SYSTEM 86/300 SERIES DIAGNOSTIC REFERENCE MANUAL

Order Number: 143896-001

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

REV.	REVISION HISTORY	PRINT DATE
001	Original issue.	1/82

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.

A572/282/ 3K DD

PREFACE

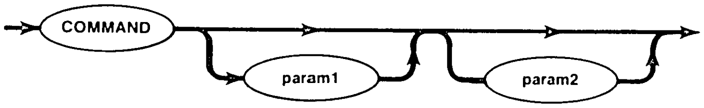
This manual describes the three diagnostic software tools supplied with the System 86/300 Series Microcomputer Systems: the System Confidence Test (SCT), the System Analysis Test (SAT), and the System Diagnostic Test (SDT). These tests check the system hardware and determine the system software's ability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

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.
[ ]	The brackets indicate optional parameters.

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:



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 86/330 Overview Manual	143898
System 86/330 Installation and Maintenance Manual	143897
User's Guide for the iSBC® 957B iAPX 86, 88 Interface and Execution Package	143979
PL/M-86 User's Guide for 8086-Based Development Systems	121636
iAPX 86, 88 Family Utilities User's Guide for 8086-Based Development Systems	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)

FIGURE

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 Analysis Test
- System Diagnostic 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 on PROMs on the processor board and is automatically invoked when power is first applied to the system or when the front panel RESET pushbutton is pressed. The SCT is described in more detail in Chapter 2.

SYSTEM ANALYSIS TEST

The System Analysis Test (SAT) provides a means to interactively exercise the system hardware with the system software for extended periods of time. This permits isolation of those elusive intermittently reported errors (subtle problems) to a given area within the System. The SAT is described in more detail in Chapter 3.

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.

RECOMMENDED TEST SEQUENCE

Upon successful completion of the SCT (no problems are encountered), the SAT may be invoked and executed to ensure that both the software and the hardware function together.

The SAT should be invoked upon receipt of the System to ensure the operational readiness of the system. It may be used occasionally as a system confidence test or whenever the occasion warrants to help isolate intermittent problems.

If a problem is encountered as a result of running the SCT, the appropriate test suite of the SDT should be invoked 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, replace the suspect board or drive (refer to Table 2-1). After replacement of the defective part, invoke the SDT.

Any problem reported by either the SCT or SAT may be used to determine a specific SDT test suite to invoke and execute. Specific routines within a particular test suite may be executed separately to further isolate problems. Execution of a specific test suite will provide more meaningful data with which to determine the failing board or drive. Upon successful completion of the SDT, run the SAT.

If a problem is reported due to the execution of the SDT, replace the suspect board, drive, or cable. After replacement, invoke the SDT again and then invoke and execute the SAT.

If an error is reported during the execution of the SAT, it points to a malfunctioning area within the system, but is not definitive. For example, an error reported during an either a Basic I/O System (BIOS) operation or an Extended I/O System (EIOS) operation to the Winchester disk does not specifically mean that the Winchester drive is defective. Such an error may point to either the iSBC 215 board, the cabling, the Winchester drive or the software itself. It may even be associated with the iSBC 86/12A board or the iSBC 056 board.

Other errors that can be reported by executing the SAT are errors associated with an I/O operation to the floppy or errors associated with the Numeric Data Processor. Other types of errors reported as a result of running the SAT may indicate that something is wrong 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 executing the SAT point to a particular problem area within the System. Therefore, if any errors are reported during the execution of the SAT test, the SDT should be invoked and executed to determine the specific problem within the System. Upon successful completion of the SAT, the System is available for system use.

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 iSBC 86/12A board and is co-resident with the iRMX 86 Bootstrap Loader and the iSBC 957B monitor. It interfaces with other software only upon termination, passing control to the iRMX 86 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, turn the System AC power switch to ON. (After about 5 seconds delay, the CRT displays a series of asterisks. 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.)

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 PIC test prompts for a character input.

3. Enter:
   i

   The SCT progress and results are displayed. Refer to Figure 2-1.

After completion of the SCT, the Bootstrap Loader loads the iRMX 86 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 by either a period (.) which indicates successful completion or by a question mark (?) which indicates that an error occurred. Figure 2-1 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 part. There are ten tests.

```
      SCT330, V1.0
        TEST                                      STATUS


     USART/TIMER                                    GO
     PIC          .*.i                              GO
     ROMCKSM      .                                 GO
     SPINNING UP >>>
     PPI          ...                               GO
     NDP          .                                 GO
     RAM TEST           TOTAL MEMORY = nnnnK
       ON BOARD                                     GO
       OFF BOARD                                    GO
       EXTENDED                                     GO

    PARITY        .                                 GO
    WINCHESTER    ..                                GO
    FLOPPY        .                                 GO
    CONTRLR INT   .                                 GO

SCT SUCCESSFUL...NOW BOOTING iRMX86
```

nnnnK = decimal number of K bytes found in the system.

Figure 2-1. Successfully Completed SCT Test Results

Table 2-1. Abnormal SCT Test Results

	POSITION					
TEST	1	2	3	4	MEANING	CORRECTIVE ACTION *
USART					If GO is not displayed the halt and run lights flash	Replace processor board Defective USART 8251A
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 struck, replace processor board
ROMCKSM	?				Checksum variation	Replace processor board
SPINUP	NOT A TEST					
PPI	?				Failure at Port A	Replace processor board
		?			Failure at Port B	Replace processor board
			?		Failure at Port C	Replace processor board
NDP	?				337 Did not respond	Replace processor board or iSBC 337 board
RAM TEST						
ONBOARD					NO GO	Replace processor board or iSBC 300 board
OFFBOARD					NO GO	Replace iSBC 056 board
EXTENDED					NO GO	Replace user-added memory board
PARITY					Parity Error	Replace iSBC 056 board
WINCHESTER		?			Winchester Disk is not formatted	Perform Disk Verify
			?		iSBC 215 Diagnostic reports an error	Replace disk controller board or Winchester drive
FLOPPY					Not Ready--Door Opened (does not prevent booting)	Insert diskette, close door Press RESET
	?				Unformatted Diskette or disk controller reported an error	Insert formatted diskette Replace disk controller
CONTRLR INT	?				Interrupt from iSBC 215 did not occur.	Replace disk controller board

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

If these tests terminate normally, control transfers to the iRMX 86 Bootstrap Loader. Should an error be detected, a message is issued to the resident serial I/O channel of the iSBC 86/12A board and control transfers to the iSBC 957B monitor at completion of the test.

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 8251 USART and the 8253 Programmable Interval Timer (PIT) on the iSBC 86/12A 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 iSBC 86/12A 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	Power fail interrupt	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	00008	00008
0	00080	000E0
1	00084	000E4
2	00088	000E8
3	0008C	000EC
4	00090	000F0
5	00094	000F4
6	00098	000F8
7	0009C	000FC

This test initializes the PIC 8259A. After mode and operation are set, the interrrupt mask register is set and read back. 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 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 FC000 through FFFFF.

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.

SPIN UP

The SCT starts the spindle motor of the Winchester drive rotating. This allows time for the spindle motor to achieve rated spin speed before invoking the Winchester test routine.

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.

TEST 5. NDP

This test checks the processor's ability to communicate with the iSBC 337 Numeric Data Processor. It writes a floating point number to the status register of the numeric data processor (NDP) and compares the source number with the value on the NDP stack. If they compare, the message "GO" appears. If they do not compare, the message "NO-GO" appears.

TEST 6. RAM TEST

This test checks the ability of the processor to communicate with RAM. The dual port RAM address space spans address locations 00000H through OFFFFH. The off-board RAM address space spans address locations 10000H through 4FFFFH. 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 056 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 list) located in memory on the RAM memory board.

The processor issues a series of instructions to the I/O wake-up address of the disk controller.

This test initializes the iSBC 215 Winchester disk controller and invokes the controller micro-diagnostics. These two functions 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 or formatted incorrectly, a "NO-GO" message appears.

TEST 9. FLEXIBLE DISK

This test checks the iSBC 218 flexible disk controller. It is identical to Winchester tests except for device numbers. If an error is detected, the message "NO-GO" appears on the CRT. An unformatted diskette or a diskette with an invalid format causes a "NO-GO" message to appear. If no errors are encountered, the message "GO" appears. If a diskette is not present, a NOT READY message appears. This message should be interpreted as a GO-type message as 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.

CHAPTER 3. SYSTEM ANALYSIS TEST

INTRODUCTION

The System Analysis Test (SAT) tests the Operating System's ability to interactively communicate with the various hardware sub-assemblies of the system. 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 LINK 86 to link the modules 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 Test

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.
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 Data Processor.

SAT INSTALLATION

All of the files required for running the SAT reside on the diagnostic diskette in the directory SATDIR 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 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. With the iRMX 86 Operating System installed, bootstrap load the iRMX 86 Operating System by entering the following:

   ```
   .b
   ```

2. Insert the diskette marked "DISKETTE, TYPE G, SYSTEM 86/330 DIAGNOSTIC" 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 the iRMX 86 Operating System. The SAT can only be invoked when the iRMX 86 Operating System is in control. The iRMX 86 Operating System may be invoked from either the Winchester drive or from the flexible disk drive. To invoke the iRMX 86 Operating System from the Winchester, enter the following:

```
.b
```

To invoke the iRMX 86 Operating System from the flexible disk drive in order to invoke the SAT, enter the following:

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

INVOKING THE SAT

With iRMX 86 Operating System in control and the necessary SAT files resident on the Winchester, you can invoke the SAT by entering the following command:

```
SAT [FOREVER]
```

Where: FOREVER (any abbreviation is recognized) is an optional parameter to exercise the SAT tests forever. If you include the FOREVER parameter, you can exit the SAT only by entering a Control-C at the terminal. If the FOREVER parameter is not included, the SAT runs for default time of five minutes.

After invocation, the SAT displays the following information at the terminal:

```
sat


iRMX 86 PL/M COMPILER V1.0
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
```

This 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.

The SAT is a GO/NO-GO test. If an error occurs, run the SDT 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:

1. the test that failed

2. the operation (System call) that failed

3. the exception code

4. the number of times the failure occurred

If an error occurs during the compilation, 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 8086-BASED DEVELOPMENT SYSTEMS for more information.

If an error occurs during the link, 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 8086-BASED DEVELOPMENT SYSTEMS MANUAL 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, the status information is displayed at the terminal in a SAT SUMMARY REPORT. When the tests run successfully, this information has the following format:

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

    SAT test program:
PASS

    Winchester tests:
PASS

    Floppy tests:
PASS

    iSBC 337 test:
PASS
```

In the previous example, the SAT SUMMARY REPORT indicates that all the tests passed. If the clock is initialized, it also displays the amount of time that the SAT test program ran. If the clock was not initialized, 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 fllexible disk drive are listed under the heading Floppy tests. Errors related to the Numeric Data Processor 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. An example of some possible error messages reported by the SAT SUMMARY REPORT is as follows:

```
    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
```

As shown in the example, 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 an EIOS attempted to delete a connection five times (via the RQ$S$DELETE$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 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 four test suites designed to diagnose system failures to a defective board or device (either the Winchester drive or the flexible disk drive). 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.

Table 4-1. SDT Test Suites

TEST SUITE	FUNCTION
SDT8612	Checks 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 iSBC 337 Numeric data processor.

SDT INSTALLATION

The SDT diagnostics reside on the Diagnostic diskette shipped with the System. If a problem with the flexible disk exists, you could not invoke the SDT diagnostics unless they were resident on the Winchester. You must install the SDT diagnostics onto the Winchester to have them resident on the Winchester disk. 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 DISKETTE, TYPE G, SYSTEM 86/330 DIAGNOSTIC into the flexible disk drive.

2. Enter the following:

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

Once installed on the Winchester disk, the SDT diagnostic 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 iRMX 86 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, enter an upper case U (press the SHIFT and u keys).

   c. In response to the SCT's PIC test prompt, 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 iRMX 86 Operating System, enter:

```
b
```

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	FLEXIBLE DISK DRIVE
SDT8612	Processor	b /SDT330/SDT8612	b :WFO:/SDT330/SDT8612
SDT056	RAM Memory	b /SDT330/SDT056	b :WFO:/SDT330/SDT056
SDT215	Disk Controller	b /SDT330/SDT215	b :WFO:/SDT330/SDT215
SDT337	Data Processor	b /SDT330/SDT337	b :WFO:/SDT330/SDT337

The recommended sequence of SDT test suites to invoke is: the SDT8612 test suite, followed by the SDT056, SDT215, and then the SDT337 tests 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 a particular SDT test suite is invoked, 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 section.

A description of the test suite and each test 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

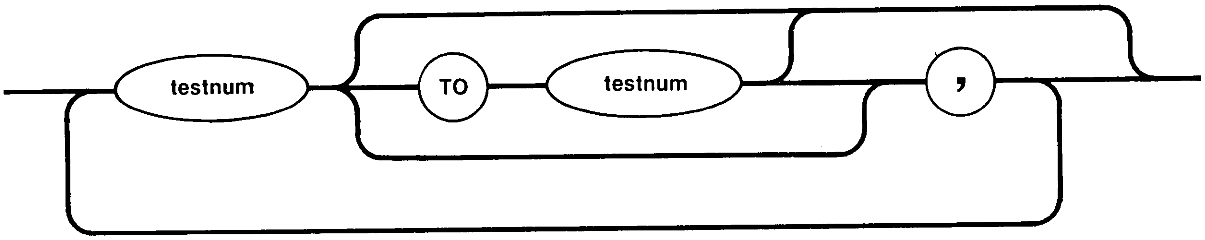
The SDT always produces numerical output in hexadecimal format. However, when you provide input to the SDT, you can specify the radix of numerical quantities by including a radix character immediately after the number. The valid radix characters include:

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:



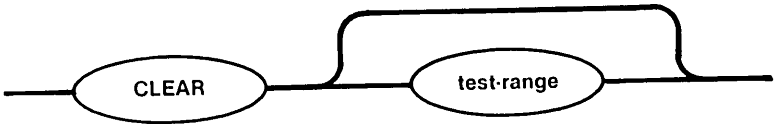
If you separate test numbers with commas, you select the individual tests (for example, 1,3 selects tests 1 and 3). If you separate two test numbers with the word TO, you select all tests between the first number and the second number (for example, 1 TO 3 selects tests 1, 2, and 3). However, if you separate 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:



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, 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
    *
```

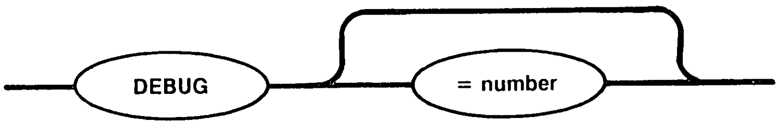
This command clears the execution and error count values for test 7.

```
    *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:



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.

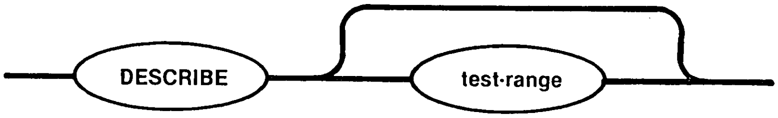
DESCRIPTION

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:



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 PATTERNS
0002H              SLIDING ONES
0003H              EXECUTE FROM RAM
*
```

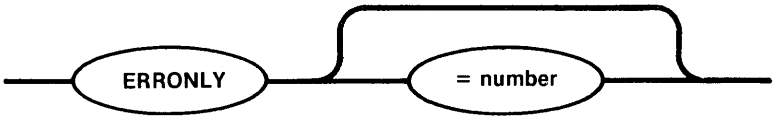
This command displays the test names and numbers for all tests in the SDT test suite (this example assumes you are running the SDT056 test suite).

```
*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:



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 debug status is set to the default value of FALSE.

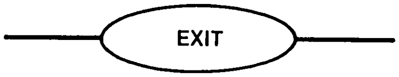
DESCRIPTION

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.

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.

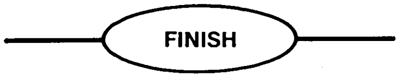
EXIT COMMAND

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:



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:



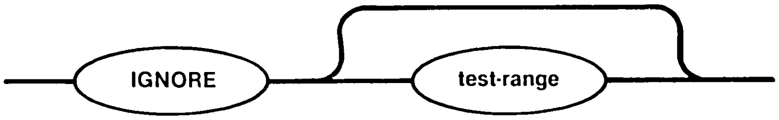
DESCRIPTION

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:



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
*
```

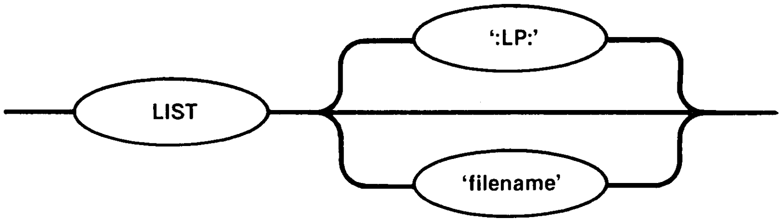
This command ignores all tests in the SDT test suite. Future TEST commands have no effect until you reverse the IGNORE status with RECOGNIZE command.

```
*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:.

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

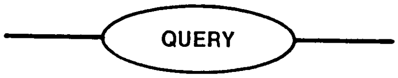
ERROR MESSAGES

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:

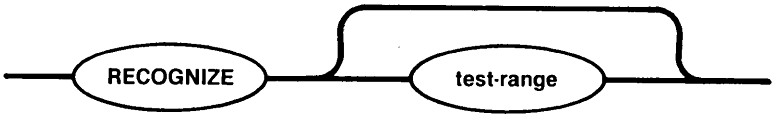


EXAMPLES

```
*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

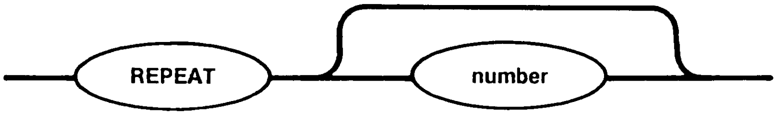
One or more of the test numbers you specified was larger than the largest test number in the SDT test suite.

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

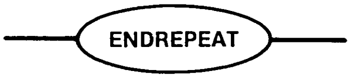
The REPEAT and ENDREPEAT commands allow you to repeat a group of commands any number of times. The REPEAT command denotes the start of the group; the ENDREPEAT command denotes the end of the group. The format of the REPEAT command is as follows:



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:



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
```

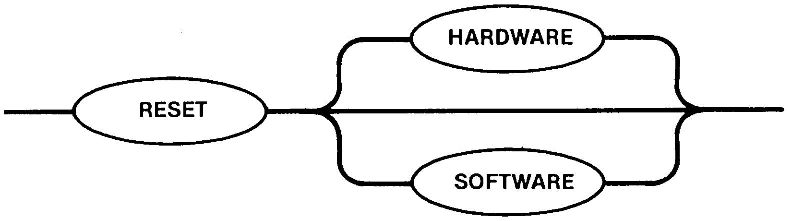
This example repeats diagnostic tests in nonsequential order. The SDT will repeat the tests until you enter a Control-C character to terminate the testing.

```
*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:



where:

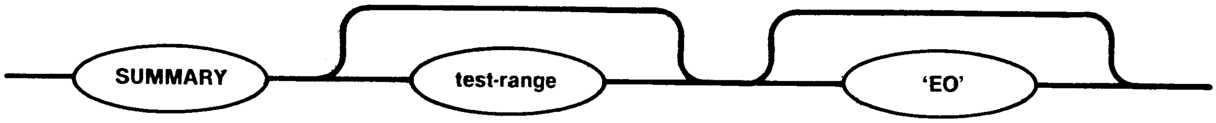
HARDWARE — Resets the hardware to its initial state. The SDT test suite may request additional information from you about the state of the hardware.

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.

SUMMARY COMMAND

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



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' — 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. 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".

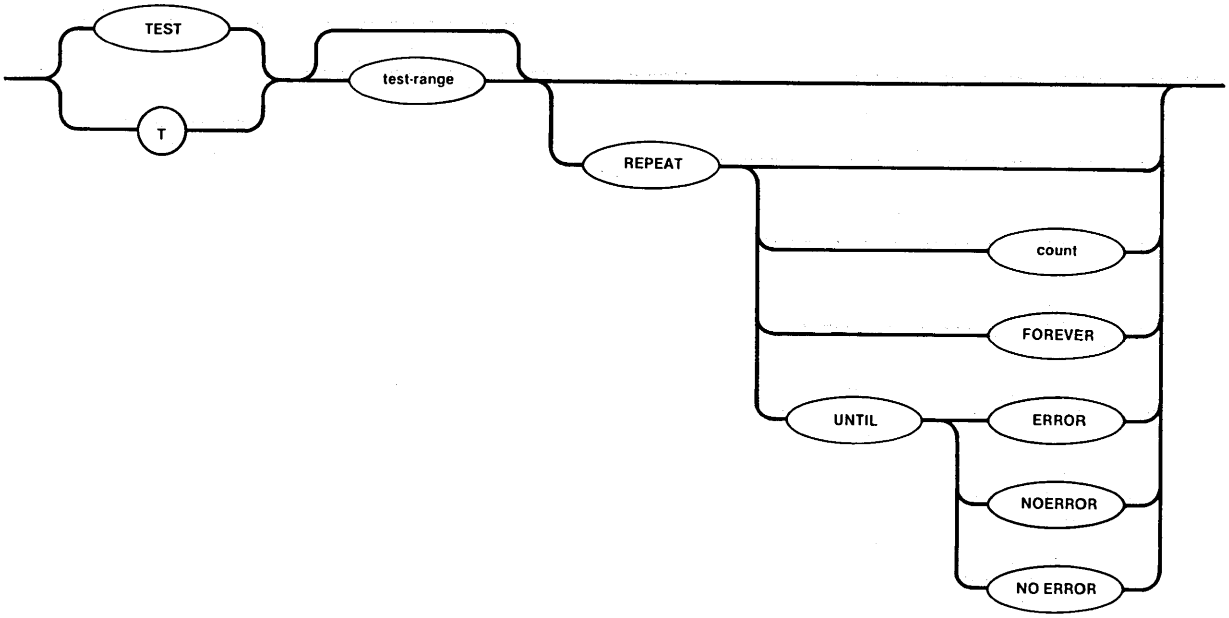
EXAMPLE

```
*SUMMARY
0000H FIXED PATTERNS                  0000 FAILED IN 0004 TRIALS
0001H ADDRESS PATTERNS                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:



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 — Repeats the tests until no errors occur.

or

UNTIL NO ERROR

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.

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".

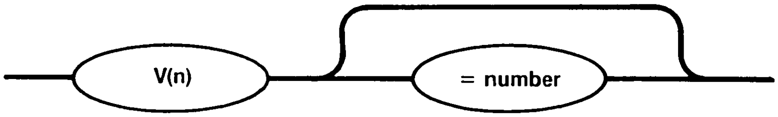
EXAMPLE

```
    *TEST
    0003H *** IGNORED ***
    0000H FIXED PATTERNS                              "PASSED"
    0001H ADDRESS PATTERNS                            "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 OFH. These variables can contain any word values. The SDT215 test is the only Intel-supplied diagnostic test that uses 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. The SDT215 test is the only Intel-supplied diagnostic test that uses the V variables. For this test suite, the following variables are meaningful:

variable	description
V(1)	If you specify an odd value (least-significant bit set to 1) for this variable, the SDT displays a message when an interrupt occurs, if that interrupt was caused by a diskette change or a seek complete status. If you specify an even value (least-significant bit set to 0), the SDT omits these messages.
V(5)	If you specify an even value for this variable, the SDT displays controller timeout and busy messages. If you specify an odd value, the SDT 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 OFH for a V variable index.

SDT8612 DIAGNOSTIC TEST

The SDT8612 diagnostic test checks the functionality of the iSBC 86/12A board and the iSBC 300 RAM Memory Extension 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 is accessed 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 300 RAM Extension Multimodule), the Winchester drive, and the iSBC 215 Winchester disk controller board must be functional. If the test is bootstrap loaded 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 SDT8612 tests and their respective functions.

Table 4-3. 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 300
6	Address Patterns Test
7	Sliding Ones Test - iSBC 86/12A
8	Sliding Ones Test - iSBC 300
9	Dual Port RAM Contention Test
A	RAM Memory Maps

SDT8612 TEST DESCRIPTIONS

TEST 0. ROM CHECKSUM

The ROM Checksum test checks the address and data lines of the ROM on the iSBC 86/12A board and calculates the checksum of the contents of the ROMs.

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.

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.

If this test fails, replace the processor or the iSBC 86/12A 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.

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 iSBC 86/12A 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.

If this test fails, replace the processor board.

TEST 5. FIXED PATTERNS - iSBC 300 BOARD

The Fixed Patterns iSBC 300 test checks RAM data lines of the iSBC 300 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 that written.

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

TEST 6. ADDRESS PATTERNS

The Address Patterns test checks the uniqueness of the RAM address lines on the iSBC 86/12A board and the iSBC 300 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.

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 iSBC 86/12A board. It first writes an all zero pattern to all of the RAM cells. The first cell is read and the results compared with the old pattern written. The cell is rewritten with the next pattern, read, and the results compared. This same sequence is repeated for each RAM cell. The sequence is repeated for each of the 16 different patterns.

If this test fails, replace the processor board.

TEST 8. SLIDING ONES - iSBC 300 BOARD

The Sliding Ones - iSBC 300A test checks RAM memory on the iSBC 300 board. It first writes an all zero pattern to all of the RAM cells. The first cell is read and the results compared with the old pattern written. The cell is then rewriten with the next pattern, read, and the results compared. This same sequence is repeated for each RAM cell. The sequence is repeated for each of the 16 different patterns.

If this test fails, replace the iSBC 300 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 iSBC 86/12A board. It causes the iSBC 215 Winchester disk controller board to perform a DMA transfer to RAM, while the iSBC 86/12A Single Board Computer 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 iSBC 86/12A board while at the same time causing the iSBC 86/12A 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 iSBC 86/12A board finishes its move operation and verifies that the data transferred from the iSBC 215 board is correct.

   If the DEBUG command is set to an even value, only a FAIL message is printed upon the occurrence of an error. When the DEBUG command is set to an odd value, this test returns two types of error messages: those occurring before the contention test and those found during the test. Error messages related to the first type are:

   ```
   iSBC 215 MICRODIAGNOSTICS FAILURE

   BUFFER I/O TRANSFER FAILURE AT ssss:nnnn
   expected xx  received yy
   ```

If an error occurs, invoke the SDT215 test. If the SDT215 test passes, suspect the bus arbitration logic or the dual port RAM control logic.

Just before the second portion (the contention portion) of the test starts, a banner "BUFFER I/O TRANSFER PASS" is displayed if Debug command is set to true. The error message related to the failure of the contention portion of the test is CONTENTION TEST FAILURE iSBC 215 DATA TRANSFER.

If an error occurs, suspect the dual port RAM control logic, especially if the SDT215 test passes.

SDT056 DIAGNOSTIC TEST

The SDT056 Diagnostic test checks the functionality of the iSBC 056 RAM Memory board. This test expects the iSBC 86/12A board to be functional. It may be called from either the Winchester disk or from the flexible disk drive. Table 4-4 lists the tests of the SDT056.

Table 4-4. SDT056 Tests

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

When the SDT056 diagnostic test suite is invoked, the following message is displayed:

```
Do you wish to change the test limits from 1000:0000
through 4000:FFFF?*
```

If you enter a carriage return or n, the default test values (initial segment 1000H and final segment 4FFFH) are used.

If you enter a y (yes), the test prompts for the altered test range. Enter the initial segment address first and then the final segment address.

The values entered must be hexadecimal values terminated with a carriage return (c/r). 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. This test 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 test suite checks the iSBC 056 RAM Memory board for odd parity errors. You must remove the top cover of the System and observe the state of the parity LED mounted on the iSBC 056 board to determine the results of this test.

The SDT056 test displays the following:

```
Parity light should be off.   (<c/r> to continue)
Parity light should be on.    (<c/r> to continue)
Parity light should be off.   (<c/r> 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 056 board.

SDT056 TEST DESCRIPTIONS

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

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.

If this test fails, replace the iSBC 056 board.

TEST 1. ADDRESS PATTERNS TEST

The Address Patterns test checks the ability of each RAM location to store a pattern which is formed by adding the four bytes of the RAM address together. Each cell is then checked for the proper contents. At the end of the pass, the parity register is checked to determine if any parity errors occurred during the test.

If this test fails, replace the iSBC 056 board.

TEST 2. SLIDING ONES PATTERN

The Sliding Ones Pattern test checks RAM memory on the iSBC 056 board. It first writes an all zero pattern to all of the RAM cells. The first cell is read and the results compared with the old pattern written. The cell is then rewritten with the next pattern, read, and the results compared. At the end of each sequence, the contents of the parity register is examined to determine if any parity errors occurred. This same 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	0F0FH	FFFFH	F0F0H
0101H	1F1FH	FEFEH	E0E0H
0303H	3F3FH	FCFCH	C0C0H
0707H	7F7FH	F8F8H	8080H

TEST 3. EXECUTE FROM RAM

The Execute From RAM test 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 pre-requiste for running this test is for tests 0 and 1 to pass. When those tests pass, the RAM cells are functional. If those tests fail, the results from this test will be extremely erratic. If any RAM cell exhibits a fault (which causes either test 0 or Test 1 to fail) might cause the code to be copied improperly or set the test limits too small, the It verifies that a program can be executed from off-board RAM with a combination ofwrites and reads that thoroughly exercises the bus arbitration logic. It writes a block of code into a series of 64K byte blocks of RAM with a pattern embedded in the middle of the code block. After executing the code, the pattern is compared.

If this test fails, replace the iSBC 056 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. The user is asked which device is to be tested. The SDT215 test consists of 17 different tests each of which check different portions of the disk controller and/or drive. Table 4-5 lists the tests.

Table 4-5. 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)

The SDT215 Diagnostic test displays the following:

SYSTEM DIAGNOSTIC TEST - 215, Vx.x

where x.x is the version number

Next, the SDT215 Diagnostic test 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 next six questions are asked.

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

TEST 0. RESET DISK TEST

The Reset 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 (iSBC 86/12A) 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 Troubleshooting Hints.

TEST 8. FORMAT DIAGNOSTIC TRACK

The Format Diagnostic Track test formats the diagnostic track with an interleave factor of 4 and the sector size determined at initialization. This test is destructive to any user data stored on the flexible disk drive and will not be run unless the unit is backed up.

If this test fails, refer to Troubleshooting Hints.

TEST 9. WRITE/READ DIAGNOSTIC TRACK

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. This test is destructive to any user data on a flexible disk drive and will not be run unless the unit is backed up.

If this test fails, refer to Troubleshooting Hints.

TEST A. DRIVE SELECTION 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 are formatted with data unique for each attached device. The data is read and compared with the data written.

This routine is destructive to any user data on the diagnostic track and should not be run unless the data is backed up or the test is specifically invoked. This test is initially ignored because it requires at least two attached drives of the same type.

If this test fails, refer to Troubleshooting Hints.

TEST B. PLATTER/HEAD SELECTION 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 Troubleshooting Hints.

TEST C. SECTOR SELECTION 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.

This test is destructive to user data residing on the flexible disk drive and will not be run unless the device is backed up.

If this test fails, refer to Troubleshooting Hints.

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 that the correct number of sectors exist (this is a predetermined number specified during initialization). The track must be formatted prior to invocation of this test.

This test is non-destructive.

If this test fails, refer to Troublshooting Hints.

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 Troubleshooting Hints.

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 access 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 Troubleshooting Hints.

TEST F. (FLEXIBLE DISK DRIVE ONLY) WRITE/READ DELETED DATA

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.

This routine may be destructive to user data if the device is not backed up.

If this test fails, refer to Troubleshooting Hints.

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.

TROUBLESHOOTING HINTS

If no asterisks appear on the terminal during the SCT test, replace the processor board. If asterisks fail to appear on the terminal, check the terminal and cable. If there is still no display, remove the RAM Memory board and the disk controller and rerun the test. Bus contention could be locking the bus.

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.

If multiple errors associated with Multibus System bus masters are ever reported, check the backplane on the card cage. There are two IC's 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 that test fails, the probable failing unit is either the disk controller or the cables. After checking that the cables are intact and correctly installed and the test still fails, replace the disk controller. If the test passes when testing the Winchester, perform a disk verification on the flexible disk drive using 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, 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. For those persons wanting to troubleshoot the flexible disk drive, refer to the troubleshooting section in the flexible disk drive manual.

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 that test fails, the probable failing unit is either the disk controller or the cables. After checking that the cables are intact and correctly installed and the test still fails, replace the disk controller. If the test passes when testing the flexible disk drive, perform a disk verification on the Winchester drive using 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, reformat the Winchester disk, and rerun the SDT215 test suite on the Winchester drive once again. If any errors occur, replace the Winchester disk drive. For those persons wanting to troubleshoot the Winchester drive, refer to the troubleshooting section in the Winchester drive manual.

CAUTION

Back up all files on the disk before reformatting. Once the drive is reformatted, all data is lost. After a format, install the iRMX Operating System onto the Winchester. Then, install the SAT and SDT Diagnostics onto the Winchester.

APPENDIX A. SDTMON SAMPLE DIAGNOSTIC TEST

SDTMON SAMPLE DIAGNOSTIC TEST

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

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

sdt$sample$test:
DO;
/*
 *
 *  SDTMON 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;
```

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

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

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 'Ødh',
        lf    LITERALLY 'Øah',
        true  LITERALLY 'Øffh',
        false LITERALLY 'Ø';

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

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

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

DECLARE version$number$name (*) BYTE DATA
    ('Version number:  Monitor Version', Ø);
```

```
$eject
/*
 *
 *  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;


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'   /* SDTMON 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 (@('SDTMON 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;
```

```
$eject
/*
 *
 *  Public procedures and variables required by SDTMON
 *
 */

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;


DECLARE user$signon (*) BYTE PUBLIC DATA
    ('SDTMON SAMPLE DIAGNOSTIC TEST, V1.0', 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);
```

```
$eject

/*
 *
 *  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

SAMPLE SDT SUBMIT FILE

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

```
;
; sdtsam.csd -- generate sample test, SDTMON based
;
plm86  :fdØ:sdt33Ø/sdtsam.p86 compact optimize(3)   &
       object(sdtsam.obj) print(sdtsam.lst)

;
; link all files together
;
link86 &
     :fdØ:sdt33Ø/sdtmon.obj,    &
     sdtsam.obj, &
     :fdØ:sdt33Ø/sdtcom.lib  &
to sdtsam.lnk print(sdtsam.mpl)

;
;locate linked file
;
loc86 &
     sdtsam.lnk to sdtsam.loc &
     reserve(ØØH to Ø1ØØFH) 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; hit 'RESET' button and enter monitor.
;
;   Boot test by 'b /user/sdtsam'.
;
```

intel®

SYSTEM 86/300 SERIES
DIAGNOSTIC REFERENCE MANUAL
143896-001

REQUEST FOR READER'S COMMENTS

Intel Corporation attempts to provide documents that meet the needs of all Intel product users. This form lets you participate directly in the documentation process.

Please restrict your comments to the usability, accuracy, readability, organization, and completeness of this document.

1. Please specify by page any errors you found in this manual.

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

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

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

5. Please rate this document on a scale of 1 to 10 with 10 being the best rating. ______

NAME ______ DATE ______
TITLE ______
COMPANY NAME/DEPARTMENT ______
ADDRESS ______
CITY ______ STATE ______ ZIP CODE ______

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.

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

O.M.S. Technical Publications

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

Printed in U.S.A.