counters of the tape controller. Sixteen 4K byte blocks are written onto the tape. The tape is then rewound, and each block is read into a different virtual address, so that each of the 24 address bits (A24 through A0) can be set to either a 0 or a 1.

Note that the maximum block size (set in subtest 4) must be set to at least 4096 bytes to successfully run this test.

To run the DMA address counter test:

1. Type **:20,9**

2. Press **Return**.

   The following message is displayed:

   > Tape Test (subtest 9) dma address counter test

   If the maximum block size has not been set for the current session, the following warning message will appear:

   > Block size not set for this drive, using default of XXXX bytes

SUBTEST 10: WRITE BLOCKS OF A DATA PATTERN (WRITE TEST)

This subtest writes a data pattern of incrementing 32-bit integers in forty 32K byte blocks to the currently selected tape drive. The pattern is constant for each block, and can be read and verified using subtest 11.

To write blocks of a data pattern to the tape drive:

1. Type **:20,10**

2. Press **Return**.

   The following message is displayed:

   > Tape test (subtest 10) write blocks of a data pattern (Write Test)

   If the maximum block size has not been set for the current session, the following warning message will appear:

   > Block size not set for this drive, using default of XXXX bytes

SUBTEST 11: READ BLOCKS OF A DATA PATTERN (READ TEST)

This subtest will read the pattern written to the selected tape drive in subtest 10.

To read the data pattern written in subtest 10,

1. Type **:20,11**
2. Press **Return**.

   This output confirms successful test completion:

   > Tape test (subtest 11) read blocks of a data pattern (Read Test)

   If the maximum block size has not been set for the current session, the following warning message will appear:

   > Block size not set for this drive, using default of XXXX bytes

SUBTEST 12: COPY DISK PARTITION TO TAPE

This subtest will copy a disk partition from the specified disk drive to the currently selected tape drive. The subtest requires the maximum block (set in subtest 4) to be set to at least 512 bytes.

To copy a disk partition to the tape drive:

1. Type **:20,12**
5. Press **Return**.

Output such as that displayed below (which assumes that drive 0 was selected and partitioned as shown, and that partition 1 was selected for back-up) is displayed on the screen:

```
Tape Test (subtest 12) copy disk partition to tape
Backup Disk Partition to Tape (norewind)
Enter Disk Drive #)default=0):0-60:
```

After entering the drive number and pressing **Return**, the following appears:

```
Volume Name: 52HIT(ecc mode)
Partition 0: start Block=0 (track 0) size (in Blocks)=328
Partition 1: start Block=328 (track 41) size (in
Blocks)=12000
Partition 2: start Block=12328 (track 1541) size (in
Blocks)=6000
Partition 3: start Block=18328 (track 2291) size (in
Blocks)=17792
Enter partition number (Default= 1):0-15:1<Return>
Enter number of disk blocks to
archive(default-12000):0-12000:<Return>
starting transfer...
Backup complete, 12000 blocks with filemark
```

To allow multiple partitions to be copied to the tape, the tape is not rewound after writing the filemark.

SUBTEST 13: RESTORE DISK PARTITION FROM TAPE

This subtest is the reverse of subtest 12; it will copy a disk partition previously written to the tape back to the specified disk. This subtest also requires that the maximum block (set in subtest 4) be set to at least 512 bytes.

CAUTION: This subtest will overwrite any information located at the specified portion of the disk.

To copy the disk partition from the tape back to the disk:

1. Type **:20,13**

2. Press **Return**.

   Output such as that displayed below (which assumes that drive 0 was selected and partitioned as shown) is displayed on the screen:

```
Tape Test (subtest 13) restore disk partition from
tape
Restore Disk Partition to Tape (norewind)
Enter Disk Drive #)default=0):0-60:
```

   After entering the drive number and pressing **Return**, the following appears:

```
Volume Name: 52HIT(ecc mode)
Partition 0: start Block=0 (track 0) size (in Blocks)=328
Partition 1: start Block=328 (track 41) size (in
Blocks)=12000
Partition 2: start Block=12328 (track 1541) size (in
Blocks)=6000
Partition 3: start Block=18328 (track 2291) size (in
Blocks)=17792
Enter partition number (NO DEFAULT):0-15:1<Return>
Enter number of disk blocks in
archive(default-12000):0-12000:<Return>
Write to disk [y/n]?y<Return>
Restore complete, 12000 blocks
```

   To allow multiple groups of disk blocks to be copied from the tape, the tape is not rewound after reading the filemark.

SUBTEST 14: COPY DISK BLOCKS TO TAPE

This subtest enables copying disk blocks to the selected tape without regard to the partition boundaries to the currently selected tape drive. This subtest also requires that the maximum block (set in subtest 4) be set to at least 512 bytes.

CAUTION: This subtest will overwrite any information located at the specified portion of the tape.

To copy disk blocks to tape:

1. Type **:20,14**

2. Press **Return**.

   Output such as that displayed below (which assumes that drive 0 was selected and partitioned as shown) is displayed on the screen:

```
Tape Test (subtest 14) copy disk blocks to tape
Backup Disk Blocks to Tape (norewind)
Enter Disk Drive #(default=0):0-60:
```

   After entering the drive number and pressing **Return**, the following appears:

```
Volume Name: 52HIT(ecc mode)
Partition 0: start Block=0 (track 0) size (in Blocks)=328
Partition 1: start Block=328 (track 41) size (in
Blocks)=12000
Partition 2: start Block=12328 (track 1541) size (in
Blocks)=6000
Partition 3: start Block=18328 (track 2291) size (in
Blocks)=17792
Enter starting block number (Default=
0):0-36120:1<Return>
Enter number of disk blocks to
archive(default-36120):0-36120:<Return>
starting transfer...
Backup complete, 4 blocks with filemark
```

   To allow multiple blocks to be copied to the tape, the tape is not rewound after writing the filemark.

SUBTEST 15: RESTORE DISK BLOCKS FROM TAPE

This subtest is the reverse of subtest 14; it will copy a disk block previously written to the tape back to the specified disk. This subtest also requires that the maximum block size (set in subtest 4) be set to at least 512 bytes.

CAUTION: This subtest will overwrite any information located at the specified portion of the disk.

To copy the disk block from the tape back to the disk:

1. Type **:20,15**

2. Press **Return**.

   Output such as that displayed below (which assumes that drive 0 was selected and partitioned as shown) is displayed on the screen:

```
        Tape Test (subtest 15) restore disk blocks from
        tape
Backup Disk Blocks to Tape (norewind)
Enter Disk Drive #)default=0):0-60:
```

   After entering the drive number and pressing **Return**, the following appears:

```
Volume Name: 52HIT(ecc mode)
Partition 0: start Block=0 (track 0) size (in Blocks)=328
Partition 1: start Block=328 (track 41) size (in
Blocks)=12000
Partition 2: start Block=12328 (track 1541) size (in
Blocks)=6000
Partition 3: start Block=18328 (track 2291) size (in
Blocks)=17792
Enter starting block number (Default=
0):0-36120:1<Return>
Enter number of disk blocks to
archive(default-36120):0-36120:<Return>
Write to disk [y/n]?y<Return>
Restore complete, 4 blocks
```

To allow multiple blocks to be copied from the tape, the tape is not rewound after reading the filemark.

SUBTEST 16: TOGGLE MULTIPLE TAPE DRIVE MODES

Subtest 16 enables you to toggle between multi-drive modes (where all drives are tested) and single drive mode (where only the drive selected by TN) is tested). The default mode is multiple.

To toggle between single and multiple tape drive modes:

1. Type **:20,16**
2. Press **Return.**

   The following output appears:

   Single tape drive mode selected.

   To change back to multiple mode, type 20:16 again, then press **Return.** The following message appears:

   Multiple tape drive mode selected.

RS-232-C Diagnostic Tests

RS-232-C DIAGNOSTIC TESTS

This chapter documents the diagnostic tests for the RS-232-C channels in your system. These channels can be found on the following:

- RS-232-C expansion boards (10-channel and 20-channel) contain 9-pin RS-232-C communications channels
- the Main Processor (CPU) board contains two 25-pin RS-232-C communications channels
- other expansion boards (such as the Ethernet Combo board) contain 9-pin RS-232-C communications channels

TEST 11: RS-232-C (8530) TEST (h)

Test 11 tests every available RS-232-C channel (see Table 1) test.

Table 1

RS-232-C CHANNEL SUBTESTS (TEST 11)

Subtest	Description
1	Data Path Check to the 8530
2	Program all options
3	Simultaneous data transfers
4	Force error conditions
5*	Tests driver pins (RTS, DTR)
6	Select channel for subtest 1
7	Select channel for subtest 2
8	Select channel for subtest 3
9	Select channel for subtest 4
10*	Select channel for subtest 5

* Loopback plug required to execute subtest.

Before executing the RS-232-C tests, determine the number of RS-232-C channels seen by the diagnostic. To do so, display the system configuration:

1. Type **CN**

2. Press **Return**.

If the system you are testing contains one 10-channel RS-232-C Expansion board, then the system configuration looks like this:

```
System configuration:
        7 mega bytes of memory
        2Mb board in slot M1
        4Mb board in slot M2
                .
                .
        RS232 ports 2 to 11 in slots: IO1
                .
                .
        MC68881 coprocessor is not present
        VME expansion board is not present
        Neither IOP nor RS422 boards are present
        Drive 0 is present
        Drive 1 is present
Memory tests will begin at 0x63000, end at 0x6FFFFC.
```

Unless you use the two-letter **CO** command, the RS-232-C subtests assume that the only terminal connected to your system is that connected to Channel 0 on the Main Processor board.

If you use CO to select another RS-232-C channel, then no matter to what channel the console is connected, that channel will not be tested. Also, if tty000 echo is on, Channel 0 will not be tested.

The RS-232-C tests require loopback capability. Since loopback plugs are not always available, subtests 1 through 4 and 6 through 9 can be executed using an internal loopback mode provided by the diagnostic. Information on external loopback plugs can be found in Appendix A.

The RS-232-C test sequence consists of subtests 1 through 5 and 6 through 10.

1. Type **h**

2. Press **Return**.

This output is generated when the tests are successfully executed:

```
RS232 TEST
Onboard: ports 0 to 1
(Subtest 1) Data Path check.  tty001
(Subtest 2) Test 8530 control options.  tty001
(Subtest 3) Data Transfer test.
Transferring data on ports:1
tty001: Loopback plug not detected, using internal loopback mode
(Subtest 4) Error Condition test. tty001
tty001: Loopback plug not detected, using internal loopback mode
(Subtest 5) Test driver pins. tty001
tty001: Loopback plug not detected.Special pin test not run!
```

Test output varies according to the number of RS-232-C channels installed in your system. This output assumes no expansion boards are present. The RS-232-C subtests are described on the following pages.

SUBTEST 1: DATA PATH CHECK TO THE 8530

The 8530 serial controller contains a baud rate register for each channel. Subtest 1 performs a data path check on each 8530 baud rate register. To execute subtest 1:

1. Type **:11,1**

2. Press **Return**.

This output is generated when the tests are successfully executed:

```
RS232 TEST
Onboard: ports 0 to 1
(Subtest 1) Data Path Check. tty001
```

SUBTEST 2: PROGRAM ALL OPTIONS

This subtest tests all options in the 8530 serial controller used by CTIX. It does so in the following sequence. The test transmits and receives the five characters "HELLO" in each of these five modes:

- 8-bit
- 7-bit
- odd parity
- even parity
- no parity

and at each of these 14 baud rates:

50	200	2400
75	300	4800
110	600	9600
134	1200	19200
150	1800	

First it transmits and receives the word "HELLO" in 8-bit characters and at 50 baud to the 8530. The subtest then transmits the same message at 75 baud, and so on to 19200 baud. It does the same for each character set and for each baud rate. To execute subtest 2:

1. Type **:11,2**

2. Press **Return**.

This output is generated when the tests are successfully executed:

```
                RS232 TEST
Onboard: ports 0 to 1
(Subtest 2) Test 8530 control options.  tty001
```

SUBTEST 3: SIMULTANEOUS DATA TRANSFERS

This subtest tests the ability of the 8530 serial controller to transmit large quantities of data at one time. Subtest 3 simultaneously transmits a total of 8400 bytes to several channels. The number of channels tested depends upon the number of RS-232-C channels installed in your system (see Table 2).

If the system you are testing contains two RS-232-C channels, then only channel 1 is tested (the terminal uses channel 0; hence, it is unavailable). In this case, all 8400 bytes are sent to that single channel.

If the system you are testing contains 22 channels, then the subtest becomes more ambitious in its distribution of bytes to channels. The first pass simultaneously transmits 1680 (8400/5 channels) bytes to channels 1 ,2, 7, 12, and 17.

Then it simultaneously transmits 2100 (8400/4 channels) bytes to channels 3, 8, 13, and 18, and so on until each channel is tested.

To execute subtest 3:

1. Type **:11,3**

2. Press **Return.**

This output is generated when the tests are successfully executed:

```
          RS232 TEST
Onboard: ports 0 to 1
(Subtest 3) Data Transfer test.
Transferring data on ports:1
tty001: Loopback plug not detected, using internal loopback mode
```

Table 2

RS-232-C SUBTEST 3, BYTES TRANSMITTED

	Channels	Bytes/Channel
12 Channels		
1st pass	1 2 7	2800 bytes
2nd pass	3 8	4100 bytes
3rd pass	4 9	4100 bytes
4th pass	5 10	4100 bytes
5th pass	6 11	4100 bytes
22 Channels		
1st pass	1 2 7 12 17	1680 bytes
2nd pass	3 8 13 18	2100 bytes
3rd pass	4 9 14 19	2100 bytes
4th pass	5 10 15 20	2100 bytes
5th pass	6 11 16 21	2100 bytes
32 Channels		
1st pass	1 2 7 12 17 22 27	1200 bytes
2nd pass	3 8 13 18 23 28	1400 bytes
3rd pass	4 9 14 19 24 29	1400 bytes
4th pass	5 10 15 20 25 30	1400 bytes
5th pass	6 11 16 21 26 31	1400 bytes
42 Channels		
1st pass	1 2 7 12 17 22 27 32 37	933 bytes
2nd pass	3 8 13 18 23 28 33 38	1050 bytes
3rd pass	4 9 14 19 24 29 34 39	1050 bytes
4th pass	5 10 15 20 25 30 35 40	1050 bytes
5th pass	6 11 16 21 26 31 36 41	1050 bytes

SUBTEST 4: FORCE ERROR CONDITIONS

Subtest 4 forces an error to the 8530 serial controller by transmitting a message at 7 bits per character, while instructing the receiver to read the message at 8 bits per character. After transmitting two characters, this should generate a framing error. Subtest 4 executes the write/read process once per channel. To execute subtest 4:

1. Type **:11,4**

2. Press **Return**.

This output is generated when the tests are successfully executed:

```
                RS232 TEST
Onboard: ports 0 to 1
(Subtest 4) Error Condition test. tty001
tty001: Loopback plug not detected, using internal loopback mode
```

SUBTEST 5: TESTS DRIVER PINS: RTS, DTR (LOOPBACK PLUG REQUIRED)

NOTE: Loopback plugs are required to execute this subtest. The success of this test depends upon accurate loopback pin assignments. See Appendix A for proper RS-232-C loopback plug pin assignments.

Subtest 5 tests the RTS, DTR, CTS, and Carrier bits (or pins) contained in the two chips located on the Main Processor board that are used to drive RS-232-C signals. It tests the loop, local ring indicator, and test bits if the channel is onboard. These bits are tested only when a Loopback plug is used in the channel being tested. This subtest also tests the pins that are looped back.

To run subtest 5,

1. Type **:11,5**

2. Press **Return.**

If no loopback plug is detected, this message is displayed:

ttyxxx: Loopback plug not detected. Special pin test not run!

Where **ttyxxx** refers to a specific channel, such as tty003 or tty022.

SUBTEST 6: SELECT CHANNEL FOR SUBTEST 1

This subtest executes subtest 1 (data path check) on a channel that you select.

1. Type **:11,6**

2. Press **Return.**

The output generated depends upon which channel you select.

SUBTEST 7: SELECT CHANNEL FOR SUBTEST 2

This subtest executes subtest 2 (program all options) on a channel that you select.

1. Type **:11,7**

2. Press **Return.**

The output generated by subtest 7 depends upon the channel you select.

SUBTEST 8: SELECT CHANNEL FOR SUBTEST 3

This subtest executes subtest 3 on a channel that you select.

1. Type **:11,8**

2. Press **Return.**

The output generated by subtest 8 depends upon the channels you select. You can select several channels per test. For example, to select channels 1, 2, and 3:

1. Type **1**

2. Press **Return.**

The integer 1 appears on the screen and this prompt appears:

Which channel? :

1. Type **2**

2. Press **Return.**

The integer 2 appears on the screen and this prompt appears:

Which channel? :

1. Type **3**

2. Press **Return.**

The integer 3 appears on the screen and this prompt appears:

Which channel? :

To execute the subtest:

1. Press **Return.**

Subtest 3 is executed according to the channels specified in the sequence above.

SUBTEST 9: SELECT CHANNEL FOR SUBTEST 4

This subtest executes subtest 4 (force error conditions) on a channel that you select. Recall that subtest 4 forces an error by transmitting a message at 7 bits per character while instructing the receiver to read that same message back at 8 bits per character. To execute subtest 9:

1. Type **:11,9**

2. Press **Return.**

The output generated by subtest 9 depends upon the channel you select.

SUBTEST 10: SELECT CHANNEL FOR SUBTEST 5

This subtest executes subtest 5 (test driver pins) on a channel that you select. Subtest 10 requires loopback plugs since internal loopback mode is not sufficient to test the driver pins. To execute subtest 10:

1. Type **:11,10**

2. Press **Return.**

The output generated by subtest 10 depends upon the channel you select.

IOP Accelerator Diagnostic Tests

IOP ACCELERATOR DIAGNOSTIC TESTS

This chapter documents the diagnostic tests for the Input/Output Processor (IOP) Accelerator expansion boards (one per system) that you can install in the IOP/RS-422 slot of the local card cage. (See the installation manual for your system for the location of the IOP slot).

NOTE: The subtests described in this section will be ignored when the board is not installed.

TEST 5: IOP ACCELERATOR BOARD TEST (i)

This series of subtests (see Table 1) ensures that the IOP Accelerator board functions properly. The IOP Accelerator board contains an MC68000 processor, a 16-bit wide data bus, and 64K bytes of memory. The IOP Accelerator board contains its own physical memory and provides access to host virtual memory. The IOP Accelerator board also contains a printer connector. Refer to Test 13, Line Printer Test, for the subtests that test that printer connector.

There are 13 IOP subtests; all of which are included in the IOP test sequence. Subtests 1 through 6 are executed from host virtual memory. Subtest 7 uploads the remaining IOP diagnostic tests from host virtual memory to the IOP Expansion board. Subtests 8 through 13 are then executed directly from the physical memory on the IOP Expansion board.

To execute the IOP test sequence:

1. Type **i**

2. Press **Return**.

   Before the Host diagnostic returns to the command prompt (command >), it asserts the reset line to the MC68000 on the IOP board, affectively turning the IOP board off. Each time Test 5 is entered into, this reset is negated, and the IOP board is booted.

   For example, if the system you are testing contains one 10-channel RS-232-C Expansion board, then this output appears when the subtests are successfully executed (all messages starting with FROM IOP: or ERROR FROM IOP: are actually generated from the IOP diagnostic program):

```
                IOP TEST
booting iop ...
FROM IOP:  IOP RAM SUBTEST 1
FROM IOP:  IOP RAM SUBTEST 2
FROM IOP:  IOP RAM SUBTEST 3
FROM IOP:  IOP RAM SUBTEST 4
FROM IOP:  IOP RAM SUBTEST 5
FROM IOP:  IOP RAM SUBTEST 6
Subtest 7 uploading IOP
Subtest 8 IOP interrupt test
Subtest 9 IOP clock data path test
FROM IOP: Testing clock 0
FROM IOP: Testing clock 1
FROM IOP: Testing clock 2
Subtest 13 IOP 232 data path check
FROM IOP:  RS232 Data Path Check.  tty002
FROM IOP:  RS232 Data Path Check.  tty003
FROM IOP:  RS232 Data Path Check.  tty004
FROM IOP:  RS232 Data Path Check.  tty005
FROM IOP:  RS232 Data Path Check.  tty006
FROM IOP:  RS232 Data Path Check.  tty007
FROM IOP:  RS232 Data Path Check.  tty008
FROM IOP:  RS232 Data Path Check.  tty009
FROM IOP:  RS232 Data Path Check.  tty010
FROM IOP:  RS232 Data Path Check.  tty011
```

Table 1

IOP DIAGNOSTIC TESTS (TEST 5)

Subtest	Description
1	Write and read 0x0000's.
2	Write and read 0xffff's
3	Write and read 0xaaaa's
4	Write and read 0x5555's
5	Write and read addresses
6	Write and read address complements
7	Upload IOP resident subtests
8	Run interrupt test
9	Run Data Path Check to 8253
10	Run Clock 0 test
11	Run Clock 1 test
12	Run Clock 2 test
13	Run Data Path Check to existing RS-232-C channels

SUBTESTS 1 TO 6: IOP MEMORY SUBTESTS

There are six subtests that perform memory tests on the static RAM on the IOP Accelerator board. Because the IOP Expansion board is fetching instructions for these tests from host virtual memory, the entire 64K bytes of memory is tested. These subtests are defined as follows:

- Subtest 1: write and read 0x0000's
- Subtest 2: write and read 0xffff's
- Subtest 3: write and read 0xaaaa's
- Subtest 4: write and read 0x5555's
- Subtest 5: write and read addresses
- Subtest 6: write and read address complements

Because the data bus onboard the IOP Accelerator board is 16-bits wide, the predefined patterns are written in 16-bit mode. You can execute any one of these individual subtests.

For example, to execute one of these subtests:

1. Type **:5,X** (where X is the subtest number)

2. Press **Return**.

   This output is generated when the tests are successfully executed:

```
            IOP TEST
booting iop ...
FROM IOP:  IOP RAM SUBTESTX
```

SUBTEST 7: UPLOAD IOP RESIDENT SUBTESTS

This subtest loads the IOP diagnostics from host virtual memory to the physical memory onboard the IOP Accelerator board.

To execute subtest 7:

1. Type **:5,7**

2. Press **Return**.

   This output is generated when the tests are successfully executed:

```
            IOP TEST
booting iop ...
Subtest 7 uploading IOP
```

NOTE: The IOP resident tests must be uploaded to execute subtests 8 through 13. Therefore, subtest 7 is automatically executed when you specify a subtest greater than 7.

SUBTEST 8: RUN INTERRUPT TEST

This subtest checks the IOP Interrupt signals generated by the MC68000 processor. To do so, the host CPU issues a command to send an interrupt signal to the IOP board. The IOP board then sends an interrupt signal to the host CPU.

To execute subtest 8:

1. Type **:5,8**

2. Press **Return**.

   This output is generated when the tests are successfully executed:

```
            IOP TEST
booting iop ...
Subtest 7 uploading IOP
Subtest 8 IOP interrupt test
```

SUBTEST 9: RUN DATA PATH CHECK TO 8253

The IOP Accelerator board contains an 8253 counter/timer chip. This subtest executes a data path check to that 8253 chip.

To execute subtest 9:

1. Type **:5,9**

2. Press **Return**.

   This output is generated when the tests are successfully executed:

```
            IOP TEST
booting iop ...
Subtest 7 uploading IOP
Subtest 9 IOP clock data path test
```

SUBTEST 10: RUN CLOCK 0 TEST

This subtest checks that Counter 0 on the 8253 chip is functioning correctly.

To execute subtest 10:

1. Type **:5,10**

2. Press **Return**.

   This output is generated when the tests are successfully executed:

```
            IOP TEST
booting iop ...
Subtest 7 uploading IOP
Subtest 10  IOP clock 0 test
FROM IOP:  Testing clock 0
```

SUBTEST 11: RUN CLOCK 1 TEST

This subtest checks that Counter 1 on the 8253 chip is functioning correctly.

To execute subtest 11:

1. Type **:5,11**

2. Press **Return**.

   This output is generated when the tests are successfully executed:

```
            IOP TEST
booting iop ...
Subtest 7 uploading IOP
Subtest 11  IOP clock 1 test
FROM IOP:  Testing clock 1
```

SUBTEST 12: RUN CLOCK 2 TEST

This subtest checks that Counter 2 on the 8253 chip is functioning correctly.

To execute subtest 12:

1. Type **:5,12**

2. Press **Return.**

   This output is generated when the tests are successfully executed:

```
                IOP TEST
booting iop ...
Subtest 7 uploading IOP
Subtest 12  IOP clock 2 test
FROM IOP:  Testing clock2
```

SUBTEST 13: RUN DATA PATH CHECK TO EXISTING RS-232-C EXPANSION CHANNELS

This subtest executes Data Path Checks to the baud rate register of each RS-232-C channel present.

To execute subtest 13:

1. Type **:5,13**

2. Press **Return.**

For example, if the system you are testing contains one 10-channel RS-232-C Expansion board, then this output is generated when the tests are successfully executed:

```
                IOP TEST
booting iop ...
Subtest 7 uploading IOP
Subtest 13  IOP 232 data path check
FROM IOP:  RS232 Data Path Check.  tty002
FROM IOP:  RS232 Data Path Check.  tty003
FROM IOP:  RS232 Data Path Check.  tty004
FROM IOP:  RS232 Data Path Check.  tty005
FROM IOP:  RS232 Data Path Check.  tty006
FROM IOP:  RS232 Data Path Check.  tty007
FROM IOP:  RS232 Data Path Check.  tty008
FROM IOP:  RS232 Data Path Check.  tty009
FROM IOP:  RS232 Data Path Check.  tty010
FROM IOP:  RS232 Data Path Check.  tty011
```

Ethernet Combo Board Diagnostic Tests

ETHERNET COMBO BOARD DIAGNOSTIC TESTS

TEST 15: ETHERNET COMBO BOARD TESTS (n)

This chapter describes the Ethernet Combo board diagnostic tests. If your system contains one or more Ethernet Combo boards, this test (test 15) can be used to ensure the functionality of thc Ethernet port (the RS-232-C channels are tested with RS-232-C diagnostic tests).

The Ethernet cable must be attached to the Ethernet port on the board. Table 1 contains a list to Ethernet Combo board diagnostic tests.

This test is ignored when the board is not installed.

Table 1

ETHERNET COMBO BOARD DIAGNOSTIC TESTS (TEST 15)

Subtest	Description
0	Default test sequence [1,2,3,6,7,8]
1*	Local RAM write/read.
2*	Host memory write/read
3*	Local EEPROM write/read
4	Local EEPROM write pattern [inverse addresses].
5	Local EEPROM read pattern set by 4
6*	Interrupt tests.
7*	Do Loopback, CRC, and Collision Tests.
8*	Do Time Domain Reflectometry tests.
9	External loopback test.
10	Configure/Select a board [print station address]
11	Toggle Debug mode.
12	Toggle packet print mode.
13	Send a test packet.
14	Retransmit packet sent in subtest 13
15	Ping-pong w/ data check
16	Reverse ping-pong w/ data check.
17	Transmit serially-numbered packets
18	Send back-to-back packets to station ZERO
19	Monitor line - save data
20	Monitor line - don't save data
21	Enable board and leave running (for ping-pong, etc.)
22	Toggle DMA channel (IOP/VME)
23	Set station address to zero
24	Toggle interactive mode

* Test sequence n and subtest 0 run these subtests

SUBTEST 1: LOCAL RAM WRITE/READ

This subtest ensures that the 64K bytes of RAM on the Ethernet Combo board is operational. A series of data patterns are written into memory, then read from memory and are compared with the original. Errors are reported. The patterns written (listed in order) include:

- all zeros
- all ones
- all A's
- all fives
- address pattern
- complement address pattern

To execute subtest 1:

1. Type **:15,1**
2. Press **Return**. For a successful test, the following is displayed:

```
FROM CTEN-X  CT-ENET MEMORY SUBTEST 1
FROM CTEN-X  CT-ENET MEMORY SUBTEST2
FROM CTEN-X  CT-ENET MEMORY SUBTEST3
FROM CTEN-X  CT-ENET MEMORY SUBTEST4
FROM CTEN-X  CT-ENET MEMORY SUBTEST5
FROM CTEN-X  CT-ENET MEMORY SUBTEST6
```

SUBTEST 2: HOST MEMORY WRITE/READ

This subtest is similar to subtest 1. Host memory performs the memory subtests (as described in subtest 1) on a 4K byte block of host memory. This verifies DMA operations on the block of memory.

To execute subtest 2:

1. Type **:15,2**
2. Press **Return.**

SUBTEST 3: LOCAL EEPROM WRITE/READ

This test verifies that the 8K byte EEPROM is operational. Two memory test patterns (as described in subtest 1) are written into the EEPROM, then are read and verified. This test takes about two minutes to complete.

To execute subtest 3:

1. Type **:15,3**
2. Press **Return.**

SUBTEST 4: LOCAL EEPROM WRITE PATTERN

This subtest writes a complemented address pattern into the EEPROM. Subtest 5 is used to read and verify this pattern.

To execute subtest 4:

1. Type **:15,4**
2. Press **Return.**

SUBTEST 5: LOCAL EEPROM READ PATTERN SET BY 4

This subtest reads and verifies the complemented address pattern written into the EEPROM by subtest 4.

To execute subtest 5:

1. Type **:15,5**

2. Press **Return**.

SUBTEST 6: INTERRUPT TEST

This subtest checks the interrupt signals generated by the Ethernet Combo board and host CPU. To do so, the host CPU issues a command to send an interrupt signal to the Ethernet Combo board. The Ethernet Combo board then sends an interrupt signal to the host CPU.

To execute subtest 6:

1. Type **:15,6**

2. Press **Return**.

SUBTEST 7: DO LOOPBACK, CRC, AND COLLISION TESTS

This three part subtest is run by the AM7990 LANCE/SIA (LAN Controller for Ethernet/Serial Interface Adapter) chip set. The subtest performs internal loopback testing and verifies that all major components on the LANCE/SIA chip set are functioning correctly.

In the loopback test, an internal loopback is generated. A packet is internally transmitted and received, and the contents of the received packet are verified with the original.

In the CRC test, the CRC of the received packet is compared with a known CRC. Any errors are reported.

In the collision test, a collision is forced somewhere on the line and should be detected by the hardware.

To execute subtest 7:

1. Type **:15,7**

2. Press **Return**.

SUBTEST 8: TIME DOMAIN REFLECTOMETRY TESTS

This subtest verifies that the Ethernet cable is not broken (discontinuity). The LANCE sends a packet out on the line and begins counting until a collision is reported back. Any discontinuity on the Ethernet line is reported.

To execute subtest 8:

1. Type **:15,8**

2. Press **Return**.

   If the cable is not broken, the following message will appear:

   No discontinuity detected.

SUBTEST 9: EXTERNAL LOOPBACK TEST

This subtest begins with the LANCE sending a packet addressed to itself. When the packet is received it is compared with the original packet. Any errors are reported.

To execute subtest 9:

1. Type **:15,9**

2. Press **Return**.

SUBTEST 10: CONFIGURE/SELECT A BOARD

This subtest does not perform a diagnostic test. Instead, it enables you to modify the Ethernet Combo board environment.

To execute subtest 10:

1. Type **:15,10**

2. Press **Return**.

   The following prompts appear. To accept the default values, simply press **Return**.

   Select CT-ethernet board [0-3]:
   Enter default # packs [for ping-pong] (current= 100):
   Enter number of buffers (current= 64):
   Enter buffer size [max= 4096] (current = 256)
   Enter current station address [0x7F63 0xFF03 0x7F03]:
   Enter logical address filter:

SUBTEST 11: TOGGLE DEBUG MODE

This subtest does not perform a diagnostic test. Instead, it enables you to toggle between being in and out of Debug mode for the test. When you are in Debug mode, extra messages are printed on the screen.

To execute subtest 11:

1. Type **:15,11**

2. Press **Return.**

   The following prompt will appear.

   DEBUG MODE IS ON.

   To turn Debug mode off, simply run the subtest again. The following message will appear:

   DEBUG MODE IS OFF.

SUBTEST 12: TOGGLE PACKET PRINT MODE

This subtest does not perform a diagnostic test. Instead, it displays the contents of each packet received across the Ethernet line on the screen. By default, it is off.

To execute subtest 12:

1. Type **:15,12**

2. Press **Return**.

   The following prompt will appear.

   PACKET PRINT MODE IS ON.

   To turn Packet Print mode off, simply run the subtest again. The following message will appear:

   PACKET PRINT MODE IS OFF.

SUBTEST 13: SEND A TEST PACKET

This subtest enables you to send a test packet to a specific address on the Ethernet line to check for accurate transmission of the test packet and for any serious transmission problems within the Ethernet line.

To execute subtest 13:

1. Type **:15,13**

2. Press **Return**.

   The following prompts appear. To accept the default values, simply press **Return**.

   ```
   Enter packet type (default = 0):
   Enter destination (default= 0):
   Enter destination #2 (default= end):
   Enter message:
   ```

   Pressing **Return** after the last prompt sends the packet.

SUBTEST 14 RETRANSMIT PACKET SEND IN SUBTEST 13

This subtest enables you to retransmit the packet generated in subtest 13. The message, destinations, and packet type will be the same as in subtest 13.

To execute subtest 14:

1. Type **:15,14**

2. Press **Return**.

SUBTEST 15: PING-PONG WITH DATA CHECK

This subtest requires two systems on the Ethernet line to be running this subtest. The two systems will transmit and receive data to and from each other.

The first station will send a 1K byte packet consisting of a sequential number of bytes (0 to 255). The second station will receive and verify this data, and then transmit the 1K byte packet back to the first station (which also verifies the data). This will be repeated until the specified number of packets has been transmitted back and forth. Any errors are displayed.

To execute subtest 15:

1. Type **:15,15**

2. Press **Return**.

   The following prompts appear. To accept the default values, simply press **Return**.

   ```
   Enter destination (default = 0):
   Enter packet size: (1=odd, 0=even)
   (default = 0):
   Enter number of packets: (current= 100):
   Enter destination #2 (default= end):
   ```

   Pressing **Return** after the last prompt starts the subtest.

SUBTEST 16: REVERSE PING-PONG WITH DATA CHECK

This test is identical to subtest 15, except that the byte sequence of the data pack is reversed (255 to 0).

To execute subtest 16:

1. Type **:15,16**

2. Press **Return**.

SUBTEST 17: TRANSMIT SERIALLY-NUMBERED PACKETS

This subtest enables you to send sequentially-numbered packets to another station on the Ethernet line. Each packet is 64 bytes and contains a number in the first four bytes which is incremented after each transmission. The receiving station will print the contents of this packet if Packet Print mode is enabled. The receiving station must be enabled by running subtest 21.

You must press **ESC** to exit this test.

To execute subtest 17:

1. Type **:15,17**

2. Press **Return**.

   The following prompts appear. To accept the default values, simply press **Return**.

   ```
   Enter packet type (default = 0):
   Enter destination (default= 0):
   Enter destination #2 (default= end):
   Enter delay [default= 15, range = 0-30]:
   ```

   Pressing **Return** after the last prompt starts the subtest.

SUBTEST 18: SEND BACK-TO-BACK PACKETS TO STATION ZERO

This subtest is designed as a troubleshooting aid for technicians debugging the Ethernet Combo board. This subtest simply sends packets (containing no data) addressed to station zero as fast as it can. The transmission can be monitored with a logic analyzer. Note that you must press **ESC** to end the test.

To execute subtest 18:

1. Type **:15,18**

2. Press **Return**.

SUBTESTS 19 AND 20: MONITOR LINE

These subtests are intended to be used as troubleshooting aids for technicians debugging the Ethernet Combo board. Both tests operate in "promiscuous" mode, thereby receiving all packets.

Subtest 19 saves data it receives. When subtest 19 is run it will print out the address of a buffer (64K byte) on the host. Received packets will be saved in this buffer in a ring fashion. The debugger can be used to inspect this buffer.

You can enter the Debugger from here by pressing **Control-B**. This will provide you with status information and the debugger prompt (DBG}). To list available debugger commands, type **he**, then press **Return**.

Subtest 20 does not save data. Instead, it increments a counter whenever a packet of data is received. This counter is printed on the screen at short intervals.

Note you must press **ESC** to exit this test.

To execute subtest 19 or 20:

1. Type **:15,19** or **:15,20**

2. Press **Return.**

SUBTESTS 21: ENABLE BOARD AND LEAVE RUNNING

This subtest does not perform a diagnostic test. Instead it turns the board on and leaves it running so the board can receive packets sent to it by subtests 13, 14, 15, 16, and 17. The board will perform a ping-pong handshake and packet validiation.

To execute subtest 21:

1. Type **:15,21**

2. Press **Return.**

SUBTESTS 22: TOGGLE DMA CHANNEL

This subtest does not perform a diagnostic test. Instead it enables you to toggle between the DMA IOP channel and the DMA VME channel.

This test can be run only on CPU-01 based systems. If you attempt to run it on a CPU-02 based system the following message will appear:

Not used on Miti2

To execute subtest 22:

1. Type **:15,22**

2. Press **Return.**

   The following message appears:

```
        CT-Ethernet test [board=n]
Booting Ethernet board
From CTEN -n: Getting Host Parameters
        Subtest 22 CTEN: Toggle DMA Channel
Selected VME DMA channel
```

SUBTESTS 23: SET STATION ADDRESS TO ZERO

This subtest does not perform a diagnostic test. Instead it enables you set the station address of the board being tested to zero (as opposed to using Subtest 10 to change the station address).

To execute subtest 23:

1. Type **:15,23**

2. Press **Return.**

SUBTESTS 24: TOGGLE INTERACTIVE MODE

This subtest does not perform a diagnostic test. Instead it enables you to decide whether or not you want to interact with the tests. When running in Interactive mode, you will be prompted for information as documented in this section. While not running in Interactive mode, the tests will use the default values and will not prompt you for information.

To execute subtest 24:

1. Type **:15,24**

2. Press **Return**.

   The following message appears:

   INTERACT IS ON

   To turn Interactive mode off, simply run the test again. The following prompt will appear:

   INTERACT IS OFF

RS-422 Expansion Board Diagnostic Tests

RS-422 EXPANSION BOARD DIAGNOSTIC TESTS

This chapter documents the diagnostic tests for the RS-422 expansion boards (2-channel and 4-channel) that you can install in the I/O portion of the local card cage.

NOTE: The subtests described in this section will be ignored when the boards are not installed.

TEST 14: RS-422 EXPANSION BOARD TEST (k)

If your system contains the 2- or 4-cluster channel RS-422 Expansion board, this test can be used to ensure the functionality of the four RS-422 cluster channels on that Expansion board. Test 14 contains eight subtests, all which are described below (see Table 1).

Table 1

RS-422 CLUSTER CHANNEL SUBTESTS (TEST 14)

Subtest	Description
1	Data Path Check (DMA and PCSAR)
2	Auto send/receive tests (2 channels)
3	Auto send/receive tests (all channels)
4	Send/receive user-specified channel
5	Send user-specified memory block
6	Receive user-specified memory block
7	Parity error read test
8	DMA address counter test
9	Reserved for manufacturer

Subtests 2, 3, 7, and 8 assume that cluster channel 0 is connected to cluster channel 1 and that cluster channel 2 is connected to cluster channel 3. Use an RS-422 cable to connect these cluster channels together. See Appendix A for proper pin assignments.

RS-422 test sequence (k) includes subtests 1, 2, 3, 7 and 8 only. It does not include subtests 4, 5, 6, and 9. To execute RS-422 subtests 1, 2, 3, 7, and 8:

1. Type **k**

2. Press **Return.**

SUBTEST 1: DATA PATH CHECK (DMA COUNT REGISTERS AND PCSAR)

This subtest ensures that the RS-422 cluster channels can communicate with the bus interface unit. To do so, it executes a data path check to the DMA count register and to the PCSAR register on the 2652 multi-protocol communications controller (MPCC) chip. The 2652 MPCC is a high-speed data transmitter used to support fast communications.

To execute subtest 1:

1. Type **:14,1**

2. Press **Return.**

SUBTEST 2: AUTO SEND/RECEIVE TESTS (2 CHANNEL)

Subtest 2 first transmits a 2 byte packet to each RS-422 cluster channel. During each subsequent transmission, the packet size is doubled, first to two words and then to a maximum size of 32,768 words. Subtest 2 executes data path checks to the idle cluster channels while waiting for the transmit and receive to complete. To execute subtest 2:

1. Type **:14,2**

2. Press **Return**.

SUBTEST 3: AUTO SEND/RECEIVE TEST (ALL CHANNELS)

This subtest does the same as subtest 2, except that it tests all four RS-422 cluster channels at the same time. It also uses a packet that starts with 100 words. It simultaneously transmits data from cluster channel 0 to cluster channel 1 and from cluster channel 2 to cluster channel 3.

Then, simultaneously, it transmits data from cluster channel 1 to cluster channel 0 and from cluster channel 3 to cluster channel 2. This test confirms that both the transmissions and receptions between the two cluster channels are correct.

This subtest cannot be run on 2-channel RS-422 Expansion boards.

To execute subtest 3:

1. Type **:14,3**

2. Press **Return**.

SUBTEST 4: TRANSMIT AND RECEIVE USER-SPECIFIED CHANNEL

Subtest 4 is similar to subtest 2, except that subtest 4 lets you specify which cluster channels, and pairs of cluster channels, to test. Unless otherwise specified, the transmitting cluster channel defaults to 0 and the receiving cluster channel to 1.

Subtest 4 uses the same rotating data pattern used by subtest 2. As with subtest 2, subtest 4 executes data path checks to the cluster channels not used to transmit and receive data.

Subtest 4 allows you select the data to be transmitted. If you enter 0 as the data pattern, this subtest selects the same data pattern that subtest 2 selects. Otherwise, subtest 4 uses the hexadecimal value you enter as the data pattern.

You can specify any combination of cluster channels, including 3 to 1, 0 to 3, 0 to 2, or 2 to 0. In addition, you may specify the transmit cluster channel only. For example, if you select cluster channel 3 as the transmit channel, then the receive channel defaults to cluster channel 2.

Likewise, if you select cluster channel 2 as the transmit channel, then cluster channel 3 becomes the default receive cluster channel. The same holds true for cluster channels 0 and 1.

To execute subtest 4:

1. Type **:14,4**
2. Press **Return**.

These prompts appear:

```
                FAST COMMUNICATIONS PORT TEST
                Fast Comm Subtest 4 - Xmit/Recv Test
Select Transmit Channel, (default= 0): 0-3:
Select Receive Channel, (default= 1): 0-3:
Enter data pattern to transfer (0= rotating pattern) (default=
0xAAAA):
```

SUBTEST 5: TRANSMIT USER-SPECIFIED MEMORY BLOCK

Subtest 5 requires a second system to execute RS-422 subtest 6, the receive subtest. Subtest 5 enables you to specify what RS-422 cluster channel to transmit from. It also lets you specify the address in memory that contains the data you want to transmit. If you specify no address, then subtest 5 defaults to the first available address in memory not used by the diagnostic. To execute subtest 5:

1. Type **:14,5**

2. Press **Return.**

SUBTEST 6: RECEIVE USER-SPECIFIED MEMORY BLOCK

Subtest 6 requires a second system to execute subtest 5, the transmit subtest. Subtest 6 enables you to specify which RS-422 cluster channel is to receive the data transmitted by subtest 5.

This subtest 6 also enables you to specify the address in memory where you want the data to be sent. If you specify no address, then subtest 6 defaults to the first available address in memory not used by the diagnostic.

To receive subtest 6:

1. Type **:14,6**

2. Press **Return.**

SUBTEST 7: PARITY ERROR READ TEST

This subtest assumes that cluster channels 0 and 1 and cluster channels 2 and 3 are connected together. Subtest 7 transmits even (wrong) parity. The even parity should be acknowledged by bit 15 in the general status register (GSR) and bit 10 in the expansion port ID register. The diagnostic then verifies that those bits registered the even parity and that they generated a parity interrupt. To execute subtest 7:

1. Type **:14,7**

2. Press **Return.**

SUBTEST 8: DMA ADDRESS COUNTER TEST

This subtest assumes that cluster channels 0 and 1 and cluster channels 2 and 3 are connected together. Subtest 8 tests the ability of the RS-422 subsystem to DMA into any DMA virtual address. To execute subtest 8:

1. Type **:14,8**

2. Press **Return.**

VME Subsystem Diagnostic Tests

TEST 17: VME SUBSYSTEM TESTS (v)

This chapter describes the subtests (see Table 1) that verify that the S/MT Series VME functions properly. The VME subtests are specific to the CMC VME card. Therefore, the term CMC VME card is used throughout this chapter.

Table 1

VME SUBSYSTEM SUBTESTS (TEST 17)

Subtest	Description
1	Read a single location from host memory
2	Write a single location from host memory
3	Read, then write, a single location to host memory
4	Copy a 1K byte block from host memory
5	Copy a 1K byte block to host memory
6	Read, write, compare a 1K byte block in host memory
7	Copy, compare a 1K block in host memory
11-17	Same as 1 through 7 in endless loop mode

The VME subtests make the assumptions that the

- VME Card Cage is installed
- VME Interface board is installed
- CMC VME card addresses the VME bus at the address specified in the VME EEPROM

See the installation manual for your system (S/222, S/320, and S/640 only) to install the VME card cage, the VME Interface board, and VME Expansion cards.

Before executing each subtest, the host CPU writes a program into the MC68000 onboard the CMC VME card to instruct the MC68000 to execute the VME subsystem tests. Additionally, the VME subsystem default test sequence consists of subtests 1 through 6 only. The subtests for the CMC VME Expansion card are described on the following pages.

To execute the VME subsystem test sequence:

1. Type **v**
2. Press **Return.**

SUBTEST 1: READ DATA

This subtest writes a program that instructs the MC68000 to read data from an address in host memory. This subtest does not check that the data read was correct, only that the read occurred.

To write and then read data to and from a location in host memory:

1. Type **:17,1**
2. Press **Return.**

SUBTEST 2: WRITE DATA

This subtest writes a program from host memory to the CMC VME card instructing the CMC VME card to write 0x9C7A to an address in host memory. The CMC VME card then checks the address in host memory to ensure that the correct address contains 0x9C7A.

To write data from host memory to the CMC VME card:

1. Type **:17,2**
2. Press **Return.**

SUBTEST 3: READ AND WRITE DATA

This subtest writes a program to the CMC VME card that instructs it to read the contents of an address in host memory. The program then writes the same data back to a different address in host memory. The host compares the data in each address to ensure that it is the same.

To execute subtest 3:

1. Type **:17,3**
2. Press **Return.**

SUBTEST 4: COPY 1K BYTE BLOCK FROM HOST MEMORY

This subtest writes a program that copies a 1K byte block of data from an address in host memory to an address on the CMC VME card. To ensure that the copy worked, the first 2 bytes in CMC VME memory are checked.

To execute subtest 4:

1. Type **:17,4**
2. Press **Return**.

SUBTEST 5: COPY 1K BYTE BLOCK TO HOST MEMORY

This subtest writes a program that copies the same 1K byte block of data from an address on the CMC VME card back into the same address in host memory. To ensure that the copy worked, the first 2 bytes in host memory are checked.

To execute subtest 5:

1. Type **:17,5**
2. Press **Return**.

SUBTEST 6: READ/WRITE/COMPARE 1K BYTE BLOCK

This subtest writes a program to the MC68000 on the CMC VME card, which instructs the CMC VME card to read 1K byte of memory from an address in host memory. That 1K byte is then written to an address on the CMC VME card. The subtest then writes 1K byte from the CMC VME card to an address in host memory. Host memory then checks the entire block of data to ensure that the data read out was the data written in.

To write, read, and compare a 1 Kbyte block of data to and from host memory:

1. Type **:17,6**

2. Press **Return**.

SUBTEST 7: COPY/COMPARE 1K BYTE BLOCK

This subtest instructs the CMC VME card to copy what it sees at an address in host memory to another address in host memory. The subtest then checks both blocks of data to determine that the copy was executed correctly.

To copy and compare a 1 Kbyte block of data in host memory:

1. Type **:17,7**

2. Press **Return**.

SUBTESTS 11 THROUGH 17

Subtests 11 through 17 execute subtests 1 through 7, respectively, in endless loop mode.

To execute any one of subtests 11 through 17:

1. Type **:17,subtest number**

   For example, to execute subtest 6 in endless loop mode:

   Type **:17,16**

2. Press **Return**.

NOTE: The only way to terminate these subtests is to press **q** or ESC/Go. This, however, only returns the diagnostic command prompt; it does not stop the CMC VME card from looping through the respective subtest. To free the CMC VME card from the endless loop, you must reboot the diagnostics.

VME Communication Controller Card Diagnostic Tests

VME COMMUNICATION CONTROLLER DIAGNOSTIC TESTS

TEST 19: VME COMMUNICATION CONTROLLER CARD TEST (b)

This chapter describes the VME Communication Controller card (also referred to as the MPCC card) diagnostic tests. If your system contains one or more VME Communication Controller cards, this test (test 19) can be used to ensure the functionality of the channels on the card.

This test contains 21 subtests, all of which are described below (see Table 1). This test is ignored when the VME Communication Controller card is not installed. You can identify a specific Communication Controller card for testing using the two-letter **MN** command. See Chapter 4 for more information on this command.

TEST FAILURES

If a subtest is successful, the command prompt appears when the subtest completes. If a subtest fails, the following error message appears:

MPCC Bd. #: <test name> failed
Test 19, Subtest #: MPCC bd. #: <test name> failed

Execute subtest 20 at this point to read the last message from the VME Communication Controller card (see the subtest 20 description later in this chapter for more information).

When a test fails, you must reload the VME Communication Controller card before you can continue. See the subtest 3 description later in this section for more information. If you try to execute a subtest without rebooting, the following error message will appear:

On bd. tester not booted.

Table 1

VME COMMUNICATION CONTROLLER CARD DIAGNOSTIC TESTS (TEST 19)

Subtest	Description
0*	Execute tests 1 through 12
1*	MPCC onboard memory access
2*	Write address to MPCC onboard memory
3*	Load and boot MPCC onboard tests
4*	Test mutual interrupts
5*	MPCC onboard local memory test
6*	MPCC onboard CIO data path test
7*	MPCC onboard SCC asynchronous data path test
8*	MPCC onboard SCC synchronous data path test
9*	MPCC onboard VME DMA test
10* **	MPCC onboard SCC DMA test
11* **	MPCC onboard asynchronous external loopback test
12*	Run subtests 5, 8, and 9 concurrently
13	Reset MPCC
14	Execute subtests 4 through 12
15	Toggle single or alternating (default) between board test mode
16	Toggle silent (default) or verbose message mode
17	Show MPCC board configuration
18	Begin user-defined sequence test
18	End user-defined sequence test
20	Print last message sent from MPCC board and TIMEOUT flag
21	Extensive onboard VME DMA test

* Subtest 0 runs these subtests.

** Loopback plug required to execute subtest.

SUBTEST 1: MPCC ONBOARD MEMORY ACCESS TEST

This subtest ensures that card memory can be accessed from the host. The host writes a bit walking pattern into memory, then resets memory. This test takes about three minutes to complete and also tests the address lines (as discussed in subtest 2).

To execute subtest 1:

1. Type **:19,1**
2. Press **Return.**

SUBTEST 2: WRITE ADDRESS AND COMPLEMENT TO MPCC ONBOARD MEMORY

This subtest checks for address line shorts.

To execute subtest 2:

1. Type **:19,2**
2. Press **Return.**

SUBTEST 3: LOAD AND BOOT MPCC ONBOARD TESTER

This subtest loads, and then boots, VME Communication Controller card tests into memory. You must execute this subtest before running subtests 4 through 12.

To execute subtest 3:

1. Type **:19,3**
2. Press **Return.**

SUBTEST 4: MUTUAL INTERRUPTS TEST

This test ensures that interrupts sent between the VME Communication Controller card and the host are received. In the first half of the test, the host sends an interrupt to the VME Communication Controller card, and the card sends a message back indicating whether or not it received the interrupt.

In the second half of the test, the VME Communiaction Controller card sends the host an interrupt, and the host indicates whether or not it received the interrupt.

To execute subtest 4:

1. Type **:19,4**
2. Press **Return**.

SUBTEST 5: MPCC ONBOARD LOCAL MEMORY TEST

This subtest causes the host to tell the VME Communication Controller card to run its local memory test. The test run by the card is similar to subtest 1. The card reports the status of this test back to the host.

To execute subtest 5:

1. Type **:19,5**
2. Press **Return**.

SUBTEST 6: MPCC ONBOARD CIO DATA PATH TEST

This subtest ensures that the CIO data path and clock interrupts are properly operating.

To execute subtest 6:

1. Type **:19,6**
2. Press **Return**.

SUBTEST 7: MPCC ONBOARD SCC ASYNCHRONOUS DATA PATH TEST

This subtest ensures the SCC data path for asynchronous data transmissions. Data is transmitted from the Transmit (Tx) lines (in 4K byte patterns) and is polled (interrupts are not used) and read by the Receive (Rx) lines. Each byte of data is compared, and any errors are reported. This test is repeated for each port on the VME Communication Controller card.

To execute subtest 7:

1. Type **:19,7**
2. Press **Return**.

SUBTEST 8: MPCC ONBOARD SCC SYNCHRONOUS DATA PATH TEST

This subtest is similar to subtest 7. It ensures the SCC data path for synchronous data transmissions. Data is transmitted from the Transmit (Tx) lines (in 4K byte patterns), and interrupts are generated at the Receive lines for receiving. Each byte of data is compared, and any errors are reported. This test is repeated for each port on the VME Communication Controller card.

To execute subtest 8:

1. Type **:19,8**
2. Press **Return**.

SUBTEST 9: MPCC ONBOARD VME DMA TEST

This test verifies that the VME Communication Controller card VME DMA is operating properly. The card transfers words of varying size to host memory, then increments the address and repeats the process. The size can be incremented up to the 4K boundary (from 0x0 to 0x7FF). This test takes two minutes to complete.

To execute subtest 9:

1. Type **:19,9**
2. Press **Return**.

SUBTEST 10: MPCC ONBOARD SCC DMA TEST

This subtest requires that a 25-pin (RS-232-C) loopback connector be attached to each channel on the VME Communiaction Controller card before running the subtest. See Appendix A for information on the loopback connector.

This subtest tests DMA on the four SCC channels on the VME Communiaction Controller card. First the channels are set up so that the SCC can receive DMA and the channels can transmit DMA. Then, a test similar to subtest 9 is executed. Words of different sizes are transmitted, received, and compared on the channel. The VME Communiaction Controller card then increments the address and repeats the process. The address can be incremented up to the 4K boundary (from 0x0 to 0xF81).

Each channel takes up to three minutes to complete the test.

To execute subtest 10:

1. Type **:19,10**
2. Press **Return**.

SUBTEST 11: EXTERNAL LOOPBACK TEST

This subtest requires that a loopback connector be attached to each channel on the VME Communication Controller card before running the subtest. See Appendix A for information on the loopback connector.

This subtest is similar to subtest 7. It is designed to ensure that the line drivers are operating properly.

To execute subtest 11:

1. Type **:19,11**

2. Press **Return.**

SUBTEST 12: CPU BUS LOAD TEST

This subtest runs the VME Communication Controller card local memory subtest (similar to subtest 5), VME DMA subtest (similar to subtest 9) and SCC synchronous subtest (similar to subtest 8) concurrently to determine in the CPU bus can handle the load.

To execute subtest 12:

1. Type **:19,12**

2. Press **Return.**

SUBTEST 13: MPCC BOARD RESET

Subtest 13 does not perform a diagnostic test. Rather, it simply resets the card. Note that you must run subtest 3 after resetting the card if you need to run more tests.

To execute subtest 13:

1. Type **:19,13**

2. Press **Return**.

SUBTEST 14: MULTIPLE SUBTESTS

This subtest enables you to run subtests 4 through 12 without having to specify each test.

To execute subtest 14:

1. Type **:19,14**

2. Press **Return**.

SUBTEST 15: BOARD TEST MODE

This subtest does not perform a diagnostic test. Instead, it enables you to determine if you will run the subtests on one VME Communication Controller card (single mode) or alternate between all VME Communication Controller cards in the system (alternate mode).

Each time you specify this subtest, the selection toggles between single and alternate (the initial default selection). For example, if you specify this subtest after booting the test, the selection will change from alternate to single. Specifying the subtest again will change the selection back to alternate.

To execute subtest 15:

1. Type **:19,15**

2. Press **Return.**

SUBTEST 16: SCREEN MESSAGE MODE

This subtest does not perform a diagnostic test. Instead, it enables you to determine if you will view the messages from the subtests on the screen (verbose mode) or just receive a message that the subtest passes at the end of the subtest (silent mode).

Each time you specify this subtest, the selection toggles between silent (the initial default selection) and verbose. For example, if you specify this subtest after booting the test, the selection will change from silent to verbose. Specifying the subtest again will change the selection back to silent.

To execute subtest 16:

1. Type **:19,16**

2. Press **Return.**

SUBTEST 17: MPCC BOARD CONFIGURATION

This subtest does not perform a diagnostic test. Instead, your system's VME Communication Controller card configuration is displayed on the screen, along with the modes selected in subtests 15 and 16.

To execute subtest 17:

1. Type **:19,17**

2. Press **Return**.

For example, the following sample configuration may appear on the screen:

```
Board o, Test Mode - alternate, Message Mode - verbose
1 MPCC boards of 4 configured are present -

board       addr          present
0           0xC0F00000    YES
1           0xC0C00000    ?
2           0xC0B00000    ?
3           0xC0A00000    ?

command>
```

SUBTESTS 18 AND 19: USER-DEFINED SEQUENCE TEST

This subtest does not perform a diagnostic test. Instead, it enables you to use the user-defined test sequence two-letter command (#=) to run multiple tests on multiple boards. See Chapter 4 for more information on this two-letter command.

In the test sequence you specify, Subtest 18 should preceed any other VME Communication Controller card subtest, and subtest 19 should follow the last VME Communication Controller card subtest.

To create a user defined test sequence:

1. Type **#=**
2. Type **19,18** (there is no period or colon before 19 in this subtest).
3. Press **Return**.
4. Type the subtests you want to run. For example.

   19,4 <Return>
   19,5 <Return>
   19,6 <Return>

5. Type **19,19** to end the user-defined sequence.

To run the user defined sequence

1. Type **#**
2. Press **Return**.

If you want to run the sequence more than once, type **X#**, with **X** indicating the number of times you want to run the sequence.

SUBTEST 20: PRINT LAST MESSAGE AND TIMEOUT FLAG

This subtest does not perform a diagnostic test. Instead, you can display the last message sent from the VME Communication Controller card flag on the screen. (This message appears after the word "onboard:" in the screen message). This subtest also displays the timeout flag value if it is set for the last test run.

To execute subtest 20:

1. Type **:19,20**

2. Press **Return.**

The following appears on the screen:

```
CTIX MPCC diag TIMEOUT flag is NOT SET
onboard: PASSED
```

SUBTEST 21: EXTENSIVE ONBOARD VME DMA TEST

This subtest runs the same test sequence as subtest 9. The major difference, however, is that subtest 21 repeats each transfer up to 40 times. This subtest can take up to 90 minutes to complete.

To execute subtest 21:

1. Type **:19,21**

2. Press **Return.**

Miscellaneous Diagnostic Tests

MISCELLANEOUS DIAGNOSTIC TESTS

This chapter describes the subtests contained in Test 18 and presented in Table 1. There is a MC68881 coprocessor test, an Uninterruptable Power Supply (UPS) line driver port test, several VME Interface board tests, and some EEPROM tests.

To execute the default test sequence (subtests 1 through 4 and 17):

1. Type **z**

2. Press **Return**.

Subtest 0 also runs the above test sequence.

LIMITS ON WRITING TO THE EEPROM

The EEPROM can only be written to twenty times in one diagnostic session. If this number is exceeded, the following message appears:

Exceeded 20 eeprom writes for the session, skipping.

The current write test is not run. You must reboot the diagnostics to write to the EEPROM again.

Table 1

MISCELLANEOUS TESTS (TEST 18)

Subtest	Description
1	MC68881 test
2	UPS line driver test
3	VME bus cycle timeout for nonpresent card
4	VME protection register check
5	VME EEPROM read pattern
6	VME EEPROM write pattern
7	VME EEPROM clear (write all zeros)
8	VME EEPROM set (write all 0xFFs)
9	VME EEPROM set type (set type to ctix or diag)
10	VME EEPROM set checksum (calculate checksum and set it) also sets type to diag
11	VME EEPROM set address and board type (also sets checksum)
12	VME EEPROM print the board types and addresses
13	VME EEPROM clear the addresses and board types
14	VME EEPROM set up 'default' table of board types
15	VME EEPROM save contents of eeprom
16	VME EEPROM restore eeprom to state before eeprom write
17	VME EEPROM confirm checksum correctness
20	Serial number ROM test (CPU-02 based systems only)

SUBTEST 1: MC68881 TEST

This subtest checks the MC68881 coprocessor by first executing a data path check on the address and data lines that connect the MC68881 and the MC68020. The test then executes a simple arithmetic operation on the MC68881.

To check the MC68881 floating point processor:

1. Type **:18,1**

2. Press **Return**.

   When the subtest is successfully completed, one of these two confirmation messages appears:

   68881 operation compare ... passed

   68881 operation compare ... FAILED

   If, however, the MC68881 fails the first portion of the test (the data path check), this error message is displayed on the terminal screen:

   68881 data path error wrote: x, read: x

   In this case, the second part of the subtest is not executed.

SUBTEST 2: UPS PORT LINE DRIVER TEST

This subtest tests the UPS line driver port on the Main Processor board. This test requires a UPS loopback plug. See Appendix A for the UPS loopback plug pin assignments. Subtest 2 transmits a signal through the loopback plug as though it were transmitting signals to the UPS module. The subtest is successful if the signal is received correctly.

To test the UPS line driver port on the Main Processor board:

1. Type **:18,2**

2. Press **Return**.

   No confirmation is displayed when the subtest is successful. If the subtest is unsuccessful, one of these two messages appears:

   FAILED: UPS line driver port failed to go low.

   FAILED: UPS line driver port failed to go high.

NOTE: Since a UPS loopback plug might not always be available, failure of this test is not a fatal condition. The diagnostic will continue to execute other tests and subtests in the respective test sequence.

SUBTEST 3: VME BUS CYCLE TIMEOUT

This subtest tests the read cycle timeout ability of the VME Interface board. This subtest does not require that any VME cards be installed in the VME Card Cage assembly.

To test the read cycle timeout ability of the VME Interface board:

1. Type **:18,3**

2. Press **Return**

   After you begin the subtest, and before the diagnostic issues a read instruction across the VME bus, this message is displayed on the screen:

   FAILED: please reset

   When this message appears, the diagnostics executes the read cycle timeout. If the read cycle timeout is unsuccessful, the earlier message remains on the terminal screen. To use the computer, you must reboot it.

   If the read cycle timeout is successful, then this message overwrites the message (FAILED: please reset) already on the screen:

   VME bus cycle complete

SUBTEST 4: VME PROTECTION REGISTER CHECK

This subtest ensures that the VME protection register is functioning correctly. The VME protection register contains 8 bits which are used to protect the areas in virtual memory accessible in User Mode.

To validate the VME protection register:

1. Type **:18,4**

2. Press **Return**.

   If the VME Protection register malfunctions, one of these two error messages is displayed on the terminal screen:

   Bus fault on user write to address (not protected)

   No bus fault on user write to address (protected)

SUBTEST 5: VME EEPROM READ PATTERN

Subtest 5 reads data, in the form of a checksum and pattern, from the EEPROM on the VME Interface board. This subtest should be executed after subtest 6, VME EEPROM write pattern.

To read the checksum and pattern written to the VME EEPROM on the VME Interface board:

1. Type **:18,5**

2. Press **Return**.

   The checksum is stored in the first 4 bytes on the EEPROM. The diagnostic looks at the checksum, reads it to ensure validity, then looks at the second 4 bytes on the EEPROM to determine if its bits are set for UNIX. If so, this message is displayed on the screen and the subtest is complete:

   VME EEPROM checksum is valid, but data is set for UNIX

   If the second 4 bytes is not set for UNIX, then this error message is displayed:

   VME EEPROM data error; x expected, read x.

SUBTEST 6: VME EEPROM WRITE PATTERN

Subtest 6 is the only subtest not included in the test sequence. It is used to write a data pattern to the EEPROM on the VME Interface board. Data is written to the EEPROM only when the checksum is not valid and when the flag bits are not set for CTIX. If the flag bits are set to CTIX, the contents of the EEPROM are saved before writing to the EEPROM.

Note that you can only run the EEPROM twenty times per diagnostic session (each system boot). See "Writing to the EEPROM," earlier in this section, for more information.

To write a data pattern to the VME EEPROM on the VME Interface board:

1. Type **:18,6**

2. Press **Return**.

   When subtest 6 is executed, this message appears on the screen:

   Checksum valid, but data set for CTIX

   If subtest 6 is not executed as part of a test sequence, then this message appears:

   Saving EEPROM contents...done

   CTIX VME driver information will be overwritten. Enter y to confirm.

To answer the prompt, you must type either y or n. In this case, a carriage return is the same as No. The latter case enables you to override the contents of the second 4 bytes in the EEPROM. Doing so will require you to restore the CTIX information. See the CTIX Operating System manual and the MightyFrame Administrator's Reference manual to do so.

SUBTEST 7: VME EEPROM CLEAR (WRITE ALL ZEROS)

Subtest 7 clears the EEPROM by writing zeros to the EEPROM on the VME Interface board and then verifies that what was written is correct. The contents of the EEPROM are saved before the write occurs. (To restore the contents of the EEPROM, run subtest 16.)

To run subtest 7:

1. Type **:18,7**

2. Press **Return.**

   The following output appears:

```
VME EEPROM clear (write all zeros)
Saving EEPROM contents...done
```

SUBTEST 8: VME EEPROM SET (WRITE ALL 0xFFs)

Subtest 8 sets the EEPROM by writing f's to the EEPROM on the VME Interface board and then verifies that what was written is correct. The contents of the EEPROM are saved before the write occurs. (To restore the contents of the EEPROM, run subtest 16.)

To run subtest 8:

1. Type **:18,8**
2. Press **Return**.

   The following output appears:

```
VME EEPROM set (write all 0xFFs)
Saving EEPROM contents...done
```

SUBTEST 9: VME EEPROM SET TYPE

Subtest 9 enables you to set the type of the VME Interface board EEPROM for CTIX or diagnostics. Setting the board for CTIX enables the device drivers to communicate with the VME controller EEPROM contents. Setting the type to diagnostics causes the device drivers to ignore the contents of the EEPROM.

A valid checksum must exist for this subtest to successfully complete. Run subtest 10 to set the checksum. If the checksum is invalid, the following message will appear:

```
VME EEPROM checksum is not valid.
```

To run subtest 9:

1. Type **:18,9**

2. Press **Return**.

   The following output appears:

```
        VME EEPROM set type (set type to ctix or diag)
Set VME EEPROM flag (0=diag, 1-ctix)
```

   Enter the board type at the prompt. If you just press **Return**, the flag will not be set.

SUBTEST 10: VME EEPROM SET CHECKSUM

Subtest 10 calculates and sets the checksum for the VME EEPROM on the VME Interface board. This checksum must be set before running subtests 9, 11 and 12. This subtest also sets the board type flag to diagnostics.

To set the checksum:

1. Type **:18,10**

2. Press **Return**.

   The following output appears:

```
        VME EEPROM set type (calculate checksum
        and set it)
        also sets type to diag.
Setting EEPROM checksum
Flag is set for the diagnostics
```

SUBTEST 11: VME EEPROM SET ADDRESS AND BOARD TYPE

Subtest 11 sets the board type and address of the EEPROM. The type flag must be set to ctix before running this subtest.

To set the address and board type:

1. Type **:18,11**

2. Press **Return**.

   The following output appears:

```
                VME EEPROM set address and board type
                (also sets checksum)
          Flag is set for the diagnostics

slot      type          address    length    int function

***existing information about each board appears here***
***(this area will be blank if no boards are configured)***

          VME EEPROM Slot #:  1-16:
          Enter board type:

                    0-no board present
                    1-CME Ethernet Controller
                    2-Interphase SMD Controller
                    3-Xylogics 722 tape controller
                    4-Interphase V/Tape Controller
                    5-MPCC
                    Other (specify): 0-127

          Enter board address (hex):
          Enter board address space length (hex)
          Zero the init function address? (y/n)

          Setting EEPROM checksum
```

   If you want to delete an entry, enter board type 0 for the empty slot.

   If you are attaching SMD drives to your system, answer y to the init function prompt.

SUBTEST 12: VME EEPROM PRINT THE BOARD TYPES AND ADDRESSES

This subtest displays the current board types and addresses stored in the EEPROM.

To display the address and board type:

1. Type **:18,12**
2. Press **Return**.

   The following output appears:

```
                VME EEPROM set address and board type
                (also sets checksum)
          Flag is set for ctix

slot      type          address    length    int function

***existing information about each board appears here***
***(this area will be blank if no boards are configured)***
```

SUBTEST 13: VME EEPROM CLEAR THE ADDRESSES AND BOARD TYPES

Subtest 13 clears the board types and addresses stored in the EEPROM by writing zeros to the portion of the EEPROM storing this information. The contents of the EEPROM are saved before the write occurs. A valid checksum is generated after the write occurs.

To run subtest 13:

1. Type **:18,13**
2. Press **Return**.

   The following output appears:

```
                VME EEPROM clear the addresses and board
                types
          Saving EEPROM contents...done
          Setting EEPROM checksum
```

SUBTEST 14: VME EEPROM SET UP 'DEFAULT' TABLE OF BOARD TYPES

Subtest 14 enables you to set up the EEPROM board type table using the default list of board parameters. Subtest 11 can be used to change or delete information that does not apply to your system. This subtest will destroy the current contents of the EEPROM. Use subtest 15 to save the contents before running subtest 14.

To set the default board table in the EEPROM:

1. Type **:18,14**

2. Press **Return**.

   The following output appears:

   VME EEPROM set up 'default' table of board types
   This will destroy the current EEPROM contents!
   Enter y to confirm:

   After responding y, the following output appears:

   Setting EEPROM checksum
   Flag is set for ctix

slot type address length int function

existing information about each board appears here

SUBTEST 15: VME EEPROM SAVE CONTENTS ON EEPROM

This subtest saves the current contents of the VME EEPROM. Use subtest 16 to restore the contents saved with subtest 15.

To save the contents of the EEPROM:

1. Type **:18,15**
2. Press **Return**.

   The following output appears:

```
VME EEPROM save contents on eeprom
Saving EEPROM contents...done
```

You cannot run this subtest twice in a row. If you do, the following message will appear the second time:

```
EEPROM contents already saved, this is it:
slot      type          address    length    int function
***existing information about each board appears here***
```

SUBTEST 16: VME EEPROM RESTORE EEPROM TO STATE BEFORE EEPROM WRITE

Subtest 16 restores the contents of the EEPROM (saved by the write tests or by subtest 15) to what they were before the write occurred. If you have written to the EEPROM during a diagnostic session, you must run this subtest before ending the session.

To restore the contents of the EEPROM:

1. Type **:18,16**

2. Press **Return**.

   The following output appears:

```
VME EEPROM restore eeprom to state before
eeprom write
Restoring VME EEPROM contents...done
```

If the contents of the EEPROM were just restored or were never saved, the following prompt appears:

```
No EEPROM restore necessary: contents were never saved
```

SUBTEST 17: VME EEPROM CONFIRM CHECKSUM CORRECTNESS

Subtest 17 confirms the validity of the EEPROM checksum by reading the EEPROM contents, generating a checksum from that read, and comparing that checksum with the existing checksum.

To verify the EEPROM checksum:

1. Type **:18,17**
2. Press **Return**.

   The following output appears:

   VME EEPROM confirm checksum correctness

SUBTEST 20: SERIAL NUMBER ROM TEST

Subtest 20 can be run only on CPU-02 based systems.

This subtest verifies the serial number ROM of the CPU-02 board. First, the contents of the ROM are read. Then, the subtest tries to write to the ROM and, byte by byte, verifies that the contents have not changed. The subtest then prints out the entire contents of the ROM

To test the CPU-02 serial number ROM:

1. Type **:18,20**
2. Press **Return**.

   The following output appears:

   Serial number ROM test
   serialnumber; (12-digit hex number)