AT&T

305-494
Issue 5

AT&T 3B2 Computer
Off-Line Diagnostic Manual

NOTICE

The information in this document is subject to change without notice. AT&T assumes no responsibility for any errors that may appear in this document.

TRADEMARKS

The following is a listing of the trademarks that are used in this manual:

- INTEL — Registered trademark of Intel Corporation
- UNIX — Registered trademark of AT&T
- WE — Registered trademark of AT&T

ORDERING INFORMATION

Additional copies of this document can be ordered by calling

Toll free:	1-800-432-6600	In the U.S.A.
	1-800-255-1242	In Canada
Toll:	1-317-352-8557	Worldwide

OR by writing to:

AT&T Customer Information Center
Attn: Customer Service Representative
P.O. Box 19901
Indianapolis, IN 46219

TRAINING INFORMATION

The *AT&T Education and Training Catalogue of Courses* and course schedules are free and can be ordered by contacting your account executive or by calling toll free:

1-800-554-6400 (extension 7150)

Training information can also be accessed electronically through our computerized catalogue:

DIAL: 1-800-662-0662 or 1-614-764-5566
LOGIN: comcats
PASSWORD: 88cat

Table of Contents

List of Figures

Chapter 1: Introduction

General

This manual contains the diagnostic phases and tests that run on the AT&T 3B2 computer system. Diagnostics are available for all versions of the System Board (SBD), the Math Accelerator Unit (MAU), the Small Computer System Interface (SCSI), and the feature cards that can be plugged into the 3B2 computer.

Standard Format

To ensure consistency, each phase and test description are presented in a standard format. Each phase description defines the phase name, type, and function. A list of the associated tests, as well as the approximate times required to run them, is also provided. Any important warnings and notes about a phase are also given. The standard phase format is as follows:

Phase Name:

Type:

Function:

Test(s):

Time:

Warnings:

Notes:

Each individual test description provides the test number, its function, and the procedure it uses. The hardware tested and any data returned are also noted in the test description. Any notes about a test are also given. The following shows the standard test format:

Test Number(s):

Function:

Procedure:

Hardware Tested:

Data Returned:

Notes:

Conventions

This document contains sample displays which will help you understand described procedures. The sample displays in this document and the displays on your terminal screen may differ slightly due to improvements in the product after this document was published. Therefore, use the displays in this document as samples of the types of data available. However, the data displayed on your terminal screen accurately reflects the software on your computer. The following conventions are used in this manual to show your terminal input and the system output.

This style of type is used to show system generated responses that are displayed on your screen.

Bold type is used to show inputs entered from your keyboard that are displayed on your screen.

The symbols < and > identify inputs from the keyboard that are not displayed on your screen, such as: <CR> carriage return, <CTRL d> control d, <ESC g> escape g, passwords, and tabs.

This style of italic type is used for notes that provide you with additional information.

Explanation of VOID/NULL in the Equipped Device Table

Upon powerup initialization or reset of the 3B2 computer, the Equipped Device Table (EDT) is created. The EDT is used to indicate what device is in a particular 3B2 computer Input/Output (I/O) slot.

Each time the system is reset, the Identification (ID) code for each device is written into the EDT. The filledt program uses the ID codes to determine the type of device in each slot. The device names are written into the EDT by filledt. The information in the EDT is used when running diagnostics.

The ID is read from the ID/Vector register. If a device is detected in an I/O slot and its ID is not recognized by filledt, the EDT will show that device as VOID.

If a subdevice has an ID of 0x0, it will be shown in the EDT as NULL.

To determine if the VOID or NULL conditions exist, boot the Diagnostic Monitor (DGMON).

Once DGMON is booted, perform the following command:

DGMON> **show** <CR>

If the 3B2 computer is equipped with a given device, and the software associated with that device has been loaded, the EDT should list that device. If a given device is not listed in the EDT, but VOID or NULL is, that device may be considered a suspect when trying to locate a problem.

Manual Organization

The remainder of this manual is organized as follows:

- Chapter 2, "Using Diagnostics," contains guidelines explaining what diagnostics are and how to use them.
- Chapter 3, "System Board Diagnostics," provides descriptions of the diagnostic phases and tests that run on two versions of the System Board (SBD). This chapter also contains diagnostics of the Math Accelerator Unit when those diagnostics are part of the system board diagnostics.
- Chapter 4, "Multiprocessor Diagnostics," contains a description of the diagnostic phases and tests that run on the Multiprocessor Enhancement card.
- Chapter 5, "Cartridge Tape Controller Diagnostics," provides descriptions of the diagnostic phases and tests that run on the Cartridge Tape Controller (CTC) card.
- Chapter 6, "Intelligent Serial Controller Diagnostics," contains descriptions of the diagnostic phases and tests that run on the Intelligent Serial Controller (ISC) card.
- Chapter 7, "Network Interface Diagnostics," gives descriptions of the diagnostic phases and tests that run on the Network Interface (NI) card.
- Chapter 8, "PORTS Diagnostics," gives descriptions of the diagnostic phases and tests that run on the PORTS card.
- Chapter 9, "EPORTS Diagnostics," gives descriptions of the diagnostic phases and tests that run on the EPORTS card.
- Chapter 10, "Remote Management Diagnostics," contains a description of the diagnostic phases and tests that run on the Remote Management card.
- Chapter 11, "Expansion Disk Controller Diagnostics," gives descriptions of the diagnostic phases and tests that run on the Expansion Disk Controller (XDC) card.
- Chapter 12, "Math Accelerator Unit Diagnostics," provides a description of diagnostic phases and tests that run on the Math Accelerator Unit.
- Chapter 13, "Virtual Cache Diagnostics," contains a description of the diagnostic phases and tests that run on the Virtual Cache (VCACHE) card.
- Chapter 14, "Small Computer System Interface Diagnostics," contains a description of the diagnostic phases and tests that run on the Small Computer System Interface.
- Chapter 15, "STARLAN Interface Diagnostics," contains a description of the diagnostic phases and tests that run on the STARLAN Local Area Network Interface.
- Chapter 16, "General Purpose Synchronous Controller Diagnostics," contains a description of the diagnostic phases and tests that run on the General Purpose Synchronous Controller (GPSC-3B) card.

Chapter 2: Using Diagnostics

Introduction

Diagnostics are used to identify and locate hardware problems in the 3B2 computer system. Five versions of the System Board (SBD) as well as the feature cards and all add-on hardware items have off-line diagnostics. By running these diagnostics, you should be able to pinpoint the source of a problem to a particular component of the hardware device. All diagnostics are functionally organized into groups called phases. Each phase consists of one or more diagnostic tests. For example, three diagnostic phases (1, 2, and 3) run on the 3B2 computer Math Accelerator Unit (MAU). Diagnostic tests are normally stored on the system hard disk. The control program for diagnostics runs in the firmware mode. Therefore, to run diagnostics, the 3B2 computer must be in the firmware mode. No other users can be logged on while the system is in the firmware mode.

The diagnostic phases run sequentially. This means that only after all of the tests of one diagnostic phase are completed, is the next phase started. Once an individual phase is started, its tests continue until all the tests pass or until one test fails. Failure of a test prevents subsequent phases from running (command options are available which cause testing to continue even if a phase fails).

Each diagnostic phase generates a report of its activity. These reports are displayed as diagnostic messages on the system console. Each message reports the phase and test number(s) and whether the phase passed or failed. The phase and test number(s) can be used to refer to the phase and test descriptions provided in this manual.

Types of Diagnostic Phases

The 3B2 computer has three types of diagnostic phases:

Normal	Normal diagnostic phases are the standard diagnostics that run each time the system is powered up or taken to the firmware mode and rebooted. The SBD and each feature card in the 3B2 computer have a set of normal diagnostics. You can also manually request normal diagnostics to run using the Diagnostic Monitor (DGMON) firmware program.
Demand	Demand diagnostic phases consist of more extensive tests. Some of the demand diagnostics check optional 3B2 computer hardware; others require more system resources to run. For these reasons, demand diagnostics are not run during powerup. You can only run demand diagnostics with manual requests to the DGMON firmware program.
Interactive	Interactive diagnostics must also be requested manually with the DGMON firmware program. The interactive diagnostics consist of tests that require your input and/or special test conditions. For example, two interactive phases on the Peripheral Ports Controller (PORTS) card require that a special test cable be attached. Be sure that you understand the function of an interactive diagnostic before requesting it because a few of the interactive phases can adversely affect the system software. For example, one interactive diagnostic may destroy the contents of the Nonvolatile Random Access Memory (NVRAM) if there is an early abort or reset. Another interactive diagnostic overwrites the contents of the Time-of-Day clock. The system informs you about the requirements and impact of an interactive diagnostic, and the system asks you to confirm that you want the diagnostic to run.

Procedures for Using Diagnostics

The following procedures explain what you need to know to be able to run diagnostics on the 3B2 computer. The procedures are given in sequential order from power up to power down.

Diagnostics Run During Powerup

When the 3B2 computer is powered up, the Power and Diagnostic indicators on the front of the unit come on simultaneously. After a few seconds, the message SELF-CHECK is displayed on the system console. This message indicates that the basic system sanity tests have started.

If a basic sanity test fails the Diagnostic indicator will begin flashing in a pattern. The flashes on the Diagnostic indicator are the only way for you to know that a sanity test failed. The number of flashes on the Diagnostic indicator represent the following failures:

- 1 flash—The system completed the basic sanity tests and is ready for the firmware password, but no system console device is connected. Check the console and/or cable connection to the system.
- 2 flashes—The processor is not working correctly.
- 3 flashes—There are problems in Programmable Read Only Memory (PROM).
- 4 flashes—The Random Access Memory (RAM) (first 16 kilobytes) failed pattern tests.
- 5 flashes—The Dual Universal Asynchronous Receiver/Transmitter (DUART) failed transmission tests.

For these problems, call your AT&T Service Representative or authorized dealer. By knowing the number of flashes on the Diagnostic indicator, your AT&T Service Representative or authorized dealer will have a better idea of the equipment to bring.

If the basic sanity tests pass, the message DIAGNOSTICS is displayed on the system console. The system then runs all normal diagnostic phases on itself. Demand and interactive diagnostic phases are not run during powerup. It takes approximately 2 minutes for the system to run all of the normal diagnostics. The exact amount of time required to run diagnostics depends on the system's hardware configuration.

The Diagnostic indicator stays on until all of the diagnostics complete successfully or one of the diagnostics tests fail. If all of the diagnostics pass, the message PASSED is displayed on the system console after the DIAGNOSTICS message, and the Diagnostic indicator goes off. If a diagnostic test failed, the message FAILED is displayed after the DIAGNOSTICS message. A SYSTEM FAILURE message is also displayed when a diagnostic test fails. The Diagnostic indicator remains on after a diagnostic failure. The system can be placed in the firmware mode by entering the firmware password (see the procedure "How to Enter Interactive Firmware Mode"). By putting the system into the firmware mode, you can rerun the diagnostics to determine which diagnostic test failed.

How to Go From the UNIX Operating System to the Firmware Mode

To run demand or interactive diagnostics, you must put the 3B2 computer into the firmware mode. This can be accomplished from the system console with the following commands: **shutdown** or **sysadm firmware**. Whichever command you use should be entered after the console login has been entered and the prompt (#) is received.

The **shutdown** command is the safest way to shut down the system because it appropriately cleans up the system before putting it into the requested state. To use **shutdown**, you must log in as the super user (root). The **shutdown** command is entered as follows:

```
# shutdown -i5 -y -g0<CR>
```

The parameter **-i** is the initialization (init) state. State 5 requests the firmware mode. The initialization states are as follows:

- 0 Powerdown
- 1,s,S Single-user mode
- 2 Multiuser mode
- 3,4 Optional modes, user-defined
- 5 Firmware mode
- 6 Stop the UNIX® operating system and reboot.

The **shutdown** command sends a warning message and a final message before it actually begins shutting down the system. As a default, it also asks you for confirmation that you want the system to be shut down. The **-y** argument preanswers this confirmation query with a "yes" answer so that the **shutdown** command can be executed immediately. Also, by default, the time delays between both the warning message and the final message and between the final message and the confirmation query are 60 seconds. For faster execution, the **-g0** argument requests that the time delays be zero (0) seconds. Please note that if any other users are on the system, this argument will log them off immediately without any warning. As a matter of courtesy, either use this argument only when no other users are logged on or be sure to inform other users that you are shutting down the system.

The system response to the **shutdown** command is as follows:

```
Shutdown started.   (Date, Time)

Broadcast Message from root (console) on Name, Date
THE SYSTEM IS BEING SHUT DOWN NOW ! ! !
Log off now or risk your files being damaged.

#
INIT: New run level: 5
The system is coming down.  Please wait.
System services are now being stopped.
Print services stopped.
Stopping job accounting
cron aborted: SIGTERM

The system is down.

SELF-CHECK

FIRMWARE MODE
```

The **sysadm** command is a part of System Administration which is a collection of menu-driven, interactive commands used for system administration. The system administration commands are designed for the novice user. The **firmware** subcommand, which calls the **shutdown** command, is found under the Machine Management Menu (see the *AT&T 3B2 Computer UNIX System V Release 3 Owner/Operator Manual* for more information on all System Administration menus and subcommands). A direct way to call the **firmware** subcommand using **sysadm** is as follows:

```
# sysadm firmware<CR>
```

The system responds with the following:

```
Running subcommand "firmware" from menu "machinemgmt",
MACHINE MANAGEMENT

Once started, this procedure CANNOT BE STOPPED.
Do you want to go to firmware "express"? [y, n, ?, q] y

Shutdown started.   (Date, Time)

Broadcast Message from root (console) on Name, Date
THE SYSTEM IS BEING SHUT DOWN NOW ! ! !
Log off now or risk your files being damaged.

#
INIT: New run level: 5
The system is coming down.  Please wait.
System services are now being stopped.
Print services stopped.
Stopping job accounting
cron aborted: SIGTERM

The system is down.

SELF-CHECK

FIRMWARE MODE
```

Like the **shutdown** command, the **sysadm** command provides a default of 60 seconds between warning messages. However, in this response, **sysadm** first asks you if you want firmware express, which is a shutdown to firmware with no time delays. For immediate shutdown (make sure no other users are on the system), enter a **y**. If you want a different time delay or the default time delay, enter an **n**. The system will then ask you how much time you want between the warning messages. For the default value, simply enter a carriage return. The option **q** allows you to quit the procedure. The option **?** is a help message that explains the warning messages and time delays. If a shutdown is initiated with **sysadm**, the system provides the same response as with the **shutdown** command.

Each of the commands discussed here (**shutdown**, **uadmin**, and **sysadm**) provides a way for you to put the 3B2 computer in the firmware mode. Regardless of which command you use, the system response you receive should conclude with the message FIRMWARE MODE. The next step is to enter the interactive firmware mode.

How to Enter the Interactive Firmware Mode

The firmware password must be used to enter the interactive firmware mode. The initial password which is loaded from firmware is **mcp**. It is recommended that each 3B2 computer be given a unique firmware password. The password (followed by a carriage return) should be entered after the response FIRMWARE MODE is displayed on the system console. Note that the password is not shown on the screen as you enter it. The response to the firmware password is as follows:

```
Enter name of program to execute [ /unix ]:
```

How to Enter the Interactive Diagnostic Monitor

The interactive Diagnostic Monitor (DGMON) firmware program can be entered after the following message appears on the console:

```
Enter name of program to execute [ /unix ]:
```

The **dgmon** command should be entered as the name of the program to execute.

```
Enter name of program to execute [ /unix ]: dgmon<CR>
```

The response to this message is a list of the possible load devices.

```
Possible load devices are:

Option Number   Slot    Type        Name
-------------------------------------
      0          0      INTEGRAL    FD5
      1          0      I/O BUS     SCSI

Enter Load Device Option Number [1  (SCSI)]: <CR>
```

Entering the option **1** or the carriage return will initiate the following response:

```
Possible subdevices are:

Option Number   Subdevice    Name
--------------------------------
      0            0        disk
      1            1        tape
      2            2        disk

Enter Subdevice Option Number [0 (disk)]:<CR>
```

Entering the option **0** or the carriage return will initiate the diagnostic monitor on the hard disk. The following message is then displayed on the system console:

```
        DIAGNOSTIC MONITOR

DGMON >
```

You are now in the interactive DGMON firmware program. Please note that this program accepts both uppercase and lowercase input. The system automatically shifts all lowercase input to uppercase. Do not be confused if an error message echoes your lowercase input as uppercase.

Diagnostic Monitor Commands

To obtain a list of the available 3B2 computer diagnostic commands, enter an **h** (help command) after the diagnostic monitor prompt:

```
DGMON > h<CR>
```

A complete menu of the commands, their options, and descriptions are displayed:

```
DGMON > h
   DIAGNOSTIC
   COMMANDS      OPTIONS                          DESCRIPTION
   ==========    =======                          ===========
   DGN           [DEVICE [DEVICE # ¦ REP=? ¦  DIAGNOSE DEVICE(S)
                      PH=?-? ¦ UCL ¦ SOAK ]] PRINT=[Y ¦ N]

   L(IST)        DEVICE                           LIST DEVICE PHASE TABLE

   S(HOW)        (NONE)                           SHOW EDT

   H(ELP)        (NONE)                           PRINT HELP MENU

   ?             (NONE)                           PRINT HELP MENU

   ERRORINFO     (NONE)                           ENABLE/DISABLE ERROR INFO

   Q(UIT)        (NONE)                           EXIT DGMON

DGMON >
```

The **dgn** command is used to request diagnostics and is described in more detail in the next section of this chapter.

To toggle the error flag to ON or OFF, execute the DGMON program from firmware mode, then enter the **errorinfo** command. You will observe a response similar to the following:

ERROR FLAG IS OFF
TOGGLE [n]

The default is an **n** (no), enter a **y** (yes) to toggle the error flag to ON or OFF.

The **l** (list) command prints diagnostic phase tables for all feature cards loaded in your 3B2 computer. For example, to print a diagnostic phase table for the Enhanced Peripheral Ports Controller (EPORTS) card, enter:

```
DGMON > l eports<CR>
```

The response to this request is the complete phase table for the EPORTS card.

```
DIAGNOSTIC PHASE TABLE FOR EPORTS
PHASE #    PHASE TYPE       PHASE DESCRIPTION
=======    ==========       =================
 1         NORMAL           EPORTS - CIO & Peripheral Sanity
 2         DEMAND           EPORTS - Upper RAM Verification
 3         DEMAND           EPORTS - Lower RAM Verification
 4         DEMAND           EPORTS - ROM Checksum
 5         DEMAND           EPORTS - Upper Chip Select Registers
 6         DEMAND           EPORTS - DMA Control Registers
 7         DEMAND           EPORTS - CPU Writable Registers
 8         DEMAND           EPORTS - Interrupt Control Registers
 9         DEMAND           EPORTS - Lower Chip Select Register
10         DEMAND           EPORTS - PIO Byte Transfers
11         DEMAND           EPORTS - PIO Word Transfers
12         DEMAND           EPORTS - DMA Byte Transfer
13         DEMAND           EPORTS - DMA Word Transfer
14         DEMAND           EPORTS - SCC Basic Sanity
15         DEMAND           EPORTS - DTR & Basic Interrupt Integrity
```

```
ENTER ANY KEY TO CONTINUE<CR>

16         DEMAND           EPORTS - SCC Receive Buffers
17         DEMAND           EPORTS - Basic DMAC & SCC Test
18         DEMAND           EPORTS - Local SCC Interrupts
19         INTERACTIVE      EPORTS - External SCC Interrupts
20         INTERACTIVE      EPORTS - External Drivers & Receivers
21         DEMAND           EPORTS - Complete DMAC & SCC Test

DGMON >
```

Similar phase tables are kept for each card. They are readable with the **l** (lowercased L) command. Simply enter an **l** and the appropriate name. The following shows the names of devices listed in this manual:

aic	Alarm Interface Card (Remote Management)
ctc	Cartridge Tape Controller card
eports	Enhanced Peripheral Ports Controller card
gpsc	General Purpose Synchronous Controller card
isc	Intelligent Serial Controller card
mau	Math Accelerator Unit
mpb	Multiprocessor Board
nau	Network Access Utilities (STARLAN interface)
ni	Network Interface card
ports	Peripheral Ports Controller card
sbd	System Board
scsi	Small Computer System Interface
vcache	Virtual Cache
xdc	Expansion Disk Controller card

The **s** (show) command displays on the console a copy of the Equipped Device Table (EDT). The EDT is created whenever the 3B2 computer is powered up or reset. The EDT lists the current configuration of the 3B2 computer, including complete specifications about the type of device loaded in each 3B2 computer Input/Output (I/O) and Performance Slot. The 3B2 computer determines a device type (PORTS, Network Interface, etc.) by reading the loaded device ID/Vector register. To see the current EDT, enter an **s** after the diagnostic monitor prompt:

```
DGMON > s<CR>
```

The output of the show command might include the following:

```
Current System Configuration

System Board memory size: 1 megabyte(s)

00 - device name = SBD       , occurrence = 0, slot = 00, ID code = 0x01
     boot device = y, board width = double, word width = 2 byte(s),
     req Q size = 0x00, comp Q size = 0x00, console ability = y,
     pump file = n subdevice(s)
     #00 = FD 5      , ID code = 0x01, #01 = HD30      , ID code = 0x03

Press any key to continue

01 - device name = NI        , occurrence = 0, slot = 01, ID code = 0x200
     boot device = n, board width = single, word width = 2 byte(s),
     req Q size = 0x0a, comp Q size = 0x19, console ability = n

Press any key to continue
```

Similar configuration lists are displayed for each device loaded in your 3B2 computer. If the SBD does not receive a match from the value in a device ID/Vector register, it lists that device as *VOID* in the EDT. As indicated in the response above, entering any keyboard character causes the EDT list to continue displaying on the system console. The EDT is also accessible while in the firmware mode with the **edt** command (simply enter **edt** as the name of the program to execute).

The **q** (quit) command is used to exit from the DGMON program (this is explained later in this chapter).

How to Run Diagnostics

The **dgn** command is used to run diagnostics. It has the following format (the options are explained below):

dgn [*device* [*device* # ¦ **rep**=*?* ¦ **ph**=*?-?* ¦ print=? ¦ **ucl** ¦ **soak**]]

device — Specifies the abbreviated name of the feature card on which you want the diagnostics to run. For example, some of the abbreviations are as follows: **sbd** (System Board), **ports** (PORTS card), and **ni** (Network Interface card), etc. If a device name is specified, it must be the second key word in the command line.

device # — Gives you the option of running diagnostic phases on a certain device. Some examples are the SBD (**sbd 0**), one of four I/O PORTS cards (**ports 0-3**), and one Network Interface card (**ni 0**). If a device number is specified, it must be identified immediately after the device name in the command line.

rep=*?* — Specifies the number of times you want the phase(s) to run. The range of allowed repetitions is from 1 through 65,536.

ph=*?-?* — Gives you the option of running a specific phase or a string of phases. When running specific phases, be sure you know which phase you want or you could cause problems to the system. For example, interactive phases will run if they are included in a string of phases, but their effect on the system may be detrimental. When possible, you must request that interactive phases be run separately. Whenever a specific phase is requested, the device to be tested must also be designated. If you are unsure of a phase type or number, use the **l** command to see the on-line phase table for a device.

print=? — The option **print=N** suppresses printing during the run. The option **print=Y** forces printing during the run. By default diagnostic messages are printed only if device #, **ph**=, or **ucl** options are specified. The option **print=?** overrides these defaults. The **print=?** option is only available in UNIX System V Release 3.2.2.

ucl — Gives you the option of running the phases in the unconditional (**ucl**) mode. In this mode, requested diagnostics do not stop when one phase fails. The results of each requested phase are displayed as the phase is completed. This option cannot be used with the **soak** option.

soak — Gives you the option of running the diagnostic phases continuously while storing a record of the failures until testing is completed. A summary of the phase failures is displayed when testing stops. This allows you to check for intermittent problems by comparing the number of failures against the number of times the phase ran. The **soak** option is stopped by either entering any character at the console or using the **rep**= option. The **soak** option cannot be used with interactive phases.

With the **dgn** command options, you can specify the device you want tested, the diagnostics you want to be run, and how you want the testing to be handled. For example, you can either request that all diagnostic phases be run on the SBD, or you can have only Phase 13 run on the SBD. To identify phase numbers and types, you can either use the **l** command to see the on-line phase tables or refer to the phase listings in this manual.

Examples of Diagnostic Requests

The following list gives some examples of diagnostic commands with different options. All of the following examples, if entered, must be followed by a carriage return as indicated.

DGMON> **dgn**<CR>

Runs all normal phases once on each feature card listed in the EDT. If any of the phases fail, testing stops and a failure message is displayed.

DGMON> **dgn sbd**<CR>

Runs all normal phases once on the SBD. If any of the phases fail, testing stops and a failure message is displayed.

DGMON> **dgn sbd ph=1-20**<CR>

Runs Phases 1 through 20 on the SBD. A check of the phase table will show how many phases there are for the particular SBD and the software release you have. Results are displayed as each phase runs.

DGMON> **dgn sbd ph=4**<CR>

Runs diagnostic Phase 4 once on the SBD. Results are displayed as the phase runs.

DGMON> **dgn sbd ph=12-16**<CR>

Runs diagnostic Phases 12 through 16 on the SBD. Each phase is run once or until a failure occurs. Results are displayed as the phases run.

DGMON> **dgn ni 0**<CR>

Runs all normal phases once on the 3B2 computer NI card. If any of the phases fail, testing stops and a failure message is displayed. Results are displayed as the phases run.

DGMON> **dgn eports**<CR>

Runs all normal phases once on all of the EPORTS cards. If any of the phases fail, testing stops and a failure message is displayed. If any of the phases fail, testing stops and a failure message is displayed.

DGMON> **dgn eports 1**<CR>

Runs all normal phases once on EPORTS card number 2. If any of the phases fail, testing stops and a failure message is displayed. Results are displayed as the phases run.

DGMON> **dgn eports 2 ucl**<CR>

Runs all normal phases once on EPORTS card number 3. Testing does not stop if a phase fails. Results are displayed as the phases run.

DGMON> **dgn sbd rep=3 ph=4**<CR>

Runs the SBD diagnostic Phase 4 three times. If any part of the phase fails, testing stops and a failure message is displayed. Results of each repetition of the phase are displayed as the phase is run.

DGMON> **dgn ucl**<CR>

Runs all normal phases once on every device in the EDT. Results of each phase are displayed as it runs. Testing does not stop if a phase fails.

DGMON> **dgn ucl rep=3 print=n**<CR>

Runs all normal phases once on every device in the EDT. Results of each phase are suppressed. Testing does not stop if a phase fails.

DGMON> **dgn ucl rep=3**<CR>

Runs all normal phases three times on every device in the EDT. Results of phases are displayed as the phases run. Testing does not stop if a phase fails.

DGMON> **dgn soak**<CR>

Runs all normal and demand phases on every device in the EDT. Testing does not stop if a phase fails, but stops if a keyboard character is entered. A summary of the phase failures is displayed when testing stops.

DGMON> **dgn ni 0 soak**<CR>

Runs all normal and demand phases on the NI card. Testing does not stop if a phase fails, but stops if a keyboard character is entered. A summary of the phase failures is displayed when testing stops.

DGMON> **dgn sbd soak rep=10 ph=13**<CR>

Runs the SBD diagnostic Phase 13 ten times. Testing does not stop if a phase fails, but stops if a keyboard character is entered. A summary of the phase failures is displayed when testing stops.

DGMON> **dgn soak rep=5**<CR>

Runs all normal and demand phases on all cards five times and displays the results when testing is completed. Testing does not stop if a phase fails, but stops if a keyboard character is entered.

DGMON> **dgn soak rep=5 print=y**<CR>

Runs all normal and demand phases on all cards five times and displays the results of each phase as they run. Testing does not stop if a phase fails, but stops if a keyboard character is entered.

Diagnostic Report Messages

All diagnostic phases use report messages to let you know the results of the diagnostic testing. These messages are displayed on the system console while the diagnostics are running (unless the soak option is used). Each report message identifies the phases and tests run and whether they passed or failed. Phase and test numbers can be used to refer to the descriptions in this manual.

The format of the report messages varies, depending on the feature card being diagnosed. However, most diagnostic messages use one of two standard formats. Diagnostic report messages identify the phase name and number, the time taken to run the phase, the number of tests run, and the results of the test(s). Either ATP (All Tests Passed) or Diagnostic PASSED is displayed if all tests pass. The following are examples of diagnostic report messages.

System Board Diagnostic Report Message

```
<<<   DIAGNOSTIC MODE  >>>
SBD Phase:  1  Test:  Central Processor Unit  Type:  NORMAL
Time Taken = ~1 second
1 2 3 4 5 6 7
*** SBD Phase 1 Diagnostic PASSED ***
```

This message indicates that SBD Phase Number 1 (Central Processor Unit, NORMAL) passed (the phase name is called Test in the message). The phase took approximately 1 second to run. The third line of the message is a list of the test numbers run by this phase (seven total). The final line of the message reports that the phase was completed and the Diagnostic PASSED.

SCSI Card Diagnostic Report Message

```
    <<<   DIAGNOSTIC MODE   >>>
Phase #10
Test: SCSI PIO Byte Transfer
Tests : 8
*** PHASE - ATP ***

     SCSI 0 (IN I/O BUS SLOT 1) DIAGNOSTICS PASSED
```

This message reports that SCSI card Phase 10 [SCSI Programmed Input/Output (PIO) Byte Transfer] passed. Test count indicates that Phase 10 consists of 8 tests.

EPORTS Card Diagnostic Report Message

```
    <<<   DIAGNOSTIC MODE   >>>
Phase: 2
Name: EPORTS - Upper RAM Verification   Type: DEMAND
Test Count: 6  Time: 30 sec.

***** EPORTS Diagnostic Phase 2 PASSED *****

       EPORTS 0 (IN I/O BUS SLOT 2) DIAGNOSTICS PASSED
```

This message reports that the EPORTS Phase 2 (Upper RAM Verification) passed. This phase, which is a demand diagnostic, consists of 6 tests and took 30 seconds to run. The diagnostic report message also specifies which EPORTS card was tested and what I/O slot it is in.

Diagnostic failure messages also vary in format, depending on the feature card that failed. The following examples show some sample failure messages:

System Board Failure Message

```
<<<   DIAGNOSTIC MODE   >>>
Test:  Hard Disk Interface DEMAND
Time taken: ~ 45 seconds per equipped drive
1 2 3 4 5 6
ERROR: Drive 0 not selected!
ERROR: Cannot recalibrate read/write heads to cylinder 0!
Disk = 0  Head = 0  Sector = 0x0  Cylinder = 0x2b8
Command opcode = 0x30        Error status byte = 0xa
Interrupt status byte = 0x0  Unit status byte = 0xa
Controller status = 0x40
Hard Disk 0 Interface Diagnostic Failed
      SBD 0 DIAGNOSTICS FAILED
DGMON >
```

This SBD failure message identifies the phase name (Hard Disk Interface), type (DEMAND), and time required to run the phase (45 seconds per equipped drive). The number of the specific test that failed is identified by the last number in the test number list (Test 6 in this example, which is the last number in the list 1 2 3 4 5 6). The error messages explain that driver 0 was not selected and that the system cannot recalibrate the read/write heads to cylinder 0. You can refer to the test description in Chapter 3, "Version 2 Hardware System Board Diagnostics," to determine the function of Test 6. Supplemental error information in the message includes the disk address and error status. The failure of this phase stops phase testing and the diagnostic monitor prompt (DGMON >) returns.

EPORTS Card Failure Message

```
     <<< DIAGNOSTIC MODE >>>

Phase: 19
Name: EPORTS - External SCC Interrupts  Type: INTERACT
Test Count: 288  Time: 5 sec.

ATTENTION-This phase requires special loop around cables!
 Hit RETURN to continue!
Type 'Q' to quite phase!
```

This message is displayed for the EPORTS card Phase 19, which is an interactive phase. This phase requires an external loop-around cable. If such a cable is attached to the Dual Universal Asynchronous Receiver/Transmitter (DUART), then you simply enter the carriage return for the phase to execute. If the cable is not attached and you still enter a carriage return, the following error message is displayed:

```
Starting Diagnostic Phase.
From Port 0 to Port 3: Character not received!
Read Reg. 0 =44, Read Reg. 1 = 7
From Port 2 to Port 1: Character not received!
Read Reg. 0 =44, Read Reg. 1 = 7
From Port 4 to Port 7: Character not received!
Read Reg. 0 =44, Read Reg. 1 = 7
From Port 6 to Port 5: Character not received!
Read Reg. 0 =44, Read Reg. 1 = 7
DSR FAILURE - test 5 :
      DTR register value 0x0
      DSR register value 0xFF
      DSR register expected value 0x0
All other ports passed interrupt tests!

*** EPORTS Diagnostic Phase 19 FAILED ###

     EPORTS 0 (IN I/O BUS SLOT 2) DIAGNOSTICS FAILED

DGMON >
```

Without the required cable, the phase fails. The failure of this phase prevents subsequent diagnostics from running. The concluding message reports that the failure occurred on the EPORTS card 0 (numbered 0) in expansion I/O bus slot 2 (in a 3B2/800 computer).

Network Interface Card Failure Message

```
Phase #19
Test: NI 3BNET TIME DOMAIN REFLECTOMETER
WARNING: NI - TRANSCEIVER CABLE MUST BE CONNECTED
TIME: 6 seconds          Tests : 21
*** ERROR - test # 10 ****
Actual: beef   Expected: b
NI 0 (IN SLOT 1) DIAGNOSTICS FAILED
```

This message reports the failure of Test 10 of the NI Phase 19. The warning message offers a clue as to why the test failed. It indicates that a transceiver cable must be connected for the phase to run. Because this phase is a demand phase, a warning message may print, but the system does not ask for confirmation from the user before running the diagnostic. Even without the required cable, the phase may attempt to run. The mismatched values for the actual and expected data indicate the failure. The message conclusion identifies the failure as occurring on the NI card 1 (numbered 1) in expansion slot 1.

How to Leave the Interactive Diagnostic Monitor

To exit from the DGMON firmware program, simply enter a **q** (quit) after the diagnostic monitor prompt:

```
DGMON> q<CR>
```

The system is now in the firmware mode. You can either boot back to the UNIX system (or some other program) or power down the system.

How to Boot From the Firmware to the UNIX Operating System

Once in the firmware mode, the response on the console should be the following:

```
Enter name of program to execute [ /unix ]:<CR>
```

To boot to the UNIX operating system, enter a carriage return or **unix** (followed by a carriage return).

The system response is the following:

```
Possible load devices are:
Option Number   Slot    Type    Name
-----------------------------------
0           0           INTEGRAL  FD5
1           0           I/O BUS   SCSI

Enter Load Device Option Number [1 (SCSI)]:<CR>
```

In this response, you are prompted to choose the floppy disk or a SCSI device as the load device. The default is a SCSI device. It is identified as option 1. If you are going to load from a hard disk or cartridge tape, enter a carriage return or a **1** followed by a carriage return. The system response is the following:

```
Option Number    Subdevice    Name
-----------------------------------
       0            0         disk
       1            1         tape

Enter Subdevice Option Number [0  (disk)]:<CR>
```

In this response, the default is the first hard disk. It is identified as option 0. You can either enter a **0** followed by a carriage return or a carriage return to initiate the boot from the hard disk. A complete report of the boot will then be displayed. This report includes any trademark acknowledgments, hardware status or error reports, and the message: *The system is coming up. Please wait.*

The boot report ends with the following login message:

```
The system is ready.

Console Login:
```

You can now log on the UNIX operating system.

How to Power Down From the Firmware Mode

While the system is in the firmware mode, simply press the Power switch to the STANDBY position.

Recommended Procedure for Running Diagnostics

The following procedure illustrates how you can use diagnostics to identify a hardware problem in your 3B2 computer. This procedure provides an organized way for running diagnostics. You should not run diagnostic phases randomly because some phases can damage the file system if they are not executed properly. As you are running the diagnostics, be sure to record the results of the testing (use a printing terminal or connect a printer to your terminal).

Execute the following procedure from the interactive diagnostic monitor:

1. Use the **s** (show) command to see what feature cards are loaded according to the EDT.
2. Execute the **dgn** command with no options to run normal phases on all loaded cards. This command will identify any card that cannot pass its normal diagnostics.
3. Request all normal phases to be run again on any card that returned a failing diagnostic message.
4. If any of the normal phases fail again, repeat those phases using the **soak** or **ucl** option and a specific number of repetitions (**rep**=). Use either the on-line phase table or this manual to verify the failing phase number.
5. Note the failing normal phases and proceed to running the demand phases on all cards. If all of the normal phases pass when you run them, a demand phase may explain the problem with the failing card.
6. If any of the demand phases fail, repeat those phases using the **soak** or **ucl** option and a specific number of repetitions (**rep**=).
7. Note any failing demand phases and proceed to running the interactive phases. Be sure that all normal and demand phases have been run on all cards listed in the EDT before running any interactive phases.
8. To run an interactive diagnostic properly, be sure you are familiar with the phase and that you understand how it is executed. These phases should be run with caution because they can cause unrecoverable system damage. You cannot use the **soak** option with any of the interactive phases.

This procedure should find any hardware problems. If you are qualified, you can repair the hardware problem yourself. For any problem that you do not understand or you cannot repair yourself, please call your AT&T Service Representative or authorized dealer. A good record of the diagnostic results will let the AT&T Service Representative or authorized dealer know what needs to be done to repair the problem. If the results cannot be understood, you will have to run the diagnostics again.

Chapter 3: System Board Diagnostics

Introduction

This chapter contains a description of the diagnostic phases and tests for the 3B2 computer System Board (SBD). This chapter is divided into two sections: Version 2 Hardware SBD Diagnostics and Version 3 Hardware SBD Diagnostics. Version 2 hardware contains the diagnostics for SBDs contained in 3B2/300, 310, and 400 computers. Version 3 hardware contains the diagnostics for SBDs contained in 3B2/500, 522, 600, 622, 700, and 1000 computers. One of the more obvious and significant differences between the two sets of tests has to do with the diagnostics for the Math Accelerator Unit (MAU). The 3B2/300 computers cannot be equipped with a MAU. The MAU diagnostic tests for the 3B2/310 and 400 computers are an integral part of the SBD diagnostic tests. Therefore, MAU diagnostic tests for the 3B2/310 and 400 computers are part of Version 2 hardware diagnostics. The MAU diagnostic tests for the computers with Version 3 hardware, however, are not part of the SBD diagnostic tests. The MAU diagnostic tests for Version 3 hardware SBDs are located in Chapter 12.

The following tables specify which releases of the operating system will run on each computer.

Version 2 Hardware-Operating System Release									
	Release								
Computer	1.0	2.0	2.0.2	2.0.4	2.0.5	2.1	3.0	3.1	3.2
3B2/300	x	x			x	x	x	x	x
3B2/310				x	x	x	x	x	x
3B2/400			x	x	x	x	x	x	x

Version 3 Hardware-Operating System Release			
	Release		
Computer	3.1.1	3.2.1	3.2.2
3B2/500A	x	x	x
3B2/500B		x	x
3B2/522		x	x
3B2/600		x	x
3B2/622		x	x
3B2/700		x	x
3B2/1000			x

Version 2 Hardware System Board Diagnostics

This section contains a description of the diagnostic phases and tests for two different Version 2 hardware System Boards (SBDs). UNIX System V Release 2.0.4 and subsequent releases support the Math Accelerator Unit (MAU) for 3B2/310 and 400 computers, but the previous releases do not. Three normal phases are used to diagnose the MAU. They are shown as Phases 4, 5, and 6 for Release 2.0.4 and subsequent releases in the following table. These phases replace the original normal phases of the same numbers for Release 2.0. As a result, these additional phases cause the original phase numbers to be incremented for Release 2.0. To reduce duplication, the phase numbers for the version that does not support the MAU (Release 2.0) are enclosed in parentheses in the phase number headings in this section. The following table further illustrates this relationship, and it can be used for reference:

PHASE DESCRIPTION	RELEASE 2.0	RELEASE 2.0.4
CPU #2 Normal	1	1
CPU #3 Normal	2	2
CPU #4 Normal	3	3
MAU #1 Normal	-	4
MAU #2 Normal	-	5
MAU #3 Normal	-	6
Memory Management #1 Normal	(4)	7
Memory Management #2 Normal	(5)	8
Memory Management #3 Normal	(6)	9
Memory Management #4 Normal	(7)	10
Dynamic Memory Demand	(8)	11
Nonvolatile Memory Interactive	(9)	12
Sanity/Interval Timer Normal	(10)	13
Control and Status Register Normal	(11)	14
DUART Interactive	(12)	15
Permanent Interrupt Demand	(13)	16
CPU Interrupt System Normal	(14)	17
Direct Memory Controller Normal	(15)	18
Floppy Disk Interface Interactive	(16)	19
Fast Hard Disk Normal	(17)	20
Extended Hard Disk Demand	(18)	21
Time-of-Day Clock Interactive	(19)	22
Hard Disk Media Check Interactive	(20)	23

The SBD is a highly integrated circuit board that serves as the primary element of the 3B2 computer. The following components are found on the SBD:

- MAU (optional for Release 2.0.4 and subsequent releases)
- Memory Management Unit (MMU)
- WE® 32002 Microprocessor [includes Central Processing Unit (CPU) and MMU]
- Dynamic Random Access Memory (DRAM) Controller
- Direct Memory Access (DMA) Subsystem [includes Direct Memory Access Controller (DMAC), Hard Disk Controller, Floppy Disk Controller, and Universal Asynchronous Receiver/Transmitter (UART)]

- Interrupt Structure
- Timers (includes Time-of-Day clock, Periodic timer, and Bus timer)
- Control and Status Register (CSR)
- Erasable Programmable Read Only Memory (EPROM)
- Nonvolatile Random Access Memory (NVRAM).

In addition, the DRAM boards and the Input/Output (I/O) expansion board plug into the SBD.

Twenty diagnostic phases for Release 2.0 or 23 diagnostic phases for Release 2.0.4 and subsequent releases run tests on all major SBD components. The Table of Contents listing will help you locate the descriptions for each SBD phase and its associated tests. The phase and test descriptions are organized numerically in the same order that the phases are run on the 3B2 computer.

Phase #(1)1 — CPU #2 Normal DGN

Phase Name:	WE 32002 Processor Module (cpu32_2)
Type:	Normal
Function:	This phase tests the operation and instruction set of the processor module.
Tests:	Test 1 — checks the logical dyadic instructions. Test 2 — checks the logical triadic instructions. Test 3 — checks the addition functions. Test 4 — checks the subtraction functions. Test 5 — checks the multiplication functions. Test 6 — checks the division functions. Test 7 — checks the modulo arithmetic functions.
Time:	1 second
Warnings:	None
Notes:	In general, most test pass/fail conditions are checked by collecting a hash value which is the composite value obtained by the summing of all test results.

Phase #(1)1 Tests

Test Number:	1
Function:	This test checks the logical dyadic instructions.
Procedure:	The logical instructions OR, AND, and EXCLUSIVE OR are exercised.
Hardware Tested:	The WE 32002 Microprocessor is tested.
Data Returned:	The failing test number (last number displayed on the system console before failure) is returned.
Notes:	The final pass/fail of this test is determined by a hash value that is collected by the correct execution of the instructions.

Test Number:	2
Function:	This test checks the logical triadic instructions.
Procedure:	The logical instructions OR, AND, and EXCLUSIVE OR are exercised.
Hardware Tested:	The WE 32002 Microprocessor is tested.
Data Returned:	The failing test number (last number displayed on the system console before failure) is returned.
Notes:	The final pass/fail of this test is determined by a hash value that is collected by the correct execution of the instructions.

==========

Test Number:	3
Function:	This test checks the ability of the CPU to perform addition.
Procedure:	The results of the calculation form a hash sum which is checked for validity.
Hardware Tested:	The WE 32002 Microprocessor is checked.
Data Returned:	The failing test number (last number displayed on the system console before failure) is returned.
Notes:	None

==========

Test Number:	4
Function:	This test checks the ability of the CPU to perform subtraction.
Procedure:	The results of the calculation form a hash sum which is checked for validity.
Hardware Tested:	The WE 32002 Microprocessor is checked.
Data Returned:	The failing test number (last number displayed on the system console before failure) is returned.
Notes:	None

==========

Test Number:	5
Function:	This test checks the ability of the CPU to perform multiplication.
Procedure:	The results of the calculation form a hash sum which is checked for validity.
Hardware Tested:	The WE 32002 Microprocessor is tested.
Data Returned:	The failing test number (last number displayed on the system console before failure) is returned.
Notes:	None

Test Number:	6
Function:	This test checks the ability of the CPU to perform division.
Procedure:	The results of the calculation form a hash sum which is checked for validity.
Hardware Tested:	The WE 32002 Microprocessor is checked.
Data Returned:	The failing test number (last number displayed on the system console before failure) is returned.
Notes:	None

==========

Test Number:	7
Function:	This test checks the ability of the CPU to perform modulo arithmetic.
Procedure:	The results of the calculation form a hash sum which is checked for validity.
Hardware Tested:	The WE 32002 Microprocessor is tested.
Data Returned:	The failing test number (last number displayed on the system console before failure) is returned.
Notes:	None

Phase #(2)2 — CPU #3 Normal DGN

Phase Name:	WE 32002 Processor Module (cpu32_3)
Type:	Normal
Function:	This phase tests the operation and instruction set of the processor module.
Tests:	Test 1 — checks the extract field instruction. Test 2 — checks the insert field instruction. Test 3 — checks the move block instruction. Test 4 — checks the test swap interlock instruction. Test 5 — checks the bit test instruction. Test 6 — checks the Save and Restore register instructions. Test 7 — checks the call procedure and RET instructions.
Time:	1 second
Warnings:	None
Notes:	None

Phase #(2)2 Tests

Test Number:	1
Function:	This test checks the extract field instruction.
Procedure:	The extract field instruction is performed for all widths and offsets.
Hardware Tested:	The WE 32002 Microprocessor is tested.
Data Returned:	The failing test number (last number displayed on the system console before failure) is returned.
Notes:	The results of the test are compared to a computed hash sum that contains all of the intermediate results. Register r7 contains the hash value that is generated by adds and rotates.

==========

Test Number:	2
Function:	This test checks the insert field instruction.
Procedure:	The insert field instruction is performed for all widths and offsets.
Hardware Tested:	The WE 32002 Microprocessor is tested.
Data Returned:	The failing test number (last number displayed on the system console before failure) is returned.
Notes:	The results of the test are compared to a computed hash sum that contains all of the intermediate results. Register r7 contains the hash value that is generated by adds and rotates.

Test Number: 3

Function: This test checks the move block instruction.

Procedure: A block of 16 words is moved from location 0x0 to the Random Access Memory (RAM) scratch area. After the block move, the locations are compared by physical addressing.

Hardware Tested: The WE 32002 Microprocessor is tested.

Data Returned: The failing test number (last number displayed on the system console before failure) is returned.

Notes: The following instruction sequence is used in this test:

```
while( r2 < 0)
{
     *r1 = r0;
     --r2;
     r0++;
     r1++;
}
```

==========

Test Number: 4

Function: This test checks the test swap interlocked instruction.

Procedure: Tests for word, half word, and byte swap are performed.

Hardware Tested: The WE 32002 Microprocessor is tested.

Data Returned: The failing test number (last number displayed on the system console before failure) is returned.

Notes: The global locations in this test are used only because they are available from other tests and are conveniently located in RAM.

==========

Test Number: 5

Function: This test checks the bit test instruction for word, half word, and byte.

Procedure: The instruction is exercised, and it compiles a hash value based on the results in the condition code portion of the Microprocessor Status register.

Hardware Tested: The WE 32002 Microprocessor is tested.

Data Returned: The failing test number (last number displayed on the system console before failure) is returned.

Notes: The results of the test are confirmed by comparing them to a generated hash sum that contains all of the intermediate results. Register r6 contains the hash value that is generated by adds and rotates.

Test Number:	6
Function:	This test checks the Save and Restore register instructions.
Procedure:	Registers r3 through r6 are loaded with a unique pattern that will be loaded into and retrieved from the registers to verify that the stack is correctly loaded and unloaded.
Hardware Tested:	The WE 32002 Microprocessor is tested.
Data Returned:	The failing test number (last number displayed on the system console before failure) is returned.
Notes:	The pass/fail condition of this test is determined by a hash value that is generated by the addition of registers r3 through r6 after the restore instruction is executed.

==========

Test Number:	7
Function:	This test checks the call procedure and RET instructions.
Procedure:	A small subroutine call is set up that returns a value in r0 to test that the call procedure and RET instructions function properly.
Hardware Tested:	The WE 32002 Microprocessor is tested.
Data Returned:	The failing test number (last number displayed on the system console before failure) is returned.
Notes:	None

Phase #(3)3 — CPU #4 Normal DGN

Phase Name:	WE 32002 Processor Module (cpu32_4)
Type:	Normal
Function:	This phase tests the operation and instruction set of the processor module.
Tests:	Test 1 — checks the PUSHAW, PUSHW, and POPW instructions. Test 2 — checks the unconditional jump instruction. Test 3 — checks the branch to subroutine and return from subroutine instructions. Test 4 — checks the jump to subroutine and return from subroutine instructions. Test 5 — checks the exception for the divide-by-zero instruction.
Time:	1 second
Warnings:	None
Notes:	None

Phase #(3)3 Tests

Test Number:	1
Function:	This test verifies the PUSHAW, PUSHW, and POP instructions.
Procedure:	The instructions are performed.
Hardware Tested:	The WE 32002 Microprocessor is tested.
Data Returned:	The failing test number (last number displayed on the system console before failure) is returned.
Notes:	None

==========

Test Number:	2
Function:	This test checks the unconditional jump instruction.
Procedure:	A jump instruction is performed. If the jump is not accomplished, the next instruction indicates an error.
Hardware Tested:	The WE 32002 Microprocessor is tested.
Data Returned:	The failing test number (last number displayed on the system console before failure) is returned.
Notes:	None

Test Number:	3
Function:	This test checks the branch to subroutine and return from subroutine instructions.
Procedure:	A small subroutine is set up that, after entered, returns on a conditional value of zero.
Hardware Tested:	The WE 32002 Microprocessor is returned.
Data Returned:	The failing test number (last number displayed on the system console before failure) is returned.
Notes:	None

==========

Test Number:	4
Function:	This test checks the jump to subroutine and return from subroutine instructions.
Procedure:	A small subroutine is set up that, after entered, returns on a conditional value of zero.
Hardware Tested:	The WE 32002 Microprocessor is returned.
Data Returned:	The failing test number (last number displayed on the system console before failure) is returned.
Notes:	This test is very similar to Test 3 except that it exercises the jump instruction rather than the branch to subroutine instruction. It also uses a different return condition.

==========

Test Number:	5
Function:	This test ensures that an exception occurs when the instruction calls for a divide-by-zero.
Procedure:	The test divides by zero and verifies that an exception is generated.
Hardware Tested:	The WE 32002 Microprocessor is tested.
Data Returned:	The failing test number (last number displayed on the system console before failure) is returned.
Notes:	A failure of this test indicates that a divide-by-zero instruction did not cause an exception.

Phase #4 — Math Accelerator Unit #1

Phase Name:	WE 32106 Math Accelerator Unit (MAU) (mau1_tst)
Type:	Normal
Function:	This phase tests for the presence of a MAU device. If it is found, the phase pattern tests the MAU Data bus, operand registers F0 through F3, and the MAU Data register.
Tests:	Test 1 — determines if a MAU is present on the SBD under test. Test 2 — pattern checks the MAU Data bus by walking a one through a field of zeros as a single precision operand to register F0. Tests 3 through 6 — pattern checks the MAU registers F0 through F3 as a double-extended precision operand with a modified data pattern of 0x55's and 0xaa's. Test 7 — pattern checks the MAU Data register as a double-extended precision operand with a modified data pattern of 0x55's and 0xaa's.
Time:	1 second
Warnings:	None
Notes:	If a MAU is not detected on the board under test, this test phase displays NTR (No Test Run) and returns an NTR condition. All operand pattern tests are checked for register-to-register cross talk.

Phase #4 Tests

Test Number:	1
Function:	This test determines if the MAU is present on the SBD.
Procedure:	A SPOP NOP is issued to see if the MAU device is present.
Hardware Tested:	The MAU is tested.
Data Returned:	A warning message indicating diagnostics not run is returned.
Notes:	None

==========

Test Number:	2
Function:	This test pattern checks the MAU Data bus.
Procedure:	A one is walked through a field of zeros as a single precision operand by moving the operand to and from register F0.
Hardware Tested:	The MAU is tested.
Data Returned:	The number of the test that failed, the actual data, and the expected data are returned.
Notes:	None

Test Numbers:	3 through 6
Function:	These tests pattern check MAU registers F0 through F3.
Procedure:	Each register is tested with modified 0x55 and 0xaa patterns, checking the other registers for cross talk.
Hardware Tested:	The MAU is tested.
Data Returned:	The number of the test that failed, the actual data, and the expected data are returned.
Notes:	None

==========

Test Number:	7
Function:	This test pattern checks the MAU Data register.
Procedure:	The Data register is tested with a modified data pattern of 0x55's and 0xaa's.
Hardware Tested:	The MAU is tested.
Data Returned:	The number of the test that failed, the actual data, and the expected data are returned.
Notes:	None

Phase #5 — Math Accelerator Unit #2

Phase Name: WE 32106 Math Accelerator Unit (MAU) (mau2_tst)

Type: Normal

Function: This phase tests that a faulty MAU can be detected. Also, all of the "Sticky" and Mask bits of the MAU Auxiliary Status Register (ASR) are tested for the ability to be set, cleared, and masked.

Tests: Test 1 — causes a MAU fault condition by dividing by zero, testing the ASR divide-by-zero "Sticky" (QS) and Mask bits (QM).

Test 2 — checks the ASR Invalid Operation "Sticky" (IS) and Mask bits (IM).

Test 3 — checks the ASR Overflow "Sticky" (OS) and Mask bits (OM).

Test 4 — checks the ASR Inexact "Sticky" (PS) and Mask bits (PM).

Test 5 — checks the ASR Underflow "Sticky" (US) and Mask bits (UM).

Test 6 — checks the ASR Integer Overflow (IO) Indicator bit.

Test 7 — checks the ASR Unordered (UO) Indicator bit.

Test 8 — checks the ASR Nontrapping Not a Number (NAN) Control (NTNC) bit.

Time: 1 second

Warnings: None

Notes: If a MAU is present, the tests are run. Otherwise, this function returns NTR (No Tests Run). The Negative (N), Zero (Z), Result Available (RA), and Partial Remainder (PR) bits are tested during Phase 6.

ASR																			
31	25	24	23-22	21	20	19	18	17	16	14	13	12	11	10	9	8	7	6	5
RA	ECP	NTNC	RC	N	Z	IO	PS	CSC	UO	IM	OM	UM	QM	PM	IS	OS	US	QS	PR

Phase #5 Tests

Test Number: 1

Function: This test checks that a faulty MAU can be detected.

Procedure: A divide-by-zero is performed, and the fact that an exception was generated is confirmed. Then, the divide-by-zero fault is inhibited, and the process is repeated. The fact that an exception is not generated is confirmed. In both cases, the MAU_ASR is checked to ensure that the correct flags were set.

Hardware Tested: The MAU is tested.

Data Returned: The number of the test that failed, the actual data, and the expected data are returned.

Notes: None

==========

Test Number: 2

Function: This test checks the ASR Invalid Operation "Sticky" (IS) and Mask bits (IM).

Procedure: An Integer to Float Conversion (ITOF) is performed with the source specified as MAU register F0. It is confirmed that an exception was generated. Then, Invalid Operation exceptions are inhibited, and the procedure is repeated. It is confirmed that an exception was not generated. In both cases, it is checked that the correct MAU_ASR flags were set.

Hardware Tested: The MAU is tested.

Data Returned: The number of the test that failed, the actual data, and the expected data are returned.

Notes: None

==========

Test Number: 3

Function: This test checks the ASR Overflow "Sticky" (OS) and Mask bits (OM).

Procedure: An Overflow exception is forced by using the MAU division instruction. It is confirmed that an exception was taken. Then, Overflow exceptions are inhibited, and the procedure is repeated. It is confirmed that an exception was not taken. In both scenarios, the MAU_ASR is checked for correct flag generation.

Hardware Tested: The MAU is tested.

Data Returned: The number of the test that failed, the actual data, and the expected data are returned.

Notes: None

Test Number:	4
Function:	This test checks the Inexact "Sticky" bit (PS) and Mask bits (PM).
Procedure:	An Inexact exception is caused by using the MAU RTOI instruction. It is confirmed that an exception was generated. Then, Inexact exceptions are blocked, and the procedure is repeated. It is confirmed that an exception was not taken. In both cases, the MAU_ASR is checked for correct flag values.
Hardware Tested:	The MAU is tested.
Data Returned:	The number of the test that failed, the actual data, and the expected data are returned.
Notes:	None

==========

Test Number:	5
Function:	This test checks the ASR Underflow "Sticky" bit (US) and Mask bits (UM).
Procedure:	A MAU Underflow exception is caused by using the MAU division instruction. It is confirmed that an exception was generated. Then, the Underflow exceptions are inhibited, and the procedure is repeated. It is checked that an exception was not generated. In both tests, the correct setting of MAU_ASR flag variables is checked.
Hardware Tested:	The MAU is tested.
Data Returned:	The number of the test that failed, the actual data, and the expected data are returned.
Notes:	None

==========

Test Number:	6
Function:	This test checks the ASR Integer Overflow (IO) Indicator bit.
Procedure:	An exception is generated by using the MAU FTOI instruction. The fact that an exception was taken is confirmed, and the correct setting of the MAU_ASR flags is checked.
Hardware Tested:	The MAU is tested.
Data Returned:	The number of the test that failed, the actual data, and the expected data are returned.
Notes:	None

Test Number:	7
Function:	This test checks the ASR Unordered (UO) Indicator bit.
Procedure:	An Unordered exception is forced by using the MAU CMP instruction. It is checked that an exception was taken and that the MAU_ASR set the correct flag values.
Hardware Tested:	The MAU is tested.
Data Returned:	The number of the test that failed, the actual data, and the expected data are returned.
Notes:	None

==========

Test Number:	8
Function:	This test checks the ASR NTNC bit.
Procedure:	An Invalid Operation condition is caused with the NTNC bit set by performing an ITOF with the source specified as a MAU register. It is confirmed that an exception was generated and that the MAU_ASR has the correct flags set.
Hardware Tested:	The MAU is tested.
Data Returned:	The number of the test that failed, the actual data, and the expected data are returned.
Notes:	None

Phase #6 — Math Accelerator Unit #3

Phase Name:	WE 32106 MAU (mau3_tst)
Type:	Normal
Function:	This phase tests all of the possible MAU operation codes.
Tests:	Tests 1 through 8 — check the CMP, CMPS, CMPE, and CMPES operation codes. Tests 9 and 10 — check the add and subtract operation codes. Tests 11 through 13 — check the multiply, divide, and REM operation codes. Test 14 — checks the NEG operation code. Test 15 — checks the ABS operation code. Test 16 — checks the SQRT operation code. Tests 17 through 21 — check the RTOI, ITOF, DTOF, FTOD, and FTOI operations.
Time:	1 second
Warnings:	None
Notes:	If a MAU is present, the tests are run. Otherwise, this function returns NTR (No Tests Run). The Negative (N), Zero (Z), Result Available (RA), and Partial Remainder (PR) bits are tested during this test phase.

Phase #6 Tests

Test Numbers:	1 through 8
Function:	These tests check the CMP, CMPS, CMPE, and CMPES operation codes.
Procedure:	Operand1 and operand2 are set to 123.0. Each compare instruction is run checking for equality. Operand1 is set to 100.0. Each compare instruction is run checking for inequality.
Hardware Tested:	The MAU is tested.
Data Returned:	The number of the test that failed, the actual data, and the expected data are returned.
Notes:	None

Test Numbers: 9 and 10

Function: These tests check the add and subtract operation codes. Many of the following tests (9 through 21) use the MAU compare instruction to determine if the test passed.

Procedure:
Operand1 is set to 39.0 and operand2 is set to 12.0.
Move operand2 to MAU F0 and operand1 to MAU F1.
Perform an Add.
Set expctd_rslt to 51.0.
Perform MAU F0 + MAU F1 -> result.
Perform a Subtract.
Set expctd_rslt to 27.0.
Perform MAU F1 - MAU F0 -> result.
Move result to MAU F0 and
expctd_rslt to MAU F1.
Compare MAU F0 with MAU F1 and check for equality.

Hardware Tested: The MAU is tested.

Data Returned: The number of the test that failed, the actual data, and the expected data are returned.

Notes: None

==========

Test Numbers: 11 through 13

Function: These tests check the multiply, divide, and REM operation codes.

Procedure:
Clear the result variable.
Perform a Multiply.
Set operand1 to 239.0 and operand2 to 11.0.
Set expctd_rslt to 2629.0.
Move operand1 to MAU F1 and operand2 to MAU F0.
Perform MAU F0 * MAU F1 -> result.
Perform a Divide.
Set operand1 to 125.0 and operand2 to 5.0.
Set expctd_rslt to 25.0.
Move operand1 to MAU F1 and operand2 to MAU F0.
Perform MAU F1 / MAU F0 -> result.
Perform a REM.
Set operand2 to 10.0.
Set expctd_rslt to 5.0.
Move operand1 to MAU F1 and operand2 to MAU F0.
Perform MAU F1 % MAU F0 -> result.
Move result to MAU F0 and expctd_rslt to MAU F1.
Compare MAU F0 and MAU F1 and check for equality.

Hardware Tested: The MAU is tested.

Data Returned: The number of the test that failed, the actual data, and the expected data are returned.

Notes: None

Test Number:	14
Function:	This test checks the Negative (NEG) operation code.
Procedure:	Set operand1 to 239.0 and expctd_rslt to -239.0. Move operand1 to MAU F0. Perform NEG MAU F0. Compare MAU F0 and expctd_rslt and check for equality.
Hardware Tested:	The MAU is tested.
Data Returned:	The number of the test that failed, the actual data, and the expected data are returned.
Notes:	None

==========

Test Number:	15
Function:	This test checks the Absolute (ABS) operation code.
Procedure:	Set operand1 to -239.0 and expctd_rslt to 239.0. Move operand1 to MAU F0. Perform ABS MAU F0. Compare MAU F0 and expctd_rslt and check for equality.
Hardware Tested:	The MAU is tested.
Data Returned:	The number of the test that failed, the actual data, and the expected data are returned.
Notes:	None

==========

Test Number:	16
Function:	This test checks the Square Root (SQRT) operation code.
Procedure:	Set operand1 to 144.0 and expctd_rslt to 12.0. Move operand1 to MAU F0. Perform SQRT MAU F0. Compare MAU F0 and expctd_rslt and check for equality.
Hardware Tested:	The MAU is tested.
Data Returned:	The number of the test that failed, the actual data, and the expected data are returned.
Notes:	None

Test Numbers:	17 through 21
Function:	These tests check the RTOI, ITOF, FTOI, DTOF, and FTOD operations.
Procedure:	RTOI: Set operand1 to 123.45 and expctd_rslt to 123.0. Perform RTOI operand1 -> MAU F0. Compare expctd_rslt and MAU F0 checking for equality. ITOF: Set operand1 to 123. Perform ITOR operand1 -> MAU F0. Compare expctd_rslt and MAU F0 checking for equality. FTOI: Set operand1 to 123.05, expctd_rslt to 123. Clear result. Move operand1 to MAU F0. Perform FTOI MAU F0 -> result. Compare result and expctd_rslt checking for equality. DTOF: Set operand1 to 0x123a, expctd_rslt to 123.0. Perform DTOF operand1 -> MAU F0. Compare expctd_rslt and MAU F0 checking for equality. FTOD: Set operand1 to 123.00, expctd_rslt to 0x123a. Move operand1 to MAU F0. Perform FTOD MAU F0 -> result. Compare result and expctd_rslt checking for equality.
Hardware Tested:	The MAU is tested.
Data Returned:	The number of the test that failed, the actual data, and the expected data are returned.
Notes:	None

Phase #(4)7 — Memory Management Unit #1

Phase Name:	Memory Management Unit Peripheral Mode (mmu1_tst)
Type:	Normal
Function:	This phase checks the internal registers and descriptor caches of the Memory Management Unit (MMU).
Tests:	Test 1 — checks the Virtual Address register. Tests 2 and 3 — check the SDCs. Tests 4 and 5 — check the LHPDC. Tests 6 and 7 — check the RHPDC. Test 8 — checks the SRAMA. Test 9 — checks the SRAMB. Tests 10 and 11 — check the Fault Address register and the Configuration register.
Time:	1 second
Warnings:	None
Notes:	None

Phase #(4)7 Tests

Test Number:	1
Function:	This test checks the Virtual Address register.
Procedure:	Read and write the Virtual Address register using full binary bit patterns.
Hardware Tested:	The Virtual Address register is tested.
Data Returned:	The failing test number, the failing test pattern, and the actual test pattern read are returned.
Notes:	None

Test Numbers:	2 and 3
Function:	These tests ensure that the Segment Descriptor Caches (SDCs) can be written to and read from.
Procedure:	A pattern test on each of the 32 segment descriptor locations is performed. Test 2 — checks the lower half of the SDC area. Test 3 — checks the upper half of the SDC area.
Hardware Tested:	The SDCs are tested.
Data Returned:	The failing test number, the failing test pattern, the actual test pattern read, and the failing SDC address are returned.
Notes:	None

==========

Test Numbers:	4 and 5
Function:	These tests ensure that the Left-Hand Page Descriptor Cache (LHPDC) can be written to and read from.
Procedure:	Four binary bit patterns are written and read back for comparison. Test 4 — checks the lower half of the LHPDC area. Test 5 — checks the upper half of the LHPDC area.
Hardware Tested:	The LHPDC is tested.
Data Returned:	The failing test number, the failing test pattern, the actual test pattern read, and the failing LHPDC address are returned.
Notes:	

==========

Test Numbers:	6 and 7
Function:	These tests ensure that the Right-Hand Page Descriptor Cache (RHPDC) can be written to and read from.
Procedure:	Four binary bit patterns are written and read back for comparison. Test 6 — checks the lower half of the RHPDC area. Test 7 — checks the upper half of the RHPDC area.
Hardware Tested:	The RHPDC is tested.
Data Returned:	The failing test number, the failing test pattern, the actual test pattern read, and the failing RHPDC address are returned.
Notes:	None

Test Number:	8
Function:	This test ensures that the Section Random Access Memory A (SRAMA) can be written to and read from.
Procedure:	Four binary patterns are written and read back for comparison.
Hardware Tested:	The SRAMA is tested.
Data Returned:	The failing test number, the failing test pattern, the actual test pattern read, and the failing SRAMA address are returned.
Notes:	None

==========

Test Number:	9
Function:	This test ensures that the Section Random Access Memory B (SRAMB) can be written to and read from.
Procedure:	Four binary patterns are written and read back for comparison.
Hardware Tested:	The SRAMB is tested.
Data Returned:	The failing test number, the failing test pattern, the actual test pattern read, and the failing SRAMB address are returned.
Notes:	None

==========

Test Numbers:	10 and 11
Function:	These tests perform pattern checks on the Fault Address and Configuration registers.
Procedure:	Each register is written with four binary patterns, and a comparison is made.
Hardware Tested:	The Fault Address register and the Configuration register are tested.
Data Returned:	The failing test number, the failing test pattern, the actual test pattern read, and the failing register address are returned.
Notes:	None

Phase #(5)8 — Memory Management Unit #2

Phase Name:	Memory Management Unit Flushing (mmu2_tst)
Type:	Normal
Function:	This phase tests the flushing capability of the Memory Management Unit (MMU).
Tests:	Test 1 — checks the SDC single entry flush. Test 2 — checks the LHPDC single entry flush. Test 3 — checks the RHPDC single entry flush. Test 4 — checks the Section 0 flush. Test 5 — checks the Section 1 flush. Test 6 — checks the Section 2 flush. Test 7 — checks the Section 3 flush.
Time:	1 second
Warnings:	None
Notes:	None

Phase #(5)8 Tests

Test Number:	1
Function:	This test checks the single entry flushing capability of the Segment Descriptor Caches (SDCs).
Procedure:	The cache is initialized by writing to Section Random Access Memory A (SRAMA). The entries to be flushed are set up by setting the G bits and specific tags. The entries are flushed by setting up the correct unique tags and indexing into the cache to clear the G bits. Entries are verified to be flushed by reading all cache entries and making sure no entry is marked as good.
Hardware Tested:	The MMU chip is tested.
Data Returned:	The failing test number (last number displayed on the system console before failure) is returned.
Notes:	None

Test Numbers:	2 and 3
Function:	These tests check the single entry flushing capability of the Page Descriptor Caches (PDCs).
Procedure:	The cache is initialized by writing to SRAMA to flush them. The entries that are to be flushed are set up by setting the G bits and specific tags. The entries are flushed by setting up the correct unique tags and indexing into the cache to clear the G bits. The entries are verified to be flushed by reading all entries in the cache to make sure that no entry is marked as good. Test 2 — checks the Left-Hand Page Descriptor Cache (LHPDC) area. Test 3 — checks the Right-Hand Page Descriptor Cache (RHPDC) area.
Hardware Tested:	The MMU chip is tested.
Data Returned:	The failing test number (last number displayed on console) is returned.
Notes:	

==========

Test Numbers:	4 through 7
Function:	These tests perform a Section flush on Sections 0 through 3.
Procedure:	All G bits are set in Sections 0 through 3. A write to SRAMA is made to clear the set bits. The G bit is checked for correct action.
Hardware Tested:	The MMU chip is tested.
Data Returned:	The failing test number (last number displayed on the system console before failure) is returned.
Notes:	None

Phase #(6)9 — Memory Management Unit #3

Phase Name: MMU Translation and Referenced and Modified (R&M) Updating (mmu3_tst)

Type: Normal

Function: This phase tests the translation capability and the R&M update.

Tests: Test 1 — checks the paged segment address translation.

Test 2 — checks the contiguous segment address translation.

Tests 3 through 6 — perform the R&M updating of the MMU.

Time: 1 second

Warnings: None

Notes: The 3B2 computer Memory Management Unit (MMU) test mapping scheme is as follows:

ADDRESS	DATA	SEGMENT(S)
0x00000000	EPROM Read/Execute	0 through 1
0x00040000	Miscellaneous Hardware Device and I/O Boards Read/Write	2 through 255
0x02000000	FW RAM Area Read/Write	256
0x02020000	MMU Text Read/Execute	257
0x02040000	MMU Tables Read/Write	258
0x02060000 0x0207ffff	Scratch RAM Read/Write	259

The Memory Management Tables in memory are set as follows (there are 64 pages per segment).

SDT TABLE	PDT TABLE
Segment 0 (0x02040000)	Pages 0-63 (0x02040900)
Segment 1 (0x02040008)	Pages 64-127 (0x0204a00)
Segments 2-258	Pages 128-16575
Segment 259 (0x02040818)	Page 16576 (0x02050c00)

Phase #(6)9 Tests

Test Numbers: 1 and 2

Function: These tests check the translation of paged and contiguous segments.

Procedure: The test procedure is as follows:

1. The Page Descriptor Table (PDT) and Segment Descriptor Table (SDT) are set up.
2. Section Random Access Memory A (SRAMA) and SRAMB are set up.
3. A data pattern is written to RAM (0x02060000 - 0x020403ff) in virtual mode.
4. The data are read back in physical mode to confirm that the translation occurred correctly.

Hardware Tested: The MMU chip is tested.

Data Returned: The failing test number, the expected data, and the actual data read are returned.

Notes: None

==========

Test Numbers: 3 through 6

Function: These tests check the R&M updating of the MMU.

Procedure: Tables are set up, a segment is referenced, a segment is written, and a signature is read in this procedure.

Hardware Tested: The MMU chip is tested.

Data Returned: The failing test number (last number displayed on the system console before failure) is returned.

Notes: None

Phase #(7)10 — Memory Management Unit #4

Phase Name: Memory Management Unit Fault

Type: Normal

Function: This phase tests the fault recognition ability of the Memory Management Unit (MMU).

Tests: Test 1 — forces a SDT length fault.

Test 2 — forces a page write fault.

Test 3 — forces an invalid segment descriptor fault.

Test 4 — forces a segment not present fault.

Test 5 — forces an object trap fault.

Time: 1 second

Warnings: The descriptor tables for memory management are set up so that the Diagnostic Monitor (DGMON) code cannot be executed while in the virtual mode.

Notes: None

Phase #(7)10 Tests

Test Number: 1

Function: This test checks the Segment Descriptor Table (SDT) length fault.

Procedure: A virtual address is referenced that has a segment select field greater than the SDT length field of Section Random Access Memory B (SRAMB).

Hardware Tested: The MMU chip is tested.

Data Returned: The failing test number, the MMU fault code, and the Fault Address register are returned.

Notes: None

==========

Test Number: 2

Function: This test checks the page write fault.

Procedure: The W bit is set in a Page Descriptor Cache (PDC) entry and writes are made to the virtual address that corresponds to that PDC entry.

Hardware Tested: The MMU chip is tested.

Data Returned: The MMU Fault Code register value and the MMU Fault Address register value are returned.

Notes: None

Test Number:	3
Function:	This test checks the invalid segment descriptor fault.
Procedure:	The descriptor tables are set up for contiguous segments. The caches are flushed. The V bit is cleared in an SDT entry. The virtual address is referenced that corresponds to that entry.
Hardware Tested:	The MMU chip is tested.
Data Returned:	The MMU Fault Code register value and the MMU Fault Address register value are returned.
Notes:	None

==========

Test Number:	4
Function:	This test checks the segment not present fault.
Procedure:	The descriptor tables are set up for contiguous segments. The caches are flushed. The P bit is cleared in an SDT entry. The virtual address is referenced that corresponds to that entry.
Hardware Tested:	The MMU chip is tested.
Data Returned:	The MMU Fault Code register value and the MMU Fault Address register value are returned.
Notes:	None

==========

Test Number:	5
Function:	This test checks the object trap fault.
Procedure:	The descriptor tables are set up for contiguous segments. The caches are flushed. The T bit is set in an SDT entry. The virtual address that corresponds to that entry is referenced.
Hardware Tested:	The MMU chip is tested.
Data Returned:	The MMU Fault Code register value and the MMU Fault Address register value are returned.
Notes:	None

Phase #(8)11 — Dynamic Memory

Phase Name:	Dynamic Memory (dram_tst)
Type:	Demand
Function:	This phase checks the Dynamic Random Access Memory (DRAM) by performing the tests listed below.
Tests:	Test 1 — checks all memory accesses on a word basis, checking for shorted address lines. Test 2 — checks all memory accesses on a word basis with alternate data patterns through the memory range. Zeros and ones are followed by ones and zeros. Test 3 — verifies dynamic memory is being refreshed by storing a data pattern in memory, waiting approximately 23 seconds, and confirming data retention. The test is repeated with complements of the previous data test pattern. Test 4 — checks memory accesses by writing words and reading back two half words; tests memory accesses by writing two half words and reading back 4 bytes; and tests memory accesses by writing and reading 4 bytes.
Time:	70 seconds
Warnings:	None
Notes:	None

Phase #(8)11 Tests

Test Number: 1

Function: This test checks all memory accesses on a word basis, checking for shorted address lines.

Procedure: The following test algorithm is performed:

Let Au be the memory address u.
0<=u<2**n where n is the bit size.
E - indicates an element exists in the set.
Let
T1 = {Au|u=0 (modulo 3)}.
T2 = {Au|u=1 (modulo 3)}.
T3 = {Au|u=2 (modulo 3)}.
Step 1: Write the all 0 word, W0, at all locations Aj E T1 and Ak E T2.
Step 2: Write the all 1 word, W1, at all locations Ai E T0.
Step 3: Read all locations Aj E T1
if output = W0 no fault indicated.
Step 4: Write the all 1 word, W1, at all locations Aj E T1.
Step 5: Read all locations Ak E T2
if output = W0 no fault indicated.
Step 6: Read all locations Ai E T0 and Aj E T1
if output = W1 no fault indicated.
Step 7: Write and then read the all 0 word at all locations Ai E T0
if output = W0 no fault indicated.
Step 8: Write and then read the all 1 word, W1, at all locations Ak E T2
if output = W1 no fault indicated.

Hardware Tested: The Dynamic Memory is tested.

Data Returned: The failing address, the expected data, and the actual data are returned.

Notes: This test checks for parity errors. If more than 10 errors are encountered, the test aborts.

Register Assignments

R0 = Scratch register
R1 = current test value
R2 = test value of zero
R3 = test value of all ones
R4 = first address under test
R5 = last address under test
R6 = current test address
R7 = error counter

Test Number:	2
Function:	This test checks all memory accesses on a word basis with alternate data patterns throughout the memory range.
Procedure:	A pattern of zeros and ones is written and read. Then, a pattern of ones and zeros is written and read. This procedure is repeated for each address under test.
Hardware Tested:	The Dynamic Memory is tested.
Data Returned:	The failing address, the expected data, and the actual data are returned.
Notes:	None

==========

Test Number:	3
Function:	This test verifies that the dynamic memory is being refreshed by storing a data pattern in memory, waiting approximately 23 seconds, and confirming data retention.
Procedure:	All of memory is cleared. After a delay of approximately 23 seconds, each memory location is read and verified. This procedure is repeated for an all ones data pattern.
Hardware Tested:	The Dynamic Memory is tested.
Data Returned:	The failing address, the expected data, and the actual data read is returned.
Notes:	None

==========

Test Number:	4
Function:	This test checks variable size accesses.
Procedure:	The test procedure is as follows: 1. Words are written and read back as shorts. The address is incremented by 200. 2. Two shorts are written and read back as bytes. The address is incremented by 400. 3. Four bytes are written and read. The address is incremented by 800.
Hardware Tested:	The Dynamic Memory is tested.
Data Returned:	The failing address, the expected data, and the actual data are returned.
Notes:	None

Phase #(9)12 — Nonvolatile RAM

Phase Name:	Nonvolatile RAM (nvram_tst)
Type:	Interactive
Function:	This phase tests the Nonvolatile Random Access Memory (NVRAM) area.
Test:	Test 1 — performs an access test and pattern check on all NVRAM locations.
Time:	1 second
Warnings:	This phase should be run with caution because the contents of NVRAM are not recoverable if lost. This phase saves and restores the contents of the NVRAM, but an unexpected system problem could destroy the saved contents before they are completely restored. Default values are used to restore NVRAM if the NVRAM contents are not sane.
Notes:	None

Phase #(9)12 Test

Test Number:	1
Function:	This test performs an access test and pattern check on the NVRAM area.
Procedure:	All 1024 nibble locations of NVRAM are tested with the test patterns 0xf, 0xa, 0x5, and 0x0.
Hardware Tested:	The NVRAM is tested.
Data Returned:	The address, the value read, and the expected value of the failing Random Access Memory (RAM) location are returned.
Notes:	NVRAM locations are located in the least significant nibble of the least significant byte located within a 32-bit data word (x 1024).

Phase #(10)13 — Sanity Interval Timer

Phase Name:	INTEL* 8254 Programmable Interval Timer/Counter (ccsit_tst)
Type:	Normal
Function:	This phase tests the access to and operation of the INTEL 8254 System Timer.
Tests:	Tests 1 through 6 — perform access tests on Counters 0 through 2 using the load most significant and load least significant byte commands. Tests 7 through 10 — check the ability to perform 2-byte load operations for Counters 1 and 2. Counters 0 is default tested. Test 11 — verifies that Counter 1 operates properly.
Time:	3 seconds
Warnings:	A failure in this phase may affect other tests which assume the timers to be inactive.
Notes:	Both Counter 0, which is used to alert the system that power has been removed, and Counter 2, which is used for I/O time-out, cannot be tested for operational capabilities.

Phase #(10)13 Tests

Test Numbers:	1 through 6
Function:	These tests perform an access test on Counters 0 through 2.
Procedure:	Using the load Most Significant Byte (MSB) and load Least Significant Byte (LSB) commands, the most significant byte and least significant byte of each counter are tested with the data patterns 0x00, 0x55, 0xaa, and 0xff. Tests 1 and 2 — check Counter 0 MSB and LSB, respectively. Tests 3 and 4 — check Counter 1 MSB and LSB, respectively. Tests 5 and 6 — check Counter 2 MSB and LSB, respectively.
Hardware Tested:	The INTEL 8254 Programmable Interval Timer/Counter is tested.
Data Returned:	The failing test number (last number displayed on the system console before failure) is returned.
Notes:	None

* Trademark of Intel Corporation

Test Numbers:	7 through 10
Function:	These tests check the ability to perform 2-byte load operations for Counters 1 and 2.
Procedure:	Each counter is loaded with specific data patterns using the load multiple byte commands, and then they are read for comparison. Tests 7 and 8 — check Counter 1 LSB and MSB, respectively. Tests 9 and 10 — check Counter 2 LSB and MSB, respectively.
Hardware Tested:	The INTEL 8254 Programmable Interval Timer/Counter is tested.
Data Returned:	The failing test number (last number displayed on the system the system console before failure) is returned.
Notes:	None

==========

Test Number:	11
Function:	This test verifies that Counter 1 is operational.
Procedure:	The timer is initialized; it is set running; it is stopped; its run is confirmed.
Hardware Tested:	The INTEL 8254 Programmable Interval Timer/Counter is tested.
Data Returned:	The failing test number (last number displayed on the system console before failure) is returned.
Notes:	None

Phase #(11)14 — Control and Status Register

Phase Name: Control and Status Register (cc_csrtst)

Type: Normal

Function: This phase performs access tests on the Control and Status Register (CSR).

Tests: Test 1 — confirms that the CSR can be read with half word and 2-byte read operations.

Tests 2 and 3 — confirm that CSR bits 11, 10, 8, 7, 5, and 4 can be set and cleared by software.

Time: 1 second

Warnings: None

Notes: CSR bits 15, 14, 12, 3, 2, 1, and 0 are tested within their respective diagnostic routines.

Control Status Register Bit Assignments

15							8
TT	PE	RR	AF	FL	FM	R	IT
7							0
IF	R	I8	I9	UI	DI	DMI	IOF

TT = Error Timer Timeout	IF = Inhibit Faults
PE = Memory Parity Error	R = Reserved
RR = System Reset Request	I8 = PIR Level 8 Int.
AF = Alignment Fault	I9 = PIR Level 9 Int.
FL = Failure LED	UI = UART Interrupt
FM = Floppy Motor On	DI = Disk Interrupt
R = Reserved	DMI = DMA Interrupt
IT = Inhibit Timers	IOF = I/O Board Fail

Phase #(11)14 Tests

Test Number: 1

Function: This test verifies that the CSR can be read with half word and 2-byte read operations.

Procedure: The CSR contents are read with half word load and 2-byte load operations. Both readings are confirmed to be the same.

Hardware Tested: The CSR is tested.

Data Returned: The failing test number (last number displayed on the system console before failure) is returned.

Notes: None

==========

Test Numbers: 2 and 3

Function: These tests verify that CSR bits 11, 10, 8, 7, 5, and 4 can be set and cleared by software.

Procedure: The test procedure is as follows:

1. The bits are cleared.
2. The bits are verified to be cleared.
3. The bits are set.
4. The bits are verified to be set.
5. The bits are cleared.

Hardware Tested: The CSR is tested.

Data Returned: The failing test number (last number displayed on the system console before failure) and the failing CSR contents are returned.

Notes: None

Phase #(12)15 — Dual UART

Phase Name: 2681 DUART Tests (duart_tst)

Type: Interactive

Function: The Dual Universal Asynchronous Receiver/Transmitter (DUART) is tested in the local loopback mode by transmitting and receiving characters. The appropriate Status bits are checked as well as the correct transmission and reception of the data. The Command and Mode registers are checked for read and write capability.

Tests: The following table lists the tests that are run in this phase:

SIDE	TEST NUMBER	TEST FUNCTION
A, B	1-16, 17-32	These tests walk a one through zeros in Mode register 2 , and they walk a zero through ones in Mode register 1.
A, B	33-36, 37-40	These tests check the Receive Data bit in the Status register so that it may be used as a flag in subsequent tests. This is done by Tests 33 through 37 and 35 through 39. At the same time, the ability to transmit and receive a character is checked by Tests 34 through 38 and 36 through 40.
A, B	41-43, 44-46	These tests walk a one through zeros with 7, 6, and 5 Data bits (8 bits already used).
A, B	47-48, 49-50	These tests check even parity and no parity (odd parity already tested).
A, B	51-52, 53-54	These tests transmit characters with both 1 and 1.5 Stop bits (2 already tested).
A, B	55-56, 57-58	These tests check the Transmitter Empty and Transmitter Ready bits in the Status register for correct operation.
A	59-60, 61-63	These tests check the Received Break and Parity Error Send/Receive bits. They also check the Overrun and First-In-First-Out (FIFO) Full Send/Receive bits.
B	64-65, 66-68	These tests check the Received Break and Parity Error Send/Receive bits. They also test the Overrun and FIFO Full Send/Receive bits.

Time: 3 seconds

Warnings: None

Notes: None

Phase #(12)15 Tests

Test Numbers: 1 through 16

Function: These tests write eight different patterns to Mode register A.

Procedure: A one is walked through zeros in Mode register 2, and a zero is walked through ones in Mode register 1.

Hardware Tested: The Mode registers are tested.

Data Returned: The failing register value is returned.

Notes: None

Test Numbers:	17 through 32
Function:	These tests write eight different patterns to Mode register B.
Procedure:	A one is walked through zeros in Mode register 2, and a zero is walked through ones in Mode register 1.
Hardware Tested:	The Mode registers are tested.
Data Returned:	The failing register value is returned.
Notes:	None

==========

Test Numbers:	33 through 36
Function:	These tests check the ability of the Universal Asynchronous Receiver/Transmitter (UART) A to transmit and receive a character correctly and to set or reset the Receive Status bit in the Status register of the UART.
Procedure:	The test procedure is as follows: 1. A character is transmitted. 2. The Receive Status bit is looped on until a time-out occurs or it is set. 3. The validity of the received character is checked if time-out does not occur. 4. The receive flag is checked to be cleared upon reading the data.
Hardware Tested:	The UART is tested.
Data Returned:	Pattern 0xk00000sr is returned, where sr is the failing Character/Status register value. The variable k is 1 if RCVRDY experiences a time-out error.
Notes:	None

Test Numbers: 37 through 40

Function: These tests check the ability of the UART B to transmit and receive a character correctly and to set or reset the Receive Status bit in the Status register of the UART.

Procedure: The test procedure is as follows:

1. A character is transmitted.
2. The Receive Status bit is looped on until a time-out occurs or it is set.
3. The validity of the received character is checked if a time-out does not occur.
4. The receive flag is checked to be cleared upon reading the data.

Hardware Tested: The UART is tested.

Data Returned: Pattern 0xk00000sr is returned, where sr is the failing Character/Status register value. The variable k is 1 if RCVRDY experiences a time-out error.

Notes: None

==========

Test Numbers: 41 through 43

Function: These tests check transmission at different data lengths on the A side.

Procedure: A one is walked through zeros at 7, 6, and 5 Data bit locations (8 was already tested).

Hardware Tested: The UART and associated Control registers are tested.

Data Returned: Raw Data: 1 = XMTRDY Status register time-out error is returned.

2 = RCVRDY Status register time-out error is returned.

3 = Improper data detected is returned.

Notes: None

==========

Test Numbers: 44 through 46

Function: These tests check transmission at different data lengths on the B side.

Procedure: A one is walked through zeros at 7, 6, and 5 Data bit locations (8 was already tested).

Hardware Tested: The UART and associated Control registers are tested.

Data Returned: Raw Data: 1 = XMTRDY Status register time-out error is returned.

2 = RCVRDY Status register time-out error is returned.

3 = Improper data detected is returned.

Notes: None

Test Numbers: 47 and 48

Function: These tests check transmission with different parities on the A side.

Procedure: A one is walked through zeros with both even parity and no parity (odd parity already tested).

Hardware Tested: The UART and associated Control registers are tested.

Data Returned: Raw Data: 1 = XMTRDY Status register time-out error is returned.

2 = RCVRDY Status register time-out error is returned.

3 = Improper data detected is returned.

Notes: None

==========

Test Numbers: 49 and 50

Function: These tests check transmission with different parities on the B side.

Procedure: A one is walked through zeros at even and no parity (odd already tested).

Hardware Tested: The UART and associated Control registers are tested.

Data Returned: Raw Data: 1 = XMTRDY Status register time-out error is returned.

2 = RCVRDY Status register time-out error is returned.

3 = Improper data detected is returned.

Notes: None

==========

Test Numbers: 51 and 52

Function: These tests check transmission at 1 and 1.5 Stop bits (2 already tested) on the A side.

Procedure: A one is walked through zeros at 1 and 1.5 stop bits.

Hardware Tested: The UART and associated Control registers are tested.

Data Returned: Raw Data: 1 = XMTRDY Status register time-out error is returned.

2 = RCVRDY Status register time-out error is returned.

3 = Improper data detected is returned.

Notes: None

Test Numbers:	53 and 54
Function:	These tests check transmission at 1 and 1.5 Stop bits on the B side (2 already tested).
Procedure:	Walk a one through zeros at 1 and 1.5 Stop bits.
Hardware Tested:	The UART and associated Control registers are tested.
Data Returned:	Raw Data: 1 = XMTRDY Status register time-out error is returned. 2 = RCVRDY Status register time-out error is returned. 3 = Improper data detected is returned.
Notes:	None

==========

Test Numbers:	55 and 56
Function:	These tests check bits, other than Error bits, of the Status register on the A side.
Procedure:	Test 55 — transmits two characters and checks txemt and txrdy for zero. Test 56 — detects the character at the receiver and checks that txemt and txrdy have been set to one.
Hardware Tested:	The nonerror bits of the UART Status register are tested.
Data Returned:	Pattern 0xk00000sr is returned, where sr equals the failing Status register value. The variable k is 1 if RCVRDY experiences a time-out error or 2 if XMTRDY experiences a time-out error.
Notes:	None

==========

Test Numbers:	57 and 58
Function:	These tests check bits, other than Error bits, of the Status register on the B side.
Procedure:	Test 57 — transmits two characters and checks txemt and txrdy for zero. Test 58 — detects the character at the receiver and checks that txemt and txrdy have been set to one.
Hardware Tested:	The nonerror bits of the UART Status register are tested.
Data Returned:	Pattern 0xk00000sr is returned, where sr equals the failing Status register value. The variable k equals 1 if RCVRDY experiences a time-out error or 2 if XMTRDY experiences a time-out error.
Notes:	None

Test Numbers: 59 and 60

Function: These tests check the Received Break and Parity Error Send/Receive bits on the A side.

Procedure: The test procedure is as follows:

1. A break character is transmitted.
2. Received Break and Parity Error bits are confirmed to be set.
3. Error bits are confirmed to be cleared after read.

Hardware Tested: The Error bits of the UART Status register are tested.

Data Returned: The failing Status register value is returned.

Notes: None

==========

Test Numbers: 61 through 63

Function: These tests check the Overrun and First-In-First-Out (FIFO) Full Send/Receive bits.

Procedure: The test procedure is as follows:

1. Three characters are transmitted.
2. The FIFO Full bit is checked to be set.
3. Two more additional characters are transmitted.
4. The Overrun and FIFO Full bits are checked to be set.
5. The overrun error is confirmed to be cleared with the RESET_ERR command to the Command register.

Hardware Tested: The Error bits of the UART Status register are tested.

Data Returned: Pattern 0xk00000sr is returned, where sr equals the failing Status register value. The variable k is 2 if XMTRDY experiences a time-out error or 3 if XMTEMT experiences a time-out error.

Notes: None

==========

Test Numbers: 64 and 65

Function: These tests check the Received Break and Parity Error Send/Receive bits on the B side.

Procedure: The test procedure is as following:

1. A break character is transmitted.
2. Received Break and Parity Error bits are confirmed to be set.
3. Error bits are confirmed to be cleared after a read.

Hardware Tested: The Error bits of the UART Status register are tested.

Data Returned: The failing Status register value is returned.

Notes: None

Test Numbers:	66 through 68
Function:	The tests check the Overrun and FIFO Full Send/Receive bits on the B side.
Procedure:	The test procedure is as follows: 1. Three characters are transmitted. 2. The FIFO Full bit is checked to be set. 3. Two more additional characters are transmitted. 4. The Overrun and FIFO Full bits are checked to be set. 5. The Overrun Error bit is checked to be cleared with the RESET_ERR command to the Command register.
Hardware Tested:	The Error bits of the UART Status register are tested.
Data Returned:	Pattern 0xk00000sr is returned, where sr equals the failing Status register value. The variable k is 2 if XMTRDY experiences a time-out error or 3 if XMTEMT experiences a time-out error.
Notes:	None

Phase #(13)16 — Permanent Interrupt

Phase Name:	Pending or Stuck Interrupts Phase (pint)
Type:	Demand
Function:	This phase tests for permanent interrupts.
Tests:	Tests 1 through 15 — check Permanent Interrupt Levels (PILs).
Time:	1 second
Warnings:	A failure of this phase may affect other tests which assume that the system is clear of interrupts.
Notes:	None

Phase #(13)16 Tests

Test Numbers:	1 through 15
Function:	These tests check all interrupt levels for pending interrupts.
Procedure:	The WE 32002 Microprocessor PIL is reduced one step at a time from Level 15 to detect any pending interrupts.
Hardware Tested:	Circuits connected to the Interrupt System are tested.
Data Returned:	The first level of the interrupt found to be stuck is returned.
Notes:	After the source of the stuck interrupt is found, this test should be executed again to complete the testing of the entire interrupt system.

Phase #(14)17 — CPU Interrupt System

Phase Name: 3B2 Computer Interrupt System Test (int_tst)

Type: Normal

Function: This phase tests the Interrupt System using trackable, noninterfering interrupt sources.

Tests: Test 1 — checks Level 15 interrupt (Periodic timer time-out).

Test 2 — checks Level 13 interrupt (UART forced interrupt).

Test 3 — checks Level 11 interrupt (floppy disk forced interrupt).

Test 4 — checks Level 9 interrupt (CSR PIR 9).

Test 5 — checks Level 8 interrupt (CSR PIR 8).

Time: 2 seconds

Warnings: A failure of this phase may affect other tests which assume that the system is clear of interrupts or that timers are turned off.

Notes: The test interrupt handler masks all interrupts in the Process Status Word (PSW) of the main program.

System Board Interrupt/CSR Bit Assignment

Level	Vector	Source	CSR Bit
15	15	System Error* and Periodic Timer	15
15	15	Parity Error	14
14	*	I/O Boards PINT20	—
13	13	UARTs and DMA Complete	03/01
12	*	I/O Boards PINT10	—
11	11	Integral Disk	02
10	*	I/O Boards PINT00	—
9	9	PIR-9 (from CSR)	04
8	8	PIR-8 (from CSR)	05

* 8-bit vector supplied by I/O board.
** System Error = Bus time-out, parity error, or I/O board failure.

Phase #(14)17 Tests

Test Number: 1

Function: This test checks that a Level 15 interrupt is generated when the Periodic timer is allowed to time-out.

Procedure: The Periodic timer is loaded with an extremely low value. The timer is allowed to time-out. It is then confirmed that both a Level 15 interrupt was taken and that the Control and Status Register (CSR) accurately reflects a timer time-out condition.

Hardware Tested: The Interrupt System, Periodic timer, and CSR are tested.

Data Returned: The contents of the CSR read from the interrupt handler are returned.

Notes: The CSR condition is tested in the interrupt handler.

==========

Test Number: 2

Function: This test checks that a Level 13 interrupt is generated when the B side of the Dual Universal Asynchronous Receiver/Transmitter (DUART) issues an interrupt request.

Procedure: The B-side DUART is reset. The Interrupt register is loaded with an interrupt request. It is confirmed that both a Level 13 interrupt is taken and that the CSR accurately reflects a pending Universal Asynchronous Receiver/Transmitter (UART) interrupt request status.

Hardware Tested: The Interrupt System, DUART, and CSR are tested.

Data Returned: The contents of the CSR read from the interrupt handler are returned.

Notes: The CSR condition is tested in the interrupt handler.

==========

Test Number: 3

Function: This test checks that a Level 11 interrupt is generated when the Floppy Disk Controller issues an interrupt request.

Procedure: The Floppy Disk Command register is loaded with a force interrupt command. It is confirmed that both a Level 11 interrupt is taken and that the CSR accurately reflects a pending disk interrupt request.

Hardware Tested: The Interrupt System, Floppy Disk Controller, and CSR are tested.

Data Returned: The contents of the CSR read from the interrupt handler are returned.

Notes: The CSR condition is tested in the interrupt handler.

Test Number:	4
Function:	This test checks that a Level 9 interrupt is generated when the Central Control (CC) CSR set PIR 9 bit command is issued.
Procedure:	The CSR PIR 9 bit is cleared. The interrupt mask is lowered. The CSR PIR 9 bit is set. It is confirmed that both a Level 9 interrupt is taken and that the CSR accurately reflects the PIR 9 request.
Hardware Tested:	The Interrupt System and CSR are tested.
Data Returned:	The contents of the CSR read from the interrupt handler are returned.
Notes:	The CSR condition is tested in the interrupt handler.

==========

Test Number:	5
Function:	This test checks that a Level 8 interrupt is generated when the CC CSR set PIR 8-bit command is issued.
Procedure:	The CSR PIR 8 bit is cleared. The interrupt mask is lowered. The CSR PIR 8 bit is set. It is confirmed that both a Level 8 interrupt is taken and that the CSR accurately reflects the PIR 8 request.
Hardware Tested:	The Interrupt System and CSR are tested.
Data Returned:	The contents of the CSR read from the interrupt handler are returned.
Notes:	The CSR condition is tested in the interrupt handler.

Phase #(15)18 — Direct Memory Controller

Phase Name: Direct Memory Access Controller (DMAC) (dmac_tst)

Type: Normal

Function: This phase tests the registers and operations of the AMD 9517 Direct Memory Access Controller (DMAC).

Tests: Tests 1 through 8 — perform pattern tests on the Current Address register and the Current Word register for all channels.

Test 9 — checks DMA operation using DUART in local loop around mode.

Time: 1 second

Warnings: None

Notes: None

Phase #(15)18 Tests

Test Numbers: 1 through 8

Function: These tests perform a pattern test on each of the Current Address and Current Word registers for each channel.

Procedure: The byte pointer in the DMAC is cleared, and a pattern is loaded for the Channel register under test. Then, the byte pointer in the DMAC is cleared again, and the data pattern reception for the same Channel register is read and confirmed.

Hardware Tested: The 9517 DMAC is tested.

Data Returned: The failing test number (last number displayed on the system console before failure) is returned.

Notes: Test numbers are assigned as follows:

Test No.	Current Address Register	Test No.	Current Word Register
1	Channel 0	2	Channel 0
3	Channel 1	4	Channel 1
5	Channel 2	6	Channel 2
7	Channel 3	8	Channel 3

Test Number: 9

Function: This test confirms that the DMAC can perform a Direct Memory Access (DMA) operation.

Procedure: The test procedure is as follows:

1. The B-side Dual Universal Asynchronous Receiver/Transmitter (DUART) is initialized to local loop around mode.
2. The DMAC is initialized for DMA operation through the DUART, and the four-character string "TEST" is transmitted.
3. The DMA job is requested, and it is waited on for completion.
4. The B side of the DUART is checked for the First-In-First-Out (FIFO Full) bit to be set in the DUART Status register. The reception of the character string "TEST" is confirmed.

Hardware Tested: The 9517 DMAC is tested.

Data Returned: The failing test number (last number displayed on the system console before failure) is tested.

Notes: Pattern 0xf0xx is returned, where xx equals the failing contents of the DUART Status register or 1 through 4 are returned, where 1 through 4 represents the number of failing characters in the string "TEST."

Phase #(16)19 — Floppy Disk Interface

Phase Name:	Floppy Interface Control Test (fic_tst)
Type:	Interactive
Function:	This phase ensures that the Floppy Disk Drive and the Floppy Disk Controller are functioning.
Tests:	Tests 1 through 3 — pattern test all Floppy Disk Controller registers that can be written to and read from. Tests 4 and 5 — verify that the entire disk is formatted and the track numbers can be read. Tests 6 through 8 — test the ability to write and read back a sector. Tests 9 through 11 — test the **step**, **stepin**, and **stepout** commands.
Time:	5 seconds
Warnings:	The test code assumes a formatted double-sided, dual-density diskette. The test code uses the last cylinder as diagnostic cylinder (track 70, sides 1 and 2). The test code is configured using the International Standards Organization (ISO) Standard Dual-Density Format specification (512 bytes/sector, 5 sectors/track).
Notes:	None

Phase #(16)19 Tests

Test Numbers:	1 through 3
Function:	These tests pattern test all Floppy Disk Controller registers that can be written to and read from.
Procedure:	Test 1 — pattern tests (0x00,0x55,0xaa,0xff) the Data register. Test 2 — pattern tests the Track register. Test 3 — pattern tests the Sector register.
Hardware Tested:	The WD2797 Floppy Disk Controller/Formatter is tested.
Data Returned:	The failing test number (last number displayed on the system console before failure) and the failing test pattern are returned.
Notes:	None

Test Numbers: 4 and 5

Function: These tests verify that the entire disk is formatted and that all tracks can be read.

Procedure: The Identification (ID) field of each track is read, beginning with track 0 and proceeding to the end of each side, comparing the actual track number read against the expected track number.

Hardware Tested: The WD2797 Floppy Disk Controller/Formatter is tested.

Data Returned: The failing test number (last number displayed on the system console before failure) is returned.

Notes: None

==========

Test Numbers: 6 through 8

Function: These tests check the ability to write and read back a sector.

Procedure: Side 1, track 70, sector 1 is written with an increasing data pattern. The track is read back, and the data is compared.

Hardware Tested: The WD2797 Floppy Disk Controller/Formatter is tested.

Data Returned: The failing test number (last number displayed on the system console before failure) is returned.

Notes: None

==========

Test Numbers: 9 through 11

Function: These tests check the **step**, **stepin**, and **stepout** commands.

Procedure: The test procedure is as follows:

1. The heads are moved to track 35, and the head movement is confirmed.
2. The **step** command is issued, and the head is confirmed to be located over track 36.
3. The **stepin** command is issued, and the head is confirmed to be located over track 37.
4. The **stepout** command is issued, and the head is confirmed to be located over track 36.

Hardware Tested: The WD2797 Floppy Disk Controller/Formatter is tested.

Data Returned: The failing test number (last number displayed on the system console before failure) is returned.

Notes: None

Phase #(17)20 — Fast Hard Disk

Phase Name:	Fast Hard Disk Controller Interface Test (fhic_tst).
Type:	Normal
Function:	This phase performs a fast test that ensures the basic integrity of the integral hard disk system.
Tests:	Test 1 — performs a pattern test on the Hard Disk Controller FIFO. Test 2 — verifies that the specify command sets up the Hard Disk Controller. Test 3 — verifies that the sense unit status command returns the correct status of the hard disk drive. Test 4 — performs a quick test of the read command and the seek command. The sanity track is read and compared to known data. Test 5 — tests the write data command. Data is written and then verified. Test 6 — writes and verifies the sanity data on the sanity track.
Time:	5 seconds per equipped drive
Warnings:	None
Notes:	None

Phase #(17)20 Tests

Test Number:	1
Function:	This test performs a pattern test on the command parameter First-In-First-Out (FIFO) for read and write capability.
Procedure:	Each byte of the FIFO is written with a different pattern and read for comparison.
Hardware Tested:	The uPD7261 Hard Disk Controller is tested.
Data Returned:	The failing test number is returned. Supplemental Data: The expected and failing test pattern, the controller status, and the error status information are returned.

Test Number: 2

Function: This test verifies that the disk controller can be set up in the required mode of operation by use of the specify command.

Procedure: The specify command is issued to set up the disk controller in the following mode.

Mode:	
MDU = 0	(MFM recording)
ECC = 0	(CRC bytes appended to data)
CRCS = 0	(CRC generator polynomial)
SSEC = 1	(Floppy-like interface)
STP = 0x0e	(Stepping rate = 30ms)

Data Length:	
CRC = 1	(Initial value of polynomial counter)
PAD = 1	(ID data pad = 0x0)
POL = 1	(Polling mode disabled)
DTL = 0x200	(Data length = 512 bytes)
ETN = **	(End track - 1)
ESN = 0x11	(End sector - 1)
GPL2 = 0x0d	(Gap length 2)
RWC = 0x0080	(Reduce write current cylinder no.)

** Value dependent on type of disk.

Hardware Tested: The uPD7261 Hard Disk Controller is tested.

Data Returned: The failing test number is returned.
Supplemental Data: The controller status and error status information are returned.

==========

Test Number: 3

Function: This test checks the sense unit status command.

Procedure: The drive Read/Write (R/W) heads are positioned to cylinder 0. The sense unit status command is executed, and the Unit Status byte is read to verify the following:

Drive is selected.
Seek is complete.
At track 000.
Drive is ready.
No write fault.

Hardware Tested: The uPD7261 Hard Disk Controller is tested.

Data Returned: The failing test number is returned.
Supplemental Data: The controller status and error status information are returned.

Test Number:	4
Function:	This test checks the seek and read commands.
Procedure:	A seek to the sanity track is issued. The sanity track is read using the read command. The data read is then compared to known sanity data. This verifies that the seek positioned the read/write heads at the correct cylinder. Comparing data read with known data verifies the read command.
Hardware Tested:	The uPD7261 Hard Disk Controller is tested.
Data Returned:	The failing test number is returned. Supplemental Data: The failing disk address, the controller status, and the error status information are returned.

==========

Test Number:	5
Function:	This test checks the write data command.
Procedure:	Test patterns are written on the diagnostic cylinder using the write command. The cylinder is written on sector 0 of each track. The write is verified by reading the data from the disk and comparing it with the known data.
Hardware Tested:	The uPD7261 Hard Disk Controller is tested.
Data Returned:	The failing test number is returned. Supplemental Data: The failing disk address, the controller status, and the error status information are returned.

==========

Test Number:	6
Function:	This test writes the sanity information on the sanity track and verifies it by doing a read comparison.
Procedure:	Sanity data is written on the diagnostic cylinder. The sanity data is read and compared with the known sanity data.
Hardware Tested:	The uPD7261 Hard Disk Controller is tested.
Data Returned:	The failing test number is returned. Supplemental Data: The controller status and error status information are returned.

Phase #(18)21 — Hard Disk Interface

Phase Name: Hard Disk Controller Interface (hic_tst)

Type: Demand

Function: This phase ensures that the hard disk and the Hard Disk Controller are functioning.

Tests: Test 1. — pattern tests the Hard Disk Controller FIFO.

Test 2 — verifies that the specify command sets up the Hard Disk Controller.

Test 3 — verifies that the sense unit status command returns the correct status of the hard disk drive.

Tests 4 and 5 — verify that the seek command moves the Read/Write (R/W) heads to the specified cylinder in both the normal and buffered modes.

Tests 6 and 7 — verify that the recalibrate command positions the R/W heads at cylinder 0 from three different start positions in both the normal and buffered modes.

Test 8 — verifies that the verify ID command correctly compares IDs from the diagnostic cylinder with a generated set.

Test 9 — verifies that the read ID command correctly reads IDs from the diagnostic cylinder. The IDs read are compared to a generated set.

Test 10 — verifies that the write data command correctly writes pattern data on the diagnostic cylinder in the multisector mode.

Test 11 — verifies that the check command executes properly by checking that the data written on the diagnostic cylinder contains the correct CRC bytes.

Test 12 — verifies that the verify data command correctly compares the data on the diagnostic cylinder with equivalent generated data.

Test 13 — verifies that the read data command correctly reads the data written on the diagnostic cylinder. The data read is compared with generated data.

Test 14 — verifies that the scan command can search and find data on the diagnostic cylinder comparable to generated data.

Test 15 — verifies that the sense interrupt status command clears the service request generated by a seek or recalibrate command. Also it verifies that the sense interrupt status command fails when no service request is pending.

Test 16 — verifies that the detect error command correctly generates an error address and error patterns to correct data errors within a sector.

Test 17 — verifies that the Hard Disk Controller can detect CRC errors.

Time: 45 seconds per equipped drive

Warnings: This diagnostic code assumes the disk is formatted and that known defects have been mapped on cylinder 0. Defects on the diagnostic cylinder which have not been mapped may cause the diagnostic to fail. The last cylinder is the diagnostic cylinder. It is the only cylinder that is written by this diagnostic unless a defective sector(s) is mapped to cylinder 0.

Phase #(18)21 Tests

Test Number: 1

Function: This test pattern tests the command parameter First-In-First-Out (FIFO) for write/read capability.

Procedure: Each byte of the FIFO is written and read with a different pattern.

Hardware Tested: The uPD7261 Hard Disk Controller is tested.

Data Returned: The failing test number is returned.
Supplemental Data: The expected and failing test patterns are returned.

==========

Test Number: 2

Function: This test verifies that the disk controller can be set up in the required modes of operation by use of the specify command.

Procedure: The specify command is issued to set up the disk controller in the following modes:

Mode:	
MDU = 0	(MFM recording)
ECC = 0	(CRC bytes appended to data)
CRCS = 0	(CRC generator polynomial)
SSEC = 1	(Floppy-like interface)
STP = 0x0e	(Stepping rate = 30ms)

Data Length:	
CRC = 1	(Initial value of polynomial counter)
PAD = 1	(ID data pad = 0x0)
POL = 1	(Polling mode disabled)
DTL = 0x200	(Data length = 512 bytes)
ETN = *	(End track - 1)
ESN = 0x11	(End sector - 1)
GPL2 = 0x0d	(Gap length 2)
RWC = 0x0080	(Reduce write current cylinder no.)
* Value dependent on type of disk.	

Hardware Tested: The uPD7261 Hard Disk Controller is tested.

Data Returned: The failing test number is returned.
Supplemental Data: The error status is returned.

Test Number: 3

Function: This test checks the sense unit status command.

Procedure: The drive R/W heads are positioned to cylinder 0. The sense unit status command is executed, and the Unit Status Byte is read to verify the following:

Drive is selected.
Seek is complete.
At track 000.
Drive is ready.
No write fault.

Hardware Tested: The uPD7261 Hard Disk Controller is tested.

Data Returned: The failing test number is returned.
Supplemental Data: The controller status and the disk address are returned.

==========

Test Numbers: 4 and 5

Function: These tests verify that the seek command moves the R/W heads of the specified drive to the correct cylinder. They also verify both the normal and buffered modes with polling enabled and disabled.

Procedure: Test 4 — seeks to three positions in the normal mode.

Test 5 — seeks to three positions in the buffered mode.

Hardware Tested: The uPD7261 Hard Disk Controller is tested.

Data Returned: The failing test number is returned.
Supplemental Data: The failing disk address and the error status information are returned.

==========

Test Numbers: 6 and 7

Function: These tests check the ability of the recalibrate command to position the R/W heads at cylinder 0 from three different positions in both the normal and buffered modes.

Procedure: Test 6 — recalibrates from three different cylinders in the normal mode.

Test 7 — recalibrates from three different cylinders in the buffered mode.

Hardware Tested: The uPD7261 Hard Disk Controller is tested.

Data Returned: The failing test number is returned.
Supplemental Data: The failing disk address and the error status information are returned.

Test Number:	8
Function:	This test checks the verify Identification (ID) command.
Procedure:	After the ID data is generated, the verify ID command is executed to determine that the ID data was correctly written.
Hardware Tested:	The uPD7261 Hard Disk Controller is tested.
Data Returned:	The failing test number is returned. Supplemental Data: The failing disk address, the controller status, and the error status information are returned.

==========

Test Number:	9
Function:	This test checks the read ID command.
Procedure:	The ID is read and saved in a buffer with the read ID command. The original ID is then compared with the generated ID data.
Hardware Tested:	The uPD7261 Hard Disk Controller is tested.
Data Returned:	The failing test number is returned. Supplemental Data: The failing disk address and the error status information are returned.

==========

Test Number:	10
Function:	This test checks the write data command.
Procedure:	Test patterns are written on the diagnostic cylinder using the write data command. The cylinder is written using the multisector feature.
Hardware Tested:	The uPD7261 Hard Disk Controller is tested.
Data Returned:	The failing test number is returned. Supplemental Data: The failing disk address and the error status information are returned as supplemental data.

==========

Test Number:	11
Function:	This test verifies the check command.
Procedure:	The data written in Test 10 is verified using the check command to ensure that the Cyclic Redundancy Check (CRC) bytes are correct.
Hardware Tested:	The uPD7261 Hard Disk Controller is tested.
Data Returned:	The failing test number is returned. Supplemental Data: The failing disk address and the error status information are returned.

Test Number:	12
Function:	This test checks the verify data command.
Procedure:	The verify data command compares the data on the diagnostic cylinder with the test data in disk buffer 1.
Hardware Tested:	The uPD7261 Hard Disk Controller is tested.
Data Returned:	The failing test number is returned. Supplemental Data: The failing disk address and the error status information are returned.

==========

Test Number:	13
Function:	This test checks the read data command.
Procedure:	The data written is read using the read data command. The data read is then compared to the original data.
Hardware Tested:	The uPD7261 Hard Disk Controller is tested.
Data Returned:	The failing test number is returned. Supplemental Data: The failing disk address and the error status information are returned.

==========

Test Number:	14
Function:	This test checks the scan command.
Procedure:	The scan command is executed to search for the data recorded on the diagnostic cylinder in sector 0. The data pattern should be the same as the data that was written by the Write Data test.
Hardware Tested:	The uPD7261 Hard Disk Controller is tested.
Data Returned:	The failing test number is returned. Supplemental Data: The failing disk address, the controller status, and the error status information are returned.

Test Number: 15

Function: This test checks the sense interrupt status command.

Procedure: The polling flag is set and a seek command is issued. The Command End High bit should be set to indicate a normal termination. It waits for the Sense Interrupt Status Request (SRQ) bit in the Status register to indicate a request for the sense interrupt status command. The sense interrupt status command is then issued, and the interrupt status byte is read to verify a normal seek end and no seek error. An abnormal termination of the command is tested by issuing the sense interrupt status command without a preceding seek command.

Hardware Tested: The uPD7261 Hard Disk Controller is tested.

Data Returned: The failing test number is returned.
Supplemental Data: The failing disk address (if applicable) and the error status information are returned.

==========

Test Number: 16

Function: This test checks the detect error command.

Procedure: A sector on the diagnostic cylinder is read into buffer 1 to ensure that it contains no Error Correction Code (ECC) errors. The sector data size is then changed from 512 bytes to 516 bytes and read into buffer 2. A single-bit error is introduced into the data and saved in buffer 2 and it is written back onto the disk. The sector data size is changed back to 512 bytes and read into buffer 2. The detect error command is then executed to obtain the error address and error correction patterns. This information is then used to correct the erroneous data in buffer 2, after which buffer 2 is compared to buffer 1 to determine that the data was corrected.

Hardware Tested: The uPD7261 Hard Disk Controller is tested.

Data Returned: The failing test number is returned.
Supplemental Data: The failing disk address and the error status information are returned.

==========

Test Number: 17

Function: This test checks for CRC errors.

Procedure: Data is read from a disk sector. Its status is checked to verify that no CRC error occurred. The sector data length is changed to 256 bytes, and the sector is read again. This should cause a CRC error to be detected. The sector length is restored.

Hardware Tested: The uPD7261 Hard Disk Controller is tested.

Data Returned: The failing test number is returned.
Supplemental Data: The failing disk address and the error status information are returned.

Phase #(19)22 — Time-of-Day Clock

Phase Name: Time-of-Day Clock Phase (tod_tst).

Type: Interactive

Function: This phase provides a fast test of the real-time clock.

Tests: Test 1 — checks the start and stop functions and the reset function of the Seconds registers.

Test 2 — executes zero test of registers for seconds, minutes, hours, days, day of week, and month.

Test 3 — does a write test of registers for seconds, minutes, hours, days, day of week, and month.

Test 4 — performs a count test of registers for tens, seconds, and tenths-of-seconds.

Phase #(19)22 Tests

Test Number: 1

Function: This test checks the start and stop functions and the reset function of the Seconds registers.

Procedure: The clock is stopped, and the 3 Seconds registers (tens, seconds, and tenths-of-seconds) are verified to be reset to zero.

Hardware Tested: The MM5817A Real-Time Clock is tested.

Data Returned: The failing test number is returned.

Notes: None

==========

Test Number: 2

Function: This test verifies that all writable registers can be zeroed.

Procedure: All registers are written, and it is verified that they all contain zero.

Hardware Tested: The MM5817A Real-Time Clock is tested.

Data Returned: The failing test number is returned.

Notes: None

Test Number:	3
Function:	This test checks all writable registers.
Procedure:	All registers are written and verified to contain the written value.
Hardware Tested:	The MM5817A Real-Time Clock is tested.
Data Returned:	The failing test number is returned. Supplemental Data: The expected data and the received data are returned.
Notes:	None

==========

Test Number:	4
Function:	This test checks the Tens, Seconds, and Tenths-of-Seconds counters.
Procedure:	The clock is started, and the tens, seconds, and tenths-of-seconds counters are read to determine that they increment properly.
Hardware Tested:	The MM5817A Real-Time Clock is tested.
Data Returned:	The failing test number is returned. Supplemental Data: The expected data and the received data are returned.
Notes:	None

Phase #(20)23 — Hard Disk Media Check

Phase Name: Hard Disk Media Phase (med_tst)

Type: Demand

Function: This phase verifies that the Cyclic Redundancy Check (CRC) codes for both data and addresses are correct over the entire disk media. This phase also verifies that the address returned by the Hard Disk Controller (HDC) is the next expected address.

Test: Test 1 — uses the check command to verify that the CRC codes for both the address and data are correct. For all CRC errors that are detected, the failing address is printed in absolute form (that is, cylinder, head, and sector).

The address returned by the HDC is compared with an expected value to ensure that the HDC knows where it is on the disk. An error message is displayed if a mismatch occurs.

Any error detected during a seek, recalibrate, or when the operational mode of the controller is specified will result in the termination of the test. If a failure occurs, the Hard Disk Interface (Phase 18) diagnostic should be run to determine the problem.

Warnings: This test code assumes that the demand phase of the Hard Disk Interface diagnostic runs ATP (All Tests Passed). It also assumes the disk is formatted and that known defects have been mapped to alternate sectors.

Phase #(20)23 Test

Test Number: 1

Function: This test checks the entire surface of the disk with the check command.

Procedure: The CRC for both the address and data is verified by the check command. Defect management is used to avoid known media defects. In addition to the CRC, the address returned by the HDC is checked to verify that the controller knows the portion of the disk it is checking.

Hardware Tested: The surface of the hard disk is tested.

Data Returned: The failing disk address of the error and any disk address mismatches are returned.
Supplemental Data: The controller status for fatal controller errors is returned.

Version 3 Hardware System Board Diagnostics

This section contains a description of the diagnostic phases and tests for the 3B2/500, 522, 600, 622, 700, and 1000 computer System Boards (SBDs). These phases and tests are applicable to UNIX System V Release 3.2.1 and subsequent releases. Note that unlike Version 2 SBD diagnostics, the MAU diagnostic tests are not included as part of the SBD diagnostic tests. The MAU diagnostics tests are located in Chapter 12.

The SBD is a highly integrated circuit board that is the primary element of the 3B2 computer. The following components are found on the SBD:

- MAU
- MMU
- Internal Cache (if present)
- WE 32100 or WE 32200 Multiprocessor
- Dynamic Random Access Memory (DRAM) Controller
- Interrupt Structure
- Timers (includes Time-of-Day clock, Periodic timer, and Bus timer)
- Control and Status Register (CSR)
- Erasable Programmable Read Only Memory (EPROM)
- Nonvolatile Random Access Memory (NVRAM).

The following 22 phases check all aspects of the Version 3 computer SBDs except the MAU which has a separate set of diagnostic phases. The MAU phases are found in Chapter 12. You can refer to the Table of Contents in Chapter 12 for a summary of the SBD phases performed. The phases are organized in the same order that they are run on the 3B2 computer.

Phase #1 — CPU Test #2 Diagnostics

Phase Name:	WE 32100 Processor Module (cpu32_2)
Type:	Normal
Function:	This phase tests the operation and instruction set of the processor module.
Tests:	Test 1 — checks the logical dyadic instructions. Test 2 — checks the logical triadic instructions. Test 3 — checks the addition functions. Test 4 — checks the subtraction functions. Test 5 — checks the multiplication functions. Test 6 — checks the division functions. Test 7 — checks the modulo arithmetic functions.
Time:	1 second
Warnings:	None
Notes:	In general, most test pass/fail conditions are checked by collecting a hash value which is the composite value obtained by the summing of all test results.

Phase #1 Tests

Test Number:	1
Function:	This test checks the logical dyadic instructions.
Procedure:	The logical instructions OR, AND, and EXCLUSIVE OR are exercised.
Hardware Tested:	The WE 32100 Microprocessor is tested.
Data Returned:	The failing test number (last number displayed on the system console before failure) is returned.
Notes:	The final pass/fail of this test is determined by a hash value that is collected by the correct execution of the instructions.

Test Number:	2
Function:	This test checks the logical triadic instructions.
Procedure:	The logical instructions OR, AND, and EXCLUSIVE OR are exercised.
Hardware Tested:	The WE 32100 Microprocessor is tested.
Data Returned:	The failing test number (last number displayed on the system console before failure) is returned.
Notes:	The final pass/fail of this test is determined by a hash value that is collected by the correct execution of the instructions.

==========

Test Number:	3
Function:	This test checks the ability of the CPU to perform addition.
Procedure:	The results of the calculation form a hash sum which is checked for validity.
Hardware Tested:	The WE 32100 Microprocessor is checked.
Data Returned:	The failing test number (last number displayed on the system console before failure) is returned.
Notes:	None

==========

Test Number:	4
Function:	This test checks the ability of the CPU to perform subtraction.
Procedure:	The results of the calculation form a hash sum which is checked for validity.
Hardware Tested:	The WE 32100 Microprocessor is checked.
Data Returned:	The failing test number (last number displayed on the system console before failure) is returned.
Notes:	None

==========

Test Number:	5
Function:	This test checks the ability of the CPU to perform multiplication.
Procedure:	The results of the calculation form a hash sum which is checked for validity.
Hardware Tested:	The WE 32100 Microprocessor is tested.
Data Returned:	The failing test number (last number displayed on the system console before failure) is returned.
Notes:	None

Test Number:	6
Function:	This test checks the ability of the CPU to perform division.
Procedure:	The results of the calculation form a hash sum which is checked for validity.
Hardware Tested:	The WE 32100 Microprocessor is checked.
Data Returned:	The failing test number (last number displayed on the system console before failure) is returned.
Notes:	None

==========

Test Number:	7
Function:	This test checks the ability of the CPU to perform modulo arithmetic.
Procedure:	The results of the calculation form a hash sum which is checked for validity.
Hardware Tested:	The WE 32100 Microprocessor is tested.
Data Returned:	The failing test number (last number displayed on the system console before failure) is returned.
Notes:	None

Phase #2 — Central Processor Unit Test #3 Diagnostics

Phase Name:	WE 32100 Processor Module (cpu32_3)
Type:	Normal
Function:	This phase tests the operation and instruction set of the processor module.
Tests:	Test 1 — checks the extract field instruction. Test 2 — checks the insert field instruction. Test 3 — checks the move block instruction. Test 4 — checks the test swap Interlock instruction. Test 5 — checks the bit test instruction. Test 6 — checks the Save and Restore register instructions. Test 7 — checks the call procedure and RET instructions. Test 8 — checks the test sign extension types. Tests 9 through 16 — checks CPU addressing modes.
Time:	1 second
Warnings:	None
Notes:	None

Phase #2 Tests

Test Number:	1
Function:	This test checks the extract field instruction.
Procedure:	The extract field instruction is performed for all widths and offsets.
Hardware Tested:	The WE 32100 Microprocessor is tested.
Data Returned:	The failing test number (last number displayed on the system console before failure) is returned.
Notes:	The results of the test are compared to a computed hash sum that contains all of the intermediate results. Register r7 contains the hash value that is generated by adds and rotates.

Test Number: 2

Function: This test checks the insert field instruction.

Procedure: The insert field instruction is performed for all widths and offsets.

Hardware Tested: The WE 32100 Microprocessor is tested.

Data Returned: The failing test number (last number displayed on the system console before failure) is returned.

Notes: The results of the test are compared to a computed hash sum that contains all of the intermediate results. Register r7 contains the hash value that is generated by adds and rotates.

==========

Test Number: 3

Function: This test checks the move block instruction.

Procedure: A block of 16 words is moved from location 0x0 to the Random Access Memory (RAM) scratch area. After the block move, the locations are compared by physical addressing.

Hardware Tested: The WE 32100 Microprocessor is tested.

Data Returned: The failing test number (last number displayed on the system console before failure) is returned.

Notes: The following instruction sequence is used in this test:

```
while( r2 < 0)
{
      *r1 = r0;
      --r2;
      r0++;
      r1++;
}
```

==========

Test Number: 4

Function: This test checks the test swap interlocked instruction.

Procedure: Tests for word, half word, and byte swap are performed.

Hardware Tested: The WE 32100 Microprocessor is tested.

Data Returned: The failing test number (last number displayed on the system console before failure) is returned.

Notes: The global locations in this test are used only because they are available from other tests and are conveniently located in RAM.

Test Number:	5
Function:	This test checks the bit test instruction for word, half word, and byte.
Procedure:	The instruction is exercised, and it compiles a hash value based on the results in the condition code portion of the Status register.
Hardware Tested:	The WE 32100 Microprocessor is tested.
Data Returned:	The failing test number (last number displayed on the system console before failure) is returned.
Notes:	The results of the test are confirmed by comparing them to a generated hash sum that contains all of the intermediate results. Register r6 contains the hash value that is generated by adds and rotates.

==========

Test Number:	6
Function:	This test checks the Save and Restore register instructions.
Procedure:	Registers r3 through r6 are loaded with a unique pattern that will be used by the save instruction to load the stack. Subsequent restore commands verify that the stack is correctly loaded and unloaded.
Hardware Tested:	The WE 32100 Microprocessor is tested.
Data Returned:	The failing test number (last number displayed on the system console before failure) is returned.
Notes:	The pass/fail condition of this test is determined by a hash value that is generated by the addition of registers r3 through r6 after the restore instruction is executed.

==========

Test Number:	7
Function:	This test checks the call procedure and RET instructions.
Procedure:	A small subroutine call is set up that returns a value in r0 to test that the call procedure and RET instructions function properly.
Hardware Tested:	The WE 32100 Microprocessor is tested.
Data Returned:	The failing test number (last number displayed on the system console before failure) is returned.
Notes:	None

Test Number: 8

Function: This test checks the test sign extension types.

Procedure: Perform the following tests:

	Byte	Half Word	Word
Unsigned byte (-1) to	-	0xff	0xff
Unsigned half word to	-	-	0x0000ffff
Signed byte (-1) to	-	-1	-1
Signed half word to	-	-	-1

Hardware Tested: The WE 32100 Microprocessor is tested.

Data Returned: The failing test number (last number displayed on the system console before failure) is returned.

Notes: None

==========

Test Numbers: 9 through 16

Function: These tests check the literal, immediate, absolute deferred, absolute, AP short offset, FP short offset, register deferred and [HB] displacement addressing modes.

Procedure: Using the global variables, test the assorted addressing modes of the CPU.

Hardware Tested: The WE 32100 Microprocessor is tested.

Data Returned: The failing test number (last number displayed on the system console before failure) is returned.

Notes: None

Phase #3 — CPU Test #4 Diagnostics

Phase Name:	WE 32100 Processor Module (cpu32_4)
Type:	Normal
Function:	This phase tests the operation and instruction set of the processor module.
Tests:	Test 1 — checks PUSHAW, PUSHW, and POPW instructions. Test 2 — checks the unconditional jump instruction. Test 3 — checks the branch to subroutine and return from subroutine instructions. Test 4 — checks the jump to subroutine and return from subroutine instructions. Test 5 — checks the exception for the divide-by-zero instruction. Tests 6 through 8 — check for exception when the three privileged registers (ISP, PCBP, PSW) are accessed in user execution level. Tests 9 and 10 — checks for exception when the PSW is accessed at any execution level except kernel. Test 11 — checks for correct exception processing during a gate instruction. Test 12 — checks that when operating at the user execution level, the appropriate exception is generated when the privileged gate instruction is attempted.
Time:	1 second
Warnings:	None
Notes:	A failure in Tests 6 through 12 may leave the microprocessor in a nonkernel execution level. Any access at that point to privileged registers or privileged instructions could cause an unexpected exception.

Phase #3 Tests

Test Number: 1

Function: This test verifies the PUSHAW, PUSHW, and POP instructions.

Procedure: The instructions are performed.

Hardware Tested: The WE 32100 Microprocessor is tested.

Data Returned: The failing test number (last number displayed on the system console before failure) is returned.

Notes: None

==========

Test Number: 2

Function: This test checks the unconditional jump instruction.

Procedure: A jump instruction is performed. If the jump is not accomplished, the next instruction indicates an error.

Hardware Tested: The WE 32100 Microprocessor is tested.

Data Returned: The failing test number (last number displayed on the system console before failure) is returned.

Notes: None

==========

Test Number: 3

Function: This test checks the branch to subroutine and return from subroutine instructions.

Procedure: A small subroutine is set up that, after entered, returns on a conditional value of zero.

Hardware Tested: The WE 32100 Microprocessor is returned.

Data Returned: The failing test number (last number displayed on the system console before failure) is returned.

Notes: None

==========

Test Number: 4

Function: This test checks the jump to subroutine and return from subroutine instructions.

Procedure: A small subroutine is set up that, after entered, returns on a conditional value of zero.

Hardware Tested: The WE 32100 Microprocessor is returned.

Data Returned: The failing test number (last number displayed on the system console before failure) is returned.

Notes: This test is similar to Test 3 except that it exercises the jump instruction rather than the branch to subroutine instruction. It also uses a different return condition.

Test Number:	5
Function:	This test ensures that an exception occurs when the instruction calls for a divide-by-zero.
Procedure:	The test divides by zero and verifies that an exception is generated.
Hardware Tested:	The WE 32100 Microprocessor is tested.
Data Returned:	The failing test number (last number displayed on the system console before failure) is returned.
Notes:	A failure of this test indicates that a divide-by-zero instruction did not cause an exception. Also note that it is this exception that puts the test into kernel execution level.

==========

Test Numbers:	6 through 8
Function:	These tests ensure that an exception is generated when the microprocessor is at the user execution level and access is attempted to the following: Test 6 — Interrupt Stack Pointer (ISP) Test 7 — Process Control Block Pointer (PCBP) Test 8 — Process Status Word (PSW)
Procedure:	The test puts the microprocessor into user execution level and attempts to access each of the above registers. Confirm that an exception was generated by the exception flag.
Hardware Tested:	The WE 32100 Microprocessor is tested.
Data Returned:	The failing test number (last number displayed on the system console before failure) is returned.
Notes:	None

==========

Test Numbers:	9 and 10
Function:	These tests confirm that an exception is generated when access to the PSW register is attempted at any level other than kernel.
Procedure:	The CPU is put into the (9) executive or (10) supervisor execution level and attempts to access the PSW register.
Hardware Tested:	The WE 32100 Microprocessor is tested.
Data Returned:	The failing test number (last number displayed on the system console before failure) is returned.
Notes:	None

Test Number:	11
Function:	This test confirms that an exception is generated when the gate instruction is executed.
Procedure:	Perform the gate instruction and confirm an exception was generated via the exc_fig variable.
Hardware Tested:	The WE 32100 Microprocessor is tested.
Data Returned:	The failing test number (last number displayed on the system console before failure) is returned.
Notes:	None

==========

Test Number:	12
Function:	This test confirms that an exception is generated operating at the user execution level and a privileged gate instruction is attempted.
Procedure:	Put the microprocessor in the user execution level and attempt to perform the gate instruction. The variable 'epsw' will indicate if a 'stack exception' (error) or 'normal exception' (correct) was processed.
Hardware Tested:	The WE 32100 Microprocessor is tested.
Data Returned:	The failing test number (last number displayed on the system console before failure) is returned.
Notes:	None

Phase #4 — Dynamic Random Access Memory Diagnostics

Phase Name:	Dynamic Memory (dram_tst)
Type:	Normal
Function:	This phase checks the Dynamic Random Access Memory (DRAM) by performing the tests listed below.
Tests:	Test 1 — checks all memory accesses on a word basis with alternate data patterns through memory range. A pattern of 55's followed by aa's. Test 2 — checks memory accesses by writing words and reading back two half words. Test 3 — checks memory accesses by writing two half words and reading back 4 bytes. Test 4 — checks memory accesses by writing and reading 4 bytes.
Time:	2 seconds per megabyte of DRAM
Warnings:	None
Notes:	None

Phase #4 Tests

Test Number:	1
Function:	This test checks all memory accesses on a word basis, checking with alternate data patterns throughout memory range.
Procedure:	A pattern of 55's and aa's is written to and read from dynamic memory.
Hardware Tested:	The DRAM is tested.
Data Returned:	The failing address, the expected data, and the actual data are returned.
Notes:	None

==========

Test Numbers:	2 through 4
Function:	These tests check variable size accesses.
Procedure:	Test 2 — checks by writing words and reading back as shorts. Increment address by 200. Test 3 — checks by writing two shorts and reading back as bytes. Increment address by 400. Test 4 — checks by writing and reading 4 bytes. Increment address by 800.
Hardware Tested:	The DRAM is tested.
Data Returned:	The failing address, the expected data, and the actual data are returned.
Notes:	None

Phase #5 — Extended Dynamic Random Access Memory (DRAM) Diagnostics

Phase Name: Extended Dynamic Memory Test (edram_tst)

Type: Demand

Function: This phase tests the dynamic memory by performing the following tests:

Tests:

Test 1 — checks all memory accesses on a word basis, checking for shorted address lines.

Test 2 — checks all memory accesses on a word basis with alternate data patterns through memory range. A pattern of 55's followed by aa's.

Test 3 — verifies dynamic memory is being refreshed by storing data pattern in memory, delaying 1 minute and confirming data retention. Repeat test with complement of previous data test pattern.

Test 4 — checks memory accesses by writing words and reading back two half words.

Test 5 — checks memory accesses by writing two half words and reading back four bytes.

Test 6 — checks memory accesses by writing and reading 4 bytes.

Tests 7 through 16 — force ECC syndrome for all single-bit errors in bits D15 through D6 by walking a one through a field of zeros using the following binary memory half word:

```
D15       D6 D5    D0       (D5 through D0 are Check bits)
0000000000  001111
```

Tests 17 through 26 — force ECC syndrome for all single bit errors in bits D31 through D22 by walking a one through a field of zeros using the following binary memory half word:

```
D31      D22 D21  D16       (D21 through D16 are Check bits)
0000000000  001111
```

Tests 27 through 36 — force ECC syndrome for all single-bit errors in bits D15 through D6 by walking a zero through a field of ones using the following binary memory half word:

```
D15       D6 D5    D0       (D5 through D0 are Check bits)
1111111111  000000
```

Tests 37 through 46 — force ECC syndrome for all single-bit errors in bits D31 through D22 by walking a zero through a field of ones using the following binary memory half word:

```
D31      D22 D21  D16       (D21 through D16 are Check bits)
1111111111  000000
```

Tests 47 through 58 — force ECC syndrome for all single-bit errors in bits D5 through D0 and D21 through D16 by storing and reading the following binary memory half word:

	D15 D6	D5 D0	(D5 through D0 are Check bits)
D5	0000000100	000001	Correct Check Bits: 100001
D4	0000000100	110000	Correct Check Bits: 100000
D3	0000000100	100110	Correct Check Bits: 101110
D2	0000000100	101001	Correct Check Bits: 101101
D1	0000000100	101010	Correct Check Bits: 101000
D0	0000000100	100100	Correct Check Bits: 100101

	D31 D22	D21 D16	(D21 through D16 are Check bits)
D21	0000000100	000001	Correct Check Bits: 100001
D20	0000000100	110000	Correct Check Bits: 100000
D19	0000000100	100110	Correct Check Bits: 101110
D18	0000000100	101001	Correct Check Bits: 101101
D17	0000000100	101010	Correct Check Bits: 101000
D16	0000000100	100100	Correct Check Bits: 100101

Tests 59 through 60 — force ECC syndrome for double and triple bit errors by storing and reading the following binary memory half word:

	D15 D6	D5 D0	(D5 through D0 are Check bits)
59 -	1010111111	000000	Double bit error D14-D12

	D31 D22	D21 D16	(D21 through D16 are Check bits)
60 -	1010111111	000000	Double bit error D30-D28

Test 61 — determines the memory size bits of each memory board to the actual memory configuration.

Time: Approximately 5 minutes

Warnings: None

Notes: Tests 7 through 22 confirm that the Fault register also indicates the correct failing syndrome bit/address information. Also, it is not possible to cause single-bit errors within the Check bit fields C6 through C0 because of hardware limitations.

Phase #5 Tests

Test Number: 1

Function: This test checks all memory accesses on a word basis, checking for shorted address lines.

Procedure: The following test algorithm is performed:

Let Au be the memory address u.
0<=u<2**n where n is the bit size.
E - indicates an element exists in the set.
Let
T1 = {Au\u=0 (modulo 3)}.
T2 = {Au\u=1 (modulo 3)}.
T3 = {Au\u=2 (modulo 3)}.
Step 1: Write the all 0 word, W0, at all locations
Aj E T1 and Ak E T2.
Step 2: Write the all 1 word, W1, at all locations
Ai E T0.
Step 3: Read all locations Aj E T1
if output = W0 no fault indicated.
Step 4: Write the all 1 word, W1, at all locations
Aj E T1.
Step 5: Read all locations Ak E T2
if output = W0 no fault indicated.
Step 6: Read all locations Ai E T0 and Aj E T1
if output = W1 no fault indicated.
Step 7: Write and then read the all 0 word
at all locations Ai E T0
if output = W0 no fault indicated.
Step 8: Write and then read the all 1 word, W1,
at all locations Ak E T2
if output = W1 no fault indicated.

Hardware Tested: The Dynamic Memory is tested.

Data Returned: The failing address, the expected data, and the actual data are returned.

Notes: The test checks for parity errors. If more than 10 errors are encountered, test aborts.

==========

Test Number: 2

Function: This test checks all memory accesses on a word basis with alternate data patterns throughout memory range.

Procedure: A pattern of 55's and aa's is written and then read.

Hardware Tested: The Dynamic Memory is tested.

Data Returned: The failing address, the expected data, and the actual data are returned.

Notes None

Test Number: 3

Function: This test verifies that dynamic memory is being refreshed by storing a data pattern in memory, delaying 60 seconds, and confirming data retention.

Procedure: All of memory is cleared, followed by a delay of approximately 60 seconds. Then each memory location is read and verified. This is repeated for an alternate data pattern.

Hardware Tested: The Dynamic Memory is tested.

Data Returned: The failing address, the expected data, and the actual data are returned.

Notes: None

==========

Test Numbers: 4 through 6

Function: These tests check variable size accesses.

Procedure: Tests 4 through 6 execute as follows:

Test 4 — checks by writing words and reading back as shorts. Increment address by 1000.

Test 5 — checks by writing two shorts and reading back as bytes. Increment address by 1000.

Test 6 — checks by writing and reading 4 bytes. Increment address by 1000.

Hardware Tested: The Dynamic Memory is tested.

Data Returned: The failing address, the expected data, and the actual data are returned.

Notes: None

==========

Test Numbers: 7 through 16

Function: These tests check that all single-bit errors can be detected and reported correctly in the Hardware Fault registers and in the Control Status and Error Register (CSER).

Procedure: Walk a one through a field of zeros in bits D6 through D15 using the following binary memory half words:

```
D15       D6 D5    D0   (D5 through D0 are Check bits)
 0000000000  001111
```

When the error is forced, read the CSER confirming that the Single-bit Error bit was set, that an interrupt was processed, and the correct syndrome bit/address information was stored in the Hardware Fault registers.

Hardware Tested: The Dynamic Memory, CSER, and Hardware Fault registers are tested.

Data Returned: The failing address, the expected data, and the actual data are returned.

Notes: None

Test Numbers: 17 through 26

Function: These tests check that all single-bit errors can be detected and reported correctly in the Hardware Fault registers and in the CSER.

Procedure: Walk a one through a field of zeros in bits D22 through D31 using the following binary memory half word:

```
D31     D22 D21 D16  (D21 through D16 are Check bits)
0000000000  001111
```

When the error is forced, read the CSER confirming that the Single-bit Error bit was set, that an interrupt was processed, and the correct syndrome bit/address information was stored in the Hardware Fault registers.

Hardware Tested: The Dynamic Memory, CSER, and Hardware Fault registers are tested.

Data Returned: The failing address, the expected data, and the actual data are returned.

Notes: None

==========

Test Numbers: 27 through 46

Function: These tests check that all single-bit errors can be detected and reported correctly in the Hardware Fault registers and in the CSER.

Procedure: Tests 27 through 36 - walk a zero through a field of ones in bits D6 through D15 using the following binary memory half words:

```
D15       D6 D5    D0   (D5 through D0 are Check bits)
 1111111111   000000
```

Tests 37 through 46 - walk a zero through a field of ones in bits D22 through D31 using the following binary memory half word:

```
D31       D22 D21  D16   (D21 through D16 are Check bits)
 1111111111   000000
```

When the error is forced, read the CSER confirming that the Single-bit Error bit was set, that an interrupt was processed, and the correct syndrome bit/address information was stored in the Hardware Fault registers.

Hardware Tested: The Dynamic Memory, CSER, and Hardware Fault registers are tested.

Data Returned: The failing address, the expected data, and the actual data are returned.

Notes: None

Test Numbers: 47 through 58

Function: These tests check that all single-bit errors in bits D5 through D0 and D21 can be detected and reported correctly in the Hardware Fault registers and in the CSER.

Procedure: Store the following binary memory half words:

	D15	D6	D5	D0	(D5 through D0 are Check bits)
#47	D5	0000000100		000001	Correct Check Bits: 100001
#48	D4	0000000100		110000	Correct Check Bits: 100000
#49	D3	0000000100		100110	Correct Check Bits: 101110
#50	D2	0000000100		101001	Correct Check Bits: 101101
#51	D1	0000000100		101010	Correct Check Bits: 101000
#52	D0	0000000100		100100	Correct Check Bits: 100101

	D31	D22	D21	D16	(D21 through D16 are Check bits)
#53	D21	0000000100		000001	Correct Check Bits: 100001
#54	D20	0000000100		110000	Correct Check Bits: 100000
#55	D19	0000000100		100110	Correct Check Bits: 101110
#56	D18	0000000100		101001	Correct Check Bits: 101101
#57	D17	0000000100		101010	Correct Check Bits: 101000
#58	D16	0000000100		100100	Correct Check Bits: 100101

When the error is forced, read the CSER confirming that the Single-bit Error bit was set, that an interrupt was processed, and the correct syndrome bit/address information was stored in the Hardware Fault registers.

Hardware Tested: The Dynamic Memory, CSER, and Hardware Fault registers are tested.

Data Returned: The failing address, the expected data, and the actual data are returned.

Notes: None

==========

Test Numbers: 59 through 60

Function: These tests check that a double-bit error can be detected and that the correct bits are identified in the Hardware Fault registers.

Procedure: Inhibit Error Correction Coding (ECC). Fault the contents of both lower and upper half words. Determine if a multiple-bit error occurred for each half word.

Hardware Tested: The Dynamic Memory is tested.

Data Returned: The failing address, the expected data, and the actual data are returned.

Notes: None

Test Number:	61
Function:	This test determines the amount of memory specified by the Memory Size bits on each board.
Procedure:	Write every 0x100000 locations beyond the present size of memory up to the end of the memory spectrum. Any access to these locations should cause a access fault. If the write succeeded then the Memory Size bits indicate an invalid size.
Hardware Tested:	The Dynamic Memory is tested.
Data Returned:	Memory board availability and size received upon error.
Notes:	In systems with two MMUs, each test accesses both MMUs. When a test fails, the failing address identifies the failing MMU. MMU0 has addresses from 0x4f000-0x4ffff, and MMU1 has addresses from 0x4b000-0x4bfff.

Phase #6 — Memory Management Unit Test #1 Diagnostics

Phase Name:	Memory Management Unit Peripheral Mode (mmu1_tst)
Type:	Normal
Function:	This phase checks the internal registers and descriptor caches of the Memory Management Unit (MMU).
Tests:	Test 1 — checks the Virtual Address register. Tests 2 and 3 — check the SDCs. Tests 4 and 5 — check the LHPDC. Tests 6 and 7 — check the RHPDC. Test 8 — checks the SRAMA. Test 9 — checks the SRAMB. Test 10 — checks the Fault Address register. Test 11 — checks the Configuration register. Test 12 — checks the mask levels of MMUs.
Time:	1 second
Warnings:	None
Notes:	None

Phase #6 Tests

Test Number:	1
Function:	This test checks the Virtual Address register.
Procedure:	Read and write the Virtual Address register using full binary bit patterns.
Hardware Tested:	The Virtual Address register is tested.
Data Returned:	The failing test number, the failing test pattern, and the actual test pattern read are returned.
Notes:	None

Test Numbers:	2 and 3
Function:	These tests ensure that the Segment Descriptor Caches (SDCs) can be written to and read from.
Procedure:	A pattern test on each of the 32 segment descriptor locations is performed. Test 2 — checks the lower half of the SDC area. Test 3 — checks the upper half of the SDC area.
Hardware Tested:	The SDCs are tested.
Data Returned:	The failing test number, the failing test pattern, the actual test pattern read, and the failing SDC address are returned.
Notes:	None

==========

Test Numbers:	4 and 5
Function:	These tests ensure that the Left-Hand Page Descriptor Cache (LHPDC) can be written to and read from.
Procedure:	Four binary bit patterns are written and read back for comparison. Test 4 — checks the lower half of the LHPDC area. Test 5 — checks the upper half of the LHPDC area.
Hardware Tested:	The LHPDC is tested.
Data Returned:	The failing test number, the failing test pattern, the actual test pattern read, and the failing LHPDC address are returned.
Notes:	None

==========

Test Numbers:	6 and 7
Function:	These tests ensure that the Right-Hand Page Descriptor Cache (RHPDC) can be written to and read from.
Procedure:	Four binary bit patterns are written and read back for comparison. Test 6 — checks the lower RHPDC area. Test 7 — checks the upper RHPDC area.
Hardware Tested:	The RHPDC is tested.
Data Returned:	The failing test number, the failing test pattern, the actual test pattern read, and the failing RHPDC address are returned.
Notes:	None

Test Number:	8
Function:	This test ensures that the Section Random Access Memory A (SRAMA) can be written to and read from.
Procedure:	Four binary patterns are written and read back for comparison.
Hardware Tested:	The SRAMA is tested.
Data Returned:	The failing test number, the failing test pattern, the actual test pattern read, and the failing SRAMA address are returned.
Notes:	None

==========

Test Number:	9
Function:	This test ensures that the Section Random Access Memory B (SRAMB) can be written to and read from.
Procedure:	Four binary patterns are written and read back for comparison.
Hardware Tested:	The SRAMB is tested.
Data Returned:	The failing test number, the failing test pattern, the actual test pattern read, and the failing SRAMB address are returned.
Notes:	None

==========

Test Numbers:	10 and 11
Function:	These tests perform pattern checks on the Fault Address and Configuration registers.
Procedure:	Each register is written with four binary patterns, and a comparison is made.
Hardware Tested:	The Fault Address Register and the Configuration Register are tested.
Data Returned:	The failing test number, the failing test pattern, the actual test pattern read, and the failing register address are returned.
Notes:	In systems with more than one MMU, diagnostics will identify the failing MMU.

==========

Test Number:	12
Function:	This test checks the MMU configurations.
Procedure:	Check MMU mask levels.
Hardware Tested:	The MMU Identification (ID) registers are tested.
Data Returned:	The failing test number, the failing test pattern, the actual test pattern read, and failing register address are returned.
Notes:	If the wrong MMU code is detected, the detected and expected codes are printed. If too many or too few MMUs are detected, the expected and detected numbers are printed. If the wrong version of MMU is detected, the expected version is printed.

Phase #7 — Memory Management Unit Test #2 Diagnostics

Phase Name:	Memory Management Unit Flushing (mmu2_tst)
Type:	Normal
Function:	This phase tests the flushing capability of the Memory Management Unit (MMU).
Tests:	Test 1 — checks the SDC single entry flush. Test 2 — checks the LHPDC single entry flush. Test 3 — checks the RHPDC single entry flush. Test 4 — checks the Section 0 flush. Test 5 — checks the Section 1 flush. Test 6 — checks the Section 2 flush. Test 7 — checks the Section 3 flush.
Time:	1 second
Warnings:	None
Notes:	None

Phase #7 Tests

Test Number:	1
Function:	This test checks the single entry flushing capability of the Segment Descriptor Caches (SDCs).
Procedure:	The cache is initialized by writing to Section Random Access Memory A (SRAMA). The entries to be flushed are set up by setting the G bits and specific tags. The entries are flushed by setting up the correct unique tags and indexing into the cache to clear the G bits. Entries are verified to be flushed by reading all cache entries and making sure no entry is marked as good.
Hardware Tested:	The MMU chip is tested.
Data Returned:	The failing test number (last number displayed on the system console before failure) is returned.
Notes:	None

Test Numbers:	2 and 3
Function:	These tests check the single entry flushing capability of the Page Descriptor Caches (PDCs).
Procedure:	The cache is initialized by writing to SRAMA to flush them. The entries that are to be flushed are set up by setting the G bits and specific tags. The entries are flushed by setting up the correct unique tags and indexing into the cache to clear the G bits. The entries are verified to be flushed by reading all entries in the cache to make sure that no entry is marked as good. Test 2 — checks the Left-Hand Page Descriptor Cache (LHPDC) area. Test 3 — checks the Right-Hand Page Descriptor Cache (RHPDC) area.
Hardware Tested:	The MMU chip is tested.
Data Returned:	The failing test number (last number displayed on the system console before failure) is returned.
Notes:	None

==========

Test Numbers:	4 through 7
Function:	These tests perform a Section flush on Sections 0 through 3.
Procedure:	All G bits are set in Sections 0 through 3. A write to SRAMA is made to clear the bits. The G bit is checked for correct action.
Hardware Tested:	The integral MMU chip is tested.
Data Returned:	The failing test number (last number displayed on the system console before failure) is returned.
Notes:	None

Phase #8 — Memory Management Unit Test #3 Diagnostics

Phase Name: MMU Translation and Referenced and Modified (R&M) Updating (mmu3_tst)

Type: Normal

Function: This phase tests the translation capability and the R&M update.

Tests: Test 1 — checks the paged segment address translation.

Test 2 — checks the contiguous segment address translation.

Tests 3 through 18 — perform the R&M updating of the MMU.

Time: 1 second

Warnings: None

Notes: The 3B2 computer Memory Management Unit (MMU) test mapping scheme is as follows:

ADDRESS	DATA	SEGMENT(S)
0x00000000	EPROM Read/Execute	0 and 1
0x00040000	Miscellaneous Hardware Device and I/O Boards Read/Write	2 through 255
0x02000000	FW RAM Area Read/Write	256
0x02020000	MMU Text Read/Execute	257
0x02040000	MMU Tables Read/Write	258
0x02060000 0x0207ffff	Scratch RAM Read/Write	259

The Memory Management Tables in memory are set as follows (there are 64 pages per segment).

SDT TABLE	PDT TABLE
Segment 0 (0x02040000)	Pages 0-63 (0x02040900)
Segment 1 (0x02040008)	Pages 64-127 (0x0204a00)
Segments 2-258	Pages 128-16575
Segment 259 (0x02040818)	Page 16576 (0x02050c00)

Phase #8 Tests

Test Numbers: 1 and 2

Function: These tests check the translation of paged and contiguous segments.

Procedure: The test procedure is as follows:

1. The Page Descriptor Table (PDT) and Segment Descriptor Table (SDT) are set up.
2. Section Random Access Memory A (SRAMA) and SRAMB are set up.
3. A data pattern is written to RAM (0x02060000 - 0x020403ff) in virtual mode.
4. The data are read back in physical mode to confirm that the translation occurred correctly.

Hardware Tested: The MMU chip is tested.

Data Returned: The failing test number, the expected data, and the actual data read are returned.

Notes: None

==========

Test Numbers: 3 through 18

Function: These tests check the R&M updating of the MMU.

Procedure: Tables are set up, a segment is referenced, a segment is written, and a signature is read in this procedure.

Hardware Tested: The MMU chip is tested.

Data Returned: The failing test number (last number displayed on the system console before failure) and the failing MMU section (0 through 3) are returned. On systems with more than one MMU, the number of the failing MMU is also returned.

Notes: None

Phase #9 — Memory Management Unit Test #4 Diagnostics

Phase Name: Memory Management Unit Fault (mmu4_tst)

Type: Normal

Function: This phase tests the fault recognition ability of the Memory Management Unit (MMU).

Tests:

Test 1 — forces a SDT length fault.

Test 2 — forces a page write fault.

Test 3 — forces an invalid segment descriptor fault.

Test 4 — forces a segment not present fault.

Test 5 — forces an object trap fault.

Test 6 — forces too many indirections fault.

Test 7 — forces segment offset fault.

Tests 8 through 31 — force access fault.

Test 32 — forces access and segment offset fault.

Test 33 — forces PDT length fault.

Test 34 — forces PDT not present fault.

Test 35 — forces page not present fault.

Test 36 — forces miss process memory fault.

Test 37 — forces R&M update memory fault.

Test 38 — forces double page hit fault.

Time: 2 seconds

Warnings: The descriptor tables for memory management are set up so that the Diagnostic Monitor (DGMON) code cannot be executed while in the virtual mode.

Notes: The entire set of tests is run once per MMU. Thus on System Boards (SBDs) with one MMU (3B2/500, 600, and 700 computers), they are run once. On the SBDs with two MMUs (3B2/1000 computer), they are run twice; once for MMU0 and once for MMU1.

Phase #9 Tests

Test Number: 1

Function: This test checks the Segment Descriptor Table (SDT) length fault.

Procedure: A virtual address is referenced that has a segment select field greater than the SDT length field of Section Random Access Memory B (SRAMB).

Hardware Tested: The MMU chip is tested.

Data Returned: The failing test number, the MMU fault code, and the Fault Address register are returned.

Notes: This test is run twice on systems with two MMUs; for example, the 3B2/1000 computer.

==========

Test Number: 2

Function: This test checks the page write fault.

Procedure: The W bit is set in a Page Descriptor Cache (PDC) entry and writes are made to the virtual address that corresponds to that PDC entry.

Hardware Tested: The MMU chip is tested.

Data Returned: The MMU Fault Code register value and the MMU Fault Address register value are returned.

Notes: This test is run twice on systems with two MMUs; for example, the 3B2/1000 computer.

==========

Test Number: 3

Function: This test checks the invalid segment descriptor fault.

Procedure: The descriptor tables are set up for contiguous segments. The caches are flushed. The V bit is cleared in an SDT entry. The virtual address is referenced that corresponds to that entry.

Hardware Tested: The MMU chip is tested.

Data Returned: The MMU Fault Code register value and the MMU Fault Address register value are returned.

Notes: This test is run twice on systems with two MMUs; for example, the 3B2/1000 computer.

Test Number: 4

Function: This test checks the segment not present fault.

Procedure: The descriptor tables are set up for contiguous segments. The caches are flushed. The P bit is cleared in an SDT entry. The virtual address is referenced that corresponds to that entry.

Hardware Tested: The MMU chip is tested.

Data Returned: The MMU Fault Code register value and the MMU Fault Address register value are returned.

Notes: This test is run twice on systems with two MMUs; for example, the 3B2/1000 computer.

==========

Test Number: 5

Function: This test checks the object trap fault.

Procedure: The descriptor tables are set up for contiguous segments. The caches are flushed. The T bit is set in an SDT entry. The virtual address that corresponds to that entry is referenced.

Hardware Tested: The MMU chip is tested.

Data Returned: Raw Data: The MMU Fault Code register value is returned.
Supplemental Data: The MMU Fault Address register value is returned.

Notes:

Test	Dependencies	Function
5	WE 32101	Test object trap fault
5	WE 32201	No test run

==========

Test Number: 6

Function: This test checks the too many indirections fault.

Procedure: The descriptor tables are set up for contiguous segments. The Indirect bit in an entry is asserted and the second word of an entry is set to point to itself. The caches are flushed. The virtual address corresponding to this entry is referenced.

Hardware Tested: The MMU chip is tested.

Data Returned: The MMU Fault Code register value and MMU Fault Address register value are returned.

Notes: This test is run twice on systems with two MMUs; for example, the 3B2/1000 computer.

Test Number: 7

Function: This test checks the segment offset fault.

Procedure: The descriptor tables are set up for the contiguous segments. The segment offset of an entry is set to zero. The caches are flushed. The virtual address corresponding to this entry is referenced.

Hardware Tested: The MMU chip is tested.

Data Returned: The MMU Fault Code register value and MMU Fault Address register value are returned.

Notes: This test is run twice on systems with two MMUs; for example, the 3B2/1000 computer.

==========

Test Numbers: 8 through 31

Function: These tests check the access fault.

Procedure: The descriptor tables are set up for contiguous segments. The appropriate access occurs when allowed, and a fault occurs when disallowed. The caches are flushed. The virtual address corresponding to this entry is referenced.

Hardware Tested: The MMU chip is tested.

Data Returned: The MMU Fault Code register value and the MMU Fault Address register value are returned.

Notes: Tests 8 through 19 execute at user level while Tests 20 through 31 execute at kernel level. This test is run twice on systems with two MMUs; for example, the 3B2/1000 computer.

Test No.	Access Allowed	Operation	Test No.	Access Allowed	Operation
8	RWE	R	20	RWE	R
9	RWE	W	21	RWE	W
10	RWE	E	22	RWE	E
11	RE	R	23	RE	R
12	RE	E	24	RE	E
13	E	E	25	E	E
14	RE	W	26	RE	W
15	EO	R	27	EO	R
16	EO	W	28	EO	W
17	Nonaccessible	R	29	Nonaccessible	R
18	Nonaccessible	W	30	Nonaccessible	W
19	Nonaccessible	E	31	Nonaccessible	E

Test Number: 32

Function: This test checks the access and segment offset fault.

Procedure: The descriptor tables are set up for contiguous segments. The entry access field is set inaccessible. The maximum offset field is set to zero. The caches are flushed. The virtual address corresponding to this entry is referenced.

Hardware Tested: The MMU chip is tested.

Data Returned: The MMU Fault Code register value and the MMU Fault Address register value are returned.

Notes: This test is run twice on systems with two MMUs; for example, the 3B2/1000 computer.

==========

Test Number: 33

Function: This test checks the Page Descriptor Table (PDT) length fault.

Procedure: The descriptor tables are set for paged segments. The page describes tables that are set up. The maximum offset is set to zero. The caches are flushed. The virtual address corresponding to this entry is referenced.

Hardware Tested: The MMU chip is tested.

Data Returned: The MMU Fault Code register value and the MMU Fault Address register value are returned.

Notes: This test is run twice on systems with two MMUs; for example, the 3B2/1000.

==========

Test Number: 34

Function: This test checks the PDT not present fault.

Procedure: The descriptor tables are set for paged segments. The descriptor tables are set with page present set to zero. The caches are flushed. The virtual address corresponding to this entry is referenced.

Hardware Tested: The MMU chip is tested.

Data Returned: The MMU Fault Code register value and the MMU Fault Address register value are returned.

Notes: This test is run twice on systems with two MMUs; for example, the 3B2/1000 computer.

Test Number:	35
Function:	This test checks the page not present fault.
Procedure:	The descriptor tables are set for paged segments. The descriptor tables are set with page set to zero. The caches are flushed. The virtual address corresponding to this entry is referenced.
Hardware Tested:	The MMU chip is tested.
Data Returned:	The MMU Fault Code register value and the MMU Fault Address register value are returned.
Notes:	This test is run twice on systems with two MMUs; for example, the 3B2/1000 computer.

==========

Test Number:	36
Function:	This test checks the miss page not present fault.
Procedure:	The descriptor tables are set for paged segments. The contents of an entry is corrupted. The caches are flushed. The virtual address corresponding to this entry is referenced.
Hardware Tested:	The MMU chip is tested.
Data Returned:	The MMU Fault Code register value and the MMU Fault Address register value are returned.
Notes:	This test is run twice on systems with two MMUs; for example, the 3B2/1000 computer.

==========

Test Number:	37
Function:	This test checks the Reference and Modify (R&M) update memory fault.
Procedure:	Reference the virtual address that corresponds to this entry. The descriptor tables are set for paged segments. The caches are flushed and configured for a bit. The contents of the correspondent segment descriptor location are corrupted. The virtual address corresponding to this entry is referenced.
Hardware Tested:	The MMU chip is tested.
Data Returned:	The MMU Fault Code register value and the MMU Fault Address register value are returned.
Notes:	This test is run twice on systems with two MMUs; for example, the 3B2/1000 computer.

Test Number: 38

Function: This test checks the double page hit fault.

Procedure: The descriptor tables are set for paged segments. The caches are flushed and two PDC entries are set up. The virtual address corresponding to this entry is referenced.

Hardware Tested: The MMU chip is tested.

Data Returned: The MMU Fault Code register is returned.

Notes:

Test	Dependencies	Function
38	WE 32101	Double page hit
38	WE 32201	No test run

Phase #10 — Nonvolatile Random Access Memory Diagnostics

Phase Name:	Nonvolatile RAM (nvram_tst)
Type:	Interactive
Function:	This phase tests the Nonvolatile Random Access Memory (NVRAM) area.
Test:	Test 1 — performs an access test and pattern check on all NVRAM locations.
Time:	1 second
Notes:	The original NVRAM contents are saved before testing and restored after testing.

Phase #10 Test

Test Number:	1
Function:	This test performs an access test and pattern check on the NVRAM area.
Procedure:	All 2048 nibble locations of NVRAM are tested with the test patterns 0xff, 0xcc, 0xaa, 0x55, 0x33, and 0x00.
Hardware Tested:	The NVRAM is tested.
Data Returned:	The address, the value read, and the expected value of the failing RAM location are returned.
Notes:	The NVRAM locations are located in the least significant nibble of the least significant byte located within a 32-bit data word (x 2048).

Phase #11 — Interrupt System Diagnostics

Phase Name:	3B2 Computer Interrupt System Test (int_tst)
Type:	Normal
Function:	This phase tests the Interrupt System using trackable, noninterfering interrupt sources.
Tests:	Test 1 — checks NMI interrupt - Sanity Timer time-out bit. Test 2 — checks NMI interrupt - Abort switch activated. Test 3 — checks NMI interrupt - Thermal CSER bit activated. Test 4 — checks Level 15 interrupt - Interval timer time-out. Test 5 — checks Level 15 interrupt - power down request. Test 6 — checks Level 15 interrupt - operational interrupt. Test 7 — checks Level 15 interrupt - uBus/BuBus received fail. Test 8 — checks Level 15 interrupt - uBus timer time-out. Test 9 — checks Level 13 interrupt - DMA complete. Test 10 — checks Level 9 interrupt - CSER PIR 9. Test 11 — checks Level 8 interrupt - CSER PIR 8. Test 12 — checks Level 13 interrupt - UART forced interrupt. Test 13 — checks Level 11 interrupt - floppy disk forced interrupt.
Time:	1 second
Warnings:	A failure of this phase may affect other tests which assume that the system is clear of interrupts or that timers are turned off.
Notes:	The test interrupt handler masks all interrupts in the Process Status Word (PSW) of the main program. The System Board (SBD) Buffered Microbus (BuBus) interrupts are not tested.

System Board Interrupt/CSR Bit Assignment			
Level	**Vector**	**Source**	**CSR Bit**
NMI	0	Sanity Timer and Abort Switch	29 and 30
NMI	0	Thermal Shut Down Request	15
15	15	Interval Timer, Power Down Req.	00 and 01
15	15	uBus/BuBus OPINT	02[
15	15	Single/Multiple Bit Error	23 and 24
15	15	uBus/BuBus Received Fail	25
15	15	uBus Timer	26
15	15	I/O Received PFAIL, PFLT, FLT	—
14	SBD/P*	BuBus (Real Time Int)	—
13	13	DUARTs and DMA Complete	03 and 04
12	SBD/P*	BuBus (Block)	—
11	11	Floppy and Floppy DMA Complete	—
10	SBD/P*	BuBus (Character)	—
9	9	PIR-9 (from CSER)	05
8	8	PIR-8 (from CSER)	06

* SBD/P - For BuBus connector: First-level interrupt vector provided by system board HW; second-level interrupt vector provided by periphery.

For 3B2 Input/Output (I/O) bus: Single vector provided by periphery.

System Board Interrupt/CSR Bit Assignment			
Level	**Vector**	**Source**	**CSR Bit**
15	15	System Error** and Periodic Timer	15
15	15	Parity Error	14
14	*	I/O Boards PINT20	—
13	13	UARTs and DMA Complete	03/01
12	*	I/O Boards PINT10	—
11	11	Integral Disk	02
10	*	I/O Boards PINT00	—
9	9	PIR-9 (from CSR)	04
8	8	PIR-8 (from CSR)	05

* 8-bit vector supplied by I/O board.
** System Error = Bus time-out, parity error, or I/O board failure.

Phase #11 Tests

Test Number:	1
Function:	This test checks that an NMI interrupt is generated when the Sanity Timer Time-out bit is set in the Control Status and Error Register (CSER).
Procedure:	The CPU interrupt level is set to 15 to block all normal interrupts. The Sanity Timer Time-out bit is set in the CSER. Confirm that an NMI interrupt is generated and the CSER accurately reflects the bit set.
Hardware Tested:	The Interrupt System and CSER are tested.
Data Returned:	The contents of the CSR read from the interrupt handler is returned.
Notes:	None

==========

Test Number:	2
Function:	This test checks that an NMI interrupt is generated when the Abort bit is set in the CSER.
Procedure:	The Abort bit in the CSER is set. Confirm that an NMI interrupt is generated and the CSER accurately reflects the bit being set.
Hardware Tested:	The Interrupt System and CSER are tested.
Data Returned:	The contents of the CSR read from the interrupt handler is returned.
Notes:	None

==========

Test Number:	3
Function:	This test checks that an NMI interrupt is generated when the Thermal Shut Down Request bit is set in the CSER.
Procedure:	The Thermal Shut Down Request bit in the CSER is set. Confirm that an NMI interrupt is generated and that the CSER accurately reflects the bit set.
Hardware Tested:	The Interrupt System and CSER are tested.
Data Returned:	The contents of the CSR read from the interrupt handler is returned.
Notes:	None

Test Numbers:	4 through 11
Function:	These tests check that each of the tested bits in the CSER generate an interrupt and that the CSER reflects their being set.
Procedure:	These tests loop through each of the bits tested, confirming that an interrupt was taken and that the CSER accurately reflects the bit being set in the interrupt handler.
Hardware Tested:	The Interrupt System and CSER are tested.
Data Returned:	The contents of the CSER read from the interrupt handler is returned.
Notes:	None

==========

Test Number:	12
Function:	This test checks that a Level 13 interrupt is generated when the B side of the Dual Universal Asynchronous Receiver/Transmitter (DUART) issues an interrupt request.
Procedure:	The B-side DUART is set, and the Interrupt register is loaded with an interrupt request. A Level 13 interrupt is taken and the CSER accurately reflects a pending Universal Asynchronous Receiver/Transmitter (UART) interrupt request status.
Hardware Tested:	The Interrupt System, DUART, and CSER are tested.
Data Returned:	The contents of the CSER as read from the interrupt handler is returned.
Notes:	None

==========

Test Number:	13
Function:	This test checks that a Level 11 interrupt is generated when the Floppy Disk Controller issues an interrupt request.
Procedure:	The Floppy Disk Command register is loaded with a force interrupt command and a Level 11 interrupt is taken.
Hardware Tested:	The Interrupt System and Floppy Disk Controller are tested.
Data Returned:	The contents of the CSER read from the interrupt handler is returned.
Notes:	None

Phase #12 — Sanity/Interval Timer Diagnostics

Phase Name: INTEL 8254 Programmable Interval Timer/Counter (sit_tst)

Type: Normal

Function: This phase tests the access to and operation of the INTEL 8254 System Timer.

Tests: Tests 1 through 6 — perform access tests on Counters 0 through 2 using the load most significant and load least significant byte commands.

Tests 7 through 12 — check the ability to perform 2-byte load operations for Counters 0, 1, and 2.

Tests 13 and 14 — verify that Counter 1 operates properly.

Tests 15 and 16 — verify that Counter 1 operates properly.

Test 17 — verifies that Counter 2 operates properly.

Tests 18 through 20 — verify the read back capability of each counter.

Time: 3 seconds

Warnings: A failure in this phase may affect other tests which assume the timers to be inactive.

Notes: None

Phase #12 Tests

Test Numbers: 1 through 6

Function: These tests perform an access test on Counters 0 through 2.

Procedure: Using the load Most Significant Byte (MSB) and load Least Significant Byte (LSB) commands, the most significant byte and least significant byte of each counter are tested with the data patterns 0x00, 0x55, 0xaa, and 0xff.

Tests 1 and 2 — check Counter 0 MSB and LSB, respectively.

Tests 3 and 4 — check Counter 1 MSB and LSB, respectively.

Tests 5 and 6 — check Counter 2 MSB and LSB, respectively.

Hardware Tested: The INTEL 8254 Programmable Interval Timer/Counter is tested.

Data Returned: The failing test number (last number displayed on the system console before failure) is returned.

Notes: None

Test Numbers:	7 through 12
Function:	These tests check the ability to perform 2-byte load operations for Counters 0, 1, and 2.
Procedure:	Each counter is loaded with specific data patterns using the load multiple byte commands, and then they are read for comparison. Tests 7 and 8 — check Counter 0 LSB and MSB, respectively. Tests 9 and 10 — check Counter 1 LSB and MSB, respectively. Tests 11 and 12 — check Counter 2 LSB and MSB, respectively.
Hardware Tested:	The INTEL 8254 Programmable Interval Timer/Counter is tested.
Data Returned:	The failing test number (last number displayed on the system console before failure) is returned.
Notes:	None

==========

Test Numbers:	13 and 14
Function:	These tests verify that Counter 0 is operational.
Procedure:	The timer is initialized; it is set running; it is stopped; its run is confirmed.
Hardware Tested:	The INTEL 8254 Programmable Interval Timer/Counter is tested.
Data Returned:	The failing test number (last number displayed on the system console before failure) is returned.
Notes:	None

==========

Test Numbers:	15 and 16
Function:	These tests verify that Counter 1 is operational and that a time-out generates an interrupt and sets the clk bit in the Control Status and Error Register (CSER).
Procedure:	The timer is initialized; it is set running; it is stopped; its run is confirmed.
Hardware Tested:	The INTEL 8254 Programmable Interval Timer/Counter is tested.
Data Returned:	The failing test number (last number displayed on the system console before failure) is returned.
Notes:	None

Test Number:	17
Function:	This test verifies that Counter 2 is operational, that a time-out generates an exception and interrupt, and that the ubto bit in the CSER is set.
Procedure:	The timer is initialized; it is set running; it is delayed for interrupt; its run is confirmed.
Hardware Tested:	The INTEL 8254 Programmable Interval Timer/Counter is tested.
Data Returned:	The failing test number (last number displayed on the system console before failure) is returned.
Notes:	None

==========

Test Numbers:	18 through 20
Function:	These tests verify that the read back command contains the correct status for each counter.
Procedure:	Initialize each counter. Check that the Status register and the contents of each counter are correct.
Hardware Tested:	The INTEL 8254 Programmable Interval Timer/Counter is tested.
Data Returned:	The failing test number (last number displayed on the system console before failure) is returned.
Notes:	None

Phase #13 — Control and Status Register Diagnostics

Phase Name: Control and Status Register (csr_tst)

Type: Normal

Function: This phase performs access tests on the Control Status and Error Register (CSER).

Tests: Tests 1 and 2 — confirm that CSER bits 10, 12, 13, and 28 can be set and cleared by software.

Test 3 — confirms that data alignment bit 28 gets set by bad data alignment access.

Time: 1 second

Warnings: None

Notes: None

Phase #13 Tests

Test Numbers: 1 and 2

Function: These tests verify that CSER bits 10, 12, 13, and 28 can be set and cleared by software.

Procedure: The test procedure is as follows:

1. Clear bits.
2. Verify bits are cleared.
3. Set bits.
4. Verify bits are set.
5. Clear bits.

Hardware Tested: The CSER is tested.

Data Returned: The failing test number (last number displayed on the system console before failure), plus the failing CSER contents are returned.

Notes: None

Test Number:	3
Function:	This test verifies that CSER bit 28 gets set due to nonword alignment access.
Procedure:	The test procedure is as follows: 1. Save the address of a variable. 2. Increment the contents of the variable. 3. Access the memory. 4. Check that CSER bit 28 is set.
Hardware Tested:	The CSER is tested.
Data Returned:	The failing test number (last number displayed on the system console before failure), plus the failing CSER contents are returned.
Notes:	None

Phase #14 — Dual UART Diagnostics

Phase Name: 2681 Dual Asynchronous Receiver/Transmitter (DUART) Tests (duart_tst)

Type: Interactive

Function: The DUART is tested in the local loopback mode by transmitting and receiving characters. The appropriate Status bits are checked as well as the correct transmission and reception of the data. The Command and Mode registers are checked for read and write capability.

Tests: The following table lists the tests that are run in this phase:

SIDE	TEST NUMBER	TEST FUNCTION
A, B	1-16, 17-32	These tests walk a one through zeros in Mode register 2, and they walk a zero through ones in Mode register 1.
A, B	33-36, 37-40	These tests check the Receive Data bit in the Status register so that it may be used as a flag in subsequent tests. This is done by Tests 33 through 37 and 35 through 39. At the same time, the ability to transmit and receive a character is checked by Tests 34 through 38 and 36 through 40.
A, B	41-47, 48-54	These tests walk a one through zeros at seven different baud rates.
A, B	55-57, 58-60	These tests walk a one through zeros with 7, 6, and 5 Data bits (8 bits already used).
A, B	61-62, 63-64	These tests check even parity and no parity (odd parity already tested).
A, B	65-66, 67-68	These tests transmit characters with both 1 and 1.5 Stop bits (2 already tested).
A, B	69-70, 71-72	These tests check the Transmitter Empty and Transmitter Ready bits in the Status register for correct operation.
A	73-74, 75-77	These tests check the Received Break and Parity Error Send/Receive bits. They also check the Overrun and First-In-First-Out (FIFO) Full Send/Receive bits.
B	78-79, 80-82	These tests check the Received Break and Parity Error Send/Receive bits. They also test the Overrun and FIFO Full Send/Receive bits.
A	83-84	These tests confirm that INTRN is activated when RxRDY A occurs and when TxRDY A occurs.
B	85-86	These tests confirm that INTRN is activated when RxRDY B occurs and when TxRDY B occurs.
A	87	This test confirms that INTRN is activated when FFULL A occurs.
B	88	This test confirms that INTRN is activated when FFULL B occurs.
A	89	This test confirms that INTRN is activated when Delta Break A is received.
B	90	This test confirms that INTRN is activated when Delta Break B is received.
A, B	92-93	These tests confirm the FIFO Shift register capacity.

Time: 5 seconds

Warnings: None

Notes: The system will specify that "CONSOLE must be connected to console side of DUART, and CONTTY port must be disconnected."

Phase #14 Tests

Test Numbers: 1 through 16

Function: These tests write eight different patterns to Mode register A.

Procedure: A one is walked through zeros in Mode register 2, and a zero is walked through ones in Mode register 1.

Hardware Tested: The SCN2681 DUART is tested.

Data Returned: The failing register number and value are returned.

Notes: None

==========

Test Numbers: 17 through 32

Function: These tests write eight different patterns to the Mode registers B.

Procedure: A one is walked through zeros in Mode register 2, and a zero is walked through ones in Mode register 1.

Hardware Tested: The SCN2681 DUART is tested.

Data Returned: The failing register number and value are returned.

Notes: None

==========

Test Numbers: 33 through 36

Function: These tests check the ability of the Universal Asynchronous Receiver/Transmitter (UART) A to transmit and receive a character correctly and to set or reset the Receive Status bit in the Status register of the UART.

Procedure: The test procedure is as follows:

1. A character is transmitted.
2. The Receive Status bit is looped on until time-out occurs or it is set.
3. The validity of the received character is checked if time-out does not occur.
4. The receive flag is checked to be cleared upon reading the data.

Hardware Tested: The SCN2681 DUART is tested.

Data Returned: The failing register number and the status of the RCVRDY and XMTRDY Time-out bits are returned.

Notes: None

Test Numbers: 37 through 40

Function: These tests check the ability of the UART B to transmit and receive a character correctly and to set or reset the Receive Status bit in the Status register of the UART.

Procedure: The test procedure is as follows:

1. A character is transmitted.
2. The Receive Status bit is looped on until a time-out occurs or it is set.
3. The validity of the received character is checked if a time-out does not occur.
4. The receive flag is checked to be cleared upon reading the data.

Hardware Tested: The SCN2681 DUART is checked.

Data Returned: The failing register number and the status of the RCVRDY and XMTRDY Time-out bits are returned.

Notes: None

==========

Test Numbers: 41 through 47

Function: These tests check transmission at different baud rates on the A side.

Procedure: A one is walked through zeros at baud rates of 110, 300, 1200, 2400, 4800, 9600, and 19200.

Hardware Tested: The SCN2681 DUART is tested.

Data Returned: The failing register number and the status of the RCVRDY and XMTRDY Time-out bits are returned.

Notes: None

==========

Test Numbers: 48 through 54

Function: These tests check transmission at different baud rates on the B side.

Procedure: A one is walked through zeros at baud rates of 110, 300, 1200, 2400, 4800, 9600, and 19200.

Hardware Tested: The SCN2681 DUART is tested.

Data Returned: The failing register number and the status of the RCVRDY and XMTRDY Time-out bits are returned.

Notes: None

Test Numbers:	55 through 57
Function:	These tests check transmission with different data lengths on the A side.
Procedure:	A one is walked through zeros at 7, 6, and 5 Data bits (8 already tested).
Hardware Tested:	The SCN2681 DUART is tested.
Data Returned:	The failing register number and the status of the RCVRDY and XMTRDY Time-out bits are returned.
Notes:	None

==========

Test Numbers:	58 through 60
Function:	These tests check transmission with different data lengths on the B side.
Procedure:	A one is walked through zeros at 7, 6, and 5 Data bits (8 already tested).
Hardware Tested:	The SCN2681 DUART is tested.
Data Returned:	The failing register number and the status of the RCVRDY and XMTRDY Time-out bits are returned.
Notes:	None

==========

Test Numbers:	61 and 62
Function:	These tests check transmission with different parities on the A side.
Procedure:	A one is walked through zeros at even and no parity (odd already tested).
Hardware Tested:	The SCN2681 DUART is tested.
Data Returned:	The failing register number and the status of the RCVRDY and XMTRDY Time-out bits are returned.
Notes:	None

==========

Test Numbers:	63 and 64
Function:	These tests check transmission with different parities on the B side.
Procedure:	Walk a one through zeros at even and no parity (odd already tested).
Hardware Tested:	The SCN2681 DUART is tested.
Data Returned:	The failing register number and the status of the RCVRDY and XMTRDY Time-out bits are returned.
Notes:	None

Test Numbers:	65 and 66
Function:	These tests check transmission at 1 and 1.5 Stop bits (2 already tested) on the A side.
Procedure:	Walk a one through zeros at 1 and 1.5 Stop bits.
Hardware Tested:	The SCN2681 DUART is tested.
Data Returned:	The failing register number and the status of the RCVRDY and XMTRDY Time-out bits are returned.
Notes:	None

==========

Test Numbers:	67 and 68
Function:	These tests check transmission at 1 and 1.5 Stop bits (2 already tested) on the B side.
Procedure:	Walk a one through zeros at 1 and 1.5 Stop bits.
Hardware Tested:	The SCN2681 DUART tested.
Data Returned:	The failing register number and the status of the RCVRDY and XMTRDY Time-out bits are returned.
Notes:	None

==========

Test Numbers:	69 and 70
Function:	These tests check the bits, other than Error bits, of the Status register on the A side.
Procedure:	Test 69 — transmits two characters and checks txemt and txrdy for zero. Test 70 — detects the character at the receiver and checks that txemt and txrdy have been set to one.
Hardware Tested:	The SCN2681 DUART is tested.
Data Returned:	The failing test number and failing Status register value are returned.
Notes:	None

==========

Test Numbers:	71 and 72
Function:	These tests check the bits (other than Error bits) of the Status register on the B side.
Procedure:	Test 71 — transmits two characters and checks txemt and txrdy for 0. Test 72 — detects the character at the receiver and checks that txemt and txrdy have been set to one.
Hardware Tested:	The SCN2681 DUART is tested.
Data Returned:	The failing test number and failing Status register value are returned.
Notes:	None

Test Numbers: 73 and 74

Function: These tests check the Received Break and Parity Error Send/Receive bits on the A side.

Procedure: The test procedure is as follows:

1. Transmit a break character.
2. Confirm that the Received Break and Parity Error Send/Receive bits are set.
3. Confirm that Error bits are cleared after read.

Hardware Tested: The SCN2681 DUART is tested.

Data Returned: The failing test number and failing Status register value are returned.

Notes: None

==========

Test Numbers: 75 through 77

Function: These tests check the Overrun and First-In-First-Out (FIFO) Full Send/Receive bits on the A side.

Procedure: The test procedure is as follows:

1. Transmit three characters.
2. Check that the FIFO Full bit is set.
3. Transmit two additional characters.
4. Check that the Overrun and FIFO Full bits are set.
5. Confirm that the Overrun Error bit is cleared by RESET_ERR command to the Command register.

Hardware Tested: The SCN2681 DUART is tested.

Data Returned: The failing test number and failing Status register value are returned.

Notes: None

==========

Test Numbers: 78 and 79

Function: These tests check the Received Break and Parity Error Send/Receive bits.

Procedure: The test procedure is as follows:

1. Transmit a break character.
2. Confirm that Received Break and Parity Error Send/Receive bits are set.
3. Confirm that Error bits are cleared after the read.

Hardware Tested: The SCN2681 DUART is tested.

Data Returned: The failing test number and failing Status register value are returned.

Notes: None

Test Numbers: 80 through 82

Function: These tests check the Overrun and FIFO Full Send/Receive bits of the A side.

Procedure: The test procedure is as follows:

1. Transmit three characters.
2. Check that the FIFO Full bit is set.
3. Transmit two additional characters.
4. Check that the Overrun and FIFO Full bits are set.
5. Confirm that the Overrun Error bit is cleared by the RESET_ERR command to the Command register.

Hardware Tested: The SCN2681 DUART is tested.

Data Returned: The failing test number and failing Status register value are returned.

Notes: None

==========

Test Numbers: 83 and 84

Function: These tests confirm that INTRN is activated when RxRDY A and TxRDY A occur for the A side.

Procedure: Call the **dint_tst( )** function to perform a complete character transmission and read. The **dint_tst( )** function stops if it detects an interrupt.

Hardware Tested: The SCN2681 DUART is tested.

Data Returned: The failing test number is returned.

Notes: None

==========

Test Numbers: 85 and 86

Function: These tests confirm that INTRN is activated when RxRDY B and TxRDY B occur for the B side.

Procedure: Call the **dint_tst( )** function to perform a complete character transmission and read. The **dint_tst( )** function stops if it detects an interrupt.

Hardware Tested: The SCN2681 DUART is tested.

Notes: None

Test Number:	87
Function:	This test confirms that INTRN is activated when FFULL A occurs.
Procedure:	Set the device up to issue an interrupt when the FFULL A condition occurs; then transmit three characters and wait for interrupt.
Hardware Tested:	The SCN2681 DUART is tested.
Data Returned:	The failing test number is returned.
Notes:	None

==========

Test Number:	88
Function:	This test confirms that INTRN is activated when FFULL B occurs.
Procedure:	Set the device up to issue an interrupt when the FFULL B condition occurs; then transmit three characters and wait for interrupt.
Hardware Tested:	The SCN2681 DUART is tested.
Data Returned:	The failing test number is returned.
Notes:	None

==========

Test Number:	89
Function:	This test confirms that INTRN is activated when Delta Break is received from A side.
Procedure:	The device issues an interrupt when a Delta Break condition occurs. Then a break character and a wait for interrupt is transmitted.
Hardware Tested:	The SCN2681 DUART is tested.
Data Returned:	The failing test number is returned.
Notes:	None

==========

Test Number:	90
Function:	This test confirms that INTRN is activated when Delta Break is received from B side.
Procedure:	The device issues an interrupt when a Delta Break condition occurs. Then a break character and a wait for interrupt is transmitted.
Hardware Tested:	The SCN2681 DUART is tested.
Data Returned:	The failing test number is returned.
Notes:	None

Test Number:	91
Function:	This test confirms that INTRN is activated when Counter Ready occurs.
Procedure:	The device issues an interrupt when the counter is ready. Calls hwtimer( ) which uses the counter. An interrupt was processed when the counter expired.
Hardware Tested:	The SCN2681 DUART is tested.
Data Returned:	The failing test number is returned.
Notes:	None

==========

Test Number:	92
Function:	This test confirms the stacking of the FIFO and Shift register characteristics on the A side.
Procedure:	Four characters are transmitted and verified upon reception.
Hardware Tested:	The SCN2681 DUART is tested.
Data Returned:	The failing test number is returned.
Notes:	None

==========

Test Number:	93
Function:	This test confirms the stacking of the FIFO and Shift register characteristics on the B side.
Procedure:	Four characters are transmitted and verified upon reception.
Hardware Tested:	The SCN2681 DUART is tested.
Data Returned:	The failing test number is returned.
Notes:	None

Phase #15 — Extended 2681 Dual UART Diagnostics

Phase Name: Extended 2681 Dual Universal Asynchronous Receiver/Transmitter (DUART) Tests (eduart_tst)

Type: Interactive

Function: The DUART is tested in the normal loopback mode by transmitting and receiving characters. The appropriate Status bits are checked as well as the correct transmission and reception of the data. The Command and Mode registers are checked for read and write capability.

Tests: The following table lists the tests that run in this phase.

SIDE	TEST NUMBER	TEST FUNCTION
B	1 through 16	These tests walk a one through zeros in Mode register 2, and walk a zero through ones in Mode register 1.
B	17 through 20	These tests check the Receive Data bit in the Status register so that it may be used as a flag in subsequent tests. This test occupies Tests 17 and 19. At the same time the ability to transmit and receive a character is tested by Tests 18 and 20.
B	21 through 27	These tests walk a one through zeros at seven different baud rates.
B	28 through 30	These tests walk a one through zeros with Data bits 7, 6, and 5 (8 bits already used).
B	31 and 32	These tests check even parity and no parity (odd parity already tested).
B	33 and 34	These tests transmit characters with 1 and then 1.5 Stop bits (2 already tested).
B	35 and 36	These tests check the Transmitter Empty and Transmitter Ready bits in the Status register for correct operation.
B	37 and 38	These tests check the Received Break and Parity Error Send/Receive bits.
B	39 through 41	These tests check the Overrun and FIFO Full Send/Receive bits.
B	42 and 43	These tests confirm that INTRN is activated when RxRDY B occurs and when TxRDY B occurs.
B	44	This test confirms that INTRN is activated when FFULL B occurs.
B	45	This test confirms that INTRN is activated when Delta Break B is received.
B	46 and 47	These tests confirm that data carrier detect can be asserted and reset.
B	48	This test confirms that INTRN is activated when Counter Ready occurs.

Time: 2 seconds

Warnings: None

Notes: This test requires the use of a loopback connector. The system console must be connected to the console side of DUART.

Phase #15 Tests

Test Numbers: 1 through 16

Function: These tests write eight different patterns to Mode register B.

Procedure: Walk a one through zeros in Mode register 2, and walk a zero through ones in Mode register 1.

Hardware Tested: The SCN2681 DUART is tested.

Data Returned: The failing test number and register value are returned.

Notes: None

==========

Test Numbers: 17 through 20

Function: These tests check the ability of the Universal Asynchronous Receive/Transmitter (UART) B to transmit and receive a character correctly, and set or reset the Receive Status bit in the Status register of the UART.

Procedure: The test procedure is as follows:

1. Transmit a character.
2. Loop on the Receive Status bit until time-out occurs or it is set.
3. Check the validity of the received character if time-out does not occur.
4. Check that the receive flag was cleared upon reading the data.

Hardware Tested: The SCN2681 DUART is tested.

Data Returned: The failing test number, failing Status register contents on the RCVRDY and XMTRDY Time-out bits, and other expected versus failing characters are returned.

Notes: None

==========

Test Numbers: 21 through 27

Function: These tests check transmission at different baud rates on the B side.

Procedure: Walk a one through zeros at baud rates 110, 300, 1200, 2400, 4800, 9600, and 19200.

Hardware Tested: The SCN2681 DUART is tested.

Data Returned: The failing test number, failing Status register contents on the RCVRDY and XMTRDY Time-out bits, and other expected versus failing characters are returned.

Notes: None

Test Numbers:	28 through 30
Function:	These tests check transmission with different data lengths on the B side.
Procedure:	Walk a one through zeros using 7, 6, and 5 Data bits (8 already tested).
Hardware Tested:	The SCN2681 DUART is tested.
Data Returned:	The failing test number, failing Status register contents on the RCVRDY and XMTRDY Time-out bits, and other expected versus failing characters are returned.
Notes:	None

==========

Test Numbers:	31 and 32
Function:	These tests check transmission with different parities on the B side.
Procedure:	Walk a one through zeros at even and no parity (odd already tested).
Hardware Tested:	The SCN2681 DUART is tested.
Data Returned:	The failing test number, failing Status register contents on the RCVRDY and XMTRDY Time-out bits, and other expected versus failing characters are returned.
Notes:	None

==========

Test Numbers:	33 and 34
Function:	These tests check transmission at 1 and 1.5 Stop bits (2 already tested) on the B side.
Procedure:	Walk a one through zeros at 1 and 1.5 Stop bits.
Hardware Tested:	The SCN2681 DUART is tested.
Data Returned:	The failing test number, failing Status register contents on the RCVRDY and XMTRDY Time-out bits, and other expected versus failing characters are returned.
Notes:	None

Test Numbers: 35 and 36

Function: These tests check the bits, other than Error bits, of the Status register on the B side.

Procedure: Test 35 — transmits two characters and checks txemt and txrdy for zero.

Test 36 — detects character at the receiver and checks that txemt and txrdy have been set to one.

Hardware Tested: The SCN2681 DUART is tested.

Data Returned: The failing test number and failing Status register value are returned.

==========

Test Numbers: 37 and 38

Function: These tests check the Received Break and Parity Error Send/Receive bits on the B side.

Procedure: The test procedure is as follows:

1. Transmit a break character.
2. Confirm that Received Break and Parity Error Send/Receive bits are set.
3. Confirm that Error bits are cleared after read.

Hardware Tested: The SCN2681 DUART is tested.

Data Returned: The failing test number and failing Status register value are returned.

Notes: None

==========

Test Numbers: 39 through 41

Function: These tests check the Overrun and First-In-First-Out (FIFO) Full Send/Receive bits on the B side.

Procedure: The test procedure is as follows:

1. Transmit three characters.
2. Check that the FIFO Full bit is set.
3. Transmit two more additional characters.
4. Check that the Overrun and FIFO Full bits are set.
5. Confirm that the Overrun Error bit is cleared by the RESET_ERR command to the Command register.

Hardware Tested: The SCN2681 DUART is tested.

Data Returned: The failing test number and failing Status register value are returned.

Notes: None

Test Numbers:	42 and 43
Function:	These tests confirm that INTRN is activated when RxRDY B and TxRDY B occur for the B side.
Procedure:	Call the **dint_tst( )** function which attempts to perform a complete character transmission and read. The **dint_tst( )** function stops if it detects an interrupt.
Hardware Tested:	The SCN2681 DUART is tested.
Data Returned:	The failing test number is returned.
Notes:	None

==========

Test Number:	44
Function:	This test confirms that INTRN is activated when FFULL B occurs.
Procedure:	The test procedure is as follows: 1. Set the device up to issue an interrupt when the FFULL B condition occurs. 2. Transmit three characters and wait for interrupt.
Hardware Tested:	The SCN2681 DUART is tested.
Data Returned:	The failing test number is returned.
Notes:	None

==========

Test Number:	45
Function:	This test confirms that INTRN is activated when Delta Break is received for the B side.
Procedure:	The test procedure is as follows: 1. Set the device up to issue an interrupt when a Delta Break condition occurs. 2. Transmit a break character and wait for interrupt.
Hardware Tested:	The SCN2681 DUART is tested.
Data Returned:	The failing test number is returned.
Notes:	None

Test Numbers:	46 and 47
Function:	These tests check for active DCD Channel B in the Input registers.
Procedure:	The test procedure is as follows: 1. Invoke data terminal ready. 2. Ensure that data carrier detect has been invoked. 3. Rescind data terminal ready. 4. Check data carrier detect is rescinded.
Hardware Tested:	The SCN2681 DUART is tested.
Data Returned:	The failing test number is returned.
Notes:	None

==========

Test Number:	48
Function:	This test confirms that INTRN is activated when Counter Ready occurs.
Procedure:	Sets the device up to issue an interrupt when the counter goes ready. Calls hwtimer( ) which uses the counter. Confirms that an interrupt was processed when the counter expired.
Hardware Tested:	The SCN2681 DUART is tested.
Data Returned:	The failing test number is returned.
Notes:	None

Phase #16 — Dual UART (DUART)/Keyboard Diagnostic

Phase name: Keyboard Verification Test (keypad)

Type: Interactive

Function: This phase verifies that the keyboard responds to the American Standard Code for Information Interchange (ASCII) set.

Tests: Test 1 — tests all ASCII characters that can be received from the keyboard.

Test 2 — performs the break key test.

Time: User dependent

Warnings: Do not run this test with an AT&T 4425 terminal.

Notes: None

Phase #16 Tests

Test Number: 1

Function: This test checks the console response to the ASCII character set.

Procedure: Request user acknowledgment; then verify requested console input is contained within the ASCII set.

Hardware Tested: The Dual Universal Asynchronous Receiver/Transmitter (DUART) 2681/Keyboard Interface is tested.

Data Returned: If there is no match, the received and expected characters are printed.

Warnings: Do not run this test with an AT&T 4425 terminal.

Notes: The control characters NUL and RS are difficult or impossible to generate on certain terminals. The AT&T 4425 terminal has no way of producing these characters. Other terminals may be able to produce them but with unusual keypresses. The following table shows some variations.

Character	Hexadecimal	Std. Key	Alt. Keys
NUL	00	control-@	control-'
RS	1E	control-^	control-.

On most terminals control-^ is shift-control-6. The AT&T 54XX and 44XX terminals require use of the alternate keys (if they support them at all). Other AT&T terminals and most non-AT&T terminals support standard keypresses.

==========

Test Number: 2

Function: This test checks the break key response.

Procedure: Verify that console can enter a break.

Hardware Tested: The DUART 2681/Keyboard Interface is tested.

Data Returned: The pass/fail status is returned.

Warnings: None

Notes: None

Phase #17 — Time-of-Day Clock Diagnostics

Phase Name:	Time-of-Day Clock Test (tod_tst).
Type:	Interactive
Function:	This phase tests the real-time clock.
Tests:	Test 1 — checks the start and stop function and the reset function of the Seconds registers. Test 2 — executes zero test of registers for seconds, minutes, hours, days, day of week, and month. Test 3 — does a write test of registers with a bit pattern. Test 4 — performs a roll test of all writable registers. Test 5 — checks for leap year.
Time:	15 seconds
Warning:	Destroys contents of Time-of-Day clock.
Notes:	None

Phase #17 Tests

Test Number:	1
Function:	This test checks the start and stop function and the reset function of the Seconds registers.
Procedure:	The clock is stopped, and the 3 Seconds registers (tens, seconds, and tenths-of-seconds) are verified to be reset to zero.
Hardware Tested:	The MM5817A Real-Time Clock is tested.
Data Returned:	The failing test number is returned.
Notes:	None

==========

Test Number:	2
Function:	This test verifies that all writable registers can be zeroed.
Procedure:	The seconds, minutes, hours, days, day of week, and months registers are zeroed and verified.
Hardware Tested:	The MM5817A Real-Time Clock is tested.
Data Returned:	The failing test number (last number displayed on the system console before failure) is returned.
Notes:	None

Test Number: 3

Function: This test checks all writable registers.

Procedure: All registers are written and verified to contain the written value.

Hardware Tested: The MM5817A Real-Time Clock is tested.

Data Returned: The failing test number is returned.
Supplemental Data: The expected data and the received data are returned.

Notes: None

==========

Test Number: 4

Function: This test checks all writable registers for 24-hour and 12-hour modes of operation.

Procedure: The test procedure is as follows:

1. Write each register with the maximum value.
2. Let the register sleep for 1 second.
3. Check that the register did not return to zero or that the next higher register was incremented.

Hardware Tested: The MM5817A Real-Time Clock is tested.

Data Returned: The failing test number is returned.
Supplemental Data: The expected data and the received data are returned.

Notes: None

==========

Test Number: 5

Function: This test performs the leap year test.

Procedure: The test procedure is as follows:

1. The Leap Year register is set and the Month and Day registers are set to February 28.
2. The clock is incremented so that February 29 occurs.
3. The Leap Year register is reset to a nonleap year. The clock is again incremented and March 1 occurs.
4. Repeat for the next two consecutive nonleap years.

Hardware Tested: The MM5817A Real-Time Clock is tested.

Data Returned: The failing test number is returned.
Supplemental Data: The expected data and the received data are returned.

Notes: None

Phase #18 — Direct Memory Access Controller Diagnostics

Phase Name: Direct Memory Access Controller (DMAC) (dmac_tst)

Type: Normal

Function: This phase tests the registers and operations of the AMD 9517 Direct Memory Access Controller (DMAC).

Tests: Tests 1 through 16 — perform pattern tests on the Current Address register and the Current Word register for all channels.

Test 17 — checks DMA operation using DUART B in local loop around mode.

Test 18 — checks DMA operation using DUART A in local loop around mode.

Test 19 — checks Fault Latch registers one and two for correct failing address data after an I/O fault has occurred.

Test 20 — checks DMA operation using DUART B in local loop around mode and perform DMA operations at every megabyte of memory.

Time: 2 seconds

Warnings: None

Notes: None

Phase #18 Tests

Test Numbers: 1 through 16

Function: These tests perform a pattern test on each of the Current Address and Current Word registers for each channel.

Procedure: The test procedure is as follows:

1. Clear byte pointer in DMAC.
2. Load pattern into lower Channel register under test.
3. Clear byte pointer in DMAC.
4. Read and confirm lower data pattern reception for Channel register under test.
5. Read and confirm upper data pattern reception for Channel register under test.

Hardware Tested: The 9517 DMAC is tested.

Data Returned: The failing test number (last number displayed on the system console before failure) and the expected versus the read pattern are returned.

Notes: Test numbers are assigned as follows:

Test No.	Current Address Register	Test No.	Current Word Register
1-2	Channel 0	3-4	Channel 0
5-6	Channel 1	7-8	Channel 1
9-10	Channel 2	11-12	Channel 2
13-14	Channel 3	15-16	Channel 3

==========

Test Number: 17

Function: This test confirms that the DMAC can perform a Direct Memory Access (DMA) operation.

Procedure: The test procedure is as follows:

1. Initialize B-side Dual Universal Asynchronous Receiver/Transmitter (DUART) to local loop around mode.
2. Initialize DMAC for DMA operation through the DUART. Job is to transmit a 4-character string "TEST."
3. Request DMA job and delay for job completion.
4. Check B side of DUART for First-In-First-Out (FIFO) Full bit set in DUART Status register and confirm reception of character string "TEST."

Hardware Tested: The 9517 DMAC is tested.

Data Returned: The failing test number (last number displayed on the system console before failure), as well as error data presented are returned.

Notes: None

==========

Test Number: 18

Function: This test confirms that the DMAC can perform a DMA operation.

Procedure: The test procedure is as follows:

1. Initialize A-side DUART to local loop around mode.
2. Initialize DMAC for DMA operation through the DUART. Job is to transmit a 4-character string "TEST."
3. Request DMA job and delay for job completion.
4. Check A side of DUART for FIFO Full bit set in DUART Status register and confirm reception of character string "TEST."

Hardware Tested: The 9517 DMAC is tested.

Data Returned: The failing test number (last number displayed on the system console before failure), as well as error data presented are returned.

Notes: None

Test Number: 19

Function: This test verifies that Fault Latch registers contain the corresponding data after a DMA operation on a faulted address.

Procedure: The test procedure is as follows:

1. Initialize B-side DUART to local loop around mode.
2. Initialize DMAC for DMA operation through the DUART.
3. Next enable the force Error Correction Coding (ECC) syndrome.
4. Write data to faulted memory address and disable FECC syndrome.
5. Request DMA job and delay for job completion.
6. Verify that Fault Latch register 1 contains the faulted address and that Fault Latch register 2 contains status that Input/Output (I/O) was master at time of fault.
7. Verify Control Status and Error Register (CSER) indicates that Fault registers were loaded.
8. Restore faulted address.

Hardware Tested: The 9517 DMAC and Fault Latch registers 1 and 2 are tested.

Data Returned: The failing test number (last number displayed on the system console before failure), as well as error data presented are returned.

Notes: None

==========

Test Number: 20

Function: This test confirms that the DMAC can perform a DMA operation at every megabyte of memory.

Procedure: The test procedure is as follows:

1. Initialize B-side DUART to local loop around mode.
2. Initialize DMAC for DMA operation through the DUART. Job is to transmit 4 bytes.
3. Request DMA job and delay for job completion.
4. Check B side of DUART for FIFO Full bit set in DUART Status register and confirm reception of 4 bytes.

Hardware Tested: The 9517 DMAC is tested.

Data Returned: The failing test number (last number displayed on the system console before failure), as well as error data presented are returned.

Notes: None

Phase #19 — Floppy Disk Interface Diagnostics

Phase Name:	Floppy Interface Control Test (fic_tst)
Type:	Interactive
Function:	This phase ensures that the Floppy Disk Drive and the Floppy Disk Controller are functioning.
Tests:	Tests 1 and 2 — verify that the Floppy Disk Controller can detect that the floppy disk drive can be enabled and disabled. Tests 3 through 5 — pattern test all Floppy Disk Controller registers that can be written to and read from. Test 6 — tests the Floppy Disk Controller register. Tests 7 and 8 — verify that the entire disk is formatted and the track numbers can be read. Tests 9 through 14 — test the ability to write to and read from a sector. Tests 15 through 17 — test the **step**, **stepin**, and **stepout** commands. Tests 18 and 19 — test the ability to write to and read from the opposite side.
Time:	16 seconds
Warnings:	The test code assumes a formatted double-sided, dual-density diskette. The test code uses the last cylinder as diagnostic cylinder (track 9, sides 1 and 2). The test code is configured using the International Standards Organization (ISO) Standard Dual-Density Format specification (512 bytes/sector, 5 sectors/track).
Notes:	None

Phase #19 Tests

Test Numbers: 1 and 2

Function: These tests enable and disable the floppy disk drive to determine if the Floppy Disk Controller can distinguish between the active and inactive drive states.

Procedure: Test 1 — disables the floppy disk drive by shutting down the floppy disk drive motor. The floppy disk drive Status register should indicate that the floppy disk drive is not ready.

Test 2 — enables the floppy disk drive by starting up the floppy disk drive motor. The floppy disk drive Status register should indicate that the floppy disk drive is ready.

Hardware Tested: The WD1793 Floppy Disk Controller/Formatter is tested.

Data Returned: The failing test number (last number displayed on the system console before failure) is returned.

Notes: None

==========

Test Numbers: 3 through 5

Function: These tests pattern test all Floppy Disk Controller registers that can be written to and read from.

Procedure: Test 3 — pattern tests (0x00,0x55,0xaa,0xff) the Data register.

Test 4 — pattern tests the Track register.

Test 5 — pattern tests the Sector register.

Hardware Tested: The WD1793 Floppy Disk Controller/Formatter is tested.

Data Returned: The failing test number (last number displayed on the system console before failure) is returned.
Supplemental Data: Raw data is the failing test pattern.

Notes: None

==========

Test Number: 6

Function: This test checks the Floppy Disk Controller register.

Procedure: Test patterns (0x00, 0x33, 0x55, 0xaa, 0xff) are written to the Floppy Disk Controller register.

Hardware Tested: The WD1793 Floppy Disk Controller/Formatter is tested.

Data Returned: The failing test number (last number displayed on the console before failure) is returned.
Supplement Data: Raw data is the failing test pattern.

Notes: None

Test Numbers: 7 and 8

Function: These tests verify that the entire disk is formatted and that all tracks can be read.

Procedure: The Identification (ID) field of each track is read, beginning with track 0 and proceeding to the end of each side, comparing the actual track number read against the expected track number.

Hardware Tested: The WD1793 Floppy Disk Controller/Formatter is tested.

Data Returned: The failing test number (last number displayed on the system console before failure) is returned.

Notes: None

==========

Test Numbers: 9 through 14

Function: These tests check the ability to write and read back a sector.

Procedure: Side 1, track 79, sector 1 is written with an increasing data pattern. The track is read back, and the data is compared. A complement data pattern is written to and read from the same area.

Hardware Tested: The WD1793 Floppy Disk Controller/Formatter is tested.

Data Returned: The failing test number (last number displayed on the system console before failure) is returned.

Notes: None

==========

Test Numbers: 15 through 17

Function: These tests check the **step**, **stepin**, and **stepout** commands and the auto update of the Track register feature.

Procedure: The test procedure is as follows:

1. The heads are moved to track 40, and the head movement is confirmed.
2. The **step** command is issued, and the head is confirmed to be located over track 41.
3. The **stepin** command is issued, and the head is confirmed to be located over track 42.
4. The **stepout** command is issued, and the head is confirmed to be located over track 41.

Hardware Tested: The WD1793 Floppy Disk Controller/Formatter is tested.

Data Returned: The failing test number (last number displayed on the system console before failure) is returned.

Notes: None

Test Numbers:	18 and 19
Function:	These tests check the ability of the Floppy Disk Controller to write to and read from opposite sides of a floppy disk.
Procedure:	The test procedure is as follows: 1. Write side 1, track 79, sector 1 with an increasing data pattern. 2. Write side 0, track 79, sector 1 with complement pattern. 3. Read side 1, track 79, sector 1 and compare the data. 4. Read side 0, track 79, sector 1 and compare the data.
Hardware Tested:	The WD1793 Floppy Disk Controller/Formatter is tested.
Data Returned:	The failing test number (last number displayed on the system console before failure) is returned.
Notes:	None

Phase #20 — Extended Floppy Disk Interface Diagnostics

Phase Name:	Extended Floppy Interface Control Test (efic_tst)
Type:	Interactive
Function:	This phase performs a floppy disk drive head position check.
Test:	Test 1 — checks the ability to write to and read from all sectors of the floppy disk.
Time:	15 minutes
Warnings:	The test code assumes a formatted double-sided, dual-density diskette. Further, the test code is configured using International Standards Organization (ISO) Standard Dual-Density Format specification.
Notes:	None

Phase #20 Test

Test Number:	1
Function:	This test writes and reads back all sectors of all tracks.
Procedure:	The test procedure is as follows: 1. Write then read track 0 of sector 1. 2. Write then read track 79 of sector 1. 3. Repeat Steps 1 and 2 for sectors 2 through 9. 4. Repeat Steps 1, 2, and 3 for tracks 1 and 78. 5. Repeat Steps 1, 2, and 3 for tracks 2 and 77. 6. Continue this pattern until all remaining tracks (3 through 76) have been tested. 7. Repeat Steps 1 through 6 for the other side of the floppy disk drive.
Hardware Tested:	The WD1793 Floppy Disk Controller/Formatter is tested.
Data Returned:	The failing test number (last number displayed on the system console before failure) and failing test pattern are returned.
Notes:	None

Phase #21 — Physical MMU Cache Diagnostics

Phase Name: MMU Physical Cache Control Program (pcache)

Type: Normal

Function: This phase checks physical Memory Management Unit (MMU) cache memory by performing the following tests:

Tests: Test 1 — checks that the double-bit memory faults work with MMU cache on.

Test 2 — checks all memory accesses on a word basis with various data patterns throughout cache memory range. The 00's followed by FF's followed by random patterns.

Test 3 — checks memory accesses by writing words and reading back two half words.

Test 4 — checks memory accesses by writing two half words and reading back 4 bytes.

Test 5 — checks memory accesses by writing and reading 4 bytes.

Test 6 — checks PCACHE tag fields.

Test 7 — checks cache entry invalidation (only in systems with two MMUs).

Time: 1 second per MMU

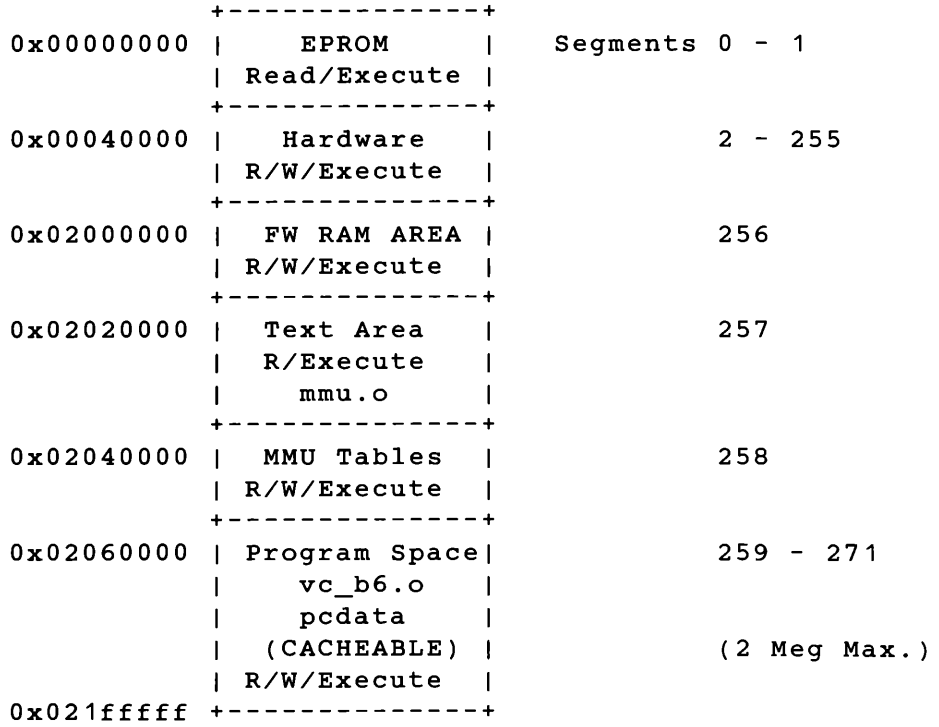
Warnings: None

Notes: Only a WE 32201, Version 2.3 or later supports caching. The WE 32201 is present only in 518B or later System Boards (SBDs). The MMUVR register has the version encoded in its two low-order hexadecimal digits. If caching is not supported by the system board, an appropriate message is displayed and the phase exits with status NTR (No Tests Run). Unsupported combinations of SBD and MMU are noted but do not cause failure.

All tests run once per MMU. On systems with two MMUs, the tests run twice. On other systems, the tests run only once (and then only if the data cache is supported).

This phase works by turning on the cache and performing accesses that are expected to hit the cache.

The MMU setup is as follows (1 to 1 mapping, Section 0 only):



==========

Phase #21 Tests

Test Number: 1

Function: This test ensures that a double-bit error can be detected in virtual mode with MMU cache enabled.

Procedure: The test inhibits Error Correction Coding (ECC), faults the contents of a word, and determines if a multiple-bit error occurred.

Hardware Tested: The Dynamic Memory Controller and MMU are tested.

Data Returned: The failing address, the actual data, and the expected data are returned.

Notes: None

==========

Test Number: 2

Function: This test checks all memory accesses on a word basis with alternate data patterns throughout memory range.

Procedure: Various patterns are written and then read.

Hardware Tested: The MMU Physical Cache (PCACHE) is tested.

Data Returned: The failing address, the actual data, and the expected data are returned.

Notes: None

Test Numbers:	3 through 5
Function:	These tests check memory accesses.
Procedure:	Test 3 — writes words and reads back as shorts. Increment address by 1000. Test 4 — writes two shorts and reads back as bytes. Increment address by 1000.
Hardware Tested:	The MMU PCACHE is tested.
Data Returned:	The failing address, the actual data, and the expected data are returned.
Notes:	None

==========

Test Number:	6
Function:	This test checks the tag field of the PCACHE.
Procedure:	Access every entry in the cache, change a Tag bit, and repeat.
Hardware Tested:	The MMU PCACHE is tested.
Data Returned:	The failing address, the actual data, and the expected data are returned.
Notes:	None

==========

Test Number:	7
Function:	This test checks the cache entry invalidation
Procedure:	Write a memory location through one MMU. Read it back. Write another value into the same location with the other MMU. Verify that the entry in the first MMU has changed.
Hardware Tested:	The MMU PCACHE is tested.
Data Returned:	The failing address, the actual data, and the expected data are returned.
Notes:	None

Phase #22 — Extended Physical MMU Cache Diagnostics

Phase Name:	MMU Extended Physical Cache Control Program (epcache)
Type:	Demand
Function:	This phase sets up Memory Management Unit (MMU), turns cache on, and executes benchmarks. For each test, if execution time with a cache on is less than a set percentage of the execution time with a cache off, that test passes.
Tests:	Test 1 — confirms that the ICACHE bit in the CPU PSW works. Test 2 — confirms that caching in MMU 0 works. Test 3 — confirms that caching in MMU 1 works. Test 4 — confirms that utilizing ICACHE bit and MMU 0 works. Test 5 — confirms that utilizing ICACHE bit and MMU 1 works.
Time:	25 seconds
Warnings:	None
Notes:	Only a WE 32201, Version 2.3 or later supports caching. The WE 32201 is present only in 518B or later System Boards (SBDs). The MMUVR register has the version encoded in its two low-order hexadecimal digits. If caching is not supported by the SBD, an appropriate message is displayed and the phase exits with status NTR (No Tests Run). After all tests run, the execution time of all benchmarks (in tenths-of-seconds and as a percent of the run time with all caches off) displays the following: FALCON MMU setup is as in PCACHE

Phase #22 Tests

Test Number:	1
Function:	This test confirms that the Instruction Cache (ICACHE) in CPU works.
Procedure:	Run the benchmark program without the ICACHE bit on. Then turn the ICACHE bit on and confirm that the benchmark program ran faster than with it off.
Hardware Tested:	The CPU ICACHE is tested.
Data Returned:	The failing test number is displayed on the system console.
Notes:	Execution time with the cache on must be less than or equal to 75 percent of execution time with the cache off.

==========

Test Numbers:	2 and 3
Function:	These tests confirm that the benchmark program executes faster when executing from the MMU Cache versus previous straight CPU time.
Procedure:	Turn the MMU cache on and execute the program. Confirm that the benchmark program executed quicker than when the CPU alone was used. Test 2 — checks the MMU 0. Test 3 — checks the MMU 1 (if present).
Hardware Tested:	The MMU Data Cache (for example, PCACHE) is tested.
Data Returned:	The failing test number is displayed on the system console.
Notes:	Execution time with the cache on must be less than or equal to 80 percent of execution time with the cache off.

==========

Test Numbers:	4 and 5
Function:	These tests confirm that the benchmark runs the quickest when both the CPUs ICACHE bit and the MMU cache are on.
Procedure:	Run the benchmark program and confirm that it ran faster than any of the previous execution times. Test 4 — checks the MMU 0. Test 5 — checks the MMU 1 (if present).
Hardware Tested:	The MMU Data Cache (for example, PCACHE) and the CPUs ICACHE bit are tested.
Data Returned:	The failing test number is displayed on the system console.
Notes:	Execution time with both caches on must be less than or equal to 69 percent of execution time with both caches off.

Chapter 4: Multiprocessor Diagnostics

Introduction

This chapter contains a description of the diagnostic phases and tests for the Multiprocessor Enhancement (MPE) card, which is referred to in software as the Multiprocessor Board (MPB). These phases and tests are applicable to the UNIX System V Release 3.2.2 and subsequent releases.

The following components are found on the System Board (SBD):

- WE 32200 Central Processing Unit (CPU)
- Math Accelerator Unit (MAU)
- Memory Management Unit (MMU)
- Interrupt Structure.

Twenty diagnostic phases run tests on all major board circuits. The Table of Contents listing will help you locate each phase and its associated tests. The phase and test descriptions are organized numerically in the same order that the phases are run on the 3B2 computer.

Phase Descriptions

Phase #1 — MPB Passive Memory Test Diagnostics

Phase Name: Random Access Memory (sram_tst)

Type: Normal

Function: This phase checks the memory by performing the following tests:

Tests: Test 1 — checks all of the memory on a word basis, checking for shorted address lines.

Memory is divided into thirds as follows:
Base test address - store 0x00L data pattern
Base test address + 1 - store 0x33L data pattern
Base test address + 2 - store 0xccL data pattern
Base test address + 3 - store 0x00L data pattern
: :: :: : ::: :: :: :: :: : :

Test 2 — checks all memory accesses on a word basis with alternate data patterns through memory range. A pattern of 55's followed by aa's.

Test 3 — checks memory accesses by writing words and reading back two half words.

Test 4 — checks memory accesses by writing two half words and reading back 4 bytes.

Test 5 — checks memory accesses by writing and reading 4 bytes.

Time: 2 seconds

Warnings: None

Notes: None

Phase #1 Tests

Test Number: 1

Function: This test checks all memory accesses on a word basis, checking for shorted address lines.

Procedure: Perform the following test algorithm:

Let Au be the memory address u.
0<=u<2**n where n is the bit size.
E - indicates an element exists in the set.
Let
T1 = {Aulu=0 (modulo 3)}
T2 = {Aulu=1 (modulo 3)}
T3 = {Aulu=2 (modulo 3)}.
Step 1: Write the all 0 word, W0, at all locations
Aj E T1 and Ak E T2.
Step 2: Write the all 1 word, W1, at all locations
Ai E T0.
Step 3: Read all locations Aj E T1
if output = W0 no fault indicated.
Step 4: Write the all 1 word, W1, at all locations
Aj E T1.
Step 5: Read all locations Ak E T2
if output = W0 no fault indicated.
Step 6: Read all locations Ai E T0 and Aj E T1
if output = W1 no fault indicated.
Step 7: Write and then read the all 0 word
at all locations Ai E T0
if output = W0 no fault indicated.
Step 8: Write and then read the all 1 word, W1,
at all locations Ak E T2
if output = W1 no fault indicated.

Hardware Tested: The MPB memory is tested.

Data Returned: The failing address, the actual data, and the expected data are returned.

Notes: None

Test Number:	2
Function:	This test checks all memory accesses on a word basis with alternate data patterns throughout memory range.
Procedure:	A pattern of 55's and aa's is written and then read.
Hardware Tested:	The MPB memory is tested.
Data Returned:	The failing address, the actual data, and the expected data are returned.
Notes:	None

==========

Test Numbers:	3 through 5
Function:	These tests check variable size accesses.
Procedure:	Test 3 — checks by writing words and reading back as shorts. Increment address by 200. Test 4 — checks by writing two shorts and reading back as bytes. Increment address by 400. Test 5 — checks by writing and read 4 bytes. Increment address by 800.
Hardware Tested:	The MPB memory is tested.
Data Returned:	The failing address, the actual data, and the expected data are returned.
Notes:	None

Phase #2 — MPB Microprocessor Unit Test #1 Diagnostics

Phase Name:	Central Processor Unit (bcpu32_1)
Type:	Normal
Function:	This phase checks the operation and instruction set of the WE 32X00 Microprocessor.
Tests:	Test 1 — checks the Condition Code register. Test 2 — checks the Microprocessor registers. Test 3 — checks the logical shift left instructions. Test 4 — checks the logical shift right instruction. Test 5 — checks the rotate instruction. Test 6 — checks the arithmetic shift left instructions. Test 7 — checks the arithmetic shift right instructions.
Time:	1 second
Warnings:	None
Notes:	None

Phase #2 Tests

Test Number:	1
Function:	This test checks the Condition Code register.
Procedure:	Test the register for the operation of the branch instructions using various states of the Condition Code register.
Hardware Tested:	The WE 32X00 Microprocessor is tested.
Data Returned:	The failing test number (last number displayed on the system console prior to failure) is returned.
Notes:	The following are the condition code assignments in the WE 32X00 Program Status Word.

```
 N  Z |V C
21 20 |19 18 X X |X X X X X |X X X X |X X X X |X X X X
```

Test Number: 2

Function: This test checks the integrity of the WE 32X00 registers using test patterns.

Procedure: Generate data test patterns that are moved through the registers to verify that each bit in each register can be set to a one and a zero and can be moved from register to register via the internal buses.

The Register Test is divided into two parts to allow the registers that contain such items as stack pointers to be restored at the completion of the test. The first part of the test will test registers 1 through 8. The second part of the test will test the FP, AP, SP, PCBP, and ISP registers.

Hardware Tested: The WE 32X00 Microprocessor is tested.

Data Returned: The failing test number (last number displayed on the system console prior to failure) is returned.

Notes: Patterns for the Register Tests are obtained by shifting until the N bit in the condition code is set.

WE 32X00 15 General Purpose Registers

Privilege Registers

R14 Interrupt Stack Pointer (ISP)
R13 Process Control Block Pointer (PCBP)
R11 Processor Status Word (PSW)

Noted Registers

R12 Stack Pointer (SP)
R10 Argument Pointer (AP)
R9 Frame Pointer (FP)

R8 - R0 General Registers

==========

Test Number: 3

Function: This tests checks the logical shift left instructions.

Procedure: Test all values of shift amount for word, half word, and byte.

Hardware Tested: The WE 32X00 Microprocessor is tested.

Data Returned: The failing test number (last number displayed on the system console prior to failure) is returned.

Notes: None

Test Number:	4
Function:	This test checks the logical shift right instructions.
Procedure:	Test all values of shift amount for word, half word, and byte.
Hardware Tested:	The WE 32X00 Microprocessor is tested.
Data Returned:	The failing test number (last number displayed on the system console prior to failure) is returned.
Notes:	None

==========

Test Number:	5
Function:	This test checks the rotate instruction.
Procedure:	Test for all values of rotate.
Hardware Tested:	The WE 32X00 Microprocessor is tested.
Data Returned:	The failing test number (last number displayed on the system console prior to failure) is returned.
Notes:	None

==========

Test Number:	6
Function:	This test checks the arithmetic shift left instructions.
Procedure:	Test all values of shift amount.
Hardware Tested:	The WE 32X00 Microprocessor is tested.
Data Returned:	The failing test number (last number displayed on the system console prior to failure) is returned.
Notes:	Arithmetic shift left is for word only, the test is done with the sign bit equal to both a one and a zero.

==========

Test Number:	7
Function:	This test checks the arithmetic shift right instructions.
Procedure:	Test all values of shift amount for word, half word, and byte.
Hardware Tested:	The WE 32X00 Microprocessor is tested.
Data Returned:	The failing test number (last number displayed on the system console prior to failure) is returned.
Notes:	None

Phase #3 — MPB Microprocessor Unit Test #2 Diagnostics

Phase Name:	Central Processor Unit (bcpu32_2)
Type:	Normal
Function:	This phase checks the operation and instruction set of the WE 32X00 Microprocessor.
Tests:	Test 1 — checks the logical dyadic instructions. Test 2 — checks the logical triadic instructions. Test 3 — checks the addition functions. Test 4 — checks the subtraction functions. Test 5 — checks the multiplication functions. Test 6 — checks the division functions. Test 7 — checks the modulo arithmetic functions.
Time:	1 second
Warnings:	None
Notes:	In general, most test pass/fail conditions are tested by collecting a hash value, which is a composite value obtained by the summing of all test results.

Phase #3 Tests

Test Number:	1
Function:	This test checks the logical dyadic instructions
Procedure:	Exercise the logical instructions OR, AND, and EXCLUSIVE OR.
Hardware Tested:	The WE 32X00 Microprocessor is tested.
Data Returned:	The failing test number (last number displayed on the system console prior to failure) is returned.
Notes:	The final pass/fail of this test is determined by a hash value that is collected by the correct execution of the instructions.

Test Number:	2
Function:	This test checks the logical triadic instructions.
Procedure:	Exercise the logical instructions OR, AND, and EXCLUSIVE OR.
Hardware Tested:	The WE 32X00 Microprocessor is tested.
Data Returned:	The failing test number (last number displayed on the system console prior to failure) is returned.
Notes:	The final pass/fail of this test is determine by a hash value that is collected by the correct execution of the instructions.

==========

Test Number:	3
Function:	This test checks the ability to perform addition.
Procedure:	The results of calculation will form a hash sum.
Hardware Tested:	The WE 32X00 Microprocessor is tested.
Data Returned:	The failing test number (last number displayed on the system console prior to failure) is returned.
Notes:	None

==========

Test Number:	4
Function:	This test checks the ability to perform subtraction.
Procedure:	The results of calculation will form a hash sum.
Hardware Tested:	The WE 32X00 Microprocessor is tested.
Data Returned:	The failing test number (last number displayed on the system console prior to failure) is returned.
Notes:	None

==========

Test Number:	5
Function:	This test checks the ability to perform multiplication.
Procedure:	The results of calculation will form a hash sum.
Hardware Tested:	The WE 32X00 Microprocessor is tested.
Data Returned:	The failing test number (last number displayed on the system console prior to failure) is returned.
Notes:	None

Test Number:	6
Function:	This test checks the ability to perform division.
Procedure:	The results of calculation will form a hash sum.
Hardware Tested:	The WE 32X00 Microprocessor is tested.
Data Returned:	The failing test number (last number displayed on the system console prior to failure) is returned.
Notes:	None

==========

Test Number:	7
Function:	This test checks the ability to perform modulo arithmetic.
Procedure:	The results of calculation will form a hash sum.
Hardware Tested:	The WE 32X00 Microprocessor is tested.
Data Returned:	The failing test number (last number displayed on the system console prior to failure) is returned.
Notes:	None

Phase #4 — MPB Microprocessor Unit Test #3 Diagnostics

Phase Name:	Central Processor Unit (bcpu32_3)
Type:	Normal
Function:	This phase tests the operation and instruction set of the microprocessor.
Tests:	Test 1 — checks the extract field instruction. Test 2 — checks the insert field instruction. Test 3 — checks the move block instruction. Test 4 — checks the swap interlocked instruction. Test 5 — checks the Test bit instruction. Test 6 — checks the Save and Restore register instructions. Test 7 — checks the call procedure and RET instruction. Test 8 — checks the test sign extension types. Test 9 through 16 — checks CPU addressing modes.
Time:	1 second
Warnings:	None
Notes:	None

Phase #4 Tests

Test Number:	1
Function:	This test checks the extract field instruction
Procedure:	Perform the instruction for all widths and offsets.
Hardware Tested:	The WE 32X00 Microprocessor is tested.
Data Returned:	The failing test number (last number displayed on the system console prior to failure) is returned.
Notes:	The results of the test are confirmed by comparison to a generated hash sum which contains all of the intermediate results. Register r7 contains the hash value that is generated by adds and rotates.

Test Number:	2
Function:	This test checks the insert field instruction.
Procedure:	Perform the Instruction for all widths and offsets.
Hardware Tested:	The WE 32X00 Microprocessor is tested.
Data Returned:	The failing test number (last number displayed on the system console prior to failure) is returned.
Notes:	The results of the test are confirmed by comparison to a generated hash sum which contains all of the intermediate results. Register r7 contains the hash value that is generated by adds and rotates.

==========

Test Number:	3
Function:	This test checks the move block instruction.
Procedure:	A block of 16 words are moved from location 0x0 to Random Access Memory (RAM) scratch area. After the block move, the locations are compared by physical addressing.
Hardware Tested:	The WE 32X00 Microprocessor is tested.
Data Returned:	The failing test number (last number displayed on the system console prior to failure) is returned.
Notes:	The following instruction sequence is used in this test:

```
while( r2 < 0)
{
*r1 = r0;
--r2;
r0++;
r1++;
}
```

==========

Test Number:	4
Function:	This test checks the swap interlocked instruction.
Procedure:	Tests for word, half word and byte swap are performed.
Hardware Tested:	The WE 32X00 Microprocessor is tested.
Data Returned:	The failing test number (last number displayed on the system console prior to failure) is returned.
Notes:	The global locations used in this test are used only because they are available from other tests and are conveniently located in RAM.

Test Number:	5
Function:	This test checks the bit test instruction for word, half word and byte.
Procedure:	Exercise the instruction and compile a hash value based on the results in the condition code portion of the Processor Status register.
Hardware Tested:	The WE 32X00 Microprocessor is tested.
Data Returned:	The failing test number (last number displayed on the system console prior to failure) is returned.
Notes:	The results of the test are confirmed by comparison to a generated hash sum which contains all the intermediate results. Register r6 contains the hash value that is generated by adds and rotates.

==========

Test Number:	6
Function:	This test checks the Save and Restore register instructions.
Procedure:	Load registers r3 through r6 with a unique pattern that will be used to by the save instruction to load the stack. Subsequent restore commands will verify the stack to be correctly loaded and unloaded.
Hardware Tested:	The WE 32X00 Microprocessor is tested.
Data Returned:	The failing test number (last number displayed on the system console prior to failure) is returned.
Notes:	The pass/fail condition of this test is determined by a hash value that is generated by the addition of registers r3 through r6 after the restore Instruction is executed.

==========

Test Number:	7
Function:	This test checks the call procedure and RET instructions.
Procedure:	Set up a small subroutine call that returns a value in r0 as a test that the call procedure and RET instructions function properly.
Hardware Tested:	The WE 32X00 Microprocessor is tested.
Data Returned:	The failing test number (last number displayed on the system console prior to failure) is returned.
Notes:	None

Test Number: 8

Function: This test checks the test sign extension types.

Procedure: Perform the following tests:

	Byte	Half word	Word
Unsigned byte (-1) to	-	0xff	0xff
Unsigned half word to	-	-	0x0000ffff
Signed byte (-1) to	-	-1	-1
Signed half word to	-	-	-1

Hardware Tested: The WE 32X00 Microprocessor is tested.

Data Returned: The failing test number (last number displayed on the system console prior to failure) is returned.

Notes: None

==========

Tests Numbers: 9 through 16

Function: These tests check the literal, immediate, absolute deferred, absolute, AP short offset, FP short offset, register deferred and [HB] displacement addressing modes.

Procedure: Using the global variables, test the assorted addressing modes of the CPU.

Hardware Tested: The WE 32X00 Microprocessor is tested.

Data Returned: The failing test number (last number displayed on the system console prior to failure) is returned.

Notes: None

Phase #5 — MPB Microprocessor Unit Test #4 Diagnostics

Phase Name:	Central Processor Unit (bcpu32_4)
Type:	Normal
Function:	This phase tests the operation and instruction set of the microprocessor.
Tests:	Test 1 — checks the PUSHAW, PUSHW and POPW instructions Test 2 — checks the unconditional jump instruction. Test 3 — checks the branch to subroutine and return from subroutine instructions. Test 4 — checks the jump to subroutine and return from subroutine instructions. Test 5 — checks for exception via divide-by-zero instruction. Tests 6 through 8 — check for exception when the three privileged registers (ISP, PCBP, PSW) are accessed in user execution level. Test 9 and 10 — check for exception when the PSW is accessed at any execution level except kernel. Test 11 — checks for correct exception processing during a gate instruction. Test 12 — checks that when operating at the user execution level, the appropriate exception is generated when the privileged gate instruction is attempted.
Time:	1 second
Warnings:	None
Notes:	A failure in Tests 6 through 12 could leave the processor in a nonkernel execution level. Any access at that point to privileged registers or privileged instructions could cause an unexpected exception.

Phase #5 Tests

Test Number: 1

Function: This test verifies the PUSHAW, PUSHW and POP instructions.

Procedure: The instructions are performed.

Hardware Tested: The WE 32X00 Microprocessor is tested.

Data Returned: The failing test number (last number displayed on the system console prior to failure) is returned.

Notes: None

==========

Test Number: 2

Function: This test verifies the unconditional jump instruction.

Procedure: Perform a jump instruction. If the jump is not accomplished, the next instruction will indicate an error.

Hardware Tested: The WE 32X00 Microprocessor is tested.

Data Returned: The failing test number (last number displayed on the system console prior to failure) is returned.

Notes: None

==========

Test Number: 3

Function: This test checks the branch to subroutine and return from subroutine instructions.

Procedure: Set up a small subroutine that is entered and that returns on a conditional value of zero.

Hardware Tested: The WE 32X00 Microprocessor is tested.

Data Returned: The failing test number (last number displayed on the system console prior to failure) is returned.

Notes: None

==========

Test Number: 4

Function: This test checks the jump to subroutine and return from subroutine instructions.

Procedure: Setup a small subroutine that is entered and that returns on a conditional value of zero.

Hardware Tested: The WE 32X00 Microprocessor is tested.

Data Returned: The failing test number (last number displayed on the system console prior to failure) is returned.

Notes: This test is very similar to Test 3 except it exercises the jump rather than branch to subroutine instruction and uses a different return condition.

Test Number:	5
Function:	This test ensures that an exception occurs when the instruction calls for a divide-by-zero.
Procedure:	Divide by zero and verify that an exception is generated.
Hardware Tested:	The WE 32X00 Microprocessor is tested.
Data Returned:	The failing test number (last number displayed on the system console prior to failure) is returned.
Notes:	A failure of this test indicates a divide-by-zero instruction did not cause an exception. Also note that it is this exception that puts the test into kernel level execution level.

==========

Tests Numbers:	6 through 8
Function:	These tests ensures that an exception is generated when the multiprocessor is at the user execution level and access is attempted to Tests 6, 7, and 8. Test 6 — Interrupt Stack Pointer (ISP). Test 7 — Process Control Block Pointer (PCBP). Test 8 — Process Status Word (PSW).
Procedure:	Puts the microprocessor into user execution level and attempts to access each of the above registers. Via the exception flag, confirm an exception was generated.
Hardware Tested:	The WE 32X00 Microprocessor is tested.
Data Returned:	The failing test number (last number displayed on the system console prior to failure) is returned.
Notes:	None

==========

Tests Numbers:	9 and 10
Function:	These tests confirm that an exception is generated when access to the Process Status Word (PSW) register is attempted at any level other than kernel.
Procedure:	Put the CPU into the (9) executive or (10) supervisor execution level and attempt to access the PSW register.
Hardware Tested:	The WE 32X00 Microprocessor is tested.
Data Returned:	The failing test number (last number displayed on the system console prior to failure) is returned.
Notes:	None

Test Number:	11
Function:	This test confirms that an exception is generated when the gate instruction is executed.
Procedure:	Perform the gate instruction and confirm an exception was generated via the bpe_flg variable.
Hardware Tested:	The WE 32X00 Microprocessor is tested.
Data Returned:	The failing test number (last number displayed on the system console prior to failure) is returned.
Notes:	None

==========

Test Number:	12
Function:	This test confirms that an exception is generated operating at the user execution level and a privileged gate instruction is attempted.
Procedure:	Put the microprocessor in the user execution level and attempt to perform the gate instruction. The variable 'bpepsw' will indicate if a 'stack exception' (error) or 'normal exception' (correct) was processed.
Hardware Tested:	The WE 32X00 Microprocessor is tested.
Data Returned:	The failing test number (last number displayed on the system console prior to failure) is returned.
Notes:	None

Phase #6 — MPB Math Accelerator Unit Test #1 Diagnostics

Phase Name:	Math Accelerator Unit (MAU) (bmau1_tst)
Type:	Normal
Function:	This phase tests the presence of a MAU device and, if found, pattern tests the MAU Data bus, operand registers F0 through F3 and the MAU Data register.
Tests:	Test 1 — pattern checks the MAU Data bus via walking a one through a field of zeros as a single precision operand to register F0. Tests 2 through 5 — pattern check the MAU registers F0 through F3 as a double-extended precision operand with a modified data pattern of 0x55's and 0xaa's. Test 6 — pattern checks the MAU Data register as a double-extended precision operand with a modified data pattern of 0x55's and 0xaa's.
Time:	1 second
Warnings:	None
Notes:	All operand pattern tests are checked for register-to-register cross talk.

Phase #6 Tests

Test Number:	1
Function:	This test pattern checks the MAU Data bus.
Procedure:	Walk a one through a field of zeros as a single precision operand by moving the operand to and from register F0.
Hardware Tested:	The WE 32X06 MAU is tested.
Data Returned:	The failing test number (last number displayed on the system console prior to failure) is returned.
Notes:	None

Test Numbers:	2 through 5
Function:	These tests pattern check the MAU registers F0 through F3.
Procedure:	Pattern test each register with a modified 0x55, 0xaa, checking the other registers for cross talk.
Hardware Tested:	The WE 32X06 MAU is tested.
Data Returned:	The failing test number (last number displayed on the system console prior to failure) is returned.
Notes:	None

==========

Test Number:	6
Function:	This test pattern checks the MAU Data register.
Procedure:	Pattern tests the Data register with a modified data pattern of 0x55's and 0xaa's.
Hardware Tested:	The WE 32X06 MAU is tested.
Data Returned:	The failing test number (last number displayed on the system console prior to failure) is returned.
Notes:	None

Phase #7 — MPB Math Accelerator Unit Test #2 Diagnostics

Phase Name: Math Accelerator Unit (MAU) (bmau2_tst)

Type: Normal

Function: This phase tests that a faulted MAU can be detected. Further, checks all of the "Sticky" and Mask bits of the MAU Auxiliary Status Register (ASR) for the ability to be set, cleared, and masked

Tests: Test 1 — causes a MAU fault condition by dividing by zero, testing the ASR divide-by-zero "Sticky" (QS) and Mask bits (QM).

Test 2 — checks the ASR Invalid Operation "Sticky" (IS) and Mask bits (IM).

Test 3 — checks the ASR Overflow. "Sticky" (OS) and Mask bits (OM).

Test 4 — checks the ASR Inexact "Sticky" (PS) and Mask bits (PM).

Test 5 — checks the ASR Underflow "Sticky" (US) and Mask bits (UM).

Test 6 — checks the ASR Integer Overflow (IO) Indicator bit.

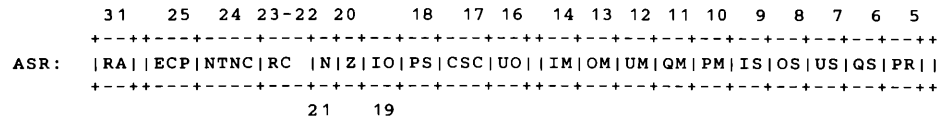
Test 7 — checks the ASR Unordered (UO) Indicator bit.

Test 8 — checks the ASR Nontrapping Not a Number (NAN) Control (NTNC) bit.

Time: 1 second

Warnings: None

Notes: The Negative (N), Zero (Z), Result Available (RA), and Partial Remainder (PR) bits are tested during Phase 8.



Phase #7 Tests

Test Number: 1

Function: This test checks that a faulted MAU can be detected.

Procedure: Perform a divide-by-zero and confirm exception was generated. Then inhibit the divide-by-zero fault and repeat the process. Confirm that an exception was not taken. In both cases, check the MAU_ASR to ensure the correct flags were set.

Hardware Tested: The WE 32X06 MAU is tested.

Data Returned: The failing test number (last number displayed on the system console prior to failure) is returned.

Notes: None

==========

Test Number: 2

Function: This test checks the ASR Invalid Operation "Sticky"(IS) and Mask bits (IM).

Procedure: Perform an Integer to Float Conversion (ITOF) with the source specified as MAU register F0. Confirm that an exception was generated. Then inhibit Invalid Operation exceptions and repeat the procedure. Confirm that an exception was not generated. In both cases, confirm that the correct MAU_ASR flags were set.

Hardware Tested: The WE 32X06 MAU is tested.

Data Returned: The failing test number (last number displayed on the system console prior to failure) is returned.

Notes: None

==========

Test Number: 3

Function: This test checks the ASR Overflow "Sticky"(OS) and Mask bits (OM).

Procedure: Force an Overflow exception using the MAU division instruction. Confirm that an exception was taken. Then inhibit Overflow exceptions and repeat the procedure. Confirm that an exception was not taken. In both scenarios, check the MAU_ASR for correct flag generation.

Hardware Tested: The WE 32X06 MAU is tested.

Data Returned: The failing test number (last number displayed on the system console prior to failure) is returned.

Notes: None

Test Number:	4
Function:	This test checks the Inexact "Sticky"(PS) and Mask bits (PM).
Procedure:	Cause an Inexact exception using the MAU RTOI instruction. Confirm that an exception was generated. Then block Inexact exceptions and repeat the procedure. Confirm that an exception was not taken. In both cases, check the MAU_ASR for correct flag values.
Hardware Tested:	The WE 32X06 MAU is tested.
Data Returned:	The failing test number (last number displayed on the system console prior to failure) is returned.
Notes:	None

==========

Test Number:	5
Function:	This test checks the ASR Underflow "Sticky"(US) and Mask bits (UM).
Procedure:	Cause a MAU Underflow exception using the MAU division instruction. Confirm that an exception was generated. Then inhibit Underflow exceptions and repeat the procedure. Check that an exception was not generated. In both tests, confirm the correct setting of MAU_ASR flag variables.
Hardware Tested:	The WE 32X06 MAU is tested.
Data Returned:	The failing test number (last number displayed on the system console prior to failure) is returned.
Notes:	None

==========

Test Number:	6
Function:	This test checks the ASR Integer Overflow (IO) Indicator bit.
Procedure:	Check that an exception is generated using the MAU FTOI instruction. Confirm that an exception was taken and that the correct MAU_ASR flags were set.
Hardware Tested:	The WE 32X06 MAU is tested.
Data Returned:	The failing test number (last number displayed on the system console prior to failure) is returned.
Notes:	None

Test Number:	7
Function:	This test checks the ASR Unordered (UO) Indicator bit.
Procedure:	Force an Unordered exception using the MAU CMP instruction. Check that an exception was taken and that the MAU_ASR set the correct flag values.
Hardware Tested:	The WE 32X06 MAU is tested.
Data Returned:	The failing test number (last number displayed on the system console prior to failure) is returned.
Notes:	None

==========

Test Number:	8
Function:	This test checks the ASR NTNC bit.
Procedure:	Cause an Invalid Operation condition with the NTNC bit set by performing an ITOF with the source specified as a MAU register. Confirm that an exception was generated and that the MAU_ASR has the correct flags set.
Hardware Tested:	The WE 32X06 MAU is tested.
Data Returned:	The failing test number (last number displayed on the system console prior to failure) is returned.
Notes:	None

Phase #8 — MPB Math Accelerator Unit Test #3 Diagnostics

Phase Name:	Math Accelerator Unit (MAU) (bmau3_tst)
Type:	Normal
Function:	This phase tests all of the possible MAU operation codes.
Tests:	Tests 1 through 8 — check the CMP, CMPS, CMPE, and CMPES operation codes. Tests 9 and 10 — check the add and subtract operation codes. Tests 11 through 13 — check the multiply, divide, and REM operation codes. Tests 14 — checks the NEG operation code. Tests 15 — checks the ABS operation code. Tests 16 — checks the SQRT operation code. Tests 17 through 21 — check the RTOI, ITOF, DTOF, FTOD and FTOI operations.
Time:	1 second
Warnings:	None
Notes:	The Negative (N), Zero (Z), Result Available (RA), and Partial Remainder (PR) bits are tested during this test phase.

Phase #8 Tests

Tests Numbers:	1 through 8
Function:	These tests check the CMP, CMPS, CMPE, and CMPES operation codes.
Procedure:	Set operand1 and operand2 to 123.0. Run each compare instruction checking for equality. Set operand1 to 100.0. Run each compare instruction checking for inequality.
Hardware Tested:	The WE 32X06 MAU is tested.
Data Returned:	The failing test number (last number displayed on the system console prior to failure) is returned.
Notes:	None

Test Numbers: 9 and 10

Function: These tests check the add and subtract operation codes. Many of the following tests (9 through 21) use the MAU compare instruction to determine if the test passed.

Procedure:
Set operand1 to 39.0 and operand2 to 12.0.
Move operand2 to MAU F0, operand1 to MAU F1.
Perform an Add.
Set expctd_rslt to 51.0.
Perform MAU F0 + MAU F1 -> result.
Perform a Subtract.
Set expctd_rslt to 27.0.
Perform MAU F1 - MAU F0 -> result.
Move result to MAU F0, expctd_rslt to MAU F1.
Compare MAU F0 with MAU F1 and check for equality.

Hardware Tested: The WE 32X06 MAU is tested.

Data Returned: The failing test number (last number displayed on the system console prior to failure) is returned.

Notes: None

==========

Tests Numbers: 11 through 13

Function: These tests check the multiply, divide, and REM operations codes.

Procedure:
Clear the result variable
Perform a Multiply.
Set operand1 to 239.0, operand2 to 11.0.
Set expctd_rslt to 2629.0.
Move operand1 to MAU F1 and operand2 to MAU F0.
Perform MAU F0 * MAU F1 -> result.
Perform a Divide.
Set operand1 to 125.0, operand2 to 5.0.
Set expctd_rslt to 25.0.
Move operand1 to MAU F1 and operand2 to MAU F0.
Perform MAU F1 / MAU F0 -> result.
Perform a REM.
Set operand2 to 10.0.
Set expctd_rslt to 5.0.
Move operand1 to MAU F1 and operand2 to MAU F0.
Perform MAU F1 % MAU F0 -> result.
Move result to MAU F0, expctd_rslt to MAU F1.
Compare MAU F0 and MAU F1 and check for equality.

Hardware Tested: The WE 32X06 MAU is tested.

Data Returned: The failing test number (last number displayed on the system console prior to failure) is returned.

Notes: None

Test Number:	14
Function:	This test checks the Negative (NEG) operation code.
Procedure:	Set operand1 to 239.0 and expctd_rslt to -239.0. Move operand1 to MAU F0. Perform NEG MAU F0. Compare MAU F0 and expctd_rslt and check for equality.
Hardware Tested:	The WE 32X06 MAU is tested.
Data Returned:	The failing test number (last number displayed on the system console prior to failure) is returned.
Notes:	None

==========

Test Number:	15
Function:	This test checks the Absolute (ABS) operation code.
Procedure:	Set operand1 to -239.0 and expctd_rslt to 239.0. Move operand1 to MAU F0. Perform ABS MAU F0. Compare MAU F0 and expctd_rslt and check for equality.
Hardware Tested:	The WE 32X06 MAU is tested.
Data Returned:	The failing test number (last number displayed on the system console prior to failure) is returned.
Notes:	None

Test Number: 16

Function: This test checks the Square Root (SQRT) operation code.

Procedure: Set operand1 to 144.0 and expctd_rslt to 12.0.
Move operand1 to MAU F0.
Perform SQRT MAU F0.
Compare MAU F0 and expctd_rslt and check for equality.

Hardware Tested: The WE 32X06 MAU is tested.

Data Returned: The failing test number (last number displayed on the system console prior to failure) is returned.

Notes: None

==========

Tests Numbers: 17 through 21

Function: These tests check the RTOI, ITOF, DTOF, FTOD and FTOI operations.

Procedure: RTOI: Set operand1 to 123.45 and expctd_rslt to 123.0.
Perform RTOI operand1 -> MAU F0.
Compare expctd_rslt and MAU F0 checking for equality.

ITOF: Set operand1 to 123.
Perform ITOR operand1 -> MAU F0.
Compare expctd_rslt and MAU F0 checking for equality.

FTOI: Set operand1 to 123.05, expctd_rslt to 123.
Clear result.
Move operand1 to MAU F0.
Perform FTOI MAU F0 -> result.
Compare result and expctd_rslt checking for equality.

DTOF: Set operand1 to 0x123a, expctd_rslt to 123.0.
Perform DTOF operand1 -> MAU F0.
Compare expctd_rslt and MAU F0 checking for equality.

FTOD: Set operand1 to 123.00, expctd_rslt to 0x123a.
Move operand1 to MAU F0.
Perform FTOD MAU F0 -> result.
Compare result and expctd_rslt checking for equality.

Hardware Tested: The WE 32X06 MAU is tested.

Data Returned: The failing test number (last number displayed on the system console prior to failure) is returned.

Notes: None

Phase #9 — MPB Memory Management Unit Test #1 Diagnostics

Phase Name:	Memory Management Unit Tests (bmmu1_tst)
Type:	Normal
Function:	This phase checks the internal registers and descriptor caches of the Memory Management Unit (MMU).
Tests:	Test 1 — checks Virtual Address register. Test 2 and 3 — check the SDCs. Test 4 and 5 — check the RHPDC. Test 6 and 7 — check the LHPDC. Test 8 — checks SRAMA. Test 9 — checks SRAMB. Test 10 through 11 — check the Fault Address register and Configuration register.
Time:	1 second
Warnings:	None
Notes:	None

Phase #9 Tests

Test Number:	1
Function:	This test checks the Virtual Address register.
Procedure:	The read/write capabilities of the internal entities with full binary bit patterns.
Hardware Tested:	The Virtual Address register is tested.
Data Returned:	The failing test number, the failing test pattern, and the actual test pattern read are returned.
Notes:	None

Tests Numbers: 2 and 3

Function: These tests check the read/write capability of the Segment Descriptor Caches (SDCs).

Procedure: A pattern test on each of the 32 segment descriptor locations are performed.

Test 2 — checks the lower half of SDC area.

Test 3 — checks the upper half of SDC area.

Hardware Tested: The SDCs are tested.

Data Returned: The failing test number, the failing test pattern, actual test pattern read, and the failing SDC address are returned.

Notes: None

==========

Tests Numbers: 4 and 5

Function: These tests check the read/write capability of the Left Hand Page Descriptor Cache (LHPDC) of the WE 32101 Microprocessor only.

Procedure: Four binary bit patterns are written and read back for comparison.

Test 4 — checks the lower half of LHPDC area.

Test 5 — checks the upper half of LHPDC area.

Hardware Tested: The LHPDC is tested.

Data Returned: The failing test number, the failing test pattern, the actual test pattern read, and the failing LHPDC address are returned.

Notes: This test is only executed on the WE 32101 Microprocessor because the WE 32201 Microprocessor does not have LHPDC areas.

==========

Tests Numbers: 6 and 7

Function: These tests check the read/write capability of the Right Hand Page Descriptor Cache (RHPDC) for the WE 32101 Microprocessor or the entire Page Descriptor Cache (PDC) for the WE 32201 Microprocessor.

Procedure: Four binary bit patterns are written and read back. Compare each binary bit pattern.

Hardware Tested: The PDC is tested.

Data Returned: The failing test number, the failing test pattern, the actual test pattern read, and the failing RHPDC address are returned.

Notes:

Test	Dependencies	Function
6	WE 32101	Lower Right Hand of the PDC area
7	WE 32101	Upper Right Hand of the PDC area
6	WE 32201	Lower PDC area (1)
7	WE 32201	Upper PDC area (1)

The WE 32201 Microprocessor does not contain LHPDCs and RHPDCs.

Test Number:	8
Function:	This test checks the read/write capability of Section Random Access Memory A (SRAMA).
Procedure:	Four binary patterns are written and read back for comparison.
Hardware Tested:	The SRAMA is tested.
Data Returned:	The failing test number, the failing test pattern, the actual test pattern read, and the failing SRAMA address are returned.
Notes:	None

==========

Test Number:	9
Function:	This test checks the read/write capability of Section Random Access Memory B (SRAMB).
Procedure:	Four binary patterns are written and read back for comparison.
Hardware Tested:	The SRAMB is tested.
Data Returned:	The failing test number, the failing test pattern, the actual test pattern read, and the failing SRAMB address are returned.
Notes:	None

==========

Tests Numbers:	10 and 11
Function:	These tests perform checks on the Fault Address and Configuration registers.
Procedure:	Each register is written with four binary patterns, and a comparison is made.
Hardware Tested:	The Fault Address register and the Configuration register are tested.
Data Returned:	The failing test number, the failing test pattern, the actual test pattern read, and the failing register address are returned.
Notes:	None

Phase #10 — MPB Memory Management Unit Test #2 Diagnostics

Phase Name:	Memory Management Unit Tests (bmmu2_tst)
Type:	Normal
Function:	This phase tests the flushing capability of the Memory Management Unit (MMU).
Tests:	Test 1 — checks the SDC single entry flush. Tests 2 and 3 — check the LPDC and RPDC single entry flush. Test 3 — checks the RPDC single entry flush. Test 4 — checks the Section 0 flush. Test 5 — checks the Section 1 flush. Test 6 — checks the Section 2 flush. Test 7 — checks the Section 3 flush.
Time:	1 second
Warnings:	None
Notes:	None

Phase #10 Tests

Test Number:	1
Function:	This test checks the single entry flushing capability of the Segment Descriptor Caches (SDCs).
Procedure:	Initialize cache by writing to Section Random Access Memory A (SRAMA) to flush them. Set up the entries that are to be flushed by setting the G bits and specific tags in them. Flush the entries by setting up the correct unique tags and indexing into the cache to clear the G bits. Make sure the entries are flushed by reading all entries in the cache to make sure that no entry is marked good.
Hardware Tested:	The MMU is tested.
Data Returned:	The failing test number (last number displayed on the system console prior to failure) is returned.
Notes:	None

Tests Numbers: 2 and 3

Function: These tests check the single entry flushing capability of the Page Descriptor Caches (PDCs).

Procedure: Initialize cache by writing to SRAMA to flush them. Set up the entries that are to be flushed by setting the G bits and specific tags in them. Flush the entries by setting up the correct unique tags and indexing into the cache to clear the G bits. Make sure the entries are flushed by reading all entries in the cache to make sure that no entry is marked good.

Hardware Tested: The MMU chip is tested.

Data Returned: The failing test number (last number displayed on the system console prior to failure) is returned.

Notes:

Test	Dependencies	Function
2	WE 32101	LHPDC
3	WE 32101	RHPDC
2	WE 32201	No tests run
3	WE 32201	No tests run

==========

Tests Numbers: 4 through 7

Function: These tests perform a Section flush on Sections 0 through 3.

Procedure: Set all G bits in Sections 0 through 3. Write to SRAMA to clear them. Check the G bit for correct action.

Hardware Tested: The MMU chip is tested.

Data Returned: The failing test number (last number displayed on the system console prior to failure) is returned.

Notes:

Tests	Dependencies	Function
4-7	WE 32101	SDC
		High Left PDC entry
		High Right PDC entry
4-7	WE 32201	SDC
		High PDC entry

Phase #11 — MPB Memory Management Unit Test #3 Diagnostics

Phase Name: Memory Management Unit (MMU) (bmmu3_tst)

Type: Normal

Function: This phase tests translation capability and the R&M update.

Tests: Test 1 — checks the paged segment address translation.

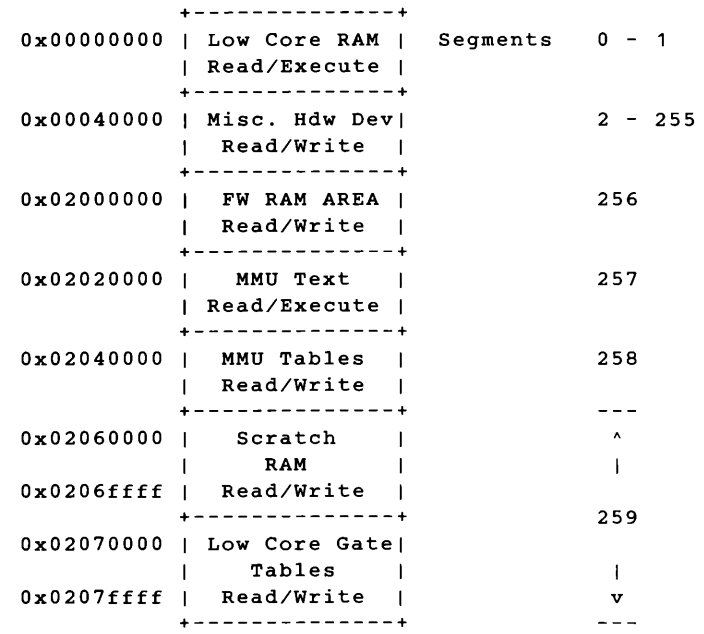
Test 2 — checks the contiguous segment address translation.

Tests 3 through 18 — perform the R&M updating of the MMU.

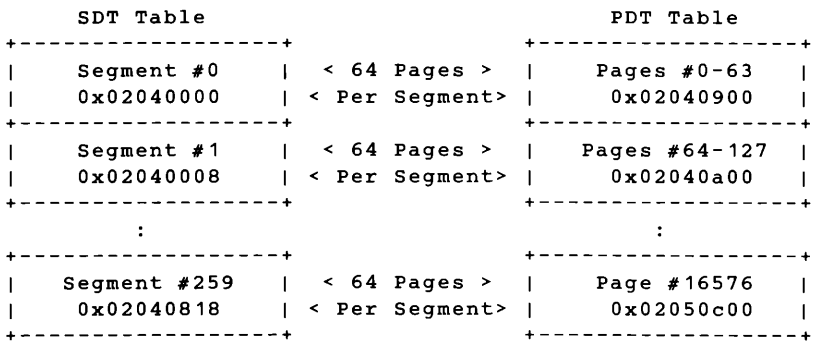
Time: 3 seconds

Warnings: None

Notes: The MMU test mapping scheme is as follows:



The Memory Management Tables in memory are set as follows:



Phase #11 Tests

Tests Numbers:	1 and 2
Function:	These tests check the translation of paged and contiguous segments.
Procedure:	Set up page descriptor tables and segment descriptor tables. Set up Section Random Access Memory A (SRAMA) and SRAMB. Write data pattern to CPU RAM (0x020600xx) in virtual mode. Read data back in physical mode to confirm that translation occurred correctly.
Hardware Tested:	The MMU chip is tested.
Data Returned:	The failing test number, the expected data, and the actual data read are returned.
Notes:	None

==========

Test Numbers:	3 through 18
Function:	These tests check the R&M updating of the MMU.
Procedure:	Set up tables, reference a segment, write to a segment, and read a signature.
Hardware Tested:	The MMU chip is tested.
Data Returned:	The failing test number (last number displayed on the system console prior to failure) and the failing MMU Section (0 through 3) are returned.
Notes:	None

Phase #12 — MPB Memory Management Unit Test #4 Diagnostics

Phase Name:	Memory Management Unit (bmmu4_tst)
Type:	Normal
Function:	This phase tests the fault recognition ability of the Memory Management Unit (MMU).
Tests:	Test 1 — forces an SDT length fault. Test 2 — forces a page write fault. Test 3 — forces an invalid segment descriptor fault Test 4 — forces a segment not present fault. Test 5 — forces an object trap fault. Test 6 — forces too many indirections fault. Test 7 — forces segment offset fault. Tests 8 through 31 — force access fault. Test 32 — forces access and segment offset fault. Test 33 — forces PDT length fault. Test 34 — forces PDT not present fault. Test 35 — forces page not present fault. Test 36 — forces miss process memory fault. Test 37 — forces R&M update memory fault. Test 38 — forces double page hit fault.
Time:	3 seconds
Warnings:	None
Notes:	None

Phase #12 Tests

Test Number:	1
Function:	This test checks the Segment Descriptor Table (SDT) length fault.
Procedure:	Reference a virtual address that has a segment select field greater than the SDT length field of Section Random Access Memory B (SRAMB).
Hardware Tested:	The MMU chip is tested.
Data Returned:	The failing test number, the MMU fault code, and the Fault Address register are returned.
Notes:	None

==========

Test Number:	2
Function:	This test checks the page write fault.
Procedure:	Set the W bit in an Page Descriptor Cache (PDC) entry. Write to the virtual address that corresponds to this PDC entry.
Hardware Tested:	The MMU chip is tested.
Data Returned:	Raw Data: The value in the MMU Fault Code register is returned. Supplemental Data: The value in the MMU Fault Address register is returned.
Notes:	None

==========

Test Number:	3
Function:	This test checks the invalid segment descriptor fault.
Procedure:	Set up the descriptor tables for contiguous segments. Flush the caches. Clear the V bit in the SDT entry. Reference the virtual address that corresponds to this entry.
Hardware Tested:	The MMU chip is tested.
Data Returned:	Raw Data: The value in the MMU Fault Code register is returned. Supplemental Data: The value in the MMU Fault Address register is returned.
Notes:	None

Test Number: 4

Function: This test checks the segment not present fault.

Procedure: Set up the descriptor tables for contiguous segments. Flush the caches. Clear the P bit in an SDT entry. Reference the virtual address that corresponds to this entry.

Hardware Tested: The MMU chip is tested.

Data Returned: Raw Data: The value in the MMU Fault Code register is returned.
Supplemental Data: The value in the MMU Fault Address register is returned.

Notes: None

==========

Test Number: 5

Function: This test checks the object trap fault.

Procedure: Set up the descriptor tables for contiguous segments. Flush the caches. Set the T bit in an SDT entry. Reference the virtual address that corresponds to this entry.

Hardware Tested: The MMU chip is tested.

Data Returned: Raw Data: The value in the MMU Fault Code register is returned.
Supplemental Data: The value in the MMU Fault Address register is returned.

Notes:

Test	Dependencies	Function
5	WE 32101	Test object trap fault
5	WE 32201	No test run

==========

Test Number: 6

Function: This test checks too many indirections fault.

Procedure: Set up the descriptor tables for contiguous segments. Assert the Indirect bit in an entry and set the second word of the entry to point to itself. Flush the caches. Reference the virtual address that corresponds to this entry.

Hardware Tested: The MMU chip is tested.

Data Returned: Raw Data: The value in the MMU Fault Code register is returned.
Supplemental Data: The value in the MMU Fault Address register is returned.

Notes: None

Test Number: 7

Function: This test checks the segment offset fault.

Procedure: Set up the descriptor tables for contiguous segments. Set the segment offset of an entry to zero. Flush the caches. Reference the virtual address that corresponds to this entry.

Hardware Tested: The MMU chip is tested.

Data Returned: Raw Data: The value in the MMU Fault Code register is returned.
Supplemental Data: The value in the MMU Fault Address register is returned.

Notes: None

==========

Test Numbers: 8 through 31

Function: These tests check the access fault.

Procedure: Set up the descriptor tables for contiguous segments. Determine that the appropriate access occurs when allowed, and a fault occurs when disallowed. Flush the caches. Reference the virtual address that corresponds to this entry.

Hardware Tested: The MMU chip is tested.

Data Returned: Raw Data: The value in the MMU Fault Code register is returned.
Supplemental Data: The value in the MMU Fault Address register is returned.

Notes: Tests 8 through 19 execute at user level while Tests 20 through 31 execute at kernel level.

Test No.	Access Allowed	Operation	Test No.	Access Allowed	Operation
8	RWE	R	20	RWE	R
9	RWE	W	21	RWE	W
10	RWE	E	22	RWE	E
11	RE	R	23	E	R
12	RE	E	24	RE	E
13	E	E	25	E	E
14	RE	W	26	RE	W
15	EO	R	27	EO	R
16	EO	W	28	EO	W
17	Nonaccessible	R	29	Nonaccessible	R
18	Nonaccessible	W	30	Nonaccessible	W
19	Nonaccessible	E	31	Nonaccessible	E

Test Number: 32

Function: This test checks the access and segment offset fault.

Procedure: Set up the descriptor tables for contiguous segments. Set the access field of an entry to nonaccessible by all. Also set the Maximum Offset field to zero. Flush the caches. Reference the virtual address that corresponds to this entry.

Hardware Tested: The MMU chip is tested.

Data Returned: Raw Data: The value in the MMU Fault Code register is returned.
Supplemental Data: The value in the MMU Fault Address register is returned.

Notes: None

==========

Test Number: 33

Function: This test checks the Page Descriptor Table (PDT) length fault.

Procedure: Set up the descriptor tables for paged segments. Set up the PDTs. Set the Maximum Offset field to zero so the PDT length fault can occur. Flush the caches. Reference the virtual address that corresponds to this entry.

Hardware Tested: The MMU chip is tested.

Data Returned: Raw Data: The value in the MMU Fault Code register is returned.
Supplemental Data: The value in the MMU Fault Address register is returned.

Notes: None

==========

Test Number: 34

Function: This test checks the PDT not present fault.

Procedure: Set up the descriptor tables for paged segments. Reset the P bit in the segment descriptor to zero to indicate PDT not present. Flush the caches. Reference the virtual address that corresponds to this entry.

Hardware Tested: The MMU chip is tested.

Data Returned: Raw Data: The value in the MMU Fault Code register is returned.
Supplemental Data: The value in the MMU Fault Address register is returned.

Notes: None

Test Number: 35

Function: This test checks the page not present fault

Procedure: Set up the descriptor tables for paged segments. Set up the PDTs with Page present to zero. Flush the caches. Reference the virtual address that corresponds to this entry.

Hardware Tested: The MMU chip is tested.

Data Returned: Raw Data: The value in the MMU Fault Code register is returned.
Supplemental Data: The value in the MMU Fault Address register is returned.

Notes: None

==========

Test Number: 36

Function: This test checks the miss process memory fault.

Procedure: Set up the descriptor tables for paged segments. Corrupt the contents of an entry so that it will be fetched for the Miss Processing. Flush the caches. Reference the virtual address that corresponds to this entry.

Hardware Tested: The MMU chip is tested.

Data Returned: Raw Data: The value in the MMU Fault Code register is returned.
Supplemental Data: The value in the MMU Fault Address register is returned.

Notes: None

==========

Test Number: 37

Function: This test checks the Reference and Modify (R&M) update memory fault.

Procedure: Set up the descriptor tables for paged segments. Flush the caches. Configure the cache for a hit. Corrupt the contents of the corresponding segment descriptor location. Reference the virtual address that corresponds to this entry.

Hardware Tested: The MMU chip is tested.

Data Returned: Raw Data: The value in the MMU Fault Code register is returned.
Supplemental Data: The value in the MMU Fault Address register is returned.

Notes: None

Test Number: 38

Function: This test checks the double page hit.

Procedure: Set up the descriptor tables for paged segments. Flush the caches. Also set up two Page Descriptor Cache (PDC) entries. Reference the virtual address that corresponds to this entry.

Hardware Tested: The MMU chip is tested.

Data Returned: Raw Data: The ualue in the MMU Fault Code register is returned.
Supplemental Data: The value in the MMU Fault Address register is returned.

Notes:

Test	Dependencies	Function
38	WE 32101	Double page hit
38	WE 32201	No test run

Phase #13 — MPB Interrupt System Diagnostics

Phase Name:	Interrupt System (int_tst)
Type:	Normal
Function:	This phase tests the Interrupt System via trackable, noninterfering interrupt sources.
Tests:	Test 1 — checks the Buffered Microbus (BuBus) Level 15 operational interrupt (System CPU active). Test 2 — checks the BuBus Level 15 operational interrupt (System CPU passive). Test 3 — checks the BuBus Level 15 multiprocessor interrupt (MPB CPU active). Test 4 — checks the BuBus Level 15 multiprocessor interrupt (MPB CPU passive). Test 5 — checks the Microbus (uBus) Level NMI multiprocessor interrupt (MPB CPU active). Test 6 — checks the uBus Level NMI multiprocessor interrupt (MPB CPU passive). Test 7 — checks the uBus Level 12 multiprocessor interrupt (MPB CPU active). Test 8 — checks the uBus Level 12 multiprocessor interrupt (MPB CPU passive). Test 9 — checks the uBus Level 10 multiprocessor interrupt (MPB CPU active). Test 10 — checks the uBus Level 10 multiprocessor interrupt (MPB CPU passive).
Time:	1 second
Warnings:	A failure of this phase may affect other tests that assume the system is clear of interrupts.

Notes: The test interrupt handler masks all interrupts in the Process Status Word (PSW) of the main program.

System Board Interrupt/CSER Bit Assignment

Level	Vector	Source	CSER Bit
NMI	0	Sanity Timer & Abort Switch	29 & 30
NMI	0	Thermal Shutdown Request	15
15	15	Interval Timer, Power Down Req.	00 & 01
15	15	uBus/BuBus OPINT	02
15	15	Single/Multiple Bit Error	23 & 24
15	15	uBus/BuBus Received Fail	25
15	15	uBus Timer	26
15	15	I/O Received PFAIL, PFLT, FLT	--
14	SBD/P*	BuBus (Real Time Int)	--
13	13	DUARTs & DMA Complete	03 & 04
12	SBD/P*	BuBus (Block)	--
11	11	Floppy & Floppy DMA Complete	--
10	SBD/P*	BuBus (Character)	--
9	9	PIR-9 (from CSER)	05
8	8	PIR-8 (from CSER)	06

* SBD/P - For BuBus connector: First- and second-level interrupt vector is provided by system board HW, for the BuBus Processor board.

Phase #13 Tests

Test Number: 1

Function: This test ensures that a Level 15 interrupt is generated when the MPB Control and Status Register (CSR) INT15 bit is set.

Procedure: Drop the CPUs interrupt Level to 14 so as to block all normal interrupts below Level 15, set the MPB CSR INT15 bit and confirm that an interrupt is generated and that the Control Status and Error Register (CSER) accurately reflects the correct scenario.

Hardware Tested: The MPB CSR, the Interrupt System, and the System CSER are tested.

Data Returned: Interrupt handler data from the CSER read is returned.

Notes: None

==========

Test Number: 2

Function: This test ensures that a Level 15 interrupt is generated when the MPB CSR INT15 bit is set by the MPB.

Procedure: Drop the CPUs interrupt Level to 14 so as to block all normal interrupts below Level 15, request the MPB to run ../bpb/bint1_tst() and confirm that an interrupt is generated and that the CSER accurately reflects the correct scenario.

Hardware Tested: The MPB, the MPB CSR, the Interrupt System, and the System CSER are tested.

Data Returned: Interrupt handler data from the CSER read is returned.

Notes: None

==========

Test Number: 3

Function: This test ensures that the MPB CSR bit, Interrupt Central Processing Unit (INTCPU), generates a local Level 15 interrupt to the MPB Processor.

Procedure: Drop the MPB Processor interrupt mask to Level 14. Set the MPB CSR bit to INTCPU. Confirm a Level 15 interrupt was taken.

Hardware Tested: The MPB Interrupt System and the MPB CSR are tested.

Data Returned: Interrupt handler data from the MPB CSR read is returned.

Notes: None

Test Number:	4
Function:	This test ensures that the MPB CSR bit, BRDINT15, generates a local Level 15 interrupt to the MPB Processor when set by the main CPU.
Procedure:	Drop the MPB Processor interrupt mask to Level 14. Request the main CPU to set BRDINT15 bit via bpbint15( ). Confirm a Level 15 interrupt was taken.
Hardware Tested:	The MPB Interrupt System and the MPB CSR are tested.
Data Returned:	Interrupt handler data from the MPB CSR read is returned.
Notes:	None

==========

Test Number:	5
Function:	This test ensures that the MPB CSR bit, ADPNMI, generates a local NMI interrupt to the MPB Processor.
Procedure:	Drop the MPB Processor interrupt mask to Level 14. Set the MPB CSR bit to ADPNMI. Confirm a NMI interrupt was taken.
Hardware Tested:	The MPB Interrupt System and the MPB CSR are tested.
Data Returned:	Interrupt handler data from the MPB CSR read is returned.
Notes:	This test is only run on the Microbus (uBus) MPB.

==========

Test Number:	6
Function:	This test ensures that the MPB CSR bit, ADPNMI, generates a local NMI interrupt to the MPB Processor when set by the main CPU.
Procedure:	Mask all interrupts on the MPB. Request the main CPU to set ADPNMI bit via bpbnmi( ). Confirm a level NMI interrupt was taken.
Hardware Tested:	The MPB Interrupt System and the MPB CSR are tested.
Data Returned:	Interrupt handler data from the MPB CSR read is returned.
Notes:	This test is only run on the uBus MPB.

==========

Test Number:	7
Function:	This test ensures that the MPB CSR bit, ADPINT12 generates a local Level 12 interrupt to the MPB Processor.
Procedure:	Drop the MPB Processor interrupt mask to Level 11. Set the MPB CSR bit to ADPINT12. Confirm a Level 12 interrupt was taken.
Hardware Tested:	The MPB Interrupt System and the MPB CSR are tested.
Data Returned:	Interrupt handler data from the MPB CSR read is returned.
Notes:	This test is only run on the uBus MPB.

Test Number:	8
Function:	This test ensures that the MPB CSR bit, ADPINT12, generates a local Level 12 interrupt to the MPB Processor when set by main the CPU.
Procedure:	Drop the MPB Processor interrupt mask to Level 11. Request the main CPU to set ADPINT12 bit via bpbint12( ). Confirm a Level 12 interrupt was taken.
Hardware Tested:	The MPB Interrupt System and the MPB CSR are tested.
Data Returned:	Interrupt handler data from the MPB CSR read is returned.
Notes:	This test is only run on the uBus MPB.

==========

Test Number:	9
Function:	This test ensures that the MPB CSR bit, ADPINT10, generates a local Level 10 interrupt to the MPB Processor.
Procedure:	Drop the MPB Processor interrupt mask to Level 9. Set the MPB CSR bit to ADPINT10. Confirm a Level 10 interrupt was taken.
Hardware Tested:	The MPB Interrupt System and the MPB CSR are tested.
Data Returned:	Interrupt handler data from the MPB CSR read is returned.
Notes:	This test is only run on the uBus MPB.

==========

Test Number:	10
Function:	This test ensures that the MPB CSR bit, ADPINT10, generates a local Level 10 interrupt to the MPB Processor when set by the main CPU.
Procedure:	Drop the MPB Processor interrupt mask to Level 9. Request the main CPU to set ADPINT10 bit via bpbint10( ). Confirm a Level 10 interrupt was taken.
Hardware Tested:	The MPB Interrupt System and the MPB CSR are tested.
Data Returned:	Interrupt handler data from the MPB CSR read is returned.
Notes:	This test is only run on the uBus MPB.

Phase #14 — MPB Main Store Diagnostics

Phase Name:	Main Store (bram_tst)
Type:	Demand
Function:	This phase checks the memory by performing the following tests.
Tests:	Test 1 — checks all memory accesses on a word basis with alternate data patterns through memory range. A pattern of 55's followed by aa's. Test 2 — checks memory accesses by writing words and reading back two half words. Test 3 — checks memory accesses by writing two half words and reading back 4 bytes. Test 4 — checks memory accesses by writing and reading 4 bytes.
Time:	7 seconds per megabyte
Warnings:	None
Notes:	None

Phase #14 Tests

Test Number:	1
Function:	This test checks all memory accesses on a word basis with alternate data patterns throughout memory range.
Procedure:	A pattern of 55's and aa's is written and then read.
Hardware Tested:	The Buffered Microbus (BuBus) Board Memory is tested.
Data Returned:	The failing address, expected value, and actual value read are returned.
Notes:	None

==========

Tests Numbers:	2 through 4
Function:	These tests check variable size accesses.
Procedure:	Test 2 — writes words and reads back as shorts. Increment address by 200. Test 3 — writes two shorts and reads back as bytes. Increment address by 400. Test 4 — writes and reads 4 bytes. Increment address by 800.
Hardware Tested:	The BuBus Board Memory is tested.
Data Returned:	The failing address, expected value, and actual value read are returned.
Notes:	None

Phase #15 — MPB Memory Alignment Fault Detection Diagnostics

Phase Name:	Fault/Alignment (bflt_tst)
Type:	Normal
Function:	This phase tests the MPBs detection and correct handling of memory and alignment faults.
Tests:	Test 1 — confirms that the MPB responds correctly to a memory fault. Test 2 — confirms that the MPB responds correctly to an unequipped memory fault. Test 3 — confirms that the MPB responds correctly to an alignment fault.
Time:	1 second
Warnings:	None
Notes:	None

Phase #15 Tests

Test Number:	1
Function:	This test confirms that the MPB responds correctly to a faulted memory location.
Procedure:	Using a specific memory location, have the main board processor load a specific memory location with known bad parity. Then let the MPB read that location. This should cause the MPB to fault and set BPERROR in the Central Status Request (CSR).
Hardware Tested:	The MPB CPU and CSR are tested.
Data Returned:	The failing test number (last number displayed on the system console prior to failure) is returned.
Notes:	None

==========

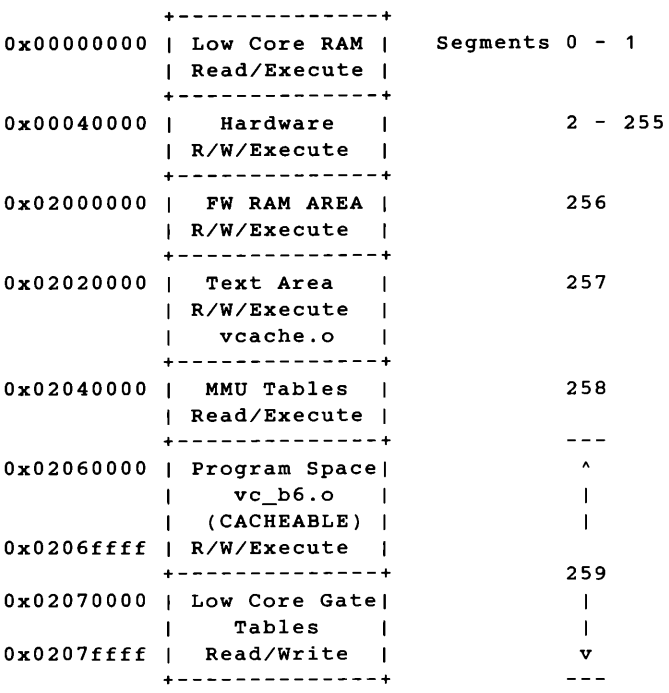
Test Number:	2
Function:	This test confirms that the MPB responds correctly when it reads an unequipped memory location.
Procedure:	Going to a specific address which is known not to be equipped, reads that location and confirms that the MPB is faulted and an exception occurs.
Hardware Tested:	The MPB CPU is tested.
Data Returned:	The failing test number (last number displayed on the system console prior to failure) is returned.
Notes:	None

Test Number:	3
Function:	This test confirms that the MPB correctly detects an alignment fault.
Procedure:	Purposely load a register with a nonword aligned address. Then attempt to read a word at that address. Confirm that the MPB is faulted and that the BPALFLT is set in the CSR.
Hardware Tested:	The MPB CPU and CSR are tested.
Data Returned:	The failing test number (last number displayed on the system console prior to failure) is returned.
Notes:	None

Phase #16 — MPB Virtual Cache Diagnostics

Phase Name:	Virtual Cache (vcache)
Type:	Normal
Function:	This phase turns the VCACHE on and off confirming read/write cache patterns.
Tests:	Test 1 — confirms that the VCACHE card correctly responds to full word hit/miss for read accesses. Test 2 — confirms that the VCACHE card correctly responds to full word hit/miss for write accesses. Test 3 — confirms that the VCACHE card correctly responds to byte miss write access. Test 4 — confirms that the VCACHE card correctly responds to half word miss write access. Test 5 — confirms that the VCACHE card correctly responds to byte hit write access. Test 6 — confirms that the VCACHE card correctly responds to half word hit write access. Test 7 — confirms that the VCACHE card flushes correctly. Test 8 — confirms that the VCACHE card flushes data correctly.
Time:	10 seconds
Warnings:	None

Notes: The MMU setup is as follows (1 to 1 mapping, Section 0 only):



Pages 0 through 3 were remapped to the end of bss to allow the VCACHE card to be tested. The rest of memory is still mapped 1 to 1. This remapping is due to the fact that low core memory is used as the VCACHE RAM area.

This test phase is only executed on Buffered Microbus (BuBus) MPB.

Phase #16 Tests

Test Number: 1

Function: This test ensures that the VCACHE card correctly responds to full word hit/miss read accesses.

Procedure: In virtual mode, write pattern to memory with the VCACHE card disabled. Enable the VCACHE card and read memory. This will load the VCACHE card. Disable the VCACHE card and change the pattern in memory. Enable the VCACHE card and perform a read. The pattern in the VCACHE card should be pattern read.

Hardware Tested: The MPB VCACHE card is tested.

Data Returned: The failing test number, the failing address, the expected pattern, and the pattern received are returned.

Notes: None

Test Number: 2

Function: This test ensures that the VCACHE card correctly responds to full word hit/miss write accesses.

Procedure: In virtual mode, write pattern to memory with the VCACHE card enabled, thus loading the VCACHE card. Disable the VCACHE card and change the pattern in memory. Enable the VCACHE card and perform a read. The pattern in the VCACHE card (the first pattern) should be pattern read.

Hardware Tested: The MPB VCACHE card is tested.

Data Returned: The failing test number, the failing address, the expected pattern, and the pattern are returned.

Notes: None

==========

Test Number: 3

Function: This test ensures that the VCACHE card correctly responds to byte miss write accesses.

Procedure: In virtual mode, write pattern to memory with byte accesses and the VCACHE card enabled. Disable the VCACHE card and change the memory test pattern. Enable the VCACHE card and verify that the pattern read is the second pattern.

Hardware Tested: The MPB VCACHE card is tested.

Data Returned: The failing test number, the failing address, the expected pattern, and the pattern are returned.

Notes: None

==========

Test Number: 4

Function: This test ensures that the VCACHE card correctly responds to half word miss write accesses.

Procedure: In virtual mode, write pattern to memory with half word accesses and the VCACHE card enabled. Disable the VCACHE card and change the memory test pattern. Enable the VCACHE card and verify that the pattern read is the second pattern.

Hardware Tested: The MPB VCACHE card is tested.

Data Returned: The failing test number, the failing address, the expected pattern, and the pattern are returned.

Notes: None

Test Number: 5

Function: This test ensures that the VCACHE card correctly responds to byte hit write accesses.

Procedure: In virtual mode, write pattern to memory with word accesses with the VCACHE card enabled. This should fill the VCACHE card. Then fill memory with the second test pattern using byte accesses. Disable the VCACHE card and verify that the second pattern was written into memory. Fill memory with the third test pattern using word accesses. Enable the VCACHE card and verify that the pattern read is the second pattern.

Hardware Tested: The MPB VCACHE card is tested.

Data Returned: The failing test number, the failing address, the expected pattern, and the pattern are returned.

Notes: None

==========

Test Number: 6

Function: This test ensures that the VCACHE card correctly responds to half word hit write accesses.

Procedure: In virtual mode, write pattern to memory using half word accesses with the VCACHE card enabled. This should fill the VCACHE card. Then fill memory with the second test pattern using half word accesses. Disable the VCACHE card and verify that the second pattern was written into memory. Fill memory with the third test pattern using half word accesses. Enable the VCACHE card and verify that the pattern read is the second pattern.

Hardware Tested: The MPB VCACHE card is tested.

Data Returned: The failing test number, the failing address, the expected pattern, and the pattern are returned.

Notes: None

==========

Test Number: 7

Function: This test ensures that the VCACHE card flushes correctly.

Procedure: In virtual mode and the VCACHE card disabled, write the memory with the first test pattern using word accesses. Enable the VCACHE card and confirm that the first test pattern was written into memory (this should load VCACHE). Disable the VCACHE card and write the second test pattern. Enable the VCACHE card and confirm that the first test pattern is present in VCACHE. Flush the VCACHE card, read memory, and confirm that the second test pattern is now read.

Hardware Tested: The MPB VCACHE card is tested.

Data Returned: The failing test number, the failing address, the expected pattern, and the pattern are returned.

Notes: None

Test Number:	8
Function:	This test ensures that the VCACHE card flushes data correctly.
Procedure:	In virtual mode and the VCACHE disabled, write memory with the first test pattern using word accesses. Enable the VCACHE card and confirm that the first test pattern was written into memory (this should load VCACHE). Disable the VCACHE card and write the second test pattern. Enable the VCACHE card and confirm that the first test pattern is present in the VCACHE card. Flush the VCACHE card, read memory, and confirm that the second test pattern is now read.
Hardware Tested:	The MPB VCACHE card is tested.
Data Returned:	The failing test number, the failing address, the expected pattern, and the pattern are returned.
Notes:	None

Phase #17 — MPB Extended Virtual Cache (VCACHE) Diagnostics

Phase Name: Virtual Cache (evcache)

Type: Demand

Function: This phase sets up the MMU, turns the VCACHE on, and executes the program "main."

Tests: Test 1 — confirms that the ICACHE bit in the CPU Process Status Word (PSW) works.

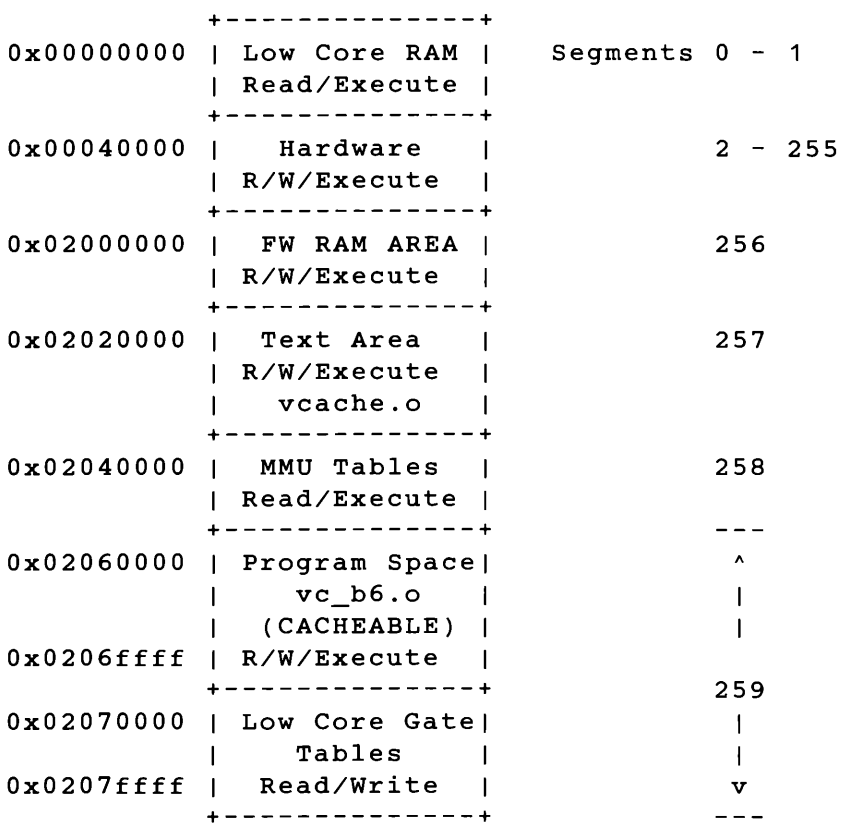
Test 2 — confirms that caching of instructions in the VCACHE card works.

Test 3 — confirms that utilizing the ICACHE bit and the VCACHE card works.

Time: 45 seconds

Warnings: None

Notes: The MMU setup is as follows (1 to 1 mapping, Section 0 only):



Pages 0 through 3 were remapped to allow the VCACHE card to be tested. The rest of memory is still mapped 1 to 1. This remapping is due in part to the fact that the low core memory area is used as the VCACHE RAM area.

This test phase is executed on the Buffered Microbus (BuBus) MPBs.

Phase #17 Tests

Test Number:	1
Function:	This test confirms that the Instruction Cache (ICACHE) in CPU works.
Procedure:	Run the benchmark program without the ICACHE bit on. Then turn the ICACHE bit on and confirm that the benchmark program ran faster than with it off.
Hardware Tested:	The CPU ICACHE is tested.
Data Returned:	The failing test number is displayed on the system console.
Notes:	None

==========

Test Number:	2
Function:	This test confirms that the benchmark program executes faster when executing from the VCACHE card versus previous straight CPU time.
Procedure:	Turn the VCACHE card on and execute the program. Confirm that the benchmark program executed quicker than when the CPU alone was used.
Hardware Tested:	The MPB VCACHE card is tested.
Data Returned:	The failing test number is displayed on the system console.
Notes:	None

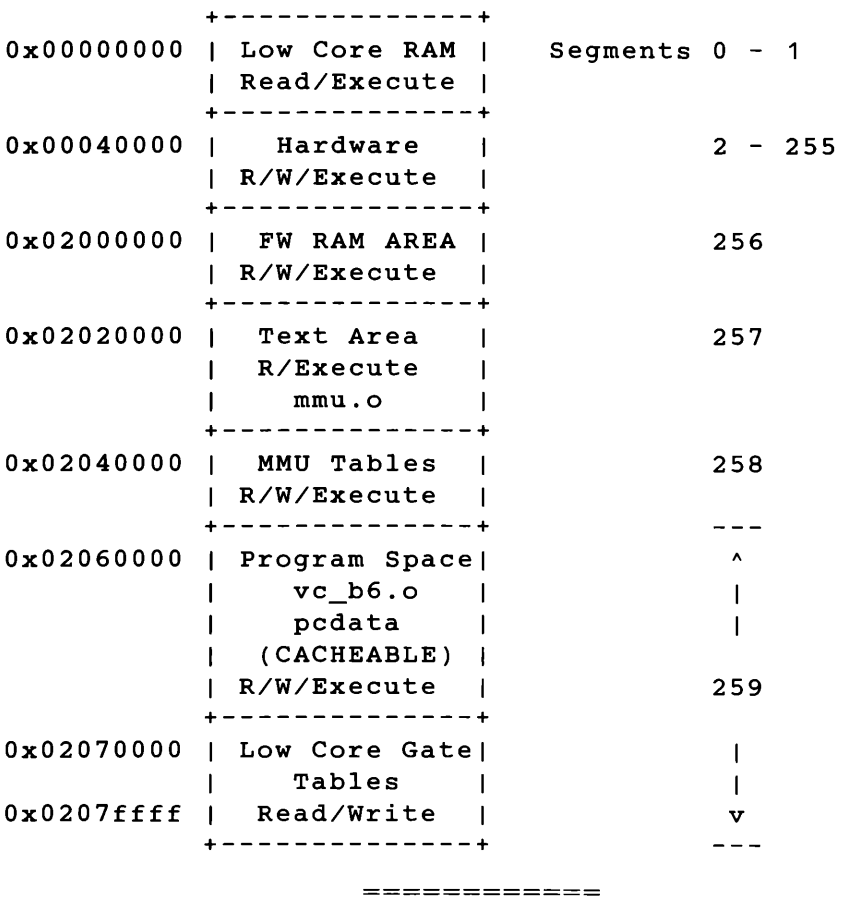
==========

Test Number:	3
Function:	This test confirms that the benchmark runs the quickest when both the CPU ICACHE bit and the VCACHE card are on.
Procedure:	Run the benchmark program and confirm that it ran faster than any of the previous execution times.
Hardware Tested:	The MPB VCACHE card and CPU ICACHE bit are tested.
Data Returned:	The failing test number is displayed on the system console.
Notes:	None

Phase #18 — MPB Physical Cache Diagnostics

Phase Name:	Physical Cache (bpc1_tst)
Type:	Normal
Function:	This phase tests the physical Memory Management Unit (MMU) cache memory by performing the following tests:
Tests:	Test 1 — checks that the double-bit memory fault works with the MMU cache on. Test 2 — checks all memory accesses on a word basis with various data patterns throughout cache memory range; 00's followed by FF's followed by random patterns. Test 3 — checks memory accesses by writing words and reading back two half words. Test 4 — checks memory accesses by writing two half words and reading back 4 bytes. Test 5 — checks memory accesses by writing and reading 4 bytes. Test 6 — checks PCACHE tag fields.
Time:	10 seconds
Warnings:	None
Notes:	These test phases will only be exercised on a PBus MPB. Only a WE 32201, Version 2.3 or later supports caching. The MMUVR register has the version encoded in its two low-order hexadecimal digits. If caching is not supported by the MPB, an appropriate message is displayed and the phase exits with status of NTR (No Tests Run). This phase works by turning on the cache and performing accesses that are expected to hit the cache.

The MMU setup is as follows (1 to 1 mapping, Section 0 only):



==========

Phase #18 Tests

Test Number: 1

Function: This test ensures that a double-bit error can be detected in virtual mode with the MMU cache enabled.

Procedure: Inhibit Error Correction Coding (ECC). Fault the contents of a word. Determine if a multiple-bit error occurred.

Hardware Tested: The Dynamic Memory Controller and MMU are tested.

Data Returned: The failing address, expected value, and actual the value read are returned.

Notes: None

Test Number:	2
Function:	This test checks all memory accesses on a word basis with alternate data patterns throughout memory range.
Procedure:	Various patterns are written and then read.
Hardware Tested:	The MMU Data Cache [for example, Physical Cache (PCACHE)] is tested.
Data Returned:	The failing address, expected value, and actual value read are returned.
Notes:	None

==========

Tests Numbers:	3 through 5
Function:	These tests check variable size accesses.
Procedure:	Test 3 — writes words and reads back as shorts. Increment address by 1000. Test 4 — writes two shorts and reads back as bytes. Increment address by 1000. Test 5 — writes and reads 4 bytes. Increment address by 1000.
Hardware Tested:	The MMU Data Cache (for example, PCACHE) is tested.
Data Returned:	The failing address, expected value, and actual value read are returned.
Notes:	None

==========

Test Number:	6
Function:	This test checks the tag field of the PCACHE.
Procedure:	Access every entry in the cache, change a Tag bit, and repeat.
Hardware Tested:	The MMU Data Cache (for example, PCACHE) is tested.
Data Returned:	The failing address, expected value, and actual value read are returned.
Notes:	None

Phase #19 — MPB Extended Physical Cache Diagnostics

Phase Name:	Physical Cache (bpc2_tst)
Type:	Demand
Function:	This phase sets up the Memory Management Unit (MMU), turns cache on, and executes benchmarks. For each test, if execution time with a cache on is less than a set percentage of the execution time with a cache off, that test passes.
Tests:	Test 1 — confirms that the ICACHE bit in the CPU Process Status Word (PSW) works. Test 2 — confirms that caching in MMU works. Test 3 — confirms that utilizing ICACHE bit and MMU works.
Time:	45 seconds
Warnings:	None
Notes:	These test phases will only be exercised on a PBus MPB. Only a WE 32201, Version 2.3 or later supports caching. The MMUVR register has the version encoded in its two low-order hexadecimal digits. If caching is not supported by the MPB, an appropriate message is displayed and the phase exits with status of FAIL indicating that an illegal configuration was detected. The MMU mapping is the same as that for the Physical Cache (PCACHE) test.

Phase #19 Tests

Test Number: 1

Function: This test confirms that the Instruction Cache (ICACHE) in CPU works.

Procedure: Run the benchmark program with the ICACHE bit off. Then turn the ICACHE bit on, and run the benchmark program again; then confirm that the benchmark program runs faster with the ICACHE bit on.

Hardware Tested: The CPU ICACHE is tested.

Data Returned: The failing test number is displayed on the system console.

Notes: Execution time with the cache on must be less than or equal to 60 percent of time with the cache off.

==========

Test Number: 2

Function: This test confirms that the benchmark program executes faster when executing from the MMU cache versus previous straight CPU time.

Procedure: Turn the MMU cache on and execute the program. Confirm that the benchmark program executed quicker than when the CPU alone was used.

Hardware Tested: The MMU Data Cache (for example, PCACHE) is tested.

Data Returned: The failing test number is displayed on the system console.

Notes: The execution time with the cache on must be less than or equal to 51 percent of execution time with the cache off.

==========

Test Number: 3

Function: This test confirms that the benchmark runs the quickest when both the CPU ICACHE bit and MMU cache are on.

Procedure: Run the benchmark program and confirm that it ran faster than any of the previous execution times.

Hardware Tested: The MMU Data Cache (for example, PCACHE) and CPU ICACHE bit are returned.

Data Returned: The failing test number is displayed on the system console.

Notes: The execution time with both caches on must be less than or equal to 45 percent of execution time with both caches off.

Phase #20 — MPB Counter Diagnostics

Phase Name:	Adaptive Counter Unit (cntr_tst)
Type:	Normal
Function:	This phase tests the uBus MPB adaptive scheduler counter by performing the following tests:
Tests:	Test 1 — confirms clock active test - passive (from SBD). Test 2 — confirms clock reset - passive (from SBD). Test 3 — confirms clock active test - active (from UBD). Test 4 — confirms clock reset - active (from UBD).
Time:	1 second
Warnings:	None
Notes:	These test phases will only be performed on the PBus MPBs.

Phase #20 Tests

Test Number:	1
Function:	This test confirms that the counter is active and counting.
Procedure:	Reset the counter and then do two successive reads separated by a 10 millisecond hardware timing delay. Confirm that the last time read is greater than the first time read and that the second time read is within a minimum range value.
Hardware Tested:	The MPB counter is tested.
Data Returned:	The failing test number is displayed on the system console.
Notes:	The minimum range value (0x03b0) was determined via actual laboratory experimentation on a 24MHz 3B2/1000.

==========

Test Number:	2
Function:	This test confirms that the counter reset function works.
Procedure:	With the timer running from the previous test, read the counter, reset the counter, and confirm that the new time value is less than the first read time value.
Hardware Tested:	The MPB counter is tested.
Data Returned:	The failing test number is displayed on the system console.
Notes:	None

Tests Numbers:	3 and 4
Function:	These tests confirm that the MPB itself can perform the same functions as the System Board (SBD) CPU.
Procedure:	Download into low core memory on the MPB static RAM area and have the MPB repeat the same identical checks as performed in Tests 1 and 2 by the SBD CPU, with the exception that minimum measurements are not made as the hardware timer is used to detect a hung MPB board.
Hardware Tested:	The MPB counter is tested.
Data Returned:	The failing test number is displayed on the system console.
Notes:	None

Chapter 5: Cartridge Tape Controller Diagnostics

Introduction

This chapter contains the diagnostic phases and tests for the 3B2 computer Cartridge Tape Controller (CTC) card. Each CTC card supports one cartridge tape drive and one floppy disk drive. The following components are found on the CTC card:

- INTEL 80186 Central Processing Unit (CPU)
- AMD 9517 Direct Memory Access Controller (DMAC)
- Disk Controller/Formatter (WD2793)
- Peripheral Control and Status Register (PCSR)
- Random Access Memory (RAM)
- Read Only Memory (ROM).

An important external component of the CTC card is the 3B2 computer Input/Output (I/O) bus. The CTC card uses the 3B2 computer I/O bus to communicate with the System Board (SBD).

If your 3B2 computer is equipped with a CTC card, the Equipped Device Table (EDT) should list it. You can use the diagnostic monitor command "show" (DGMON > **s**) to display the EDT. If the CTC card is not listed in the EDT and a VOID or NULL is listed, one or more of the following hardware devices on the CTC card may be faulty.

- INTEL 80186 Microprocessor
- CTC ROM
- CTC ID/Vector register
- CTC Address/Data bus
- CTC interface to the 3B2 computer I/O bus.

Eighteen diagnostic phases run tests on all major CTC card components. The Table of Contents listing will help you locate the descriptions for each CTC card phase and its associated tests. The phase descriptions are organized numerically in the same order that the phases are run on the 3B2 computer.

Phase Descriptions

Phase #1 — Common I/O and Peripheral Sanity Test

Phase Name: Cartridge Tape I/O Sanity (cio)

Type: Normal

Function: This phase verifies that the CTC card Common I/O (CIO) interface is functioning properly.

Test: Test 1 — verifies that the diagnostic results can be transferred to the System Board (SBD).

Time: 1 second

Warnings: None

Notes: All successful diagnostics return the PASS result flag to the Diagnostic Monitor (DGMON). The DGMON then displays ATP (All Tests Passed) on the system console. If Phase 1 fails, the following hardware may be faulty:

- INTEL 80186 Microprocessor
- CTC RAM/ROM
- 3B2 computer I/O bus Interface Acknowledge Circuitry
- CTC interface to the 3B2 computer I/O bus
- CTC Address/Data bus.

Phase #1 Test

Test Number: 1

Function: This test verifies that the CIO hardware and firmware are functioning properly.

Procedure: This test uses the following standard procedure:

1. The I/O slot number of the CTC card in the 3B2 computer is determined.
2. The CTC card is reset.
3. The sysgen data block is initialized.
4. The CTC card is initialized (sysgen) by sending express [Interrupt 0 (INT0)] and attention (INT1) interrupts sequentially.
5. The X86 code is downloaded by using the CIO firmware command [Download Memory (DLM)].
6. Execution of the phase is started by using the CIO firmware command [Force Call to Function (FCF)].
7. A function call to **"phasend( )"** is made when the phase is complete. Phasend returns the test results to the SBD.

Hardware Tested: The CTC interface to the 3B2 computer I/O bus is tested.

Data Returned: None

Phase #2 — PCSR Write/Read Test

Phase Name:	Cartridge Tape Control and Status Register (pcsr)
Type:	Demand
Function:	This phase diagnoses and reports any bit interdependency of the CTC Peripheral Control and Status Register (PCSR).
Tests:	Test 1 — clears and reads the PCSR bit 0. Test 2 — clears and reads the PCSR bit 1. Test 3 — clears and reads the PCSR bit 3. Test 4 — clears and reads the PCSR bit 4.
Time:	1 second
Warnings:	None
Notes:	None

Phase #2 Tests

Test Numbers:	1 through 4
Function:	These tests verify the ability to write to and read from the PCSR.
Procedure:	Each bit is cleared with a write and verified with a read.
Hardware Tested:	The CTC PCSR is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

Phase #3 — Upper RAM Write/Read Test

Phase Name:	Cartridge Tape Upper RAM
Type:	Demand
Function:	This phase diagnoses and reports any errors in the operation of the CTC RAM and refresh circuitry.
Tests:	Tests 1 through 4 — walk data patterns through the upper 64 kilobytes of RAM. Tests 5 and 6 — check the upper 64 kilobytes of RAM and refresh circuitry.
Time:	20 seconds
Warnings:	None
Notes:	None

Phase #3 Tests

Test Numbers:	1 through 4
Function:	These tests check the operation of the upper RAM.
Procedure:	The following test sequence is used to test the upper RAM (addresses 0x80000 through 0x9fffe): 1. All memory locations are written with 0's in ascending (small addresses to large addresses) order. 2. All memory locations are read, expecting 0's, and then all memory is written with 1's in descending order. 3. All memory locations are read, expecting 1's, and then all memory is written with 5's in ascending order. 4. All memory locations are read, expecting 5's, and then all memory is written with A's in descending order.
Hardware Tested:	The upper RAM locations are tested.
Data Returned:	The number of the failing test, the actual data, and the expected data are returned.
Notes:	None

Test Numbers: 5 and 6

Function: These tests verify that the upper RAM and refresh circuitry are functioning properly.

Procedure: All memory locations are read, expecting A's, and then all memory is written with the following patterns in ascending order:

pat(00)=0xF606 pat(08)=0xF603 pat(16)=0x7FB8 pat(24)=0x401B
pat(01)=0xE727 pat(09)=0xD732 pat(17)=0x6E99 pat(25)=0x612A
pat(02)=0xD25A pat(10)=0xA245 pat(18)=0x59F4 pat(26)=0x0A5F
pat(03)=0xC37B pat(11)=0x8374 pat(19)=0x48D5 pat(27)=0x2B6E
pat(04)=0xBAEC pat(12)=0x1988 pat(20)=0x3D41 pat(28)=0xBD96
pat(05)=0xABCD pat(13)=0x38B9 pat(21)=0x2C60 pat(29)=0x9CA7
pat(06)=0x91A2 pat(14)=0x5FC0 pat(22)=0x141F pat(30)=0xE5DC
pat(07)=0x8083 pat(15)=0x7EF1 pat(23)=0x053E pat(31)=0xC4ED

The patterns are written according to the following formula:

memory location(i) = pat(i modulo 32)
where i = {0, 1, 2 ... 32K} where K=1024
a memory location = 16 bits.

All memory locations will be read in ascending order after a 1-second delay to test the refresh circuitry. The refresh rate is each 2.048 milliseconds. The time needed to read or write all 64 kilobytes of memory (16 bits on each read or write) is approximately 20.0 milliseconds (125 nanoseconds per cycle, it takes 5 cycles to perform a read or write). A delay of 1 second, in addition to the 20.0 milliseconds needed to write all memory, is sufficient to test the refresh circuitry. This phase executes from the lower 64 kilobytes of RAM.

Hardware Tested: The upper RAM and refresh circuitry are tested.

Data Returned: The test number that failed, the expected data, and the actual data are returned.

Notes: None

Phase #4 — Lower RAM Write/Read Test

Phase Name:	Cartridge Tape Lower RAM
Type:	Demand
Function:	This phase repeats the test sequence of Phase 3 except it uses the lower RAM address range. This phase executes from the upper 64 kilobytes of RAM. The stack, which is usually located in the lower 64 kilobytes of RAM, is moved to the upper 64 kilobytes of RAM before the tests begin.
Tests:	Tests 1 through 4 — walk data patterns through the lower 64 kilobytes of RAM. Tests 5 and 6 — check the lower 64 kilobytes of RAM and refresh circuitry.
Time:	20 seconds
Warnings:	None
Notes:	None

Phase #4 Tests

Test Numbers:	1 through 4
Function:	Tests 1 through 4 check the operation of the lower RAM.
Procedure:	The following test sequence is used to test the lower RAM (addresses 0x80000 through 0x9fffe): 1. All memory locations are written with 0's in ascending (small addresses to large addresses) order. 2. All memory locations are read, expecting 0's, and then all memory is written with 1's in descending order. 3. All memory locations are read, expecting 1's, and then all memory is written with 5's in ascending order. 4. All memory locations are read, expecting 5's, and then all memory is written with A's in descending order.
Hardware Tested:	The lower RAM locations are tested.
Data Returned:	The number of the failing test, the actual data, and the expected data are returned.
Notes:	None

Test Numbers: 5 and 6

Function: These tests verify that the lower RAM and refresh circuitry are functioning properly.

Procedure: All memory locations are read, expecting A's, and then all memory is written with the following patterns in ascending order:

pat(00)=0xF606 pat(08)=0xF603 pat(16)=0x7FB8 pat(24)=0x401B
pat(01)=0xE727 pat(09)=0xD732 pat(17)=0x6E99 pat(25)=0x612A
pat(02)=0xD25A pat(10)=0xA245 pat(18)=0x59F4 pat(26)=0x0A5F
pat(03)=0xC37B pat(11)=0x8374 pat(19)=0x48D5 pat(27)=0x2B6E
pat(04)=0xBAEC pat(12)=0x1988 pat(20)=0x3D41 pat(28)=0xBD96
pat(05)=0xABCD pat(13)=0x38B9 pat(21)=0x2C60 pat(29)=0x9CA7
pat(06)=0x91A2 pat(14)=0x5FC0 pat(22)=0x141F pat(30)=0xE5DC
pat(07)=0x8083 pat(15)=0x7EF1 pat(23)=0x053E pat(31)=0xC4ED

The patterns are written according to the following formula:

memory location(i) = pat(i modulo 32)
where i = {0, 1, 2 ... 32K} where K=1024
a memory location = 16 bits.

All memory locations will be read in ascending order after a 1-second delay to test the refresh circuitry. The refresh rate is each 2.048 milliseconds. The time needed to read or write all 64 kilobytes of memory (16 bits on each read or write) is approximately 20.0 milliseconds (125 nanoseconds per cycle, it takes 5 cycles to perform a read or write). A delay of 1 second, in addition to the 20.0 milliseconds needed to write all memory, is sufficient to test the refresh circuitry. This phase executes from the lower 64 kilobytes of RAM.

Hardware Tested: The lower RAM and refresh circuitry are tested.

Data Returned: The test number that failed, the expected data, and the actual data are returned.

Notes: None

Phase #5 — ROM Check Sum Test

Phase Name:	Cartridge Tape ROM Check Sum (rom) Test
Type:	Demand
Function:	This phase verifies the integrity of the CTC ROM.
Test:	Test 1 — calculates a ROM check sum and compares it to the check sum stored in ROM when the ROM was initially programmed.
Time:	3 seconds
Warnings:	None
Notes:	The stored ROM check sum is offset 16 bytes from the end of the ROM address space.

Phase #5 Test

Test Number:	1
Function:	This test verifies the integrity of the CTC ROM.
Procedure:	A check sum is calculated by reading the CTC ROM. The calculated check sum is compared with the check sum stored in ROM when the ROM was initially programmed.
Hardware Tested:	The CTC ROM is tested.
Data Returned:	The number of the test that failed, the actual check sum value, and the expected check sum value are returned.

Phase #6 — CPU Chip Select Test

Phase Name:	Upper Chip Select Register Tests (cpu_1)
Type:	Demand
Function:	This phase tests the operation of the Upper Memory Chip Select (UMCS), Peripheral Access Chip Select (PACS), Middle Memory Chip Select (MMCS), and Middle Peripheral Chip Select (MPCS) registers of the INTEL 80186 CPU.
Tests:	Tests 1 through 12 — check the UMCS register. Tests 13 through 29 — check the PACS register. Tests 30 through 37 — check the MMCS register. Tests 38 through 48 — check the MPCS register.
Time:	2 seconds
Warnings:	None
Notes:	If any test in this phase fails, the following hardware may be faulty: ■ INTEL 80186 Microprocessor ■ INTEL 80186 Microprocessor interface to the CTC Address/Data bus.

Phase #6 Tests

Test Numbers:	1 through 12
Function:	These tests verify that the UMCS register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The UMCS register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.
Notes:	None

==========

Test Numbers:	13 through 29
Function:	These tests verify that the PACS register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The PACS register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.
Notes:	None

Test Numbers:	30 through 37
Function:	These tests verify that the MMCS register is functional.
Procedure:	A valid data pattern is written to the register, and it is verified with a read.
Hardware Tested:	The MMCS register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.
Notes:	None

==========

Test Numbers:	38 through 48
Function:	These tests verify that the MPCS register is functional.
Procedure:	A valid data pattern is written to the register, and it is verified with a read.
Hardware Tested:	The MPCS register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.
Notes:	None

Phase #7 — CPU DMA Internal Test

Phase Name:	DMA Control Register Tests (cpu_2)
Type:	Demand
Function:	This phase verifies that the internal registers of the INTEL 80186 Direct Memory Access Controller (DMAC) are functioning properly.
Tests:	Tests 1 through 14 — check the DMA0 Control register. Tests 15 through 31 — check the DMA0 Terminal Count register. Tests 32 through 48 — check the DMA0 Destination (low) register. Tests 49 through 53 — check the DMA0 Destination (high) register. Tests 54 through 70 — check the DMA0 Source (low) register. Tests 71 through 75 — check the DMA0 Source (high) register. Tests 76 through 90 — check the DMA1 Control register. Tests 91 through 107 — check the DMA1 Terminal Count register. Tests 108 through 124 — check the DMA1 Destination (low) register. Tests 125 through 141 — check the DMA1 Source (low) register. Tests 142 and 143 — no tests run.
Time:	2 seconds
Warnings:	None
Notes:	If any test in this phase fails, the following hardware may be faulty: ■ INTEL 80186 Microprocessor ■ INTEL 80186 Microprocessor interface to the CTC Address/Data bus.

Phase #7 Tests

Test Numbers:	1 through 14
Function:	These tests verify that the Direct Memory Access 0 (DMA0) Control register is functional.
Procedure:	A valid data pattern is written to the register, and it is verified with a read.
Hardware Tested:	The DMA0 Control register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.
Notes:	None

Test Numbers:	15 through 31
Function:	These tests verify that the DMA0 Terminal Count register is functional.
Procedure:	A valid data pattern is written to the register, and it is verified with a read.
Hardware Tested:	The DMA0 Terminal Count register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.
Notes:	None

==========

Test Numbers:	32 through 48
Function:	These tests verify that the DMA0 Destination (low) register is functional.
Procedure:	A valid data pattern is written to the register, and it is verified with a read.
Hardware Tested:	The DMA0 Destination (low) register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.
Notes:	None

==========

Test Numbers:	49 through 53
Function:	These tests verify that the DMA0 Destination (high) register is functional.
Procedure:	A valid data pattern is written to the register, and it is verified with a read.
Hardware Tested:	The DMA0 Destination (high) register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.
Notes:	None

==========

Test Numbers:	54 through 70
Function:	These tests verify that the DMA0 Source (low) register is functional.
Procedure:	A valid data pattern is written to the register, and it is verified with a read.
Hardware Tested:	The DMA0 Source (low) register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.
Notes:	None

==========

Test Numbers:	71 through 75
Function:	These tests verify that the DMA0 Source (high) register is functional.
Procedure:	A valid data pattern is written to the register, and it is verified with a read.
Hardware Tested:	The DMA0 Source (high) register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.
Notes:	None

Test Numbers: 76 through 90
Function: These tests verify that the DMA1 Control register is functional.
Procedure: A valid data pattern is written to the register, and it is verified with a read.
Hardware Tested: The DMA1 Control register is tested.
Data Returned: The test number that failed, the actual data, and the expected data are returned.
Notes: None

==========

Test Numbers: 91 through 107
Function: These tests verify that the DMA1 Terminal Count register is functional.
Procedure: A valid data pattern is written to the register, and it is verified with a read.
Hardware Tested: The DMA1 Terminal Count register is tested.
Data Returned: The test number that failed, the actual data, and the expected data are returned.
Notes: None

==========

Test Numbers: 108 through 124
Function: These tests verify that the DMA1 Destination (low) register is functional.
Procedure: A valid data pattern is written to the register, and it is verified with a read.
Hardware Tested: The DMA1 Destination (low) register is tested.
Data Returned: The test number that failed, the actual data, and the expected data are returned.
Notes: None

==========

Test Numbers: 125 through 141
Function: These tests verify that the DMA1 Source (low) register is functional.
Procedure: A valid data pattern is written to the register, and it is verified with a read.
Hardware Tested: The DMA1 Source (low) register is tested.
Data Returned: The test number that failed, the actual data, and the expected data are returned.
Notes: None

Phase #8 — CPU Timer Test

Phase Name:	CPU Writable Register Tests (cpu_3)
Type:	Demand
Function:	This phase tests the operation of the internal Timer registers of the INTEL 80186 CPU.
Tests:	Tests 1 through 17 — check the Timer0 Count register. Tests 18 through 34 — check the Timer0 MCA register. Tests 35 through 51 — check the Timer0 MCB register. Tests 52 through 59 — check the Timer0 Mode register. Tests 60 through 77 — check the Timer1 Count register. Tests 78 through 94 — check the Timer1 MCA register. Tests 95 through 111 — check the Timer1 MCB register. Tests 112 through 120 — check the Timer1 Mode register. Tests 121 through 137 — check the Timer2 Count register. Tests 138 through 154 — check the Timer2 MCA register. Tests 155 through 158 — check the Timer2 Mode register.
Time:	2 seconds
Warnings:	None
Notes:	If any test in this phase fails, the following hardware may be faulty: ■ INTEL 80186 Microprocessor ■ INTEL 80186 Microprocessor interface to the CTC Address/Data bus.

Phase #8 Tests

Test Numbers:	1 through 17
Function:	These tests verify that the Timer0 Count register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The Timer0 Count register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.
Notes:	None

Test Numbers:	18 through 34
Function:	These tests verify that the Timer0 Maximum Count A (MCA) register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The Timer0 MCA register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.
Notes:	None

==========

Test Numbers:	35 through 51
Function:	These tests verify that the Timer0 Maximum Count B (MCB) register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The Timer0 MCB register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.
Notes:	None

==========

Test Numbers:	52 through 59
Function:	These tests verify that the Timer0 Mode register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The Timer0 Mode register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.
Notes:	None

==========

Test Numbers:	60 through 77
Function:	These tests verify that the Timer1 Count register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The Timer1 Count register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.
Notes:	None

Test Numbers:	78 through 94
Function:	These tests verify that the Timer1 MCA register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The Timer1 MCA register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.
Notes:	None

==========

Test Numbers:	95 through 111
Function:	These tests verify that the Timer1 MCB register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The Timer1 MCB register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.
Notes:	None

==========

Test Numbers:	112 through 120
Function:	These tests verify that the Timer1 Mode register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The Timer1 Mode register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.
Notes:	None

==========

Test Numbers:	121 through 137
Function:	These tests verify that the Timer2 Count register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The Timer2 Count register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.
Notes:	None

Test Numbers:	138 through 154
Function:	These tests verify that the Timer2 MCA register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The Timer2 MCA register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.
Notes:	None

==========

Test Numbers:	155 through 158
Function:	These tests verify that the Timer2 Mode register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The Timer2 Mode register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.
Notes:	None

Phase #9 — CPU Interrupt Controller Test

Phase Name:	Interrupt Controller Register Test (cpu_4)
Type:	Demand
Function:	This phase verifies that the internal registers of the INTEL 80186 CPU Interrupt Controller (IC) are functioning properly.
Tests:	Tests 1 through 8 — check the IC In-service register. Tests 9 through 11 — check the Interrupt Request register. Tests 12 through 20 — check the Interrupt Mask register. Tests 21 through 23 — check the Interrupt Priority Mask register. Tests 24 through 28 — check the Interrupt Status register. Tests 29 through 40 — check the DMA0 and DMA1 Control registers. Tests 41 through 56 — check the INT0 and INT1 Control registers. Tests 57 through 69 — check the INT2 and INT3 Control registers.
Time:	2 seconds
Warnings:	None
Notes:	If any test in this phase fails, the following hardware may be faulty: ■ INTEL 80186 Microprocessor ■ INTEL 80186 Microprocessor interface to the CTC Address/Data bus.

Phase #9 Tests

Test Numbers:	1 through 8
Function:	These tests verify that the IC In-service register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The IC In-service register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.
Notes:	None

Test Numbers:	9 through 11
Function:	These tests verify that the Interrupt Request register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The Interrupt Request register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.
Notes:	None

==========

Test Numbers:	12 through 20
Function:	These tests verify that the Interrupt Mask register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The Interrupt Mask register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.
Notes:	None

==========

Test Numbers:	21 through 23
Function:	These tests verify that the Interrupt Priority Mask register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The Interrupt Priority Mask register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.
Notes:	None

==========

Test Numbers:	24 through 28
Function:	These tests verify that the Interrupt Status register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The Interrupt Status register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.
Notes:	None

Test Numbers:	29 through 40
Function:	These tests verify that the Direct Memory Access 0 (DMA0) and DMA1 Control registers are functional.
Procedure:	Valid data patterns are written to the registers and verified with a read.
Hardware Tested:	The DMA0 and DMA1 Control registers are tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.
Notes:	None

==========

Test Numbers:	41 through 56
Function:	These tests verify that the Interrupt 0 (INT0) and INT1 Control registers are functional.
Procedure:	Valid data patterns are written to the registers and verified with a read.
Hardware Tested:	The INT0 and INT1 Control registers are tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.
Notes:	None

==========

Test Numbers:	57 through 69
Function:	These tests verify that the INT2 and INT3 Control registers are functional.
Procedure:	Valid data patterns are written to the registers and verified with a read.
Hardware Tested:	The INT2 and INT3 Control registers are tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.
Notes:	None

Phase #10 — CPU Lower Chip Select Test

Phase Name:	Lower Chip Select Register Tests (cpu_5)
Type:	Demand
Function:	This phase verifies that the Lower Memory Chip Select (LMCS) register of the INTEL 80186 CPU is functioning properly.
Tests:	Tests 1 through 12 — check the LMCS register.
Time:	2 seconds
Warnings:	None
Notes:	If any test in this phase fails, the following hardware may be faulty: ■ INTEL 80186 Microprocessor ■ INTEL 80186 Microprocessor interface to the CTC Address/Data bus.

Phase #10 Tests

Test Numbers:	1 through 12
Function:	These tests verify that the LMCS register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The LMCS register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.
Notes:	None

Phase #11 — Programmed Input/Output (PIO) Byte Test

Phase Name: PIO Byte Transfer Test (pio_1)

Type: Demand

Function: This phase tests the ports interface to the 3B2 computer I/O bus.

Tests: Test 1 — tests the data pattern 0x01 at every address in a page of SBD DPDRAM.

Test 2 — tests the data pattern 0x02 at every address in a page of SBD DPDRAM.

Test 3 — tests the data pattern 0x04 at every address in a page of SBD DPDRAM.

Test 4 — tests the data pattern 0x08 at every address in a page of SBD DPDRAM.

Test 5 — tests the data pattern 0x10 at every address in a page of SBD DPDRAM.

Test 6 — tests the data pattern 0x20 at every address in a page of SBD DPDRAM.

Test 7 — tests the data pattern 0x40 at every address in a page of SBD DPDRAM.

Test 8 — tests the data pattern 0x80 at every address in a page of SBD DPDRAM.

Time: 60 seconds

Warnings: None

Notes: This phase performs PIO (write and read) in bytes. If any test in this phase fails, the following hardware may be faulty:

- INTEL 80186 Microprocessor
- INTEL 80186 Microprocessor interface to the CTC Address/Data bus
- CTC interface to the 3B2 computer I/O bus.

Phase #11 Tests

Test Numbers:	1 through 8
Function:	This test verifies the operation of the CTC interface to the 3B2 computer I/O bus.
Procedure:	A one is walked through a field of zeros at every address of a page of System Board (SBD) Dual Port Dynamic Random Access Memory (DPDRAM) and verified with a read.
Hardware Tested:	The CTC interface to the 3B2 computer I/O bus is tested.
Data Returned:	The CTC interface to the 3B2 computer I/O bus is tested. A byte value read from the SBD DPDRAM, a short value read from the SBD DPDRAM, and the SBD DPDRAM failing address are returned.
Notes:	None

Phase #12 — Programmed Input/Output (PIO) Word Test

Phase Name: PIO Word Transfer Test (pio_2)

Type: Demand

Function: This phase tests the CTC interface to the 3B2 computer I/O bus.

Tests: Test 1 — tests the data pattern 0x0001 at every address in a page of SBD DPDRAM.

Test 2 — tests the data pattern 0x0002 at every address in a page of SBD DPDRAM.

Test 3 — tests the data pattern 0x0004 at every address in a page of SBD DPDRAM.

Test 4 — tests the data pattern 0x0008 at every address in a page of SBD DPDRAM.

Test 5 — tests the data pattern 0x0010 at every address in a page of SBD DPDRAM.

Test 6 — tests the data pattern 0x0020 at every address in a page of SBD DPDRAM.

Test 7 — tests the data pattern 0x0040 at every address in a page of SBD DPDRAM.

Test 8 — tests the data pattern 0x0080 at every address in a page of SBD DPDRAM.

Test 9 — tests the data pattern 0x0100 at every address in a page of SBD DPDRAM.

Test 10 — tests the data pattern 0x0200 at every address in a page of SBD DPDRAM.

Test 11 — tests the data pattern 0x0400 at every address in a page of SBD DPDRAM.

Test 12 — tests the data pattern 0x0800 at every address in a page of SBD DPDRAM.

Test 13 — tests the data pattern 0x1000 at every address in a page of SBD DPDRAM.

Test 14 — tests the data pattern 0x2000 at every address in a page of SBD DPDRAM.

Test 15 — tests the data pattern 0x4000 at every address in a page of SBD DPDRAM.

Test 16 — tests the data pattern 0x8000 at every address in a page of SBD DPDRAM.

Time: 60 seconds

Warnings: None

Notes: This phase tests PIO (write and read) in words. If any test in this phase fails, the following hardware may be faulty:

- INTEL 80186 Microprocessor
- INTEL 80186 Microprocessor interface to the CTC Address/Data bus
- CTC interface to the 3B2 computer I/O bus.

Phase #12 Tests

Test Numbers: 1 through 16

Function: These tests verify the operation of the ports interface to the 3B2 computer I/O bus.

Procedure: A one is walked through a field of zeros at every address of a page of System Board (SBD) Dual Port Dynamic Random Access Memory (DPDRAM), and it is verified with a read.

Hardware Tested: The CTC interface to the 3B2 computer I/O bus is tested.

Data Returned: The number of the test that failed, the word value read from the SBD DPDRAM, and the SBD DPDRAM failing address are returned.

Notes: None

Phase #13 — DMA Transfer Byte Test

Phase Name: DMA Byte Transfer Tests (dmabyt)

Type: Demand

Function: This phase diagnoses and reports any errors in the operation of the CTC Direct Memory Access (DMA) channels (byte width).

Tests: Test 1 — checks DMA0 from the SBD DPDRAM to the CTC RAM.

Test 2 — checks DMA0 from the CTC RAM to the SBD DPDRAM.

Test 3 — checks DMA1 from the SBD DPDRAM to the CTC RAM.

Test 4 — checks DMA1 from the CTC RAM to the SBD DPDRAM.

Time: 2 seconds

Warnings: None

Notes: This phase tests DMA transfers in bytes. If any test in this phase fails, the following hardware may be faulty:

- INTEL 80186 Microprocessor
- INTEL 80186 Microprocessor interface to the CTC Address/Data bus
- CTC interface to the 3B2 computer I/O bus.

Phase #13 Tests

Test Number: 1

Function: This test verifies that Direct Memory Access Channel 0 (DMA0) from the System Board (SBD) Dual Port Dynamic Random Access Memory (DPDRAM) to the CTC Random Access Memory (RAM) is functional.

Procedure: Data is written to the SBD DPDRAM. That data is transferred to the CTC RAM and compared for integrity.

Hardware Tested: The CTC DMA0 (INTEL 80186) from the SBD to the CTC card is tested.

Data Returned: The test number that failed, the actual data, and the expected data are returned.

Notes: None

==========

Test Number: 2

Function: This test verifies that DMA0 from the CTC RAM to the SBD DPDRAM is functional.

Procedure: Data is written to the CTC RAM. That data is transferred to the SBD DPDRAM and compared for integrity.

Hardware Tested: The CTC DMA0 (INTEL 80186) from the CTC to the SBD is tested.

Data Returned: The test number that failed, the actual data, and the expected data are returned.

Notes: None

Test Number:	3
Function:	This test verifies that DMA1 from the SBD DPDRAM to the CTC RAM is functional.
Procedure:	Data is written to the SBD DPDRAM. That data is transferred to the CTC RAM and compared for integrity.
Hardware Tested:	The CTC DMA1 (INTEL 80186) from the SBD to the CTC card is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.
Notes:	None

==========

Test Number:	4
Function:	This test verifies that DMA1 from the CTC RAM to the SBD DPDRAM is functional.
Procedure:	Data is written to the CTC RAM. That data is transferred to the SBD DPDRAM and compared for integrity.
Hardware Tested:	The CTC DMA1 (INTEL 80186) from the CTC to the SBD is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.
Notes:	None

Phase #14 — DMA Transfer Word Test

Phase Name: DMA Word Transfer Tests (dmawrd)

Type: Demand

Function: This phase diagnoses and reports any errors in the operation of the CTC Direct Memory Access (DMA) channels (word width).

Tests: Test 1 — checks DMA0 from the SBD DPDRAM to the CTC RAM.

Test 2 — checks DMA0 from the CTC RAM to the SBD DPDRAM.

Test 3 — checks DMA1 from the SBD DPDRAM to the CTC RAM.

Test 4 — checks DMA1 from the CTC RAM to the SBD DPDRAM.

Time: 2 seconds

Warnings: None

Notes: This phase tests DMA transfers in words (16 bits). If any test in this phase fails, the following hardware may be faulty:

- INTEL 80186 Microprocessor
- INTEL 80186 Microprocessor interface to the CTC Address/Data bus
- CTC interface to the 3B2 computer I/O bus.

Phase #14 Tests

Test Number: 1

Function: This test verifies that Direct Memory Access Channel 0 (DMA0) from the System Board (SBD) Dual Port Dynamic Random Access Memory (DPDRAM) to the CTC Random Access Memory (RAM) is functional.

Procedure: Data is written to the SBD DPDRAM. That data is transferred to the CTC RAM and verified for data integrity.

Hardware Tested: The INTEL 80186 DMA0 is tested.

Data Returned: The test number that failed, the actual data, and the expected data are returned.

Notes: None

==========

Test Number: 2

Function: This test verifies that DMA0 from the CTC RAM to the SBD DPDRAM is functional.

Procedure: Data is written to the CTC RAM. That data is transferred to the SBD DPDRAM and verified for integrity.

Hardware Tested: The CTC DMA0 (INTEL 80186) from the CTC to the SBD is tested.

Data Returned: The test number that failed, the actual data, and the expected data are returned.

Notes: None

Test Number:	3
Function:	This test verifies that DMA1 from the SBD DPDRAM to the CTC RAM is functional.
Procedure:	Data is written to the SBD DPDRAM. That data is transferred to the CTC RAM and compared for integrity.
Hardware Tested:	The CTC DMA1 (INTEL 80186) from the SBD to the CTC is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.
Notes:	None

==========

Test Number:	4
Function:	This test verifies that DMA1 from the CTC RAM to the SBD DPDRAM is functional.
Procedure:	Data is written to the CTC RAM. That data is transferred to the SBD DPDRAM and compared for integrity.
Hardware Tested:	The CTC DMA1 (INTEL 80186) from the CTC to the SBD is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.
Notes:	None

Phase #15 — CTC/DMAC Register Test

Phase Name:	CTC/DMAC Register Tests (dmac_tst)
Type:	Demand
Function:	This phase tests the registers of the AMD 9517 Direct Memory Access Controller (DMAC).
Tests:	Tests 1 and 2 — perform pattern tests on the Current Address and Current Word registers, respectively, for Channel 0.
Time:	2 seconds
Warnings:	None
Notes:	None

Phase #15 Tests

Test Numbers:	1 and 2
Function:	These tests perform pattern checks on the Current Address and Current Word registers of the AMD 9517 DMAC for Channel 0.
Procedure:	The byte pointer is cleared in the DMAC. Then, a pattern is loaded for the Channel register under test. The byte pointer is cleared again, and the data is read and confirmed for the channel under test.
Hardware Tested:	The AMD 9517 DMAC chip is tested.
Data Returned:	The number of the test that failed, the actual data, and the expected data are returned.
Notes:	Test 1 checks the Channel 0 Current Address register, and Test 2 checks the Channel 0 Current Word register.

Phase #16 — CTC Tape/DMAC Test

Phase Name:	CTC Tape/DMAC Interface Test (ctc_tst)
Type:	Interactive
Function:	This phase checks the operation of the Direct Memory Access Controller (DMAC), the Floppy Disk Controller, and the cartridge tape drive.
Tests:	Tests 1 through 3 — performs pattern tests on Data register, Track register, and Sector register. Tests 4 through 127 — see Tests 4 through 127 of this phase. Tests 128 through 130 — check the **stepin**, **stepout**, and **step** commands.
Time:	8 to 16 minutes (dependent upon 6- or 12-stream tapes)
Warnings:	The test code assumes that a formatted nonwrite protected cartridge tape is in the tape drive. Any unknown defects may cause a test failure.
Notes:	The test will exercise 12-stream tapes as well as 6-stream tapes.

Phase #16 Tests

Test Numbers:	1 through 3
Function:	These tests pattern check all Floppy Disk Controller registers that can be read from and written to.
Procedure:	Test 1 — pattern tests (0xff, 0xaa, 0x55, 0x00) the Data register. Test 2 — pattern tests the Track register. Test 3 — pattern tests the Sector register.
Hardware Tested:	The WD2797 Floppy Disk Controller/Formatter is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.
Notes:	None

Test Numbers: 4 through 127

Function: See the following procedures.

Procedure:

Test	Test Execution
(4)	Home tape stream 1.
(5)	seek stream 1, segment 0, sector 1.
(6)	Read stream 1, segment 0, sector 1, into buffer.
(7)	Check pdinfo for valid sanity.

This sequence checks Identification (ID) information on each stream and writes a unique pattern to each stream.

(8) Home tape and seek to segment 5, stream 1.
(9) Read ID into buffer (stream 1, segment 5).
(10) Compare ID information with segment number (stream 1, segment 5).
(11) Write pattern onto tape (stream 1, segment 5).

(12) Home tape and seek to segment 5, stream 3.
(13) Read ID into buffer (stream 3, segment 5).
(14) Compare ID information with segment number (stream 3, segment 5).
(15) Write pattern onto tape (stream 3, segment 5).

(16) Home tape and seek to segment 5, stream 5.
(17) Read ID into buffer (stream 5, segment 5).
(18) Compare ID information with segment number (stream 5, segment 5).
(19) Write pattern onto tape (stream 5, segment 5).

(20) Home tape and seek to segment 5, stream 7.
(21) Read ID into buffer (stream 7, segment 5).
(22) Compare ID information with segment number (stream 7, segment 5).
(23) Write pattern onto tape (stream 7, segment 5).

(24) Home tape and seek to segment 5, stream 9.
(25) Read ID into buffer (stream 9, segment 5).
(26) Compare ID information with segment number (stream 9, segment 5).
(27) Write pattern onto tape (stream 9, segment 5).

(28) Home tape and seek to segment 5, stream 11.
(29) Read ID into buffer (stream 11, segment 5).
(30) Compare ID information with segment number (stream 11, segment 5).
(31) Write pattern onto tape (stream 11, segment 5).

(32) Home tape and seek to segment 5, stream 2.
(33) Read ID into buffer (stream 2, segment 5).
(34) Compare ID information with segment number (stream 2, segment 5).
(35) Write pattern onto tape (stream 2, segment 5).

(36) Home tape and seek to segment 5, stream 4.
(37) Read ID into buffer (stream 4, segment 5).
(38) Compare ID information with segment number (stream 4, segment 5).
(39) Write pattern onto tape (stream 4, segment 5).

(40) Home tape and seek to segment 5, stream 6.
(41) Read ID into buffer (stream 6, segment 5).
(42) Compare ID information with segment number (stream 6, segment 5).
(43) Write pattern onto tape (stream 6, segment 5).

(44) Home tape and seek to segment 5, stream 8.
(45) Read ID into buffer (stream 8, segment 5).
(46) Compare ID information with segment number (stream 8, segment 5).
(47) Write pattern onto tape (stream 8, segment 5).

(48) Home tape and seek to segment 5, stream 10.
(49) Read ID into buffer (stream 10, segment 5).
(50) Compare ID information with segment number (stream 10, segment 5).
(51) Write pattern onto tape (stream 10, segment 5).

(52) Home tape and seek to segment 5, stream 12.
(53) Read ID into buffer (stream 12, segment 5).
(54) Compare ID information with segment number (stream 12, segment 5).
(55) Write pattern onto tape (stream 12, segment 5).

After patterns are read, zeros are written/read to each stream. This sequence reads the patterns that have been previously written.

(56) Home tape and seek to stream 12, segment 5.
(57) Read segment into buffer (stream 12, segment 5).
(58) Compare written pattern with read pattern.
(59) Write zeros onto tape (stream 12, segment 5).
(60) Read data from tape into buffer (stream 12, segment 5).
(61) Compare written pattern (zeros) with read pattern.

(62) Home tape and seek to stream 10, segment 5.
(63) Read segment into buffer (stream 10, segment 5).
(64) Compare written pattern with read pattern.
(65) Write zeros onto tape (stream 10, segment 5).
(66) Read data from tape into buffer (stream 10, segment 5).
(67) Compare written pattern (zeros) with read pattern.

(68) Home tape and seek to stream 8, segment 5.
(69) Read segment into buffer (stream 8, segment 5).
(70) Compare written pattern with read pattern.
(71) Write zeros onto tape (stream 8, segment 5).
(72) Read data from tape into buffer (stream 8, segment 5).
(73) Compare written pattern (zeros) with read pattern.

(74) Home tape and seek to stream 6, segment 5.
(75) Read segment into buffer (stream 6, segment 5).
(76) Compare written pattern with read pattern.
(77) Write zeros onto tape (stream 6, segment 5).
(78) Read data from tape into buffer (stream 6, segment 5).
(79) Compare written pattern (zeros) with read pattern.

(80) Home tape and seek to stream 4, segment 5.
(81) Read segment into buffer (stream 4, segment 5).
(82) Compare written pattern with read pattern.
(83) Write zeros onto tape (stream 4, segment 5).
(84) Read data from tape into buffer (stream 4, segment 5).
(85) Compare written pattern (zeros) with read pattern.

(86) Home tape and seek to stream 2, segment 5.
(87) Read segment into buffer (stream 2, segment 5).
(88) Compare written pattern with read pattern.
(89) Write zeros onto tape (stream 2, segment 5).
(90) Read data from tape into buffer (stream 2, segment 5).
(91) Compare written pattern (zeros) with read pattern.

(92) Home tape and seek to stream 11, segment 5.
(93) Read segment into buffer (stream 11, segment 5).
(94) Compare written pattern with read pattern.
(95) Write zeros onto tape (stream 11, segment 5).
(96) Read data from tape into buffer (stream 11, segment 5).
(97) Compare written pattern (zeros) with read pattern.

(98) Home tape and seek to stream 9, segment 5.
(99) Read segment into buffer (stream 9, segment 5).
(100) Compare written pattern with read pattern.
(101) Write zeros onto tape (stream 9, segment 5).
(102) Read data from tape into buffer (stream 9, segment 5).
(103) Compare written pattern (zeros) with read pattern.

(104) Home tape and seek to stream 7, segment 5.
(105) Read segment into buffer (stream 7, segment 5).
(106) Compare written pattern with read pattern.
(107) Write zeros onto tape (stream 7, segment 5).
(108) Read data from tape into buffer (stream 7, segment 5).
(109) Compare written pattern (zeros) with read pattern.

(110) Home tape and seek to stream 5, segment 5.
(111) Read segment into buffer (stream 5, segment 5).
(112) Compare written pattern with read pattern.
(113) Write zeros onto tape (stream 5, segment 5).
(114) Read data from tape into buffer (stream 5, segment 5).
(115) Compare written pattern (zeros) with read pattern.

(116) Home tape and seek to stream 3, segment 5.
(117) Read segment into buffer (stream 3, segment 5).
(118) Compare written pattern with read pattern.
(119) Write zeros onto tape (stream 3, segment 5).
(120) Read data from tape into buffer (stream 3, segment 5).
(121) Compare written pattern (zeros) with read pattern.

(122) Home tape and seek to stream 1, segment 5.
(123) Read segment into buffer (stream 1, segment 5).
(124) Compare written pattern with read pattern.
(125) Write zeros onto tape (stream 1, segment 5).
(126) Read data from tape into buffer (stream 1, segment 5).
(127) Compare written pattern (zeros) with read pattern.

Hardware Tested: The WD2797 Floppy Disk Controller/Formatter is tested.

Data Returned: The failing test number, the actual data, and the expected data are returned.

Notes: None

Test Numbers:	128 through 130
Function:	These tests check the operation of the **stepin**, **stepout**, and **step** commands.
Procedure:	Test 128 — attempts to force a seek to segment 122 to erroneously seek to segment 121. The seek will verify that it stopped at segment 121. A **stepin** command is executed, and it is confirmed that the head is now located over segment 122. Test 129 — attempts to force a seek to segment 5 to erroneously seek to segment 6. A **stepout** command is executed, and it is confirmed that the head is now located over segment 5. Test 130 — issues a **step** command and confirms that the head is now located over segment 4.
Hardware Tested:	The WD2797 Floppy Disk Controller/Formatter is tested.
Data Returned:	The failing test number, the actual data, and the expected data are returned.
Notes:	None

Phase #17 — CTC Floppy/DMAC Test

Phase Name:	CTC Floppy Disk/DMAC Interface Control Test (fic_tst)
Type:	Interactive
Function:	This phase tests the Floppy Interface Control.
Tests:	Tests 1 through 3 — perform pattern checks on all controller registers that can be written and read. Tests 4 and 5 — verify that all Floppy Disk Controller registers can be written to and read from. Tests 6 through 14 — check the ability to write and read back a sector. Tests 15 through 17 — check the **stepin**, **stepout**, and **step** commands.
Time:	25 seconds
Warnings:	The test code assumes that a formatted nonwrite protected diskette is in the floppy disk drive.
Notes:	None

Phase #17 Tests

Test Numbers:	1 through 3
Function:	These tests perform a pattern test on all WD2793 Floppy Disk Controller registers that can be written and read.
Procedure:	Valid data patterns are written to the registers, and they are verified with a read.
Hardware Tested:	The WD2793 Floppy Disk Controller is tested.
Data Returned:	The number of the test that failed, the actual data, and the expected data are returned.
Notes:	Test 1 pattern tests (0xff, 0xaa, 0x55, 0x00) the Data register. Test 2 pattern tests the Track register. Test 3 pattern tests the Sector register.

==========

Test Numbers:	4 and 5
Function:	These tests check that the floppy disk is formatted and that all tracks can be read.
Procedure:	The Identifier (ID) field of each track is read, beginning with track 0 and proceeding to the end of each side, comparing the actual track number read against the expected track number.
Hardware Tested:	The WD2793 Floppy Disk Controller/Formatter is tested.
Data Returned:	The number of the test that failed, the actual data, and the expected data are returned.
Notes:	None

Test Numbers:	6 through 14
Function:	These tests check the ability to write and read back a sector.
Procedure:	Both sides are written with an increasing data pattern starting with the last track of sector 1. The data is read back and verified.
Hardware Tested:	The WD2793 Floppy Disk Controller/Formatter is tested.
Data Returned:	The number of the test that failed, the actual data, and the expected data are returned.
Notes:	None

==========

Test Numbers:	15 through 17
Function:	These tests check the **stepin**, **stepout**, and **step** commands respectively.
Procedure:	1. A move to track 40 is made, and the head movement is verified. Verification is done by the floppy controller. 2. A **step** command is issued, and the head is verified to be over track 41. 3. A **stepin** command is issued, and the head is verified to be over track 42. 4. A **stepout** command is issued, and the head is verified to be over track 41.
Hardware Tested:	The WD2793 Floppy Disk Controller/Formatter is tested.
Data Returned:	The number of the test that failed, the actual data, and the expected data are returned.
Notes:	None

Phase #18 — Fast CTC Tape/DMAC Test

Phase Name:	Fast CTC Tape/DMAC Interface Test (fctc_tst)
Type:	Interactive
Function:	This phase checks the operation of the Direct Memory Access Controller (DMAC), the Floppy Disk Controller, and the cartridge tape drive.
Tests:	See Tests 1 through 52 of this phase.
Time:	4 to 8 minutes
Warnings:	The test code assumes that a formatted nonwrite protected cartridge tape is in the tape drive. Any unknown defects may cause a test failure.
Notes:	This phase does not perform extensive testing of the CTC board or cartridge tape drive. This is a subset of the tests found in Phase 16 of CTC diagnostics.

Phase #18 Tests

Test Numbers:	1 through 3
Function:	These tests perform a pattern test on all WD2793 Floppy Disk Controller registers that can be written and read.
Procedure:	Valid data patterns are written to the registers, and they are verified with a read.
Hardware Tested:	The WD2793 Floppy Disk Controller is tested.
Data Returned:	The number of the test that failed, the actual data, and the expected data are returned.
Notes:	Test 1 pattern tests (0xff, 0xaa, 0x55, 0x00) the Data register. Test 2 pattern tests the Track register. Test 3 pattern tests the Sector register.

Test Numbers: 4 through 52

Function: See the following procedures.

Procedure:

Test	Test Execution
(4)	Home stream 1.
(5)	seek stream 1, segment 0, sector 1.
(6)	Read stream 1, segment 0, sector 1 into buffer.
(7)	Check pdinfo for valid sanity.

This sequence checks Identification (ID) information on each stream.

(8) Home tape and seek to segment 6, stream 1.
(9) Read ID into buffer (stream 1, segment 6).
(10) Compare ID information with segment number (stream 1, segment 6).

(11) Home tape and seek to segment 7, stream 3.
(12) Read ID into buffer (stream 3, segment 7).
(13) Compare ID information with segment number (stream 3, segment 7).

(14) Home tape and seek to segment 8, stream 5.
(15) Read ID into buffer (stream 5, segment 8).
(16) Compare ID information with segment number (stream 5, segment 8).

(17) Home tape and seek to segment 9, stream 7.
(18) Read ID into buffer (stream 7, segment 9).
(19) Compare ID information with segment number (stream 7, segment 9).

(20) Home tape and seek to segment 10, stream 9.
(21) Read ID into buffer (stream 9, segment 10).
(22) Compare ID information with segment number (stream 9, segment 10).

(23) Home tape and seek to segment 11, stream 11.
(24) Read ID into buffer (stream 11, segment 11).
(25) Compare ID information with segment number (stream 11, segment 11).

(26) Home tape and seek to segment 12, stream 2.
(27) Read ID into buffer (stream 2, segment 12).
(28) Compare ID information with segment number (stream 2, segment 12).

(29) Home tape and seek to segment 13, stream 4.
(30) Read ID into buffer (stream 4, segment 13).
(31) Compare ID information with segment number (stream 4, segment 13).

(32) Home tape and seek to segment 14, stream 6.
(33) Read ID into buffer (stream 6, segment 14).
(34) Compare ID information with segment number (stream 6, segment 14).

(35) Home tape and seek to segment 15, stream 8.
(36) Read ID into buffer (stream 8, segment 15).
(37) Compare ID information with segment number (stream 8, segment 15).

(38) Home tape and seek to segment 16, stream 10.
(39) Read ID into buffer (stream 10, segment 16).
(40) Compare ID information with segment number (stream 10, segment 16).

(41) Home tape and seek to segment 17, stream 12.
(42) Read ID into buffer (stream 12, segment 17).
(43) Compare ID information with segment number (stream 12, segment 17).

(44) Home tape and seek to segment 6, stream 2.
(45) Write pattern onto tape (stream 2, segment 6).
(46) Read pattern into buffer (stream 2, segment 6).
(47) Compare pattern with buffer data (stream 2, segment 6).
(48) Write zeros onto tape (stream 2, segment 6).
(49) Read zeros into buffer (stream 2, segment 6).
(50) Compare zeros with buffer data (stream 2, segment 6).

(51) Home and seek to stream 1, segment 122
(52) Seek to segment 121.

Hardware Tested: The WD2797 Floppy Disk Controller/Formatter is tested.

Data Returned: The number of the test that failed, the actual data, and the expected data are returned.

Notes: None

Chapter 6: Intelligent Serial Controller Diagnostics

Introduction

This chapter contains the diagnostic phases and tests for the 3B2 computer Intelligent Serial Controller (ISC) card. Diagnostics test the following components on the ISC card:

- INTEL 80186 Central Processing Unit (CPU)
- Peripheral Control and Status Register (PCSR)
- Random Access Memory (RAM)
- Read Only Memory (ROM)
- Universal Synchronous/Asynchronous Receiver/Transmitter (USART).

An important external component of the ISC card is the 3B2 computer Input/Output (I/O) bus. The ISC card uses the 3B2 computer I/O bus to communicate with the UNIX operating system.

If your 3B2 computer is equipped with an ISC card, the Equipped Device Table (EDT) should list it. You can use the diagnostic monitor command show (DGMON> **s**) to display the EDT. If the ISC card is not listed in the EDT and a VOID or NULL is listed, the following hardware devices on the ISC card may be faulty:

- INTEL 80186 CPU
- ISC ROM
- ISC ID/Vector register
- ISC Address/Data bus
- ISC interface to the 3B2 computer I/O bus.

Eighteen diagnostic phases run tests on all major ISC card components. Phase 1 is a normal diagnostic phase. Phases 2 through 13 are demand diagnostic phases, and Phases 14 through 18 are interactive diagnostic phases.

Phases 14 through 18 require special cabling. Phase 14 is not used for diagnostics on the ISC card. Phase 14 is a debug monitor used by software developers; it also requires a clock source. The cabling information for Phases 14 through 18 is given in their sections of this chapter.

Use the Table of Contents to locate the desired description for each ISC card phase and its associated tests. The phase and test descriptions are organized in numerical order.

Phase Descriptions

Phase #1 — Common I/O and Peripheral Sanity

Phase Name: CIO/Peripheral Sanity Test (cio)

Type: Normal

Function: This phase verifies that the ISC Common I/O (CIO) interface is functioning properly.

Test: Test 1 — verifies that the diagnostic results can be transferred to the 3B2 computer System Board (SBD).

Time: 1 second

Warnings: None

Notes: Phase 1 is the normal phase that checks the basic sanity of the ISC card. All ISC diagnostic phases use a standard procedure to execute their tests and return the results to the SBD. An ISC card is considered insane if it cannot perform the procedure described in Test 1; this procedure runs until completion unless a fault occurs.

All successful diagnostics return the PASS result flag to the Diagnostic Monitor (DGMON). The DGMON then displays ATP (All Tests Passed) on the system console. When a failure occurs, the diagnostics return the FAIL result flag to the DGMON. The DGMON then displays the failing phase number, the actual data, and the expected data. If this phase fails, the following hardware may be faulty:

- INTEL 80186 CPU
- ISC Random Access Memory (RAM)/Read Only Memory (ROM)
- 3B2 computer I/O bus Interface Acknowledge Circuitry (IAC)
- ISC interface to the 3B2 computer I/O bus
- ISC Address/Data bus.

Phase #1 Test

Test Number: 1

Function: This test verifies that the CIO hardware and firmware are functioning properly.

Procedure: This test uses the following standard procedure:

1. The I/O slot number of the ISC card in the 3B2 computer is determined.
2. The serial ports are reset.
3. The sysgen data block is initialized.
4. The serial ports are initialized (sysgen) by sending express [Interrupt 0 (INT0)] and attention (INT1) interrupts sequentially.
5. The INTEL 80186 test code (phase) is downloaded by using the CIO firmware command [Download Memory (DLM)].
6. Execution of the downloaded phase is started by using the CIO firmware command [Force Call to Function (FCF)].
7. A function call to "phasend( ) " is made when the phase is complete. Phasend returns the test results to the SBD.

Hardware Tested: The ISC interface to the 3B2 computer I/O bus is tested.

Data Returned: None

Phase #2 — PCSR Read/Write Test

Phase Name:	PCSR Read/Write Tests (pcsr)
Type:	Demand
Function:	This phase diagnoses and reports any bit interdependency of the ISC Peripheral Control and Status Register (PCSR).
Tests:	Test 1 — clears and reads PCSR bit 0. Test 2 — clears and reads PCSR bit 1. Test 3 — clears and reads PCSR bit 3.
Time:	1 second
Warnings:	None
Notes:	None

Phase #2 Tests

Test Numbers:	1 through 3
Function:	These tests verify the ability to read and write the PCSR.
Procedure:	Each bit is cleared and verified with a read.
Hardware Tested:	The ISC PCSR is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

Phase #3 — Upper RAM Verification Test

Phase Name:	Upper Memory Verification (ram_h)
Type:	Demand
Function:	This phase verifies that the upper ISC RAM is functioning properly.
Tests:	Test 1 — writes 0x0000 to each memory location and verifies with a read. Test 2 — writes 0xffff to each memory location and verifies with a read. Test 3 — writes 0x0000 to each memory location and verifies with a read. Test 4 — writes 0x0101 to each memory location and verifies with a read. Test 5 — writes 0x0202 to each memory location and verifies with a read. Test 6 — writes 0x0404 to each memory location and verifies with a read. Test 7 — writes 0x0808 to each memory location and verifies with a read. Test 8 — writes 0x1010 to each memory location and verifies with a read. Test 9 — writes 0x2020 to each memory location and verifies with a read. Test 10 — writes 0x4040 to each memory location and verifies with a read. Test 11 — writes 0x8080 to each memory location and verifies with a read. Test 12 — writes 0x5555 to each memory location and verifies with a read. Test 13 — writes 0xaaaa to each memory location and verifies with a read. Test 14 — writes 0xdead to each memory location and verifies with a read. Test 15 — writes 0x0000 to each memory location and verifies with a read.
Time:	12 seconds
Warnings:	None
Notes:	If this phase fails, the suspected faulty hardware may be the upper ISC RAM (addresses 0x4000 to end).

Phase #3 Tests

Test Numbers:	1 through 15
Function:	These tests verify that the upper portion of the ISC RAM is functional (from 0x4000 to last memory location).
Procedure:	These tests write known data patterns to every memory location (upper memory) and verify each data pattern with a read.
Hardware Tested:	The ISC RAM (upper portion — in 64-kilobyte banks) is tested.
Data Returned:	The number of the test that failed, the data pattern read, and the data pattern written are returned.

Phase #4 — Lower RAM Verification Test

Phase Name:	Lower Memory Verification (ram_l)
Type:	Demand
Function:	This phase verifies that the lower ISC RAM is functioning properly.
Tests:	Test 1 — writes 0x0000 to each memory location and verifies with a read. Test 2 — writes 0x1111 to each memory location and verifies with a read. Test 3 — writes 0x5555 to each memory location and verifies with a read. Test 4 — writes 0xaaaa to each memory location and verifies with a read. Test 5 — walks a one through a field of zeros in every memory location and verifies with a read. Test 6 — walks a zero through a field of ones in every memory location and verifies with a read.
Time:	15 seconds
Warnings:	None
Notes:	Locations 0x000 through 0x5ff are reserved by the ISC hardware and are therefore not available for testing. If this phase fails, the suspected faulty hardware may be the lower ISC RAM (addresses 0x600 to 0x3fff).

Phase #4 Tests

Test Numbers:	1 through 6
Function:	These tests verify that the lower 16 kilobytes of ISC RAM (0x600 to 0x3fff) are functioning properly.
Procedure:	These tests write known data patterns to every memory location (lower memory) and verify each data pattern with a read.
Hardware Tested:	The ISC RAM (lower portion) is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

Phase #5 — Unique Address Test

Phase Name:	Unique Address Test (addrtst)
Type:	Demand
Function:	This phase verifies ISC RAM addresses by writing, reading, and comparing each unique address as the test data pattern.
Test:	Test 1 — writes the unique address of the target memory location into the memory location itself and verifies by reading the memory location.
Time:	4 seconds
Warnings:	None
Notes:	If this phase fails, the faulty hardware may be the ISC RAM.

Phase #5 Test

Test Number:	1
Function:	This test verifies the the ISC RAM can be addressed.
Procedure:	This test writes the actual address values from 0x3fff to the end of memory as the test data pattern to the location addressed by the values and reads the values back as verification.
Hardware Tested:	The ISC RAM is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

Phase #6 — ROM Check Sum

Phase Name:	Peripheral ROM Check Sum (rom)
Type:	Demand
Function:	This phase verifies the integrity of the ISC ROM.
Test:	Test 1 — calculates a ROM check sum and compares it to the check sum stored in the ROM when the ROM was initially programmed.
Time:	1 second
Notes:	The stored ROM check sum is offset 16 bytes from the end of the ROM address space.

Phase #6 Test

Test Number:	1
Function:	This test verifies the integrity of the ISC ROM.
Procedure:	This test calculates a check sum on the ISC ROM and compares the calculated check sum with the check sum stored in the ROM when it was initially programmed.
Hardware Tested:	The ISC ROM is tested.

Phase #7 — Upper Chip Select Register Test

Phase Name:	Upper Chip Select Register Test (cpu_1)
Type:	Demand
Function:	This phase tests the operation of the Upper Memory Chip Select (UMCS), Peripheral Access Chip Select (PACS), Middle Memory Chip Select (MMCS), and Middle Peripheral Chip Select (MPCS) registers of the INTEL 80186 CPU.
Tests:	Tests 1 through 12 — check the UMCS register. Tests 13 through 29 — check the PACS register. Tests 30 through 37 — check the MMCS register. Tests 38 through 48 — check the MPCS register.
Time:	2 seconds
Warnings:	None
Notes:	If any test in this phase fails, the following hardware areas may be faulty: ■ INTEL 80186 CPU ■ INTEL 80186 CPU interface to the ISC Address/Data bus.

Phase #7 Tests

Test Numbers:	1 through 12
Function:	These tests verify that the UMCS register is functional.
Procedure:	A valid data pattern is written to the register, and it is verified with a read.
Hardware Tested:	The INTEL 80186 CPU UMCS register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	13 through 29
Function:	These tests verify that the PACS register is functional.
Procedure:	A valid data pattern is written to the register, and it is verified with a read.
Hardware Tested:	The INTEL 80186 CPU PACS register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

Test Numbers:	30 through 37
Function:	These tests verify that the MMCS register is functional
Procedure:	A valid pattern is written to the register, and it is verified with a read.
Hardware Tested:	The INTEL 80186 CPU MMCS register is tested.
Data Returned:	The test numbers that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	38 through 48
Function:	These tests verify that the MPCS register is functional.
Procedure:	A valid data pattern is written to the register, and it is verified with a read.
Hardware Tested:	The INTEL 80186 CPU MPCS register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

Phase #8 — DMA Control Registers Test

Phase Name: DMA Control Register Tests (cpu_2)

Type: Demand

Function: This phase verifies that the internal registers of the INTEL 80186 CPU Direct Memory Access Controller (DMAC) are functioning properly.

Tests: Tests 1 through 14 — check the DMA0 Control register.

Tests 15 through 31 — check the DMA0 Terminal Count register.

Tests 32 through 48 — check the DMA0 Destination (low) register.

Tests 49 through 53 — check the DMA0 Destination (high) register.

Tests 54 through 70 — check the DMA0 Source (low) register.

Tests 71 through 75 — no tests run.

Tests 76 through 89 — check the DMA1 Control register.

Tests 90 through 106 — check the DMA1 Terminal Count register.

Tests 107 through 123 — check the DMA1 Destination (low) register.

Tests 124 through 140 — check the DMA1 Source (low) register.

Tests 141 through 143 — no tests run.

Time: 2 seconds

Warnings: None

Notes: If any test in this phase fails, the following hardware areas may be faulty:

- INTEL 80186 CPU
- INTEL 80186 CPU interface to the ISC Address/Data bus.

Phase #8 Tests

Test Numbers: 1 through 14

Function: These tests verify that the Direct Memory Access 0 (DMA0) Control register is functional.

Procedure: A valid data pattern is written to the register, and it is verified with a read.

Hardware Tested: The INTEL 80186 CPU DMA0 Control register is tested.

Data Returned: The test number that failed, the actual data, and the expected data are returned.

Test Numbers: 15 through 31
Function: These tests verify that the DMA0 Terminal Count register is functional.
Procedure: A valid data pattern is written to the register, and it is verified with a read.
Hardware Tested: The INTEL 80186 CPU DMA0 Terminal Count register is tested.
Data Returned: The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers: 32 through 48
Function: These tests verify that the DMA0 Destination register is functional.
Procedure: A valid data pattern is written to the register, and it is verified with a read.
Hardware Tested: The INTEL 80186 CPU DMA0 Destination register is tested.
Data Returned: The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers: 49 through 53
Function: These tests verify that the DMA0 Destination (high) register is functional.
Procedure: A valid data pattern is written to the register, and it is verified with a read.
Hardware Tested: The INTEL 80186 DMA0 Destination (high) register is tested.
Data Returned: The test number that failed, the actual data, and the expected data are returned.
Notes: None

==========

Test Numbers: 54 through 70
Function: These tests verify that the DMA0 Source (low) register is functional.
Procedure: A valid data pattern is written to the register, and it is verified with a read.
Hardware Tested: The INTEL 80186 CPU DMA0 Source register is tested.
Data Returned: The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers: 76 through 89
Function: These tests verify that the DMA1 Control register is functional.
Procedure: A valid data pattern is written to the register and verified with a read.
Hardware Tested: The INTEL 80186 DMA1 Control register is tested.
Data Returned: The test number that failed, the actual data, and the expected data are returned.

Test Numbers:	90 through 106
Function:	These tests verify that the DMA1 Terminal Count register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The INTEL 80186 DMA1 Terminal Count register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	107 through 123
Function:	These tests verify that the DMA1 Destination (low) register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The INTEL 80186 DMA1 Destination (low) register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	124 through 140
Function:	These tests verify that the DMA1 Source (low) register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The INTEL 80186 DMA1 Source (low) register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

Phase #9 — CPU Writable Register Test

Phase Name:	CPU Writable Register Tests (cpu_3)
Type:	Demand
Function:	This phase tests the Timer0 internal registers of the INTEL 80186 CPU.
Tests:	Tests 1 through 17 — check the Timer0 Count register. Tests 18 through 34 — check the Timer0 MCA register. Tests 35 through 51 — check the Timer0 MCB register. Tests 52 through 59 — check the Timer0 Mode register.
Time:	2 seconds
Warnings:	None
Notes:	If any test fails, the following hardware may be faulty: ■ INTEL 80186 CPU ■ INTEL 80186 CPU interface to the ISC Address/Data bus.

Phase #9 Tests

Test Numbers:	1 through 17
Function:	These tests verify that the Timer0 Count register is functional.
Procedure:	A valid data pattern is written to the register, and it is verified with a read.
Hardware Tested:	The INTEL 80186 CPU Timer0 Count register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

Test Numbers:	18 through 34
Function:	These tests verify that the Timer0 Maximum Count A (MCA) register is functional.
Procedure:	A valid data pattern is written to the register, and it is verified with a read.
Hardware Tested:	The INTEL 80186 CPU Timer0 MCA register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	35 through 51
Function:	These tests verify that the Timer0 Maximum Count B (MCB) register is functional.
Procedure:	A valid data pattern is written to the register, and it is verified with a read.
Hardware Tested:	The INTEL 80186 CPU Timer0 MCB register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	52 through 59
Function:	These tests verify that the Timer0 Mode register is functional.
Procedure:	A valid data pattern is written to the register, and it is verified with a read.
Hardware Tested:	The INTEL 80186 CPU Timer0 Mode register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

Phase #10 — Interrupt Control Register Test

Phase Name:	Interrupt Control Register Tests (cpu_4)
Type:	Demand
Function:	This phase verifies that the internal registers of the INTEL 80186 CPU Interrupt Controller (IC) are functioning properly.
Tests:	Tests 1 through 8 — check the IC In-service register. Tests 9 through 11 — check the Interrupt Request register. Tests 12 through 20 — check the Interrupt Mask register. Tests 21 through 24 — check the Interrupt Priority Mask register. Tests 25 through 29 — check the Interrupt Status register. Tests 30 through 33 — check the Interrupt Timer register. Tests 34 through 37 — check the DMA0 Control register. Tests 38 through 41 — check the DMA1 Control register. Tests 42 through 50 — check the INT0 Control register. Tests 51 through 59 — check the INT1 Control register. Tests 60 through 65 — check the INT2 Control register. Tests 66 through 69 — check the INT3 Control register.
Time:	2 seconds
Warnings:	None
Notes:	If any test fails, the following hardware may be faulty: ■ INTEL 80186 CPU ■ INTEL 80186 CPU interface to the ISC Address/Data bus.

Phase #10 Tests

Test Numbers:	1 through 8
Function:	These tests verify that the IC In-service register is functional.
Procedure:	A valid data pattern is written to the register, and it is verified with a read.
Hardware Tested:	The INTEL 80186 CPU IC In-service register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

Test Numbers: 9 through 11
Function: These tests verify that the Interrupt Request register is functional.
Procedure: A valid data pattern is written to the register, and it is verified with a read.
Hardware Tested: The INTEL 80186 CPU Interrupt Request register is tested.
Data Returned: The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers: 12 through 20
Function: These tests verify that the Interrupt Mask register is functional.
Procedure: A valid data pattern is written to the register, and it is verified with a read.
Hardware Tested: The INTEL 80186 CPU Interrupt Mask register is tested.
Data Returned: The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers: 21 through 24
Function: These tests verify that the Interrupt Priority Mask register is functional.
Procedure: A valid data pattern is written to the register, and it is verified with a read.
Hardware Tested: The INTEL 80186 CPU Interrupt Priority Mask register is tested.
Data Returned: The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers: 25 through 29
Function: These tests verify that the Interrupt Status register is functional.
Procedure: A valid data pattern is written to the register, and it is verified with a read.
Hardware Tested: The INTEL 80186 CPU Interrupt Status register is tested.
Data Returned: The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers: 30 through 33
Function: These tests verify that the Interrupt Timer register is functional.
Procedure: A valid data pattern is written to the register, and it is verified with a read.
Hardware Tested: The INTEL 80186 CPU Interrupt Timer register is tested.
Data Returned: The test number that failed, the actual data, and the expected data are returned.

Test Numbers:	34 through 37
Function:	These tests verify that the Direct Memory Access 0 (DMA0) Control register is functional.
Procedure:	A valid data pattern is written to the register, and it is verified with a read.
Hardware Tested:	The INTEL 80186 CPU DMA0 Control register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	38 through 41
Function:	These tests verify that the DMA1 Control register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The INTEL 80186 DMA1 Control register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	42 through 50
Function:	These tests verify that the Interrupt 0 (INT0) Control register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The INTEL 80186 INT0 Control register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	51 through 59
Function:	These tests verify that the INT1 Control register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The INTEL 80186 INT1 Control register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

Test Numbers:	60 through 65
Function:	These tests verify that the INT2 Control register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The INTEL 80186 INT2 Control register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	66 through 69
Function:	These tests verify that the INT3 Control register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The INTEL 80186 INT3 Control register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

Phase #11 — Download/Upload Common I/O Test

Phase Name: Test of Common I/O Download/Uploads Capability (load test)

Function: This phase verifies that data downloaded to the ISC RAM can be successfully uploaded to 3B2 computer memory.

Test: Test 1 — downloads and uploads known data to and from the ISC RAM.

Time: 1 second

Warnings: None

Notes: This test moves a block of data from the 3B2 computer memory to ISC RAM and back to the 3B2 computer memory. If the test fails, the following hardware may be faulty:

- INTEL 80186 CPU
- ISC RAM
- 3B2 computer RAM
- INTEL 80186 CPU interface to the ISC Address/Data bus.

Phase #11 Test

Test Number: 1

Function: This test verifies the download and upload ISC capability.

Procedure: The 3B2 computer resident portion of the diagnostic generates a block of data containing an incrementing pattern. The data block is downloaded into the ISC RAM and is then uploaded into the 3B2 computer memory. The uploaded data block is compared with the original data block downloaded to the ISC RAM. The size of the data block is 0x700.

Hardware Tested: The ISC download and upload firmware is tested.

Data Returned: The block of data originally downloaded into ISC RAM is returned.

Phase #12 — Verify DOS/ZOMBIE Commands Test

Phase Name:	Send Determine Operational Status (DOS)/ZOMBIE Commands to ISC (ciotests)
Type:	Demand
Function:	This phase verifies ISC execution of the DOS and ZOMBIE commands.
Tests:	Test 1 — verifies operation of the DOS ISC commands. Test 2 — verifies operation of the ZOMBIE ISC commands.
Time:	2 seconds
Warnings:	None
Notes:	None

Phase #12 Tests

Test Number:	1
Function:	This test verifies the proper execution of the DOS ISC commands.
Procedure:	The ISC executes the DOS firmware commands and returns the completion code to 3B2 computer resident diagnostic software.
Hardware Tested:	The ISC ROM firmware is tested.
Data Returned:	The execution condition PASS or FAIL is returned.

==========

Test Number:	2
Function:	This test verifies the proper execution of the ZOMBIE ISC commands.
Procedure:	The ISC firmware executes the ZOMBIE commands and returns the completion code to the 3B2 computer resident diagnostic software.
Hardware Tested:	The ISC ROM firmware is tested.
Data Returned:	The execution condition PASS or FAIL is returned.

Phase #13 — External DMA Tests

Phase Name: External DMA Static Tests (extdma)

Type: Demand

Function: This phase checks the external ISC Direct Memory Access Controller (DMAC) chip.

Tests: Test 1 — writes 0x0000 to the DMAC registers; data is verified by performing a read and compare.

Test 2 — writes 0xffff to the DMAC registers; data is verified by performing a read and compare.

Test 3 — writes 0x0000 to the DMAC registers; data is verified by performing a read and compare.

Test 4 — writes 0xaaaa to the DMAC registers; data is verified by performing a read and compare.

Test 5 — writes 0x5555 to the DMAC registers; data is verified by performing a read and compare.

Test 6 — writes a walking ones pattern to the DMAC registers; data is verified by performing a read and compare.

Test 7 — writes a walking zeros pattern to the DMAC registers; data is verified by performing a read and compare.

Time: 3 seconds

Warnings: None

Notes: The ISC card, in addition to the internal INTEL 80186 CPU DMAC, has an external DMAC.

If any test fails, the following hardware may be faulty:

- INTEL 80186 CPU
- External DMAC chip.

Phase #14 — Enter ISC Debug Mode Test

Phase Name: Send Debug Command to ISC

Type: Interactive

Function: This phase is used to enter the ISC ROM resident debug monitor.

Test: Test 1 — issues the debug command to the ISC.

Time: 1 second

Warnings: A terminal must be connected to Channel B of the ISC to run this phase.

Notes: The ISC card contains a debug program in ROM which uses the INTEL 80186 CPU and Channel B of the 8274 Universal Synchronous/Asynchronous Receiver/Transmitter (USART) to control an American Standard Code for Information Interchange (ASCII) terminal. This debug program is useful while developing and testing software developed for Channel A of the ISC. This facility cannot be used to debug code for Channel B.

Since the ISC does not have an internal baud rate generator, the transmit and receive clocks must be provided to Channel B while using the ASCII terminal. The ASCII terminal setup should be for 7-bit word length, 1 stop bit, no parity, and 9600 baud (bits per second). Figure 6-1 shows the ISC to terminal pinout information.

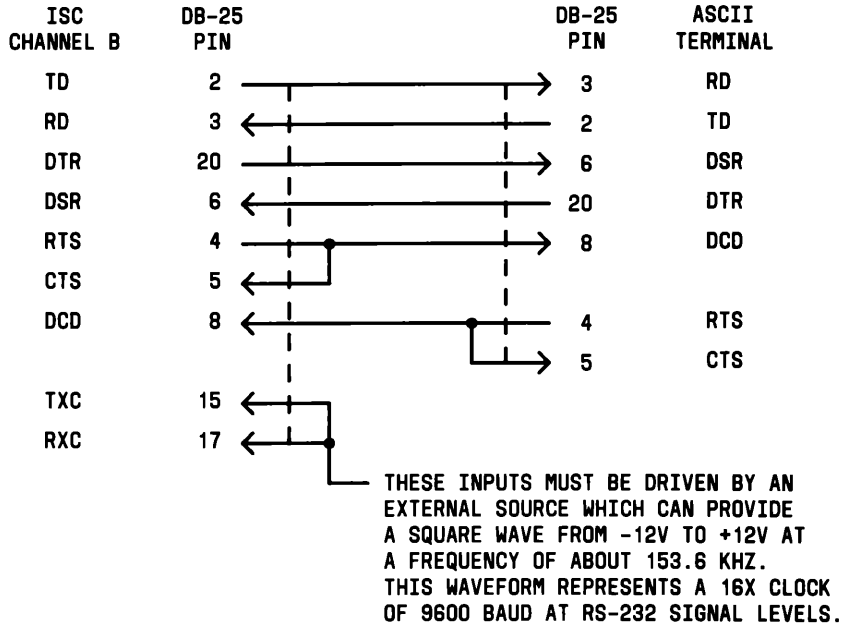


Figure 6-1: ISC to ASCII Terminal Connection for Debug Operation

Phase #14 Test

Test Number:	1
Function:	This test verifies that the ISC can enter the debug mode. Upon recognition of this command, the ISC executes the ROM resident debug monitor code. This facility is primarily a debugging capability. It is assumed that a terminal is attached to Channel B of the ISC as the interface to the debug monitor.
Procedure:	The ISC firmware enters the ROM resident debug monitor when the 3B2 computer resident diagnostic software issues the debug command.
Hardware Tested:	The ISC ROM firmware and ISC Channel B are tested.
Data Returned:	None

Phase #15 — ISC Channel A Loopback Test

Phase Name: ISC External Channel A Loopback Test (iscaloop)

Type: Interactive

Function: This phase verifies data transmission through Channel A of the ISC Universal Synchronous/Asynchronous Receiver/Transmitter (USART). Data from the 3B2 computer memory and the ISC RAM is looped back to the ISC RAM and 3B2 computer memory. The test makes two passes. On the first pass, data is looped back with the ISC NRZI circuitry disabled. On the second pass, the NRZI circuitry is enabled.

Test: Test 1 — loops data from 3B2 computer memory to the ISC.

Time: 4 seconds

Warnings: A special external loopback cable must be attached to Channel A of the ISC to run this phase.

Notes: In this phase , the ISC Channel A is tested in a loopback of the Channel A transmit data line to Channel A receive data line. In order to run this test, you must provide the loopback connection of signals as shown in Figure 6-2.

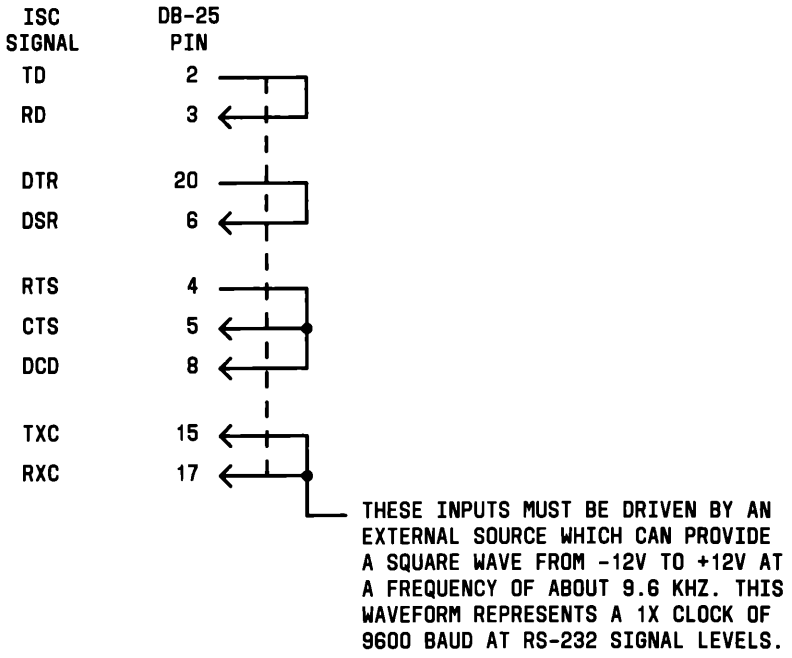


Figure 6-2: ISC Loopback Connection for Phases 15 and 16

Phase #15 Test

Test Number: 1

Function: This test verifies data transmission through Channel A of the ISC USART.

Procedure: A block of data (0x700 bytes) is generated which contains the following data patterns:

0x00	256 bytes
0xff	256 bytes
0xaa	256 bytes
0x55	256 bytes
incrementing pattern	256 bytes
walking ones	256 bytes
alternating bytes of 0x00 and 0xff	256 bytes

Hardware Tested: This test checks the ISC USART. Other major components which have already been verified to a degree are also tested.

If this phase fails, the following hardware may be faulty:

- ISC Channel A USART
- ISC external Direct Memory Access Controller (DMAC) chip
- INTEL 80186 CPU interface to the ISC Address/Data bus
- Missing or faulty loopback cable.

Data Returned: The test number that failed, the actual data, and the expected data are returned.

Additional error conditions returned are as follows:

- Lost DCD or missing loop plug
- Lost CTS
- Overrun error
- CRC error
- Bad Direct Memory Access (DMA) receive terminal count
- Bad DMA transmit terminal count
- ISC receiving incomplete or missing loop plug
- Data miscompare in ISC
- Data miscompare in 3B2 computer.

Phase #16 — ISC Channel B Loopback Test

Phase Name: ISC External Channel B Loopback Test (iscbloop)

Type: Interactive

Function: This phase verifies data transmission through Channel B of the ISC Universal Synchronous/Asynchronous Receiver/Transmitter (USART). Data from the 3B2 computer memory and the ISC RAM is looped back to the ISC RAM and 3B2 computer memory. The test actually consists of two iterations. On the first pass, data is looped back with the ISC NRZI circuitry disabled. On the second pass, the NRZI circuitry is enabled.

Test: Test 1 — loops data from 3B2 computer memory to the ISC.

Time: 4 seconds

Warnings: A special external loopback cable must be attached to Channel B of the ISC to run this phase.

Notes: In this phase, the ISC Channel B is tested in a loopback of Channel B transmit data line to the Channel B receive data line. In order to run this test, you must provide the loopback connection of signals as shown in Figure 6-2.

Phase #16 Test

Test Number: 1

Function: This test verifies data transmission through Channel B of the ISC USART.

Procedure: A block of data (0x700 bytes) is generated which contains the following data patterns:

0x00	256 bytes
0xff	256 bytes
0xaa	256 bytes
0x55	256 bytes
incrementing pattern	256 bytes
walking ones	256 bytes
alternating bytes of 0x00 and 0xff	256 bytes

Hardware Tested: This test checks the ISC USART. Other major components which have already been verified to a degree are also tested.

If this phase fails, the following hardware may be faulty:

- ISC Channel A USART
- ISC external Direct Memory Access Controller (DMAC) chip
- INTEL 80186 CPU interface to the ISC Address/Data bus
- Missing or faulty loopback cable.

Data Returned: The test number that failed, the actual data, and the expected data are returned.

Additional error conditions returned are as follows:

- Lost DCD or missing loop plug
- Lost CTS
- Overrun error
- CRC error
- Bad Direct Memory Access (DMA) receive terminal count
- Bad DMA transmit terminal count
- ISC receiving incomplete or missing loop plug
- Data miscompare in ISC
- Data miscompare in 3B2 computer.

Phase #17 — ISC Channel A-to-B Loopback Test

Phase Name: ISC External Channel A-to-B Loopback Test (iscabloop)

Type: Interactive

Function: This phase verifies data transmission out of Channel A and back to Channel B of the ISC Universal Synchronous/Asynchronous Receiver/Transmitter (USART). Data from the 3B2 computer memory and the ISC RAM is looped back to the ISC RAM and 3B2 computer memory. The test consists of two passes. On the first pass, data is looped back with the ISC circuitry disabled. On the second pass, the ISC circuitry is enabled.

Test: Test 1 — loops data from the 3B2 computer memory to the ISC.

Time: 4 seconds

Warnings: A special external loopback cable must be attached between Channel A and Channel B of the ISC to run this phase.

Notes: In this phase, the ISC Channels A and B are tested with a loopback of the Channel A transmit data line to the Channel B receive data line. In order to run this test, you must provide the loopback connection of signals illustrated in Figure 6-3.

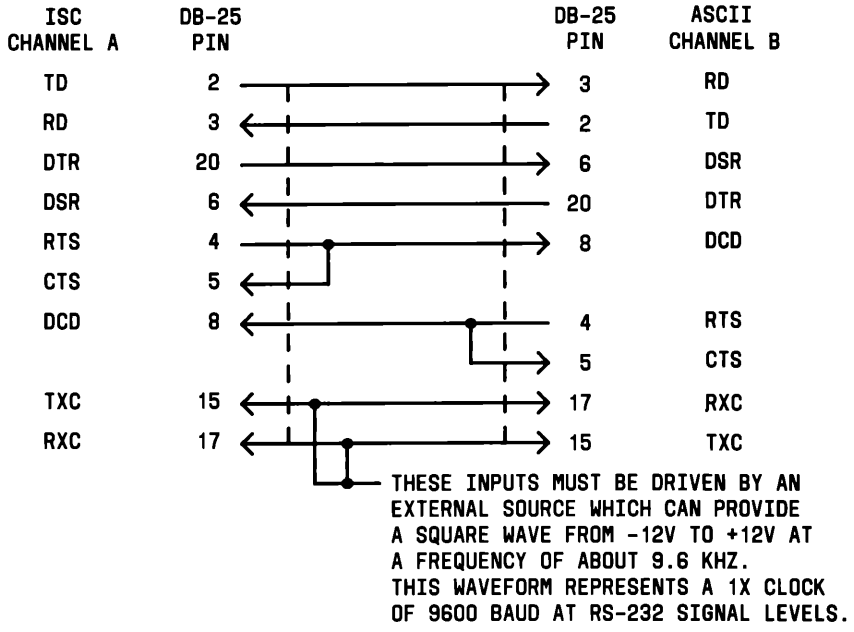


Figure 6-3: ISC Loopback Connection for Phases 17 and 18

Phase #17 Test

Test Number: 1

Function: This test verifies data transmission out of Channel A and back to Channel B of the ISC USART.

Procedure: A block of data (0x700 bytes) is generated which contains the following data patterns:

0x00	256 bytes
0xff	256 bytes
0xaa	256 bytes
0x55	256 bytes
incrementing pattern	256 bytes
walking ones	256 bytes
alternating bytes of 0x00 and 0xff	256 bytes

The data block is accessed from 3B2 computer memory by the diagnostic program downloaded in the ISC. The data block is then moved by Direct Memory Access (DMA) from ISC RAM to the ISC Channel A USART in SDLC format. The test data is returned immediately over the loopback cable to the USART Channel B and ISC RAM. The loopback data is compared to the original data before it is transferred back to 3B2 computer memory where it is again compared to the original data.

Hardware Tested: This test checks the ISC USART. Other major components which have already been verified to a degree are also tested.

If this phase fails, some areas of suspected faulty hardware may be:

- ISC A Channels and B USART
- ISC external Direct Memory Access Controller (DMAC) chip
- INTEL 80186 CPU interface to the ISC Address/Data bus
- Missing or faulty loopback cable.

Data Returned: The test number that failed, the actual data, and the expected data are returned.

Additional error conditions returned are:

- Lost DCD or missing loop plug
- Lost CTS
- Overrun error
- CRC error
- Bad DMA receive terminal count
- Bad DMA transmit terminal count
- ISC receiving incomplete or missing loop plug
- Data miscompare in ISC
- Data miscompare in 3B2 computer.

Phase #18 — ISC Channel B-to-A Loopback Test

Phase Name: ISC External Channel B-to-A Loopback Test (iscbaloop)

Type: Interactive

Function: This phase verifies data transmission out of Channel B and back to Channel A of the ISC Universal Synchronous/Asynchronous Receiver/Transmitter (USART). Data from the 3B2 computer memory and the ISC RAM is looped back to the ISC RAM and 3B2 computer memory. The test consists of two passes. On the first pass, data is looped back with the ISC circuitry disabled. On the second pass, the ISC circuitry is enabled.

Test: Test 1 — loops data from 3B2 computer memory to the ISC.

Time: 4 seconds

Warnings: A special external loopback cable must be attached between Channel A and Channel B of the ISC to run this phase.

Notes: In this phase, the ISC Channels A and B are tested in a loopback of the Channel B transmit data line to the Channel A receive data line. In order to run this test, you must provide the loopback connection of signals illustrated in Figure 6-3.

Phase #18 Test

Test Number: 1

Function: This test verifies data transmission out of Channel B and back to Channel A of the ISC USART. Data from the 3B2 computer memory and the ISC RAM is looped back to the ISC RAM and 3B2 computer memory.

Procedure: A block of data (0x700 bytes) is generated which contains the following data patterns:

0x00	256 bytes
0xff	256 bytes
0xaa	256 bytes
0x55	256 bytes
incrementing pattern	256 bytes
walking ones	256 bytes
alternating bytes of 0x00 and 0xff	256 bytes

The data block is accessed from 3B2 computer memory by the diagnostic program downloaded in the ISC. The data block is then moved by Direct Memory Access (DMA) from the ISC RAM to the ISC Channel B USART in SDLC format. The test data is returned immediately over the loopback cable to the USART Channel A and ISC RAM. The loopback data is compared to the original data before it is transferred back to 3B2 computer memory where it is again compared to the original data.

Hardware Tested: This test checks the ISC USART. Other major components which have already been verified to a degree are also tested.

If this phase fails, the faulty hardware may be:

- ISC Channels A and B
- ISC external Direct Memory Access Controller (DMAC) chip
- INTEL 80186 CPU interface to the ISC Address/Data bus
- Missing or faulty loopback cable.

Data Returned: The test number that failed, the actual data, and the expected data are returned.

Additional error conditions returned are as follows:

- Lost DCD or missing loop plug
- Lost CTS
- Overrun error
- CRC error
- Bad DMA receive terminal count
- Bad DMA transmit terminal count
- ISC receiving incomplete or missing loop plug
- Data miscompare in ISC
- Data miscompare in 3B2 computer.

Chapter 7: Network Interface Diagnostics

Introduction

This chapter contains the diagnostic phases and tests for the 3B2 computer Network Interface (NI) card. The 3B2 computer NI card provides access to AT&T 3BNET, a local computer network that allows file transfer and remote command execution among UNIX system-based machines. The following components are found on the NI card:

- INTEL 80186 Central Processing Unit (CPU)
- Peripheral Control and Status Register (PCSR)
- Random Access Memory (RAM)
- Read Only Memory (ROM)
- INTEL 82586 Ethernet Controller
- INTEL 82501 Serial Interface.

An important external component of the NI card is the 3B2 computer Input/Output (I/O) bus. The NI card uses the 3B2 computer I/O bus to access the System Board (SRD) Dynamic Random Access Memory (DRAM).

If your 3B2 computer is equipped with a NI card, the Equipped Device Table (EDT) should list it. You can use the diagnostic monitor command "show" (DGMON > **s**) to print a copy of the EDT. If the NI card is not listed in the EDT and a VOID or NULL is listed, one or more of the following hardware devices on the NI card may be faulty:

- INTEL 80186 Microprocessor
- NI ROM
- NI ID/Vector register
- NI Address/Data bus.

Nineteen diagnostic phases run tests on the NI card. The Table of Contents listing will help you locate the descriptions for each NI card phase and its associated tests. The phase descriptions are organized numerically in the same order that the phases are run on the 3B2 computer.

Phase Descriptions

Phase #1 — Common I/O Sanity Phase

Phase Name: NI I/O Sanity (cio)

Type: Normal

Function: This phase tests the basic operation of the 3B2 computer NI card Common I/O (CIO) interface.

Test: Test 1 — verifies that the diagnostic results can be transferred to the System Board (SBD).

Time: 1 second

Warnings: None

Notes: This phase and Phase 18 are the normal diagnostic phases that check the basic sanity of the NI card. All NI diagnostic phases use the same procedure to execute their tests and return the results to the SBD. The NI card is considered insane if it cannot follow this procedure.

All successful phases return the PASS result flag to the Diagnostic Monitor (DGMON). The DGMON then displays ATP (All Tests Passed) on the system console. If Phase 1 fails, the following hardware may be faulty:

- INTEL 80186 Microprocessor
- NI RAM/ROM
- Interface between the NI card and the 3B2 computer I/O bus
- NI Address/Data bus.

Phase #1 Test

Test Number: 1

Function: This test verifies that the CIO hardware and firmware are functioning properly.

Procedure: This test uses the following standard procedure:

1. The I/O slot number of the NI card in the 3B2 computer is determined.
2. The NI card is reset.
3. The sysgen data block is initialized.
4. The NI card is initialized (sysgen) by sending express [Interrupt 0 (INT0)] and attention (INT1) interrupts sequentially.
5. The X86 code is downloaded by using the CIO firmware command [Download Memory (DLM)].
6. Execution of the phase is started by using the CIO firmware command [Force Call to Function (FCF)].
7. A function call is made to **"phasend( )"** when the test is complete. Phasend is the routine that responds to the SBD.
8. The PASS result flag is returned to the DGMON. The DGMON then displays ATP (All Tests Passed) on the system console.

Hardware Tested: The interface between the NI card and the 3B2 computer I/O bus is tested.

Data Returned: None

Phase #2 — PCSR Write/Read Test

Phase Name:	NI Control - Status Register (pcsr)
Type:	Demand
Function:	This phase diagnoses and reports any bit interdependency of the NI Peripheral Control and Status Register (PCSR).
Tests:	Test 1 — clears and reads the PCSR bit 0. Test 2 — clears and reads the PCSR bit 1. Test 3 — clears and reads the PCSR bit 3. Test 4 — clears and reads the PCSR bit 4.
Time:	1 second
Warnings:	None
Notes:	None

Phase #2 Tests

Test Numbers:	1 through 4
Function:	These tests diagnose and report any bit interdependency of the PCSR.
Procedure:	A bit is cleared and verified with a read.
Hardware Tested:	The NI PCSR is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

Phase #3 — Upper RAM Write/Read Test

Phase Name:	NI Upper RAM (ram_h)
Type:	Demand
Function:	This phase verifies the operation of the upper half of the NI RAM.
Tests:	Test 1 — writes 0x0000 to each memory location and verifies with a read. Test 2 — writes 0xffff to each memory location and verifies with a read. Test 3 — writes 0x5555 to each memory location and verifies with a read. Test 4 — writes 0xaaaa to each memory location and verifies with a read. Test 5 — walks a one through a field of zeros in every memory location and verifies with a read. Test 6 — walks a zero through a field of ones in every memory location and verifies with a read.
Time:	10 seconds
Warnings:	None
Notes:	None

Phase #3 Tests

Test Numbers:	1 through 6
Function:	These tests verify the operation of the upper half of the NI RAM, addresses 0x4000 to 0x7fff.
Procedure:	A known data pattern is written to every memory location (upper half) and each is verified with a read.
Hardware Tested:	The NI RAM (upper half) is tested.
Data Returned:	The number of the test that failed, the data pattern read, and the data pattern written are returned.

Phase #4 — Lower RAM Write/Read Test

Phase Name:	NI Lower RAM (ram_l)
Type:	Demand
Function:	This phase verifies the operation of the lower half of the NI RAM.
Tests:	Test 1 — writes 0x0000 to each memory location and verifies with a read. Test 2 — writes 0xffff to each memory location and verifies with a read. Test 3 — writes 0x5555 to each memory location and verifies with a read. Test 4 — writes 0xaaaa to each memory location and verifies with a read. Test 5 — walks a one through a field of zeros in every memory location and verifies with a read. Test 6 — walks a zero through a field of ones in every memory location and verifies with a read.
Time:	10 seconds
Warnings:	None
Notes:	None

Phase #4 Tests

Test Numbers:	1 through 6
Function:	These tests verify the operation of the lower half of the NI RAM, addresses 0x0000 to 0x3fff.
Procedure:	A known data pattern is written to every memory location (lower half) and each is verified with a read.
Hardware Tested:	The NI RAM (lower half) is tested.
Data Returned:	The number of the test that failed, the data pattern read, and the data pattern written are returned.

Phase #5 — ROM Check Sum Test

Phase Name:	NI ROM (rom)
Type:	Demand
Function:	This phase verifies the integrity of the NI ROM.
Test:	Test 1 — calculates a ROM check sum and compares it to the check sum stored in ROM when ROM was initially programmed.
Time:	1 second
Warnings:	None
Notes:	None

Phase #5 Test

Test Number:	1
Function:	This test verifies the integrity of the NI ROM.
Procedure:	A check sum is calculated (by reading) on the NI ROM. The calculated check sum is compared with the check sum stored in ROM when ROM was initially programmed.
Hardware Tested:	The NI ROM is tested.
Data Returned:	The number of the test that failed, the expected check sum value, and the actual check sum value are returned.

Phase #6 — CPU Chip Select Test

Phase Name: NI CPU_1 (cpu_1)

Type: Demand

Function: This phase tests the operation of the Upper Memory Chip Select (UMCS), Peripheral Access Chip Select (PACS), and Middle Peripheral Chip Select (MPCS) registers of the INTEL 80186 CPU.

Tests: Tests 1 through 11 — check the UMCS register.

Tests 12 through 28 — check the PACS register.

Tests 29 through 40 — check the MPCS register.

Time: 1 second

Warnings: None

Notes: If any test in this phase fails, the following hardware may be faulty:

- INTEL 80186 Microprocessor
- INTEL 80186 Microprocessor interface to the NI Address/Data bus.

Phase #6 Tests

Test Numbers: 1 through 11

Function: These tests verify that the UMCS register is functional.

Procedure: A valid data pattern is written to the register and verified with a read.

Hardware Tested: The UMCS register is tested.

Data Returned: The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers: 12 through 28

Function: These tests verify that the PACS register is functional.

Procedure: A valid data pattern is written to the register and verified with a read.

Hardware Tested: The PACS register is tested.

Data Returned: The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers: 29 through 40

Function: These tests verify that the MPCS register is functional.

Procedure: A valid data pattern is written to the register and verified with a read.

Hardware Tested: The MPCS register is tested.

Data Returned: The test number that failed, the actual data, and the expected data are returned.

Phase #7 — CPU DMA Internal Test

Phase Name: NI CPU_2 (cpu_2)

Type: Demand

Function: This phase tests the operation of the internal registers of the INTEL 80186 Direct Memory Access Controller (DMAC).

Tests: Tests 1 through 16 — check the DMA0 Terminal Count register.

Tests 17 through 33 — check the DMA0 Destination (low) register.

Tests 34 through 38 — check the DMA0 Destination (high) register.

Tests 39 through 56 — check the DMA0 Source (low) register.

Tests 57 through 61 — check the DMA1 Source (high) register.

Tests 62 through 78 — check the DMA1 Terminal Count register.

Tests 79 through 95 — check the DMA1 Destination (low) register.

Tests 96 through 112 — check the DMA1 Source (low) register.

Tests 113 through 143 — no tests run.

Time: 1 second

Warnings: None

Notes: If any test in this phase fails, the following hardware may be faulty:

- INTEL 80186 Microprocessor
- INTEL 80186 Microprocessor interface to the NI Address/Data bus.

Phase #7 Tests

Test Numbers: 1 through 16

Function: These tests verify that the DMA0 Terminal Count register is functional.

Procedure: A valid data pattern is written to the register and verified with a read.

Hardware Tested: The DMA0 Terminal Count register is tested.

Data Returned: The test number that failed, the actual data, and the expected data are returned.

Test Numbers:	17 through 33
Function:	These tests verify that the DMA0 Destination (low) register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The DMA0 Destination (low) register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	34 through 38
Function:	These tests verify that the DMA0 Destination (high) register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The DMA0 Destination (high) register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	39 through 56
Function:	These tests verify that the DMA0 Source (low) register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The DMA0 Source (low) register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	57 through 61
Function:	These tests verify that the DMA1 Source (high) register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The DMA1 Source (high) register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	62 through 78
Function:	These tests verify that the DMA1 Terminal Count register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The DMA1 Terminal Count register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

Test Numbers:	79 through 95
Function:	These tests verify that the DMA1 Destination (low) register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The DMA1 Destination (low) register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	96 through 112
Function:	These tests verify that the DMA1 Source (low) register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The DMA1 Source (low) register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

Phase #8 — CPU Timer Test

Phase Name: NI CPU_3 (cpu_3)

Type: Demand

Function: This phase tests the operation of the internal Timer registers of the INTEL 80186 CPU.

Tests: Tests 1 through 17 — check the Timer0 Count register.

Tests 18 through 34 — check the Timer0 MCA register.

Tests 35 through 51 — check the Timer0 MCB register.

Tests 52 through 59 — check the Timer0 Mode register.

Tests 60 through 77 — check the Timer1 Count register.

Tests 78 through 94 — check the Timer1 MCA register.

Tests 95 through 111 — check the Timer1 MCB register.

Tests 112 through 120 — check the Timer1 Mode register.

Tests 121 through 137 — check the Timer2 Count register.

Tests 138 through 154 — check the Timer2 MCA register.

Tests 155 through 158 — check the Timer2 Mode register.

Time: 1 second

Warnings: None

Notes: If any test in this phase fails, the following hardware may be faulty:

- INTEL 80186 Microprocessor
- INTEL 80186 Microprocessor interface to the NI Address/Data bus.

Phase #8 Tests

Test Numbers: 1 through 17

Function: These tests verify that the Timer0 Count register is functional.

Procedure: A valid data pattern is written to the register and verified with a read.

Hardware Tested: The Timer0 Count register is tested.

Data Returned: The test number that failed, the actual data, and the expected data are returned.

Test Numbers:	18 through 34
Function:	These tests verify that the Timer0 Maximum Count A (MCA) register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The Timer0 MCA register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	35 through 51
Function:	These tests verify that the Timer0 Maximum Count B (MCB) register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The Timer0 MCB register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	52 through 59
Function:	These tests verify that the Timer0 Mode register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The Timer0 Mode register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	60 through 77
Function:	These tests verify that the Timer1 Count register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The Timer1 Count register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	78 through 94
Function:	These tests verify that the Timer1 MCA register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The Timer1 MCA register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

Test Numbers:	95 through 111
Function:	These tests verify that the Timer1 MCB register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The Timer1 MCB register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	112 through 120
Function:	These tests verify that the Timer1 Mode register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The Timer1 Mode register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	121 through 137
Function:	These tests verify that the Timer2 Count register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The Timer2 Count register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	138 through 154
Function:	These tests verify that the Timer2 MCA register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The Timer2 MCA register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	155 through 158
Function:	These tests verify that the Timer2 Mode register is functional.
Procedure:	A valid pattern is written to the register and verified with a read.
Hardware Tested:	The Timer2 Mode register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

Phase #9 — CPU Interrupt Controller

Phase Name: NI CPU_4 (cpu_4)

Type: Demand

Function: This phase tests the operation of the internal registers of the INTEL 80186 CPU Interrupt Controller (IC).

Tests: Tests 1 through 8 — check the IC In-service register.

Tests 9 through 11 — check the Interrupt Request register.

Tests 12 through 20 — check the Interrupt Mask register.

Tests 21 through 24 — check the Interrupt Priority Mask register.

Tests 25 through 36 — check the DMA0 and DMA1 Control registers.

Tests 37 through 45 — check the INT0 and INT1 Control registers.

Tests 46 through 51 — check the INT2 and INT3 Control registers.

Time: 1 second

Warnings: None

Notes: If any test in this phase fails, the following hardware may be faulty:

- INTEL 80186 Microprocessor
- INTEL 80186 Microprocessor interface to the NI Address/Data bus.

Phase #9 Tests

Test Numbers: 1 through 8

Function: These tests verify that the IC In-service register is functional.

Procedure: A valid data pattern is written to the register and verified with a read.

Hardware Tested: The IC In-service register is tested.

Data Returned: The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers: 9 through 11

Function: These tests verify that the Interrupt Request register is functional.

Procedure: A valid data pattern is written to the register and verified with a read.

Hardware Tested: The Interrupt Request register is tested.

Data Returned: The test number that failed, the actual data, and the expected data are returned.

Test Numbers:	12 through 20
Function:	These tests verify that the Interrupt Mask register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The Interrupt Mask register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	21 through 24
Function:	These tests verify that the Interrupt Priority Mask register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The Interrupt Priority Mask register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	25 through 36
Function:	These tests verify that the Direct Memory Access 0 (DMA0) and DMA1 Control registers are functional.
Procedure:	A valid data pattern is written to the registers and each is verified with a read.
Hardware Tested:	The DMA0 and DMA1 Control registers are tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	37 through 45
Function:	These tests verify that the Interrupt 0 (INT0) and INT1 Control registers are functional.
Procedure:	A valid data pattern is written to the registers and each is verified with a read.
Hardware Tested:	The INT0 and INT1 Control registers are tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	46 through 51
Function:	These tests verify that the INT2 and INT3 Control registers are functional.
Procedure:	A valid data pattern is written to the registers and each is verified with a read.
Hardware Tested:	The INT2 and INT3 Control registers are tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

Phase #10 — CPU Lower Chip Select Test

Phase Name:	NI CPU_5 (cpu_5)
Type:	Demand
Function:	This phase tests the operation of the Lower Memory Chip Select (LMCS) register of the INTEL 80186 CPU.
Tests:	Tests 1 through 12 — check the LMCS register.
Time:	1 second
Warnings:	None
Notes:	If any test in this phase fails, the following hardware may be faulty: ■ INTEL 80186 Microprocessor ■ INTEL 80186 Microprocessor interface to the NI Address/Data bus.

Phase #10 Tests

Test Numbers:	1 through 12
Function:	These tests verify that the LMCS register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The LMCS register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

Phase #11 — Programmed Input/Output (PIO) Byte

Phase Name: NI Byte PIO (pio_1)

Type: Demand

Function: This phase tests the NI card interface to the 3B2 computer I/O bus.

Tests: Test 1 — tests the data pattern 0x01 at every address in a page of SBD DPDRAM.

Test 2 — tests the data pattern 0x02 at every address in a page of SBD DPDRAM.

Test 3 — tests the data pattern 0x04 at every address in a page of SBD DPDRAM.

Test 4 — tests the data pattern 0x08 at every address in a page of SBD DPDRAM.

Test 5 — tests the data pattern 0x10 at every address in a page of SBD DPDRAM.

Test 6 — tests the data pattern 0x20 at every address in a page of SBD DPDRAM.

Test 7 — tests the data pattern 0x40 at every address in a page of SBD DPDRAM.

Test 8 — tests the data pattern 0x80 at every address in a page of SBD DPDRAM.

Time: 40 seconds

Warnings: None

Notes: This phase performs PIO (write and read) in bytes. If any test in this phase fails, the following hardware may be faulty:

- INTEL 80186 Microprocessor
- INTEL 80186 Microprocessor interface to the NI Address/Data bus
- Interface between the NI card and the 3B2 computer I/O bus.

Phase #11 Tests

Test Numbers: 1 through 8

Function: These tests verify the operation of the NI card interface to the 3B2 computer I/O bus.

Procedure: A one is walked through a field of zeros at every address of a page of System Board (SBD) Dual Port Dynamic Random Access Memory (DPDRAM) and verified with a read.

Hardware Tested: The interface between the NI card and the 3B2 computer I/O bus is tested.

Data Returned: A byte value read from SBD DPDRAM, a short value read from SBD DPDRAM, and the SBD DPDRAM failing address are returned.

Phase #12 — Programmed Input/Output (PIO) Word

Phase Name: NI Word PIO (pio_2)

Type: Demand

Function: This phase tests the NI card interface to the 3B2 computer I/O bus.

Tests: Test 1 — tests the data pattern 0x0001 at every address in a page of SBD DPDRAM.

Test 2 — tests the data pattern 0x0002 at every address in a page of SBD DPDRAM.

Test 3 — tests the data pattern 0x0004 at every address in a page of SBD DPDRAM.

Test 4 — tests the data pattern 0x0008 at every address in a page of SBD DPDRAM.

Test 5 — tests the data pattern 0x0010 at every address in a page of SBD DPDRAM.

Test 6 — tests the data pattern 0x0020 at every address in a page of SBD DPDRAM.

Test 7 — tests the data pattern 0x0040 at every address in a page of SBD DPDRAM.

Test 8 — tests the data pattern 0x0080 at every address in a page of SBD DPDRAM.

Test 9 — tests the data pattern 0x0100 at every address in a page of SBD DPDRAM.

Test 10 — tests the data pattern 0x0200 at every address in a page of SBD DPDRAM.

Test 11 — tests the data pattern 0x0400 at every address in a page of 8BD DPDRAM.

Test 12 — tests the data pattern 0x0800 at every address in a page of SBD DPDRAM.

Test 13 — tests the data pattern 0x1000 at every address in a page of SBD DPDRAM.

Test 14 — tests the data pattern 0x2000 at every address in a page of SBD DPDRAM.

Test 15 — tests the data pattern 0x4000 at every address in a page of SBD DPDRAM.

Test 16 — tests the data pattern 0x8000 at every address in a page of SBD DPDRAM.

Time: 40 seconds

Warnings: None

Notes: This phase performs PIO (write and read) in words. If any test in this phase fails, the following hardware may be faulty:

- INTEL 80186 Microprocessor
- INTEL 80186 Microprocessor interface to the NI Address/Data bus
- Interface between the NI card and the 3B2 computer I/O bus.

Phase #12 Tests

Test Numbers: 1 through 16

Function: These tests verify the operation of the NI card interface to the 3B2 computer I/O bus.

Procedure: A one is walked through a field of zeros at every address of a page of System Board (SBD) Dual Port Dynamic Random Access Memory (DPDRAM) and verified with a read.

Hardware Tested: The interface between the NI card and the 3B2 computer I/O bus is tested.

Data Returned: The number of test that failed, a word value read from SBD DPDRAM, and the SBD DPDRAM failing address are returned.

Phase #13 — DMA Transfer Byte Test

Phase Name: NI DMA Transfer Byte (dmabyt)

Type: Demand

Function: This phase diagnoses and reports any errors in the operation of the NI Direct Memory Access (DMA) channels (byte width).

Tests: Test 1 — checks DMA0 from the SBD DPDRAM to the NI RAM.

Test 2 — checks DMA0 from the NI RAM to the SBD DPDRAM.

Test 3 — checks DMA1 from the SBD DPDRAM to the NI RAM.

Test 4 — checks DMA1 from the NI RAM to the SBD DPDRAM.

Time: 1 second

Warnings: None

Notes: This phase does DMA transfers in bytes. If any test in this phase fails, the following hardware may be faulty:

- INTEL 80186 Microprocessor
- INTEL 80186 Microprocessor interface to the NI Address/Data bus
- Interface between the NI card and the 3B2 computer I/O bus.

Phase #13 Tests

Test Number: 1

Function: This test verifies that Direct Memory Access Channel 0 (DMA0) from the System Board (SBD) Dual Port Dynamic Random Access Memory (DPDRAM) to the NI RAM is functional.

Procedure: Data is written [Programmed Input/Output (PIO)] to the SBD DPDRAM. That data is then transferred to the NI RAM, and it is compared for integrity.

Hardware Tested: The NI DMA0 (INTEL 80186) from the SBD to the NI is tested.

Data Returned: The test number that failed, the actual data, and the expected data are returned.

==========

Test Number: 2

Function: This test verifies that DMA0 from the NI RAM to the SBD DPDRAM is functional.

Procedure: Data is written to the NI RAM. That data is then transferred to the SBD DPDRAM and compared for integrity.

Hardware Tested: The NI DMA0 (INTEL 80186) from the NI to the SBD is tested.

Data Returned: The test number that failed, the actual data, and the expected data are returned.

Test Number:	3
Function:	This test verifies that DMA1 from the SBD DPDRAM to the NI RAM is functional.
Procedure:	Data is written [Programmed Input/Output (PIO)] to the SBD DPDRAM. That data is then transferred to the NI RAM and compared for integrity.
Hardware Tested:	The NI DMA1 (INTEL 80186) from the SBD to the NI is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Number:	4
Function:	This test verifies that DMA1 from the NI RAM to the SBD DPDRAM is functional.
Procedure:	Data is written to the NI RAM. That data is transferred to the SBD DPDRAM and compared for integrity.
Hardware Tested:	The NI DMA1 (INTEL 80186) from the NI to the SBD is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

Phase #14 — DMA Transfer Word Test

Phase Name: NI DMA Word (dmawrd)

Type: Demand

Function: This phase diagnoses and reports any errors in the operation of the NI Direct Memory Access (DMA) channels (word width).

Tests: Test 1 — checks DMA0 from the SBD DPDRAM to the NI RAM.

Test 2 — checks DMA0 from the NI RAM to the SBD DPDRAM.

Test 3 — checks DMA1 from the SBD DPDRAM to the NI RAM.

Test 4 — checks DMA1 from the NI RAM to the SBD DPDRAM.

Time: 1 second

Warnings: None

Notes: This phase does DMA transfers in words (16 bits). If any test in this phase fails, the following hardware may be faulty:

- INTEL 80186 Microprocessor
- INTEL 80186 Microprocessor interface to the NI Address/Data bus
- Interface between the NI card and the 3B2 computer I/O bus.

Phase #14 Tests

Test Number: 1

Function: This test verifies that Direct Memory Access Channel 0 (DMA0) from the System Board (SBD) Dual Port Dynamic Random Access Memory (DPDRAM) to the NI RAM is functional.

Procedure: Data is written [Programmed Input/Output (PIO)] to the SBD DPDRAM. That data is then transferred to the NI RAM and compared for integrity.

Hardware Tested: The NI DMA0 (INTEL 80186) from the SBD to the NI is tested.

Data Returned: The test number that failed, the actual data, and the expected data are returned.

==========

Test Number: 2

Function: This test verifies that DMA0 from the NI RAM to the SBD DPDRAM is functional.

Procedure: Data is written to the NI RAM. That data is then transferred to the SBD DPDRAM and compared for integrity.

Hardware Tested: The NI DMA0 (INTEL 80186) from the NI to the SBD is tested.

Data Returned: The test number that failed, the actual data, and the expected data are returned.

Test Number:	3
Function:	This test verifies that DMA1 from the SBD DPDRAM to the NI RAM is functional.
Procedure:	Data is written PIO to the SBD DPDRAM. That data is then transferred to the NI RAM and compared for integrity.
Hardware Tested:	The NI DMA1 (INTEL 80186) from the SBD to the NI is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Number:	4
Function:	This test verifies that DMA1 from the NI RAM to the SBD DPDRAM is functional.
Procedure:	Data is written to the NI RAM. That data is then transferred to the SBD DPDRAM and compared for integrity.
Hardware Tested:	The NI DMA1 (INTEL 80186) from the NI to the SBD is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

Phase #15 — NI Internal Loop (INTEL 82586)

Phase Name: NI Internal Loop (INTEL 82586), 64 Bytes (en_int_1)

Type: Demand

Function: This phase verifies the operation of the NI card INTEL 82586 Ethernet Controller.

Tests: Test 1 — checks the INTEL 82586 Ethernet Controller initialization.

Test 2 — checks the INTEL 82586 Ethernet Controller receive frame area initialization.

Tests 3 through 11 — check the INTEL 82586 Ethernet Controller configuration command.

Tests 12 through 14 — check the INTEL 82586 Ethernet Controller receive activities.

Tests 15 through 24 — check the INTEL 82586 Ethernet Controller packet transmission and status.

Test 25 — checks the INTEL 82586 Ethernet Controller interface to the Address/Data bus.

Test 500 — checks the INTEL 82586 Ethernet Controller interface to the Control bus.

Time: 3 seconds

Warnings: The NI card to the Ethernet transceiver cable must be connected for this phase to run correctly.

Notes: This phase internally loops a 64-byte packet. If any test in this phase fails, the following hardware may be faulty:

- INTEL 82586 Ethernet Controller
- NI card to the Ethernet transceiver cable
- INTEL 80186 CPU and the INTEL 82586 Ethernet Controller Control/Status bus
- INTEL 82586 Ethernet Controller interface to the NI Address/Data bus
- NI RAM/ROM.

Phase #15 Tests

Test Number: 1

Function: This test verifies that the INTEL 82586 Ethernet Controller passes its initialization procedure.

Procedure: The System Configuration Pointer (SCP) and Intermediate System Control Pointer (ISCP) are in NI ROM. The INTEL 80186 CPU sets up the System Control Block (SCB) in NI RAM.

Beginning at address 0xffff6, the SCP specifies both the width of the Data bus used by the INTEL 82586 Ethernet Controller and the location of the ISCP. The ISCP contains the address of the SCB and the Busy bit. The Busy bit is used to indicate that the INTEL 82586 Ethernet Controller is being initialized.

The SCB is a block of memory in NI RAM. The INTEL 80186 CPU uses the SCB to issue commands that are to be executed by the INTEL 82586 Ethernet Controller. The INTEL 82586 Ethernet controller uses the SCB to report the status of command execution.

The initialization procedure is as follows:

1. The CPU sets the ISCP Busy bit, issues a reset to the INTEL 82586 Ethernet Controller, and asserts a Channel Attention (CA) interrupt.
2. The INTEL 82586 Ethernet Controller reads the SCP, determines the bus width, and fetches the ISCP address. From the ISCP, it fetches and stores the SCB address.
3. The INTEL 82586 Ethernet controller then clears the ISCP Busy bit and the SCB command word. It signals an interrupt (LINT21) to the INTEL 80186 CPU and waits for the next CA.

Hardware Tested: The INTEL 82586 Ethernet Controller is tested.

Data Returned: The number of the failing test, the actual value of ISCP Busy bit, and the expected value of ISCP Busy bit are returned.

==========

Test Number: 2

Function: This test verifies that the INTEL 82586 Ethernet Controller receive frame area can be initialized.

Procedure: The INTEL 82586 Ethernet Controller receive frame area is initialized in the NI RAM. This test checks the SCB status line to verify that the initialization procedure was not aborted. The CA can then be serviced.

Hardware Tested: The INTEL 82586 Ethernet Controller is tested.

Data Returned: The number of the failing test, the actual value of the SCB status line, and the expected value of the SCB status line are returned.

Test Numbers: 3 through 11

Function: These tests verify that the INTEL 82586 Ethernet Controller can be configured.

Procedure: Tests 3 through 11 execute as follows:

1. The INTEL 82586 Ethernet Controller configuration command structure is initialized in NI RAM.
2. The SCB command line is checked for any pending commands (Test 3).
3. The configuration command is composed.
4. The CA is sent.
5. The SCB command line is waited on to be cleared which indicates that the configuration command has been completed (Test 4).
6. The SCB status line is waited on to indicate that the configuration command was executed without error (Test 5).
7. The CA is serviced.
8. The dump command structure is initialized in the NI RAM.
9. The SCB command line is checked for any pending commands (Test 6).
10. The dump command is composed.
11. The CA is sent.
12. The SCB command line is checked for dump command completion (Test 7).
13. The SCB status line is checked for any errors in the dump command execution (Test 8).
14. The SCB command line is checked for any pending commands (Test 9).
15. The unique network address is initialized in the NI RAM.
16. The address command is composed.
17. The CA is sent.
18. The SCB command line is checked for address command completion (Test 10).
19. The SCB status line is checked for any address command execution errors (Test 11).

Hardware Tested: The INTEL 82586 Ethernet Controller is tested.

Data Returned: The test number that failed, the actual data, and the expected data are returned.

Test Numbers: 12 through 14

Function: These tests verify that the receive activities of the INTEL 82586 Ethernet Controller can be initiated.

Procedure: Tests 12 through 14 execute as follows:

1. The SCB command line is checked for any pending commands (Test 12).
2. The SCB RUC command is composed which enables the INTEL 82586 Ethernet Controller for frame reception.
3. The CA is sent.
4. The SCB command line is checked for SCB RUC command completion (Test 13).
5. The SCB RUS Ready bit is verified to have been set (Test 14). The RUS Ready bit indicates that the INTEL 82586 Ethernet Controller is ready for frame reception.

Hardware Tested: The INTEL 82586 Ethernet Controller is tested.

Data Returned: The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers: 15 through 24

Function: These tests verify that the INTEL 82586 Ethernet Controller transmits a packet.

Procedure: Tests 15 through 24 execute as follows:

1. The transmit command area is initialized in NI RAM.
2. The Transmit Buffer Descriptor (TBD) is initialized in NI RAM.
3. The Transmit buffer is filled with data in the NI RAM.
4. The SCB command line is checked for any pending commands (Test 15).
5. The transmit command is composed.
6. The CA is sent.
7. The packet is transmitted.
8. The SCB command line is checked for transmit command completion (Test 16).
9. The SCB status line is checked for any errors in the transmit command execution (Test 17).
10. The SCB status line is checked for frame reception (Test 18).
11. The transmit command block is checked for any transmission errors (Test 19).
12. The Receive Frame Descriptor (RFD) status is checked for any errors in frame reception (Test 20).
13. The "C" bit is checked in the RFD for the end of frame reception (Test 21).

14. The Receive Buffer Descriptor (RBD) is checked for the end of the frame (Test 22).
15. The RBD is checked for the use of the Receive buffer (Test 23).
16. The actual byte count of the packet received is checked (Test 24).
17. The CA is serviced.

Hardware Tested: The INTEL 82586 Ethernet Controller is tested.

Data Returned: The test number that failed, the actual data, and the expected data are returned.

==========

Test Number: 25

Function: This test verifies that the INTEL 82586 Ethernet Controller interface to the NI Address/Data bus is functional.

Procedure: A string comparison is performed between the Receive buffer and the Transmit buffer. The CA can then be serviced.

Hardware Tested: The INTEL 82586 Ethernet Controller interface to the NI Address/Data bus is tested.

Data Returned: The test number that failed, the actual data, and the expected data are returned.

==========

Test Number: 500

Function: This test verifies that the INTEL 82586 Ethernet Controller can service CA interrupts.

Procedure: The CA acknowledge flags are checked to determine if they have been cleared.

Hardware Tested: The INTEL 82586 Ethernet Controller is tested.

Data Returned: The number of the failing test is returned, and both the actual and expected values are verified to be dead.

Phase #16 — NI Internal Loop (INTEL 82586)

Phase Name: NI Internal Loop (INTEL 82586), 1500 Bytes (en_int_2)

Type: Demand

Function: This phase verifies the operation of the NI card INTEL 82586 Ethernet Controller.

Tests: Test 1 — checks the INTEL 82586 Ethernet Controller initialization.

Test 2 — checks the INTEL 82586 Ethernet Controller receive frame area initialization.

Tests 3 through 11 — check the INTEL 82586 Ethernet Controller configuration command.

Tests 12 through 14 — check the INTEL 82586 Ethernet Controller receive activities.

Tests 15 through 24 — check the INTEL 82586 Ethernet Controller packet transmission and status.

Test 25 — checks the INTEL 82586 Ethernet Controller interface to the Address/Data bus.

Test 500 — checks the INTEL 82586 Ethernet Controller interface to the Control bus.

Time: 3 seconds

Warnings: The NI card to the Ethernet transceiver cable must be connected for this phase to run correctly.

Notes: This phase internally loops a 1500-byte packet. If any test in this phase fails, the following hardware may be faulty:

- INTEL 82586 Ethernet Controller
- NI card to the Ethernet transceiver cable
- INTEL 80186 CPU and the INTEL 82586 Ethernet Controller Control/Status bus
- INTEL 82586 Ethernet Controller interface to the NI Address/Data bus
- NI RAM/ROM.

Phase #16 Tests

Test Number: 1

Function: This test verifies that the INTEL 82586 Ethernet Controller passes its initialization process.

Procedure: The System Configuration Pointer (SCP) and Intermediate System Control Pointer (ISCP) are in NI ROM. The INTEL 80186 CPU sets up the System Control Block (SCB) in NI RAM.

Beginning at address 0xffff6, the SCP specifies both the width of the Data bus used by the INTEL 82586 Ethernet Controller and the location of the ISCP. The ISCP contains the address of the SCB and the Busy bit. The Busy bit is used to indicate that the INTEL 82586 Ethernet Controller is being initialized.

The SCB is a block of memory in NI RAM. The INTEL 80186 CPU uses the SCB to issue commands that are to be executed by the INTEL 82586 Ethernet Controller. The INTEL 82586 Ethernet Controller uses the SCB to report the status of command execution.

The initialization procedure is as follows:

1. The CPU sets the ISCP Busy bit, issues a reset to the INTEL 82586 Ethernet Controller, and asserts a Channel Attention (CA) interrupt.
2. The INTEL 82586 Ethernet Controller reads the SCP, determines the bus width, and fetches the ISCP address. From the ISCP, it fetches and stores the SCB address.
3. The INTEL 82586 Ethernet Controller then clears the ISCP Busy bit and the SCB command word. It signals an interrupt (LINT21) to the INTEL 80186 CPU and waits for the next CA.

Hardware Tested: The INTEL 82586 Ethernet Controller is tested.

Data Returned: The number of the failing test, the actual value of ISCP Busy bit, and the expected value of ISCP Busy bit are returned.

==========

Test Number: 2

Function: This test verifies that the INTEL 82586 Ethernet Controller receive frame area can be initialized.

Procedure: The INTEL 82586 Ethernet Controller receive frame area is initialized in NI RAM. This test checks the SCB status line to make sure the initialization procedure was not aborted. The CA can then be serviced.

Hardware Tested: The INTEL 82586 Ethernet Controller is tested.

Data Returned: The number of the failing test, the actual value of the SCB status line, and the expected value of the SCB status line are returned.

Test Numbers: 3 through 11

Function: These tests verify that the INTEL 82586 Ethernet Controller can be configured.

Procedure: Tests 3 through 11 execute as follows:

1. The INTEL 82586 Ethernet Controller configuration command structure is initialized in NI RAM.
2. The SCB command line is checked for any pending commands (Test 3).
3. The configuration command is composed.
4. The CA is sent.
5. The SCB command line is waited on to be cleared (Test 4). This indicates that the configuration command has been completed.
6. The SCB status line is waited on to indicate that the configuration command was executed without error (Test 5).
7. The CA is serviced.
8. The dump command structure is initialized in NI RAM.
9. The SCB command line is checked for any pending commands (Test 6).
10. The dump command is composed.
11. The CA is sent.
12. The SCB command line is checked for dump command completion (Test 7).
13. The SCB status line is checked for any errors in dump command execution (Test 8).
14. The SCB command line is checked for any pending commands (Test 9).
15. The unique network address is initialized in the NI RAM.
16. The address command is composed.
17. The CA is sent.
18. The SCB command line is checked for address command completion (Test 10).
19. The SCB status line is checked for any address command execution errors (Test 11).

Hardware Tested: The INTEL 82586 Ethernet Controller is tested.

Data Returned: The test number that failed, the actual data, and the expected data are returned.

Test Numbers: 12 through 14

Function: These tests verify that the receive activities of the INTEL 82586 Ethernet Controller can be initiated.

Procedure: Tests 12 through 14 execute as follows:

1. The SCB command line is checked for any pending commands (Test 12).
2. The SCB RUC command is composed which enables the INTEL 82586 Ethernet Controller for frame reception.
3. The CA is sent.
4. The SCB command line is checked for SCB RUC command completion (Test 13).
5. The SCB RUS Ready bit is checked to be set (Test 14). The RUS Ready bit indicates that the INTEL 82586 Ethernet Controller is ready for frame reception.

Hardware Tested: The INTEL 82586 Ethernet Controller is tested.

Data Returned: The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers: 15 through 24

Function: These tests verify that the INTEL 82586 Ethernet Controller transmits a packet.

Procedure: Tests 15 through 24 execute as follows:

1. The Transmit command area is initialized in NI RAM.
2. The Transmit Buffer Descriptor (TBD) is initialized in NI RAM.
3. The Transmit buffer in NI RAM is filled with data.
4. The SCB command line is checked for any pending commands (Test 15).
5. The transmit command is composed.
6. The CA is sent.
7. The packet is transmitted.
8. The SCB command line is checked for transmit command completion (Test 16).
9. The SCB status line is checked for any errors in the transmit command execution (Test 17).
10. The SCB status line is checked for frame reception (Test 18).
11. The transmit command block is checked for any transmission errors (Test 19).
12. The Receive Frame Descriptor (RFD) status is checked for any errors in frame reception (Test 20).
13. The "C" bit in the RFD is checked for the end of frame reception (Test 21).

14. The Receive Buffer Descriptor (RBD) is checked for the end of the frame (Test 22).
15. The RBD is checked for the use of the Receive buffer (Test 23).
16. The actual byte count of the packet received is checked (Test 24).
17. The CA is serviced.

Hardware Tested: The INTEL 82586 Ethernet Controller is tested.

Data Returned: The test number that failed, the actual data, and the expected data are returned.

==========

Test Number: 25

Function: This test verifies that the INTEL 82586 Ethernet Controller interface to the NI Address/Data bus is functional.

Procedure: A string comparison between the Receive buffer and the Transmit buffer is performed. The CA can then be serviced.

Hardware Tested: The INTEL 82586 Ethernet Controller interface to the NI Address/Data bus is tested.

Data Returned: The test number that failed, the actual data, and the expected data are returned.

==========

Test Number: 500

Function: This test verifies that the INTEL 82586 Ethernet Controller Control/Status bus interface is functional.

Procedure: The CA acknowledge flags are checked to determine if they have been cleared.

Hardware Tested: The INTEL 82586 Ethernet Controller Control/Status bus interface is tested.

Data Returned: The number of the failing test is returned, and the both actual and expected values are checked to be dead.

Phase #17 — NI External Loop (INTEL 82501)

Phase Name: NI External Loop (INTEL 82501) (en_ext_1)

Type: Demand

Function: This phase verifies that the INTEL 82586 Ethernet Controller can externally loop around a packet to the INTEL 82501 Serial Interface.

Tests: Test 1 — checks the INTEL 82586 Ethernet Controller initialization.

Test 2 — checks the INTEL 82586 Ethernet Controller receive frame area initialization.

Tests 3 through 11 — check the INTEL 82586 Ethernet Controller configuration command.

Tests 12 through 14 — check the INTEL 82586 Ethernet Controller receive activities.

Tests 15 through 24 — check the INTEL 82586 Ethernet Controller packet transmission and status.

Test 25 — checks the INTEL 82586 Ethernet Controller interface to the Address/Data bus.

Test 500 — checks the INTEL 82586 Ethernet Controller interface to the Control bus.

Time: 3 seconds

Warnings: The NI card to the Ethernet transceiver cable must be connected.

Notes: This phase externally loops a 12-byte diagnostic packet to the INTEL 82501 Serial Interface. If any test in this phase fails, the following hardware may be faulty:

- INTEL 82586 Ethernet Controller
- INTEL 82501 Serial Interface
- NI card to the Ethernet transceiver cable
- Ethernet transceiver
- INTEL 80186 CPU and the INTEL 82586 Ethernet Controller Control/Status bus
- INTEL 82586 Ethernet Controller interface to the NI Address/Data bus
- NI RAM/ROM.

Phase #17 Tests

Test Number: 1

Function: This test verifies that the INTEL 82586 Ethernet Controller passes its initialization procedure.

Procedure: The System Configuration Pointer (SCP) and Intermediate System Control Pointer (ISCP) are in NI ROM. The INTEL 80186 CPU sets up the System Control Block (SCB) in NI RAM.

Beginning at address 0xffff6, the SCP specifies both the width of the Data bus used by the INTEL 82586 Ethernet Controller and the location of the ISCP. The ISCP contains the address of the SCB and the Busy bit. The Busy bit is used to indicate that the INTEL 82586 Ethernet Controller is being initialized.

The SCB is a block of memory in NI RAM. The INTEL 80186 CPU uses the SCB to issue commands that are to be executed by the INTEL 82586 Ethernet Controller. The INTEL 82586 Ethernet Controller uses the SCB to report the status of command execution.

The initialization procedure is as follows:

1. The CPU sets the ISCP Busy bit, issues a reset to the INTEL 82586 Ethernet Controller, and asserts a Channel Attention (CA) interrupt.
2. The INTEL 82586 Ethernet Controller reads the SCP, determines the bus width, and fetches the ISCP address. From the ISCP, it fetches and stores the SCB address.
3. The INTEL 82586 Ethernet Controller then clears the ISCP Busy bit and the SCB command word. It signals an interrupt (LINT21) to the INTEL 80186 CPU and waits for the next CA.

Hardware Tested: The INTEL 82586 Ethernet Controller is tested.

Data Returned: The number of the failing test, the actual value of ISCP Busy bit , and the expected value of ISCP Busy bit are returned.

==========

Test Number: 2

Function: This test verifies that the INTEL 82586 Ethernet Controller receive frame area can be initialized.

Procedure: The INTEL 82586 Ethernet Controller receive frame area is initialized in NI RAM. This test checks the SCB status line to make sure that the initialization procedure was not aborted. The CA can then be serviced.

Hardware Tested: The INTEL 82586 Ethernet Controller is tested.

Data Returned: The number of the failing test, the actual value of the SCB status line, and the expected value of the SCB status line are returned.

Test Numbers: 3 through 11

Function: These tests verify that the INTEL 82586 Ethernet Controller can be configured.

Procedure: Tests 3 through 11 execute as follows:

1. The INTEL 82586 Ethernet Controller configuration command structure is initialized in NI RAM.
2. The SCB command line is checked for any pending commands (Test 3).
3. The configuration command is composed.
4. The CA is sent.
5. The SCB command line is waited on to be cleared (Test 1). This indicates that the configuration command has been completed.
6. The SCB status line is waited on to indicate that the configuration command was executed without error (Test 5).
7. The CA is serviced.
8. The dump command structure is initialized in NI RAM.
9. The SCB command line is checked for any pending commands (Test 6).
10. The dump command is composed.
11. The CA is sent.
12. The SCB command line is checked for dump command completion (Test 7).
13. The SCB status line is checked for any errors in dump command execution (Test 8).
14. The SCB command line is checked for any pending commands (Test 9).
15. The unique network address is initialized in NI RAM.
16. The address command is composed.
17. The CA is sent.
18. The SCB command line is checked for address command completion (Test 10).
19. The SCB status line is checked for any address command execution errors (Test 11).

Hardware Tested: The INTEL 82586 Ethernet Controller is tested.

Data Returned: The test number that failed, the actual data, and the expected data are returned.

Test Numbers: 12 through 14

Function: These tests verify that the receive activities of the INTEL 82586 Ethernet Controller can be initiated.

Procedure: Tests 12 through 14 execute as follows:

1. The SCB command line is checked for any pending commands (Test 12).
2. The SCB RUC command is composed which enables the INTEL 82586 Ethernet Controller for frame reception.
3. The CA is sent.
4. The SCB command line is checked for SCB RUC command completion (Test 13).
5. The SCB RUS Ready bit is checked to be set (Test 14). The RUS Ready bit indicates that the INTEL 82586 Ethernet Controller is ready for frame reception.

Hardware Tested: The INTEL 82586 Ethernet Controller is tested.

Data Returned: The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers: 15 through 24

Function: These tests verify that the INTEL 82586 Ethernet Controller transmits a packet.

Procedure: Tests 15 through 24 execute as follows:

1. The transmit command area is initialized in NI RAM.
2. The Transmit Buffer Descriptor (TBD) is initialized in NI RAM.
3. The 82501 Serial Interface loopback bit is set.
4. The Transmit buffer is filled with data in NI RAM.
5. The SCB command line is checked for any pending commands (Test 15).
6. The transmit command is composed.
7. The CA is sent.
8. The packet is transmitted.
9. The SCB command line is checked for transmit command completion (Test 16).
10. The SCB status line is checked for any errors in the transmit command execution (Test 17).
11. The SCB status line is checked for frame reception (Test 18).
12. The transmit command block is checked for any transmission errors (Test 19).
13. The Receive Frame Descriptor (RFD) status is checked for any errors in frame reception (Test 20).
14. The "C" bit is checked in the RFD for the end of frame reception (Test 21).

15. The Receive Buffer Descriptor (RBD) is checked for the end of the frame (Test 22).
16. The RBD is checked for the use of the Receive buffer (Test 23).
17. The actual byte count of the packet received is checked (Test 24).
18. The CA is serviced.

Hardware Tested: The INTEL 82586 Ethernet Controller is tested.

Data Returned: The test number that failed, the actual data, and the expected data are returned.

==========

Test Number: 25

Function: This test verifies that the INTEL 82586 Ethernet Controller interface to the NI Address/Data bus is functional.

Procedure: A string comparison between the Receive buffer and the Transmit buffer is performed. The CA can then be serviced.

Hardware Tested: The INTEL 82586 Ethernet Controller interface to the NI Address/Data bus is tested.

Data Returned: The test number that failed, the actual data, and the expected data are returned.

==========

Test Number: 500

Function: This test verifies that the INTEL 82586 Ethernet Control/Status bus interface is functional.

Procedure: The CA acknowledge flags are checked to be cleared.

Hardware Tested: The INTEL 82586 Ethernet Controller Control/Status bus interface is tested.

Data Returned: The number of the failing test is returned, and both the actual and expected values are verified to be dead.

Phase #18 — NI External Loop (XCVR)

Phase Name: NI External Loop (XCVR) (en_ext_2)

Type: Normal

Function: This phase verifies that the INTEL 82586 Ethernet Controller can externally loop around a packet to the Ethernet transceiver.

Tests: Test 1 — checks the INTEL 82586 Ethernet Controller preinitialization.

Test 2 — checks the INTEL 82586 Ethernet Controller initialization.

Tests 3 through 11 — check the INTEL 82586 Ethernet Controller configuration command.

Tests 12 through 14 — check the INTEL 82586 Ethernet Controller receive activities.

Tests 15 through 24 — check the INTEL 82586 Ethernet Controller packet transmission and status.

Test 25 — checks the INTEL 82586 Ethernet Controller interface to the Address/Data bus.

Test 500 — checks the INTEL 82586 Ethernet Controller interface to the Control bus.

Time: 3 seconds

Warnings: The NI card to the Ethernet transceiver cable must be connected for this phase to run correctly.

Notes: This phase loops around a 12-byte diagnostic packet to the Ethernet transceiver. If any test in this phase fails, the following hardware may be faulty:

- INTEL 82586 Ethernet Controller
- INTEL 82501 Serial Interface
- NI card to the Ethernet transceiver cable
- Ethernet transceiver
- INTEL 80186 CPU and the INTEL 82586 Ethernet Controller Control/Status bus
- INTEL 82586 Ethernet Controller interface to the NI Address/Data bus
- NI RAM/ROM.

Phase #18 Tests

Test Number: 1

Function: This test verifies that the INTEL 82586 Ethernet Controller passes its preinitialization process.

Procedure: The System Configuration Pointer (SCP) and Intermediate System Control Pointer (ISCP) are in INTEL 80186 CPU ROM. The INTEL 80186 CPU sets up the System Control Block (SCB) in NI RAM.

Beginning at address 0xffff6, the SCP specifies both the width of the Data bus used by the INTEL 82586 Ethernet Controller and the location of the ISCP. The ISCP contains the address of the SCB and the Busy bit. The Busy bit is used to indicate that the INTEL 82586 Ethernet Controller is being initialized.

The SCB is a block of memory in NI RAM. The INTEL 80186 CPU uses the SCB to issue commands that are to be executed by the INTEL 82586 Ethernet Controller. The INTEL 82586 Ethernet Controller uses the SCB to report the status of command execution.

The initialization procedure is as follows:

1. The CPU sets the ISCP Busy bit, issues a reset to the INTEL 82586 Ethernet Controller, and asserts a Channel Attention (CA) interrupt.
2. The INTEL 82586 Ethernet Controller reads the SCP, determines the bus width, and fetches the ISCP address. From the ISCP, it fetches and stores the SCB address.
3. The INTEL 82586 Ethernet Controller then clears the ISCP Busy bit and the SCB command word. It signals an interrupt (LINT21) to the INTEL 80186 CPU and waits for the next CA.

Hardware Tested: The INTEL 82586 Ethernet Controller is tested.

Data Returned: The test number that failed, the actual data, and the expected data are returned.

==========

Test Number: 2

Function: This test verifies that the INTEL 82586 Ethernet Controller passes its initialization process.

Procedure: The INTEL 82586 Ethernet Controller receive frame area is initialized in NI RAM. This test checks the SCB status line to verify that the initialization procedure was not aborted. The CA can then be serviced.

Hardware Tested: The INTEL 82586 Ethernet Controller is tested.

Data Returned: The test number that failed, the actual data, and the expected data are returned.

Test Numbers: 3 through 11

Function: These tests verify that the INTEL 82586 Ethernet Controller can be configured.

Procedure: Tests 3 through 11 execute as follows:

1. The INTEL 82586 Ethernet Controller configuration command structure is initialized in NI RAM.
2. The SCB command line is checked for any pending commands (Test 3).
3. The configuration command is composed.
4. The CA is sent.
5. The SCB command line is waited on to be cleared (Test 4). This indicates that the configuration command has been completed.
6. The SCB status line is waited on to indicate that the configuration command was executed without error (Test 5).
7. The CA is serviced.
8. The dump command structure is initialized in NI RAM.
9. The SCB command line is checked for any pending commands (Test 6).
10. The dump command is composed.
11. The CA is sent.
12. The SCB command line is checked for dump command completion (Test 7).
13. The SCB status line is checked for any errors in dump command execution (Test 8).
14. The SCB command line is checked for any pending commands (Test 9).
15. The unique network address is initialized in NI RAM.
16. The address command is composed.
17. The CA is sent.
18. The SCB command line is checked for address command completion (Test 10).
19. The SCB status line is checked for any address command execution errors (Test 11).

Hardware Tested: The INTEL 82586 Ethernet Controller is tested.

Data Returned: The test number that failed, the actual data, and the expected data are returned.

Test Numbers: 12 through 14

Function: These tests verify that the receive activities of the INTEL 82586 Ethernet Controller can be initiated.

Procedure: Tests 12 through 14 execute as follows:

1. The SCB command line is checked for any pending commands (Test 12).
2. The SCB RUC command is composed which enables the INTEL 82586 Ethernet Controller for frame reception.
3. The CA is sent.
4. The SCB command line is checked for SCB RUC command completion (Test 13).
5. The SCB RUS Ready bit is checked to be set (Test 14). The RUS Ready bit indicates that the INTEL 82586 Ethernet Controller is ready for frame reception.

Hardware Tested: The INTEL 82586 Ethernet Controller is tested.

Data Returned: The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers: 15 through 24

Function: These tests verify that the INTEL 82586 Ethernet Controller transmits a packet.

Procedure: Tests 15 through 24 execute as follows:

1. The transmit command area is initialized in NI RAM.
2. The Transmit Buffer Descriptor (TBD) is initialized in NI RAM.
3. The Transmit buffer is filled with data in NI RAM.
4. The SCB command line is checked for any pending commands (Test 15).
5. The transmit command is composed.
6. The CA is sent.
7. The packet is transmitted.
8. The SCB command line is checked for transmit command completion (Test 16).
9. The SCB status line is checked for any errors in the transmit command execution (Test 17).
10. The SCB status line is checked for frame reception (Test 18).
11. The transmit command block is checked for any transmission errors (Test 19).
12. The Receive Frame Descriptor (RFD) status is checked for any errors in frame reception (Test 20).
13. The "C" bit in the RFD is checked for the end of frame reception (Test 21).

14. The Receive Buffer Descriptor (RBD) is checked for the end of the frame (Test 22).
15. The RBD is checked for the use of the Receive buffer (Test 23).
16. The actual byte count of the packet received is checked (Test 24).
17. The CA is serviced.

Hardware Tested: The INTEL 82586 Ethernet Controller and the INTEL 82501 Serial Interface are tested.

Data Returned: The test number that failed, the actual data, and the expected data are returned.

==========

Test Number: 25

Function: This test verifies that the INTEL 82586 Ethernet Controller interface to the NI Address/Data bus is functional.

Procedure: A string comparison between the Receive buffer and the Transmit buffer is performed. The CA can then be serviced.

Hardware Tested: The INTEL 82586 Ethernet Controller interface to the NI Address/Data bus is tested.

Data Returned: The test number that failed, the actual data, and the expected data are returned.

==========

Test Number: 500

Function: This test verifies that the INTEL 82586 Ethernet Controller Control/Status bus interface is functional.

Procedure: The CA acknowledge flags are checked to be cleared.

Hardware Tested: The INTEL 82586 Ethernet Controller Control/Status bus interface is tested.

Data Returned: The number of the failing test is returned, and both the actual and expected values are verified to be dead.

Phase #19 — Time Domain Reflectometer

Phase Name:	NI Time Domain Reflectometer (TDR)
Type:	Demand
Function:	This phase diagnoses and reports any errors in the Network Coaxial Media.
Tests:	Tests 1 and 2 — check the INTEL 82586 Ethernet Controller initialization. Test 3 — verifies that the INTEL 82586 Ethernet Controller receive unit is off. Test 4 — verifies that the INTEL 82586 Ethernet Controller is not performing any commands. Test 5 — verifies that the INTEL 82586 Ethernet Controller completed the TDR command. Test 6 — verifies that the TDR command finished within its time limit. Test 7 — determines if a transceiver error was detected. Test 8 — determines if an open was detected on the Coaxial Media.
Time:	6 seconds
Warnings:	The NI card to the transceiver drop cable must be connected for this phase to run properly.
Notes:	None

Phase #19 Tests

Test Number:	1
Function:	This test verifies that the INTEL 82586 Ethernet Controller passed its initialization procedure.
Procedure:	The INTEL 82586 Ethernet Controller initialization structures are initialized. The Busy bit is checked to be set.
Hardware Tested:	The INTEL 82586 Ethernet Controller is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.
Notes:	If this test fails, the INTEL 82586 Ethernet controller could not complete the initialization procedure.

Test Number:	2
Function:	This test verifies that the INTEL 82586 Ethernet Controller passed its initialization procedure.
Procedure:	The status bits (CXCNR) are checked.
Hardware Tested:	The INTEL 82586 Ethernet Controller is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.
Notes:	If this test fails, the INTEL 82586 Ethernet Controller cannot complete the initialization procedure.

==========

Test Number:	3
Function:	This test verifies that the INTEL Ethernet Controller 82586 receive unit is off.
Procedure:	The System Control Block (SCB) RUS bit is verified to be set. If so, the Receive Unit Abort bit is set. A Channel Attention (CA) interrupt is sent. The status is checked to see if the receive unit is shut off.
Hardware Tested:	The INTEL 82586 Ethernet Controller is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.
Notes:	None

==========

Test Number:	4
Function:	This test verifies that the INTEL 82586 Ethernet Controller is not performing any commands.
Procedure:	The SCB command word is checked.
Hardware Tested:	The INTEL 82586 Ethernet Controller is checked.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.
Notes:	None

Test Number:	5
Function:	This test verifies that the INTEL 82586 Ethernet Controller performed the TDR command.
Procedure:	The command block is initialized for the TDR command. The CA interrupt is sent. The SCB command word is checked.
Hardware Tested:	The INTEL 82586 Ethernet Controller and the INTEL 82501 Serial Interface are tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.
Notes:	None

==========

Test Numbers:	6 through 8
Function:	These tests verify that no errors were detected in the Network Coaxial Media.
Procedure:	The completion of the TDR command is checked to be within the allowed time. Transceiver errors are verified not to have been detected. Coaxial opens are verified not to have been detected.
Hardware Tested:	The Network Coaxial Media is tested.
Data Returned:	The test number that failed is returned. The actual value is dead. The expected value for Test 6 is tdr.b. For Tests 7 and 8, the expected value is tdr.time.
Notes:	If Test 8 fails, an open was detected on the media. To determine the distance to the detected open, take the actual data, which is hexadecimal, and convert it to decimal. Then multiply the decimal value by 10 for meters or 32.80 for feet. The final value is the distance to the detected open. Accuracy is within plus or minus 10 meters or 32.80 feet.

Chapter 8: PORTS Diagnostics

Introduction

This chapter contains the diagnostic phases and tests for the 3B2 computer Peripheral Ports Controller (PORTS) card. The PORTS card provides access to the 3B2 computer for peripheral equipment, such as terminals and printers. Each PORTS card contains four serial ports and one parallel port. The following components are found on the PORTS card:

- INTEL 80186 Central Processing Unit (CPU)
- Peripheral Control and Status Register (PCSR)
- Random Access Memory (RAM)
- Read Only Memory (ROM)
- Dual Universal Asynchronous Receiver/Transmitter (DUART).

An important external component of the PORTS card is the 3B2 computer Input/Output (I/O) bus. The PORTS card uses the 3B2 computer I/O bus to communicate with the System Board (SBD).

If your 3B2 computer is equipped with a PORTS card, the Equipped Device Table (EDT) should list it. You can use the diagnostic monitor command "show" (DGMON > **s**) to print a copy of the EDT. If the PORTS card is not listed in the EDT and a VOID or NULL is listed, the following hardware devices on the PORTS card may be faulty:

- INTEL 80186 Microprocessor
- PORTS ROM
- PORTS ID/Vector register
- PORTS Address/Data bus
- PORTS interface to the 3B2 computer I/O bus.

Twenty-two diagnostic phases run tests on all major PORTS card components. The Table of Contents listing will help you locate the descriptions for each PORTS card phase and its associated tests. The phase descriptions are organized numerically in the same order that the phases are run on the 3B2 computer.

There are several additional tests in Phases 17 through 22 which only run on certain versions of the I/O PORTS card. A test is run to see which version of the PORTS card you have. The appropriate tests for the card are then automatically run. These additional tests are identified in this chapter by enclosing the test number in "( )".

Phase Descriptions

Phase #1 — PORTS—Common I/O and Peripheral Sanity

Phase Name: PORTS — CIO and Peripheral Sanity (cio)

Type: Normal

Function: This phase verifies that the PORTS card Common I/O (CIO) interface is functioning properly.

Test: Test 1 — verifies that the diagnostic results can be transferred to the System Board (SBD).

Time: 1 second

Warnings: None

Notes: Phases 1, 21, and 22 are the normal diagnostic phases that check the basic sanity of the PORTS card. All PORTS diagnostic phases use a standard procedure to execute their tests and return the results to the SBD. A PORTS card is considered insane if it cannot follow this procedure. The procedure runs until completion unless a fault occurs.

All successful phases return the PASS result flag to the Diagnostic Monitor (DGMON). The DGMON then prints ATP (All Tests Passed) on the system console. If Phase 1 fails, the following hardware may be faulty:

- INTEL 80186 Microprocessor
- PORTS RAM/ROM
- 3B2 computer I/O bus Interface Acknowledge Circuitry
- PORTS interface to the 3B2 computer I/O bus
- PORTS Address/Data bus.

Phase #1 Test

Test Number: 1

Function: This test verifies that the CIO hardware and firmware are functioning properly.

Procedure: This test uses the following standard procedure:

1. The I/O slot number of the PORTS card in the 3B2 computer is determined.
2. The ports are reset.
3. The sysgen data block is initialized.
4. The ports initialized (sysgen) by sending express [Interrupt 0 (INT0)] and attention (INT1) interrupts sequentially.
5. The X86 code is downloaded by using the CIO firmware command [Download Memory (DLM)].
6. Execution of the phase is started by using the CIO firmware command [Force Call to Function (FCF)].
7. A function call to **"phasend( )"** is made when the phase is complete. Phasend returns the test results to the SBD.

Hardware Tested: The PORTS interface to the 3B2 computer I/O bus is tested.

Data Returned: None

Phase #2 — PORTS—PCSR Read/Write

Phase Name:	PORTS — PCSR Read/Write (pcsr)
Type:	Demand
Function:	This phase diagnoses and reports any bit interdependency of the PORTS Peripheral Control and Status Register (PCSR).
Tests:	Test 1 — clears and reads the PCSR bit 0. Test 2 — clears and reads the PCSR bit 1. Test 3 — clears and reads the PCSR bit 3. Test 4 — clears and reads the PCSR bit 4.
Time:	1 second
Warnings:	None
Notes:	None

Phase #2 Tests

Test Numbers:	1 through 4
Function:	These tests verify the ability to read and write the PCSR.
Procedure:	Each bit is cleared with a write and verified with a read.
Hardware Tested:	The PORTS PCSR is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

Phase #3 — PORTS—Upper RAM Verification

Phase Name:	PORTS — Upper RAM Verification (ram_h)
Type:	Demand
Function:	This phase verifies that the upper 16 kilobytes of PORTS RAM are functioning properly.
Tests:	Test 1 — writes 0x0000 to each memory location and verifies with a read. Test 2 — writes 0x1111 to each memory location and verifies with a read. Test 3 — writes 0x5555 to each memory location and verifies with a read. Test 4 — writes 0xaaaa to each memory location and verifies with a read. Test 5 — walks a one through a field of zeros in every memory location and verifies with a read. Test 6 — walks a zero through a field of ones in every memory location and verifies with a read.
Time:	12 seconds
Warnings:	None
Notes:	None

Phase #3 Tests

Test Numbers:	1 through 6
Function:	These tests verify the operation of the upper 16 kilobytes of PORTS RAM (addresses 0x4000 to 0x7fff).
Procedure:	A known data pattern is written to every memory location (upper half) and verified with a read.
Hardware Tested:	The PORTS RAM (upper half) is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

Phase #4 — PORTS—Lower RAM Verification

Phase Name:	PORTS — Lower RAM Verification (ram_l)
Type:	Demand
Function:	This phase verifies that the lower 16 kilobytes of PORTS RAM are functioning properly.
Tests:	Test 1 — writes 0x0000 to each memory location and verifies with a read. Test 2 — writes 0x1111 to each memory location and verifies with a read. Test 3 — writes 0x5555 to each memory location and verifies with a read. Test 4 — writes 0xaaaa to each memory location and verifies with a read. Test 5 — walks a one through a field of zeros in every memory location and verifies with a read. Test 6 — walks a zero through a field of ones in every memory location and verifies with a read.
Time:	12 seconds
Warnings:	None
Notes:	None

Phase #4 Tests

Test Numbers:	1 through 6
Function:	These tests verify the operation of the lower 16 kilobytes of PORTS RAM (addresses 0x0000 to 0x3fff).
Procedure:	A known data pattern is written to every memory location (lower half) and verified with a read.
Hardware Tested:	The PORTS RAM (lower half) is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

Phase #5 — PORTS—ROM Check Sum

Phase Name:	PORTS — ROM Check Sum (rom)
Type:	Demand
Function:	This phase verifies the integrity of the PORTS ROM.
Test:	Test 1 — calculates a ROM check sum and compares it to the check sum stored in ROM when the ROM was initially programmed.
Time:	3 seconds
Warnings:	None
Notes:	The stored ROM check sum is offset 16 bytes from the end of the ROM address space.

Phase #5 Test

Test Number:	1
Function:	This test verifies the integrity of the PORTS ROM.
Procedure:	A check sum is calculated by reading the PORTS ROM. The calculated check sum is compared with the check sum stored in ROM when the ROM was initially programmed.
Hardware Tested:	The PORTS ROM is tested.
Data Returned:	The number of test that failed, the actual check sum value, and the expected check sum value are returned.

Phase #6 — PORTS—Upper Chip Select Registers

Phase Name:	PORTS — Upper Chip Select Registers (cpu_1)
Type:	Demand
Function:	This phase tests the operation of the Upper Memory Chip Select (UMCS), Peripheral Access Chip Select (PACS), Middle Memory Chip Select (MMCS), and Middle Peripheral Chip Select (MPCS) registers of the INTEL 80186 CPU.
Tests:	Tests 1 through 12 — check the UMCS register. Tests 13 through 29 — check the PACS register. Tests 30 through 37 — check the MMCS register. Tests 38 through 48 — check the MPCS register.
Time:	2 seconds
Warnings:	None
Notes:	If any test in this phase fails, the following hardware may be faulty: ■ INTEL 80186 Microprocessor ■ INTEL 80186 Microprocessor interface to the PORTS Address/Data bus.

Phase #6 Tests

Test Numbers:	1 through 12
Function:	These tests verify that the UMCS register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The UMCS register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	13 through 29
Function:	These tests verify that the PACS register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The PACS register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

Test Numbers:	30 through 37
Function:	These tests verify that the MMCS register is functional
Procedure:	A valid pattern is written to the register, and it is verified with a read.
Hardware Tested:	The MMCS register is tested.
Data Returned:	The test numbers that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	38 through 48
Function:	These tests verify that the MPCS register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The MPCS register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

Phase #7 — PORTS—DMA Control Registers

Phase Name:	PORTS — DMA Control Registers (cpu_2)
Type:	Demand
Function:	This phase verifies that the internal registers of the INTEL 80186 Direct Memory Access Controller (DMAC) are functioning properly.
Tests:	Tests 1 through 14 — check the DMA0 Control register. Tests 15 through 31 — check the DMA0 Terminal Count register. Tests 32 through 48 — check the DMA0 Destination (low) register. Tests 49 through 53 — no tests run. Tests 54 through 70 — check the DMA0 Source (low) register. Tests 71 through 75 — no tests run. Tests 76 through 89 — check the DMA1 Control register. Tests 90 through 106 — check the DMA1 Terminal Count register. Tests 107 through 123 — check the DMA1 Destination (low) register. Tests 124 through 140 — check the DMA1 Source (low) register. Tests 141 through 143 — no tests run.
Time:	2 seconds
Warnings:	None
Notes:	If any test in this phase fails, the following hardware may be faulty: ■ INTEL 80186 Microprocessor ■ INTEL 80186 Microprocessor interface to the PORTS Address/Data bus.

Phase #7 Tests

Test Numbers:	1 through 14
Function:	These tests verify that the Direct Memory Access 0 (DMA0) Control register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The DMA0 Control register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

Test Numbers: 15 through 31
Function: These tests verify that the DMA0 Terminal Count register is functional.
Procedure: A valid data pattern is written to the register and verified with a read.
Hardware Tested: The DMA0 Terminal Count register is tested.
Data Returned: The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers: 32 through 48
Function: These tests verify that the DMA0 Destination (low) register is functional.
Procedure: A valid data pattern is written to the register and verified with a read.
Hardware Tested: The DMA0 Destination (low) register is tested.
Data Returned: The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers: 54 through 70
Function: These tests verify that the DMA0 Source (low) register is functional.
Procedure: A valid data pattern is written to the register and verified with a read.
Hardware Tested: The DMA0 Source (low) register is tested.
Data Returned: The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers: 76 through 89
Function: These tests verify that the DMA1 Control register is functional.
Procedure: A valid data pattern is written to the register and verified with a read.
Hardware Tested: The DMA1 Control register is tested.
Data Returned: The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers: 90 through 106
Function: These tests verify that the DMA1 Terminal Count register is functional.
Procedure: A valid data pattern is written to the register and verified with a read.
Hardware Tested: The DMA1 Terminal Count register is tested.
Data Returned: The test number that failed, the actual data, and the expected data are returned.

Test Numbers:	107 through 123
Function:	These tests verify that the DMA1 Destination (low) register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The DMA1 Destination (low) register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	124 through 140
Function:	These tests verify that the DMA1 Source (low) register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The DMA1 Source (low) register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

Phase #8 — PORTS—CPU Writable Registers

Phase Name:	PORTS — CPU Writable Registers (cpu_3)
Type:	Demand
Function:	This phase tests the operation of the internal Timer registers of the INTEL 80186 CPU.
Tests:	Tests 1 through 17 — check the Timer0 Count register. Tests 18 through 34 — check the Timer0 MCA register. Tests 35 through 51 — check the Timer0 MCB register. Tests 52 through 59 — check the Timer0 Mode register. Tests 60 through 76 — check the Timer1 Count register. Tests 77 through 93 — check the Timer1 MCA register. Tests 94 through 110 — check the Timer1 MCB register. Tests 111 through 118 — check the Timer1 Mode register. Tests 119 through 135 — check the Timer2 Count register. Tests 136 through 152 — check the Timer2 MCA register. Tests 153 through 156 — check the Timer2 Mode register. Tests 157 and 158 — no tests run.
Time:	2 seconds
Warnings:	None
Notes:	If any test in this phase fails, the following hardware may be faulty: ■ INTEL 80186 Microprocessor ■ INTEL 80186 Microprocessor interface to the PORTS Address/Data bus.

Phase #8 Tests

Test Numbers:	1 through 17
Function:	These tests verify that the Timer0 Count register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The Timer0 Count register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

Test Numbers:	18 through 34
Function:	These tests verify that the Timer0 Maximum Count A (MCA) register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The Timer0 MCA register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	35 through 51
Function:	These tests verify that the Timer0 Maximum Count B (MCB) register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The Timer0 MCB register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	52 through 59
Function:	These tests verify that the Timer0 Mode register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The Timer0 Mode register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	60 through 76
Function:	These tests verify that the Timer1 Count register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The Timer1 Count register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	77 through 93
Function:	These tests verify that the Timer1 MCA register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The Timer1 MCA register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

Test Numbers:	94 through 110
Function:	These tests verify that the Timer1 MCB register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The Timer1 MCB register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	111 through 118
Function:	These tests verify that the Timer1 Mode register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The Timer1 Mode register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	119 through 135
Function:	These tests verify that the Timer2 Count register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The Timer2 Count register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	136 through 152
Function:	These tests verify that the Timer2 MCA register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The Timer2 MCA register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	153 through 156
Function:	These tests verify that the Timer2 Mode register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The Timer2 Mode register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

Phase #9 — PORTS—Interrupt Control Registers

Phase Name:	PORTS — Interrupt Control Registers (cpu_4)
Type:	Demand
Function:	This phase verifies that the internal registers of the INTEL 80186 CPU Interrupt Controller (IC) are functioning properly.
Tests:	Tests 1 through 8 — check the IC In-service register. Tests 9 through 11 — check the Interrupt Request register. Tests 12 through 20 — check the Interrupt Mask register. Tests 21 through 24 — check the Interrupt Priority Mask register. Tests 25 through 29 — check the Interrupt Status register. Tests 30 through 33 — check the Interrupt Timer Control register. Tests 34 through 37 — check the DMA0 Control register. Tests 38 through 41 — check the DMA1 Control register. Tests 42 through 50 — check the INT0 Control register. Tests 51 through 59 — check the INT1 Control register. Tests 60 through 65 — check the INT2 Control register. Tests 66 through 69 — check the INT3 Control register.
Time:	2 seconds
Warnings:	None
Notes:	If any test in this phase fails, the following hardware may be faulty: ■ INTEL 80186 Microprocessor ■ INTEL 80186 Microprocessor interface to the PORTS Address/Data bus.

Phase #9 Tests

Test Numbers:	1 through 8
Function:	These tests verify that the IC In-service register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The IC In-service register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

Test Numbers:	9 through 11
Function:	These tests verify that the Interrupt Request register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The Interrupt Request register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	12 through 20
Function:	These tests verify that the Interrupt Mask register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The Interrupt Mask register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	21 through 24
Function:	These tests verify that the Interrupt Priority Mask register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The Interrupt Priority Mask register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	25 through 29
Function:	These tests verify that the Interrupt Status register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The Interrupt Status register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	30 through 33
Function:	These tests verify that the Interrupt Timer Control register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The Interrupt Timer Control register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

Test Numbers:	34 through 37
Function:	These tests verify that the Direct Memory Access 0 (DMA0) Control register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The DMA0 Control register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	38 through 41
Function:	These tests verify that the DMA1 Control register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The DMA1 Control register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	42 through 50
Function:	These tests verify that the Interrupt 0 (INT0) Control register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The INT0 Control register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	51 through 59
Function:	These tests verify that the INT1 Control register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The INT1 Control register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

Test Numbers:	60 through 65
Function:	These tests verify that the INT2 Control register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The INT2 Control register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	66 through 69
Function:	These tests verify that the INT3 Control register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The INT3 Control register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

Phase #10 — PORTS—Lower Chip Select Register

Phase Name:	PORTS — Lower Chip Select Register (cpu_5)
Type:	Demand
Function:	This phase verifies that the Lower Memory Chip Select (LMCS) register of the INTEL 80186 CPU is functioning properly.
Tests:	Tests 1 through 12 — check the LMCS register.
Time:	2 seconds
Warnings:	None
Notes:	If any test in this phase fails, the following hardware may be faulty: ■ INTEL 80186 Microprocessor ■ INTEL 80186 Microprocessor interface to the PORTS Address/Data bus.

Phase #10 Tests

Test Numbers:	1 through 12
Function:	These tests verify that the LMCS register is functional.
Procedure:	A valid data pattern is written to the register and verified with a read.
Hardware Tested:	The LMCS register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

Phase #11 — PORTS—PIO Byte Transfers

Phase Name: PORTS — Programmed Input/Output (PIO) Byte Transfers (pio_1)

Type: Demand

Function: This phase tests the PORTS interface to the 3B2 computer I/O bus.

Tests: Test 1 — tests the data pattern 0x01 at every address in a page of SBD DPDRAM.

Test 2 — tests the data pattern 0x02 at every address in a page of SBD DPDRAM.

Test 3 — tests the data pattern 0x04 at every address in a page of SBD DPDRAM.

Test 4 — tests the data pattern 0x08 at every address in a page of SBD DPDRAM.

Test 5 — tests the data pattern 0x10 at every address in a page of SBD DPDRAM.

Test 6 — tests the data pattern 0x20 at every address in a page of SBD DPDRAM.

Test 7 — tests the data pattern 0x40 at every address in a page of SBD DPDRAM.

Test 8 — tests the data pattern 0x80 at every address in a page of SBD DPDRAM.

Time: 40 seconds

Warnings: None

Notes: This phase performs PIO (write and read) in bytes. If any test in this phase fails, the following hardware may be faulty:

- INTEL 80186 Microprocessor
- INTEL 80186 Microprocessor interface to the PORTS Address/Data bus
- PORTS interface to the 3B2 computer I/O bus.

Phase #11 Tests

Test Numbers:	1 through 8
Function:	These tests verify the operation of the PORTS interface to the 3B2 computer I/O bus.
Procedure:	A one is walked through a field of zeros at every address of a page of System Board (SBD) Dual Port Dynamic Random Access Memory (DPDRAM) and verified with a read.
Hardware Tested:	The PORTS interface to the 3B2 computer I/O bus is tested.
Data Returned:	A byte value read from the SBD DPDRAM, a short value read from SBD DPDRAM, and the SBD DPDRAM failing address are returned.

Phase #12 — PORTS—PIO Word Transfers

Phase Name: PORTS — Programmed Input/Output (PIO) Word Transfers (pio_2)

Type: Demand

Function: This phase tests the PORTS interface to the 3B2 computer I/O bus.

Tests: Test 1 — tests the data pattern 0x0001 at every address in a page of SBD DPDRAM.

Test 2 — tests the data pattern 0x0002 at every address in a page of SBD DPDRAM.

Test 3 — tests the data pattern 0x0004 at every address in a page of SBD DPDRAM.

Test 4 — tests the data pattern 0x0008 at every address in a page of SBD DPDRAM.

Test 5 — tests the data pattern 0x0010 at every address in a page of SBD DPDRAM.

Test 6 — tests the data pattern 0x0020 at every address in a page of SBD DPDRAM.

Test 7 — tests the data pattern 0x0040 at every address in a page of SBD DPDRAM.

Test 8 — tests the data pattern 0x0080 at every address in a page of SBD DPDRAM.

Test 9 — tests the data pattern 0x0100 at every address in a page of SBD DPDRAM.

Test 10 — tests the data pattern 0x0200 at every address in a page of SBD DPDRAM.

Test 11 — tests the data pattern 0x0400 at every address in a page of SBD DPDRAM.

Test 12 — tests the data pattern 0x0800 at every address in a page of SBD DPDRAM.

Test 13 — tests the data pattern 0x1000 at every address in a page of SBD DPDRAM.

Test 14 — tests the data pattern 0x2000 at every address in a page of SBD DPDRAM.

Test 15 — tests the data pattern 0x4000 at every address in a page of SBD DPDRAM.

Test 16 — tests the data pattern 0x8000 at every address in a page of SBD DPDRAM.

Time: 40 seconds

Warnings: None

Notes: This phase performs PIO (write and read) in words. If any test in this phase fails, the following hardware may be faulty:

- INTEL 80186 Microprocessor
- INTEL 80186 Microprocessor interface to the PORTS Address/Data bus
- PORTS interface to the 3B2 computer I/O bus.

Phase #12 Tests

Test Numbers: 1 through 16

Function: These tests verify the operation of the PORTS interface to the 3B2 computer I/O bus.

Procedure: A one is walked through a field of zeros at every address of a page of System Board (SBD) Dual Port Dynamic Random Access Memory (DPDRAM), and it is verified with a read.

Hardware Tested: The PORTS interface to the 3B2 computer I/O bus is tested.

Data Returned: The number of the test that failed, the word value read from the SBD DPDRAM, and the SBD DPDRAM failing address are returned.

Phase #13 — PORTS—DMA Byte Transfers

Phase Name: PORTS — DMA Byte Transfers (dmabyt)

Type: Demand

Function: This phase diagnoses and reports any errors in the operation of the PORTS Direct Memory Access (DMA) channels (byte width).

Tests: Test 1 — checks DMA0 from the SBD DPDRAM to the PORTS RAM.

Test 2 — checks DMA0 from the PORTS RAM to the SBD DPDRAM.

Test 3 — checks DMA1 from the SBD DPDRAM to the PORTS RAM.

Test 4 — checks DMA1 from the PORTS RAM to the SBD DPDRAM.

Time: 2 seconds

Warnings: None

Notes: This phase performs DMA transfers in bytes. If any test in this phase fails, the following hardware may be faulty:

- INTEL 80186 Microprocessor
- INTEL 80186 Microprocessor interface to the PORTS Address/Data bus
- PORTS interface to the 3B2 computer I/O bus.

Phase #13 Tests

Test Number: 1

Function: This test verifies that Direct Memory Access Channel 0 (DMA0) from the System Board (SBD) Dual Port Dynamic Random Access Memory (DPDRAM) to the PORTS RAM is functional.

Procedure: Data is written [Programmed Input/Output (PIO)] to the SBD DPDRAM. That data is transferred to the PORTS RAM, and then it is compared for integrity.

Hardware Tested: The PORTS DMA0 (INTEL 80186) from the SBD to the PORTS card is tested.

Data Returned: The test number that failed, the actual data, and the expected data are returned.

Test Number:	2
Function:	This test verifies that DMA0 from the PORTS RAM to the SBD DPDRAM is functional.
Procedure:	Data is written to the PORTS RAM. That data is transferred to the SBD DPDRAM, and it is compared for integrity.
Hardware Tested:	The PORTS DMA0 (INTEL 80186) from the PORTS to the SBD is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Number:	3
Function:	This test verifies that DMA1 from the SBD DPDRAM to the PORTS RAM is functional.
Procedure:	Data is written (PIO) to the SBD DPDRAM. That data is transferred to the PORTS RAM, and then it is compared for integrity.
Hardware Tested:	The PORTS DMA1 (INTEL 80186) from the SBD to the PORTS is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Number:	4
Function:	This test verifies that DMA1 from the PORTS RAM to the SBD DPDRAM is functional.
Procedure:	Data is written to the PORTS RAM. That data is transferred to the SBD DPDRAM, and it is compared for integrity.
Hardware Tested:	The PORTS DMA1 (INTEL 80186) from the PORTS to the SBD is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

Phase #14 — PORTS—DMA Word Transfers

Phase Name: PORTS — DMA Word Transfers (dmawrd)

Type: Demand

Function: This phase diagnoses and reports any errors in the operation of the PORTS Direct Memory Access (DMA) channels (word width).

Tests: Test 1 — checks DMA0 from the SBD DPDRAM to the PORTS RAM.

Test 2 — checks DMA0 from the PORTS RAM to the SBD DPDRAM.

Test 3 — checks DMA1 from the SBD DPDRAM to the PORTS RAM.

Test 4 — checks DMA1 from the PORTS RAM to the SBD DPDRAM.

Time: 2 seconds

Warnings: None

Notes: This phase performs DMA transfers in words (16 bits). If any test in this phase fails, the following hardware may be faulty:

- INTEL 80186 Microprocessor
- INTEL 80186 Microprocessor interface to the PORTS Address/Data bus
- PORTS interface to the 3B2 computer I/O bus.

Phase #14 Tests

Test Number: 1

Function: This test verifies that Direct Memory Access Channel 0 (DMA0) from the System Board (SBD) Dual Port Dynamic Random Access Memory (DPDRAM) to the PORTS RAM is functional.

Procedure: Data is written [Programmed Input/Output (PIO)] to the SBD DPDRAM. That data is transferred to the PORTS RAM and verified for data integrity.

Hardware Tested: The PORTS DMA0 (INTEL 80186) from the SBD to the PORTS card is tested.

Data Returned: The test number that failed, the actual data, and the expected data are returned.

==========

Test Number: 2

Function: This test verifies that DMA0 from the PORTS RAM to the SBD DPDRAM is functional.

Procedure: Data is written to the PORTS RAM. That data is transferred to the SBD DPDRAM and verified for integrity.

Hardware Tested: The PORTS DMA0 (INTEL 80186) from the PORTS to the SBD is tested.

Data Returned: The test number that failed, the actual data, and the expected data are returned.

Test Number:	3
Function:	This test verifies that DMA1 from the SBD DPDRAM to the PORTS RAM is functional.
Procedure:	Data is written (PIO) to the SBD DPDRAM. That data is transferred to the PORTS RAM, and then it is compared for integrity.
Hardware Tested:	The PORTS DMA1 (INTEL 80186) from the SBD to the PORTS card is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Number:	4
Function:	This test verifies that DMA1 from the PORTS RAM to the SBD DPDRAM is functional.
Procedure:	Data is written to the PORTS RAM. That data is transferred to the SBD DPDRAM, and it is compared for integrity.
Hardware Tested:	The PORTS DMA1 (INTEL 80186) from the PORTS to the SBD is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

Phase #15 — PORTS—Printer Data Register

Phase Name: PORTS — Printer Data Register (print_1)

Type: Interactive

Function: This phase verifies that the PORTS parallel Printer Data register is functioning properly.

Tests: Tests 1 through 4 — check the ability to read/write Printer Data register.

Test 5 — checks the bit interdependency of Printer Data register.

Time: 1 second

Warnings: There should *not* be a printer attached to the printer port during this phase. This test does not check the data that is sent *from* the Printer Data register. It only checks the data that is being sent *to* the Printer Data register.

Notes: A warning message accompanies this phase and is displayed on the system console. If any test in this phase fails, the following hardware may be faulty:

- Printer parallel Data register circuitry
- Component interface to the PORTS Address/Data bus
- PORTS RAM/ROM.

Phase #15 Tests

Test Numbers: 1 through 4

Function: These tests verify that the Printer Data register can be written correctly.

Procedure: Various data patterns are written to the register and verified with reads.

Hardware Tested: The WE300A is tested.

Data Returned: The number of the failing test, the actual value in the Printer Data register, and the expected value in the Printer Data register are returned.

==========

Test Number: 5

Function: This test checks for bit interdependency in the Printer Data register.

Procedure: A one is walked through each bit position in the Printer Data register and verified with a read.

Hardware Tested: The WE300A is tested.

Data Returned: The number of the failing test, the actual value in the Printer Data register, and the expected value in the Printer Data register are returned.

Phase #16 — PORTS—Printer CSR Register

Phase Name: PORTS — Printer CSR Test (print_2)

Type: Interactive

Function: This phase verifies the ability to access the printer Control and Status Register (CSR).

Tests: Tests 1 through 8 — check the Printer Control bits.

Tests 9 through 12 — check the Printer Status bits.

Time: 1 second

Warnings: The printer loop around cable must be connected.

Notes: The Printer Status bits are tested by strapping them to the lower nibble of the Printer Data register. Values are written to the Data register and then read from the CSR. Bit assignments are as follows:

Bit 0 - Printer Reset
Bit 1 - Printer Strobe
Bit 2 - Not Used
Bit 3 - Not Used
Bit 4 - Printer Busy
Bit 5 - Printer Paper Empty
Bit 6 - Printer Select
Bit 7 - Printer Fault.

Only bits 0 and 1 are used for printer control. Bits 2 and 3 are tested but not used.

A warning message accompanies this phase and is displayed on the system console. If any test in this phase fails, the following hardware may be faulty:

- Printer loop around cable
- PORTS Address/Data bus
- PORTS RAM/ROM.

Phase #16 Tests

Test Numbers:	1 through 8
Function:	These tests check the ability to read/write the Printer Control bits in the CSR.
Procedure:	Various bit patterns are written to the lower nibble of the CSR and verified with reads. Only the first two bits are used for printer control. The fourth bit is writable but is not currently used.
Hardware Tested:	The 74LS173 and LS244 Integrated Circuits (ICs) are tested.
Data Returned:	The number of the failing test, the actual value in printer CSR, and the expected value in printer CSR are returned.

==========

Test Numbers:	9 through 12
Function:	These tests check the ability to read Printer Status bits in the printer CSR.
Procedure:	With the loop around cable connected, various bit patterns are written to the lower nibble of the CSR. The CSR is read and verified to have the correct bits set.
Hardware Tested:	The 74LS173 and LS244 ICs are tested.
Data Returned:	The number of the failing test, the actual value in the printer CSR, and the expected value in the printer CSR are returned.

Phase #17 — PORTS—DUART 0 Internal Loop

Phase Name: PORTS — DUART (SC2681) 0 Internal Loop Test (duart0_1)

Type: Demand

Function: This phase verifies the ability of the Dual Universal Asynchronous Receiver/Transmitter (DUART) 0 to do internal loop around at various baud rates.

Tests: Tests 1 through 10 — check the internal loop of the ASCII characters through DUART 0, Port 1.

Tests 11 through 20 — check the internal loop of the ASCII characters through DUART 0, Port 2.

Test 21 — verifies operation of the FIFO Full and Overrun Error bits for DUART 0, Port 1.

Test 22 — verifies operation of the FIFO Full and Overrun Error bits for DUART 0, Port 2.

Test (23) — checks the ability of the DUART to send an interrupt when the Receive Data register on DUART 0, Port 1 is enabled and has received a character.

Test (24) — checks the ability of the DUART to send an interrupt when the Transmit Data register on DUART 0, Port 1 is enabled and after character has been transmitted.

Test (25) — checks the ability of the DUART to send an interrupt when the Receive Data register on DUART 0, Port 2 is enabled and has received a character.

Test (26) — checks the ability of the DUART to send an interrupt when the Transmit Data register on DUART 0, Port 2 is enabled and after a character has been transmitted.

Test (27) — checks the ability of the DUART to send an interrupt when the Receive buffer is full (3 characters) on DUART 0, Port 1.

Test (28) — checks the ability of the DUART to send an interrupt when the Receive buffer is full (3 characters) on DUART 0, Port 2.

Test (29) — checks the ability of the DUART to send an interrupt when a break character is detected on DUART 0, Port 1.

Test (30) — checks the ability of the DUART to send an interrupt when a break character is detected on DUART 0, Port 2.

Test (31) — checks the ability of the DUART to send an interrupt when "Counter Ready" is detected on DUART 0.

Time: 7 seconds

Warnings: None

Notes: If any test in this phase fails, the following hardware may be faulty:

- DUART (SC2681) and its 3.6864-megahertz crystal
- DUART interface to the PORTS Address/Data bus
- PORTS RAM/ROM.

Phase #17 Tests

Test Numbers: 1 through 5

Function: These tests verify that DUART 0, Port 1 can internally loop American Standard Code for Information Interchange (ASCII) characters at various baud rates.

Procedure: Port 1 is initialized for internal loop around using 7 bits and no parity. The ASCII characters are transmitted at 300, 1200, 4800, 9600, and 19200 baud. The received characters are verified to match the ones transmitted.

Hardware Tested: The DUART 0 (SC2681) is tested.

Data Returned: The number of the failing test, the actual character received, and the character transmitted are returned.

==========

Test Numbers: 6 through 10

Function: These tests verify that the DUART 0, Port 1 can internally loop ASCII characters at various baud rates.

Procedure: Port 1 is initialized for internal loop around using 8 bits and no parity. The ASCII characters are transmitted at 300, 1200, 4800, 9600, and 19200 baud. The received characters are verified to match the ones transmitted.

Hardware Tested: The DUART 0 (SC2681) is tested.

Data Returned: The number of the failing test, the actual character received, and the character transmitted are returned.

==========

Test Numbers: 11 through 15

Function: These tests verify that the DUART 0, Port 2 can internally loop ASCII characters at various baud rates.

Procedure: Port 2 is initialized for internal loop around using 7 bits and no parity. The ASCII characters are transmitted at 300, 1200, 4800, 9600, and 19200 baud. The received characters are verified to match the ones transmitted.

Hardware Tested: The DUART 0 (SC2681) is tested.

Data Returned: The number of the failing test, the actual character received, and the character transmitted are returned.

Test Numbers:	16 through 20
Function:	These tests verify that the DUART 0, Port 2 can internally loop ASCII characters at various baud rates.
Procedure:	Port 2 is initialized for internal loop around using 8 bits and no parity. The ASCII characters are transmitted at 300, 1200, 4800, 9600, and 19200 baud. The received characters are verified to match the ones transmitted.
Hardware Tested:	The DUART 0 (SC2681) is tested.
Data Returned:	The number of the failing test, the actual character received, and the character transmitted are returned.

==========

Test Number:	21
Function:	This test verifies the operation of the First-In-First-Out (FIFO) Full and Overrun Error bits for DUART 0, Port 1.
Procedure:	Port 1 is initialized for internal loop around using 7 bits and no parity. The baud rate is set to 19200. Three characters are transmitted, and it is verified that the FIFO Full bit is set. Then, two additional characters are sent, and the Overrun Error bit is checked to be set.
Hardware Tested:	The DUART 0 (SC2681) is tested.
Data Returned:	The number of the failing test and the actual value in the Status register are returned.

==========

Test Number:	22
Function:	This test verifies the operation of the FIFO Full and Overrun Error bits for DUART 0, Port 2.
Procedure:	Port 2 is initialized for internal loop around using 7 bits and no parity. The baud rate is set to 19200. Three characters are transmitted, and the FIFO Full bit is checked to be set. Then, two additional characters are sent, and the Overrun Error bit is checked to be set.
Hardware Tested:	The DUART 0 (SC2681) is tested.
Data Returned:	The number of the failing test and the actual value in the Status register are returned.

Test Number: (23)

Function: This test verifies the ability of the DUART to send an interrupt when the Receive Data register on DUART 0, Port 1 is enabled and has received a character.

Procedure: Port 1 is initialized to transmit and receive interrupts. The interrupts consist of 7 bits, no parity, and 2 Stop bits. The baud rate is set to 9600 bits per second (bps). Set the DUART to send an interrupt when the Receive Data register is enabled and has received a character; then check that the interrupt was correctly received.

Hardware Tested: The DUART 0 (SC2681) is tested.

Data Returned: The number of the failing test, the ASCII character received, and the ASCII character transmitted are returned.

==========

Test Number: (24)

Function: This test verifies the ability of the DUART to send an interrupt when the Transmit Data register on DUART 0, Port 1 is enabled and after a character has been transmitted.

Procedure: Port 1 is initialized to transmit and receive interrupts. The interrupts consist of 7 bits, no parity, and 2 Stop bits. The baud rate is set to 9600 bps. Set the DUART to send an interrupt when the Transmit Data register is enabled and the Transmit buffer is empty; then check that the interrupt was correctly received.

Hardware Tested: The DUART 0 (SC2681) is tested.

Data Returned: The number of the failing test, the ASCII character received, and the ASCII character transmitted are returned.

==========

Test Number: (25)

Function: This test verifies the ability of the DUART to send an interrupt when the Receive Data register on DUART 0, Port 2 is enabled and has received a character.

Procedure: Port 2 is initialized to transmit and receive interrupts. The interrupts consist of 7 bits, no parity, and 2 Stop bits. The baud rate is set to 9600 bps. Set the DUART to send an interrupt when the Receive Data register is enabled and has received a character; then check that the interrupt was correctly received.

Hardware Tested: The DUART 0 (SC2681) is tested.

Data Returned: The number of the failing test, the ASCII character received, and the ASCII character transmitted are returned.

Test Number: (26)

Function: This test verifies the ability of the DUART to send an interrupt when the Transmit Data register on DUART 0, Port 2 is enabled and after a character has been transmitted.

Procedure: Port 2 is initialized to transmit and receive interrupts. The interrupts consist of 7 bits, no parity, and 2 Stop bits. The baud rate is set to 9600 bps. Set the DUART to send an interrupt when the Transmit Data register is enabled and the Transmit buffer is empty; then check that the interrupt was correctly received.

Hardware Tested: The DUART 0 (SC2681) is tested.

Data Returned: The number of the failing test, the ASCII character received, and the ASCII character transmitted are returned.

==========

Test Number: (27)

Function: This test verifies the ability of the DUART to send an interrupt when the Receive buffer is full (3 characters) on DUART 0, Port 1.

Procedure: Port 1 is initialized to transmit and receive interrupts. The interrupts consist of 8 bits, no parity, and 2 Stop bits. The baud rate is set to 9600 bps. Transmit three characters over DUART 0, Port 1. Check for an interrupt. Verify that the character transmission is correct.

Hardware Tested: The DUART 0 (SC2681) is tested.

Data Returned: The number of the failing test, the ASCII character received, and the ASCII character transmitted are returned.

==========

Test Number: (28)

Function: This test verifies the ability of the DUART to send an interrupt when the Receive buffer is full (3 characters) on DUART 0, Port 2.

Procedure: Port 2 is initialized to transmit and receive interrupts. The interrupts consist of 8 bits, no parity, and 2 Stop bits. The baud rate is set to 9600 bps. Transmit three characters over DUART 0, Port 2. Check for an interrupt. Verify that the character transmission is correct.

Hardware Tested: The DUART 0 (SC2681) is tested.

Data Returned: The number of the failing test, the ASCII character received, and the ASCII character transmitted are returned.

Test Number: (29)

Function: This test verifies the ability of the DUART to send an interrupt when a break character is detected on DUART 0, Port 1.

Procedure: Port 1 is initialized to transmit and receive interrupts. The interrupts consist of 7 bits, no parity, and 2 Stop bits. The baud rate is set to 9600 bps. Set the DUART to send an interrupt when a break character is detected. Enable transmitter and receiver. Start break and check that the interrupt was correctly received.

Hardware Tested: The DUART 0 (SC2681) is tested.

Data Returned: The number of the failing test, the ASCII character received, and the ASCII character transmitted are returned.

==========

Test Number: (30)

Function: This test verifies the ability of the DUART to send an interrupt when a break character is detected on DUART 0, Port 2.

Procedure: Port 2 is initialized to transmit and receive interrupts. The interrupts consist of 7 bits, no parity, and 2 Stop bits. The baud rate is set to 9600 bps. Set the DUART to send an interrupt when a break character is detected. Enable transmitter and receiver. Start break and check that the interrupt was correctly received.

Hardware Tested: The DUART 0 (SC2681) is tested.

Data Returned: The number of the failing test, the ASCII character received, and the ASCII character transmitted are returned.

==========

Test Number: (31)

Function: This test verifies the ability of the DUART to send an interrupt when "Counter Ready" is detected on DUART 0.

Procedure: Port 2 is initialized to transmit and receive interrupts. The interrupts consist of 7 bits, no parity, and 2 Stop bits. The baud rate is set to 9600 bps. Set the DUART to send an interrupt when "Counter Ready" is detected. Enable transmitter and receiver. Reset and start the counter. Check that the interrupt was correctly received.

Hardware Tested: The DUART 0 (SC2681) is tested.

Data Returned: The number of the failing test, the ASCII character received, and the ASCII character transmitted are returned.

Phase #18 — PORTS—DUART 1 Internal Loop

Phase Name: PORTS — DUART (SC2681) 1 Internal Loop Test (duart1_1)

Type: Demand

Function: This phase verifies the ability of Dual Universal Asynchronous Receiver/Transmitter (DUART) 1 to do internal loop around at various baud rates.

Tests: Tests 1 through 10 — check the internal loop of the ASCII characters through DUART 1, Port 1.

Tests 11 through 20 — check the internal loop of the ASCII characters through DUART 1, Port 2.

Test 21 — verifies the operation of the FIFO Full and Overrun Error bits for DUART 1, Port 1.

Test 22 — verifies the operation of the FIFO Full and Overrun Error bits for DUART 1, Port 2.

Test (23) — checks the ability of the DUART to send an interrupt when the Receive Data register on DUART 1, Port 1 is enabled and has received a character.

Test (24) — checks the ability of the DUART to send an interrupt when the Transmit Data register on DUART 1, Port 1 is enabled and after a character has been transmitted.

Test (25) — checks the ability of the DUART to send an interrupt when the Receive Data register on DUART 1, Port 2 is enabled and has received a character.

Test (26) — checks the ability of the DUART to send an interrupt when the Transmit Data register on DUART 1, Port 2 is enabled and after a character has been transmitted.

Test (27) — checks the ability of the DUART to send an interrupt when the Receive buffer is full (3 characters) on DUART 1, Port 1.

Test (28) — checks the ability of the DUART to send an interrupt when the Receive buffer is full (3 characters) on DUART 1, Port 2.

Test (29) — checks the ability of the DUART to send an interrupt when a break character is detected on DUART 1, Port 1.

Test (30) — checks the ability of the DUART to send an interrupt when a break character is detected on DUART 1, Port 2.

Test (31) — checks the ability of the DUART to send an interrupt when "Counter Ready" is detected on DUART 1.

Time: 10 seconds

Warnings: None

Notes: If any test in this phase fails, the following hardware may be faulty:

- The DUART (SC2681) and its 3.6864-megahertz crystal
- DUART interface to the PORTS Address/Data bus
- PORTS RAM/ROM.

Phase #18 Tests

Test Numbers: 1 through 5

Function: These tests verify that DUART 1, Port 1 can internally loop American Standard Code for Information Interchange (ASCII) characters at various baud rates.

Procedure: Port 1 is initialized for internal loop around using 7 bits and no parity. The ASCII characters are transmitted at 300, 1200, 4800, 9600, and 19200 baud. The received character is verified to match the one transmitted.

Hardware Tested: The DUART 1 (SC2681) is tested.

Data Returned: The number of the test that failed, the actual character received, and the character transmitted are returned.

==========

Test Numbers: 6 through 10

Function: These tests verify that DUART 1, Port 1 can internally loop ASCII characters at various baud rates.

Procedure: Port 1 is initialized for internal loop around using 8 bits and no parity. The ASCII characters are transmitted at 300, 1200, 4800, 9600, and 19200 baud. The received character is verified to match the one transmitted.

Hardware Tested: The DUART 1 (SC2681) is tested.

Data Returned: The number of the failing test, the actual character received, and the character transmitted are returned.

==========

Test Numbers: 11 through 15

Function: These tests verify that DUART 1, Port 2 can internally loop ASCII characters at various baud rates.

Procedure: Port 2 is initialized for internal loop around using 7 bits and no parity. The ASCII characters are transmitted at 300, 1200, 4800, 9600, and 19200 baud. The received character is verified to match the one transmitted.

Hardware Tested: The DUART 1 (SC2681) is tested.

Data Returned: The number of the failing test, the actual character received, and the character transmitted are returned.

Test Numbers: 16 through 20

Function: These tests verify that DUART 1, Port 2 can internally loop ASCII characters at various baud rates.

Procedure: Port 2 is initialized for internal loop around using 8 bits and no parity. The ASCII characters are transmitted at 300, 1200, 4800, 9600, and 19200 baud. The received character is verified to match the one transmitted.

Hardware Tested: The DUART 1 (SC2681) is tested.

Data Returned: The number of the failing test, the actual character received, and the character transmitted are returned.

==========

Test Number: 21

Function: This test verifies the operation of the First-In-First-Out (FIFO) Full and Overrun Error bits for DUART 1, Port 1.

Procedure: Port 1 is initialized for internal loop around using 7 bits and no parity. The baud rate is set to 19200. Three characters are transmitted, and the FIFO Full bit is verified to be set. Two additional characters are transmitted, and the Overrun Error bit is verified to be set.

Hardware Tested: The DUART 1 (SC2681) is tested.

Data Returned: The number of the failing test and the actual value in the Status register are returned.

==========

Test Number: 22

Function: This test verifies the operation of the FIFO Full and Overrun Error bits for DUART 1, Port 2.

Procedure: Port 2 is initialized for internal loop around using 7 bits and no parity. The baud rate is set to 19200. Three characters are transmitted, and the FIFO Full bit is verified to be set. Then, two additional characters are transmitted, and the Overrun Error bit is verified to be set.

Hardware Tested: The DUART 1 (SC2681) is tested.

Data Returned: The number of the failing test and the actual value in the Status register are returned.

Test Number: (23)

Function: This test verifies the ability of the DUART to send an interrupt when the Receive Data register on DUART 1, Port 1 is enabled and has received a character.

Procedure: Port 1 is initialized to transmit and receive interrupts. The interrupts consist of 7 bits, no parity, and 2 Stop bits. The baud rate is set to 9600 bits per second (bps). Set the DUART to send an interrupt when the Receive Data register is enabled and has received a character; then check that the interrupt was correctly received.

Hardware Tested: The DUART 1 (SC2681) is tested.

Data Returned: The number of the failing test, the ASCII character received, and the ASCII character transmitted are returned.

==========

Test Number: (24)

Function: This test verifies the ability of the DUART to send an interrupt when the Transmit Data register on DUART 1, Port 1 is enabled and after a character has been transmitted.

Procedure: Port 1 is initialized to transmit and receive interrupts. The interrupts consist of 7 bits, no parity, and 2 Stop bits. The baud rate is set to 9600 bps. Set the DUART to send an interrupt when the Transmit Data register is enabled and the Transmit buffer is empty; then check that the interrupt was correctly received.

Hardware Tested: The DUART 1 (SC2681) is tested.

Data Returned: The number of the failing test, the ASCII character received, and the ASCII character transmitted are returned.

==========

Test Number: (25)

Function: This test verifies the ability of the DUART to send an interrupt when the Receive Data register on DUART 1, Port 2 is enabled and has received a character.

Procedure: Port 2 is initialized to transmit and receive interrupts. The interrupts consist of 7 bits, no parity, and 2 Stop bits. The baud rate is set to 9600 bps. Set the DUART to send an interrupt when the Receive Data register is enabled and has received a character; then check that the interrupt was correctly received.

Hardware Tested: The DUART 1 (SC2681) is tested.

Data Returned: The number of the failing test, the ASCII character received, and the ASCII character transmitted are returned.

Test Number: (26)

Function: This test verifies the ability of the DUART to send an interrupt when the Transmit Data register on DUART 1, Port 2 is enabled and after a character has been transmitted.

Procedure: Port 2 is initialized to transmit and receive interrupts. The interrupts consist of 7 bits, no parity, and 2 Stop bits. The baud rate is set to 9600 bps. Set the DUART to send an interrupt when the Transmit Data register is enabled and the Transmit buffer is empty; then check that the interrupt was correctly received.

Hardware Tested: The DUART 1 (SC2681) is tested.

Data Returned: The number of the failing test, the ASCII character received, and the ASCII character transmitted are returned.

==========

Test Number: (27)

Function: This test verifies the ability of the DUART to send an interrupt when the Receive buffer is full (3 characters) on DUART 1, Port 1.

Procedure: Port 1 is initialized to transmit and receive interrupts. The interrupts consist of 8 bits, no parity, and 2 Stop bits. The baud rate is set to 9600 bps. Transmit three characters over DUART 1, Port 1. Check for an interrupt. Verify that the character transmission is correct.

Hardware Tested: The DUART 1 (SC2681) is tested.

Data Returned: The number of the failing test, the ASCII character received, and the ASCII character transmitted are returned.

==========

Test Number: (28)

Function: This test verifies the ability of the DUART to send an interrupt when the Receive buffer is full (3 characters) on DUART 1, Port 2.

Procedure: Port 2 is initialized to transmit and receive interrupts. The interrupts consist of 8 bits, no parity, and 2 Stop bits. The baud rate is set to 9600 bps. Transmit three characters over DUART 1, Port 2. Check for an interrupt. Verify that the character transmission is correct.

Hardware Tested: The DUART 1 (SC2681) is tested.

Data Returned: The number of the failing test, the ASCII character received, and the ASCII character transmitted are returned.

Test Number: (29)

Function: This test verifies the ability of the DUART to send an interrupt when a break character is detected on DUART 1, Port 1.

Procedure: Port 1 is initialized to transmit and receive interrupts. The interrupts consist of 7 bits, no parity, and 2 Stop bits. The baud rate is set to 9600 bps. Set the DUART to send an interrupt when a break character is detected. Enable transmitter and receiver. Start break and check that the interrupt was correctly received.

Hardware Tested: The DUART 1 (SC2681) is tested.

Data Returned: The number of the failing test, the ASCII character received, and the ASCII character transmitted are returned.

==========

Test Number: (30)

Function: This test verifies the ability of the DUART to send an interrupt when a break character is detected on DUART 1, Port 2.

Procedure: Port 2 is initialized to transmit and receive interrupts. The interrupts consist of 7 bits, no parity, and 2 Stop bits. The baud rate is set to 9600 bps. Set the DUART to send an interrupt when a break character is detected. Enable transmitter and receiver. Start break and check that the interrupt was correctly received.

Hardware Tested: The DUART 1 (SC2681) is tested.

Data Returned: The number of the failing test, the ASCII character received, and the ASCII character transmitted are returned.

==========

Test Number: (31)

Function: This test verifies the ability of the DUART to send an interrupt when "Counter Ready" is detected on DUART 1.

Procedure: Port 2 is initialized to transmit and receive interrupts. The interrupts consist of 7 bits, no parity, and 2 Stop bits. The baud rate is set to 9600 bps. Set the DUART to send an interrupt when "Counter Ready" is detected. Enable transmitter and receiver. Reset and start the counter. Check that the interrupt was correctly received.

Hardware Tested: The DUART 1 (SC2681) is tested.

Data Returned: The number of the failing test, the ASCII character received, and the ASCII character transmitted are returned.

Phase #19 — PORTS—DUART 0 External Loop

Phase Name: PORTS — DUART (SC2681) 0 External Loop (duart0_2)

Type: Interactive

Function: This phase verifies the ability of Dual Universal Asynchronous Receiver/Transmitter (DUART) 0 to do external loop around at various baud rates.

Tests: Test 1 — checks the Port 1 DTR.

Tests 2 through 11 — check the external loop of the ASCII characters through DUART 0, Port 1.

Test 12 — checks the Port 2 DTR.

Tests 13 through 22 — check the external loop of the ASCII characters through DUART 0, Port 2.

Test (23) — checks the ability of the DUART to send an interrupt when the Receive Data register on DUART 0, Port 1 is enabled and has received a character.

Test (24) — checks the ability of the DUART to send an interrupt when the Transmit Data register on DUART 0, Port 1 is enabled and after a character has been transmitted.

Test (25) — checks the ability of the DUART to send an interrupt when the Receive Data register on DUART 0, Port 2 is enabled and has received a character.

Test (26) — checks the ability of the DUART to send an interrupt when the Transmit Data register on DUART 0, Port 2 is enabled and after a character has been transmitted.

Test (27) — checks the ability of the DUART to send an interrupt when Receive buffer full (3 characters) is detected on DUART 0, Port 1.

Test (28) — checks the ability of the DUART to send an interrupt when Receive buffer full (3 characters) is detected on DUART 0, Port 2.

Test (29) — checks the ability of the DUART to send an interrupt when a break character is detected on DUART 0, Port 1.

Test (30) — checks the ability of the DUART to send an interrupt when a break character is detected on DUART 0, Port 2.

Test (31) — checks the ability of the DUART to send an interrupt when "Counter Ready" is detected on DUART 0.

Time: 7 seconds

Warnings: This phase requires two special loop around plugs. They must be connected to the first and second serial jack connections of the PORTS card under test.

Notes: Figure 8-1 shows the loop around plug wiring. The DTR is implemented by using bits 0 and 1 of the DUART general purpose Output register. These two outputs are inverted and internally strapped to bits 0 and 1 of the general purpose Input register.

A warning message accompanies this phase and displays on the system console. If any test in this phase fails, the following hardware may be faulty:

- DUART (SC2681)
- RS-232 driver chips
- Loop around connector
- DUART interface to the PORTS Address/Data bus
- PORTS RAM/ROM.

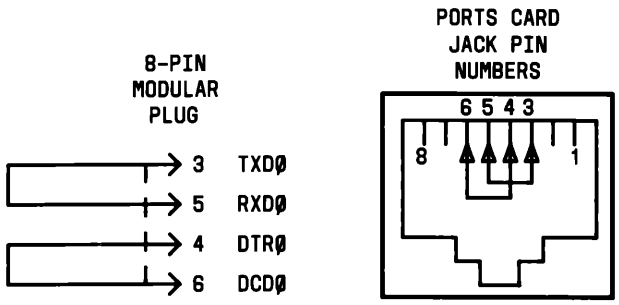


Figure 8-1: Loop Around Plug Wiring

Phase #19 Tests

Test Number:	1
Function:	This test checks the Data Terminal Ready (DTR) for Port 1.
Procedure:	The DTR is asserted by writing to the general purpose Output register. The DTR is verified to be asserted by reading the general purpose Input register.
Hardware Tested:	The DUART 0 (SC2681) is tested.
Data Returned:	The number of the failing test, the actual value of the Input register, and the expected value of the Input register are returned.

Test Numbers:	2 through 6
Function:	These tests verify that DUART 0, Port 1 can externally loop American Standard Code for Information Interchange (ASCII) characters at various baud rates.
Procedure:	Port 1 is initialized for external loop around using 7 bits, no parity, and 2 Stop bits. This requires setting both the transmit and receive ports in the same mode. The ASCII characters are transmitted at 300, 1200, 4800, 9600, and 19200 baud. The received character is verified to match the one transmitted.
Hardware Tested:	The DUART 0 (SC2681) is tested.
Data Returned:	The number of the failing test, the actual character received, and the character transmitted are returned.

==========

Test Numbers:	7 through 11
Function:	These tests verify that DUART 0, Port 1 can externally loop ASCII characters at various baud rates.
Procedure:	Port 1 is initialized for external loop around using 8 bits, no parity, and 2 Stop bits. The ASCII characters are transmitted at 300, 1200, 4800, 9600, and 19200 baud. The received character is verified to match the one transmitted.
Hardware Tested:	The DUART 0 (SC2681) is tested.
Data Returned:	The number of the failing test, the actual character received, and the character transmitted are returned.

==========

Test Number:	12
Function:	This test checks the DTR for Port 2.
Procedure:	The DTR is asserted for Port 2 by writing to the general purpose Output register. The DTR is verified to be asserted by reading the general purpose Input register. Bits 1 and 2 of the general purpose Input and Output registers are tied together. The data read from the Input register is inverted.
Hardware Tested:	The DUART 0 (SC2681) is tested.
Data Returned:	The number of the failing test, the actual value in the Input register, and the expected value in the Input register are returned.

Test Numbers: 13 through 17

Function: These tests verify that DUART 0, Port 2 can externally loop ASCII characters at various baud rates.

Procedure: Port 2 is initialized for external loop around using 7 bits, no parity, and 2 Stop bits. The ASCII characters are transmitted at 300, 1200, 4800, 9600, and 19200 baud. The received character is verified to match the one transmitted.

Hardware Tested: The DUART 0 (SC2681) is tested.

Data Returned: The number of the failing test, the ASCII value received, and the ASCII value transmitted are returned.

==========

Test Numbers: 18 through 22

Function: These tests verify that DUART 0, Port 2 can externally loop ASCII characters at various baud rates.

Procedure: Port 2 is initialized for external loop around using 8 bits, no parity, and 2 Stop bits. The ASCII characters are transmitted at 300, 1200, 4800, 9600, and 19200 baud. The received character is verified to match the one transmitted.

Hardware Tested: The DUART 0 (SC2681) is tested.

Data Returned: The number of the failing test, the ASCII value received, and the ASCII value transmitted are returned.

==========

Test Number: (23)

Function: This test verifies the ability of the DUART to send an interrupt when the Receive Data register on DUART 0, Port 1 is enabled and a character is received.

Procedure: Port 1 is initialized to transmit and receive interrupts. The interrupts consist of 7 bits, no parity, and 2 Stop bits. The baud rate is set to 9600 bits per second (bps). Set the DUART to send an interrupt when the Receive Data register is enabled and has received a character; then check that the interrupt was correctly received.

Hardware Tested: The DUART 0 (SC2681) is tested.

Data Returned: The number of the failing test, the ASCII character received, and the ASCII character transmitted are returned.

Test Number: (24)

Function: This test verifies the ability of the DUART to send an interrupt when the Transmit Data register on DUART 0, Port 1 is enabled and after a character has been transmitted.

Procedure: Port 1 is initialized to transmit and receive interrupts. The interrupts consist of 7 bits, no parity, and 2 Stop bits. The baud rate is set to 9600 bps. Set the DUART to send an interrupt when the Transmit Data register is enabled and a character has been transmitted; then check that the interrupt was correctly received.

Hardware Tested: The DUART 0 (SC2681) is tested.

Data Returned: The number of the failing test, the ASCII character received, and the ASCII character transmitted are returned.

==========

Test Number: (25)

Function: This test verifies the ability of the DUART to send an interrupt when the Receive Data register on DUART 0, Port 2 is enabled and a character is received.

Procedure: Port 2 is initialized to transmit and receive interrupts. The interrupts consist of 7 bits, no parity, and 2 Stop bits. The baud rate is set to 9600 bps. Set the DUART to send an interrupt when the Receive Data register is enabled and has received a character; then check that the interrupt was correctly received.

Hardware Tested: The DUART 0 (SC2681) is tested.

Data Returned: The number of the failing test, the ASCII character received, and the ASCII character transmitted are returned.

==========

Test Number: (26)

Function: This test verifies the ability of the DUART to send an interrupt when the Transmit Data register on DUART 0, Port 2 is enabled and after a character has been transmitted.

Procedure: Port 2 is initialized to transmit and receive interrupts. The interrupts consist of 7 bits, no parity, and 2 Stop bits. The baud rate is set to 9600 bps. Set the DUART to send an interrupt when the Transmit Data register is enabled and a character has been transmitted; then check that the interrupt was correctly received.

Hardware Tested: The DUART 0 (SC2681) is tested.

Data Returned: The number of the failing test, the ASCII character received, and the ASCII character transmitted are returned.

Test Number: (27)

Function: This test verifies the ability of the DUART to send an interrupt when Receive buffer full (3 characters) is detected on DUART 0, Port 1.

Procedure: Port 1 is initialized to transmit and receive interrupts. The interrupts consist of 8 bits, no parity, and 2 Stop bits. The baud rate is set to 9600 bps. Transmit three characters over DUART 0, Port 1. Check for an interrupt. Verify that the character transmission is correct.

Hardware Tested: The DUART 0 (SC2681) is tested.

Data Returned: The number of the failing test, the ASCII character received, and the ASCII character transmitted are returned.

==========

Test Number: (28)

Function: This test verifies the ability of the DUART to send an interrupt when Receive buffer full (3 characters) is detected on DUART 0, Port 2.

Procedure: Port 2 is initialized to transmit and receive interrupts. The interrupts consist of 8 bits, no parity, and 2 Stop bits. The baud rate is set to 9600 bps. Transmit three characters over DUART 0, Port 2. Check for an interrupt. Verify that the character transmission is correct.

Hardware Tested: The DUART 0 (SC2681) is tested.

Data Returned: The number of the failing test, the ASCII character received, and the ASCII character transmitted are returned.

==========

Test Number: (29)

Function: This test verifies the ability of the DUART to send an interrupt when a break character is detected on DUART 0, Port 1.

Procedure: Port 1 is initialized to transmit and receive interrupts. The interrupts consist of 7 bits, no parity, and 2 Stop bits. The baud rate is set to 9600 bps. Set the DUART to send an interrupt when a break character is detected. Enable transmitter and receiver. Start break and check that the interrupt was correctly received.

Hardware Tested: The DUART 0 (SC2681) is tested.

Data Returned: The number of the failing test, the ASCII character received, and the ASCII character transmitted are returned.

Test Number: (30)

Function: This test verifies the ability of the DUART to send an interrupt when a break character is detected on DUART 0, Port 2.

Procedure: Port 2 is initialized to transmit and receive interrupts. The interrupts consist of 7 bits, no parity, and 2 Stop bits. The baud rate is set to 9600 bps. Set the DUART to send an interrupt when a break character is detected. Enable transmitter and receiver. Start break and check that the interrupt was correctly received.

Hardware Tested: The DUART 0 (SC2681) is tested.

Data Returned: The number of the failing test, the ASCII character received, and the ASCII character transmitted are returned.

==========

Test Number: (31)

Function: This test verifies the ability of the DUART to send an interrupt when "Counter Ready" is detected on DUART 0.

Procedure: Port 2 is initialized to transmit and receive interrupts. The interrupts consist of 7 bits, no parity, and 2 Stop bits. The baud rate is set to 9600 bps. Set the DUART to send an interrupt when "Counter Ready" is detected. Enable transmitter and receiver. Reset and start the counter. Check that the interrupt was correctly received.

Hardware Tested: The DUART 0 (SC2681) is tested.

Data Returned: The number of the failing test, the ASCII character received, and the ASCII character transmitted are returned.

Phase #20 — PORTS—DUART 1 External Loop

Phase Name: PORTS — DUART (SC2681) 1 External Loop (duart1_2)

Type: Interactive

Function: This phase verifies the ability of Dual Universal Asynchronous Receiver/Transmitter (DUART) 1 to do external loop around at various baud rates.

Tests: Test 1 — checks the Port 1 DTR.

Tests 2 through 11 — check the external loop of the ASCII characters through DUART 1, Port 1.

Test 12 — checks the Port 2 DTR.

Tests 13 through 22 — check the external loop of the ASCII characters through DUART 1, Port 2.

Test (23) — checks the ability of the DUART to send an interrupt when the Receive Data register on DUART 1, Port 1 is enabled and has received a character.

Test (24) — checks the ability of the DUART to send an interrupt when the Transmit Data register on DUART 1, Port 1 is enabled and after a character has been transmitted.

Test (25) — checks the ability of the DUART to send an interrupt when the Receive Data register on DUART 1, Port 2 is enabled and has received a character.

Test (26) — checks the ability of the DUART to send an interrupt when the Transmit Data register on DUART 1, Port 2 is enabled and after a character has been transmitted.

Test (27) — checks the ability of the DUART to send an interrupt when Receive buffer full (3 characters) is detected on DUART 1, Port 1.

Test (28) — checks the ability of the DUART to send an interrupt when Receive buffer full (3 characters) is detected on DUART 1, Port 2.

Test (29) — checks the ability of the DUART to send an interrupt when a break character is detected on DUART 1, Port 1.

Test (30) — checks the ability of the DUART to send an interrupt when a break character is detected on DUART 1, Port 2.

Test (31) — checks the ability of the DUART to send an interrupt when "Counter Ready" is detected on DUART 1.

Time: 7 seconds

Warnings: This phase requires two special loop around plugs. They must be connected to the third and fourth serial jack connections of the PORTS card under test.

Notes: Figure 8-1 shows the loop around plug wiring. (See Phase 19.)

The DTR is implemented by using bits 0 and 1 of the DUART general purpose Output register. These two outputs are inverted and internally strapped to bits 0 and 1 of the general purpose Input register.

A warning message accompanies this phase and displays on the system

console. If any test in this phase fails, the following hardware may be faulty:

- DUART (SC2681)
- RS-232 driver chips
- Loop around connector
- DUART interface to the PORTS Address/Data bus
- PORTS RAM/ROM.

Phase #20 Tests

Test Number: 1

Function: This test checks the Data Terminal Ready (DTR) for Port 1.

Procedure: The DTR is asserted for Port 1 by writing to the general purpose Output register. The DTR is verified to be asserted by reading the general purpose Input register.

Hardware Tested: The DUART 1 (SC2681) is tested.

Data Returned: The number of the failing test, the actual value of the Input register, and the expected value of the Input register are returned.

==========

Test Numbers: 2 through 6

Function: These tests verify that DUART 1, Port 1 can externally loop American Standard Code for Information Interchange (ASCII) characters at various baud rates.

Procedure: Port 1 is initialized for external loop around using 7 bits, no parity, and 2 Stop bits. This requires setting both the transmit and receive ports in the same mode. The ASCII characters are transmitted at 300, 1200, 4800, 9600, and 19200 baud. The received character is verified to match the one transmitted.

Hardware Tested: The DUART 1 (SC2681) is tested.

Data Returned: The number of the failing test, the actual character received, and the character transmitted are returned.

Test Numbers:	7 through 11
Function:	These tests verify that DUART 1, Port 1 can externally loop ASCII characters at various baud rates.
Procedure:	Port 1 is initialized for external loop around using 8 bits, no parity, and 2 Stop bits. The ASCII characters are transmitted at 300, 1200, 4800, 9600, and 19200 baud. The received character is verified to match the one transmitted.
Hardware Tested:	The DUART 1 (SC2681) is tested.
Data Returned:	The number of the failing test, the actual character received, and the character transmitted are returned.

==========

Test Number:	12
Function:	This test checks DTR for Port 2.
Procedure:	The DTR is asserted for Port 2 by writing to the general purpose Output register. The DTR is verified to be asserted by reading the general purpose Input register. Bits 1 and 2 of the general purpose Input and Output registers are tied together. The data read from the Input register is inverted.
Hardware Tested:	The DUART 1 (SC2681) is tested.
Data Returned:	The number of the failing test, the actual value in the Input register, and the expected value in the Input register are returned.

==========

Test Numbers:	13 through 17
Function:	These tests verify that DUART 1, Port 2 can externally loop ASCII characters at various baud rates.
Procedure:	Port 2 is initialized for external loop around using 7 bits, no parity, and 2 Stop bits. The ASCII characters are transmitted at 300, 1200, 4800, 9600, and 19200 baud. The received character is verified to match the one transmitted.
Hardware Tested:	The DUART 1 (SC2681) is tested.
Data Returned:	The number of the failing test, the ASCII value received, and the ASCII value transmitted are returned.

Test Numbers: 18 through 22

Function: These tests verify that DUART 1, Port 2 can externally loop ASCII characters at various baud rates.

Procedure: Port 2 is initialized for external loop around using 8 bits, no parity, and 2 Stop bits. The ASCII characters are transmitted at 300, 1200, 4800, 9600, and 19200 baud. The received character is verified to match the one transmitted.

Hardware Tested: The DUART 1 (SC2681) is tested.

Data Returned: The number of the failing test, the ASCII value received, and the ASCII value transmitted are returned.

==========

Test Number: (23)

Function: This test verifies the ability of the DUART to send an interrupt when the Receive Data register on DUART 1, Port 1 is enabled and a character is received.

Procedure: Port 1 is initialized to transmit and receive interrupts. The interrupts consist of 7 bits, no parity, and 2 Stop bits. The baud rate is set to 9600 bits per second (bps). Set the DUART to send an interrupt when the Receive Data register is enabled and has received a character; then check that the interrupt was correctly received.

Hardware Tested: The DUART 1 (SC2681) is tested.

Data Returned: The number of the failing test, the ASCII character received, and the ASCII character transmitted are returned.

==========

Test Number: (24)

Function: This test verifies the ability of the DUART to send an interrupt when the Transmit Data register on DUART 1, Port 1 is enabled and after a character has been transmitted.

Procedure: Port 1 is initialized to transmit and receive interrupts. The interrupts consist of 7 bits, no parity, and 2 Stop bits. The baud rate is set to 9600 bps. Set the DUART to send an interrupt when the Transmit Data register is enabled and a character has been transmitted; then check that the interrupt was correctly received.

Hardware Tested: The DUART 1 (SC2681) is tested.

Data Returned: The number of the failing test, the ASCII character received, and the ASCII character transmitted are returned.

Test Number: (25)

Function: This test verifies the ability of the DUART to send an interrupt when the Receive Data register on DUART 1, Port 2 is enabled and a character is received.

Procedure: Port 2 is initialized to transmit and receive interrupts. The interrupts consist of 7 bits, no parity, and 2 Stop bits. The baud rate is set to 9600 bps. Set the DUART to send an interrupt when the Receive Data register is enabled and has received a character; then check that the interrupt was correctly received.

Hardware Tested: The DUART 1 (SC2681) is tested.

Data Returned: The number of the failing test, the ASCII character received, and the ASCII character transmitted are returned.

==========

Test Number: (26)

Function: This test verifies the ability of the DUART to send an interrupt when the Transmit Data register on DUART 1, Port 2 is enabled and after a character has been transmitted.

Procedure: Port 2 is initialized to transmit and receive interrupts. The interrupts consist of 7 bits, no parity, and 2 Stop bits. The baud rate is set to 9600 bps. Set the DUART to send an interrupt when the Transmit Data register is enabled and a character has been transmitted; then check that the interrupt was correctly received.

Hardware Tested: The DUART 1 (SC2681) is tested.

Data Returned: The number of the failing test, the ASCII character received, and the ASCII character transmitted are returned.

==========

Test Number: (27)

Function: This test verifies the ability of the DUART to send an interrupt when Receive buffer full (3 characters) is detected on DUART 1, Port 1.

Procedure: Port 1 is initialized to transmit and receive interrupts. The interrupts consist of 8 bits, no parity, and 2 Stop bits. The baud rate is set to 9600 bps. Transmit three characters over DUART 1, Port 1. Check for an interrupt. Verify that the character transmission is correct.

Hardware Tested: The DUART 1 (SC2681) is tested.

Data Returned: The number of the failing test, the ASCII character received, and the ASCII character transmitted are returned.

Test Number: (28)

Function: This test verifies the ability of the DUART to send an interrupt when Receive buffer full (3 characters) is detected on DUART 1, Port 2.

Procedure: Port 2 is initialized to transmit and receive interrupts. The interrupts consist of 8 bits, no parity, and 2 Stop bits. The baud rate is set to 9600 bps. Transmit three characters over DUART 1, Port 2. Check for an interrupt. Verify that the character transmission is correct.

Hardware Tested: The DUART 1 (SC2681) is tested.

Data Returned: The number of the failing test, the ASCII character received, and the ASCII character transmitted are returned.

==========

Test Number: (29)

Function: This test verifies the ability of the DUART to send an interrupt when a break character is detected on DUART 1, Port 1.

Procedure: Port 1 is initialized to transmit and receive interrupts. The interrupts consist of 7 bits, no parity, and 2 Stop bits. The baud rate is set to 9600 bps. Set the DUART to send an interrupt when a break character is detected. Enable transmitter and receiver. Start break and check that the interrupt was correctly received.

Hardware Tested: The DUART 1 (SC2681) is tested.

Data Returned: The number of the failing test, the ASCII character received, and the ASCII character transmitted are returned.

==========

Test Number: (30)

Function: This test verifies the ability of the DUART to send an interrupt when a break character is detected on DUART 1, Port 2.

Procedure: Port 2 is initialized to transmit and receive interrupts. The interrupts consist of 7 bits, no parity, and 2 Stop bits. The baud rate is set to 9600 bps. Set the DUART to send an interrupt when a break character is detected. Enable transmitter and receiver. Start break and check that the interrupt was correctly received.

Hardware Tested: The DUART 1 (SC2681) is tested.

Data Returned: The number of the failing test, the ASCII character received, and the ASCII character transmitted are returned.

Test Number:	(31)
Function:	This test verifies the ability of the DUART to send an interrupt when "Counter Ready" is detected on DUART 1.
Procedure:	Port 2 is initialized to transmit and receive interrupts. The interrupts consist of 7 bits, no parity, and 2 Stop bits. The baud rate is set to 9600 bps. Set the DUART to send an interrupt when "Counter Ready" is detected. Enable transmitter and receiver. Reset and start the counter. Check that the interrupt was correctly received.
Hardware Tested:	The DUART 1 (SC2681) is tested.
Data Returned:	The number of the failing test, the ASCII character received, and the ASCII character transmitted are returned.

Phase #21 — PORTS—DUART 0 Loop Sanity

Phase Name:	PORTS—DUART (SC2681) 0 Loop Sanity (duart0_3)
Type:	Normal
Function:	This phase verifies the sanity of Dual Universal Asynchronous Receiver/Transmitter (DUART) 0.
Tests:	Test 1 — checks the internal loop of ASCII characters through DUART 0, Port 1 at 19200 baud. Test 2 — checks the internal loop of ASCII characters through DUART 0, Port 2 at 19200 baud. Test (3) — checks the ability of the DUART to send an interrupt when the Receive Data register on DUART 0, Port 1 is enabled and a character is received. Test (4) — checks the ability of the DUART to send an interrupt when the Transmit Data register on DUART 0, Port 1 is enabled and after a character is transmitted. Test (5) — checks the ability of the DUART to send an interrupt when the Receive Data register on DUART 0, Port 2 is enabled and a character is received. Test (6) — checks the ability of the DUART to send an interrupt when the Transmit Data register on DUART 0, Port 2 is enabled and after a character is transmitted.
Time:	2 seconds
Warnings:	None
Notes:	If any test in this phase fails, the following hardware may be faulty: ■ DUART (SC2681) ■ DUART interface to the PORTS Address/Data bus ■ PORTS RAM/ROM.

Phase #21 Tests

Test Number:	1
Function:	This test verifies that the DUART 0, Port 1 can internally loop American Standard Code for Information Interchange (ASCII) characters at 19200 baud.
Procedure:	The DUART is initialized for internal loop around using 7 bits and no parity. The baud is set at 19200. The transmitted and received characters are verified to match.
Hardware Tested:	The DUART 0 (SC2681) is tested.
Data Returned:	The number of the failing test, the ASCII character received, and the ASCII character transmitted are returned.

Test Number: 2

Function: This test verifies that the DUART 0, Port 2 can internally loop ASCII characters at 19200 baud.

Procedure: The DUART is initialized for internal loop around using 7 bits and no parity. The baud is set at 19200. The transmitted and received characters are verified to match.

Hardware Tested: The DUART 0 (SC2681) is tested.

Data Returned: The number of the failing test, the ASCII character received, and the ASCII character transmitted are returned.

==========

Test Number: (3)

Function: This test verifies the ability of the DUART to send an interrupt when the Receive Data register on DUART 0, Port 1 is enabled and a character is received.

Procedure: Port 1 is initialized to transmit and receive interrupts. The interrupts consist of 7 bits, no parity, and 2 Stop bits. The baud rate is set to 9600 bits per second (bps). Set the DUART to send an interrupt when the Receive Data register is enabled and has received a character; then check that the interrupt was correctly received.

Hardware Tested: The DUART 0 (SC2681) is tested.

Data Returned: The number of the failing test, the ASCII character received, and the ASCII character transmitted are returned.

==========

Test Number: (4)

Function: This test verifies the ability of the DUART to send an interrupt when the Transmit Data register on DUART 0, Port 1 is enabled and after a character is transmitted.

Procedure: Port 1 is initialized to transmit and receive interrupts. The interrupts consist of 7 bits, no parity, and 2 Stop bits. The baud rate is set to 9600 bps. Set the DUART to send an interrupt when the Transmit Data register is enabled and a character has been transmitted; then check that the interrupt was correctly received.

Hardware Tested: The DUART 0 (SC2681) is tested.

Data Returned: The number of the failing test, the ASCII character received, and the ASCII character transmitted are returned.

Test Number: (5)

Function: This test verifies the ability of the DUART to send an interrupt when the Receive Data register on DUART 0, Port 2 is enabled and a character is received.

Procedure: Port 2 is initialized to transmit and receive interrupts. The interrupts consist of 7 bits, no parity, and 2 Stop bits. The baud rate is set to 9600 bps. Set the DUART to send an interrupt when the Receive Data register is enabled and has received a character; then check that the interrupt was correctly received.

Hardware Tested: The DUART 0 (SC2681) is tested.

Data Returned: The number of the failing test, the ASCII character received, and the ASCII character transmitted are returned.

==========

Test Number: (6)

Function: This test verifies the ability of the DUART to send an interrupt when the Transmit Data register on DUART 0, Port 2 is enabled and after a character is transmitted.

Procedure: Port 2 is initialized to transmit and receive interrupts. The interrupts consist of 7 bits, no parity, and 2 Stop bits. The baud rate is set to 9600 bps. Set the DUART to send an interrupt when the Transmit Data register is enabled and a character has been transmitted; then check that the interrupt was correctly received.

Hardware Tested: The DUART 0 (SC2681) is tested.

Data Returned: The number of the failing test, the ASCII character received, and the ASCII character transmitted are returned.

Phase #22 — PORTS—DUART 1 Loop Sanity

Phase Name:	PORTS—DUART (SC2681) 1 Loop Sanity (duart1_3)
Type:	Normal
Function:	This phase verifies the sanity of Dual Universal Asynchronous Receiver/Transmitter (DUART) 1.
Tests:	Test 1 — checks the internal loop of ASCII characters through DUART 1, Port 1 at 19200 baud. Test 2 — checks the internal loop of ASCII characters through DUART 1, Port 2 at 19200 baud. Test (3) — checks the ability of the DUART to send an interrupt when the Receive Data register on DUART 1, Port 1 is enabled and a character is received. Test (4) — checks the ability of the DUART to send an interrupt when the Transmit Data register on DUART 1, Port 1 is enabled and after a character is transmitted. Test (5) — checks the ability of the DUART to send an interrupt when the Receive Data register on DUART 1, Port 2 is enabled and a character is received. Test (6) — checks the ability of the DUART to send an interrupt when the Transmit Data register on DUART 1, Port 2 is enabled and after a character is transmitted.
Time:	2 seconds
Warnings:	None.
Notes:	If any test in this phase fails, the following hardware may be faulty: ■ DUART (SC2681) ■ DUART interface to the PORTS Address/Data bus ■ PORTS RAM/ROM.

Phase #22 Tests

Test Number:	1
Function:	This test verifies that the DUART 1, Port 1 can internally loop American Standard Code for Information Interchange (ASCII) characters at 19200 baud.
Procedure:	The DUART is initialized for internal loop around using 7 bits and no parity. The baud is set at 19200. The transmitted and received characters are verified to match.
Hardware Tested:	The DUART 1 (SC2681) is tested.
Data Returned:	The number of the failing test, the ASCII character received, and the ASCII character transmitted are returned.

Test Number: 2

Function: This test verifies that the DUART 1, Port 2 can internally loop ASCII characters at 19200 baud.

Procedure: The DUART is initialized for internal loop around using 7 bits and no parity. The baud is set at 19200. The transmitted and received characters are verified to match.

Hardware Tested: The DUART 1 (SC2681) is tested.

Data Returned: The number of the failing test, the ASCII character received, and the ASCII character transmitted are returned.

==========

Test Number: (3)

Function: This test verifies the ability of the DUART to send an interrupt when the Receive Data register on DUART 1, Port 1 is enabled and a character is received.

Procedure: Port 1 is initialized to transmit and receive interrupts. The interrupts consist of 7 bits, no parity, and 2 Stop bits. The baud rate is set to 9600 bits per second (bps). Set the DUART to send an interrupt when the Receive Data register is enabled and has received a character; then check that the interrupt was correctly received.

Hardware Tested: The DUART 1 (SC2681) is tested.

Data Returned: The number of the failing test, the ASCII character received, and the ASCII character transmitted are returned.

==========

Test Number: (4)

Function: This test verifies the ability of the DUART to send an interrupt when the Transmit Data register on DUART 1, Port 1 is enabled and after a character is transmitted.

Procedure: Port 1 is initialized to transmit and receive interrupts. The interrupts consist of 7 bits, no parity, and 2 Stop bits. The baud rate is set to 9600 bps. Set the DUART to send an interrupt when the Transmit Data register is enabled and a character has been transmitted; then check that the interrupt was correctly received.

Hardware Tested: The DUART 1 (SC2681) is tested.

Data Returned: The number of the failing test, the ASCII character received, and the ASCII character transmitted are returned.

Test Number: (5)

Function: This test verifies the ability of the DUART to send an interrupt when the Receive Data register on DUART 1, Port 2 is enabled and a character is received.

Procedure: Port 2 is initialized to transmit and receive interrupts. The interrupts consist of 7 bits, no parity, and 2 Stop bits. The baud rate is set to 9600 bps. Set the DUART to send an interrupt when the Receive Data register is enabled and has received a character; then check that the interrupt was correctly received.

Hardware Tested: The DUART 1 (SC2681) is tested.

Data Returned: The number of the failing test, the ASCII character received, and the ASCII character transmitted are returned.

==========

Test Number: (6)

Function: This test verifies the ability of the DUART to send an interrupt when the Transmit Data register on DUART 1, Port 2 is enabled and after a character is transmitted.

Procedure: Port 2 is initialized to transmit and receive interrupts. The interrupts consist of 7 bits, no parity, and 2 Stop bits. The baud rate is set to 9600 bps. Set the DUART to send an interrupt when the Transmit Data register is enabled and a character has been transmitted; then check that the interrupt was correctly received.

Hardware Tested: The DUART 1 (SC2681) is tested.

Data Returned: The number of the failing test, the ASCII character received, and the ASCII character transmitted are returned.

Chapter 9: EPORTS Diagnostics

Introduction

This chapter contains the diagnostic phases and tests for the 3B2 computer Enhanced Peripheral Ports Controller (EPORTS) card. The EPORTS card provides access to the 3B2 computer for peripheral equipment, such as modems, terminals and serial printers. Each EPORTS card contains eight serial ports. The following components are found on the EPORTS card:

- INTEL 80186 Central Processing Unit (CPU)
- Serial Communication Controller (SCC)
- Random Access Memory (RAM)
- Programmable Read Only Memory (PROM)
- Direct Memory Access Controllers (DMACs)
- Full duplex capability RS-232-C ports.

An important external component of the EPORTS card is the 3B2 computer Input/Output (I/O) bus. The EPORTS card uses the 3B2 computer I/O bus to communicate with the System Board (SBD).

If your 3B2 computer is equipped with an EPORTS card, the Equipped Device Table (EDT) should list it. You can use the diagnostic monitor command "show" (DGMON > **s**) to print a copy of the EDT. If the EPORTS card is not listed in the EDT and a VOID or NULL is listed, one or more of the following hardware devices on the EPORTS card may be faulty:

- INTEL 80186 Microprocessor
- EPORTS Read Only Memory (ROM)
- EPORTS ID/Vector register
- EPORTS Address/Data bus
- EPORTS interface to the 3B2 computer I/O bus.

Twenty-one diagnostic phases run tests on all major EPORTS card components. The Table of Contents listing on the previous page will help you locate the descriptions for each EPORTS card phase and its associated tests. The phase and test descriptions are organized numerically in the same order that the phases are run on the 3B2 computer.

Phase Descriptions

Phase #1 — EPORTS — CIO and Peripheral Sanity

Phase Name:	EPORTS — CIO and Peripheral Sanity
Type:	Normal
Function:	This phase verifies the integrity of the Common I/O (CIO) queue.
Test:	Test 1 — verifies that the diagnostic results can be transferred to the System Board (SBD).
Time:	1 second
Warnings:	None
Notes:	None

Phase #1 Test

Test Number:	1
Function:	This test verifies that the CIO hardware and firmware are functioning properly.
Procedure:	This test uses the following standard procedure:

1. The I/O slot number of the EPORTS card in the 3B2 computer is determined.
2. The ports are reset.
3. The sysgen data block is initialized.
4. The ports are initialized (sysgen) by sending express [Interrupt 0 (INT0)] and attention (INT1) interrupts sequentially.
5. The X86 code is downloaded by using the CIO firmware command [Download Memory (DLM)].
6. Execution of the phase is started by using the CIO firmware command [Force Call to Function (FCF)].
7. A function call to "**phasend( )**" is made when the phase is complete. Phasend returns the test results to the SBD.

Hardware Tested:	The EPORTS interface to the 3B2 computer I/O bus is tested.
Data Returned:	None

Phase #2 — EPORTS — Upper RAM Verification

Phase Name:	EPORTS — Upper RAM Verification (ram_h)
Type:	Demand
Function:	This phase diagnoses and reports any errors in the operation of the EPORTS RAM and refresh circuitry. This phase tests the upper 64 kilobytes of peripheral RAM.
Tests:	Tests 1 through 4 — write a pattern into RAM; then read it back and write the next pattern. Test 5 — fills RAM with a random pattern, waits 5 seconds, then verifies the write. This test checks the refresh circuitry. Test 6 — check for any interaction problems between bit cells in RAM.
Time:	30 seconds
Warnings:	None
Notes:	None

Phase #2 Tests

Test Numbers:	1 through 6
Function:	These tests verify the operation of the upper 16 kilobytes of EPORTS RAM (addresses 0x4000 to 0x7fff).
Procedure:	A known data pattern is written to every memory location (upper half) and verified with a read.
Hardware Tested:	The EPORTS RAM (upper half) is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

Phase #3 — EPORTS — Lower RAM Verification

Phase Name:	EPORTS — Lower RAM Verification (ram_l)
Type:	Demand
Function:	This phase diagnoses and reports any errors in the operation of the EPORTS RAM and refresh circuitry. This phase tests the lower 64 kilobytes of peripheral RAM.
Tests:	Tests 1 through 4 — write a pattern into RAM; then read it back and write the next pattern. Test 5 — fills RAM with a random pattern, waits 5 seconds, then verifies the write. This test checks the refresh circuitry. Test 6 — check for any interaction problems between bit cells in RAM.
Time:	30 seconds
Warnings:	None
Notes:	None

Phase #3 Tests

Test Numbers:	1 through 6
Function:	These tests verify that the lower 16 kilobytes of EPORTS RAM (addresses 0x0000 to 0x3fff) are functioning properly.
Procedure:	A known data pattern is written to every memory location (lower half) and verified with a read.
Hardware Tested:	The EPORTS RAM (lower half) is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

Phase #4 — EPORTS — ROM Check Sum

Phase Name:	EPORTS — ROM Check Sum (rom)
Type:	Demand
Function:	This phase verifies the integrity of the EPORTS ROM.
Test:	Test 1 — calculates the check sum of the EPORTS ROM.
Time:	3 seconds
Warnings:	None
Notes:	None

Phase #4 Test

Test Number:	1
Function:	This test verifies the integrity of the EPORTS ROM.
Procedure:	A check sum is calculated by reading the EPORTS ROM. The calculated check sum is compared with the check sum stored in ROM when the ROM was initially programmed.
Hardware Tested:	The EPORTS ROM is tested.
Data Returned:	The number of test that failed, the actual check sum value, and the expected check sum value are returned.

Phase #5 — EPORTS — Upper Chip Select Registers

Phase Name:	EPORTS — Upper Chip Select Registers (cpu_1)
Type:	Demand
Function:	This phase tests the Upper Memory Chip Select (UMCS), Peripheral Access Chip Select (PACS), and Middle Peripheral Chip Select (MPCS) on the INTEL 80186 CPU.
Tests:	Test 1 through 12 — check the UMCS register. Tests 13 through 29 — check the PACS register. Tests 30 through 37 — no tests run. Tests 38 through 48 — check the MPCS register.
Time:	2 seconds
Warnings:	None
Notes:	None

Phase #5 Tests

Test Numbers:	1 through 12
Function:	These tests verify that the UMCS register is functional.
Procedure:	Write a valid data pattern to the register, read the register, and compare the value read with the value that was written.
Hardware Tested:	The UMCS register is tested.
Data Returned:	Supplemental Data: The value that was written. Raw Data: The value that was read.
Notes:	The values written to the register are 0xc038, 0xe038, 0xf038, 0xf838, 0xfc38, 0xfe38, 0xff38, 0xffb8, and 0xfff8.

==========

Test Numbers:	13 through 29
Function:	These tests verify that the PACS register is functional.
Procedure:	Write a valid data pattern to the register, read the register, and compare the value read with the value that was written.
Hardware Tested:	The PACS register is tested.
Data Returned:	Supplemental Data: The value that was written. Raw Data: The value that was read.
Notes:	The values written to the register are 0x0038, 0x003a, 0x003c, 0x0078, 0x00b8, 0x0138, 0x0238, 0x0438, 0x0838, 0x1038, 0x2038, 0x4038, and 0x8038.

Test Numbers:	38 through 48
Function:	These tests verify that the MPCS register is functional.
Procedure:	Write a valid data pattern to the register, read the register, and compare the value read with the value that was written.
Hardware Tested:	The MPCS register is tested.
Data Returned:	Supplemental Data: The value that was written. Raw Data: The value that was read.
Notes:	The values written to the register are 0x80b8, 0x80b9, 0x80ba, 0x80bc, 0x80f8, 0x81b8, 0x82b8, 0x84b8, 0x88b8, 0x90b8, 0xa0b8, and 0xc0b8. The INTEL 80186 INT2 and INT3 Control registers are tested.

Phase #6 — EPORTS — DMA Control Registers

Phase Name:	EPORTS — DMA Control Registers (cpu_2)
Type:	Demand
Function:	This phase diagnoses and reports any errors in the operation of the INTEL 80186 CPU.
Tests:	Test 1 through 14 — check the DMA0 Control register. Tests 15 through 31 — check the DMA0 Terminal Count register. Tests 32 through 48 — check the DMA0 Destination (low) register. Tests 49 through 53 — no tests run. Tests 54 through 70 — check the DMA0 Source (low) register. Tests 71 through 75 — no tests run. Tests 76 through 90 — check the DMA1 Control register. Tests 91 through 107 — check the DMA1 Terminal Count register. Tests 108 through 124 — check the DMA1 Destination (low) register. Tests 125 through 141 — check the DMA1 Source (low) register.
Time:	2 seconds
Warnings:	None
Notes:	None

Phase #6 Tests

Test Numbers:	1 through 14
Function:	These tests verify that the Direct Memory Access 0 (DMA0) Control register is functional.
Procedure:	Write a valid data pattern to the register, read the registers, and compare the value read with the expected value.
Hardware Tested:	The DMA0 Control register is tested.
Data Returned:	Supplemental Data: The value that was written. Raw Data: The value that was read.
Notes:	The values written to the register are 0x8000, 0x4000, 0x2000, 0x1000, 0x0800, 0x0400, 0x0200, 0x0100, 0x0080, 0x0040, 0x0020, 0x0010, 0x0001, and 0x0000.

==========

Test Numbers:	15 through 31
Function:	These tests verify that the DMA0 Terminal Count register is functional.
Procedure:	Write a valid data pattern to the register, read the register, and compare the value read with the expected value.
Hardware Tested:	The DMA0 Terminal Count register is tested.
Data Returned:	Supplemental Data: The value that was written. Raw Data: The value that was read.
Notes:	The values written to the register are 0x0000 and walk a one through a field of 16 zeros.

==========

Test Numbers:	32 through 48
Function:	These tests verify that the DMA0 Destination (low) register is functional.
Procedure:	Write a valid data pattern to the register, read the register, and compare the value read with the expected value.
Hardware Tested:	The DMA0 Destination (low) register is tested.
Data Returned:	Supplemental Data: The value that was written. Raw Data: The value that was read.
Notes:	The values written to the register are 0x0000 and walk a one through a field of 16 zeros.

Test Numbers:	54 through 70
Function:	These tests verify that the DMA0 Source (low) register is functional.
Procedure:	Write a valid data pattern to the register, read the register, and compare the value read with the expected value.
Hardware Tested:	The DMA0 Source (low) register is tested.
Data Returned:	Supplemental Data: The value that was written. Raw Data: The value that was read.
Notes:	The values written to the register are 0x00 and walk a one through a field of 16 zeros.

==========

Test Numbers:	76 through 90
Function:	These tests verify that the DMA1 Control register is functional.
Procedure:	Write a valid data pattern to the register, read the register, and compare the value read with the expected value.
Hardware Tested:	The DMA1 Control register is tested.
Data Returned:	Supplemental Data: The value that was written. Raw Data: The value that was read.
Notes:	The values written to the register are 0x8000, 0x4000, 0x2000, 0x1000, 0x0800, 0x0400, 0x0200, 0x0100, 0x0080, 0x0040, 0x0020, 0x0010, 0x0001, and 0x0000.

==========

Test Numbers:	91 through 107
Function:	The tests verify that the DMA1 Terminal Count register is functional.
Procedure:	Write a valid data pattern to the register, read the register, and compare the value read with the expected value.
Hardware Tested:	The DMA1 Terminal Count register is tested.
Data Returned:	Supplemental Data: The value that was written. Raw Data: The value that was read.
Notes:	The values written to the register are 0x0000 and walk a one through a field of 16 zeros.

Test Numbers:	108 through 124
Function:	These tests verify that the DMA1 Destination (low) register is functional.
Procedure:	Write a valid data pattern to the register, read the register, and compare the value read with the expected value.
Hardware Tested:	The DMA1 Destination (low) register is tested.
Data Returned:	Supplemental Data: The value that was written. Raw Data: The value that was read.
Notes:	The values written to the register are 0x0000 and walk a one through a field of 16 zeros.

==========

Test Numbers:	125 through 141
Function:	These tests verify that the DMA1 Source (low) register is functional.
Procedure:	Write a valid data pattern to the register, read the register, and compare the value read with the expected value.
Hardware Tested:	The DMA1 Source (low) register is tested.
Data Returned:	Supplemental Data: The value that was written. Raw Data: The value that was read.
Notes:	The values written to the register are 0x0000 and walk a one through a field of 16 zeros.

Phase #7 — EPORTS — CPU Writable Registers

Phase Name:	EPORTS — CPU Writable Registers (cpu_3)
Type:	Demand
Function:	This phase tests the operation of the internal Timer registers of the INTEL 80186 CPU.
Tests:	Tests 1 through 17 — check the Timer0 Count register. Tests 18 through 34 — check the Timer0 MCA register. Tests 35 through 51 — check the Timer0 MCB register. Tests 52 through 59 — check the Timer0 Mode register. Tests 60 through 120 — no tests run. Tests 121 through 137 — check the Timer2 Count register. Tests 138 through 154 — check the Timer2 MCA register. Tests 155 through 158 — check the Timer2 Mode register.
Time:	4 seconds
Warnings:	None
Notes:	None

Phase #7 Tests

Test Numbers:	1 through 17
Function:	These tests verify that the Timer0 Count register is functional.
Procedure:	Write a valid data pattern to the register, read the register, and compare the value read with the value that was written.
Hardware Tested:	The Timer0 Count register is tested.
Data Returned:	Supplemental Data: The value that was written. Raw Data: The value that was read.
Notes:	The values written to the register are 0x0000 and walk a one through a field of 16 zeros.

Test Numbers:	18 through 34
Function:	These tests verify that the Timer0 Maximum Count A (MCA) register is functional.
Procedure:	Write a valid data pattern to the register, read the register, and compare the value read with the value that was written.
Hardware Tested:	The Timer0 MCA register is tested.
Data Returned:	Supplemental Data: The value that was written. Raw Data: The value that was read.
Notes:	The values written to the register are 0x0000 and walk a one through a field of 16 zeros.

==========

Test Numbers:	35 through 51
Function:	These tests verify that the Timer0 Maximum Count B (MCB) register is functional.
Procedure:	Write a valid data pattern to the register, read the register, and compare the value read with the value that was written.
Hardware Tested:	The Timer0 MCB register is tested.
Data Returned:	Supplemental Data: The value that was written. Raw Data: The value that was read.
Notes:	The values written to the register are 0x0000 and walk a one through a field of 16 zeros.

==========

Test Numbers:	52 through 59
Function:	These tests verify that the Timer0 Mode register is functional.
Procedure:	Write a valid data pattern to the register, read the register, and compare the value read with the value that was written.
Hardware Tested:	The Timer0 Mode register is tested.
Data Returned:	Supplemental Data: The value that was written. Raw Data: The value that was read.
Notes:	The values written to the register are 0x0000, 0x0001, 0x0002, 0x0004, 0x0008, 0x0010, 0x2000, 0x4000, and 0x8000.

Test Numbers:	121 through 137
Function:	These tests verify that the Timer2 Count register is functional.
Procedure:	Write a valid data pattern to the register, read the register, and compare the value read with the value that was written.
Hardware Tested:	The Timer2 Count register is tested.
Data Returned:	Supplemental Data: The value that was written. Raw Data: The value that was read.
Notes:	The values written to the register are 0x0000 and walk a one through a field of 16 zeros.

==========

Test Numbers:	138 through 154
Function:	These tests verify that the Timer2 MCA register is functional.
Procedure:	Write a valid data pattern to the register, read the register, and compare the value read with the value that was written.
Hardware Tested:	The Timer2 MCA register is tested.
Data Returned:	Supplemental Data: The value that was written. Raw Data: The value that was read.
Notes:	The values written to the register are 0x0000 and walk a one through a field of 16 zeros.

==========

Test Numbers:	155 through 158
Function:	These tests verify that the Timer2 Mode register is functional.
Procedure:	Write a valid data pattern to the register, read the register, and compare the value read with the value that was written.
Hardware Tested:	The Timer2 Mode register is tested.
Data Returned:	Supplemental Data: The value that was written. Raw Data: The value that was read.
Notes:	The values written to the register are 0x0000, 0x0x0020, 0x0x2000, and 0xc001 - read 0x8021.

Phase #8 — EPORTS — Interrupt Control Registers

Phase Name:	EPORTS — Interrupt Control Registers (cpu_4)
Type:	Demand
Function:	This phase verifies that the internal registers of the INTEL 80186 CPU Interrupt Controller (IC) are functioning properly.
Tests:	Tests 1 through 8 — check the IC In-service register. Tests 9 through 11 — check the Interrupt Request register. Tests 12 through 20 — check the Interrupt Mask register. Tests 21 through 23 — check the Interrupt Priority Mask register. Tests 24 through 28 — check the Interrupt Status register. Tests 29 through 40 — check the DMA0 and DMA1 Control registers. Tests 41 through 56 — check the INT0 and INT1 Control registers. Tests 57 through 69 — check the INT2 and INT3 Control registers.
Time:	2 seconds
Warnings:	None
Notes:	None

Phase #8 Tests

Test Numbers:	1 through 8
Function:	These tests verify that the IC In-service register is functional.
Procedure:	Write a valid data pattern to the register, read the register, and compare the value read with the value that was written.
Hardware Tested:	The IC In-service register is tested.
Data Returned:	Supplemental Data: The value that was written. Raw Data: The value that was read.
Notes:	The values written to the register are 0x0000, 0x0001, 0x0004, 0x0008, 0x0010, 0x0020, 0x0040, 0x0080, 0x0100, and 0x0200.

Test Numbers:	9 through 11
Function:	These tests verify that the Interrupt Request register is functional.
Procedure:	Write a valid data pattern to the register, read the register, and compare the value read with the value that was written.
Hardware Tested:	The Interrupt Request register is tested.
Data Returned:	Supplemental Data: The value that was written. Raw Data: The value that was read.
Notes:	The values written to the register are 0x0, 0x0004, and 0x0008.

==========

Test Numbers:	12 through 20
Function:	These tests verify that the Interrupt Mask register is functional.
Procedure:	Write a valid data pattern to the register, read the register, and compare the value read with the value that was written.
Hardware Tested:	The Interrupt Mask register is tested.
Data Returned:	Supplemental Data: The value that was written. Raw Data: The value that was read.
Notes:	The values written to the register are 0x0, 0x0001, 0x0004, 0x0008, 0x0010, 0x0020, 0x0040, 0x0080, and 0x0100.

==========

Test Numbers:	21 through 23
Function:	These tests verify that the Interrupt Priority Mask register is functional.
Procedure:	Write a valid data pattern to the register, read the register, and compare the value read with the value that was written.
Hardware Tested:	The Interrupt Priority Mask register is tested.
Data Returned:	Supplemental Data: The value that was written. Raw Data: The value that was read.
Notes:	The values written to the register are 0x0, 0x0001, 0x0002, and 0x0004.

Test Numbers: 24 through 28

Function: These tests verify that the Interrupt Status register is functional.

Procedure: Write a valid data pattern to the register, read the register, and compare the value read with the value that was written.

Hardware Tested: The Interrupt Status register is tested.

Data Returned: Supplemental Data: The value that was written.
Raw Data: The value that was read.

Notes: The values written to the register are 0x0, 0x0001, 0x0002, 0x0004, and 0x8000.

==========

Test Numbers: 29 through 40

Function: These tests verify that the Direct Memory Access 0 (DMA0) and DMA1 Control registers are functional.

Procedure: Write a valid data pattern to the register, read the register, and compare the value read with the value that was written.

Hardware Tested: The DMA0 and DMA1 Control register are tested.

Data Returned: Supplemental Data: The value that was written.
Raw Data: The value that was read.

Notes: The values written to the register are 0x0, 0x0001, 0x0002, and 0x0004.

==========

Test Numbers: 41 through 56

Function: These tests verify that the Interrupt 0 (INT0) and INT1 Control registers are functional.

Procedure: Write a valid data pattern to the register, read the register, and compare the value read with the value that was written.

Hardware Tested: The INT0 and INT1 Control registers are tested.

Data Returned: Supplemental Data: The value that was written.
Raw Data: The value that was read.

Notes: The values written to the register are 0x0, 0x0001, 0x0002, 0x0004, 0x0008, 0x0010, 0x0020, 0x0040, and 0x0080.

==========

Test Numbers: 57 through 69

Function: These tests verify that the INT2 and INT3 Control registers are functional.

Procedure: Write a valid data pattern to the register, read the register, and compare the value read with the value that was written.

Hardware Tested: The INT2 and INT3 Control registers are tested.

Data Returned: Supplemental Data: The value that was written.
Raw Data: The value that was read.

Notes: The values written to the register are 0x0, 0x0001, 0x0002, 0x0004, 0x0008, and 0x0010.

Phase #9 — EPORTS — Lower Chip Select Register

Phase Name:	EPORTS — Lower Chip Select Register (cpu_5)
Type:	Demand
Function:	This phase verifies that the Lower Memory Chip Select (LMCS) register of the INTEL 80186 CPU is functioning properly.
Tests:	Tests 1 through 7 — check the LMCS register.
Time:	4 seconds
Warnings:	None
Notes:	This phase is part of a "not yet complete" diagnostic package and its operation cannot be guaranteed outside of this package.

Phase #9 Tests

Test Numbers:	1 through 7
Function:	These tests verify that the LMCS register is functional.
Procedure:	Write a valid data pattern to the register, read the register, and compare the value read with the value that was written.
Hardware Tested:	The LMCS register is tested.
Data Returned:	Supplemental Data: The value that was written. Raw Data: The value that was read.
Notes:	The values written to the register are 0x00f8, 0x01f8, 0x03f8, 0x07f8, 0x0ff8, 0x1ff8, and 0x3ff8.

Phase #10 — EPORTS — Programmed Input/Output (PIO) Byte Transfers

Phase Name:	EPORTS — PIO Byte Transfers (pio_1)
Type:	Demand
Function:	This phase tests byte transfers to the 3B2 computer I/O bus.
Tests:	Tests 1 through 8 — verifies the operation of the I/O bus interface.
Time:	80 seconds
Warnings:	None

Phase #10 Tests

Test Numbers:	1 through 8
Function:	These tests verify the operation of the I/O bus interface.
Procedure:	Write a one through a field of zeros in all memory locations of a page of System Board (SBD) RAM.
Hardware Tested:	The EPORTS interface to the 3B2 computer I/O bus is tested.
Data Returned:	The data returned are the test count that failed, the address of the failure, and the data pattern that failed. The test count is the bit that was written. The address is represented as 0x8 9????.
Notes:	Walk a one through a field of 8 zeros is the value written to the memory location.

Phase #11 — EPORTS — Programmed Input/Output (PIO) Word Transfers

Phase Name:	EPORTS — PIO Word Transfer (pio_2)
Type:	Demand
Function:	This phase tests the I/O Address/Data bus.
Tests:	Tests 1 through 16 — walk a one through a field of 16 zeros in every memory location of one page of memory.
Time:	80 seconds
Warnings:	None
Notes:	None

Phase #11 Tests

Test Numbers:	1 through 16
Function:	These tests verify the operation of the 3B2 computer I/O bus.
Procedure:	Write a one through a field of zeros in all locations of one page of System Board (SBD) RAM.
Hardware Tested:	The EPORTS interface to the 3B2 computer I/O bus is tested.
Data Returned:	The data returned are the test count that failed, the address of the failure, and the data pattern that failed. The test count is the bit that was written. The address is represented as 0x8 9????.
Notes:	Walk a one through a field of 16 zeros is the value written to the register.

Phase #12 — EPORTS — DMA Byte Transfers

Phase Name:	EPORTS — DMA Byte Transfers (dmabyt)
Type:	Demand
Function:	This phase diagnoses and reports any errors in the operation of INTEL 80186 Direct Memory Access (DMA) channels (byte width).
Tests:	Test 1 — check DMA0 from the SBD to the EPORTS card. Test 2 — check DMA0 from the EPORTS card to the SBD. Test 3 — check DMA1 from the SBD to the EPORTS card. Test 4 — check DMA1 from the EPORTS card to the SBD.
Time:	2 seconds
Warnings:	None
Notes:	The Direct Memory Access Channel 1 (DMA1) is unused and therefore untested.

Phase #12 Tests

Test Number:	1
Function:	This test verifies that DMA0 from the System Board (SBD) to the EPORTS card is functional.
Procedure:	Perform Programmed Input/Output (PIO) to the SBD, DMA that data to the EPORTS card, and compare the value written with the value that was read.
Hardware Tested:	The EPORTS DMA0 is tested.
Data Returned:	Supplemental Data: The value that was written. Raw Data: The value that was read.
Notes:	None

==========

Test Number:	2
Function:	This test verifies that DMA0 from the EPORTS card to the SBD is functional.
Procedure:	Write EPORTS RAM, DMA that data to the SBD, and compare the value written with the value that was read.
Hardware Tested:	The EPORTS DMA0 is tested.
Data Returned:	Supplemental Data: The value that was written. Raw Data: The value that was read.
Notes:	None

Test Number:	3
Function:	This test verifies that DMA1 from the SBD to the EPORTS card is functional.
Procedure:	Perform PIO to the SBD, DMA that data to the I/O board, and compare the value written with the value that was read.
Hardware Tested:	The EPORTS DMA1 is tested.
Data Returned:	Supplemental Data: The value that was written. Raw Data: The value that was read.
Notes:	None

==========

Test Number:	4
Function:	This test verifies that DMA1 from the EPORTS card to the SBD is functional.
Procedure:	Write EPORTS RAM, DMA that data to the SBD, and compare the value written with the value that was read.
Hardware Tested:	The EPORTS DMA1 is tested.
Data Returned:	Supplemental Data: The value that was written. Raw Data: The value that was read.
Notes:	None

Phase #13 — EPORTS — DMA Word Transfers

Phase Name:	EPORTS — DMA Word Transfers (dmawrd)
Type:	Demand
Function:	This phase diagnoses and reports any errors in the word transfer operation of the INTEL 80186 Direct Memory Access (DMA) channels (word width).
Tests:	Test 1 — check DMA0 from the SBD to the EPORTS card. Test 2 — check DMA0 from the EPORTS card to the SBD. Test 3 — check DMA1 from the SBD to the EPORTS card. Test 4 — check DMA1 from the EPORTS card to the SBD.
Notes:	The Direct Memory Access Channel 1 (DMA1) is unused and therefore untested.
Time:	2 seconds
Warnings:	None
Notes:	None

Phase #13 Tests

Test Number:	1
Function:	This test verifies that DMA0 word transfers from the System Board (SBD) to the EPORTS card is functional.
Procedure::	Perform Programmed Input/Output (PIO) to the SBD, DMA that data to the EPORTS card, and compare the value written with the value that was read.
Hardware Tested:	The EPORTS DMA0 is tested.
Data Returned:	Supplemental Data: The value that was written. Raw Data: The value that was read.
Notes:	None

==========

Test Number	2
Function:	This test verifies that DMA0 from the EPORTS card to the SBD is functional.
Procedure:	Write EPORTS RAM, DMA that data to the SBD, and compare the value written with the value that was read.
Hardware Tested:	The EPORTS DMA0 is tested.
Data Returned:	Supplemental Data: The value that was written. Raw Data: The value that was read.
Notes:	None

Test Number:	3
Function:	This test verifies that DMA1 from the SBD to the EPORTS card is functional.
Procedure:	Perform PIO to the SBD, DMA that data to the I/O board, and compare the value written with the value that was read.
Hardware Tested:	The EPORTS DMA1 is tested.
Data Returned:	Supplemental Data: The value that was written. Raw Data: The value that was read.
Notes:	None

==========

Test Number:	4
Function:	This test verifies that DMA1 from the EPORTS card to the SBD is functional.
Procedure:	Write EPORTS RAM, DMA that data to the SBD, and compare the value written with the value that was read.
Hardware Tested:	The EPORTS DMA1 is tested.
Data Returned:	Supplemental Data: The value that was written. Raw Data: The value that was read.
Notes:	None

Phase #14 — EPORTS — SCC Basic Sanity

Phase Name:	EPORTS — SCC Basic Sanity
Type:	Demand
Function:	This phase tests the basic sanity of each Serial Communication Controller (SCC) by looping a character in local loop mode on each channel of the SCC.
Test:	Test 1 — local loops a character at 9600 baud with 2 Stop bits and no parity for each channel. Channels A and B of SCC0 through SCC3 are tested.
Time:	12 seconds
Warnings:	None
Notes:	None

Phase #15 — EPORTS — DTR and Basic Interrupt Integrity

Phase Name:	EPORTS — DTR and Basic Interrupt Integrity (dtr)
Type:	Demand
Function:	This phase diagnoses the Data Terminal Ready (DTR) register by writing and reading all bit combinations. The phase then tests basic integrity of the Serial Communication Controller (SCC) interrupt and acknowledge circuitry.
Tests:	Tests 1 through 256 — write a number equal to the test number minus 1 into the DTR register and read it back. Test 257 — causes a transmit buffer empty interrupt on the SCC furthest electrically from the INTEL 80186 Microprocessor, and checks to see that the proper interrupt was taken. This will verify the basic sanity of the interrupt and acknowledge circuitry to SCCs.
Time:	1 second
Warnings:	None
Notes:	None

Phase #16 — EPORTS — SCC Receive Buffers

Phase Name:	EPORTS — SCC Receive Buffers (rx_buf)
Type:	Demand
Function:	This phase diagnoses the Receive buffer of the Serial Communication Controller (SCC) by seeing if it can be filled without losing characters. Also, checks if the SCC generates the proper interrupt if the Receive buffer overflows.
Tests:	Tests 1 through 4 — transmit three characters in local loop mode before reading them back. Tests 5 through 8 — transmit four characters in local loop mode to force a buffer overflow interrupt.
Time:	5 seconds
Warnings:	None
Notes:	None

Phase #17 — EPORTS — Basic DMAC and SCC Test

Phase Name:	EPORTS — Basic DMAC and SCC Test (dmac)
Type:	Demand
Function:	This phase uses the Serial Communication Controllers (SCCs) in local loop around mode to see if each Direct Memory Access (DMA) channel can transfer a character properly between SCC and Dynamic Random Access Memory (DRAM).
Tests:	Test 1 — check transfer of character from SCC0 Channel A, Direct Memory Access Controller 0 (DMAC0) Channels 0 and 1. Test 2 — check transfer of character from SCC0 Channel B, DMAC0 Channels 2 and 3. Test 3 — check transfer of character from SCC1 Channel A, DMAC1 Channels 0 and 1. Test 4 — check transfer of character from SCC1 Channel B, DMAC1 Channels 2 and 3. Test 5 — check transfer of character from SCC2 Channel A, DMAC2 Channels 0 and 1. Test 6 — check transfer of character from SCC2 Channel B, DMAC2 Channels 2 and 3. Test 7 — check transfer of character from SCC3 Channel A, DMAC3 Channels 0 and 1. Test 8 — check transfer of character from SCC3 Channel B, DMAC3 Channels 2 and 3.
Time:	15 seconds
Warnings:	None
Notes:	None

Phase #18 — EPORTS — Local SCC Interrupts

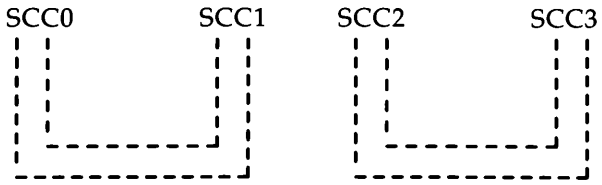
Phase Name:	EPORTS — SCC Local Interrupts (int1)
Type:	Demand
Function:	This phase diagnoses the receive buffer break interrupts. These can be tested without an external loop around connector. For Serial Communication Controller (SCC) 0 through 3, send a break character and check that the proper interrupt occurred. Test for each channel on each SCC.
Tests:	Test 1 — check SCC0 Channel A. Test 2 — check SCC0 Channel B. Test 3 — check SCC1 Channel A. Test 4 — check SCC1 Channel B. Test 5 — check SCC2 Channel A. Test 6 — check SCC2 Channel B. Test 7 — check SCC3 Channel A. Test 8 — check SCC3 Channel B.
Time:	4 seconds
Warnings:	None
Notes:	This test will not stop on first failure but will continue until all ports have been tested. If a port(s) fails, a failure message is displayed and the other ports are then tested.

Phase #19 — EPORTS — External SCC Interrupts

Phase Name: EPORTS — External SCC Interrupts (int2)

Type: Interactive

Function: This phase checks the external looping cable from one port to another on the EPORTS board.



All eight port channels are tested for the following interrupts in this phase:

parity error
framing error
CTS changed
DCD dropped

Tests: Tests 1 through 4 — check transmission of interrupts from SCC0 Channel A (Port 0) to SCC1 Channel B (Port 3):

Test 1 — parity error.
Test 2 — framing error.
Test 3 — CTS change.
Test 4 — DCD dropped.

Tests 5 through 8 — check transmission of interrupts from SCC1 Channel B (Port 3) to SCC0 Channel A (Port 0):

Test 5 — parity error.
Test 6 — framing error.
Test 7 — CTS change.
Test 8 — DCD dropped.

Tests 9 through 12 — check transmission of interrupts from SCC1 Channel A (Port 2) to SCC0 Channel B (Port 1):

Test 9 — parity error.
Test 10 — framing error.
Test 11 — CTS change.
Test 12 — DCD dropped.

Tests 13 through 16 — check transmission of interrupts from SCC0 Channel B (Port 1) to SCC1 Channel A (Port 2):

Test 13 — parity error.
Test 14 — framing error.
Test 15 — CTS change.
Test 16 — DCD dropped.

Tests 17 through 20 — check transmission of interrupts from SCC2 Channel A (Port 4) to SCC3 Channel B (Port 7):

Test 17 — parity error.
Test 18 — framing error.
Test 19 — CTS change.
Test 20 — DCD dropped.

Tests 21 through 24 — check transmission of interrupts from SCC3 Channel B (Port 7) to SCC2 Channel A (Port 4):

Test 21 — parity error.
Test 22 — framing error.
Test 23 — CTS change.
Test 24 — DCD dropped.

Tests 25 through 28 — check transmission of interrupts from SCC3 Channel A (Port 6) to SCC2 Channel B (Port 5):

Test 25 — parity error.
Test 26 — framing error.
Test 27 — CTS change.
Test 28 — DCD dropped.

Tests 29 through 32 — check transmission of interrupts from SCC2 Channel B (Port 5) to SCC3 Channel A (Port 6):

Test 29 — parity error.
Test 30 — framing error.
Test 31 — CTS change.
Test 32 — DCD dropped.

Tests 33 through 288 — verify that data can be looped from the DTR register to the DSR register.

Time: 5 seconds

Warnings: None

Notes: This phase will not stop on first failure but will continue until all ports have been tested. If a port(s) fails, a failure message is displayed and the other ports are then tested.

Phase #19 Tests

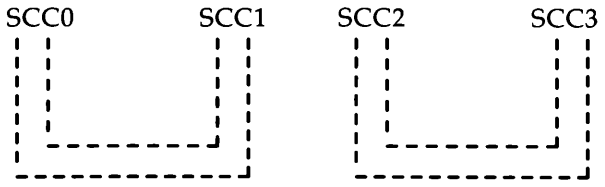
Test Numbers:	33 through 288
Function:	These tests verify that values can be written into the Data Terminal Ready (DTR) register, then looped to and read from the DSR register.
Procedure:	All valid patterns (0x0 through 0xff) are written into the DTR register, then looped to the DSR register. The original value is then compared to the value in the DSR register.
Hardware	The DTR and DSR registers are tested.
Data Returned:	The expected and actual values in the DSR register are returned.
Notes:	Tests 33 through 288 will only be run on EPORTS boards equipped with a DSR register. These tests check all combinations (0x0 - 0xff) of values that can be written in the DTR register and looped to and read from the DSR register.

Phase #20 — EPORTS — External Drivers and Receivers

Phase Name: EPORTS — External Drivers and Receivers (xmit_recv)

Type: Interactive

Function: This phase provides external looping cable from one port to another on the EPORTS board.



The transmit-receive drivers test will send the entire American Standard Code for Information Interchange (ASCII) character set for each Serial Communication Controller (SCC) at the maximum and minimum supported baud rates (38.4 kilobyte, 50).

Tests: Test 1 — SCC0 Channel A transmit - SCC1 Channel B receive.
SCC0 Channel A receive - SCC1 Channel B transmit.

Test 2 — SCC1 Channel A transmit - SCC0 Channel B receive.
SCC1 Channel A receive - SCC0 Channel B transmit.

Test 3 — SCC2 Channel A transmit - SCC3 Channel B receive.
SCC2 Channel A receive - SCC3 Channel B transmit.

Test 4 — SCC3 Channel A transmit - SCC2 Channel B receive.
SCC3 Channel A receive - SCC2 Channel B transmit.

Time: 300 seconds

Warnings: None

Notes: This test will not stop on first failure but will continue until all ports are tested. If a port(s) fails, a failure message is displayed and the other ports are then tested.

Data Returned: The expected and actual data values are returned.

Phase #21 — EPORTS — Complete DMAC and SCC Test

Phase Name: EPORTS — Complete DMAC and SCC Test (scc_dmac)

Type: Demand

Function: This phase tests the Serial Communication Controller (SCC) and Dynamic Memory Access Controller (DMAC) together with all possible baud rates, character bit sizes, parity, Stop bits, various address locations in Dynamic Random Access Memory (DRAM) and various sizes of data blocks to be transferred. The tests are done in local loop mode so they can run without special loop around cables connected to the EPORTS board.

Tests: Test 1 — check SCC0 and DMAC0--Port 0.

Test 2 — check SCC0 and DMAC0--Port 1.

Test 3 — check SCC1 and DMAC1--Port 2.

Test 4 — check SCC1 and DMAC1--Port 3.

Test 5 — check SCC2 and DMAC2--Port 4.

Test 6 — check SCC2 and DMAC2--Port 5.

Test 7 — check SCC3 and DMAC3--Port 6.

Test 8 — check SCC3 and DMAC3--Port 7.

Time: 900 seconds

Warnings: None

Notes: This test will not stop on first failure but will continue until all ports are tested. If a port(s) fails, a failure message is displayed and the other ports are then tested.

Data Returned: The parameters of the failed test (baud rate, Stop bit, character size, block size, source address, and destination address), type of failure, expected data, and actual data are returned.

All addresses are physical addresses. The baud rate, parity, Stop bit, and character size parameters are returned in a numeric code which must be translated with the following tables:

Returned Baud Rate	Baud Rate Setting
bd=0	50
bd=1	75
bd=2	110
bd=3	134.5
bd=4	150
bd=5	200
bd=6	300
bd=7	600
bd=8	1200
bd=9	1800

Returned Baud Rate	Baud Rate Setting
bd=10	2400
bd=11	4800
bd=12	9600
bd=13	19.2K
bd=14	38.4K

Returned Parity	Parity Setting
par=0	None
par=1	Odd
par=2	Even

Returned Stop Bit Value	Stop Bit Setting
stop=0	2 Stop bits
stop=1	1.5 Stop bits
stop=2	1 Stop bit

Returned Character Size Value	Character Size Setting
ch sz=0	8 bits
ch sz=1	7 bits
ch sz=2	6 bits
ch sz=3	5 bits

Chapter 10: Remote Management Diagnostics

Introduction

This chapter contains a description of the diagnostic phases for the 3B2 computer Remote Management card. The Remote Management card is referred to as the Alarm Interface Card (AIC) in software. If your computer is equipped with a Remote Management card, the Equipped Device Table (EDT) should list it. You can use the diagnostic monitor command "show" (DGMON> **s**) to display the EDT. If the Remote Management card is not listed in the EDT and a VOID or NULL is listed, the Remote Management card may be faulty.

Three diagnostic phases run tests on all major Remote Management Card components. The Table of Contents listing on the previous page will help you locate the descriptions for each diagnostic phase and its associated tests. The phase descriptions are organized numerically in the same order that the phases are run on the 3B2 computer.

Phase Descriptions

Phase #1 — AIC Control and Status Register

Phase Name:	AIC Control and Status Register (aic_csr)
Type:	Interactive
Function:	This phase ensures that every bit can be set and cleared by software.
Tests:	Tests 1 and 2 — confirm that the Inhibit Alarms bit can be set and cleared. Tests 3 and 4 — confirm that the Software Minor Alarm bit can be set and cleared. Tests 5 and 6 — confirm that the Software Major Alarm bit can be set and cleared. Tests 7 and 8 — confirm that the Enable Timers bit can be set and cleared. Tests 9 through 11 — confirm that the Clear Timers bit can be set (active low) and that it is automatically reset by hardware 130 milliseconds later (logic level of high). Tests 12 and 13 — confirm that the AC Fail Alarm bit can be set and cleared. Tests 14 and 15 — confirm that the Low Battery Alarm bit can be set and cleared. Tests 16 and 17 — confirm that the TTL Input 2 bit can be set and cleared. Tests 18 and 19 — confirm that the TTL Input 1 bit can be set and cleared. Tests 20 and 21 — confirm that the Insane Alarm bit can be set and cleared. Tests 22 through 24 — confirm that the Time-out bit can be set and cleared. Tests 25 and 26 — confirm that the DTR bit can be set and cleared.
Time:	120 seconds
Warnings:	This phase requires the use of a special loop around test connector. The Inhibit Alarms bit remains inhibited to prevent any adverse side effects until the AIC driver initializes the board.
Notes:	To set a bit in the AIC Control and Status Register (CSR), you have to perform a read of the address assigned for the particular bit. A write to the same location does not have an effect. Also note that because the Remote Management card does not provide an interrupt vector, an exception is generated [the Central Processing Unit (CPU) is faulted]. Therefore, the exception handler is used to detect an interrupt from the AIC board.

Phase #1 Tests

Test Numbers: 1 and 2

Function: These tests confirm that the Inhibit Alarms bit of the CSR can be set and cleared.

Procedure: Determine the address of the board under test. Then set the 'inhib_alrms' bit of the CSR. Read the CSR confirming that it is set. Clear the bit by reading the 'enb_alrms' bit of the CSR. Read the CSR confirming that the bit is cleared.

Hardware Tested: The AIC CSR is tested.

Data Returned: The failing test number (last number displayed on the system console prior to failure) is returned.

Notes: The Inhibit Alarms bit is set at the end of this test to enable the testing of the other CSR bits without side effects.

==========

Test Numbers: 3 and 4

Function: These tests confirm that the Software Minor (SWMINOR) Alarm bit of the CSR can be set and cleared.

Procedure: Set the SWMINOR Alarm bit of the CSR. Read the CSR confirming that it is set. Clear the bit by reading the 'clr_swminor' bit of the CSR. Read the CSR confirming that the bit is cleared.

Hardware Tested: The AIC CSR is tested.

Data Returned: The failing test number (last number displayed on the system console prior to failure) is returned.

Notes: None

==========

Test Numbers: 5 and 6

Function: These tests confirm that the Software Major (SWMAJOR) Alarm bit of the CSR can be set and cleared.

Procedure: Set the SWMAJOR Alarm bit of the CSR. Read the CSR confirming that it is set. Clear the bit by reading the 'clr_swmajor' bit of the CSR. Read the CSR confirming that the bit is cleared.

Hardware Tested: The AIC CSR is tested.

Data Returned: The failing test number (last number displayed on the system console prior to failure) is returned.

Notes: None

Test Numbers:	7 and 8
Function:	These tests confirm that the Enable Timers (SANENBL) bit of the CSR can be set and cleared.
Procedure:	Set the SANENBL bit of the CSR. Read the CSR confirming that it is set. Clear the bit by reading the 'dis_timer' bit of the CSR. Read the CSR confirming that the bit is cleared.
Hardware Tested:	The AIC CSR is tested.
Data Returned:	The failing test number (last number displayed on the system console prior to failure) is returned.
Notes:	None

==========

Test Numbers:	9 through 11
Function:	These tests confirm that the Clear Timers (CLRTMRS0) bit can be set (active low) and that it is automatically reset by hardware 130 milliseconds later.
Procedure:	Set (active low) the CLRTMRS0 bit of the CSR. Read the CSR confirming that it is set (active low). Delay 140 milliseconds. Read the CSR confirming that the bit is reset (high).
Hardware Tested:	The AIC CSR is tested.
Data Returned:	The failing test number (last number displayed on the system console prior to failure) is returned.
Notes:	None

==========

Test Numbers:	12 and 13
Function:	These tests confirm that the AC Fail Alarm bit of the CSR can be set and cleared.
Procedure:	Set the AC Fail Alarm bit of the CSR. Read the CSR confirming that it is set. Clear the bit by reading the 'clr_acfail' bit of the CSR. Read the CSR confirming that the bit is cleared.
Hardware Tested:	The AIC CSR is tested.
Data Returned:	The failing test number (last number displayed on the system console prior to failure) is returned.
Notes:	None

Test Numbers: 14 and 15

Function: These tests confirm that the Low Battery (LOWBAT) Alarm bit of the CSR can be set and cleared.

Procedure: Set the LOWBAT Alarm bit of the CSR. Read the CSR confirming that it is set. Clear the bit by reading the 'clr_lowbat' bit of the CSR. Read the CSR confirming that the bit is cleared.

Hardware Tested: The AIC CSR is tested.

Data Returned: The failing test number (last number displayed on the system console prior to failure) is returned.

Notes: None

==========

Test Numbers: 16 and 17

Function: These tests confirm that the TTL Input 2 (EXTTL2) bit of the CSR can be set and cleared.

Procedure: Set the EXTTL2 bit of the CSR. Read the CSR confirming that it is set. Clear the bit by reading the 'clr_ttl21' bit of the CSR. Read the CSR confirming that the bit is cleared.

Hardware Tested: The AIC CSR is tested.

Data Returned: The failing test number (last number displayed on the system console prior to failure) is returned.

Notes: None

==========

Test Numbers: 18 and 19

Function: These tests confirm that the TTL Input 1 (EXTTL1) bit of the CSR can be set and cleared.

Procedure: Set the EXTTL1 bit of the CSR. Read the CSR confirming that it is set. Clear the bit by reading the 'clr_ttl11' bit of the CSR. Read the CSR confirming that the bit is cleared.

Hardware Tested: The AIC CSR is tested.

Data Returned: The failing test number (last number displayed on the system console prior to failure) is returned.

Notes: None

Test Numbers:	20 and 21
Function:	These tests confirm that the Insane Alarm bit of the CSR can be set and cleared.
Procedure:	Set the Insane Alarm bit of the CSR. Read the CSR confirming that it is set. Clear the bit by reading the 'clr_insane' bit of the CSR. Read the CSR confirming that the bit is cleared.
Hardware Tested:	The AIC CSR is tested.
Data Returned:	The failing test number (last number displayed on the system console prior to failure) is returned.
Notes:	None

==========

Test Numbers:	22 through 24
Function:	These tests confirm that the Time-out 1 (TMOUT1) bit of the CSR can be set and cleared.
Procedure:	Delay for interrupt. Test 22 — confirms interrupt is taken. Test 23 — reads the CSR confirming that it is set, and clears the bit by reading the 'clr_int20' bit of the CSR. Test 24 — reads the CSR confirming that the bit is cleared.
Hardware Tested:	The AIC CSR is tested.
Data Returned:	The failing test number (last number displayed on the system console prior to failure) is returned.
Notes:	Setting the TMOUT1 bit generates an interrupt which is detected and confirmed in this phase. The interrupt handler issues the Clear Pending PINT20 bit command.

==========

Test Numbers:	25 and 26
Function:	These tests confirm that the Data Terminal Ready (DTR) bit of the CSR can be set and cleared.
Procedure:	Set the DTR bit of the CSR. Read the CSR and confirm the setting. Clear the DTR bit in the CSR. Read the CSR and confirm the setting.
Hardware Tested:	The AIC CSR is tested.
Data Returned:	The failing test number (last number displayed on the system console prior to failure) is returned.
Notes:	None

Phase #2 — AIC Alarm Generation

Phase Name: AIC Alarm Generation (aic_alrms)

Type: Interactive

Function: This phase tests that the alarm generation circuitry is operational.

Tests: Test 1 — confirms that setting the SWMAJOR bit in CSR generates an alarm to the autodialer.

Test 2 — confirms that setting the SWMINOR bit in CSR generates an alarm to the autodialer.

Test 3 — confirms that the Sanity timer does not time-out when the timer is disabled.

Test 4 — confirms that the first sanity time-out does not generate an alarm to the autodialer.

Time: 180 seconds

Warnings: The special connector used in this test phase can cause other AIC phases to fail! Remove it after this test phase is executed. When this test terminates: the system will be reset, the UNIX operating system will shut down, and the AIC status will be lost.

Notes: All tests require the use of a special loop around connector which connects the Software Major (SWMAJOR) Output bit to the EXTTL1 and EXTTL2 Input bits, and the Software Minor (SWMINOR) Output bit to the ACFAIL and Low Battery (LOWBAT) Input bits. Further note that the testing of the interrupt and alarm generation when the ACFAIL and LOWBAT bits are set is default tested whenever a minor alarm is generated. Also note that because the AIC does not provide an interrupt vector, the Central Processing Unit (CPU) is faulted. Therefore, the exception handler is used to detect correct interrupt processing.

Phase #2 Tests

Test Number: 1

Function: This test confirms that setting the SWMAJOR bit in the Control and Status Register (CSR) generates a major alarm to the autodialer.

Procedure: With the special connector in place, disable the timers, set the SWMAJOR bit, and confirm that a major alarm is output by detection of the one TTL1 input signal and ACFAIL/LOWBAT bit being set to a logic level '1' in the CSR.

Hardware Tested: The AIC alarm generation circuitry is tested.

Data Returned: The failing test number (last number displayed on the system console prior to failure) is returned.

Notes: None

Test Number:	2
Function:	This test confirms that setting the SWMINOR bit in the CSR generates a minor alarm to the autodialer.
Procedure:	With the alarms enabled and the special connector in place, set the SWMINOR Alarm bit in the CSR. Confirm that a minor alarm is output by the AIC by detecting that the ACFAIL and LOWBAT input signals were driven to a logic level '1' in CSR.
Hardware Tested:	The AIC alarm generation circuitry is tested.
Data Returned:	The failing test number (last number displayed on the system console prior to failure) is returned.
Notes:	When ACFAIL and LOWBAT inputs are driven high, a PINT20 should occur.

==========

Test Numbers:	3 and 4
Function:	These tests confirm that the first sanity time-out does not generate an alarm to the autodialer.
Procedure:	With the alarms enabled and the special connector in place, clear the timer and Timer counter, wait for clearance to take place. Test 3 — delays for 65 seconds and observes that the timer does not time-out with the timer disabled. Test 4 — enables the timer and delays for 65 seconds, allowing the first sanity time-out to occur. Confirm that an alarm is not generated by reading the CSR bits EXTTL1, EXTTL2, ACFAIL, and LOWBAT. All of these bits are at a low logic level.
Hardware Tested:	The AIC alarm generation circuitry is tested.
Data Returned:	The failing test number (last number displayed on the system console prior to failure) is returned.
Notes:	None

Phase #3 — AIC Timer Execution

Phase Name:	AIC Timer Execution (aic_timers)
Type:	Normal
Function:	This phase tests the Sanity timer functions as specified.
Tests:	Test 1 — confirms that the first occurrence of a sanity time-out generates an interrupt by asserting the I/O bus PINT20 signal. Test 2 — confirms that the timers are cleared by the setting of the CLRTMRS0 bit of the AIC CSR and observing that only an interrupt is generated after allowing the sanity time-out to occur. Test 3 — confirms that a system reset request is generated when a second sanity time-out occurs.
Time:	1 second
Warning:	The last test phase causes a system reset.
Notes:	None

Phase #3 Tests

Test Number:	1
Function:	This test confirms that the first sanity time-out generates an interrupt and sets the Time-out 1 (TMOUT1) bit in the Control and Status Register (CSR).
Procedure:	Clear the timers; enable Sanity timer and wait 140 milliseconds. Confirm, by the interrupt handler, that an interrupt is generated and that the TMOUT1 bit in the CSR is set.
Hardware Tested:	The AIC Sanity timer is tested.
Data Returned:	The failing test number (last number displayed on the system console prior to failure) is returned.
Notes:	None

Test Number:	2
Function:	This test confirms that the Clear Timers 0 (CLRTMRS0) bit of the CSR does indeed clear the timers and time-out counter.
Procedure:	Clear the timers; enable Sanity timer and wait 65 seconds. Confirm, by the interrupt handler, that an interrupt is generated and that the TMOUT1 bit in the CSR is set.
Hardware Tested:	The AIC Sanity timer and CSR are tested.
Data Returned:	The failing test number (last number displayed on the system console prior to failure) is returned.
Notes:	This test phase, should it fail, may reset the system. If a reset occurs, this test failed.

==========

Test Number:	3
Function:	This test confirms that a second sanity time-out produces a system reset.
Procedure:	Delay for another 65 seconds allowing the Sanity timer to time-out a second time.
Hardware Tested:	The AIC Sanity timer and CSR are tested.
Data Returned:	The failing test number (last number displayed on the system console prior to failure) is returned.
Notes:	This test phase terminates with a system reset. A successful completion message is not issued inasmuch as there is no way to trap a reset exception back to this diagnostic test phase. Therefore, if the system resets itself, this test phase completed successfully.

Chapter 11: Expansion Disk Controller Diagnostics

Introduction

This chapter contains a description of the diagnostic phases and tests for the Expansion Disk Controller (XDC) card. Diagnostics are provided to verify the sanity of the Common I/O (CIO) interface between the 3B2 computer Input/Output (I/O) bus and the XDC, the XDC itself, the interface between the XDC and hard disk(s), and the hard disk(s).

All of the XDC diagnostic tests are executed by the XDC INTEL 80186 Central Processing Unit (CPU).

If your 3B2 computer is equipped with an XDC card, the Equipped Device Table (EDT) should list it. You can use the diagnostic monitor command "show" (DGMON > **s**) to print a copy of the EDT. If the XDC card is not listed in the EDT and a VOID or NULL is listed, one or more of the following hardware devices on the XDC card may be faulty:

- INTEL 80186 Microprocessor
- XDC Read Only Memory (ROM)/Random Access Memory (RAM)
- XDC Hard Disk Controller (HDC)
- XDC Address/Data bus
- XDC interface to the 3B2 computer I/O bus.

There are 22 phases associated with XDC diagnostics. The first phase is a normal phase, and the rest are demand phases. Each phase is designed to test a subunit or portion of a subunit of the XDC. Normal phases are executed as part of the normal boot sequence. Demand phases, as the name implies, are executed only when requested. Demand phases perform more exhaustive tests on the XDC.

The common tests are the first set of tests executed in the XDC diagnostics. Basic sanity of the INTEL 80186 Microprocessor is being checked.

ROM-based checkerboard (all 5's) and inverted checkerboard (all A's) test patterns are written to and then read from each of the following registers:

- Data Registers—AX, BX, CX, and DX (16-bit registers)
- Pointer and Index Registers—Stack Pointer (SP), Base Pointer (BP), Source Index (SI), and Destination Index (DI)
- Segment Registers—Code (CS), Data (DS), Stack (SS), and Extra (ES).

Application-specific tests perform basic sanity checks of the XDC Dynamic Random Access Memory (DRAM). The purpose of this test is to establish minimum sanity in DRAM. Data is written to and read from a small range (10 locations) of memory locations.

Twenty-two diagnostic phases run tests on all major XDC card components. The Table of Contents listing on the previous page will help you locate the descriptions for each diagnostic phase and its associated tests. The phase descriptions are organized numerically in the same order that the phases are run on the 3B2 computer.

Phase Descriptions

Phase #1 — XDC Common I/O Sanity Test

Phase Name:	XDC I/O Sanity Test (cio.c)
Type:	Normal
Function:	This phase verifies that the XDC card Common I/O (CIO) interface is functioning properly.
Test:	Test 1 — verifies that the diagnostic results can be transferred to the System Board (SBD).
Time:	1 second
Warnings:	None
Notes:	None

Phase #1 Test

Test Number:	1
Function:	This test verifies that the CIO hardware and firmware are functioning properly.
Procedure:	This test uses the following standard procedure:

1. The I/O slot number of the XDC card in the 3B2 computer is determined.
2. The XDC card is reset.
3. The sysgen data block is initialized.
4. The XDC card is initialized (sysgen) by sending the express [Interrupt 0 (INT0)] and attention (INT1) interrupts sequentially.
5. The X86 code is downloaded by using the CIO firmware command [Download Memory (DLM)].
6. Execution of the phase is started by using the CIO firmware command [Force Call to Function (FCF)].
7. A function call to **"phasend( )"** is made when the phase is complete. Phasend returns the test results to the SBD.

Hardware Tested:	The XDC interface to the 3B2 computer I/O bus is tested.
Data Returned:	None

Phase #2 — PCSR Write/Read Test

Phase Name:	Peripheral Control and Status Register Write/Read (pcsr)
Type:	Demand
Function:	This phase diagnoses and reports any bit interdependency of the Peripheral Control and Status Register (PCSR).
Tests:	Test 1 — writes and reads PCSR bit 0. Test 2 — writes and reads PCSR bit 1. Test 3 — writes and reads PCSR bit 2. Test 4 — writes and reads PCSR bit 4.
Time:	1 second
Warnings:	None
Notes:	None

Phase #2 Tests

Test Numbers:	1 through 4
Function:	These tests verify that the PCSR register is functional.
Procedure:	A valid pattern is written to the register and verified with a read.
Hardware Tested:	The XDC PCSR is tested.
Data Returned:	The test number that failed, the actual values, and the expected values are returned.

Phase #3 — Upper RAM Write/Read Test

Phase Name:	XDC Upper RAM
Type:	Demand
Function:	This phase diagnoses and reports any errors in the operation of the XDC upper RAM and refresh circuitry.
Tests:	Tests 1 through 4 — walk data patterns through the upper 64 kilobytes of RAM. Test 5 — checks the upper 64 kilobytes of the upper RAM and refresh circuitry.
Time:	12 seconds
Warnings:	None
Notes:	None

Phase #3 Tests

Test Numbers:	1 through 4
Function:	These tests check the operation of the upper RAM.
Procedure:	The following test sequence is used to test the upper RAM (addresses 0x80000 through 0x9fffe): 1. All memory locations are written with 0's in ascending (small addresses to large addresses) order. 2. All memory locations are read, expecting 0's, and then all memory is written with 1's in descending order. 3. All memory locations are read, expecting 1's, and then all memory is written with 5's in ascending order. 4. All memory locations are read, expecting 5's, and then all memory is written with A's in descending order.
Hardware Tested:	The upper RAM locations are tested.
Data Returned:	The number of the failing test, the actual data, and the expected data are returned.
Notes:	None

Test Number: 5

Function: This test verifies that the upper RAM and refresh circuitry are functioning properly.

Procedure: All memory locations are read, expecting A's, and then all memory is written with the following patterns in ascending order:

pat(00)=0xF606 pat(08)=0xF603 pat(16)=0x7FB8 pat(24)=0x401B
pat(01)=0xE727 pat(09)=0xD732 pat(17)=0x6E99 pat(25)=0x612A
pat(02)=0xD25A pat(10)=0xA245 pat(18)=0x59F4 pat(26)=0x0A5F
pat(03)=0xC37B pat(11)=0x8374 pat(19)=0x48D5 pat(27)=0x2B6E
pat(04)=0xBAEC pat(12)=0x1988 pat(20)=0x3D41 pat(28)=0xBD96
pat(05)=0xABCD pat(13)=0x38B9 pat(21)=0x2C60 pat(29)=0x9CA7
pat(06)=0x91A2 pat(14)=0x5FC0 pat(22)=0x141F pat(30)=0xE5DC
pat(07)=0x8083 pat(15)=0x7EF1 pat(23)=0x053E pat(31)=0xC4ED

The patterns are written according to the following formula:

memory location(i) = pat(i modulo 32)
where i = {0, 1, 2 ... (32K-1)} where K=1024
a memory location = 16 bits

All memory locations will be read in ascending order after a 1-second delay to test the refresh circuitry. The refresh rate is 2.048 milliseconds. The time needed to read or write all 64 kilobytes of memory (16 bits on each read or write) is approximately 20.0 milliseconds (125 nanoseconds per cycle; it takes 5 cycles to do a read or write). A delay of 1 second in addition to the 20.0 milliseconds needed to write all memory is sufficient to test the refresh circuitry. This phase executes from the lower 64 kilobytes of RAM.

Hardware Tested: The upper 64 kilobytes of the RAM and refresh circuitry are tested.

Data Returned: The test number that failed, the expected data, and the actual data are returned. If the refresh circuit is not working, the test code (which resides in RAM) will probably be destroyed, and you will get a phase time-out rather than a normal failure.

Phase #4 — Lower RAM Write/Read Test

Phase Name: XDC Lower RAM

Type: Demand

Function: This phase repeats the test sequence of Phase 3 except it uses the lower RAM address range. This phase executes from the upper 64 kilobytes of RAM. The stack, which is normally located in the lower 64 kilobytes of RAM, is moved to the upper 64 kilobytes of RAM before the tests begin.

Tests: Tests 1 through 4 — walk data patterns through the lower 64 kilobytes of RAM.

Test 5 — checks the lower 64 kilobytes of the RAM and refresh circuitry.

Time: 12 seconds

Warnings: None

Notes: If the refresh circuit is not working, the test code (which resides in RAM) will probably be destroyed, and you will get a phase time-out message rather than a normal failure.

Phase #4 Tests

Test Numbers: 1 through 4

Function: These tests check the operation of the lower RAM.

Procedure: The following test sequence is used to test the lower RAM (addresses 0x80000 through 0x9fffe):

1. All memory locations are written with 0's in ascending (small addresses to large addresses) order.
2. All memory locations are read, expecting 0's, and then all memory is written with 1's in descending order.
3. All memory locations are read, expecting 1's, and then all memory is written with 5's in ascending order.
4. All memory locations are read, expecting 5's, and then all memory is written with A's in descending order.

Hardware Tested: The operation of the lower RAM locations are tested.

Data Returned: The number of the failing test, the actual data, and the expected data are returned.

Notes: None

Test Number: 5

Function: This test verifies that the lower RAM and refresh circuitry are functioning properly.

Procedure: All memory locations are read, expecting A's, and then all memory is written with the following patterns in ascending order:

pat(00)=0xF606 pat(08)=0xF603 pat(16)=0x7FB8 pat(24)=0x401B
pat(01)=0xE727 pat(09)=0xD732 pat(17)=0x6E99 pat(25)=0x612A
pat(02)=0xD25A pat(10)=0xA245 pat(18)=0x59F4 pat(26)=0x0A5F
pat(03)=0xC37B pat(11)=0x8374 pat(19)=0x48D5 pat(27)=0x2B6E
pat(04)=0xBAEC pat(12)=0x1988 pat(20)=0x3D41 pat(28)=0xBD96
pat(05)=0xABCD pat(13)=0x38B9 pat(21)=0x2C60 pat(29)=0x9CA7
pat(06)=0x91A2 pat(14)=0x5FC0 pat(22)=0x141F pat(30)=0xE5DC
pat(07)=0x8083 pat(15)=0x7EF1 pat(23)=0x053E pat(31)=0xC4ED

The patterns are written according to the following formula:

memory location(i) = pat(i modulo 32)
where i = {0, 1, 2 ... (32K-1)} where K=1024
a memory location = 16 bits.

All memory locations will be read in ascending order after a 1-second delay to test the refresh circuitry. The refresh rate is 2.048 milliseconds. The time needed to read or write all 64 kilobytes of memory (16 bits on each read or write) is approximately 20.0 milliseconds (125 nanoseconds per cycle; it takes 5 cycles to do a read or write). A delay of 1 second in addition to the 20.0 milliseconds needed to write all memory is sufficient to test the refresh circuitry. This phase executes from the lower 64 kilobytes of RAM.

Hardware Tested: The lower 64 kilobytes of the RAM and refresh circuitry are tested.

Data Returned: The test number that failed, the expected data, and the actual data are returned. If the refresh circuit is not working, the test code (which resides in RAM) will probably be destroyed, and you will get a phase time-out rather than a normal failure.

Phase #5 — ROM Check Sum Test

Phase Name:	XDC ROM Check Sum Test
Type:	Demand
Function:	This phase tests the integrity of the contents of the XDC ROM.
Test:	Test 1 — performs a check sum on the XDC ROM.
Time:	2 seconds
Warnings:	None
Notes:	None

Phase #5 Test

Test Number:	1
Function:	This test verifies the correct ROM check sum.
Procedure:	Calculating and verifying check sum.
Hardware Tested:	The XDC ROM is tested.
Data Returned:	The expected check sum and the calculated check sum are returned.
Notes:	The contents of the ROM are checked by calculating a check sum value based on the following formula: check sum = (check sum rotated left by one) (plus) (contents of current word). A byte in ROM contains the check sum value of the ROM based on the formula above. The final check sum value is compared with the expected check sum value in ROM for that particular release of the ROM. The check sum value is calculated on a byte boundary.

Phase #6 — CPU Upper Chip Select Test

Phase Name:	Upper Chip Select Register Test (cpu_1)
Type:	Demand
Function:	This phase checks the Upper Memory Chip Select (UMCS), Peripheral Access Chip Select (PACS), Middle Memory Chip Select (MMCS), and Middle Peripheral Chip Select (MPCS) registers of the INTEL 80186 CPU.
Tests:	Tests 1 through 12 — check the UMCS register. Tests 13 through 29 — check the PACS register. Tests 30 through 37 — check the MMCS register. Tests 38 through 48 — check the MPCS register.
Time:	2 seconds
Warnings:	None
Notes:	None

Phase #6 Tests

Test Numbers:	1 through 12
Function:	These tests verify that the UMCS register is functional.
Procedure:	Write a valid data pattern to the register, read the register, and compare the value read with the value that was written.
Hardware Tested:	The UMCS register is tested.
Data Returned:	Supplemental Data: The value written is returned. Raw Data: The value read is returned.
Notes:	The values written to the register are 0x0000, 0xc03a, 0xc039, 0xc038, 0xe038, 0xf038, 0xf838, 0xf38, 0xfe38, 0xff38, 0xffb8, and 0xfff8.

==========

Test Numbers:	13 through 29
Function:	These tests verify that the PACS register is functional.
Procedure:	Write a valid data pattern to the register, read the register, and compare the value read with the value that was written.
Hardware Tested:	The PACS register is tested.
Data Returned:	Supplemental Data: The value written is returned. Raw Data: The value read is returned.
Notes:	The values written to the register are 0x0038, 0x003a, 0x003c, 0x0078, 0x00b8, 0x0138, 0x0238, 0x0438, 0x0838, 0x1038, 0x2038, 0x4038, and 0x8038.

Test Numbers:	30 through 37
Function:	These tests verify that the MMCS register is functional.
Procedure:	Write a valid data pattern to the register, read the register, and compare the value read with the value that was written.
Hardware Tested:	The MMCS register is tested.
Data Returned:	Supplemental Data: The value written is returned. Raw Data: The value read is returned.
Notes:	The values written to the register are 0x0 and 0x9f8.

==========

Test Numbers:	38 through 48
Function:	These tests verify that the MPCS register is functional.
Procedure:	Write a valid data pattern to the register, read the register, and compare the value read with the value that was written.
Hardware Tested:	The MPCS register is tested.
Data Returned:	Supplemental Data: The value written is returned. Raw Data: The value read is returned.
Notes:	The values written to the register are 0x80b8, 0x80b9, 0x80ba, 0x80bc, 0x80f8, 0x81b8, 0x82b8, 0x84b8, 0x88b8, 0x90b8, 0xa0b8, and 0xc0b8.

Phase #7 — CPU DMA Internal Test

Phase Name: DMA Control Register Tests (cpu_2)

Type: Demand

Function: This phase checks the Direct Memory Access 0 (DMA0) and DMA1 Control registers.

Tests: Tests 1 through 143 — check the DMA0 and DMA1 Control registers.

Time: 2 seconds

Warnings: None

Notes: None

Phase #7 Tests

Test Numbers: 1 through 143

Function: These tests verify that the DMA0 and DMA1 Control registers are functional.

Procedure: The registers being tested and the test patterns used are as follows:

Registers	Test Pattern(s)
Control Word	Walks a one in a field of zeros.
Transfer Count	Walks a one in a field of zeros.
Dest. Pointer Upper	Walks a one in a field of zeros.
Dest. Pointer Lower	Walks a one in a field of zeros.
Source Pointer Upper	Walks a one in a field of zeros.
Source Pointer Lower	Walks a one in a field of zeros.

The same tests are performed on both DMA0 and DMA1 Control registers.

Hardware Tested: The DMA0 and DMA1 Control registers are tested.

Data Returned: The number of the test that failed, the actual values, and the expected values are returned.

Notes: None

Phase #8 — CPU Timer Test

Phase Name: CPU Timer Test (cpu_3)

Type: Demand

Function: This phase tests the Timer registers of the INTEL 80186 CPU.

Tests: Tests 1 through 17 — check the Timer0 Count register.

Tests 18 through 34 — check the Timer0 MCA register.

Tests 35 through 51 — check the Timer0 MCB register.

Tests 52 through 59 — check the Timer0 Mode register.

Tests 60 through 77 — check the Timer1 Count register.

Tests 78 through 94 — check the Timer1 MCA register.

Tests 95 through 111 — check the Timer1 MCB register.

Tests 112 through 120 — check the Timer1 Mode register.

Tests 121 through 137 — check the Timer2 Count register.

Tests 138 through 154 — check the Timer2 MCA register.

Tests 155 through 158 — check the Timer2 Mode register.

Time: 2 seconds

Warnings: None

Notes: The registers being tested and the test patterns used are as follows:

Registers	Test Pattern(s)
Mode/Control	Walks a one in a field of zeros.
Max. Count B (Timer0 and Timer1 only)	Walks a one in a field of zeros.
Max. Count A	Walks a one in a field of zeros.
Count	Walks a one in a field of zeros.

Phase #8 Tests

Test Numbers:	1 through 17
Function:	These tests verify that the Timer0 Count register is functional.
Procedure:	A valid data pattern is written to the register, and then the register is read and verified.
Hardware Tested:	The Timer0 Count register is tested.
Data Returned:	The number of the test that failed, the actual data, and the expected data are returned.
Notes:	None

==========

Test Numbers:	18 through 34
Function:	These tests verify that the Timer0 Maximum Count A (MCA) register is functional.
Procedure:	A valid data pattern is written to the register, and then the register is read and verified.
Hardware Tested:	The Timer0 MCA register is tested.
Data Returned:	The number of the test that failed, the actual data, and the expected data are returned.
Notes:	None

==========

Test Numbers:	35 through 51
Function:	These tests verify that the Timer0 Maximum Count B (MCB) register is functional.
Procedure:	A valid data pattern is written to the register, and then the register is read and verified.
Hardware Tested:	The Timer0 MCB register is tested.
Data Returned:	The number of the test that failed, the actual data, and the expected data are returned.
Notes:	None

Test Numbers:	52 through 59
Function:	These tests verify that the Timer0 Mode register is functional.
Procedure:	A valid data pattern is written to the register, and then the register is read and verified.
Hardware Tested:	The Timer0 Mode register is tested.
Data Returned:	The number of the test that failed, the actual data, and the expected data are returned.
Notes:	The values written to the register are 0x0000, 0x0001, 0x0002, 0x0004, 0x0008, 0x0010, 0x2000, 0x4000, and 0x8000.

==========

Test Numbers:	60 through 77
Function:	These tests verify that the Timer1 Count register is functional.
Procedure:	A valid data pattern is written to the register, and then the register is read and verified.
Hardware Tested:	The Timer1 Count register is tested.
Data Returned:	The number of the test that failed, the actual data, and the expected data are returned.
Notes:	None

==========

Test Numbers:	78 through 94
Function:	These tests verify that the Timer1 MCA register is functional.
Procedure:	A valid data pattern is written to the register, and then the register is read and verified.
Hardware Tested:	The Timer1 MCA register is tested.
Data Returned:	The number of the test that failed, the actual data, and the expected data are returned.
Notes:	None

Test Numbers:	95 through 111
Function:	These tests verify that the Timer1 MCB register is functional.
Procedure:	A valid data pattern is written to the register, and then the register is read and verified.
Hardware Tested:	The Timer1 MCB register is tested.
Data Returned:	The number of the test that failed, the actual data, and the expected data are returned.
Notes:	None

==========

Test Numbers:	112 through 120
Function:	These tests verify that the Timer1 Mode register is functional.
Procedure:	A valid data pattern is written to the register, and then the register is read and verified.
Hardware Tested:	The Timer1 Mode register is tested.
Data Returned:	The number of the test that failed, the actual data, and the expected data are returned.
Notes:	The values written to the register are 0x0001, 0x0002, 0x0004, 0x0008, 0x0010, 0x0020, 0x2000, 0x4000, and 0x8000.

==========

Test Numbers:	121 through 137
Function:	These tests verify that the Timer2 Count register is functional.
Procedure:	A valid data pattern is written to the register, and then the register is read and verified.
Hardware Tested:	The Timer2 Count register is tested.
Data Returned:	The number of the test that failed, the actual data, and the expected data are returned.
Notes:	None

Test Numbers:	138 through 154
Function:	These tests verify that the Timer2 MCA register is functional.
Procedure:	A valid data pattern is written to the register, and then the register is read and verified.
Hardware Tested:	The Timer2 MCA register is tested.
Data Returned:	The number of the test that failed, the actual data, and the expected data are returned.
Notes:	None

==========

Test Numbers:	155 through 158
Function:	These tests verify that the Timer2 Mode register is functional.
Procedure:	A valid data pattern is written to the register, and then the register is read and verified.
Hardware Tested:	The Timer2 Mode register is tested.
Data Returned:	The number of the test that failed, the actual data, and the expected data are returned.
Notes:	The values written to the register are 0x0x0, 0x0x0020, 0x0x2000, and 0xc001 - read 0x8021.

Phase #9 — CPU Interrupt Controller

Phase Name: CPU Interrupt Controller Register Tests (cpu_4)

Type: Demand

Function: This phase verifies that the internal registers of the INTEL 80186 CPU Interrupt Controller (IC) are functioning properly.

Tests: Tests 1 through 8 — check the IC In-service register.

Tests 9 through 11 — check the Interrupt Request register.

Tests 12 through 20 — check the Interrupt Mask register.

Tests 21 through 23 — check the Interrupt Priority Mask register.

Tests 24 through 28 — check the Interrupt Status register.

Tests 29 through 40 — check the DMA0 and DMA1 Control registers.

Tests 41 through 56 — check the INT0 and INT1 Control registers.

Tests 57 through 69 — check the INT2 and INT3 Control registers.

Time: 2 seconds

Warnings: None

Notes: The registers being tested and the test patterns used are as follows:

Registers	Test Pattern(s)
Interrupt Controller In-service	Walks a one in a field of zeros.
Interrupt Request	Walks a one in a field of zeros.
Interrupt Mask	Walks a one in a field of zeros.
Interrupt Priority Mask	Walks a one in a field of zeros.
Interrupt Control (INT0 and INT1)	Walks a one in a field of zeros.
Interrupt Control (INT2 and INT3)	Walks a one in a field of zeros.

Phase #9 Tests

Test Numbers: 1 through 8

Function: These tests verify that the IC In-service register is functional.

Procedure: A valid data pattern is written to the register, and then the register is read and verified.

Hardware Tested: The IC In-service register is tested.

Data Returned: The number of the test that failed, the actual data, and the expected data are returned.

Notes: The values written to the register are 0x0000, 0x0001, 0x0004, 0x0008, 0x0010, 0x0020, 0x0040, 0x0080, 0x0100, and 0x0200.

Test Numbers:	9 through 11
Function:	These tests verify that the Interrupt Request register is functional.
Procedure:	A valid data pattern is written to the register, and then the register is read and verified.
Hardware Tested:	The Interrupt Request register is tested.
Data Returned:	The number of the test that failed, the actual data, and the expected data are returned.
Notes:	The values written to the register are 0x0, 0x0004, and 0x0008.

==========

Test Numbers:	12 through 20
Function:	These tests verify that the Interrupt Mask register is functional.
Procedure:	A valid data pattern is written to the register, and then the register is read and verified.
Hardware Tested:	The Interrupt Mask register is tested.
Data Returned:	The number of the test that failed, the actual data, and the expected data are returned.
Notes:	The values written to the register are 0x0, 0x0001, 0x0004, 0x0008, 0x0010, 0x0020, 0x0040, 0x0080, and 0x0100.

==========

Test Numbers:	21 through 23
Function:	These tests verify that the Interrupt Priority Mask register is functional.
Procedure:	A valid data pattern is written to the register, and then the register is read and verified.
Hardware Tested:	The Interrupt Priority Mask register is tested.
Data Returned:	The number of the test that failed, the actual data, and the expected data are returned.
Notes:	The values written to the register are 0x0, 0x0001, 0x0002, and 0x0004.

==========

Test Numbers:	24 through 28
Function:	These tests verify that the Interrupt Status register is functional.
Procedure:	A valid data pattern is written to the register, and then the register is read and verified.
Hardware Tested:	The Interrupt Status register is tested.
Data Returned:	The number of the test that failed, the actual data, and the expected data are returned.
Notes:	The values written to the register are 0x0, 0x0001, 0x0002, 0x0004, and 0x8000. The Interrupt/Status register cannot be tested if run under the diagnostic monitor.

Test Numbers:	29 through 40
Function:	These tests verify that the Direct Memory Access 0 (DMA0) and DMA1 Control registers are functional.
Procedure:	A valid data pattern is written to the register, and then the register is read and verified.
Hardware Tested:	The DMA0 and DMA1 Control registers are tested.
Data Returned:	The number of the test that failed, the actual data, and the expected data are returned.
Notes:	The values written to the registers are 0x0, 0x0001, 0x0002, and 0x0004.

==========

Test Numbers:	41 through 56
Function:	These tests verify that the Interrupt 0 (INT0) and INT1 Control registers are functional.
Procedure:	A valid data pattern is written to the register, and then the register is read and verified.
Hardware Tested:	The INT0 and INT1 Control registers are tested.
Data Returned:	The number of the test that failed, the actual data, and the expected data are returned.
Notes:	The values written to the registers are 0x0, 0x0001, 0x0002, 0x0004, 0x0008, 0x0010, 0x0020, 0x0040, and 0x0080.

==========

Test Numbers:	57 through 69
Function:	These tests verify that the INT2 and INT3 Control registers are functional.
Procedure:	A valid data pattern is written to the register, and then the register is read and verified.
Hardware Tested:	The INT2 and INT3 Control registers are tested.
Data Returned:	The number of the test that failed, the actual data, and the expected data are returned.
Notes:	The values written to the registers are 0x0, 0x0001, 0x0002, 0x0004, 0x0008, and 0x0010.

Phase #10 — CPU Lower Chip Select Test

Phase Name:	Lower Chip Select Register Tests (cpu_5)
Type:	Demand
Function:	This phase tests the Lower Memory Chip Select (LMCS) register of the INTEL 80186 CPU.
Tests:	Tests 1 through 12 — check the LMCS register.
Time:	2 seconds
Warnings:	None
Notes:	None

Phase #10 Tests

Test Numbers:	1 through 12
Function:	These tests verify that the LMCS register is functional.
Procedure:	A valid data pattern is written to the register, and then the register is read and verified.
Hardware Tested:	The LMCS register is tested.
Data Returned:	The number of the test that failed, the actual data, and the expected data are returned.
Notes:	The values written to the LMCS register are 0x03f8, 0x07f8, 0x0ff8, 0x1ff8, and 0x3ff8.

Phase #11 — Programmed Input/Output (PIO) Byte Transfer Test

Phase Name:	PIO Byte Transfer Test (pio_1)
Type:	Demand
Function:	This phase tests the PIO Address/Data bus capability of the XDC to and from the System Board (SBD) in byte mode.
Tests:	Test 1 — walks a one through a field of zeros in every memory location of one page of the SBD DPDRAM. Tests 2 through 8 — no tests run.
Time:	45 seconds
Warnings:	None
Notes:	To test the PIO to the SBD, bytes are written to the SBD memory with the PIO. The data bytes written are then retrieved from the SBD by using Direct Memory Access (DMA). The results are compared with the expected data. To test the PIO from the SBD, bytes are written to the XDC Random Access Memory (RAM) and then transferred to the SBD by DMA. The data is then retrieved from the SBD with the PIO and compared with the expected data.

Phase #11 Test

Test Number:	1
Function:	This test verifies the operation of the I/O bus.
Procedure:	A one is written through a field of zeros in all memory locations of one page of the SBD Dual Port Dynamic Random Access Memory (DPDRAM).
Hardware Tested:	The I/O bus is tested.
Data Returned:	The data returned are the test count that failed, the address of the failure, and the data pattern that failed. The test count is the bit that was written. The address is represented as 0x8! 9????.
Notes:	None

Phase #12 — Programmed Input/Output (PIO) Word Transfer Test

Phase Name:	PIO Word Transfer Test (pio_2)
Type:	Demand
Function:	This phase tests the PIO Address/Data bus capability of the XDC to and from the System Board (SBD) in word mode.
Tests:	Test 1 through 16 — walk a one through a field of zeros in every memory location of one page of the SBD DPDRAM.
Time:	60 seconds
Warnings:	None
Notes:	To test the PIO to the SBD, words are written to the SBD memory with the PIO. The data words written are then retrieved from the SBD by using Direct Memory Access (DMA). The results are compared with the expected data. To test the PIO from the SBD, words are written to the XDC Random Access Memory (RAM) and then transferred to the SBD by DMA. The data is then retrieved from the SBD with the PIO and compared with the expected data.

Phase #12 Tests

Test Numbers:	1 through 16
Function:	These tests verify the operation of the I/O bus.
Procedure:	A one is written through a field of zeros in all memory locations of one page of the SBD Dual Port Dynamic Random Access Memory (DPDRAM).
Hardware Tested:	The I/O bus is tested.
Data Returned:	The data returned are the test count that failed, the address of the failure, and the data pattern that failed. The test count is the bit that was written. The address is represented as 0x8i 9????.
Notes:	None

Phase #13 — DMA Byte Transfer Test

Phase Name:	DMA Byte Transfer Tests (dmabyt)
Type:	Demand
Function:	This phase tests the Direct Memory Access (DMA) capability to and from the System Board (SBD) in byte mode.
Tests:	Tests 1 through 4 — test the operation of both DMA channels.
Time:	2 seconds
Warnings:	None
Notes:	To test DMA from the SBD, bytes are written to the SBD by the Programmed Input/Output (PIO), and then transferred to the XDC using the DMA. To test DMA to the SBD, bytes are written to the XDC Random Access Memory (RAM), and then transferred to the SBD by DMA. The bytes are then retrieved from the SBD by the PIO and compared to the expected values.

Phase #13 Tests

Test Number:	1
Function:	This test verifies that Direct Memory Access Channel 0 (DMA0) from the SBD to the I/O board is functional.
Procedure:	Data is sent to the SBD by the PIO. Then, that data is transferred to the I/O board by DMA. The data is compared for validity.
Hardware Tested:	The INTEL 80186 CPU DMA0 is tested.
Data Returned:	The number of the test that failed, the actual data, and the expected data are returned.
Notes:	None

==========

Test Number:	2
Function:	This test verifies that DMA0 from the I/O board to the SBD is functional.
Procedure:	Data is written to the I/O board RAM. Then, that data is sent to the SBD by DMA. The data is compared for validity.
Hardware Tested:	The INTEL 80186 CPU DMA0 is tested.
Data Returned:	The number of the test that failed, the actual data, and the expected data are returned.
Notes:	None

Test Number:	3
Function:	This test verifies that DMA1 from the SBD to the I/O board is functional.
Procedure:	Data is sent to the SBD by the PIO. Then, that data is sent to the I/O board by DMA. The data is compared for validity.
Hardware Tested:	The INTEL 80186 CPU DMA1 is tested.
Data Returned:	The number of the test that failed, the actual data, and the expected data are returned.
Notes:	None

==========

Test Number:	4
Function:	This test verifies that DMA1 from the I/O board to the SBD is functional.
Procedure:	Data is written to the I/O board RAM. Then, that data is sent to the SBD by DMA. The data is compared for validity.
Hardware Tested:	The INTEL 80186 CPU DMA1 is tested.
Data Returned:	The number of the test that failed, the actual data, and the expected data are returned.
Notes:	None

Phase #14 — DMA Word Transfer Tests

Phase Name: DMA Word Transfer Tests (dmawrd)

Type: Demand

Function: This phase tests the Direct Memory Access (DMA) capability to and from the System Board (SBD) in word mode.

Tests: Tests 1 through 4 — test the operation of both DMA channels.

Time: 2 seconds

Warnings: None

Notes: This phase is designed to test the DMA capability to and from the SBD in word mode. The procedure is the same as Phase 13 except that words are used instead of bytes.

To test DMA from the SBD, words are written to the SBD by the Programmed Input/Output (PIO), and then they are transferred to the XDC using DMA.

To test DMA to the SBD, words are written to the XDC Random Access Memory (RAM), and then they are transferred to the SBD by DMA. The words are then retrieved from the SBD by the PIO, and they are compared against the expected values.

Phase #14 Tests

Test Number: 1

Function: This test verifies that Direct Memory Access Channel 0 (DMA0) from the SBD to the I/O board is functional.

Procedure: Data is sent to the SBD by the PIO. Then, that data is transferred to the I/O board by DMA. The data is compared for validity.

Hardware Tested: The INTEL 80186 CPU DMA0 is tested.

Data Returned: The number of the test that failed, the actual data, and the expected data are returned.

Notes: None

==========

Test Number: 2

Function: This test verifies that DMA0 from the I/O board to the SBD is functional.

Procedure: Data is written to the I/O board RAM. Then, that data is sent to the SBD by DMA. The data is compared for validity.

Hardware Tested: The INTEL 80186 CPU DMA0 is tested.

Data Returned: The number of the test that failed, the actual data, and the expected data are returned.

Notes: None

Test Number:	3
Function:	This test verifies that DMA1 from the SBD to the I/O board is functional.
Procedure:	Data is sent to the SBD by the PIO. Then, that data is sent to the I/O board by DMA. The data is compared for validity.
Hardware Tested:	The INTEL 80186 CPU DMA1 is tested.
Data Returned:	The number of the test that failed, the actual data, and the expected data are returned.
Notes:	None

==========

Test Number:	4
Function:	This test verifies that DMA1 from the I/O board to the SBD is functional.
Procedure:	Data is written to the I/O board RAM. Then, that data is sent to the SBD by DMA. The data is compared for validity.
Hardware Tested:	The INTEL 80186 CPU DMA1 is tested.
Data Returned:	The number of the test that failed, the actual data, and the expected data are returned.
Notes:	None

Phase #15 — Hard Disk Controller (HDC) Basic Sanity Test

Phase Name: XDC FIFO and Specify Command Test (xdc_1)

Type: Demand

Function: This phase checks the XDC First-In-First-Out (FIFO) and the specify initialization command.

Tests: Tests 1 through 8 — test the I/O of the FIFO with eight patterns.

Test 9 — initializes the XDC with the specify command.

Time: 2 seconds

Warnings: None

Notes: This phase is designed to test the basic sanity of the XDC. The tests are designed to verify the sanity of the FIFO and the specify command. The FIFO is filled with eight patterns and the data written are then retrieved for comparison. The data patterns used are as follows:

Pattern	Data
1	0x0f
2	0xf0
3	0xcc
4	0x33
5	0xaa
6	0x55
7	0xaa
8	0x55

These patterns are designed to test bit-wise interdependence and transitional fault (data propagation from 1 byte of the FIFO to the next). If the FIFO passes all of the tests, the specify command is used to initialize the XDC to the proper mode for further tests. The mode of the XDC is initialized to the following:

Mode:	
MDU = 0	(MFM recording)
ECC = 0	(CRC is appended to data field)
SSEC = 1	(Floppy-like interface)
STP = 0x0e	(Stepping rate = 30ms)
Data Length:	
PAD = 1	(ID data pad = 0x0)
POL = 1	(Polling mode disable)
DTL = 0x200	(Data Length = 512 bytes)
ETN = 0x3	(End track - 1)
ESN = 0x11	(End sector - 1)
GPL2 = 0xd	(Gap Length 2)
RWC = 0x80	(Reduce write current cylinder number)

Phase #15 Tests

Test Numbers:	1 through 8
Function:	These tests check the FIFO.
Procedure:	Eight data patterns are written to the FIFO and checked for validity.
Hardware Tested:	The XDC FIFO is tested.
Data Returned:	The number of the test that failed, the actual data, and the expected data are returned.
Notes:	None

==========

Test Number:	9
Function:	This test checks that the XDC can be initialized with the specify command.
Procedure:	If the FIFO is determined to be good, the XDC is initialized with the specify command.
Hardware Tested:	The XDC is tested.
Data Returned:	The number of the test that failed, the actual data, and the expected data are returned.
Notes:	None

Phase #16 — HDC XD Status Tests

Phase Name:	Sense Unit Status Command Tests (xdc_2)
Type:	Demand
Function:	This phase checks the XDC sense unit status command.
Tests:	Test 1 — initializes the HDC chip. Test 2 — recalibrates the drive. Test 3 — tests for end of seek. Test 4 — sends the sense unit status command to the HDC. Test 5 — checks for sense unit status command completion. Test 6 — checks for drive ready and drive selected.
Time:	2 seconds
Warnings:	None
Notes:	None

Phase #16 Tests

Test Number:	1
Function:	This test initializes the Hard Disk Controller (HDC) chip.
Procedure:	The HDC chip is initialized.
Hardware Tested:	The XDC is tested.
Data Returned:	The number of the test that failed, the actual data, and the expected data are returned.
Notes:	None

==========

Test Number:	2
Function:	This test recalibrates the drive.
Procedure:	The drive is recalibrated.
Hardware Tested:	The XDC is tested.
Data Returned:	The number of the test that failed, the actual data, and the expected data are returned.
Notes:	None

Test Number:	3
Function:	This test checks for end of seek.
Procedure:	The end of seek is verified.
Hardware Tested:	The XDC is tested.
Data Returned:	The number of the test that failed, the actual data, and the expected data are returned.
Notes:	None

==========

Test Number:	4
Function:	This test sends the sense unit status command to the HDC.
Procedure:	The sense unit status command is sent to the HDC.
Hardware Tested:	The XDC is tested.
Data Returned:	The number of the test that failed, the actual data, and the expected data are returned.
Notes:	None

==========

Test Number:	5
Function:	This test checks for the completion of the sense unit status command.
Procedure:	Completion of the sense unit status command is verified.
Hardware Tested:	The XDC is tested.
Data Returned:	The number of the test that failed, the actual data, and the expected data are returned.
Notes:	None

==========

Test Number:	6
Function:	This test checks for drive ready and drive selected.
Procedure:	The XDC is checked for drive ready and drive selected.
Hardware Tested:	The XDC is tested.
Data Returned:	The number of the test that failed, the actual data, and the expected data are returned.
Notes:	None

Phase #17 — HDC XD Seek Tests

Phase Name: XDC Seek Tests (xdc_3)

Type: Demand

Function: This phase tests the ability of the XDC to seek to various regions of hard disk.

Tests: Test 1 — uses the specify command to initialize the NEC chip.

Test 2 — returns the heads to cylinder 0.

Test 3 — seeks to cylinder 0.

Test 4 — reads sector 0 to determine the physical size of the disk.

Test 5 — tests for read command completion of disk.

Tests 6 through 25 — seek to various regions of disk0.

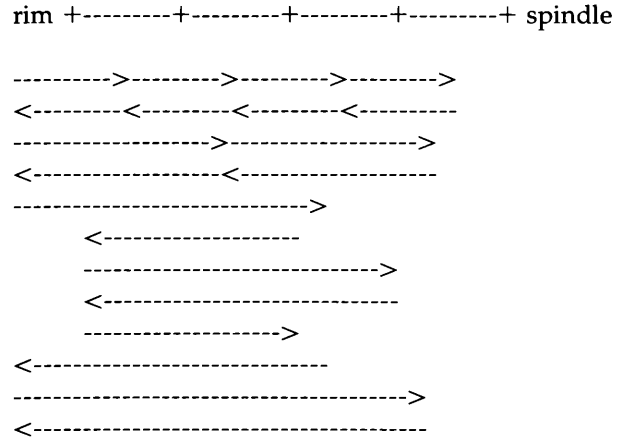
Tests 26 through 100 — no tests run.

Tests 101 through 125 — are identical to Tests 6 through 25 except disk1 is tested.

Time: 8 seconds

Warnings: None

Notes: This phase is designed to test the ability of the heads to move from cylinder to cylinder. The disk is divided into four regions between the rim and the spindle of the disk. The cylinders selected for the seek tests and the test sequence are as follows:



The sequence (from top to bottom in the figure) of seeks selected is designed to perform seeks between any adjacent regions in both the forward (toward the spindle) and backward directions. The sequence requires no "special trip" for the heads to set up for the next seek. At the end of all the tests, the heads should be at cylinder 0. Normal seek (not buffer seek) will not be tested since it is a feature not used by XDC. Overlap seek will be attempted when more than one disk drive is equipped.

Phase #17 Tests

Test Number: 1

Function: This test initializes the NEC chip with the specify command.

Procedure: The specify command is used to initialize the NEC chip.

Hardware Tested: The NEC chip is tested.

Data Returned: The number of the test that failed, the actual data, and the expected data are returned.

Notes: None

==========

Test Number: 2

Function: This test returns the heads to cylinder 0.

Procedure: The heads are returned to 0.

Hardware Tested: The XDC is tested.

Data Returned: The number of the test that failed, the actual data, and the expected data are returned.

Notes: None

==========

Test Number: 3

Function: This test seeks to cylinder 0.

Procedure: Seek to cylinder 0.

Hardware Tested: The XDC is tested.

Data Returned: The number of the test that failed, the actual data, and the expected data are returned.

Notes: None

==========

Test Number: 4

Function: This test determines the physical size of the disk by reading sector 0.

Procedure: Read sector 0.

Hardware Tested: The XDC is tested.

Data Returned: The number of the test that failed, the actual data, and the expected data are returned.

Notes: None

Test Number:	5
Function:	This test checks for the read disk command completion.
Procedure:	The read disk command is exercised.
Hardware Tested:	The XDC is tested.
Data Returned:	The number of the test that failed, the actual data, and the expected data are returned.
Notes:	None

==========

Test Numbers:	6 through 25
Function:	These tests check the ability to seek to various regions of disk0.
Procedure:	Seeks are performed on various regions of disk0.
Hardware Tested:	The XDC is tested.
Data Returned:	The number of the test that failed, the actual data, and the expected data are returned.
Notes:	None

==========

Test Numbers:	101 through 125
Function:	These tests check the ability to seek to various regions of disk1.
Procedure:	Seeks are performed on various regions of disk1.
Hardware Tested:	The XDC is tested.
Data Returned:	The number of the test that failed, the actual data, and the expected data are returned.
Notes:	None

Phase #18 — HDC XD Recalibration Tests

Phase Name: Expansion Disk (XD) Recalibration Tests (xdc_4)

Type: Demand

Function: This phase tests the ability of the XDC to return to cylinder 0 from various regions of the hard disk.

Tests: Test 1 — uses the specify command to initialize the NEC chip.

Test 2 — returns the heads to cylinder 0.

Test 3 — seeks to cylinder 0.

Test 4 — reads sector 0 to determine the physical size of the disk.

Test 5 — tests for read command completion of disk.

Tests 6 through 9 — seek to various regions of disk0.

Tests 10 through 100 — no tests run.

Tests 101 through 109 — are identical to Tests 1 through 9 except disk1 is tested.

Time: 5 seconds

Warnings: None

Notes: This phase is designed to test the ability of the disk to move its head to cylinder 0 (not by seek). The disk is divided into four regions. Recalibration is performed from each of the four regions. The test sequence is as follows:

A seek to one-fourth of all cylinders from the rim is done.
Recalibrate

A seek to one-half of all cylinders from the rim is done.
Recalibrate

A seek to one-fourth of all cylinders from the spindle is done.
Recalibrate

A seek to spindle is done.
Recalibrate.

Phase #18 Tests

Test Number: 1

Function: This test initializes the NEC chip with the specify command.

Procedure: The specify command is used to initialize the NEC chip.

Hardware Tested: The NEC chip is tested.

Data Returned: The number of the test that failed, the actual data, and the expected data are returned.

Notes: None

==========

Test Number: 2

Function: This test returns the heads to cylinder 0.

Procedure: The heads are returned to cylinder 0.

Hardware Tested: The XDC is tested.

Data Returned: The number of the test that failed, the actual data, and the expected data are returned.

Notes: None

==========

Test Number: 3

Function: This test seeks to cylinder 0.

Procedure: A seek is made to cylinder 0.

Hardware Tested: The XDC is tested.

Data Returned: The number of the test that failed, the actual data, and the expected data are returned.

Notes: None

==========

Test Number: 4

Function: This test determines the physical size of the disk by reading sector 0.

Procedure: Sector 0 is read.

Hardware Tested: The XDC is tested.

Data Returned: The number of the test that failed, the actual data, and the expected data are returned.

Notes: None

Test Number:	5
Function:	This test checks for the read disk command completion.
Procedure:	The read disk command completion command is exercised.
Hardware Tested:	The XDC is tested.
Data Returned:	The number of the test that failed, the actual data, and the expected data are returned.
Notes:	None

==========

Test Numbers:	6 through 9
Function:	These tests check the ability to seek to various regions of disk0.
Procedure:	Seeks are performed to various regions of disk0.
Hardware Tested:	The XDC is tested.
Data Returned:	The number of the test that failed, the actual data, and the expected data are returned.
Notes:	None

==========

Test Numbers:	101 through 109
Function:	These tests check the ability to seek to various regions of disk1.
Procedure:	Seeks are performed to various regions of disk1.
Hardware Tested:	The XDC is tested.
Data Returned:	The number of the test that failed, the actual data, and the expected data are returned.
Notes:	None

Phase #19 — HDC XD DATA ID Tests

Phase Name:	Verify ID Tests (xdc_5)
Type:	Demand
Function:	This phase tests the XDC verify ID command.
Tests:	Test 1 — uses the specify command to initialize the NEC chip. Test 2 — returns the heads to cylinder 0. Test 3 — reads the physical drive information and defect table. Test 4 — uses the verify ID command to verify the ID field of the diagnostic cylinder. Test 5 — returns the heads to cylinder 0. Tests 6 through 100 — no tests run. Tests 101 through 105 — are identical to Tests 1 through 5 except disk1 is tested.
Time:	5 seconds
Warnings:	None
Notes:	This phase is designed to test the verify ID command of the XDC. The Identification (ID) field of each sector is read and compared with the expected ID field in memory by the XDC. The XDC retrieves the expected ID field in memory by Direct Memory Access (DMA). The verify ID command will be attempted on all the tracks within the diagnostic cylinder.

Phase #19 Tests

Test Number: 1

Function: This test initializes the NEC chip with the specify command.

Procedure: The specify command is used to initialize the NEC chip.

Hardware Tested: The NEC chip is tested.

Data Returned: The number of the test that failed, the actual data, and the expected data are returned.

Notes: None

==========

Test Number: 2

Function: This test returns the heads to cylinder 0.

Procedure: The heads are returned to cylinder 0.

Hardware Tested: The XDC is tested.

Data Returned: The number of the test that failed, the actual data, and the expected data are returned.

Notes: None

==========

Test Number: 3

Function: This test reads the physical drive information and defect table.

Procedure: The physical drive information and defect table are read.

Hardware Tested: The XDC is tested.

Data Returned: The number of the test that failed, the actual data, and the expected data are returned.

Notes: None

Test Number:	4
Function:	This test uses the verify ID command to verify the ID field of the diagnostic cylinder.
Procedure:	The verify ID command is used to verify the ID field of the diagnostic cylinder.
Hardware Tested:	The XDC is tested.
Data Returned:	The number of the test that failed, the actual data, and the expected data are returned.
Notes:	None

==========

Test Number:	5
Function:	This test returns the heads to cylinder 0.
Procedure:	The heads are returned to cylinder 0.
Hardware Tested:	The XDC is tested.
Data Returned:	The number of the test that failed, the actual data, and the expected data are returned.
Notes:	None

==========

Test Numbers:	101 through 105
Function:	These tests are the same as Tests 1 through 5 except disk1 is tested.
Procedure:	Same as Tests 1 through 5 except for disk1.
Hardware Tested:	The XDC is tested.
Data Returned:	The number of the test that failed, the actual data, and the expected data are returned.
Notes:	None

Phase #20 — HDC XD Read ID Tests

Phase Name:	Read ID Tests (xdc_6)
Type:	Demand
Function:	This phase tests the ability of the XDC to perform the read ID command.
Tests:	Test 1 — uses the specify command to initialize the NEC chip. Test 2 — returns the heads to cylinder 0. Test 3 — reads the physical drive information and defect table. Test 4 — reads and compares the ID field of the diagnostic cylinder. Test 5 through 100 — no tests run. Tests 101 through 104 — are identical to Tests 1 through 4 except disk1 is tested.
Time:	3 seconds
Warnings:	None
Notes:	This phase is designed to test the read ID command of the XDC. The Identification (ID) field of each sector is read by the XDC and transferred to memory by Direct Memory Access (DMA). The ID written into memory can then be examined. The read ID command will be attempted on all tracks within the diagnostic cylinder.

Phase #20 Tests

Test Number:	1
Function:	This test initializes the NEC chip with the specify command.
Procedure:	The specify command is used to initialize the NEC chip.
Hardware Tested:	The NEC chip is tested.
Data Returned:	The number of the test that failed, the actual data, and the expected data are returned.
Notes:	None

==========

Test Number:	2
Function:	This test returns the heads to cylinder 0.
Procedure:	The heads are returned to cylinder 0.
Hardware Tested:	The XDC is tested.
Data Returned:	The number of the test that failed, the actual data, and the expected data are returned.
Notes:	None

Test Number:	3
Function:	This test reads the physical drive information and defect table.
Procedure:	The physical drive information and defect table are read.
Hardware Tested:	The XDC is tested.
Data Returned:	The number of the test that failed, the actual data, and the expected data are returned.
Notes:	None

==========

Test Number:	4
Function:	This test reads and compares the ID field of the diagnostic cylinder.
Procedure:	The diagnostic cylinder ID field is read and compared.
Hardware Tested:	The XDC is tested.
Data Returned:	The number of the test that failed, the actual data, and the expected data are returned.
Notes:	None

==========

Test Numbers:	101 through 104
Function:	These tests are the same as Tests 1 through 4 except that disk1 is used.
Procedure:	Same as Tests 1 through 4 except that disk1 is tested.
Hardware Tested:	The XDC is tested.
Data Returned:	The number of the test that failed, the actual data, and the expected data are returned.
Notes:	None

Phase #21 — HDC XD Read/Write Tests

Phase Name: Data Read/Write Tests (xdc_7)

Type: Demand

Function: This phase performs a Read/Write test on the diagnostic cylinder.

Tests:

Test 1 — uses the specify command to initialize the NEC chip.

Test 2 — moves the heads to cylinder 0.

Test 3 — reads the physical drive information and defect table.

Test 4 — writes the diagnostics cylinder with data.

Test 5 — returns the heads to cylinder 0.

Test 6 — uses the check command to verify the correct generation of the CRC code for the data.

Test 7 — returns the heads to cylinder 0.

Test 8 — reads data from the diagnostic cylinder and compares it to data in a buffer.

Test 9 — returns the heads to cylinder 0.

Tests 10 through 100 — no tests run.

Tests 101 through 109 — are identical to Tests 1 through 9 except disk1 is tested.

Time: 50 seconds

Warnings: None

Notes: This phase is designed to test the read/write capabilities of the disk. A complete cylinder will be written and read. The tests are done over the diagnostic cylinder. During the course of this phase, the check command is used to verify that the CRC bytes are correct. The verify data command is used to check the data. The test sequence is as follows:

Write - Write data to the disk.
Check - Check that CRC bytes are correct.
Verify - Verify data written against the buffers.
Read - Read the data and then compare.
Error Detection - Read the data with induced errors.
CRC - Check CRC Error Detection.

Phase #22 — HDC XD Scan Tests

Phase Name:	Sense Interrupt Status Tests (xdc_8)
Type:	Demand
Function:	This phase tests the sense interrupt status command.
Tests:	Test 1 — uses the specify command to initialize the NEC chip. Test 2 — waits for the specify command to finish. Test 3 — moves the heads back to cylinder 0. Test 4 — reads the physical drive information and defect table. Test 5 — seeks to the diagnostic cylinder. Test 6 — issues the sense interrupt status command. Test 7 — waits for the sense interrupt status command to finish. Test 8 — uses the specify command to reinitialize the NEC chip. Test 9 — moves the heads back to cylinder 0. Tests 101 through 109 — are identical to Tests 1 through 9 except disk1 is tested.
Time:	110 seconds
Warnings:	None
Notes:	When a change of disk status occurs, the XDC interrupts the host CPU. The sense interrupt status command reveals the cause of interrupt, such as seek end, disk ready change, seek error, or equipment check. A seek is issued in this phase, and then a wait for the SRQ (Sense Interrupt Request) flag in the Status register is performed. When the SRQ flag is asserted, the sense interrupt status command is issued, and seek end is expected in the interrupt status. The abnormal termination of the command is tested by attempting to execute a sense interrupt status command not preceded by a change of disk status (no seek issued).

Chapter 12: Math Accelerator Unit Diagnostics

Introduction

This chapter contains the diagnostic phases and tests for the 3B2 computer Math Accelerator Unit (MAU). The MAU is a WE 31006, WE 32106, or WE 32206 Microprocessor that plugs into the System Board (SBD), providing enhanced computational abilities for the 3B2 computer. The MAU only exist on Version 3 computers.

If your 3B2 computer is equipped with a MAU, the Equipped Device Table (EDT) should list it. You can use the diagnostic monitor command "show" (DGMON > **s**) to print a copy of the EDT. If the MAU is not listed in the EDT and a VOID or NULL is listed, the MAU may be faulty. Three diagnostic phases run tests on the MAU.

The Table of Contents listing on the previous page will help you locate each phase and its associated test. The phase and test descriptions are organized numerically in the same order that the phases are run on the 3B2 computer.

Phase Descriptions

Phase #1 — Math Accelerator Unit #1 Diagnostics

Phase Name:	Math Accelerator Unit (MAU) (mau1_tst)
Type:	Normal
Function:	This phase tests for the presence of a MAU device, and if it is found, the phase pattern tests the MAU Data bus, operand registers F0 through F3, and the MAU Data register.
Tests:	Test 1 — pattern tests the MAU Data bus by walking a one through a field of zeros as a single precision operand to register F0. Tests 2 through 5 — pattern test the MAU registers F0 through F3 as a double-extended precision operand with a modified data pattern of 0x55 and 0xaa. Test 6 — pattern checks the MAU Data register as a double-extended precision operand with a modified data pattern of 0x55 and 0xaa.
Time:	1 second
Warnings:	None
Notes:	All operand pattern tests are checked for register-to-register cross talk.

Phase #1 Tests

Test Number:	1
Function:	This test pattern checks the MAU Data bus.
Procedure:	A one is walked through a field of zeros as a single precision operand by moving the operand to and from register F0.
Hardware Tested:	The MAU is tested.
Data Returned:	The failing test number (last number displayed on the system console prior to failure) is returned.
Notes:	None

Test Numbers:	2 through 5
Function:	These tests pattern check MAU registers F0 through F3.
Procedure:	Each register is tested with modified 0x55 and 0xaa patterns, checking the other registers for cross talk.
Hardware Tested:	The MAU is tested.
Data Returned:	The failing test number (last number displayed on the system console prior to failure) is returned.
Notes:	None

==========

Test Number:	6
Function:	This test pattern checks the MAU Data register.
Procedure:	The Data register is tested with a modified data pattern of 0x55 and 0xaa.
Hardware Tested:	The MAU is tested.
Data Returned:	The failing test number (last number displayed on the system console prior to failure) is returned.
Notes:	None

Phase #2 — Math Accelerator Unit #2 Diagnostics

Phase Name: Math Accelerator Unit (MAU) (mau2_tst)

Type: Normal

Function: This phase tests that a faulty MAU can be detected. Also, all of the "Sticky" and Mask bits of the MAU Auxiliary Status Register (ASR) are tested for the ability to be set, cleared, and masked.

Tests: Test 1 — causes a MAU fault condition by dividing by zero, testing the ASR divide-by-zero "Sticky" (QS) and Mask bits (QM).

Test 2 — checks the ASR Invalid Operation "Sticky" (IS) and Mask bits (IM).

Test 3 — checks the ASR Overflow "Sticky" (OS) and Mask bits (OM).

Test 4 — checks the ASR Inexact "Sticky" (PS) and Mask bits (PM).

Test 5 — checks the ASR Underflow "Sticky" (US) and Mask bits (UM).

Test 6 — checks the ASR Integer Overflow (IO) Indicator bit.

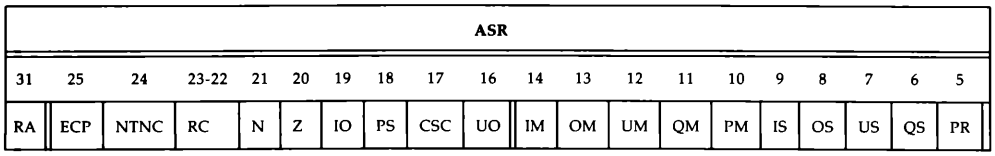
Test 7 — checks the ASR Unordered (UO) Indicator bit.

Test 8 — checks the ASR Nontrapping Not a Number (NAN) Control (NTNC) bit.

Time: 1 second

Warnings: None

Notes: The Negative (N), Zero (Z), Result Available (RA), and Partial Remainder (PR) bits are tested during the mau3_tst() diagnostic phase.



ASR																			
31	25	24	23-22	21	20	19	18	17	16	14	13	12	11	10	9	8	7	6	5
RA	ECP	NTNC	RC	N	Z	IO	PS	CSC	UO	IM	OM	UM	QM	PM	IS	OS	US	QS	PR

Phase #2 Tests

Test Number: 1

Function: This test checks that a faulty MAU can be detected.

Procedure: A divide-by-zero is performed, and the fact that an exception was generated is confirmed. Then, the divide-by-zero fault is inhibited, and the process is repeated. The fact that an exception is not generated is confirmed. In both cases, the MAU_ASR is checked to ensure that the correct flags were set.

Hardware Tested: The MAU is tested.

Data Returned: The failing test number (last number displayed on the system console prior to failure) is returned.

Notes: None

==========

Test Number: 2

Function: This test checks the ASR Invalid Operation "Sticky" (IS) and Mask bits (IM).

Procedure: An Integer to Float Conversion (ITOF) is performed with the source specified as MAU register F0. It is confirmed that an exception was generated. Then, Invalid Operation exceptions are inhibited, and the procedure is repeated. It is confirmed that an exception was not generated. In both cases, it is checked that the correct MAU_ASR flags were set.

Hardware Tested: The MAU is tested.

Data Returned: The failing test number (last number displayed on the system console prior to failure) is returned.

Notes: None

==========

Test Number: 3

Function: This test checks the ASR Overflow "Sticky" (OS) and Mask bits (OM).

Procedure: An Overflow exception is forced by using the MAU division instruction. It is confirmed that an exception was taken. Then, Overflow exceptions are inhibited, and the procedure is repeated. It is confirmed that an exception was not taken. In both scenarios, the MAU_ASR is checked for correct flag generation.

Hardware Tested: The MAU is tested.

Data Returned: The failing test number (last number displayed on the system console prior to failure) is returned.

Notes: None

Test Number:	4
Function:	This test checks the Inexact "Sticky" (PS) and Mask bits (PM).
Procedure:	An Inexact exception is caused by using the MAU RTOI instruction. It is confirmed that an exception was generated. Then, Inexact exceptions are blocked, and the procedure is repeated. It is confirmed that an exception was not taken. In both cases, the MAU_ASR is checked for correct flag values.
Hardware Tested:	The MAU is tested.
Data Returned:	The failing test number (last number displayed on the system console prior to failure) is returned.
Notes:	None

==========

Test Number:	5
Function:	This test checks the ASR Underflow "Sticky" (US) and Mask bits (UM).
Procedure:	A MAU Underflow exception is caused by using the MAU division instruction. It is confirmed that an exception was generated. Then, the Underflow exceptions are inhibited, and the procedure is repeated. It is checked that an exception was not generated. In both tests, the correct setting of MAU_ASR flag variables is checked.
Hardware Tested:	The MAU is tested.
Data Returned:	The failing test number (last number displayed on the system console prior to failure) is returned.
Notes:	None

==========

Test Number:	6
Function:	This test checks the ASR Integer Overflow (IO) Indicator bit.
Procedure:	An exception is generated by using the MAU FTOI instruction. The fact that an exception was taken is confirmed, and the correct setting of the MAU_ASR flags is checked.
Hardware Tested:	The MAU is tested.
Data Returned:	The failing test number (last number displayed on the system console prior to failure) is returned.
Notes:	None

Test Number:	7
Function:	This test checks the ASR Unordered (UO) Indicator bit.
Procedure:	An Unordered exception is forced by using the MAU CMP instruction. It is checked that an exception was taken and that the MAU_ASR set the correct flag values.
Hardware Tested:	The MAU is tested.
Data Returned:	The failing test number (last number displayed on the system console prior to failure) is returned.
Notes:	None

==========

Test Number:	8
Function:	This test checks the ASR NTNC bit.
Procedure:	An Invalid Operation condition is caused with the NTNC bit set by performing an ITOF with the source specified as a MAU register. It is confirmed that an exception was generated and that the MAU_ASR has the correct flags set.
Hardware Tested:	The MAU is tested.
Data Returned:	The failing test number (last number displayed on the system console prior to failure) is returned.
Notes:	None

Phase #3 — Math Accelerator Unit #3 Diagnostics

Phase Name:	Math Accelerator Unit (MAU) (mau3_tst)
Type:	Normal
Function:	This phase tests all of the possible MAU operation codes.
Tests:	Tests 1 through 8 — check the CMP, CMPS, CMPE, and CMPES operation codes. Tests 9 and 10 — check the add and subtract operation codes. Tests 11 through 13 — check the multiply, divide, and REM operation codes. Test 14 — checks the NEG operation code. Test 15 — checks the ABS operation code. Test 16 — checks the SQRT operation code. Tests 17 through 21 — check the RTOI, ITOF, DTOF, FTOD, and FTOI operations.
Time:	1 second
Warnings:	None
Notes:	The Negative (N), Zero (Z), Result Available (RA), and Partial Remainder (PR) bits are tested during this test phase.

Phase #3 Tests

Test Numbers:	1 through 8
Function:	These tests check the CMP, CMPS, CMPE and CMPES operation codes.
Procedure:	Operand1 and operand2 are set to 123.0. Each compare instruction is run checking for equality. Operand1 is set to 100.0. Each compare instruction is run checking for inequality.
Hardware Tested:	The MAU is tested.
Data Returned:	The failing test number (last number displayed on the system console prior to failure) is returned.
Notes:	None

Test Numbers: 9 and 10

Function: These tests check the add and subtract operation codes. Many of the following tests (9 through 21) use the MAU compare instruction to determine if the test passed.

Procedure:
Operand1 is set to 39.0 and operand2 is set to 12.0.
Move operand2 to MAU F0 and operand1 to MAU F1.
Perform an Add.
Set expctd_rslt to 51.0.
Perform MAU F0 + MAU F1 -> result.
Perform a Subtract.
Set expctd_rslt to 27.0.
Perform MAU F1 - MAU F0 -> result.
Move result to MAU F0 and expctd_rslt to MAU F1.
Compare MAU F0 with MAU F1 and check for equality.

Hardware Tested: The MAU is tested.

Data Returned: The failing test number (last number displayed on the system console prior to failure) is returned.

Notes: None

==========

Test Numbers: 11 through 13

Function: These tests check the multiply, divide, and REM operations codes.

Procedure:
Clear the result variable.
Perform a Multiply.
Set operand1 to 239.0 and operand2 to 11.0.
Set expctd_rslt to 2629.0.
Move operand1 to MAU F1 and operand2 to MAU F0.
Perform MAU F0 * MAU F1 -> result.
Perform a Divide.
Set operand1 to 125.0 and operand2 to 5.0.
Set expctd_rslt to 25.0.
Move operand1 to MAU F1 and operand2 to MAU F0.
Perform MAU F1 / MAU F0 -> result.
Perform a REM.
Set operand2 to 10.0.
Set expctd_rslt to 5.0.
Move operand1 to MAU F1 and operand2 to MAU F0.
Perform MAU F1 % MAU F0 -> result.
Move result to MAU F0 and expctd_rslt to MAU F1.
Compare MAU F0 and MAU F1 and check for equality.

Hardware Tested: The MAU is tested.

Data Returned: The failing test number (last number displayed on the system console prior to failure) is returned.

Notes: None

Test Number:	14
Function:	This test checks the Negative (NEG) operation code.
Procedure:	Set operand1 to 239.0 and expctd_rslt to -239.0. Move operand1 to MAU F0. Perform -(MAU F0). Compare MAU F0 and expctd_rslt and check for equality.
Hardware Tested:	The MAU is tested.
Data Returned:	The failing test number (last number displayed on the system console prior to failure) is returned.
Notes:	None

==========

Test Number:	15
Function:	This test checks the Absolute (ABS) operation code.
Procedure:	Set operand1 to -239.0 and expctd_rslt to 239.0. Move operand1 to MAU F0. Perform ABS MAU F0. Compare MAU F0 and expctd_rslt and check for equality.
Hardware Tested:	The MAU is tested.
Data Returned:	The failing test number (last number displayed on the system console prior to failure) is returned.
Notes:	None

==========

Test Number:	16
Function:	This test checks the Square Root (SQRT) operation code.
Procedure:	Set operand1 to 144.0 and expctd_rslt to 12.0. Move operand1 to MAU F0. Perform SQRT MAU F0. Compare MAU F0 and expctd_rslt and check for equality.
Hardware Tested:	The MAU is tested.
Data Returned:	The failing test number (last number displayed on the system console prior to failure) is returned.
Notes:	None

Test Numbers:	17 through 21
Function:	These tests check the RTOI, ITOF, DTOF, FTOD, and FTOI operations.
Procedure:	RTOI: Set operand1 to 123.45 and expctd_rslt to 123.0. Perform RTOI operand1 -> MAU F0. Compare expctd_rslt and MAU F0 checking for equality. ITOF: Set operand1 to 123. Perform ITOR operand1 -> MAU F0. Compare expctd_rslt and MAU F0 checking for equality. FTOI: Set operand1 to 123.05 and expctd_rslt to 123. Clear result. Move operand1 to MAU F0. Perform FTOI MAU F0 -> result. Compare result and expctd_rslt checking for equality. DTOF: Set operand1 to 0x123a and expctd_rslt to 123.0. Perform DTOF operand1 -> MAU F0. Compare expctd_rslt and MAU F0 checking for equality. FTOD: Set operand1 to 123.00 and expctd_rslt to 0x123a. Move operand1 to MAU F0. Perform FTOD MAU F0 -> result. Compare result and expctd_rslt checking for equality.
Hardware Tested:	The MAU is tested.
Data Returned:	The failing test number (last number displayed on the system console prior to failure) is returned.
Notes:	None

Chapter 13: Virtual Cache Diagnostics

Introduction

This chapter contains a description of the diagnostic phases and tests for the 3B2 computer Virtual Cache (VCACHE) card. If your 3B2 computer is equipped with a VCACHE card, the Equipped Device Table (EDT) should list it. You can use the diagnostic monitor command show (DGMON> **s**) to display the EDT. If the VCACHE card is not listed in the EDT and a NULL or VOID is, the VCACHE card may be faulty. Two diagnostic phases test all aspects of the VCACHE card.

The VCACHE card is optional on the AT&T 3B2/500 computers. The VCACHE card is required on the 3B2/600 and 3B2/700 computers running the UNIX System V Release 3.1.1 operating system. The VCACHE card is not used in 3B2 computers running the UNIX System V Release 3.2.2 operating system.

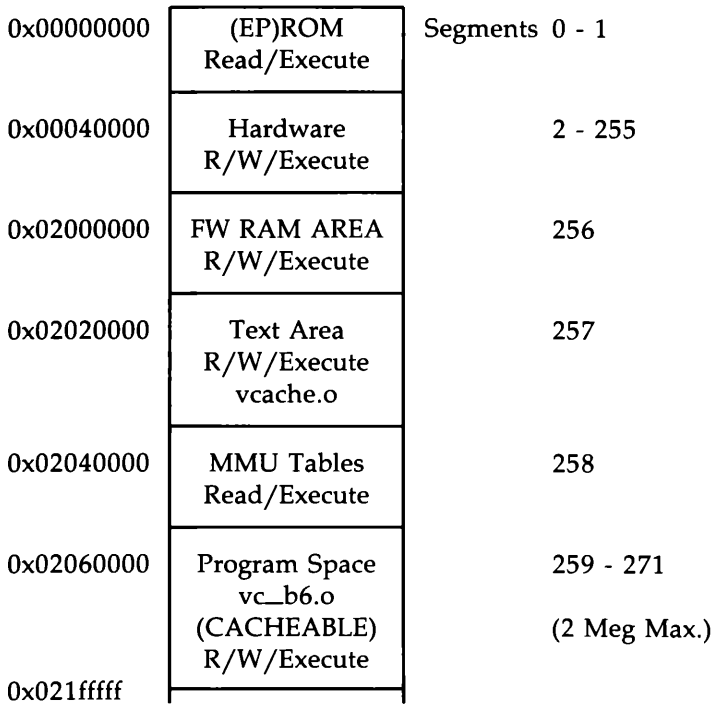
Two diagnostic phases run tests on all major VCACHE card components. The Table of Contents Listing on the previous page will help you locate the descriptions of each VCACHE card diagnostic phase and its associated tests. The phase and test descriptions are organized numerically in the same order that the phases run on the 3B2 computer.

Phase Descriptions

Phase #1 — Virtual Cache Diagnostics

Phase Name:	Virtual Cache (VCACHE) card
Type:	Normal
Function:	This phase sets up the Memory Management Unit (MMU), turns the VCACHE card on, and executes the program "main."
Tests:	Test 1 — pattern tests the VCACHE card memory in physical mode. Test 2 — checks hit/miss accesses for full word read. Test 3 — checks hit/miss accesses for full word write. Test 4 — confirms that the VCACHE card correctly responds to byte miss write accesses. Test 5 — confirms that the VCACHE card correctly responds to half word miss write accesses. Test 6 — confirms that the VCACHE card correctly responds to byte hit write accesses. Test 7 — confirms that the VCACHE card correctly responds to half word hit write accesses. Test 8 — confirms that the VCACHE card flushes correctly. Test 9 — confirms that the VCACHE card flushes data correctly.
Time:	5 seconds
Warnings:	None

Notes: The MMU setup is as follows (1 to 1 mapping, Section 0 only):



==========

Phase #1 Tests

Test Number: 1

Function: This test pattern tests the VCACHE card RAM devices in physical mode.

Procedure: Using the test patterns 0xffffffffL, 0xccccccccL, 0xaaaaaaaaL, 0x55555555L, 0x33333333L and 0x00000000L, pattern test the RAM devices on the VCACHE card.

Hardware Tested: The VCACHE card is tested.

Data Returned: The failing test number, the failing location, and the failing pattern are returned.

Notes: None

Test Number:	2
Function:	This test ensures that the VCACHE card correctly responds to full word hit/miss read accesses.
Procedure:	In virtual mode, write a pattern to memory with the VCACHE card disabled. Enable the VCACHE card and read memory. This will load the VCACHE card. Disable the VCACHE card and change the pattern in memory. Enable the VCACHE card and perform a read. The pattern in the VCACHE card should be pattern read.
Hardware Tested:	The VCACHE card is tested.
Data Returned:	The failing test number, the failing address, the expected pattern, and the pattern received are returned.
Notes:	None

==========

Test Number:	3
Function:	This test ensures that the VCACHE card correctly responds to full word hit/miss write accesses.
Procedure:	In virtual mode, write pattern to memory with the VCACHE card enabled, thus loading the VCACHE card. Disable the VCACHE card and change the pattern in memory. Enable the VCACHE card and perform a read. The pattern in the VCACHE card (the first pattern) should be pattern read.
Hardware Tested:	The VCACHE card is tested.
Data Returned:	The failing test number, the failing address, the expected pattern, and the pattern received are returned.
Notes:	None

==========

Test Number:	4
Function:	This test ensures that the VCACHE card correctly responds to byte miss write accesses.
Procedure:	In virtual mode, write pattern to memory using byte accesses with the VCACHE card enabled. Disable the VCACHE card and change the memory test pattern. Enable the VCACHE card and verify that the pattern read is the second pattern.
Hardware Tested:	The VCACHE card is tested.
Data Returned:	The failing test number, the failing address, the expected pattern, and the pattern received are returned.
Notes:	None

Test Number: 5

Function: This test ensures that the VCACHE card correctly responds to half word miss write accesses.

Procedure: In virtual mode, write pattern to memory using half word accesses with the VCACHE card enabled. Disable the VCACHE card and change the memory test pattern. Enable the VCACHE card and verify that the pattern read is the second pattern.

Hardware Tested: The VCACHE card is tested.

Data Returned: The failing test number, the failing address, the expected pattern, and the pattern are returned.

Notes: None

==========

Test Number: 6

Function: This test ensures that the VCACHE card correctly responds to byte hit write accesses.

Procedure: In virtual mode, write pattern to memory using word accesses with the VCACHE card enabled. This should fill the VCACHE card. Then fill memory with second test pattern using byte accesses. Disable the VCACHE card and verify that the second pattern was written into memory. Fill memory with the third test pattern using word accesses. Enable the VCACHE card and verify that the pattern read is the second pattern.

Hardware Tested: The VCACHE card is tested.

Data Returned: The failing test number, the failing address, the expected pattern, and the pattern are returned.

Notes: None

==========

Test Number: 7

Function: This test ensures that the VCACHE card correctly responds to half word hit write accesses.

Procedure: In virtual mode, write pattern to memory using half word accesses with the VCACHE card enabled. This should fill the VCACHE card. Then fill memory with second test pattern using half word accesses. Disable the VCACHE card and verify that the second pattern was written into memory. Fill memory with the third test pattern using half word accesses. Enable the VCACHE card and verify that the pattern read is the second pattern.

Hardware Tested: The VCACHE card is tested.

Data Returned: The failing test number, the failing address, the expected pattern, and the pattern are returned.

Notes: None

Test Number:	8
Function:	This test ensures that the VCACHE card flushes correctly.
Procedure:	In virtual mode and with the VCACHE card disabled, write the memory with the test pattern using word accesses. Enable the VCACHE card and confirm that the pattern was written into memory (this should load the VCACHE card). Disable the VCACHE card and write the second test pattern. Enable the VCACHE card and confirm that the test pattern is present in the VCACHE card. Flush the VCACHE card, read the memory, and confirm that the second test pattern is now read.
Hardware Tested:	The VCACHE card is tested.
Data Returned:	The failing test number, the failing address, the expected pattern, and the pattern are returned.
Notes:	None

==========

Test Number:	9
Function:	This test ensures that the VCACHE card flushes data correctly.
Procedure:	In virtual mode and with the VCACHE card disabled, write the memory with the test pattern using word accesses. Enable the VCACHE card and confirm that the pattern was written into memory (this should load VCACHE card). Disable the VCACHE card and write the second test pattern. Enable the VCACHE card and confirm that the test pattern is present in the VCACHE card. Flush the VCACHE card, read the memory, and confirm that the second test pattern is now read.
Hardware Tested:	The VCACHE card is tested.
Data Returned:	The failing test number, the failing address, the expected pattern, and the pattern are returned.
Notes:	None

Phase #2 — Extended Virtual Cache Diagnostics

Phase Name: Extended Virtual Cache (EVCACHE)

Type: Demand

Function: This phase sets up the Memory Management Unit (MMU), turns the VCACHE card on, and executes the program "main."

Tests: Test 1 — confirms that the ICACHE bit in the CPU Process Status Word (PSW) works.

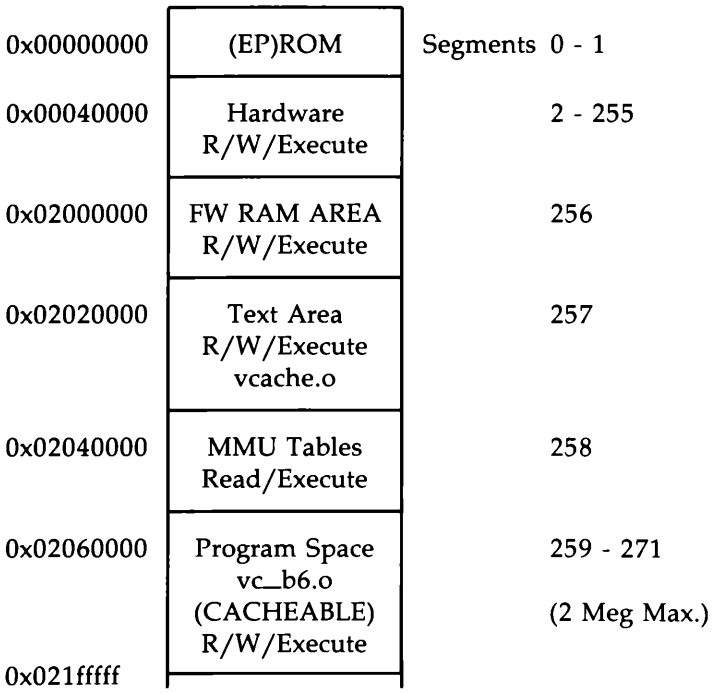
Test 2 — confirms that caching of instructions in the VCACHE card works.

Test 3 — confirms that using the ICACHE bit and the VCACHE card works.

Time: 25 seconds

Warnings: None

Notes: The MMU setup is as follows (1 to 1 mapping, Section 0 only):



Phase #2 Tests

Test Number: 1

Function: This test ensures that the Instruction Cache (ICACHE) in CPU works.

Procedure: Run the benchmark program without the ICACHE bit on. Then turn the ICACHE bit on and confirm that the benchmark program ran faster than with it off.

Hardware Tested: The CPU ICACHE is tested.

Data Returned: The failing test number is displayed on the system console.

Notes: None

==========

Test Number: 2

Function: This test ensures that the benchmark program executes faster when executing with the VCACHE card versus previous straight CPU time.

Procedure: Turn the VCACHE card on and execute the program. Confirm that the benchmark program executed faster than when the CPU alone was used.

Hardware Tested: The VCACHE card is tested.

Data Returned: The failing test number is displayed on the system console.

Notes: None

==========

Test Number: 3

Function: This test ensures that the benchmark runs faster when both the CPU ICACHE bit and the VCACHE card are on.

Procedure: Run the benchmark program, and confirm that the benchmark program runs faster than any of the previously executed times.

Hardware Tested: The VCACHE card and CPU ICACHE bit are tested.

Data Returned: The failing test number is displayed on the system console.

Notes: None

Chapter 14: Small Computer System Interface Diagnostics

Introduction

This chapter contains a description of the diagnostic phases for the 3B2 computer Small Computer System Interface (SCSI) devices. If your 3B2 computer is equipped with a SCSI device, the Equipped Device Table (EDT) should list it. You can use the diagnostic monitor command "show" (DGMON> **s**) to display the EDT. If a SCSI device is not listed in the EDT and a VOID or NULL is listed, the SCSI device may be considered faulty.

Twenty-four diagnostic phases check all aspects of the SCSI. The first 16 phases test the Host Adapter card. The Table of Contents listing will help you locate each phase and its associated tests. The phase and test descriptions are organized numerically in the same order that the phases are run on the 3B2 computer.

Phase Descriptions

Phase #1 — SCSI CIO Sanity

Phase Name: SCSI CIO Sanity Test

Type: Demand

Function: This phase tests the integrity of the CIO queue.

Test: Test 1 — verifies that the diagnostic results can be transferred to the SBD.

Time: 1 second

Warnings: None

Notes: This phase is used to determine the basic sanity of the Common I/O (CIO) portion of the SCSI. The "procedure" used in this phase is used by all other diagnostic phases to execute tests and return results to the System Board (SBD). If the SCSI cannot execute this phase, it will not successfully run any other phase. When this phase fails, a "Phase Time-out" message usually occurs. This means that the SBD tried to communicate with the SCSI, but timed out while waiting for a response. If this phase fails, the following hardware may be faulty:

- INTEL 80186 Microprocessor
- SCSI Input/Output (I/O) interface circuitry
- SCSI Random Access Memory (RAM)/Read Only Memory (ROM)
- SCSI Address/Data bus.

Phase #1 Test

Test Number: 1

Function: This test verifies the integrity of the CIO queue.

Procedure: This test uses the following standard procedure:

1. The I/O slot number of the Host Adapter card is determined.
2. The SCSI is reset.
3. The sysgen data block is initialized.
4. The SCSI is initialized (sysgen) by sending express [Interrupt 0 (INT0)] and attention (INT1) interrupts sequentially.
5. The X86 code is downloaded by using the CIO firmware command [Force Call to Function (FCF)].

Hardware Tested: The CIO hardware on the Host Adapter is tested.

Data Returned: The test number that failed, the actual data, and the expected data are returned.

Phase #2 — SCSI Upper RAM Write/Read

Phase Name:	SCSI Upper RAM Write/Read Test
Type:	Demand
Function:	This phase diagnoses and reports errors in the SCSI RAM and refresh circuitry.
Tests:	Tests 1 through 6 — checks the upper 64 kilobytes of peripheral RAM.
Time:	12 seconds
Warnings:	None
Notes:	None

Phase #2 Tests

Test Numbers:	1 through 4
Function:	These tests write patterns of 0's and 1's in the upper 64 kilobytes of peripheral RAM.
Procedure:	A 1-second loop is created for the Central Processing Unit (CPU) and pseudo-random data is read back and checked.
Hardware Tested:	The upper RAM locations are tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	5 and 6
Function:	These tests diagnose and report errors in the SCSI RAM and refresh circuitry.
Procedure:	The CPU is put in a 1-second loop. The pseudo-random data is read back. If the refresh is not working, the pattern is mangled.
Hardware Tested:	The upper RAM locations are tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

Phase #3 — SCSI Lower RAM Write/Read

Phase Name: SCSI Lower RAM Write/Read Tests.

Type: Demand

Function: This phase diagnoses and reports errors in the Host Adapter RAM and refresh circuitry.

Tests: Tests 1 through 6 — check the lower 64 kilobytes of peripheral RAM.

Time: 12 seconds

Warnings: None

Notes: None

Phase #3 Tests

Test Numbers: 1 through 4

Function: These tests perform memory pattern testing.

Procedure: These tests write patterns of 0's and 1's in the lower 64 kilobytes of peripheral RAM.

Hardware Tested: The lower RAM locations are tested.

Data Returned: The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers: 5 and 6

Function: These tests diagnose and report errors in the lower 64 kilobytes of RAM and refresh circuitry.

Procedure: The CPU is put in a 1-second loop. The pseudo-random data is read back. If the refresh is not working, the pattern is mangled.

Hardware Tested: The lower RAM locations are tested.

Data Returned: The test number that failed, the actual data, and the expected data are returned.

Phase #4 — SCSI ROM Check Sum

Phase Name:	SCSI ROM Check Sum Test
Type:	Demand
Function:	This phase verifies the integrity of the SCSI ROM contents.
Test:	Test 1 — performs a check sum on the I/O ROM.
Time:	2 seconds
Warnings:	None
Notes:	None

Phase #4 Test

Test Number:	1
Function:	This test verifies the integrity of the I/O ROM.
Procedure:	This test reads the ROM byte-by-byte, calculating the check sum value.
Hardware Tested:	The SCSI ROM is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

Phase #5 — SCSI CPU Upper Chip Select

Phase Name:	SCSI CPU Upper Chip Select Test
Type:	Demand
Function:	This phase tests the operation of the Upper Memory Chip Select (UMCS), Peripheral Access Chip Select (PACS), Middle Memory Chip Select (MMCS), and Middle Peripheral Chip Select (MPCS) registers of the INTEL 80186 CPU.
Tests:	Tests 1 through 12 — check the UMCS register. Tests 13 through 29 — check the PACS register. Tests 30 through 37 — check the MMCS register. Tests 38 through 48 — check the MPCS register.
Time:	2 seconds
Warnings:	None
Notes:	None

Phase #5 Tests

Test Numbers:	1 through 12
Function:	These tests verify the UMCS register.
Procedure:	Write a valid data pattern to the register, read the register, and compare the value with the value that was written.
Hardware Tested:	The UMCS register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	13 through 29
Function:	These tests verify that the PACS register is functional.
Procedure:	Write a valid data pattern to the register, read the register, and compare the value read with the value that was written.
Hardware Tested:	The PACS register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

Test Numbers:	30 through 37
Function:	These tests verify that the MMCS register is functional.
Procedure:	Write a valid data pattern to the register, read the register, and compare the value read with the value that was written.
Hardware Tested:	The MMCS register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.
Notes:	None

==========

Test Numbers:	38 through 48
Function:	These tests verify that the MPCS register is functional.
Procedure:	Write a valid data pattern to the register, read the register, and compare the value read with the value that was written.
Hardware Tested:	The MPCS register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

Phase #6 — SCSI CPU DMA Internal

Phase Name: SCSI CPU DMA Internal Test

Type: Demand

Function: This phase verifies that the internal registers of the INTEL 80186 Direct Memory Access Controller (DMAC) are functioning properly.

Tests: Tests 1 through 14 — check the DMA0 Control register.

Tests 15 through 31 — check the DMA0 Terminal Count register.

Tests 32 through 48 — check the DMA0 Destination (low) register.

Tests 49 through 53 — check the DMA0 Destination (high) register.

Tests 54 through 70 — check the DMA0 Source (low) register.

Tests 71 through 75 — check the DMA0 Source (high) register.

Tests 76 through 90 — check the DMA1 Control register.

Tests 91 through 107 — check the DMA1 Terminal Count register.

Tests 108 through 124 — check the DMA1 Destination (low) register.

Tests 125 through 141 — check the DMA1 Source (low) register.

Time: 2 seconds

Warnings: None

Notes: If any test in this phase fails, the following hardware may be faulty:

- INTEL 80186 Microprocessor
- INTEL 80186 Microprocessor interface to the SCSI Address/Data bus.

Phase #6 Tests

Test Numbers:	1 through 14
Function:	These tests verify that the Direct Memory Access 0 (DMA0) Control register is functional.
Procedure:	A valid data pattern is written to the register, and it is verified with a read.
Hardware Tested:	The DMA0 Control register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	15 through 31
Function:	These tests verify that the DMA0 Terminal Count register is functional.
Procedure:	A valid data pattern is written to the register, and it is verified with a read.
Hardware Tested:	The DMA0 Terminal Count register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	32 through 48
Function:	These tests verify that the DMA0 Destination (low) register is functional.
Procedure:	A valid data pattern is written to the register, and it is verified with a read.
Hardware Tested:	The DMA0 Destination (low) register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	49 through 53
Function:	These tests verify that the DMA0 Destination (high) register is functional.
Procedure:	A valid data pattern is written to the register, and it is verified with a read.
Hardware Tested:	The DMA0 Destination (high) register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	54 through 70
Function:	These tests verify that the DMA0 Source (low) register is functional.
Procedure:	A valid data pattern is written to the register, and it is verified with a read.
Hardware Tested:	The DMA0 Source (low) register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

Test Numbers:	71 through 75
Function:	These tests verify that the DMA0 Source (high) register is functional.
Procedure:	A valid data pattern is written to the register, and it is verified with a read.
Hardware Tested:	The DMA0 Source (high) register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	76 through 90
Function:	These tests verify that the DMA1 Control register is functional.
Procedure:	A valid data pattern is written to the register, and it is verified with a read.
Hardware Tested:	The DMA1 Control register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	91 through 107
Function:	These tests verify that the DMA1 Terminal Count register is functional.
Procedure:	A valid data pattern is written to the register, and it is verified with a read.
Hardware Tested:	The DMA1 Terminal Count register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	108 through 124
Function:	These tests verify that the DMA1 Destination (low) register is functional.
Procedure:	A valid data pattern is written to the register, and it is verified with a read.
Hardware Tested:	The DMA1 Destination (low) register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	125 through 141
Function:	These tests verify that the DMA1 Source (low) register is functional.
Procedure:	A valid data pattern is written to the register, and it is verified with a read.
Hardware Tested:	The DMA1 Source (low) register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

Phase #7 — SCSI CPU Timer

Phase Name: SCSI CPU Timer Test

Type: Demand

Function: This phase tests the operation of the internal Timer registers of the INTEL 80186 CPU.

Tests: Tests 1 through 17 — check the Timer0 Count register.

Tests 18 through 34 — check the Timer0 MCA register.

Tests 35 through 51 — check the Timer0 MCB register.

Tests 52 through 59 — check the Timer0 Mode register.

Tests 60 through 77 — check the Timer1 Count register.

Tests 78 through 94 — check the Timer1 MCA register.

Tests 95 through 111 — check the Timer1 MCB register.

Tests 112 through 120 — check the Timer1 Mode register.

Tests 121 through 137 — check the Timer2 Count register.

Tests 138 through 154 — check the Timer2 MCA register.

Tests 155 through 158 — check the Timer2 Mode register.

Time: 2 seconds

Warnings: None

Notes: If any test in this phase fails, the following hardware may be faulty:

- INTEL 80186 Microprocessor
- INTEL 80186 Microprocessor interface to the SCSI Address/Data bus.

Phase #7 Tests

Test Numbers: 1 through 17

Function: These tests verify that the Timer0 Count register is functional.

Procedure: A valid data pattern is written to the register, and it is verified with a read.

Hardware Tested: The Timer0 Count register is tested.

Data Returned: The test number that failed, the actual data, and the expected data are returned.

Test Numbers:	18 through 34
Function:	These tests verify that the Timer0 Maximum Count A (MCA) register is functional.
Procedure:	A valid data pattern is written to the register, and it is verified with a read.
Hardware Tested:	The Timer0 MCA register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	35 through 51
Function:	These tests verify that the Timer0 Maximum Count B (MCB) register is functional.
Procedure:	A valid data pattern is written to the register, and it is verified with a read.
Hardware Tested:	The Timer0 MCB register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	52 through 59
Function:	These tests verify that the Timer0 Mode register is functional.
Procedure:	A valid data pattern is written to the register, and it is verified with a read.
Hardware Tested:	The Timer0 Mode register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	60 through 77
Function:	These tests verify that the Timer1 Count register is functional.
Procedure:	A valid data pattern is written to the register, and it is verified with a read.
Hardware Tested:	The Timer1 Count register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	78 through 94
Function:	These tests verify that the Timer1 MCA register is functional.
Procedure:	A valid data pattern is written to the register, and it is verified with a read.
Hardware Tested:	The Timer1 MCA register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	95 through 111
Function:	These tests verify that the Timer1 MCB register is functional.
Procedure:	A valid data pattern is written to the register, and it is verified with a read.
Hardware Tested:	The Timer1 MCB register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	112 through 120
Function:	These tests verify that the Timer1 Mode register is functional.
Procedure:	A valid data pattern is written to the register, and it is verified with a read.
Hardware Tested:	The Timer1 Mode register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	121 through 137
Function:	These tests verify that the Timer2 Count register is functional.
Procedure:	A valid data pattern is written to the register, and it is verified with a read.
Hardware Tested:	The Timer2 Count register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	138 through 154
Function:	These tests verify that the Timer2 MCA register is functional.
Procedure:	A valid data pattern is written to the register, and it is verified with a read.
Hardware Tested:	The Timer2 MCA register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	155 through 158
Function:	These tests verify that the Timer2 Mode register is functional.
Procedure:	A valid data pattern is written to the register, and it is verified with a read.
Hardware Tested:	The Timer2 Mode register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

Phase #8 — SCSI CPU Interrupt Controller

Phase Name: SCSI CPU Interrupt Controller

Type: Demand

Function: This phase verifies that the internal registers of the INTEL 80186 CPU Interrupt Controller (IC) are functioning properly.

Tests: Tests 1 through 8 — check the IC In-service register.

Tests 9 through 11 — check the Interrupt Request register.

Tests 12 through 20 — check the Interrupt Mask register.

Tests 21 through 23 — check the Interrupt Priority Mask register.

Tests 24 through 28 — check the Interrupt Status register.

Tests 29 through 40 — check the DMA0 and DMA1 Control registers.

Tests 41 through 56 — check the INT0 and INT1 Control registers.

Tests 57 through 69 — check the INT2 and INT3 Control registers.

Time: 2 seconds

Warnings: None

Notes: If any test in this phase fails, the following hardware may be faulty:

- INTEL 80186 Microprocessor
- INTEL 80186 Microprocessor interface to the SCSI Address/Data bus.

Phase #8 Tests

Test Numbers: 1 through 8

Function: These tests verify that the IC In-service register is functional.

Procedure: A valid data pattern is written to the register, and it is verified with a read.

Hardware Tested: The IC In-service register is tested.

Data Returned: The test number that failed, the actual data, and the expected data are returned.

Test Numbers:	9 through 11
Function:	These tests verify that the Interrupt Request register is functional.
Procedure:	A valid data pattern is written to the register, and it is verified with a read.
Hardware Tested:	The Interrupt Request register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	12 through 20
Function:	These tests verify that the Interrupt Mask register is functional.
Procedure:	A valid data pattern is written to the register, and it is verified with a read.
Hardware Tested:	The Interrupt Mask register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	21 through 23
Function:	These tests verify that the Interrupt Priority Mask register is functional.
Procedure:	A valid data pattern is written to the register, and it is verified with a read.
Hardware Tested:	The Interrupt Priority Mask register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	24 through 28
Function:	These tests verify that the Interrupt Status register is functional.
Procedure:	A valid data pattern is written to the register, and it is verified with a read.
Hardware Tested:	The Interrupt Status register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	29 through 40
Function:	These tests verify that the Direct Memory Access 0 (DMA0) and DMA1 Control registers are functional.
Procedure:	Valid data patterns are written to the registers, and they are verified with a read.
Hardware Tested:	The DMA0 and DMA1 Control registers are tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

Test Numbers:	41 through 56
Function:	These tests verify that the Interrupt 0 (INT0) and INT1 Control registers are functional.
Procedure:	Valid data patterns are written to the registers, and they are verified with a read.
Hardware Tested:	The INT0 and INT1 Control registers are tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Numbers:	57 through 69
Function:	These tests verify that the INT2 and INT3 Control registers are functional.
Procedure:	Valid data patterns are written to the registers, and they are verified with a read.
Hardware Tested:	The INT2 and INT3 Control registers are tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

Phase #9 — SCSI CPU Lower Chip Select

Phase Name:	SCSI CPU Lower Chip Select
Type:	Demand
Function:	This phase verifies that the Lower Memory Chip Select (LMCS) register of the INTEL 80186 CPU is functioning properly.
Tests:	Tests 1 through 12 — check the LMCS register.
Time:	2 seconds
Warnings:	None
Notes:	If any test in this phase fails, the following hardware may be faulty: ■ INTEL 80186 Microprocessor ■ INTEL 80186 Microprocessor interface to the SCSI bus.

Phase #9 Tests

Test Numbers:	1 through 12
Function:	These tests verify that the LMCS register is functional.
Procedure:	A valid data pattern is written to the register, and it is verified with a read.
Hardware Tested:	The LMCS register is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

Phase #10 — SCSI Programmed Input/Output (PIO) Byte Transfer

Phase Name: SCSI PIO Byte Transfer

Type: Demand

Function: This phase tests the I/O Address/Data bus.

Tests: Test 1 — tests the data pattern 0x01 at every address in a page of SBD DPDRAM.

Test 2 — tests the data pattern 0x02 at every address in a page of SBD DPDRAM.

Test 3 — tests the data pattern 0x04 at every address in a page of SBD DPDRAM.

Test 4 — tests the data pattern 0x08 at every address in a page of SBD DPDRAM.

Test 5 — tests the data pattern 0x10 at every address in a page of SBD DPDRAM.

Test 6 — tests the data pattern 0x20 at every address in a page of SBD DPDRAM.

Test 7 — tests the data pattern 0x40 at every address in a page of SBD DPDRAM.

Test 8 — tests the data pattern 0x80 at every address in a page of SBD DPDRAM.

Time: 45 seconds

Warnings: None

Notes: This phase performs PIO (write and read) in bytes. If any test in this phase fails, the following hardware may be faulty:

- INTEL 80186 Microprocessor
- INTEL 80186 Microprocessor interface to the SCSI Address/Data bus
- The SCSI interface to the 3B2 computer I/O bus.

Phase #10 Tests

Test Numbers: 1 through 8

Function: These tests verify the operation of the I/O Address/Data bus.

Procedure: A one is walked through a field of zeros at every address of a page of System Board (SBD) Dual Port Dynamic Random Access Memory (DPDRAM) and verified with a read.

Hardware Tested: The SCSI interface to the 3B2 computer I/O bus is tested.

Data Returned: A byte value read from the SBD DPDRAM, a short value read from SBD DPDRAM, and the SBD DPDRAM failing address are returned.

Phase #11 — SCSI Programmed Input/Output (PIO) Word Transfer

Phase Name: SCSI PIO Word Transfer

Type: Demand

Function: This phase tests the SCSI interface to the 3B2 computer I/O bus.

Tests: Test 1 — tests the data pattern 0x0001 at every address in a page of SBD DPDRAM.

Test 2 — tests the data pattern 0x0002 at every address in a page of SBD DPDRAM.

Test 3 — tests the data pattern 0x0004 at every address in a page of SBD DPDRAM.

Test 4 — tests the data pattern 0x0008 at every address in a page of SBD DPDRAM.

Test 5 — tests the data pattern 0x0010 at every address in a page of SBD DPDRAM.

Test 6 — tests the data pattern 0x0020 at every address in a page of SBD DPDRAM.

Test 7 — tests the data pattern 0x0040 at every address in a page of SBD DPDRAM.

Test 8 — tests the data pattern 0x0080 at every address in a page of SBD DPDRAM.

Test 9 — tests the data pattern 0x0100 at every address in a page of SBD DPDRAM.

Test 10 — tests the data pattern 0x0200 at every address in a page of SBD DPDRAM.

Test 11 — tests the data pattern 0x0400 at every address in a page of SBD DPDRAM.

Test 12 — tests the data pattern 0x0800 at every address in a page of SBD DPDRAM.

Test 13 — tests the data pattern 0x1000 at every address in a page of SBD DPDRAM.

Test 14 — tests the data pattern 0x2000 at every address in a page of SBD DPDRAM.

Test 15 — tests the data pattern 0x4000 at every address in a page of SBD DPDRAM.

Test 16 — tests the data pattern 0x8000 at every address in a page of SBD DPDRAM.

Time: 60 seconds

Warnings: None

Notes: This phase performs PIO (write and read) in words. If any test in this phase fails, the following hardware may be faulty:

- INTEL 80186 Microprocessor
- INTEL 80186 Microprocessor interface to the SCSI Address/Data bus
- SCSI interface to the 3B2 computer I/O bus.

Phase #11 Tests

Test Numbers: 1 through 16

Function: These tests verify the operation of the I/O bus.

Procedure: A one is walked through a field of zeros at every address of a page of System Board (SBD) Dual Port Dynamic Random Access Memory (DPDRAM), and it is verified with a read.

Hardware Tested: The SCSI interface to the 3B2 computer I/O bus is tested.

Data Returned: The number of the test that failed, the word value read from the SBD DPDRAM, and the SBD DPDRAM failing address are returned.

Phase #12 — SCSI DMA Byte Transfer

Phase Name: SCSI DMA Byte Transfer

Type: Demand

Function: This phase diagnoses and reports any errors in the operation of the Direct Memory Access (DMA) channels (byte width).

Tests: Test 1 — checks DMA0 from the SBD DPDRAM to the SCSI RAM.

Test 2 — checks DMA0 from the SCSI RAM to the SBD DPDRAM.

Time: 2 seconds

Warnings: None

Notes: This phase performs DMA transfers in bytes. If any test in this phase fails, the following hardware may be faulty:

- INTEL 80186 Microprocessor
- INTEL 80186 Microprocessor interface to the SCSI Address/Data bus
- The SCSI interface to the 3B2 computer I/O bus.

Phase #12 Tests

Test Number: 1

Function: This test verifies that Direct Memory Access Channel 0 (DMA0) from the System Board (SBD) Dual Port Dynamic Random Access Memory (DPDRAM) to the SCSI RAM is functional.

Procedure: Data is written [Programmed Input/Output (PIO)] to the SBD DPDRAM. That data is transferred to the SCSI RAM and compared for integrity.

Hardware Tested: The SCSI DMA0 (INTEL 80186) is tested.

Data Returned: The test number that failed, the actual data, and the expected data are returned.

==========

Test Number: 2

Function: This test verifies that DMA0 from the SCSI RAM to the SBD DPDRAM is functional.

Procedure: Data is written to the SCSI RAM. That data is transferred to the SBD DPDRAM and compared for integrity.

Hardware Tested: The SCSI DMA0 (INTEL 80186) is tested.

Data Returned: The test number that failed, the actual data, and the expected data are returned.

Phase #13 — SCSI DMA Word Transfer

Phase Name: SCSI DMA Word Transfer (dmawrd)

Type: Demand

Function: This phase diagnoses and reports any errors in the operation of the SCSI Direct Memory Access (DMA) channels (word width).

Tests: Test 1 — checks DMA0 from the SBD DPDRAM to the SCSI RAM.

Test 2 — checks DMA0 from the SCSI RAM to the SBD DPDRAM.

Time: 2 seconds

Warnings: None

Notes: This phase performs DMA transfers in words (16 bits). If any test in this phase fails, the following hardware may be faulty:

- INTEL 80186 Microprocessor
- INTEL 80186 Microprocessor interface to the SCSI Address/Data bus
- The SCSI interface to the 3B2 computer I/O bus.

Phase #13 Tests

Test Number: 1

Function: This test verifies that Direct Memory Access Channel 0 (DMA0) from the System Board (SBD) Dual Port Dynamic Random Access Memory (DPDRAM) to the SCSI RAM is functional.

Procedure: Data is written [Programmed Input/Output (PIO)] to the SBD. That data is transferred to the SCSI RAM and verified for data integrity.

Hardware Tested: The INTEL 80186 DMA0 is tested.

Data Returned: The test number that failed, the actual data, and the expected data are returned.

==========

Test Number: 2

Function: This test verifies that DMA0 from the SCSI RAM to the SBD DPDRAM is functional.

Procedure: Data is written to the SCSI RAM. That data is transferred to the SBD DPDRAM and verified for integrity.

Hardware Tested: The SCSI DMA0 (INTEL 80186) is tested.

Data Returned: The test number that failed, the actual data, and the expected data are returned.

Phase #14 — SCSI FIFO

Phase Name: SCSI FIFO Phase

Type: Demand

Function: This phase checks the four First-In-First-Out (FIFO) devices on the Host Adapter.

Tests: Test 1 — tests for computer-to-SCSI FIFO (a unique pattern to each memory location in FIFO).

Test 2 — pattern tests for computer-to-SCSI FIFO (0x0).

Test 3 — pattern tests for computer-to-SCSI FIFO (0xFFFF).

Test 4 — pattern tests for computer-to-SCSI FIFO (0x5555).

Test 5 — pattern tests for computer-to-SCSI FIFO (0xAAAA).

Test 6 — pattern tests for computer-to-SCSI FIFO (0xA5A5).

Test 7 — pattern tests for computer-to-SCSI FIFO (unique pattern to each memory location in FIFO).

Test 8 — pattern tests for SCSI-to-computer FIFO (0x0).

Test 9 — pattern tests for SCSI-to-computer FIFO (0xFFFF).

Test 10 — pattern tests for SCSI-to-computer FIFO (0x5555).

Test 11 — pattern tests for SCSI-to-computer FIFO (0xAAAA).

Test 12 — pattern tests for SCSI-to-computer FIFO (0xA5A5).

Test 13 — verifies that FIFO related PCSR bits are correct on reset.

Test 14 — verifies that the FIFO Full flags (in PCSR) stay inactive for all not-full conditions.

Test 15 — checks the FIFO Full flag.

Test 16 — checks SCSI-to-computer FIFO for empty flags.

Test 17 — checks SCSI-to-computer FIFO for full condition.

Time: 5 seconds

Warnings: None

Notes: None

Phase #14 Tests

Test Numbers: 1 through 12

Function: These tests perform pattern checks on the FIFO device.

Procedure: Bit patterns are written to and read from the FIFO device.

Hardware Tested: The FIFO device is tested.

Data Returned: The test number that failed, the actual data, and the expected data are returned.

==========

Test Number: 13

Function: This test verifies that FIFO related Peripheral Control and Status Register (PCSR) bits are correct on reset.

Procedure: The Host Adapter and computer FIFO device are reset. The full and empty PCSR bits are verified.

Hardware Tested: The FIFO devices are tested.

Data Returned: The test number that failed, the actual data, and the expected data are returned.

==========

Test Number: 14

Function: This test verifies that FIFO Full flags (in PCSR) stay inactive for all not-full conditions.

Procedure: Bytes are added to the computer-to-SCSI FIFO device until the device reaches full minus 1 byte capacity. As each byte is inserted, the FIFO Full flag is verified not to be set.

Hardware Tested: The FIFO device is tested.

Data Returned: The test number that failed, the actual data, and the expected data are returned.

==========

Test Number: 15

Function: This test checks the FIFO Full flag.

Procedure: The computer-to-SCSI FIFO devices are filled, and the FIFO Full flag is set.

Hardware Tested: The FIFO Full flag is tested.

Data Returned: The test number that failed, the actual data, and the expected data are returned.

Test Number:	16
Function:	This test verifies SCSI-to-computer FIFO device for empty flags.
Procedure:	When the SCSI-to-computer FIFO is empty, the PCSR flag indicates an empty condition.
Hardware Tested:	The FIFO Full flag is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Number:	17
Function:	This test verifies SCSI-to-computer FIFO device for full condition.
Procedure:	When the SCSI-to-computer FIFO device is full, the PCSR flag indicates a full condition.
Hardware Tested:	The FIFO Full flag is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

Phase #15 — SCSI External DMA Word

Phase Name:	SCSI External DMA Word
Type:	Demand
Function:	This phase verifies the 8237 Direct Memory Access Controller (DMAC) chip.
Tests:	Tests 1 and 2—check the SBD to the Host Adapter interface.
Time:	80 seconds
Warnings:	None
Notes:	Both of these tests use 1-kilobyte transfers. During the phase, these two tests are repeated 400 times to help locate possible intermittent errors.

Phase #15 Tests

Test Number:	1
Function:	This test verifies that Direct Memory Access Channel 1 (DMA1) from the System Board (SBD) to the Host Adapter First-In-First-Out (FIFO) register is functional.
Procedure:	A bit pattern is written to and read from the FIFO register.
Hardware Tested:	The 8237 DMAC is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Number:	2
Function:	This test verifies that the DMA0 from the Host Adapter FIFO register to the SBD memory is functional.
Procedure:	A bit pattern is written to and read from the SBD memory.
Hardware Tested:	The 8237 DMAC is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

Phase #16 — SCSI Controller

Phase Name:	SCSI Controller
Type:	Demand
Function:	This phase verifies that the SCSI Controller chip is properly functioning.
Tests:	Test 1 — checks the Command register for the correct reset value. Test 2 — checks the Control register for the correct reset value. Test 3 — checks the Destination register for the correct reset value. Test 4 — checks the Auxiliary Status register for the correct reset value. Test 5 — checks the ID register for the correct reset value. Test 6 — checks the Interrupt register for the correct reset value. Test 7 — checks the Source register for the correct reset value. Test 8 — checks the Diagnostic register for the correct reset value. Test 9 — checks the Transfer Count register for the correct reset value. Test 10 — checks the Transfer Count (msb) register for the correct reset value. Test 11 — checks the Transfer Count (lsb) register for the correct reset value. Test 12 — verifies the diagnostic loopback command to verify data paths on the chip.
Time:	3 seconds
Warnings:	None
Notes:	None

Phase #16 Tests

Test Number:	1
Function:	This test checks the Command register for reset.
Procedure:	Write a bit pattern into and read from the Command register.
Hardware Tested:	The SCSI Controller chip is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

Test Number: 2
Function: This test checks the Control register for reset.
Procedure: Write a bit pattern into and read from the Control register.
Hardware Tested: The SCSI Controller chip is tested.
Data Returned: The test number that failed, the actual data, and the expected data are returned.

==========

Test Number: 3
Function: This test checks the Destination register for reset.
Procedure: Write a bit pattern into and read from the Destination register.
Hardware Tested: The SCSI Controller chip is tested.
Data Returned: The test number that failed, the actual data, and the expected data are returned.

==========

Test Number: 4
Function: This test checks the Auxiliary Status register for reset.
Procedure: Write a bit pattern into and read from the Auxiliary Status register.
Hardware Tested: The SCSI Controller chip is tested.
Data Returned: The test number that failed, the actual data, and the expected data are returned.

==========

Test Number: 5
Function: This test checks the Identification (ID) register for reset.
Procedure: Write a bit pattern into and read from the ID register.
Hardware Tested: The SCSI Controller chip is tested.
Data Returned: The test number that failed, the actual data, and the expected data are returned.

==========

Test Number: 6
Function: This test checks the Interrupt register for reset.
Procedure: Write a bit pattern into and read from the Interrupt register.
Hardware Tested: The SCSI Controller chip is tested.
Data Returned: The test number that failed, the actual data, and the expected data are returned.

Test Number:	7
Function:	This test checks the Source register for reset.
Procedure:	Write a bit pattern into and read from the Source register.
Hardware Tested:	The SCSI Controller chip is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Number:	8
Function:	This test checks the Diagnostic register for reset.
Procedure:	Write a bit pattern into and read from the Diagnostic register.
Hardware Tested:	The SCSI Controller chip is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Number:	9
Function:	This test checks the Transfer Count register for reset.
Procedure:	Write a bit pattern into and read from the Transfer Count register.
Hardware Tested:	The SCSI Controller chip is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Number:	10
Function:	This test checks the Transfer Count [most significant byte (msb)] register for reset.
Procedure:	Write a bit pattern into and read from the Transfer Count (msb) register.
Hardware Tested:	The SCSI Controller chip is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

Test Number:	11
Function:	This test checks the Transfer Count [least significant byte (lsb)] register for reset.
Procedure:	Write a bit pattern into and read from the Transfer Count (lsb) register.
Hardware Tested:	The SCSI Controller chip is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Number:	12
Function:	This test verifies data paths on the chip using diagnostic loopback commands.
Procedure:	Diagnostic commands are sent to the SCSI Controller. The SCSI chip loops the data in the commands throughout the internal registers. The data is echoed back to the INTEL 80186 Microprocessor. A count pattern of 1 to 256 is used.
Hardware Tested:	The SCSI Controller chip is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

Phase #17 — SCSI Reset Test

Phase Name: SCSI Reset Phase Test

Type: Demand

Function: This phase resets the SCSI bus, causing target controllers to initiate reset diagnostics. Basic commands are sent to the target controllers to establish sanity and identification.

Tests: Test 1 — checks for incorrect cabling and termination.

Test 2 — checks for a minimum of at least one target controller.

Test 3 — verifies that the Host Adapter ID is different from all other target controller IDs.

Test 4 — verifies that at least one LU is associated with each target controller.

Time: 60 seconds

Warnings: The configuration of the SCSI bus is displayed during this phase. You must verify that the display is correct. The diagnostics only exercise what the system believes is associated with the SCSI system.

Notes: None

Phase #17 Tests

Test Number: 1

Function: This test checks for incorrect cabling and termination.

Procedure: Clear the SCSI Bus Reset latch.

Hardware Tested: The cabling and bus terminator are tested.

Data Returned: If this test fails, the message, "Reset Error: possible SCSI bus cable or termination problem" is displayed on the system console.

==========

Test Number: 2

Function: This test verifies that the Host Adapter Identification (ID) is different from that of all other target controllers.

Procedure: A command is sent to the Host Adapter.

Hardware Tested: All of the target controllers on the bus are tested.

Data Returned: If this test fails, the message "Error: Host Adapter Target ID match at ID xx" is displayed on the system console.

Test Number:	3
Function:	This test checks for a minimum of one target controller.
Procedure:	Commands are sent using all possible bus IDs, one of which must be present.
Hardware Tested:	The SCSI bus and target controllers are tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Number:	4
Function:	This test verifies that at least one Logic Unit (LU) is associated with a target controller.
Procedure:	Commands are sent to all possible bus IDs, one of which must be present.
Hardware Tested:	The target controller and LU are tested.
Data Returned:	If the test fails, the message "Error: Target ID xx has no LUs" is displayed on the system console.

Phase #18 — SCSI Controller Buffer Test

Phase Name: SCSI Controller Buffer Test

Type: Demand

Function: This phase resets the SCSI bus, causing the target controllers to initiate any reset diagnostics. Basic commands are sent to the target controllers to establish sanity and identification. Target controllers are pattern tested.

Tests: Test 1 — checks for incorrect cabling and termination.

Test 2 — checks for a minimum of at least one target controller.

Test 3 — verifies that the Host Adapter ID is different from all target controller IDs.

Test 4 — verifies that at least one LU is associated with a target controller.

Test 5 — writes and reads pattern tests of target controller buffers.

Time: 240 seconds

Warnings: The configuration of the SCSI bus is displayed during this phase. You must verify that the display is correct. The diagnostics only exercise what the system believes is associated with the SCSI system.

Notes: None

Phase #18 Tests

Test Number: 1

Function: This test checks for incorrect cabling and termination.

Procedure: Clear the SCSI Bus Reset latch.

Hardware Tested: The cabling and bus terminator are tested.

Data Returned: If this test fails, the message "Reset Error: possible SCSI bus cable or termination problem" is displayed on the system console.

==========

Test Number: 2

Function: This test verifies that the Host Adapter Identification (ID) is different from that of all target controllers.

Procedure: A command is sent to the Host Adapter.

Hardware Tested: All of the target controllers on the bus are tested.

Data Returned: If this test fails, the message "Error: Host Adapter Target ID match at ID xx" is displayed on the system console.

Test Number:	3
Function:	This test checks for a minimum of one target controller.
Procedure:	Commands are sent using all possible bus IDs, one of which must be present.
Hardware Tested:	The SCSI bus and target controllers are tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Number:	4
Function:	This test verifies that at least one Logic Unit (LU) is associated with a target controller.
Procedure:	Commands are sent using all possible bus IDs, one of which must be present.
Hardware Tested:	The target controller and LU are tested.
Data Returned:	If this test fails, the message "Error: Target ID xx has no LUs" is displayed on the system console.

==========

Test Number:	5
Function:	This test writes multiple patterns to and reads them from the buffer memory on the target controller.
Procedure:	Data is written to and read from the target controller.
Hardware Tested:	The SCSI bus and target controllers are tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

Phase #19 — SCSI Controller Self-Test

Phase Name: SCSI Controller Self-Test

Type: Demand

Function: This phase resets the SCSI bus, causing the target controllers to initiate any reset diagnostics. Basic commands are sent to the target controllers to establish sanity and identification.

Tests: Test 1 — checks for incorrect cabling and termination.

Test 2 — checks for a minimum of at least one target controller.

Test 3 — verifies that the Host Adapter ID is different from all target controller IDs.

Test 4 — verifies that at least one LU is associated with a target controller.

Test 5 — verify target controller self-test commands.

Time: 240 seconds

Warnings: The configuration of the SCSI bus is displayed during this phase. You must verify that the display is correct. The diagnostics only exercise what the system believes is associated with the SCSI system.

Notes: None

Phase #19 Tests

Test Number: 1

Function: This test checks for incorrect cabling and termination.

Procedure: Clear the SCSI Bus Reset latch.

Hardware Tested: The cabling and bus terminator are tested.

Data Returned: If this test fails, the message "Reset Error: possible SCSI bus cable or termination problem" is displayed on the system console.

==========

Test Number: 2

Function: This test verifies that the Host Adapter Identification (ID) is different from that of all target controllers.

Procedure: A command is sent to the Host Adapter.

Hardware Tested: All of the target controllers on the bus are tested.

Data Returned: If this test fails, the message "Error: Host Adapter Target ID match at ID xx" is displayed on the system console.

Test Number:	3
Function:	This test checks for a minimum of one target controller.
Procedure:	Commands are sent using all possible bus IDs, one of which must be present.
Hardware Tested:	The SCSI bus and target controllers are tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Number:	4
Function:	This test verifies that at least one Logic Unit (LU) is associated with a target controller.
Procedure:	Commands are sent using all possible bus IDs, one of which must be present.
Hardware Tested:	The target controller and LU are tested.
Data Returned:	If this test fails, the message "Error: Target ID xx has no LUs" is displayed on the system console.

==========

Test Number:	5
Function:	This test checks the target controller hardware.
Procedure:	Self-test commands are sent to each target controller.
Hardware Tested:	The target controller is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

Phase #20 — SCSI Controller/Media Self-Test

Phase Name: SCSI Controller/Media Self-Test

Type: Demand

Function: This phase resets the SCSI bus, causing the target controllers to initiate any reset diagnostics. Basic commands are sent to the target controllers to establish sanity and identification.

Tests: Test 1 — checks for incorrect cabling and termination.

Test 2 — checks for a minimum of at least one target controller.

Test 3 — verifies that the Host Adapter ID is different from all target controller IDs.

Test 4 — verifies that at least one LU is associated with a target controller.

Test 5 — verify target controller self-test commands.

Time: 240 seconds

Warnings: The configuration of the SCSI bus is displayed during this phase. You must verify that the display is correct. The diagnostics only exercise what the system believes is associated with the SCSI system.

Notes: This phase is not run on the tape units.

Phase #20 Tests

Test Number: 1

Function: This test checks for incorrect cabling and termination.

Procedure: Clear the SCSI Bus Reset latch.

Hardware Tested: The cabling and bus terminator are tested.

Data Returned: If this test fails, the message "Reset Error: possible SCSI bus cable or termination problem" is displayed on the system console.

==========

Test Number: 2

Function: This test verifies that the Host Adapter Identification (ID) is different from that of all target controllers.

Procedure: A command is sent to the Host Adapter.

Hardware Tested: All of the target controllers on the bus are tested.

Data Returned: If this test fails, the message "Error: Host Adapter Target ID match at ID xx" is displayed on the system console.

Test Number:	3
Function:	This test checks for a minimum of one target controller.
Procedure:	Commands are sent using all possible bus IDs, one of which must be present.
Hardware Tested:	The SCSI bus and target controllers are tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Number:	4
Function:	This test verifies that at least one Logic Unit (LU) is associated with a target controller.
Procedure:	Commands are sent using all possible bus IDs, one of which must be present.
Hardware Tested:	The target controller and LU are tested.
Data Returned:	If this test fails, the message "Error: Target ID xx has no LUs" is displayed on the system console.

==========

Test Number:	5
Function:	This test checks the target controller and media.
Procedure:	The target controller performs a self-test with the media.
Hardware Tested:	The target controller and LU are tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

Phase #21 — SCSI Interactive Tape Test

Phase Name: SCSI Interactive Tape Test

Type: Interactive

Function: This phase resets the SCSI bus, causing the target controllers to initiate any reset diagnostics. Basic commands are sent to the target controllers to establish sanity and identification.

Tests: Test 1 — checks for incorrect cabling and termination.

Test 2 — checks for a minimum of at least one target controller.

Test 3 — verifies that the Host Adapter ID is different from all target controller IDs.

Test 4 — verifies that at least one LU is associated with a target controller.

Test 5 — verifies that tape unit can be written to and read from.

Time: 180 seconds

Warnings: The configuration of the SCSI bus is displayed during this phase. You must verify that the display is correct. The diagnostics only exercise what the system believes is associated with the SCSI system.

Notes: None

Phase #21 Tests

Test Number: 1

Function: This test checks for incorrect cabling and termination.

Procedure: Clear the SCSI Bus Reset latch.

Hardware Tested: The cabling and bus terminator are tested.

Data Returned: If this test fails, the message "Reset Error: possible SCSI bus cable or termination problem" is displayed on the system console.

==========

Test Number: 2

Function: This test verifies that the Host Adapter Identification (ID) is different from that of all target controllers.

Procedure: A command is sent to the Host Adapter.

Hardware Tested: All of the target controllers on the bus are tested.

Data Returned: If this test fails, the message "Error: Host Adapter Target ID match at ID xx" is displayed on the system console.

Test Number:	3
Function:	This test checks for a minimum of one target controller.
Procedure:	Commands are sent using all possible bus IDs, one of which must be present.
Hardware Tested:	The SCSI bus and target controllers are tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Number:	4
Function:	This test verifies that at least one Logic Unit (LU) is associated with a target controller.
Procedure:	Commands are sent using all possible bus IDs, one of which must be present.
Hardware Tested:	The target controller and LU are tested.
Data Returned:	If this test fails, the message "Error: Target ID xx has no LUs" is displayed on the system console.

==========

Test Number:	5
Function:	This test writes multiple patterns to and reads them from selected tape units.
Procedure:	Sequential data blocks are written to and read from the tape unit.
Hardware Tested:	The tape unit is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

Phase #22 — SCSI Disk Write/Read Test

Phase Name: SCSI Disk Write/Read Test

Type: Demand

Function: This phase resets the SCSI bus, causing the target controllers to initiate any reset diagnostics. Basic commands are sent to the target controllers to establish sanity and identification.

Tests: Test 1 — checks for incorrect cabling and termination.

Test 2 — checks for a minimum of at least one target controller.

Test 3 — verifies that the Host Adapter ID is different from all target controller IDs.

Test 4 — verifies that at least one LU is associated with a target controller.

Test 5 — verifies write/read access of the SCSI disks.

Time: 60 seconds

Warnings: The configuration of the SCSI bus is displayed during this phase. You must verify that the display is correct. The diagnostics only exercise what the system believes is associated with the SCSI system.

Notes: None

Phase #22 Tests

Test Number: 1

Function: This test checks for incorrect cabling and termination.

Procedure: Clear the SCSI Bus Reset latch.

Hardware Tested: The cabling and bus terminator are tested.

Data Returned: If this test fails, the message "Reset Error: possible SCSI bus cable or termination problem" is displayed on the system console.

==========

Test Number: 2

Function: This test verifies that the Host Adapter Identification (ID) is different from that of all target controllers.

Procedure: A command is sent to the Host Adapter.

Hardware Tested: All of the target controllers on the bus are tested.

Data Returned: If this test fails, the message "Error: Host Adapter Target ID match at ID xx" is displayed on the system console.

Test Number:	3
Function:	This test checks for a minimum of one target controller.
Procedure:	Commands are sent using all possible bus IDs, one of which must be present.
Hardware Tested:	The SCSI bus and target controllers are tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Number:	4
Function:	This test verifies that at least one Logic Unit (LU) is associated with a target controller.
Procedure:	Commands are sent using all possible bus IDs, one of which must be present.
Hardware Tested:	The target controller and LU are tested.
Data Returned:	If this test fails, the message "Error: Target ID xx has no LUs" is displayed on the system console.

==========

Test Number:	5
Function:	This test writes multiple patterns to and reads them from all SCSI disks on the system.
Procedure:	Write to and read from the diagnostic area of the disk.
Hardware Tested:	The disk controller and disk drive are tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

Phase #23 — SCSI Interactive Disk Write/Read

Phase Name:	SCSI Interactive Disk Write/Read
Type:	Interactive
Function:	This phase resets the SCSI bus, causing the target controllers to initiate any reset diagnostics. Basic commands are sent to the target controllers to establish sanity and identification.
Tests:	Test 1 — checks for incorrect cabling and termination. Test 2 — checks for a minimum of at least one target controller. Test 3 — verifies that the Host Adapter ID is different from all target controller IDs. Test 4 — verifies that at least one LU is associated with a target controller. Test 5 — verifies write/read access of selected SCSI disks.
Time:	240 seconds
Warnings:	The configuration of the SCSI bus is displayed during this phase. You must verify that the display is correct. The diagnostics only exercise what the system believes is associated with the SCSI system.
Notes:	None

Phase #23 Tests

Test Number:	1
Function:	This test checks for incorrect cabling and termination.
Procedure:	Clear the SCSI Bus Reset latch.
Hardware Tested:	The cabling and bus terminator are tested.
Data Returned:	If this test fails, the message "Reset Error: possible SCSI bus cable or termination problem" is displayed on the system console.

==========

Test Number:	2
Function:	This test verifies that the Host Adapter Identification (ID) is different from that of all target controllers.
Procedure:	A command is sent to the Host Adapter.
Hardware Tested:	All of the target controllers on the bus are tested.
Data Returned:	If this test fails, the message "Error: Host Adapter Target ID match at ID xx" is displayed on the system console.

Test Number:	3
Function:	This test checks for a minimum of one target controller.
Procedure:	Commands are sent using all possible bus IDs, one of which must be present.
Hardware Tested:	The SCSI bus and target controllers are tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Number:	4
Function:	This test verifies that at least one Logic Unit (LU) is associated with a target controller.
Procedure:	Commands are sent using all possible bus IDs, one of which must be present.
Hardware Tested:	The target controller and LU are tested.
Data Returned:	If this test fails, the message "Error: Target ID xx has no LUs" is displayed on the system console.

==========

Test Number:	5
Function:	This test writes multiple patterns to and reads them from selected disk devices.
Procedure:	Run extensive write and read testing on diagnostic areas of the selected disk.
Hardware Tested:	The disk controller and disk drive are tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

Phase #24 — SCSI Configuration Status

Phase Name:	Configuration Status
Type:	Demand
Function:	This phase evaluates the SCSI configuration generated by the system during boot.
Tests:	Test 1 — checks ledt sanity word. Test 2 — checks for a minimum of at least one target controller. Test 3 — verifies that at least one LU is associated with a target controller. Test 4 — verifies that the Host Adapter ID is different from all target controller IDs.
Time:	5 seconds
Warnings:	None
Notes:	None

Phase #24 Tests

Test Number:	1
Function:	This test checks ledt sanity word.
Procedure:	Valid configuration information is placed in the table.
Hardware Tested:	The Host Adapter is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Number:	2
Function:	This test checks for a minimum of one target controller.
Procedure:	Read the equipage table. Determine if target controller(s) is listed.
Hardware Tested:	The target controller(s) is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

==========

Test Number:	3
Function:	This test verifies that at least one Logic Unit (LU) is associated with a target controller.
Procedure:	Read the equipage table. Determine if LU(s) is listed.
Hardware Tested:	The target controllers and LUs are tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

Test Number:	4
Function:	This test verifies that the Host Adapter Identification (ID) is different from that of all target controllers.
Procedure:	Read the equipage table and verify that the Host Adapter has a unique ID.
Hardware Tested:	The target controllers are tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

Chapter 15: STARLAN Interface Diagnostics

Introduction

This chapter contains a description of the diagnostic phases and tests for the 3B2 computer STARLAN Local Area Network (LAN) interface. The card is referred to as the Network Access Unit (NAU) in software. If your 3B2 computer is equipped with a STARLAN interface, the Equipped Device Table (EDT) should list it. You can use the diagnostic monitor command "show" (DGMON> **s**) to display the EDT. If the STARLAN interface is not listed in the EDT and a VOID or NULL is listed, the STARLAN interface may be considered a suspect of faulty hardware. Twenty-one diagnostic phases test all aspects of STARLAN.

The Table of Contents listing on the previous page will help you locate the desired phase and test descriptions for the STARLAN interface. The phase and test descriptions are organized in numerical order.

Phase Descriptions

Phase #1 — Common I/O Sanity Test

Phase Name: Network Access Unit I/O SANITY (cio)

Type: Normal

Function: This phase checks the basic operation of the Network Access Unit — Common I/O (CIO) interface.

Test: Test 1 — verifies that the diagnostic results can be transferred to the System Board (SBD).

Time: 1 second

Warnings: None

Notes: This phase, in conjunction with Phase 18, is used to determine the basic sanity of the Network Access Unit. The following tests are used by all other diagnostic phases to execute their tests and return the results to the SBD. If the Network Access Unit cannot follow the procedure in this phase, it is considered insane. If this phase fails, the following hardware may be faulty:

- INTEL 80186 Microprocessor
- Network Access Unit Random Access Memory (RAM)/Read Only Memory (ROM)
- Network Access Unit interface to the 3B2 Input/Output (I/O) bus
- Network Access Unit Address/Data bus.

Phase #1 Test

Test Number: 1

Function: This test verifies that the CIO hardware and firmware are functioning properly.

Procedure: This test uses the following standard procedure:

1. The I/O slot number of the Network Access Unit in the 3B2 computer is determined.
2. The Network Access Unit is reset.
3. The system data block is initialized.
4. The Network Access Unit is initialized (sysgen) by sending express [Interrupt 0 (INT0)] and attention (INT1) interrupts sequentially.
5. The Network Access Unit test code (phase) is downloaded by using the CIO firmware command [Download Memory (DLM)].

6. Execution of the downloaded phase is started by using the CIO firmware command [Force Call to Function (FCF)].
7. A function call to "**phasend**( )" is made when the phase is complete. Phasend returns the test results to the SBD. The DGMON in turn displays the message, ATP (All Tests Passed) on the system console.

Hardware Tested: The Network Access Unit interface to the 3B2 I/O bus is tested.

Data Returned: None

Phase #2 — PCSR Write/Read Test

Phase Name:	Network Access Unit Control and Status Register (pcsr)
Type:	Demand
Function:	This phase diagnoses and reports any bit interdependency of the Network Access Unit Peripheral Control and Status Register (PCSR).
Tests:	Test 1 — clears and reads PCSR bit 0. Test 2 — clears and reads PCSR bit 1. Test 3 — clears and reads PCSR bit 3. Test 4 — clears and reads PCSR bit 4.
Time:	1 second
Warnings:	None
Notes:	None

Phase #2 Tests

Test Numbers:	1 through 4
Function:	These tests diagnose and report any bit interdependency of the PCSR.
Procedure:	Clear a bit and verify with a read.
Hardware Tested:	The Network Access Unit PCSR is tested.
Data Returned	The failed test number, the actual data, and the expected data are returned.

Phase #3 — Upper RAM Write/Read Test

Phase Name: Network Access Unit Upper RAM (ram_h)

Type: Demand

Function: This phase verifies the operation of the upper half of the Network Access Unit RAM.

Tests: Test 1 — writes 0x0000 to each memory location and verifies the write with a read.

Test 2 — writes 0xFFFF to each memory location and verifies the write with a read.

Test 3 — writes 0x5555 to each memory location and verifies the write with a read.

Test 4 — writes 0xAAAA to each memory location and verifies the write with a read.

Test 5 — walks a one through a field of zeros in every memory location and verifies the walk with a read.

Test 6 — walks a zero through a field of ones in every memory location and verifies the walk with a read.

Time: 10 seconds

Warnings: None

Notes: None

Phase #3 Tests

Test Numbers: 1 through 6

Function: These tests verify the operation of the upper half of the Network Access Unit RAM, addresses 0x4000 to 0x7FFF.

Procedure: These tests write a known data pattern to every memory location (upper half) and verify the write with a read.

Hardware Tested: The Network Access Unit RAM (upper half) is tested.

Data Returned The failed test number, the data pattern read, and the data pattern written are returned.

Phase #4 — Lower RAM Write/Read Test

Phase Name: Network Access Unit Lower RAM (ram_l)

Type: Demand

Function: This phase checks the operation of the lower half of the Network Access Unit RAM.

Tests: Test 1 — writes 0x0000 to each memory location and verifies the write with a read.

Test 2 — writes 0xFFFF to each memory location and verifies the write with a read.

Test 3 — writes 0x5555 to each memory location and verifies the write with a read.

Test 4 — writes 0xAAAA to each memory location and verifies the write with a read.

Test 5 — walks a one through a field of zeros in every memory location and verifies the walk with a read.

Test 6 — walks a zero through a field of ones in every memory location and verifies the walk with a read.

Time: 10 seconds

Warnings: None

Notes: None

Phase #4 Tests

Test Numbers: 1 through 6

Function: These tests verify the operation of the lower half of the Network Access Unit RAM, addresses 0x0000 to 0x3FFF.

Procedure: Write a known data pattern to every memory location (lower half) and verify the write with a read.

Hardware Tested: The Network Access Unit RAM (lower half) is tested.

Data Returned The failed test number, the data pattern read, and the data pattern written are returned.

Phase #5 — ROM Check Sum Test

Phase Name: Network Access Unit ROM (rom)

Type: Demand

Function: This phase verifies the integrity of the Network Access Unit ROM.

Test: This phase calculates a ROM check sum and compares it to the check sum stored in ROM when the ROM was initially programmed.

Time: 1 second

Warnings: None

Notes: Every Network Access Unit Erasable Programmable Read Only Memory (EPROM) set is initially loaded with the same data. This load contains all but 6 bytes of the data required to complete a Network Access Unit EPROM load. The six omitted bytes are those that contain the individual Network Access Unit address. Because the address of each Network Access Unit is unique, the data that is written into these six registers later is different for each Network Access Unit. A check sum is generated over the unchanging data and written into each Network Access Unit EPROM set. When generating this check sum, the generator finds the hexadecimal value FF in each of the three unwritten registers (0xFF is the value found in unprogrammed EPROM registers).

Prior to insertion in a Network Access Unit, each Network Access Unit EPROM set is loaded with the 6 address bytes that differentiate that Network Access Unit from all others.

To generate the same check sum as contained in the Network Access Unit EPROM, the diagnostics substitute 0xFF for the actual data contained in the six Network Access Unit Address registers.

Phase #5 Test

Test Number: 1

Function: This test verifies the integrity of the Network Access Unit ROM.

Procedure: Calculate a check sum over the Network Access Unit ROM and compare the calculated check sum with the check sum stored in ROM when it was initially programmed.

Hardware Tested: The Network Access Unit ROM is tested.

Data Returned The failed test number, the expected check sum value, and the actual check sum value are returned.

Phase #6 — CPU Chip Select Test

Phase Name: Network Access Unit CPU_1 (cpu_1)

Type: Demand

Function: This phase tests the operation of the Upper Memory Chip Select (UMCS), Peripheral Access Chip Select (PACS), and Middle Peripheral Chip Select (MPCS) registers of the INTEL 80186 CPU.

Tests: Tests 1 through 11 — check the UMCS register.

Tests 12 through 28 — check the PACS register.

Tests 29 through 40 — check the MPCS register.

Time: 1 second

Warnings: None

Notes: If any test in this phase fails, the following hardware may be faulty:

- The INTEL 80186 Microprocessor
- The INTEL 80186 Microprocessor interface to the Network Access Unit Address/Data bus.

Phase #6 Tests

Test Numbers: 1 through 11

Function: These tests verify that the UMCS register is functional.

Procedure: Write a valid data pattern to the register and verify with a read.

Hardware Tested: The UMCS register is tested.

Data Returned The failed test number, the actual value, and the expected value are returned.

==========

Test Numbers: 12 through 28

Function: These tests verify that the PACS register is functional.

Procedure: Write a valid data pattern to the register and verify with a read.

Hardware Tested: The PACS register is tested.

Data Returned The failed test number, the actual value, and the expected value are returned.

==========

Test Numbers: 29 through 40

Function: These tests verify that the MPCS register is functional.

Procedure: Write a valid data pattern to the register and verify with a read.

Hardware Tested: The MPCS register is tested.

Data Returned The failed test number, the actual value, and the expected value are returned.

Phase #7 — CPU DMA Internal Test

Phase Name: Network Access Unit CPU_2 (cpu_2)

Type: Demand

Function: This phase tests the following internal registers of the INTEL 80186 Microprocessor Direct Memory Access Controller (DMAC).

Tests: Tests 1 through 16 — check the DMA0 Terminal Count register.

Tests 17 through 33 — check the DMA0 Destination (low) register.

Tests 34 through 38 — check the DMA0 Destination (high) register.

Tests 39 through 56 — check the DMA0 Source (low) register.

Tests 57 through 61 — check the DMA0 Source (high) register.

Tests 62 through 78 — check the DMA1 Terminal Count register.

Tests 79 through 95 — check the DMA1 Destination (low) register.

Tests 96 through 112 — check the DMA1 Source (low) register.

Tests 113 through 143 — no tests run.

Time: 1 second

Warnings: None

Notes: If any test in this phase fails, the following hardware may be faulty:

- INTEL 80816 Microprocessor
- INTEL 80186 Microprocessor interface to the Network Access Unit Address/Data bus.

Phase #7 Tests

Test Numbers: 1 through 16

Function: These tests verify that the Direct Memory Access 0 (DMA0) Terminal Count register is functional.

Procedure: Write a valid data pattern to the register and verify with a read.

Hardware Tested: The DMA0 Terminal Count register is tested.

Data Returned The failed test number, the actual value, and the expected value are returned.

Test Numbers:	17 through 33
Function:	These tests verify that the DMA0 Destination (low) register is functional.
Hardware Tested:	The DMA0 Destination (low) register is tested.
Data Returned	The failed test number, the actual value, and the expected value are returned.

==========

Test Numbers:	34 through 38
Function:	These tests verify that the DMA0 Destination (high) register is functional.
Procedure:	Write a valid data pattern to the register and verify with a read.
Hardware Tested:	The DMA0 Destination (high) register is tested.
Data Returned	The failed test number, the actual value, and the expected value are returned.

==========

Test Numbers:	39 through 56
Function:	These tests verify that the DMA0 Source (low) register is functional.
Procedure:	Write a valid data pattern to the register and verify with a read.
Hardware Tested:	The DMA0 Source (low) register is tested.
Data Returned	The failed test number, the actual value, and the expected value are returned.

==========

Test Numbers:	57 through 61
Function:	These tests verify that the DMA0 Source (high) register is functional.
Procedure:	Write a valid data pattern to the register and verify with a read.
Hardware Tested:	The DMA0 Source (high) register is tested.
Data Returned	The failed test number, the actual value, and the expected value are returned.

==========

Test Numbers:	62 through 78
Function:	These tests verify that the DMA1 Terminal Count register is functional.
Procedure:	Write a valid data pattern to the register and verify with a read.
Hardware Tested:	The DMA1 Terminal Count register is tested.
Data Returned	The failed test number, the actual value, and the expected value are returned.

Test Numbers:	79 through 95
Function:	These tests verify that the DMA1 Destination (low) register is functional.
Procedure:	Write a valid data pattern to the register and verify with a read.
Hardware Tested:	The DMA1 Destination (low) register is tested.
Data Returned	The failed test number, the actual value, and the expected value are returned.

==========

Test Numbers:	96 through 112
Function:	These tests verify that the DMA1 Source (low) register is functional.
Procedure:	Write a valid data pattern to the register and verify with a read.
Hardware Tested:	The DMA1 Source (low) register is tested.
Data Returned	The failed test number, the actual value, and the expected value are returned.

Phase #8 — CPU Timer Test

Phase Name: Network Access Unit CPU_3 (cpu_3)

Type: Demand

Function: This phase tests the operation of the internal Timer registers of the INTEL 80186 CPU.

Tests: Tests 1 through 17 — check the Timer0 Count register.

Tests 18 through 34 — check the Timer0 MCA register.

Tests 35 through 51 — check the Timer0 MCB register.

Tests 52 through 59 — check the Timer0 Mode register.

Tests 60 through 77 — check the Timer1 Count register.

Tests 78 through 94 — check the Timer1 MCA register.

Tests 95 through 111 — check the Timer1 MCB register.

Tests 112 through 120 — check the Timer1 Mode register.

Tests 121 through 137 — check the Timer2 Count register.

Tests 138 through 154 — check the Timer2 MCA register.

Tests 155 through 158 — check the Timer2 Mode register.

Time: 1 second

Warnings: None

Notes: If any test in this phase fails, the following hardware may be faulty:

- INTEL 80186 Microprocessor
- INTEL 80186 Microprocessor interface to the Network Access Unit Address/Data bus.

Phase #8 Tests

Test Numbers: 1 through 17

Function: These tests verify that the Timer0 Count register is functional.

Procedure: Write a valid data pattern to the register and verify with a read.

Hardware Tested: The Timer0 Count register is tested.

Data Returned The failed test number, the actual value, and the expected value are returned.

Test Numbers:	18 through 34
Function:	These tests verify that the Timer0 Maximum Count A (MCA) register is functional.
Hardware Tested:	The Timer0 MCA register is tested.
Data Returned	The failed test number, the actual value, and the expected value are returned.

==========

Test Numbers:	35 through 51
Function:	These tests verify that the Timer0 Maximum Count B (MCB) register is functional.
Procedure:	Write a valid data pattern to the register and verify with a read.
Hardware Tested:	The Timer0 MCB register is tested.
Data Returned	The failed test number, the actual value, and the expected value are returned.

==========

Test Numbers:	52 through 59
Function:	These tests verify that the Timer0 Mode register is functional.
Procedure:	Write a valid data pattern to the register and verify with a read.
Hardware Tested:	The Timer0 Mode register is tested.
Data Returned	The failed test number, the actual value, and the expected value are returned.

==========

Test Numbers:	60 through 77
Function:	These tests verify that the Timer1 Count register is functional.
Procedure:	Write a valid data pattern to the register and verify with a read.
Hardware Tested:	The Timer1 Count register is tested.
Data Returned	The failed test number, the actual value, and the expected value are returned.

==========

Test Numbers:	78 through 94
Function:	These tests verify that the Timer1 MCA register is functional.
Procedure:	Write a valid data pattern to the register and verify with a read.
Hardware Tested:	The Timer1 MCA register is tested.
Data Returned:	The failed test number, the actual value, and the expected value are returned.

Test Numbers: 95 through 111
Function: These tests verify that the Timer1 MCB register is functional.
Procedure: Write a valid data pattern to the register and verify with a read.
Hardware Tested: The Timer1 MCB register is tested.
Data Returned: The failed test number, the actual value, and the expected value are returned.

==========

Test Numbers: 112 through 120
Function: These tests verify that the Timer1 Mode register is functional.
Procedure: Write a valid data pattern to the register and verify with a read.
Hardware Tested: The Timer1 Mode register is tested.
Data Returned: The failed test number, the actual value, and the expected value are returned.

==========

Test Numbers: 121 through 137
Function: These tests verify that the Timer2 Count register is functional.
Procedure: Write a valid data pattern to the register and verify with a read.
Hardware Tested: The Timer2 Count register is tested.
Data Returned: The failed test number, the actual value, and the expected value are returned.

==========

Test Numbers: 138 through 154
Function: These tests verify that the Timer2 MCA register is functional.
Procedure: Write a valid data pattern to the register and verify with a read.
Hardware Tested: The Timer2 MCA register is tested.
Data Returned: The failed test number, the actual value, and the expected value are returned.

==========

Test Numbers: 155 through 158
Function: These tests verify that the Timer2 Mode register is functional.
Procedure: Write a valid data pattern to the register and verify with a read.
Hardware Tested: The Timer2 Mode register is tested.
Data Returned: The failed test number, the actual value, and the expected value are returned.

Phase #9 — CPU Interrupt Controller

Phase Name: Network Access Unit CPU_4 (cpu_4)

Type: Demand

Function: This phase verifies that the internal registers of the INTEL 80186 CPU Interrupt Controller (IC) are functioning properly.

Tests: Tests 1 through 8 — check the IC In-service register.

Tests 9 through 11 — check the Interrupt Request register.

Tests 12 through 20 — check the Interrupt Mask register.

Tests 21 through 24 — check the Interrupt Priority Mask register.

Tests 25 through 36 — check the DMA0 and DMA1 Control registers.

Tests 37 through 45 — check the INT0 and INT1 Control registers.

Tests 46 through 51 — check the INT2 and INT3 Control registers.

Tests 52 through 69 — no tests run.

Time: 1 second

Warnings: None

Notes: If any test in this phase fails, the following hardware may be faulty:

- INTEL 80186 Microprocessor
- INTEL 80186 Microprocessor interface to the Network Access Unit Address/Data bus.

==========

Phase #9 Tests

Test Numbers: 1 through 8

Function: These tests verify that the IC In-service register is functional.

Procedure: Write a valid data pattern to the register and verify with a read.

Hardware Tested: The IC In-service register is tested.

Data Returned: The failed test number, the actual value, and the expected value are returned.

Test Numbers:	9 through 11
Function:	These tests verify that the Interrupt Request register is functional.
Procedure:	Write a valid data pattern to the register and verify with a read.
Hardware Tested:	The Interrupt Request register is tested.
Data Returned:	The failed test number, the actual value, and the expected value are returned.

==========

Test Numbers:	12 through 20
Function:	These tests verify that the Interrupt Mask register is functional.
Procedure:	Write a valid data pattern to the register and verify with a read.
Hardware Tested:	The Interrupt Mask register is tested.
Data Returned:	The failed test number, the actual value, and the expected value are returned.

==========

Test Numbers:	21 through 24
Function:	These tests verify that the Interrupt Priority Mask register is functional.
Procedure:	Write a valid data pattern to the register and verify with a read.
Hardware Tested:	The Interrupt Priority Mask register is tested.
Data Returned:	The failed test number, the actual value, and the expected value are returned.

==========

Test Numbers:	25 through 36
Function:	These tests verify that the Direct Memory Access 0 (DMA0) and DMA1 Control registers are functional.
Procedure:	Write a valid data pattern to the register and verify with a read.
Hardware Tested:	The DMA0 and DMA1 Control registers are tested.
Data Returned:	The failed test number, the actual value, and the expected value are returned.

Test Numbers:	37 through 45
Function:	These tests verify that the Interrupt 0 (INT0) and INT1 Control registers are functional.
Procedure:	Write a valid data pattern to the register and verify with a read.
Hardware Tested:	The INT0 and INT1 Control registers are tested.
Data Returned:	The failed test number, the actual value, and the expected value are returned.

==========

Test Numbers:	46 through 51
Function:	These tests verify that the INT2 and INT3 Control registers are functional.
Procedure:	Write a valid data pattern to the register and verify with a read.
Hardware Tested:	The INT2 and INT3 registers are tested.
Data Returned:	The failed test number, the actual value, and the expected value are returned.

Phase #10 — CPU Lower Chip Select Test

Phase Name:	Network Access Unit CPU_5 (cpu_5)
Type:	Demand
Function:	This phase tests the Lower Memory Chip Select (LMCS) register of the INTEL 80186 CPU.
Tests:	Tests 1 through 12 — check the LMCS register.
Time:	1 second
Warnings:	None
Notes:	If any test in this phase fails, the following hardware may be faulty: ■ INTEL 80186 Microprocessor ■ INTEL 80186 Microprocessor interface to the Network Access Unit Address/Data bus.

Phase #10 Tests

Test Numbers:	1 through 12
Function:	These tests verify that the LMCS register is functional.
Procedure:	Write a valid data pattern to the register and verify with a read.
Hardware Tested:	The LMCS register is tested.
Data Returned:	The failed test number, the actual value, and the expected value are returned.

Phase #11 — CPU Sanity Maze Test

Phase Name: Network Access Unit CPU_6 (cpu_6)

Type: Demand

Function: This phase tests the ability of the INTEL 80186 CPU to execute its instruction set.

Tests:

Test 1 — checks the jump, call, and RET instructions.

Test 2 — checks the conditional jumps and Flag Register bits.

Test 3 — checks the CMP instruction and Data bits.

Test 4 — checks registers (as source and destination) and word or byte field encoding.

Test 5 — checks Segment registers.

Test 6 — checks all addressing modes.

Test 7 — checks segment override prefix and address—object instructions.

Test 8 — checks memory write, xchg, and accumulator—memory move instructions.

Test 9 — verifies the push and pop commands.

Test 10 — checks the addressing line.

Test 11 — checks the OR, AND, and EXCLUSIVE OR test instructions.

Test 12 — checks the shift and rotate instructions.

Test 13 — checks the negative, add, subtract, increment, and decrement instructions.

Test 14 — checks the string instructions.

Test 15 — checks the unsigned and signed multiplication and division instructions.

Test 16 — checks the daa, das, aaa, aas, aam, aad, chw, and cwd instructions.

Time: 1 second

Warnings: None

Phase #11 Tests

Test Numbers:	1 through 16
Function:	These tests verify the integrity of the INTEL 80186 CPU.
Procedure:	Run CPU through "sanity maze" to test its ability to execute its instruction set.
Hardware Tested:	The INTEL 80186 CPU is tested.
Data Returned:	The failed test number, the actual value, and the expected value are returned.

Phase #12 — Programmed Input/Output (PIO) Byte

Phase Name: Network Access Unit Byte PIO (pio_1)

Type: Demand

Function: This phase checks the Network Access Unit interface to the 3B2 I/O bus.

Tests: Test 1— checks the data pattern 0x01 at every address in a page of SBD DPDRAM.

Test 2 — checks the data pattern 0x02 at every address in a page of SBD DPDRAM.

Test 3 — checks the data pattern 0x04 at every address in a page of SBD DPDRAM.

Test 4 — checks the data pattern 0x08 at every address in a page of SBD DPDRAM.

Test 5 — checks the data pattern 0x10 at every address in a page of SBD DPDRAM.

Test 6 — checks the data pattern 0x20 at every address in a page of SBD DPDRAM.

Test 7 — checks the data pattern 0x40 at every address in a page of SBD DPDRAM.

Test 8 — checks the data pattern 0x80 at every address in a page of SBD DPDRAM.

Time: 40 seconds

Warnings: None

Notes: This phase performs PIO (write and read) in bytes. If any test in this phase fails, the following hardware may be faulty:

- INTEL 80186 Microprocessor
- INTEL 80186 Microprocessor interface to the Network Access Unit Address/Data bus
- Network Access Unit interface to the 3B2 I/O bus.

Phase #12 Tests

Test Numbers:	1 through 8
Function:	These tests check the operation of the Network Access Unit interface to the 3B2 I/O bus.
Procedure:	Walk a one through a field of zeros at every address of a page of System Board (SBD) Dual Port Dynamic Random Access Memory (DPDRAM) and verify with a read.
Hardware Tested:	The Network Access Unit interface to the 3B2 I/O bus is tested.
Data Returned:	The failed test number, the expected value, and the actual value are returned.

Phase #13 — Programmed Input/Output (PIO) Word

Phase Name: Network Access Unit Word PIO (pio_2)

Type: Demand

Function: This phase checks the Network Access Unit interface to the 3B2 I/O bus.

Tests: Test 1 — checks the data pattern 0x0001 at every address in a page of SBD DPDRAM.

Test 2 — checks the data pattern 0x0002 at every address in a page of SBD DPDRAM.

Test 3 — checks the data pattern 0x0004 at every address in a page of SBD DPDRAM.

Test 4 — checks the data pattern 0x0008 at every address in a page of SBD DPDRAM.

Test 5 — checks the data pattern 0x0010 at every address in a page of SBD DPDRAM.

Test 6 — checks the data pattern 0x0020 at every address in a page of SBD DPDRAM.

Test 7 — checks the data pattern 0x0040 at every address in a page of SBD DPDRAM.

Test 8 — checks the data pattern 0x0080 at every address in a page of SBD DPDRAM.

Test 9 — checks the data pattern 0x0100 at every address in a page of SBD DPDRAM.

Test 10 — checks the data pattern 0x0200 at every address in a page of SBD DPDRAM.

Test 11 — checks the data pattern 0x0400 at every address in a page of SBD DPDRAM.

Test 12 — checks the data pattern 0x0800 at every address in a page of SBD DPDRAM.

Test 13 — checks the data pattern 0x1000 at every address in a page of SBD DPDRAM.

Test 14 — checks the data pattern 0x2000 at every address in a page of SBD DPDRAM.

Test 15 — checks the data pattern 0x4000 at every address in a page of SBD DPDRAM.

Test 16 — checks the data pattern 0x8000 at every address in a page of SBD DPDRAM.

Time: 40 seconds

Warnings: None

Notes: This phase performs PIO (write and read) in words. If any test in this phase fails, the following hardware may be faulty:

- INTEL 80186 Microprocessor
- INTEL 80186 Microprocessor interface to the Network Access Unit Address/Data bus
- Network Access Unit interface to the 3B2 I/O bus.

Phase #13 Tests

Test Numbers: 1 through 16

Function: These tests check the Network Access Unit interface to the 3B2 I/O bus.

Procedure: Walk a one through a field of zeros at every address of a page of System Board (SBD) Dual Port Dynamic Random Access Memory (DPDRAM) and verify with a read.

Hardware Tested: The Network Access Unit interface to the 3B2 I/O bus is tested.

Data Returned: The failed test number, the expected value, and the actual value are returned.

Phase #14 — DMA Transfer Byte Test

Phase Name: Network Access Unit Direct Memory Access (DMA) Byte (dmabyt)

Type: Demand

Function: This phase diagnoses and reports any errors in the Network Access Unit Direct Memory Access (DMA) channels (byte width).

Tests: Test 1 — checks DMA0 from the SBD DPDRAM to the Network Access Unit RAM.

Test 2 — checks DMA0 from the Network Access Unit RAM to the SBD DPDRAM.

Test 3 — checks DMA1 from the SBD DPDRAM to the Network Access Unit RAM.

Test 4 — checks DMA1 from the Network Access Unit RAM to the SBD DPDRAM.

Time: 1 second

Warnings: None

Notes: This phase performs DMA transfers in bytes. If any test in this phase fails, the following hardware may be faulty:

- INTEL 80186 Microprocessor
- INTEL 80186 Microprocessor interface to the Network Access Unit Address/Data bus
- Network Access Unit interface to the 3B2 I/O bus.

Phase #14 Tests

Test Number: 1

Function: This test verifies that Direct Memory Access Channel 0 (DMA0) from the System Board (SBD) Dual Port Dynamic Random Access Memory (DPDRAM) to the Network Access Unit RAM is functional.

Procedure: Write data to the SBD DPDRAM. DMA that data to the Network Access Unit RAM, and compare the data integrity.

Hardware Tested: The Network Access Unit DMA0 (INTEL 80186) from the SBD to the Network Access Unit is tested.

Data Returned: The failed test number, the actual value, and the expected value are returned.

Test Number:	2
Function:	This test verifies that DMA0 from the Network Access Unit RAM to the SBD DPDRAM is functional.
Procedure:	Write data to the Network Access Unit RAM, DMA that data to the SBD DPDRAM, and compare the data integrity.
Hardware Tested:	The Network Access Unit DMA0 (INTEL 80186) from the Network Access Unit to the SBD is tested.
Data Returned:	The failed test number, the actual value, and the expected value are returned.

==========

Test Number:	3
Function:	This test verifies that DMA1 from the SBD DPDRAM to the Network Access Unit RAM is functional.
Procedure:	Write data to the SBD DPDRAM, DMA that data to the Network Access Unit RAM, and compare the data integrity.
Hardware Tested:	The Network Access Unit DMA1 (INTEL 80186) from the SBD to the Network Access Unit is tested.
Data Returned:	The failed test number, the actual value, and the expected value are returned.

==========

Test Number:	4
Function:	This test verifies that DMA1 from the Network Access Unit RAM to the SBD DPDRAM is functional.
Procedure:	Write data to the Network Access Unit RAM, DMA that data to SBD DPDRAM, and compare the data integrity.
Hardware Tested:	The Network Access Unit DMA1 (INTEL 80186) from the Network Access Unit to the SBD is tested.
Data Returned:	The failed test number, the actual value, and the expected value are returned.

Phase #15 — DMA Transfer Word Test

Phase Name: Network Access Unit DMA WORD (dmawrd)

Type: Demand

Function: This phase diagnoses and reports errors in the Network Access Unit Direct Memory Access (DMA) channels (word width).

Tests: Test 1 — checks DMA0 from the SBD DPDRAM to the Network Access Unit RAM.

Test 2 — checks DMA0 from the Network Access Unit RAM to the SBD DPDRAM.

Test 3 — checks DMA1 from the SBD DPDRAM to the Network Access Unit RAM.

Test 4 — checks DMA1 from the Network Access Unit RAM to the SBD DPDRAM.

Time: 1 second

Warnings: None

Notes: This phase performs DMA transfers in words (16 bits). If any test in this phase fails, the following hardware may be faulty:

- INTEL 80186 Microprocessor
- INTEL 80186 Microprocessor interface to the Network Access Unit Address/Data bus
- Network Access Unit interface to the 3B2 I/O bus.

Phase #15 Tests

Test Number: 1

Function: This test verifies that Direct Memory Access Channel 0 (DMA0) from the System Board (SBD) Dual Port Dynamic Random Access Memory (DPDRAM) to the Network Access Unit RAM is functional.

Procedure: Write data to the SBD DPDRAM, DMA that data to the Network Access Unit RAM, and compare the data integrity.

Hardware Tested: This test checks the Network Access Unit DMA0 (INTEL 80186) from the SBD to the Network Access Unit is tested.

Data Returned: The failed test number, the actual value, and the expected value are returned.

Test Number:	2
Function:	This test verifies that DMA0 from the Network Access Unit RAM to the SBD DPDRAM is functional.
Procedure:	Write data to the Network Access Unit RAM, DMA that data to the SBD DPDRAM, and compare the data integrity.
Hardware Tested:	The Network Access Unit DMA0 (INTEL 80186) from the Network Access Unit to the SBD is tested.
Data Returned:	The failed test number, the actual value, and the expected value are returned.

==========

Test Number:	3
Function:	This test verifies that DMA1 from the SBD DPDRAM to the Network Access Unit RAM is functional.
Procedure:	Write data to the SBD DPDRAM, DMA that data to the Network Access Unit RAM, and compare the data integrity.
Hardware Tested:	The Network Access Unit DMA1 (INTEL 80186) from the SBD to the Network Access Unit is tested.
Data Returned:	The failed test number, the actual value, and the expected value are returned.

==========

Test Number:	4
Function:	This test verifies that DMA1 from the Network Access Unit RAM to the SBD DPDRAM is functional.
Procedure:	Write data to the Network Access Unit RAM, DMA that data to the SBD DPDRAM, and compare the data integrity.
Hardware Tested:	The Network Access Unit DMA1 (INTEL 80186) from the Network Access Unit to the SBD is tested.
Data Returned:	The failed test number, the actual value, and the expected value are returned.

Phase #16 — INTEL 82586 Internal Loop

Phase Name: Network Access Unit INTEL 82586 Internal Loop — 64 Bytes (sl_int_1)

Type: Demand

Function: This phase verifies the operation of the Network Access Unit INTEL 82586 Local Area Network (LAN) Coprocessor.

Tests: Test 1 — checks the INTEL 82586 LAN Coprocessor initialization.

Test 2 — checks the INTEL 82586 LAN Coprocessor receive frame area initialization.

Test 3 — checks the INTEL 82586 LAN Coprocessor self-diagnostic.

Tests 6 through 11 — check the INTEL 82586 LAN Coprocessor configuration command.

Tests 12 through 14 — check the INTEL 82586 LAN Coprocessor receive activities.

Tests 15 through 24 — check the INTEL 82586 LAN Coprocessor packet transmission and status.

Test 25 — checks the INTEL 82586 LAN Coprocessor interface to the Address/Data bus.

Test 500 — checks the INTEL 82586 LAN Coprocessor interface to the Control bus.

Time: 3 seconds

Warnings: None

Notes: This phase internally loops a 64-byte packet. If any test in this phase fails, the following hardware may be faulty:

- INTEL 82586 LAN Coprocessor
- INTEL 80186 Microprocessor and the INTEL 82586 LAN Coprocessor Control/Status bus
- INTEL 82586 LAN Coprocessor interface to the Network Access Unit Address/Data bus
- Network Access Unit RAM/ROM.

Phase #16 Tests

Test Number: 1

Function: This test verifies that the INTEL 82586 LAN Coprocessor passes its initialization procedure.

Procedure: The System Configuration Pointer (SCP) and Intermediate System Control Pointer (ISCP) are in Network Access Unit ROM. The INTEL 80186 Microprocessor sets up the System Control Block (SCB) in Network Access Unit RAM.

The SCP begins at address 0xFFFF6. It has two purposes: first purpose is to specify the width of the Data bus used by the INTEL 82586 LAN Coprocessor, and the second purpose is to specify the location of the ISCP. The ISCP contains the address of the SCB and the Busy bit. The Busy bit is used to indicate that the INTEL 82586 LAN Coprocessor is being initialized.

The SCB is a block of memory in Network Access Unit RAM. The INTEL 80186 Microprocessor uses the SCB for issuing commands to be executed by the INTEL 82586 LAN Coprocessor. The INTEL 82586 LAN Coprocessor uses the SCB for reporting the status of command execution.

Hardware Tested: The INTEL 82586 LAN Coprocessor is tested.

Data Returned: The failed test number, the actual value, and the expected value are returned.

==========

Test Number: 2

Function: This test initializes the INTEL 82586 LAN Coprocessor receive frame buffer area and checks the SCB status line to determine if the initialization procedure has succeeded.

Procedure: Initialize the INTEL 82586 LAN Coprocessor receive frame area in Network Access Unit RAM. Check the SCB status line to make sure that the initialization procedure was not aborted (Test 2). Service the Channel Attention (CA).

Hardware Tested: The INTEL 82586 LAN Coprocessor is tested.

Data Returned: The failed test number, the actual value, and the expected value are returned.

Test Numbers: 3 through 5

Function: These tests verify that the INTEL 82586 LAN Coprocessor can pass its internal self-diagnostics.

Procedure: Tests 3 through 5 execute as follows:

1. Initialize the diagnose command structure in Network Access Unit RAM.
2. Check the SCB command line for any pending commands (Test 3).
3. Compose the diagnose command.
4. Send the CA.
5. Check the SCB command line for diagnose command completion (Test 4).
6. Check the SCB status line for any errors in diagnose command completion.
7. Diagnose command execution (Test 5).

Hardware Tested: The INTEL 82586 LAN Coprocessor is tested.

Data Returned: The failed test number, the actual value, and the expected value are returned.

==========

Test Numbers: 6 through 11

Function: These tests check that the INTEL 82586 LAN Coprocessor can be configured.

Procedure: Tests 6 through 11 execute as follows:

1. Initialize the INTEL 82586 LAN Coprocessor configuration command structure in Network Access Unit RAM.
2. Check the SCB command line for any pending commands (Test 6).
3. Compose the configuration command.
4. Send the CA.
5. Wait for the SCB command line to be cleared, indicating the configuration command has been completed (Test 7).
6. Wait for the SCB status line to indicate the configuration command was executed without error (Test 8).
7. Service the CA.
8. Initialize the unique network address in Network Access Unit RAM.
9. Compose the address command.
10. Send the CA.

11. Check the SCB command line for any pending commands (Test 9).
12. Check the SCB command line for address command completion (Test 10).
13. Check the SCB status line for any address command execution errors (Test 11).

Hardware Tested: The INTEL 82586 LAN Coprocessor is tested.

Data Returned: The failed test number, the actual value, and the expected value are returned.

==========

Test Numbers: 12 through 14

Function: These tests verify that the receive activities of the INTEL 82586 LAN Coprocessor can be initiated.

Procedure: Tests 12 through 14 execute as follows:

1. Check the SCB command line for any pending commands (Test 12).
2. Compose the SCB RUC command which will enable the INTEL 82586 LAN Coprocessor for frame reception.
3. Send the CA.
4. Check the SCB command line for the SCB RUC command completion (Test 13).
5. Check if the SCB RUS Ready bit has been set (Test 14). The RUS Ready bit indicates that the INTEL 82586 LAN Coprocessor is ready for frame reception.

Hardware Tested: The INTEL 82586 LAN Coprocessor is tested.

Data Returned: The failed test number, the actual value, and the expected value are returned.

==========

Test Numbers: 15 through 24

Function: These tests verify that the INTEL 82586 LAN Coprocessor transmits a packet.

Procedure: Tests 15 through 24 execute as follows:

1. Initialize the transmit command area in Network Access Unit RAM.
2. Initialize the Transmit Buffer Descriptor (TBD) in Network Access Unit RAM.
3. Fill the Transmit buffer in Network Access Unit RAM with data.
4. Check the SCB command line for any pending commands (Test 15).
5. Compose the transmit command.
6. Send the CA.
7. Transmit the packet.
8. Check the SCB command line for transmit command completion (Test 16).
9. Check the SCB status line for any errors in the transmit command execution (Test 17).

10. Check the SCB status line for frame reception (Test 18).
11. Check the transmit command block for any transmission errors (Test 19).
12. Check the Receive Frame Descriptor (RFD) status for any errors in frame reception (Test 20).
13. Check the "C" bit in the RFD for the end of frame reception (Test 21).
14. Check the Receive Buffer Descriptor (RBD) for the end of the frame (Test 22).
15. Check the RBD for the use of the Receive buffer (Test 23).
16. Check the actual byte count of the packet received (Test 24).
17. Service the CA.

Hardware Tested: The INTEL 82586 LAN Coprocessor is tested.

Data Returned: The failed test number, the actual value, and the expected value are returned.

==========

Test Number: 25

Function: This test verifies that the INTEL 82586 LAN Coprocessor interface to the Network Access Unit Address/Data bus is functional.

Procedure: String compare the Receive buffer against the Transmit buffer. Service the CA.

Hardware Tested: The INTEL 82586 LAN Coprocessor interface to the Network Access Unit Address/Data bus is tested.

Data Returned: The failed test number, the actual value, and the expected value are returned.

==========

Test Number: 500

Function: This test verifies that the INTEL 82586 LAN Coprocessor can service CA interrupts.

Procedure: Check that the CA acknowledge flags have been cleared.

Hardware Tested: The INTEL 82586 LAN Coprocessor is tested.

Data Returned: The failed test number, the actual value, and the expected value are returned.

Phase #17 — INTEL 82586 Internal Loop

Phase Name: Network Access Unit INTEL 82586 Internal Loop — 1500 Bytes (sl_int_2)

Type: Demand

Function: This phase verifies the Network Access Unit INTEL 82586 STARLAN Network Controller.

Tests: Test 1 — checks the INTEL 82586 LAN Coprocessor initialization.

Test 2 — checks the INTEL 82586 LAN Coprocessor receive frame area initialization.

Tests 3 through 8 — check the INTEL 82586 LAN Coprocessor configuration command.

Tests 9 through 11 — check the INTEL 82586 LAN Coprocessor receive activities.

Tests 12 through 21 — check the INTEL 82586 LAN Coprocessor packet transmission and status.

Test 22 — checks the INTEL 82586 LAN Coprocessor interface to the Address/Data bus.

Tests 23 through 27 — no tests run.

Test 500 — checks the INTEL 82586 LAN Coprocessor interface to the Control bus.

Time: 3 seconds

Warnings: None

Notes: This phase internally loops a 1500-byte packet. If any test in this phase fails, the following hardware may be faulty:

- INTEL 82586 LAN Coprocessor
- INTEL 80186 Microprocessor and the INTEL 82586 LAN Coprocessor Control/Status bus
- INTEL 82586 LAN Coprocessor interface to the Network Access Unit Address/Data bus
- Network Access Unit RAM/ROM.

Phase #17 Tests

Test Number: 1

Function: This test verifies that the INTEL 82586 LAN Coprocessor passes its initialization process.

Procedure: The System Configuration Pointer (SCP) and Intermediate System Control Pointer (ISCP) are in Network Access Unit ROM. The INTEL 80186 Microprocessor sets up the System Control Block (SCB) in Network Access Unit RAM.

The SCP begins at address 0xFFFF6. It has two purposes: the first purpose is to specify the width of the Data bus used by the INTEL 82586 LAN Coprocessor, the second purpose is to specify the location of the ISCP. The ISCP contains the address of the SCB and the Busy bit. The Busy bit is used to indicate that the INTEL 82586 LAN Coprocessor is being initialized.

The SCB is a block of memory in Network Access Unit RAM. The INTEL 80186 Microprocessor uses the SCB for issuing commands to be executed by the INTEL 82586 LAN Coprocessor. The INTEL 82586 LAN Coprocessor uses the SCB to report the status of the command execution.

Hardware Tested: The INTEL 82586 LAN Coprocessor is tested.

Data Returned: The failed test number, the actual value, and the expected value are returned.

==========

Test Number: 2

Function: This test verifies that the INTEL 82586 LAN Coprocessor receive frame area can be initialized.

Procedure: Initialize the INTEL 82586 LAN Coprocessor receive frame area in Network Access Unit RAM. Check the SCB status line to make sure that the initialization procedure was not aborted (Test 2). Service the Channel Attention (CA).

Hardware Tested: The INTEL 82586 LAN Coprocessor is tested.

Data Returned: The failed test number, the actual value, and the expected value are returned.

Test Numbers: 3 through 8

Function: These tests verify that the INTEL 82586 LAN Coprocessor can be configured.

Procedure: Tests 3 through 8 execute as follows:

1. Initialize the INTEL 82586 LAN Coprocessor configuration command structure in Network Access Unit RAM.
2. Check the SCB command line for any pending commands (Test 3).
3. Compose the configuration command.
4. Send the CA.
5. Wait for the SCB command line to be cleared, indicating the configuration command has been completed (Test 4).
6. Wait for the SCB status line to indicate the configuration command was executed without error (Test 5).
7. Service the CA.
8. Initialize the unique network address in Network Access Unit RAM.
9. Check the SCB command line for any pending commands (Test 6).
10. Compose the address command.
11. Send the CA.
12. Check the SCB command line for address command completion (Test 7).
13. Check the SCB status line for any address command execution errors (Test 8).

Hardware Tested: The INTEL 82586 LAN Coprocessor is tested.

Data Returned: The failed test number, the actual value, and the expected value are returned.

==========

Test Numbers: 9 through 11

Function: These tests verify that the receive activities of the INTEL 82586 LAN Coprocessor can be initiated.

Procedure: Tests 9 through 11 execute as follows:

1. Check the SCB command line for any pending commands (Test 9).
2. Compose the SCB RUC command, which will enable the INTEL 82586 LAN Coprocessor for frame reception.
3. Send the CA.

4. Check the SCB command line for SCB RUC command completion (Test 10).
5. Check if the SCB RUS Ready bit has been set (Test 11). The RUS Ready bit indicates that the INTEL 82586 LAN Coprocessor is ready for frame reception.

Hardware Tested: The INTEL 82586 LAN Coprocessor is tested.

Data Returned: The failed test number, the actual value, and the expected value are returned.

==========

Test Numbers: 12 through 21

Function: These tests verify that the INTEL 82586 LAN Coprocessor transmits a packet.

Procedure: Tests 12 through 21 execute as follows:

1. Initialize the transmit command area in Network Access Unit RAM.
2. Initialize the Transmit Buffer Descriptor (TBD) in Network Access Unit RAM.
3. Fill the Transmit buffer in Network Access Unit RAM with data.
4. Check the SCB command line for any pending commands (Test 12).
5. Compose the transmit command.
6. Send the CA.
7. Transmit the packet.
8. Check the SCB command line for transmit command completion (Test 13).
9. Check the SCB status line for any errors in the transmit command execution (Test 14).
10. Check the SCB status line for frame reception (Test 15).
11. Check the transmit command block for any transmission errors (Test 16).
12. Check the Receive Frame Descriptor (RFD) status for any errors in frame reception (Test 17).
13. Check the "C" bit in the RFD, for the end of frame reception (Test 18).
14. Check the Receive Buffer Descriptor (RBD) for the end of the frame (Test 19).
15. Check the RBD for the use of the Receive buffer (Test 20).
16. Check the actual byte count of the packet received (Test 21).
17. Service the CA.

Hardware Tested: The INTEL 82586 LAN Coprocessor is tested.

Data Returned: The failed test number, the actual value, and the expected value are returned.

Test Number:	22
Function:	This test verifies that the INTEL 82586 LAN Coprocessor interface to the Network Access Unit Address/Data bus is functional.
Procedure:	String compare the Receive buffer against the Transmit buffer. Service the CA.
Hardware Tested:	The INTEL 82586 LAN Coprocessor interface to the Network Access Unit Address/Data bus is tested.
Data Returned:	The failed test number, the actual value, and the expected value are returned.

==========

Test Number:	500
Function:	This test verifies that the INTEL 82586 LAN Coprocessor Control/Status bus interface is functional.
Procedure:	Check that the CA acknowledge flags have been cleared.
Hardware Tested:	The INTEL 82586 LAN Coprocessor Control/Status bus interface is tested.
Data Returned:	The failed test number, the actual value, and the expected value are returned.

Phase #18 — STARLAN External Loop (JACK)

Phase Name: Network Access Unit 12-Byte External Loop (sl_ext_1)

Type: Normal

Function: This phase verifies the ability of the INTEL 82586 LAN Coprocessor to externally loop around a packet.

Tests: Test 1 — checks the INTEL 82586 LAN Coprocessor preinitialization.

Test 2 — checks the INTEL 82586 LAN Coprocessor initialization.

Tests 3 through 8 — check the INTEL 82586 LAN Coprocessor configuration command.

Tests 9 through 11 — check the INTEL 82586 LAN Coprocessor receive activities.

Tests 12 through 21 — check the INTEL 82586 LAN Coprocessor packet transmission and status.

Test 22 — checks the INTEL 82586 LAN Coprocessor interface to the Address/Data bus.

Tests 23 through 25 — no tests run.

Test 500 — checks the INTEL 82586 LAN Coprocessor interface to the Control/Status bus.

Time: 15 seconds

Warnings: It is possible for this phase to fail due to severe media congestion. This failure can occur when a "bus hog" is monopolizing the medium, or when traffic on the medium is so heavy that the repeated attempts by the Network Access Unit to access the medium fail. In cases where the Network Access Unit passes all but its external loop around phases, follow the procedure described in the notes below.

Notes: This phase externally loops a 12-byte diagnostic packet to the INTEL 82586 LAN Coprocessor. If any test in this phase fails, the following hardware may be faulty:

- INTEL 82586 LAN Coprocessor
- STARLAN Network cable
- Network Access Unit "input" and/or "output" jack(s)
- Network Access Unit STARLAN Network Serial Interface circuitry
- INTEL 80186 Microprocessor and the INTEL 82586 LAN Coprocessor Control/Status bus
- INTEL 82586 LAN Coprocessor interface to the Network Access Unit Address/Data bus
- Network Access Unit RAM/ROM.

Phase #18 Tests

Test Number: 1

Function: This test verifies that the INTEL 82586 LAN Coprocessor passes its preinitialization process.

Procedure: The System Configuration Pointer (SCP) and Intermediate System Control Pointer (ISCP) are in Network Access Unit ROM. The INTEL 80186 Microprocessor sets up the System Control Block (SCB) in Network Access Unit RAM.

Beginning at address 0xffff6, the SCP specifies both the width of the Data bus used by the INTEL 82586 LAN Coprocessor and the location of the ISCP. The ISCP contains the address of the SCB and the Busy bit. The Busy bit is used to indicate that the INTEL 82586 LAN Coprocessor is being initialized.

The SCB is a block of memory in Network Access Unit RAM. The INTEL 80186 Microprocessor uses the SCB to issue commands that are to be executed by the INTEL 82586 LAN Coprocessor. The INTEL 82586 LAN Coprocessor uses the SCB to report the status of command execution.

The initialization procedure is as follows:

1. The INTEL 80186 Microprocessor sets the ISCP Busy bit, issues a reset to the INTEL 82586 LAN Coprocessor, and asserts a Channel Attention (CA) interrupt.
2. The INTEL 82586 LAN Coprocessor reads the SCP, determines the bus width, and fetches the ISCP address. From to the ISCP, it fetches and stores the SCB address.
3. The INTEL 82586 LAN Coprocessor then clears the ISCP Busy bit and SCB command word. It signals an interrupt (LINT21) to the INTEL 80186 Microprocessor and waits for the next CA.

Hardware Tested: The INTEL 82586 LAN Coprocessor is tested.

Data Returned: The failed test number, the actual value, and the expected value are returned.

==========

Test Number: 2

Function: This test verifies that the INTEL 82586 LAN Coprocessor passes its initialization process.

Procedure: Initialize the INTEL 82586 LAN Coprocessor receive frame area in Network Access Unit RAM. Check the SCB status line to make sure that the initialization procedure was not aborted (Test 2) and service the CA.

Hardware Tested: The INTEL 82586 LAN Coprocessor is tested.

Data Returned: The failed test number, the actual value, and the expected value are returned.

Test Numbers: 3 through 8

Function: These tests verify that the INTEL 82586 LAN Coprocessor can be configured.

Procedure: Tests 3 through 8 execute as follows:

1. Initialize the INTEL 82586 LAN Coprocessor configuration command structure in Network Access Unit RAM.
2. Check the SCB command line for any pending commands (Test 3).
3. Compose the configuration command.
4. Send the CA.
5. Wait for the SCB command line to be cleared, indicating the configuration command has been completed (Test 4).
6. Wait for the SCB status line to indicate the configuration command was executed without error (Test 5).
7. Service the CA.
8. Initialize the unique network address in Network Access Unit RAM.
9. Check the SCB command line for any pending commands (Test 6).
10. Compose the address command.
11. Send the CA.
12. Check the SCB command line for address command completion (Test 7).
13. Check the SCB status line for any address command execution errors (Test 8).

Hardware Tested: The INTEL 82586 LAN Coprocessor is tested.

Data Returned: The failed test number, the actual value, and the expected value are returned.

==========

Test Numbers: 9 through 11

Function: These tests verify that the receive activities of the INTEL 82586 LAN Coprocessor can be initiated.

Procedure: Tests 9 through 11 execute as follows:

1. Check the SCB command line for any pending commands (Test 9).
2. Compose the SCB RUC command, which will enable the INTEL 82586 LAN Coprocessor for frame reception.
3. Send the CA.

4. Check the SCB command line for SCB RUC command completion (Test 10).
5. Check if the SCB RUS Ready bit has been set (Test 11). The RUS Ready bit indicates that the INTEL 82586 LAN Coprocessor is ready for frame reception.

Hardware Tested: The INTEL 82586 LAN Coprocessor is tested.

Data Returned: The failed test number, the actual value, and the expected value are returned.

==========

Test Numbers: 12 through 21

Function: These tests verify that the INTEL 82586 LAN Coprocessor transmits a packet.

Procedure: Tests 12 through 21 execute as follows:

1. Initialize the transmit command area in Network Access Unit RAM.
2. Initialize the Transmit Buffer Descriptor (TBD) in the Network Access Unit RAM.
3. Fill the Transmit buffer in Network Access Unit RAM with data.
4. Check the SCB command line for any pending commands (Test 12).
5. Compose the transmit command.
6. Send the CA.
7. Transmit the packet.
8. Check the SCB command line for transmit command completion (Test 13).
9. Check the SCB status line for any errors in the transmit command execution (Test 14).
10. Check the SCB status line for frame reception (Test 15).
11. Check the transmit command block for any transmission errors (Test 16).
12. Check the Receive Frame Descriptor (RFD) status for any errors in frame reception (Test 17).
13. Check the "C" bit in the RFD, for the end of frame reception (Test 18).
14. Check the Receive Buffer Descriptor (RBD) for the end of the frame (Test 19).
15. Check the RBD for the use of the Receive buffer (Test 20).
16. Check the actual byte count of the packet received (Test 21).
17. Service the CA.

Hardware Tested: The INTEL 82586 LAN Coprocessor and STARLAN Network interface circuitry is tested.

Data Returned: The failed test number, the actual value, and the expected value are returned.

Test Number:	22
Function:	This test verifies that the INTEL 82586 LAN Coprocessor interface to the Network Access Unit Address/Data bus is functional.
Procedure:	String compare the Receive buffer against the Transmit buffer. Service the CA.
Hardware Tested:	The INTEL 82586 LAN Coprocessor interface to the Network Access Unit Address/Data bus is tested.
Data Returned:	The failed test number, the actual value, and the expected value are returned.

==========

Test Number:	500
Function:	This test verifies that the INTEL 82586 LAN Coprocessor Control/Status bus interface is functional.
Procedure:	Check that the CA acknowledge flags have been cleared.
Hardware Tested:	The INTEL 82586 LAN Coprocessor Control/Status bus interface is tested.
Data Returned:	The failed test number, the actual value, and the expected value are returned.

Phase #19 — STARLAN External Loop (JACK)

Phase Name: Network Access Unit 1500-Byte External Loop (sl_ext_2)

Type: Demand

Function: This phase verifies the ability of the INTEL 82586 LAN Coprocessor to externally loop around a 1500-byte packet.

Tests: Test 1 — checks the INTEL 82586 LAN Coprocessor preinitialization.

Test 2 — checks the INTEL 82586 LAN Coprocessor initialization.

Tests 3 through 8 — check the INTEL 82586 LAN Coprocessor configuration command.

Tests 9 through 11 — check the INTEL 82586 LAN Coprocessor receive activities.

Tests 12 through 21 — check the INTEL 82586 LAN Coprocessor packet transmission and status.

Test 22 — checks the INTEL 82586 LAN Coprocessor interface to the Address/Data bus.

Tests 23 through 26 — no tests run.

Test 500 — checks the INTEL 82586 LAN Coprocessor interface to the Control/Status bus.

Time: 15 seconds

Warnings: It is possible for this phase to fail due to severe media congestion. This failure can occur when a "bus hog" is monopolizing the medium, or when traffic on the medium is so heavy that the repeated attempts by the Network Access Unit to access the medium fail. In cases where the Network Access Unit passes all except its external loop around phases, follow the procedure described in the notes below.

Notes: This phase externally loops a 1500-byte diagnostic packet to the INTEL 82586 LAN Coprocessor.

Immediately following installation, and whenever media problems are suspected, external loopback tests should be run twice: once with the Network Access Unit disconnected from the STARLAN Network (Network Access Unit jacks empty) and once with the Network Access Unit connected to the STARLAN Network.

In the case where the Network Access Unit is connected to a STARLAN Network, the packet is routed through a Network Access Unit STARLAN Network media interface jack and onto the STARLAN Network. Successful completion of this phase with the Network Access Unit configured as described tests the STARLAN Network interface circuitry, media interface connectivity through the jack, and the STARLAN Network itself.

The other configuration described tests the ability of the coprocessor to externally loop serial data to itself through the Network Access Unit STARLAN Network media interface jacks. When both Network Access Unit jacks are empty, they should loopback and properly terminate the Network Access Unit STARLAN Network interface circuitry. This configuration tests the Network Access Unit up to the media interface jacks. It also tests the Network Access Unit STARLAN Network loop around and, to some degree, termination capabilities, as provided through the "input" and "output" jacks.

If any test in this phase fails, the following hardware may be faulty:

- INTEL 82586 LAN Coprocessor
- STARLAN Network cable
- Network Access Unit "input" and/or "output" jack(s)
- Network Access Unit STARLAN Network Serial Interface circuitry
- INTEL 80186 Microprocessor and the INTEL 82586 LAN Coprocessor Control/Status bus
- INTEL 82586 LAN Coprocessor interface to the Network Access Unit Address/Data bus
- Network Access Unit RAM/ROM.

Phase #19 Tests

Test Number:	1
Function:	This test verifies that the INTEL 82586 LAN Coprocessor passes its preinitialization process.
Procedure:	The System Configuration Pointer (SCP) and Intermediate System Control Pointer (ISCP) are in Network Access Unit ROM. The INTEL 80186 Microprocessor sets up the System Control Block (SCB) in Network Access Unit RAM.
Hardware Tested:	The INTEL 82586 LAN Coprocessor is tested.
Data Returned:	The failed test number, the actual value, and the expected value are returned.

==========

Test Number:	2
Function:	This test verifies that the INTEL 82586 LAN Coprocessor passes its initialization process.
Procedure:	Initialize the INTEL 82586 LAN Coprocessor receive frame area in Network Access Unit RAM. Check the SCB status line to make sure that the initialization procedure was not aborted (Test 2). Service the Channel Attention (CA).
Hardware Tested:	The INTEL 82586 LAN Coprocessor is tested.
Data Returned:	The failed test number, the actual value, and the expected value are returned.

Test Numbers: 3 through 8

Function: These tests verify that the INTEL 82586 LAN Coprocessor can be configured.

Procedure: Tests 3 through 8 execute as follows:

1. Initialize the INTEL 82586 LAN Coprocessor configuration command structure in Network Access Unit RAM.
2. Check the SCB command line for any pending commands (Test 3).
3. Compose the configuration command.
4. Send the CA.
5. Wait for the SCB command line to be cleared, indicating the configuration command has been completed (Test 4).
6. Wait for the SCB status line to indicate the configuration command was executed without error (Test 5).
7. Service the CA.
8. Initialize the unique network address in Network Access Unit RAM.
9. Check the SCB command line for any pending commands (Test 6).
10. Compose the address command.
11. Send the CA.
12. Check the SCB command line for address command completion (Test 7).
13. Check the SCB status line for any address command execution errors (Test 8).

Hardware Tested: The INTEL 82586 LAN Coprocessor is tested.

Data Returned: The failed test number, the actual value, and the expected value are returned.

==========

Test Numbers: 9 through 11

Function: These tests verify that the receive activities of the INTEL 82586 LAN Coprocessor can be initiated.

Procedure: Tests 9 through 11 execute as follows:

1. Check the SCB command line for any pending commands (Test 9).
2. Compose the SCB RUC command to enable the INTEL 82586 LAN Coprocessor for frame reception.
3. Send the CA.

4. Check the SCB command line for SCB RUC command completion (Test 10).
5. Check if the SCB RUS Ready bit has been set (Test 11). The RUS Ready bit indicates that the INTEL 82586 LAN Coprocessor is ready for frame reception.

Hardware Tested: The INTEL 82586 LAN Coprocessor is tested.

Data Returned: The failed test number, the actual value, and the expected value are returned.

==========

Test Numbers: 12 through 21

Function: These tests verify that the INTEL 82586 LAN Coprocessor transmits a packet.

Procedure: Tests 12 through 21 execute as follows:

1. Initialize the transmit command area in Network Access Unit RAM.
2. Initialize the Transmit Buffer Descriptor (TBD) in Network Access Unit RAM.
3. Fill the Transmit buffer in Network Access Unit RAM with data.
4. Check the SCB command line for any pending commands (Test 12).
5. Compose the transmit command.
6. Send the CA.
7. Transmit the packet.
8. Check the SCB command line for transmit command completion (Test 13).
9. Check the SCB status line for any errors in the transmit command execution (Test 14).
10. Check the SCB status line for frame reception (Test 15).
11. Check the transmit command block for any transmission errors (Test 16).
12. Check the Receive Frame Descriptor (RFD) status for any errors in frame reception (Test 17).
13. Check the "C" bit in the RFD for the end of frame reception (Test 18).
14. Check the Receive Buffer Descriptor (RBD) for the end of the frame (Test 19).
15. Check the RBD for the use of the Receive buffer (Test 20).
16. Check the actual byte count of the packet received (Test 21).
17. Service the CA.

Hardware Tested: The INTEL 82586 LAN Coprocessor and STARLAN Network interface circuitry are tested.

Data Returned: The failed test number, the actual value, and the expected value are returned.

Test Number:	22
Function:	This test verifies that the INTEL 82586 LAN Coprocessor interface to the Network Access Unit Address/Data bus is functional.
Procedure:	String compare the Receive buffer against the Transmit buffer. Service the CA.
Hardware Tested:	The INTEL 82586 LAN Coprocessor interface to the Network Access Unit Address/Data bus is tested.
Data Returned:	The failed test number, the actual value, and the expected value are returned.

==========

Test Number:	500
Function:	This test verifies that the INTEL 82586 LAN Coprocessor Control/Status bus interface is functional.
Procedure:	Check that the CA acknowledge flags have been cleared.
Hardware Tested:	The INTEL 82586 LAN Coprocessor Control/Status bus interface is tested.
Data Returned:	The failed test number, the actual value, and the expected value are returned.

Phase #20 — Broadcast Transmit/Receive Test

Phase Name: Network Access Unit Broadcast Transmit/Receive (sl_bc)

Type: Demand

Function: This phase verifies the ability of the INTEL 82586 LAN Coprocessor to transmit and receive a broadcast packet.

Tests: Test 1 — checks the INTEL 82586 LAN Coprocessor preinitialization.

Test 2 — checks the INTEL 82586 LAN Coprocessor initialization.

Tests 3 through 8 — check the INTEL 82586 LAN Coprocessor configuration command.

Tests 9 through 11 — check the INTEL 82586 LAN Coprocessor receive activities.

Tests 12 through 21 — check the INTEL 82586 LAN Coprocessor packet transmission and status.

Test 22 — checks the INTEL 82586 LAN Coprocessor interface to the Address/Data bus.

Test 500 — checks the INTEL 82586 LAN Coprocessor interface to the Control/Status bus.

Time: 15 seconds

Warnings: It is possible for this phase to fail due to severe media congestion. This can occur when a "bus hog" is monopolizing the medium, or when traffic on the medium is so heavy that the repeated attempts by the Network Access Unit to access the medium fail. In cases where the Network Access Unit passes all except its external loop around phases, follow the procedure described in the notes below.

Notes: This phase externally loops a 12-byte broadcast packet to the INTEL 82586 LAN Coprocessor.

Immediately following installation, and whenever media problems are suspected, external loopback tests should be run twice: once with the Network Access Unit disconnected from the STARLAN Network (Network Access Unit jacks empty) and once with the Network Access Unit connected to the STARLAN Network.

In the case where the Network Access Unit is connected to a STARLAN Network, the packet is routed through a Network Access Unit STARLAN Network media interface jack and onto the STARLAN Network. Successful completion of this phase with the Network Access Unit configured as described tests the STARLAN Network interface circuitry, media interface connectivity via the jack, and the STARLAN Network itself.

The other configuration described tests the ability of the coprocessor to externally loop serial data to itself through the Network Access Unit STARLAN Network media interface jacks. When both Network Access Unit jacks are empty, they should loopback and properly terminate the Network Access Unit STARLAN Network interface circuitry. This configuration tests the Network Access Unit up to the media interface jacks. It also tests the Network Access Unit STARLAN Network loop around and, to some degree, termination capabilities, as provided through the "input" and "output" jacks.

If any test in this phase fails, the following hardware may be faulty:

- INTEL 82586 LAN Coprocessor
- STARLAN Network cable
- Network Access Unit "input" and/or "output" jack(s)
- Network Access Unit STARLAN Network Serial Interface circuitry
- INTEL 80186 Microprocessor and the INTEL 82586 LAN Coprocessor Control/Status bus
- INTEL 82586 LAN Coprocessor interface to the Network Access Unit Address/Data bus
- Network Access Unit RAM/ROM.

Phase #20 Tests

Test Number:	1
Function:	This test verifies that the INTEL 82586 LAN Coprocessor passes its preinitialization process.
Procedure:	The System Configuration Pointer (SCP) and Intermediate System Control Pointer (ISCP) are in Network Access Unit ROM. The INTEL 80186 Microprocessor sets up the System Control Block (SCB) in Network Access Unit RAM.
Hardware Tested:	The INTEL 82586 LAN Coprocessor is tested.
Data Returned:	The failed test number, the actual value, and the expected value are returned.

Test Number: 2

Function: This test verifies that the INTEL 82586 LAN Coprocessor passes its initialization process.

Procedure: Initialize the INTEL 82586 LAN Coprocessor receive frame area in the Network Access Unit RAM. Check the SCB status line to make sure that the initialization procedure was not aborted (Test 2). Service the Channel Attention (CA).

Hardware Tested: The INTEL 82586 LAN Coprocessor is tested.

Data Returned: The failed test number, the actual value, and the expected value are returned.

==========

Test Numbers: 3 through 8

Function: These tests verify that the INTEL 82586 LAN Coprocessor can be configured.

Procedure: Tests 3 through 8 execute as follows:

1. Initialize the INTEL 82586 LAN Coprocessor configuration command structure in Network Access Unit RAM.
2. Check the SCB command line for any pending commands (Test 3).
3. Compose the configuration command.
4. Send the CA.
5. Wait for the SCB command line to be cleared, indicating the configuration command has been completed (Test 4).
6. Wait for the SCB status line to indicate the configuration command was executed without error (Test 5).
7. Service the CA.
8. Initialize the unique network address in Network Access Unit RAM.
9. Check the SCB command line for any pending commands (Test 6).
10. Compose the address command.
11. Send the CA.
12. Check the SCB command line for address command completion (Test 7).
13. Check the SCB status line for any address command execution errors (Test 8).

Hardware Tested: The INTEL 82586 LAN Coprocessor is tested.

Data Returned: The failed test number, the actual value, and the expected value are returned.

Test Numbers: 9 through 11

Function: These tests verify that the receive activities of the INTEL 82586 LAN Coprocessor can be initiated.

Procedure: Tests 9 through 11 execute as follows:

1. Check the SCB command line for any pending commands (Test 9).
2. Compose the SCB RUC command, which will enable the INTEL 82586 LAN Coprocessor for frame reception.
3. Send the CA.
4. Check the SCB command line for SCB RUC command completion (Test 10).
5. Ensure that the SCB RUS Ready bit has been set (Test 11). The RUS Ready bit indicates that the INTEL 82586 LAN Coprocessor is ready for frame reception.

Hardware Tested: The INTEL 82586 LAN Coprocessor is tested.

Data Returned: The failed test number, the actual value, and the expected value are returned.

==========

Test Numbers: 12 through 21

Function: These tests verify that the INTEL 82586 LAN Coprocessor transmits a packet.

Procedure: Tests 12 through 21 execute as follows:

1. Initialize the transmit command area in Network Access Unit RAM.
2. Initialize the Transmit Buffer Descriptor (TBD) in Network Access Unit RAM.
3. Fill the Transmit buffer in Network Access Unit RAM with data.
4. Check the SCB command line for any pending commands (Test 12).
5. Compose the transmit command.
6. Send the CA.
7. Transmit the packet.
8. Check the SCB command line for transmit command completion (Test 13).
9. Check the SCB status line for any errors in the transmit command execution (Test 14).
10. Check the SCB status line for frame reception (Test 15).
11. Check the transmit command block for any transmission errors (Test 16).
12. Check the Receive Frame Descriptor (RFD) status for any errors in frame reception (Test 17).
13. Check the "C" bit in the RFD, for the end of frame reception (Test 18).
14. Check the Receive Buffer Descriptor (RBD) for the end of the frame (Test 19).

15. Check the RBD for the use of the Receive buffer (Test 20).
16. Check the actual byte count of the packet received (Test 21).
17. Service the CA.

Hardware Tested: The INTEL 82586 LAN Coprocessor and STARLAN Network interface circuitry are tested.

Data Returned: The failed test number, the actual value, and the expected value are returned.

==========

Test Number: 22

Function: This test verifies that the INTEL 82586 LAN Coprocessor interface to the Network Access Unit Address/Data bus is functional.

Procedure: String compare the Receive buffer against the Transmit buffer, and service the CA.

Hardware Tested: The INTEL 82586 LAN Coprocessor interface to the Network Access Unit Address/Data bus is tested.

Data Returned: The failed test number, the actual value, and the expected value are returned.

==========

Test Number: 500

Function: This test verifies that the INTEL 82586 LAN Coprocessor Control/Status bus interface is functional.

Procedure: Check that the CA acknowledge flags have been cleared.

Hardware Tested: The INTEL 82586 LAN Coprocessor Control/Status bus interface is tested.

Data Returned: The failed test number, the actual value, and the expected value are returned.

Phase #21 — Multicast Transmit/Receive Test

Phase Name: Network Access Unit Multicast Transmit/Receive (sl_mc)

Type: Demand

Function: This phase tests the ability of the INTEL 82586 LAN Coprocessor to transmit and receive a multicast packet.

Tests: Test 1 — checks the INTEL 82586 LAN Coprocessor preinitialization.

Test 2 — checks the INTEL 82586 LAN Coprocessor initialization.

Tests 3 through 8 — check the INTEL 82586 LAN Coprocessor configuration command.

Tests 9 through 11 — check the INTEL 82586 LAN Coprocessor receive activities.

Tests 12 through 21 — check the INTEL 82586 LAN Coprocessor packet transmission and status.

Test 22 — checks the INTEL 82586 LAN Coprocessor interface to the Address/Data Bus.

Tests 23 through 26 — no tests run.

Test 500 — checks the INTEL 82586 LAN Coprocessor interface to the Control/Status bus.

Time: 15 seconds

Warnings: It is possible for this phase to fail due to severe media congestion. This can occur when a "bus hog" is monopolizing the medium, or when traffic on the medium is so heavy that the repeated attempts by the Network Access Unit to access the medium fail. In cases where the Network Access Unit passes all except its external loop around phases, follow the procedure described in the notes below.

Notes: This phase externally loops a 12-byte multicast packet to the INTEL 82586 LAN Coprocessor.

Immediately following installation, and whenever media problems are suspected, external loopback tests should be run twice: once with the Network Access Unit disconnected from the STARLAN Network (Network Access Unit jacks empty) and once with the Network Access Unit connected to the STARLAN Network.

In the case where the Network Access Unit is connected to a STARLAN Network, the packet is routed through a Network Access Unit STARLAN Network media interface jack and onto the STARLAN Network. Successful completion of this phase with the Network Access Unit configured as described tests the STARLAN Network interface circuitry, media interface connectivity via the jack, and the STARLAN Network itself.

The other configuration described tests the ability of the coprocessor to externally loop serial data to itself through the Network Access Unit STARLAN Network media interface jacks. When both Network Access Unit jacks are empty, they should loopback and properly terminate the Network Access Unit STARLAN Network interface circuitry. This configuration tests the Network Access Unit up to the media interface jacks. It also tests the Network Access Unit STARLAN Network loop around and, to some degree, termination capabilities, as provided through the "input" and "output" jacks.

If any test in this phase fails, the following hardware may be faulty:

- INTEL 82586 LAN Coprocessor
- STARLAN Network cable
- Network Access Unit "input" and/or "output" jack(s)
- Network Access Unit STARLAN Network Serial Interface circuitry
- INTEL 80186 Microprocessor and the INTEL 82586 LAN Coprocessor Control/Status bus
- The INTEL 82586 LAN Coprocessor interface to the Network Access Unit Address/Data bus
- Network Access Unit RAM/ROM.

Phase #21 Tests

Test Number: 1

Function: This test verifies that the INTEL 82586 LAN Coprocessor passes its preinitialization process.

Procedure: The System Configuration Pointer (SCP) and Intermediate System Control Pointer (ISCP) are in Network Access Unit ROM. The INTEL 80186 Microprocessor sets up the System Control Block (SCB) in Network Access Unit RAM.

Beginning at address 0xffff6, the SCP specifies both the width of the Data bus used by the INTEL 82586 LAN Coprocessor and the location of the ISCP. The ISCP contains the address of the SCB and the Busy bit. The Busy bit is used to indicate that the INTEL 82586 LAN Coprocessor is being initialized.

The SCB is a block of memory in Network Access Unit RAM. The INTEL 80186 Microprocessor uses the SCB to issue commands that are to be executed by the INTEL 82586 LAN Coprocessor. The INTEL 82586 LAN Coprocessor uses the SCB to report the status of command execution.

The initialization procedure is as follows:

1. The INTEL 80186 Microprocessor sets the ISCP Busy bit issues a reset to the INTEL 82586 LAN Coprocessor, and asserts a Channel Attention (CA) interrupt.
2. The INTEL 82586 LAN Coprocessor then reads the SCP, determines the bus width, and fetches the ISCP address. From to the ISCP, it fetches and stores the SCB address.

3. The INTEL 82586 LAN Coprocessor then clears the ISCP Busy bit and clears the SCB command word. It signals an interrupt (LINT21) to the INTEL 80186 Microprocessor and waits for the next CA.

Hardware Tested: The INTEL 82586 LAN Coprocessor is tested.

Data Returned: The failed test number, the actual value, and the expected value are returned.

==========

Test Number: 2

Function: This test verifies that the INTEL 82586 LAN Coprocessor passes its initialization process.

Procedure: Initialize the INTEL 82586 LAN Coprocessor receive frame area in Network Access Unit RAM. Check the SCB status line to make sure that the initialization procedure was not aborted (Test 2). Service the Channel Attention (CA).

Hardware Tested: The INTEL 82586 LAN Coprocessor is tested.

Data Returned: The failed test number, the actual value, and the expected value are returned.

==========

Test Numbers: 3 through 8

Function: These tests verify that the INTEL 82586 LAN Coprocessor can be configured.

Procedure: Tests 3 through 8 execute as follows:

1. Initialize the INTEL 82586 LAN Coprocessor configuration command structure in Network Access Unit RAM.
2. Check the SCB command line for any pending commands (Test 3).
3. Compose the configuration command.
4. Send the CA.
5. Wait for the SCB command line to be cleared, indicating the configuration command has been completed (Test 4).
6. Wait for the SCB status line to indicate the configuration command was executed without error (Test 5).
7. Service the CA.
8. Initialize the unique network address in Network Access Unit RAM.
9. Check the SCB command line for any pending commands (Test 6).
10. Compose the address command.
11. Send the CA.

12. Check the SCB command line for address command completion (Test 7).
13. Check the SCB status line for any address command execution errors (Test 8).

Hardware Tested: The INTEL 82586 LAN Coprocessor is tested.

Data Returned: The failed test number, the actual value, and the expected value are returned.

==========

Test Numbers: 9 through 11

Function: These tests verify that the receive activities of the INTEL 82586 LAN Coprocessor can be initiated.

Procedure: Tests 9 through 11 execute as follows:

1. Check the SCB command line for any pending commands (Test 9).
2. Compose the SCB RUC command, which will enable the INTEL 82586 LAN Coprocessor for frame reception.
3. Send the CA.
4. Check the SCB command line for SCB RUC command completion (Test 10).
5. Check if the SCB RUS Ready bit has been set (Test 11). The RUS Ready bit indicates that the INTEL 82586 LAN Coprocessor is ready for frame reception.

Hardware Tested: The INTEL 82586 LAN Coprocessor is tested.

Data Returned: The failed test number, the actual value, and the expected value are returned.

==========

Test Numbers: 12 through 21

Function: These tests verify that the INTEL 82586 LAN Coprocessor transmits a packet.

Procedure: Tests 12 through 21 execute as follows:

1. Initialize the transmit command area in Network Access Unit RAM.
2. Initialize the Transmit Buffer Descriptor (TBD) in Network Access Unit RAM.
3. Fill the Transmit buffer in the Network Access Unit RAM with data.
4. Check the SCB command line for any pending commands (Test 12).
5. Compose the transmit command.
6. Send the CA.
7. Transmit the packet.
8. Check the SCB command line for transmit command completion (Test 13).
9. Check the SCB status line for any errors in the transmit command execution (Test 14).

10. Check the SCB status line for frame reception (Test 15).
11. Check the transmit command block for any transmission errors (Test 16).
12. Check the Receive Frame Descriptor (RFD) status for any errors in frame reception (Test 17).
13. Check the "C" bit in the RFD for the end of frame reception (Test 18).
14. Check the Receive Buffer Descriptor (RBD) for the end of the frame (Test 19).
15. Check the RBD for the use of the Receive buffer (Test 20).
16. Check the actual byte count of the packet received (Test 21).
17. Service the CA.

Hardware Tested: The INTEL 82586 LAN Coprocessor and the STARLAN Network interface circuitry are tested.

Data Returned: The failed test number, the actual value, and the expected value are returned.

==========

Test Number: 22

Function: This test verifies that the INTEL 82586 LAN Coprocessor interface to the Network Access Unit Address/Data bus is functional.

Procedure: String compare the Receive buffer against the Transmit buffer. Service the CA.

Hardware Tested: The INTEL 82586 LAN Coprocessor interface to the Network Access Unit Address/Data bus is tested.

Data Returned: The failed test number, the actual value, and the expected value are returned.

==========

Test Number: 500

Function: This test verifies that the INTEL 82586 LAN Coprocessor Control/Status bus interface is functional.

Procedure: Check that the CA acknowledge flags have been cleared.

Hardware Tested: The INTEL 82586 LAN Coprocessor Control/Status bus interface is tested.

Data Returned: The failed test number, the actual value, and the expected value are returned.

Chapter 16: General Purpose Synchronous Controller

Introduction

This chapter contains the diagnostic phases and tests for the 3B2 computer General Purpose Synchronous Controller (GPSC-3B) card. The GPSC-3B card is an optional feature used to communicate over synchronous channels. The GPSC-3B card supports synchronous peripherals such as synchronous modems, terminals, and printers.

The GPSC-3B card includes:

- Two physical full-duplex synchronous ports
- Communication over a serial line at a baud rate of up to 64K bits per second
- 16 MHz INTEL 80186 Microprocessor-based hardware
- Electrical interfaces for a number of popular domestic and international protocols
- GPSC-3B offloads the 3B2 computer by processing interface protocols
- Simultaneous support of multiple electrical and communication protocols
- Plug-in type connections.

An important external component of the GPSC-3B card is the 3B2 computer Input/Output (I/O) bus. The GPSC-3B card uses the I/O bus to communicate with the System Board (SBD).

If your 3B2 computer is equipped with a GPSC-3B card, the Equipped Device Table (EDT) should list it. You can use the diagnostic monitor command "show" (DMGON> **s**) to print a copy of the EDT. If the GPSC-3B card is not listed in the EDT and a VOID or NULL is listed, one or more of the following hardware devices on the GPSC-3B card may be faulty:

- INTEL 80186 Microprocessor
- GPSC-3B Read Only Memory (ROM)
- GPSC-3B ID/Vector register
- GPSC-3B Address/Data bus
- GPSC-3B interface to the 3B2 computer I/O bus.

Twenty-seven diagnostic phases run tests on all major GPSC-3B card components. The Table of Contents listing on the previous page will help you locate the descriptions for each GPSC-3B card phase and its associated tests. The phase descriptions are organized numerically in the same order that the phases are run on the 3B2 computer.

Phase Descriptions

Phase #1 — GPSC-3B — CIO and Peripheral Sanity

Phase Name: GPSC-3B — CIO Sanity and Peripheral Sanity

Type: Normal

Function: This phase verifies that the GPSC-3B Common I/O (CIO) interface is functioning properly.

Test: Test 1 — verifies that the diagnostic results can be transferred to the System Board (SBD).

Time: 1 second

Warnings: None

Notes: None

Phase #1 Test

Test Number: 1

Function: This test verifies that the CIO hardware and firmware are functioning properly.

Procedure: This test uses the following standard procedure:

1. The I/O slot number of the GPSC-3B card in the 3B2 computer is determined.
2. The ports are reset.
3. The sysgen data block is initialized.
4. The ports are initialized (sysgen) by sending express [Interrupt (INT0)] and attention (INT1) interrupts sequentially.
5. The X86 code is downloaded by using the CIO firmware command [Download Memory (DLM)].
6. Execution of the phase is started by using the CIO firmware command [Force Call to Function (FCF)].
7. A function call to **"phasend( )"** is made when the phase is complete. Phasend returns the test results to the SBD.

Hardware Tested: The GPSC-3B interface to the 3B2 computer I/O bus is tested.

Data Returned: None

Phase #2 — GPSC-3B — Upper RAM Verification

Phase Name:	GPSC-3B — Upper RAM Verification (ram_h)
Type:	Demand
Function:	This phase diagnoses and reports any errors in the operation of the GPSC-3B RAM and refresh circuit. This phase tests the upper 384 kilobytes of peripheral RAM.
Tests:	Tests 1 through 4 — write a pattern into RAM, then read it back and write the next pattern. Test 5 — fills RAM with a random pattern, waits 5 seconds, then verifies the write. This test checks the refresh circuitry. Test 6 — checks for any interaction problems between bit cells in RAM.
Time:	30 seconds
Warnings:	None
Notes:	None

Phase #2 Tests

Test Numbers:	1 through 6
Function:	These tests verify the operation of the upper 16 kilobytes of the GPSC-3B RAM (addresses 0x4000 to 0x7fff).
Procedure:	A known data pattern is written to every memory location (upper half) and verified with a read.
Hardware Tested:	The GPSC-3B RAM (upper half) is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

Phase #3 — GPSC-3B — Lower RAM Verification

Phase Name:	GPSC-3B — Lower RAM Verification (ram_l)
Type:	Demand
Function:	This phase diagnoses and reports any errors in the operation of the GPSC-3B RAM and refresh circuit. This program tests the lower 384 kilobytes of peripheral RAM.
Tests:	Tests 1 through 4 — write a pattern into RAM, then read it back and write the next pattern. Test 5 — fills RAM with a random pattern, waits 5 seconds, then verifies the write. This test checks the refresh circuitry. Test 6 — checks for any interaction problems between bit cells in RAM.
Time:	30 seconds
Warnings:	None
Notes:	None

Phase #3 Tests

Test Numbers:	1 through 6
Function:	These tests verify that the lower 16 kilobytes of the GPSC-3B RAM (addresses 0x0000 to 0x3fff) are functioning properly.
Procedure:	A known data pattern is written to every memory location (lower half) and verified with a read.
Hardware Tested:	The GPSC-3B RAM (lower half) is tested.
Data Returned:	The test number that failed, the actual data, and the expected data are returned.

Phase #4 — GPSC-3B — ROM Check Sum

Phase Name:	GPSC-3B — ROM Check Sum (rom)
Type:	Demand
Function:	This phase verifies the integrity of the GPSC-3B ROM.
Test:	Test 1 — calculates the check sum of the GPSC-3B ROM.
Time:	3 seconds
Warnings:	None
Notes:	None

Phase #4 Test

Test Number:	1
Function:	This test verifies the integrity of the GPSC-3B ROM.
Procedure:	A check sum is calculated by reading the GPSC-3B ROM. The calculated check sum is compared with the check sum stored in ROM when the ROM was initially programmed.
Hardware Tested:	The GPSC-3B ROM is tested.
Data Returned:	The number of test that failed, the actual check sum value, and the expected check sum value are returned.

Phase #5 — GPSC-3B — Upper Chip Select Registers

Phase Name:	GPSC-3B — Upper Chip Select Registers (cpu_1)
Type:	Demand
Function:	This phase tests the Upper Memory Chip Select (UMCS), Peripheral Access Chip Select (PACS), and Middle Peripheral Chip Select (MPCS) registers of the INTEL 80186 CPU.
Tests:	Test 1 through 7 — check the UMCS register. Tests 8 through 24 — check the PACS register. Tests 25 — no tests run. Tests 26 through 34 — check the MPCS register.
Time:	2 seconds
Warnings:	None
Notes:	None

Phase #5 Tests

Test Numbers:	1 through 7
Function:	These tests verify that the UMCS register is functional.
Procedure:	Write a valid data pattern to the register, read the register, and compare the value read with the value that was written.
Hardware Tested:	The UMCS register is tested.
Data Returned:	Supplemental Data: The value written is returned. Raw Data: The value read is returned.
Notes:	The values written to the register are 0xf038, 0xf838, 0xfc38, 0xfe38, 0xff38, 0xffb8, and 0xfff8.

==========

Test Numbers:	8 through 24
Function:	These tests verify that the PACS register is functional.
Procedure:	Write a valid data pattern to the register, read the register, and compare the value read with the value that was written.
Hardware Tested:	The PACS register is tested.
Data Returned:	Supplemental Data: The value written is returned. Raw Data: The value read is returned.
Notes:	The values written to the register are 0x0038, 0x0039, 0x003a, 0x003c, 0x0078, 0x00b8, 0x0138, 0x0238, 0x0438, 0x0838, 0x1038, 0x2038, 0x4038, and 0x8038.

Test Numbers:	26 through 34
Function:	These tests verify that the MPCS register is functional.
Procedure:	Write a valid data pattern to the register, read the register, and compare the value read with the value that was written.
Hardware Tested:	The MPCS register is tested.
Data Returned:	Supplemental Data: The value written is returned. Raw Data: The value read is returned.
Notes:	The values written to the register are 0x80f8, 0x80f9, 0x80fa, 0x80fc, 0x81f8, 0x82f8, 0x84f8, 0x88f8, and 0x90f8.

Phase #6 — GPSC-3B — DMA Control Registers

Phase Name:	GPSC-3B — DMA Control Registers (cpu_2)
Type:	Demand
Function:	This phase diagnoses and reports any errors in the operation of the INTEL 80186 CPU.
Tests:	Test 1 through 14 — check the DMA0 Control register. Tests 15 through 31 — check the DMA0 Terminal Count register. Tests 32 through 48 — check the DMA0 Destination (low) register. Tests 49 through 53 — no tests run. Tests 54 through 70 — check the DMA0 Source (low) register. Tests 71 through 75 — no tests run. Tests 76 through 89 — check the DMA1 Control register. Tests 90 through 106 — check the DMA1 Terminal Count register. Tests 107 through 123 — check the DMA1 Destination (low) register. Tests 124 through 140 — check the DMA1 Source (low) register.
Time:	2 seconds
Warnings:	None
Notes:	None

Phase #6 Tests

Test Numbers:	1 through 14
Function:	These tests verify that the Direct Memory Access 0 (DMA0) Control register is functional.
Procedure:	Write a valid data pattern to the register, read the register, and compare the value read with the expected value.
Hardware Tested:	The DMA0 Control register is tested.
Data Returned:	Supplemental Data: The value written is returned. Raw Data: The value read is returned.
Notes:	The values written to the register are 0x8000, 0x4000, 0x2000, 0x1000, 0x0800, 0x0400, 0x0200, 0x0100, 0x0080, 0x0040, 0x0020, 0x0010, 0x0001, and 0x0000.

==========

Test Numbers:	15 through 31
Function:	These tests verify that the DMA0 Terminal Count register is functional.
Procedure:	Write a valid data pattern to the register, read the register, and compare the value read with the expected value.
Hardware Tested:	The DMA0 Terminal Count register is tested.
Data Returned:	Supplemental Data: The value written is returned. Raw Data: The value read is returned.
Notes:	The values written to the register are 0x0000 and walk a one through a field of 16 zeros.

==========

Test Numbers:	32 through 48
Function:	These tests verify that the DMA0 Destination (low) register is functional.
Procedure:	Write a valid data pattern to the register, read the register, and compare the value read with the expected value.
Hardware Tested:	The DMA0 Destination (low) register is tested.
Data Returned:	Supplemental Data: The value written is returned. Raw Data: The value read is returned.
Notes:	The values written to the register are 0x0000 and walk a one through a field of 16 zeros.

Test Numbers:	54 through 70
Function:	These tests verify that the DMA0 Source (low) register is functional.
Procedure:	Write a valid data pattern to the register, read the register, and compare the value read with the expected value.
Hardware Tested:	The DMA0 Source (low) register is tested.
Data Returned:	Supplemental Data: The value written is returned. Raw Data: The value read is returned.
Notes:	The values written to the register are 0x00 and walk a one through a field of 16 zeros.

==========

Test Numbers:	76 through 89
Function:	These tests verify that the DMA1 Control register is functional.
Procedure:	Write a valid data pattern to the register, read the register, and compare the value read with the expected value.
Hardware Tested:	The DMA1 Control register is tested.
Data Returned:	Supplemental Data: The value written is returned. Raw Data: The value read is returned.
Notes:	The values written to the register are 0x8000, 0x4000, 0x2000, 0x1000, 0x0800, 0x0400, 0x0200, 0x0100, 0x0080, 0x0040, 0x0020, 0x0010, 0x0001, and 0x0000.

==========

Test Numbers:	90 through 106
Function:	These tests verify that the DMA1 Terminal Count register is functional.
Procedure:	Write a valid data pattern to the register, read the register, and compare the value read with the expected value.
Hardware Tested:	The DMA1 Terminal Count register is tested.
Data Returned:	Supplemental Data: The value written is returned. Raw Data: The value read is returned.
Notes:	The values written to the register are 0x0000 and walk a one through a field of 16 zeros.

Test Numbers:	107 through 123
Function:	These tests verify that the DMA1 Destination (low) register is functional.
Procedure:	Write a valid data pattern to the register, read the register, and compare the value read with the expected value.
Hardware Tested:	The DMA1 Destination (low) register is tested.
Data Returned:	Supplemental Data: The value written is returned. Raw Data: The value read is returned.
Notes:	The values written to the register are 0x0000 and walk a one through a field of 16 zeros.

==========

Test Numbers:	124 through 140
Function:	These tests verify that the DMA1 Source (low) register is functional.
Procedure:	Write a valid data pattern to the register, read the register, and compare the value read with the expected value.
Hardware Tested:	The DMA1 Source (low) register is tested.
Data Returned:	Supplemental Data: The value written is returned. Raw Data: The value read is returned.
Notes:	The values written to the register are 0x0000 and walk a one through a field of 16 zeros.

Phase #7 — GPSC-3B — CPU Writable Registers

Phase Name:	GPSC-3B — CPU Writable Registers (cpu_3)
Type:	Demand
Function:	This phase tests the operation of the internal Timer registers of the INTEL 80186 CPU.
Tests:	Tests 1 through 57 — no tests run. Tests 58 through 74 — check the Timer1 Count register. Tests 75 through 91 — check the Timer1 MCA register. Tests 92 through 107 — check the Timer1 MCB register. Tests 108 through 113 — check the Timer1 Mode register. Tests 114 through 117 — no tests run. Tests 118 through 134 — check the Timer2 Count register. Tests 135 through 151 — check the Timer2 MCA register. Tests 152 through 155 — check the Timer2 Mode register.
Time:	4 seconds
Warnings:	None
Notes:	None

Phase #7 Tests

Test Numbers:	58 through 74
Function:	These tests verify that the Timer1 Count register is functional.
Procedure:	Write a valid data pattern to the register, read the register, and compare the value read with the value that was written.
Hardware Tested:	The Timer1 Count register is tested.
Data Returned:	Supplemental Data: The value written is returned. Raw Data: The value read is returned.
Notes:	The values written to the register are 0x0000 and walk a one through a field of 16 zeros.

Test Numbers:	75 through 91
Function:	These tests verify that the Timer1 Maximum Count A (MCA) register is functional.
Procedure:	Write a valid data pattern to the register, read the register, and compare the value read with the value that was written.
Hardware Tested:	The Timer1 MCA register is tested.
Data Returned:	Supplemental Data: The value written is returned. Raw Data: The value read is returned.
Notes:	The values written to the register are 0x0000 and walk a one through a field of 16 zeros.

==========

Test Numbers:	92 through 107
Function:	These tests verify that the Timer1 Maximum Count B (MCB) register is functional.
Procedure:	Write a valid data pattern to the register, read the register, and compare the value read with the value that was written.
Hardware Tested:	The Timer1 MCB register is tested.
Data Returned:	Supplemental Data: The value written is returned. Raw Data: The value read is returned.
Notes:	The values written to the register are 0x0000 and walk a one through a field of 16 zeros.

==========

Test Numbers:	108 through 113
Function:	These tests verify that the Timer1 Mode register is functional.
Procedure:	Write a valid data pattern to the register, read the register, and compare the value read with the value that was written.
Hardware Tested:	The Timer1 Mode register is tested.
Data Returned:	Supplemental Data: The value written is returned. Raw Data: The value read is returned.
Notes:	The values written to the register are 0x0000, 0x0001, 0x0002, 0x0004, 0x0008, 0x0010, and 0x2000.

Test Numbers:	118 through 134
Function:	These tests verify that the Timer2 Count register is functional.
Procedure:	Write a valid data pattern to the register, read the register, and compare the value read with the value that was written.
Hardware Tested:	The Timer2 Count register is tested.
Data Returned:	Supplemental Data: The value written is returned. Raw Data: The value read is returned.
Notes:	The values written to the register are 0x0000 and walk a one through a field of 16 zeros.

==========

Test Numbers:	135 through 151
Function:	These tests verify that the Timer2 MCA register is functional.
Procedure:	Write a valid data pattern to the register, read the register, and compare the value read with the value that was written.
Hardware Tested:	The Timer2 MCA register is tested.
Data Returned:	Supplemental Data: The value written is returned. Raw Data: The value read is returned.
Notes:	The values written to the register are 0x0000 and walk a one through a field of 16 zeros.

==========

Test Numbers:	152 through 155
Function:	These tests verify that the Timer2 Mode register is functional.
Procedure:	Write a valid data pattern to the register, read the register, and compare the value read with the value that was written.
Hardware Tested:	The Timer2 Mode register is tested.
Data Returned:	Supplemental Data: The value written is returned. Raw Data: The value read is returned.
Notes:	The values written to the register are 0x0000, 0x0x0020 0x0x2000, and 0xc001 - read 0x8021.

Phase #8 — GPSC-3B — Interrupt Control Registers

Phase Name:	GPSC-3B — Interrupt Control Registers (cpu_4)
Type:	Demand
Function:	This phase tests the operation of the internal registers of the INTEL 80186 CPU Interrupt Controller (IC).
Tests:	Test 1 through 8 — check the IC In-service register. Tests 9 through 11 — check the Interrupt Request register. Tests 12 through 20 — check the Interrupt Mask register. Tests 21 through 23 — check the Interrupt Priority Mask register. Tests 24 through 28 — check the Interrupt Status register. Tests 29 through 40 — check the DMA0 and DMA1 Control registers. Tests 41 through 56 — check the INT0 and INT1 Control registers. Tests 57 through 69 — check the INT2 and INT3 Control registers.
Time:	2 seconds
Warnings:	None
Notes:	None

Phase #8 Tests

Test Numbers:	1 through 8
Function:	These tests verify that the IC In-service register is functional.
Procedure:	Write a valid data pattern to the register, read the register, and compare the value read with the value that was written.
Hardware Tested:	The IC In-service register is tested.
Data Returned:	Supplemental Data: The value written is returned. Raw Data: The value read is returned.
Notes:	The values written to the register are 0x0000, 0x0001, 0x0004, 0x0008, 0x0010, 0x0020, 0x0040, 0x0080, 0x0100, and 0x0200.

Test Numbers:	9 through 11
Function:	These tests verify that the Interrupt Request register is functional.
Procedure:	Write a valid data pattern to the register, read the register, and compare the value read with the value that was written.
Hardware Tested:	The Interrupt Request register is tested.
Data Returned:	Supplemental Data: The value written is returned. Raw Data: The value read is returned.
Notes:	The values written to the register are 0x0, 0x0004, and 0x0008.

==========

Test Numbers:	12 through 20
Function:	These tests verify that the Interrupt Mask register is functional.
Procedure:	Write a valid data pattern to the register, read the register, and compare the value read with the value that was written.
Hardware Tested:	The Interrupt Mask register is tested.
Data Returned:	Supplemental Data: The value written is returned. Raw Data: The value read is returned.
Notes:	The values written to the register are 0x0, 0x0001, 0x0004, 0x0008, 0x0010, 0x0020, 0x0040, 0x0080, and 0x0100.

==========

Test Numbers:	21 through 23
Function:	These tests verify that the Interrupt Priority Mask register is functional.
Procedure:	Write a valid data pattern to the register, read the register, and compare the value read with the value that was written.
Hardware Tested:	The Interrupt Priority Mask register is tested.
Data Returned:	Supplemental Data: The value written is returned. Raw Data: The value read is returned.
Notes:	The values written to the register are 0x0, 0x0001, 0x0002, and 0x0004.

Test Numbers:	24 through 28
Function:	These tests verify that the Interrupt Status register is functional.
Procedure:	Write a valid data pattern to the register, read the register, and compare the value read with the value that was written.
Hardware Tested:	The Interrupt Status register is tested.
Data Returned:	Supplemental Data: The value written is returned. Raw Data: The value read is returned.
Notes:	The values written to the register are 0x0, 0x0001, 0x0002, 0x0004, and 0x8000.

==========

Test Numbers:	29 through 40
Function:	These tests verify that the Direct Memory Access 0 (DMA0) and DMA1 Control registers are functional.
Procedure:	Write a valid data pattern to the register, read the register, and compare the value read with the value that was written.
Hardware Tested:	The DMA0 and DMA1 Control registers are tested.
Data Returned:	Supplemental Data: The value written is returned. Raw Data: The value read is returned.
Notes:	The values written to the register are 0x0, 0x0001, 0x0002, and 0x0004.

==========

Test Numbers:	41 through 56
Function:	These tests verify that the INT0 and INT1 Control registers are functional.
Procedure:	Write a valid data pattern to the register, read the register, and compare the value read with the value that was written.
Hardware Tested:	The INT0 and INT1 Control registers are tested.
Data Returned:	Supplemental Data: The value written is returned. Raw Data: The value read is returned.
Notes:	The values written to the register are 0x0, 0x0001, 0x0002, 0x0004, 0x0008, 0x0010, 0x0020, 0x0040, and 0x0080.

==========

Test Numbers:	57 through 69
Function:	These tests verify that the INT2 and INT3 Control registers are functional.
Procedure:	Write a valid data pattern to the register, read the register, and compare the value read with the value that was written.
Hardware Tested:	The INT2 and INT3 Control registers are tested.
Data Returned:	Supplemental Data: The value written is returned. Raw Data: The value read is returned.
Notes:	The values written to the register are 0x0, 0x0001, 0x0002, 0x0004, 0x0008, and 0x0010.

Phase #9 — GPSC-3B — Lower Chip Select Register

Phase Name:	GPSC-3B — Lower Chip Select Register (cpu_5)
Type:	Demand
Function:	This phase tests the operation of the Lower Memory Chip Select (LMCS) register of the INTEL 80186 CPU.
Tests:	Tests 1 through 5 — check the LMCS register.
Time:	4 seconds
Warnings:	None

Phase #9 Tests

Test Numbers:	1 through 5
Function:	These tests verify that the LMCS register is functional.
Procedure:	Write a valid data pattern to the register, read the register, and compare the value read with the value that was written.
Hardware Tested:	The LMCS register is tested.
Data Returned:	Supplemental Data: The value written is returned. Raw Data: The value read is returned.
Notes:	The values written to the register are 0x03f8, 0x07f8, 0x0ff8, 0x1ff8, and 0x3ff8.

Phase #10 — GPSC-3B — Programmed Input/Output (PIO) Byte Transfers

Phase Name:	GPSC-3B — PIO Byte Transfers (pio_1)
Type:	Demand
Function:	This phase tests byte transfers over the 3B2 computer I/O bus.
Tests:	Tests 1 through 8 — walk a one through a field of 8 zeros in every memory location of one page of memory. Test 9 — tests the Bus Abort Feature (BAF) with PIO.
Time:	80 seconds
Warnings:	None

Phase #10 Tests

Test Numbers:	1 through 8
Function:	These tests verify the operation of the I/O bus interface.
Procedure:	Write a one through a field of zeros in all memory locations of a page of System Board (SBD) RAM.
Hardware Tested:	The interface to the 3B2 computer I/O bus is tested.
Data Returned:	The data returned are the test count that failed, the address of the failure, and the data pattern that failed. The test count is the bit that was written. The address is represented as 0xc9???? or 0xd????.
Notes:	Walk a one through a field of 8 zeros is the value written to the memory locations.

==========

Test Number:	9
Function:	This test verifies that the BAF works for PIO from the SBD.
Procedure:	Set the BAF timer (Timer1) to a small value. Perform PIO from the SBD to the GPSC-3B. Should get a Timer1 interrupt indicating BAF.
Hardware Tested:	The BAF timer is tested.
Data Returned:	None
Notes:	None

Phase #11 — GPSC-3B — Programmed Input/Output (PIO) Word Transfers

Phase Name:	GPSC-3B — PIO Word Transfer (pio_2)
Type:	Demand
Function:	This phase tests the I/O Address/Data bus.
Tests:	Tests 1 through 16 — walk a one through a field of 16 zeros in every memory location of one page of memory.
Time:	80 seconds
Warnings:	None
Notes:	None

Phase #11 Tests

Test Numbers:	1 through 16
Function:	These tests verify the operation of the I/O bus.
Procedure:	Write a one through a field of zeros in all locations of one page of SBD RAM.
Hardware Tested:	The I/O bus is tested.
Data Returned:	The data returned are the test count that failed, the address of the failure, and the data pattern that failed. The test count is the bit that was written. The address is represented as 0xc9???? or 0xd????.
Notes:	Walk a one through a field of 8 zeros is the value written to the memory locations.

Phase #12 — GPSC-3B — DMA Byte Transfer

Phase Name:	GPSC-3B — DMA Byte Transfer (dmabyt)
Type:	Demand
Function:	This phase diagnoses and reports any errors in the operation of the INTEL 80186 DMA channel (byte width).
Tests:	Test 1 —checks DMA0 from the SBD to the GPSC-3B card. Test 2 —checks DMA0 from the GPSC-3B card to the SBD. Test 3 —checks Bus Abort Feature (BAF) in the DMA mode.
Time:	2 seconds
Warnings:	None
Notes:	The DMA1 is unused and therefore untested.

Phase #12 Tests

Test Number:	1
Function:	This tests verifies that Direct Memory Access Channel 0 (DMA0) from the System Board (SBD) to the GPSC-3B card is functional.
Procedure:	Perform Programmed Input/Output (PIO) to the SBD, DMA that data to the GPSC-3B card, and compare the values at each.
Hardware Tested:	The DMA0 is tested.
Data Returned:	Supplemental Data: The value written is returned. Raw Data: The value read is returned.
Notes:	Walk a one through a field of zeros is the value written to the register.

Test Number:	2
Function:	This test verifies that DMA0 from the GPSC-3B card to the SBD is functional.
Procedure:	Write GPSC-3B RAM, DMA that data to the SBD, and compare the values at each.
Hardware Tested:	The DMA0 is tested.
Data Returned:	Supplemental Data: The value written is returned. Raw Data: The value read is returned.
Notes:	Walk a one through a field of zeros is the value written to the register.

==========

Test Number:	3
Function:	This test verifies that BAF works for a DMA.
Procedure:	Set the BAF timer (Timer1) to a small value. Perform a DMA from the SBD to peripheral; should get a Timer1 interrupt indicating BAF.
Hardware Tested:	The BAF timer is tested.
Data Returned:	None
Notes:	None

Phase #13 — GPSC-3B — DMA Word Transfer

Phase Name:	GPSC-3B — DMA Word Transfer (dmawrd)
Type:	Demand
Function:	This phase diagnoses and reports any errors in the word transfer operation of the INTEL 80186 Direct Memory Access (DMA) channels (word width).
Tests:	Test 1 —checks DMA0 from the SBD to the GPSC-3B card. Test 2 —checks DMA0 from the GPSC-3B card to the SBD.
Time:	2 seconds
Warnings:	None
Notes:	The DMA channel 1 is unused and therefore untested.

Phase #13 Tests

Test Number:	1
Function:	This test verifies that Direct Memory Access Channel 0 (DMA0) from System Board (SBD) to the GPSC-3B card is functional.
Procedure:	Perform Programmed Input/Output (PIO) to the SBD, DMA that data to the GPSC-3B card, and compare the values at each.
Hardware Tested:	The DMA0 is tested.
Data Returned:	Supplemental Data: The value written is returned. Raw Data: The value read is returned.
Notes:	Walk a one through a field of zeros is the value written to the register.

==========

Test Number:	2
Function:	This test verifies that DMA0 from the GPSC-3B card to the SBD is functional.
Procedure:	Write GPSC-3B RAM, DMA that data to the SBD, and compare the values at each.
Hardware Tested:	The DMA0 is tested.
Data Returned:	Supplemental Data: The value written is returned. Raw Data: The value read is returned.
Notes:	Walk a one through a field of zeros is the value written to the register.

Phase #14 — GPSC-3B — SCC Basic Sanity

Phase Name:	GPSC-3B — SCC Basic Sanity
Type:	Demand
Function:	This phase tests the basic sanity of each Serial Communication Controller (SCC) by looping a character in local loop mode on each channel of the SCC.
Test:	Test 1 — local loops a character at 9600 baud with 2 Stop bits and no parity for Channels A and B.
Time:	12 seconds
Warnings:	None
Notes:	None

Phase #15 — GPSC-3B — Interrupt Integrity

Phase Name:	GPSC-3B — Interrupt Integrity (int_scc)
Type:	Demand
Function:	This phase tests the basic integrity of the Serial Communication Controller (SCC) interrupt and acknowledge circuitry.
Test:	Test 1 — causes a transmit buffer empty interrupt on the SCC furthest electrically from the INTEL 80186 Microprocessor, then checks to see that the proper interrupt was taken. This will verify the basic sanity of the interrupt and acknowledge circuitry to SCCs.
Time:	1 second
Warnings:	None
Notes:	None

Phase #16 — GPSC-3B — SCC Receive Buffers

Phase Name:	GPSC-3B — SCC Receive Buffers (rx_buf)
Type:	Demand
Function:	This phase diagnoses the Receive buffer of the Serial Communication Controller (SCC) by seeing if it can be filled without losing characters. Also, this phase verifies that the SCC generates the proper interrupt if the Receive buffer overflows.
Tests:	Test 1 — transmits three characters in local loop mode before reading them back. Test 2 — transmits four characters in local loop mode to force a buffer overflow interrupt.
Time:	5 seconds
Warnings:	None
Notes:	None

Phase #17 — GPSC-3B — Basic DMAC and SCC Test

Phase Name:	GPSC-3B — Basic DMAC and SCC Test (dmac)
Type:	Demand
Function:	This phase uses the Serial Communication Controllers (SCCs) in local loop around mode to see if each Direct Memory Access (DMA) channels can transfer a character properly between the SCC and Dynamic Random Access Memory (DRAM).
Tests:	Test 1 — checks transfer of character from SCC0 Channel A, Direct Memory Access Controller 0 (DMAC0) Channels 0 and 1. Test 2 — checks transfer of character from SCC0 Channel B, DMAC0 Channels 2 and 3.
Time:	15 seconds
Warnings:	None
Notes:	None

Phase #18 — GPSC-3B — Application DMA Register Test

Phase name: ADMA Test

Type: Demand

Function: This phase uses the Serial Communication Controllers (SCCs) in local loop around mode to see if each Direct Memory Access (DMA) channel can transfer a character properly between the SCC and the specified pages of the Dynamic Random Access Memory (DRAM).

Tests: Tests 1 through 12 — check the values for the ADMA register which are abcdabcdabcdabcd, where "abcd" equals to 0000 to 1011 which covers all 12 pages.

Tests 13 through 28 — walk a one through a field of zeros through each channel of the ADMA page register.

Time: 60 seconds

Warings: None

Notes: None

Phase #18 Tests

Tests Numbers: 1 through 12

Function: These tests verify that the ADMA Page register will work on all 12 pages.

Procedure: Transfer a character to each of the pages of Dynamic Random Access Memory (DRAM) using the SCC in local loop mode.

Hardware Tested: The ADMA Page register is tested

Data Returned: The failing test number is returned.

Notes: The values written to the ADMA Page register are the following:

Test	Value
1	0x0000
2	0x1111
3	0x2222
4	0x3333
5	0x4444
6	0x5555
7	0x6666
8	0x7777
9	0x8888
10	0x9999
11	0xaaaa
12	0xbbbb

Test Numbers: 13 through 28

Function: These tests walk a one through a field of zeros on each channel of the ADMA Page register to test for bits tied together.

Procedure: Transfer a character to each of the pages of DRAM using the SCC in local loop mode.

Hardware Tested: The ADMA Page register is tested.

Data Returned: The failing test number is returned.

Notes: The values written to the ADMA Page register are the following:

Test	Value
13	0x0001
14	0x0002
15	0x0004
16	0x0008
17	0x0010
18	0x0020
19	0x0040
20	0x0080
21	0x0100
22	0x0200
23	0x0400
24	0x0800
25	0x1000
26	0x2000
27	0x4000
28	0x8000

Phase #19 — GPSC-3B — Application Control Register Test

Phase name:	Application Control Register (ACR) Test
Type:	Demand
Function:	This phase tests the ACR by writing and reading a specified value to and from the register with special attention paid to bit 14 (PFAIL).
Tests:	Test 1 — checks bits 0 through 7, 9, 10, 12, and 13. Test 2 — checks bit 8 (EEPROM Chip Select). Test 3 — checks bit 15 (currently spare). Test 4 — checks bit 14 (PFAIL).
Time:	40 seconds
Warnings:	None
Notes:	Bit 8 should be turned on only once because it reduces the life of the Electrically Erasable Programmable Read Only Memory (EEPROM). The direction of the TRxC pins (Channels A and B transmit clocks) on the Serial Communication Controller (SCC) (and on 8536) must be set to input before enabling (setting to "0") either of the Transmit Clock Input Control bits (bit 12 or bit 13). Bit 11 is not tested in Phase 19. It is the data output of the EEPROM and is tested by Phase 27.

Phase #19 Tests

Test Number:	1
Function:	This test checks bits 0 through 7, 9, 10, 12, and 13.
Procedure:	Write data to the ACR, read it back, and compare it to the expected value.
Hardware Tested:	The ACR is tested.
Data Returned:	The value actually read from the ACR and the expected value are returned.
Notes:	The values written to the ACR are 0x0001 to 0x36ff, with bits 8 and 11 always zero.

==========

Test Number:	2
Function:	This test checks bit 8 (EEPROM Chip Select).
Procedure:	Write data to the ACR, read it back, and compare it to the expected value.
Hardware Tested:	The ARC is tested.
Data Returned:	The value actually read from the ACR and the expected value are returned.
Notes:	The value written to the ACR register is 0x0100.

Test Number:	3
Function:	This test checks bit 15 (currently spare).
Procedure:	Write data to the ACR, read it back, and compare it to the expected value.
Hardware Tested:	The ARC is tested.
Data Returned:	The value actually read from the ACR and the expected value are returned.
Notes:	The value written to the ACR register is 0xb000.

==========

Test Number:	4
Function:	This test checks bit 14 (PFAIL).
Procedure:	Write data to the ACR, read it back, and compare it to the expected value.
Hardware Tested:	The ARC is tested.
Data Returned:	The value actually read from the ACR and the expected value are returned.
Notes:	The value written to the ACR is 0x7000.

Phase #20 — GPSC-3B — Local SCC Interrupts

Phase Name:	GPSC-3B — SCC Local Interrupts (int_1lp)
Type:	Demand
Function:	This phase diagnoses the receive buffer break interrupts. These can be tested without an external loop around connector. Send a break character and check that the proper interrupt occurred. Test for each channel on the Serial Communication Controller (SCC).
Tests:	Test 1 — checks SCC Channel A. Test 2 — checks SCC Channel B.
Notes:	This test will not stop on the first failure but will continue until all ports have been tested. If a port(s) fails, a failure message is displayed; the other ports are then tested.

Phase #21 — Counter/Timer and Parallel I/O Test

Phase name: Counter/Timer and Parallel I/O Test (cntr_tmr)

Type: Demand

Function: This phase tests the operation of the three counter/timers.

Tests: Tests 1 through 32 — walk a one through a field of zeros and a zero through a field of ones on both Ports A and B with the Enable bit turned off.

Tests 33 and 34 — verify Counters 1 and 2 using the input clock coming in from the TRXC of the SCC.

Tests 35 through 37 — verify that each of the three timers can generate interrupts.

Time: 40 seconds

Warnings: None

Notes: The Counter 3 test resides in Phase 26 because it requires a clock coming in from the external cable

Phase #21 Tests

Test Numbers: 1 through 32

Function: These tests verify the operation of Ports A and B.

Procedure: Walk a one through a field of zeros and a zero through a field of ones on both Ports A and B with the Enable bit turned off.

Hardware Tested: The 8536 Counter/Timer chip is tested.

Data Returned: None

Notes: The following data is written to Ports A and B.

Test Number	Port	Value
1	A	0x01
2	A	0xfe
3	B	0x01
4	B	0xfe
5	A	0x02
6	A	0xfd
7	B	0x02
8	B	0xfd
9	A	0x04
10	A	0xfb
11	B	0x04
12	B	0xfb
13	A	0x08
14	A	0xf7

Test Number	Port	Value
15	B	0x08
16	B	0xf7
17	A	0x10
18	A	0xef
19	B	0x10
20	B	0xef
21	A	0x20
22	A	0xdf
23	B	0x20
24	B	0xdf
25	A	0x40
26	A	0xbf
27	B	0x40
28	B	0xbf
29	A	0x80
30	A	0x7f
31	B	0x80
32	B	0x7f

==========

Test Numbers: 33 and 34

Function: These tests check Counters 1 and 2 using the input clock coming in from the TRXC of the Serial Communication Controller (SCC).

Procedure: Program the Counter/Timer (CTR/TMR) to take its input clock for Timer1 and Timer2 from the TRXC pin of the SCC.

Hardware Tested: The 8536 Counter/Timer chip is tested.

Data Returned: None

==========

Test Numbers: 35 through 37

Function: These tests verify that each of the three timers can generate interrupts.

Procedure: Program each of the three timers to use their internal clock sources and interrupt when they have completely counted down.

Hardware Tested: The 8536 Counter/Timer chip is tested.

Data Returned: None

Phase #22 — Basic Synchronous Test

Phase name:	Basic Synchronous Test (local loop)
Type:	Demand
Function:	This phase diagnoses synchronous, byte sync, and bit sync protocols.
Tests:	Tests 1 through 32 — verify that the GPSC-3B card can transfer data.
Time:	60 seconds
Warnings:	None
Notes:	None

Phase #22 Tests

Test Numbers: 1 through 32

Function: These tests diagnose Serial Communication Controller (SCC) synchronous modes in local loopback mode.

Procedure: Program the SCC and the Direct Memory Access Controller (DMAC) to transfer characters in the following synchronous modes:

Monosync and SDLC
Character size: 5, 6, 7, and 8 bits/character
Encoding mode: NRZ, NRZI, FM0, and FM1

Hardware Tested: The SCC is tested.

Data Returned: None

Notes: The SCC is programmed as follows:

Test	Char Size	Encoding	Mode
1	8	NRZ	SDLC
2	8	NRZ	MONOSYNC
3	8	NRZI	SDLC
4	8	NRZI	MONOSYNC
5	8	FM0	SDLC
6	8	FM0	MONOSYNC
7	8	FM1	SDLC
8	8	FM1	MONOSYNC
9	7	NRZ	SDLC
10	7	NRZ	MONOSYNC
11	7	NRZI	SDLC
12	7	NRZI	MONOSYNC
13	7	FM0	SDLC
14	7	FM0	MONOSYNC
15	7	FM1	SDLC
16	7	FM1	MONOSYNC
17	6	NRZ	SDLC
18	6	NRZ	MONOSYNC
19	6	NRZI	SDLC
20	6	NRZI	MONOSYNC
21	6	FM0	SDLC
22	6	FM0	MONOSYNC
23	6	FM1	SDLC
24	6	FM1	MONOSYNC
25	5	NRZ	SDLC
26	5	NRZ	MONOSYNC
27	5	NRZI	SDLC
28	5	NRZI	MONOSYNC
29	5	FM0	SDLC
30	5	FM0	MONOSYNC
31	5	FM1	SDLC
32	5	FM1	MONOSYNC

Phase #23 — GPSC-3B — Complete DMAC and SCC Test

Phase Name: GPSC-3B2 — Complete DMAC and SCC Test (scc_dmac)

Type: Demand

Function: This phase checks the Serial Communication Controller (SCC) and Direct Memory Access Controller (DMAC) together with all possible baud rates, character bit sizes, parity, Stop bits, various address locations in Dynamic Random Access Memory (DRAM) and various size of data blocks to be transferred. The tests are done in local loop mode so they can run without special loop around cables connected to the board.

Tests: Test 1 — checks SCC and DMAC0.

Test 2 — checks SCC and DMAC0.

Data Returned: The parameters of the failed test (baud rate, Stop bit, character size, block size, source address, and destination address), type of failure, expected data, and actual data are returned.

All addresses and physical addresses are returned. The baud rate, parity, Stop bit, and character size parameters are returned in a numeric code which must be translated with the following tables:

Returned Baud Rate	Baud Rate Setting
bd=0	50
bd=1	75
bd=2	110
bd=3	134.5
bd=4	150
bd=5	200
bd=6	300
bd=7	600
bd=8	1200
bd=9	1800
bd=10	2400
bd=11	4800
bd=12	9600
bd=13	19.2K
bd=14	38.4K

Returned Parity	Parity Setting
par=0	None
par=1	Odd
par=2	Even

Returned Stop Bit Value	Stop Bit Setting
stop=0	2 Stop bits
stop=1	1.5 Stop bits
stop=2	1 Stop bit

Returned Character Size Value	Character Size Setting
ch sz=0	8 bits
ch sz=1	7 bits
ch sz=2	6 bits
ch sz=3	5 bits

Notes: This test will not stop on first failure but will continue until all ports are tested. If a port(s) fails, a failure message is displayed, the other ports are then tested.

Phase #24 — GPSC-3B — External SCC Interrupts

Phase Name: GPSC-3B External SCC Interrupts (int_elp)

Type: Interactive

Function: This phase provides the external looping cable from one port to another on the GPSC-3B card.

All eight port channels are tested for the following interrupts in this phase:

parity error
framing error
CTS changed
DCD dropped

Tests: Tests 1 through 4 — check transmission of interrupts from SCC Channel A (Port 0) to SCC Channel B (Port 1):

Test 1 — Parity error.
Test 2 — Framing error.
Test 3 — CTS change.
Test 4 — DCD dropped.

Tests 5 through 8 — check transmission of interrupts from SCC Channel B (Port 1) to SCC Channel A (Port 0):

Test 5 — Parity error.
Test 6 — Framing error.
Test 7 — CTS change.
Test 8 — DCD dropped.

Phase #25 — GPSC-3B External Drivers and Receivers

Phase Name:	GPSC-3B External Drivers and Receivers (xmit_recv)
Type:	Interactive
Function:	This phase provides external looping cable from one port to another on the GPSC-3B card. The transmit-receive drivers test will send the entire American Standard Code for Information Interchange (ASCII) character set for each Serial Communication Controller (SCC) at the maximum and minimum supported baud rates (38.4 kilobyte, 50).
Tests:	Test 1 — SCC Channel A transmit - SCC Channel B receive (38400 baud) Test 2 — SCC Channel A receive - SCC0 Channel B transmit (38400 baud) Test 3 — SCC Channel A transmit - SCC Channel B receive (50 baud) Test 4 — SCC3 Channel A receive - SCC2 Channel B transmit (50 baud)
Data Returned:	The expected and actual data values are returned.
Notes:	This test will not stop on first failure but will continue until all ports are tested. If a port(s) fails, a failure message is displayed; the other ports are then tested.

Phase #26 — Complete Synchronous Test (External Loop)

Phase Name:	Complete Synchronous Test (External Loop)
Type:	Interactive
Function:	This phase diagnoses Serial Communication Controller (SCC), the Direct Memory Access Controller (DMAC), and balanced and unbalanced transceivers. Also tests encoding/decoding modes, internal/external clocking, and clock recovery using DPLL.
Tests:	No tests run.
Time:	0 seconds
Warnings:	None
Notes:	This phase can only be run with special equipment that is only available only at the factory. It requires a set of special cables and a special loopback circuit board.
Hardware Tested:	The SCC, DMAC, and the balanced and unbalanced drivers and receivers are tested in this phase. Counter/Timer 3 is also tested with an external clock in this phase.

Phase #27 — Complete EEPROM

Phase Name: EEPROM Test

Type: Interactive

Function: This phase diagnoses Electronically Erasable Programmable Read Only Memory (EEPROM) erase, read, and write capability using selected addresses and all addresses. Test the operation of the EEPROM and pattern tests selected 64X16-bit registers.

Tests: Tests 1 through 64 — verify that all registers of the EEPROM have been erased.

Tests 65 through 128 — verify that a zero has been written to all registers of the EEPROM.

Test 129 — verifies that Register 10 has been erased.

Test 130 — verifies that Register 15 has been erased.

Test 131 — verifies that Register 20 has been erased.

Test 132 — verifies that Register 25 has been erased.

Test 133 — verifies that Register 30 has been erased.

Test 134 — verifies that Register 50 has been erased.

Tests 135 through 198 — verify that all registers of the EEPROM have been written to the number value of the register.

Tests 199 and 200 — verify the erase/write disable operation.

Tests 201 through 264 — verify that the contents of the EEPROM have been restored to original values.

Time: 3 seconds

Warnings: This phase will test the operation of the GPSC-3B EEPROM device. The device manufacturer does not guarantee proper operation after 10,000 write/erase cycles per register. After 10,000 write/erase cycles to a particular register, that register could be rendered useless. This diagnostic performs an erase/write cycle to each of the 64 available registers a maximum of 5 times. If this diagnostic fails, the attempt to restore the original contents of the EEPROM cannot be guaranteed.

Notes: All diagnostics test the EEPROM device and the Application Control Register (ACR). Therefore, if a test fails it cannot be determined which device caused the failure.

Phase #27 Tests

Test Numbers:	1 through 64
Function:	These tests verify that all registers of the EEPROM have been erased.
Procedure:	Read all registers and compare with 0xFFFF.
Hardware Tested:	The EEPROM is tested.
Data Returned:	Supplemental Data: The values read from the registers are returned. Raw Data: The register numbers are returned.

==========

Test Numbers:	65 through 128
Function:	These tests verify that a zero has been written to all registers of the EEPROM.
Procedure:	Read all registers and compare with zero.
Hardware Tested:	The EEPROM is tested.
Data Returned:	Supplemental Data: The values read from the registers are returned. Raw Data: The register numbers are returned.

==========

Test Number:	129
Function:	This test verifies that Register 10 has been erased.
Procedure:	Read Register 10 and compare with 0xFFFF.
Hardware Tested:	The EEPROM is tested.
Data Returned:	Supplemental Data: The value read from the register is returned. Raw Data: The register number is returned.

==========

Test Number:	130
Function:	This test verifies that Register 15 has been erased.
Procedure:	Read Register 15 and compare with 0xFFFF.
Hardware Tested:	The EEPROM is tested.
Data Returned:	Supplemental Data: The value read from the register is returned. Raw Data: The register number is returned.

Test Number: 131

Function: This test verifies that Register 20 has been erased.

Procedure: Read Register 20 and compare with 0xFFFF.

Hardware Tested: The EEPROM is tested.

Data Returned: Supplemental Data: The value read from the register is returned.
Raw Data: The register number is returned.

==========

Test Number: 132

Function: This test verifies that Register 25 has been erased.

Procedure: Read Register 25 and compare with 0xFFFF.

Hardware Tested: The EEPROM is tested.

Data Returned: Supplemental Data: The value read from the register is returned.
Raw Data: The register number is returned.

==========

Test Number: 133

Function: This test verifies that Register 30 has been erased.

Procedure: Read Register 30 and compare with 0xFFFF.

Hardware Tested: The EEPROM is tested.

Data Returned: Supplemental Data: The value read from the register is returned.
Raw Data: The register number is returned.

==========

Test Number: 134

Function: This test verifies that Register 50 has been erased.

Procedure: Read Register 50 and compare with 0xFFFF.

Hardware Tested: The EEPROM is tested.

Data Returned: Supplemental Data: The value read from the register is returned.
Raw Data: The register number is returned.

==========

Test Numbers: 135 through 198

Function: These tests verify that all registers of the Electrically Erasable Programmable Read Only Memory (EEPROM) have been written to the number value of the register (for example, 0 in Register 0, ... , 22 in Register 22, ..., 63 in Register 63).

Procedure: Read all registers and compare with the appropriate data.

Hardware Tested: The EEPROM is tested.

Data Returned: Supplemental Data: The value read from the registers is returned.
Raw Data: The register numbers are returned.

Test Numbers:	199 and 200
Function:	These tests verify the erase/write disable operation.
Procedure:	Disable write and erase operation on the EEPROM, then try to erase Register 33, try to write Register 21.
Hardware Tested:	The EEPROM is tested.
Data Returned:	Supplemental Data: The value read from the register is returned. Raw Data: The value expected to be read from the register is returned.

==========

Test Numbers:	201 through 264
Function:	These tests verify that the contents of the EEPROM have been restored to original values.
Procedure:	Write original contents to EEPROM; read it back and compare.
Hardware Tested:	The EEPROM is tested.
Data Returned:	Supplemental Data: The value read from the register is returned. Raw Data: The value expected to be read from the register is returned.

Glossary

The following is a glossary of terms used in this document.

3BNET
A high-speed networking system that facilitates file transfer, remote command execution, and network management functions among 3B2 computers.

8-bit ASCII interface
An interface standard that allows 8-bit coded data to be transmitted between computers or between a computer and a terminal.

access
The ability to send data to, or receive data from, a peripheral unit or storage device.

ASCII Code
American Standard Code for Information Interchange — An 8-bit code that is used when transmitting data between computers or between a computer and a terminal.

baud
A unit of measurement for data transmission speed, such as bits per second.

bit
One binary digit (0 or 1).

block
One or more bytes treated as a unit for reading and writing data, usually 512, 1024, or 2048 bytes.

boot or bootstrap
To perform a routine that starts the UNIX operating system running a computer. The main steps are loading the UNIX operating system from the boot device into main memory, configuring the system, and starting the system running.

boot device
The device from which the boot program is loaded.

boot program
A software program that loads the operating system into the computer.

byte
8 bits — The equivalent of one character of text.

cartridge tape
A cartridge containing magnetic tape used for storing information.

cartridge tape drive
A mass storage device that uses removable cartridge tapes.

command
A word or string of letters and/or special characters that instructs the computer to perform a task.

command line
A set of commands and arguments that tell the computer to perform one or more processing tasks. Commands always begin immediately after the shell prompt and are separated with semicolons. They can continue for several (terminal) lines, up to 256 characters.

console terminal
See "system console."

contty
Name of the unlabeled serial jack (located next to the CONSOLE jack) located on the back of a 3B2 computer.

CTC
Cartridge Tape Controller — An I/O card that provides interface circuitry for cartridge tape drives. A CTC card is not used with a SCSI tape.

default
A fixed value that is used when a value is not assigned to a variable.

DGMON
Diagnostic Monitor — An on-line diagnostic tool that is used to initiate diagnostic tests of 3B2 computer circuit cards.

diagnostics
A series of tests your 3B2 computer performs to check its operation.

DMAC
Direct Memory Access Controller — A unit that controls memory accesses.

DUART
Dual Universal Asynchronous Receiver/Transmitter — Two integrated circuits that each perform serial-to-parallel and parallel-to-serial conversion of digital data.

EDT
Equipped Device Table — A table that contains system configuration information.

EIA
Electronics Industries Association — An organization that maintains standards for the electronic industry.

EPORTS feature
Enhanced Peripheral Ports Controller — A feature card and associated software that provides eight RS-232-C serial interfaces to support peripheral devices such as terminals, modems, and serial printers.

error
A condition that results when the requested operation could not be performed.

error message
An indication that an error has been detected.

failure
Inability of a unit to perform its function.

fault
A condition that causes a device to fail to perform its required task.

feature card
Any add-on circuit board that provides an interface to peripherals or expands the systems capabilities.

FIFO
First-In-First-Out — A queuing technique in which, the next item to be retrieved, is the item that has been in the queue for the longest period of time.

file
A collection of data that has a name.

file system
A file arrangement on a segment of the hard disk that can be mounted or unmounted. (Mounting a file makes it available to the user.)

firmware
Microprograms, usually in Read Only Memory (ROM).

firmware state
The operating state of the 3B2 computer that allows you to make a floppy key, run diagnostics, change the firmware password, dump the system image to a floppy disk or hard disk, test hardware and firmware, and display the system generic version.

floppy disk
A flat, flexible disk on which programs and data can be stored.

floppy disk drive
A device that allows your 3B2 computer to read and write information to and from floppy disks.

hard disk (drive)
A device used to store large amounts of data in a short amount of time.

hardware
The physical components of a computer.

initialize
See "boot."

ISC feature
Intelligent Serial Controller — A feature card and associated hardware that allows the 3B2 computer to communicate in a synchronous environment.

K or Kilobyte
Abbreviations for the number 1024. The "K" comes from kilo, meaning one thousand. Therefore, one Kilobyte equals 1024 bytes. A full 8½- by 11-inch page contains about 2000 characters or 2 Kbytes of data.

local area network
A means of connecting computers that reside in the same room, building, or group of buildings which allows individual nodes (computers) to share resources.

login
A string of letters and/or numbers that lets the 3B2 computer know with whom it is communicating.

M or Megabyte
Abbreviations for million. One Mbyte or megabyte is 2^{20} (1,048,576) bytes when you are considering Random Access Memory. However, it is 10^6 bytes when you are considering SCSI hard disks.

MAU
Math Accelerator Unit — A chip that provides IEEE standard floating point support for the microprocessor.

MMU
Memory Management Unit — A chip that provides support for running the paging scheme of memory management. The chip makes use of tables maintained by the kernel for performing address translations.

modem
A device used to transmit digital data over voice telephone lines. (The word "modem" derives from MODulate and DEModulate.)

modular connector
Modular telephone jacks or plugs. The 4-pin modular connectors are used for telephone connections, and 8-pin modular connectors are used for data connections.

MPE feature
Multiprocessor Enhancement — A feature card [referred to as Multiprocessor Enhancement (MPE) card and Multiprocessor Board (MPB)] and associated software that off-loads user processes from the primary processor to improve the performance of the system.

multiuser state
The operating state of the 3B2 computer that allows more than one person to use the computer simultaneously. Also called "run level 2."

NI
Network Interface — A local Network Interface card that allows interconnection of computers.

NVRAM
Nonvolatile Random Access Memory — A type of Random Access Memory (RAM) used to save essential information that would otherwise be lost when power is removed.

off-line diagnostics
Diagnostics that are performed in the firmware mode when the operating system is not active.

operating state
An "environment" (defined internally by the computer) in which the computer must be running for specified functions to be performed. The 3B2 computer has seven operating states (0 through 6): powerdown state (0), single-user state (1), multiuser state (2), two unassigned states (3 and 4), firmware state (5), and reboot state (6). Also called "run levels."

operating system
Software that controls the execution of programs. An operating system may provide services such as resource allocation scheduling, input/output control, and data management.

parallel
More than one bit at a time.

password
Passwords are encrypted strings of characters that a user must enter to identify himself or herself to the computer. Passwords can be any mixture of numbers, special characters, or alpha characters (uppercase or lowercase).

peripheral devices
Devices such as terminals, printers, or modems that can be connected to a computer.

port
The architectural term for a connection between a computer and an input/output device (such as a terminal) through which data is transferred. The physical connection between the computer and another device is made through a modular jack on the computer.

PORTS feature
Peripheral Ports Controller — A feature card and associated software that provides four RS-232-C serial ports and one parallel port which support peripheral devices such as terminals, modems, and serial printers.

PROM
Programmable Read Only — A type of Read Only Memory (ROM) that can be programmed.

prompt
A symbol (or a string of letters and/or numbers) that the computer displays on the terminal screen as a signal that it is waiting for input (such as a login or a command) from you. The **$** is the UNIX system shell default prompt.

queue
A list of the processes and/or data (in sequential order) that are being saved for processing.

RAM
Random Access Memory — A type of memory in which you can directly access any location in memory.

reboot
Perform a routine that restarts the UNIX operating system. The main steps are loading the UNIX operating system from the hard disk into main memory (present contents are overwritten), configuring the system, and restarting the system.

removable storage media
Floppy disks and cartridge tapes.

ROM
Read Only Memory — A type of memory chip that can be read but cannot be written. It is used in microcomputers that always execute the same program.

root directory
The highest ranked directory in a UNIX operating system file system. Root is the directory in which all other file systems are created. It is represented by the symbol /.

root login
The login ID that you must use when you want to do system administration tasks or work that requires you to shut down the machine. Also called "super-user login."

RS-232-C
The standard serial interface adopted by the Electronics Industries Association.

run level
See "operating state."

sanity
A term that refers to basic processor operation. A processor that is functioning correctly is sane. A processor that is functioning incorrectly is insane.

SCSI
Small Computer System Interface — An ANSI-defined standard for attaching intelligent peripherals to computers.

SCSI device
A disk or tape drive that is connected to a computer system through the SCSI bus.

SCSI Host Adapter
A feature card that provides the interface between the SCSI bus and SCSI devices.

serial
One bit at a time.

shell
A UNIX system program that processes communication between a user and a 3B2 computer. The shell is also known as a command language interpreter because it translates user commands into a language understandable by the computer.

software
Programs that enable a computer to perform its functions.

super user
See "root login."

System Administration menus
A special set of menus listing interactive commands that help you perform system administration tasks on a 3B2 computer.

system board
The card holding the main logic circuits that comprise the processing center of a computer.

system console
The terminal from which your 3B2 computer is controlled. The system console is connected through the CONSOLE jack on the back of the computer. This terminal is normally used by the system administrator.

terminal
A device with input capability similar to that of a typewriter allowing you to communicate with a computer. This terminal is normally connected through an EPORTS or PORTS card.

UNIX operating system
A general purpose, multiuser, interactive, multitasking operating system developed by AT&T. (Refer to "shell.")

XDC
Expansion Disk Controller — An expansion card that allows two additional hard disk drives to be added to the system.

Index

R

S

U

V

Your comments and suggestions are appreciated and will help us to provide the best documentation for your use.

1. How would you rate this document for COMPLETENESS? (Please Circle)

   Excellent Adequate Poor
   4-------- ---------3-------- ---------2-------- ---------1-------- ---------0

2. Identify any information that you feel should be included or removed.

   _____

   _____

3. How would you rate this document for ACCURACY of information? (Please Circle)

   Excellent Adequate Poor
   4-------- ---------3-------- ---------2-------- ---------1-------- ---------0

4. Specify page and nature of any error(s) found in this document.

   _____

   _____

5. How would you rate this document for ORGANIZATION of information? (Please Circle)

   Excellent Adequate Poor
   4-------- ---------3-------- ---------2-------- ---------1-------- ---------0

6. Describe any format or packaging problems you have experienced with this document.

   _____

   _____

7. Do you have any general comments or suggestions regarding this document?

   _____

   _____

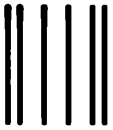
8. We would like to know a little about your background as a user of this document:

   A. Your job function _____ .

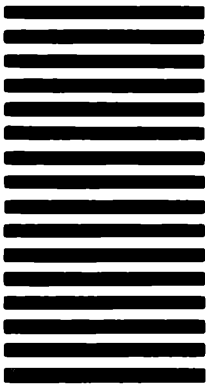
   B. Number of years experience with computer hardware: operation _____ , maintenance _____ .

   C. Number of years experience with computer software: user _____ , programmer _____ .

Your Name _____ Phone No. _____
Company _____
Address _____
City & State _____ Zip Code _____



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES



BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 1999 GREENSBORO, N.C.

POSTAGE WILL BE PAID BY ADDRESSEE

DOCUMENTATION SERVICES
2400 Reynolda Road
Winston-Salem, N.C. 27106-9989

Do Not Tear—Fold Here and Tape

COMCODE 105720627
OFF-LINE DIAGNOSTIC
SELECT CODE 305-494
ISSUE 5