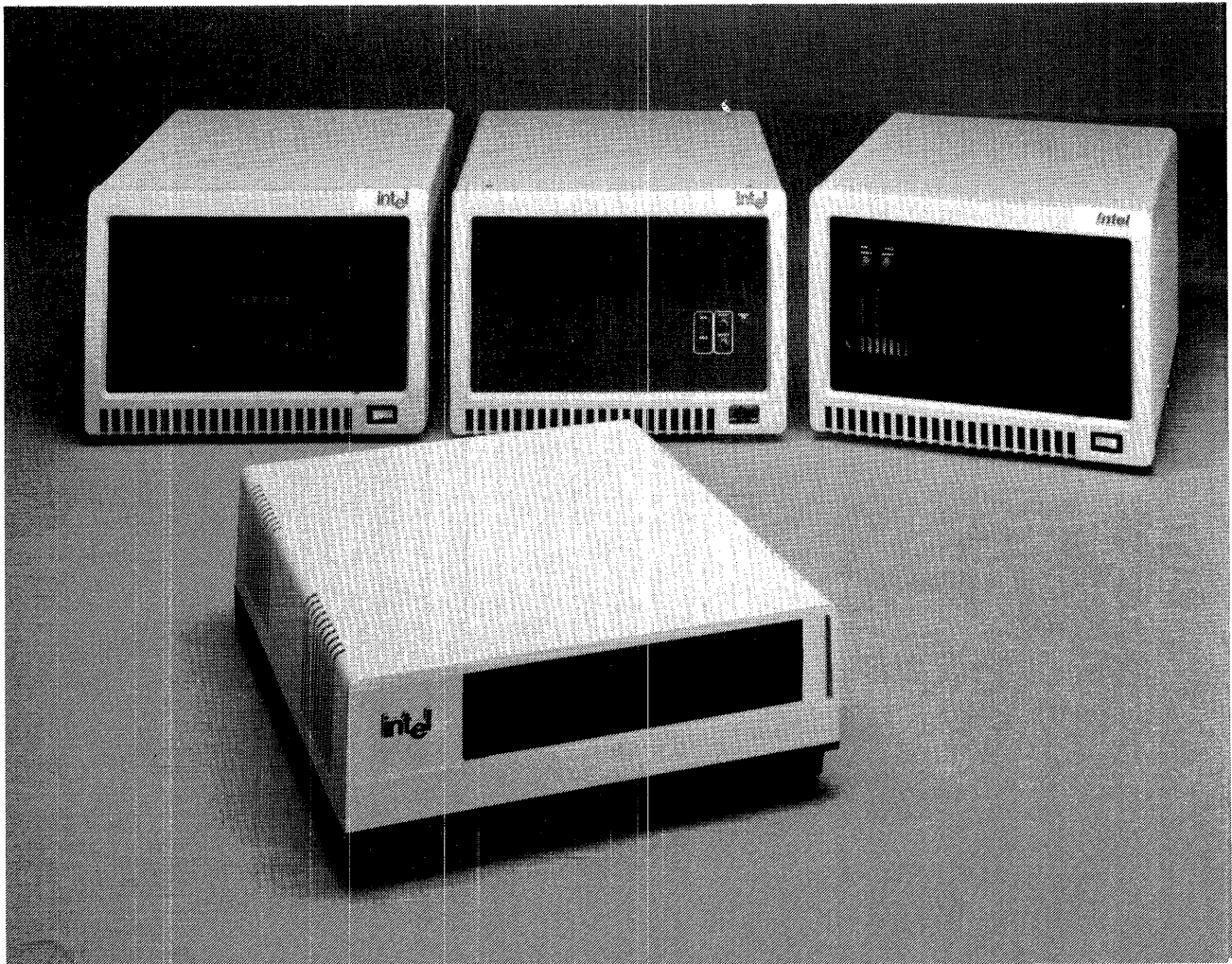




# SYSTEM 86/300 SERIES DIAGNOSTIC MAINTENANCE MANUAL



**SYSTEM 86/300 SERIES  
DIAGNOSTIC MAINTENANCE  
MANUAL**

Order Number: 144813-004

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 or to keep current the information contained in this document.

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

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

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

Intel Corporation makes no warranty for the use of its products and assumes no responsibility for any errors which may appear in this document nor does it make a commitment to update the information contained herein.

Intel retains the right to make changes to these specifications at any time, without notice.

Contact your local sales office to obtain the latest specifications before placing your order.

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

BITBUS	iLBX	iPDS	Plug-A-Bubble
COMMputer	i <sub>m</sub>	iRMX	PROMPT
CREDIT	iMMX	iSBC	Promware
Data Pipeline	Insite	iSBX	QUEX
Genius	int <sub>e</sub> l	iSDM	QUEST
Δ	int <sub>e</sub> lBOS	iSXM	Ripplemode
I	Intelelevision	Library Manager	RMX/80
i	int <sub>e</sub> l <sub>i</sub> gent Identifier	Megachassis	RUPI
I <sup>2</sup> ICE	int <sub>e</sub> l <sub>i</sub> gent Programming	MICROMAINFRAME	Seamless
ICE	Inteltec	MULTIBUS	SOLO
iCS	Intellink	MULTICHANNEL	SYSTEM 2000
iDBP	iOSP	MULTIMODULE	UPI
iDIS			

MDS is an ordering code only and is not used as a product name or trademark. MDS® is a registered trademark of Mohawk Data Sciences Corporation.

MULTIBUS is a patented Intel bus.

© 1984, Intel Corporation

REV.	REVISION HISTORY	DATE
-001	Original issue. Contains information formerly found in SYSTEM 86/300 SERIES DIAGNOSTIC REFERENCE MANUAL. This issue updates the SCT, adds the SDT8630 test, and updates the remaining SDT test suites.	06/82
-002	Adds the SDT534 and SDT309 test suites.	09/82
-003	Updates RSAT material.	07/83
-004	Deletes RSAT material; reorganizes SCT and SDT material.	05/84

**CAUTION**

This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause interference to radio communications. It has been tested and found to comply with the limits for Class A Computing Device pursuant to Subpart J of Part 15 of FCC rules, which are designed to provide reasonable protection against such interference when operated in a commercial environment. Operation of this equipment in a residential area is likely to cause interference, in which case the user, at his own expense, will be required to take whatever measures may be required to correct the interference.





This manual explains how to use the System Confidence Test (SCT) and System Diagnostic Test (SDT) programs available for Intel System 86/300 Series Microcomputer Systems. You can use these programs to determine whether the system is operating properly and as an aid in troubleshooting malfunctions in the system's hardware. The information in this manual is intended for service and engineering personnel who understand how the System 300 Series microcomputers work.

### WARNING

There are hazardous voltages present within 86/300 Series Microcomputer systems. To avoid electrical shock and damage to the system observe the following precautions:

1. Do not work on the system unless you are technically qualified to do so.
2. Disconnect the power cord before performing any type of maintenance or service.
3. Always refer to the appropriate reference manuals for the correct service procedures for your system.

Briefly, the subjects covered in each chapter are as follows:

- Chapter 1 describes the System Confidence Test in detail. It tells you how to invoke the SCT and how to interpret the results of its tests. It also includes troubleshooting hints.
- Chapter 2 introduces the System Diagnostic Tests (SDTs). It tells you how to invoke the tests and how use the Test Management commands to control test execution and error reporting.
- Chapter 3 describes the SDTs for the processor boards used in the System 86/300 Series microcomputers.
- Chapter 4 describes the SDTs for the memory boards used in the System 86/300 Series microcomputers.

- Chapter 5 describes the System Diagnostic Tests for mass storage components used in the System 86/300 Series microcomputers.
- Chapter 6 describes the System Diagnostic Tests available for the optional communications expansion boards used in System 86/300 Series microcomputers.
- Appendix A describes how to modify the System Diagnostic Tests to suit your particular system.

#### RELATED PUBLICATIONS

The following manuals contain information about the System 86/310 Microcomputer.

*System 310 Installation and Operation Guide*, Order Number 173211

*System 310 Hardware Integration Guide*, Order Number 173203

*System 310 Hardware Maintenance Manual*, Order Number 173208

The following manuals contain information about the System 86/380 and 86/330A Microcomputers.

*Introduction To The System 86/380 And System 86/330A Microcomputer Systems*, Order Number 172758

*System 86/330A Hardware Reference Manual*, Order Number 172759

*System 86/380 Hardware Reference Manual*, Order Number 172761

The following manuals contain information about software used on the System 86/300 Series microcomputers.

*iRMX™86 Release 5 Basic I/O System Reference Manual for System 86/300 Series Microcomputer Systems*, Order Number 172766

*iRMX86 Release 5 Extended I/O System Reference Manual for System 86/300 Series Microcomputer Systems*, Order Number 172767

The following manuals contain information about hardware components used in System 86/300 Series microcomputer systems.

*iSBC® 86/14 and iSBC 86/30 Single Board Computer Hardware Reference Manual*, Order Number 144044

*iSBC 86/12A Single Board Computer Hardware Reference Manual, Order Number 9803074*

*iSBC 337 MULTIMODULE™ Numeric Data Processor Hardware Reference Manual, Order Number 142887*

*iSBC 016A/032A/064A/028A/056A RAM Board Hardware Reference Manual, Order Number 143572*

*iSBC 308/309 Memory Management and Protection MULTIMODULE Board Hardware Reference Manual, Order Number 144686*

*iSBC 215G Winchester Disk Controller Hardware Reference Manual, Order Number 144780*

*iSBX™ 218A Flexible Diskette Controller Board Hardware Reference Manual, Order Number 145911*

*iSBX 351 Serial MULTIMODULE Board Hardware Reference Manual, Order Number 9803190*

*iSBC 534 Four-Channel Communications Expansion Board Hardware Reference Manual, Order Number 9800450*

*iSBC 544 Intelligent Communications Controller Board Hardware Reference Manual, Order Number 9800616*







## SERVICE INFORMATION

United States customers may obtain service and repair assistance by contacting the Intel Product Service Center in Phoenix, Arizona. Customers outside the United States should contact their sales source (Intel Sales Office or Authorized Distributor) for service information.

Before calling the Product Service Center you should have the following information:

- The date you received the product.
- The complete part number (including the dash number) of the product. This number is usually silk-screened onto printed circuit boards and stamped on the label of other products.
- The serial number of the product. This is usually silk-screened onto printed circuit boards and stamped on the label of other products.
- Your shipping and billing addresses.
- A purchase order number for billing purposes if your Intel product warranty has expired.
- Extended warranty agreement information, if applicable.

### SERVICE AND REPAIR ASSISTANCE

Use the following telephone numbers to contact the Intel Product Service Marketing Administration group:

Regional Telephone Numbers		TWX Numbers
Western Region	(602) 869-4951	910-951-1330
Midwestern Region	(602) 869-4392	910-951-0687
Eastern Region	(602) 869-4045	
International	(602) 869-4391	

Always contact the Intel Product Service Marketing Administration group before returning a product to Intel for repair. When you make the request you will be given a repair authorization number, shipping instructions, and other information that will help Intel provide you with fast, efficient service.

If you are returning a product because of damage sustained during shipment or if the product is out of warranty, a purchase order is required before Intel can initiate repair.

Use the original factory packaging material in preparing a product for shipment to the repair center. If that material is not available, ensure the product is adequately protected by wrapping it in cushioning material before enclosing it in a heavy-duty corrugated shipping carton. All cartons should be labeled "FRAGILE" to ensure careful handling. If a printed circuit board is being returned, a material such as Air Cap TH-240, manufactured by the Sealed Air Corporation of Hawthorne, New Jersey, should be used to give adequate cushioning.

Address and ship only to the address specified by Intel Product Service Marketing Administration group personnel.



<b>CHAPTER 1</b>	<b>PAGE</b>
<b>SYSTEM CONFIDENCE TEST</b>	
Running the SCT .....	1-1
Test Descriptions .....	1-4
USART/Timer Test .....	1-4
USART/Timer Errors .....	1-4
Programmable Interrupt Controller Test .....	1-5
PIC Errors .....	1-5
ROM Test .....	1-5
Programmable Peripheral Interface Test .....	1-6
PPI Errors .....	1-6
Memory Test .....	1-6
Memory Errors .....	1-7
Winchester Test .....	1-7
Winchester Errors .....	1-8
Flexible Diskette Test .....	1-9
Flexible Diskette Errors .....	1-9
Tape Test .....	1-9
Tape Errors .....	1-9
Additional Troubleshooting Hints .....	1-10
 <b>CHAPTER 2</b>	
<b>USING THE SYSTEM DIAGNOSTIC TESTS</b>	
System 86/300 Diagnostic Diskettes .....	2-2
Installing the SDT Test Suites on Winchester Disk .....	2-2
Loading an SDT into Memory .....	2-3
Test Management Commands .....	2-4
Command Notation and Usage .....	2-5
Syntax Diagrams .....	2-4
Abbreviations .....	2-5
Continuation Lines and Comments .....	2-6
Command Delimiters .....	2-6
Input Radices .....	2-7
Test Range Parameter .....	2-7
CLEAR Command .....	2-8
DEBUG Command .....	2-8
DESCRIBE Command .....	2-9
ERRONLY Command .....	2-10
EXIT Command .....	2-10
IGNORE Command .....	2-11
LIST Command .....	2-11
QUERY Command .....	2-12
RECOGNIZE Command .....	2-13
REPEAT and ENDREPEAT Commands .....	2-13
RESET Command .....	2-15
SUMMARY Command .....	2-15
TEST Command .....	2-17

	<b>PAGE</b>
V Command .....	2-19
Interrupting the Diagnostic Tests .....	2-20
 <b>CHAPTER 3</b>	
<b>PROCESSOR SYSTEM DIAGNOSTIC TESTS</b>	
SDT8630 Test Suite .....	3-1
SDT8630 Initialization .....	3-1
Using the RESET Command .....	3-3
SDT8630 Test 0, ROM Checksum .....	3-3
SDT8630 Test 1, Parallel Port .....	3-3
SDT8630 Test 2, PIC .....	3-3
SDT8630 Test 3, Timer .....	3-4
SDT8630 Test 4, Fixed Patterns .....	3-5
SDT8630 Test 5, Address March .....	3-5
SDT8630 Test 6, iSBC 86/30 Sliding Ones .....	3-5
SDT8630 Test 7, Dual Port RAM Contention .....	3-5
SDT8612 Test Suite .....	3-7
SDT8612 Test 0, ROM Checksum .....	3-8
SDT8612 Test 1, Parallel Port .....	3-8
SDT8612 Test 2, PIC .....	3-8
SDT8612 Test 3, Timer .....	3-9
SDT8612 Test 4, Fixed Patterns (iSBC 86/12A Board) .....	3-9
SDT8612 Test 5, Fixed Patterns (iSBC 300A Board) .....	3-10
SDT8612 Test 6, Address Patterns .....	3-10
SDT8612 Test 7, Sliding Ones .....	3-10
SDT8612 Test 8, Sliding Ones (iSBC 300A Board) .....	3-10
SDT8612 Test 9, Dual Port RAM Contention .....	3-10
SDT337 Test Suite .....	3-12
 <b>CHAPTER 4</b>	
<b>MEMORY SYSTEM DIAGNOSTIC TESTS</b>	
SDTRAM Test Suite .....	4-1
SDTRAM Initialization .....	4-2
Using the RESET Command .....	4-7
SDTRAM Test 0, Fixed Patterns .....	4-10
SDTRAM Test 1, Address March .....	4-10
SDTRAM Test 2, Sliding Ones Pattern .....	4-11
SDTRAM Test 3, Execute from RAM .....	4-11
SDTRAM Test 4, A-Series Parity Logic and RAMs .....	4-11
SDTRAM Test 5, A-Series Interrupt Detection .....	4-12
SDTRAM Test 6, C-Series Check Bit Logic .....	4-12
SDTRAM Test 7, C-Series Check Bit RAMs .....	4-12
SDTRAM Test 8, C-Series Error Correction .....	4-12
SDTRAM Test 9, C-Series Interrupt Detection .....	4-13
SDTRAM Test A, Rd/Wr Scope Loop Utility .....	4-13
SDT309 Test Suite .....	4-14
SDT309 Initialization .....	4-14
SDT309 Test 0, Interrupt Processing .....	4-16
SDT309 Test 1, Exception Conditions .....	4-17
SDT309 Test 2, Physical Address Verification .....	4-19

	<b>PAGE</b>
SDT309 Test 3, Memory Bounds .....	4-20
SDT309 Test 4, Memory Access .....	4-20
SDT309 Test 5, DMA I/O .....	4-20
 <b>CHAPTER 5</b>	
<b>MASS STORAGE SYSTEM DIAGNOSTIC TESTS</b>	
SDTWIN Test Suite .....	5-1
SDTWIN Initialization .....	5-2
Using the RESET Command .....	5-5
SDTWIN Test 0, Reset/Initialize Disk Test .....	5-5
SDTWIN Test 1, ROM Checksum Test .....	5-5
SDTWIN Test 2, RAM Window Test .....	5-5
SDTWIN Test 3, RAM Address Test .....	5-6
SDTWIN Test 4, Transfer Status .....	5-6
SDTWIN Test 5, Buffer I/O Test .....	5-6
SDTWIN Test 6, Format Diagnostic Track .....	5-6
SDTWIN Test 7, Microdiagnostic Test .....	5-7
SDTWIN Test 8, Verify Format/Format Test .....	5-7
SDTWIN Test 9, Seek/Verify Test .....	5-7
SDTWIN Test A, Worst-Case Seek Test .....	5-8
SDTWIN Test B, Read/Write/Verify Diagnostic Track ..	5-8
SDTWIN Test C, Drive Selection Test .....	5-8
SDTWIN Test D, Platter/Head Selection Test .....	5-8
SDTWIN Test E, Sector Selection Test .....	5-9
SDTWIN Test F, Overlap Seek Test .....	5-9
SDTWIN Test 10, Alternate Track Test .....	5-9
SDTWIN Test 11, Zero Fill Test .....	5-9
SDTWIN Test 12, Data Overrun Test .....	5-9
SDTWIN Test 13, Auto-Increment Test .....	5-10
SDTWIN Test 14, Write All/Read/Compare Test .....	5-10
SDTWIN Test 15, Random Write/Read/Verify Test .....	5-11
SDTWIN Test 16, Select Next Drive Under Test .....	5-11
SDTWIN Test 17, Format Utility .....	5-11
SDTWIN Test 1A, Unload Heads for Shutdown .....	5-12
SDTWIN Test 1B, Display/Edit Defective Track List Utility .....	5-12
SDTWIN Test 1C, Display/Clear Error Log Utility .....	5-12
Error Messages .....	5-12
Error Logs .....	5-14
SDT218 Test Suite .....	5-17
SDT218 Initialization .....	5-18
Using the RESET Command .....	5-20
SDT218 Test 0, Format Diagnostic Track .....	5-21
SDT218 Test 1, Seek/Verify Test .....	5-21
SDT218 Test 2, Write/Read Diagnostic Track .....	5-21
SDT218 Test 3, Drive Selection Test .....	5-21
SDT218 Test 4, Side/Head Selection .....	5-21
SDT218 Test 5, Sector Selection Test .....	5-22
SDT218 Test 6, Track Verify Test .....	5-22
SDT218 Test 7, Diskette Verify Test .....	5-22
SDT218 Test 8, Write and Read Deleted Data .....	5-22
SDT218 Utilities for the iSBC 215 Environment .....	5-22
SDT218 Utilities for the iSBC 86/30 Environment .....	5-27
SDT218 Messages .....	5-29

	<b>PAGE</b>
<b>CHAPTER 6</b>	
<b>COMMUNICATIONS SYSTEM DIAGNOSTIC TESTS</b>	
SDT351 Test Suite .....	6-1
SDT351 Initialization .....	6-1
Using the RESET Command .....	6-4
SDT351 Test 0, PIT Initialization .....	6-5
SDT351 Test 1, USART Initialization .....	6-6
SDT351 Test 2, USART Interrupts .....	6-8
SDT351 Test 3, Baud Rate Verification .....	6-9
SDT351 Test 4, Character Transmitter .....	6-10
SDT351 Test 5, Character Receiver .....	6-10
SDT534 Test Suite .....	6-11
SDT534 Initialization .....	6-11
Using the RESET SOFTWARE Command .....	6-13
Using the RESET HARDWARE Command .....	6-15
SDT534 Test 0, PIT Initialization .....	6-15
SDT534 Test 1, USART Initialization .....	6-16
SDT534 Test 2, PIC Initialization .....	6-17
SDT534 Test 3, PPI Port C Verification .....	6-17
SDT534 Test 4, USART Interrupt Verification .....	6-17
SDT534 Test 5, PIT Timer 4 and 5 Interrupt Verification .....	6-19
SDT534 Test 6, Baud Rate Verification .....	6-20
SDT534 Test 7, USART Load Test .....	6-20
SDT534 Test 8, USART Character Transmitter .....	6-20
SDT534 Test 9, USART Character Receiver .....	6-21
SDT544 Test Suite .....	6-22
SDT544 Initialization .....	6-22
Using the RESET SOFTWARE Command .....	6-25
Using the RESET HARDWARE Command .....	6-26
SDT544 Test 0, MULTIBUS RAM Test .....	6-27
SDT544 Test 1, iRMX86/XENIX 86 Firmware Jump-Out Command Test .....	6-28
SDT544 Test 2, Dual Port RAM .....	6-29
SDT544 Test 3, Board Interrupt Verification .....	6-29
SDT544 Test 4, iRMX 86/XENIX 86 Firmware Verification .....	6-30
SDT544 Test 5, PIT Initialization .....	6-31
SDT544 Test 6, USART Initialization .....	6-32
SDT544 Test 7, PIC Initialization .....	6-33
SDT544 Test 8, PPI Port A Verification .....	6-34
SDT544 Test 9, USART Interrupt Verification .....	6-34
SDT544 Test A, Baud Rate Verification .....	6-35
SDT544 Test B, Port Character Transmitter .....	6-35
SDT544 Test C, Port Character Receiver .....	6-36
 <b>APPENDIX A</b>	
<b>CONFIGURING THE SYSTEM DIAGNOSTIC TESTS</b>	
Making Changes to the Test Suites .....	A-1
 <b>INDEX</b>	

**TABLES**

<b>TABLE</b>	<b>TITLE</b>	<b>PAGE</b>
3-1	SDT8630 Tests .....	3-1
3-2	SDT8612 Tests .....	3-7
4-1	SDTRAM Tests .....	4-1
4-2	SDT309 Tests .....	4-14
5-1	SDTWIN Tests .....	5-1
5-2	SDT218 Tests .....	5-17
6-1	SDT351 Tests .....	6-1
6-2	USART Modes of Operation .....	6-6
6-3	SDT534 Tests .....	6-11
6-4	SDT544 Tests .....	6-22

**FIGURES**

<b>FIGURE</b>	<b>TITLE</b>	<b>PAGE</b>
1-1	System Confidence Test Display .....	1-3







The System Confidence Test (SCT) is a self-test program supplied as part of the system firmware in factory-standard System 86/300 Series microcomputers. The SCT tests the following major subsystems:

- Serial I/O port on the processor board
- Parallel I/O port on the processor board
- RAM on the processor and memory boards
- ROM on the processor board
- Priority interrupt controller on the processor board
- Winchester disk drive and controller
- Flexible diskette drives and controller
- Tape drive and controller

There are two versions of the SCT: one for systems with Winchester drives (SCT 86/300W firmware) and one for systems having only flexible drives (SCT 86/300F firmware). The two SCT versions are identical except that SCT 86/300F contains no tests for the Winchester or Tape systems.

#### **RUNNING THE SCT**

In order to run the SCT, the processor board must be at least partly functional, otherwise the tests cannot begin. The microprocessor must be running and able to access at least part of the on-board ROM and RAM, and the RS-232 port on the processor board (normally used for the system console) must be working well enough that the test results can be reported on the terminal.

To run the SCT, proceed as follows:

1. Set up a Video Display Terminal for 9600 baud operation (see note following step 2), with parity off, and connect it to the processor board's RS-232 port on the rear panel. Turn the terminal on and let it warm up.
2. Turn on the system. (If it is already on, press the Reset switch.) After a short delay, the system begins sending a series of asterisks to the terminal.

**NOTE**

Strictly speaking, the terminal does not have to be set up for 9600 baud, since the firmware contains an automatic baud rate determining feature that is activated when you enter an uppercase "U" just after power-up (step 3). However, if the terminal is not set to 9600 baud, no asterisks will be displayed on the terminal at power-up or reset.

3. Enter an uppercase U on the terminal's keyboard. The U initiates the SCT. (If you do not enter the U, the system eventually starts the SCT anyway, with slightly different results--this is discussed later in this section.)
4. Shortly after you enter the uppercase U, the system will display a message asking you for input from the keyboard. The display appears approximately as follows:

<u>TEST</u>	<u>STATUS</u>
USART/Timer	GO
PIC . *. INPUT "I"	

At this point you have four options:

- a. Do nothing. If you do not respond within six seconds, the PIC (programmable interrupt controller) test is reported "NO GO" and control of the system passes to the system monitor program. You can press the Reset switch to start testing again.
- b. Cause the SCT to continue by entering any character except CONTROL-C, CONTROL-D, or CONTROL-L. (CONTROL-D and CONTROL-L will cause the system to "hang up.")
- c. Abort testing and return control of the system to the system monitor by entering CONTROL-C.
- d. Select extended memory testing by entering the letter m (upper- or lowercase). For each error, the SCT displays the segment and offset address of the error, the expected data and the actual data read at that location, and the Exclusive-OR of the actual and expected data (showing the locations of the bits that were in error). See "Memory Tests" later in this Chapter.

As the SCT proceeds, the terminal display shows which part of the system is being tested and the status of that part. The display shown in Figure 1-1 is typical of one produced by a Winchester-disk-based

---

```

SCT 86/300W, V1.2          Copyright 1982, 1983 Intel Corporation
TEST                      STATUS
USART/ Timer              GO
PIC                       *. Input "I" .      GO
ROMCKSM                   GO
PPI                       GO
  Processor Subsystem    GO

Onboard                   GO
Offboard                  GO
  Memory Subsystem      Total Memory = 768K    GO

Winchester                ...                GO
Floppy                    .                  GO
Tape                      .                  OFFLINE
  Boot Subsystem        GO
SCT Successful...Now Booting System

```

Figure 1-1. System Confidence Test Display

---

system. For testing purposes, the system is divided into three subsystems: processor, memory, and boot (disk) subsystems. The subsystems are further divided into component parts, such as the priority interrupt controller or the parallel port in the processor subsystem. As each part passes a test, the SCT displays a period (.) just to the right of the part's name; a failed test is indicated by a question mark. If the part passes all tests, its status is GO; any failed test produces a NO GO status. A diskette or tape drive may be reported OFFLINE if no diskette is installed or the drive is disconnected.

Shown below is an abbreviated SCT display that occurs when you do not respond at initialization by entering the uppercase "U." This display indicates only the status of the major subsystems.

```

SCT 86/300W, V1.2          Copyright 1982, 1983 Intel Corporation

  Processor Subsystem    GO
  Memory Subsystem      Total Memory = 768K    GO
  Boot Subsystem        GO

SCT Successful...Now Booting System

```

Notice that (if you have a diskette installed) as the SCT accesses the flexible diskette drive during testing, the head load indicator on the front of the drive goes on and you can hear the head mechanism as it moves from track to track. This can be a useful indication of

processor and disk subsystem activity if the RS-232 port is not working, or if there is no terminal connected to the port.

You can abort the SCT at any time by entering CONTROL-C on the keyboard or by pressing the Interrupt switch on the front panel. These actions return control of system to the system monitor.

If the tests run without errors, control transfers to the Bootstrap Loader. If the SCT detects an error, control passes to the system monitor at completion of the test.

In situations where you need to reload the system software frequently you can bypass the SCT (and save the contents of RAM) by pressing the Interrupt switch within two seconds after pressing the Reset switch. You may find this useful during system software debugging.

### **TEST DESCRIPTIONS**

The following sections describe the individual SCT tests in detail. For each test there is an operational description followed by an interpretation of the test's error messages.

#### **USART/TIMER TEST**

The USART/TIMER test checks the 8251A Universal Synchronous/Asynchronous Receiver-Transmitter (USART) and the 8253 Programmable Interval Timer (PIT) on the processor board. First the USART and PIT are initialized. Next the USART is tested by transmitting a character (the first asterisk that appears on the display screen after power-up or reset), then reading the USART's status register. If the TxReady bit in the status register indicates that the USART was able to shift the character out of its transmit buffer, the USART and PIT are considered to have passed their tests. This does not guarantee that the terminal connected to the RS-232 port received the character.

#### **USART/Timer Errors**

If the HALT and RUN indicators flash alternately (System 86/330 and 86/380) or the the RUN indicator flashes on and off (System 86/310), replace the processor board. If no asterisks appear on the terminal and the HALT and RUN indicators do not flash, take the following corrective actions:

1. Check that the terminal is operating properly and that it is set to 9600 baud, no parity.
2. Check that the RS-232 cable is securely connected.
3. Check the voltage on the +12 and -12 volt power supplies.
4. Replace the processor board.

**PROGRAMMABLE INTERRUPT CONTROLLER TEST**

The Programmable Interrupt Controller Test checks the 8259A Programmable Interrupt Controller (PIC) on the processor board. First the test program initializes the PIC and reads the mask register within the PIC. The test program then tests the PIC using interrupt requests from three sources: the timer, the USART's transmitter, and the USART's receiver. If any of these interrupts does not occur, an error condition is reported on the terminal. In testing the USART receiver interrupt, the test program prompts you to enter a character on the terminal's keyboard (as explained earlier under "Running the SCT"). If the receiver interrupt request does not come within six seconds, the test program indicates an error condition and goes on to subsequent tests.

**Priority Interrupt Controller Errors**

Errors found during the PIC test are displayed on the terminal's screen as follows:

PIC	?	*?	INPUT "I" i?	NO GO
	(1)	(2)	(3)	(4)

1. A question mark at position 1 indicates that the timer interrupt did not occur. To correct this malfunction, replace the processor board.
2. A question mark at position 2 indicates that the USART transmit interrupt did not occur. The "\*" displayed just in front of the question mark is the character sent by the USART to check the transmit interrupt. To correct this malfunction replace the processor board.
3. A question mark at position 3 indicates that the USART receive interrupt did not occur. The "i" just in front of the question mark represents the character entered on the keyboard in response to the INPUT "I" prompt. If the interrupt does not occur within six seconds, the question mark is displayed and the test program goes on to other tests. To correct this malfunction (if you entered a character and still got an error) replace the processor board.
4. The "NO GO" message is displayed when any interrupt does not occur.

**ROM TEST**

The ROM test verifies the contents of the ROM on the processor board and checks that the address and data lines to the ROM are operating properly.

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

#### **PROGRAMMABLE PERIPHERAL INTERFACE TEST**

The Programmable Peripheral Interface Test checks the 8255A Programmable Peripheral Interface (PPI) that drives the parallel port on the processor board. The test first initializes the PPI and sends a value to each of the PPI's three ports. The test then reads each port and the data read is compared with the data sent.

#### **NOTE**

If your system is connected to an Intel Micro-computer Development System via the parallel port, the SCT will report the PPI "OFFLINE" and will go on to subsequent tests.

#### **PPI Errors**

Errors found during the PPI test are reported on the terminal's display screen as follows:

PPI	?	?	?	NO GO
	(1)	(2)	(3)	(4)

1. A question mark at position 1 indicates a failure at port A. Replace the processor board to correct this malfunction.
2. A question mark at position 2 indicates a failure at port B. Replace the processor board to correct the malfunction.
3. A question mark at position 3 indicates a failure at port C. Replace the processor board to correct this malfunction.
4. A failure at any of the ports results in a "NO GO" status.

#### **MEMORY TEST**

The Memory Test checks the system's Random Access Memory (RAM), both on and off the processor board. For testing purposes, the RAM is divided into two areas: on-board RAM, which consists of all RAM on the processor board; off-board RAM, which consists of the RAM normally contained on separate memory boards in factory-standard versions of the system.

In two successive passes, the memory test writes a checkerboard pattern of ones and zeros (A55AH on one pass; 5AA5H on the next) into RAM. During each pass the test reads the RAM to verify that the data read matches the data written and that the data bytes are at the proper addresses. This verifies that the RAM devices, data paths, and address lines are functioning properly. Memory locations from 0000:0000H through C000:000FH (768K) are tested, provided that all blocks of memory are contiguous (no gaps beginning at 16K boundaries).

Extended testing, which goes beyond the checkerboard pattern test and provides more detailed error reporting, may be selected by entering "m" during the PIC test.

### Memory Errors

If any errors are found during testing, a NO GO message appears in the status column for the area (on-board or off-board) in which the error occurred. If a failure is reported in on-board memory, replace the processor board; otherwise replace the appropriate memory board.

Errors found during extended testing (selected by entering "m" during the PIC test) are reported on the terminal as follows:

**Error at    ssss:aaaa    RECrrrr    EXPeeee    XORxxxxxxxxxxxxxxxx**

where

**sss**            is the segment portion of the address where the error occurred.

**aaaa**            is the offset portion of the address where the error occurred.

**rrrr**            is the data actually read at the address where the error occurred.

**eeee**            is the data that the test program expected to find at the address where the error occurred.

**x...**            is a 16-bit binary value representing the Exclusive-OR of **eeee** and **rrrr**, showing the locations of the bits that were in error.

### WINCHESTER TEST

The Winchester Test checks the iSBC 215G<sup>®</sup> Winchester Disk Controller and the Winchester disk drive. The Winchester test has three subtests; each subtest must be passed before the next one can start. The subtests are as follows:

- Winchester controller and drive initialization



- Winchester controller on-board diagnostics invocation
- Winchester controller interrupt generation

### Winchester Errors

Errors found during the Winchester test are reported on the terminal as follows:

<b>Winchester</b>	?	?	?		<b>NO GO</b>
	(1)	(2)	(3)		(4)

1. A question mark in position 1 indicates that the Winchester controller and drive could not be initialized.

Since the processor communicates with the Winchester controller by means of "control blocks" in memory, a memory failure can cause a Winchester initialization failure as well. If a memory error is reported, replace the appropriate memory board before replacing any Winchester component.

If no memory error is reported, take the following steps, checking the system after each step.

- a. In 8-inch drive systems, ensure that the head is unlocked. Check all interconnecting cables.
- b. Make sure that the disk is properly formatted, using a program such as DISKVERIFY, which is supplied with the iRMX 86 operating system. If a format problem is found, reformat the disk.

### CAUTION

Reformatting destroys all data on the disk.  
Make a back-up copy of the files on the disk  
before reformatting.

- c. Replace the Winchester controller (after checking that jumpers are properly set).
  - d. Replace the Winchester drive.
2. A question mark at position 2 indicates that an error was found by the Winchester controller's diagnostics. Corrective action is the same as in steps a through d for the initialization error.
  3. A question mark at position 3 indicates that a Winchester controller interrupt did not occur. If the PIC also failed its test, replace the processor board; otherwise, replace the Winchester controller board.







## CHAPTER 2 USING THE SYSTEM DIAGNOSTIC TESTS

The System Diagnostic Tests (SDTs) are collections of tests (test suites) that can be used to determine the cause of malfunctions in System 86/300 Series Microcomputers. SDTs are available for most standard and optional circuit boards used in the 86/300 Series systems. Once an SDT has been loaded into memory, you can use its tests individually or in groups, and you can choose detailed error reporting or simple pass/fail error messages.

The SDTs and their functions are as follows:

- SDT8630 tests the CPU, USART, ROM, RAM, PPI (programmable peripheral interface), PIT (programmable interval timer), PIC (programmable interrupt controller), and associated logic circuitry on the 86/30 processor board.
- SDT8612 tests the CPU, USART, ROM, RAM, PPI, PIT, PIC and associated logic circuitry on the iSBC 86/12A processor board.
- SDT337 tests the 8087 Numeric Processor Extension on the processor board.
- SDTRAM tests memory board RAM devices, address and data lines, and parity circuitry.
- SDT309 tests the memory mapping and protection features of the iSBC 309 Memory Management board.
- SDTWIN tests the Winchester disk drives and the 8089 I/O Processor, ROM, RAM, and other circuitry on the iSBC 215B and iSBC 215G Winchester Disk Controller boards. It also verifies the circuitry on iSBC 220 controllers and SMD drives (currently not supported in the System 310).
- SDT218 tests the iSBX 218 (and 218A) Flexible Disk Controller Board and flexible diskette drives.
- SDT351 tests the USART and PIT on the iSBX 351 Serial Communications Expansion Board.
- SDT534 tests the USARTs, PITs, and PICs on the iSBC 534 Four Channel Communications Expansion Board.
- SDT544 tests the USARTs, PPI, PITs, and PICs on the iSBC 544 Intelligent Communications Controller Board.

**SYSTEM 86/300 DIAGNOSTIC DISKETTES**

Intel distributes the System 86/300 diagnostic software on two 8" or five 5¼" flexible diskettes. The diskettes are labeled as follows:

SYP 86/300 DIAGNOSTICS, # *n*

DS/DD iRMX 86 DISKETTE  
PART NO. xxxxxx-yyy  
©1982, 1983 INTEL CORP.

where *n* is the diskette number. The contents of the diskettes are as follows:

- 8" diskette #1 contains the executable SDT program files. Use this diskette when loading the SDTs.
- 8" diskette #2 contains configuration files for the SDTs. Use these files to change the default values that the SDTs use at initialization for test limits and other variables. See Appendix A for additional information.
- 5¼" diskettes #1A and #1B contain the same SDT program files as 8" diskette #1.
- 5¼" diskettes #2A, #2B, and #2C contain the same configuration files as 8" diskette #2.

**INSTALLING SDT TEST SUITES ON WINCHESTER DISK**

The SDT test suites reside on the diagnostic diskettes shipped with your system. If you are using the iRMX 86 operating system, you should, upon receipt of the diagnostic diskettes, install the SDT tests on the Winchester disk. That way, if you have a problem with either the Winchester or flexible diskette drives, you will have an alternate means of invoking the tests.

If you are using the XENIX\* 86 operating system, you must invoke the diagnostics from a flexible diskette. In this case, the following discussion does not apply.

The following procedure assumes that you are familiar with the iRMX 86 operating system. Refer to the *iRMX 86 Operator's Manual* for detailed information.

Install the SDTs on the Winchester disk as follows:

1. After loading the iRMX 86 operating system from your Winchester disk, invoke the SUPER command. When the prompt

**ENTER PASSWORD:**

\*XENIX is a trademark of Microsoft Corporation.

appears, enter the password associated with the system manager.

2. Create a Winchester drive directory for the SDTs by entering

```
CREATDIR :SD: SDTDIR
```

3. Attach the flexible disk device using the ATTACHDEVICE command:

```
ATTACHDEVICE www AS :FD0:
```

where *www* is the physical device name of the drive to be used. Use the physical device name WFDD0 for an 8" double-sided double-density flexible diskette drive when the flexible diskette controller board is mounted on the iSBC 215G Winchester Controller board. (This is the factory-standard configuration for systems that have a Winchester drive installed.) Use the physical device name WMFDX0 for a 5¼" flexible diskette drive.

4. Insert 8" diagnostic diskette 1 or 5¼" diagnostic diskette 1A into the flexible diskette drive.
5. Enter

```
COPY :FD0:SDTDIR/SDTnnnn
    TO :SD:SDTDIR/SDTnnnn
```

where *nnnn* is the number of an SDT. Repeat this step for each SDT on 8" diskette 1 or 5¼" diskettes 1A and 1B. (See the list of SDTs at the beginning of this chapter.)

#### LOADING AN SDT INTO MEMORY

To run tests in an SDT test suite, you must first load the SDT into memory. Proceed as follows.

1. Press the Reset switch on the system's front panel.
2. After you press the Reset switch, the system begins to send a series of asterisks to the terminal's screen. As soon as the asterisks appear, enter an uppercase U on the terminal.
3. Shortly after you enter the uppercase U, the system displays a message asking you for input from the keyboard, as follows:

<u>TEST</u>	<u>STATUS</u>
USART/Timer	GO
PIC	. *. INPUT "I"

When this message appears, enter CONTROL-C on the terminal or press the Interrupt switch on the system front panel. This passes control to the system monitor program.

4. If the SDT test suites are contained on the Winchester disk boot the desired test by entering

*b /SDTDIR/testnumber*

where *testnumber* is one of the following SDT test suite identifiers: SDT8630, SDT8612, SDT337, SDTRAM, SDTWIN, SDT218, SDT351, SDT534, SDT544, or SDT309.

You must leave a space between the "b" and the first "/" in the invocation line. You may use upper- or lowercase characters.

To invoke the tests from a flexible diskette, enter

*b :WF0:/SDTDIR/testnumber*

if the flexible diskette controller is mounted on the iSBC 215G Winchester Controller board (the standard location in Winchester-disk-based systems), or enter

*b :PF0:/SDTDIR/testnumber*

if the flexible disk controller is mounted on the iSBC 86/30 Single Board Computer.

After invoking a particular SDT test suite, you can run all of the tests in the suite, run a selected series of tests, loop on a particular test or series of tests, or run a single test by using the test management commands described in the next section.

#### TEST MANAGEMENT COMMANDS

The test management commands, available with all of the SDT test suites, allow you to select specific tests, control the way in which the tests run, and control the extent and manner of error reporting. The commands are as follows:

- 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 from a test to the system monitor.

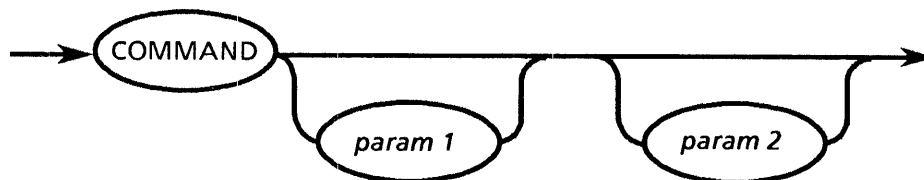
- **IGNORE** causes the **TEST** and **SUMMARY** commands to ignore selected tests.
- **LIST** copies all console I/O to a file.
- **QUERY** displays the status of **DEBUG** and **ERRONLY**.
- **RECOGNIZE** reverses the effect of an **IGNORE**.
- **REPEAT/ENDREPEAT** begins and ends a loop of commands.
- **RESET** resets the hardware and/or software.
- **SUMMARY** prints a result log of the tests that have run.
- **TEST** selects and runs one or more SDT tests.
- **V** displays and sets the the values of global variables used by the tests.

#### COMMAND NOTATION AND USAGE

This section explains the notational conventions and the various rules of use that apply to the Test Management Commands.

#### Syntax Diagrams

The Test Management Command descriptions use syntax diagrams to illustrate the various ways commands may be entered. The tracks on the diagrams indicate the possible paths through the various combinations of a command's elements. For example, the following diagram illustrates a command with two optional parameters.



F-0126

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

```

COMMAND
COMMAND param1
COMMAND param2
COMMAND param1 param2

```

To use the diagram, start at the left-hand side and follow the tracks through all those elements that you want to use. The arrows on the tracks indicate the proper direction of travel. When leaving a track, you must continue in the direction of the smooth curve; sharp turns and going against the arrows are not allowed.



The following notational conventions are used for elements within the syntax diagrams:

- Elements shown in uppercase letters are literal quantities and must be entered using exactly the characters shown. They may, however, be entered in upper- or lowercase characters. In addition, certain command names and keywords may be abbreviated, as described later in this section.
- Elements shown in lowercase italics are variable quantities. You must enter the appropriate value or symbol, as specified in the test descriptions.

### Abbreviations

When you enter commands, you can abbreviate command names, and keywords used with the commands, to their first three characters. For example, you can abbreviate

```
TEST 1 REPEAT UNTIL ERROR
```

as follows:

```
TES 1 REP UNT ERR
```

### Continuation Lines and Comments

You can continue a command on the next line by using an 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.

You can also insert a comment by entering /\*.

For example, the command

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

is a continued command with comments inserted using both methods. 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 enter more than one command on a single line. To do this, you must delimit the commands with a semicolon (;). For example, the SDT runs the following commands as if you entered them on separate lines.

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

**Input Radices**

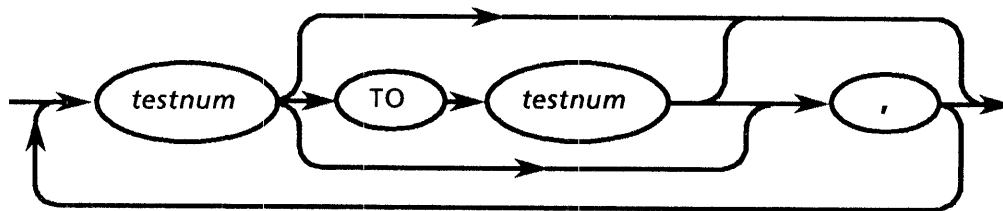
The SDT always displays numerical output in hexadecimal form. However, when you provide input to the SDT, you can specify the radix of numerical quantities by including a radix character immediately after the number. The valid radix characters include:

<u>radix</u>	<u>character</u>	<u>example</u>
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 Parameter**

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 the range of test numbers on which the command is to operate. The syntax for *test-range* is as follows:



F-0127

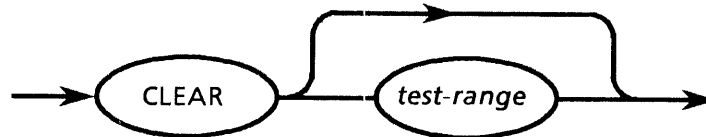
To select individual tests, separate the test numbers with commas. For example, 1,3 selects tests 1 and 3. If you separate two test numbers with the word TO, you also select all tests between the first number and the second number (for example, 1 TO 3 selects tests 1, 2, and 3); however, when you separate 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.

**CLEAR COMMAND**

Each test has an execution count and an error count, showing how many times the test has run and how many errors occurred. For each test in the test range (except those specified in an IGNORE command), CLEAR sets the execution count and error count to zero.

Command syntax is as follows:



F-0128

where *test-range* is the 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 to zero.

Examples:

```
*CLEAR
*
```

Clears the execution and error count values for all tests in the SDT test suite.

```
*CLE 7
*
```

Clears the execution and error count values for test 7.

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

Clears the execution and error count values for tests 1, 2, 3, 4, and 8.

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 test suite.

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

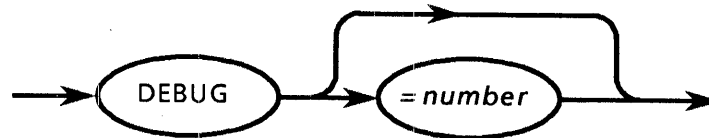
When specifying test numbers using "a TO b," a must be less than b.

**DEBUG COMMAND**

During execution, a test can return debug messages. A debug message lists detailed information about test failures that is useful if you are writing a diagnostic test or troubleshooting the hardware. The

DEBUG command controls whether or not the SDT displays debug messages at the console during test execution.

Command syntax is as follows:



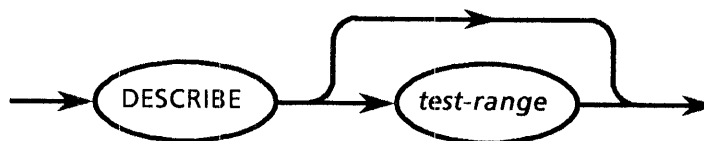
F-0129

where *number* is a value that determines the debug status. An even value (least-significant bit set to 0) sets the debug status to FALSE (off). An odd value (least-significant bit set to 1) sets the debug status to TRUE (on). If you omit the number parameter, the SDT displays the current value of DEBUG. The default value is FALSE.

If you set DEBUG to TRUE and ERRONLY to FALSE (refer to the description of the ERRONLY command), the SDT displays the name and number of each test before running the test; it also displays the usual debug messages 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 specified tests. Command syntax is as follows:



F-0130

where *test-range* is the 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.

Example:

```
*DES 1 TO 5,7
```

lists the test names and numbers for tests 1, 2, 3, 4, 5, and 7. If you are running the SDT215 test suite, the display looks like this:

0001H	TRANSFER STATUS
0002H	BUFFER I/O TEST
0003H	ROM CHECKSUM TEST
0004H	RAM WINDOW TEST
0005H	RAM ADDRESS TEST
0007H	SEEK/VERIFY TEST
*	

Error messages:

**ERROR: test out of range**

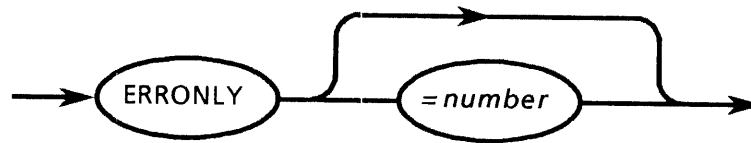
One or more of the test numbers specified was larger than the largest test number in the SDT test suite.

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

When specifying test numbers using "a TO b," a must be less than b.

**ERRONLY COMMAND**

The ERRONLY command controls whether or not the SDT displays status messages on the terminal during test execution. Command syntax is as follows:



F-0131

where *number* is a value that 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 *number*, the SDT displays the current value of ERRONLY. The default value is FALSE.

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.) If you set ERRONLY to FALSE, the SDT displays test results for both passing and failing tests.

**EXIT COMMAND**

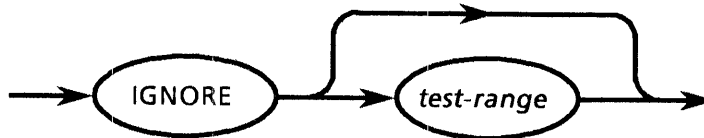
The EXIT command causes an exit from the SDT and returns control to the system monitor. This allows you to load other programs, such as the other SDT test suites or the iRMX 86 Operating System. Command syntax is as follows:



F-0132

**IGNORE COMMAND**

The IGNORE command causes specified tests to be bypassed when you invoke the TEST command. After the IGNORE command is invoked for a test, it remains ignored until you issue a RECOGNIZE command for that test. Command syntax is as follows:



F-0134

where *test-range* is the range of test numbers that are ignored during a TEST command. If you omit the test range, all tests in the test suite are ignored. If you specify the number of a test that has already been ignored, the IGNORE command has no effect on that test.

**Error Messages:****ERROR: test out of range**

One or more of the test numbers 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 using "a TO b," a must be less than b.

**Examples:**

```
*IGNORE
*
```

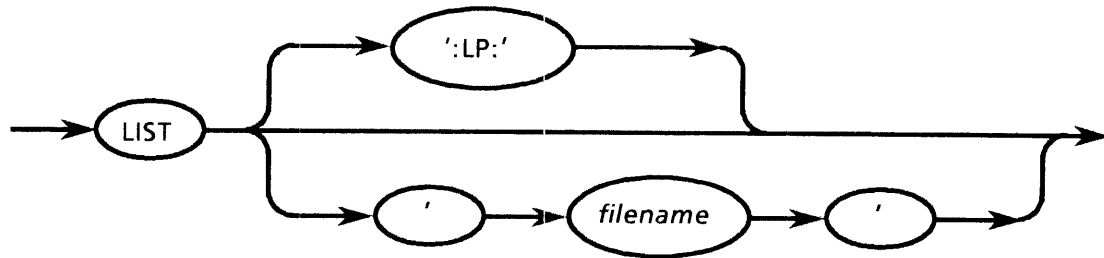
ignores all tests in the SDT test suite. Future TEST commands have no effect until you recognize the tests using the RECOGNIZE command.

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

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

**LIST COMMAND**

The LIST command causes the SDT to copy all terminal I/O to a specified file or device. Command syntax is as follows:



F-0135

where

**' :LP: '** represents the line printer. You must enclose **:LP:** in single quote characters. If you use this parameter, LIST copies terminal I/O to the line printer connected to your system.

#### NOTE

If there is no line printer connected to your system, specifying **' :LP: '** causes the system to malfunction, requiring you to reset the system and reload the SDT.

**'filename'** is an ISIS-II filename. You must enclose the filename in single quote characters.

You can specify an ISIS-II file to receive terminal I/O only if your system is connected to an Intel Microprocessor Development System through the processor board's parallel interface.

If you use 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 and the Microprocessor Development System. To recover from this error, you must reload the SDT software.

#### QUERY COMMAND

The QUERY command displays the current logical values of the DEBUG and ERRONLY command variables. Command syntax is as follows:



F-0136

Examples:

\*QUERY

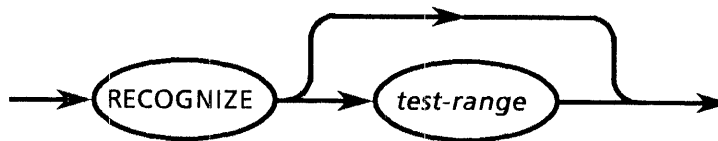
Produces the following display:

```

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 run when the TEST command is invoked. Command syntax is as follows:



F-0137

where *test-range* is the 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 has already been recognized, the RECOGNIZE command has no effect on that test.

Error Messages:

**ERROR: test out of range**

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

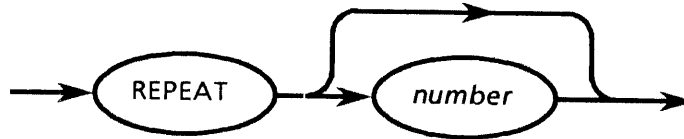
**ERROR: in "a TO b", b is less than a**

When specifying test numbers using "a TO b," a must be less than b.

### REPEAT AND 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. Command syntax is as follows:





F-0138



F-0139

where *number* is the number of times the group of commands is to be repeated. If you omit the number parameter, the delimited commands are repeated indefinitely, until you enter CONTROL-C or reset the system.

The REPEAT and ENDREPEAT commands provide a mechanism for creating command loops. When you enter the REPEAT command, the SDT issues the following prompt:

.\*

to remind you that succeeding commands are part of a REPEAT loop. The SDT does not execute any of the commands until you enter the ENDREPEAT command to end the loop.

You can nest REPEAT loops by entering another REPEAT command in response to the prompt. If you do this, the SDT issues the following prompt:

..\*

indicating that succeeding commands are part of the nested loop. You can end the nested loop by entering an ENDREPEAT command. 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 adds an additional period to its prompt. The SDT issues an error message if you attempt to nest more than eight levels of REPEAT loops.

Examples:

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

The preceding sequence repeats tests 0, 1, and 3 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

```

The preceding sequence is a nested repeat loop in which the SDT runs test 0 followed by five iterations of test 9. It continues this testing sequence until you enter a CONTROL-C.

Error Messages:

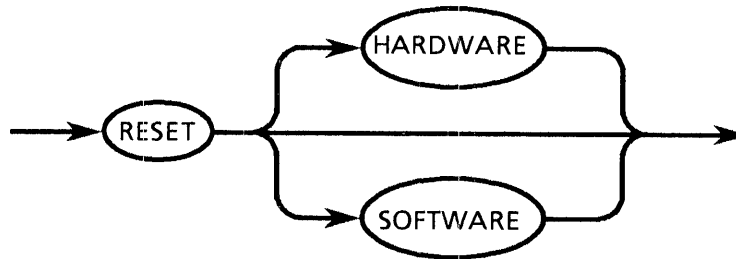
**ERROR: too many nested IFs or REPEATs**

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

**RESET COMMAND**

The RESET command initializes the test software and the hardware that it is testing. The RESET command does not reset the values of the DEBUG and ERRONLY commands, nor does it specify that ignored tests are to be recognized, in all cases. After using the RESET command, you should examine the individual SDT test suites to determine the state of ignored tests.

Command syntax is as follows:



where

F-0140

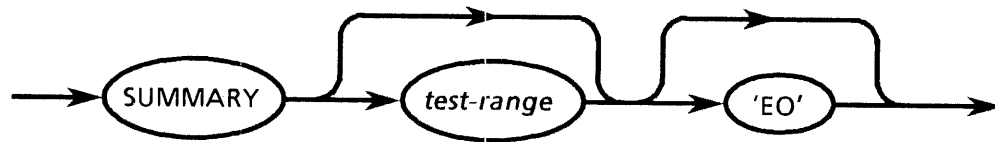
**HARDWARE** resets the hardware to its initial state. The SDT test suite may request additional information from you about the state of the hardware.

**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 specified tests. Command syntax is as follows:



F-0141

where

*test-range* is 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 tests in the SDT test suite.

'EO' displays the "error only" information. If you specify this parameter, SUMMARY displays information only for failed tests in the test range. Otherwise, SUMMARY displays information for all tests in the test range.

The SUMMARY command displays the name and number of each test in the test range, followed by the number of test tries and the number of failures (both in hexadecimal). You can reset the number of tries and failures to zero by entering the CLEAR command.

Information about the number of tries and failures does not appear for any tests specified in the IGNORE command. Instead, the following message appears:

**\*\*\* IGNORED \*\*\***

Example:

```

*SUMMARY

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

This command displayed a log of test results for all tests in the current test suite, which in this case is the SDTRAM test suite. IGNORE status is set for the last test; that test does not run and SUMMARY does not display information for it.

Error Messages:

**ERROR: test out of range**

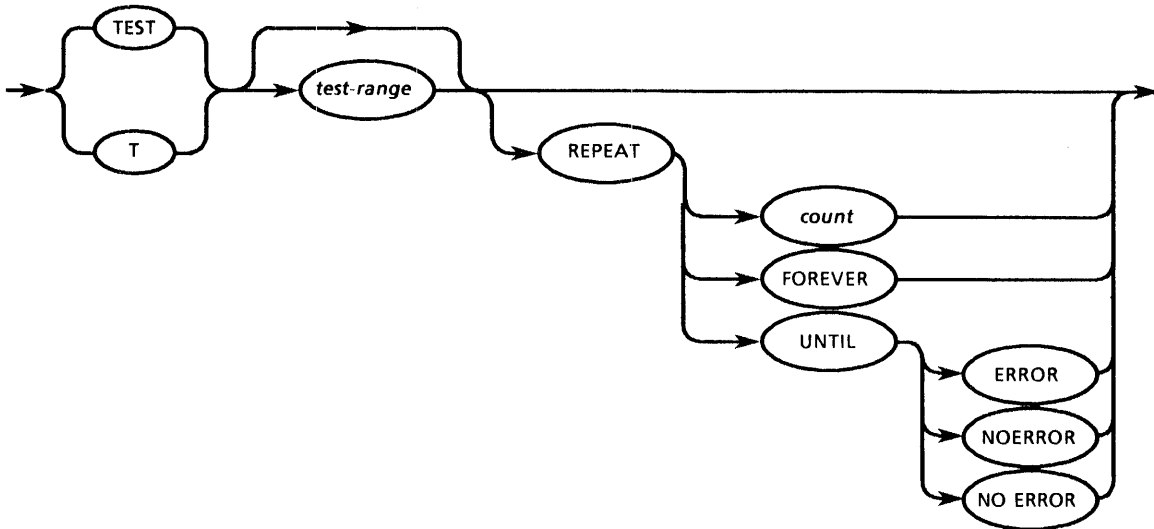
One or more of the test numbers you specified was larger than the largest test number in the SDT test suite.

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

When specifying test numbers using "a TO b," a must be less than b.

**TEST COMMAND**

The TEST command selects and runs diagnostic tests. Command syntax is as follows:



F-0142

where

*test-range*

is the range of tests to run. The SDT runs all tests in the test range except those you have specified in an IGNORE command. If you omit this parameter, the SDT assumes the test range to be all tests in the test suite.

REPEAT

means repeat 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*

is the number of times to repeat the tests.

FOREVER

means repeat the tests indefinitely.

UNTIL ERROR

means repeat the tests until an error occurs.

UNTIL NO ERROR

means repeat the tests until no errors occur.

The TEST command runs all tests in the test suite except those specified in an IGNORE command. See the IGNORE command description 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 commands. (See the descriptions of the DEBUG and ERRONLY commands.)

Normally, the SDT displays the number, name, and result (passed or failed) of each test. If a test has "ignored" status, 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. In addition, detailed debug messages that individual tests generate to describe error conditions are displayed. If you set DEBUG to an even value (FALSE), the SDT omits the pre-test message and the detailed debug messages.

If you set the ERRONLY variable to an odd value (TRUE), the SDT displays results only for failed tests. 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. If you set ERRONLY to an even value (FALSE), the SDT displays test results for both passing and failing tests.

#### NOTE

If you set ERRONLY to TRUE and issue the Test Command using the REPEAT, REPEAT FOREVER, or REPEAT UNTIL ERROR parameters, you may find it impossible to stop test execution with CONTROL-C. If all tests run without errors, ERRONLY causes the SDT to omit messages to the terminal. Since the SDT can respond to a CONTROL-C only during terminal I/O, any CONTROL-C you enter will be ineffective. In this case, you must press the INTERRUPT switch to stop test execution. You must then reload the SDT test suite to run further tests.

Error Messages:

#### **ERROR: test out of range**

One or more of the test numbers 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 using "a TO b," a must be less than b.

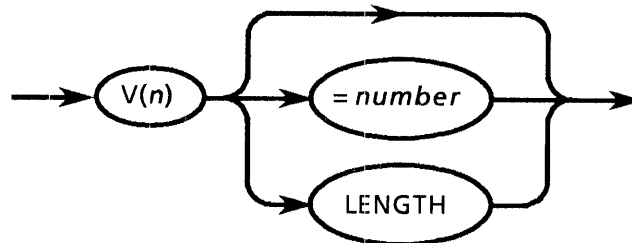
Example:

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

This command ran each of the tests in the suite once, except for test 3, which has "ignored" status. This example assumes the SDTRAM test suite is the current test suite.

#### V COMMAND

The V command displays or sets values for global variables used by the tests. Command syntax is as follows:



F-0143

where

*n* is the variable's number, in the range 0 through 0FH. Not all Intel-supplied diagnostic tests make use of global variables. You can use these variables if you write your own diagnostic tests.

*number* is the value to which the global variable is set. Variables can be set to any 16-bit value. If you omit value, the SDT displays the current value of the global variable.

*LENGTH* is the length of the list of variables whose current value you want to display.

Examples:

```
*V(B)=100
```

assigns the hexadecimal value 100 to variable V(B).

\*V(C)

displays the current value of the variable V(C).

\*V(0) len 6

displays the current values of variables V(0) through V(5).

Error Messages:

**ERROR: "V" variable out of bounds**

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

### INTERRUPTING THE DIAGNOSTIC TESTS

To interrupt a test that is running and regain control of the system, enter CONTROL-C on the terminal. CONTROL-C causes the SDT to abort the current test or command and return control to you at the console.

The SDT can respond to CONTROL-C only when it is performing console I/O. Therefore, there may be a delay in obtaining control from a diagnostic test that performs console I/O infrequently.



**CHAPTER 3  
PROCESSOR SYSTEM DIAGNOSTIC TESTS**

This chapter contains detailed descriptions of the System Diagnostic Tests for the processor boards and related modules used in System 86/300 Series microcomputers. Included are descriptions for the following tests:

- SDT8630, which tests the iSBC 86/30 Single Board Computer
- SDT8612, which tests the iSBC 86/12A Single Board Computer and the iSBC 300A RAM Expansion MULTIMODULE board.
- SDT337, which tests the iSBC 337 Numeric Data Processor

**SDT8630 TEST SUITE**

The SDT8630 test suite contains tests for the the iSBC 86/30 Single Board Computer. Table 3-1 lists the SDT8630 tests and their functions.

Table 3-1. SDT8630 Tests

Test	Function
0	ROM Checksum Test
1	8255 Parallel Port Test
2	8259 Interrupt Test
3	8253 Timer Test
4	Fixed Patterns Test
5	Address March Test
6	Sliding Ones Test
7	Dual Port RAM Contention Test

**SDT8630 INITIALIZATION**

When you load the SDT8630 diagnostic test suite into memory it displays the following message:



**SYSTEM DIAGNOSTIC TEST - 8630, Vx.y**  
**Copyright 1982, 1983 Intel Corporation**

where x.y is the version number of the test suite.

After the displaying the initial message, SDT8630 asks a series of questions about the hardware configuration, waiting after each question for your response. The current (or default) setting is enclosed in brackets. For example, in the following question:

**Set CPU clock to (8 or [5]) MHz\***

you can choose 8 or 5 MHz as the clock frequency. In this case, 5 is the default response. If you enter a carriage return alone, SDT8630 assumes the default value is to be used.

The SDT8630 initialization sequence is as follows:

**Set CPU clock to (8 or [5]) MHz\***

In response, enter the clock frequency, in MHz.

**iSBC 309 installed: (y or [n])\***

Answer yes if you have an iSBC 309 memory management board installed on your iSBC 86/30 board.

**Test rx interrupt: ([y] or n)\***

Answer no if you do not want to test the the RxRDY interrupt from the serial port.

**Are rx and tx interrupts or-tied: (y or [n])\***

Answer yes if you have modified your board to take advantage of the interrupt combining feature.

SDT8630 next examines memory locations to determine whether gaps exist in the range of tested memory.

**NOTE**

The memory tests in this suite check only the lowest 60K bytes of memory (addresses 00000H through 0F000H of the iSBC 86/30's on-board memory). This test does not check the iSBC 304 RAM Expansion board. The SDT056 test suite should be run to check memory locations above 0F000H.

If SDT8630 discovers a gap within the range of on-board memory, it displays the following message:

**non-existent memory begins between xxxx:0000 and yyyy:0000**

where *xxxx* and *yyyy* are segment values that indicate the start and stop, respectively, of a range of 16K bytes. This message indicates that a gap in memory exists somewhere in the specified 16K bytes. If SDT8630 displays this message, it automatically ignores all memory tests.

Once memory has been examined for gaps, the initialization sequence is complete. At this point, SDT8630 displays its prompt character (\*), indicating that it is ready to accept commands.

**Using the RESET Command**

When you invoke the **RESET** or **RESET HARDWARE** commands, SDT8630 displays the same series of questions as it does at initialization; this allows you to set the iSBC 86/30 board to a known initial state. There is no question sequence when you invoke the **RESET SOFTWARE** command.

**SDT8630 TEST 0, ROM CHECKSUM**

The ROM Checksum test checks the address and data lines to the ROM on the processor board. It also calculates the checksum of the contents of the ROMs and compares the calculated checksum to a value stored in the ROMs.

Test 0 can return messages of the following form:

**Rom Checksum Error, type Byte: Expected = *ee* Received = *rr***

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

**SDT8630 TEST 1, PARALLEL PORT**

Test 1 checks the PPI (Programmable Peripheral Interface). It writes data into the PPI, reads the data back from the PPI, and compares both values.

Test 1 can return the following error message:

**Error at port *p*: Expected = *ee* Received = *rr***

where *p* is the port in the PPI (A, B, or C) and the values *ee* and *rr* are the expected and received values.

**SDT8630 TEST 2, PIC**

Test 2 tests the 8259 Programmable Interrupt Controller (PIC). It initializes the PIC and causes three interrupts to occur:

- USART's RxRDY signal
- USART's TxRDY signal
- Timer Interrupt

To test the RxRDY interrupt, the test prompts you with a ">". When this prompt appears, enter any character from the keyboard to force the USART interrupt.

Test 2 can return the following messages:

**No Interrupt Detected: Expected = *eint***

The test expected an interrupt to occur at level *eint*, but no interrupt occurred. This error message does not necessarily indicate a malfunction; the test will send it if you do not enter a character within the allowed time-out period during the RxRDY interrupt test.

**Wrong Interrupt Detected: Expected = *eint* Received *rint***

The test expected an interrupt to occur at level *eint* but the interrupt occurred at level *rint* instead.

### **SDT8630 TEST 3, TIMER**

Test 3 checks the 8253 Programmable Interval Timer. First, the CPU initializes the timer and the timer begins counting down from a predetermined value. The CPU then enters a timing loop. When the CPU finishes the loop, it reads the timer's contents twice without stopping the counter in the timer ("on the fly"). The value read must be within a given range of the expected value or an error is reported.

Test 3 can return the following error message:

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

This message indicates that the timer count is out of limits or that you selected the 8 MHz clock at initialization and the actual clock frequency is 5 MHz (or vice versa). The limits are as follows (8 MHz values are enclosed in parentheses; 5 MHz values are not):

	no iSBC 309	iSBC 309 installed
<i>xxxx</i>		
upper limit	0AE15H (0C550H)	00546H (063B1H)
lower limit	0AC15H (0C348H)	00346H (061B1H)
<i>yyyy</i>		
upper limit	05C4AH (08C11H)	00999H (0C675H)
lower limit	0585AH (08880H)	00759H (0C435H)

**SDT8630 TEST 4, FIXED PATTERNS**

The Fixed Patterns test is most useful in checking for stuck data lines to the RAM on the processor board. It writes a checkerboard pattern in two passes throughout RAM (AAAAH and 5555H in alternate passes), checking the actual contents of each address against the value written at the address during each pass.

Test 4 can return the following error message:

**Error at ssss:nnnn**

**Expected xxxx received yyyy reread rrrr xor zzzzzzzzzzzzzzzzzzzzz**

where ssss:nnnn is the address where the error occurred, xxxx and yyyy are the expected and actual values at the location, rrrr is the value obtained when the location was read again, and zzzzzzzzzzzzzzzzzzzzz is the Exclusive-OR of the expected and actual values (in binary).

**SDT8630 TEST 5, ADDRESS MARCH**

The Address March test writes a background pattern into each word of memory. Then it "marches" the complement of that pattern through memory, one word at a time. If the test encounters a location that contains something other than the background pattern, it reports an error. Test 5 returns the same error messages as Test 4.

**SDT8630 TEST 6, iSBC® 86/30 SLIDING ONES**

The iSBC 86/30 Sliding Ones test writes a series of patterns into all RAM locations, checking for errors that are sensitive to particular patterns of data. 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 6 returns the same error messages as Test 4.

**SDT8630 TEST 7, DUAL PORT RAM CONTENTION**

The RAM Contention test forces dual port RAM contention on the processor board. It does this by causing the iSBC 215G Winchester Disk Controller board to perform a DMA transfer to processor board RAM at the same time the processor is trying to move data in that RAM.

**NOTE**

If your system does not contain a Winchester controller board, this test will report a failure. The system may appear to hang up until the test's fail-safe timeout period has elapsed.

When invoked, Test 7 will do the following:

1. Initialize the iSBC 215 Winchester Disk Controller.
2. Write a test pattern in processor board RAM and transfer it to iSBC 215 board RAM.
3. Write zeros in processor board RAM, then cause the iSBC 215 board to return the test pattern to processor RAM.
4. Verify that the returned pattern matches the original. (If the pattern matches, the iSBC 215 board is working well enough to continue.) If the pattern does not match, report errors.
5. Write zeros to processor board RAM memory.
6. Initiate a DMA transfer of the original test pattern from the Winchester controller board to a buffer in processor board RAM while the processor simultaneously reads and writes data in an adjacent buffer in processor board RAM.
7. When the Winchester controller board signals that it is finished (by issuing an interrupt), the processor board finishes its data manipulations and verifies that the data transferred from the iSBC 215 board is correct.

Test 7 returns the following error messages:

**reset error**

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

**86/30 to 215 failure**

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

**215 to 86/30 failure**

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

**transfer buffer miscompare at *nnnn:mmmm*  
expected *ee* received *rr***

While comparing the original test pattern with the pattern returned from the Winchester controller, the test compared a value in the processor board's buffer with the corresponding value in the Winchester controller's buffer and found a mismatch. The value *nnnn:mmmm* indicates the address of the mismatch; *ee* and *rr* indicate the expected and received values.

**contending access failure**

During dual port RAM contention, either the test received an error from the controller or a data mismatch occurred.

If the test returns one of these messages, invoke the SDT215 test. If the SDT215 test indicates no errors with the Winchester controller, replace the processor board.

**SDT8612 TEST SUITE**

The SDT8612 test suite tests the iSBC 86/12A Single Board Computer and the iSBC 300A RAM Expansion Board. Table 3-2 lists the SDT8612 tests and their functions.

Table 3-2. SDT8612 Tests

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

When you load the SDT8612 test suite into memory, it displays the following message:

**SYSTEM DIAGNOSTIC TEST - 8612, Vx.y**  
**Copyright 1982, 1983 Intel Corporation**

where x.y is the version number. At this point, SDT8612 displays its prompt character (\*), indicating that it is ready to accept commands.

#### **SDT8612 TEST 0, ROM CHECKSUM**

The ROM Checksum test checks the address and data lines to the ROM on the processor board. It also calculates the checksum of the contents of the ROMs and compares the calculated checksum to a value contained in the ROMs.

Test 0 can return the following error message:

**Rom Checksum Error, type Byte: Expected = ee Received = rr**

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

#### **SDT8612 TEST 1, PARALLEL PORT**

Test 1 checks the PPI (Programmable Peripheral Interface). It writes data into the PPI, reads the data back from the PPI, and compares both values.

Test 1 can return messages of the following form:

**Error at port p: Expected = ee Received = rr**

where *p* is the port in the PPI (A, B, or C) and the values *ee* and *rr* are the expected and received values.

#### **SDT8612 TEST 2, PIC**

Test 2 tests the 8259 Programmable Interrupt Controller (PIC). It initializes the PIC and causes three interrupts to occur:

- USART's RxRDY signal
- USART's TxRDY signal
- Timer Interrupt

To test the RxRDY interrupt, the test prompts you with an ">". When this prompt appears, enter any character from the keyboard to force the USART interrupt.





value obtained when the location was read again, and zzzzzzzzzzzzzzzzz is the Exclusive-OR of the expected and actual values (in binary).

**SDT8612 TEST 5, FIXED PATTERNS (iSBC<sup>®</sup> 300A BOARD)**

Test 5 performs the same tests as Test 4, but it tests the iSBC 300A RAM Expansion board, if installed on the processor board. It returns the same error messages as Test 4.

If this test reports an error, replace the iSBC 300A board (after first making sure that the problem is not caused by the processor board).

**SDT8612 TEST 6, ADDRESS PATTERNS**

The Address Patterns test checks the address lines to the RAM on the processor board and the iSBC 300A board. A pattern that is formed by adding the four bytes (two each from segment and offset) of the address together is written into each location in memory. Then, each RAM cell is read and the value read is compared against the value written. This test returns the same error messages as test 4.

**SDT8612 TEST 7, SLIDING ONES**

The Sliding Ones test writes a series of patterns into all RAM locations, checking for errors that are sensitive to particular patterns of data. 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 7 returns the same error messages as test 4.

**SDT8612 TEST 8, SLIDING ONES (iSBC<sup>®</sup> 300A BOARD)**

Test 8 checks the memory on the iSBC 300 RAM Expansion board, using the same tests that Test 7 performs on the processor board.

This test returns the same error messages as Test 4.

**SDT8612 TEST 9, DUAL PORT RAM CONTENTION**

The RAM Contention test forces dual port RAM contention on the processor board. It does this by causing the iSBC 215 Winchester Disk Controller board to perform a DMA transfer to processor board RAM at the same time the processor is trying to move data in that RAM.

**NOTE**

If your system does not contain a Winchester controller board, this test will report a failure. The system may appear to hang up until the test's fail-safe timeout period has elapsed.

Test 9 does the following:

1. Initializes the iSBC 215 Winchester Disk Controller.
2. Writes a test pattern in processor board RAM and transfers it to iSBC 215 board RAM.
3. Writes zeros in processor board RAM, then causes the iSBC 215 board to return the test pattern to processor RAM.
4. Verifies that the returned pattern matches the original; reports any errors. If the pattern matches, the 215 board is working well enough to continue the test.
5. Writes zeros to processor board RAM memory.
6. Initiate a DMA transfer of the original test pattern from the Winchester controller board to a buffer in processor board RAM, while the processor simultaneously reads and writes data in an adjacent buffer in processor board RAM.
7. When the Winchester controller board signals that it is finished (by issuing an interrupt), the processor board finishes its data manipulations and verifies that the data transferred from the iSBC 215 board is correct.

Test 9 returns the following error messages:

**reset error**

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

**86/12A to 215 failure**

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

**215 to 86/12A failure**

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

**transfer buffer miscompare at *nnnn:mmmm*  
expected *ee* received *rr***

While comparing the original test pattern with the pattern returned from the Winchester controller, the test compared a value in the processor board's buffer with the corresponding value in the Winchester controller's buffer and found a mismatch. The value *nnnn:mmmm* indicates the address of the mismatch; *ee* and *rr* indicate the expected and received values, respectively.

**contending access failure**

During dual port RAM contention, either the test received an error from the controller or a data mismatch occurred.

If the test returns one of these messages, invoke the SDT215 test. If the SDT215 test indicates no errors with the Winchester controller, replace the processor board.

**SDT337 TEST SUITE**

The SDT337 Diagnostic Test Suite consists of one test that checks all of the functions of the iSBC 337 Numeric Data Processor.

When you load the SDT337 test suite into memory, it displays the following message:

**SYSTEM DIAGNOSTIC TEST - 337, Vx.y  
Copyright 1982, 1983 Intel Corporation**

where x.y is the version number. At this point, SDT337 displays its prompt character (\*), indicating that it is ready to accept commands.

SDT337 returns error messages of the following form:

**Opcode failure - *instruction name***

where *instruction name* is the name of the 8087 instruction that failed.

If SDT337 reports an error, replace the iSBC 337 board (after first running SDT8630 or SDT8612 to make sure the processor is functional).



## CHAPTER 4 MEMORY SYSTEM DIAGNOSTIC TESTS

This chapter contains detailed descriptions of the System Diagnostic Tests for memory boards and related modules used in System 300 Series microcomputers. Included are descriptions of the following tests:

- SDTRAM, which tests the iSBC 056A 256K RAM Board and most other MULTIBUS-compatible memory boards
- SDT309, which tests the iSBC 309 Memory Management Board

### SDTRAM TEST SUITE

SDTRAM aids the user in detecting and isolating faults on the memory boards of the 86/300 Series microcomputers. The test supports Intel's A, B, and C series of memory boards and provides limited support for No-Parity boards. SDTRAM also tests memory board ECC/Parity and may be used to test memory boards other than those listed without testing ECC/Parity.

SDTRAM consists of nine test routines and one utility and can hold configuration data for 16 memory boards. When invoked, SDTRAM tests all boards defined in the configuration.

The memory range checked by this test is from 0B000H to the top of memory. Table 4-1 lists the tests of the SDTRAM.

Table 4-1. SDTRAM Tests

Test	Function
0	Fixed Patterns
1	Address March
2	Sliding Ones Pattern
3	Execute from RAM
4	A-Series Parity Logic and RAMs
5	A-Series Interrupt Detection

Table 4-1. SDTRAM Tests (Continued)

Test	Function
6	C-Series Check Bit Logic
7	C-Series Check Bit RAMs
8	C-Series Error Correction
9	C-Series Interrupt Detection
A	Rd/Wr Scope Loop Utility

**SDTRAM INITIALIZATION**

Booting SDTRAM automatically invokes the reset software procedures, which provide the user with means to alter the test limits and parameters. Reset software procedures may be invoked at any time during memory testing by entering the command RESET SOFTWARE or RESET.

Prompts for numbers are answered with decimal, hexadecimal, octal, or binary values terminated with a carriage return. The desired radix (default is hex) is indicated by using one of these characters: h or H (hex), t or T (decimal), q or Q (octal), y or Y (binary).

In an input overflow condition, only the least significant digits are used and leading zeros are unnecessary. Default values are printed with each query; to retain the indicated default values, the user need only enter a carriage return in response. When the user changes the values, the new values entered become the default values.

The reset software routine displays a main menu and waits for the user to make a selection. After processing the selection, the routine returns to the main menu. The user may exit the reset software routine by selecting the menu item labeled "Exit." "Exit" is the default value and may be selected by entering a carriage return.

**EXAMPLE:****Configuration Main Menu:**

- |                               |                                 |
|-------------------------------|---------------------------------|
| ( 0) Specify number of boards | ( 5) Set operational switches   |
| ( 1) Define board parameters  | ( 6) Select board for testing   |
| ( 2) Change board parameters  | ( 7) Deselect board for testing |
| ( 3) Display board parameters | ( 8) Set preconfigured defaults |
| ( 4) Exit                     |                                 |

Each main menu item is described in the following sections. Each of the user prompts displays the default value, surrounded by brackets, for the parameter in question. Some prompts also display a range of possible input values on the prompt line, in the form x-y.

**EXAMPLE:**

Enter Number of Boards in The System: 01..16 [01]\*

In this example, the range is 1-16 and the current value (default) is 1.

**SPECIFY NUMBER OF BOARDS**

This selection prompts the user for the number of memory boards in the system. SDTRAM uses this number in subsequent operations, so it should be the first selection made.

**EXAMPLE:**

Enter Number of Boards in The System: 01..16 [01]\*

The default of one board is assumed by SDTRAM to be the 128K (no-parity) on-board memory. Any additional memory should be specified by the user.

**DEFINE BOARD PARAMETERS**

This selection prompts the user for board parameters starting with board number one and ending with the number of boards specified. The user must specify the number of boards before choosing this selection.

**EXAMPLE:**

Board Number: 01..02 [01]\*

Define Board Type:

(0) A - Series

(2) C - Series

(1) B - Series

(3) No - Parity

[2]\*

Initial Segment: 0000H..FFFFH [1000H]\*

Final Segment: 0000H..FFFFH [7FFFH]\*

ECC Port: 0000H..FFFFH [01C0H]\*

Do You Intend to Invoke The Interrupt Test?

(0) No (1) Yes

[1]\*

Interrupt Origin:

(0) NMI

(2) Slave PIC

(1) Master PIC

[0]\*

If the response to the preceding question is 1 or 2, the test asks the following question:

Interrupt Level: 0..7 [0]\*  
 Bus Configuration:  
     (0) MULTIBUS                      (1) LBX  
     [1]\*  
 ECC Mode:  
     (0) Correcting                    (1) Non-Correcting  
     [0]\*

If the initial and final segments are equal, one paragraph (16 bytes) is tested. After the user enters the memory range, the reset software routine makes a preliminary check for nonexistent memory within the test range. If the test detects nonexistent memory, an error message appears and the user must accept or reject the range values.

**EXAMPLE:**

```

*** Range Appears to Contain Non-Existent Memory,
    Do You Want to Use These Values?
    (0) No                              (1) Yes
[0]*
  
```

The routine issues no prompts for information inapplicable to the board type.

**CHANGE BOARD PARAMETERS**

This selection prompts the user for a board number and then the parameters for that board. The prompts are the same as for Define Board Parameters except that it prompts for one board at a time.

**DISPLAY BOARD PARAMETERS**

This selection displays the board type and number of all the defined boards, and then prompts for a board number to be displayed. The default board number starts at 1 and rises to the number of boards specified. All board parameters can be displayed by entering a carriage return after each prompt.

**EXAMPLE:**

```

Defined Boards:
  Board No: 01 Type: C - Series (Selected)
  Board No: 02 Type: A - Series (Deselected)

Board Number to Display: 01..02 [01]*
  
```

## Board No: 01 Type: C - Series

Initial Segment = 1000H  
 Final Segment = 7FFFH  
 ECC Status Port = 01C0H  
 Interrupt Origin = NMI  
 Bus Configuration = iLBX  
 ECC Mode = Correcting

## Board No: 02 Type: A - Series

Initial Segment = 8000H  
 Final Segment = BFFFH  
 Parity Port = 0000H  
 Interrupt Origin = Master PIC, Level 04

**EXIT**

This is the default selection that causes an exit to the command line prompt. At this point, tests and other functions may be invoked. If Fast\_Test is selected, a message is printed to remind the user. If the first board defined is a C-Series board running on the iLBX bus, a message appears regarding the C-Series test.

When Fast\_Test is set, it performs a quick sampling of each chip. For more comprehensive testing, the user can set No-Fast\_Test.

**NOTE**

The C-Series Check Bit RAM test has been set to IGNORE. This test fails if SDTRAM is located on the board being tested. If the range 0-64K does not reside on the C-Series board, the tests can be set to RECOGNIZE and will then execute properly.

**SET OPERATIONAL SWITCHES**

This selection prints the current settings of the four Operational Switches, prints a menu, and then prompts for a change selection. (For more information, see "Operational Switches" later in this chapter.)

**EXAMPLE:**

## Current Switch Settings:

Fast\_Test No-Scope\_Loop Status\_Line No-Stop\_on\_Error

## Set Switches Menu:

( 0) Fast\_Test ( 5) No-Fast\_Test  
 ( 1) Scope\_Loop ( 6) No-Scope\_Loop  
 ( 2) Status\_Line ( 7) No-Status\_Line  
 ( 3) Stop\_on\_Error ( 8) No-Stop\_on\_Error  
 ( 4) Exit  
 [4]\*



If Scope\_Loop is selected, SDTRAM prompts for Scope Loop parameters.

**EXAMPLE:**

Set Scope\_Loop Parameters:

Segment: 0000H..FFFFH [0000H]\*

Offset: 0000H..FFFFH [0000H]\*

Type of Access:

(0) Read Byte

(2) Write Byte

(1) Read Word

(3) Write Word

[0]\*

Write Pattern: 0000H..FFFFH [0000H]\*

Delay in 10ths of a mSec: 00000..65535 [00000]\*

**SELECT BOARD FOR TESTING**

This selection prints the board number, board type, and Selected/Deselected setting for all defined boards. It then prompts the user to select a board for testing. The default board number is the first deselected board found or the last specified board.

**EXAMPLE:**

Defined Boards

Board No: 01 Type: C - Series (Selected)

Board No: 02 Type: A - Series (Deselected)

Board number two is the default because it is the first deselected board.

**DESELECT BOARD FOR TESTING**

This selection does the opposite of "Select Board for Testing"; it chooses boards that are not to be tested.

**SET PRECONFIGURED DEFAULTS**

This function causes all parameters to be reset to the values defined during SDTRAM invocation. The routine offers the user a chance to decline the reset function by answering the prompt yes or no.

**EXAMPLE:**

Are You Sure You Want to Set Preconfigured Defaults?

(0) No

(1) Yes

[0]\*

The default for this prompt is always no.

**ERRORS**

The following errors may be encountered during reset software dialogues.

Entering an unspecified range value produces the following error message and causes the routine to reissue the prompt.

**\*\*\* ERROR: value out of range -- r. Please try again.**

Where

r = the user's response

Entering invalid numeric values produces the following error message and causes the routine to reissue the prompt.

**\*\*\* ERROR: Invalid input. Please try again.**

If input for a menu selection is out of range, the following message appears and the menu is reprinted.

**\*\*\* ERROR: Not a valid selection -- r. Please try again.**

Where

r = the user's response

When the user enters illegal segment values, one of the following messages appears, and the system continues to prompt the user until a legal range is entered.

**\*\*\* ERROR: Segments Less Than 1000 Are Illegal,  
Enter a New Range**

**\*\*\* ERROR: Final Segment Must Be Greater Than Initial,  
Enter a New Range**

**\*\*\* ERROR: Overlaps Range xxxx to yyyy Specified for Board  
zz, Try Again**

If a memory scan fails on a range entered, the following message appears:

**\*\*\* ERROR: Range Appears to Contain Non-Existent Memory,  
Do You Want to Use These Values?**

(0) No  
[0]\*

(1) Yes

**Using the RESET Command**

When the user invokes the RESET or RESET SOFTWARE commands, SDTRAM displays the same sequence of questions as it does at initialization.

The RESET HARDWARE command initializes ECC/Parity logic and writes a pattern to the range on all applicable boards defined in the reset software dialogue. SDTRAM automatically invokes this routine when it starts up. The reset hardware routine may be invoked at any time by typing the command RESET HARDWARE.

### DEBUG LEVELS

SDTRAM has three significant debug levels that control the type of error information displayed when it detects an error. The debug level is set by entering the command

**\*DEBUG = xxxx**

Where

xxxx = a hex number from 0 to FFFFH

The levels (values) significant in SDTRAM are 0, 1, and 3.

When DEBUG=0, the tests do not print any error, but print the test name and a PASS or FAIL indicator.

When DEBUG=1, some of the tests try to interpret the error values. They execute to completion without printing errors; when the test is finished, it prints a small summary that may include the location of a suspect chip.

EXAMPLE:

```

0000H      Fixed Patterns
*** ERROR: While Testing Board No: 01 Type: C-Series
      First Error at 4000:000, Last Error at BFFF:0000
          Suspect Data Chip at Location U33
0000H      Fixed patterns                               FAILED <==

```

The third line of debug level=1 errors varies and may take any one of the following forms:

```

      Suspect ECC Chip at Location U11
      Suspect Parity Chip at Location 52
      Multiple Bit Failure, Suspect Addressing Problem
      Failing Data Bit is Bit 9

```

When DEBUG=3, all error information is printed. Most of the tests print out a standard error format with expected and received values.

EXAMPLE:

```

** Parity ERROR: Bank = 0 Row = 1 Board No: b Type: t
*** ERROR: at 'SSSS:NNNN' Board No: b Type: t
expected = xxxx received = yyyy reread = rrrr xor = zzzzzzzzzz

```

Where

SSSS:NNNN = address of the failing RAM location  
 b = board number (1-16)  
 t = board type (A, B, C series, or No-Parity)  
 xxxx = word value written  
 yyyy = word value initially read back  
 rrrr = word value read back during second read  
 zzzzzzzzzz = Exclusive-OR of the expected and received values

## OPERATIONAL SWITCHES

SDTRAM utilizes four operational switches: Fast Test, Scope Loop, Status Line, and Stop On Error. All four operational switches are set and reset by selecting the configuration main menu item, "Set Operational Switches."

### FAST\_TEST

Setting Fast\_Test causes some of the routines in SDTRAM to execute faster. These routines test only the first 16 bytes in each 4K block over the specified range, ensuring a sampling from each chip and catching most types of errors. Some types of errors cannot be detected with Fast\_Test.

### SCOPE\_LOOP

Setting Scope\_Loop causes SDTRAM to prompt for Scope\_Loop parameters. These parameters are stored in the V-variables and may be changed by using the command syntax

**V(xx)=yy**

Where

xx = the number of the V-variable  
 yy = the desired set value

After Scope\_Loop is set, the SUMMARY and DESCRIBE commands indicate an added test titled "Rd/Wr Scope Loop Utility", test 0AH. The Scope Loop Utility is invoked the same way the tests are invoked.

### STATUS\_LINE

When Status\_Line is set and Fast\_Test is not, most tests print a status line indicating the segment being tested.

**EXAMPLE:**

0000H Fixed Patterns  
Currently Testing Board No: 01 Type: C-Series at [4000:XXXX]

This feature informs the user that the test is still executing. Setting `Fast_Test` prevents the status line from being printed because it would defeat the purpose of the `Fast_Test` feature.

**STOP\_ON\_ERROR**

With `Stop_On_Error` set and `DEBUG=3`, SDTRAM stops printing after the first set of errors and then waits for the user to enter a carriage return. `Stop_On_Error` can be useful for detecting intermittent errors. If a routine fails with `Stop_On_Error` set, the first encountered error is preserved on the screen. `Stop_On_Error` has no effect when `DEBUG=1`.

**TEST DESCRIPTIONS**

The following sections provide detailed descriptions of the routines in the SDTRAM suite.

**SDTRAM TEST 0, FIXED PATTERNS**

This test checks the ability of each RAM location to store word values 5555H and AAAAH. This pattern of alternating ones and zeros is a standard checkerboard pattern test, useful for finding stuck data lines.

The errors displayed depend on the `DEBUG` level set. See the section in this chapter titled "Debug Levels."

**SDTRAM TEST 1, ADDRESS MARCH**

This test checks that each location in the test range has a unique address. The test range is written with a background pattern. Then each word is read and verified, written with the complement, and then reread. If the test encounters a cell that is not supposed to have been complemented yet, but does not contain the background pattern, an error is logged. This error may be either a data error or an address error. If test 0 (which cannot detect an address error) does not report an error at the same location, then the error is an address error; otherwise the error is a data error.

The errors displayed depend on the `DEBUG` level set. See the section in this chapter titled "Debug Levels." The bit(s), marked by the `XOR` field in the message, highlights the data bit(s) that has an addressing problem. On a board arranged in rows (such as the iSBC 012C), this bit information specifies the chip at fault. If several chips appear to be at fault, the user should check for stuck or shorted address lines.

**SDTRAM TEST 2, SLIDING ONES PATTERN**

This test is a form of galpet (galloping pattern) used to check memory for pattern-sensitive errors. A series of sweeps writes hex values to memory and looks for changes. The hex values 0000, 8001, C003, F00F, F81F, FC3F, FE7F, and FFFF are used.

The errors displayed depend on the DEBUG level set. See the section in this chapter titled "Debug Levels."

**SDTRAM TEST 3, EXECUTE FROM RAM**

This test verifies that programs can be executed from off-board RAM. It copies a 64-byte block of code into a series of 16K-byte blocks of RAM. A pattern is embedded in the middle of the code block. The code block fills the rest of the 16K block with copies of itself. After executing the code, the test verifies the pattern in all copies. The test repeats this process for all 16K blocks in the test range. This test emphasizes a highly mixed combination of reads and writes, to thoroughly exercise bus arbitration logic and bus drivers.

Since unpredictable results may occur if the test limits are too small or if the code block is improperly copied, this test is initially set to be IGNORED. As a minimum prerequisite for running this test, the system must pass the Fixed Patterns test and the Address March test. Additionally, the test checks to see that the test limits are large enough before executing.

The errors displayed depend on the DEBUG level set. See the section in this chapter titled "Debug Levels."

The following message appears if the test range is too small and DEBUG = true:

```
*** ERROR: bounds too small for execution  
Board no: b Type: t
```

**SDTRAM TEST 4, A-SERIES PARITY LOGIC AND RAMS**

This test verifies the parity logic and RAM on A-Series boards. The test simulates parity errors at each location and verifies that the errors are detected. The accesses are byte reads and writes with two different patterns that verify both even and odd parity states.

Following is an example of errors when DEBUG=1:

```
*** ERROR: While Testing Board No: 01 Type: A-Series  
First Error at 4000:0000, Last Error at 5F00:0006
```

When DEBUG=3:

```
*** ERROR: at 4000:0000 Board No: 01 Type: A-Series  
Forced Parity Error Not Detected, Pattern = 08
```

**SDTRAM TEST 5, A-SERIES INTERRUPT DETECTION**

This test verifies that the A-Series board being tested can generate an interrupt and that the interrupt is detectable by the processor boards. It can test interrupts from the Master PIC, Slave PIC, or NMI. When testing for an interrupt from the Master or Slave PIC, the test first verifies that the expected interrupt can be cleared. If the interrupt cannot be cleared, an error message appears and the test fails. It then attempts to cause the interrupt by forcing an error on the board. If the interrupt remains undetected, the test prints an error message and fails. The test ignores all interrupt request lines except the one specified in the reset software routine.

Following are examples of errors displayed when DEBUG=1 or 3.

**\*\*\* ERROR: Cannot Clear Interrupt, Check Port Addressing**

**\*\*\* ERROR: Expected Interrupt Did Not Occur,  
interrupt expected = 00001000 interrupt received = 00000001**

**\*\*\* ERROR: NMI Interrupt Did Not Occur**

**SDTRAM TEST 6, C-SERIES CHECK BIT LOGIC**

This test verifies the check bit generation logic for C-Series boards. It writes all 65536 patterns to the first location in the board range. For each write, it verifies the correct ECC syndrome bits are generated.

Following is an example of errors displayed when DEBUG=1 or 3.

**\*\*\* ERROR: Bad Check Bits Generated at 8000:0000  
Pattern = DEAD Expected 3F Received 3C**

**SDTRAM TEST 7, C-SERIES CHECK BIT RAMS**

This is a RAM test of the ECC RAM chips. The test writes the two patterns, FFDAH and FEFFH, that generate ECC patterns 15H and 2AH. The two patterns produce alternating bit patterns on the ECC RAMs. The test writes the pattern and then reads the ECC RAM bits. If the ECC RAM bits do not match the expected pattern, an error is logged. This test fails if SDTRAM executes from the board being tested.

The errors displayed depend on the DEBUG level set. See the section in this chapter titled "Debug Levels."

**SDTRAM TEST 8, C-SERIES ERROR CORRECTION**

This test verifies the error detection/correction of single-bit errors on C-Series boards. The test writes multiple patterns to the first location in the range, starting with zero and incrementing by three. For each pattern, 16 errors are injected by inverting each bit one at a

time. The altered pattern is written with diagnostic mode set, which prevents writing of ECC bits, and then read with correcting mode set and verified as correct.

The errors displayed depend on the DEBUG level set. See the section in this chapter titled "Debug Levels."

#### **SDTRAM TEST 9, C-SERIES INTERRUPT DETECTION**

This test verified that the C-Series board being tested can generate a detectable interrupt. It tests interrupts from the Master PIC, Slave PIC, and NMI. When testing for an interrupt from the Master or Slave PIC, the test verifies that the expected interrupt can be cleared. If the interrupt cannot be cleared, an error message appears and the test fails. The test then attempts to cause the interrupt by forcing an error on the board. If the interrupt goes undetected, the test prints a message and fails. The test ignores all interrupt request lines except the one specified in reset software.

Following are examples of errors displayed when DEBUG=1 or 3.

**\*\*\* ERROR: Cannot Clear Interrupt, Check Port Addressing**

**\*\*\* ERROR: Expected Interrupt Did Not Occur,  
interrupt expected = 00001000 interrupt received = 00000001**

**\*\*\* ERROR: NMI Interrupt Did Not Occur**

#### **SDTRAM TEST A, RD/WR SCOPE LOOP UTILITY**

This utility is used to loop on a memory location for the purpose of providing oscilloscope signals. It only appears in the test list if the operational switch "Scope\_Loop" has been set, and it disappears from the list when "No\_Scope\_Loop" is set. The parameters for looping are held in the V-variables and can be altered through the command line. The user can terminate the loop by entering CONTROL-C.

When the utility is invoked, the current values of the parameters are displayed.

#### **EXAMPLE:**

\*test a

#### **Scope Loop Parameters:**

```
v(0):v(1)  -Address = 8000:0000
v(2)      -Pattern = CAFE
v(3)      -Type of Access = Read Byte
           0 - Read Byte
           1 - Read Word
           2 - Write Byte
           3 - Write Word
           - Delay = 5/10ths of a mSec
```

**\*\*\* To Quit Looping, Type CTRL-C \*\*\***



**SDT309 TEST SUITE**

The SDT309 test suite contains tests for the iSBC 309 Memory Management board. Table 4-2 lists the tests and their functions.

Table 4-2. SDT309 Tests

Test	Function
0	Interrupt Processing
1	Exception Conditions
2	Physical Address Verification
3	Memory Bounds
4	Memory Access
5	DMA I/O

**SDT309 INITIALIZATION**

When invoked, SDT309 displays the following message:

```

SYSTEM DIAGNOSTIC TEST - 309, Vx.y
Copyright 1982, 1983 Intel Corporation

```

where x.y is the version number.

Next, SDT309 determines whether its processes can be mapped into available memory. It displays the following information to indicate the resultant mapping:

**PROCESS CONFIGURATION**

```

          PAGES/ PHYSICAL          PAGES/ PHYSICAL
PROCESS  BLOCK  BLOCK  PAGE  PROCESS  BLOCK  BLOCK  PAGE
=====  =====  =====  =====  =====  =====  =====  =====
number  blknum  pages  firstpg  number  blknum  pages  firstpg
.       .       .       .       .       .       .       .
.       .       .       .       .       .       .       .
.       .       .       .       .       .       .       .

```

This display lists the 64K memory blocks allocated for each user process, the number of 2K pages in each block, and the number of the physical page that corresponds to the first page of each block.

SDT309 assigns contiguous memory for the pages in a block. It assigns "no access" to all pages that do not represent physical memory and to all nonspecified pages. It maps block 0 of all processes into the same physical block in memory. It also prevents any other block from overlapping block 0 by reassigning the start of that block to the first location after block 0 and truncating the block by an amount equal to the overlapped portion.

If errors occur in the process configuration, SDT309 adjusts the configuration to make it a valid one. SDT309 returns the following messages to indicate such adjustments:

**ERROR: BLOCK BEGINS BEYOND THE END OF AVAILABLE MEMORY**  
**PROCESS NUMBER** *process-number*, **BLOCK NUMBER** *block-number*  
**TRUNCATED TO ZERO**

**ERROR: BLOCK 0 INCORRECTLY MAPPED**  
**PROCESS NUMBER** *process-number*, **BLOCK 0 REMAPPED TO**  
**PHYSICAL PAGE 0 WITH 32 PAGES**

**ERROR: NUMBER OF PAGES IS GREATER THAN 32**  
**PROCESS NUMBER** *process-number*, **BLOCK NUMBER** *block-number*  
**RESET TO 32**

**ERROR: PAGES IN THE BLOCK EXTEND BEYOND**  
**THE END OF AVAILABLE MEMORY**  
**PROCESS NUMBER** *process-number*, **BLOCK NUMBER** *block-number*  
**TRUNCATED TO** *new-limit*

**ERROR: PHYSICAL PAGES 0 THROUGH 31 ARE RESERVED**  
**PROCESS NUMBER** *process-number*, **BLOCK NUMBER** *block-number*  
**REMAPPED TO PAGE 32 WITH** *num-blocks* **BLOCKS**

**ERROR: YOUR SPECIFIED PAGE LIMIT IS GREATER THAN**  
**YOUR AVAILABLE MEMORY.**  
**I'M USING** *actual-limit* **AS THE UPPER BOUND**

**ERROR: YOUR SPECIFIED PAGE LIMIT MUST BE**  
**AT LEAST 31 (FOR 64K)**

To change the default configuration of the processes that the test uses, the user can modify the SDT309 configuration file. By modifying this file, the user can also change the board configuration that the tests expect, the order in which the tests run, the initial state of the tests, and other settings. Refer to Appendix A, "Configuring the Diagnostic Tests", for more information.

After verifying (and adjusting) the process configuration, SDT309 attempts to map the configuration to the iSBC 309 board. It maps

each page onto the board and then reads the resultant mapping. If the value read fails to match the value written, the test displays the following error message, if the status of the DEBUG command is TRUE.

**ERROR: MAPPED VALUE DOES NOT MATCH VALUE READ  
EXPECTED: *ee*, RECEIVED: *rr***

where *ee* is the value that the SDT309 mapped into the board and *rr* is the value that it read.

The test displays the following message, regardless of the setting of the DEBUG command:

**ERROR: UNIT FAILED MAPPING THE MEMORY CONFIGURATION**

If these messages appear, they indicate a faulty iSBC 309 board. The user can attempt to run the SDT309 tests to verify the problem, but the tests may produce erroneous results.

After initialization, SDT309 displays the prompt character (\*), indicating that it is ready to accept commands.

The RESET and RESET HARDWARE commands also initialize SDT309; the process is the same as described earlier. The RESET SOFTWARE command has no effect.

#### **SDT309 TEST 0, INTERRUPT PROCESSING**

Test 0 verifies that the processor can service interrupts while the iSBC 309 board is in system and user modes. It checks internal and external interrupts in both modes.

To check external interrupts, Test 0 sets a flag and sets up the timer to generate an interrupt. Then the test enters a timing loop that increments the counter and checks the flag. When the interrupt occurs, an interrupt service routine resets the flag, verifies the state of the iSBC 309 system request and trap flags, and returns to the test at the point of interruption.

The test then checks the flag to see that it has returned to its original value. If the interrupt does not occur, or if the iSBC 309 flag is in an incorrect state, the test returns an error message. If control returns to an unexpected location, the test fails, leaving the system in an unknown condition; the system may have to be reset to recover from such an error.

To check internal interrupts, the test sets a flag and issues an INT instruction. The interrupt service routine resets the flag, verifies the state of the iSBC 309 system request and trap flags, and returns to the point of interruption. The test then checks to see that the flag is still set and the iSBC 309 flags are correct. Test 0 can return the following error messages:

**ERROR: CANNOT ENABLE MMU.**

**ERROR: NO INTERRUPT OCCURRED IN SYSTEM MODE.**

**ERROR: NO INTERRUPT OCCURRED IN USER MODE.**

**ERROR: [SYSTEM MODE] MMU DID NOT RECOGNIZE  
INTERNAL INTERRUPT**

**ERROR: [SYSTEM MODE] MMU DID NOT RECOGNIZE  
MASKABLE INTERRUPT**

**ERROR: [SYSTEM MODE] UNABLE TO GENERATE INTERNAL  
INTERRUPT**

**ERROR: [USER MODE] UNABLE TO GENERATE INTERNAL  
INTERRUPT.**

If Test 0 reports an error, the user should verify that the other SDT test suites run without error. The SDT309 configuration file should be checked to ensure that the process configuration is correct. (See Appendix A.) If the error persists, the iSBC 309 board should be replaced.

#### **SDT309 TEST 1, EXCEPTION CONDITIONS**

This test verifies that the iSBC 309 board generates the expected exception conditions in system and user modes. To do this, it performs the following operations:

- Port I/O operations. The test expects exceptions in user mode if the board is configured to prohibit them.
- Enable/disable interrupt. The test expects exceptions for disable interrupt in user mode.
- Map memory. In system mode, the test expects no exceptions when setting the accesses. In user mode, if user I/O is allowed, the test expects no exceptions; it also expects the map memory command to have no effect. Otherwise, the test expects an exception in user mode.
- Set process register command. In system mode, the test expects no exceptions. In user mode, if user I/O is allowed, the test expects the command to have no effect. Otherwise, the test expects an exception in user mode.
- System call command. In user mode, the test verifies that a trap occurs and that the command sets the SR and TR flags.
- Clear status command. In system mode, the test verifies that the SR and TR flags have been reset and that no exceptions occur. In user mode, the test verifies that the clear status command causes no exceptions.

Test 1 can return the following error messages:

**ERROR: MAPPED VALUE DOES NOT MATCH VALUE READ  
EXPECTED: *ee*, RECEIVED: *rr***

with any of the following messages as the next line of the message:

**CLEAR STATUS COMMAND FAILED.  
INPUT UNSUCCESSFUL.  
OUTPUT UNSUCCESSFUL.  
SYSTEM CALL COMMAND CAUSED EXCEPTION.  
UNABLE TO DISABLE INTERRUPTS.  
UNABLE TO ENABLE INTERRUPTS.  
UNABLE TO MAP WITH R ACCESS.  
UNABLE TO MAP WITH RW ACCESS.  
UNABLE TO MAP WITH W ACCESS.**

The value *ee* is the value that the test attempted to map into the iSBC 309 board, and *rr* is the value the test received when verifying the mapping.

**ERROR: USER MODE FAILURE:**

Any of the following messages can appear as the second line of this message:

**ABLE TO DISABLE INTERRUPTS.  
ABLE TO MAP WITH R ACCESS.  
ABLE TO MAP WITH RW ACCESS.  
ABLE TO MAP WITH W ACCESS.**

**CANNOT EXECUTE PROCESS REGISTER COMMAND**

The test can return this message only if the iSBC 309 board is configured to allow I/O in user mode.

**CLEAR STATUS COMMAND FAILED.**

**EXECUTED SET/CLEAR MODE WITHOUT EXCEPTION.**

**INPUT SUCCESSFUL.**

The test can return this message only if the iSBC 309 board is configured to prohibit I/O in user mode.

**INPUT UNSUCCESSFUL.**

The test can return this message only if the iSBC 309 board is configured to allow I/O in user mode.

**OUTPUT SUCCESSFUL.**

The test can return this message only if the iSBC 309 board is configured to prohibit I/O in user mode.

**OUTPUT UNSUCCESSFUL.**

The test can return this message only if the iSBC 309 board is configured to allow I/O in user mode.

**PROCESS REGISTER COMMAND EXECUTED.**

The test can return this message only if the iSBC 309 board is configured to prohibit I/O in user mode.

**UNABLE TO ENABLE INTERRUPTS.****ERROR: UNIT FAILED MAPPING THE MEMORY CONFIGURATION**

If Test 1 reports an error, the user should verify that the other SDT test suites run without error and then check the SDT309 configuration file to ensure that the process configuration is correct. If the error persists, the iSBC 309 board should be replaced.

**SDT309 TEST 2, PHYSICAL ADDRESS VERIFICATION**

Test 2 verifies that the pages of a physical process reside at the physical locations allotted for them. First the test writes an identifier to a page of the process while the iSBC 309 board is enabled. It then disables the iSBC 309 board and reads the physical address of the page to verify that the identifier is there. The test checks each page of each process; it does not check the memory for block 0.

Test 2 can return the following error message:

**ERROR: P = process, B = block, PG = page,  
EXPECTED = ee, ACTUAL = aa**

where

*process* is the process number where the error occurred.

*block* is the block number where the error occurred.

*page* is the page number where the error occurred.

*ee* is the expected value.

*aa* is the actual value.

If Test 2 reports an error, the user should verify that the other SDT test suites run without error and then check the SDT309

configuration file to ensure that the process configuration is correct. If the error persists, the iSBC 309 board should be replaced.

#### **SDT309 TEST 3, MEMORY BOUNDS**

This test verifies that the iSBC 309 board ignores the high-order bits of any addresses that are larger than allowed. The verification procedure is similar to that used in Test 2. The test can return the same error message that Test 2 returns.

If Test 3 reports an error, the user should verify that the other SDT test suites run without error and then check the SDT309 configuration file to ensure that the process configuration is correct. If the error persists, the iSBC 309 board should be replaced.

#### **SDT309 TEST 4, MEMORY ACCESS**

The Memory Access Test verifies that the access assigned to each page properly protects the page. It sets the access for all pages in each process to R, W, and then to R/W. It then verifies the access for each page by attempting read and write operations. This test does not check the pages of block 0.

Test 4 can return the following error message:

**ERROR: P = process, B = block, PG = page,  
ACCESS = acc, OPER = op**

where

*process* is the process number where the error occurred.

*block* is the block number where the error occurred.

*page* is the page number where the error occurred.

*acc* is the type of access allocated to the page.

*op* is the type of operation attempted (R or W).

If Test 4 reports an error, the user should verify that the other SDT test suites run without error and then check the SDT309 configuration file to ensure that the process configuration is correct. If the error persists, the iSBC 309 board should be replaced.

#### **SDT309 TEST 5, DMA I/O**

The DMA I/O Test verifies that a DMA transfer can occur from the iSBC 215 to each page of each configured process. First the test enables the iSBC 309 board, places it into system mode, and starts the DMA from a process. Then it disables the iSBC 309 board,

determines the physical address of the transfer from the memory map, and checks the data area for errors.

#### NOTE

This test should not be run unless your system contains an iSBC 215 board. In addition, if there is an error that causes data to be transferred into memory containing code or data required by the SDT309, you may have to reset the system to recover.

Test 5 can return the following error messages:

**ERROR: HOST TO 215 TRANSFER FAILURE**

**ERROR: 215 TO HOST TRANSFER FAILURE**

**ERROR: RECEIVER BUFFER NOT ZEROED AFTER MMU ENABLED**

**PROCESS** = *process*, **BLOCK** = *block*, **BYTE** = *byte*,

**RECEIVED** = *rec*

where

*process* is the process number where the error occurred.

*block* is the block number where the error occurred.

*byte* is the byte number of the error.

*rec* is the nonzero value encountered.

**ERROR: TRANSFER BUFFER MISCOMPARE**

**PROCESS** = *process*, **BLOCK** = *block*, **BYTE** = *byte*,

**EXPECTED** = *exp*, **RECEIVED** = *rec*

where

*process* is the process number where the error occurred.

*block* is the block number where the error occurred.

*byte* is the byte number of the error.

*exp* is the expected value.

*rec* is the received value.

If Test 5 reports an error, the user should verify that the other SDT test suites run without error and then check the SDT309 configuration file to ensure that the process configuration is correct. If the error persists, the iSBC 309 board should be replaced.







This chapter contains detailed descriptions of the System Diagnostic Tests for mass storage components used in System 86/300 Series microcomputers. The following tests are described:

- SDTWIN, which tests the iSBC 215B and 215G Winchester Controllers and the associated Winchester drives.
- SDT218, which tests the iSBX 218 and 218A Flexible Diskette Controller boards.

### **SDTWIN TEST SUITE**

SDTWIN provides testing and fault analysis to aid the user in detecting and isolating problems with Winchester disk drives and the iSBC 215/220 controller boards. When invoking this test suite, remember that SDTWIN has two versions: SDTWIN8 (for 8-inch drives) and SDTWIN5 (for 5¼-inch drives). (Unless otherwise specified, throughout this chapter SDTWIN refers to both SDTWIN5 and SDTWIN8.)

Table 5-1 lists the SDTWIN tests and their functions. The tests and the procedures are described in detail later in the chapter.

Table 5-1. SDTWIN Tests

Test	Function
0	Reset/Initialize Disk Test
1	ROM Checksum Test
2	RAM Window Test
3	RAM Address Test
4	Transfer Status
5	Buffer I/O Test
6	Format Diagnostic Tracks Test
7	Microdiagnostic Test
8	Verify Format/Format Test
9	Seek/Verify Test
A	Worst-Case Seek Test

Table 5-1. SDTWIN Tests (Continued)

Test	Function
B	Read/Write/Verify Diagnostic Track Test
C	Drive Selection Test
D	Platter/Head Selection Test
E	Sector Selection Test
F	Overlap Seek Test
10	Alternate Track Test
11	Zero Fill Test
12	Data Overrun Test
13	Auto-Increment Test
14	Write All/Read/Compare Test
15	Random Write/Read/Verify Test
16	Select Next Drive Under Test
17	Format Utility
1A	Unload Heads for Shutdown
1B	Display/Edit Defective Track List Utility
1C	Display/Clear Error Log Utility

**SDTWIN INITIALIZATION**

When the user invokes SDTWIN, the test signs on with the test name and version. The SDTWIN5 prompts with the following:

**WHAT SIZE OF 5¼" WINCHESTER DRIVE DO YOU HAVE?**

- 0. 12Mb
- 1. 19Mb
- 2. 40Mb

**ENTER 0, [1], OR 2: \*r**

(**Note:** Throughout this chapter, "r" indicates a user response.)

Entering a carriage return selects the default setting (19Mb). (**Note:** Option 2 (40Mb) is not currently supported in the System 310.) This question is not asked in SDTWIN8.

For each reset software command, the following question is asked after the sign-on message:

**DO YOU WANT VARIABLES RESET TO THEIR DEFAULT VALUES? (Y/[N]) \*r**

If the user answers yes to this query, the routine asks for the 5¼-inch Winchester drive size (same prompt as listed above for the SDTWIN5). If the user answers no to this query, then the current

controller board wakeups (port addresses) are displayed. The user is now asked the following question:

**DO YOU WANT TO CHANGE THE CURRENT WAKEUP ASSIGNMENTS? (Y/[N]): \*r**

If the user answers yes to this query, the routine asks for the new wakeup assignments. After the new wakeup assignments are specified, the wakeup addresses of the controller boards are again displayed and the test repeats the preceding question. The test continues to display the current wakeups until the user answers no to the question, at which time the reset procedure executes a controller response test. Results of the response test are displayed for each controller. If none of the controllers respond, the system displays the following error message:

**NO RESPONSE**

There are two possible causes for a no response error: 1) a defective controller board or 2) no controller board installed in the system.

When the reset routine encounters a no response error during this test, the controller response test is repeated until at least one controller board responds. After the controller response test passes, the characteristics for any drives connected to the responding controller are displayed.

The routine now prompts the user for changes to the drive characteristics.

**DO YOU WANT TO CHANGE DRIVE CHARACTERISTICS? (Y/[N]) \*r**

If the user answers yes to this question, SDTWIN displays the current drive characteristics and prompts the user for new parameters. The question will be repeated until a response of no is given.

If the user answers no to this question, the current drive characteristics are displayed, and the reset procedure performs a drive response test on all drives. For drives that display a characteristic of 0 total cylinders, the drive response test assumes that these drives are unused or unattached and displays a "not used" message. When at least one drive responds for each controller, test results are displayed and the user is asked:

**DO YOU WANT ALL RESPONDING DRIVES TESTED (Y/[N]) \*r**

Answering yes to this query causes all responding drives to be tested. A no response to this question initiates individual questions for testing each drive.

The routine then asks:

**ARE ANY DRIVES BACKED UP? (Y/[N]) \*r**

If the user responds no to this question, any tests within the SDTWIN suite that are destructive to user data cannot be performed.

An affirmative answer allows all tests to execute, including tests that destroy user data. A warning message is displayed and the question is repeated as follows:

**WARNING: A DRIVE BACKED UP MAY BE ERASED.  
ARE YOU SURE DRIVE(S) ARE BACKED UP? (Y/[N]) \*r**

For each drive responding to the drive response test, the routine asks the user:

**IS iSBC-xxx (yyy) DRIVE n BACKED UP? (Y/[N]) \*r**

Where

xxx = controller number  
yyy = wakeup address for that controller  
n = unit number  
r = user's response

If the user responds no to this question, any tests within the SDTWIN suite that are destructive to user data cannot be performed.

If the user answers yes to the question

**DO YOU WANT TO CHANGE DRIVE CHARACTERISTICS?  
(Y/[N]) \*r**

SDTWIN displays the current drive characteristics and prompts the user for new parameters. After entering new values, the routine repeats the preceding question, and continues to do so until the user answers the question with no. Once the question receives a negative reply, SDTWIN performs the drive response test and prompts the user for the drives to be tested and backup status. To change the drive characteristic information, the user should know the following information about his drive:

- Wakeup
- Drive number
- Total number of cylinders
- Number of fixed heads
- Number of removable heads
- Number of sectors per track
- Number of bytes per sector
- Total number of alternate cylinders

This information can usually be found in the hardware reference manual that came with the drive or system.

**Using the RESET Command**

The user reset procedures (reset software and reset hardware) allow the user to change the board or drive parameters for SDTWIN if the board configuration has been changed. After changing the board configuration, the user can invoke SDTWIN and implement the reset procedures (enter "reset" or "res") to change the board or drive parameters. When the reset commands are invoked, SDTWIN displays the same sequence of questions as it does at initialization.

The SDTWIN procedure queries the user for correct board parameters and, where appropriate, provides a list of the acceptable parameter values. Some queries specify a default or current parameter value by enclosing the value in brackets. The user can select this value by entering a single carriage return.

**TEST DESCRIPTIONS**

This section describes each test and utility within the SDTWIN test suite.

**SDTWIN TEST 0, RESET/INITIALIZE DISK TEST**

This routine verifies the controller's ability to respond. It sets the channel controller block busy flag to busy and resets the controller under test. The channel controller busy flag is then tested for a nonbusy condition. If the flag fails to change to nonbusy within one second, the test fails. Provided the controller under test responds, the controller initializes four times, once for each selected drive. If a hard error is detected from the controller while initializing any drive, the test fails.

**SDTWIN TEST 1, ROM CHECKSUM TEST**

This routine verifies that the controller firmware is valid. It invokes the controller on-board ROM checksum test that calculates the checksum of its own firmware by adding the contents of all ROM locations (excluding the stored checksum). It then compares the calculated checksum value to the original checksum stored in the ROM. Each ROM has its own byte checksum, providing one checksum for the even bytes and another checksum for the odd bytes. If the test detects a checksum error, the controller reports an error condition and this test fails.

**SDTWIN TEST 2, RAM WINDOW TEST**

This routine verifies that all of the controller's RAM is functioning properly. It forces the controller to execute code from the MULTIBUS interface for a RAM window test. The RAM window test

sets all of RAM to zero except one word that is set to all ones (0FFFFH). This value is walked through all of memory. After every move, all of RAM is checked to see if the word of all ones walked through memory. The entire process is repeated with all of memory set to ones, and a word of zeros is walked through memory. If a RAM window test error is detected, the test fails.

#### **SDTWIN TEST 3, RAM ADDRESS TEST**

This test ensures that the controller's RAM address lines are working and that each bit of RAM can be turned on or off. It forces the controller to execute code from the MULTIBUS interface for a RAM address test that tests the RAM on-board the controller. The RAM address test copies the contents of on-board ROM into on-board RAM and then compares them. The contents of RAM are then inverted and added to the contents of ROM, which should produce a value of zero. If a RAM address test error is detected, the test fails.

#### **SDTWIN TEST 4, TRANSFER STATUS TEST**

This test checks the basic handshaking of the controller and makes sure the controller can properly retain drive information. It instructs the controller to transfer the error status information from its 12-byte error status buffer for the drive under test into a data buffer in host memory. If the status information does not transfer or contains error information, the test fails.

#### **SDTWIN TEST 5, BUFFER I/O TEST**

This test verifies that data transfers to and from the controller and host are possible. It uses two 600H-byte buffers in host memory: a write buffer and a read buffer. The test fills the write buffer with a counting pattern between 0 and 600H and the read buffer with all zeros. The controller loads the contents of its internal memory into the read buffer and the contents of the write buffer into internal memory. The test now compares the read and write buffers against each other and if it detects errors in the controller error status information or the read and write buffers miscompare, the test fails.

#### **SDTWIN TEST 6, FORMAT DIAGNOSTIC TRACKS TEST**

This test verifies that the unit under test can properly format a track. It formats one of the tracks on the diagnostic cylinder (last cylinder) and then verifies the format. The diagnostic track (head) selected for the format is the first diagnostic track not listed in the defective track list starting with head 0. If the defect list specifies all the diagnostic tracks, the test fails.

The format verification checks for proper cylinder and head, granularity, track type, and pattern formatted by using the read sector ID and read data functions. Any discrepancies in the verification cause the test to fail.

**SDTWIN TEST 7, MICRODIAGNOSTIC TEST**

The diagnostic track (last cylinder, head 0) is formatted because the microdiagnostic (on-board the controller) expects this track to have been formatted.

This test invokes the on-board controller microdiagnostic. The microdiagnostic test starts with a seek to the diagnostics track (last cylinder, head 0, sector 0) and performs a read ID to verify the track position. It then writes and reads sector 0 with a 55AAH data pattern and verifies that the data read matches the data written. If the routine encounters an error, the controller reports an error condition and this test fails. The microdiagnostic performs a quick GO/NO GO self-diagnostic test that verifies internal data and status electronics and checks position and read/write electronics in the disk drive under test.

**SDTWIN TEST 8, VERIFY FORMAT/FORMAT TEST**

This test does the following:

1. Verifies the format of cylinder 0, head 0 for proper cylinder and head, granularity, and track type.
2. If the format of cylinder 0, head 0 verifies, then all remaining data tracks are verified. A data track is defined as any track except a track in the last two cylinders or a track in an assigned alternate cylinder.
3. If the format of cylinder 0, head 0 does not verify, a format procedure may occur. An actual format is not performed unless the drive is backed up. A warning message is displayed and the test 17 formats the cylinder. After a format function, all data tracks are verified.
4. Test failures can occur for the following reasons:
  - The format of cylinder 0, head 0 does not verify and the drive is not backed up
  - The format of cylinder 0, head 0 verifies but some other data track will not verify
  - After formatting, some data tracks will not verify

**SDTWIN TEST 9, SEEK/VERIFY TEST**

This test expects both the diagnostic track and track 0 to be formatted before the test will pass. The test seeks to the diagnostic track of the disk drive under test and instructs the controller to read data into the controller buffer and verify the function on sector 0. The controller checks the ECCs and verifies the sector ID and data field for the sector. The test repeats the process for the first cylinder, head 0, sector 0. The test fails when a discrepancy in checking the ECCs, sector ID, or data fields occurs.



**SDTWIN TEST A, WORST-CASE SEEK TEST**

This test exercises the disk drive and controller seek mechanisms. It exercises the seek mechanism of the drive under test by performing various sequences of seeks on the drive. The algorithm used seeks tracks 0, 1, 0, 2, 0, 3, etc. to the last data track. It then seeks data tracks last, last -1, last -2, etc. to the first data track. Finally, it seeks data tracks center, center +1, center -1, center +2, center -2, etc. until all data tracks are covered. After each seek, the sector ID is read. The test fails when a discrepancy in the sector ID occurs. During this test, expect some vibration in the immediate area of the drive enclosure.

**SDTWIN TEST B, WRITE/READ/VERIFY DIAGNOSTIC TRACK TEST**

This test verifies the internal data and status electronics and checks position and read/write electronics in the disk drive under test from the MULTIBUS interface. This test expects the diagnostic track to be formatted before it can run without errors. The test is functionally equal to the microdiagnostic test except all testing instructions come from the host. The test starts with a seek to the diagnostics track and performs a read sector ID to verify the track position. It then writes and reads sector 0 with a 55AAH data pattern and verifies that the data read matches the data written. After each step in the test is executed, the controller's operational status byte and error status buffer are examined for an error. The data read is also checked for a 55AAH pattern. The test fails if an error cannot be recovered.

**SDTWIN TEST C, DRIVE SELECTION TEST**

This test checks the drive selection circuitry; it requires more than one drive to be attached to the controller under test. The routine writes a unique data pattern on the diagnostic track of each drive attached to the controller under test. The test reads back the data pattern and compares it against the pattern written. If the write and read data patterns do not match, the test fails. The test also fails because of hard errors detected by the controller.

**SDTWIN TEST D, PLATTER/HEAD SELECTION TEST**

This test formats the first and last tracks on the diagnostic cylinder (last cylinder) with a unique data pattern. The data is then read from the tracks and checked for the pattern written. If the write and read data patterns do not match, the test fails. The test also fails due to any hard error conditions detected by the controller that cannot be recovered. This test is expected to check the platter and head selection circuitry.

**SDTWIN TEST E, SECTOR SELECTION TEST**

This test verifies that each sector of a track can be addressed uniquely. It writes the sector number into each sector of head 0 on the diagnostic cylinder and then reads back each sector number and checks it for accuracy. If a sector number read does not agree with the actual sector number, the test fails. The test also fails because of hard errors detected by the controller.

**SDTWIN TEST F, OVERLAP SEEK TEST**

This test verifies that the controller can properly handle overlapped operations when more than one drive is attached to the controller; it requires at least two drives be attached to the controller under test, hereafter called drives A and B. The test starts by seeking to the last diagnostic track of each drive and then initiating a seek on drive A to track 0. Before the seek can finish, the test checks the sector ID on drive B. After that, the routine confirms a completed seek on drive A and checks the sector ID. The test fails when a sector ID is incorrect or a seek does not complete. The test also fails because of hard errors detected by the controller.

**SDTWIN TEST 10, ALTERNATE TRACK TEST**

This routine verifies that the alternate track function operates correctly. The test formats the first diagnostic track on the drive under test as an alternate track. It then formats the last diagnostic track as a defective track that points to the first diagnostic track. The test reads the first sector of the last diagnostic track, which points to the alternate track. Then, from the alternate track, the test reads the sector ID. The routine checks that the data returned by reading the sector ID specifies that the last cylinder and head accessed were from the alternate track. Finally, the test verifies that the flag field indicates that the last track accessed is an alternate track.

**SDTWIN TEST 11, ZERO FILL TEST**

This test checks the zero fill capabilities of the controller. The routine performs writes of various lengths (nonintegral sectors less than one sector) to the diagnostic track (last cylinder, head 0, sector 0). After each write, a read/compare operation (of the entire sector in each case) checks to verify that the data read matches the data written and that the remainder of the sector is filled with zeros. If the data read does not match the data written or if the remainder of the sector is not filled with zeros, the test fails.

**SDTWIN TEST 12, DATA OVERRUN TEST**

This test checks for a possible data overrun error condition. The routine writes data to the diagnostic track (last cylinder, head 0,

sector 0) and then performs reads of various lengths (nonintegral sectors less than one sector) to verify that the data read is valid and that no data overrun occurred. This test assumes that if a data overrun occurs, it destroys the data immediately following the buffer into which the data is being read. If this happens, the test fails.

#### **SDTWIN TEST 13, AUTO-INCREMENT TEST**

This test checks the auto-increment feature of the controller under test. It performs two one-sector writes: one to sector zero of the diagnostic track and one to the last sector. The test follows this with a single sector read to verify that the second sector of data was actually written to the correct sector. If not, the test fails. Two sector reads are then performed to verify that they also yield the correct data. If not, the test fails. Since only one cylinder is allocated to diagnostic use, the auto-cylinder increment cannot be checked.

#### **SDTWIN TEST 14, WRITE ALL/READ/COMPARE TEST**

All but the last two cylinders must be formatted before this test can run without errors.

This routine tests the entire data area and ensures the unique addressability of every sector on the drive under test. The test first writes a unique data pattern to all data sectors on the drive under test. It then goes back and performs a read of each data sector. After the test reads each sector, it checks the sector for the unique data pattern. If the pattern read does not match the pattern written, the test fails. The unique data stored on each sector includes:

- 10 bytes of unique ID for each sector on the drive under test,
- 10 bytes of 6DB6H followed by
- 10 bytes of the complement of 6DB6H

The pattern repeats until it fills the sector. Additionally, the 6DB6H word used in the pattern is extracted from V(5). The pattern can be changed by changing the value in V(5).

When desired, this test can perform the read after each write instead of writing all data first and then reading it back. To perform read after writes, the user must set V(6) to a nonzero value. The default of V(6) is 0.

This routine can be forced to test only a particular cylinder and/or head. The user can specify this option by putting the cylinder number in V(3) and/or the head number in V(4). The default values in V(3) and V(4) are 0FFFFH which indicates to do all cylinders and heads.

#### **CAUTION**

This test is destructive to user data and cannot be run unless the unit is backed up.

**SDTWIN TEST 15, RANDOM WRITE/READ/VERIFY TEST**

This test expects all but the last two cylinders to be formatted before it can run without errors. The routine runs for about five minutes, using random cylinders, heads, and sectors. Testing consists of writing random data, reading the data, and comparing data read with data written. If the data read and data written do not match, the test fails. This test complements the write all/read/compare test, but performs a quicker checkout of the same circuitry. The major difference is that this test does not check every single track on the disk under test.

**CAUTION**

This test is destructive to user data and cannot be run unless the unit is backed up.

**SDTWIN TEST 16, SELECT NEXT DRIVE UNDER TEST**

This test increments V(0) through V(2) to point at the next controller or drive under test. For example, suppose a system has one controller and three drives to test and drive 0 is currently under test. This test would increment the drive under test to drive 1. If drive 2 is under test then this test rotates the drive under test back to drive 0. When multiple controllers are in a system, this test increments to the next controller when V(0, 1, 2) are pointing to the last drive attached to the controller under test.

In a system equipped with three drives, a user tests all three drives by entering:

**T REP 3**

This instructs a diagnostic or test monitor to repeat all tests three times. Thus, the first loop tests the first drive and executes this test, which increments V(0,1,2) to point to the second drive. The second loop tests the second drive and executes this test, which increments V(0,1,2) to point to the third drive. The third and last loop tests the third drive and this test rotates V(0,1,2) to point back to the first drive. The diagnostic or test monitor Summary command indicates a collective total of test trials and failures for all drives. The error log shows individual errors applicable to each drive.

**SDTWIN TEST 17, FORMAT UTILITY**

This utility formats the data tracks and alternate tracks on the drive under test, provided the drive is backed up. It will not format over any of the defective track information. Additionally, known defective tracks are not assigned alternates unless V(A) default value is 0; alternates are not assigned unless the user changes the value in

V(A) to something other than zero. Alternates are assigned from the last data track downward when required.

#### NOTE

The following tests provide an interface to commonly used utilities. Each of these tests is initially ignored.

#### **SDTWIN TEST 1A, UNLOAD HEADS FOR SHUTDOWN**

This utility positions the heads on all drives under test to the diagnostic cylinder in preparation for shutdown (power off).

#### **SDTWIN TEST 1B, DISPLAY/EDIT DEFECTIVE TRACK LIST UTILITY**

This utility permits the user to see, add, change, or delete entries in the defective track list on the last cylinder minus 1. To display or edit the track list, the user can read the list from the disk and display it. After making any changes, the list must be written back to the disk, otherwise any changes made are lost. If the disk does not have a defective track list, the user can create a null defective track list in the same way.

#### **SDTWIN TEST 1C, DISPLAY/CLEAR ERROR LOG UTILITY**

This utility permits the user to see and/or clear any or all of the error logs.

#### **ERROR MESSAGES**

There are four types of errors that can cause SDTWIN to fail:

- No response errors -- displayed when the controller and/or drive does not respond within its given worst case time
- Hard errors -- displayed when a controller reports a nonrecoverable error condition
- Soft errors -- displayed when a controller reports a recoverable error condition
- Exception errors -- displayed when the controller/drive responds and no errors are reported but something obviously is wrong, such as data read differs from data written

When an error occurs during testing, information about that error is stored in an error log; a separate log is maintained for each type of error. The error logs reside in RAM and help the user isolate problems with the Winchester subsystem.

Soft errors encountered during testing do not cause a test to fail. They are only displayed to let the user know when a soft error occurs. Excessive soft errors could be a sign of impending nonrecoverable errors. The error count for each unique soft error is shown in the soft error log.

#### NO RESPONSE ERROR MESSAGES

A typical no response error message looks like this:

```

NO RESPONSE ERROR: iSBC 220 (100H) COUNT = 1T
DRIVE FAILURE COUNT
 0  1  2  3  TEST  FUNC  ST$SEMA  OP$STAT  BUSY$1
 1  0  0  0  8H   8H   0H         0H         FFH

```

This example shows that a no response error occurred while executing test 8 on drive 0 attached to the iSBC 220 board assigned to wakeup 100H. Additional information is also given for:

- FUNC (function)
- ST\$SEMA (status semaphore)
- OP\$STAT (operation status)
- BUSY\$1 (busy 1 flag)

These are specific fields within the controller channel control block and I/O parameter block.

#### HARD ERROR MESSAGES

A typical hard error message looks similar to this:

```

HARD ERROR:
iSBC WAKEUP UNIT CYLDR HDOP$STAT FUNC ERRSTAT TEST
220 100      0H  0H    0H C1H   8H   4000H   8H

TRIAL ERRCNT
1H    1H

```

#### OPERATION STATUS (OP\$STAT) EXPLANATION:

OPERATION COMPLETE

HARD ERROR

INITIATE TRACK SEEK FUNCTION (FUNC)

#### HARD ERROR STATUS (ERRSTAT) EXPLANATION:

SELECTED UNIT NOT READY

This display shows a hard error occurred on iSBC 220 at wakeup 100H, unit 0, cylinder 0, head 0 while executing test 8. The description of the error is given by OP\$STAT (operation status), FUNC (function), and ERRSTAT (error status). These are fields within the I/O blocks of the controller and are explained in lines 4

through 9 of the display. As shown, the operation completed with a hard error per the OP\$STAT, the FUNC was an initiate track seek function, and the ERRSTAT indicates the selected unit was not ready. Further, the error occurred during trial 1 of test 8 and the error count (ERRCNT) is one (first time for this error).

#### **SOFT ERROR MESSAGES**

A typical soft error message looks identical to a hard error message except the heading reads "SOFT ERROR", the ERRSTAT describes the soft error status, the explanation of the ERRSTAT is for a soft error, and the accumulated number of retries is shown. The retry count is an accumulated total of automatic retries by the controller and user-initiated retries. To reset the accumulated total to zero, the user should invoke SDTWIN routine 1C. The user can specify the number of retries allowed by changing the value of the V-variable V(C), which has a default value of 10 (0AH).

#### **EXCEPTION ERROR MESSAGES**

An exception error message indicates a condition exists that prevents the test from passing. Exception errors are not reported as errors by the controller or drive, but they do cause the test to fail. Additional information shown is identical to a hard error display.

#### **ERROR LOGS**

Just as there are four different types of errors and error messages, there are four different error logs:

- No response error log -- contains information about errors that occurred because the controller and/or drive failed to respond during the response test
- Hard error log -- contains information about errors that are nonrecoverable even after retries
- Soft error log -- contains information about errors that are recoverable
- Exception log -- contains information about errors not reported as errors by the controller or drive

#### **NO RESPONSE ERROR LOG**

The no response error log provides the user with the following information:

- The number of the controller or drive that failed to respond to the response test

- The controller response failure count which tells the user how many times the specified board has failed the response test
- The drive response failure test which tells the user how many times the specified drive has failed the response test

Each time the controller or drive response test is executed and a unit fails, the appropriate failure count is incremented. Resetting the system does not clear the failure counts and the counts continue to increment until the user invokes SDTWIN test 1C to clear the log.

#### **HARD ERROR LOG**

The hard error log contains information about errors that have occurred and the user has tried to recover without success. This log holds a maximum of 20 entries; if more than 20 entries are necessary, each new entry replaces the last entry placed in the log. The original 19 entries are retained until the user invokes SDTWIN test 1C.

This log provides the following information:

- Unit, cylinder, and head number where the error occurred
- Operational status of the unit
- Function code
- Error status
- Test number
- Number of times user retried error recovery
- Test trial
- Error count

#### **SOFT ERROR LOG**

The soft error log contains information about errors that have occurred and that the user successfully recovered. This log can hold a maximum of 20 entries; if more than 20 entries are necessary, each new entry replaces the last entry placed in the log. The original 19 entries are retained until the user invokes SDTWIN test 1C to clear the log.

This log provides the following information:

- Unit, cylinder, and head number where the error occurred
- Operational status of the unit
- Function code



- Test number
- Test trial
- Error count

**EXCEPTION LOG**

The hard error log contains information about errors that have occurred and that the user has tried to recover without success. This log holds a maximum of 20 entries; if more than 20 entries are necessary, each new entry replaces the last entry placed in the log. The original 19 entries are retained until the user invokes SDTWIN test 1C to clear the log.

This log provides the following information:

- Unit, cylinder, and head number where the error occurred
- Operational status of the unit
- Function code
- Error code
- Test number
- Number of times user retried error recovery
- Test trial
- Error count

The information displayed by the exception log is essentially the same as the hard error log. However, rather than an error status, the exception log displays an error code. The meaning of the error codes is as follows:

<b>CODE</b>	<b>MEANING</b>
1	Transfer count different than expected
2	More retries than allowed
3	Actual cylinder or head number different than expected
4	Actual sector size different than expected
5	Track format type different than expected
6	Pattern formatted different than pattern read
7	Defect list shows all diagnostic tracks defective
8	Too many defective diagnostic tracks per defect list
9	Pattern written different from pattern read
10	Cannot do test with current defect list
11	Error detected during overlap seek

CODE	MEANING
12	Pattern read is not properly filled with zeros
13	More data transferred than requested
14	Test needs unit backed up to execute
15	Track 0 has a failure that is not allowed
16	Ran out of alternate tracks to assign
17	Defect list is not valid
18	Two or more drives per controller needed for test to execute
19	An interrupt was expected but not received
20	Status semaphore never indicated status to be posted as expected
21	Hard error detected with initialization function
22	Controller does not respond from a reset

### SDT218 TEST SUITE

The SDT218 test suite contains nine tests and 40 utility routines for the iSBX 218 and 218A Flexible Diskette Controller boards. A test is provided for each major area of the flexible diskette drive system. The utilities perform single functions such as reading or writing selected sectors on the diskette. Table 5-2 lists the tests and their functions.

Table 5-2. SDT218 Tests

Test	Function
0	Format Test
1	Seek/Verify Test
2	Write/Read Test
3	Drive Selection Test
4	Side/Head Selection Test
5	Sector Selection Test
6	Track Verify Test
7	Diskette Verify Test
8	Write/Read Deleted Data

**SDT218 INITIALIZATION**

The user can load the SDT218 test suite into the system as described in Chapter 3. Upon initial loading, SDT218 displays the following message:

**SYSTEM DIAGNOSTIC TEST - 218, Vx.y**  
**Copyright 1983 Intel Corporation.**

where x.y is the version number.

After displaying the initial message, SDT218 asks a series of questions about the hardware characteristics and the extent of the testing to be done. SDT218 pauses after each question for a user response. Current (or default) settings are enclosed in brackets. For example, in the following question:

**SPECIFY DECIMAL NUMBER OF TRACKS PER SURFACE:**  
**[77T]**

The user can either enter the decimal number of tracks on the diskette (followed by a carriage return) or accept the default value, 77, if appropriate. The default value can be selected by entering a carriage return.

The user can back up to the previous question by entering a "b" in response to any of the questions (except the last).

The first SDT218 prompt is as follows:

**ENTER CODE OF DEVICE TO BE TESTED:**  
**(0) DEFAULT 8" FLOPPY**  
**(1) DEFAULT 5.25" FLOPPY**  
**(2) OTHER**  
**[0] \*r**

(Note: Throughout this chapter, r is the user's response.)

If the user chooses (0), the SDT218 assumes that the unit being tested is an 8-inch flexible diskette drive, and that it is the only such drive present. If the user chooses (1), the SDT218 assumes that the unit being tested is a 5¼-inch flexible diskette drive, and that it is the only such drive present.

Note that entering a response of (0) or (1) ends the reset software dialogue. Refer to the list below for default parameters for options (0) and (1).

**CAUTION**

If the user chooses (0) or (1), SDT218 automatically assumes that the unit to be tested is backed up. Some test routines in SDT218 are destructive to user data. When specifying a default device (options 0 or 1), a blank double-sided, double-density diskette must be used to run this test suite.

**DEFAULT PARAMETERS**

<b>PARAMETER</b>	<b>5¼-INCH DRIVE</b>	<b>8-INCH DRIVE</b>
HOST	iSBC 215G	iSBC 215G
MEDIA FORMAT	DS/DD	DS/DD
UNIT TESTED	0	0
TRACKS/SURFACE	40	77
SECTORS/TRACK	8	26
BYTES/SECTOR	512	256
BACKED UP	YES	YES

If the user chooses (2), the procedure displays the following:

```
** ASSUMING NO-DEFAULT DEVICE **
ENTER CODE FOR HOST BOARD:
  (0) iSBC 215
  (1) iSBC 86/30
[0] *r
```

If the user chooses (0), the test assumes that the iSBX 218's host board is an iSBC 215 Winchester Disk Controller board. If the user chooses (1), the test assumes that the host board is the iSBC 86/30 Single Board Computer.

Next, SDT218 asks the following two questions:

```
ENTER CODE FOR MEDIA SIZE:
  (0) 8"
  (1) 5.25"
[0] *r

ENTER CODE FOR MEDIA TYPE:
  (0) DOUBLE-SIDED, DOUBLE-DENSITY
  (1) SINGLE-SIDED, DOUBLE-DENSITY
  (2) SINGLE-SIDED, SINGLE-DENSITY
[0] *r
```

SDT218 asks the following question for units 0 through 3 (where  $n=0, 1, 2, 3$ ):

```
IS UNIT  $n$  BEING TESTED? (Y OR [N]) *r
```

After the user selects the unit to be tested, the procedure asks

```
IS THIS UNIT BACKED UP? (Y OR [N]) *r
```

**CAUTION**

If the user answers yes to this question, some tests in this test suite will write on the diskette under test, possibly destroying data.

SDT218 next asks about diskette characteristics:

**SPECIFY DECIMAL NUMBER OF TRACKS PER SURFACE:**

[nT]

**SPECIFY DECIMAL NUMBER OF SECTORS PER TRACK:**

[nT]

**SPECIFY DECIMAL NUMBER OF BYTES PER SECTOR:**

[nT]

where *n* is the current setting.

**ENTER A 1 TO 5 DIGIT DECIMAL RANDOM NUMBER SEED [57]\***

Any number may be entered; this value is used to derive a random number used during testing.

The final question that SDT218 asks is

**DO YOU WISH TO USE THE UTILITY TESTS? (Y OR [N]) \*r**

If the user answers yes, he can invoke all of the utilities (as Tests 09H through 33H) in addition to Tests 0 through 08H. If the user answers no, he will not have access to the utilities (unless SDT218 is reinitialized).

**Using the RESET Command**

If the user enters the RESET or RESET SOFTWARE command, SDT218 displays the same sequence of questions as at initialization. The RESET HARDWARE command initializes the hardware; it produces a "pass" or "fail" message, depending on whether the hardware initialization was successful or not. The user should use RESET HARDWARE whenever diskettes are changed in the drives.

**CAUTION**

Since flexible diskettes do not have diagnostic tracks, tests 0, 2, 3, 4, and 8 use the innermost data track as a diagnostic track. Any data on that track will be destroyed. These tests will not run unless the user answers yes to "Is this unit backed up?" at initialization or after invoking the RESET command.

**SDT218 TEST 0, FORMAT DIAGNOSTIC TRACK**

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

**SDT218 TEST 1, SEEK/VERIFY TEST**

First this test seeks to the diagnostic track and verifies the data records there. For the iSBC 86/30 board environment, data verification uses the diskette controller's read ID and read data commands; the iSBC 215 environment uses the Winchester controller's verify command to verify ECCs. After verifying the diagnostic track, the test seeks to track 1 and performs the same verification. Test 1 repeats this sequence for all recognized units.

**SDT218 TEST 2, WRITE/READ DIAGNOSTIC TRACK**

The Write/Read Diagnostic Track test transfers one sector of contiguous data from system memory to the diagnostic track. After writing the data, it performs a read and compares the contents of the write and read buffers. The user must format the diagnostic track before invoking this test. There are two methods for formatting the track: the diskette can be formatted using the iRMX 86 FORMAT command, or the format diagnostic track test (Test 0) can be used.

**SDT218 TEST 3, DRIVE SELECTION**

The Drive Selection test verifies that each attached device can be accessed. It does this in the following manner:

- Accesses all attached drives to determine if ready.
- Writes on the diagnostic track on each of the available devices with data unique to each attached device.
- Reads and compares the data read with the data written.

This test is initially "ignored" because it requires at least two attached drives of the same general type (for example, two 8-inch flexible diskette drives). Therefore, to use the test, the user must "recognize" it with the RECOGNIZE command.

**SDT218 TEST 4, SIDE/HEAD SELECTION**

The Side/Head Selection test verifies that each side and head can be uniquely addressed. First, the test accesses the diagnostic track of all attached devices and writes the track and head number in the ID

field of each attached device. It then reads the ID field of each attached device and compares the data read with that written.

#### **SDT218 TEST 5, SECTOR SELECTION**

The Sector Selection test verifies that each sector is uniquely addressable. The test first accesses the diagnostic track of each attached device and writes a unique sector number into the ID field of each sector. It then reads each sector and compares the contents with that written.

#### **SDT218 TEST 6, TRACK VERIFY**

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

#### **SDT 218 TEST 7, DISKETTE VERIFY**

In the iSBC 215 board environment, this test reads IDs from all sectors and uses the controller verify command. In the iSBC 86/30 environment, the test performs read ID and read data commands and checks the status registers for CRC errors. This test is nondestructive. The diskette must be formatted prior to running this test.

#### **SDT218 TEST 8, WRITE AND READ DELETED DATA**

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

#### **SDT218 UTILITIES FOR THE iSBC® 215 ENVIRONMENT**

This section describes the utility programs available for use when testing a system that has the iSBX 218 Flexible Diskette Controller mounted on the iSBC 215 Winchester Disk Controller.

The user can assign values to parameters used in these utilities by invoking the V command. (See Chapter 3.) For example, the following sequence:

```
*V(B) = 1
*V(C) = 250
*TEST 19
```

assigns values of 1 and 250 to V(B) and V(C), respectively. It then invokes Test 19, the Seek Test, causing the drive under test to seek to track 250 using head 1.

The utilities are invoked in the same way as the tests described previously in this section (i.e., using the characters TES or TEST, followed by a hexadecimal number). Many of the utilities require that certain conditions be set up before they are invoked. For example, the Seek Utility requires that the Select Unit Utility (Test 10) be invoked first to select the drive to be tested.

The first 10 utilities (Tests 9H through 12H) are "select" utilities whose primary purpose is to set up conditions for the other utilities; relevant select utilities are listed, where appropriate, in the utility descriptions.

**Test 9, Help:** provides information about using V(B) and V(C) variables.

**Test A, Select Unit:** uses V(B) to specify the default unit.

**Test B, Select Cylinder:** uses V(B) to specify the cylinder number. The user should choose a value less than or equal to the tracks per surface specified using the Reset Hardware command.

**Test C, Select Head:** uses V(B) to specify the head number. The user should choose a value less than or equal to the number of fixed surfaces specified using the Reset Hardware command.

**Test D, Set Sector Count:** uses V(B) to specify the sector count. The sector count multiplied by the number of bytes per sector must be less than or equal to 0FFFEH.

**Test E, Select Interrupt:** uses V(B) to specify the interrupt and retry modes.

**Test F, Select Read Buffer:** uses V(B) to specify the offset of BUFFER, the user's selected read buffer; the base remains the same (data segment).

**Test 10, Select Write Buffer:** uses V(B) to specify the offset of BUFFER, the user's selected write buffer; the base remains the same (data segment).

**Test 11, Increment Value:** increments the values for V(B) and V(C). V(B) is incremented by V(D) and V(C) is incremented by V(E).

**Test 12, Decrement Value:** decrements the values for V(B) and V(C) as described above for Test 11.

**Test 13, Reset:** resets the controller by issuing a reset command, an interrupt clear command, and a channel attention command.

**Test 14, Initialize:** initializes the controller and all units specified at initialization or using the RESET command.



**Test 15, Recalibrate:** causes a recalibration of the selected unit to cylinder 0. Relevant selection routine: UNIT.

**Test 16, Error Status:** transfers error status and sector address information from the controller and displays it, if any bits are set in the status register.

**Test 17, System to Controller Write:** writes from the write buffer to controller memory. V(B) contains the controller memory address, and V(C) contains the number of bytes to write. Relevant selection routine: WRITE BUFFER.

**Test 18, Controller to System Write:** transfers data from the controller's (buffer) memory to the selected read buffer. V(B) contains the controller address and V(C) contains the number of bytes to transfer. Relevant selection routine: READ BUFFER.

**Test 19, Seek:** causes the specified head to seek, on the selected unit, to the specified cylinder. V(B) contains the head number; V(C) contains the cylinder number. Relevant selection routine: UNIT.

**Test 1A, Read and Display ID:** reads and displays the track ID of the last track accessed on the selected unit. V(B) contains the value for the unit number. Relevant selection routines: CYLINDER, HEAD.

**Test 1B, Verify:** uses the controller's verify command on the selected unit and track. V(B) contains the cylinder number, V(C) contains the beginning sector number. Relevant selection routines: SECTOR COUNT, UNIT.

**Test 1C, Read:** reads sectors from the selected unit into the read buffer. V(B) contains the cylinder number; V(C) contains the beginning sector number. Relevant selection routines: SECTOR COUNT, UNIT, READ BUFFER.

**Test 1D, Read Deleted Data:** reads sectors of deleted data from the selected unit and track, storing it in the read buffer. V(B) contains the cylinder number, and V(C) contains the beginning sector number.

**Test 1E, Write:** writes data from the write buffer to sector(s) on the selected unit and track. V(B) contains the cylinder number, and V(C) contains the beginning sector number. Relevant selection routines: SECTOR COUNT, UNIT, WRITE BUFFER.

**Test 1F, Write Deleted Data:** writes data from the write buffer to sectors on the selected unit and track. The data written out is marked deleted. V(B) contains the cylinder number and V(C) contains the beginning sector number. Relevant selection routines: SECTOR COUNT, UNIT, WRITE BUFFER.

**Test 20, Write Controller Buffer:** writes to disk from controller memory. V(B) contains the cylinder number, and V(C) contains the beginning sector number. Relevant selection routine: SECTOR COUNT.

**Test 21, Alternate Seek:** causes a continuous seek to occur between two cylinders. V(B) contains the cylinder\$1 number and V(C) contains the cylinder\$2 number. The user must enter a CONTROL-C to stop this test.

**Test 22, Disk Seek:** exercises the seek and verify function; it moves the heads using the following algorithm:

```

DO FOREVER;
  n = first$track;
  m = last$track;
  REPEAT
    verify (n);
    verify (m);
    n = n + 1;
    m = m - 1;
  UNTIL n= last$track;
END;
```

Values for m and n are based on hardware. Relevant selection routine: UNIT.

**Test 23, Random Seek:** randomly seeks on the selected unit until the user enters CONTROL-C. Relevant selection routine: UNIT.

**Test 24, Format Track:** formats the selected track using bytes 0 through 3 of the selected write buffer for the format data pattern. V(B) contains the type of format (0 for data, 40H for alternate, and 80H for defective). V(C) contains the interleave factor. Relevant selection routines: UNIT, CYLINDER, WRITE BUFFER, READ BUFFER.

**Test 25, Format Platter:** formats the entire selected side (platter) of the selected unit. It uses the pattern specified by bytes 0 through 3 of the write buffer as the data pattern for the format operation. V(B) contains the type of format (0 for data, 40H for alternate, and 80H for defective). V(C) contains the interleave factor. Relevant selection routines: UNIT, HEAD, WRITE BUFFER.

**Test 26, Format Drive:** formats all platters of the selected drive. It uses bytes 0 through 3 of the selected write buffer for the pattern the controller is to use. V(B) contains the type of format (0 for data, 40H for alternate, and 80H for defective). V(C) contains the interleave factor. Relevant selection routines: UNIT, WRITE BUFFER.

**Test 27, Write Cylinder:** writes data on the selected unit, platter, and head. It then reads the data and compares it with the original data. V(B) contains the cylinder number and V(C) contains the beginning sector number. The head last accessed is used in this command. Relevant selection routines: UNIT, HEAD, WRITE BUFFER, READ BUFFER.

**Test 28, Track Check:** verifies that all track address bits can be addressed correctly. The test writes to the selected sector on all tracks for head 0 on the selected drive. The track number is written into every byte of each track's data field. When writing is complete, the data on each track is compared with the track number. V(B) contains the sector number. Relevant selection routine: UNIT.

**Test 29, Random Write:** writes to random sectors. V(B) contains the number of sectors. The data written includes sector, head, track, and a fill pattern. The sectors are read back and compared in the same order as written. Relevant selection routine: UNIT.

**Test 2A, Initiate Seek:** initiates a seek to a specified cylinder but does not wait for the operation to be completed. V(B) contains the cylinder number and V(C) contains the number of the head on the selected unit. Relevant selection routine: UNIT.

**Test 2B, Wait for Seek:** waits for the seek started by the Initiate Seek Utility to be completed.

**Test 2C, Pause:** generates a delay of V(B) times 0.1 millisecond.

**Test 2D, ECC:** calculates and displays the Error Checking Code (ECC) remainder for a specified block of system memory (in the same segment as the write buffer). V(B) contains the offset address and V(C) contains the number of bytes. If the count is less than four, no calculations are performed.

**Test 2E, Display Memory:** displays system memory, starting at the offset, for a specified count. V(B) contains the offset address and V(C) contains the count bytes. The test displays memory locations from the same segment as the read buffer (user's data segment).

**Test 2F, Compare Buffers:** compares the read and write buffers, byte-by-byte, for a specified count. V(B) contains the byte count. If DEBUG is TRUE, the routine displays any bytes that do not match. Relevant selection routines: WRITE BUFFER, READ BUFFER.

**Test 30, Fill Read Buffer:** places copies of a specified data word in the read buffer. V(B) contains the count for the number of copies, and V(C) contains the data word. Relevant selection routine: READ BUFFER.

**Test 31, Fill Write Buffer:** places copies of a specified data word in the write buffer. V(B) contains the count for the number of copies, and V(C) contains the data word. Relevant selection routine: READ BUFFER.

**Test 32, Display IOPB:** displays the iSBC 215 board's IOPB.

**Test 33, Clean Head:** may be used to clean the flexible diskette drive's heads. It loads the heads against the cleaning diskette for 30 seconds. A suitably prepared head cleaning diskette should be loaded in the selected unit before this test is invoked. V(B) specifies the number of the cleaning track to be used on the cleaning diskette.

**SDT218 UTILITIES FOR THE iSBC® 86/30 ENVIRONMENT**

This section describes the utility programs available for use when testing a system that has the iSBX 218 Flexible Diskette Controller mounted on the iSBC 86/30 Single Board Computer.

The user can assign values to parameters used in these utilities by invoking the V command. (See Chapter 3.) For example, the following sequence

```
*V(B) = 1
*v(C) = 250
*TEST 18
```

assigns values of 1 and 250 to V(B) and V(C), respectively. It then invokes Test 18, the Seek Test, causing the drive under test to seek to track 250 using head 1.

The utilities are invoked in the same way as the tests described previously in this section (i.e., using the characters TES or TEST followed by a hexadecimal number). Many of the utilities require that certain conditions be set up before they are invoked. For example, the Seek Utility requires that the Select Unit Utility (Test 10) be invoked first to select the drive to be tested.

The first 10 utilities (Tests 9H through 12H) are select utilities whose primary purpose is to set up conditions for the other utilities; relevant select utilities are listed, where appropriate, in the utility descriptions.

**Tests 9 through 12:** identical to Tests 9 through 12 for the iSBC 215 environment.

**Test 13, Reset:** initializes the iSBC 218 board by issuing the Reset, Sense Interrupt Status, and Recalibrate commands.

**Test 14, Initialize:** identical to Test 14 for the iSBC 215 environment.

**Test 15, Recalibrate:** identical to Test 15 for the iSBC 215 environment.

**Test 16, Sense Drive Status:** verifies that status can be sensed from drive 0. It initializes the 8272 Flexible Disk Controller chip on the iSBX 218 board with the drive parameters given in response to the RESET command prompts.

**Test 17, Sense Interrupt Status:** sends a sense interrupt status command to the 8272, then reads the result bytes in the 8272. This operation continues until an "all-clear" status is obtained, then a status message is displayed. This utility may be needed after a faulty calibration or seek operation, to ensure that the 8272 status is cleared.

**Test 18, Seek:** identical to Test 19 for the iSBC 215 environment.

**Test 19, Read and Display ID:** identical to Test 1A for the iSBC 215 environment.

**Test 1A, Verify:** emulates the iSBC 215 board Verify command by reading one byte from the specified sector and allowing the 8272 to check ID fields and CRCs. V(B) = cylinder, V(C) = beginning sector. Relevant selection routines: SECTOR COUNT, UNIT, READ BUFFER.

**Test 1B, Read:** identical to Test 1C for the iSBC 215 environment.

**Test 1C, Read Deleted Data:** identical to Test 1D for the iSBC 215 environment.

**Test 1D, Write:** identical to Test 1E for the iSBC 215 environment.

**Test 1E, Write Deleted Data:** identical to Test 1F for the iSBC 215 environment.

**Test 1F, Alternate Seek:** identical to Test 21 for the iSBC 215 environment.

**Test 20, Disk Seek:** identical to Test 22 for the iSBC 215 environment.

**Test 21, Random Seek:** identical to Test 23 for the iSBC 215 environment.

**Test 22, Format Track:** formats the selected track. It uses byte 0 of the selected write buffer for the data pattern that the controller writes. V(B) contains the interleave factor. Relevant selection routines: UNIT, CYLINDER, WRITE BUFFER, READ BUFFER.

**Test 23, Format Platter:** formats the entire selected side (platter) of the diskette in the selected unit. It uses byte 0 of the selected write buffer for the data pattern that the controller writes. V(b) contains the interleave factor. Relevant selection routines: UNIT, HEAD, WRITE BUFFER.

**Test 24, Format Drive:** formats the entire diskette in the selected unit. It uses byte 0 of the selected write buffer for the pattern that the controller writes. V(b) contains the interleave factor. Relevant selection routines: UNIT, WRITE BUFFER.

**Test 25, Write Cylinder:** identical to Test 27 for the iSBC 215 environment.

**Test 26, Track Check:** identical to Test 28 for the iSBC 215 environment.

**Test 27, Random Write:** identical to Test 29 for the iSBC 215 environment.

**Test 28, CRC:** verifies the CRC (cyclic redundancy check) and Read/Write circuitry. It first formats the diagnostic track and writes

4K bytes of data. Then it checks the 8272 Flexible Diskette Controller status to verify that the write operation was completed, and it reads the entire track back to make a comparison. An error is reported if the test finds a mismatch in the 4K data pattern that is not due to a media defect. The 4K byte write operation checks the auto-increment feature of the 8272.

**Test 29, Initiate Seek:** identical to Test 2A for the iSBC 215 environment.

**Test 2A, Wait for Seek:** identical to Test 2B for the iSBC 215 environment.

**Test 2B, Pause:** identical to Test 2C for the iSBC 215 environment.

**Test 2C, Display Memory:** identical to Test 2E for the iSBC 215 environment.

**Test 2D, Compare Buffers:** identical to Test 2F for the iSBC 215 environment.

**Test 2E, Fill Read Buffer:** identical to Test 30 for the iSBC 215 environment.

**Test 2F, Fill Write Buffer:** identical to Test 31 for the iSBC 215 environment.

**Test 30, Display IOPB:** displays the I/O Processor Byte (IOPB). Unlike the IOPB in the iSBC 215 board environment, this IOPB is used by the test software only.

**Test 31, Clean Head:** identical to Test 33 for the iSBC 215 environment.

#### SDT218 MESSAGES

Using the DEBUG command, the user can select either "pass/fail" error reporting or detailed error reporting for the SDT218 tests. Detailed error messages have the following general form:

```

error op
uu UNIT NUMBER
bbbbbbbb - OLD * CIB STATUS * NEW - bbbbbbbb
** error type **
status bits ** ERROR STATUS BITS **
message

```

	Desired	Actual
cyl	xxxx	yyyy
head	xx	yy
sect	xx	yy
Number of retries -		nnn

The meanings of the various fields within the error message are as follows:

*error op* is a message that indicates the type of operation that resulted in the error. The messages are as follows:

**DIAG ERROR** (The microdiagnostics returned an error.)

**FORMAT ERROR** (A format error occurred.)

**LBUF ERROR** (An I/O error occurred.)

**READ ERROR** (A read error has occurred.)

**READID ERROR** (A read sector ID error occurred.)

**RESET ERROR** (A reset error occurred.)

**SEEK ERROR** (A seek error occurred.)

**TRANST ERROR** (A transfer status error occurred.)

**VERIFY ERROR** (A verify error occurred.)

**WRITE ERROR** (A write error occurred.)

**WRTBUF ERROR** (An error occurred while writing the controller buffer to the disk.)

*uu* is the hexadecimal unit number of the device on which the error occurred.

*bbbbbbbb* are binary values that represent the value of the controller invocation block before and after the test used the transfer error status function to determine the type of error.

*status bits* is a binary value representing the contents of the error status bytes. For the iSBC 215 environment, this field takes the form

*hhhhhhh\$hhhhhhh sssssss*

where *hhhhhhh* is the hard error status and *sssssss* is the soft error status. For the iSBC 86/30 environment, this field takes the form

*iiiiiii\$jjjjjjj\$kkkkkkkkk*

or

*iiiiiii\$jjjjjjj\$kkkkkkkkk\$lllllll*

where *i*, *j*, *k*, and *l* represent the bits in status registers ST0, ST1, ST2, and ST3.

*error type* is one of the following messages:

**HARD ERROR REPORTED BY CONTROLLER**

**SOFT ERROR REPORTED BY CONTROLLER**

*message* is one of the following messages:

**8272 CONFUSED** (iSBC 86/30 environment only.)

**8272 TIMEOUT** (iSBC 86/30 environment only.)

**CYLINDER ADDRESS MISCOMPARE** (The ID field contains a cylinder address that is different from the expected cylinder address.)

**DATA FIELD ERROR** (The controller detected a correctable error in the data field of a sector. The user should examine the "number of retries" field of this message.)

**DIAGNOSTIC FAULT** (A Winchester controller microdiagnostic error occurred--iSBC 215 environment only.)

**DRIVE FAULT** (A fault in the specified drive occurred and is characterized by a read/write fault, a position fault, a power fault, or a speed fault.)

**END OF MEDIA** (The controller detected an end of media signal.)

**ID FIELD ERROR** (The controller detected a correctable error in the ID field of a sector. The user should examine the "number of retries" field of this message.)

**ILLEGAL FORMAT**

**ILLEGAL SECTOR SIZE** (The initialized sector does not agree with the formatted sector size.)

**INVALID ADDRESS** (The test attempted to access a cylinder beyond the range of available tracks, including alternates.)

**INVALID COMMAND** (The controller was issued an invalid function or parameter.)

**INVALID INTERRUPT** (iSBC 86/30 environment only.)



**MISSING ADDRESS MARK****NO DATA** (iSBC 86/30 environment only.)**NO INDEX****RAM ERROR** (The RAM test was not successful--iSBC 215 environment only.)**RATE ERROR****RESERVED****ROM ERROR** (The Winchester controller detected an error during the ROM test--iSBC 215 environment only.)**SECTOR NOT FOUND** (The controller could not find the specified sector.)**SEEK ERROR** (The controller detected a seek error.)**SEEK IN PROGRESS ERROR****SELECTED UNIT NOT READY** (The selected unit is either not ready, not connected, or not responding to attempts to connect it.)**WRITE PROTECTION FAULT** (The test attempted to write to a write-protected unit.)

*xxxx* is the cylinder, head, and sector location where the test intended to perform the operation.

*yyyy* is the cylinder, head, and sector location where the operation was actually performed.

*nnn* is the number of times the controller tried to perform the specified operation.

The following additional errors may be reported:

**BUSY ERROR**

The controller was not ready to start the operation. Resetting the controller is necessary; resetting the system may also be necessary.

**SEEK ERROR TIMOUT**

The seek complete interrupt did not occur within the allowed time-out period.

**NON-INTERRUPT TIME OUT**

The controller did not complete the operation within the allowed time-out period.

**INTERRUPT TIME OUT**

The controller interrupt did not occur before the allowed time-out period.

**CHANNEL *n* BUSY****CHANNEL *n* NOT BUSY**

where *n* is either 1 or 2. This message reports the state of the 8089 channel busy flags (iSBC 215 environment only).





## CHAPTER 6 COMMUNICATIONS SYSTEM DIAGNOSTIC TESTS

This chapter contains detailed descriptions of the System Diagnostic Tests for the optional communications boards available for System 300 Series microcomputers. Included are descriptions of the following System Diagnostic Tests:

- SDT351, which tests the iSBX 351 Serial Communications Expansion MULTIMODULE board
- SDT534, which tests the iSBC 534 Four Channel Communications Expansion board
- SDT544, which tests the iSBC 544 Intelligent Communications Controller board

### SDT351 TEST SUITE

The SDT351 test suite contains six tests for the iSBX 351 Serial Communications Expansion board. Table 6-1 lists the tests and their functions.

Table 6-1. SDT351 Tests

Test Number	Function
0	PIT Initialization
1	USART Initialization
2	USART Interrupts
3	Baud Rate Verification
4	Character Transmitter
5	Character Receiver

### SDT351 INITIALIZATION

When invoked, SDT351 displays the following message:

**SYSTEM DIAGNOSTIC TEST - 351, Vx.y**  
**Copyright 1982, 1983 Intel Corporation**

where x.y is the version number.

After the initial invocation message, SDT351 asks a series of questions about the hardware configuration, waiting after each question for your response. This prompting sequence allows you to modify the default settings for the tests to provide for variations in the iSBX 351 installation.

As each question is displayed, SDT351 encloses the default setting within brackets. For example, in the following question:

**LOOPBACK CONNECTOR INSTALLED ? (y or [n])**

you can enter y (yes) or n (no). In this case, n is the default setting. If you enter a carriage return alone, SDT 351 assumes the default setting. SDT351 indicates that it is ready for your response by displaying an asterisk just to the right of the question.

The sequence of questions is as follows:

**RESTORE HARDWARE VARIABLES TO DEFAULTS ?  
(y or [n])\***

where a "y" response causes all of the hardware settings described in this section to revert to their default values.

**CONTINUE TESTING [ iSBX Jx ] ? ([y] or n)\***

where x is the number of the iSBX connector on which the iSBX 351 board is installed (either J3 or J4). A "y" response to this prompt causes all further dialog and testing to refer to the port at the specified connector. An "n" response generates the prompt:

**NOW TESTING [ iSBX Jz] :**

where z is the number of the other iSBX connector. (For example, if x=3, then z=4.) All further dialog and testing subsequently refers to the port at the z connector.

**LOOPBACK CONNECTOR INSTALLED ? (y or [n])**

A loopback connector is a shorting plug that connects the transmit data output and receive data input pins together in order to loop back the output data to the input. A loopback connector is required for receive testing by all tests except the Character Receiver Test.

**NOTE**

You should install a loopback connector on the ports under test for the USART Initialization, USART Interrupt, and Baud Rate tests. Connect a terminal for the Character Transmitter and Character Receiver tests.

If you specify a loopback connector but do not install it, the USART Initialization and USART Interrupts tests fail. If you do not specify a loopback connector but do install it, the USART Initialization Test, USART Interrupts Test, and Character Transmitter Test will produce overrun errors. Refer to the *iSBX 351 Serial Communications Expansion MULTIMODULE Hardware Reference Manual* for details.

#### **ALTER INTERRUPTS CONFIGURATION ? (y or [n])**

A "y" response generates this prompt:

**TX INTERRUPT LEVEL [tt]\*  
RX INTERRUPT LEVEL [rr]\***

where *tt* is the current transmit interrupt level, with a range of zero to FFH, and *rr* is the current receive interrupt level, also with a range of zero to FFH.

The values for *tt* and *rr* can be equal. Since the interrupt controller only supports levels of zero to seven, if you specify a higher level, SDT351 interprets this as an "interrupts disabled" command. The test then masks off the interrupts for all tests except the USART Interrupts Test. If you do not install interrupt jumpers on both the iSBX 351 and the iSBC 86/30 boards to the levels you specified, the specification is meaningless. In these situations, the system can generate a "Tx (or Rx) INTERRUPT DID NOT OCCUR" message.

#### **ALTER TIMER CONFIGURATION ? (y or [n])**

A "y" response generates the following prompt:

**TIMERS 0 & 1 CASCADED ? (y or [n])**

The test supports cascaded timers for testing off-board uses of the system clock and timers. If you enter an "n," the following prompt appears:

**BAUD RATE TIMER JUMPER (0..2) [t]\***

where *t* is the number of the timer connected to the 8251A USART baud rate input clock. The timer number can be 0, 1, or 2. If the *t* value specified does not correspond to the physical jumper setting on the iSBX 351 board, all tests except the PIT test will fail. Note that the baud rate timer must not be cascaded.

A yes response to the previous question causes the following message to appear:

**\*\* TIMERS 0 & 1 CASCADED:  
\*\* BAUD TIMER RESET TO 2**

This message advises you that timers 0 and 1 are now connected in cascade and that timer 2 is the baud rate timer to be used by the test program.

**CLOCK FREQUENCY JUMPER (0..2)**  
**0=153.6K, 1=1.23M, 2=2.46M [c]**

where c is the current setting. This prompt allows you to change the iSBX 351 baud rate clock frequency. If you select the 153.6KHz clock, the maximum baud rate for all tests is 4800. Otherwise, the system (or your response to a RESET SOFTWARE prompt) specifies the baud rate.

After initialization is complete, SDT351 displays its prompt (\*), indicating that it is ready to accept commands.

**Using the RESET Command**

If you enter the RESET, RESET HARDWARE, or RESET SOFTWARE command, SDT351 asks a series of questions about the settings of certain variables to be used during testing. After each question, you can enter a new value for the variable or you can accept the present setting by simply pressing the RETURN key. This section describes the RESET SOFTWARE question sequence. The RESET HARDWARE questions are the same as those displayed at SDT351 invocation; these are described earlier in this section.

The RESET SOFTWARE questions allow you to modify the settings of variables used in the Character Transmitter and Character Receiver tests (Tests 4 and 5). All other SDT351 tests control their own software settings.

The sequence of questions is as follows:

**RESTORE SOFTWARE VARIABLES TO DEFAULTS ? (y or [n])\***

A "y" response causes all of the variables to assume their default values.

**ENTER NUMBER OF STOP BITS USING INDEX BELOW:**

**0= 1.0 STOP BITS**  
**1= 1.5 STOP BITS**  
**2= 2.0 STOP BITS [s]\***

where s is the current setting. This prompt allows you to change the number of stop bits in the asynchronous communication format.

**ENTER CHAR LENGTH AS FOLLOWS:**

**0= 5 BITS**  
**1= 6 BITS**  
**2= 7 BITS**  
**3= 8 BITS [c]\***

where c is the current setting. You can specify the number of bits needed to define a character. The actual number of bits transmitted

is the character length plus a parity bit, if parity is enabled, and one or two stop bits.

**ENABLE PARITY ? (y or [n])**  
**USE EVEN PARITY ? (y or [n])**

An "n" response disables parity checking. The second part of the prompt, USE EVEN PARITY, appears only if you respond with a "y" to the first prompt and allows you to select either odd or even parity.

The next prompt allows you to set the baud rate:

**ENTER BAUD RATE USING INDEX BELOW:**

0= 75 baud  
 1= 110 baud  
 2= 150 baud  
 3= 300 baud  
 4= 600 baud  
 5= 1200 baud  
 6= 2400 baud  
 7= 4800 baud  
 8= 9600 baud  
 9=19200 baud [b]\*

where *b* is the current setting. Only the listed baud rates are allowed. If you choose the 153.6 kHz clock frequency during initialization the maximum baud rate is 4800 bits per second.

**ECHO CHARS TO PERIPHERAL DEVICE [ v(1) ] ? (y or [n])\***

This question allows you to control whether input sent from the peripheral in the Character Receiver Test will be echoed back to the peripheral connected to the port under test.

If you also want the characters to be echoed at the system console (the terminal connected to the processor board's serial port) answer yes to the following question.

**ECHO CHARS TO SYSTEM CONSOLE SCREEN [ v(2) ] ?**  
**([y] or n)\***

You can also set the previous two variables by changing global variables V(1) and V(2) using the "V" test management command.

#### **SDT351 TEST 0, PIT INITIALIZATION**

This test initializes each of the three timers in the 8253 Programmable Interrupt Timer. The test performs the following sequence for each timer:

- Verifies the initialization of the three timer subtests
- Initializes and starts each timer



- Checks to see if each timer counted down
- Sequentially tests the timers with their interrupts disabled

If the test reads a value equal to the the one originally loaded into the timer, it assumes the timer is not running. This produces the following error message:

**ERROR: TIMER *t* DID NOT COUNT, MODE = *m***

where *t* is the timer number and *m* is the PIT mode byte. (Refer to the *iSBX 351 Serial Communications Expansion MULTIMODULE Hardware Reference Manual*.)

**TEST 1, USART INITIALIZATION**

Test 1 initializes the 8251A USART for various asynchronous modes of operation. For each mode tested, the test initializes the USART with the combination of stop bit, parity, character length, and baud rate parameters listed in Table 6-2.

In some cases the 5- and 6-bit character tests can cause a terminal to lock up. This lock-up condition can occur if the terminal interprets the characters received as control characters. If this happens, you can continue testing by resetting the terminal or turning the terminal off, then back on.

If you have connected a terminal rather than a loopback connector to the RS-232 port, the information displayed on the terminal may be garbled. This simply means that the modes being tested are not compatible with the terminal's operation.

Table 6-2. USART Modes of Operation

Test	Number of stop bits	Parity	Character Length	Baud Rate	Division Ratio
async0	1.0	disabled	8 bits	110	16
async1	1.0	disabled	8 bits	19200	16
async2	1.0	disabled	8 bits	9600	16
async3	1.0	disabled	7 bits	9600	16
async4	1.0	disabled	6 bits	9600	16
async5	1.0	disabled	5 bits	9600	16
async6	1.0	odd	8 bits	9600	16
async7	1.0	even	8 bits	9600	16
async8	1.5	even	8 bits	9600	16
async9	2.0	even	8 bits	9600	16

For each mode, Test 1 does the following:

- Initializes the USART
- Uses the USART to transmit a set of ten characters
- Checks the USART status after each character

If you do not specify a loopback connector during initialization, and the iSBX 351 board receives a character during the test, the test generates an "unexpected character received" error message.

If you do specify a loopback connector during initialization, the test also does the following:

- Checks the USART receive status
- Checks the character received against the character sent

The test uses different character sets for testing the different character length modes, so that it can perform receive validation.

This test can generate the following error messages:

**ERROR: ASYNCH MODE: SB=*s.s* LEN=*l*  
 PARITY = *x, y*  
 USART STATUS=*uu* BAUD=*bbbb***

where

- s.s* is the number of stop bits.
- l* is the number of bits per character.
- x* indicates parity is disabled (0) or enabled (1).
- y* indicates parity is odd (0) or even (1).
- uu* is the USART status byte (hexadecimal).
- bbbb* is the baud rate (decimal).

**ERROR: TRANSMIT FAILED: USART STATUS=*ss*  
 RECEIVE FAILED: CHAR SENT=*cc* RECEIVED=*dd***

**ERROR: UNEXPECTED CHAR RECEIVED WITHOUT  
 LOOPBACK CONNECTOR  
 USART STATUS=*ss* CHAR SENT=*cc* RECEIVED=*dd***

where

- ss* is the hexadecimal value of the USART status byte

*cc* is the hexadecimal value of the character sent byte

*dd* is the hexadecimal value of the character received byte

### **SDT351 TEST 2, USART INTERRUPTS**

Test 2 checks and verifies that the 8251A USART generates interrupts correctly. To test interrupts, the interrupt jumpers on both the iSBX 351 board and the host CPU board must be set, and you must enter values corresponding to these settings at SDT 351 invocation or after invoking the RESET HARDWARE command. In addition, to test the receive interrupts you must install a loopback connector.

The USART transmits a stream of 210 characters each for baud rates of 300, 9600, and 19,200. For each character, the test does the following:

- Checks transmit and receive interrupt counters
- Determines whether appropriate interrupts occurred
- Checks that the interrupts occurred only once

The error messages generated by Test 2 are:

**ERROR: RECEIVE INTERRUPT DID NOT OCCUR**  
USART STATUS=*ss* BAUD=*bbbb*

**ERROR: TRANSMIT INTERRUPT DID NOT OCCUR**  
USART STATUS=*ss* BAUD=*bbbb*

**ERROR: EXTRA RECEIVE INTERRUPT(S):**  
EXPECTED=*xxxx* ACTUAL=*aaaa*  
USART STATUS=*ss* BAUD=*bbbb*

**ERROR: EXTRA TRANSMIT INTERRUPT(S):**  
EXPECTED=*xxxx* ACTUAL=*aaaa*  
USART STATUS=*ss* BAUD=*bbbb*

**ERROR: UNEXPECTED CHAR RECEIVED WITH NO LOOPBACK CONNECTOR:**  
CHAR SENT=*cc* RECEIVED=*dd*

**ERROR: RECEIVE FAILED: CHAR EXPECTED=*cc***  
**RECEIVED=*dd***  
USART STATUS=*ss* BAUD=*bbbb*

**ERROR: USART STAYING UNREADY ON TRANSMIT**  
**USART STATUS=ss BAUD=bbbb**

**ERROR: BAD STATUS**  
**USART STATUS=ss**

where

<i>ss</i>	is the hexadecimal value of the USART status byte
<i>bbbb</i>	is the baud rate under test (decimal)
<i>xxxx</i>	is the number of interrupts expected (word)
<i>aaaa</i>	is the number of interrupts that actually occurred (word)
<i>cc</i>	is the hexadecimal value of the character transmitted
<i>dd</i>	is the hexadecimal value of the character received

#### **SDT351 TEST 3, BAUD RATE VERIFICATION**

Test 3 verifies that the iSBX 351 clock generator, PIT, and USART can accurately transmit at a specified baud rate. It compares the number of characters transmitted in one second with a benchmark number that corresponds to the specified baud rate. The test checks baud rates of 110 and 2400 bits per second.

The test initializes the timer with the appropriate count value for each baud rate and initializes the USART with the following values:

- One stop bit
- Parity disabled
- Eight-bit character length
- Asynchronous mode
- Transmit and receive interrupt levels disabled

The test then sets the processor fail-safe timer for one second and causes the USART to transmit characters as fast as its transmit-ready status can be restored. At the end of the one second interval, the test compares the number of characters transmitted to the benchmark value.

The test can generate the following error message:

**ERROR: USART STAYING UNSTEADY ON TRANSMIT**

**ERROR: WRONG BAUD RATE: ACTUAL=aaaa  
EXPECTED=bbbb**

where *aaaa* is the decimal number of bits received in one second and *bbbb* is the decimal number of bits expected in one second.

**SDT351 TEST 4, CHARACTER TRANSMITTER**

Test 4 acts as a character generator that sends a known character set to be displayed on a peripheral device. If you use a loopback connector (this must be specified at initialization), characters are read in and discarded after each transmission. If you do not specify a loopback connector at initialization and one is installed, the test will detect an overrun condition and report a "bad status" error. In normal operation, the test will report an error for only two reasons: bad status or a timeout while waiting for the transmit-ready status bit.

The test transmits the following character set:

ABCDEFGHIJKLMNOPQRSTUVWXYZ !"#%\*()  
abcdefghijklmnopqrstuvwxyz0123456789

Possible error messages are:

**ERROR: TRANSMIT FAILED, USART STATUS=ss**

**ERROR: BAD STATUS ON USART=ss**

where *ss* is the hexadecimal value of the USART status byte.

**SDT351 TEST 5, CHARACTER RECEIVER**

Test 5 allows you to control input to SDT351 directly from a peripheral. It is the only test that allows receive testing (other than through the loopback connector).

Test 5 forces the DEBUG flag TRUE so that the following prompt can be displayed.

**\* READY TO RECEIVE CHARS, SEND CTRL Z TO QUIT \***

The test receives characters from the peripheral device through the iSBX 351 serial port. The test displays the characters it receives according to the echo controls you set in response to the RESET SOFTWARE prompts.

If the iSBX 351 board does not receive a character within ten seconds, the test reports an error. The test continues until you send a CONTROL-Z (01AH) or ten seconds go by while waiting for the next character.

Test 5 error messages:

**ERROR: TIMEOUT WAITING FOR FIRST CHAR, USART  
STATUS=ss**

**TIMED OUT AFTER CHAR(S) RECEIVED, CTRL Z INSERTED**

where ss is the hexadecimal value of the USART status byte.

### **SDT534 TEST SUITE**

The SDT534 test suite contains tests for the iSBC 534 Four Channel Communications Expansion Boards. SDT534 consists of ten tests. Table 6-3 lists the tests and their functions.

Table 6-3. SDT534 Tests

Test Number	Function
0	PIT Initialization
1	USART Initialization
2	PIC Initialization
3	PPI Port C Verification
4	USART Interrupt Verification
5	PIT Timer 4 and 5 Interrupt Verification
6	PIC Baud Rate Generation Verification
7	USART Load Test
8	USART Character Transmitter Test
9	USART Character Receiver Test

### **SDT534 INITIALIZATION**

When invoked, the SDT534 test suite displays the following message, where x.y is the version number.

**SYSTEM DIAGNOSTIC TEST - 534, Vx.y  
Copyright 1982, 1983 Intel Corporation**

After the initial message, SDT534 asks a series of questions about the hardware configuration, pausing after each question to allow you to respond. Current (or default) settings are enclosed in brackets. For example, in the following question:

**ENTER NUMBER OF iSBC 534 BOARDS IN THE SYSTEM [1] \***

the 1 displayed within brackets is the current setting for the number of iSBC 534 boards in the system. Enter a carriage return alone to use the current setting; otherwise, enter a new value followed by a carriage return. If your iSBC 534 board is in the default condition (as shipped from the factory with the iSXM 100 extension module) you can, in response to these prompts, enter only a carriage return.

**CAUTION**

When you enter values in response to any SDT prompts, you must ensure that these values represent valid hardware or software specifications. If the values you enter do not match the actual hardware or software configuration, the tests cannot produce reliable results.

The sequence of questions is as follows:

**ENTER NUMBER OF iSBC 534 BOARDS IN THE SYSTEM [x] \***

where  $x$  is the present setting for the number of iSBC 534 boards in the system. In response, you should enter the number of boards in your system.

**ENTER BASE PORT ADDRESS FOR BOARD  $n$  [xx] \***

where  $n$  is the number of the iSBC 534 board and  $xx$  is the hexadecimal value of the default base address. Enter the I/O base address of the iSBC 534 board. The test asks this question for each of the iSBC 534 boards in your system.

**ENTER BOARD  $k$  PIC  $n$  INTERRUPT LEVEL [m] \***

where  $k$  is the number of the iSBC 534 board,  $n$  is the number of the PIC on the iSBC 534 board (0 or 1) and  $m$  is the default interrupt level of the processor board. The test issues this prompt for both PICs on each iSBC 534 board.

You should specify the interrupt level (0 - 7) to which you have connected the PIC on the iSBC 534 board. Entering an interrupt level greater than 7 indicates that the PIC is not connected to any interrupt line. The default level is the standard configuration for 86/300 Series systems.

**ENTER THE NUMBER OF THE BOARD TO CHECK [x] \***

where x is the default number of the board to be checked. In response, you should enter the number of the iSBC 534 board that you want the SDT534 to test. (The test checks only one board at a time.)

After the initialization sequence, SDT 534 displays its prompt character (\*) and waits for you to enter a command. If you want to change any hardware or software variables used in the SDT534 tests, you can use the **RESET**, **RESET HARDWARE**, or **RESET SOFTWARE** commands. These commands cause SDT534 to display a series of questions that allows you to change the variable values.

**Using the RESET SOFTWARE Command**

The **RESET SOFTWARE** command allows you to set variables used by the SDT 534 tests. If your iSBC 534 board is in the default condition, as shipped from the factory with the iSXM 100 Extension Module, you can leave these variables in their initial default settings.

The **RESET SOFTWARE** question sequence is as follows:

**SET GLOBAL V VARIABLES TO DEFAULT (Y OR [N]) \***

If you answer yes, the test sets the following variables to default values for use with tests 8 and 9:

Variable	Value	Description
V(0)	0FFH	Specifies the number of the USART that receives or transmits the character string in Tests 8 and 9. The default value (0FFH) specifies that no USART receives the character string.
V(1)	0	Determines whether Test 9 displays the character string at the terminal connected to the tested USART. The default value (0) specifies that no such display appears. An odd value (least-significant bit = 1) causes the test to display the character string.
V(2)	0FFH	Determines whether Test 9 displays the character string at the system console (the terminal connected to the USART on the processor board). The default value (0FFH) specifies that the display appears at the terminal. An even value (least-significant bit = 0) prevents the test from displaying the character string.



You can modify these global variables by using the V Command described in Chapter 2.

**INHIBIT USART USAGE IN TESTS (Y OR [N]) \***

If you answer yes to this question, the next question allows you to specify which USARTs to omit from testing. If you answer no, the test assumes all four USARTs are to be tested.

a yes response produces the following question for each USART:

**INHIBIT USART *n* USAGE [*s*] \***

where *n* is the number of the USART (0 - 3) and *s* is the present setting (Y or N). If you answer yes, this USART is not tested.

**USART CONFIGURATION FOR TEST 8 AND 9 (Y OR [N]) \***

If you answer yes to this prompt, SDT534 asks four additional questions about USART mode settings used in the USART Character Transmitter and Character Receiver tests. If you answer no, the test assumes a default set of modes and omits further questions. Refer to the *iSBC 534 Four Channel Communications Expansion Board Hardware Reference Manual* for more information about setting up the USART.

The USART configuration questions are as follows:

**ENTER MODE BYTE FOR USART *n* [*xx*] \***

where *n* is the number of the USART (0 - 3) and *xx* is the present value of the USART mode byte. Enter a new value use the default mode instruction byte.

SDT534 next prompts you to enter sync characters:

**ENTER SYNC CHAR *n* [*xx*] \***

where *n* is the number of the sync character (1 or 2) and *xx* is the present hexadecimal value of the sync character. Enter a new hexadecimal value for the sync character or use the default. The test issues this prompt for each sync character. If you are using asynchronous mode, enter a zero for each sync character. If you are using synchronous mode with one sync character, enter a zero for the second sync character.

Next, SDT534 asks for the USART command byte:

**ENTER COMMAND BYTE FOR USART *n* [*xx*] \***

where *n* is the number of the USART and *xx* is the present value of the command instruction byte. Enter a new value for the command instruction byte or use the default. The test issues this prompt for each of the four USARTs.

**NOTE**

SDT534 prohibits you from using the command instruction byte to perform an internal reset. It masks off the internal reset bit, preventing you from setting that bit.

Next, SDT534 asks for a timer count:

**ENTER TIMER COUNT FOR USART  $n$  [xxxx] \***

where  $n$  is the number of the USART (0-3) and xxxx is a word value that the test assumes it should load into the counter for use as an initial countdown value. Enter a new countdown value or use the default. The test issues this prompt for each of the four USARTs.

**Using the RESET HARDWARE Command**

The RESET HARDWARE command initiates a series of questions that allows you to change the settings of hardware variables used in testing the iSBC 534 board. If your iSBC 534 board is in the default condition as shipped from the factory, you can leave these hardware variables in their default settings.

The first several questions are the same as those that appear at SDT534 initialization; these questions have been described previously and will not be repeated here. The other questions are as follows:

**CHANGE USART-TIMER CONFIGURATION (Y OR [N]) \***

If you answer no, the test omits further questions and assumes the standard configuration as shipped from the factory. If you answer yes, the test asks the following question for each USART:

**ENTER TIMER NO FOR USART  $n$  TRANSMIT/RECEIVE [ $t$ ] \***

where  $n$  is the USART number (0 - 3) and  $t$  is the number of the baud rate generator to be used for the USART. Specify the baud rate generator (0 - 5) as it is configured on the board. The default setting is the standard configuration as shipped from the factory.

**TIMERS 4 AND 5 CASCADED ([Y] OR N) \***

Answer yes if the iSBC 534 board under test is set up with baud rate generators BGD4 and BGD5 connected in series rather than in parallel. If they are connected in parallel, answer no to this prompt.

**SDT534 TEST 0, PIT INITIALIZATION**

This test initializes each timer on each 8253 PIT and tests for successful initialization. The test checks each timer in sequence, verifying that the timer has been initialized and that it is running.

Test 0 can return the following error messages:

**ERROR: TIMER  $n$  DID NOT INITIALIZE, MODE =  $m$**

**ERROR: TIMER  $n$  DID NOT COUNT, MODE =  $m$**

where  $n$  is the number of the timer (0-5) and  $m$  is the kind of initialization attempted.

The possible values of  $m$  are as follows:

0	Binary count, two counts follow the mode command
1	Binary count, the high-order byte follows the mode command
2	Binary count, the low-order byte follows the mode command
3	BCD count, two counts follow the mode command
4	BCD count, the high-order byte follows the mode command
5	BCD count, the low-order byte follows the mode command

#### **SDT534 TEST 1, USART INITIALIZATION**

This test initializes each USART in both synchronous and asynchronous modes and tests the following options:

- Synchronous Mode: parity enabled and disabled; even and odd parity; one and two sync characters; 5- through 8-bit character lengths
- Asynchronous Mode: baud rates of 110, 9600, and 19,200; parity enabled and disabled; even and odd parity; 5-through 8-bit character lengths; 1, 1.5, and 2 stop bits.

Test 1 verifies the initialization by sending a character, receiving the character, and verifying that sent and received characters are the same.

This test can return the following error messages:

**ERROR: ASYNC USART  $n$  : STATUS= $ss$  ACTUAL= $aa$  EXPECTED= $ee$   
PARITY= $p,v$  LEN= $l$  BAUD= $bbbb$  SB= $s.b$**

where

$n$	is the number of the USART (0 - 3) on which the error occurred
$ss$	is the contents (hexadecimal) of the status register
$aa$	is the character pattern received
$ee$	is the expected character character pattern

<i>p</i>	indicates parity is enabled (1) or disabled (0)
<i>v</i>	indicates whether parity is even (1) or odd (0)
<i>l</i>	is the number of bits in the character transmitted or received
<i>bbbb</i>	is the baud rate
<i>s.b</i>	is the number of stop bits (1.0, 1.5, or 2.0) used in the communication process

**ERROR: SYNC USART *n* : STATUS = *ss* ACTUAL = *aa* EXPECTED = *ee*  
 PARITY = *p,v* LEN = *l* BAUD = *bbbb* SYNC CHARS = *cc***

where the values for *n*, *ss*, *aa*, *ee*, *p*, *v*, *l*, and *bbbb* are the same as for the previous message, and *cc* is the hexadecimal value of the sync characters the USART is programmed to insert into the data stream.

#### **SDT534 TEST 2, PIC INITIALIZATION**

Test 2 initializes each PIC by writing 16H into ICW1 and 0 into ICW2. The test then sets and reads the interrupt mask, using different patterns to verify each bit in the mask. After verifying the mask, test sends and receives a character pattern via port C of the PPI. It then compares the input with the output to verify the PPI operation.

This test can return the following error message:

**ERROR: PIC *n*: ACTUAL = *aa* EXPECTED = *ee***

where *n* is the number of the PIC under test and *aa* and *ee* are the actual and expected values, respectively, sent or received via port C.

#### **SDT534 TEST 3, PPI PORT C VERIFICATION**

Test 3 sends and receives a character pattern via port C of the PPI. It then compares the input with the output to verify the PPI operation.

This test can return the following error message:

**ERROR: PPI PORT C FAILURE:  
 ACTUAL = *aa* EXPECTED = *ee***

where *aa* and *ee* are the actual and expected values, respectively, sent or received via port C.

#### **SDT534 TEST 4, USART INTERRUPT VERIFICATION**

Test 4 verifies that each USART generates the correct transmit and receive interrupts at PIC 0. It initializes all USARTs

(including the ones omitted from testing at initialization) with the transmit and receive functions disabled. It then unmask all interrupts on PIC 0 and repeats the following sequence 50 times for all four USARTs, using low medium and high baud rates.

- Sends a character to the USART
- Verifies that the processor board receives the correct transmit and receive interrupts
- Verifies the status of the USART and verifies that the character sent matches the character received
- Verifies that the processor board received no other interrupts

This test can return the following error messages:

**ERROR: BOARD  $n$ , PIC  $p$ : CANNOT SET MASK,  
ACTUAL =  $aa$  EXPECTED =  $ee$**

where

- $n$  is the number of the iSBC 534 board (from 0 to the number of boards in the system).
- $p$  is the number of the PIC (0 - 1)
- $aa$  is the hexadecimal value of the mask as read from the PIC
- $ee$  is the expected hexadecimal value of the mask, as set by the test.

**ERROR: CANNOT SET ONBOARD PIC MASK,  
ACTUAL=  $aa$ , EXPECTED =  $ee$**

where  $aa$  is the hexadecimal value of the mask as read from the on-board PIC and  $ee$  is the expected hexadecimal value of the mask as set by the test.

**ERROR: WRONG INTERRUPT OCCURRED, USART =  $u$ ,  
BAUD =  $bbbb$ , WRONG INTERRUPT =  $i$**

where

- $u$  is the number of the USART (0 - 3) that transmitted the interrupt
- $bbbb$  is the baud rate of the USART
- $i$  is the interrupt level (0 - 7) of the interrupt that occurred

**ERROR: RECEIVE INTERRUPT DID NOT OCCUR,  
 USART = *u*, BAUD = *bbbbbb*,  
 USART RECEIVE STATUS = *ss***

**ERROR: TRANSMIT INTERRUPT DID NOT OCCUR,  
 USART = *u*, BAUD = *bbbbbb*,  
 USART RECEIVE STATUS = *ss***

**ERROR: EXTRA RECEIVE INTERRUPT OCCURRED,  
 USART = *u*, BAUD = *bbbbbb*,  
 USART RECEIVE STATUS = *ss***

**ERROR: EXTRA TRANSMIT INTERRUPT OCCURRED,  
 USART = *u*, BAUD = *bbbbbb*,  
 USART TRANSMIT STATUS = *ss***

**ERROR: RECEIVE FAILED, USART = *u*, BAUD = *bbbbbb*,  
 USART RECEIVE STATUS = *ss*  
 CHAR ACTUAL = *aa*, CHAR EXPECTED = *ee***

**ERROR: TRANSMIT FAILED, USART = *u*, BAUD = *bbbbbb*,  
 USART TRANSMIT STATUS = *ss***

where

*u* is the number of the USART (0 - 3).

*bbbbbb* is the baud rate of the USART.

*ss* is the hexadecimal value of the USART status byte.

*aa* is the hexadecimal value of the character read from the USART.

*ee* is the hexadecimal value of the character that the test sent to the USART.

#### **SDT534 TEST 5, TIMER 4 AND 5 INTERRUPT VERIFICATION**

Test 5 programs timers 4 and 5 to count down and generate an interrupt when they reach zero. The test checks each timer at each interrupt level, masking the levels not used during each check.

This test can return the following error messages:

**ERROR: BOARD *n*, PIC *p*: CANNOT SET MASK, ACTUAL = *aa*, EXPECTED = *ee***

where

*n* is the number of the iSBC 534 board (from 0 to the number of boards in the system).

*p* is the number of the PIC (0 - 1).

*aa* is the hexadecimal value of the mask as read from the PIC.

*ee* is the expected hexadecimal value of the mask.

**ERROR: CANNOT SET ONBOARD PIC MASK, ACTUAL = *aa*, EXPECTED = *ee***

where *aa* is the hexadecimal value of the mask as read from the PIC and *ee* is the expected hexadecimal value of the mask as set by the test

**ERROR: TIMER *t*: EXPECTED INTERRUPT DID NOT OCCUR**

where *t* is the number of the timer (4 - 5) that did not generate an interrupt.

**ERROR: TIMER *t*: INTERRUPT OCCURRED WHEN NOT EXPECTED**

where *t* = the number of the timer (4 - 5) that interrupted unexpectedly.

#### **SDT534 TEST 6, BAUD RATE VERIFICATION**

Test 6 checks each USART to verify that it is transmitting and receiving at the proper baud rate. The test does this by determining the number of characters transmitted or received in a known period of time. The test checks baud rates of 110 and 2400.

This test can return the following error message:

**ERROR: BAUD RATE ON USART *u* : ACTUAL = *aaaaaa* EXPECTED = *eeeeee***

where *u* is the number of the USART and *aaaaaa* and *eeeeee* are the actual and expected baud rates, respectively.

#### **SDT534 TEST 7, USART LOAD TEST**

This test places the iSBC 534 board in test mode. Then it initializes each USART to a different baud rate and enables the transmit and receive interrupts. The test next transmits and receives characters, checking each input character to verify that it matches the corresponding output character. The test also sets a timer to check that each interrupt occurs within a given period of time.

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

#### **SDT534 TEST 8, USART CHARACTER TRANSMITTER TEST**

Initially, this test is set to "ignored" status; when recognized using the RECOGNIZE Command, this test sends an ASCII character string to the USART specified by the V(0) global variable. The test sets the

iSBC 534 board to normal mode so that characters can be received at the terminal connected to the port under test.

This test can return the following error messages:

**ERROR: NO USART SPECIFIED**

Before running this test, you must use the V command to declare a USART to test.

**ERROR: IMPROPER USART #*u* SPECIFIED**

where *u* is the number of the USART that you specified by setting the V(0) global variable. This error can occur if you set the V(0) global variable to a value outside the range 0 to 3 during initialization.

**SDT534 TEST 9, USART CHARACTER RECEIVER TEST**

Initially, this test is set to "ignored" status; when recognized using the Recognize Command, Test 9 checks the ability of a USART to receive characters from a terminal.

After invoking this test, you enter characters at a terminal connected to the RS-232 port controlled by the USART specified by the V(0) global variable. The test places the characters it receives in a wrap-around buffer, echoing them at the terminals specified by the V(1) and V(2) global variables.

The rate at which the test echoes characters does not have a direct relationship to the the rate at which you enter those characters. Therefore, if you type characters faster than the test echoes them, some characters are lost.

You can terminate the test by entering a CONTROL-Z character.

This test can return the following error messages:

**ERROR: NO USART SPECIFIED**

Before running this test, you must use the V command to declare a USART to test.

**ERROR: IMPROPER USART #*u* SPECIFIED**

where *u* is the number of the USART that you specified by setting the V(0) global variable. This error can occur if you set the V(0) global variable to a value outside the range 0 to 3.

**ERROR: NO CHARACTER RECEIVED; USART STATUS = *ss***

where *ss* is the hexadecimal value of the USART's status word. The test displays this message if it does not receive the first character within 30 seconds.



**SDT544 TEST SUITE**

The SDT544 test suite contains tests for the iSBC 544 Intelligent Communications Controller board. SDT544 consists of thirteen diagnostic tests. The tests and their functions are listed in Table 6-4.

Table 6-4. SDT544 Tests

Test	Function
0	MULTIBUS® RAM test
1	8085 Jump Out test
2	Dual Port RAM test
3	544 board interrupt verification
4	iRMX™86 firmware verification
5	PIT initialization
6	USART initialization
7	PIC initialization
8	PPI verification
9	USART interrupt verification
A	Baud rate verification
B	USART character transmitter test
C	USART character receiver test

**SDT544 INITIALIZATION**

When invoked, the SDT544 diagnostic test suite displays the following message:

**SYSTEM DIAGNOSTIC TEST - 544, Vx.y**  
**Copyright 1983 Intel Corporation**

where x.y is the version number.

When invoked, the SDT544 test suite asks a series of questions about the configuration of your system. If the iSBC 544 board is in the

default condition (as shipped from the factory) you can use the default values for each of the questions.

Whenever SDT544 asks a question, it encloses the current (or default) setting in brackets. For example, in the following question:

**ENTER THE NUMBER OF iSBC 544 BOARDS IN THE SYSTEM [2] \***

the 2 displayed in brackets is the current value. Enter a carriage return alone to accept the default setting; enter an appropriate new value followed by a carriage return to change the setting.

### CAUTION

When you enter values in response to any of the following questions, you must ensure that these values represent valid hardware or software configurations. If the values you enter do not match your actual hardware or software configuration, the test will produce unreliable results.

The first question in the initialization sequence is as follows:

**ENTER THE NUMBER OF iSBC 544 BOARDS IN THE SYSTEM [x] \***

where *x* is the default value.

If the number you enter is not between one and four, SDT544 displays the following error message and requests a new response:

**ILLEGAL NUMBER OF BOARD: *n***  
**LEGAL VALUES ARE: 1, 2, 3, 4**

The system repeats the following questions once for each board.

**ENTER DUAL PORT RAM BASE ADDRESS FOR BOARD *n***  
**ENTER SEGMENT [*ssss*] \* *aaaa***  
**ENTER OFFSET [*oooo*] \* *ffff***

where

- |             |   |
|-------------|---|
| <i>n</i>    | is the number of the board.   |
| <i>ssss</i> | is the current segment portion of the RAM base address (in hexadecimal).                                      |
| <i>aaaa</i> | is your response, stating the actual segment portion of the RAM base address. Enter the value in hexadecimal. |
| <i>oooo</i> | is the current offset portion of the RAM base address (in hexadecimal).                                       |

*ffff* is your response, stating the actual offset portion of the RAM base address. Enter the value in hexadecimal.

If the sum of the base address and offset you entered is not on a 1000H boundary, the system displays the following error message and requests a new response:

**ILLEGAL BASE ADDRESS: *aaaa:ffff*  
MUST BE ON A 1000 HEX BOUNDARY**

If you specify a base address that forces some of the RAM addresses to extend beyond 64K (the MULTIBUS System Bus cannot access all RAM), the test displays the following error message:

**\*\* WARNING \*\* WITH DUAL PORT RAM SIZE OF *ssss*  
AND A BASE ADDRESS OF *aaaa:ffff*  
THE SYSTEM WILL NOT BE ABLE TO ACCESS BOARD *n***

where

*ssss* is the size of the dual port RAM (in hexadecimal).  
The size should be 4000H.

*aaaa* is the value you entered as the actual segment portion of the RAM base address (in hexadecimal).

*ffff* is the value you entered as the actual offset portion of the RAM base address (in hexadecimal).

*n* is the number of the board under test.

**ENTER BOARD *n* SEND INTERRUPT LEVEL [*xx*] \* *yy***

where

*n* is the board number that you are presently configuring.

*xx* is the currently specified send interrupt level.

*yy* is your response, stating the interrupt level that the board is jumpered for.

If you enter an interrupt level greater than seven, the test displays the following error message:

**SEND INTERRUPT ON BOARD *n* NOT CONNECTED**

If you indicated that you have more than one iSBC 544 board in your system, the test displays the following prompt:

**ENTER THE NUMBER OF THE BOARD TO CHECK [*x*] \***

where *x* is the number of the board currently specified.

If you enter a number greater than the number of boards you indicated were in your system, the test displays the following error message and requests a new response:

**THERE IS NO BOARD  $n$  SPECIFIED IN THE SYSTEM  
LEGAL BOARD NUMBERS ARE: 1, ...  $m$**

where  $n$  is your response to the previous prompt and  $m$  is the number of boards that you indicated are in the system.

After you have responded to all of the prompts, the system attempts to read the firmware version from the board. The system displays the following message whether or not the read attempt was successful:

**iRMX86/XENIX FIRMWARE VERSION  $V_{x.y}$**

where  $x.y$  is the version number the system read from the board. If the board is not present or there are errors in the RAM, an erroneous version number results.

If the version number displayed is 1.1 (or less), the test also displays the following warning:

**\*\*\* WARNING: THIS TEST SUITE MAY NOT OPERATE \*\*\*  
\*\*\* CORRECTLY ON ANY VERSION OF THE iRMX \*\*\*  
\*\*\* 86/XENIX FIRMWARE PRIOR TO VERSION 1.2 \*\*\***

#### Using the RESET SOFTWARE Command

Using the RESET or RESET SOFTWARE command, you can change the settings of test variables used by SDT544. SDT544 prompts you for answers to a series of questions, as follows:

**CHANGE RESET\$SOFTWARE VARIABLES BACK TO  
CONFIGURATION DEFAULT VALUES (Y OR [N]) \***

A yes response causes the test to reset all variables to their default values. A no response causes the test to continue with the following question:

**SET GLOBAL V VARIABLES TO DEFAULT (Y or [N]) \***

where a "y" response sets  $V(0)$ ,  $V(1)$ , and  $V(2)$  as follows:

$V(0) = FFH$  This variable is used by Test 11 to specify the port that is to receive a character string. Valid values include 0 to 3H.

$V(1) = 0$  This variable is used by Test 11 to control the echo of the character string back to the sending device. The value 0 means no echo.

V(2) = FFH      This variable is used by Test 11 to control the echo to the system console (the terminal connected to processor board's RS-232 port). FFH means yes, the characters will be echoed at the system console.

These variables can be modified using the V command.

#### **INHIBIT PORT USAGE IN TESTS (Y or [N]) \***

A no response causes the next question to be skipped; a yes response causes the system to ask this question:

#### **INHIBIT PORT *n* USAGE [*s*] \***

where *n* is the number of the port and *s* is the default response. A yes answer inhibits use of the specified port; a no answer allows use of the port.

The test repeats this question for each port (0-3). After you respond to the prompt for port 3, the test displays the following prompt:

#### **SPECIFY BAUD RATE FOR TESTS B and C (Y or [N]) \***

A no response ends the reset software prompts; a yes response causes the test to display the following prompt:

#### **BAUD RATE FOR PORT *n* (*bbbb*) \* *rrrr***

where

*n*                    is the number of the port.

*bbbb*                is the current baud rate (in decimal). The current value is the value at initialization or the last value you specified when you invoked RESET SOFTWARE.

*rrrr*                is your response, selected from one of the following baud rates: 19,200, 9600, 4800, 2400, 1200, 600, 300, 150, and 110.

The test repeats the baud rate question for each of the four ports (0-3). If you enter an invalid baud rate, the test displays the following error message and waits for a new response:

**INVALID BAUD RATE: *rrrr***  
**SELECT FROM THE FOLLOWING BAUD RATE VALUES:**  
**110T, 150T, 300T, 600T, 1200T, 2400T, 4800T, 9600T, 19200T**

#### **Using the RESET HARDWARE Command**

The RESET or RESET HARDWARE command allows you to change the settings of hardware variables used by the SDT544 tests. SDT544

asks a series of questions, waiting after each question for your response. The sequence of questions is as follows:

**CHANGE RESET\$HARDWARE VARIABLES BACK TO CONFIGURATION DEFAULT VALUES (Y OR [N]) \***

A yes response causes the test to reset all of the variables to their default values and omit further questions. If you answer no, SDT544 proceeds through the same series of questions that it does at initialization; these questions were described earlier and will not be repeated here. After the initialization sequence, SDT544 asks the following additional questions:

**CHANGE LOOPBACK CONNECTOR FLAGS (Y OR [N]) \***

If you answer yes, the test displays the following prompt, once for each port:

**LOOPBACK CONNECTOR ON SERIAL PORT *p* (Y OR [N]) \***

where *p* is a value of zero to three, representing one of the serial ports on the iSBC 544 board.

Answer yes if your system has a loopback connector on the indicated serial port. If you answer yes when the loopback connector is not installed for the specified port, tests 6 and 9 will report errors.

**SDT544 TEST 0, MULTIBUS RAM TEST**

The MULTIBUS RAM test checks the ability to read from and write to the iSBC 544 dual port RAM over the MULTIBUS System Bus. Always run this test before running the others, since the system bus must be able to access the iSBC 544's dual port RAM correctly in order to communicate with the 8085A microprocessor.

If you set DEBUG to FALSE, this test aborts on the first error encountered. If you set DEBUG to TRUE, the test prints an error message for every error it finds. All errors in this test are of the following form:

**ERROR: MULTIBUS RAM ACCESS. LOCATION = *ssss:0000*,  
VALUE READ = *rr*, VALUE EXPECTED = *ee***

where

*ssss* is the base (segment) portion of the address referenced via the MULTIBUS structure.

*0000* is the offset portion of the address referenced via the MULTIBUS structure.

*rr* is the hexadecimal value the test reads at location *ssss:0000*.

*ee* the hexadecimal value the test expects at location *ssss:0000*.

**SDT544 TEST 1, iRMX™ 86/XENIX 86  
FIRMWARE JUMP OUT COMMAND TEST**

This test checks the iRMX 86/XENIX 86 firmware's ability to exercise the iSBC 544 board's 8085A microprocessor. The firmware should cause the 8085 microprocessor to "jump out" and execute instructions the system processor has loaded into the dual port RAM. Test 0 (the MULTIBUS RAM test) should already have been run and passed.

The test can fail for the following reasons:

- The 8085A microprocessor can't access the dual port RAM correctly.
- The firmware cannot cause the 8085A microprocessor to "jump out" to the RAM.
- The 8085A microprocessor cannot execute instructions correctly.
- There was a failure in the RAM used by this test.

The following error message is produced for all errors detected by this test:

**ERROR: JUMPOUT FAILED. iRMX86/XENIX FIRMWARE STATUS = *ff*,  
MULTIBUS ADDRESS = *ssss:0000*, 8085 ADDRESS = *aaaa*,  
VALUE READ = *rr*, VALUE EXPECTED = *ee***

where

*ff* is the iRMX86/XENIX firmware status. Values other than one through seven indicate errors.

*ssss* is the segment portion of the address (the first location in output line 0) referenced via the MULTIBUS System Bus.

*0000* is the offset portion of the address (the first location in output line 0) referenced via the MULTIBUS System Bus.

*aaaa* is the address (the first location in output line 0) referenced by the on-board 8085A microprocessor.

*rr* is the hexadecimal value the test read at location *ssss:0000*.

*ee* is the hexadecimal value the test expected at location *ssss:0000*.

**SDT544 TEST 2, DUAL PORT RAM TEST**

You can use Test 2 to determine whether the dual port RAM can be accessed correctly by both the on-board 8085A microprocessor and the MULTIBUS System Bus.

This test sets up a fail-safe timer to prevent an error from hanging up the system processor. If the test gets no response within the allotted time, it displays the following message:

**ERROR: TIMEOUT IN DUAL PORT RAM TEST.  
MULTIBUS ADDRESS = ssss:0000**

where *ssss* is the segment portion of the address referenced and *0000* is the offset portion of the address referenced.

The test displays all other error messages in the following format:

**ERROR: DUAL PORT RAM TEST. VALUE EXPECTED = ee  
MULTIBUS: VALUE READ = rr, LOCATION ADDRESS = ssss:0000  
8085: VALUE READ = kk, LOCATION ADDRESS = llll**

where

<i>ee</i>	is the value the test expected in location <i>ssss:0000</i> .
<i>rr</i>	is the value read in location <i>ssss:0000</i> over the MULTIBUS System Bus.
<i>ssss</i>	is the segment portion of the location accessed over the MULTIBUS System Bus.
<i>0000</i>	is the offset portion of the location accessed over the MULTIBUS System Bus.
<i>kk</i>	is the value the 8085A microprocessor on the iSBC 544 read.
<i>llll</i>	is the location referenced by the 8085A microprocessor.

**SDT544 TEST 3, BOARD INTERRUPT VERIFICATION**

Test 3 checks the ability of the iSBC 544 board to send an interrupt request to the system processor.

This test can produce the following error messages:

**ERROR: INTERRUPT DID NOT OCCUR. BOARD INTERRUPT = e  
FIRMWARE: COMMAND BYTE = cc, STATUS BYTE = ss**

**ERROR: UNEXPECTED INTERRUPT OCCURRED FROM BOARD.  
BOARD INTERRUPT = e  
FIRMWARE: COMMAND BYTE = cc, STATUS BYTE = ss**



where

- e* is either "ENABLED" or "DISABLED".
- cc* is the contents (in hexadecimal) of the board command byte.
- ss* is the contents (in hexadecimal) of the board status byte.

#### **SDT544 TEST 4, IRMX 86 /XENIX 86 FIRMWARE VERIFICATION**

Test 4 checks the firmware's execution of the reset, output, parameter, and continue commands. If you install a loopback connector on a port, the test verifies the input command for that port.

This test can produce the following error messages:

**ERROR: BAD STATUS ON PARAMETER COMMAND FOR PORT *p***  
**FIRMWARE STATUS = *ss***

**ERROR: BAD STATUS ON OUTPUT COMMAND FOR PORT *p***  
**FIRMWARE STATUS = *ss***

**ERROR: WRONG OUTPUT BUFFER POINTERS FOR PORT *p***  
**NUMBER CHARS SPECIFIED BY SYSTEM = *oo***  
**NUMBER CHARS SPECIFIED BY FIRMWARE = *tt***

where

- p* is the port number (0, 1, 2, 3).
- ss* is the firmware status byte (in hexadecimal). Values other than one through seven indicate errors.
- oo* is the number (in hexadecimal) of characters that the test puts in the output buffer.
- tt* is the number (in hexadecimal) of characters in the buffer that the firmware recognizes.

The following message results from a timeout by the fail-safe timer; it occurs if you specify a loopback connector but do not install it.

**ERROR: INPUT NOT RECEIVED BY PORT *p***

where *p* is the port number (0, 1, 2, 3).

The test can also display the following three error messages if you specify the loopback connector for the port:

**ERROR: INPUT DOES NOT MATCH OUTPUT FOR PORT *p***  
**NUMBER CHARS OUTPUT = *hh*, NUMBER CHARS INPUT = *ii***

**ERROR: BAD STATUS ON INPUT COMMAND FOR PORT *p***  
**FIRMWARE STATUS = *ss***

**ERROR: WRONG INPUT BUFFER POINTERS FOR PORT *p***  
**NUMBER OF CHARS SPECIFIED BY SYSTEM = *ee***  
**NUMBER OF CHARS SPECIFIED BY FIRMWARE = *ff***

where

- p* is the port number (0, 1, 2, 3).
- hh* is the number of characters the test transmits.
- ii* is the number of characters the board receives via the loopback connector.
- ss* is the firmware status byte (in hexadecimal).
- ee* is the actual number of characters (in hexadecimal) the test removed from the input buffer.
- ff* is the number of characters (in hexadecimal) recognized by the firmware that the test took from the input buffer.

If the fail-safe timer reaches its maximum count before the board responds to the command, the test displays one of the following error messages:

**ERROR: TIMEOUT ON PARAMETER COMMAND FOR PORT *p***

**FIRMWARE: COMMAND BYTE = *aa*, STATUS BYTE = *bb***

**ERROR: TIMEOUT ON OUTPUT COMMAND FOR PORT *p***

**FIRMWARE: COMMAND BYTE = *aa*, STATUS BYTE = *bb***

**ERROR: TIMEOUT ON INPUT COMMAND FOR PORT *p***

**FIRMWARE: COMMAND BYTE = *aa*, STATUS = *bb***

where

- p* is the port number (0, 1, 2, 3).
- aa* is the firmware command byte (in hexadecimal).
- bb* is the firmware status byte (in hexadecimal).

#### **SDT544 TEST 5, PIT INITIALIZATION**

The purpose of Test 3 is to check whether 8253 PIT can be properly set up for operation. To do this, the test loads a count value into each timer within the 8253 and starts the timer counting. After a delay,

the reads the count to see if the timer is actually counting. Each timer in the PIT is checked using both binary and BCD counts.

Test 5 can produce the following error message:

**ERROR: TIMER *t* NOT RUNNING**

This test sets up a fail-safe timer (on the processor board); if the iSBC544 is nonresponsive, the test displays the following message:

**ERROR: TIMEOUT OCCURRED WHILE TESTING TIMER *t*  
FIRMWARE: COMMAND BYTE = *cc*, STATUS = *ss***

where

- t* is the timer number (0 to 5).
- cc* is the hexadecimal value of the firmware command byte.
- ss* is the hexadecimal value of the firmware status byte.

#### **SDT544 TEST 6, USART INITIALIZATION**

Test 6 initializes the USART in asynchronous mode. It then confirms the initialization by transmitting characters using the following:

- Baud rates of 110, 9600, 19,200
- Character lengths of five to eight characters
- Parity enabled and disabled
- Even and odd parity
- 1, 1.5, and 2 stop bits

If you did not specify a loopback connector, the test masks the interrupts on the iSBC 544 board's PIC and checks that the USART can transmit a character in each of the modes. If you specify a loopback connector, the test also checks that the USART can receive characters. Test 6 can produce the following error message:

**ERROR: ASYNCH USART *u*, STATUS = *ss*, PARITY = *p,e*,  
LEN = *l*, BAUD = *bbbbbb*, SB = *k.k*  
CHAR RECEIVED = *r***

where

- u* is the port number (0, 1, 2, 3).
- ss* is the hexadecimal value of the status byte read from the USART.

<i>e</i>	is 0 for parity disabled and 1 for parity enabled.
<i>p</i>	is 0 if parity is odd and 1 if parity is even.
<i>l</i>	is the character length (5, 6, 7, 8 bits).
<i>bbbb</i>	is the decimal value of the baud rate.
<i>k.k</i>	is the number of stop bits (1.0, 1.5, 2.0).
<i>r</i>	is the hexadecimal value of the character received, if you specified a loopback connector for the port being tested.

If the iSBC 544 board does not respond within the allowed time, the test displays this message:

**ERROR: TIMEOUT IN USART *u* INITIALIZATION  
FIRMWARE: COMMAND BYTE = *cc*, STATUS BYTE = *ss***

where

<i>u</i>	is the port number (0 to 3).
<i>cc</i>	is the command byte of the firmware.
<i>ss</i>	is the status byte of the firmware.

#### **SDT544 TEST 7, PIC INITIALIZATION**

Test 7 sets and verifies the mask in the PIC using various patterns to verify each bit in the mask. The test initializes the PIC with the ICW1 parameter set to 16H and the ICW2 parameter set to 0.

If the test detects any failures, it displays the following error message:

**ERROR: PIC INITIALIZATION. MASK SET = *mm*, MASK READ = *rr***

where *mm* is the hexadecimal value the test writes to the PIC and *rr* is the hexadecimal value the test reads from the PIC.

If the iSBC 544 board does not respond within the allowed time, the test displays the following error message:

**ERROR: TIMEOUT IN PIC INITIALIZATION  
FIRMWARE: COMMAND BYTE = *cc*, STATUS BYTE = *ss***

where *cc* is the hexadecimal value of the firmware command byte and *ss* is the hexadecimal value of the firmware status byte.

**SDT544 TEST 8, PPI PORT A VERIFICATION**

This test initializes the PPI and tests port A by writing a pattern to port A and then reading the pattern back.

This test can generate the following error messages:

**ERROR: PPI PORT A VERIFICATION FAILED**  
**PATTERN: WRITTEN = *ww*, READ = *rr***

where *ww* is the hexadecimal value of the pattern the test writes to the output port and *rr* is the hexadecimal value of the pattern the test reads from the input port.

**ERROR: TIMEOUT DURING PPI VERIFICATION.**  
**FIRMWARE: COMMAND BYTE = *cc*, STATUS BYTE = *ss***

where *cc* is the hexadecimal value of the firmware command byte and *ss* is the hexadecimal value of the firmware status byte.

**SDT544 TEST 9, USART INTERRUPT VERIFICATION**

This test verifies that each USART generates the correct interrupts at the PIC. If you specify a loopback connector for a port (and connect it to the port), the test checks both the transmit and receive interrupt for that port. If you do not specify a loopback connector for a port, the test checks only the transmit interrupt.

The test can generate the following error messages:

**ERROR: TRANSMIT INTERRUPT DID NOT OCCUR FROM USART *u***  
**USART STATUS = *ss***

**ERROR: RECEIVE INTERRUPT DID NOT OCCUR FROM USART *u***  
**USART STATUS = *ss***

**ERROR: WRONG INTERRUPT OCCURRED, USART STATUS = *ss***  
**INTERRUPT LEVEL: EXPECTED = *e*, RECEIVED = *r***

**ERROR: BAD USART *u* STATUS ON TRANSMIT, STATUS = *ss***

**ERROR: BAD USART *u* STATUS ON RECEIVE, STATUS = *ss***

**ERROR: CHARACTER DOES NOT MATCH FROM USART *u***  
**CHAR: TRANSMITTED = *tt*, RECEIVED = *cc*,**  
**USART RECEIVE STATUS = *ss***

**ERROR: TIMEOUT WHILE TESTING USART *u* INTERRUPTS**

where

*u* is the port number (0 - 3).

*ss* is the USART status in hexadecimal.

<i>e</i>	is the level of interrupt the test expected.
<i>r</i>	is the level of interrupt the test received.
<i>tt</i>	is the hexadecimal value of the character the test transmitted.
<i>cc</i>	is the hexadecimal value of the character the test received.

#### **SDT544 TEST A, BAUD RATE VERIFICATION**

Test A verifies that each USART is transmitting at the correct baud rate. The test does this by counting the number of characters transmitted in a known period of time. It checks baud rates of 110 and 2400 on each USART.

The test can generate the following error messages:

**ERROR: WRONG BAUD RATE ON USART *u***  
**BAUD RATE: SET = *bbbb*, ACTUAL = *aaaa***

where

<i>u</i>	is the port number (0 to 3).
<i>bbbb</i>	is the baud rate (in decimal) the test sets for the port.
<i>aaaa</i>	is the baud rate at which the USART is transmitting.

**ERROR: TIMEOUT ON USART *u* BAUD RATE VERIFICATION**  
**FIRMWARE: COMMAND BYTE = *cc*, STATUS BYTE = *ss***

where *cc* is the hexadecimal value of the firmware command byte and *ss* is the hexadecimal value of the firmware status byte. This message occurs if the iSBC 544 board does not respond within the allowed time.

#### **SDT544 TEST B, PORT CHARACTER TRANSMITTER TEST**

Test B sends an ASCII character string to the port you specify using the V(0) variable.

This test can generate the following error messages:

**ERROR: NO PORT NUMBER SPECIFIED**

**ERROR: ILLEGAL PORT NUMBER (*p*) SPECIFIED**

**ERROR: TIMEOUT ON PORT *p* CHARACTER TRANSMITTER TEST.**  
**FIRMWARE: COMMAND BYTE = *cc*, STATUS BYTE = *ss***

where

- p* is the port number you specified.
- cc* is the hexadecimal value of the firmware command byte.
- ss* is the hexadecimal value of the firmware status byte.

**SDT544 TEST C, PORT CHARACTER RECEIVER TEST**

This test reads a character string from the port you specify using the V(0) variable. The V(1) and V(2) variables control how the input characters are echoed. Setting V(1) to TRUE causes the character to be echoed at the terminal connected to the port under test. Setting V(2) to TRUE causes the character to be echoed on the system console. If you enter characters faster than the test can echo them, some characters will be lost. The test reports a timeout error if it does not receive the first character within 30 seconds. The test terminates when it reads a CONTROL-Z.

This test can generate the following error messages:

**ERROR: NO PORT NUMBER SPECIFIED**

**ERROR: ILLEGAL PORT NUMBER (*p*) SPECIFIED**

**ERROR: TIMEOUT ON PORT *p* CHARACTER TRANSMITTER TEST.  
FIRMWARE: COMMAND BYTE = *cc*, STATUS BYTE = *ss***

where

- p* is the port number you specified.
- cc* is the hexadecimal value of the firmware command byte.
- ss* is the hexadecimal value of the firmware status byte.



## APPENDIX A CONFIGURING THE SYSTEM DIAGNOSTIC TESTS

Diagnostic diskettes #2 (8") or #2A and #2B (5 $\frac{1}{4}$ ") contain configuration files that you can modify to tailor your SDT218, SDT309, SDT351, SDT534, and SDT544 test suites to your requirements. The configuration files for a particular test suite reside in a directory called SDTxxx.DIR, where xxx is the number of the test suite. (For example, the configuration files for SDT309 reside in directory SDT309.DIR.) Each directory contains several files that you need to reconfigure the test suite. One of the files is the configuration file; its name is SDTxxxCNF.P86, where xxx is the number of the test suite. The configuration files are PL/M-86 source files that consist of variable declarations used by the test suite. By modifying the declared values, you can change the configuration of the test suite.

### MAKING CHANGES TO THE TEST SUITES

Examine the configuration files before making any changes to them; they describe the configuration of the test suites as released at the factory. If you decide that the test suites are already in the proper configuration, you need not do anything further. You can ignore the procedures in this section and use the test suites as described.

If you elect to make changes to a configuration file, you must edit and compile the file, then link it to the library of test suites to produce an executable test suite. The diagnostic diskette provides a SUBMIT file for each configurable diagnostic that performs this process for you.

To modify one of the test suites, perform the following steps:

### CAUTION

If you wish to save your present copy of SDTxxx, first rename it SDTxxx.OLD before starting the reconfiguration process.

1. If you have not already attached your flexible diskette drive, use the following command:

```
-ATTACHDEVICE www AS :FD0:)
```

where *www* is the physical device name of the device to be attached. Use the physical device name WFDD0 if you have a double-sided, double-density flexible diskette drive installed in



an iSBC 215 board environment. Refer to the iRMX 86 Operator's Manual for more information on the ATTACHDEVICE command.

2. Insert the diagnostic diskette containing the configuration files.
3. Enter the following command to make SDTDIR the default directory:

```
-ATTACHFILE /SDTDIR
```

4. Enter the following command to install the necessary configuration files:

```
-SUBMIT :FD0:INSTALL.CSD (SDTxxx)
```

where xxx is the SDT number (351, 534, etc.)

5. Edit the configuration file residing in directory SDTxxx.DIR/SRC/SDTxxxCNF.P86 and make the necessary changes.

6. Invoke the SUBMIT file to compile the configuration file, link it to the test suite, and produce an executable module. Use the following command:

```
-SUBMIT SDTxxx.DIR/CSD/SDTxxx.CSD
```

where xxx is the number of the SDT (309, 351, etc.).

This command produces an executable module, SDTxxx, and places it in directory SDTDIR.



- Abbreviations,
  - use in commands, 2-6
- Address March Test,
  - iSBC 86/30 board, 3-5
  - memory boards, 4-10
- Address Patterns Test,
  - iSBC 86/12A, 3-10
- Baud rate verification,
  - iSBC 351 board, 6-9
  - iSBC 534 board, 6-20
  - iSBC 544 board, 6-35
- Buffer I/O Test, iSBC 215 board,
  - 5-6
- Character Receiver Test,
  - iSBC 351 board, 6-10
  - iSBC 534 board, 6-21
  - iSBC 544 board, 6-36
- Character Transmitter Test,
  - iSBC 351 board, 6-10
  - iSBC 534 board, 6-20
  - iSBC 544 board, 6-35
- Checksum test,
  - iSBC 86/30 board, 3-3
  - iSBC 86/12A board, 3-8
  - iSBC 215 board, 5-5
  - processor ROM, 1-5
- Cleaning diskette drive heads,
  - 5-26, 5-29
- CLEAR Command, 2-8
- Commands, test management, 2-4
  - abbreviations in, 2-6
  - continuation lines, 2-6
  - delimiters, 2-6
  - radices, 2-7
  - test range parameter, 2-7
- Communications boards, 6-1
- DEBUG Command, 2-8
- DESCRIBE Command, 2-9
- Diskette Verify Test, 5-22
- Drive Selection Test,
  - SDTWIN, 5-8
  - SDT218, 5-21
- Dual Port RAM Contention Test,
  - iSBC 86/30 board, 3-5
  - iSBC 86/12A board, 3-10
- ENDREPEAT Command, 2-13
- ERRONLY Command, 2-10
- Error count, 2-8
- Error messages, 2-8, 2-10, 2-18
- Execute From RAM Test, 4-11
- Execution count, 2-8
- EXIT Command, 2-10
- Fixed Patterns Test,
  - iSBC 300A board, 3-10
  - iSBC 86/12A board, 3-9
  - iSBC 86/30 board, 3-5
  - memory boards, 4-10
- Flexible diskette drive
  - SDT tests, 1-9, 5-1, 5-17
- Format Diagnostic Tracks Test,
  - 5-6
- Global Variables, 2-19, 5-22
- IGNORE Command, 2-5, 2-11, 2-16, 2-18
- Interrupting SDTs, 2-20
- iSBC 056A board, 4-1
- iSBC 215 board, 1-7, 5-1
- iSBC 218 board, 5-1, 5-17
- iSBC 300A board, 3-10
- iSBC 309 board, 3-2, 4-1
- iSBC 337 board, 3-1, 3-12
- iSBC 351 board, 6-1
- iSBC 534 board, 6-1, 6-11
- iSBC 544 board, 6-1, 6-22
- iSBC 86/12A board, 3-1, 3-7
- iSBC 86/30 board, 3-1
- LIST Command, 2-11
- loops, command. See REPEAT and ENDREPEAT
- Memory tests, 1-6, 3-2, 3-5, 3-9, 3-10, 4-1

- Microdiagnostic Test, iSBC 215 board, 5-7
- Numeric Processor Extension, 3-1, 3-12
- Overlap Test, iSBC 215, 5-9
- Parallel ports. *See* programmable peripheral interface tests
- Parity Test, iSBC 056A board, 4-11
- PIC. *See* programmable interrupt controller tests
- PIT. *See* programmable interval timer tests
- PPI. *See* programmable peripheral interface tests
- Programmable interrupt controller tests,
  - System Confidence Test, 1-2, 1-5
  - System Diagnostic Tests, iSBC 86/12A board, 3-8
  - iSBC 86/30 board, 3-3
  - iSBC 534 board, 6-17
  - iSBC 544 board, 6-33
- Programmable interval timer tests,
  - System Confidence Test, 1-4
  - System Diagnostic Tests, iSBC 86/30 board, 3-4
  - iSBC 351 board, 6-5
  - iSBC 534 board, 6-15, 6-19
  - iSBC 544 board, 6-31
- Programmable peripheral interface tests,
  - System Confidence Test, 1-6
  - System Diagnostic Tests, iSBC 86/12A board, 3-8
  - iSBC 86/30 board, 3-3
  - iSBC 534 board, 6-17
  - iSBC 544 board, 6-34
- QUERY Command, 2-12
- RAM. *See* memory tests
- RECOGNIZE Command, 2-13
- REPEAT Command, 2-13
- RESET Command, 2-15
- ROM. *See* checksum test
- SCT. *See* System Confidence Test
- SDT. *See* System Diagnostic Tests
- SDTRAM test suite, 4-1
- SDTWIN test suite, 5-1
- SDT218 test suite, 5-17
- SDT309 test suite, 4-14
- SDT337 test suite, 3-12
- SDT351 test suite, 6-1
- SDT534 test suite, 6-11
- SDT544 test suite, 6-22
- SDT8612 test suite, 3-7
- SDT8630 test suite, 3-1
- Sector Selection Test, iSBC 215 board, 5-9
- Seek/Verify Test, iSBC 215 board, 5-7
- Sliding Ones Tests. *See* memory tests
- SUMMARY Command, 2-15
- syntax, 2-5
- System Confidence Test
  - running the tests, 1-1
  - test descriptions, 1-4
  - troubleshooting hints, 1-10
- System Diagnostic Tests, 2-1
  - available tests, 2-1
  - distribution diskettes, 2-2
  - installing on Winchester disk, 2-2
  - interrupting the tests, 2-20
  - running the tests, 2-3
  - See also* Commands, test management
- TEST Command, 2-17
- Test management commands. *See* Commands, test management
- Timer. *See* Programmable interval timer tests
- Track Verify Test, 5-8, 5-22
- Transfer Status Test, 5-6
- Troubleshooting hints, 1-10
- USART tests,
  - System Confidence Test, 1-4
  - System Diagnostic Tests, iSBC 351, 6-6, 6-8, 6-9

- iSBC 534, 6-16, 6-17,  
6-6-20
- iSBC 544, 6-32, 6-34,  
6-35
- Utilities, SDT218, 5-22
- V Command, 2-19
- Winchester,
  - System Confidence Test,  
1-7
  - System Diagnostic Test  
(SDTWIN), 5-1
  - typical disk characteristics,  
5-5
- Write/Read Deleted Data Test,  
5-22
- Write/Read Diagnostic Track  
Test, 5-8, 5-21



### REQUEST FOR READER'S COMMENTS

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

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

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

---

---

---

---

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

---

---

---

---

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

---

---

---

---

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

---

---

---

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

NAME \_\_\_\_\_ DATE \_\_\_\_\_

TITLE \_\_\_\_\_

COMPANY NAME/DEPARTMENT \_\_\_\_\_

ADDRESS \_\_\_\_\_

CITY \_\_\_\_\_ STATE \_\_\_\_\_ ZIP CODE \_\_\_\_\_

(COUNTRY)

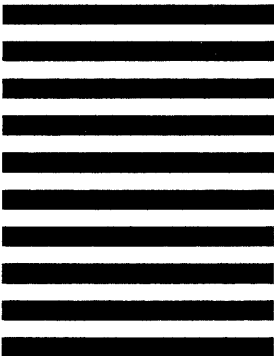
Please check here if you require a written reply.

**WE'D LIKE YOUR COMMENTS . . .**

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



**NO POSTAGE  
NECESSARY  
IF MAILED  
IN THE  
UNITED STATES**



**BUSINESS REPLY MAIL**  
FIRST CLASS PERMIT NO. 79 BEAVERTON, OR

POSTAGE WILL BE PAID BY ADDRESSEE

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

**ISO-N TECHNICAL PUBLICATIONS HF2-1-830**



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

Printed in U.S.A.

144813-004  
IS-009/3K/0584/OSPS  
OEM SYSTEMS