S/MT™ SERIES
DIAGNOSTICS MANUAL

NOTICE

Specifications Subject to Change.

Second Edition — November 1988 09-02110-01

WARNING

This equipment generates, uses, and can radiate radio frequency energy and if not installed and used in accordance with the instruction manual, may cause interference to radio communications. It 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 in which case the user at his own expense will be required to take whatever measures may be required to correct the interference.

S/MT Series Diagnostics Quick Reference

Loading the Diagnostics with CTIX Running

1. Shut down the operating system.
2. Insert the diagnostic tape in the tape drive when the following prompt appears:

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

3. Type **reboot,** then press **RETURN** (or simply press the reset switch).

Loading the Diagnostics with Power Off

1. Insert the diagnostic tape into the tape drive.

2. Turn on the power by pressing the power switch to the on position.

The diagnostics screen is displayed as follows:

```
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 slot X is configured for dma slot X
     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 >
```

Two-Letter Commands

The two-letter commands are shown in Table 1. A complete description of the commands is provided in Chapter 4, "Two-Letter Commands."

Table 1. Two-Letter Commands

Command	Description
General	
??	list available two\-letter commands
HE	help screen
PM	page mode
LE	line printer echo
Disk and Tape	
RC	repeat count
EC	ECC error correction enable/disable
SEx	disk soft error count
DD	display disk error statistics
CD	clear disk error statistics
IM	interrupt mode
Unit Selection	
DN	disk drive number
TN	tape drive number
EN	Ethernet Combo board number
MN	VME Communication Controller card number
System Mode	
RE	reboot
PE	NMI/parity enable
CA	cache enable
SPx	system priority level

Table 1. Two-Letter Commands (continued)

Command	Description
Display Status	
CN	system configuration
DE	display errors
DS	display test sequence subtests
User Selection	
CO	change console
# =	user-defined test sequence

Test Sequence Summary

Number refers to test by number and is used to run subtests. A letter can be used to run an entire test sequence. Table 2 describes the test sequence numbers and letters.

Table 2. Test Sequences

Number	Letter	Description
CPU Tests		
0	m	memory
1	p	page map (CPU-01 systems only)
2	o	physical memory
3	r	parity (CPU-01 systems only)
4	c	clock
10	a	Map RAM (CPU-01 systems only)
10	a	data cache and PTC RAM (CPU-02 systems only)
12	e	page protection
13	l	line printer
Disk and Tape		
6	w	hard disk drive tests
9	u	general disk tests
16	t	QIC tape drive tests (CPU-01 systems only)
20	j	general tape tests
I/O Processor Boards		
11	h	RS-232-C tests
5	i	IOP tests
15	n	Ethernet Combo board tests
14	k	RS-422 tests
VME		
17	v	VME subsystem (ethernet)
19	b	VME Communication Controller card tests
Miscellaneous		
18	z	miscellaneous tests

Table of Contents

S/MT Series Diagnostics Quick Reference

Chapter 1. Overview

Chapter 2. Getting Started

Chapter 3. Command Interpreter

Chapter 3. Command Interpreter (continued)

Chapter 4. Two-Letter Commands

Chapter 5. CPU-01 Diagnostic Tests

Chapter 5. CPU-01 Diagnostic Tests (continued)

Chapter 5. CPU-01 Diagnostic Tests (continued)

Chapter 6. CPU-02 Diagnostic Tests

Chapter 6. CPU-02 Diagnostic Tests (continued)

Chapter 6. CPU-02 Diagnostic Tests (continued)

Chapter 7. Disk and Tape Diagnostic Tests

Chapter 7. Disk and Tape Diagnostic Tests (continued)

Chapter 7. Disk and Tape Diagnostic Tests (continued)

Chapter 8. RS-232-C Channel Diagnostic Tests

Chapter 9. IOP Accelerator Board Diagnostic Tests

Chapter 9. IOP Accelerator Board Diagnostic Tests (continued)

Chapter 10. Ethernet Combo Board Diagnostic Tests

Chapter 10. Ethernet Combo Board Diagnostic Tests (continued)

Chapter 11. RS-422 Expansion Board Diagnostic Tests

Chapter 11. RS-422 Expansion Board Diagnostic Tests (continued)

Chapter 12. VME Subsystem Diagnostic Tests

Chapter 13. VME Communication Controller Board Diagnostic Tests

Chapter 13. VME Comunnication Controller Board Diagnostic Tests (continued)

Chapter 14. Miscellaneous Diagnostic Tests

Chapter 14. Miscellaneous Diagnostic Tests (continued)

Appendix A. Cable and Connector Specifications

Appendix A. Cable and Connector Specifications (continued)

Glossary

Index

Illustrations

Illustrations (continued)

Tables

Tables (continued)

Tables (continued)

CHAPTER 1

Overview

INTRODUCTION

The diagnostic tests 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 and each contain tests specific to the CPU-01 and CPU-02 versions of the Main Processor (CPU) board. Except where called out in the text, the tests for the rest of the system hardware are identical.

The diagnostics tapes are not interchangeable. The CPU-02 version of the diagnostics will not run on a CPU-01 version of the Main Processor board, and vice versa. Make sure you have the right tape for your system.

The installation manual 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 system)
- S/220 (a CPU-01 system)
- S/221 (a CPU-01 system)
- S/222 (a CPU-01 system)
- S/320 (a CPU-01 system)
- S/480 (a CPU-02 system)
- S/640 (a CPU-02 system)

Consult the current sales guide for a complete list and description of all S/MT Series computer systems and their Main Processor (CPU) boards.

ABOUT THIS MANUAL

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

Who Should Use This Manual

This manual is intended for use by service personnel who are familiar with the S/MT Series computer systems and are capable of performing diagnostic tests on computerized equipment. Each diagnostic test is described according to what it tests, the language required to execute the test, and what the test output means.

Purpose of This Manual

This purpose of this manual is to:

- Help you ensure proper computer operation.
- Show you how to identify problems when they occur.
- Refer you to related documentation.

Manual Organization

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

- Chapter 1, "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 and the optional parallel printer, and load the diagnostic programs from the QIC tape.
- Chapter 3, "Command Interpreter," describes the language required to execute the diagnostic programs.
- Chapter 4, "Two-Letter Commands," presents individual descriptions of two-letter commands.
- Chapter 5, "CPU-01 Diagnostic Tests," describes the diagnostic tests for systems using the CPU-01 version of the Main Processor (CPU) board.
- Chapter 6, "CPU-02 Diagnostic Tests," describes the diagnostic tests for systems using the CPU-02 version of the Main Processor (CPU) board.

- Chapter 7, "Disk and Tape Diagnostic Tests," contains hard disk and tape drive tests for systems using the CPU-01 and CPU-02 versions of the Main Processor (CPU) board.

- Chapter 8, "RS-232-C Diagnostic Tests," documents the diagnostic tests for the RS-232-C channels in your system.

- Chapter 9, "IOP Accelerator Diagnostic Tests," describes the tests for the Input/Output Processor (IOP) Accelerator Expansion board.

- Chapter 10, "Ethernet Combo Board Diagnostic Tests," describes the tests for the Ethernet Combo board.

- Chapter 11, "RS-422 Expansion Board Diagnostic Tests," documents the diagnostic tests for the RS-422 Expansion boards.

- Chapter 12, "VME Subsystem Diagnostic Tests," contains the subtests specific to the CMC VME board used to verify VME functions.

- Chapter 13, "VME Communication Controller Diagnostic Tests," documents the tests used to ensure the functionality of one or more VME Communication Controller cards.

- Chapter 14, "Miscellaneous Diagnostic Tests," includes tests for the MC68881 coprocessor, the Uninterruptible Power Supply (UPS), the VME Interface board, and some EEPROM tests.

- Appendix A, "Cable Specifications," contains pin assignments and information for making your own cables and the loopback connectors required for some diagnostic tests.

A glossary and appendix are also included.

How to Use This Manual

Each set of diagnostic tests is documented by hardware type and organized into chapters. Each chapter is separated and identified by a tabed page.

As you run a series of tests, refer to the specific chapter 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.

RELATED DOCUMENTATION

The documents described below provide additional information related to the S/MT Series diagnostic tests.

- **Installation**
  S/120 Installation
  S/220 Installation
  S/221 and S/222 Installation
  S/320 and S/640 Installation
  S/480 Installation

- **Hardware**
  S/220 and S/320 Hardware
  S/120 Addendum to the S/220 and S/320 Hardware Manual
  S/MT Series CPU-02 Technical Reference
  S/480 Technical Reference
  S/MT Series I/O Processor (IOP) Expansion Technical Reference
  S/MT Series RS-422 Expansion Technical Reference
  S/MT Series RS-232-C Expansion Technical Reference
  S/MT Series VME Communications Controller Card Technical Reference
  S/MT Series VME Expansion Technical Reference
  S/MT Series Ethernet Combo Board Technical Reference

- **System Administration**
  S/MT Series CTIX Administrator's Reference

The *S/120 Installation Manual* contains 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* contains 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* contains 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* contains 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/480 Installation Manual* contains proceudres for installing, powering on, and booting the S/480 system, and for connecting peripherals and adding expansion boards and hard disk drives.

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

The *S/MT Series CPU-02 Technical Reference Manual* contains a functional description and user interface definition for the 25MHz Main Processor (CPU) board.

The *S/480 Technical Reference Manual* contains a functional description, software interface, and theory of operation for the S/480 system.

The *S/MT Series I/O Processor (IOP) Expansion Board Technical Reference Manual* contains a functional and circuitry description of the Input/Output Processor board, and a theory of operation and software interface description.

The *S/MT Series RS-422 Expansion Technical Reference Manual* describes the hardware on the RS-422 Expansion board including the clock, bus, and channel control. An onboard line printer interface is also described.

The *S/MT Series RS-232-C Expansion Technical Reference Manual* describes the hardware incorporated on the 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 *S/MT Series VME Communications Controller Card Technical Reference Manual* contains a functional description and user interface for the VME Communications Controller card.

The *S/MT Series VME Expansion Technical Reference Manual* contains the functional and circuit description, theory of operation, and software interfaces for the VME Interface board.

The *S/MT Series Ethernet Combo Board Technical Reference Manual* contains the functional and circuitry description, and software interfaces for the Ethernet Combo board.

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

CHAPTER 2

Getting Started

INTRODUCTION

This chapter introduces the S/MT Series diagnostic tests. Procedures are included to connect a terminal and a parallel line printer to your system and to load the diagnostic tests. Also included is a brief description of how the tests operate.

The diagnostic tests are used to ensure correct system operation and to diagnose computer malfunctions when they occur. There are three levels 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 THE TERMINAL

Before you begin, make sure that you connect only *one* terminal to Channel 0 on the Main Processor (CPU) board. You must follow the directions in this section to ensure that the diagnostics will work properly.

You can use several types of terminals:

- Convergent Technologies TO-250 and TO-300
- Any terminal with asynchronous RS-232-C protocol, such as the Freedom 100 or VT-100 terminals
- A PT or GT-type terminal used in Emulate mode (see "Starting Up Terminals in Emulate Mode" later in this chapter)
- Any ASCII or ANSI compatible terminal

Consult the installation manual for the terminal you are using and make sure the terminal is configured to operate as follows:

- Baud rate – 9600
- Data bits – 8
- Stop bits – 1
- Parity – none

You need a crossed (twisted) 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. Attach the 25-pin connector on one end of the cable to Channel 0 on the Main Processor (CPU) board.

2. Attach the 25-pin connector at the other end of the cable to 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. To issue diagnostic commands from another port, refer to the two-letter command **CO** as described in Chapter 4, "Two-Letter Commands."

STARTING UP THE TERMINAL

Any RS-232-C terminal will operate as the console terminal for displaying the diagnostics command modes and subtests. However, if you use a PT or GT-type terminal, you need to start up the terminal in Emulate mode, which makes the system see the PT or GT as an RS-232-C terminal.

To start up a PT or GT terminal in Emulate mode:

1. Connect the PT or GT to Channel 0 on the Main Processor (CPU) board as shown in Figure 2-1.

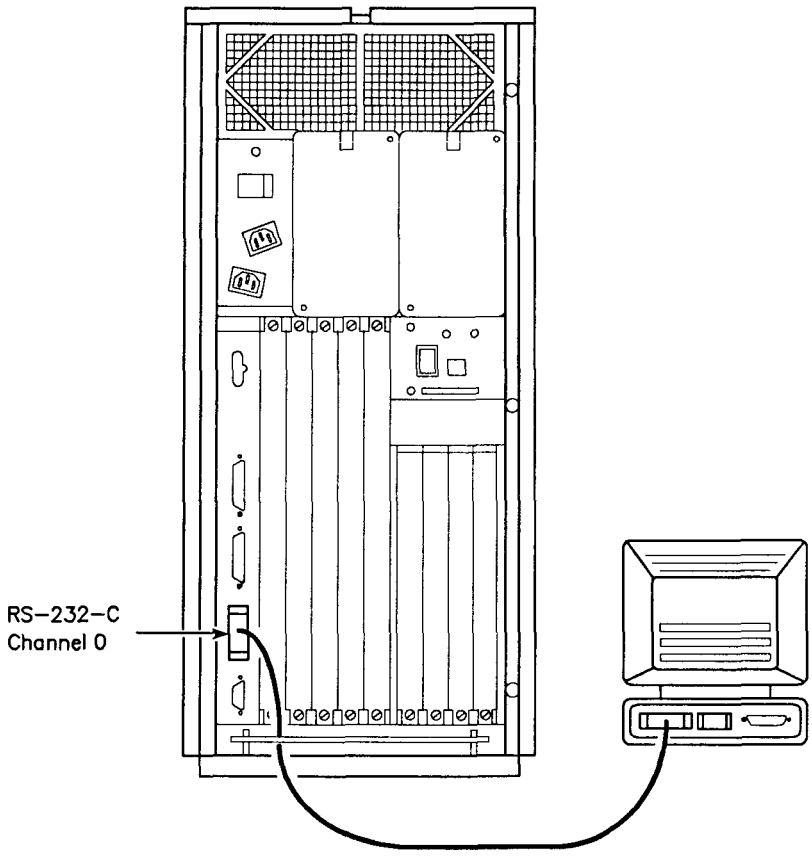


Figure 2-1. Connecting a Terminal to Channel 0

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 hold 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. Type **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.

There are some restrictions for PT and GT terminals that contain a version 1.0 boot ROM:

- 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:

  – Type / instead of ?
  – Type . instead of :
  – Type uppercase letters instead of lowercase letters.

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

You can still display the command modes and subtests when using a terminal containing a version 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 A PARALLEL PRINTER

A parallel line printer can be attached to the Parallel Printer connector on the Main Processor (CPU) board to print the output from the diagnostic tests. (You cannot connect a serial printer to the Main Processor board.)

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

Connect the parallel line printer as follows:

1. Attach the DB25 end of the Centronics-compatible parallel printer cable to the connector labeled Parallel Printer on the Main Processor (CPU) board.

2. Attach the connector at the other end of the cable to the connector on the parallel line printer as shown in Figure 2-2.

Note Load the diagnostic test program as described under "Loading the Diagnostic Tests." Then enable the parallel printer by performing the line printer echo test as described in Chapter 4.

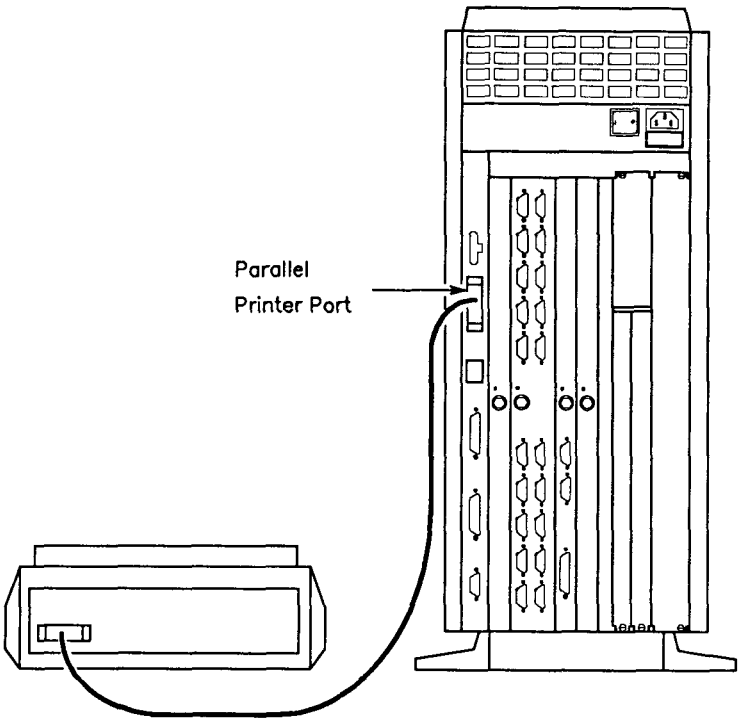


Figure 2-2. Attaching the Parallel Line Printer

LOADING THE DIAGNOSTIC TESTS

After you connect the terminal and, if necessary, the parallel line printer, you can load the diagnostic tests from the quarter-inch cartridge (QIC) tape. The diagnostic tests can be loaded when the system is running the CTIX operating system 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, "Command Interpreter") until an error occurs. If an error does occur, the error light-emitting diode (LED) on the status panel illuminates. The error message is displayed when you connect a terminal to Channel 0 on the Main Processor (CPU) board. Refer to the installation manual for your system for a summary of error and status codes. Refer to the hardware manual for your system for a complete description of the boot sequence.

With CTIX Running

When the CTIX operating system is running, you must terminate all currently running processes before loading the diagnostic tests. Refer to the *S/MT Series CTIX Administrator's Reference Manual* for instructions on properly shutting down the operating system.

Wait for the following prompt to appear on the screen:

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

Then load the diagnostics as follows:

1. Insert the diagnostic tape into the tape drive.
2. Type **reboot**
3. Press **RETURN**, or press down the Reset switch on the front of the Main Processor (CPU) board.

An example of the screen that is displayed when the diagnostics are fully loaded is given in the section "Diagnostic Screen."

With Power Off

To load the diagnostic tests when power is off:

1. Insert the diagnostic tape into the tape drive.

2. Lock the system in "reset" by placing the Reset switch on the front of the Main Processor (CPU) board in the up position.

3. Press the power switch on the rear of your system to the ON position.

4. If applicable, power on any external expansion devices connected to your system.

5. Press down the Reset switch on the front of the Main Processor (CPU) board.

The system powers up and loads the diagnostic tests into memory. An example of the screen that is displayed when the tests are fully loaded is given in the section "Diagnostic Screen."

DIAGNOSTIC SCREEN

The diagnostic screen looks like this:

```
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 slot X is configured for dma slot X
    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>
```

The diagnostic screen displays the following information:

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

- The diagnostic command prompt

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

DIAGNOSTIC TEST CONDITIONS

Listed below are several conditions to consider before 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 4KB 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, etc.) causes the diagnostic to be reset and the system configuration to be displayed. Use the escape character 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.

CHAPTER 3

Command Interpreter

INTRODUCTION

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 proper diagnostic software.

DIAGNOSTIC TEST MODES

Two test modes are available in the diagnostic program:

- Attended mode
  - Normal mode
  - Long error mode
  - Continuous error mode
- Unattended mode

Attended Mode

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

NORMAL ATTENDED MODE

In the normal attended mode, an error causes the diagnostic to reset 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 the escape character (ESC, GO, etc.) or typing **q**.

CONTINUOUS ERROR MODE

This error-reporting mode displays error messages on the terminal screen but 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 Terminal Ready (DTR) is asserted.)

Unattended mode causes the diagnostic to execute test sequence **d** until an error is encountered. The error causes the error LED on the status panel to flash, and the diagnostic waits for a terminal to be attached. Then the corresponding error message is displayed when a terminal is connected to Channel 0 on the Main Processor (CPU) 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 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 as follows:

1. Type **??**
2. Press **RETURN**.

Some of the two-letter commands are displayed as follows:

```
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, 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. Display the Help screen as follows:

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 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>
```

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 as follows:

1. Type **?**

2. Press **RETURN.**

Table 3-1 contains some of the test sequences in the diagnostic program.

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 and display the test sequence number. The following command lists the clock subtests and the clock test sequence number as follows:

1. Type **?c**

2. Press **RETURN.**

The output resembles the example below:

```
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 and subtest pairs contained in a selected test sequence. List the tests that make up system test sequence **d** as follows:

1. Type **DS**

2. Press **RETURN.**

This prompt is displayed:

```
Enter Test Sequence command letter:
```

3. Type **d**

4. Press **RETURN.**

The output resembles the following example:

```
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 as follows:

- **Repeat-Count**

  This *optional* parameter is a positive integer giving the number of times the entire test is to be executed. If no repeat count is given in the command, 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 a 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

Some examples of command structures that you can use to execute test sequences, tests, and subtests are described in this section. 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 as follows:

1. Type **m**
2. Press **RETURN.**

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

1. Type **m;p;a**
2. Press **RETURN.**

Execute any one, or all three, of the test sequences in continuous error mode as follows:

1. Type **5|mC;pC;aC**
2. Press **RETURN.**

You can also execute each test several times by typing the number of repetitions before the command-letter as follows:

1. Type **4|mC;8pC;2aC**

2. Press **RETURN.**

Execute subtest 5 of memory test sequence 0 as shown:

1. Type **:0,5**

2. Press **RETURN.**

Execute subtest 5 of memory sequence 10 times in Continuous mode as follows:

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

2. Press **RETURN.**

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 as follows:

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

2. Press **RETURN.**

Execute that list of subtests 10 times as indicated:

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

2. Press **RETURN**.

Execute that list of subtests 10 times in Continuous mode as follows:

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

2. 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 as shown:

5|mL;pL;aL

Instead, use the two-letter command **#=** to create your own test sequence, which you can then execute in long mode (**#L**). Refer to Chapter 4, "Two-Letter Commands," for more information.

HELPFUL HINTS

Here are some hints to consider before continuing:

- When you initiate a diagnostic session, the system configuration is displayed on the terminal screen. Verify that a piece of equipment is properly connected if the diagnostics fails to recognize equipment you believe is attached to the system.

  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 continues 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.

CHAPTER 4

Two-Letter Commands

INTRODUCTION

This chapter describes the two-letter commands that perform functions and establish the environment for the diagnostic tests.

DISPLAY TWO-LETTER COMMANDS

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.

Display the list of two-letter commands as follows:

1. Type **??**
2. Press **RETURN**.

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

1. Type **HE**
2. Press **RETURN**.

Help (HE)

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

Display the online help screen as follows:

1. Type **HE**

2. Press **RETURN**.

The following 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 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 COMMANDS

This section describes the page mode and line printer echo commands.

Page Mode (PM)

By default, page mode is disabled so that the output from the diagnostic tests is continuously scrolled through the terminal screen. 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, page mode is more useful.

Enable page mode as follows:

1. Type **PM**
2. Press **RETURN**.

The following 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.

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

You must press one of the three letters indicated on the screen to continue.

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.

Two ways to disable page mode are:

- Type **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 Parallel Printer connector on the Main Processor (CPU) board (not to a printer connector on an expansion board). To connect the line printer, refer to Chapter 2.

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.

Enable line printer echo (use the same steps to disable line printer echo) as follows:

1. Type **LE**

2. Press **RETURN**.

The following 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.
```

DISK AND TAPE COMMANDS

This section discusses disk and tape commands. Commands included in this section are:

- Repeat count
- Error correction
- Disk soft error count
- Display disk error statistics
- Clear disk error statistics
- Interrupt mode.

Repeat Count (RC)

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

You can set the number of retries by entering a number of your choice or by cycling 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.

Select the number of retries yourself (for example, 20) as shown:

1. Type **RC20** (or **RC 20**).

2. Press **RETURN**.

The following message is displayed:

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

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

Alternately, you can cycle through the three settings listed previously as follows:

1. Type **RC**

2. Press **RETURN**.

   The following message is displayed:

```
Perform transfer but no retries
```

   The message 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.

3. Type **RC**

4. Press **RETURN**.

   The following 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.

Error Correction Code Enable/Disable (EC)

This two-letter command enables you to change the ability of a disk drive to do error correction code (ECC) correction but does not change the format of the disk. When the default is disabled, it forces the controller to report the error and creates a more stringent test.

Enable ECC correction as indicated:

1. Type **EC**
2. Press **RETURN**.

The following message is displayed.

```
ECC correction is enabled.
```

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 and SCSI 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).

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 stops on an error. The default value is 0.

Determine the current number of soft errors allowed as follows:

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

The following message is displayed.

```
x ECC correctable errors allowed.
```

Set a limit on the number of soft errors as shown:

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. Command **DD** enables you to look at this information any time.

Display the disk error log as follows:

1. Type **DD**

2. Press **RETURN**.

The contents of the disk error log are displayed.

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

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

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 without displaying a screen message.

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.

Disable and enable interrupt mode as follows:

1. Type **IM**
2. Press **RETURN**.

The following message is displayed:

```
Disk/QIC Interrupt Mode disabled
```

UNIT SELECTION COMMANDS

This section describes unit selection commands which include: disk drive identification (ID) number, tape drive ID number, ethernet combo ID number, and VME communication controller card ID number.

Disk Drive Number (DN)

This two-letter command enables you to select the 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 expansion X-Box or V-Box devices.

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.

Determine the drive presently selected as follows:

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

The correlation between SCSI ID numbers and diagnostic ID numbers is shown in Table 4-1.

Table 4-1. Drive Identification Numbers

SCSI ID	Diagnostic ID	Usage
0	64	unused
1	68	always QIC tape
2	72	unused
3	76	unused
4	80	SCSI drive 2*
5	84	SCSI drive 1*
6	88	SCSI drive 0*
7		system

* Non-SCSI drive ID numbers will depend upon your unit configuration Refer to the /etc/system file SCSI map in your system.

There are two ways to select a drive ID 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.

Select a disk drive (for example, the drive number D2) as follows:

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

2. Press **RETURN**.

The following message is displayed:

```
Disk Drive 2 is selected.
```

Alternately, you can issue the two-letter command without specifying a number and cycle the diagnostic through all possible drive numbers as follows:

1. Type **DN**

2. Press **RETURN**.

   The following message is displayed:

```
Disk Drive 1 is selected.
```

3. Type **DN**

4. Press **RETURN**.

The following 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 lets you select the SCSI tape drive (CPU-02 version systems only) or half-inch tape drive from which to execute the tape drive tests.

Determine the tape drive presently selected as indicated:

1. Type **TN?**

2. Press **RETURN**.

Refer to Table 4-1 to determine the correlation between SCSI ID numbers and diagnostic ID numbers.

Each controller can support up to eight tape drives. The controller on the Main Processor (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, etc.

To determine the tape drive ID number, use the following formula:

TN = (controller# * 8) + unit#

Select a tape drive (for example, the tape drive 2) as follows:

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

2. Press **RETURN**.

The following message is displayed:

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

You can also issue the two-letter command without specifying a number, and cycle the diagnostic through all possible tape drive ID numbers.

1. Type **TN**

2. Press **RETURN**.

   The following message is displayed:

```
Tape Drive 1 is selected.
```

3. Type **TN**

4. Press **RETURN**.

   The following 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 so that you can run tests.

Select an Ethernet Combo board (for example, board number 2) as follows:

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

2. Press **RETURN**.

The following message is displayed:

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

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

VME Communication Controller Board Number (MN)

This two-letter command enables you to select the VME Communications Controller board (also called the MPCC board) so that you can run tests.

Select a VME Communication Controller board (for example, board number 2) as follows:

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

The following message is displayed:

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

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

SYSTEM MODE COMMANDS

This section describes system mode commands which include: reboot, NMI/parity enable, cache enable, and system priority level.

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.

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 as follows:

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.

All three interrupts are enabled when the diagnostic is loaded to detect errors as soon as possible. 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.

Disable NMIs and parity errors as follows:

1. Type **PE**

2. Press **RETURN**.

The following 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 CPU on the Main Processor (CPU) board contains an on-chip, instruction-only cache memory that improves the performance of the CPU. The cache increases execution efficiency by providing a quick-store for instructions.

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

Disable the cache as indicated:

1. Type **CA**
2. Press **RETURN.**

The following 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 CPU. 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 4-2. You can only set the interrupt levels to integers 0 through 7.

Change the interrupt level to 2 in this manner:

Table 4-2. System Priority Levels (SPLs)

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

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 the test.

Display the system priority level as follows:

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

2. Press **RETURN**.

DISPLAY STATUS COMMANDS

This section describes display status commands which include system configuration and display test sequence subtests.

System Configuration (CN)

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

Display the system configuration as follows:

1. Type **CN**
2. Press **RETURN**.

The system configuration is displayed on the terminal screen and resembles the following:

```
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 are present but contain invalid Volume Home Blocks (VHBs).

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.

Display the error log as follows:

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, list the tests that make up system test sequence **d** in this manner:

1. Type **DS**
2. Press **RETURN**.

   The following 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 COMMANDS

This section describes user selection commands which include change console and user-defined test sequence.

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.

Change consoles as follows:

1. Type **CO**

2. Press **RETURN**.

The following 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:
```

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 terminal to the console terminal 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.

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!
```

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

Create your own test sequence as follows:

1. Type **#=**
2. Press **RETURN**.

   The following 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**.

   The following 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.

Execute the new test sequence in this manner:

1. Type #
2. Press **RETURN**.

CHAPTER 5

CPU-01 Diagnostic Tests

INTRODUCTION

This chapter describes the diagnostic tests for the CPU-01 version of the Main Processor (CPU) board.

The CPU-01 version of the Main Processor (CPU) board can be distinguished from the CPU-02 version by the location of the Uninterruptible Power Supply (UPS) connector (see Figure 5-1).

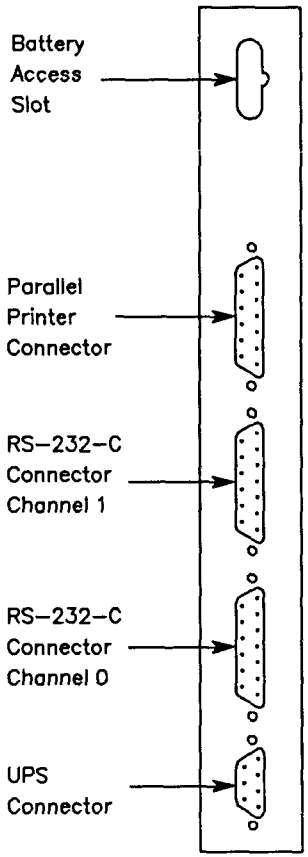


Figure 5-1. Main Processor Board (CPU-01 Version)

Table 5-1 describes the CPU-01 diagnostic tests.

Table 5-1. 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

TEST 0: MEMORY (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 5-2 contains a summary of memory subtests.

Table 5-2. Memory Subtests

Subtest	Description
1	Tests logic control to each 1MB bank of memory
2	Writes/reads 0x00000000s (32-bit)
3	Writes/reads 0xFFFFFFFFs (32-bit)
4	Writes/reads 0xAAAAAAAAs (32-bit)
5	Writes/reads 0x54545454s (32-bit)
6	Writes/reads addresses (32-bit)
7	Writes/reads address complements (32-bit)
8	Writes/reads 0x0000s (16-bit)
9	Writes/reads 0xFFFFs (16-bit)
10	Writes/reads 0xAAAAs (16-bit)
11	Writes/reads 0x5454s (16-bit)
12	Writes/reads addresses (16-bit)
13	Writes/reads address complements (16-bit)
15	Sets the memory limits for the above subtests
17	Swaps the top half of memory with the bottom half

Table 5-2. Memory Subtests (continued)

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	2731
Compare blocks			32

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.

Note that 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

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. These tests each consist of two passes, which are:

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

Subtests 2 and 13 delay approximately 15 seconds between passes to detect refresh errors. To prevent memory access, the delay is executed by the 68020 instruction cache, which is enabled. Subtests 2 through 13 are described in Table 5-3.

Table 5-3. Memory Subtests 2 Through 13

Subtest	Purpose
2,8	Writes and reads 0s to every location in memory in 32-bit and 16-bit modes, respectively.
3,9	Writes and reads fs in hexadecimal format to every location in memory in 32-bit and 16-bit modes, respectively.
4,10	Writes and reads as in hexadecimal format to every location in memory in 32-bit and 16-bit modes, respectively.
5,11	Writes and reads 54s 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	Writes and reads the addresses of each location in memory in 32-bit and 16-bit modes, respectively.
7,13	Writes and reads the one's complements of addresses to every location in memory in 32-bit and 16-bit modes, respectively.

Execute subtests 1 through 13 as follows:

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
```

Execute one memory subtest, for example subtest 3, as follows:

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

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 14.

- 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 purpose of subtests 20 through 32 are described in Table 5-4.

Table 5-4. Memory Subtests 20 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

REQUIRED INPUT

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

Table 5-5. Input for Memory Subtests 15, 20–32

Type of Input	Subtests
Starting address	15 and 20 through 32
Ending address	15, 22, 23, 26, 27, 30, 31, 32
Data pattern	20, 22, 24, 26, 28, 30
Destination address	23, 27, 31
Number of loops	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 (0x7C000, end at 400000).

Hex Starting Address

The hexadecimal starting address is 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 hexadecimal ending address is 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            = >
```

Address Block to Compare

The address block to compare is the starting hexadecimal address of the block of memory between the hexadecimal starting address and the hexadecimal ending address which is to be compared.

```
Enter address block to compare      = >
```

Hex Data Pattern

The hexadecimal data pattern is 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 hexadecimal destination address is 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 decimal number of loops is the number of times you want to execute the subtest, according to the parameters established above. The prompt looks like this:

```
Enter 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

For the first option, type the subtest number next to the prompt to execute another subtest. To exit the menu, press **RETURN**.

For the second option, a table displays the interactive subtests. 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.

Execute subtest 15 as follows:

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

The following output is displayed:

```
MEMORY TEST
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 2 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.

Execute subtest 17 separately (it is automatically executed by test sequence **m**) as follows:

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. Execute one of these three subtests as follows:

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

2. Press **RETURN**. These prompts are displayed. Enter data at 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. Execute one of these three subtests as follows:

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

2. Press **RETURN.**

   The following 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.**

The following 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. Execute one of these three subtests as follows:

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

2. Press **RETURN.**

   The following 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.**

The following 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.

Execute one of these three subtests as follows:

1. Type **:0,** *subtest number*
2. Press **RETURN**.

   The following 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.**

The following 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. Execute subtest 32 as follows:

1. Type **:0,32**

2. Press **RETURN.**

   The following 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.**

The following 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 (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:

- Writes addresses to two locations on two different pages in memory
- Swaps the page table entries for the two entries
- Reads the entries to verify that the values were swapped

Execute the Page Map test as follows:

1. Type **p**
2. Press **RETURN**.

The following 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 (o)

The physical memory map enables the CTIX operating system to track the access history of each physical page. Tracking allows the operating system 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.

The physical memory map:

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 nonmaskable interrupts (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.

Execute the physical memory map test sequence (subtests 1 through 3) as follows:

1. Type **o**

2. Press **RETURN**.

Subtests 1 through 3 are confirmed as follows:

```
PHYSICAL MEMORY MAP TESTS
PHYSICAL MEMORY MAP Subtest 1
PHYSICAL MEMORY MAP Subtest 2
PHYSICAL MEMORY MAP Subtest 3
```

Subtest 1: Test All Bit Combinations in 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 in one map register.

Execute subtest 1 as follows:

1. Type **:2,1**

2. Press **RETURN**.

The following 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: Test Access Register Bits

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.

Execute subtest 2 as follows:

1. Type **:2,2**
2. Press **RETURN**.

The following output confirms successful test completion:

```
PHYSICAL MEMORY MAP TESTS
PHYSICAL MEMORY MAP SUBTEST 2
```

This error message is displayed when subtest 2 fails:

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

Subtest 3: Walk 0 and 1 Through Modified Bits of Access Registers

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.

Execute subtest 3 as follows:

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

The following output confirms successful test completion:

```
PHYSICAL MEMORY MAP TESTS
PHYSICAL MEMORY MAP SUBTEST 3
```

This error message is displayed when subtest 3 fails:

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

TEST 3: PARITY (r)

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

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

These subtests are described on the following pages.

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

Execute the parity test sequence (subtests 1 through 3) as follows:

1. Type **r**

2. Press **RETURN**.

The following 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: Test GSR Parity Error Bits

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 1MB block of physical memory. Subtest 1 does not destroy memory. Prior to this test, the contents of memory are saved.

Execute subtest 1 as follows:

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

The following 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: Test All Available GSR Physical Page Numbers

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.

Execute subtest 2 as follows:

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

The following 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 Program with 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.

Execute subtest 3 as follows:

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

If a parity error occurs, the subtest is successful and the following 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 (c)

The eight clock subtests are listed in Table 5-6.

Table 5-6. Clock Subtests

Subtest	Description
1	Perform Data Path Check to 8259 and 8253
2	Tests Timer 0
3	Tests Timer 1
4	Tests Timer 2
5	Tests clock calendar
6	Sets the clock calendar
7	Reads the data and time from the clock calendar
8	Tests special frequency calibration mode

Execute the clock test sequence (subtests 1 through 8) as follows:

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 and 8253

Execute subtest 1 as follows:

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

The following output is displayed:

```
CLOCK TEST
Data Path Check
```

Subtest 2: Test Timer 0

Execute subtest 2 as follows:

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

The following output is displayed:

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

Subtest 3: Test Timer 1

Execute subtest 3 as follows:

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

The following output is displayed:

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

Subtest 4: Test Timer 2

Execute subtest 4 as follows:

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

The following output is displayed:

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

Subtest 5: Test Clock Calendar

Execute subtest 5 as follows :

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

The following 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 Clock Calendar

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

Execute subtest 6 as follows:

1. Type **:4,6**
2. Press **RETURN.**

The following 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 Date and Time

Execute subtest 7 as follows:

1. Type **:4,7**

2. Press **RETURN**.

The following 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

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

Execute subtest 8 as follows:

1. Type **:4,8**

2. Press **RETURN**.

The following 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 (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 5s to the static RAMs instead of the 54s that the memory subtests write to the dynamic RAMs. The map RAM subtests are listed in Table 5-7.

Table 5-7. Map RAM Subtests

Subtest	Description
1	Writes/reads 0x00000000s
2	Writes/reads 0xFFFFFFFFs
3	Writes/reads 0xAAAAAAAAs
4	Writes/reads 0x55555555s
5	Writes/reads addresses
6	Writes/reads 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 mode) 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.

Execute the entire test sequence as follows:

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 0x00000000s

Execute subtest 1 as follows:

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 0xFFFFFFFFs

Execute subtest 2 as follows:

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 0xAAAAAAAAs

Execute subtest 3 as follows:

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 0x55555555s

Execute subtest 4 as follows:

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

Execute subtest 5 as follows:

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

Execute subtest 6 as follows:

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 (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. Refer to 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 2GB. The two page protection subtests are presented in Table 5-8.

Table 5-8. 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.

Execute the two page protection subtests as follows:

1. Type **e**
2. Press **RETURN**.

The following 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: Check 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
- Operating system read/write
- User read
- User read/write

Execute subtest 1 as follows:

1. Type **:12,1**

2. Press **RETURN**.

The diagnostic displays the following 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: Read and Write to Mapped RAM and Unused I/O Space

This subtest reads and writes data in User mode to every page in the two virtual memory regions. Then, it tests the Unused memory region in both User and Supervisor modes. 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 1MB. User access to all VME regions is denied during this part of subtest 2.

Execute subtest 2 as follows:

1. Type **:12,2**

2. Press **RETURN**.

Failure of subtest 2 can generate one of the following 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 (l)

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 5-9.

Table 5-9. Line Printer Subtests

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

Execute the test sequence as follows:

1. Type **l** (lowercase **L**).
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 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).

Execute subtest 1 as follows:

1. Type **:13,1**
2. Press **RETURN**.

Output like the following is displayed as the subtest is executed:

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

Subtest 2: Print and Check Rotating Pattern on Printer Using Interrupts

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

Execute subtest 2 as follows:

1. Type **:13,2**
2. Press **RETURN**.

Output like the following is displayed as the subtest is executed:

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

Subtest 3: Check Printer Connector Control Lines

Execute subtest 3 as follows:

1. Type **:13,3**
2. Press **RETURN**.

Output like the following is displayed as the subtest is executed:

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

Subtest 4: Check/Display IOP or RS-422 Line Printer Status Register

Execute subtest 4 as follows:

1. Type **:13,4**
2. Press **RETURN**.

Output like the following 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: Check Interrupt Status

Execute subtest 5 as follows:

1. Type **:13,5**
2. Press **RETURN**.

Output like the following 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: Check IOP or RS-422 Printer Connector Control Lines

Execute subtest 6 as follows:

1. Type **:13,6**

2. Press **RETURN**.

Output like the following 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
```

CHAPTER 6

CPU-02 Diagnostic Tests

INTRODUCTION

This chapter describes the diagnostic tests for the CPU-02 version of the Main Processor (CPU) board.

The CPU-02 version of the Main Processor (CPU) board can be distinguished from the CPU-01 version by the location of the Uninterruptible Power Supply (UPS) connector (see Figure 6-1).

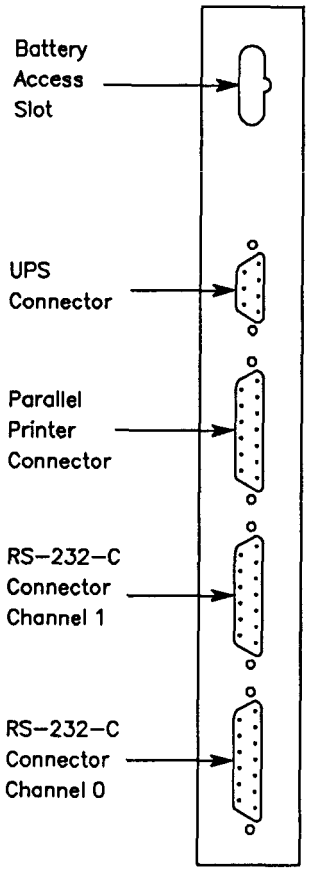


Figure 6-1. Main Processor Board (CPU-02 Version)

Table 6-1 describes the CPU-02 diagnostic tests.

Table 6-1. 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

TEST 0: MEMORY (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 6-2 contains a summary of memory subtests.

Table 6-2. Memory Subtests

Subtest	Description
1	Tests logic control to each 2MB bank of memory
2	Writes/reads 0x00000000s (32-bit)
3	Writes/reads 0xFFFFFFFFs (32-bit)
4	Writes/reads 0xAAAAAAAAs (32-bit)
5	Writes/reads 0x55555555s (32-bit)
6	Writes/reads addresses (32-bit)
7	Writes/reads address complements (32-bit)
8	Writes/reads addresses
9	Writes/reads address complements
10	Writes/reads 0x00000000s
11	Writes/reads 0xFFFFFFFFs
12	Writes/reads 0xAAAAAAAAs
13	Writes/reads 0x55555555s
14	Performs memory test on check bits (addr, ~addr)
15	Sets memory limits for above subtests
16	Performs 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

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.

Noninteractive Memory 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. Each test consists 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 by the 68020 instruction cache which is enabled. Subtests 2 through 7 are described in Table 6-3.

Table 6-3. Memory Subtests 2 Through 7

Subtest	Purpose
2	Writes and reads 0s in hexadecimal format to every location in memory in 32-bit mode.
3	Writes and reads Fs in hexadecimal format to every location in memory in 32-bit mode.
4	Writes and reads as in hexadecimal format to every location in memory in 32-bit mode.
5	Writes and reads 0x55s in hexadecimal format to every location in memory in 32-bit mode.
6	Writes and reads the addresses of each location in memory in 32-bit mode.
7	Writes and reads the one's complements of addresses to every location in memory in 32-bit mode.

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 error-correction code (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 XOR 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 XOR 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.

Execute subtests 1 through 13 as follows:

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
```

Execute one memory subtest, for example subtest 3, as follows:

1. Type **:0,3**

2. Press **RETURN.**

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

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 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 listed in Table 6-4.

Table 6-4. Memory Subtests 15 and 20-32

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

REQUIRED INPUT

Table 6-5 summarizes the input you must provide to execute subtests 15 and 20 through 32.

Table 6-5. Input for Memory Tests 15 and 20-32

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

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 (0x7C000, end at 400000).

Hex Starting Address

The hexadecimal starting address is 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 hexadecimal ending address is 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            = >
```

Address Block to Compare

The address block to compare is the starting hexadecimal address of the block of memory between the hexadecimal starting address and the hexadecimal ending address which is to be compared.

```
Enter address block to compare          = >
```

Hex Data Pattern

The hexadecimal data pattern is 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 hexadecimal destination address is 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 decimal number of loops is the number of times you want to execute the subtest, according to the parameters established above. The prompt looks like this:

```
Enter 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 these two options:

- Execute another interactive subtest
- Exit the Interactive Subtest menu

To execute the first option, type the subtest number next to the prompt. Otherwise, press **RETURN** to exit the menu.

The Interactive Subtest menu presents a table of the interactive subtests. 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.

Execute subtest 15 in this manner:

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**.

   The following prompt is displayed:

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

5. Type the ending hexadecimal address.

6. Press **RETURN**.

   The following 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.

Execute one of these three subtests as follows:

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

2. Press **RETURN**.

   The following 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. Execute one of these three subtests as follows:

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. Execute one of these three subtests as follows:

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.

Execute one of these three subtests in this manner:

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.**

The following 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. Execute subtest 32 as follows:

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 (o)

The physical memory map allows the CTIX operating system to track the access history of each physical page. Tracking allows the operating system 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 doing 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 nonmaskable interrupts (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.

Execute the physical memory map test sequence (subtests 1 through 3) in this manner:

1. Type **o**

2. Press **RETURN**.

Subtests 1 through 3 are confirmed as follows:

```
PHYSICAL MEMORY MAP TESTS
PHYSICAL MEMORY MAP Subtest 1
PHYSICAL MEMORY MAP Subtest 2
PHYSICAL MEMORY MAP Subtest 3
```

Subtest 1: Test All Bit Combinations in One Access Register

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

Execute subtest 1 as follows:

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: Test Access Register Bits

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.

Execute subtest 2 as indicated:

1. Type **:2,2**

2. Press **RETURN.**

This output confirms successful test completion:

```
PHYSICAL MEMORY MAP TESTS
PHYSICAL MEMORY MAP SUBTEST 2
```

This error message is displayed when subtest 2 fails:

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

Subtest 3: Walk 0 and 1 Through Modified Bits of Access Registers

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.

Execute subtest 3 as follows:

1. Type **:2,3**

2. Press **RETURN**.

This output confirms successful test completion:

```
PHYSICAL MEMORY MAP TESTS
PHYSICAL MEMORY MAP SUBTEST 3
```

This error message is displayed when subtest 3 fails:

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

TEST 4: CLOCK (c)

The eight clock subtests are listed in Table 6-6.

Table 6-6. Clock Subtests

Subtest	Description
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

Execute the clock test sequence (subtests 1 through 8) as follows:

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 and 8253

Execute subtest 1 in this manner:

1. Type **:4,1**

2. Press **RETURN**.

This output is displayed:

```
CLOCK TEST
Data Path Check
```

Subtest 2: Test Timer 0

Execute subtest 2 as follows:

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

This output is displayed:

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

Subtest 3: Test Timer 1

Execute subtest 3 as follows:

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

Execute subtest 4 as follows:

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

This output is displayed:

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

Subtest 5: Test Clock Calendar

Execute subtest 5 as indicated:

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 Clock Calendar

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

Execute subtest 6 as follows:

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 Date and Time

Execute subtest 7 as indicated:

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

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

Execute subtest 8 as follows:

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 6-7.

Table 6-7. Data Cache and Page Table Cache Subtests

Subtest	Description
1	Tests data cache memory
2	Tests data cache tag memory
3	Tests DMA map memory
4	Tests group table memory
5	Tests page table cache memory
6	Tests page table cache tag memory
7	Forces NMI on PTC fill
8	Tests page table cache fill
9	Tests pointer register
10	Tests indirect pointer register
11	Tests data cache memory test through virtual space: 0x100000 - 0x120000

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

- Walking 1s
- Walking 0s
- All 0s
- All Fs
- All 55s
- All As
- 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 by 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 is approximately 16 seconds. When each subtest is run individually, you are prompted to enter in a delay count in hexidecimal. The default, 0x1000000, corresponds to approximately 16 seconds and is selected by pressing **RETURN**. Each count adds approximately 720 nanoseconds of delay time.

Execute the entire test sequence as follows (note that the indirect pointer register tests, subtest 10, and the data cache memory test through virtual space, subtest 11, are not executed at this time, but they can be executed individually).

1. Type **a**

2. Press **RETURN**.

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

```
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.
```

Subtest 1: Data Cache Memory

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.

Execute subtest 1 as follows:

1. Type **:10,1**

2. Press **RETURN**.

The following output appears on the screen:

```
Data cache memory tests.
Enter delay count: (hex) (default: 0x1000000)
```

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

Subtest 2: Data Cache Tag Memory

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.

Execute subtest 2 in this manner:

1. Type **:10,2**

2. Press **RETURN**.

The following output appears on the screen:

```
Data cache tag memory test.
Enter delay count: (hex) (default: 0x1000000)
```

There is a delay after the write pass to detect refresh errors. When running these subtests as part of the test sequence (for example, test sequence **a**), the delay is approximately 16 seconds. When each subtest is run individually, you are prompted to enter in a delay count in hexidecimal. The default, 0x1000000, corresponds to approximately 16 seconds and is selected by pressing **RETURN**. Each count adds approximately 720 nanoseconds of delay time.

Subtest 3: DMA Map Memory

Execute subtest 3 as follows:

1. Type **:10,3**
2. Press **RETURN**.

The following output appears on the screen:

```
DMA map memory test.
```

Subtest 4: Group Table Memory

Execute subtest 4 as follows:

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

The following output appears on the screen:

```
Group table memory test.
```

Subtest 5: Page Table Cache Memory

Execute subtest 5 as follows:

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

The following output appears on the screen:

```
Page table cache memory test.
```

Subtest 6: Page Table Cache Tag Memory

Execute subtest 6 as follows:

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

The following output appears on the screen:

```
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.

Execute subtest 7 as follows:

1. Type **:10,7**

2. Press **RETURN.**

The following output appears on the screen:

```
Force NMI on PTC fill.
```

Subtest 8: Page Table Cache Fill

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 1s test across the data bus, and then the address bus.

Execute subtest 8 as follows:

1. Type **:10,8**

2. Press **RETURN.**

The following output appears on the screen:

```
Page Table Cache fill test.
```

Subtest 9: Pointer Register

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. Refer to the *S/MT Series CPU-02 Technical Reference Manual* for more information.)

Execute subtest 9 as follows:

1. Type **:10,9**

2. Press **RETURN.**

The following output appears on the screen:

```
Pointer register tests
```

Subtest 10: Indirect Pointer Register

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, the Expansion board is used as the DMA device. (Channels 0 and 1 must be connected with a RS-422 loopback cable. Refer to Chapter 11, "RS-422 Expansion Board Diagnostic Tests" for more information.) If an RS-422 Expansion board is not installed, the currently selected tape drive is used as the DMA device. (Be sure that a tape is loaded in the tape drive.)

Execute subtest 10 as follows:

1. Type **:10,10**
2. Press **RETURN**.

The following output appears on the screen:

```
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 tape drive.

Subtest 11: Data Cache Memory Through Virtual Space: 0x100000) – 0x120000

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

Execute subtest 11 as follows:

1. Type **:10,11**

2. Press **RETURN**.

The following output appears on the screen:

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

There is a delay after the write pass to detect refresh errors. When running these subtests as part of the test sequence (for example, test sequence **a**), the delay is approximately 16 seconds. When each subtest is run individually, you are prompted to enter in a delay count in hexadecimal. The default, 0x1000000, corresponds to approximately 16 seconds and is selected by pressing **RETURN**. Each count adds approximately 720 nanoseconds of delay time.

TEST 12: PAGE PROTECTION (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 2 gigabytes. The two page protection subtests are described in Table 6-8.

Table 6-8. 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 the I/O space.

Execute the two page protection subtests as follows:

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: Check 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 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
- Operating system read/write
- User read
- User read/write

Execute subtest 1 as follows:

1. Type **:12,1**

2. Press **RETURN**.

The diagnostic displays the following 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: Read and Write to Virtual Memory and 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 1MB. User access to all VME regions is denied during this part of subtest 2.

Execute subtest 2 as follows:

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 (l)

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 eight subtests described in Table 6-9.

Table 6-9. Line Printer Subtests

Subtest	Description
1	Checks and displays the line printer status register
2	Prints a rotating pattern on the printer using interrupts
3	Checks the control lines of the printer connector with a loopback plug
4	Same as subtest 1 for the line printer connector on the IOP or RS-422 Expansion board
5	Same as subtest 2 for the line printer connector on the IOP or RS-422 Expansion board
6	Same as subtest 3 (requires a loopback plug) for the line printer connector on the IOP or RS-422 Expansion board
7	Tests the Bidirectional Printer Port in Master Mode
8	Tests the Bidirectional Printer Port in Slave Mode

Execute the test sequence as follows:

1. Type **l** (lowercase **L**).

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 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).

Execute subtest 1 as follows:

1. Type **:13,1**

2. Press **RETURN**.

Output like the following is displayed as the subtest is executed:

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

Subtest 2: Print and Check Rotating Pattern on Printer Using Interrupts

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

Execute subtest 2 as follows:

1. Type **:13,2**
2. Press **RETURN**.

Output like the following is displayed as the subtest is executed:

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

Subtest 3: Check Control Lines of Printer Connector

Execute subtest 3 as follows:

1. Type **:13,3**

2. Press **RETURN.**

Output like the following is displayed as the subtest is executed:

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

Subtest 4: Check/Display IOP or RS-422 Line Printer Status Register

Execute subtest 4 as follows:

1. Type **:13,4**

2. Press **RETURN.**

Output like the following 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: Check Interrupt Status

Execute subtest 5 as follows:

1. Type **:13,5**
2. Press **RETURN**.

Output like the following 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: Check IOP or RS-422 Printer Connector Control Lines

Execute subtest 6 as follows:

1. Type **:13,6**
2. Press **RETURN**.

Output like the following 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

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

Begin with the slave system, and execute the subtest as follows:

1. Type **:13,8**
2. Press **RETURN**.

Then start the master system, and execute the subtest as follows:

1. Type **:13,7**
2. Press **RETURN**.

The two systems proceed to communicate with each other. First the master will transmit a byte of data and the slave reads it. Then the slave transmits the one's complement of the data back to the master. This interaction between the master and slave continues through a walking 1s pattern for the data.

CHAPTER 7

Disk and Tape Diagnostic Tests

INTRODUCTION

This chapter describes the hard disk and tape drive tests listed in Table 7-1.

Table 7-1. Storage Media Diagnostic Tests

Number	Letter	Description
6	w	Tests hard disk drives
9	u	General Disk Test
16	t	Tests QIC tape drive (CPU-01 systems only)
20	j	General Tape Test

Several drive controllers are discussed in this chapter. Make sure that you refer to the appropriate section for your particular system.

Caution Several of the disk drive subtests are destructive, which means that information contained on the disk is destroyed when these 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 *S/MT Series CTIX Administrator's Reference Manual* for information on backing up the contents of a disk drive before you execute a disk test.

Likewise, be careful when you execute the tape tests because they too can destroy information stored on the tape. Use a blank tape when you execute any tape test sequence.

DISK DRIVE TESTS (w AND u)

The hard disk drive subtests (Test 6), listed in Table 7-2, test every aspect of the currently selected disk drive. (The disk drive is selected with the **DN** command. Refer to Chapter 4 for a description of this command and a list of drive identification numbers.)

Table 7-2. Hard Disk Drive Subtests (Tests 6 and 9)

Subtest	Description
1*	Recalibrates the disk
2	Formats the disk
3*	Tests disk write
4	Tests disk read
5*	Tests random seeks
6	Writes multiple blocks (interactive)
7	Reads multiple blocks (interactive)
8	Spares a sector (interactive)
9*	Forces DMA faults
10	Initializes VHB and BBT
11	Writes VHB and BBT
12	Prints VHB and BBT
14	Toggles alien disk mode
15*	Checks disk DMA register integrity
20	Prints CTIX file system
21	Finds unreadable blocks
23	Tests disk surface
25†	Sorts BBT
29	Tests harmonic seek stress
30	Clones disk
31	Tests quick disk write and check read

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

† Test not applicable to SCSI disk drives.

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**).

Note The test output listings for the disk drive subtests are examples only, and do not imply that the default settings will always be the same.

Bad Block Information for Non-SCSI Drives

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, as described in Table 7-3.

Table 7-3. Bad Block Information Location

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/480	N/A
S/640	in bags on the inner surface of the front and rear doors

Bad Block Information for SCSI Drives

Each SCSI disk drive is shipped with manufacturer's bad block information on a track that is not accessible to the user. This manufacturer's defect list does not change. If additional bad blocks are found, they are added to the grown defect list on the drive either by Diagnostics surface test (23), spare sector test (8), or by the CTIX operating system if it attempts to write to a block that it finds to be bad.

The Volume Home Block (VHB) contains information about bad blocks that the operating system has attempted to read and found to be bad. The next time the operating system tries to write to that block, it adds it to the defect list on the drive and removes it from the bad block table (BBT).

Disk Drive Controller Registers

Disk drives 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.

DISK DRIVES

For disk drives, the error messages include a "Response = #" phrase in which # is a 2 byte hexadecimal number. 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 shown in Table 7-4.

Table 7-4. Error Message Response Codes

Code	Meaning
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)

ST506 DISK DRIVES

For ST506 drives only (not SCSI or SMD), 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 7-5.

Table 7-5. 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

The Error register contains the specific error status codes that pertain to the completion of a command. The bits are described in Table 7-6.

Table 7-6. 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

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

SCSI DRIVES

For SCSI disk drives, error information appears if the response # was 1, 2, or 4. The line will give the command that the SCSI drive was executing, an error, an error key, and a code similar to the following:

```
SCSI error status: cmd 0x## err 0x## key )x##
```

The command, error, and key information is decoded as shown in Tables 7-7, 7-8, and 7-9.

Table 7-7. Direct Access Commands (cmd)

00h	TEST UNIT READY
01h	REZERO UNIT
03h	REQUEST SENSE
04h	FORMAT UNIT
07h	REASSIGN BLOCKS
08h	READ
0AH	WRITE
0Bh	SEEK
12H	INQUIRY
15h	MODE SELECT
16h	RESERVE
17h	RELEASE
18h	COPY
1Ah	MODE SENSE
1Bh	START/STOP UNIT
1Ch	RECEIVE DIAGNOSTIC RESULTS
1Dh	SEND DIAGNOSTIC
1Eh	PREVENT/ALLOW MEDIUM REMOVAL
25h	READ CAPACITY
28h	READ
2Ah	WRITE
2Bh	SEEK
2Eh	WRITE AND VERIFY
2Fh	VERIFY
30h	SEARCH DATA HIGH
31h	SEARCH DATA EQUAL
32h	SEARCH DATA LOW
33h	SET LIMITS
34h	PRE-FETCH
35h	FLUSH CACHE
36h	LOCK/UNLOCK CACHE
37h	READ DEFECT DATA
39h	COMPARE
3Ah	COPY AND VERIFY
3Bh	WRITE BUFFER
3Ch	READ BUFFER
3Eh	READ LONG
3Fh	WRITE LONG

Table 7-8. Error Numbers (err)

Error	Description
0x01	Parity error on SCSI bus
0x02	Excess data after DMA exhausted
0x04	Time out error
0x08	Invalid command
0x10	SCSI bus was reset and high level driver has not cleared condition
0x20	Unknown protocol error

Table 7-9. Key Descriptions (key)

Key	Description
0h	NO SENSE – indicates that there is no specific sense key information to be reported for the designated logical unit. This would be the case for a successful command or a command that received a CHECK CONDITION status because one of the filemark, EOM, or ILI bits is set to one.
1h	RECOVERED ERROR – indicates that the last command completed succesfully with some recovery action performed by the target. Details may be determinable by examining the additional sense bytes and the information bytes.
2h	NOT READY – indicates that the logical unit addressed cannot be accessed. Operator intervention may be required to correct this condition.

Table 7-9. Key Descriptions (key) (continued)

Key	Description
3h	MEDIUM ERROR – indicates that the command terminated with a nonrecovered error condition that was probably caused by a flaw in the medium or an error in the recorded data. This sense key may also be returned if the target is unable to distinguish between a flaw in the medium and a specific hardware failure (sense key 4h).
4h	HARDWARE ERROR – indicates that the target detected a nonrecoverable hardware failure (for example, controller failure, device failure, parity error, etc.) while performing the command or during a self test.
5h	ILLEGAL REQUEST – indicates that there was an illegal parameter in the command descriptor block or in the additional parameters supplied as data for some commands (FORMAT UNIT, SEATCH DATA, etc.). If the target detects an invalid parameter in the command descriptor block, then it shall terminate the command without altering the medium. If the target detects an invalid parameter in the additional parameters supplied as data, then the target may have already altered the medium.
6h	UNIT ATTENTION – indicates that the removable medium may have been changed or the target has been reset.
7h	DATA PROJECT – indicates that a command that reads or writes the medium was attempted on a block that is protected from this operation.
8h	BLANK CHECK – indicates that a write-once read-multiple device or a sequential-access device encountered blank medium or format-defined end-of-medium indication while reading, or a write-once read-multiple device encountered a nonblank medium while writing.
9h	Vendor Unique – this sense key is available for reporting vendor unique conditions.

Table 7-9. Key Descriptions (key) (continued)

Key	Description
Ah	COPY ABORTED – indicates a COPY, COMPARE, or COPY and VERIFY command was aborted due to an error condition on the source device, the destination device, or both.
Bh	ABORTED COMMAND – indicates that the target aborted the command. The initiator may be able to recover by trying the command again.
Ch	EQUAL – indicates a SEARCH DATA command has satisfied an equal comparison.
Dh	VOLUME OVERFLOW – indicates that a buffered peripheral device has reached the end-of-medium and it remains in the buffer that has not been written to the medium. A RECOVER BUFFERED DATA command(s) may be issued to read the unwritten data from the buffer.
Eh	MISCOMPARE – indicates that the source data did not match the data read from the medium.

SMD DRIVES

Detailed SMD controller status and error register information is discussed in the S/MT Series VME SMD Controller Card Manual.

SELECTING THE DISK DRIVE

An S/MT Series computer system can contain several disk drives. Disk drive 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 the following 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**

2. Press **RETURN.**

The following 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 following order:

- 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 Disk

This subtest initializes the disk, brings the head back to 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.**

The following 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, save the contents of the disk as described in the *S/MT Series CTIX Administrator's Reference Manual.*

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(1) performs this function. See the *CTIX Operating System Manual* for a description of the iv command.

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)?
```

Typing **n** returns you to the command prompt. To reformat, type **y**; the following output is displayed:

```
The Predefined Volume Block formats are:
      62CP-HITACHI
      84CP-HITACHI
      85CP-MAXTOR
      86CP-MICROPOLIS
      87CP-MINI SCRIBE
      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 appear (if you instead select one of the above format types, these prompts are 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):
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:
```

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.

Subtest 3: Write

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 series CTIX 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 pass 0
Read address is 0x71000, write address is 0x67000
```

Subtest 4: Read

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.**

The following 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.
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, this output is displayed while the subtest is executed:

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

Subtests 6-7: Write/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, 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, 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.**

If the disk drive is not a SCSI, the following prompts are displayed:

```
DISK TEST Drive0 [56ATA]
(Subtest 8) Spare a sector function.
Volume Name: 56ATA (ecc mode)
Give a physical sector number:
```

If you press **RETURN** (Cylinder, Head, and Byte), the following prompts 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 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 does not appear if the sector is already spared.

If the disk drive is a SCSI disk, the following prompt is displayed:

```
Give physical sector:
Enter sector number
Another? (Y or N)
```

Note If you suspect that there are bad sectors, run subtest 21 and note the bad sectors. Repeat subtest 8 and enter the bad physical sector number in the preceding display.

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.**

The 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 following output is displayed:

```
      DISK TEST Drive0 [56ATA]
(Subtest 10)  Initialize VHB and BBT .
Do you want to change the VHB on Drive0 (Y or N)?
```

Typing **n** returns you to the command prompt. To reformat, type **y**; the following output is displayed:

```
The Predefined VHB formats are:
      62CP-HITACHI
      84CP-HITACHI
      85CP-MAXTOR
      86CP-MICROPOLIS
      87CP-MINI SCRIBE
      143CP-MAXTOR
      190CP-MAXTOR
      300CP-Priam807
      301CP-NT8212
      302CP-DX332
      450CP-EAGLE
      600CP-EAGLEXP
Enter format [?      ype to display VHB format]  (press return to
specify your own format)
```

If you specify your own format, the following prompts appear (if you instead select one of the above format types, these prompts are 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):
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):
Enter y to approve vhb parameters, enter n to abort:
```

The last prompt provides the opportunity to abort the test. Enter **y** to approve VHB parameters, enter **n** to abort. When the parameters are approved, 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 with **y**. Typing **n** returns you to the command 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 shown 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.
Total number of usable 1K logical blocks = 142519
Partition 0: start    Block = 0,        size (in Blocks) = 36120
Partition 1: start    Block = 336,      size (in Blocks) = 16000
Partition 2: start    Block = 16226,    size (in Blocks) = 8000
Partition 3: start    Block = 24336,    size (in Blocks) = 20000
Partition 4: start    Block = 44336,    size (in Blocks) = 40000
Partition 5: start    Block = 84336,    size (in Blocks) = 58184
Loader starts at block 3
BBT starts at block 1 (size = 1 Blocks).
The BBT contains 0 entries.
Grown defect list
Cylinder 126   head 0    sector 0
```

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.**

The following 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 appear (if you instead select one of the above format types, these prompts are 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):
Is the disk exchangeable (Y or N):
Does the drive use head select bit 3 (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.**

The following output is displayed to confirm that the 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:

```
DISK TEST Drive0 [56ATA]
(Subtest 20) Print CTIX File System.
Volume Name: 56ATA
Enter partition number to describe (Default = 1)
```

After entering a partition number and pressing **RETURN**, output similar to the sample below appears:

```
#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.**

The following prompts are displayed:

```
DISK TEST Drive0 [56ATA]
(Subtest 21) Find unreadable blocks
Volume Name:
Enter start block (Default = 0):
Enter # of blocks (Default = 36120):
Enter repeat count (Default = 1):
```

Subtest 23: Surface

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.**

The following 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:

```
Rotating Data Pattern = 0XDB6, Start Sector = 51, # Sector = 76704,
Increment = 17, Repeats = 1, Retries = 10
```

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.

Note This subtest has no effect on SCSI disk drives.

To sort the BBT:

1. Type **:6,25**

2. Press **RETURN.**

The following 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, 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 appears:

```
Added Bad Block: Cylinder XXX, Head X, Sector XX
Used track XXXX as the alternate
```

The subtest prompts you for permission to continue on error.

Subtest 29: Harmonic Seek Stress

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.**

The following output is displayed to indicate test execution:

```
DISK TEST Drive0
(Subtest 29) Harmonic Seek stress test.
Volume Name: 56ATA
Seeks: Low to High
Seeks: 080
Seeks: increasing to 80 then decreasing seeks to 0
```

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 5MB 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.**

The following 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
```

Subtest 31: Quick Disk Write and Check Read

This subtest does a quick function test of the disk drive and controller. The test writes a data pattern to 2000 randomly assigned data blocks and then reads these blocks back to verify that they are correct.

Caution Since Write Disk destroys all information on the disk drive, you should save information already on the disk. Refer to the *S/MT series CTIX Administrator's Reference Manual* to do so.

To execute the quick disk write and check read subtest:

1. Type **:6,31**
2. Press **RETURN.**

The following prompt is displayed:

```
(Subtest 31) Quick Disk Write and Check Read test
Do you want to write to Drive 00 [m1310]?
Read address is 0xXXXX, Write address is 0xXXXX
```

TEST 16: QIC CARTRIDGE TAPE (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 described in Table 7-10. Subtests 4 and 5 consist of numerous options that are detailed in this chapter.

Table 7-10. QIC Tape Subtests

Subtest	Description
1	Resets and initializes tape
2	Performs adverse DMA tests
3	Rewinds tape
4	Runs interactive diagnostics
	0 runs utilities 1 rewinds tape 2 issues write command 3 issues read command 4 writes blocks 5 reads blocks 6 displays control lines 7 issues write filenames 8 issues read filenames a resets controller b issues retension command c issues read status command d dumps blocks with filemark e reads blocks up to filemark f displays pages of memory g displays previous tape status h adverse DMA test

Table 7-10. QIC Tape Subtests (continued)

Subtest	Description
5	Runs interactive utilities 0 runs diagnostics 1 rewinds tape 3 copies disk partition to tape 4 restores disk partition from tape 5 runs tape write/read test 6 runs write phase only 7 runs read phase only 8 copies disk blocks to tape 9 restores disk blocks from tape a resets tape b retensions tape c reads tape status d sets tape data format e erases test
6	Executes write phase of tape test
7	Executes read phase of tape test
8	Sets up parameters of tape test
9	Erases tape
10	Retensions tape

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 display messages used in this chapter are examples only and do not reflect limitations or default settings.

When the tests are successfully executed, this message is displayed:

```
        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. It executes a data path check to the tape DMA byte count register, resets the tape controller, and reads the tape status.

Execute Subtest 1 as follows:

1. Type **:16,1**

2. Type **RETURN.**

Upon successful completion of this subtest, the following message is displayed:

```
Qici - controller reset
```

Subtest 2: Adverse DMA

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 this word from memory and write it to the disk. The subtest reports that the tape controller subsystem detected the parity error, and that the status bit in the Tape Condition register was set. Correct parity is then restored to the 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.

Execute subtest 2 as follows:

1. Type **:16,2**

2. Press **RETURN.**

The following 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. Rewind the tape as follows:

1. Type **:16,3**

2. Press **RETURN.**

The following output is displayed:

```
Rewinding the tape ... Qic - 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. The *IEEE Tape Interface Specification* describes how they can be most effectivly used.

Display the QIC tape diagnostic options as follows:

1. Type **:16,4**
2. Press **RETURN.**

The following 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)$
```

The prompt changes to **Q(diag)** to indicate that you are in the diagnostic menu of the QIC tape test.

To execute a tape diagnostic option, simply 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:

```
Qici write block @ memory address:0x3E6C8M
How many 512 byte blocks? 0-100000:M
```

(5): READ BLOCKS

This option reads the blocks of data sent to tape from memory in the above option, verifying that the transfer was done properly. It provides the following prompt:

```
Qici read block @ memory address:0x3E6C8
How many 512 byte blocks? 0-100000 :
```

(6): DISPLAY CONTROL LINES

This option displays the control lines. It provides the following output:

```
signals 0xE: not ready, no data error, exception, not ack
```

(7): ISSUE WRITE FILEMARK

This option can be 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 provides the following 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. It presents the following output:

```
tape status:0x81 Reset Occurred
```

In this message, as in the Utilities, the decode for the hexadecimal number (0x81) appears 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 the following output:

```
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 as shown in the following display:

```
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 as shown in the following display:

```
previous tape status
tape status:0x00
```

(h): ADVERSE DMA (WRITES TO TAPE)

This option tests the ability to detect three types of errors:

- Parity error

- Page protection error
- Address error

Subtest 5: Run Interactive Utilities

This subtest consists of 15 options. Access the interactive tape utilities as follows:

1. Type **:16,5**
2. Press **RETURN.**

The following list of utilities is displayed on your screen:

```
qici utilities
qici - 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 a tape utility option, simply 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 to the tape. To allow multiple partitions to be copied, 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 from 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, 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 from memory to tape in option 6 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 it lets you copy blocks instead of partitions from disk to tape. It displays the following 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 it lets you restore blocks instead of partitions to disk. It displays the following 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 TAPE

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. For a complete description, refer to the same option in the diagnostics section.

(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 utility options 6 and 7.

(e): ERASE

This option erases the tape currently in the tape drive. It outputs the following:

```
    Erase tape test
    Rewinding the tape ... Qic - controller reset
Rewind okay
```

Subtest 6: Write Phase of Tape

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.

Execute subtest 6 as follows:

1. Type **:16,6**

2. Type **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

This subtest verifies that the data written from memory to tape in subtest 6 is correct. Execute subtest 7 as follows:

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

This subtest lets you specify the number of blocks and the data pattern for subtests 6 and 7. Execute subtest 8 as follows:

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

This subtest erases the contents of the tape in the QIC tape drive. Execute subtest 9 as follows:

1. Type **:16,9**

2. Press **RETURN.**

The following 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

This subtest retensions the tape, rewinds it, and resets the tape controller. Execute subtest 10 as follows:

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 (j)

The general tape drive subtests test every aspect of the SCSI half-inch tape drive installed in your system. The tape drive subtests are described in Table 7-11.

Table 7-11. 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 tape drive identification numbers are described in Table 4-1 in Chapter 4. The SCSI tape drive in a CPU-02 system is initially selected by the tests. To determine the currently selected tape drive:

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

2. Press **RETURN.** The following message 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.

The tape drive number can be changed with the two-letter **TN** command. To do this, first to determine the number of the tape drive you want to select.

Each SCSI controller can support up to eight tape drives. The controller on the Main Processor (CPU) board is designated as 0. Each controller on the VME bus is designated as 1, 2, etc., 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. Refer to 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.**

The following message is displayed:

```
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 select a drive that is not attached to your system. Use the diagnostic screen described in Chapter 2, "Getting Started" to determine which controllers and drives are attached to your system.

Executing the Tape Drive Test Sequence

The general tape drive test sequence (**j**) consists of 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 When you issue this command, the diagnostic writes information to the tape which destroys any previously written data. Load a blank or scratch tape before issuing 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 appears:

```
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, the following message is displayed:

```
Waiting for drive X to finish rewinding...done
```

Subtest 1: Reset and Initialize

This subtest resets and initializes the controller for the selected tape drive and prints the controller status on the screen. Execute subtest 1 as follows:

1. Type **:20,1**
2. Press **RETURN.**

Test execution is confirmed with the following output:

```
Tape Test (subtest 1) reset and initialize
Controller X [controllername] Parameters:none
```

Subtest 2: Set Default Controller Parameters

This subtest sets predefined parameters for the selected controller. Note that this subtest does not apply to SCSI tape drives. Execute subtest 2 as follows:

1. Type **:20,2**
2. Press **RETURN**.

The following message is 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. Execute subtest 3 as follows:

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 session
```

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 uses 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 are 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 is displayed:

```
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 are displayed:

```
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 are displayed 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 prompts you for the desired current tape density. If the selected drive uses the Density Select Bit, the following prompt is displayed:

```
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 is displayed:

```
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.

Run subtest 5 as follows:

1. Type **:20,5**
2. Press **RETURN.**

The following message is displayed:

```
Not currently supported by this controller.
```

Subtest 6: Rewind the Tape

This subtest rewinds the current tape. Run subtest 6 as follows:

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 message is displayed:

```
    Tape Test (subtest 6) rewind the tape
Drive 8 [Interphase] Status: <online, ready, tape rewinding,
```

Subtest 7: Erase Tape

This subtest erases the entire length of the current tape drive. Run subtest 7 as follows:

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 Next Filemark

This subtest seeks 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

This subtest exercises the DMA address counters of the tape controller. Sixteen 4KB 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 is displayed:

```
Block size not set for this drive, using default of XXXX bytes
```

Subtest 10: Write

This subtest writes a data pattern of incrementing 32-bit integers in forty 32KB blocks to the currently selected tape drive. The pattern is constant for each block, and can be read and verified using subtest 11.

Write blocks of a data pattern to the tape drive as follows:

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 is displayed:

```
Block size not set for this drive, using default of XXXX bytes
```

Subtest 11: Read

This subtest reads the pattern written to the selected tape drive in subtest 10.

Read the data pattern written in subtest 10 as follows:

1. Type **:20,11**

2. Press **RETURN.**

The following 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 appears:

```
Block size not set for this drive, using default of XXXX bytes
```

Subtest 12: Copy Disk Partition to Tape

This subtest copies 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**
2. 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 message is displayed:

```
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 copies 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 overwrites 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**.

The following output is displayed (which assumes that drive 0 was selected and partitioned):

```
    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**, this output is displayed:

```
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 copies disk blocks to the selected tape without regard to the partition boundaries. This subtest requires that the maximum block (set in subtest 4) be set to at least 512 bytes.

Caution This subtest overwrites any information located at the specified portion of the tape.

Copy disk blocks to tape as follows:

1. Type **:20,14**
2. Press **RETURN.**

The following output is displayed (which assumes that drive 0 was selected and partitioned as shown):

```
      Tape Test (subtest 14) copy disk blocks to tape
Backup Disk Blocks to Tape (norewind)
Enter Disk Drive #(default=0):0-60:
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 copies a disk block previously written to the tape back to the specified disk. This subtest requires that the maximum block size (set in subtest 4) be set to at least 512 bytes.

Caution This subtest overwrites 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**.

The following output is displayed (which assumes that drive 0 was selected and partitioned as shown):

```
    Tape Test (subtest 15) restore disk blocks from tape
Backup Disk Blocks to Tape (norewind)
Enter Disk Drive #)default=0):0-60:
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 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.

Select single tape drive mode as follows:

1. Type **:20,16**
2. Press **RETURN**.

The following message 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.
```

CHAPTER 8

RS-232-C Channel Diagnostic Tests

INTRODUCTION

This chapter describes 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) (h)

Test 11 tests every available RS-232-C channel as indicated by Table 8-1.

Table 8-1. RS-232-C Channel Subtests (Test 11)

Subtest	Description
1	Performs data Path Check to the 8530
2	Programs all options
3	Performs simultaneous data transfers
4	Forces error conditions
5*	Tests driver pins (RTS, DTR)
6	Selects channel for subtest 1
7	Selects channel for subtest 2
8	Selects channel for subtest 3
9	Selects channel for subtest 4
10*	Selects 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 as follows:

1. Type **CN**
2. Press **RETURN**.

If the system you are testing contains one 10-channel RS-232-C Expansion board, 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 connected to Channel 0 on the Main Processor (CPU) board.

If you use **CO** to select another RS-232-C channel, no matter what channel the console is connected to, that channel is not tested. Also, if tty000 echo is on, Channel 0 is not 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.

Execute the entire test as follows:

1. Type **h**

2. Press **RETURN**.

The following 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 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.

Execute subtest 1 as follows:

1. Type **:11,1**

2. Press **RETURN**.

The following 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. 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, 75, 110, 134, 150, 200, 300, 600, 1200, 1800, 2400, 4800, 9600, 19200

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.

Execute subtest 2 as follows:

1. Type **:11,2**
2. Press **RETURN**.

2. Press **RETURN**.

The following 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 as indicated in Table 8-2.

Table 8-2. 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

If the system you are testing contains two RS-232-C channels, only channel 1 is tested (since the terminal uses channel 0, it is unavailable). In this case, all 8400 bytes are sent to that single channel.

If the system you are testing contains 22 channels, 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.

Execute subtest 3 as follows:

1. Type **:11,3**

2. Press **RETURN.**

The following 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
```

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, a framing error is generated. Subtest 4 executes the write/read process once per channel.

Execute subtest 4 as follows:

1. Type **:11,4**

2. Press **RETURN**.

The following 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: Test Driver Pins: RTS, DTR

Note Loopback plugs are required to execute this subtest. The success of this test depends upon accurate loopback pin assignments. Refer to 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 (CPU) 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.

Execute subtest 5 as follows:

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 as follows:

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 as follows:

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 as follows:

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, select channels 1, 2, and 3 as follows:

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, press **RETURN**. Subtest 3 is executed according to the channels specified in the sequence above.

Subtest 9: Select Channel for Subtest 4

Subtest 9 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.

Execute subtest 9 as follows:

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

Subtest 10 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.

Execute subtest 10 as follows:

1. Type **:11,10**

2. Press **RETURN**.

The output generated by subtest 10 depends upon the channel you select.

CHAPTER 9

IOP Accelerator Board Diagnostic Tests

INTRODUCTION

This chapter describes the diagnostic tests for the Input/Output Processor (IOP) Accelerator board that you can install (one per system) in the IOP/RS-422 slot of the local card cage. (Refer to the installation manual for your system for the location of the IOP slot).

Note The subtests described in this section are ignored when the board is not installed.

TEST 5: IOP ACCELERATOR BOARD (i)

This series of subtests ensures that the IOP Accelerator board functions properly. The IOP Accelerator board contains an MC68000 CPU, a 16-bit wide data bus, and 64KB of memory. It contains its own physical memory and provides access to host virtual memory. It 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, shown in Table 9-1, that 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.

Table 9-1. IOP Diagnostic Tests (Test 5)

Subtest	Description
1	Writes and reads 0x0000s
2	Writes and reads 0xffffs
3	Writes and reads 0xaaaas
4	Writes and reads 0x5555s
5	Writes and reads addresses
6	Writes and reads address complements
7	Uploadss IOP resident subtests
8	Runs interrupt test
9	Runs Data Path Check to 8253
10	Runs Clock 0 test
11	Runs Clock 1 test
12	Runs Clock 2 test
13	Runs Data Path Check to existing RS-232-C channels

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, effectively 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, the following 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
```

Subtests 1 to 6: IOP Memory

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 64KB of memory is tested. These subtests are defined as follows:

- Subtest 1 – write and read 0x0000s
- Subtest 2 – write and read 0xffffs
- Subtest 3 – write and read 0xaaaas
- Subtest 4 – write and read 0x5555s
- Subtest 5 – write and read addresses
- Subtest 6 – write and read address complements

Because the IOP Accelerator board data bus is 16 bits wide, the predefined patterns are written in 16-bit mode. You can execute the subtests individually.

For example:

1. Type **:5,X** (where X is the subtest number).

2. Press **RETURN.**

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

```
      IOP TEST
booting iop ...
FROM IOP:  IOP RAM SUBTEST X
```

Subtest 7: Upload IOP Resident

This subtest loads the IOP diagnostics from host virtual memory to the physical memory on the IOP Accelerator board.

To execute subtest 7:

1. Type **:5,7**

2. Press **RETURN.**

The following 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

This subtest checks the IOP Interrupt signals generated by the MC68000 processor. During this subtest, 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**.

The following 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**.

The following 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

This subtest checks that Counter 0 on the 8253 chip is functioning correctly.

To execute subtest 10:

1. Type **:5,10**

2. Press **RETURN**.

The following 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

This subtest checks that Counter 1 on the 8253 chip is functioning correctly.

To execute subtest 11:

1. Type **:5,11**
2. Press **RETURN**.

The following 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

This subtest checks that Counter 2 on the 8253 chip is functioning correctly.

To execute subtest 12:

1. Type **:5,12**

2. Press **RETURN**.

The following 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 clock 2
```

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.

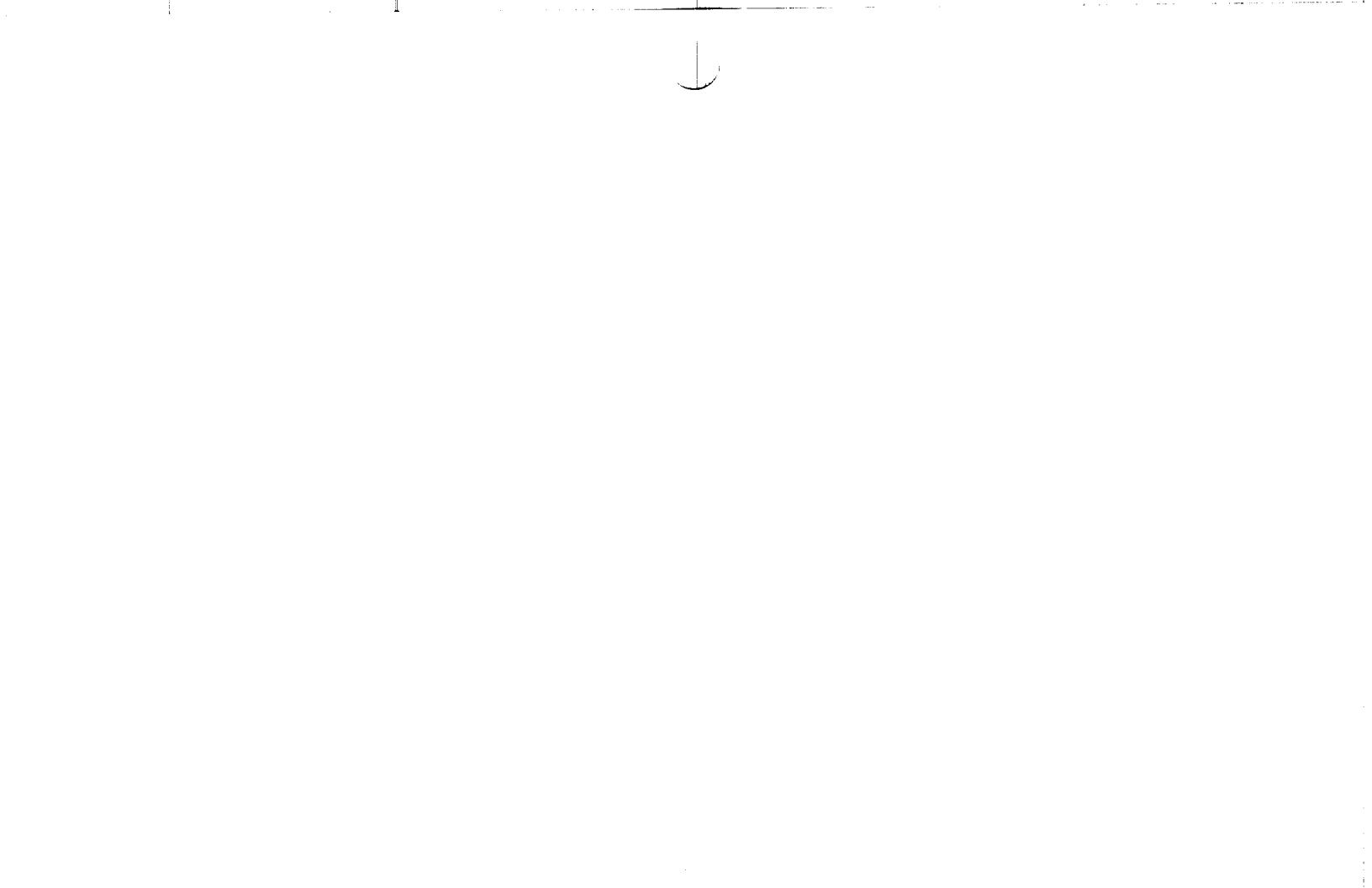
To execute subtest 13:

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
```



CHAPTER 10

Ethernet Combo Board Diagnostic Tests

INTRODUCTION

This chapter describes the Ethernet Combo board diagnostic tests.

TEST 15: ETHERNET COMBO BOARD (n)

If your system contains one or more Ethernet Combo boards, this test (test 15) can be used to ensure the functionality of the 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. Ethernet Combo board diagnostic tests are listed in Table 10-1.

This test is ignored when the board is not installed.

Table 10-1. Ethernet Combo Board Tests (Test 15)

Subtest	Description
0	Performs default test sequence [1,2,3,6,7,8]
1*	Writes/reads local RAM
2*	Writes/reads host memory
3*	Writes/reads local EEPROM
4	Writes pattern to local EEPROM [inverse addresses]
5	Reads local EEPROM set by 4
6*	Tests interrupts
7*	Does Loopback, CRC, and Collision tests
8*	Does Time Domain Reflectometry tests
9	Tests external loopback
10	Configures/Selects a board [print station address]
11	Toggles Debug mode
12	Toggles packet print mode
13	Sends a test packet
14	Retransmits packet sent in subtest 13
15	Checks ping-pong w/ data
16	Checks reverse ping-pong w/ data
17	Transmit serially-numbered packets
18	Sends back-to-back packets to station ZERO
19	Monitors line - save data
20	Monitors line - don't save data
21	Enables board and leave running (for ping-pong, etc.)
22	Toggles DMA channel (IOP/VME)
23	Sets station address to zero
24	Toggles interactive mode

* Test sequence n and subtest 0 run these subtests.

Subtest 1: RAM Write/Read

This subtest ensures that the 64KB of RAM on the Ethernet Combo board is operational. A series of data patterns is written into memory, then read from memory and compared with the original. Errors are reported. The patterns written (listed in order) include:

- All zeros
- All ones
- All As
- All fives
- Address pattern
- Complement address pattern

Execute subtest 1 as follows:

1. Type **:15,1**
2. Press **RETURN**.

When the test is successfully completed, the following is displayed:

```
FROM CTEN-X  CT-ENET MEMORY SUBTEST 1
FROM CTEN-X  CT-ENET MEMORY SUBTEST 2
FROM CTEN-X  CT-ENET MEMORY SUBTEST 3
FROM CTEN-X  CT-ENET MEMORY SUBTEST 4
FROM CTEN-X  CT-ENET MEMORY SUBTEST 5
FROM CTEN-X  CT-ENET MEMORY SUBTEST 6
```

Subtest 2: Host Memory Write/Read

This subtest, similar to subtest 1, verifies DMA operations on the block of memory. Host memory performs the memory subtests (as described in subtest 1) on a 4KB block of host memory.

Execute subtest 2 as follows:

1. Type **:15,2**
2. Press **RETURN**.

Subtest 3: 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.

Execute subtest 3 as follows:

1. Type **:15,3**

2. Press **RETURN**.

Subtest 4: EEPROM Write Pattern

This subtest writes a complemented address pattern into the EEPROM. Subtest 5 is used to read and verify this pattern.

Execute subtest 4 as follows:

1. Type **:15,4**
2. Press **RETURN**.

Subtest 5: EEPROM Read Pattern Set by 4

This subtest reads and verifies the complemented address pattern written into the EEPROM by subtest 4.

Execute subtest 5 as follows:

1. Type **:15,5**
2. Press **RETURN**.

Subtest 6: Interrupt

This subtest checks the interrupt signals generated by the Ethernet Combo board and host CPU. In this subtest, 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.

Execute subtest 6 as follows:

1. Type **:15,6**

2. Press **RETURN**.

Subtest 7: Loopback, CRC, and Collision

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.

Execute subtest 7 as follows:

1. Type **:15,7**

2. Press **RETURN**.

Subtest 8: Time Domain Reflectometry

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.

Execute subtest 8 as follows:

1. Type **:15,8**

2. Press **RETURN**.

If the cable is not broken, the following message is displayed:

```
No discontinuity detected.
```

Subtest 9: External Loopback

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.

Execute subtest 9 as follows:

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.

Execute subtest 10 as follows:

1. Type **:15,10**
2. Press **RETURN**.

The following prompts are displayed. To accept the default values, simply press **RETURN** again.

```
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 enter and exit Debug mode for a test. When you are in Debug mode, extra messages are printed on the screen.

Execute subtest 11 as follows:

1. Type **:15,11**
2. Press **RETURN**.

The following prompt is displayed.

```
DEBUG MODE IS ON.
```

To turn Debug mode off, simply run the subtest again. The following message is displayed:

```
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.

Execute subtest 12 as follows:

1. Type **:15,12**
2. Press **RETURN**.

The following prompt is displayed:

```
PACKET PRINT MODE IS ON.
```

To turn Packet Print mode off, simply run the subtest again. The following message is displayed:

```
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 verify accurate transmission and detect transmission problems with the Ethernet line.

Execute subtest 13 as follows:

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 Sent in Subtest 13

This subtest enables you to retransmit the packet generated in subtest 13. The message, destinations, and packet type are the same as in subtest 13.

Execute subtest 14 as follows:

1. Type **:15,14**

2. Press **RETURN**.

Subtest 15: Ping-Pong with Data Check

Two systems on the Ethernet line are required to run this subtest. The two systems transmit and receive data to and from each other.

The first station sends a 1KB packet consisting of a sequential number of bytes (0 to 255). The second station receives and verifies this data, then transmits the packet back to the first station (which also verifies the data). This pattern is repeated until the specified number of packets has been transmitted back and forth. Any errors are displayed.

Execute subtest 15 as follows:

1. Type **:15,15**

2. Press **RETURN**.

The following prompts appear. To accept the default values, press **RETURN** again.

```
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).

Execute subtest 16 as follows:

1. Type **:15,16**
2. Press **RETURN**.

Subtest 17: Transmit Serially-Numbered Packets

This subtest allows 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 must be enabled by running subtest 21. The receiving station prints the contents of this packet if Packet Print mode is enabled.

To exit this test, press **ESC**.

Execute subtest 17 as follows:

1. Type **:15,17**

2. Press **RETURN**.

The following prompts are displayed. To accept the default values, simply press **RETURN** again.

```
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.

Execute subtest 18 as follows:

1. Type **:15,18**
2. Press **RETURN**.

Subtests 19 and 20: Monitor Line

These subtests are designed as troubleshooting aids for technicians debugging the Ethernet Combo board. Both tests receive all packets.

Subtest 19 saves data it receives. When subtest 19 is run it prints out the address of a buffer (64KB) on the host. Received packets are saved in this buffer in a ring fashion. The debugger can be used to inspect this buffer.

Press **CONTROL-B** to enter the Debugger. 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.

To exit this test, press **ESC**.

Execute subtest 19 or 20 as follows:

1. Type **:15,19** or **:15,20**

2. Press **RETURN**.

Subtest 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 performs a ping-pong handshake and packet validiation.

Execute subtest 21 as follows:

1. Type **:15,21**

2. Press **RETURN**.

Subtest 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 systems. If you attempt to run it on a CPU-02 system the following message is displayed:

```
Not used on Mitl2
```

Execute subtest 22 as follows:

1. Type **:15,22**
2. Press **RETURN.**

The following message is displayed:

```
    CT-Ethernet test [board = n]
Booting Ethernet board
From CTEN -n: Getting Host Parameters
    Subtest 22 CTEN: Toggle DMA Channel
Selected VME DMA channel
```

Subtest 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).

Execute subtest 23 as follows:

1. Type **:15,23**

2. Press **RETURN**.

Subtest 24: Toggle Interactive Mode

This subtest does not perform a diagnostic test. Instead it allows you interact with the diagnostic tests. When running in Interactive mode, you are prompted for information as documented in this section. While not running in Interactive mode, the tests use the default values and do not prompt you for information.

Execute subtest 24 as follows:

1. Type **:15,24**

2. Press **RETURN**.

The following message is displayed:

```
INTERACT IS ON
```

To turn Interactive mode off, simply run the test again. The following prompt is displayed:

```
INTERACT IS OFF
```

CHAPTER 11

RS-422 Expansion Board Diagnostic Tests

INTRODUCTION

This chapter describes the diagnostic tests for the RS-422 expansion boards (2-channel and 4-channel) that are installed in the I/O portion of the local card cage.

Note The subtests described in this section are ignored when the RS-422 Expansion boards are not installed.

TEST 14: RS-422 EXPANSION BOARD (k)

If your system contains the 2- or 4-cluster channel RS-422 Expansion board, test 14 can be used to ensure the functionality of the four RS-422 cluster channels on the board. This test contains eight subtests as described in Table 11-1.

Table 11-1. RS-422 Cluster Channel Subtests (Test 14)

Subtest	Description
1	Performs data path check (DMA and PCSAR)
2	Performs auto send/receive tests (2 channels)
3	Performs auto send/receive tests (all channels)
4	Performs send/receive to user-specified channel
5	Sends user-specified memory block
6	Receives user-specified memory block
7	Tests parity error read
8	Tests DMA address counter
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. Refer to 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.

Execute subtests 1, 2, 3, 7, and 8 as follows:

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. 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.

Execute subtest 1 as follows:

1. Type **:14,1**
2. Press **RETURN**.

Subtest 2: Auto Send/Receive (2 Channel)

Subtest 2 transmits a 2 byte (one word) packet to each RS-422 cluster channel. During each subsequent transmission, the packet size is doubled (two words, four words, and so on 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.

Execute subtest 2 as follows:

1. Type **:14,2**

2. Press **RETURN**.

Subtest 3: Auto Send/Receive (All Channels)

Subtest 3 does the same as subtest 2, except that it tests all four RS-422 cluster channels at the same time. This subtest 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 all transmissions and receptions are correct.

This subtest cannot be run on 2-channel RS-422 Expansion boards.

Execute subtest 3 as follows:

1. Type **:14,3**

2. Press **RETURN.**

Subtest 4: Transmit/Receive User-Specified Channel

Subtest 4 is similar to subtest 2, except that subtest 4 lets you specify which cluster channels to test. Unless otherwise specified, the transmitting cluster channel defaults to 0 and the receiving cluster channel to 1.

You can specify any combination of cluster channels, including 3 to 1, 0 to 3, 0 to 2, or 2 to 0. You may also 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, cluster channel 3 becomes the default receive cluster channel. The same holds true for cluster channels 0 and 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.

Execute subtest 4 as fllows:

1. Type **:14,4**

2. Press **RETURN.**

The following prompts are displayed:

```
        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 which 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 diagnostics.

Execute subtest 5 as follows:

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 also enables you to specify the address in memory where the data is sent. If you specify no address, subtest 6 defaults to the first available address in memory not used by the diagnostics.

Execute subtest 6 as follows:

1. Type **:14,6**

2. Press **RETURN**.

Subtest 7: Parity Error Read

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 diagnostic verifies that bit 15 in the general status register (GSR) and bit 10 in the expansion port ID register detect even parity and generate a parity interrupt.

Execute subtest 7 as follows:

1. Type **:14,7**

2. Press **RETURN**.

Subtest 8: DMA Address Counter

This subtest assumes that cluster channels 0-1 and 2-3 are connected together. Subtest 8 tests the ability of the RS-422 subsystem to DMA any DMA virtual address.

Execute subtest 8 as follows:

1. Type **:14,8**
2. Press **RETURN**.

CHAPTER 12

VME Subsystem Diagnostic Tests

INTRODUCTION

This chapter describes the diagnostic tests that verify that the VME functions properly.

TEST 17: VME SUBSYSTEM (v)

The VME subtests are specific to the CMC VME board. Therefore, the term CMC VME board is used throughout this chapter.

The VME subtests make the assumptions that:

- The VME Card Cage is installed
- The VME Interface board is installed
- The CMC VME board addresses the VME bus at the address specified in the VME EEPROM

Refer to the installation manual for your system (S/222, S/320, S/480, and S/640 only) to install the VME card cage, the VME Interface board, and VME Expansion boards.

The subtests for the CMC VME Expansion board are listed in Table 12-1. Before executing each subtest, the host CPU writes a program into the MC68000 on the CMC VME board to instruct it to execute the VME subsystem tests. The VME subsystem default test sequence consists of subtests 1 through 6 only.

Table 12-1. VME Subsystem Subtests (Test 17)

Subtest	Description
1	Reads a single location from host memory
2	Writes a single location from host memory
3	Reads, then writes, a single location to host memory
4	Copies a 1KB block from host memory
5	Copies a 1KB block to host memory
6	Reads, writes, compares a 1KB block in host memory
7	Copies, compares a 1KB block in host memory
11-17	Same as 1 through 7 in endless loop mode

Execute the VME subsystem test sequence as follows:

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.

Execute subtest 1 as follows:

1. Type **:17,1**

2. Press **RETURN**.

Subtest 2: Write Data

This subtest writes a program from host memory to the CMC VME board instructing the board to write 0x9C7A to an address in host memory. The CMC VME board then checks the address in host memory to ensure that the correct address contains 0x9C7A.

Execute subtest 2 as follows:

1. Type **:17,2**

2. Press **RETURN**.

Subtest 3: Read and Write Data

This subtest writes a program that instructs the CMC VME board to read the contents of an address in host memory. The data is then written to a different address in host memory, and the host compares the data in each address to ensure that it is the same.

Execute subtest 3 as follows:

1. Type **:17,3**

2. Press **RETURN.**

Subtest 4: Copy Block from Host Memory

This subtest writes a program that copies a 1KB block of data from an address in host memory to an address on the CMC VME board. The first 2 bytes in CMC VME memory are then checked to verify that the copy was successful.

Execute subtest 4 as follows:

1. Type **:17,4**

2. Press **RETURN.**

Subtest 5: Copy Block to Host Memory

This subtest writes a program that copies the same 1KB block of data from an address on the CMC VME board back into the same address in host memory. The first 2 bytes in host memory are then checked to verify that the data was successfully copied.

Execute subtest 5 as follows:

1. Type **:17,5**

2. Press **RETURN**.

Subtest 6: Read/Write/Compare Block

This subtest writes a program to the MC68000 (on the CMC VME board) which instructs it to read 1KB of memory from an address in host memory and write it to an address on the CMC VME board. The 1KB is then written to an address in host memory, where it is compared with the original 1KB.

Execute subtest 6 as follows:

1. Type **:17,6**

2. Press **RETURN**.

Subtest 7: Copy/Compare Block

This subtest instructs the CMC VME board to copy data from 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.

Execute subtest 7 as follows:

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.

Execute one of these subtests as follows:

1. Type **:17,** *subtest number*

   For example, to execute subtest 6 in endless loop mode, type **:17,6**

2. Press **RETURN**.

Note Pressing **q** or **ESC/GO** returns the diagnostic command prompt; however, it does not stop the CMC VME board from looping through the respective subtest. To free the CMC VME board from the endless loop, reboot the diagnostics.

CHAPTER 13

VME Communication Controller Board Diagnostic Tests

INTRODUCTION

This chapter describes the VME Communication Controller board (also referred to as the MPCC board) diagnostic tests.

TEST 19: VME COMMUNICATION CONTROLLER BOARD (b)

If your system contains one or more VME Communication Controller boards, test 19 can be used to ensure the functionality of the channels on the board.

This test contains 21 subtests, all of which are described in Table 13-1. Test 19 is ignored when the VME Communication Controller board is not installed. You can identify a specific Communication Controller board for testing using the two-letter **MN** command. Refer to Chapter 4 for more information on this command.

Table 13-1. VME Communication Controller (Test 19)

Subtest	Description
0*	Executes tests 1 through 12
1*	Tests MPCC onboard memory access
2*	Writes address to MPCC onboard memory
3*	Loads and boots MPCC onboard tests
4*	Tests mutual interrupts
5*	Tests MPCC onboard local memory
6*	Tests MPCC onboard CIO data path
7*	Tests MPCC onboard SCC asynchronous data path
8*	Tests MPCC onboard SCC synchronous data path
9*	Tests MPCC onboard VME DMA
10* †	Tests MPCC onboard SCC DMA
11* †	Tests MPCC onboard asynchronous external loopback test
12*	Runs subtests 5, 8, and 9 concurrently
13	Resets MPCC
14	Executes subtests 4 through 12
15	Toggles single or alternating (default) between board test mode
16	Toggles silent (default) or verbose message mode
17	Shows MPCC board configuration
18	Begins user-defined sequence test
18	Ends user-defined sequence test
20	Prints last message sent from MPCC board and TIMEOUT flag
21	Performs extensive onboard VME DMA tests

* Subtest 0 runs these subtests.

† Loopback plug required to execute subtest.

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 board (refer to the subtest 20 description later in this chapter for more information).

When a test fails, you must reload the VME Communication Controller board before you can continue. Refer to the subtest 3 description later in this section for more information. If you try to execute a subtest without rebooting, the following error message appears:

```
On bd. tester not booted.
```

Subtest 1: MPCC Memory Access

This subtest ensures that card memory can be accessed from the host. The host writes a walking bit pattern into memory, then resets memory, and tests the address lines (as discussed in subtest 2). This test takes about three minutes to complete.

Execute subtest 1 as follows:

1. Type **:19,1**
2. Press **RETURN**.

Subtest 2: Write Address and Complement to MPCC Memory

This subtest checks for address line shorts.

Execute subtest 2 as follows:

1. Type **:19,2**
2. Press **RETURN**.

Subtest 3: Load and Boot MPCC Tester

This subtest loads, and then boots, VME Communication Controller board tests into memory. You must execute this subtest before running subtests 4 through 12.

Execute subtest 3 as follows:

1. Type **:19,3**

2. Press **RETURN**.

Subtest 4: Mutual Interrupts

This test ensures that interrupts sent between the VME Communication Controller board and the host are received. In the first half of the test, the host sends an interrupt to the VME Communication Controller board, and the board sends a message back indicating whether or not it received the interrupt.

In the second half of the test, the VME Communication Controller board sends the host an interrupt, and the host indicates whether or not it received the interrupt.

Execute subtest 4 as follows:

1. Type **:19,4**

2. Press **RETURN**.

Subtest 5: MPCC Local Memory

This subtest causes the host to tell the VME Communication Controller board to run its local memory test. The test run by the board is similar to subtest 1. The board reports the status of this test back to the host.

Execute subtest 5 as follows:

1. Type **:19,5**
2. Press **RETURN**.

Subtest 6: MPCC CIO Data Path

This subtest ensures that the CIO data path and clock interrupts are operating properly.

Execute subtest 6 as follows:

1. Type **:19,6**
2. Press **RETURN**.

Subtest 7: MPCC SCC Asynchronous Data Path

This subtest tests the SCC data path for asynchronous data transmissions. Data is transmitted from the Transmit (Tx) lines (in 4KB 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 board.

Execute subtest 7 as follows:

1. Type **:19,7**

2. Press **RETURN**.

Subtest 8: MPCC SCC Synchronous Data Path

This subtest is similar to subtest 7. It verifies 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 board.

Execute subtest 8 as follows:

1. Type **:19,8**

2. Press **RETURN**.

Subtest 9: MPCC VME DMA

This test verifies that the VME Communication Controller board VME DMA is operating properly. The board 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.

Execute subtest 9 as follows:

1. Type **:19,9**

2. Press **RETURN**.

Subtest 10: MPCC OSCC DMA

This subtest requires that a 25-pin (RS-232-C) loopback connector be attached to each channel on the VME Communication Controller board before running the subtest. Refer to Appendix A for information on the loopback connector.

This subtest tests DMA on the four SCC channels on the VME Communication Controller board. First the channels are set up so that the SCC can receive and 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 Communication Controller board 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.

Execute subtest 10 as follows:

1. Type **:19,10**
2. Press **RETURN**.

Subtest 11: External Loopback

This subtest requires that a loopback plug be attached to each channel on the VME Communication Controller board before running the subtest. Refer to Appendix A for information on the loopback plug.

This subtest is similar to subtest 7. It is designed to ensure that the line drivers are operating properly.

Execute subtest 11 as follows:

1. Type **:19,11**

2. Press **RETURN.**

Subtest 12: CPU Bus Load

This subtest runs the VME Communication Controller board 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 if the CPU bus can handle the load.

Execute subtest 12 as follows:

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 board. Note that if you need to run more tests after resetting the card, you must run subtest 3.

Execute subtest 13 as follows:

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.

Execute subtest 14 as follows:

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 select whether the subtests run on one VME Communication Controller board (single mode) or alternate between all VME Communication Controller boards 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 changes from alternate to single. Specifying the subtest again changes the selection back to alternate.

Execute subtest 15 as follows:

1. Type **:19,15**

2. Press **RETURN**.

Subtest 16: Screen Message Mode

This subtest does not perform a diagnostic test. Instead, it allows you to select between viewing the messages from the subtests on the screen (verbose mode) or just receiving a message indicating that the subtest passed 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 changes from silent to verbose. Specifying the subtest again changes the selection back to silent.

Execute subtest 16 as follows:

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 board configuration is displayed on the screen, along with the modes selected in subtests 15 and 16.

Execute subtest 17 as follows:

1. Type **:19,17**

2. Press **RETURN.**

The following sample configuration appears 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

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. Refer to Chapter 4 for more information on this two-letter command.

In the test sequence you specify, subtest 18 should precede any other VME Communication Controller board subtest, and subtest 19 should follow the last VME Communication Controller board subtest.

Create a user defined test sequence as follows:

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.

Execute the user defined sequence as follows:

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, it displays the last message sent from the VME Communication Controller board flag on the screen (this message appears after the word "onboard:" in the screen message) and also displays the timeout flag value if it was set for the last test run.

Execute subtest 20 as follows:

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

Subtest 21 runs the same test sequence as subtest 9. The major difference, however, is that subtest 21 repeats each transfer up to 40 times and can take up to 90 minutes to complete.

Execute subtest 21 as follows:

1. Type **:19,21**
2. Press **RETURN.**

CHAPTER 14

Miscellaneous Diagnostic Tests

INTRODUCTION

This chapter describes the subtests contained in test 18 as shown in Table 14-1.

Table 14-1. Miscellaneous Subtests (Test 18)

Subtest	Description
1	Tests MC68881
2	Tests UPS line driver
3	Tests VME bus cycle timeout for nonpresent card
4	Checks VME protection register
5	Reads VME EEPROM pattern
6	Writes VME EEPROM pattern
7	Clears VME EEPROM (writes all zeros)
8	Sets VME EEPROM (writes all 0xFFs
9	Sets VME EEPROM type (sets type to ctix or diag)
10	Sets VME EEPROM checksum (calculates checksum and sets it) also sets type to diag
11	Sets VME EEPROM address and board type (also sets checksum)
12	Prints VME EEPROM board types and addresses
13	Clears VME EEPROM addresses and board types
14	Sets VME EEPROM 'default' table of board types
15	Saves VME EEPROM contents of EEPROM
16	Restores VME EEPROM to state before EEPROM write
17	Confirms VME EEPROM checksum correctness
20	Tests serial number ROM test (CPU-02 systems only)

The default test sequence consists of subtests 1 through 4 and 17. Execute these subtests as follows:

1. Type **z**

2. Press **RETURN.**

Subtest 0 also runs the default 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 is displayed:

```
Exceeded 20 eeprom writes for the session, skipping.
```

When this occurs, the current write test is not run. You must reboot the diagnostics to write to the EEPROM again.

Subtest 1: MC68881

This subtest checks the MC68881 floating point coprocessor. It executes a data path check on the address and data lines that connect the MC68881 and the MC68020, then performs a simple arithmetic operation on the MC68881.

Execute subtest 1 as follows:

1. Type **:18,1**
2. Press **RETURN**.

When the subtest is completed successfully, one of the following messages is displayed:

```
68881 operation compare ... passed
```

```
68881 operation compare ... FAILED
```

If, however, the MC68881 fails the first portion of the test (the data path check), the following error message is displayed:

```
68881 data path error wrote: x, read: x
```

When this occurs, the second part of the subtest is not executed.

Subtest 2: UPS Port Line Driver

This subtest tests the UPS line driver port on the Main Processor (CPU) board. This test requires a UPS loopback plug. (Refer to 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 when the signal is received correctly.

Execute subtest 2 as follows:

1. Type **:18,2**

2. Press **RETURN.**

No confirmation is displayed when the subtest is successful. If the subtest is unsuccessful, one of the following messages is displayed:

```
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 continues 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 boards 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, the following message is displayed:

```
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 the following message overwrites the message (FAILED: please reset) already on the screen:

VME bus cycle complete

Subtest 4: VME Protection Register

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.

Execute subtest 4 as follows:

1. Type **:18,4**
2. Press **RETURN**.

If the VME Protection register malfunctions, one of the following error messages is displayed:

Bus fault on user write to address (not protected)

No bus fault on user write to address (protected)

Subtest 5: VME 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.

Execute subtest 5 as follows:

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 CTIX. If so, the following message is displayed 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 CTIX, then the following error message is displayed:

```
VME EEPROM data error; x expected, read x.
```

Subtest 6: VME Write Pattern

Subtest 6, the only subtest not included in the test sequence, 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 it.

Note that you can only run the EEPROM twenty times per diagnostic session (each system boot). Refer to the section "Writing to the EEPROM" 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, the following message is displayed:

```
Checksum valid, but data set for CTIX
```

If subtest 6 is not executed as part of a test sequence, the following 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 *S/MT Series CTIX Administrator's Reference* manual to do so.

Subtest 7: VME Clear (Write All Zeros)

Subtest 7 clears the EEPROM by writing zeros to the EEPROM on the VME Interface board, then verifies that it is correct. The contents of the EEPROM are saved before the write occurs. (To restore the contents of the EEPROM, run subtest 16.)

Run subtest 7 as follows:

1. Type **:18,7**
2. Press **RETURN**.

The following output is displayed:

```
VME EEPROM clear (write all zeros)
Saving EEPROM contents...done
```

Subtest 8: VME Set (Write All 0xFFs)

Subtest 8 sets the EEPROM by writing 0xFFs to the EEPROM on the VME Interface board, then verifies that it is correct. The contents of the EEPROM are saved before the write occurs. (To restore the contents of the EEPROM, run subtest 16.)

Run subtest 8 as follows:

1. Type **:18,8**
2. Press **RETURN**.

The following message is displayed:

```
VME EEPROM set (write all 0xFFs)
Saving EEPROM contents...done
```

Subtest 9: VME 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 is displayed:

```
VME EEPROM checksum is not valid.
```

Run subtest 9 as follows:

1. Type **:18,9**
2. Press **RETURN**.

The following message is displayed:

```
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; simply pressing RETURN does not set the flag.

Subtest 10: VME 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.

Execute subtest 10 as follows:

1. Type **:18,10**

2. Press **RETURN**.

The following message is displayed:

```
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 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.

Set the address and board type as follows:

1. Type **:18,11**

2. Press **RETURN.**

The following message is displayed:

```
            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 Print the Board Types and Addresses

This subtest displays the current board types and addresses stored in the EEPROM.

Display the address and board type as follows:

1. Type **:18,12**

2. Press **RETURN**.

The following output is displayed:

```
            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 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.

Run subtest 13 as follows:

1. Type **:18,13**

2. Press **RETURN**.

The following message is displayed:

```
VME EEPROM clear the addresses and board types
Saving EEPROM contents...done
Setting EEPROM checksum
```

Subtest 14: VME 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. This subtest destroys the current contents of the EEPROM. Subtest 11 can be used to change or delete information that does not apply to your system. Use subtest 15 to save the contents before running subtest 14.

Execute subtest 14 as follows:

1. Type **:18,15**
2. Type **:18,14**
3. Press **RETURN**.

The following message is displayed:

```
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 message is displayed:

```
        Setting EEPROM checksum
        Flag is set for ctix

slot      type      address  length   int   function

(existing information about each board appears here)
```

Subtest 15: VME 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.

Save the contents of the EEPROM as follows:

1. Type **:18,15**
2. Press **RETURN**.

The following message is displayed:

```
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 is displayed 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 Restore EEPROM to State Before 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, run this subtest before ending the session.

Restore the contents of the EEPROM as follows:

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 is displayed:

```
No EEPROM restore necessary: contents were never saved
```

Subtest 17: VME 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.

Verify the EEPROM checksum as follows:

1. Type **:18,17**
2. Press **RETURN**.

The following message is displayed:

```
VME EEPROM confirm checksum correctness
```

Subtest 20: Serial Number ROM Test

Subtest 20 can be run only on CPU-02 systems. It verifies the serial number ROM of the CPU-02 board. This subtest reads the contents of the ROM, then 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.

Test the CPU-02 serial number ROM as follows:

1. Type **:18,20**

2. Press **RETURN**.

The following message is displayed:

```
Serial number ROM test
serial number; (12-digit hex number)
```

APPENDIX A

Cable and Connector Specifications

INTRODUCTION

This appendix defines pin assignments for the cables and loopback connectors (plugs) required to execute the S/MT Series diagnostic programs.

- 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
- 50-pin SCSI connector
- 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 the S/MT Series computer systems must use Convergent Technologies (or equivalent) recommended terminators and cables. The cables incorporate special bulk cable and shielding terminations (that is, 360° shielding from connector shell to connector shell with low energy 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

This cable connects DTE equipment to 25-pin connectors on the Main Processor (CPU) board. Figure A-1 gives the pin connections for this cable.

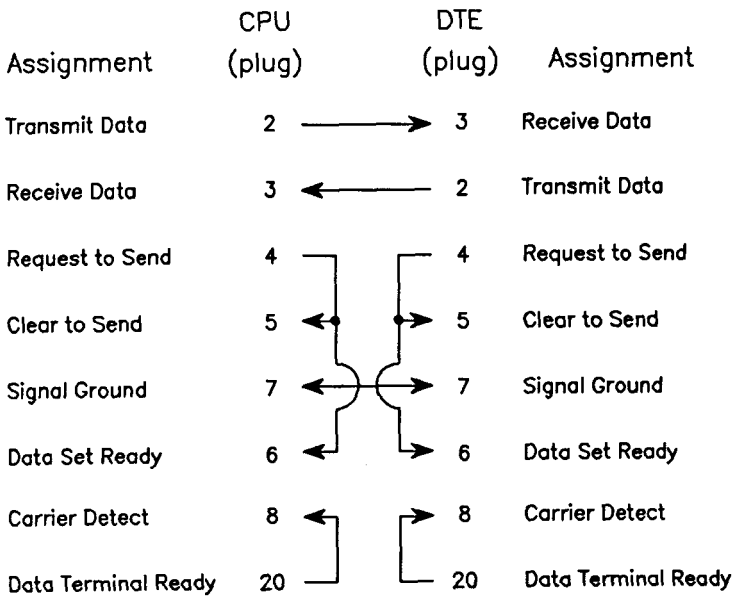


Figure A-1. Crossed RS-232-C Cable Connections

Figure A-2 illustrates how to construct this cable.

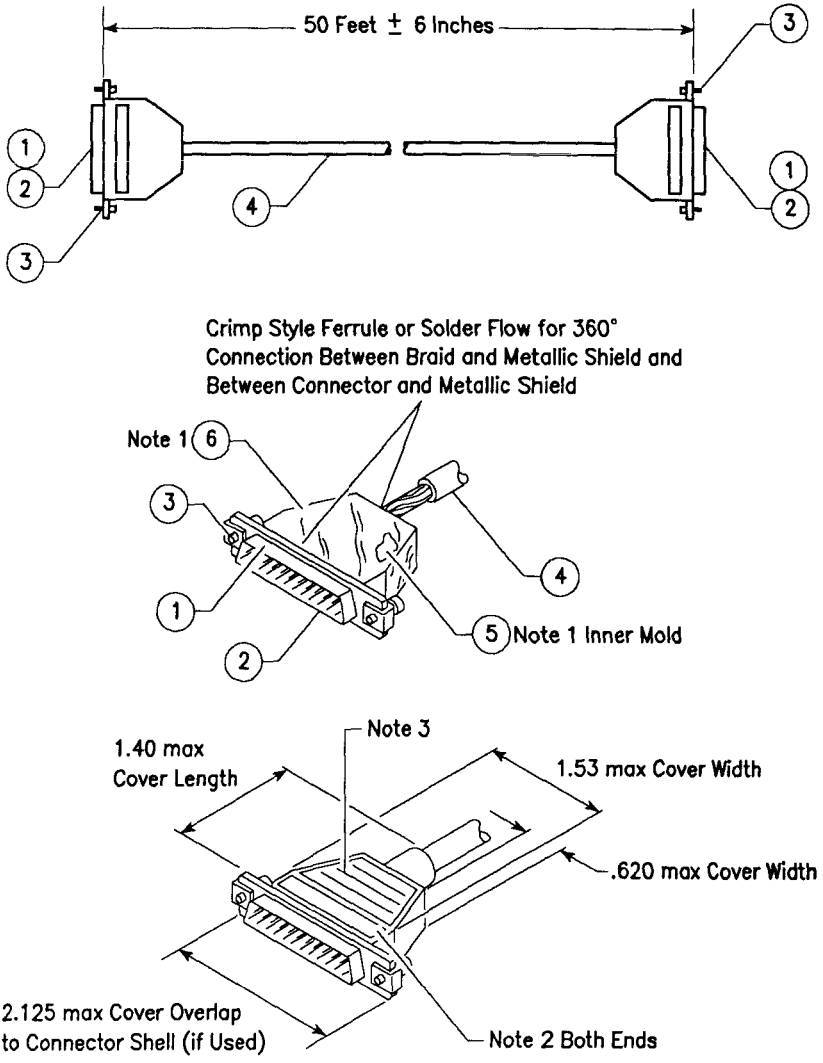


Figure A-2. Crossed RS-232-C Cable Construction

Table A-1 lists the parts required to construct this cable.

Table A-1. Crossed RS-232-C Cable Parts List

Item	Quantity	Description
1	2	25-pin plug assembly (male). Use AMP part number 207464-2 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.

1. Data terminal devices that have metal connectors use Convergent Technologies 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

This cable connects DTE equipment to an RS-232-C Expansion board. Figure A-3 gives the pin connections for this cable.

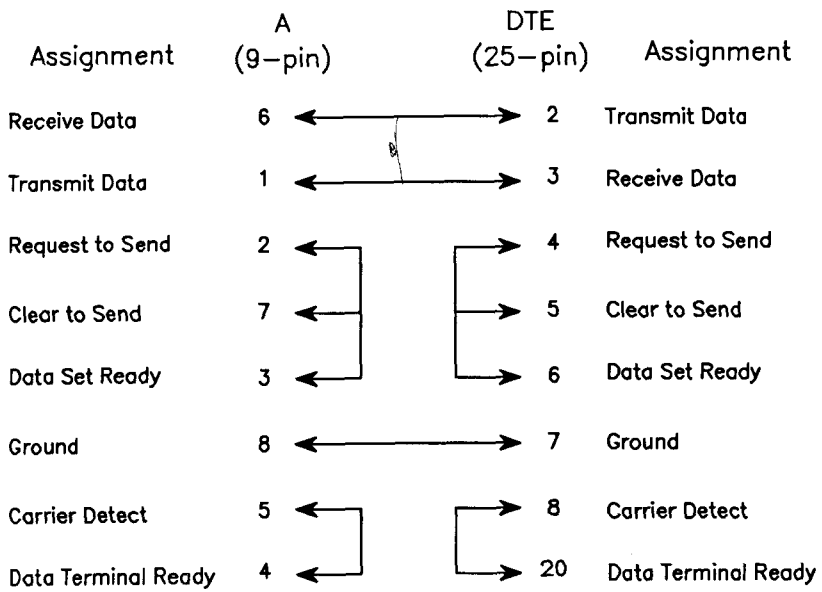


Figure A-3. 9-Pin to 25-Pin Cable Connections

Figure A-4 illustrates how to build this cable.

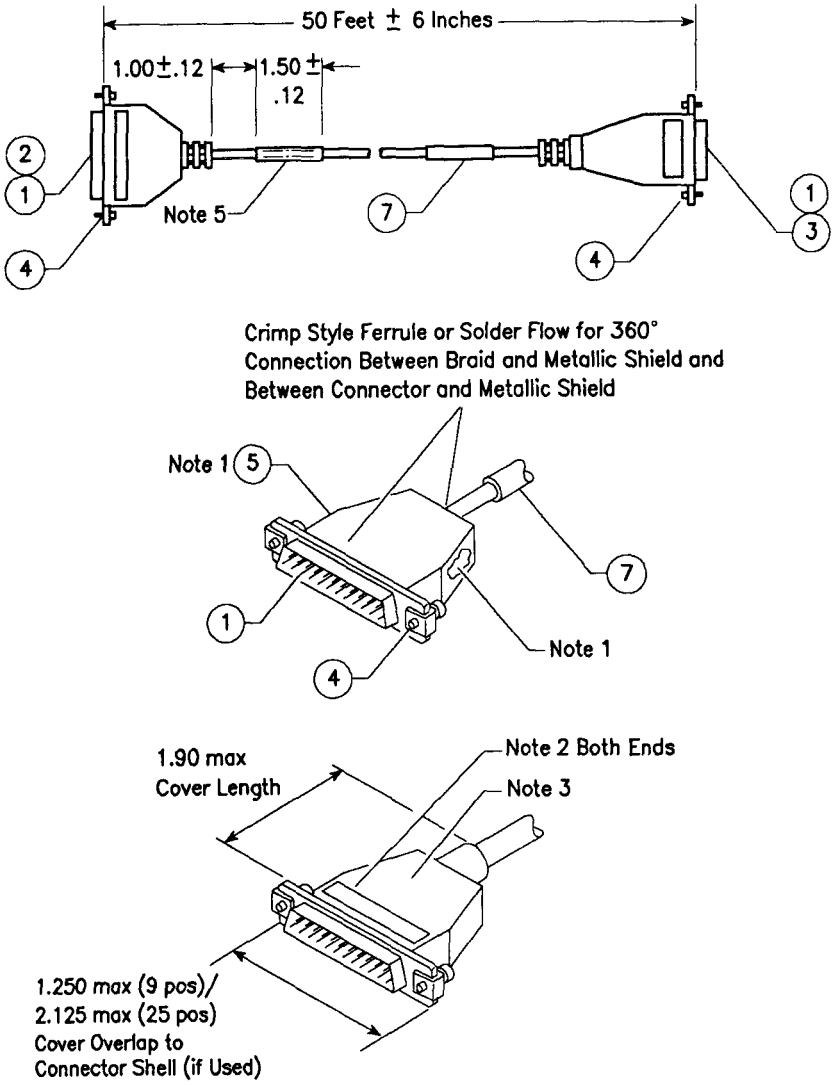


Figure A-4. 9-Pin to 25-Pin Cable Construction

Table A-2 lists the parts required to construct this cable.

Table A-2. 9-Pin to 25-Pin RS-232-C Cable Parts List

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	as required	Copper foil tape
6	2 feet	Black copper wire, 300V
7	3 feet	3-conductor shielded cable

When constructing the crossed 9-pin to 25-pin RS-232-C cable, note the following:

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

The parallel printer cable connects parallel printers to S/MT Series systems. Figure A-5 gives the pin connections for this cable.

A(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	⟷	24	LPPRESENT–
13, 15	⟷	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)

Figure A-5. Parallel Line Printer Cable Connections

Figure A-6 illustrates assembly of this cable.

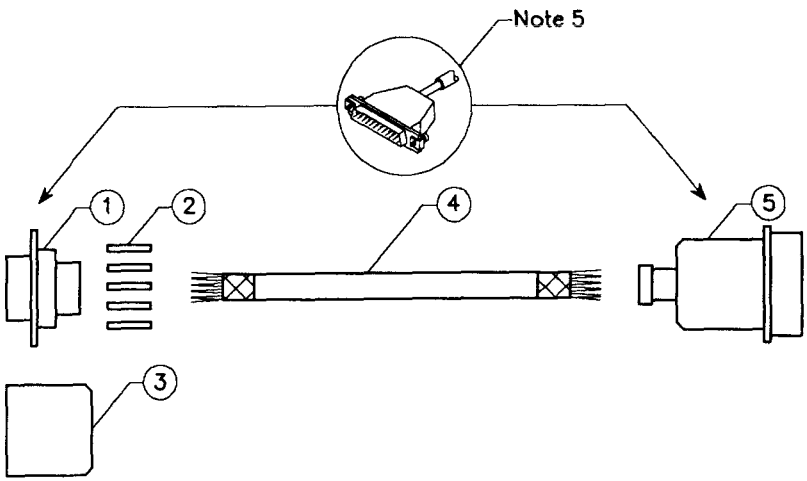


Figure A-6. Parallel Line Printer Cable Construction

Table A-3 lists the parts required to construct this cable.

Table A-3. Parallel Printer Cable Parts List

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

When constructing the parallel printer cable, note the following:

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

2. A twisted pair consists of a signal and 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 can be 10 feet (3 meters) maximum. Longer cables must be approved by the printer manufacturer.

BIDIRECTIONAL PARALLEL PRINTER PORT CABLE

The bidirectional parallel printer cable is used with the bidirectional parallel printer port diagnostic tests in CPU-02 systems. Figure A-7 gives the pin connections for this cable.

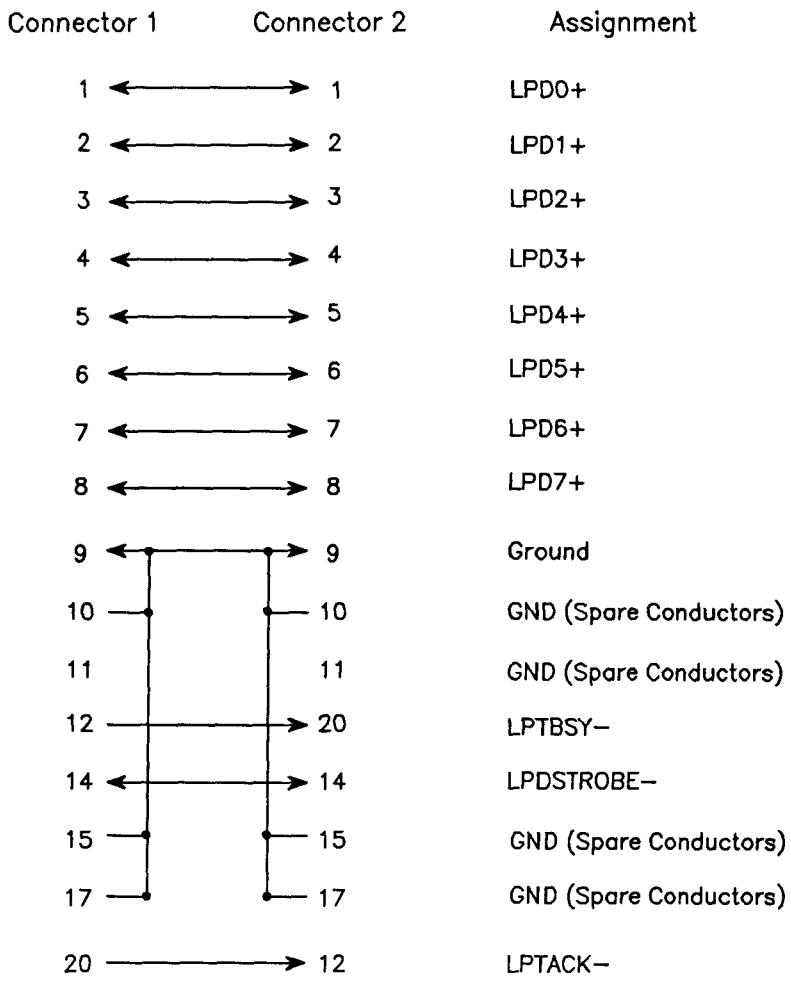


Figure A-7. Bidirectional Parallel Cable Connections

Figure A-8 illustrates assembly of this cable.

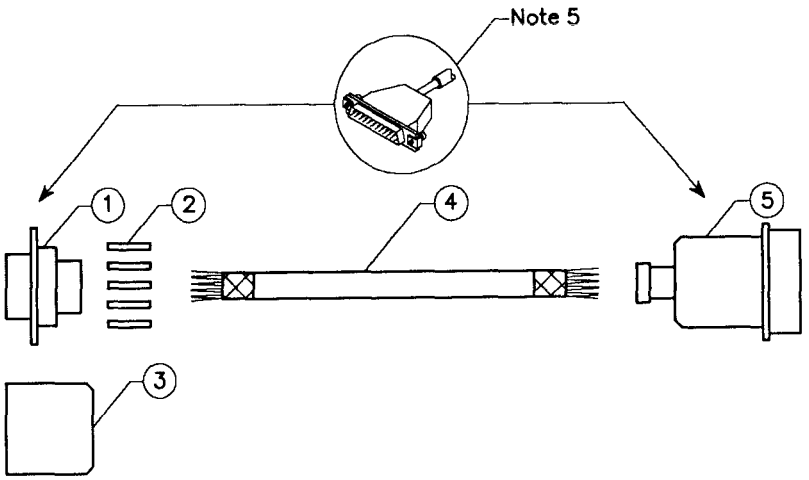


Figure A-8. Bidirectional Parallel Cable Construction

Table A-4 lists the parts required to construct this cable.

Table A-4. Parallel Printer Cable Parts List

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	5 feet	30-conductor shielded cable. Use Alpha part number 5279/15 or equivalent.
5	1	36-pin Blue Ribbon type connector assembly (male). Use AMP part number 57-30360 or equivalent.
6	as required	Copper foil (part number 80-00025).

When constructing the parallel printer cable, note the following:

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

2. A twisted pair consists of a signal and 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 can be 5 feet (1.5 meters) maximum. Longer cables must be approved by the printer manufacturer.

RS-422 CLUSTER COMMUNICATIONS CABLE

This cable connects RS-422 terminals to the RS-422 Expansion board. Figure A-9 gives the pin connections for this cable.

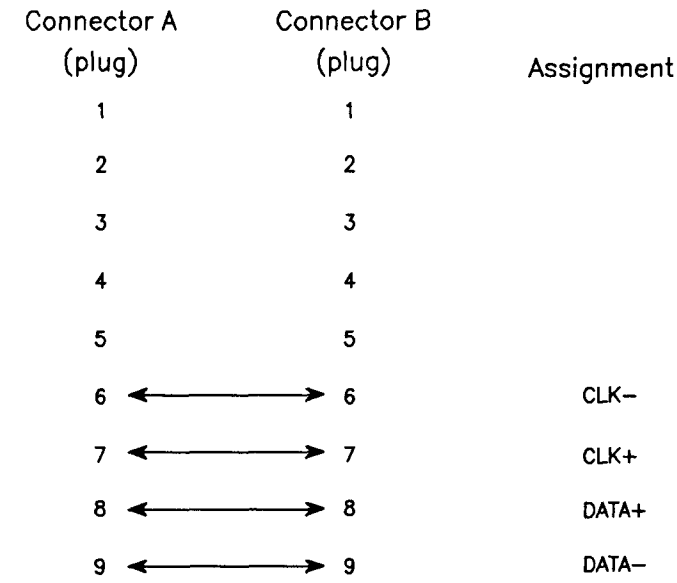


Figure A-9. RS-422 Cluster Cable Connections

Figure A-10 illustrates how to construct this cable.

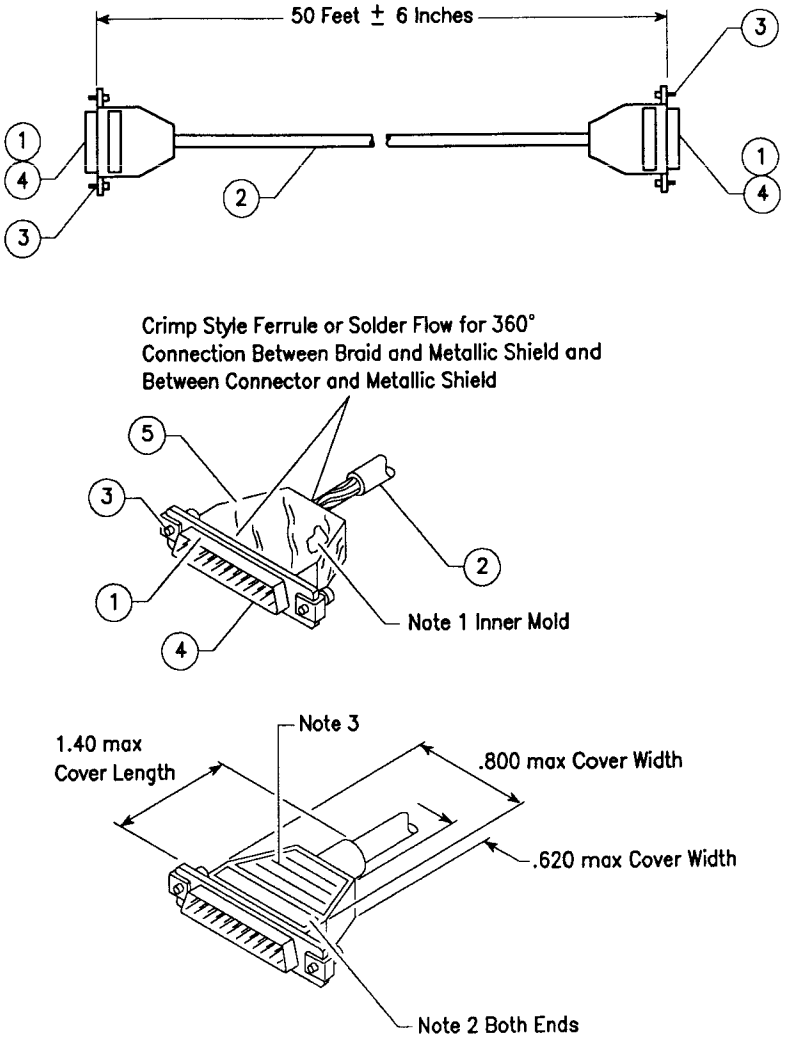


Figure A-10. RS-422 Cluster Cable Construction

Table A-5 lists the parts required to construct this cable.

Table A-5. RS-422 Cable Parts List

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	as required	Copper foil (part number 80-00025).

When constructing the RS-422 cable, note the following:

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).

50-PIN SCSI CONNECTOR

Figure A-11 shows the SCSI connector. The SCSI interface conforms to the single-ended alternative 1 SCSI standard in ANSI standard X3T9.2 (Rev. 17B or above). Refer to the *Small Computer System Interface* (SCSI) specification for more information.

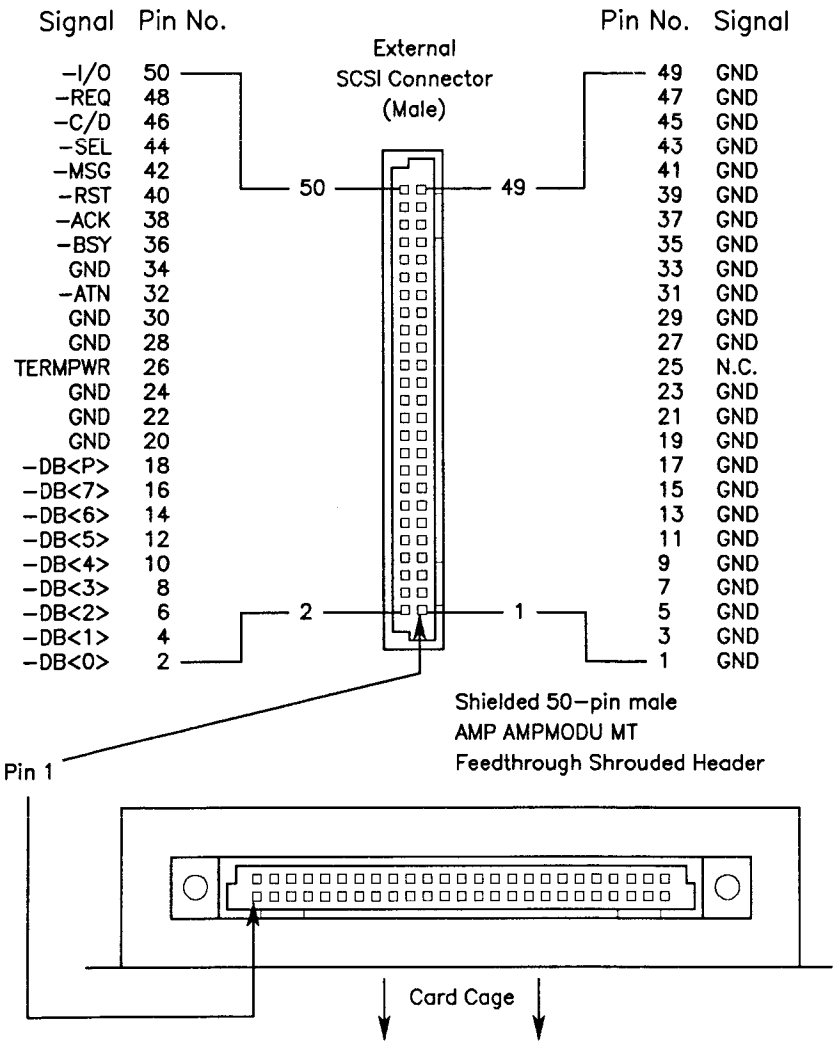


Figure A-11. 50-Pin SCSI Connector

RS-232-C 25-PIN LOOPBACK PLUG

The RS-232-C 25-pin loopback plug is used with the RS-232-C and VME MPCC board diagnostic tests. Figure A-12 gives the pin connections for this loopback plug.

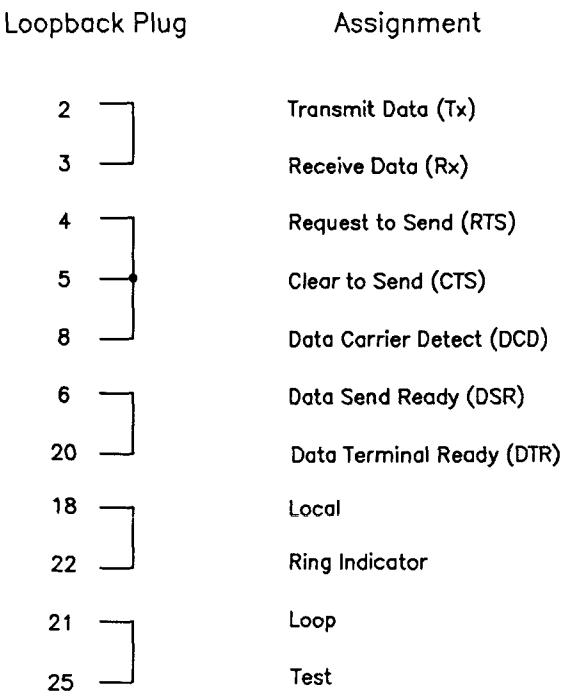


Figure A-12. RS-232-C 25-Pin Loopback Plug Connections

Table A-6 lists the parts required to construct this plug.

Table A-6. RS-232-C 25-Pin Loopback Plug Parts

Item	Quantity	Description
1	11	Contact, crimp pin
2	1	25-pin D-type connector plug
3	2	Screw and retainer (male)
4	1 foot	Copper foil tape
5	1.5 feet	Black copper wire, 26 AWG, 300 Volt

RS-232-C 9-PIN LOOPBACK PLUG

The RS-232-C 9-pin loopback plug is used with the RS-232-C diagnostic tests. Figure A-13 gives the pin connections for this plug.

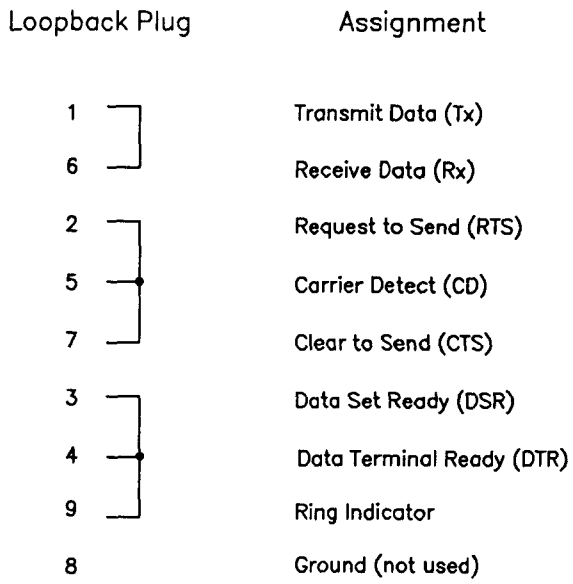


Figure A-13. RS-232-C 9-Pin Loopback Plug Connections

Table A-7 lists the parts required to construct this plug.

Table A-7. RS-232-C 9-Pin Loopback Plug Parts

Item	Quantity	Description
1	8	Contact, crimp pin
2	1	9-pin D-type connector plug
3	2	Screw and retainer (male)
4	1 foot	Copper foil tape
5	2 feet	Black copper wire, 26 AWG, 300 Volt

UPS LOOPBACK PLUG

The uninteruptible power supply (UPS) loopback plug is used with the UPS diagnostic tests. Figure A-14 gives the pin connections for this plug.

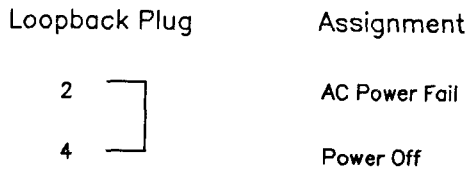


Figure A-14. UPS Loopback Plug Connections

Table A-8 lists the parts required to construct this plug.

Table A-8. UPS Loopback Plug Parts

Item	Quantity	Description
1	2	Contact, crimp pin
2	1	9-pin D-type connector plug
3	2	Screw and retainer (male)
4	1 foot	Copper foil tape
5	0.2 feet	Black copper wire, 26 AWG, 300 Volt

LINE PRINTER LOOPBACK PLUG

The line printer loopback plug is used with the line printer diagnostic tests. Figure A-15 gives the pin connections for this plug.

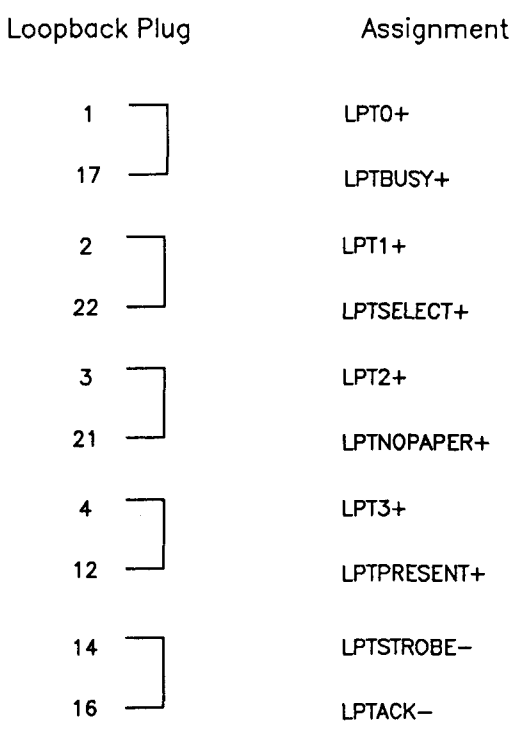


Figure A-15. Line Printer Loopback Plug Connections

Table A-9 lists the parts required to construct this plug.

Table A-9. Line Printer Loopback Plug Parts

Item	Quantity	Description
1	10	Contact, crimp pin
2	1	25-pin D-type connector plug (female)
3	2	Screw and retainer (male)
4	1 foot	Copper foil tape
5	1.5 feet	Black copper wire, 26 AWG, 300 Volt

GLOSSARY

arbitration	Settling a dispute between two or more devices, such as which device gains bus control when both devices request it.
assert	Place in logical true state. An active high signal is asserted when it is high. An active low signal is asserted when it is low.
asynchronous	A procedure, or protocol, that does not require responses at exact time intervals and clock edges, but rather within a window of time.
autovector interrupt	During an autovector interrupt acknowledge cycle, the interrupt device does not return an interrupt vector; rather, the CPU interrupt level determines the vector.
binary	Base 2 number system using 0s and 1s.

BIU Bus Interface Unit. DMA device control element that handles the local system DMA request/grant arbitration and routes DMA data between memory and holding buffers.

block Refers to 512 bytes of data.

boot A sequence of steps that brings a device up to initialization status. A boot sequence typically resides in ROM. The CPU executes the boot sequence after a reset or after a power-on.

bootstrap See **boot**.

boundary The applicable least significant address bits, generally the 3, 2, or 1 LSBs.

buffer A storage element, such as a latch DIP, that holds data intended for another location. A buffer is also a logical gate that increases power or delays a signal.

buffered A signal that is routed through a gate without changing the signal's logical level, (as opposed to inverted, which does change the signal's logical level). Signals are buffered for various reasons, including to delay them, to increase their power, or to enable them under only certain conditions (such as with a tri-state buffer).

bus cycle Any access requiring control of the local busses, for example memory or register accesses.

cache: hits and misses The lower virtual or physical address bits address the PTC and memory cache. The remaining upper address bits are compared with the tag field obtained from the addressed cache entry. If the compare is equal, and the tag valid bit is set, a cache hit occurs. If the compare is unequal, or the tag valid bit is not set, a cache miss occurs. See also **tag valid bit.**

concatenation Combining address fields obtained from different elements to obtain a single address, such as a PTC page pointer and virtual byte index combined to obtain the physical memory address.

CRC Cyclic Redundancy Check. Less complicated method (than ECC) of data encoding designed to guarantee data reliability.

CTIX Popular multiuser, multitasking, virtual memory operating system from Bell Labs. (The Bell Labs version is UNIX; CTIX is a proprietary version of UNIX.)

data control PAL and buffer logic that controls the routing of data bytes between main memory, the memory cache, the CPU, and DMA devices.

data overrun Condition where a master transmits data and the slave cannot receive it before the master transmits more data.

data path check Diagnostic test method that uses a "walking 0 and walking 1" technique for verifying the data paths.

data underrun Condition where a slave does not supply data requested by a master in time.

demand paged Refers to virtual pages being brought into physical memory from the disk as required by the currently operating software (either a user application or the operating system), and then returned to the disk when not needed, all of which is invisible to a user.

diagnostics Programs designed to determine correct functionality of hardware and also to aid partially in fault location.

dirty bit Bit for each location in the memory cache. The bit is set when the cache location is written to (during either a cache hit or a miss).

DMA Direct Memory Access. A DMA is a direct access to memory without requiring CPU intervention. (Note that in some systems, such as a VME bus system, any bus access to memory, including a CPU access to memory, is considered DMA.)

DMA mapper The virtual-to-physical address translator for DMA accesses. It maps a 16MB DMA space to (up to) 64MB of physical memory. Each DMA mapper entry contains two pieces of information, a page pointer and page permission bits.

doorbell interrupt — A handshake protocol where a device interrupts another device, and the device being interrupted responds by servicing the interrupt and resetting the interrupt request on the device generating the interrupt. A timeout may or not be implemented.

ECC — Error Correction Coding. Data reliability coding performed on and appended to data during a write. During a read, the ECC code is compared with the read data to verify that the data is unchanged.

group table — The first level of the MMU. It is implemented onboard. It contains 16 entries, each supplying a 14-bit segment table pointer (the starting address of a segment table). See also **PTC miss**.

handshake — Protocol, such as a data transfer handshake, where a master device asserts a trigger signal (such as Transfer Ready), then waits for a slave device to respond within a specified period with some form of acknowledge.

hexadecimal — Base 16 number system that uses letters A through F for 10, 11, 12, 13, 14, and 15. Thus, 1Ah equals 16 + 10 = 26.

I/O Input/Output. I/O involves data transfers with peripherals external to the computer, such as disk drives or terminals. An I/O device or I/O register enables I/O transfers to occur.

interrupt Signal or field sent to the CPU by a device to indicate a timing, status, or error condition exists.

Kernel Name referring to the CTIX operating system program. This program implements the computer software processes and furnishes the means for interprocess communication,

LED Light Emitting Diode. An LED is an indicator that lights to indicate a condition. LEDs can be under hardware control, software control, or both.

line Eight contiguous and addressable bytes of data. All main memory accesses are line accesses, although data control may only buffer a portion of the line to or from the memory cache, the CPU, or a DMA device. Memory cache data entries are lines.

longword Four bytes.

LSB Least Significant Bit.

main memory Actual memory locations implemented in RAM on expansion boards. Main memory and the memory cache constitute physical memory.

main system One of two S/MT Series systems, the main system and the VME subsystem. The main system contains all non-VME bus locations.

mapped Refers to fields that are converted to obtain a corresponding field or positioning, such as mapping virtual to physical addresses, or mapping register bit locations to LED positions.

master Device that controls a bus for a given activity, such as a DMA transfer. The DMA master outputs an address, address strobe and data strobe(s), and other signals defined by the bus, then waits for the device addressed (the slave) to respond. (A typical response is a data strobe acknowledge). The master always initiates the bus activity; the slave always responds.

memory cache A 128KB cache that holds the most recently CPU-accessed 128KB of physical memory. (DMA cache misses do not cause a cache update.) Each memory cache entry contains four pieces of information: eight bytes of data (a line), a dirty bit, a tag field, and a tag valid bit. See also **cache: hits and misses.**

negate Place in logical false state. An active high signal is negated when it is low. An active low signal is negated when it is high.

offboard Refers to a hardware location that is on a different board than the one being discussed. Also refers to a procedure that originates on a different board, such as offboard DMA.

onboard Refers to a hardware location that is on the board being discussed. Also refers to a procedure that originates on the board, such as onboard DMA.

page In a virtual system, memory is sectioned off into pages of a set length. The contents (data and code) of most of the pages are located on a disk; only a very small percentage of pages actually reside in physical memory. The programmer does not distinguish between pages that are on a disk or in memory. A physical page refers to a page that is in physical memory. A virtual page is the page addressed by the programmer. Both types of pages are 4096 bytes in S/MT Series systems.

page pointer The upper 14 bits of the physical memory address. It points to one of (up to) 16K physical pages. The pointer is concatenated with the lower 12 bits of the virtual address, the byte offset, to obtain the complete 26-bit physical address. The page pointer is obtained from either the PTC (PTC hit), page table (PTC miss), or DMA mapper (DMA cycle).

page table Third level of the MMU. It resides in physical memory. The page table contains 16 entries, each entry containing a 14-bit physical page pointer and page permission bits for one physical page. See also **page pointer**.

Page Table Cache — See **PTC.**

per-character interrupt — I/O controller generated interrupt for single character exchange with a peripheral.

physical address — Memory address obtained from concatenation of the page pointer and the byte index.

physical memory — Actual memory locations implemented in hardware. The 128KB memory cache and main memory constitute physical memory.

process working set — Set of physical pages (pages that are a part of processes currently being executed by the CPU).

protocol — Activity involving two or more devices, where the devices must act in accordance with a predefined sequence of steps. Acquiring bus control in a multi-bus-device system requires following an arbitration protocol. Exchanging data involves a data protocol.

PTC The PTC is an 8KB cache that holds physical page addresses of the most recently accessed 2K virtual pages. Each PTC location contains four entries: page pointer, page permission bits, tag field, and tag valid bit. Each valid PTC entry contains the page pointer and permission bits corresponding to the same in the page table. See also **page table** and **page pointer.**

PTC miss A PTC miss causes an access to the MMU tables. The group table supplies the 14-bit segment table pointer, which is concatenated with the 10-bit segment index, to obtain the physical address to the segment table entry. (The lower two address bits to segment and page table entries always = 00b). The segment table supplies a 20-bit page table pointer, which is concatenated with the 4-bit page table index, to address the page table. The page table supplies the 14-bit physical page pointer. See also **concatenation.**

QIC Quarter Inch Cartridge.

RAM Random Access Memory. RAM is physical memory that can be read or written.

ROM Read-Only Memory. ROM is memory that can be read but not written. It comes preprogrammed, for example, as a computer bootstrap sequence.

segment table The second level of the MMU. It resides in physical memory and contains 1024 entries, each containing a 20-bit page table pointer and two page table permission bits. See also **PTC miss.**

sequencer A device that executes a series of two or more states, where a state defines a certain point in some activity or operation. Other devices, such as a decoder, can be directed by the sequencer to perform state-defined specific actions.

serial A serial data stream is a single bit stream, with a new bit transferred at each clock.

slave See **master.**

state machine A device that executes a sequence of states defined by a binary field. In some or all of these states, the machine decodes inputs to generate one or more outputs. State machines are frequently implemented by PALs.

supervisor One of two MC68020 CPU operating modes. Supervisor mode contains the complete CPU instruction set and typically accesses hardware locations not available to user mode, where locations are used to implement and maintain the operating system. See also **user**.

synchronize To cause the critical rising or falling edges (or both edges) of a signal to occur simultaneously with other signal edges. Synchronizing may also ensure that a signal occurs at an expected time in a sequence, such as at a certain point in a bus cycle.

synchronous A procedure, or protocol, that requires a response at an exact time and clock edge.

tag A descriptor field. In a CPU-02 system, tag fields are used to qualify the corresponding virtual and physical memory addresses, respectively, of PTC and memory cache locations. The tag valid bit indicates if the tag field is valid. See also **tag valid bit**, **PTC**, **memory cache**, and **cache: hits and misses**.

tag valid bit Each cache location contains a valid bit that indicates if the corresponding tag field at that location is valid. If the valid bit is not set, a cache miss occurs. See also **cache: hits and misses.**

terminal count The condition where the requested number of transfers (or events) have occurred, indicated by a counting element asserting a terminal count output.

timeout Period of time in which a required response must occur before an error condition or termination of operation occurs.

tri-state A third state available to a signal in addition to high or low. A tri-stated signal appears as a high impedance with no effect on other devices connected to the line, and any other device driving the line high or low controls the line. Tri-stated lines are often used when a signal comes from more than one source.

trigger Signal or group of signals that activates a device from idle, wait, or off status.

user One of two MC68020 CPU operating modes. User mode contains a less powerful instruction set than supervisor and is designed mainly for applications processing. User also refers to a programmer that executes user-mode applications. See also **supervisor.**

vector An address that points to a location, which itself contains an address that the CPU ultimately branches to.

virtual address Address output by the CPU. A virtual memory address contains a virtual page address and byte offset. The byte offset is the same for the physical page address.

virtual memory Memory address space seen by the processor. The contents of virtual memory are located either on the disk or in physical memory.

VME subsystem One of two S/MT Series systems, the main system and the VME subsystem. The VME subsystem contains all VME bus locations. See also **main system.**

wait state State of a processor or controller where the device waits for a response, without changing any internal or external status.

wait states Wait states are states where the CPU suspends activity until it receives a prompt (usually an acknowledge) to complete a cycle. The access speed of a device accessed by the CPU often determines how many wait states the CPU executes.

write back policy Policy where a memory cache write miss cycle causes current data in the cache at the addressed location to be written back to memory if the data is dirty (was written to since it was loaded into the cache). A write back policy compares with a write through policy, where data is immediately written through to memory during any cache write, thus making write backs unnecessary.

Index

C (continued)

C (continued)

D

D (continued)

E

F

G, H

I

J, K

L

M

M (continued)

N, O, P

Q, R

S

T

T (continued)

U

V, W, X, Y, Z