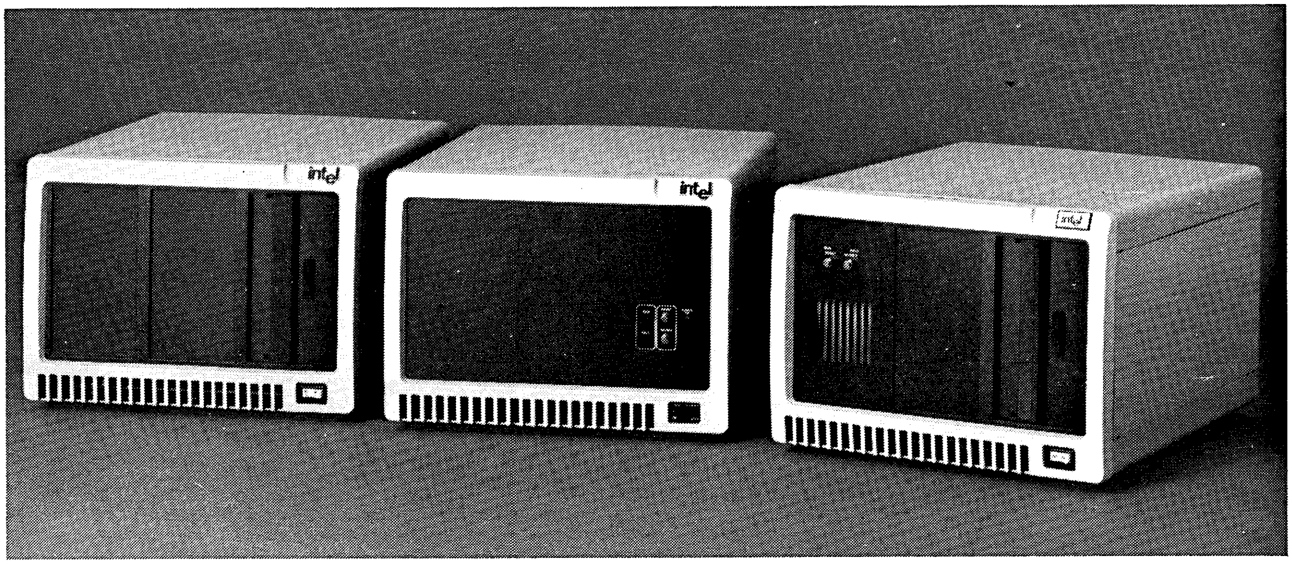
SYSTEM 86/300 SERIES DIAGNOSTIC MAINTENANCE MANUAL



Intel Corporation, 3065 Bowers Avenue, Santa Clara, California 95051

Order Number: 144813-001

SYSTEM 86/300 SERIES DIAGNOSTIC MAINTENANCE MANUAL

Order Number: 144813-001

Intel Corporation, 3065 Bowers Avenue, Santa Clara, California 95051

REV.	REVISION HISTORY	PRINT DATE
-001	Original issue. Contains information formerly found in SYSTEM 86/300 SERIES DIAGNOSTIC REFERENCE MANUAL. This issue updates the SCT, adds the SDT8630 test, and updates the remaining SDT test suites.	6/82

Additional copies of this manual or other Intel literature may be obtained from:

Literature Department
Intel Corporation
3065 Bowers Avenue
Santa Clara, CA 95051

The Information in this document is subject to change without notice.

Intel Corporation makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Intel Corporation assumes no responsibility for any errors that may appear in this document. Intel Corporation makes no commitment to update nor to keep current the information contained in this document.

Intel Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in an Intel product. No other circuit patent licenses are implied.

Intel software products are copyrighted by and shall remain the property of Intel Corporation. Use, duplication or disclosure is subject to restrictions stated in Intel's software license, or as defined as ASPR 7-104.9(a)(9).

No part of this document may be copied or reproduced in any form or by any means without the prior written consent of Intel Corporation.

The following are trademarks of Intel Corporation and its affiliates and may be used only to identify Intel products:

BXP
CREDIT
i
ICE
iCS
im
iMMX
Insite
Intel
Int$_{e}$l
Intelevision
Intellec
iOSP
iRMX
iSBC
iPDS
iSBX
Library Manager
MCS
Megachassis
Micromainframe
Mircromap
Multibus
Multichannel
Multimodule
Plug-A-Bubble
PROMPT
RMX/80
System 2000
UPI

PREFACE

This manual describes the three diagnostic software tools supplied with the System 86/300 Series Microcomputer Systems: the System Confidence Test (SCT), the System Analysis Test (SAT), and the System Diagnostic Test (SDT). These tests check the system hardware and determine the system software's ability to run on that hardware.

WARNING

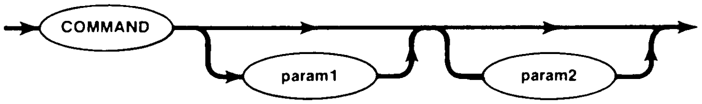
Dangerous voltages are present withing the System. Before removing or installing boards or devices, disconnect the power cord. In addition, only qualified technical personnel should attempt to modify System hardware. Failure to observe these precautions can result in personal injury and circuit damage.

NOTATIONAL CONVENTIONS

This manual uses the following notational convention to illustrate syntax.

UPPERCASE	Uppercase information must be entered exactly as shown. You can, however, enter this information in uppercase or lowercase.
lowercase	Lowercase fields contain variable information. You must enter the appropriate value or symbol for variable fields.
underscore	In examples of dialog at the terminal, user input is underscored to distinguish it from system output.
[ ]	Brackets surround optional parameters.
<>	Angle brackets surround optional fields in messages displayed by the diagnostics.

Chapter 4 of this manual uses the "railroad track" schematic to illustrate the syntax of the SDT commands. This syntax consists of what looks like an aerial view of a model railroad setup, with syntactic elements scattered along the track. To interpret the command syntax, you start at the left side of the schematic, follow the track through all the syntactic elements you desire (sharp turns and backing up are not allowed), and exit at the right side of the schematic. The syntactic elements that you encounter, separated by spaces, comprise a valid command. For example, a command that consists of a command name and two optional parameters would have the following schematic representation:



x-067

You could enter this command in any of the following forms:

```
COMMAND
COMMAND param1
COMMAND param2
COMMAND param1 param2
```

The arrows indicate the possible flow through the tracks; they are omitted in the remainder of the manual.

RELATED PUBLICATIONS

The following manuals provide additional information that may be helpful to users of this manual.

Manual	Number
System 86/330A Overview Manual	144680
System 86/330A Installation and Maintenance Manual	144777

Manual	Number
User's Guide for the iSBC® 957B iAPX 86, 88 Interface and Execution Package	143979
PL/M-86 User's Guide	121636
iAPX 86, 88 Family Utilities User's Guide	121616
iRMX™ 86 Nucleus Reference Manual	9803122
iRMX™ 86 Basic I/O System Reference Manual	9803123
iRMX™ 86 Extended I/O System Reference Manual	143308

CONTENTS

PAGE

CHAPTER 1
GENERAL DESCRIPTION

CHAPTER 2
SYSTEM CONFIDENCE TEST

CHAPTER 3
SYSTEM ANALYSIS TEST

CHAPTER 4
SYSTEM DIAGNOSTIC TEST

CONTENTS (continued)

CONTENTS (continued)

FIGURES

TABLES

CHAPTER 1. GENERAL DESCRIPTION

INTRODUCTION

There are three groups of diagnostic test packages used to determine the operational readiness of the System 86/300 Series Microcomputer Systems. These diagnostic packages are:

- System Confidence Test
- System Diagnostic Test
- System Analysis Test

SYSTEM CONFIDENCE TEST

The System Confidence Test (SCT) provides a means to quickly ascertain if the System is exhibiting any catastrophic errors. It resides in PROM on the processor board and is automatically invoked when you first apply power to the system or when you press the front panel RESET pushbutton. The SCT is described in more detail in Chapter 2.

SYSTEM DIAGNOSTIC TEST

The System Diagnostic Test (SDT) provides a means of isolating a problem to a specific board or drive by permitting the operator to specify certain test(s) and parameters in order to determine which board is not working. The SDT is described in more detail in Chapter 4.

SYSTEM ANALYSIS TEST

The System Analysis Test (SAT) provides a means to interactively exercise the system hardware with the iRMX 86 Operating System for extended periods of time. This permits isolation of elusive intermittently-reported errors (subtle problems) to a given area within the System. You can use the SAT only when the iRMX 86 Operating System is running in your system. The SAT is described in more detail in Chapter 3.

RECOMMENDED TEST SEQUENCE

This section describes the recommended sequence of running the diagnostic tests. It assumes that your system uses the iRMX 86 Operating System, which is a requirement for running the SAT. If your system uses the Xenix operating system (Xenix is a trademark of Microsoft), you cannot run the SAT. In such a case, you should ignore references to the SAT. However, you can run the SCT and SDT regardless of the operating system you use.

The recommended sequence of running the diagnostic tests is:

1. Run the SCT whenever you apply power or reset the system (this happens automatically).
2. Run the SAT when you first receive the System and whenever you change or repair the System. This ensures that the hardware and the iRMX 86 software work together.
3. Run the SDT test suites whenever you experience a problem with the SCT or SAT and whenever you suspect there is a problem with a particular device or subassembly. The SDT test suites further isolate problems.

The following sections discuss this process in more detail.

RUNNING THE SCT

The SCT resides in PROM and is invoked whenever you apply power to the system or press the RESET pushbutton. Therefore, it is always the first test to run. The SCT provides a basic check of System hardware.

RUNNING THE SAT

Upon receipt of your System, you should run the SAT immediately after the SCT, even if SCT completes successfully and you suspect no problems. The SAT checks the operational readiness of the System and ensures that both the hardware and the iRMX 86 software function together. Later, you can use the SAT as an occasional confidence test, or you can use it whenever the occasion warrants to help isolate intermittent problems.

If the SAT reports an error, it points to a malfunctioning area within the system, but it is not definitive. For example, if the SAT reports an error while performing a Basic I/O System (BIOS) operation or an Extended I/O System (EIOS) operation to the Winchester disk, it does not specifically mean that the Winchester drive is defective. Such an error potentially could be caused by several different kinds of malfunctioning hardware. You must run the SDT test suites to isolate the problem further.

Other errors that can be reported by executing the SAT are errors associated with an I/O operation to the flexible disk drive or errors associated with the 8087 Numeric Processor E[illegible]on (NPX). In addition, SAT errors can indicate problems with memory (both addressing and size), the compilation of the tasks, the linking of the files, the device drivers, etc. The errors reported as a result of running the SAT point to a particular problem area within the System. Therefore, if the SAT reports any errors, you should invoke the SDT to determine the specific problem within the System. Upon successful completion of the SAT, the System is available for system use.

RUNNING THE SDT TEST SUITES

If you encounter a problem as a result of running the SCT or SAT, you should invoke the appropriate test suite of the SDT to aid in the isolation of the problem to a specific board or drive. If the SDT cannot be invoked due to problems with the drive or the inability to communicate with the terminal, check the configuration of all boards in the System (refer to the INSTALLATION AND MAINTENANCE MANUAL for your System for configuration information). If the configuration is correct, replace the suspect board or drive (refer to Table 2-1). After replacement of the defective part, invoke the SDT.

WARNING

> Dangerous voltages are present within the System. Before removing or installing boards or devices, disconnect the power cord. In addition, only qualified technical personnel should attempt to modify System hardware. Failure to observe these precautions can result in personal injury and circuit damage.

You can use the information returned by the SCT and SAT to determine a specific SDT test suite to invoke. You can then invoke individual routines within a particular test suite to further isolate problems. Execution of a specific test suite provides more meaningful data with which to determine the failing board or drive.

If the the SDT reports a problem, check the configuration of all boards in the System. If the boards are in their default configurations, replace the suspect board, drive, or cable. After fixing the problem, invoke the SDT test suite again. If no errors occur, invoke the SAT to ensure that repairs made to one portion of the System have not affected other portions.

CHAPTER 2. SYSTEM CONFIDENCE TEST

INTRODUCTION

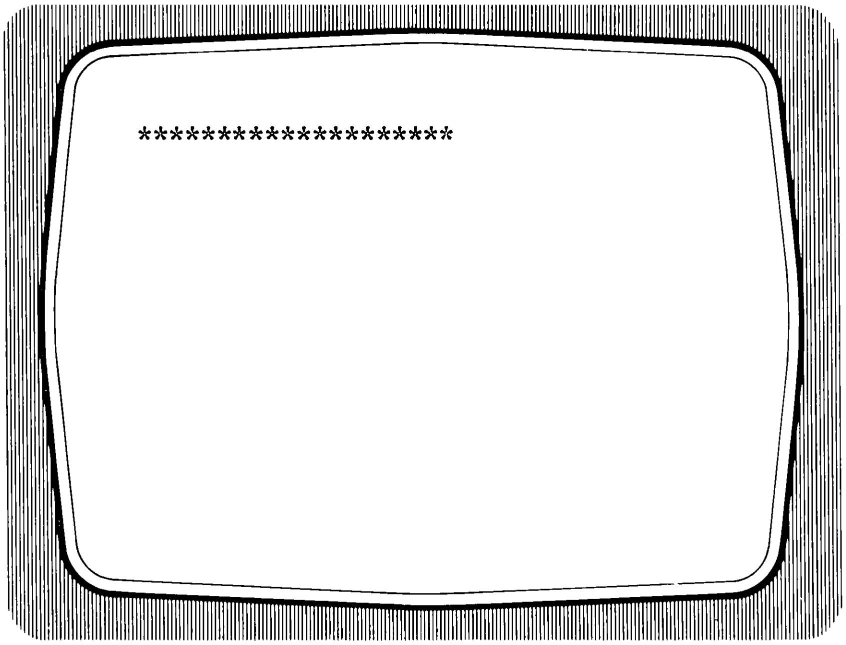
The System Confidence Test (SCT) provides a level of diagnostic testing that determines if major components of the System are malfunctioning. It receives control upon system power-up or reset.

The SCT resides in PROM on the processor board and is co-resident with the Operating System's Bootstrap Loader and the iSBC 957B monitor. It interfaces with other software only upon termination, passing control to the Bootstrap Loader if no errors were encountered. If it finds an error, it passes control to the iSBC 957B monitor.

Upon initialization, or as a result of pressing the front panel RESET switch pushbutton, the System Confidence Test (SCT) is initiated.

To invoke the System Confidence Test, proceed as follows:

1. With the user-supplied terminal turned on and connected to the System, turn the System AC power switch to ON. (After about 5 seconds delay, the CRT displays a series of asterisks. See Figure 2-1. Note that asterisks might not be displayed if the terminal is not set for 9600 baud. If nothing appears on the CRT within 10 seconds, go to step 2. This time varys depending on the amount of RAM in the system. The more RAM, the longer it takes for the asterisks to appear. If you do not set the baud rate to 9600 on the attached terminal, the SCT will execute, but the Operating System will not. The preconfigured Operating System requires that the terminal be set for 9600 baud in order to operate.)



```
********************
```

Figure 2-1. Initial Display

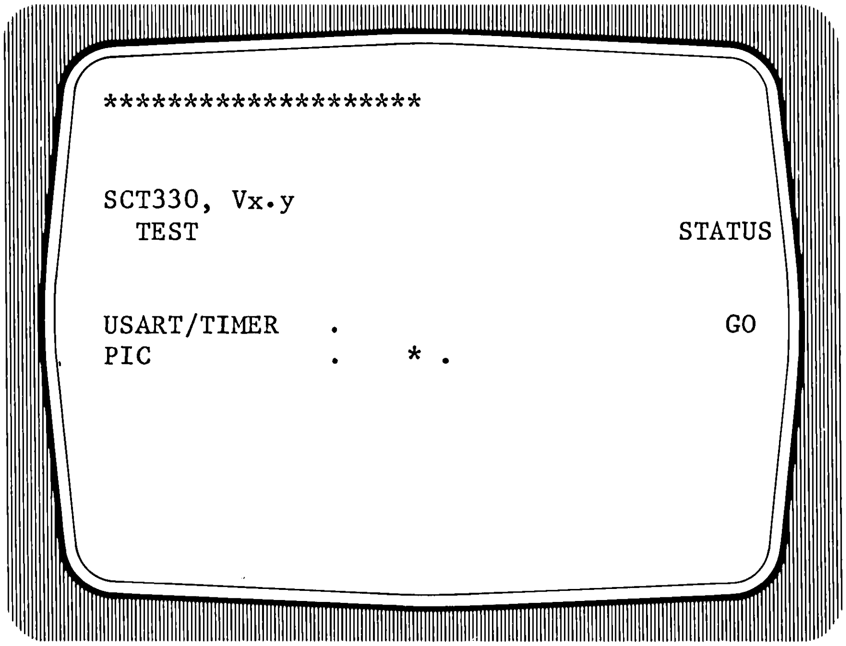
2. Enter:

   An uppercase U. (Press both the SHIFT key and then the u key.)

   This invokes the baud rate search and the SCT. The SCT then displays a header message and some additional information. After displaying the line containing the characters "PIC", it stops and waits for you to enter a character in response to this PIC test (see Figure 2-2).

NOTE

After entering the uppercase U, you have six seconds to respond to the PIC test by entering a character. If you fail to respond within six seconds, the PIC test times out and presents a NO-GO for the test results. In such a case, you can press the front panel RESET button to restart the SCT.



```
********************


SCT330, Vx.y
  TEST                                  STATUS


USART/TIMER    .                          GO
PIC            .    * .
```

Figure 2-2. Display of PIC Test Prompt

3. Enter:

   Any character except Control-D

   The SCT continues running, displaying results as it proceeds. Refer to Figure 2-3 for a display of a successfully-completed SCT.

When the SCT runs, it either flashes the front panel lights (to indicate a problem in communicating with the terminal) or it attempts to run all its tests. This means that errors occurring in early tests can propogate through to later tests. If the SCT reports a number of errors, correct the errors in the order they are listed in the SCT display. The later errors may be caused by the same things that caused the earlier errors; thus they may disappear when you correct the first errors.

After completion of the SCT, the Bootstrap Loader loads the Operating System from the Winchester, assuming that the SCT test results are "GO" and that the Operating System file resides on the Winchester.

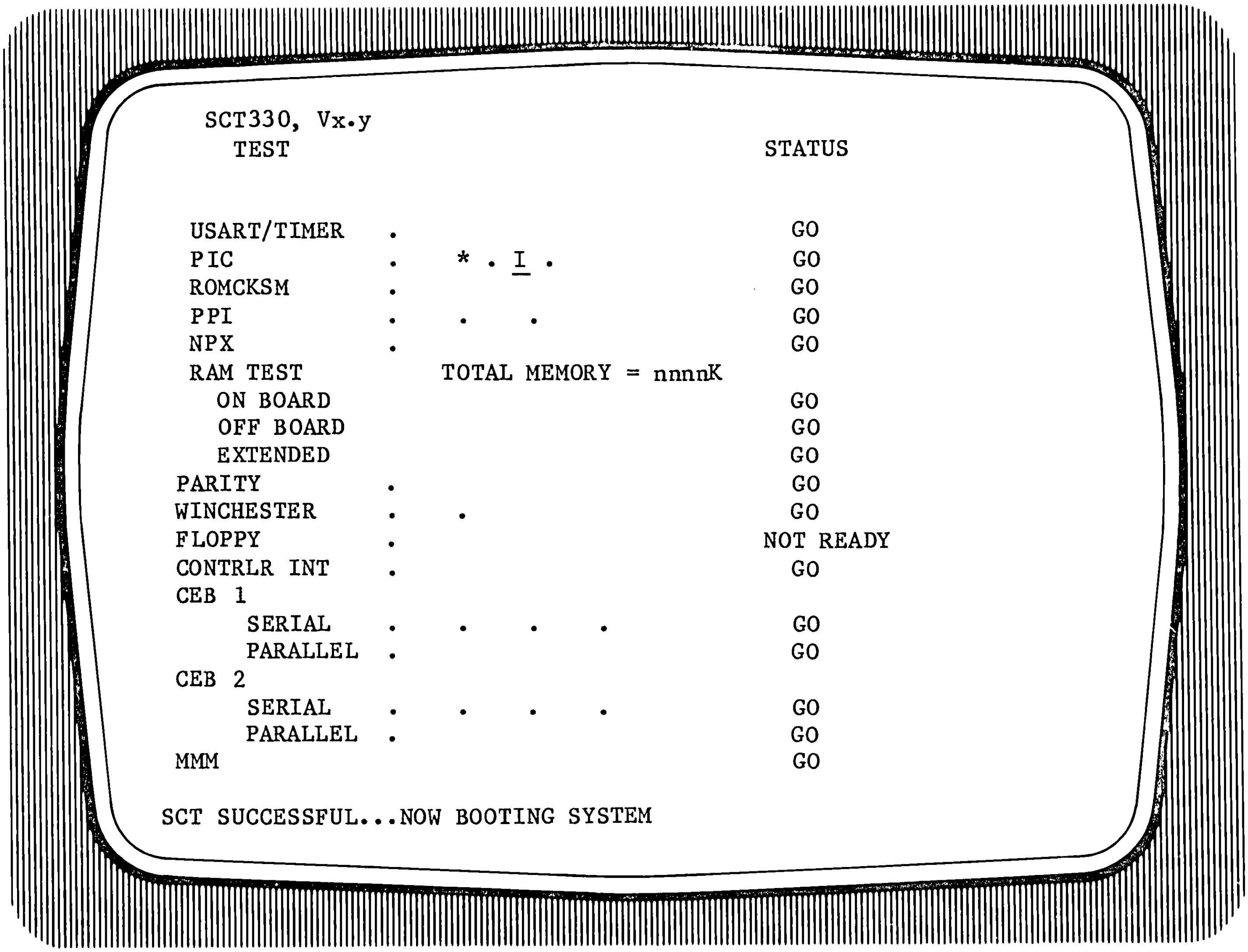
The progress of each routine within a specific test is indicated either by a period (.) which indicates successful completion or by a question mark (?) which indicates that an error occurred. Figure 2-3 depicts a successfully-completed SCT test. Table 2-1 defines an abnormal test result related to the question mark°s position within the CRT display. It also lists the error encountered and the failing subassembly.

Before performing the corrective actions listed in Table 2-1, always check the configuration of the boards in the system. It is possible that an invalid jumper connection on one board could cause the SCT to return an error message for any number of tests. Refer to the INSTALLATION AND MAINTENANCE MANUAL for your system to determine the proper board configuration.

WARNING

Only qualified technical personnel should perform the corrective actions listed in Table 2-1. Attempts by unqualified personnel to service the hardware can result in personal injury and damage to the system. Always disconnect the power cord prior to installing or removing a board or peripheral device. Failure to observe this precaution can result in personal injury and circuit damage.

The display in Figure 2-3 contains optional portions that might not appear when you run the SCT. The optional portions are the NPX (Numeric Processor Extension) test, the CEB (Communications Expansion Board) tests, and the MMM (Memory Management Module) test. If you remove the 8087 NPX from your system, the NPX test does not display information at the terminal. If your system does not include iSBC 534 Communication Expansion Boards or an iSBC 309 Memory Management Board, the CEB and MMM tests do not display information at the terminal.



```
 SCT330, Vx.y
   TEST                                    STATUS


  USART/TIMER   .                            GO
  PIC           .    * . I .                 GO
  ROMCKSM       .                            GO
  PPI           .    .    .                  GO
  NPX           .                            GO
  RAM TEST         TOTAL MEMORY = nnnnK
    ON BOARD                                 GO
    OFF BOARD                                GO
    EXTENDED                                 GO
 PARITY         .                            GO
 WINCHESTER     .    .                       GO
 FLOPPY         .                          NOT READY
 CONTRLR INT    .                            GO
 CEB  1
      SERIAL    .    .    .    .             GO
      PARALLEL  .                            GO
 CEB  2
      SERIAL    .    .    .    .             GO
      PARALLEL  .                            GO
 MMM                                         GO

SCT SUCCESSFUL...NOW BOOTING SYSTEM
```

---

nnnnK = decimal number of K bytes found in the system

Figure 2-3. Successfully-Completed SCT Test Results

Table 2-1. Abnormal SCT Test Results

TEST	POSITION 1 2 3 4	MEANING	CORRECTIVE ACTION *
USART		If GO is not displayed	Replace processor board
PIC	?	TMRO INT did not occur	Replace processor board
	. ?	Transmit INT did not occur	Replace processor board
	. . ?	Receive INT did not occur	If key was pressed, replace processor board
ROMCKSM	?	Checksum variation	Replace processor board
PPI	?	Failure at Port A	If iSBC 957B download link is connected, this might not be an error; otherwise, replace processor board
	. ?	Failure at Port B	Replace processor board
	. . ?	Failure at Port C	Replace processor board
NPX	?	8087 did not respond	Replace processor board, iSBC 337 board, or 8087
RAM TEST			
ONBOARD		NO GO	Replace processor board
OFFBOARD		NO GO	Replace iSBC 056A board
EXTENDED		NO GO	Replace user-added memory board
PARITY		Parity Error	Replace iSBC 056A board
WINCHESTER	?	Initialization error	If RAM test also fails, replace RAM board; otherwise, perform following steps, running SDT215 after each step to determine if error still occurs: 1. Ensure that head is unlocked 2. Run DISKVERIFY to check disk format 3. If damaged, reformat disk (this destroys data on disk) 4. Replace disk controller board 5. Replace Winchester drive or boards

* If more diagnostic information is needed, invoke the appropriate System Diagnostic Test (SDT).

Table 2-1. Abnormal SCT Test Results (continued)

TEST	POSITION				MEANING	CORRECTIVE ACTION *
	1	2	3	4		
WINCHESTER (con't)	•	?			iSBC 215 Diagnostic	Perform following steps, running SDT215 after each step to determine if error still occurs: 1. Ensure that head is unlocked 2. Run DISKVERIFY to check disk format 3. If damaged, reformat disk (this destroys data on disk) 4. Replace disk controller board 5. Replace Winchester drive or boards
FLOPPY					NOT READY--Door Opened (does not prevent loading of Operating System)	Insert diskette, close door, press RESET
	?				Unformatted Diskette or disk controller reported an error	Perform following steps, running SDT215 after each step, to determine if error still occurs: 1. Run DISKVERIFY to check disk format 2. If damaged, reformat disk (this destroys data on disk) 3. Replace disk controller board 4. Replace disk drive
CONTRLR INT	?				Interrupt from iSBC 215 did not occur	If PIC test passed, replace controller board; otherwise, replace processor board

* If more diagnostic information is needed, invoke the appropriate System Diagnostic Test (SDT).

Table 2-1. Abnormal SCT Test Results (continued)

TEST	POSITION 1 2 3 4	MEANING	CORRECTIVE ACTION *
CEB (1 and 2)			
SERIAL	?	Serial channel 1 error	Replace iSBC 534 board
	. ?	Serial channel 2 error	Replace iSBC 534 board
	. . ?	Serial channel 3 error	Replace iSBC 534 board
	. . . ?	Serial channel 4 error	Replace iSBC 534 board
PARALLEL	?	Parallel port C error	Replace iSBC 534 board
MMM	?	Memory management flag error	Replace iSBC 309 board
* If more diagnostic information is needed, invoke the appropriate System Diagnostic Test (SDT).			

If these tests terminate normally, control transfers to the Bootstrap Loader. Should an error be detected, a message is issued to the resident serial I/O channel of the processor board and control transfers to the iSBC 957B monitor at completion of the test.

TEST DESCRIPTIONS

The following sections describe the individual SCT tests in more detail.

TEST 1. USART/TIMER

This test checks the processor's ability to communicate with its on-board USART and thus establish a communication path via the RS232 serial I/O port to the user-installed CRT/keyboard.

The USART/TIMER test routine initializes the 8251A USART and the 8253 Programmable Interval Timer (PIT) on the processor board. The status word from the USART is validated. If the status word is valid, a message "GO" is printed out on the attached CRT. If the status word is invalid, the HALT and RUN lights on the front panel flash indicating a defective processor board.

With communication established with the USART, the attached CRT can display the SCT results.

TEST 2. PIC INITIALIZATION

The second test checks the ability of the Programmable Interrupt Controller (PIC) on the processor board to generate the appropriate interrupt levels. Table 2-2 lists the various interrupt level assignments and Table 2-3 lists the interrupt vector address locations.

Table 2-2. Interrupt Assignments

Level	Description	Priority
NMI	Not used	8 (Highest)
IR0	Floating point exception	7
IR1	Multibus interrupt 1 - console	6
IR2	On-board timer	5
IR3	Available to the user	4
IR4	Line Printer	3
IR5	Multibus interrupt 5 - disk	2
IR6	Serial I/O Receive	1
IR7	Serial I/O Transmit	0

Table 2-3. Interrupt Vector Addresses

Interrupt	Monitor	iRMX 86
NMI	00008H	00008H
0	00080H	000E0H
1	00084H	000E4H
2	00088H	000E8H
3	0008CH	000ECH
4	00090H	000F0H
5	00094H	000F4H
6	00098H	000F8H
7	0009CH	000FCH

This test initializes the 8259A PIC. After it sets the mode and operation, it sets and reads the interrrupt mask register. If the masks are not correct, a single "?" is displayed on the terminal and the test terminates. If the masks are correct, individual interrupts are invoked (i.e. timer, USART transmit, USART receive). This test awaits an input from the user at the terminal. If no input is received within seven seconds, a NO-GO message is generated. You may enter any character except a Control-D.

TEST 3. ROM CHECKSUM

This test verifies that the address and data lines of the on-board ROM are intact. The on-board ROM address space spans address locations 0F8000h through 0FFFFFh.

This test accesses all ROM locations and calculates a checksum of their contents. The calculated checksum is compared with the recorded checksum stored in ROM. If the contents match, a "GO" message appears on the CRT. If the contents do not match, the "NO-GO" message appears.

TEST 4. PPI INITIALIZATION

This test checks the processor's ability to communicate with its on-board Programmable Peripheral Interface (PPI). It checks all three ports of the PPI.

The PPI 8255 is initialized and a value is sent to each of its three ports. Each port is then read. The word read is compared with that sent. If they compare, the message "GO" appears. If they do not compare, the message "NO-GO" appears.

NOTE

> If your system is connected to an Intel Microcomputer Development System via the iSBC 957B package, this test will report a failure at port A and return a "NO-GO" message. This is a normal occurrence and does not necessarily indicate a problem with your hardware.

TEST 5. NPX (OPTIONAL TEST)

This test checks the processor's ability to communicate with the 8087 Numeric Processor Extension (NPX). It writes a floating-point number to the stack of the NPX and compares the source number with the value on the NPX stack. If they compare, the message "GO" appears. If they do not compare, the message "NO-GO" appears.

If the 8087 NPX is not present in the system, this test displays no information.

TEST 6. RAM TEST

This test checks the ability of the processor to communicate with RAM. The dual port RAM spans address locations 00000H through 1FFFFH. The off-board RAM spans address locations 20000H through 5FFFFH. Added memory is automatically checked. But, the address of the memory added must be contiguous. A "GO" message is still displayed after the Extended memory test even if additional memory is not added.

If no errors are detected, a "GO" message appears on the terminal. If an error is detected, a "NO-GO" message appears.

TEST 7. PARITY

This test verifies that the RAM memory board can detect a parity error and generate the appropriate response. The routine writes a test value to a RAM location while the memory parity controller is set to one format, reinitializes the controller to the opposite format, and attempts to read the original value. This should generate a parity error which causes the message "GO" to appear on the CRT. If no parity error occurs, the message "NO-GO" appears on the CRT. The parity error is cleared before proceeding.

Note that a parity error does not generate an interrupt. If the user desires to take advantage of the parity interrupt logic within the iSBC 056A board, then an interrupt connection must be made and an interrupt handler provided by the user.

TEST 8. WINCHESTER

This test verifies that the processor can communicate with the disk controller. The processor communicates with the disk controller via four control blocks (linked lists) located in memory on the RAM memory board. If the RAM Test returned a "NO-GO" message, this test may also return a "NO-GO" message due to its inability to access the control blocks.

This test initializes the iSBC 215 Winchester disk controller and invokes the controller micro-diagnostics. These two operations also test the controller's ROM and RAM. After each function, the status word from the controller is examined. If the Winchester drive did not spin up, the controller returns an invalid status word for the initialization function. Receipt of a valid status word causes the "GO" message to appear. If the status word is invalid, the message "NO-GO" appears.

If the Winchester is not formatted, if it is formatted incorrectly, or if the diagnostic track is not formatted, a "NO-GO" message appears.

TEST 9. FLOPPY

This test checks the iSBC 218 flexible disk controller without destroying data on the flexible diskette. It is identical to the Winchester initialization test except for device numbers. If an error is detected, the message "NO-GO" appears on the CRT. If you have an iRMX 86-formatted diskette or a Xenix bootable diskette (Xenix is a trademark of Microsoft) in the diskette drive and no errors are encountered, the message "GO" appears. An unformatted diskette or a diskette with an invalid format causes a "NO-GO" message to appear. If a diskette is not present, a NOT READY message appears. This message should be interpreted as a GO-type message, because the iRMX 86 Operating System is still bootstrap loaded even if the diskette is not present. If the flexible disk drive is not connected (off line), a NOT READY message appears.

TEST 10. CONTRLR INT

This test verifies that the disk controller can generate an interrupt. If an interrupt is not received, the message "NO-GO" appears on the CRT. If no errors are encountered, the message "GO" appears.

TEST 11. CEB (FOR OPTIONAL BOARDS)

This test checks the processor's ability to communicate with off-board USARTs. It tests the USARTs on one or two iSBC 534 Communication Expansion Boards. It initializes and examines the status word of each USART on the iSBC 534 board. It also checks the processor's ability to communicate with port C of the Communication Expansion Board's 8255A PPI (as in Test 4). If any status word is invalid, or if the processor cannot communicate with the PPI, the test displays a question mark (?) and a "NO-GO" message at the terminal. If all four status words and the PPI are valid, the test displays the "GO" message.

This test displays information only when one or more iSBC 534 boards are present in the system. It displays the "CEB 1" information for an iSBC 534 board whose I/O base address is jumpered for 30h. It displays the "CEB 2" information for an iSBC 534 board whose I/O base address is jumpered for 40h. If no iSBC 534 board is present, this test displays no information.

TEST 12. MMM (FOR OPTIONAL BOARDS)

This test verifies that the iSBC 309 Memory Manager board is present and in a valid state. If the board is present, the test reads the system request and trap flags. If the flags indicate an invalid state, the test displays the "NO-GO" message; otherwise it displays the "GO" message.

If the iSBC 309 board is not present in the system, this test displays no information.

TROUBLESHOOTING HINTS

Whenever the SCT indicates an error, check the configuration of the individual boards before replacing any of the hardware. An invalid jumper configuration on one board might cause the tests for that board to return errors. However, the same invalid configuration might instead cause other tests to return errors. Refer to the INSTALLATION AND MAINTENANCE MANUAL for your system to determine the proper board configurations.

If no asterisks appear at the terminal during the SCT test, first check the baud rate switches on the terminal to ensure that they select 9600 baud. If this does not correct the problem, replace the processor board. If this fails, check the terminal and cable. If there is still no display, remove the RAM Memory board and the disk controller and rerun the test. There could be a conflict between the priorities of the boards that are bus masters, causing a bus contention problem.

If (after replacing the disk controller board and verifying that the correct voltage is present) the Winchester drive still fails to spin up during the SCT test, check the cables and the RAM Memory board. In order for the Winchester drive motor to spin up, the address jumper configuration of the RAM memory board has to be correct.

CHAPTER 3. SYSTEM ANALYSIS TEST

INTRODUCTION

The System Analysis Test (SAT) tests the ability of the iRMX 86 Operating System to interactively communicate with the various hardware subassemblies of the system. This test requires the iRMX 86 Operating System to be running in your System. If you use a different operating system, you should skip this chapter; you cannot run the SAT.

When invoked, the SAT performs the following operations:

1. Invokes the PL/M-86 Compiler to compile one of the SAT test modules.
2. Invokes LINK86 to link the module's object code with other SAT modules and iRMX 86 interface libraries.
3. Using the linked test, the SAT creates an iRMX86 job and several test tasks.

The SAT test then runs, displaying status messages once every minute. Table 3-1 lists the SAT tests.

Table 3-1. SAT Tests

Test	Function
SAT report test	Tests the ability to communicate with the user's terminal via the USART.
BIOS Winchester test	Tests the ability to use iRMX 86 Basic I/O System (BIOS) calls to communicate with the Winchester drive.
BIOS Floppy test	Tests the ability to use iRMX 86 Basic I/O System (BIOS) calls to communicate with the flexible disk drive.
EIOS Winchester test	Tests the ability to use iRMX 86 Extended I/O System (EIOS) calls to communicate with the Winchester drive.

Table 3-1. SAT Tests (continued)

Test	Function
EIOS Floppy test	Tests the ability to use iRMX 86 Extended I/O System (EIOS) calls to communicate with the flexible disk drive.
iSBC 337 test	Tests the ability to perform numerical operations via the 8087 Numeric Processor Extension.

SAT INSTALLATION

All of the files required for running the SAT reside in the directory SATDIR on the diskette marked "SYSTEM 86/300 SERIES DIAGNOSTICS # 1", as released with the System. Table 3-2 lists these files.

Table 3-2. SAT Files

File	Description
SATTEST.P86	PL/M 86 source code for the SAT tests
SATTST.LIT	Literal definitions for the SATTEST.P86 module
SAT.LIT	Literal definitions for the SATTEST.P86 module
SAT.EXT	External declarations for the SATTEST.P86 module
SAT.LIB	Library which contains the other SAT test modules
LINK.CSD	SUBMIT file to link SAT test program
HPIFC.LIB	Human Interface compact interface library
EPIFC.LIB	EIOS compact interface library
IPIFC.LIB	BIOS compact interface library
RPIFC.LIB	Nucleus compact interface library

Upon receipt of your System, you must copy these files from the diagnostic diskette to a directory called SATDIR which must reside in your default directory (USER) on the Winchester disk. The diagnostic diskette contains a SUBMIT file called SATINSTALL.CSD which performs this operation. To invoke this SUBMIT file (and install the SAT files on the Winchester disk), perform the following:

1. After installing the iRMX 86 Operating System software onto the Winchester disk (refer to the INSTALLATION AND MAINTENANCE MANUAL for your system), run the SCT and bootstrap load the Operating System by pressing the front panel RESET button.

2. Insert the diskette marked "SYSTEM 86/300 SERIES DIAGNOSTICS # 1" into the flexible disk drive.

3. Enter the following:

```
-SUBMIT :FD0:SATINSTALL.CSD
```

   (This copies all necessary SAT files to the Winchester disk and places them in a directory called SATDIR in your default directory. It takes about 3 minutes to complete.)

After SAT installation, control returns to you under the iRMX 86 Operating System. You can then invoke the SAT as described in the next section.

If the Operating System software is not installed on the Winchester disk, you can bootstrap load the correct version of the Operating System from the flexible disk, as follows:

1. Press the INTRPT button on the front panel to interrupt into the monitor.

2. Insert the diskette marked "INSTALLATION" into the flexible disk drive.

3. Enter the following command:

```
.b :WF0:/SYSTEM/RMX86.WD0
```

INVOKING THE SAT

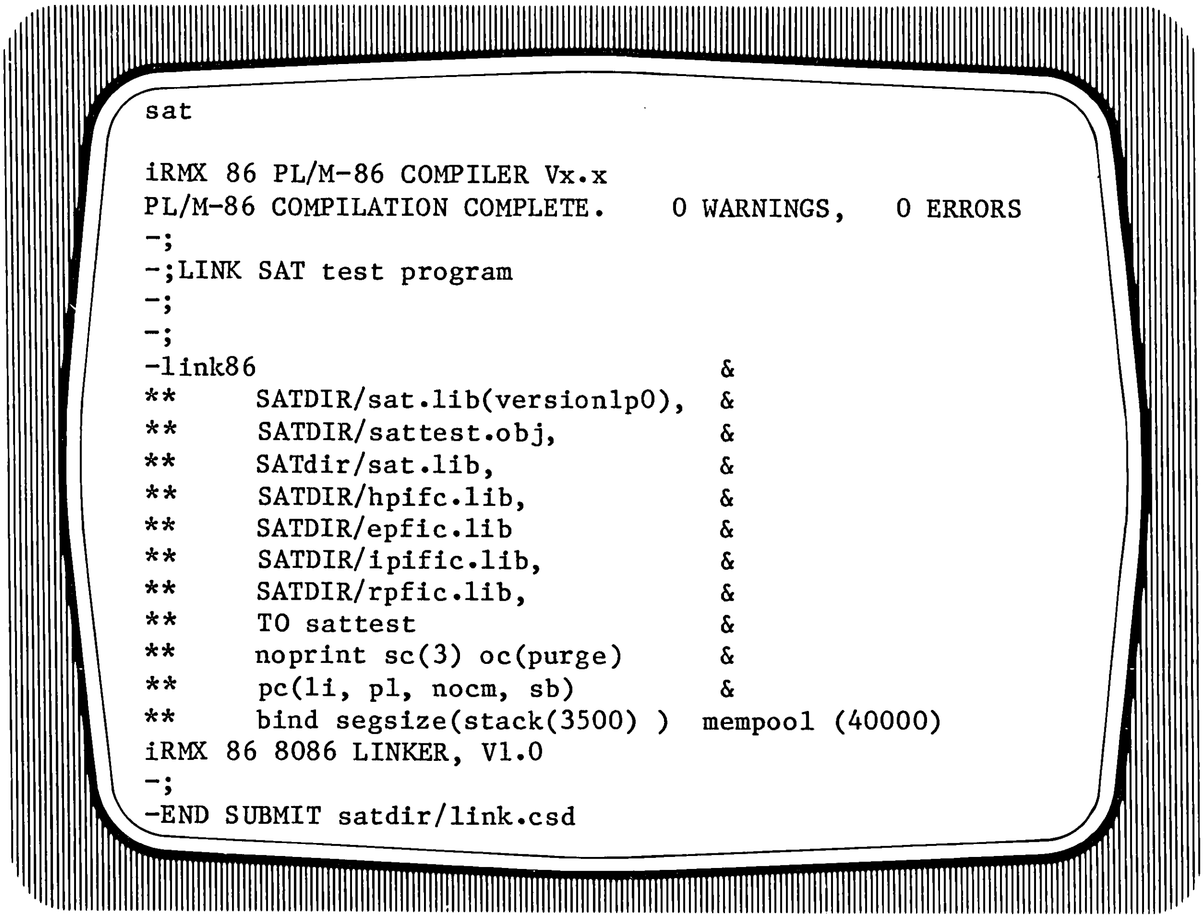
In order for the SAT to function, the correct version of the Operating System must be running in your system and the Winchester disk must contain the necessary SAT files. The PL/M-86 compiler and LINK86 must also reside on the Winchester disk. That being the case, you can invoke the SAT by entering the following command:

```
SAT [FOREVER]
```

where:

FOREVER	An optional parameter that exercises the SAT tests forever. If you include the FOREVER parameter (or any abbreviation), you can exit the SAT only by entering a Control-C at the terminal. If you omit this parameter, the SAT runs for a default time of five minutes.

After invocation, the SAT displays at the terminal the information listed in Figure 3-1. This information indicates the status of the first portion of the SAT: the compiling and linking of the test program. If no errors occur during the first portion, the SAT tests begin running. However, if an error occurs, the SAT stops and returns control to you at the terminal.



```
sat

iRMX 86 PL/M-86 COMPILER Vx.x
PL/M-86 COMPILATION COMPLETE.    0 WARNINGS,    0 ERRORS
-;
-;LINK SAT test program
-;
-;
-link86                                &
**      SATDIR/sat.lib(version1p0),    &
**      SATDIR/sattest.obj,            &
**      SATdir/sat.lib,                &
**      SATDIR/hpifc.lib,              &
**      SATDIR/epfic.lib               &
**      SATDIR/ipific.lib,             &
**      SATDIR/rpfic.lib,              &
**      TO sattest                     &
**      noprint sc(3) oc(purge)        &
**      pc(li, pl, nocm, sb)           &
**      bind segsize(stack(3500) )  mempool (40000)
iRMX 86 8086 LINKER, V1.0
-;
-END SUBMIT satdir/link.csd
```

Figure 3-1. Initial SAT Display

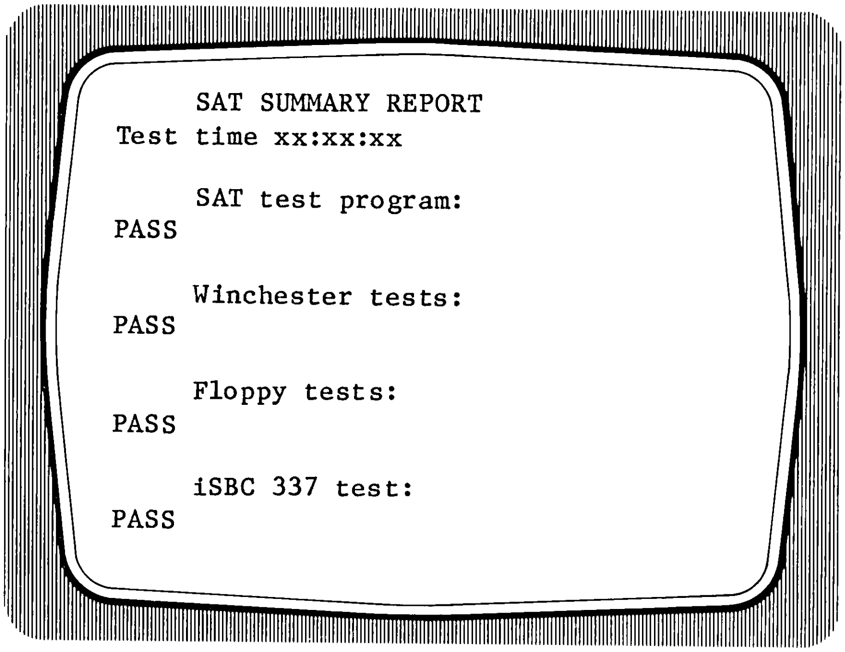
The SAT is a GO/NO-GO test. If an error occurs in the SAT, you should run SDT tests to obtain more information. SAT error messages are created by the iRMX 86 Operating System. To fully understand them requires a knowledge of the iRMX 86 Operating System and the code. The error messages provide the following information:

- the test that failed
- the operation (system call) that failed
- the exception code
- the number of times the failure occurred

If an error occurs during the compilation portion of the SAT, you can obtain more information about the error by examining the program listing. The compiler writes this list information to a file named SATDIR/SATTEST.LST. Examine this file to obtain more information. Refer to the PL/M-86 USER'S GUIDE for more information.

If an error occurs during the link portion of the SAT, you can obtain more information about the error by examining the map file. The linker writes the link map to a file named SATDIR/SATTEST.MP1. Refer to the iAPX 86, 88 FAMILY UTILITIES USER'S GUIDE for more information about the LINK86 errors.

If no errors occur during the compilation and link phases, the SAT tests start running. Once every minute, they display status information at the terminal in a SAT Summary Report. When the tests run successfully, this information appears as listed in Figure 3-2.

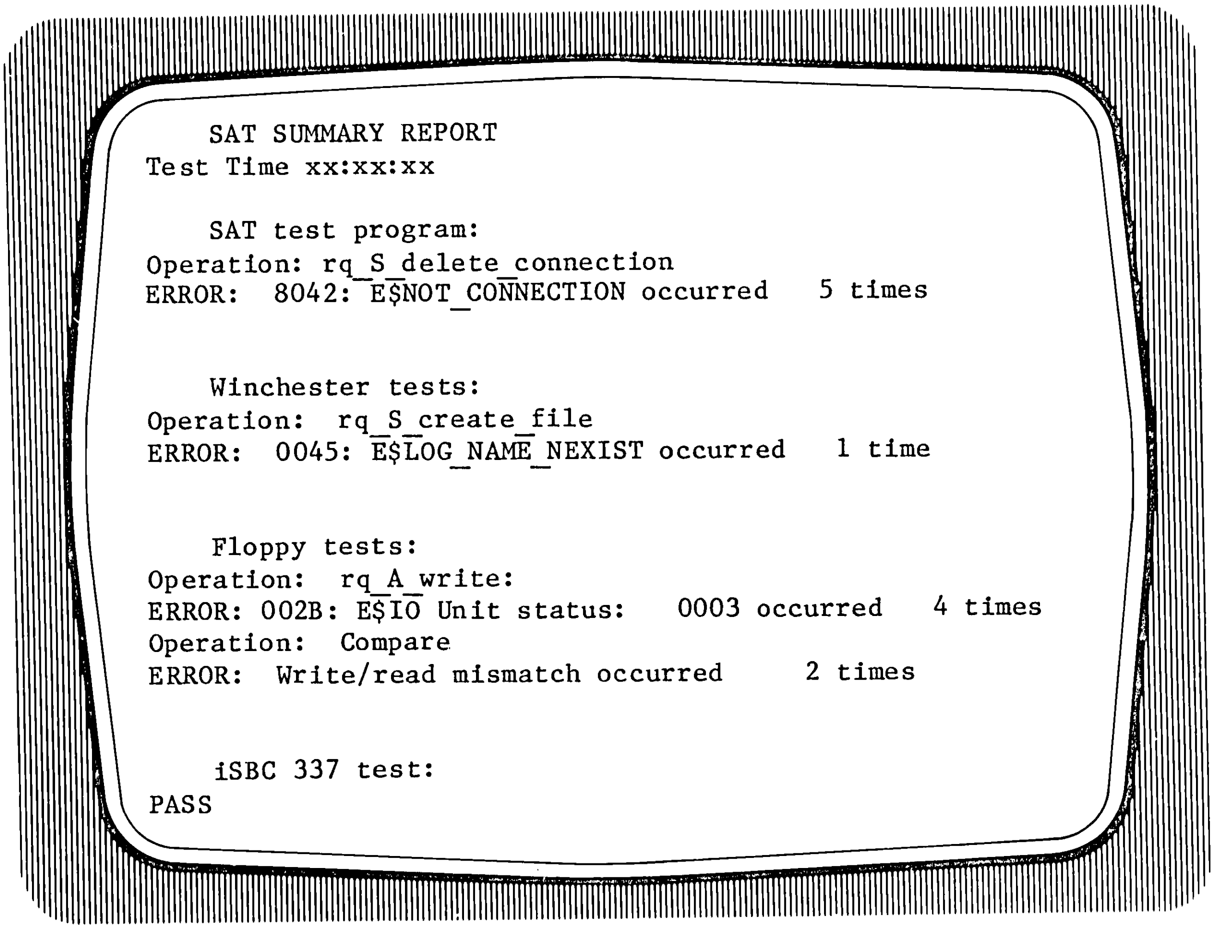


```
          SAT SUMMARY REPORT
     Test time xx:xx:xx

          SAT test program:
     PASS

          Winchester tests:
     PASS

          Floppy tests:
     PASS

          iSBC 337 test:
     PASS
```

Figure 3-2. SAT Summary Report (With No Errors)

In Figure 3-2, the SAT Summary Report indicates that all the tests passed. If you initialized the clock before invoking the SAT (by using the iRMX 86 DATE and TIME commands), the report also displays the amount of time that the SAT test program ran. If you did not initialize the clock, the elapsed time displays 00:00:00. Any errors encountered during the SAT main task are listed under the heading "SAT test program". These errors are related to the creating and deleting of tasks and/or connections to temporary files. Errors related to the BIOS and EIOS tests on the Winchester are listed under the heading Winchester test. Errors related to the BIOS and EIOS tests on the flexible disk drive are listed under the heading Floppy tests. Errors related to the Numeric Processor Extension test are listed under the heading iSBC 337 test.

The report contains a summary of all errors encountered up to the point at which the report is generated. Thus, errors that occurred during the first few minutes of the SAT test program are reported and continue to be reported once every minute until the SAT test program terminates. Figure 3-3 contains an example of some possible error messages reported by the SAT Summary Report.



```
    SAT SUMMARY REPORT
Test Time xx:xx:xx

    SAT test program:
Operation: rq_S_delete_connection
ERROR:  8042: E$NOT_CONNECTION occurred   5 times


    Winchester tests:
Operation:  rq_S_create_file
ERROR:  0045: E$LOG_NAME_NEXIST occurred   1 time


    Floppy tests:
Operation:  rq_A_write:
ERROR: 002B: E$IO Unit status:   0003 occurred   4 times
Operation:  Compare
ERROR:  Write/read mismatch occurred     2 times


    iSBC 337 test:
PASS
```

Figure 3-3. SAT Summary Report (With Errors)

As shown in Figure 3-3, error reporting is done in sets of two lines. The first line identifies the operation that failed. The second line describes the type of error. Following the type of error, the number of times this error occurred is also displayed. Under the title SAT test program, the failing operation indicates that the EIOS attempted to delete a connection five times (via the RQSDELETE$CONNECTION system call). This signifies that the SAT test's main task was attempting to delete a connection to a temporary file that did not exist. The reason it did not exist is indicated by the next error message which specifies that the temporary file was not created.

When an iRMX 86 system call is displayed as the failing operation, refer to the appropriate iRMX 86 manual for further information concerning the reported error.

Table 3-3 lists all of the system calls which can appear within the SAT SUMMARY REPORT. Refer to iRMX 86 NUCLEUS REFERENCE MANUAL for more information about Nucleus System calls, the iRMX 86 BASIC I/O SYSTEM REFERENCE MANUAL for information about BIOS system calls, and the iRMX 86 EXTENDED I/O SYSTEM REFERENCE MANUAL for information about EIOS System calls.

Table 3-3. System Call Index

Nucleus System Calls	BIOS System Calls	EIOS System Calls
rq_create_mailbox	rq_A_open	rq_S_attach_file
rq_create_task	rq_A_read	rq_S_create_file
rq_create_segment	rq_A_seek	rq_S_delete_connection
rq_delete_segment	rq_A_truncate	rq_S_open
rq_delete_task	rq_A_write	rq_S_read
rq_get_priority		rq_S_seek
rq_receive_message		rq_S_truncate_file
rq_send_message		rq-S_write
rq_sleep		

If an E$IO error occurs on an iRMX 86 BIOS operation, the unit status for the device is displayed after the error message but before the number of occurrences. The unit status gives the status of the device at the time the error occurred. Refer to the iRMX 86 BASIC I/O SYSTEM REFERENCE MANUAL for an explaination of the unit status.

If more diagnostic information is needed, invoke the System Diagnostic Test. Refer to Chapter 4.

CHAPTER 4. SYSTEM DIAGNOSTIC TEST

INTRODUCTION

The System Diagnostic Test (SDT) is a collection of test suites designed to diagnose system failures to a defective board or device. Each test suite exercises a specific portion of the System 86/300 Series Microcomputer System. Table 4-1 lists the test suites of the SDT and the sequence in which they should be executed.

Some of the test suites exercise optional hardware which may not be available on all systems. When you run the SDT, you should invoke only the test suites that exercise the hardware in your system.

Table 4-1. SDT Test Suites

TEST SUITE	FUNCTION
SDT8630	If the iSBC 86/30 board is the main processor board in your system, you can use this test to check the procesor's CPU, USART, ROM, RAM, PPI, Timer, and the associated gating circuitry on the iSBC 86/30 board. It also tests portions of the iSBC 215 Winchester Disk Controller board.
SDT8612	If the iSBC 86/12A board is the main processor board in your system, you can use this test to check the procesor's CPU, USART, ROM, RAM, PPI, Timer, and the associated gating circuitry on the iSBC 86/12A board. It also tests portions of the iSBC 215 Winchester Disk Controller board.
SDT056	Checks the RAM memory board's RAM cells, and RAM address and data lines and parity circuitry.
SDT215	Checks the functionality of the disk controller by verifying the co-processor, ROM, RAM, and other circuity on the iSBC 215 Winchester disk controller board and the circuitry on the iSBC 218 flexible disk controller. It also checks the functionality of the Winchester drive and the flexible disk drive.
SDT337	Checks the 8087 Numeric Processor Extension.

SDT INSTALLATION

The SDT diagnostics reside on the Diagnostic diskettes shipped with the System. If you are using the iRMX 86 Operating System, you should, upon receipt of the diagnostics, install them onto the Winchester. If you are using an operating system other than the iRMX 86 Operating System, you must invoke the diagnostics from flexible diskette. In such a case, you should ignore the following installation discussion.

Installing the diagnostics on the Winchester disk provides you with two sets of diagnostics: one on diskette and another on the Winchester disk. Normally, you invoke the versions of the diagnostics that reside on the Winchester disk. However, if you experience problems with the Winchester disk, you can attempt to invoke the diagnostics from the flexible diskettes.

To install the SDT diagnostics on the Winchester, proceed as follows:

1. After successfully bootstrap loading the iRMX 86 Operating System, insert the diskette marked "SYSTEM 86/300 SERIES DIAGNOSTIC DISKETTE #1" into the flexible disk drive.

2. Enter the following:

   ```
   SUBMIT :FD0:SDTINSTALL.CSD
   ```

   This SUBMIT file creates on the Winchester disk a directory called SDTDIR and copies the individual SDT test suites into this directory.

3. Insert the diskette marked "SYSTEM 86/300 SERIES DIAGNOSTIC DISKETTE #2" into the flexible disk drive and enter the following command:

   ```
   DIR :FD0:SDTDIR
   ```

   This lists the SDT test suites contained on the second diagnostic diskette.

4. Look at the display produced. If you need any of the diagnostic test suites listed, copy them individually to the SDTDIR directory on the Winchester disk. For example, to copy the SDT8612 test suite, enter the following command:

   ```
   COPY :FD0:SDTDIR/SDT8612 TO /SDTDIR/SDT8612
   ```

Once installed on the Winchester disk, the SDT diagnostics can be invoked from either the Winchester or from the flexible disk drive. The ability to invoke the SDT from either the Winchester disk or from the flexible disk is useful during troubleshooting.

SDT INVOCATION

You must first gain access to the iSBC 957B monitor before you can invoke any of the SDT test suites. To invoke an SDT test suite, proceed as follows:

1. To gain access to the iSBC 957B monitor from the Operating System, perform the following:
   a. On the System front panel, press the RESET pushbutton. (This resets the registers to a known condition.)
   b. When asterisks are displayed on the terminal (refer to Figure 2-1), enter an upper case U (press the SHIFT and u keys).
   c. In response to the SCT's PIC test prompt (refer to Figure 2-2), press the INTRPT pushbutton on the System front panel.
   d. To invoke the appropriate SDT test suite, enter one of the Bootstrap Load commands listed in Table 4-2.
2. To gain access to the iSBC 957B monitor from an SDT test suite in order to invoke the same or another SDT test suite, proceed as follows:
   a. At the terminal, enter EXIT.
   b. To invoke the appropriate SDT test suite, enter one of the Bootstrap Load commands listed in Table 4-2.
3. To bootstrap load the Operating System, proceed as follows:
   a. At the terminal, enter EXIT. This returns control to the monitor from the test suite.
   b. At the terminal, enter:

      ```
      b
      ```

      This bootstrap loads the Operating System.

Table 4-2 lists the commands used to invoke each of the SDT test suites.

Table 4-2. SDT Test Suite Invocation

TEST SUITE	BOARD CHECKED	ENTER THE FOLLOWING TO BOOTSTRAP LOAD FROM WINCHESTER	ENTER THE FOLLOWING TO BOOTSTRAP LOAD FROM FLEXIBLE DISK DRIVE
SDT8612	Processor	b /SDTDIR/SDT8612	b :WFO:/SDTDIR/SDT8612
SDT8630	Processor	b /SDTDIR/SDT8630	b :WFO:/SDTDIR/SDT8630
SDT056	RAM Memory	b /SDTDIR/SDT056	b :WFO:/SDTDIR/SDT056
SDT215	Disk Controller	b /SDTDIR/SDT215	b :WFO:/SDTDIR/SDT215
SDT337	Numeric Processor	b /SDTDIR/SDT337	b :WFO:/SDTDIR/SDT337

The recommended sequence of SDT test suites to invoke is:

1. The test suite for the processor (either SDT8630 or SDT8612)
2. The SDT056 test suite
3. The SDT215 test suite
4. The SDT337 test suite

Of course, when the SCT or SAT test results point to a particular problem area, you can select the appropriate SDT test suite to ascertain the failing unit.

Once you invoke a particular SDT test suite, you can run all tests, run a specific series of tests, loop on a particular test or tests, or run one test by using the available test management commands. These commands are described in the following sections.

A description of the test suites and all the tests within each test suite are described in their respective sections.

TEST MANAGEMENT COMMANDS

A standard set of test management commands is available with all the SDT test suites. These commands provide the ability to select and run the SDT tests. You can enter the test management commands regardless of the test suite you load into your system.

The test management commands that are available include:

CLEAR	Resets the execution and error count values
DEBUG	Enables or disables the printing of debug messages (or lists the status of the command)
DESCRIBE	Prints a description of the tests
ERRONLY	Enables or disables the printing of status messages (or lists the status of the command)
EXIT	Returns control to the iSBC 957B monitor
FINISH	Calls a user-supplied finish routine
IGNORE	Causes the TEST and SUMMARY commands to ignore a test or tests
LIST	Copies all console I/O to a file
QUERY	Displays the status of DEBUG and ERRONLY
RECOGNIZE	Reverses the effect of an IGNORE
REPEAT/ ENDREPEAT	Begins and ends a loop of commands
RESET	Resets the hardware and/or software
SUMMARY	Prints a result log of the tests that have run
TEST	Selects and runs the SDT tests
V	Displays and sets the global variables V(n), where n ranges from 0 through 0FH

The following sections describe these commands in detail. The first section contains information that applies to all the test management commands. The remaining sections describe the individual commands in detail.

COMMAND SYNTAX

Later sections of this chapter discuss the syntax of the individual test management commands. However, this section contains additional information about command syntax that applies to all commands.

Abbreviations

Each of the test management commands has a command name. Keywords are also used as parameters to many of the commands. When you enter commands, you can abbreviate the command names and keywords to their first three characters. For example, you can abbreviate the following command:

```
TEST 1 REPEAT UNTIL ERROR
```

to:

```
TES 1 REP UNT ERR
```

Continuation Characters and Comments

You can continue a command on more than one line by using the ampersand (&) as a continuation character. This character informs the SDT that the command continues on the next line. The SDT treats all characters entered after the ampersand but before the end-of-line character (carriage return) as comments.

Also, you can specify a comment by entering the characters:

```
/*
```

All characters which follow these characters in the line are considered comments. For example, you can enter the following command:

```
TEST 1                   &   Run the first test
**REPEAT UNTIL ERROR     /*  until an error occurs
```

This command illustrates the two kinds of comments. The SDT displays the double asterisk at the beginning of the second line to indicate that the line is a continuation line.

Command Delimiters

You can specify multiple commands on a single line. To do this, you must delimit the commands with a semicolon (;). For example, you can enter the following commands on a single line:

```
IGNORE 1; TEST REPEAT 5; SUMMARY
```

The SDT runs these commands as if you entered them on separate lines. Refer to later sections for descriptions of the individual commands.

Input Radices

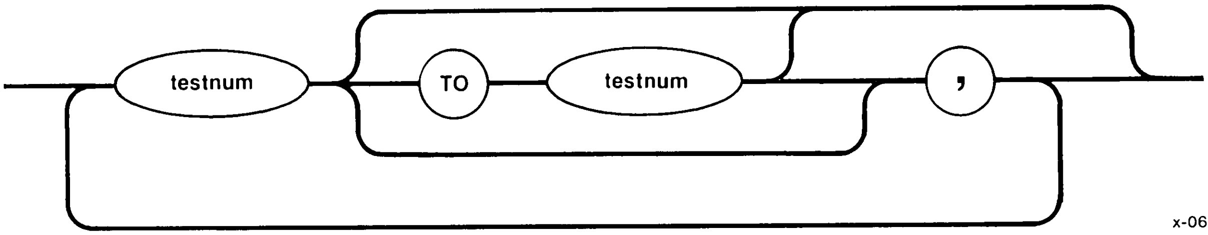
The SDT always produces numerical output in hexadecimal format. However, when you provide input to the SDT, you can specify the radix of numerical quantities by including a radix character immediately after the number. The valid radix characters include:

radix	character	example
hexadecimal	h or H	16h, 7Ch
decimal	t or T	23t, 100T
octal	q or Q	27q, 33Q
binary	y or Y	101y, 11100Y

If you omit the radix character in numerical input, the SDT assumes the number is hexadecimal.

Test Range

Throughout this chapter, the command descriptions use the term "test-range" as a parameter name. When a command description lists this term as a parameter, you must enter a range of test numbers on which the command is to operate. The format for entering a range of test numbers is as follows:



x-068

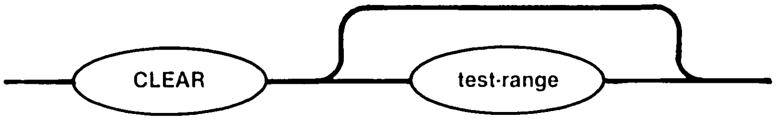
If you separate test numbers with commas, you select the individual tests (for example, 1,3 selects tests 1 and 3). If you separate two test numbers with the word TO, you select all tests between the first number and the second number (for example, 1 TO 3 selects tests 1, 2, and 3). However, if you separate test numbers with TO, the first number must be smaller than the second number. You can use a combination of commas and the word TO when entering the test range (for example, 0 TO 2,4,6 TO 8 selects tests 0, 1, 2, 4, 6, 7, and 8).

Interrupting the Diagnostic Tests

In certain cases, you may want to interrupt a test that is currently running and regain control at the system console. To do this, you must enter the Control-C character. The Control-C character causes the SDT to abort the current test or command and return control to you at the console. However, the SDT can respond to a Control-C only when it is performing console I/O. Therefore, if you interrupt a diagnostic test that performs console I/O infrequently, there may be a delay in obtaining control from the test.

CLEAR COMMAND

The CLEAR command resets the execution count and error count for the specified tests. The format of this command is as follows:



x-069

where:

test-range	Range of test numbers upon which the command operates. If you omit the test range, CLEAR resets the execution count and error count for all tests in the SDT test suite.

DESCRIPTION

Each time you enter a SUMMARY command for a particular test, the SDT displays information about that test, including the number of times the test has been run and the number of errors encountered. The CLEAR command resets this information. For each test in the test range (excepting those for which you have specified the IGNORE command), CLEAR sets the execution count and error count to zero.

ERROR MESSAGES

```
ERROR: test out of range
```

One or more of the test numbers that you specified was larger than the largest test number in the SDT test suite.

```
ERROR: in "a TO b", b is less than a
```

When specifying test numbers with the terms "a TO b", the number "a" must be less than the number "b".

EXAMPLES

```
*CLEAR
*
```

This command clears the execution and error count values for all tests in the SDT test suite.

```
*CLE 7
*
```

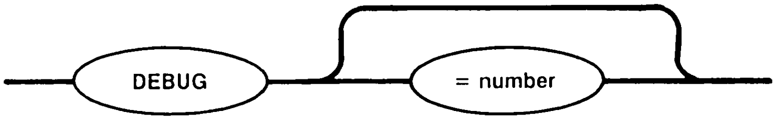
This command clears the execution and error count values for test 7.

```
*CLE 1 TO 4,8
*
```

This command clears the execution and error count values for tests 1, 2, 3, 4, and 8.

DEBUG COMMAND

The DEBUG command determines whether the SDT displays debug messages at the console during test execution. The format of this command is as follows:



x-070

where:

number	Value which determines the debug status. An even value (least-significant bit set to 0) sets the debug status to FALSE. An odd value (least-significant bit set to 1) sets the debug status to TRUE. If you omit the number parameter, the SDT displays the current value of DEBUG. Initially, the debug status is set to the default value of FALSE.

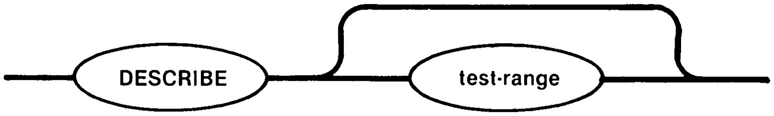
DESCRIPTION

During execution, a diagnostic test can return three kinds of messages, debug messages, status messages, and error messages. An error message indicates a failure of a diagnostic test. A status message returns information on the general status of the diagnostic tests. A debug message lists detailed information about the failing test. A debug message is usually important only to the writer of a diagnostic test or to a person who is debugging a particular piece of hardware.

The DEBUG command informs the SDT whether to display the debug messages. If you set DEBUG to TRUE, the SDT displays, for some failing tests, detailed messages produced by the diagnostic tests that describe the failure in terms of specific test operations. Also, if you set DEBUG to TRUE and ERRONLY to FALSE (refer to the description of the ERRONLY command for more information), the SDT displays the name and number of each test before running the test; it also displays the usual information after the test runs. If you set DEBUG to FALSE, the SDT omits the information that precedes the test and the detailed debug messages.

DESCRIBE COMMAND

The DESCRIBE command displays the names and numbers of the specified tests. The test names indicate the hardware being tested. The format of this command is as follows:



x-071

where:

test-range	Range of test numbers for which DESCRIBE displays information. If you omit the test range, DESCRIBE displays information about all tests in the SDT test suite.

ERROR MESSAGES

ERROR: test out of range

One or more of the test numbers that you specified was larger than the largest test number in the SDT test suite.

ERROR: in "a TO b", b is less than a

When specifying test numbers with the terms "a TO b", the number "a" must be less than the number "b".

EXAMPLES

```
*DESCRIBE
0000H               FIXED PATTERNS
0001H               ADDRESS MARCH
0002H               SLIDING ONES
0003H               EXECUTE FROM RAM
*
```

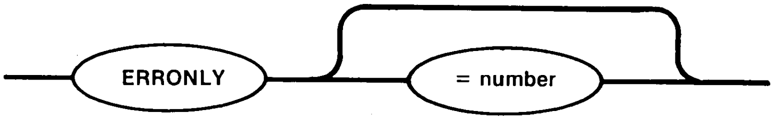
This command displays the test names and numbers for all tests in the SDT test suite (this example assumes you are running the SDT056 test suite).

```
*DES 1 TO 5,7
0001H               TRANSFER STATUS
0002H               BUFFER I/O TEST
0003H               ROM CHECKSUM TEST
0004H               RAM WINDOW TEST
0005H               RAM ADDRESS TEST
0007H               SEEK/VERIFY TEST
```

This command lists the test names and numbers for tests 1, 2, 3, 4, 5, and 7 (this example assumes you are running the SDT215 test suite).

ERRONLY COMMAND

The ERRONLY command determines whether the SDT displays status messages at the console during test execution. The format of this command is as follows:



x-072

where:

number	Value which determines the ERRONLY status. An even value (least-significant bit set to 0) sets the ERRONLY status to FALSE. An odd value (least-significant bit set to 1) sets the ERRONLY status to TRUE. If you omit the number parameter, the SDT displays the current value of ERRONLY. Initially, the erronly status is set to the default value of FALSE.

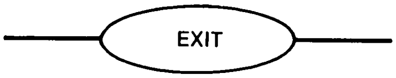
DESCRIPTION

During execution, a diagnostic test can return three kinds of messages, debug messages, status messages, and error messages. An error message indicates a failure of a diagnostic test. A status message returns information on the general status of the diagnostic tests. A debug message lists detailed information about the failing test.

The ERRONLY command informs the SDT whether to display the status messages. If you set ERRONLY to TRUE, the SDT displays test results only for tests that fail. It does not display information for tests that pass, nor does it display the names and numbers of the tests before running them, even if DEBUG is set to TRUE (refer to the description of the DEBUG command for more information). If you set ERRONLY to FALSE, the SDT displays test results for both passing and failing tests.

EXIT COMMAND

The EXIT command exits the SDT and returns control to the iSBC 957B monitor. From the monitor you can bootstrap load other modules, such as other SDT test suites or the iRMX 86 Operating System. The format of this command is as follows:



x-073

FINISH COMMAND

The FINISH command calls a customer-written procedure which returns the hardware to a consistent state. You should enter this command after aborting (with the Control-C character) any customer-written diagnostic test which leaves the hardware in an inconsistent state. You do not need to enter the FINISH command after aborting Intel-supplied diagnostic tests. The format of this command is as follows:

x-074

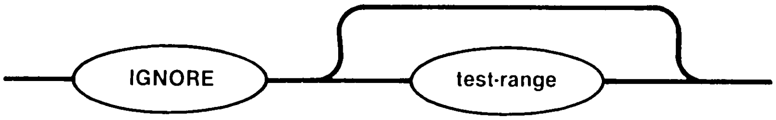
DESCRIPTION

If you write your own diagnostic tests, these tests may leave the hardware in an inconsistent state if aborted (using Control-C). The FINISH command provides a mechanism to call a customer-written procedure to return the hardware to a consistent state.

The contents of the customer-written procedure depend on the contents of the customer-written diagnostic tests. However, in order to be called with the FINISH command, the customer-written finish procedure must be a PUBLIC procedure named FINISH and have no parameters. You must link this procedure to the SDT module. Refer to Appendix A for more information about adding tests to the SDT.

IGNORE COMMAND

The IGNORE command declares tests which do not run when you invoke the TEST command. This IGNORE status for a test remains in effect until you issue a RECOGNIZE command for the test. The format of this command is as follows:



x-075

where:

test-range	Range of test numbers which are ignored during a TEST command. If you omit the test range, all tests in the SDT test suite are ignored. If you specify the number of a test that is already ignored, the IGNORE command has no effect for that test.

ERROR MESSAGES

```
ERROR: test out of range
```

One or more of the test numbers you specified was larger than the largest test number in the SDT test suite.

```
ERROR: in "a TO b", b is less than a
```

When specifying test numbers with the terms "a TO b", the number "a" must be less than the number "b".

EXAMPLES

```
*IGNORE
*
```

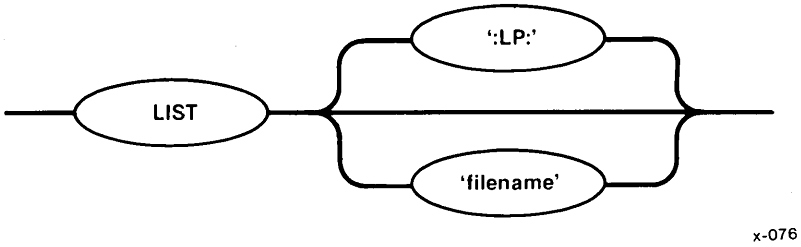
This command ignores all tests in the SDT test suite. Future TEST commands have no effect until you reverse the IGNORE status with RECOGNIZE command.

```
*IGN 3,6 TO 9
*
```

This command ignores tests 3, 6, 7, 8, and 9.

LIST COMMAND

The LIST command causes the SDT to copy all console I/O to the specified file. The format of this command is as follows:



where:

':LP:'	Line printer. You must enclose the :LP: in single quote characters. If you specify this parameter, LIST copies console I/O to the line printer connected to your System 86/300 Series Microcomputer System. However, if there is no line printer connected to your system, specifying ':LP:' causes the system to malfunction, requiring you to reload the SDT.
'filename'	ISIS-II filename. You must enclose the filename in single quote characters. If your system is connected to an Intel Microprocessor Development System via a parallel load cable, you can specify an ISIS-II filename to receive a copy of all console I/O. However, if your system is not connected to an Intel Microprocessor Development System, you cannot copy console I/O to a file other than :LP:. For instructions to connect your system to an Intel Microcomputer Development System, refer to the USER'S GUIDE FOR THE ISBC 957B iAPX 86, 88 INTERFACE AND EXECUTION PACKAGE and to your system's INSTALLATION AND MAINTENANCE MANUAL.

If you specify the LIST command without parameters, LIST stops copying console I/O and closes the current list file (if any).

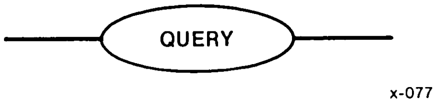
ERROR MESSAGES

Bad EMDS Connection

This message indicates a bad communication link between the System 86/300 Series Microcomputer System and the Intel Microprocessor Development System. To recover from this error, you must reload the SDT software.

QUERY COMMAND

The QUERY command displays the current values of the DEBUG and ERRONLY global variables. The format of this command is as follows:

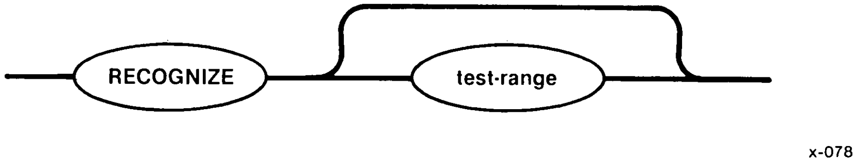


EXAMPLES

```
*QUERY
DEBUG=0000
ERRONLY=0000
*
```

RECOGNIZE COMMAND

The RECOGNIZE command reverses the effect of all or part of a previously-issued IGNORE command, allowing the specified tests to be run when the TEST command is invoked. The format of this command is as follows:



where:

test-range	Range of test numbers upon which the command operates. If you omit the test range, RECOGNIZE operates on all tests in the SDT test suite. If you specify the number of a test that is already recognized, the RECOGNIZE command has no effect for that test.

ERROR MESSAGES

ERROR: test out of range

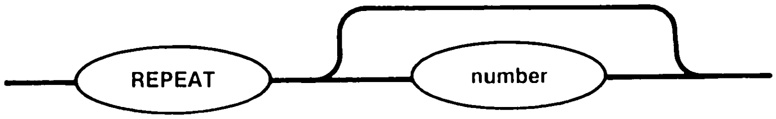
One or more of the test numbers you specified was larger than the largest test number in the SDT test suite.

ERROR: in "a TO b", b is less than a

When specifying test numbers with the terms "a TO b", the number "a" must be less than the number "b".

REPEAT/ENDREPEAT COMMANDS

The REPEAT and ENDREPEAT commands allow you to repeat a group of commands any number of times. The REPEAT command denotes the start of the group; the ENDREPEAT command denotes the end of the group. The format of the REPEAT command is as follows:



x-079

where:

number	The number of times the group of commands is to be repeated. If you omit the number parameter, the delimited commands are repeated indefinitely.

The format of the ENDREPEAT command is as follows:

x-080

DESCRIPTION

The REPEAT and ENDREPEAT commands provide a mechanism for creating command loops. When you enter the REPEAT command, the SDT responds by issuing a period (.) followed by an asterisk (*) as a prompt. This prompt reminds you that any succeeding commands you enter are part of a REPEAT loop. The SDT does not execute any of the succeeding commands until you enter the ENDREPEAT command to end the REPEAT loop.

You can nest REPEAT loops by entering a REPEAT command in response to the period-asterisk prompt. If you do this, the SDT responds by issuing two periods (..) followed by an asterisk (*) as a prompt. Any succeeding commands that you enter are part of the nested loop. You can end the nested loop by entering an ENDREPEAT command in response to the period-period-asterisk prompt. However, the SDT does not execute any commands until you end the outermost REPEAT loop with an ENDREPEAT command.

You can nest up to eight levels of REPEAT loops. At each level, the SDT responds with an additional period in its prompt. The SDT issues an error message if you attempt to nest more than eight levels of REPEAT loops.

ERROR MESSAGES

```
ERROR: too many nested IFs or REPEATs
```

You attempted to create more than eight levels of REPEAT loops.

EXAMPLES

```
*REPEAT
.*TEST 1
.*TEST 0
.*TEST 3
.*ENDREPEAT
```

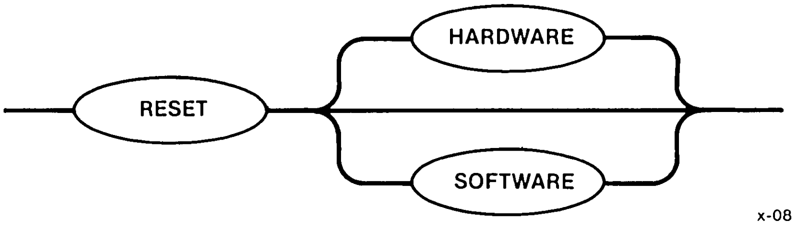
This example repeats diagnostic tests in nonsequential order. The SDT will repeat the tests until you enter a Control-C character to terminate the testing.

```
*REPEAT
.*TEST 0
.*REPEAT 5
..*TEST 9
..*ENDREPEAT
.*ENDREPEAT
```

This example illustrates a nested repeat loop. In this example, the SDT runs test 0 followed by five iterations of test 9. It continues this testing sequence until you enter a Control-C character.

RESET COMMAND

The RESET command resets the software and hardware to their initial states at program start. The format of this command is as follows:



x-081

where:

HARDWARE	Resets the hardware to its initial state. The SDT test suite may request additional information from you about the state of the hardware.
SOFTWARE	Resets the SDT software to its initial state. The SDT test suite may request additional information about test ranges and devices.

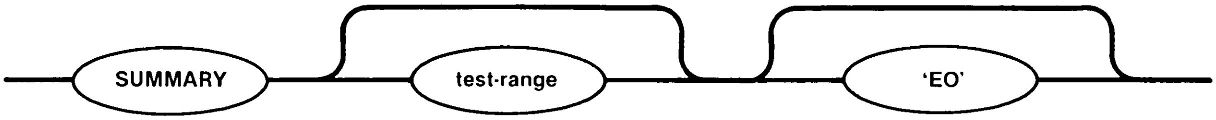
If you specify the RESET command without parameters, the SDT first resets the software and then resets the hardware.

DESCRIPTION

The RESET command allows you to reset the software and hardware portions of the SDT test suite to their initial states without forcing you to exit and re-invoke the test suite. However, the RESET command does not reset the values of the DEBUG and ERRONLY commands, nor does it specify that ignored tests are to be recognized in all cases. You should examine the individual SDT test suites to determine the state of ignored tests after reset.

SUMMARY COMMAND

The SUMMARY command displays a log of test results for the specified tests. The format of this command is as follows:



x-082

where:

test-range	The range of test numbers for which a test summary is required. If you omit this parameter, the SDT assumes the test range to be all the tests in the SDT test suite.
'EO'	Displays the "error only" information. If you specify this parameter, SUMMARY displays information only for failing tests in the test range. Otherwise, SUMMARY displays information for all tests in the test range.

DESCRIPTION

The SUMMARY command displays the name and number of each test in the test range followed by the number of trials and the number of failures (both in hexadecimal). It flags failing tests with the characters:

```
<===
```

You can reset the number of trials and failures by entering the CLEAR command.

The information concerning the number of trials and the number of failures does not appear for any tests for which you have specified the IGNORE command. Instead, the following message appears in place of the trials/failures information:

```
*** IGNORED ***
```

ERROR MESSAGES

```
ERROR: test out of range
```

One or more of the test numbers you specified was larger than the largest test number in the SDT test suite.

```
ERROR: in "a TO b", b is less than a
```

When specifying test numbers with the terms "a TO b", the number "a" must be less than the number "b".

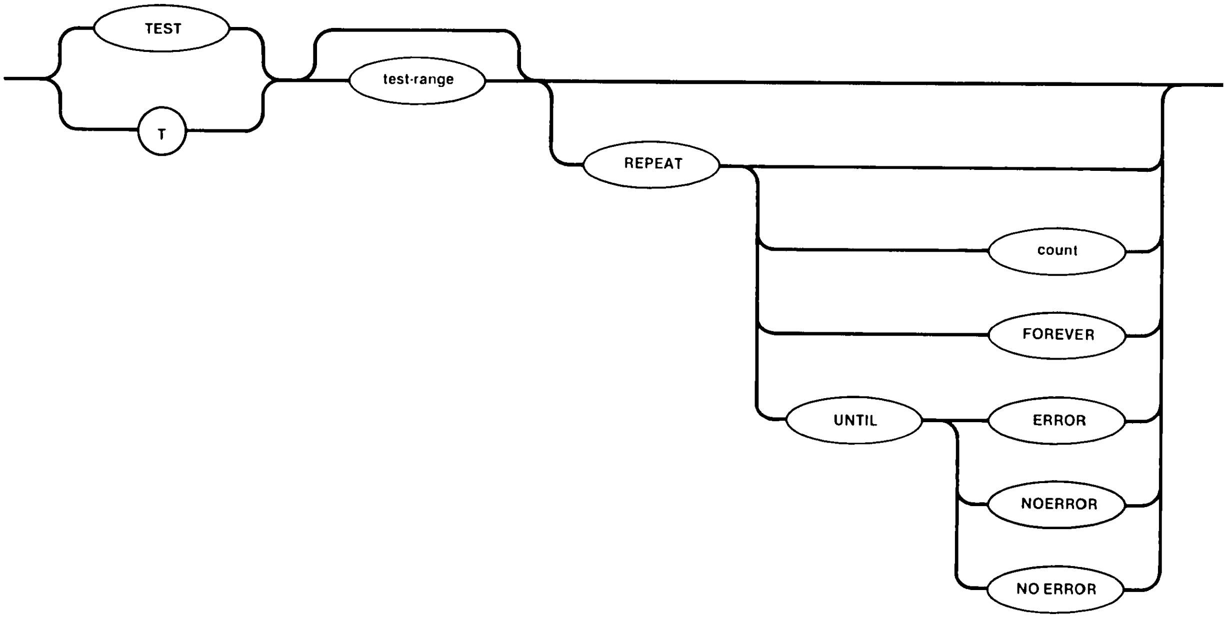
EXAMPLE

```
*SUMMARY
0000H FIXED PATTERNS                    0000 FAILED IN 0004 TRIALS
0001H ADDRESS MARCH                     0000 FAILED IN 0004 TRIALS
0002H SLIDING ONES                      0000 FAILED IN 0004 TRIALS
0003H EXECUTE FROM RAM                  *** IGNORED ***
*
```

This command displays the log of test results for all tests in the current test suite (this example assumes the SDT056 test suite). IGNORE status is set for the last test; it does not run and SUMMARY does not display information for it.

TEST COMMAND

The TEST command selects and runs diagnostic tests. The format of this command is as follows:



x-083

where:

test-range	Range of tests which are to run. The SDT runs all tests in the test range except for those which you have specified in an IGNORE command. If you omit this parameter, the SDT assumes the test range to be all tests in the SDT test suite.
REPEAT	Repeats the tests as specified in the succeeding parameters. If you omit the succeeding parameters, the SDT repeats the tests indefinitely. If you omit this parameter, the SDT runs the tests once.
count	Specifies the number of times to repeat the tests.
FOREVER	Repeats the tests indefinitely.
UNTIL ERROR	Repeats the tests until an error occurs.
UNTIL NOERROR or UNTIL NO ERROR	Repeats the tests until no errors occur.

DESCRIPTION

The TEST command runs the diagnostic tests in the current SDT test suite, as specified in the test-range parameter. It runs all tests except those that have IGNORE status set. Refer to the IGNORE command for more information.

The amount of information that the TEST command displays after running a test depends on the settings of the DEBUG and ERRONLY global variables (refer to the description of the DEBUG and ERRONLY commands for more information). Normally, the SDT displays the number, name, and result of each test (PASSED or FAILED) after executing the test. If a test is ignored, the name field for the test contains the message:

```
*** IGNORED ***
```

If you set the DEBUG variable to an odd value (TRUE), the SDT displays the name and number of each test before running the test; it also displays the usual information after the test runs, plus any detailed debug messages that the individual tests produce to describe the error conditions. If you set the DEBUG variable to an even value (FALSE), the SDT omits the information that precedes the test and the detailed debug messages.

If you set the ERRONLY variable to an odd value (TRUE), the SDT displays test results only for tests that fail. It does not display information for tests which pass, nor does it display the names and numbers of the tests before running them, even if DEBUG is set to TRUE. If you set the ERRONLY variable to an even value (FALSE), the SDT displays test results for both passing and failing tests.

NOTE

If you set the ERRONLY variable to TRUE and issue the TEST command with the REPEAT parameter (REPEAT, REPEAT FOREVER, or REPEAT UNTIL ERROR), you may find it impossible to stop the test execution with the Control-C character. If all the tests run without errors, the state of the ERRONLY variable causes the SDT to omit messages to the terminal. Since the SDT can respond to a Control-C only when a test is performing I/O, this means that any Control-C you enter will be ineffective. In such a case, you must press the INTRPT button to stop test execution. You must then reload the appropriate SDT test suite to run further SDT diagnostic tests.

ERROR MESSAGES

```
    ERROR: test out of range
```

One or more of the test numbers you specified was larger than the largest test number in the SDT test suite.

```
    ERROR: in "a TO b", b is less than a
```

When specifying test numbers with the terms "a TO b", the number "a" must be less than the number "b".

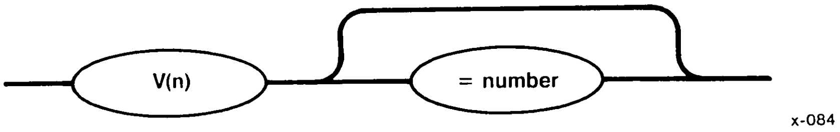
EXAMPLE

```
*TEST
0003H *** IGNORED ***
0000H FIXED PATTERNS                                   "PASSED"
0001H ADDRESS MARCH                                    "PASSED"
0002H SLIDING ONES                                     "PASSED"
*
```

This example runs each of the tests in the test suite once, except for test 3, which has IGNORE status set. This example assumes the SDT056 test suite is the current test suite.

V COMMAND

The V command displays or sets word values for global variables. These global variables affect the operation of all the tests in the SDT test suite. The format of the V command is as follows:



where:

n	Number of the variable, in the range 0 through 0FH. These variables can contain any word values. Not all Intel-supplied diagnostic tests make use of these global variables. However, you can make use of these variables if you write your own diagnostic tests.
number	Value to which you want to set the global variable. V variables can be set to any 16-bit (word) value. If you omit the number parameter, the SDT displays the current value of the global variable.

DESCRIPTION

V variables provide word values which you can set or display. For Intel-supplied test suites, the following variables are meaningful:

variable	description
V(5)	If you specify an even value for this variable, the SDT215 test suite displays controller timeout and busy messages. If you specify an odd value, SDT215 omits these messages.

You can write your own diagnostic tests that read or change the V variables. Refer to Appendix A for more information concerning user-written diagnostic tests.

ERROR MESSAGES

ERROR: "V" variable out of bounds

You specified a value greater than 0FH for a V variable index.

SDT8630 DIAGNOSTIC TEST

If your system contains an iSBC 86/30 board, you can use the SDT8630 test suite to check the functionality of the board. It can also be used to check the iSBC 304 RAM Memory Expansion Board, if you have added that board to your system. This diagnostic test suite can be invoked from either the Winchester drive (once the SDT has been installed) or from the flexible disk drive. Note that the iSBC 86/30 board and the device that you access to bootstrap load this SDT test suite must be at least marginally functional. That is, when this test is bootstrap loaded from the Winchester, the processor board, the Winchester drive, and the iSBC 215 Winchester disk controller board must be functional. If you bootstrap load this test from the flexible disk drive, the processor board, the flexible disk drive, portions of the iSBC 215 board, and the iSBC 218 flexible disk controller board must be functional. Table 4-3 lists each of the SDT8630 tests and their respective functions.

Table 4-3. SDT8630 Tests

TEST	FUNCTION
0	ROM Checksum
1	8255 Parallel Port Test
2	8259 Interrupt Test
3	8253 Timer Test
4	Fixed Patterns - iSBC 86/30
5	Fixed Patterns - iSBC 304
6	Address March Test
7	Sliding Ones Test - iSBC 86/30
8	Sliding Ones Test - iSBC 304
9	Dual Port RAM Contention Test

When invoked, the SDT8630 diagnostic test suite displays the following information:

```
SYSTEM DIAGNOSTIC TEST - 8630, Vx.x
```

where x.x is the version number.

Next, the SDT8630 diagnostic test prompts you for a command. If you wish to test hardware that is not in the system default condition, your first command should be a RESET command. This causes the SDT8630 to display a series of prompts, allowing you to specify the configuration of your hardware. The next section lists the prompts, along with the default values.

RESET HARDWARE PROMPTS

If you enter the RESET command, the SDT8630 asks questions about the hardware configuration and prompts you for responses. For these questions, the SDT8630 indicates the default response with brackets, such as:

```
(Y OR [N])
```

In this case, N (or no) is the default response. If you enter a carriage return alone, the SDT8630 assumes the default response.

The prompts are:

```
Set cpu clock to (8 or [5])*
```

In response, enter the clock frequency, in MHz.

```
is iSBC 304 installed? (y or [n])*
```

If you answer yes, the SDT8630 assumes that an iSBC 304 RAM Expansion Board is attached to your processor board. In this case, it assumes on-board memory ranges from 0 through 3FFFFh. If you answer no, the SDT8630 assumes that no iSBC 304 board is attached. In this case, it assumes that on-board memory ranges from 0 through 1FFFFh. It also sets tests 5 and 7 to be ignored.

The SDT8630 next examines memory locations to determne whether gaps exist in the range of on-board memory. If it discovers an address in the range of on-board memory that does not represent an actual memory location, it displays the following message:

```
non-existent memory begins between <xxxx>:0000 and <yyyy>:0000
```

where <xxxx> and <yyyy> are segment values that indicate a range of 16K bytes. This message indicates that a gap in memory exists somewhere in the specified 16K bytes.

If the SDT8630 displays this message, it automatically sets all memory tests to be ignored.

SDT8630 TEST DESCRIPTIONS

The SDT8630 consists of a number of tests which check the functionality of the iSBC 86/30 board. Two of the tests check the functionality of an iSBC 304 RAM Memory Expansion Board. These two tests are originally set to be ignored when you invoke the test suite. If you add an iSBC 304 board to your processor board, you can use the RECOGNIZE or RESET commands to enable these tests.

WARNING

> Only qualified technical personnel should perform the corrective actions listed in this section. Attempts by unqualified personnel to service the hardware can result in personal injury and damage to the System. In addition, always disconnect the power cord prior to installing or removing a board or peripheral device. Failure to observe this precaution can result in personal injury and circuit damage.

Test 0. ROM Checksum

The ROM Checksum test checks the address and data lines of the ROM on the processor board and calculates the checksum of the contents of the ROMs.

This test can return messages of the following format:

- Rom Checksum Error, <type> Byte: Expected = <ee> Received = <rr>

  where <type> is Even or Odd and the values <ee> and <rr> are the expected and received values.

If this test fails, replace the processor board.

Test 1. 8255 Parallel Port Test

The 8255 Parallel Port test checks the processor's ability to communicate with the PPI (Programmable Peripheral Interface). This routine writes data into the PPI, reads it from the PPI, and compares both values.

This test can return messages of the following format:

- Error at port <p>: Expected = <ee> Received = <rr>

  where <p> is the port number of the PPI (either A, B, or C) and the values <ee> and <rr> are the expected and received values.

If this test fails, replace the processor board.

Test 2. 8259 Interrupt Test

The 8259 Interrupt test checks the processor's ability to communicate with the Programmable Interrupt Controller (PIC). It initializes the PIC and causes three interrupts to occur. These three interrupts are:

USART's RxRDY signal
USART's TxRDY signal
PIT's Interrupt 2

The test prompts with a ">" character for the user to enter a character from the keyboard to force the USART interrupt.

This test can return the following messages:

- No Interrupt Detected: Expected = <eint>

  The test expected an interrupt to occur at level <eint>, but no interrupt occurred.

- Wrong Interrupt Detected: Expected = <eint> Received <rint>

  The test expected an interrupt to occur at level <eint>; however, the interrupt occurred at level <rint> instead.

If this test fails, replace the processor board.

Test 3. 8253 Timer Test - Read on the Fly

The 8253 Timer test checks the ability of the 8253 timer and its input clock network to count down at a programmed rate from a predetermined value. The 8253 timer is initialized and set to a predetermined value. The CPU is placed in a timing loop. When the CPU finishes, the 8253 timer contents are read on the fly twice. The value read must be within a given range of the expected value or an error exists.

This test can return messages of the following format:

- 1st READ-ON-THE-FLY, EXPECTED <expec1>, RECEIVED <xxxx>
  2nd READ-ON-THE-FLY, EXPECTED <expec2>, RECEIVED <yyyy>

  The value read is outside the expected limits. When the clock is set for 5 Mhz, the <expec1> value is 0AD15H and the <expec2> value is 5A4AH. The first value read, <xxxx>, must lie between 0AC15h and 0AE15h. The second value, <yyyy>, must lie between 0584Ah and 05C4Ah. When the clock is set for 8 Mhz, the <expec1> value is 0C465H and the <expec2> value is 8ADAH. In this case, the first value read, <xxxx>, must lie between 0C348H and 0C550H. The second value, <yyyy>, must lie between 8880H and 8C11H.

If this test fails, replace the processor board.

Test 4. Fixed Pattern - iSBC 86/30 Board

The Fixed Pattern test checks the RAM data lines of the processor board. It writes a pattern to each RAM cell using one of the two patterns (5555H and AAAAH) per pass. It then checks the contents of each cell against the value written.

This test can return error messages of the following format:

- error at <ssss:nnnn>
  expected<xxxx> received<yyyy> reread<rrrr> xor<zzzzzzzzzzzzzzzz>

  The <ssss:nnnn> is the segment base and offset of the failing RAM location, <xxxx> and <yyyy> are the expected and received values, <rrrr> is the value obtained when the location was read again, and <zzzzzzzzzzzzzzzz> is the exclusive OR of the expected and received values (in binary).

If this test fails, replace the processor board.

Test 5. Fixed Patterns - iSBC 304 Board

The Fixed Patterns iSBC 304 test is originally set to be ignored. If you add an iSBC 304 board to your processor board, you can use this test to check RAM data lines of the iSBC 304 board. It writes a pattern to each RAM cell using one of two patterns (5555H or AAAAH) per pass. It then checks each cell and compares the value read to the value written.

This test can return the same error messages that test 4 returns.

If this test fails, replace the iSBC 304 board or the processor board.

Test 6. Address March

The Address March test checks the memory on the processor board and any RAM expansion board that you have installed on the processor board. It writes a background pattern to each word of memory. Then it "marches" the complement of that pattern through memory, one word at a time. If the test encounters a cell that contains something other than the background pattern, it logs a error.

If you enter the RESET command and specify that your system contains an iSBC 304 board, this test checks all 256K bytes of memory on the processor board and the RAM expansion board. Therefore, if your system does not include a RAM expansion board, any errors returned for memory in the range of 2000:0 through 3000:FFFF indicate problems with the iSBC 056A memory board, not the processor board. This allows you to check for failures that cross board boundaries.

This test can return the same error messages that test 4 returns.

If this test fails, determine the location of the problem. If the failing memory resides on the processor board or the RAM expansion board, replace the failing board. If the failing memory resides on the iSBC 056A board, run the SDT056 test suite.

Test 7. Sliding Ones - iSBC 86/30 Board

The Sliding Ones - iSBC 86/30 test checks RAM memory on the processor board. It first writes a pattern to all of the RAM cells. Then it reads the first cell and compares the pattern read with the pattern written. Then it rewrites the cell with the next pattern, reads the pattern, and compares the pattern read with the pattern written. This sequence is repeated for each RAM cell. The sequence is repeated for each of the 16 different patterns. The following values are used for the different patterns:

0000
0101H
0303H
0707H
0F0FH
1F1FH
3F3FH
7F7FH
FFFFH
FEFEH
FCFCH
F8F8H
F0F0H
E0E0H
C0C0H
8080H

This test can return the same error messages that test 4 returns.

If this test fails, replace the processor board.

Test 8. Sliding Ones - iSBC 304 Board

The Sliding Ones - iSBC 304 test is originally set to be ignored. If you add an iSBC 304 board to your processor board, you can run this test to check the memory on the RAM Expansion board. This test performs the same operations on the iSBC 304 board that test 7 performs on the processor board.

This test can return the same error messages that test 4 returns.

If this test fails, replace the RAM Expansion board or the processor board.

Test 9. Dual Port RAM Contention Test

The Dual Port RAM Contention test forces dual port RAM contention on the processor board. It causes the iSBC 215 Winchester disk controller board to perform a DMA transfer to RAM, while the processor moves strings of data. There are no disk accesses during this test. The following steps are performed.

1. Initialize the iSBC 215 Winchester Disk Controller.
2. Write a test pattern in memory and transfer it to RAM memory on board the iSBC 215 board using the BUFFER I/O command.
3. Write zeros to memory and cause the iSBC 215 board to return the test pattern using the BUFFER I/O command.
4. Write zeros to RAM memory, transfer the pattern from RAM memory on the iSBC 215 board to buffers on the processor board while at the same time causing the processor board to read and write its own patterns (AAAAH and 5555H) in adjacent buffers all in dual port memory. This test loops until the iSBC 215 board signifies that it is finished by issuing an interrupt. The processor board finishes its move operation and verifies that the data transferred from the iSBC 215 board is correct.

This test can return the following error messages:

- reset error

  The test attempted to reset the iSBC 215 controller but did not receive acknowledgement from the controller.

- 86/30 to 215 failure

  The test attempted to transfer the test pattern from processor board memory to controller memory. However, the controller returned a status error or was unable to complete the transfer within the timeout period.

- 215 to 86/30 failure

  The test attempted to transfer the test pattern from controller memory to processor board memory. However, the controller returned a status error or was unable to complete the transfer within the timeout period.

- transfer buffer miscompare at <nnnn:mmmm>
  expected <ee> received <rr>

  The test compared the value in the processor board's dual port RAM buffer (the value that was sent to the controller) with the value in the iSBC 215 dual port RAM buffer (the value that the controller sent back) and found a mismatch. The <nnnn:mmmm> indicates the address of the mismatch. The <ee> and <rr> values indicate the expected and received values.

- contending access failure

  The test attempted to transfer the pattern while the processor board and iSBC 215 board were contending for dual port memory, as described in step 4 previously. Either the test received an error from the controller, as described in the first two messages, or a data mismatch occurred, as described in the third message.

If the test returns one of these messages, invoke the SDT215 test. If the SDT215 test passes, suspect the bus arbitration logic or the dual port RAM control logic.

SDT8612 DIAGNOSTIC TEST

If your system contains an iSBC 86/12A board, you can use the SDT8612 test suite to check the functionality of the iSBC 86/12A board and the iSBC 300A RAM Memory Expansion Board. This diagnostic test suite may be invoked from either the Winchester drive (once the SDT has been installed) or from the flexible disk drive. Note that the iSBC 86/12A board and the device that you access to bootstrap load this SDT test suite must be at least marginally functional. That is, when this test is bootstrap loaded from the Winchester, the processor board (iSBC 86/12A Single Board Computer and the iSBC 300A RAM Expansion Board), the Winchester drive, and the iSBC 215 Winchester disk controller board must be functional. If you bootstrap load this test from the flexible disk drive, the processor board, the flexible disk drive, portions of the iSBC 215 board, and the iSBC 218 flexible disk controller board must be functional. Table 4-4 lists each of the SDT8612 tests and their respective functions.

Table 4-4. SDT8612 Tests

TEST	FUNCTION
0	ROM Checksum
1	8255 Parallel Port Test
2	8259 Interrupt Test
3	8253 Timer Test
4	Fixed Patterns - iSBC 86/12A
5	Fixed Patterns - iSBC 300A
6	Address Patterns Test
7	Sliding Ones Test - iSBC 86/12A
8	Sliding Ones Test - iSBC 300A
9	Dual Port RAM Contention Test

When invoked, the SDT8612 diagnostic test suite displays the following information:

```
SYSTEM DIAGNOSTIC TEST - 8612, Vx.x
```

where x.x is the version number.

SDT8612 TEST DESCRIPTIONS

The SDT8612 consists of a number of tests which check the functionality of the iSBC 86/12A board. Two of the tests check the functionality of the iSBC 300A RAM Memory Expansion Board.

WARNING

> Only qualified technical personnel should perform the corrective actions listed in this section. Attempts by unqualified personnel to service the hardware can result in personal injury and damage to the system. In addition, always disconnect the power cord prior to installing or removing a board or peripheral device. Failure to observe this precaution can result in personal injury and circuit damage.

Test 0. ROM Checksum

The ROM Checksum test checks the address and data lines of the ROM on the processor board and calculates the checksum of the contents of the ROMs.

This test can return messages of the following format:

- Rom Checksum Error, <type> Byte: Expected = <ee> Received = <rr>

  where <type> is Even or Odd and the values <ee> and <rr> are the expected and received values.

If this test fails, replace the processor board.

Test 1. 8255 Parallel Port Test

The 8255 Parallel Port test checks the processor's ability to communicate with the PPI (Programmable Peripheral Interface). This routine writes data into the PPI, reads it from the PPI, and compares both values.

This test can return messages of the following format:

- Error at port <p>: Expected = <ee> Received = <rr>

  where <p> is the port number of the PPI (either A, B, or C) and the values <ee> and <rr> are the expected and received values.

If this test fails, replace the processor board.

Test 2. 8259 Interrupt Test

The 8259 Interrupt test checks the processor's ability to communicate with the Programmable Interrupt Controller (PIC). It initializes the PIC and causes three interrupts to occur. These three interrupts are:

USART's RxRDY signal
USART's TxRDY signal
PIT's Interrupt 2

The test prompts with a ">" character for the user to enter a character from the keyboard to force the USART interrupt.

This test can return the following messages:

- No Interrupt Detected: Expected = <eint>

  The test expected an interrupt to occur at level <eint>, but no interrupt occurred.

- Wrong Interrupt Detected: Expected = <eint> Received <rint>

  The test expected an interrupt to occur at level <eint>; however, the interrupt occurred at level <rint> instead.

If this test fails, replace the processor board.

Test 3. 8253 Timer Test - Read on the Fly

The 8253 Timer test checks the ability of the 8253 timer and its input clock network to count down at a programmed rate from a predetermined value. The 8253 timer is initialized and set to a predetermined value. The CPU is placed in a timing loop. When the CPU finishes, the 8253 timer contents are read on the fly twice. The value read must be within a given range of the expected value or an error exists.

This test can return messages of the following format:

- 1st READ-ON-THE-FLY, EXPECTED <expec1>, RECEIVED <xxxx>
  2nd READ-ON-THE-FLY, EXPECTED <expec2>, RECEIVED <yyyy>

  The value read is outside the expected limits. When the clock is set for 5 Mhz, the <expec1> value is 0AD15H and the <expec2> value is 5A4AH. The first value read, <xxxx>, must lie between 0AC15h and 0AE15h. The second value, <yyyy>, must lie between 0584Ah and 05C4Ah. When the clock is set for 8 Mhz, the <expec1> value is 0C465H and the <expec2> value is 8ADAH. In this case, the first value read, <xxxx>, must lie between 0C348H and 0C550H. The second value, <yyyy>, must lie between 8880H and 8C11H.

If this test fails, replace the processor board.

Test 4. Fixed Pattern - iSBC 86/12A Board

The Fixed Pattern test checks the RAM data lines of the processor board. It writes a pattern to each RAM cell using one of the two patterns (5555H and AAAAH) per pass. It then checks the contents of each cell against the value written.

This test can return error messages of the following format:

- error at <ssss:nnnn>
  expected<xxxx> received<yyyy> reread<rrrr> xor<zzzzzzzzzzzzzzzz>

  The <ssss:nnnn> is the segment base and offset of the failing RAM location, <xxxx> and <yyyy> are the expected and received values, <rrrr> is the value obtained when the location was read again, and <zzzzzzzzzzzzzzzz> is the exclusive OR of the expected and received values (in binary).

If this test fails, replace the processor board.

Test 5. Fixed Patterns - iSBC 300A Board

The Fixed Patterns - iSBC 300A checks RAM data lines of the iSBC 300A board. It writes a pattern to each RAM cell using one of two patterns (5555H or AAAAH) per pass. It then checks each cell and compares the value read to the value written.

This test can return the same error messages that test 4 returns.

If this test fails, replace the iSBC 300A board or the processor board.

Test 6. Address Patterns

The Address Patterns test checks the uniqueness of the RAM address lines on the processor board and the iSBC 300A board. The pattern written to each cell is formed by adding the four bytes of the cell address together. Then each RAM cell is read and the value compared against the value written.

This test can return the same error messages that test 4 returns.

If this test fails, replace the processor board.

Test 7. Sliding Ones - iSBC 86/12A Board

The Sliding Ones - iSBC 86/12A test checks RAM memory on the processor board. It first writes a pattern to all of the RAM cells. Then it reads the first cell and compares the pattern read with the pattern written. Then it rewrites the cell with the next pattern, reads the pattern, and compares the pattern read with the pattern written. This sequence is repeated for each RAM cell. The sequence is repeated for each of the 16 different patterns. The following values are used for the different patterns:

0000
0101H
0303H
0707H
0F0FH
1F1FH
3F3FH
7F7FH
FFFFH
FEFEH
FCFCH
F8F8H
F0F0H
E0E0H
C0C0H
8080H

This test can return the same error messages that test 4 returns.

If this test fails, replace the processor board.

Test 8. Sliding Ones - iSBC 300A Board

The Sliding Ones - iSBC 300A test checks the memory on the RAM Expansion board. This test performs the same operations on the iSBC 300A board that test 7 performs on the processor board.

This test can return the same error messages that test 4 returns

If this test fails, replace the RAM Expansion board or the processor board.

Test 9. Dual Port RAM Contention Test

The Dual Port RAM Contention test forces dual port RAM contention on the processor board. It causes the iSBC 215 Winchester disk controller board to perform a DMA transfer to RAM, while the processor moves strings of data. There are no disk accesses during this test. The following steps are performed.

1. Initialize the iSBC 215 Winchester Disk Controller.
2. Write a test pattern in memory and transfer it to RAM memory on board the iSBC 215 board using the BUFFER I/O command.
3. Write zeros to memory and cause the iSBC 215 board to return the test pattern using the BUFFER I/O command.
4. Write zeros to RAM memory, transfer the pattern from RAM memory on the iSBC 215 board to buffers on the processor board while at the same time causing the processor board to read and write its own patterns (AAAAH and 5555H) in adjacent buffers all in dual port memory. This test loops until the iSBC 215 board signifies that it is finished by issuing an interrupt. The processor board finishes its move operation and verifies that the data transferred from the iSBC 215 board is correct.

This test can return the following error messages:

- reset error

  The test attempted to reset the iSBC 215 controller but did not receive acknowledgement from the controller.

- 86/12A to 215 failure

  The test attempted to transfer the test pattern from processor board memory to controller memory. However, the controller returned a status error or was unable to complete the transfer within the timeout period.

- 215 to 86/12A failure

  The test attempted to transfer the test pattern from controller memory to processor board memory. However, the controller returned a status error or was unable to complete the transfer within the timeout period.

- transfer buffer miscompare at <nnnn:mmmm>
  expected <ee> received <rr>

  The test compared the value in the processor board's dual port RAM buffer (the value that was sent to the controller) with the value in the iSBC 215 dual port RAM buffer (the value that the controller sent back) and found a mismatch. The <nnnn:mmmm> indicates the address of the mismatch. The <ee> and <rr> values indicate the expected and received values.

- contending access failure

  The test attempted to transfer the pattern while the processor board and iSBC 215 board were contending for dual port memory, as described in step 4 previously. Either the test received an error from the controller, as described in the first two messages, or a data mismatch occurred, as described in the third message.

If the test returns one of these messages, invoke the SDT215 test. If the SDT215 test passes, suspect the bus arbitration logic or the dual port RAM control logic.

SDT056 DIAGNOSTIC TEST

The SDT056 Diagnostic test checks the functionality of a Multibus-compatible memory board. It can check any such board; however, it expects the board to contain a parity register similar to the one contained on the iSBC 056A board. If the memory board you test does not contain this parity register:

- The test cannot toggle the parity LED as described in this section
- Tests 1 and 2 always report parity errors

This section assumes you are running the test on an iSBC 056A memory board.

This test expects the processor board to be functional. It may be called from either the Winchester disk or from the flexible disk drive. Table 4-5 lists the tests of the SDT056.

Table 4-5. SDT056 Tests

TEST NUMBER	FUNCTION
0	Fixed Patterns
1	Address March
2	Sliding Ones Pattern
3	Execute from RAM

When invoked, the SDT056 diagnostic test suite displays the following information:

```
SYSTEM DIAGNOSTIC TEST - 056, Vx.x
```

where x.x is the version number.

Next, the SDT056 diagnostic test prompts you for a command. If you wish to test hardware that is not in the system default condition, or if you wish to change the test range, your first command should be a RESET command. This causes the SDT056 to display a series of prompts, allowing you to change the default values. There are two kinds prompts that the SDT056 displays: reset software and reset hardware. It displays the prompts in response to the corresponding RESET SOFTWARE or RESET HARDWARE commands. RESET without parameters displays all prompts. The next two sections list the prompts, along with the default values.

RESET SOFTWARE PROMPTS

If you enter the RESET or RESET SOFTWARE command, the SDT056 asks questions about the test limits and prompts you for responses. For these questions, the SDT056 indicates the default response with brackets, such as:

```
(Y OR [N])
```

In this case, N (or no) is the default response. If you enter a carriage return alone, the SDT056 assumes the default response.

The SDT056 prompts are:

```
Change test limits (y or [n])?*
```

If you answer yes, the SDT056 prompts you for more information about the test limits. If you answer no, it omits the remainder of the reset software prompts and assumes a default set of test limits, as follows:

- Initial segment address of 2000H
- Final segment address of 5FFFH
- Parity flag register port address of 0

If you answer yes to the first prompt, the SDT056 issues the following prompts:

```
initial segment: ([2000H])*
```

Respond by entering the address of the initial segment to be tested (or use the default).

```
final segment: ([5FFFH])*
```

Respond by entering the address of the final segment to be tested (or use the default).

The values entered in response to these two prompts must be hexadecimal values terminated with a carriage return. They must specify the base portion of the address; the offset portion is assumed to be zero for the initial segment and 0F for the final segment. Only the last four digits are used; leading zeros are not necessary. If the initial and final segments are equal, one paragraph (16 bytes) is tested. The SDT056 will not accept the following:

- An initial segment value less than 1000H
- A final segment value less than the initial segment value
- Segment values greater than FBFFH

The SDT056 next issues the following prompt:

```
Change parity port to: ([0000H])*
```

Respond by entering the port address of the parity flag register as configured on the board (or use the default, if it is accurate). The only legitimate values for this port address are 0 through 0FH and 40H through 4FH.

RESET HARDWARE PROMPTS

WARNING

> This portion of the test involves removing the top cover. It exposes risks of fire and electric shock. Only persons qualified in electronic hardware should perform this portion of the test. After removing the top cover, take care not to touch anything inside.

After you enter the RESET command to reset the hardware, the SDT056 test suite checks the iSBC 056A RAM Memory board for parity errors. You must remove the top cover of the System and observe the state of the parity LED mounted on the iSBC 056A board to determine the results of this test.

The SDT056 test displays the following:

```
Parity light should be off.   (<cr> to continue)
Parity light should be on.    (<cr> to continue)
Parity light should be off.   (<cr> to continue)
```

The operator observes an error if at any stage the state of the parity LED does not match the prompt. If an error occurs, replace the iSBC 056A board.

SDT056 TEST DESCRIPTIONS

The SDT056 consists of four tests with which to check the functionality of the iSBC 056A RAM Memory board.

WARNING

Only qualified technical personnel should perform the corrective actions listed in this section. Attempts by unqualified personnel to service the hardware can result in personal injury and damage to the system. In addition, always remove system power prior to installing or removing a board or peripheral device. Failure to observe this precaution can result in personal injury and circuit damage.

Test 0. Fixed Patterns Test

The Fixed Patterns test checks the ability of each RAM location to store word values 5555H and AAAAH. This pattern of alternating ones and zeros is useful for determining problems with data lines. It is a standard checkerboard type RAM memory test.

This test can return the following error message:

```
    error at <ssss:nnnn>
    expected = <xxxx> received = <yyyy> reread = <rrrr> xor = <zzzzzzzzzzzzzzzz>
```

where <ssss:nnnn> indicates the segment and offset of the failing RAM location, <xxxx> and <yyyy> are the expected and received values, <rrrr> is the value obtained when the location was read again, and <zzzzzzzzzzzzzzzz> is the exclusive OR of the expected and received values (in binary). The xor field indicates the failing data bits. On a board arranged in banks and rows (such as the iSBC 056A board), this information also indicates the RAM chip at fault. If several chips appear to be at fault, it is possible that the data lines are stuck or shorted together.

If this test fails, replace the iSBC 056A board.

Test 1. Address March Test

The Address March test writes a background pattern to each word in the test range. Then it "marches" the complement of that pattern through the test range, one word at a time. If the test encounters a cell that contains something other than the background pattern, it logs an error. At the end of the pass, the test checks the parity register to determine if any parity errors occurred during the test.

This test can return the same error message that test 0 returns. If this test indicates an error for a location in which test 0 did not indicate an error, the error is an address error. If tests 0 and 1 both return errors for the same location, the error is a data error. The xor field of the error message indicates the RAM chip at fault. If several chips appear to be at fault, it is possible that the address lines are stuck or shorted together.

This test can also return the following error message:

```
PARITY ERROR:  BANK NUMBER = <x>   ROW NUMBER = <y>
```

This indicates that a parity error occurred, where <x> specifies the bank of memory in which the error occurred. Possible values include 0 (error in the high-order byte), 1 (error in the low-order byte), BOTH (error in both bytes), and NEITHER (spurious error, suggesting that the parity logic is faulty). The <y> indicates the row of memory and can have a value of 0 or 1.

If this test fails, replace the iSBC 056A board.

Test 2. Sliding Ones Pattern

The Sliding Ones Pattern test checks RAM memory on the iSBC 056A board. It first writes a pattern to all of the RAM cells. Then it reads the first cell and compares the pattern read with the pattern written. Then it rewrites the cell with the next pattern, reads the pattern, and compares the pattern read with the pattern written. This sequence is repeated for each RAM cell. At the end of each sequence, the test examines the contents of the parity register to determine if any parity errors occurred. The sequence is repeated for each of the 16 different patterns. The following values are used for the different patterns:

0000	0F0FH	FFFFH	F0F0H
0101H	1F1FH	FEFEH	E0E0H
0303H	3F3FH	FCFCH	C0C0H
0707H	7F7FH	F8F8H	8080H

This test can return the same error messages that tests 0 and 1 return. If this test indicates an error for a location which neither test 0 or 1 indicated an error, the error is probably a pattern sensitivity error. Otherwise it is a data or address error.

If this test fails, replace the iSBC 056A board.

Test 3. Execute From RAM

The Execute from RAM test verifies that a program can be executed from off-board RAM. It is initially set to be ignored. When tests 0 and 1 pass, this test can be exercised by first using the RECOGNIZE command. The minimum prerequisite for running this test is for tests 0 and 1 to pass. When those tests pass, the RAM cells are functional. If either of those tests fail, the results from this test will be extremely erratic, because this test will be unable to copy code into the test area correctly.

This test uses a combination of writes and reads to thoroughly exercise the bus driver logic. It writes a block of code into a series of 64K-byte blocks of RAM in the test range. This code contains a pattern embedded in the middle of it. After executing the code, the test reads the pattern and compares it to the expected value.

Before you run this test, you must set the test limits large enough to contain the entire block of code that this test loads into RAM. If the test range is too small, this test displays the following message without running:

```
bounds too small for execution
```

This test can also return the same error message that test 0 returns.

If this test fails, replace the iSBC 056A RAM Memory board.

SDT215 DIAGNOSTIC TEST

The SDT215 Diagnostic test checks the functionality of the disk controller and the device (Winchester or floppy). This test checks only one device at a time. It asks the user which device is to be tested. The SDT215 test consists of 17 different tests, each of which checks different portions of the disk controller and/or drive. Table 4-6 lists the tests.

Table 4-6. SDT215 Tests

TEST NUMBER	FUNCTION
0	Reset Test
1	Transfer Status
2	Buffer I/O Test
3	ROM Checksum Test
4	RAM Window Test
5	RAM Address Test
6	Micro-Diagnostic
7	Seek/Verify Test
8	Format Test
9	Write/Read Test
A	Drive Selection
B	Platter/Head Test
C	Sector Selection
D	Track Verify
E	Platter Verify
F	Overlap Test (Winchester test only) or Write/Read Deleted Data (Flexible disk drive test only)

When you invoke the SDT215 diagnostic test, it displays information, asks questions about the device to be tested, and prompts you for responses. For any question requiring a yes or no response, the test indicates the default response with brackets, such as:

(Y OR [N])

In this case, N (or no) is the default response. If you enter a carriage return alone in response to a question of this kind, the test assumes the default response.

When invoked, the SDT215 diagnostic test displays the following information:

```
SYSTEM DIAGNOSTIC TEST - 215, Vx.x
```

where x.x is the version number

Next, the SDT215 Diagnostic test prompts you for reset software information. You should enter the values that reflect the device you are testing (Winchester or flexible disk drive). If you want to change these values while you are running the SDT215 test suite, you can use the RESET command as described earlier in this chapter. If you want to change the device being tested (such as changing from Winchester to flexible disk drive), you should enter RESET without parameters to reset both the hardware and the software.

RESET SOFTWARE PROMPTS

Immediately after the SDT215 displays its sign-on message (or when you enter the RESET command to reset the software), the SDT215 prompts for answers to the following questions:

> ENTER A 1 TO 5 DIGIT DECIMAL RANDOM NUMBER SEED.
> (This number is used to derive a random number.)
>
> WHICH UNIT IS BEING TESTED?
> WINCHESTER (0) OR FLOPPY (1) - ENTER NUMBER
>
> IS UNIT 0 BEING TESTED (Y OR [N])
>
> (The standard configuration of the system assigns unit 0 for both the Winchester and the flexible disk drives.)
>
> If the answer is no, the system sequentially prompts if units 1, 2, or 3 are being tested.
>
> If the answer is yes, the next questions are asked.
>
> IS THIS UNIT BACKED-UP (Y OR [N]).
>
> If the answer is yes, certain tests within this test suite may destroy data on the disk under test.
>
> DO YOU WANT TO USE THE INITIALIZATION DEFAULTS (Y OR [N]).
>
> If the answer is yes, the system displays the following:
>
> PASS
> iSBC/VERSION Vx.x

If the answer is no, the SDT215 requests the following hardware information:

NUMBER OF TRACKS/SURFACE

Default number for Winchester is 520 (decimal).
Default number for flexible disk is 77 (decimal).

NUMBER OF FIXED SURFACES

Default number for Winchester is 5.
This question is not asked when testing a flexible disk.

NUMBER OF REMOVABLE SURFACES

Default is 2 for a flexible disk device.
This question is not asked when testing a Winchester.

NUMBER OF SECTORS/TRACK

Default number for Winchester is 12.
Default number for flexible disk is 26.

NUMBER OF BYTES/SECTOR

Default number for Winchester is 1024.
Default number for flexible is 256.

NUMBER OF ALTERNATE TRACKS

Default number for Winchester is 10.
This question is not asked when testing a flexible disk.

FM OR MFM (0 or 1)

Default is MFM
This question is not asked when testing a Winchester.

Note that when a "b" is entered in response to any of the questions, the system display will back up to the previous question.

SDT215 TEST DESCRIPTIONS

The SDT215 consists of a number of tests with which to check the functionality of the iSBC 215 board, the Winchester disk drive, and the flexible disk drive.

WARNING

Only qualified technical personnel should perform the corrective actions listed in this section. Attempts by unqualified personnel to service the hardware can result in personal injury and damage to the system. In addition, always disconnect the power cord prior to installing or removing a board or peripheral device. Failure to observe this precaution can result in personal injury and circuit damage.

Test 0. Reset Disk Test

The Reset Disk test checks the ability of the controller to reset on only the proper wake-up address. This test sends a reset signal and a channel attention signal to the disk controller. Upon receipt of the channel attention signal, the disk controller begins its initialization process by fetching initialization data from ROM and calculates the address of the wake-up block in system memory. The wake-up block is fetched and a link with the I/O communication blocks established.

If this test fails, replace the disk controller.

Test 1. Transfer Status

The Transfer Status test checks the basic handshaking between the processor board and the disk controller board. It ensures that the attached units can be selected by executing the transfer error status function for each attached device.

If this test fails, replace the disk controller board.

Test 2. Buffer I/O Test

The Buffer I/O test checks the validity of the data transfers to and from the disk controller. This routine transfers all possible 8-bit numbers from memory to the controller and back. This is the first occurrence of any DMA operation.

If this test fails, replace the disk controller board.

Test 3. Checksum Test

The Checksum test calculates the checksum of the firmware. It sums the contents of all ROM locations (excluding the stored checksum) with carry added in. It then compares the calculated checksum value to the original checksum value stored in the ROM. Each ROM cell has its own byte checksum providing a checksum for even bytes and one for odd bytes.

If this test fails, replace the disk controller board.

Test 4. RAM Window Test

The RAM Window test writes a word of all zeros to all of the controller's RAM locations and then fills one cell with all ones. This value is then written to each memory cell in turn. After each write, every RAM cell is tested to determine if the word of all ones written to one cell has an adverse effect on any other cell. Next, all cells are set to all ones with only one cell set to all zeros. This value is then propagated through each memory cell in turn. After each move, every RAM cell is again tested to determine if the word of all zeros has an adverse effect on any other location.

If this test fails, replace the disk controller board.

Test 5. RAM Address Test

The RAM Address test copies the contents of the controller's ROM into its RAM and then compares both. The contents of RAM are then inverted and ANDed to the contents of RAM, which should produce a value of zero. This test ensures that the RAM address lines are operational and that each bit in RAM can be toggled.

If this test fails, replace the disk controller board.

Test 6. Micro-Diagnostic Test

The Micro-Diagnostic test performs on-board diagnostic self-test functions on the iSBC 215 Winchester disk controller to ascertain the functionality of the disk controller.

If this test fails, replace the disk controller board.

Test 7. Seek/Verify Test

The Seek/Verify test causes either the Winchester drive to seek to the diagnostic track (track 519 decimal) or the flexible disk drive to seek to the last track (track 76 decimal), select head zero, and then read the ID and data fields and verify the ECCs. The routine then causes the device to access track 0 for Winchester and single-density flexible disk drive (track 1 double-density flexible disk drive) and then reads the ID and Data fields and verifies the ECCs.

If this test fails, refer to the "Troubleshooting Hints" section at the end of this chapter.

Test 8. Format Diagnostic Track

CAUTION

> Since flexible diskettes do not have diagnostic tracks, this test destroys user data on a flexible disk drive. It is not run for flexible disk drives unless you answer yes to the "IS THIS UNIT BACKED-UP" prompt and specifically invoke the test.

The Format Diagnostic Track test formats the diagnostic track with an interleave factor of 4 and the sector size determined at initialization.

If this test fails, refer to the "Troubleshooting Hints" section at the end of this chapter.

Test 9. Write/Read Diagnostic Track

CAUTION

> Since flexible diskettes do not have diagnostic tracks, this test destroys user data on a flexible disk drive. It is not run for flexible disk drives unless you answer yes to the "IS THIS UNIT BACKED-UP" prompt and specifically invoke the test.

The Write/Read Diagnostic Track test transfers a predetermined number of sectors of contiguous data from system memory to the diagnostic track. After writing the data, a read is performed and the contents of the write and read buffers are compared.

The diagnostic track must be formatted before invoking this test. If this test fails, refer to the "Troubleshooting Hints" section at the end of this chapter.

Test A. Drive Selection Test

CAUTION

> Since flexible diskettes do not have diagnostic tracks, this test destroys user data on a flexible disk drive. It is not run for flexible disk drives unless you answer yes to the "IS THIS UNIT BACKED-UP" prompt and specifically invoke the test.

The Drive Selection test verifies that each attached device can be addressed. All attached drives are accessed to determine if ready. The diagnostic track on each of the available devices is formatted with data unique for each attached device. The data is read and compared with the data written.

This test is initially ignored because it requires at least two attached drives of the same type.

If this test fails, refer to the "Troubleshooting Hints" section at the end of this chapter.

Test B. Platter/Head Selection Test

CAUTION

> Since flexible diskettes do not have diagnostic tracks, this test destroys user data on a flexible disk drive. It is not run for flexible disk drives unless you answer yes to the "IS THIS UNIT BACKED-UP" prompt and specifically invoke the test.

The Platter/Head Selection test verifies that a platter and a head can be uniquely addressed. It accesses the diagnostic track of all attached devices and writes the track and head number in the ID field of each attached device.

If this test fails, refer to the "Troubleshooting Hints" section at the end of this chapter.

Test C. Sector Selection Test

CAUTION

> Since flexible diskettes do not have diagnostic tracks, this test destroys user data on a flexible disk drive. It is not run for flexible disk drives unless you answer yes to the "IS THIS UNIT BACKED-UP" prompt and specifically invoke the test.

The Sector Selection test verifies that each sector is uniquely addressable. The diagnostic track of each attached device is accessed and a unique sector number is written into the ID field of each sector. Each sector is then read and the contents compared with that written.

If this test fails, refer to the "Troubleshooting Hints" section at the end of this chapter.

Test D. Track Verify Test

The Track Verify test performs a seek beginning at track 7 and accesses every thirteenth track. At each track, it verifies one sector of data. It does this for all attached units. The track must be formatted prior to invocation of this test.

This test is non-destructive.

If this test fails, refer to the "Troubleshooting Hints" section at the end of this chapter.

Test E. Platter Verify Test

The Platter Verify test reads each ID field from all sectors of all cylinders of all heads on all attached units. This routine is non-destructive. It takes about ten and one-half minutes to execute this test on a Winchester drive and about two and one-half minutes to execute on a flexible disk drive.

If this test fails, refer to the "Troubleshooting Hints" section at the end of this chapter.

Test F. (Winchester Only) Overlap Test

The Overlap test verifies the ability of the disk controller to properly handle overlapped seek operations on two attached Winchester devices. The attached devices are issued a recalibrate operation which accesses track 0. Next, one device executes a seek operation to its diagnostic track while the other verifies a sector at track 0.

This routine is executed only when more than one device of the same type is attached, and is initially ignored.

If this test fails, refer to the "Troubleshooting Hints" section at the end of this chapter.

Test F. (Flexible Disk Drive Only) Write/Read Deleted Data

CAUTION

> This test destroys user data on a flexible disk drive. It is not run unless you answer yes to the "IS THIS UNIT BACKED-UP" prompt and specifically invoke the test.

The Write/Read Deleted Data test verifies the write and read deleted data function of the flexible disk drive. This routine writes deleted data on the last track, reads the deleted data, and then compares the data read to the data written.

If this test fails, refer to the "Troubleshooting Hints" section at the end of this chapter.

SDT215 MESSAGES

If you set the DEBUG command to TRUE, the SDT215 tests can return the following error messages:

BUSY ERROR	The test could not initiate a function because the controller's channel 1 is busy when it should be available.
INTERRUPT TIME-OUT	The controller has not interrupted to indicate that the operation is complete.
NON-INTERRUPT TIME-OUT	The controller has not signalled that the operation is complete for a function that had inhibited the operation complete interrupt.
TIME-OUT ON SEEK COMPLETE	The controller has not signalled that the seek is complete.

In addition, if you set the DEBUG command to TRUE, and the error bit in the Controller Invocation Block (CIB) of the iSBC 215 controller is set, the SDT215 tests return messages of the following format:

```
<error type>
<uu>  UNIT NUMBER
<bbbbbbbb> - OLD * CIB STATUS * NEW - <bbbbbbbb>
**  <error type>  **
<cccccccc cccccccc$cccccccc>  ERROR STATUS BITS **
<message>

           Desired        Actual
cyl         <xxxx>         <xxxx>
head        <xx>           <xx>
sect        <xx>           <xx>
Number of retries -        <num>
```

where:

<error type>	Message that indicates the type of operation that resulted in the error. Possible values include:

DIAG ERROR	The micro-diagnostics returned an error.
FORMAT ERROR	A format error occurred.
LBUF ERROR	An I/O buffer error occurred.
READ ERROR	A read error occurred.
READID ERROR	A read sector ID error occurred.
RESET ERROR	A reset error occurred.
SEEK ERROR	A seek error occurred.
TRANST ERROR	A transfer error status error occurred.
VERIFY ERROR	A verify error occurred.
WRITE ERROR	A write error occurred.
WRTBUF ERROR	An error occurred while writing the controller buffer to the disk.

<uu>	Hexadecimal unit number of the device on which the error occurred.
<bbbbbbbb>	Binary values which represent the value of the Controller Invocation Block before and after the test used the transfer error status function to determine the type of error.

<cccccccc cccccccc$cccccccc>

Binary value representing the contents of the error status byte and word.

<error type> This field can contain one of the following messages:

HARD ERROR REPORTED BY CONTROLLER

SOFT ERROR REPORTED BY CONTROLLER

<message> One of the following messages:

CYLIN. ADDR MISC	Cylinder address miscompare. The ID field contains a cylinder address that is different from the expected cylinder address.
DATA FIELD	The controller detected a correctable error in the data field of a sector. Examine the "Number of retries" field of this message.
DEFECTIVE ALTERNATE	The alternate cylinder is also defective.
DIAGNOSTIC FAULT	A micro-diagnostic error occurred.
DRIVE FAULT	A fault in the specified drive occurred and is characterized by a read/write fault, a position fault, a power fault, or a speed fault.
END OF MEDIA	The controller detected an end of media.
ID FIELD	The controller detected a correctable error in the ID field of a sector. Examine the "Number of retries" field of this message.
ILLEGAL SECTOR SIZE	The initialized sector size does not agree with the formatted sector size.
INVALID ADDRESS	The test attempted to access a cylinder beyond the range of available tracks (including alternates).
INVALID COMMAND	The controller was issued an invalid function or parameter.

SECTOR NOT FOUND	The controller could not find the specified sector.
SEEK ERROR	The controller detected a seek error.
SELECTED UNIT NOT READY	The selected unit is either not ready, not connected, or not responding to attempts to connect it.
SYNC NOT FOUND	The read electronics could not synchronize on either a data or an ID field.
WRITE PROTECTION FAULT	The test attempted to write to a write-protected unit.

<xxxx> and <xx>	Location on the disk where the test intended to perform the operation (Desired) and where the operation was actually performed (Actual). These values list the cylinder, head, and sector numbers.
<num>	Number of times the controller tried to perform the specified operation.

SDT337 DIAGNOSTIC

The SDT337 Diagnostic test consists of one test which checks all of the functions of the iSBC 337 Numeric Data Processor. If this test fails, replace the iSBC 337 Numeric Data Processor board or the processor board.

WARNING

Only qualified technical personnel should replace boards in the System. Attempts by unqualified personnel to service the hardware can result in personal injury and damage to the system. In addition, always disconnect the power cord prior to installing or removing a board. Failure to observe this precaution can result in personal injury and circuit damage.

If the DEBUG command is set to TRUE, the SDT337 test can return messages of the following format:

```
Opcode failure - <instruction name>
```

Where <instruction name> is the name of the 8087 instruction that failed.

TROUBLESHOOTING HINTS

Whenever an SDT test indicates an error, check the configuration of the individual boards before replacing any of the hardware. An invalid jumper configuration on one board might cause the diagnostic tests for that board to return errors. However, the same invalid configuration might instead cause other diagnostic tests to return errors. Refer to the INSTALLATION AND MAINTENANCE MANUAL for your system to determine the proper board configurations.

If multiple errors associated with Multibus System bus masters are reported, check the backplane on the card cage. There are two ICs on the backplane which are involved with the priority resolution scheme of the System. The priority resolution circuit on the backplane resolves bus contention between the processor board and the disk controller board.

If a seek, read, or write error is reported when testing a particular drive (Winchester or flexible disk drive), it is difficult to isolate a problem to a single defective part (board, drive, or cable). However, you can ascertain the malfunctioning part by performing the following sequences:

1. If a failure occurs during tests 7 through F of the SDT215 test suite while testing a flexible disk drive, run the entire SDT215 test suite on the Winchester.

    - If the tests fail, the probable failing unit is either the disk controller or the cables. To isolate the problem further, ensure that the cables are intact and correctly installed and rerun the tests. If the tests still fail, replace the disk controller.

    - If the tests pass when testing the Winchester and you normally use the iRMX 86 Operating System, bootstrap load the Operating System and run the Disk Verification Utility on the flexible disk. You invoke this utility by entering the DISKVERIFY Human Interface command. The Disk Verification Utility verifies the data structures of iRMX 86 physical and named volumes. It can also be used to recreate file structures on a damaged volume in order to salvage some of the valid data. If this fails or if you are going to run a different operating system, reformat the flexible diskette. If format errors occur, replace the diskette and rerun the entire SDT215 test suite on the flexible disk drive once again. If it fails again, replace the flexible disk drive.

2. If a failure occurs during tests 7 through F of the SDT215 test suite while testing a Winchester drive, run the entire SDT215 test suite on the flexible disk drive.

    - If the tests fail, the probable failing unit is either the disk controller or the cables. To isolate the problem further, ensure that the cables are intact and correctly installed and rerun the tests. If the tests still fail, replace the disk controller.

    - If the tests pass when testing the flexible disk and you normally use the iRMX 86 Operating System, bootstrap load the Operating System and run the Disk Verification Utility on the Winchester disk. You invoke this utility by entering the DISKVERIFY Human Interface command. The Disk Verification Utility verifies the data structures of iRMX 86 physical and named volumes. It can also be used to recreate file structures on a damaged volume in order to salvage some of the valid data. If this fails or if you normally run a different operating system, reformat the Winchester diskette and rerun the entire SDT215 test suite on the Winchester disk once again. If any errors occur, replace the Winchester disk drive.

CAUTION

Back up all files on the disk before reformatting. Once the drive is reformatted, all data is lost. After reformatting, install the Operating System onto the Winchester.

APPENDIX A. SDT SAMPLE DIAGNOSTIC TEST

The following sample diagnostic test source listing contains additional information for those who wish to create their own diagnostic test.

```
$title ('SDT SAMPLE DIAGNOSTIC TEST SUITE')
/************************************************************************
*
*   TITLE:          Sample diagnostic test
*
*   DATE:           December 23, 1981
*
*   ABSTRACT:       This module contains four short tests to be run under
*                   the SDT.
*
*   LANGUAGE DEPENDENCIES:   PLM86
*
************************************************************************/

sdt$sample$test:
DO;
/*
 *
 *      SDT routines and variables
 *
 *
 */
td$display:
    PROCEDURE (string$ptr) EXTERNAL;
    DECLARE string$ptr POINTER;
END td$display;

td$display$char:
    PROCEDURE (char) EXTERNAL;
    DECLARE char WORD;
END td$display$char;

td$read$line:
    PROCEDURE (buffer$ptr) EXTERNAL;
    DECLARE buffer$ptr POINTER;
END td$read$line;

td$set$tdt$ptr:
    PROCEDURE (td$ptr) EXTERNAL;
    DECLARE tdt$ptr POINTER;
END td$set$tdt$ptr;

td$start:
    PROCEDURE EXTERNAL;
END td$start;
```

```
$eject
td$new$line:
    PROCEDURE EXTERNAL;
END td$new$line;

DECLARE td$version(4)   BYTE      EXTERNAL,
        td$erronly      WORD      EXTERNAL,
        td$debug        WORD      EXTERNAL;
/*
 *
 *      constant and literal declaration
 *
 */

DECLARE cr          LITERALLY '0dh',
        lf          LITERALLY '0ah',
        true        LITERALLY '0ffh',
        false       LITERALLY '0';

DECLARE pass$name (*) BYTE DATA
    ('Pass:  Always passes', 0);

DECLARE fail$name (*) BYTE DATA
    ('Fail:  Always fails', 0);

DECLARE input$string$name (*) BYTE DATA
    ('Input string:  Reads a string', 0);

DECLARE version$number$name (*) BYTE DATA
    ('Version number:  Monitor Version', 0);
/*
 *
 *      test procedures
 *
 */

pass:  PROCEDURE BYTE PUBLIC;
    RETURN true;
END pass;

fail:  PROCEDURE BYTE PUBLIC;
    IF td$debug
        THEN CALL td$display (@('Proceeding to fail.', cr, lf, 0));
    RETURN false;
END fail;
```

```
$eject
input$string:  PROCEDURE BYTE PUBLIC;
    DECLARE buffer(123) BYTE;

    CALL td$display(@('Input example -- enter a string ', 0));
    CALL td$read$line(@buffer);

    IF buffer(0) = 'Y'        /*   SDT uppercases all input. */
        THEN DO;
            IF td$debug THEN
                   CALL td$display (@('Test passes if the string ',
                        'begins with a "y" or "Y" ', cr, lf, 0));
            RETURN pass;
          END;

        ELSE DO;
            IF td$debug THEN
                   CALL td$display (@('Test fails if the string does',
                        ' not begin with a "y" or "Y" ', cr, lf, 0));
            RETURN fail;
          END;
END input$string;

version$number: PROCEDURE BYTE PUBLIC;
    DECLARE I BYTE;

    CALL td$display (@('SDT Version number ', 0));

    DO I = 0 TO 3;
        CALL td$display$char (td$version(I));
    END;

    CALL td$new$line;
    RETURN true;

END version$number;
/*
 *
 *      Public procedures and variables required by SDT
 *
 */

user$reset$software:
    PROCEDURE PUBLIC REENTRANT;
    CALL td$display (@('User software RESET invoked', cr, lf, 0));
END user$reset$software;

user$reset$hardware:
    PROCEDURE PUBLIC REENTRANT;
    CALL td$display (@('User hardware RESET invoked', cr, lf, 0));
END user$reset$hardware;
```

```
$eject
DECLARE user$signon (*) BYTE PUBLIC DATA
    ('SDT SAMPLE DIAGNOSTIC TEST, V 1.0', 0);

DECLARE user$copyright (*) BYTE PUBLIC DATA
    ('(C) INTEL CORP., 1982');

DECLARE user$number$of$tests WORD PUBLIC DATA (4);

DECLARE user$tdt (4) STRUCTURE
    (flag          BYTE,
     overlay       BYTE,
     addr          POINTER,
     name$ptr      POINTER,
     err$cnt       WORD,
     exec$cnt      WORD)  PUBLIC INITIAL
    (0,  0,  @pass,  @pass$name,  0,  0,
     0,  0,  @fail,  @fail$name,  0,  0,
     0,  0,  @input$string,  @input$string$name,  0,  0,
     0,  0,  @version$number,  @version$number$name,  0,  0);

/*
 *
 *      Main program
 *
 */

    CALL td$set$tdt$ptr (@user$tdt);         /*  Necessary only if more
                                                 than one test descriptor
                                                 table is in use  */

    CALL td$start;      /* this routine should be called once only  */

END sdt$sample$test;
```

SDTMON SAMPLE DIAGNOSTIC TEST

The following is an example of an SDT SUBMIT file which compiles, links, and locates an SDT diagnostic and places it in its own library.

```
;
;   sdtsam.csd -- generate sample test, SDT based
;
plm86   sdtdir/sdtsam.p86 compact optimize(3)  &
        object(sdtsam.obj)  print(sdtsam.lst)

;
;   link all files together
;
link86  :fd0:sdtdir/sdtmon.obj,         &
        sdtsam.obj,                     &
        :fd0:sdtdir/sdtcom.lib          &
    to  sdtsam.lnk  print(sdtsam.mp1)

;
;   locate linked file
;
loc86   sdtsam.lnk  to  sdtsam.loc     &
        reserve(00h to 0100Fh)  print(sdtsam.mp2)

;
;   generate bootable library
;
delete  sdtsam     ; remove any old copies

lib86
create sdtsam
add sdtsam.loc to sdtsam
exit

;
;   SDTSAM test now ready; press RESET button to enter monitor
;
;   Boot test by entering:   b /user/sdtsam
;
```

INDEX

Underscored entries are primary references.

INDEX (continued)

REQUEST FOR READER'S COMMENTS

Intel's Technical Publications Departments attempt to provide publications that meet the needs of all Intel product users. This form lets you participate directly in the publication process. Your comments will help us correct and improve our publications. Please take a few minutes to respond.

Please restrict your comments to the usability, accuracy, readability, organization, and completeness of this publication. If you have any comments on the product that this publication describes, please contact your Intel representative. If you wish to order publications, contact the Intel Literature Department (see page ii of this manual).

1. Please describe any errors you found in this publication (include page number).

2. Does the publication cover the information you expected or required? Please make suggestions for improvement.

3. Is this the right type of publication for your needs? Is it at the right level? What other types of publications are needed?

4. Did you have any difficulty understanding descriptions or wording? Where?

5. Please rate this publication on a scale of 1 to 5 (5 being the best rating). ______

NAME ______ DATE ______

TITLE ______

COMPANY NAME/DEPARTMENT ______

ADDRESS ______

CITY ______ STATE ______ ZIP CODE ______
(COUNTRY)

Please check here if you require a written reply. ☐

WE'D LIKE YOUR COMMENTS . . .

This document is one of a series describing Intel products. Your comments on the back of this form will help us produce better manuals. Each reply will be carefully reviewed by the responsible person. All comments and suggestions become the property of Intel Corporation.

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 79 BEAVERTON, OR

POSTAGE WILL BE PAID BY ADDRESSEE

Intel Corporation
5200 N.E. Elam Young Pkwy.
Hillsboro, Oregon 97123

OMO Technical Publications

INTEL CORPORATION, 3065 Bowers Avenue, Santa Clara, California 95051 (408) 987-8080

Printed in U.S.A.