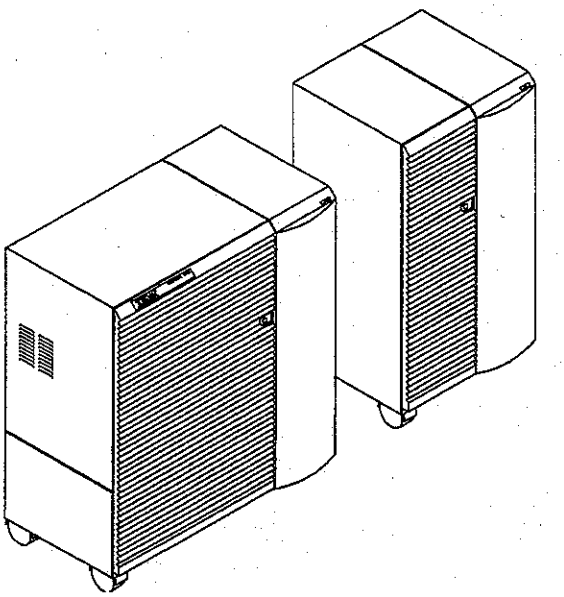
PS/2 Server Diagnostics Manual



IBM

71G2213

IBM PS/2 Server
Diagnostics Manual

Customer service:

United States:
Hardware: **1-800-IBM-SERV (1-800-426-7378)**
Software: **1-800-241-1620**

Canada:
Hardware/Software: **1-800-465-6666**

Outside of Canada or the U.S.:
Hardware/Software: **Contact the Place of Purchase**

IBM PS/2 Server Diagnostics Manual **Part Number 71G2213** (March, 1993)

Table of Contents

Chapter 4 : Functional Description of Field Maintenance Tests

List of Figures

List of Tables

WARNING

CAUTION: ONLY QUALIFIED, AUTHORIZED SERVICE PERSONNEL SHOULD RUN THE TESTS CONTAINED ON THE DIAGNOSTIC DISKETTES.

WHEN TROUBLESHOOTING THE PS/2 SERVER 195/295, FOLLOW ALL SAFETY PRECAUTIONS LISTED IN THE BEGINNING OF THE INSTALLATION GUIDE OF THE PS/2 SERVER SYSTEM/MAINTENANCE INFORMATION MANUAL

Who should read this guide

IBM's PS/2 Server computers are full-function servers designed to operate in Local Area Network (LAN) environments that require dependable application support for user's personal computers. This guide is intended for use by the individual or individuals responsible for maintaining and troubleshooting a PS/2 Server 195/295 computer on one or more networks.

This guide assumes that you are familiar with the administrative and functional requirements of your network.

Further reading

The following manuals provide additional, related reference material:

- *IBM PS/2 Server 195/295 System/Maintenance Information Manual*, International Business Machines Corporation, 1993

  This manual above provides the following:

  - Detailed installation procedures for the basic server product. In addition, the guide explains in non-technical terms how the server functions and the services it can provide.

  - Administrative procedures that explains how to manage, maintain, and operate the server on a daily basis, and provides general troubleshooting information.

  - Detailed hardware description of server options; how to plan for, install, and set up Micro Channel adapters and peripherals, IDC-controlled SCSI devices, and system modules.

Manual organization

This guide provides information about the operation of IBM PS/2 Server diagnostics, as well as procedures to follow when isolating server hardware faults using the diagnostic tests.

This guide is organized as follows:

Chapter 1, **General Description**, introduces the server diagnostics, their capabilities and components.

Chapter 2, **Diagnostic Analysis Procedures**, describes the procedures for locating a Field Replaceable Unit (FRU) using the diagnostic tests.

Chapter 3, **Functional Description Of DIAGMON Tests**, describes the tests and test routines located on the PS/2 Server Diagnostics Monitor Tests diskette.

Chapter 4, **Functional Description of Field Maintenance Tests**, describes the tests and test routines located on the PS/2 Server Field Maintenance System Tests diskette.

Conventions used in this guide

This guide observes the conventions listed in this section.

Programs, filenames, and pathnames are capitalized. For example:

PDAUTIL - a system utility
AUTOEXEC.BAT - a system filename
C:\NETWARE - an MS-DOS drive and subdirectory pathname
F:\SYSTEM - a NetWare logical drive and subdirectory pathname
SYS: - a NetWare volume

Italicized characters or words are descriptive names for items that appear in error or information messages or that you must replace with appropriate values when typing a command or response to a prompt. For example, the word

password

in a prompt means that you type in the appropriate password.

The Return, Enter, or ↵ key is referred to as the Enter key.

Warnings and cautions are presented in this form:

▲ **Warning: Before powering down or resetting the server, always alert network users, exit from all applications, and follow normal network operating system shutdown procedures.**

Task lists and checklists are presented in this form:

- ✓ Unpack the server.
- ✓ Cable the server.
- ✓ Install the server's dress kit.

Chapter 1: General Description

General Description

This manual describes the diagnostic test programs that can be executed on the PS/2 Server Models 195 and 295 for fault analysis. The diagnostic package consists of diagnostic test programs on two diskettes:

- ✓ IBM PS2 Server Diagnostic Monitor Tests diskette
- ✓ IBM PS/2 Server Field Maintenance System Tests diskette

The diagnostic test programs on the IBM PS/2 Server Diagnostic Monitor Tests diskette can be run after the server enters the Diagnostic Monitor (DIAGMON) mode.

The diagnostic test programs on the IBM PS/2 Server Field Maintenance System Tests diskette can be run by booting the server after the diskette has been inserted in drive A:.

DIAGMON Tests

There is a ROM-based diagnostic program resident on each of the server's processor modules. When the server is placed in DIAGMON mode, this ROM-based diagnostic program takes control over the server's operation and, along with the programs found on the Diagnostic Monitor Tests diskette, is used to test system components.

Diagnostic tests on the Diagnostic Monitor Tests diskette is divided up into module sections which can contain single or multiple tests. Single tests are executable files with a .ABS suffix. Multiple tests (group tests) can be run by executing batch files (.BAT suffix). DIAGMON can also be accessed from remote locations via the processor-module communications ports. For additional information that describes the particular details of the DIAGMON diagnostic tests, refer to Chapter 3 in this manual.

System components tested

The PS/2 Server contains the following major board components that are tested using the diagnostic monitor program:

- Processor (CPU) board
- Memory board
- Intelligent Disk Controller (IDC) board
- Remote Maintenance Processor (RMP) board

Diagnostic tests are also available for certain Micro Channel network adapter boards. Additional diagnostics for these adapters may be provided by the manufacturer on the option diskette. Refer to the documentation provided with these adapters for a description of their diagnostic programs.

The other primary system components that must be referenced through visual observation by the user while running diagnostics are described below.

Placing the server in DIAGMON mode

There are two basic methods of placing the server in DIAGMON mode for running the DIAGMON diagnostic tests on the Server:

✓ Power-up or reset of the server with the two front-panel buttons depressed.

✓ Using the PS/2 Field Maintenance System Tests diskette - provides access to the DIAGMON boot program via a menu selection.

❖ **Before the user attempts to run the diagnostic program, the following conditional aspects of the server should be checked:**

1. **Ascertain that the server system has been configured correctly. If any boards are misaligned or removed from their proper position, the server will indicate a configuration problem.**

2. **Check the seating of all system boards. For assurance , complete a server reset.**

3. **Ensure that the network is operating properly (network or application software has not crashed or is hung-up operationally).**

DIAGMON can operate from the system console (keyboard and VGA) or from an RS-232 terminal connected to the serial port on the processor board. The RS-232 terminal can be connected directly (using a null modem cable) or through telephone lines (using a modem on each end). Most commercially available modems are supported by DIAGMON.

DIAGMON selects which console it uses based on the setting of the AP/FP switch located on the processor board. When DIAGMON is running on a processor board designated as FP, it uses the serial port. When running on a processor board designated as AP, DIAGMON uses the keyboard and VGA monitor. on multiprocessor systems, if both processors are designated as AP, only the processor located in slot 1 will be able to use the system console. Refer to Figure 1-1 for processor board AP/FP switch position settings.

▲ **If DIAGMON cannot access its console, it flashes all three LEDs located on the processor board. On processors designated as FP, LEDs flash if DIAGMON fails to detect a modem or RS-232 terminal connected to the serial port. On processors designated as AP, LEDs flash if DIAGMON fails to detect a keyboard or VGA monitor. For example, if both processors are designated as AP, the processor located in slot 0 will flash its LEDs, while the processor in slot 0 will have access to the system console.**

DIAGMON is responsive to configuration changes that affect its console selection. For example, if you enter DIAGMON on a single processor system that is setup as an FP, the VGA screen will be blank because DIAGMON is expecting a console connected to its serial port. If no such console is available, the LEDs on the CPU board will flash. If you intend to use the system console, you may move the AP/FP switch to the AP position

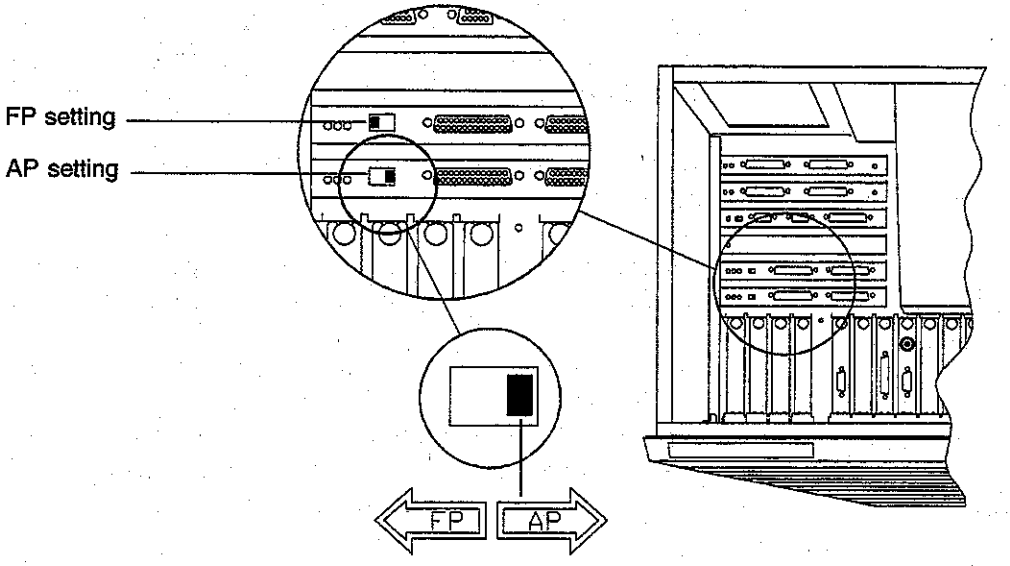


Figure 1-1
AP/FP switch locations

(to the right when facing the front of the system cabinet). DIAGMON will detect the change in the AP/FP configuration on that board and will start using the system console (after displaying its banner on the VGA screen). Alternatively, if you intend to use an RS-232 terminal as a console, you may plug one in to the serial port at anytime (using a null-modem cable). As soon as DIAGMON detects the presence of the console, it will display its banner and start accepting commands.

▲ **Changing the AP/FP switch setting on a multprocessor system may cause POST errors the next time the system reboots. If you change the AP/FP configuration in order to select a specific console for a DIAGMON, ensure that the switches on both processors are set back to their original position before rebooting the system.**

▲ **On single processor systems, if the processor board is designated as FP, the system may appear to be disabled when entering DIAGMON. If you intend to use the system console to run DIAGMON, move the AP/FP switch located on the processor board to the AP position (to the right when facing the front of the PS/2 Server cabinet). DIAGMON will detect the configuration change, display its prompt on the VGA screen, and start accepting input from the keyboard.**

The following sections describe the two basic ways to setup the server to exercise the DIAGMON diagnostics.

Diagnostic setup using server front panel buttons

The server will be setup in DIAGMON mode by simultaneously depressing the buttons on the front panel (see Figure 1) when the server is powered-on or reset. This option may be invoked if the server experiences a fatal system fault that prevents the server from booting from drive A:.

1. **Boot the DIAGMON by depressing both front panel switches plus the DC power switch. Keep the switches depressed approximately five seconds.**

Diagnostic setup using the field maintenance system tests diskette

The user may also desire to just run the DIAGMON test diagnostics from the Field Maintenance System Tests diskette. This diskette offers additional diagnostic tests that can be selected from a pull-down menu bar and is available only if the server boots-up properly. For additional information that describes the particular details of the diagnostic test functions, refer to Chapter 4 in this manual.

Running DIAGMON Tests

1. **After the server enters the DIAGMON mode, the diagnostic banner will be displayed on the monitor.**

2. **At the prompt, enter the "load" command:**

   ld

3. **Insert the IBM PS/2 Server Diagnostic Monitor Tests floppy diskette in drive A.**

4. **Enter the name of the desired test. For example, entering the command:**

   ld fd cpu.bat

   and pressing <Enter> will load the tests found in the batch file "cpu.bat" from the floppy disk (fd).

5. **Type the command**

   GO

   and press <Enter>

The user can load either batch files (*.BAT files) or diagnostic images (*.ABS files). You load . ABS files when you desire to run one particular test. Loading the .BAT files will run all the batch diagnostic files. You can view all .BAT and .ABS files by entering DIR at the A: prompt. See Appendix A for a list of .BAT file contents.

Field Maintenance System Tests

The Field Maintenance System Tests is a menu driven option program that interacts with the user through screen menu displays and dialog boxes. The Field Maintenance test diskette allows the user to choose the following test functions:

1. **Run diagnostics on the PS/2 Server.**
2. **Enter the diagnostic monitor program.**
3. **Display the ECC memory error log.**
4. **Display the NMI log.**
5. **Exit to run other desired DOS diagnostic programs.**

System components tested

Diagnostic test routines and test selections run from the IBM PS/2 Server Field Maintenance System Tests diskette allows the user to:

- Run tests on system devices (keyboard, communication ports, diskette drives, VGA board, etc.)
- Run ".DGS" programs for testing Micro Channel adapters (provided by manufacturers on an option diskette).
- Run ".DGS" programs for testing adaptes for use on a PS/2 Server
- Run tests on available system memory (memory board and memory on local CPU processor) and multiprocessor interaction.
- Capture ECC and NMI errors and diaplay them in error log files

- Setup the system in DIAGMON mode (for running system board tests from the IBM PS/2 Diagnostics Monitor Tests diskette)

- Allow other diagnostics to be run from other component manufacturers (e.g., DOS diagnostics supplied with Micro Channel based network adapters in place of (or in addition to) ".DGS" programs).

For further information concerning the functional aspects and operation of the Field Maintenance Tests, refer to Chapter 4 in this manual.

Running the Field Maintenence System Tests

This program requires a server console; you cannot run it from a remote location. To start the diagnostic tests using the Field Maintenance System Test diskette, follow this procedure:

1. **Insert the IBM PS/2 Server Field Maintenance System Tests diskette in drive A: and reboot the server.**

Server front-panel buttons, LCD, and LED's

The server is provided with a front panel that contains a liquid crystal display (LCD), two light-emitting diodes (LED's), two front-panel buttons, and a speaker (Figure 1-1). The LEDs are visible without opening the cabinet. To view the LCD or access the buttons, you unlock the door covering the front panel display and device bays.

The system software lights the green LED when you power up the server. When lit, the LED indicates that system software has detected power and this power is within the server specifications.

- **The green LED remains lit as long as the server is receiving external AC powerif the Maximum Availability Supprt System/2 (MASS/2) feature is installed on your PS/2 Server. The LED blinks**

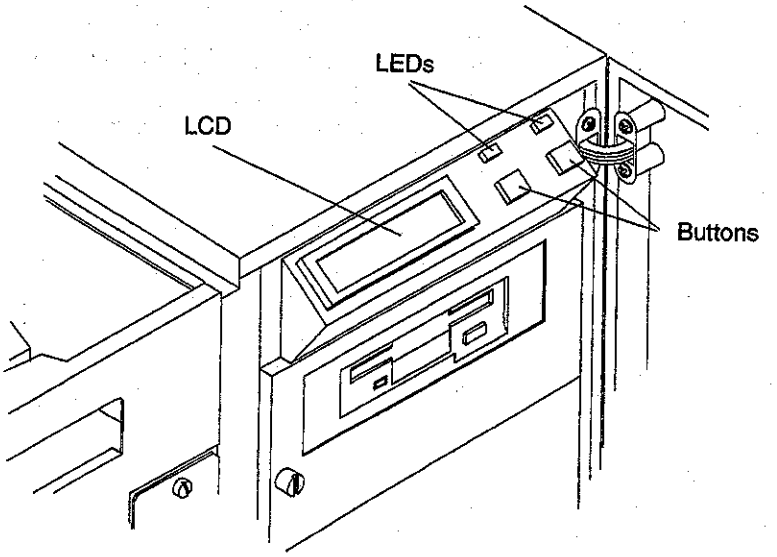


Figure 1-2
Front-panel buttons, LCD, and LEDs

if uninterruptible power supply (UPS) power is activated and the server is notified of the power failure.

The amber LED is used as an error indicator. When the LED is off, the system is operating normally. The LED is turned on if errors are detected during POST. Once the server starts operating in the OS/2 or NetWare environment, the error LED is controlled by the PS/2 Server system software. If the LED is on and blinking, a non-fatal error has occurred. When the LED is on and steady, a fatal error has occurred.

The LCD displays error and informational messages at startup and during server operation. These messages and their meanings are listed in Chapter 2 in this manual and in the *"Error Messages" section of the IBM PS/2 Server Administrator's Guide.*

In an OS/2 and NetWare environment, the user can reset the amber LED and the LCD by pressing both front-panel buttons.

The speaker is programmed to issue beep codes when some errors occur, usually those that occur befor the console monitor is initialized. Refer to Chapter 2 in this manual and the "Error Messages" section of the *IBM PS/2 Server Administrator's Guide* for a listing and description of these beep codes.

The front -panel buttons are used to scroll through error messages and for troubleshooting. If both buttons are held down at power-up or reset, the server starts up in diagnostic mode.

Chapter 2: Diagnostic Analysis Procedures

Diagnostic Analysis Procedures

Introduction

This chapter describes the diagnostic analysis procedures for the PS/2 Server Models 195 and 295. It outlines the maintenance analysis procedures to follow when exercising the diagnostics and describes the symptoms, probable causes and specific actions to be taken to resolve server malfunctions. Included sections cover the specific analysis instructions, rules, and presents analysis charts containing error correction information pertaining to system error indications from beep codes, POST errors (displayed on the monitor), operating errors (LCD indications), and the SCSI error key indications. A description covering diagnostic operation within both the level 2 and 3 environments is also addressed in this chapter. A level 2 diagnostic is executed from the processor board's external cache. A level 3 diagnostic is executed from system memory.

Analysis Procedure Instructions

DIAGMON diagnostic operation

All DIAGMON diagnostics operate in a similar manner. Level 2 diagnostics are executed from the processor board's external cache. The diagnostic is loaded individually for each particular test group (Processor (CPU), Memory, IDC, or RMP boards). To load a diagnostic into the external cache, the following monitor command must be executed.

Syntax: L[OA]D [[FD] filename | [PP | SP] [pathname]]

The FD argument names the floppy drive as the load device. PP and SP specify the parallel and serial ports respectively. If no arguments are specified, the load device is the last one that was specified by the LD or DEV monitor commands.

The syntax of the command line that executes a level 2 diagnostic is as follows:

GO [HELP] [[PASS x] [SEC x]]

where x = some numeric character that indicates how many times the test will be run.

The command line arguments shown above are common to all the diagnostics.

Diagnostics that test system memory allow two additional arguments: SIZE and BANK. These arguments are allowed on the command line at the same time making command lines like the following possible:

GO [HELP] [[SIZE x] [BANK x] [PASS x] [SEC x]]

If the GO command is typed without any arguments all sections of the diagnostic are executed once specified.

If the HELP argument is entered on the command line information will be displayed concerning syntax of the command line and the available section numbers and the tests associated with them. This argument overrides any of the others. Once the information is displayed, the program terminates and returns to the monitor prompt.

If the PASS argument is enterd on the command line the selected test sequence (default on command line specified) will be executed the specified number of times. Any value greater than zero is valid for this argument. If the argument is not specified, the default pass count value of 1 is used.

If the SEC argument is entered on the command line only the specified diagnostic section will be executed.

If the BANK argument is entered on the command line and is allowed with the particular diagnostic in question, memory testing will start at the address boundry corresponding to the beginning of the specified bank. The default starting address is the lowest address of the lowest numbered responding bank. If the SIZE argument is entered on the command line and is allowed with the particular diagnostic in question, memory testing will be restricted to a range from the starting address to that address plus the size argument in mega bytes. The default size is the highest address of the highest numbered responding memory bank.

If an invalid argument is found on the command line the diagnostic will terminate after displaying an error message and the proper syntax.

In the event of an error, the user will be prompted for a response to the question posed as to whether to continue, loop, exit or quit.

The continue option will cause the diagnostic to move on to the next test case.

The loop option will repeatedly execute all instructions that led up to the error.

The exit option will terminate the current section and continue on to the next one.

The quit option will terminate the diagnostic and return to the monitor prompt.

An example of a Diagnostic Load for the DRAM test is shown below:

ld fd DRAM.abs

Maintenance Analysis Methodology

▲ **Contact the IBM Spport Structure for assistance if a system malfunction is suspected. The user should not run ANY diagnostics without supervision from an authorized IBM service representative.**

The user may encounter error conditions while using the server. The server returns error codes and messages when an error occurs during normal operation. The user can use these codes and messages to isolate and identify probable causes of the malfunction. This section presents descriptive information for these error conditions and provides suggestive diagnostic test action to be taken to isolate the problem and determine the most likely FRU.

The server performs a power-on self-test (POST) at startup, returning an error indication if a system component fails to respond properly or if the server configuration encountered is not the same as what is contained in CMOS.

Errors that occur in the PS/2 server are reported using various combinations of either POST Error codes, Beep codes, front panel yellow error LED and LCD (operating errors) or the diagnostic LED's on the system module boards. Errors reported utilizing these methods are tabulated in this section as follows :

- Table 2-1. POST Beep Codes
- Table 2-2. POST errors displayed on Monitor
- Table 2-3. SCSI error keys
- Table 2-4. Operating errors (LCD)
- Table 2-5. RMP IP bus tests

The error log generated and displayed when the PS/2 Server Field Maintenance System Tests diskette is invoked will show all ECC memory errors and non-maskable interrupts (NMI's) that occur on the server system. This error log is described and shown in Chapter 4 of this manual.

Diagnostic Test Action

Several server operation pre-requisites should be satisfied before the user attempts to run the diagnostic programs. These pre-diagnostic checks on the server system should be accomplished prior to invoking the diagnostic monitor :

1. **Ensure that the server system is configured properly. If component boards are removed or rearranged, the server will indicate a configuration problem.**

2. **Turn power off and make sure all boards are seated correctly. Perform a reset of the server.**

3. **Ensure that the network is operating properly.**

4. **Refer to the Troubleshooting chapter (Chapter 6) of the PS/2 Server Administrator's Guide to help identify common trouble symptoms and the problems that most likely will cause them.**

If during pre-check, any apparent hardware or network problems are encountered and cannot be simply corrected, the user should:

1. **Contact their Support Structure for assistance.**

Problem Analysis Charts

The following tables outline the various error conditions that may occur within the server. A description of the failure, most likely FRU replacement and the particular diagnostic test that should be run to determine the problem is included in each table for the respective error indicator. Entry into these tables should only be initiated after executing any recommended actions suggested in the "Troubleshooting" section of the *IBM PS/2 Server Administrator's Guide.*

Table 2-1. POST Beep Code Indication

POST Beep Code	Failure Description	Most Likely FRU	Suggested Diagnostic Test*
1-1-3	CMOS Write/Read Test Failure	CMOS on Processor (CPU) Board	Run CPU.BAT
1-1-4	BIOS ROM Checksum Failure	Processor (CPU) Board	Run CPU.BAT
1-2-1	Programmable Interval Timer	Processor (CPU) Board - RTC	Run CPU.BAT
1-2-2	DMA Initialization	Processor (CPU) Board	Run DMA.ABS test and CPU.BAT
1-2-3	DMA Page Register Write/Read	Processor (CPU) Board or possibly a Micro Channel adapter	Run DMA.ABS test and CPU.BAT. Also, run the diagnostics supplied with your Micro Channel adapter
1-3-3 through 2-4-4	First 64K RAM Chip or Data Line	Memory Board or processor (CPU) Board	Run IPMEM.ABS, CPU.BAT, and MEMORY.BAT
3-1-1	Slave DMA Register	Micro Channel Adapter	Run DMA.ABS and CPU.BAT
3-1-2	Master DMA Register	Processor (CPU) Board or Possibly Micro Channel adapter	Run DMA.ABS and CPU.BAT. Also, run the diagnostics supplied with your Micro Channel adapter

*** Tests indicated in this column are found on the IBM PS/2 Server Diagnostics Monitor diskette**

Table 2-1. POST Beep Code Indication (Continued)

POST Beep Code	Failure Description	Most Likely FRU	Suggested Diagnostic Test*
3-1-4	Slave Interrupt Mask Register	Processor (CPU) Board	Run CPU.BAT
3-2-4	Keyboard Controller	Processor (CPU) Board	Run CPU.BAT
3-3-4	Screen Memory	Processor (CPU) Board or Video Graphics Adapter	Run Memory Group Tests
3-4-1	Screen Initialization	Processor (CPU) Board or Video Graphics Adapter	Check VGA - Run CPU.BAT
3-4-2	Screen Retrace	Processor (CPU) Board - Video Graphics Adapter	Check VGA - Run CPU.BAT
4-2-1	Timer Tick Interrupt	Processor (CPU) Board	TIMER.ABS and CPU.BAT
4-2-2	Shutdown	Possibly One or More System Boards	Check RMP on 295. Run SYSTEM.BAT
4-2-3	Gate A20 Failure	Processor (CPU) Board	Run CPU.BAT
4-2-4	Unexpected Interrupt In Protected Mode	Possibly One or More System Boards (processor, memory, IDC)	Run SYSTEM.BAT

*** Tests indicated in this column are found on the IBM PS/2 Server Diagnostics Monitor diskette**

Table 2-1. POST Beep Code Indication (Continued)

POST Beep Code	Failure Description	Most Likely FRU	Suggested Diagnostic Test*
4-3-3	Interval Timer Channel 2 Tests	Processor (CPU) Board	Run RTC.ABS and CPU.BAT
4-3-4	Time-of-day Clock Test Failure	Processor (CPU) Board	Run TIMER.ABS and CPU.BAT
4-4-1	Serail Port Test Failure	Processor (CPU) Board	Run SERIAL.ABS and CPU.BAT
4-4-2	Parallel Port Test Failure	Processor (CPU) Board-	Run PRINTER.ABS and CPU.BAT
4-4-3	Math Co-processor Test failure	Processor (CPU) Board	Run CPU.BAT
low tone 1-1-2	Processor Module Select Failure	Processor (CPU) Board	RunCPU.BAT and SYSTEM.BAT
low tone 1-1-3	Extended CMOS RAM	Processor (CPU) Board	Run CMOS.ABS and CPU.BAT
low tone 2-1-1	No IP-Bus RAM Detected In System	Memory Board or Processor (CPU)	MEMORY.BAT, IP.BAT and CPU.BAT
low tone 2-1-2	Bad Server Shutdown Byte (No LCD Message)	Possibly One or More System Boards (processor, memory, IDC)	Run CPU.BAT
low tone 2-1-3	IP-Bus Failure:	Processor (CPU) or Memory Board	Run IP.BAT and MEMORY.BAT

*** Tests indicated in this column are found on the IBM PS/2 Server Diagnostics Monitor diskette**

Table 2-1. POST Beep Code Indication (Continued)

POST Beep Code	Failure Description	Most Likely FRU	Suggested Diagnostic Test*
low tone 2-1-4	A BIOS image RAM ECC Check	Processor (CPU) Board	Run Processor Board Group Tests
low tone 2-2-1	Bad Mailbox Registers(s) - No LCD Message	Processor (CPU) Board or Memory Board	Run Processor and Memory Group Tests
low tone 2-2-3	Bad Diagnostic Monitor Entry Vector - Missing Or Misprogrammed	Processor (CPU) Board	(NO TEST - will not BootDIAGMON)
low tone 2-2-4	RAM Image Write Protection Failure	Processor (CPU) Board-	Run CPU.BAT
low tone 2-3-1	FP/AP Parameter Transfer Area Mismatch	Processor (CPU) Board	Run CPU.BAT, IP.BAT and MEMORY.BAT
low tone 2-3-2	Not Enough Memory For AP To Operate	Processor (CPU) or Memory Board	Run CPU.BAT and MEMORY.BAT
low tone 2-3-3	Bad Processor Module	Processor (CPU) Board	Run CPU.BAT
low tone 2-3-4	Bad Procesor Module Offset or Memory Size	Processor (CPU) or Memory Board	Run CPU.BAT
low tone 2-4-1	Bad S486 Internal Cache RAM	Processor (CPU) or Memory Board	Run CPU.BAT
low tone 2-1-3	IP-Bus Failure:	Processor (CPU) or Memory Board	Run CPU.BAT and MEMORY.BAT

*** Tests indicated in this column are found on the IBM PS/2 Server Diagnostics Monitor diskette**

Table 2-1. POST Beep Code Indication (Continued)

POST Beep Code	Failure Description	Most Likely FRU	Suggested Diagnostic Test*
low tone 2-4-2	Bad 486 "487" Internal Co-processor	Processor (CPU) Board	Run CPU.BAT
low tone 2-4-3	Bad Server Memory Mapper	Processor (CPU) Board	Run CPU.BAT and MEMORY.BAT

*** Tests indicated in this column are found on the IBM PS/2 Server Diagnostics Monitor diskette**

Table 2-2. POST Errors Displayed On Monitor

Code	Server Console Message	Most Likely FRU	Suggested DIAGMON Diagnostic Test	Suggested Field Maintenance Tests
103	Processor Module Failure	Processor (CPU) Board	Run TIMER.ABS and CPU.BAT	MEMORY/CPU and SYSTEM UNIT
104	Processor Module Failure	Processor (CPU) Board	Run PIC.ABS, COU.BAT and SYSTEM.BAT	MEMORY/CPU and SYSTEM UNIT
108	Processor Module Failure	Processor (CPU) Board	Run TIMER.ABS, CPU.BAT and SYSTEM.BAT	MEMORY/CPU and SYSTEM UNIT
110	Memory Parity Error At Address	Memory Board	Run DRAM.ABS and MEMORY.BAT	MEMORY/CPU and SYSTEM UNIT
111	I/O Adapter Parity Error At Address	Processor (CPU) Board or Possibly Micro Channel Area	Run CPU.BAT and MCA.BAT	SYSTEM UNIT
112	Watchdog Timeout	Processor (CPU) Board	Run CPU.BAT	SYSTEM UNIT
113	DMA Bus Timeout	Processor (CPU) Board	Run DMA.ABS and CPU.BAT and SYSTEM.BAT	MEMORY/CPU
114	Option ROM Checksum Failure	Memory Board	Run CPU.BAT	VIDEO GRAPHICS ARRAY

Table 2-2. POST Errors Displayed On Monitor (Continued)

Code	Server Console Message	Most Likely FRU	Suggested DIAGMON Diagnostic Test	Suggested Field Maintenance Tests
130	Processor Module failure	Processor (CPU) Board	Run CPU.BAT	MEMORY/CPU
132	Processor Module failure	Processor (CPU) Board	Run CPU.BAT	MEMORY/CPU
161	Real Time Clock Failure (Battery)	Processor (CPU) Board or battery	RTC.ABS and CPU.BAT	MEMORY/CPU and SYSTEM UNIT
162	Real Time Clock Failure (CRC)	Processor (CPU) Board	RTC.ABS and CPU.BAT	———
163	Time And Date Not Set	Processor (CPU) Board	RTC.ABS and CPU.BAT	———
164	Invalid Configuration Information (Memory)	Processor (CPU) Board or Memory Board	CU.BAT and MEMORY.BAT	MEMORY/CPU
165	Invalid Configuration Information (Micro Channel Adapter)	Micro Channel adapter - first check server configuration using reference disk	MCA.BAT	Run manufacturers Mico Channel adapter diagnostic(s)
166	Invalid Configuration Information (Micro Channel Adapter)	Micro Channel adapter - first check server configuration using reference disk	MCA.BAT	Run manufacturers Mico Channel adapter diagnostic(s)

Table 2-2. POST Errors Displayed On Monitor (Continued)

Code	Server Console Message	Most Likely FRU	Suggested DIAGMON Diagnostic Test	Suggested Field Maintenance Tests
167	Invalid Configuration Information (Diskette)	Processor (CPU) Board - first check that the drive itself is properly configured and working	Run FDC.BAT	Check that the drives are properly configured and working
168	Invalid Configuration Information (Processor Module)	Processor (CPU) Board -first check that the system board itself is properly configured using reference disk	Run CPU.BAT	Check that the processor boards are properly configured using the reference disk
201	General Memory Failure	Processor (CPU) or Memory Board	Run CPU.BAT and MEMORY.BAT	MEMORY/CPU

Table 2-2. POST Errors Displayed On Monitor (Continued)

Code	Server Console Message	Most Likely FRU	Suggested DIAGMON Diagnostic Test	Suggested Field Maintenance Tests
202	General Memory Failure	Processor and/or Memory Board	Run CPU.BAT and MEMORY.BAT	MEMORY/CPU
203	General Memory Failure	Processor and/or Memory Board	Run CPU.BAT and MEMORY.BAT	MEMORY/CPU
301	Keyboard Failure	Keyboard, Processor (CPU) Board or IP Backplane	Run CPU.BAT	KEYBOARD
304	Keyboard Failure	Keyboard, Processor (CPU) Board or IP Backplane	Run CPU.BAT	KEYBOARD
306	Keyboard Failure	Keyboard, Processor (CPU) Board or IP Backplane	Run CPU.BAT	KEYBOARD
307	Stuck Key Failure	Keyboard, Processor (CPU) Board or IP Backplane	Run CPU.BAT	KEYBOARD
601	Diskette Subsystem Failure	IP backplane; also oheck diskette drive, and/or diskette drive cable	Run FDC.ABS and SYSTEM.BAT	DISKETTE DRIVE

Table 2-2. POST Errors Displayed On Monitor (Continued)

Code	Server Console Message	Most Likely FRU	Suggested DIAGMON Diagnostic Test	Suggested Field Maintenance Tests
602	Diskette Drive 0 Failure	Processor (CPU) Board; also check diskette drive, and/or diskette drive cable	Run FDC.BAT	DISKETTE DRIVE
901	Parallel Port Failure	Processor (CPU) Board	Run PRINTER.ABS and CPU.BAT	SYSTEM BOARD PARALLEL PORT
1101	Serial Port Failure	Processor (CPU) Board	Run SERIAL.ABS and CPU.BAT	SYSTEM BOARD SERIAL PORT
1780	Fixed Disk Drive 0 Failure	IDC Board; also check cables or disk drive 0	Run SCSI.BAT	———
1781	Fixed Disk Drive 1 Failure	IDC Board; also check cables or disk drive 1	Run SCSI.BAT	———
1782	Fixed Disk Reset Failure	IDC Board; also check cables or disk drive	Run SCSI.BAT	———
1790	Fixed Disk Drive 0 Failure	IDC Board; also check cables or disk drive 0	Run SCSI.BAT	———

Table 2-2. POST Errors Displayed On Monitor (Continued)

Code	Server Console Message	Most Likely FRU	Suggested DIAGMON Diagnostic Test	Suggested Field Maintenance Tests
1791	Fixed Disk Drive 1 Failure	IDC Board; also check cables or disk drive 1	Run SCSI.BAT	———
2401	Video Failure	Processor (CPU) Board	Run CPU.BAT	VIDEO GRAPHICS ARRAY
8602	Auxiliary Device Failure	Processor (CPU) Board or Mouse Device; check that the mouse is plugged into the correct connector	Run CPU.BAT	———
9902	FP/AP Switches Have Same Setting	Processor (CPU) Board; also check that the processor's AP/FP switch is set to correct location	Run CPU.BAT	———
9903	System Configuration Mismatch	Processor (CPU) Board; also check server configuration with reference disk	Run MEMORY.BAT and CPU.BAT	———

Table 2-2. POST Errors Displayed On Monitor (Continued)

Code	Server Console Message	Most Likely FRU	Suggested DIAGMON Diagnostic Test	Suggested Field Maintenance Tests
9904	AP Error During POST. AP Not Useable	Processor (CPU - AP Module)	Run CPU.BAT and MEMORY.BAT	MEMORY/CPU
9906	Invalid System Memory Configuration	Processor (CPU) Board; also check server configuration using reference disk	Run CPU.BAT and MEMORY.BAT	MEMORY/CPU
9907	CPU Disk Interrupt	Processor (CPU) Board or Mico Channel Area; also check server configuration using reference disk	Run CPU.BAT and MEMORY.BAT	———
9910	Disk Controller Board Failure	IDC Board	Run SCSI.BAT	———
9911	Hard Disk Drive Failure	IDC Board cable or disk drive	Run SCSI.BAT	———
9912	Hard Disk Configuration Error	IDC Board; also check server configuration using reference disk	Run SCSI.BAT	———
9913	Security Key Not Installed	Processor (CPU) Board; also verify key is installed	Run CPU.BAT and SCSI.BAT	———

Table 2-2. POST Errors Displayed On Monitor (Continued)

Code	Server Console Message	Most Likely FRU	Suggested DIAGMON Diagnostic Test	Suggested Field Maintenance Tests
9920	IP Slot Configuration Mismatch	IDC Board or RMP Board; also check server configuration using reference disk	Run SCSI.BAT and IPRMP.BAT	———
9921	IP Slot Configuration Mismatch	Processor (CPU) Board; also check server configuration using reference disk	CPU.BAT	———
9922	Invalid IP Slot Configuration Data	Processor (CPU) Board; also check server configuration using reference disk	CPU.BAT	———

Table 2-2. POST Errors Displayed On Monitor (Continued)

Code	Server Console Message	Most Likely FRU	Suggested DIAGMON Diagnostic Test	Suggested Field Maintenance Tests
9923	RMP In Slot Not Useable	RMP; also check server configuration using reference disk	Run RMP.BAT	MEMORY/CPU
9925	Cache Subsystem Not Useable	Processor (CPU) Board	Run CPU.BAT	———
9926	Memory Error Detection Not Useable	Memory Board	Run MEMORY.BAT	MEMORY/CPU
9927	FP/APModule Parameter Discrepancy	Processor (CPU) Board	Run CMOS.ABS and CPU.BAT	———
9928	FP/APModule Parameter Discrepancy	Processor (CPU) Board	Run CMOS.ABS and CPU.BAT	———
9929	Boot CPU Handoff Failed	Processor (CPU) Board	Run CPU.BAT	———
9930	Boot CPU Configured For Non-zero Base	Processor (CPU) Board	Run CPU.BAT; also check server configuration	———
9931	CMOS Configuration Revision Mismatch	Processor (CPU) Board	Run CPU.BAT; also check server configuration	———
9950	External Cache Disabled	Processor (CPU) Board	Run CPU.BAT	———

Table 2-2. POST Errors Displayed On Monitor (Continued)

Code	Server Console Message	Most Likely FRU	Suggested DIAGMON Diagnostic Test	Suggested Field Maintenance Tests
9951	Internal and External Cache Disabled	Processor (CPU) Board	Run CPU.BAT	———
9952	Total IP RAM limited	Memory or Processor (CPU) Board	Run CPU.BAT and MEMORY.BAT	———
9980	2-bit ECC Error At Address	Memory Board	Run MEMORY.BAT	MEMORY/CPU
9981	IP Bus Grant Timeout Error	Possibly One or More System boards (processor, Bemory, IDC)	Run CPU.BAT and SYSTEM.BAT	SYSTEM UNIT
9982	IP Bus Data Strobe Timeout	Possibly One or More System boards (processor, Bemory, IDC)	Run SYSTEM.BAT	SYSTEM UNIT and MEMORY/CPU
9983	IP Bus Parity Error (Responder Report)	Possibly One or More System boards (processor, Bemory, IDC)	Run CPU.BAT, MEMORY.BAT and SCSI.BAT	SYSTEM UNIT and MEMORY/CPU
9984	IP Bus Parity Error (Local Report)	Possibly One or More System boards (processor, Bemory, IDC)	Run CPU.BAT, MEMORY.BAT and SCSI.BAT	SYSTEM UNIT and MEMORY/CPU

Table 2-2. POST Errors Displayed On Monitor (Continued)

Code	Server Console Message	Most Likely FRU	Suggested DIAGMON Diagnostic Test	Suggested Field Maintenance Tests
9985	Intermittent NMI Occurred	Possibly One or More System boards (processor, Bemory, IDC)	Run CPU.BAT, MEMORY.BAT and SCSI.BAT	SYSTEM UNIT and MEMORY/CPU

Table 2-3. SCSI Error Keys

Code	Server Console Message	Most Likely FRU	Suggested DIAGMON Diagnostic Test	Suggested Field Maintenance Tests
01	Recovered Error - Recovery Action Had To Be Performed By Disk Drive	Refer to Error Log Entry (Event Log) Available Through PDA Management Utility (if block address is different)	Run SCSI.BAT	———
02	Not ready. The Disk Drive Cannot Be Accessed	Operator intervention Is Required. Possible Power Or Cabling Problems	None (check disk configuration via the reference disk)	———
03	Medium Error. - Command Terminated Due To Media Flaw Or By An Error In The Recorded Data	Hard disk drive	Run SCSI.BAT	———

Table 2-3. SCSI Error Keys (Continued)

Code	Server Console Message	Most Likely FRU	Suggested DIAGMON Diagnostic Test	Suggested Field Maintenance Tests
04	Hardware Error. A Nonrecoverable Hardware Error Was Encountered (e.g., disk drive failure, parity error, etc.)	Hard disk drive	Run SCSI.BAT	———
05	Illegal Request - An Illegal Parameter In The Command Or In The Additional Required Parameters	Hard disk drive	Run SCSI.BAT	———
FF	Drive Unit Attention - Disk Drive Cannot Be Accessed	Hard disk drive (operator intervention is required)	Run SCSI.BAT (check disk configuration via the reference disk)	———

Table 2-4. Operating Errors (LCD)

Code	Server Console Message	Most Likely FRU	Suggested DIAGMON Diagnostic Test	Suggested Field Maintenance Tests
System Error Message: NMI	PS/2 Generic Sources, Server Specific Sources And IP-Bus NMI Reported In RMP Log and/or NVRAM (Error Log)	Refer To Error Log	Run CPU.BAT and SYSTEM.BAT	Exercise The PS/2 Field Maintenance System Tests Diskette (MEMORY/CPU; Error Log utility - NMI errors)).
Boot Failure Error(s) In POST	Errors Detected During POST. Recorded on RMP Event Log.	Refer To Error Log	Run CPU.BAT and SYSTEM.BAT	———
ECC Memory Error, Read Failure	Transient Error Occurs In The Read Path	Refer To ECC Error Log	Run MEMORY.BAT	Exercise The PS/2 Field Maintenance System Tests Diskette (MEMORY/CPU; Error Log utility - ECC errors)
ECC Memory Error Soft Failure	Memory Error Corrected By Writing (Scrub) Correct Data Back To Memory Over The Error.	Refer To ECC Error Log	Run MEMORY.BAT	Exercise The PS/2 Field Maintenance System Tests Diskette (MEMORY/CPU; Error Log utility - ECC errors)
ECC Memory Error Hard Failure	Memory Error Not Corrected Via Scrubbing	Refer To ECC Error Log	Run MEMORY.BAT	Exercise The PS/2 Field Maintenance System Tests Diskette (Error Log utility - ECC errors)
ECC Memory Error Tracking Stopped	Memory Error Tracking Stopped To Reduce Overhead	Refer To ECC Error Log	Run MEMORY.BAT	Exercise The PS/2 Field Maintenance System Tests Diskette v2.0 (Error Log utility - ECC errors)

Table 2-5. RMP IP Bus Tests

Test Description	FRU Tested	Suggested DIAGMONDiagnostic Test
IPRMP Tests The Bus Interface Between The Processor And RMP Boards. CPU Access To The RMP's CSR Registers And IP Bus Are Tested.	Processor (CPU) Board / RMP	Run RMP.BAT
Test RMPIP Bus Loopback Buffers. For Each Buffer A Series Of Double Word Patterns Are Written And Read Back For Verification.	Processor (CPU) Board / RMP	Run RMP.BAT

Chapter 3: Functional Description of DIAGMON Tests

Functional Description of DIAGMON Tests

Functional Overview

This chapter describes the functional aspects of the various diagnostic test routines that are exercised on the PS/2 Server 195 and 295 to isolate and determine which FRU to replace to return the server to normal operation. The diagnostic monitor is a ROM-based diagnostic program. Each diagnostic is divided up into module sections which can contain single or multiple tests. The diagnostic programs are written in assembly lanaguage or C, and they may be embedded in ROM (assembly language only) or loaded from an external device. When external they reflect the form of absolute files with an extension .ABS. The .ABS files allows the diagnostician to execute individual diagnostic tests within a particular board group. Whereas, the .BAT tests will run the complete set of tests prescribed within the board group. Each diagnostic is an executable file with an .ABS suffix.

Each processor module contains a diagnostic monitor. The Server can be started with a diagnostic monitor running on each processor module. Server system components and their respective diagnostic test modules are described in this chapter.

Processor (CPU) Board Group

The following diagnostic tests are available on the diagnostic monitor for each processor (CPU) board group :

- CMOS
- CPUMEM
- Dual CPU
- Peripheral Interrupt Controller (PIC)
- Parallel Port Printer
- Real Time Clock (RTC)
- Serial Controller
- Microchannel System Timer

A functional description of each test is provided below:

CMOS

A series of memory tests that execute on the 8K of CMOS RAM on the processor board. The test consists of three major sections: RAM pattern, RAM address, and RAM MATS + tests.

Section 1 of the diagnostic is the single pattern test. A single pattern is written to each CMOS RAM location. Then the entire CMOS RAM is read back for verification. The steps of the test are listed as follows:

1. **Initialize test data to 0x00**
2. **Fill each extended CMOS location with test pattern**
3. **Read each location and compare against expected data**

4. **Increment test data by 0x55. If all possible patterns have been exhausted, then terminate test. Otherwise go to step 2.**

Section 2 of the diagnostic is the address pattern test. A unique value is written to each memory location and then read back for verification. The steps of the test are listed as follows :

1. **Write address value of CMOS location being tested to that location.**

2. **Go to step 1. Continue until entire extended CMOS RAM has been filled.**

3. **Read back each CMOS location and compare against expected data for correctness.**

4. **Continue until entire CMOS RAM has been verified.**

Section 3 of the diagnostic executes the MATS+ algorithm on the CMOS RAM. The steps of the test are :

1. **Write zero to locations 1 to n-1.**

2. **Write one to location 0.**

3. **Test for zero at locations 1 to n-1.**

4. **Then write ones from locations 1 to n-1.**

5. **Test for ones at locations 1 to n-1.**

6. **Then write zero at locations 0 to n-2.**

7. **Test for ones at location n-1.**

8. **Test for zero at location 0.**

9. **Write zero to location n-1.**

In the event of an error, the following message is displayed :

!ERROR: Address XX Expected XX Observed XX XOR XX

The Address value is the offset of the CMOS location where the error occured. The Expected value is the data that was written to the CMOS location. The Observed value is the data read back from the CMOS location. The XOR value is the exclusive OR of the expected and observed data values.

CPUMEM

Tests the memory mapping and cache snooping features of the processor (CPU) board. The tests consist of four major sections: Memory Mapping Offset and Size, Cache Snoop, Cache Snoop with adjusted offset, and Snoop with non-adjusted offset. The sections of this diagnostic are as follows:

Section 1 (Memory Mapping Offset and Size Test) tests the read address offset hardware. The mapping of the CPU addresses as they are affected by various values in the offset and offset size registers and enabling offset (mapping of CPU memory addresses enabled). The offset is the base address of the CPU's window into IP bus memory. The offset size value defines the top address of the CPU's window into IP bus memory. With offset enabled, the CPU memory addresses are affected by the offset and offset size registers. After each set of offsets and offset sizes are tested, IP bus memory is read back to ensure that the data is correct. The steps of the test are listed as follows :

1. **Set up memory size.**
2. **Enable RMW on the memory board.**
3. **Initialize a section of memory with an incrementing pattern.**
4. **Write values to offset and offset size registers.**
5. **Read back memory location and compare against known data**
6. **Increment to next location**
7. **Repeat steps 5 and 6 until memory has been verified.**

In the event of an error, the following message is displayed to the use

!ERROR: Address XX Offset XX OffStSz XX Mem_Size XX Expected XX Observed XX XOR XX

The Address value is the memory location where the failure occured. The offset value is the data contained in the offset register at the time of the failure. The OffStSz value is the data in the offset size register. The Mem Size is the data in the memory size register. The expected value is the correct data. The observed value is the actual data read back from the memory location. The XOR value is the exclusive OR between the expected and observed data values.

Section 2 tests write and read access through the snoop tag RAM diagnostic space. The steps of the test are:

1. **Enable self snoop mode.**
2. **Set up offset, offset size and memory size registers for 128 Mbytes.**
3. **Initialize snoop tag RAM.**
4. **Calculate physical address in system memory based on a cache offset.**
5. **Write a value of 0 to physical address of system memory that was calculated in step 4.**
6. **Read back snoop tag RAM.**
7. **Increment to next snoop tag location. Repeat until all locations have been verified.**

!ERROR: Address XX Entry XX Expected XX Observed XX XOR XX

The Address value is the system memory location that matches the snoop tag entry. The Entry value is the 1K byte section of the snoop RAM where the failure occured. The Expected value is the correct data. The Observed

value is the actual data that was read back from the snoop RAM. The XOR value is the exclusive OR between the expected and observed data values.

Section 3 tests write and read access to the snooping tag RAM in conjunction with varying offset, offset size and memory size values. The offset line is enabled, so the mapping of CPU memory addresses are affected by the offset and offset size registers. Therefore , the write offset hardware is also tested. The offset and offset size values are adjusted by subtracting a value of one megabyte. The steps of the test are:

1. **Enable self snoop mode.**
2. **Initialize snoop tag RAM.**
3. **Set up offset, offset size and memory size registers on the processor board.**
4. **Write a value of 0 to main memory.**
5. **Read back snoop tag RAM.**
6. **Increment to next location.**
7. **Repeat steps 5 and 6 until all the snoop tag RAM has been read back and verified.**

In the event of an error, the following message is displayed :

!ERROR: Address XX Offset XX OffStSz XX Mem Size XX Expected XX Observed XX XOR XX

Section 4 tests the write and read access to the snooping tag RAM in conjunction with varying offset and offset size values. The offset line is enabled and the write offset hardware is tested. The memory size is held constant at 128 Mbytes. The offset and offset sizes in this test are not adjusted. The steps of the diagnostic are :

1. **Enable self snoop mode.**
2. **Set up memory size for 128 Mbytes.**

3. **Initialize snoop tag RAM.**

4. **Set up offset and offset size registers.**

5. **Calculate system memory physical offset based on snoop cache RAM offset.**

6. **Write to system memory address calculated in step 5.**

7. **Read back snoop tag RAM.**

8. **Increment to next snoop tag location.**

9. **Repeat steps 7 and 8 until entire snoop tag RAM has been read back and verified.**

In the event of an error, the following message is displayed :

!ERROR: Address XX Offset XX OffStSz XX Mem_Size XX Expected XX Observed XX XOR XX

DUALCPU

The DUALCPU module tests the capability of each processor to become master of the IP bus (memory board arbiter). It exercises the memory board arbiter by executing instructions simultaneously within both processors. The diagnostic consists of three sections :

- IP bus byte transfer test
- IP bus word transfer test
- IP bus double-word transfer test

There are two cases that should be tested. They are as follows:

1. **Both processor boards executing diagnostic from external cache.**

2. **Foreground processor executing diagnostic from system memory with caching enabled and background processor executing from external cache.**

Test case two is the simplest to set up and verifies that the two processors can execute code simultaneously. The steps are :

1. **Load diagnostic into the processor's external cache.**

2. **Execute the GO2 command. This command copies the diagnostic from one processor's external cache to the other processor's cache and starts execution.**

Test case two verifys that caching and snooping is working properly on the foreground processor. The purpose of also having the background processor executing is to increase traffic across the IP bus and exercise the memory board's arbiter.

Section 1 of the diagnostic executes a series of byte accesses across the IP bus to system memory. The steps of the test are :

1. **Initialize test pattern to 0x00.**

2. **Write a byte to memory.**

3. **Increment to next byte address.**

4. **Repeat steps 2 and 3 until section of memory under test has been initialized.**

5. **Read back memory location and compare against known data.**

6. **Increment to next byte address.**

7. **Repeat steps 5 and 6 until entire section of memory under test has been read back and verified.**

8. **Increment data by 0x55. Go to step 2. Repeat until all data patterns are used.**

Section 2 of the diagnostic executes a series of word transfers across the IP bus to system memory. The steps of the test are :

1. **Write a word to memory.**
2. **Increment to next word address.**
3. **Repeat steps 1 and 2 until section of memory under test has been initialized.**
4. **Read back memory location and compare against known data.**
5. **Increment to next word address.**
6. **Repeat steps 5 and 6 until the entire section of memory has been read back and verified.**
7. **Increment test data by 0x5555.**
8. **Go to step 2. Repeat until all data patterns are used.**

Section 3 of the diagnostic executes a series of double word transfers across the IP bus to system memory. The steps of the test are listed as follows :

1. **Initialize test pattern to 0x00000000.**
2. **Write a double word to memory.**
3. **Increment to next double word address.**
4. **Repeat steps 2 and 3 until section of memory under test has been initialized.**
5. **Read back memory location and compare against known data.**
6. **Increment to next double word address.**

7. **Repeat steps 5 and 6 until the entire section of memory has been read back and verified.**

8. **Increment test data by 0x55555555. Go to step 2. Repeat until all test patterns have been used.**

Peripheral Interrupt Controller (PIC)

A multiple interrupt level test that verifies that the processor board can generate and handle interrupts from different sources and priority levels. The interrupt instructions of this diagnostic test originate from three sources: Microchannel system timer, Real Time Clock alarm, and system expansion slots 3 thru 5. The system timer interrupt is serviced by the master controller and Real Time Clock and the slave controller.

The interrupts used in the diagnostic are from three sources. They are the Microchannel system timer, the real time clock alarm and system expansion slots 3 thru 5. The system timer interrupt is serviced by master controller and the real time clock and expansion slot interrupts are handled by the slave controller. The steps of the test are as follows :

1. **Start countdown operation on system timer.**

2. **Set up real time clock alarm and start update operation.**

3. **Generate interrupt from the expansion slot by reading an empty FIFO on one of the SIOP channels.**

The expansion slot interrupt is generated continuously while the system timer and real time clock are running. When interrupts from the system timer and real time clock are recieved, part of the servicing is restarting them so that operation continues as long as the expansion slot interrupt is still being generated.

At the end of the test, a count is checked for how many of each interrupt was generated and serviced. An error condition exists if any of these counts are zero. In that case the following message is displayed :

!ERROR: Number of Interrupts: Timer XX RTC XX Expansion Slot XX

The Timer value is the number of interrupts that were generated by the system timer and serviced by the master interrupt controller. The RTC value is the number of interrupts that were generated by the real time clock alarm and handled by the slave interrupt controller. The Expansion Slot value is the number of interrupts that were generated by the IDC controller board and serviced by the slave interrupt controller.

Parallel Port Printer

Tests the functionality of the parallel port which is integrated into the Intel 82304 Microchannel chip set. Consists of a single Parallel Port Pattern test.

A single pattern is written to data port. Then the data port is read back for verification. The steps of the test are :

1. **Initialize test data to 0x00.**
2. **Write data port with test pattern.**
3. **Read back data port and compare against expected data.**
4. **Increment test data by 0x55. If all possible patterns have been exhausted, then terminate test. Otherwise go to step 2.**

In the event of an error, the following message is displayed to the user:

!ERROR: Address XX Expected XX Observed XX XOR XX

The Address value is the location of the data port. The Expected value is the correct data that was supposed to have been read back from the data port. The Observed value is the actual data that was read back. The XOR value is the exclusive OR between the expected and observed data values.

- **Each base address of the parallel port is also tested. The various base addresses are 0x3BC, 0x378 and 0x278. Each is configured through POS register 0x102 during a system board setup.**

!ERROR: Address XX Expected XX Observed XX XOR XX

The Address value is the memory location where the failure occured. The Expected value is the correct data that was suppose to have been read back from memory. The Observed value is the data that was actually read back from memory. The XOR value is the exclusive OR between the expected and observed data values.

In a case where the two processors are not communicating properly, the following message is displayed :

!ERROR: Interrupt Not Pending

Microchannel Real Time Clock (RTC)

Tests the functionality of the DS/285 real time clock chip and how it interfaces with the Intel 82304 local I/O support chip. All tests in this diagnostic are specfic to the RTC chip. It functions arew not driven or controlled by the 82304 chip support logic.

The diagnostic contains six sections. The sections of the diagnostic are :

- Section 1: RTC 2 Timer Test
- Section 2: RTC Daylight Savings Test
- Section 3: RTC Flags Test
- Section 4: RTC Time Set Test
- Section 5: RTC Update in Progress Test
- Section 6: RTC Update Test

Section 1 sets up a countdown operation on the tone generator timer and then initiates an update operation on the RTC (Real Time Clock). This is to ensure that the RTC and the tone generator can operate in conjunction with each other. The steps of the test are :

1. **Disables the RTC update operation.**
2. **Enable the oscillator but hold the countdown chain in reset.**

3. **Disable channel check, parity check, speaker data and the tone generation timer.**

4. **Program the tone generator timer. Set timer rate, read and write counter bits, mode 3 counter mode and the binary counter.**

5. **Load channel 2 count register with timer data.**

6. **Enable update operation on the RTC.**

7. **Poll the UIP bit on RTC register A and wait for update operation to complete. Bit will go to 0.**

8. **Enable the tone generation timer gate.**

9. **Poll bit 0 of system control port B and wait for time countdown to complete. Bit will go to 0.**

In the case of an error, the following message is displayed :

!ERROR: Address XX RTC Flag XX Address XX Timer Flag XX

The Address values are the IO ports of the RTC registers and the tone generator status register. The RTC and Timer Flag values are the data that is read out of the respective status registers.

Section 2 ensures that the daylight savings time feature of the RTC is functioning properly. The steps of the test are :

1. **Disable the RTC oscillator, but hold it in reset.**

2. **Enable RTC and set operation mode.**

3. **Disable RTC update and daylight savings time operation. Set up for binary data format and 24 hour mode.**

4. **Set time, calendar and alarm information by writing the appropriate RAM bytes.**

5. **Enable the update and daylight savings time operation.**

6. **Poll the UIP bit on RTC register A to wait for operation to complete.**

7. **Disable the update operation. Read back time information from RAM and verify that time was set properly and that the daylight savings time operation was completed. Data read back from RAM is compared with known data for verification.**

In the case of an error , the following message is displayed to the user.

!ERROR: Expected: Hour XX Minutes XX Seconds XX Observed: Hour: XX Minutes XX Seconds XX

The above data displays the expected and actual data after the operation has been completed.

Section 3 ensures that the periodic, alarm and update ended interrupts are functioning. The RTC is started and then the state of each of the interrupts are checked on a status register. Note that this is strictly a poll for interrupt test. The processor board interrupt controller is not involved in this test. The steps of the test are as follows :

1. **Disable the RTC oscillator and hold it in reset.**

2. **Enable RTC and set operation mode.**

3. **Disable update operation.**

4. **Set time, calendar and alarm information by writing the appropriate RAM bytes.**

5. **Enable the update operation.**

6. **Poll the UIP bit on RTC register A to wait for operation to complete.**

7. **Disable the RTC update operation.**

8. **Read back RTC register C and check for the PF, AF and UF bits. These are the periodic, alarm and update ended interrupt status**

flags respectively. At the end of the test a one should appear in each of these bits.

In the event of an error, the following message is displayed :

!ERROR: Address XX Expected XX Observed XX

The Address value is the IO port from which the status was read. This value will be RTC register C. The Expected value is the correct data. A correct value can be 0x40, 0x20 and 0x10 which represent the PF, AF and UF bits respectively. The Observed data is the actual data that was read from the IO port. An incorrect value will be 0x00.

Section 4 tests the RTC set feature. An update operation is started on the RTC. A reading of the time that was set during the update is taken while the RTC is still in operation. Another reading is taken with the RTC disabled. This is to ensure that data is preserved while the RTC changes from one operation mode to another. In each state, there are two readings of the time taken at one second apart. The steps of the test are :

1. **Disable update operation.**
2. **Disable oscillator and hold in reset.**
3. **Enable RTC and set operation mode.**
4. **Enable update operation.**
5. **Read the seconds byte from the RTC RAM. Read seconds byte again and compare with first reading. If they match, an error exists because the RTC is not running.**
6. **Disable update operation.**
7. **Read the seconds byte from RTC RAM. Read the byte again and compare with first reading. If readings don't match, it indicates that the RTC has not been disabled.**

In the event of an error, the following message is displayed :

!ERROR: Address XX Old XX New XX

The Address value is the IO port from where the data was read. The Old value is the first reading of the RTC second byte and the New value is the second reading. Section 5 tests the RTC update in progress feature. The steps of the test are :

1. **Disable RTC update operation.**
2. **Disable oscillator and hold in reset.**
3. **Enable RTC and set operation mode.**
4. **Enable update operation.**
5. **Poll the UIP bit on RTC register A. Increment high counter variable if bit is active and the low counter variable if bit is low.**
6. **Compare software counters and ensure that both are not zero. If both are zero, then that means that there was no update activity at all.**

In the event of an error, the following message is displayed :

!ERROR: Address XX Expected XX Observed XX

The Address value is the IO port from where the data was read. The Expected value is the correct data. The Observed value is the actual data that was read back for verification.

Section 6 tests the time update feature of the RTC. The time on the RTC is set. Then a new time is set and an update operation is enabled. The time data bytes are then read back and checked for validity. The steps of the test are :

1. **Disable oscillator and hold in reset. Enable RTC update operation.**
2. **Write time and calendar bytes to RAM.**

3. **Enable update operation.**

4. **Poll UIP bit on RTC register A to ensure that update operation was completed.**

5. **Disable RTC update operation.**

6. **Read back time and calendar data from RAM.**

7. **Compare against known time and calendar data for verification.**

In the event of an error, the following message is displayed :

!ERROR: Expected day:mm:dd:yy:hr:min:sec Observed day:mm:dd:yy:hr:min:sec

The Expected value is the known time and calendar data. The Observed value is the time that was read back from the RTC RAM.

Serial Controller

Tests the functionality of the NS16550AF chip and support logic for the serial port which is integral within the Microchannel chip set. Consists of two diagnostic tests: Serial Register Pattern and Serial Internal Loopback.

Section 1 of the diagnostic is the single pattern test. A single pattern is written to a register on the NS16550AF. Then the register is read back for verification.

The steps of the test are :

1. **Initialize test data to 0x00.**

2. **Write register with test pattern.**

3. **Read back data port and compare against expected data.**

4. **Increment test data by 0x55. If all possible patterns have been exhausted, then terminate test. Otherwise go to step 2.**

The serial internal loopback test uses the chip's internal loopback mode to test all internal data paths. Each valid baud rate is tested.

The steps of the test are:

1. **Set serial controller chip into loopback mode.**
2. **Write character to transmit hold register.**
3. **Wait for data received by polling the DR bit on the line status register.**
4. **Read back receive buffer data and compare with correct data.**
5. **Disable loopback mode.**

The following message is displayed in the event of an error :

!ERROR: Address XX Expected XX Observed XX XOR XX

Address value is the I/O port location that was being read back. The Expected value is the correct data. The Observed value is the actual data that was read back. The XOR value is the exclusive OR between the expected and actual data.

Microchannel System Timer

This module tests the system timers and microchannel address decoding capability of the Intel 82309 address bus controller. The timers are integrated on the local I/O support chip. Consists of two diagnostic tests : System timer 0, and tone generator timer.

Section 1: System Timer 0 Test

Section 2: Tone Generator Timer Test

Section 1 tests system timer 0 by verifying that an interrupt can be generated on it and handled properly. The steps of the test are :

1. **Set the system timer count rate, read and write counter bits, mode 3 counter mode and the binary counter.**

2. **Write 0x00 to the channel 2 count registers.**

3. **Start system timer and wait for interrupt.**

4. **Wait for a software variable to become 1 as an indication that the interrupt occured and has been serviced. This verifies that the interrupt request was sent to the CPU properly via the INTR line and the interrupt serviced.**

If the interrupt does not occur, an error condition exists and the following message is displayed :

!ERROR: Timer 0 Interrupt Test: Interrupt NOT Recieved

Section 2 tests the microchannel tone generator timer. The steps of the test are :

1. **Disable channel and parity check and speaker data on the tone generator timer.**

2. **Set the countdown rate, read and write counter bits, mode 3 operating mode and the binary counter.**

3. **Write channel 2 count register with data.**

4. **Enable tone generator timer.**

5. **Poll bit 1 on system control port B. Mark whether signal is high or low.**

6. **Compare the high and low counts. They should be nonzero. If not, an error condition exists.**

In the event of an error, the following message is displayed :

!ERROR: Tone Generator Timer Test : Output Inactive

Memory Board Hardware Group

The following tests make up the Diagnostic Module for the memory board hardware group:

- IPMEM (IP Bus Timeout/Loopback Buffer Test)
- EDAC (EDC Chip Tests)
- DRAM (DRAM Tests)

As in the case of the Processor (CPU) board, each of the above files tests a functional block of the memory board. It is recommended that while testing, the memory board diagnostics be executed in the above order. With each diagnostic, more of the hardware is being tested and a more advance feature is utilized. IPMEM tests the IP bus interface between the processor and memory boards. EDAC tests the EDC chips on the memory board. DRAM tests the host access to the memory banks, but requires that the hardware that is tested by IPMEM and EDAC be functional.

IPMEM

IPMEM tests the IP bus interface between the processor and memory boards. This is done by testing access to the memory board's IP bus loopback buffers. Furthermore the IP bus timeout function is also tested.

The diagnostic contains two sections. The sections of the diagnostic are listed as follows:

- Section 1: IP Bus Timeout Test
- Section 2: IP Bus Loopback Buffer Test

Section 1 tests for an IP bus timeout. The steps of the test are :

1. **Write an invalid memory location.**
2. **Read the DSTO (Data Strobe Timeout) bit on the memory board's error status register.**

In the event that an error occurs, the following message is displayed :

!ERROR: Data Strobe Time Out did not occur on memory board

This error indicates that the DSTO bit did not go active as required when the invalid memory location was written.

Section 2 tests the memory board's IP bus data interface. For each buffer in the interface, a series of double word patterns 0x00, 0x55, 0xAA and 0xFF are used as test data. The steps of the test are :

1. **Initialize test data pattern to 0.**
2. **Write to buffer location.**
3. **Immediately read back the buffer and check against the test data.**
4. **Increment to next buffer location. Go to step 2.**
5. **If all buffer locations have been tested, increment the test data pattern by 0x55555555. Go to step 2.**
6. **If all test patterns have been used, then test is finished.**

!ERROR: Address XX Expected XX Observed XX XOR XX

The Address value is the location of the buffer where the error occured. The Expected value is the data that was written to the buffer. The Observed value is the actual data that was read back from the buffer. The XOR value is the exclusive OR between expected and observed data values.

Error Detection and Correction (EDAC)

EDAC tests the memory board Error Detection and Correction (EDC) chips. The diagnostic contains five sections. The sections of the diagnostic are :

- Section 1: Data Memory (RMW Off) Test
- Section 2: Checkbit Generation Test

- Section 3: EDAC Data Latch Loopback Test
- Section 4: Checkbit Memory Test
- Section 5: EDAC Error Correction Test

Section 1 is a data memory test with RMW off. This test is intended to determine the addressability of each bank. This test is not exhaustive in the sense that it only tests the first megabyte of each bank of memory.

Section 2 tests the checkbit generation, error detection and correction features of the EDAC chips.

Section 3 tests the data path loopback of the EDAC data latches. This verifes that the data in, data out and internal data paths are all functional.

For sections 1, 2 and 3, the following message is displayed in the event of an error.

!ERROR: Address XX Expected XX Observed XX XOR XX

The Address value is the register to which the data was extracted. This can be either a memory location , the upper or lower EDAC data latches or the syndrome register. The Expected value is the correct data. The Observed data is what was read back for verification. The XOR value is the exclusive OR between the expected and observed data.

Section 4 tests the checkbit memory. The testing scheme is as follows:

1. **Write data to memory that will generate the correct checkbit value.**
2. **The memory boards error status register is then checked for any errors (Data strobe timeout, bus grant timeout, or single and double bit errors). If an error is detected in the register, the checkbits are then read back from the syndrome register. If no error occurs, checkbit memory is operational.**

In the event of an error, the following message is displayed :

!ERROR: Address XX Status XX Syndrome XX

The Address value is the memory location where the failure occured. The Status value is the data from the memory board error status register. The Syndrome value is the data from the syndrome register.

Section 5 consists of four tests for the EDAC chips. These tests check the syndrome generation, error detection and correction logic of the EDAC chips. The tests are data single bit error, data double bit error, checkbits single bit error and detect no bit error. Each test is explained in the following paragraphs.

The Data Single Bit Error test introduces a single bit error into the data that is written into memory. Then data is read back with ECC enabled to verify that data has not been corrupted.

The Checkbits Single Bit Error test corrupts a bit in the checkbit data that is generated and writes the corrupt data to the EDAC diagnostic latches for testing.

The Detect No Bit Error ensures that the EDAC chips are functioning properly in normal operating mode. Normal in the sense that no bit errors are introduced into the data that is written to memory or the checkbit data that is generated. The steps of the test are listed as follows:

In the event of an error, the following message is displayed :

!ERROR: Address XX Expected XX Observed XX XOR XX

The Address value is the location in data memory where the error occured. The Expected value is the correct data. The Observed value is the data that was read back from memory. The XOR value is the exclusive OR between the expected and observed data. This message indicates that the data read back from memory was incorrect.

If an error occurs while checking the error and interrupt status, the following message is displayed :

!ERROR: Address XX Expected XX Observed XX

This message means that the value read from the syndrome register does not match the value that was generated beforehand. The Expected value is the generated one. The Observed value is the data read out of the syndrome register. The XOR value is the exclusive OR between the expected and observed data values.

Dynamic RAM Module

The Dynamic RAM (DRAM) module tests the host access to the DRAM banks. It is a level 2 and 3 diagnostic. The diagnostic contains nine sections. The sections of the diagnostic test are :

- Section 1: DRAM Address Pattern Test
- Section 2: DRAM Single Pattern Test
- Section 3: DRAM Data Uniqueness Test
- Section 4: DRAM Random Data Test
- Section 5: DRAM MATS+ Test
- Section 6: DRAM Marching Sequence Test
- Section 7: DRAM Checkerboard Test
- Section 8: DRAM Walking Ones Test
- Section 9: DRAM Walking Zeroes Test

Since DRAM is also a level 3 diagnostic, it can also be executed from IP bus memory. Below are the necessary monitor commands to execute the diagnostic out of IP bus memory. The steps are :

1. **MV**
2. **CACHE ON**

The previous commands will move the diagnostic to IP bus memory and then the CPU cache is enabled. To execute the diagnostic, type the command GO.

Section 1 of the diagnostic fills each memory location with the address of that memory location that is being tested. The steps of the test are :

1. **Write memory location with least significant byte of the memory address that is being tested.**
2. **Increment to next memory byte address.**
3. **Repeat steps 1 and 2 until entire memory has been filled and then go to step 4.**
4. **Read back memory location and compare against test data.**
5. **Increment to next memory byte address.**
6. **Repeat steps 4 and 5 until entire memory has been read back and checked.**

Section 2 of the diagnostic fills each memory location with a single pattern. The pattern used is 0xAA. The steps of the test are :

1. **Write memory location with test pattern.**
2. **Increment to next memory byte address.**
3. **Repeat steps 1 and 2 until entire memory has been filled and then go to step 4.**
4. **Read back memory location and compare against test data.**
5. **Increment to next memory byte address.**
6. **Repeat steps 4 and 5 until entire memory has been read back and checked.**

Section 3 of the diagnostic fills each memory location with the address of the location plus a step increment. The increment used in the test is 0x03. The steps of the test are :

1. **Initialized test data to step increment.**
2. **Write memory location with test pattern.**
3. **Increment test data by the step value.**
4. **Increment to next memory byte address.**
5. **Repeat steps 2 thru 4 until memory has been filled and then go to step 6.**
6. **Read back memory location and compare against test data.**
7. **Increment to next memory byte address.**
8. **Repeat steps 6 and 7 until entire memory has been read back and checked.**

Section 4 of the diagnostic is pseudo random data test. The random data is generated and then written and read back from memory. The steps of the test are :

1. **Generate single random value.**
2. **Write value to memory location.**
3. **Read back memory location and check against generated data value.**
4. **Increment to next memory byte address. Go to step 1.**
5. **Repeat steps 1 thru 4 until entire memory has been tested.**

Section 5 of the diagnostic executes a MATS+ algorithm on the memory banks. The steps of the test are :

1. **Write memory location with test pattern.**

2. **Increment to next memory byte address.**

3. **Repeat steps 1 and 2 until entire memory has been filled and then go to step 4.**

4. **Write location 0 with the complement of the test pattern.**

5. **Read back memory starting at location 1.**

6. **Read back memory starting at location 0. (a) Read back memory location and compare against test data. (b) Increment to next memory byte address. (c) Repeat steps 6a and 6b until entire memory has been read back and verified.**

Section 6 of the diagnostic executes the Ready algorithm on the memory banks. The steps of the test are :

1. **Start first test sequence. (a) Write double word location with 0. (b) Increment to next double word location. (c) Repeat steps 1a and 1b until entire memory has been initialized to 0. (d) Read back memory location and checked for 0. (e) Write memory location with 0xFFFFFFFF. (f) Read back memory location and check for data of 0xFFFFFFFF. (g) Write memory location with 0. (h) Read back memory location and check for data value of 0. (i) Write memory location with 0xFFFFFFFF. (j) Increment to next memory location. (k) Repeat steps 1d thru 1j until entire memory has been tested and then go to step 2.**

2. **Start second test sequence. (a) Read back memory location and check for data value of 0xFFFFFFFF. (b) Write memory location with value of 0. (c) Read back memory location and check for value of 0. (d) Write memory location with 0xFFFFFFFF. (e) Increment to next memory double word location. (f) Repeat steps 2a thru 2e until entire memory has been tested and then go to step 3.**

3. **Start third test sequence. The start address for this sequence begins at the top of memory.**

4. **Start fourth test sequence. The start address for this sequence begins at the top of memory.**

Section 7 of the diagnostic fills the memory banks with alternating patterns of 0xAA and 0x55. The steps of the test are :

1. **1) Write memory location with test data. (a) if an even address, use the data pattern 0xAA. (b) If an odd address, use the data pattern 0x55.**
2. **Increment to the next memory byte address.**
3. **Repeat steps 1 and 2 until entire memory has been initialized and then go to step 4.**
4. **Read back memory location and check against test data.**
5. **Increment to next memory byte address.**
6. **Repeat steps 4 and 5 until entire memory has been read back and verified.**

Section 8 of the diagnostic executes a walking ones test on the memory banks. The steps of the test are listed as follows:

1. **Write memory location with data value of 0.**
2. **Increment to next memory byte address.**
3. **Repeat steps 1 and 2 until entire memory has been initialized to 0 and then go to step 4.**
4. **Write each byte address bit to 1.**
5. **After each byte address bit has been written to 1, read back memory location and compare against test data for verification.**
6. **Repeat steps 4 and 5 until all the bits have been checked and then go to step 7.**

7. **Increment to next memory byte address.**

8. **Repeat steps 4 thru 7 until all memory locations have been tested.**

At the end of the test, the entire memory array is filled with ones.

Section 9 of the diagnostic executes a walking zeroes test on the memory banks. The steps of the test are :

1. **Write memory location with 0xFF.**

2. **Increment to next memory byte address.**

3. **Repeat steps 1 and 2 until entire memory has been initialized to 0xFF.**

4. **Write each address bit to 0.**

5. **After each address bit has been written to 0, read back memory location and compare against test data for verification.**

6. **Repeat steps 4 and 5 until all the address bits have been checked and then go to step 7.**

7. **Increment to next memory byte address.**

8. **Repeat steps 4 thru 7 until all memory locations have been tested.**

At the end of the test, the entire memory array will be filled with zeroes.

In the event of an error, the following message will be displayed to the user.

!ERROR: Address XX Expected XX Observed XX XOR XX

The Address value is the location in memory where the error occured. The Expected value is the correct data. The Observed value is the data that was read back from the memory location being checked. The XOR value is the exclusive OR between the expected and observed data values.

❖ **NOTE: All sections are byte write and read tests except for section 6 which is a double word write and read test.**

SCSI Board Hardware Group

The following tests make up the diagnostic module for the SCSI hardware group :

- IPSCSI (SCSI IP Bus Tests)
- SIOP (SCSI Processor Tests)
- SCSISCR (SCSI Single and Dual Channel Tests)

It is recommended that the tests for the SCSI board be executed in the above order because each diagnostic tests a more advance feature. IPSCSI tests the IP bus interface between the Processor and SCSI boards. The next test SIOP checks out the basic functionality of the SIOP chip. Finally SCSISCR tests out the SCSI board's interrupt generation, channel to channel communication and channel to device functionality via SCSI script operations.

IPSCSI Test

IPSCSI tests the IP bus interface between the Processor and the SCSI boards. Host access to the data paths of both channels A and B, the ID and LED registers are tested.

IPSCSI contains three sections. The sections of the diagnostic are listed as follows :

- Section 1: IP Bus Loopback Buffer Tests
- Section 2: SCSI ID Test
- Section 3: SCSI LED Test

Section 1 of the diagnostic tests host access to the primary and secondary IP bus buffers for both the A and B SIOP controller channels. Various pattern algorithms are used. They are :

1. Single Pattern
2. Unique Pattern
3. Alternating Patterns
4. Block Patterns
5. Walking Zeroes
6. Walking Ones

In the Single Pattern test, each buffer is tested with the double word patterns of 0x00, 0x55, 0xAA and 0xFF. Each buffer is accessed individually and then read back for verification. The steps of the test are :

1. **Initialize data pattern.**
2. **Write data pattern to buffer.**
3. **Read back the buffer and compare against expected data.**
4. **Set next data pattern.**
5. **Increment to next buffer location. Then go to step 2.**
6. **Repeat steps 2 thru 5 until all buffer locations have been tested.**

The Unique Pattern test fills each buffer with a unique pattern. The initial pattern used is 0x11111111 and is incremented by the same value for each buffer that is tested. All buffers are written and then read back individually for verification. The steps of the test are listed as follows :

1. **Initialize data pattern to 0x11111111.**
2. **Write to data buffer.**

3. **Increment data pattern by 0x11111111.**

4. **Increment to next data buffer. Go to step 2.**

5. **Repeat steps 2 thru 4 until all buffers have been filled.**

6. **Read data back from data buffer and compare against expected data.**

7. **Increment to next buffer address. Go to step 6.**

8. **Repeat steps 6 and 7 until all buffers have been read back.**

9. **Increment test data by 0x11111111. Go to step 2.**

The Alternating Patterns test fills each buffer with a pattern and then the next buffer with the complement of the first pattern until the entire set of buffers is filled with alternating patterns. The patterns used are double words of 0x00, 0x55, 0xAA and 0xFF. Finally the buffers are read back and checked for any data mismatches. The steps of the test are :

1. **Initialize test data pattern.**

2. **Write to data buffer. If the buffer address is even, write buffer with the unaltered data. If the buffer address is odd, write buffer with the complement of the test data.**

3. **Increment to next buffer location. Go to step 2.**

4. **Repeat steps 2 and 3 until all buffer locations have been written.**

5. **Read back data buffer and compare against expected data.**

6. **Increment to next buffer location. Go to step 5.**

7. **Repeat steps 5 and 6 until all the buffers have been read back.**

8. **Set next data pattern. Go to step 2.**

9. **Repeat step 8 until all patterns have been exhausted.**

The Block Patterns test fills the entire set of buffer with a single pattern and reads back all the buffers to check for any data mismatches. The patterns used are double words of 0x00, 0x55, 0xAA and 0xFF. The steps of the test are :

1. **Initialize test data.**
2. **Write to buffer location.**
3. **Increment to next buffer location. Go to step 2.**
4. **Repeat steps 2 and 3 until all the buffers have been written.**
5. **Read buffer location and compare against expected data.**
6. **Increment to next buffer location. Go to step 5.**
7. **Repeat steps 5 and 6 until all buffer locations have been read back.**
8. **Set next data pattern. Go to step 2.**
9. **Repeat step 8 until all data patterns are exhausted.**

The Walking Zeroes test walks a zero through each individual buffer. Each time one of the walking patterns is written to the buffer under test, it is read back for verification. The steps of the test are :

1. **Write buffer location to 0xFFFFFFFF.**
2. **Write 0 to one bit of the buffer location.**
3. **Read back buffer location and compare against expected data.**
4. **Repeat steps 2 and 3 until all data bits in the buffer have been written to 0.**
5. **Increment to next buffer location. Go to step 1.**
6. **Repeat step 5 until all buffer locations have been tested.**

The Walking Ones test walks a one through each individual buffer. Each time one of the walking patterns is written to the buffer under test, it is read back for verification.

1. **Write buffer location to 0xFFFFFFFF.**
2. **Write 1 to one bit of the buffer location.**
3. **Read back buffer location and compare against expected data.**
4. **Repeat steps 2 and 3 until all data bits in the buffer have been written to 1.**
5. **Increment to next buffer location. Go to step 1.**
6. **Repeat step 5 until all buffer locations have been tested.**

In the event of an error, the following message is displayed :

!ERROR: Address 0xXX Expected 0xXX Observed 0xXX XOR 0xXX

In the above message, Address indicates the location of the failing data buffer. The Expected value indicates the data that was suppose to have been written to the buffer. The Observed value indicates the actual data that was read back from the buffer being tested. The XOR value is the exclusive OR value between the expected and actual data.

Section 2 probes the expansion slots for the proper SCSI controller ID. If the SLOT argument has been used, the slot that has been specified is probed. The steps involved in the test are :

1. Read IDC board ID register.

2. If ID is proper, then probe is successful and display a message.

If the probe is successful, the following message will be displayed to the user :

SCSI Controller found in slot xx

Section 3 tests the LED wiring status by asserting and then deasserting the LED bit on the LED register. The status is checked on the SCSI controller board status register. The steps of the test are :

1. **Read SCSI board status register to ensure that LED status is not active.**

2. **Enable LED bit on the LED register.**

3. **Read SCSI board status register to ensure that LED status is active.**

In the event of an error, the following message is displayed.

!ERROR: Address 0xXX Expected 0xXX Observed 0xXX XOR 0xXX

The Address value indicates the location of the register where the failure occured. The Expected value is the data that was suppose to be contained in the register when it was read back. The Observed value is the data that was read back from the register. The XOR value is the exclusive or between the expected and observed data values.

SIOP Tests

SCSI Input/Output Processor (SIOP) tests the host access to the SCSI controller chip. In addition it also tests access to the DMA and SCSI FIFOs and all the possible interrupts that do not require the SIOP channel to be connected to any external devices (disk, tape drive, SIOP etc). The purpose of these tests are to ensure that the SIOP can function in SCSI script mode. The first group of tests checks out host access to the SIOP's register set. A series of pattern tests are performed on the SIOP registers. Host access ensures that the registers of the SIOP can be configured properly.

Both the SCSI and DMA FIFOs are tested with a variety of pattern tests. These tests ensure that data can be moved in and out of the FIFOs without any difficulty.

Finally the interrupts that can be generated without the SIOP being connected to an external device are tested. Basically an interrupt is initiated

by manipulating the SIOP registers and then a status register is polled to see if the interrupt has occured.

SIOP contains four sections. The sections of the diagnostic are :

Section 1: SIOP Register Test

Section 2: SIOP SCSI FIFO Test

Section 3: SIOP DMA FIFO Test

Section 4: SIOP Interrupt Test

The syntax of the diagnostic operation is as follows:

Syntax: GO [HELP] [CH channel] [PASS number] [SEC number] [SLOT number]]

The CH argument selects the channel to be tested. This is done by typing either the character A or B.

The SLOT argument selects the controller to be tested. This is specified by the value number. The valid slot values are 3 thru 5.

Section 1 of the diagnostic tests host access to the SIOP register set. This is to ensure that the chip can be configured properly for SCSI Script operation. This section contains five tests. They are listed as follows:

1. SIOP Reset
2. SIOP Register Pattern
3. SIOP Register Address
4. SIOP ID Register
5. Dual SIOP Uniqueness

The SIOP Reset test ensures that the chip is being reset properly. A software reset is initiated on each channel and then the registers are read

back to ensure that they are set to their correct default values. The steps of the test are :

1. **Assert the RST on the SIOP's DCNTL register.**
2. **Deassert the DCNTL register's RST bit.**
3. **Read SIOP register and compare against known default value.**
4. **Increment to next register. Go to step 3.**
5. **Repeat steps 3 and 4 until all registers have been read back and checked.**

The SIOP Register Pattern test writes a single pattern to the SIOP registers and then reads them back for verification. The data patterns used are bytes of 0x00, 0x55, 0xAA and 0xFF. The steps of the test are :

1. **Initialize data pattern to 0x00.**
2. **Write data to register.**
3. **Read back register and compare against known data.**
4. **Increment data pattern by 0x55. Go to step 2.**
5. **Repeat step 4 until all data patterns have been exhausted.**
6. **Increment to next register location. Go to step 1.**
7. **Repeat step 6 until all registers have been tested.**

The SIOP Register Address test writes a unique pattern to each SIOP register and then reads them back for data verification. The data patterns used are the address offsets f the registers that is being tested. The steps of the test are :

1. **Write the value of the test register address to the register being tested.**

2. **Increment to the next register. Go to step 1.**

3. **Repeat step 2 until all registers have been filled with data.**

4. **Read data back from register and compare against known data.**

5. **Increment to next register location. Go to step 4.**

6. **Repeat step 5 until all registers have been read back and checked.**

The SIOP ID Register tests both the SCSI chip id (SCID) and destination (SDID) registers. Each time the pattern changes, data is read back and checked for correctness. These two registers are essential to script operation. The chip id register is the SIOP's own id that is asserted on the IDC bus. The destination register is the SIOP's target. The steps of the test are :

1. **Initialize data pattern to 0x01.**

2. **Write data to register.**

3. **Read back data and compare with known data.**

4. **Shift data left 1 bit. Go to step 2.**

5. **Repeat step 4 until all bits have been tested.**

The Dual SIOP Uniqueness test writes data to the registers of both the SIOP channels and then read back for verification. The purpose of this test is to ensure that one channel can be accessed with corrupting the data on the other one. The steps in the test are :

1. **Initialize data pattern to 0x00.**

2. **Write data to register on channel A SIOP.**

3. **Write complement of data to register on channel B SIOP.**

4. **Read back data from register on channel A SIOP and compare against known data.**

5. **Read back data from register on channel B SIOP and compare against known data.**

6. **Increment to next register location. Go to step 2.**

7. **Repeat step 6 until all register locations have been tested.**

8. **Increment test data pattern by 0x55. Go to step 2.**

9. **Repeat step 8 until all data patterns have been used.**

In the event of an error, the following message is displayed :

!ERROR: Address 0xXX Expected 0xXX Observed 0xXX XOR 0xXX

The Address value is the location of the failing register. The Expected value is the data that is suppose to have been read back from the register. The Observed value is the actual data that was read back from the register. The XOR value is the exclusive OR of the expected and observed data values.

Sections 2 and 3 test the SIOP SCSI and DMA FIFOs respectively. The following paragraphs describe how they are tested.

The SCSI FIFO is an 8 transfer deep and byte wide. Known data is written to the FIFO and then read back for verification. The steps in testing the SCSI FIFO are as follows:

1. **Set up parity and load method. These two elements are controlled by the EPG (Enable Parity Generation) bit on the SCNTL0 register and the AESP (Assert Even SCSI Parity) bit on the SCNTL1 register. The parity types and load methods are described as follows:**

2. **Write the SCSI FIFO access bit to 1 on the CTEST4 register. This enables the FIFO to accept data.**

3. **Load the SCSI FIFO with the data by writing to the SODL register.**

The data is then read back for verification through the CTEST3 register. The parity is read back through the CTEST2 register.

Disable FIFO access by writing the SCSI FIFO enable bit on the CTEST4 register to 0. The DMA FIFO is a 36X8 bit FIFO that is divided into 4 sections, each being 9-bits wide and 8 transfers deep. Each of these sections are known as "byte lanes". Each can be tested by writing known data into the FIFO and read back out of the FIFO.

Write the appropriate value the chip test register 4 (CTEST4). This enables the appropriate byte lane. The values for enabling and disabling the byte lanes are as follows:

1. **Write the parity bit of chip test register 7 (CTEST7) and the 8 bit data value to the CTEST6 register to write the desired data pattern to the FIFO.**

2. **Read back the data out of the FIFO from the CTEST6 register.**

3. **Read back the parity bit in CTEST2 register for parity check.**

4. **Disable FIFO access by writing 0x00 to the CTEST4 register.**

Remote Maintenance Processor Board Group

The following tests make up the diagnostic module for the RMP hardware group:

- IPRMP (RMP IP Bus Tests)

IPRMP tests the IP bus interface between the processor and RMP boards. Access to the IP bus accessible registers on the RMP from the processor board is tested. In addition the IP bus loopback buffers on the RMP are tested.

Remote Maintenance Processor Diagnostic Module

IPRMP tests IP bus interface between the RMP and Processor boards. Host access to the RMP's CSR registers and IP bus data interface are tested.

The diagnostic contains two sections. The sections of the diagnostic are listed as follows:

Section 1: IP Bus RMP CSR Test .

Section 2: IP Bus RMP Loopback Buffer Test.

The syntax of the diagnostic operation is shown as follows:

Syntax: GO [HELP] [[PASS number] [SEC number]]

Section 1 tests the access to the common RMP CSR over the IP bus interface. For each register, a byte pattern is written and then read back for verification. The series of patterns used are 0x00, 0x55, 0xAA and 0xFF. The steps of the test are listed as follows:

1. **Initialize data pattern to 0x00**
2. **Write data to register**
3. **Read back data from register and compare against known data**
4. **Increment test data pattern by 0x55. Go to step 2**
5. **Repeat steps 2 thru 4 until all data patterns have been used. Then go to step 6**
6. **Increment to next register location. Go to step 1**
7. **Repeat steps 2 thru 6 until all registers have been tested.**

Section 2 tests the RMP IP bus loopback buffers. For each buffer, the series of double word patterns 0x00, 0x55, 0xAA and 0xFF written and then read back for verification. The steps of the test are listed as follows:

1. **Initialize data pattern to 0**
2. **Write data to loopback buffer**
3. **Read back loopback buffer and compare against known data**
4. **Complement test data**
5. **Write data to loopback buffer**
6. **Read back loopback buffer and compare against known data**
7. **Increment test data pattern by 0x55555555. Go to step 2**
8. **Repeat steps 2 thru 7 until all patterns have been used.**

In the case of an error occuring for any of the sections, the following message is displayed to the user.

!ERROR: Address XX Expected XX Observed XX XOR XX

The Address value is the register location where the error occured. The Expected value is the data that was written to the buffer. The Observed value is the actual data that was read back from the buffer. The XOR value is the exclusive OR between expected and observed data values.

Chapter 4: Functional Description of Field Maintenance Tests

Functional Description of Field Maintenance Tests

This chapter describes the functional aspects and operation of the various diagnostic tests that are available utilizing the Field Maintenance Test diskette. As introduced in chapter 1, this diskette is a a menu driven option program which allows the user to interact with the diagnostic testing process through menus and dialog boxes displayed on the monitor. These menus present various testing parameters and optional choices to the user. A description and illustration of the various menu options that may be selected are shown below.

Running the Field Maintenance Test diskette

This program requires a server console; you cannot run it from a remote location. To start the diagnostic tests using the Field Maintenance Test diskette, follow this procedure:

1. **Insert the IBM PS/2 Server Field Maintenance System diskette in drive A: and reboot the server.**

After POST is completed, a banner screen is displayed on the console as shown in Figure 4-1. This diagnostic menu initially provides the user with the capability to exercise the following diagnostic test options:

1. **Run Advanced Diagnostics**
2. **Run Memory/CPU Diagnostics**
3. **Enter the Diagnostic Monitor program**
4. **Display ECC error log**

5. **Display NMI error log**

6. **Exit the Field Maintenance Tests to run other DOS diagnostics**

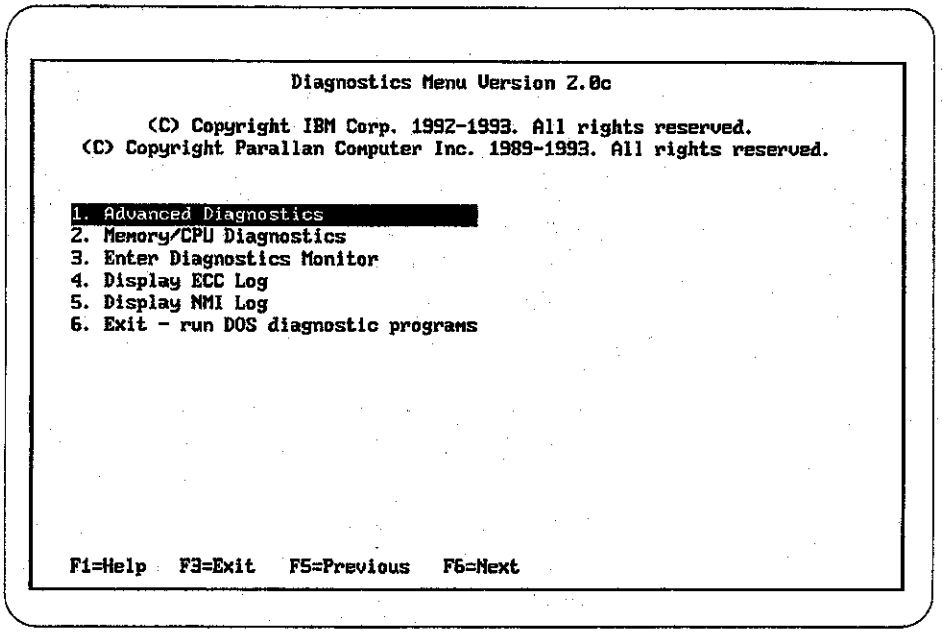


Figure 4-1
Main menu example.

Advanced Diagnostics

A typical user interactive scenario is described in the following section assuming that the Advanced Diagnostics item has been selected from the Main Menu.

A message is displayed on the console indicating that the tests are loaded into memory. You are prompted with a question - "If you desire to load more diagnostic tests, remove the diskette from the default drive and then

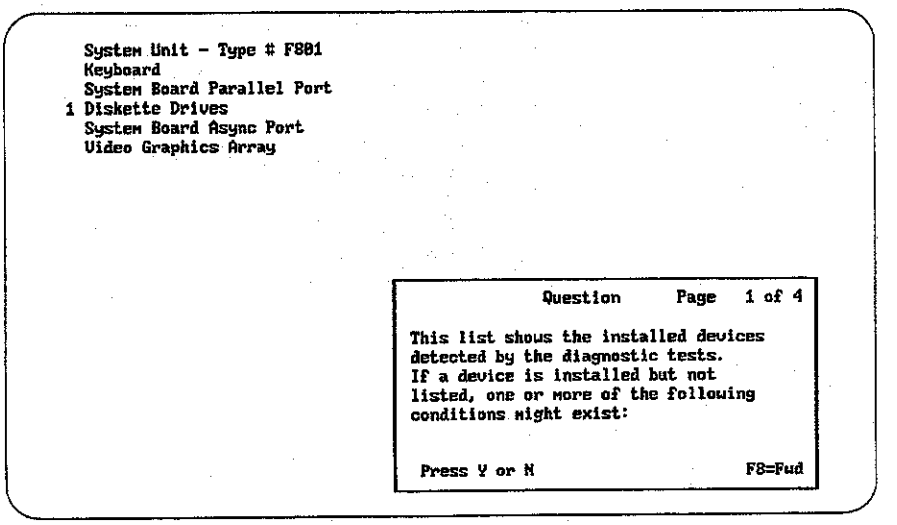


Figure 4-2
Device list example

insert the next diskette". If you select (Y), the system will allow you to load other diagnostics such as those that are provided with other option diskettes for Micro Channel adapters.

When (N) is selected, the Device List menu is displayed, as shown in the following Figure 4-2. This list shows the installed devices detected by the diagnostic tests.

Figure 4-3 shows the Device Test Menu which allows you to test the devices listed . You may test all the devices listed or run tests on individual devices. Use the up and down arrow keys to highlight the selections.

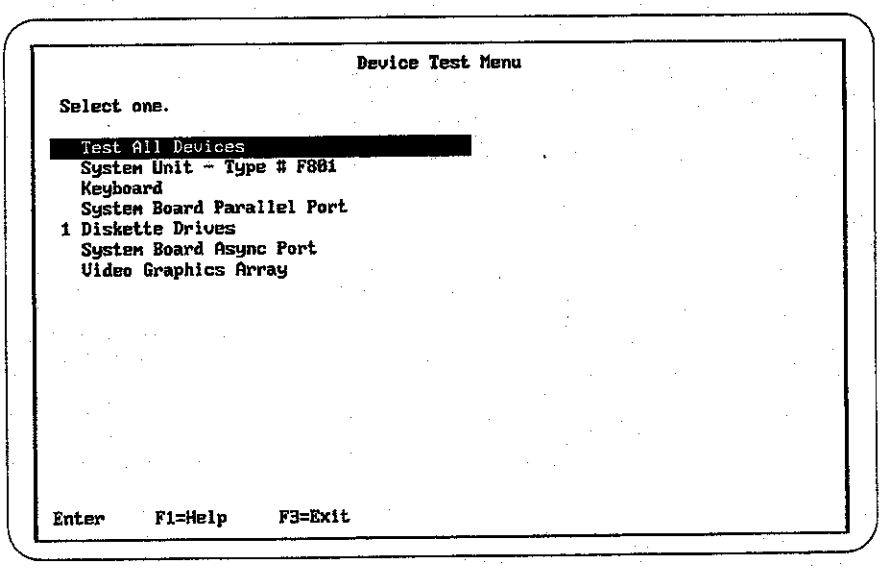


Figure 4-3
Device Test Menu Example

After a selection is made, a "Test Selection Menu" is displayed (Figure 4-4).

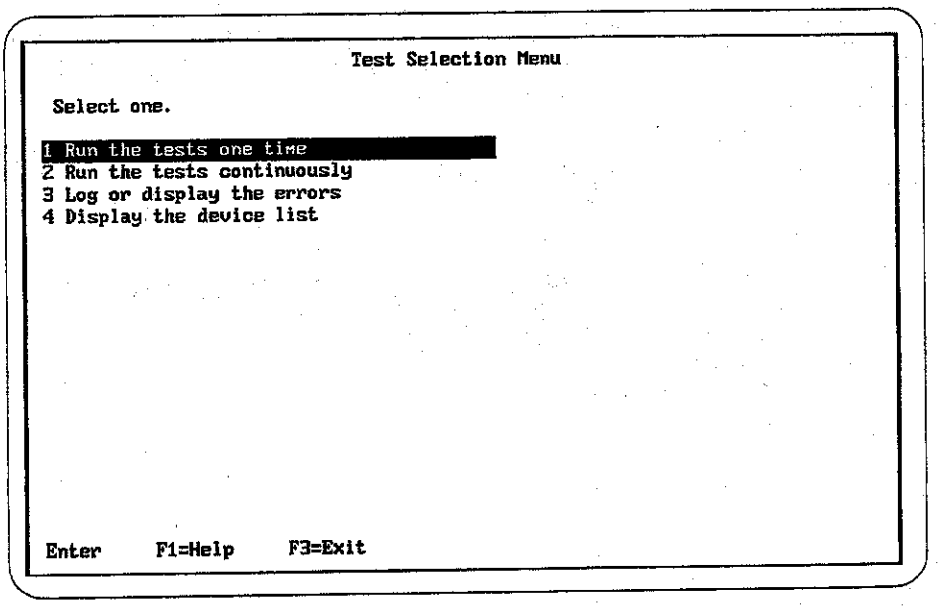


Figure 4-4
Test Selection Example

Figure 4-4 shows the Test Selection Menu which allows the user to invoke the following test actions:

1. **Run the tests one time.**
2. **Run the tests continuously**
3. **Log or display the errors**
4. **Display the device list**

If you select item number 3 from the Test Selection Menu, the program will forward you to the error log menu (See Figure 4-5) to allow a selection of any of those options.

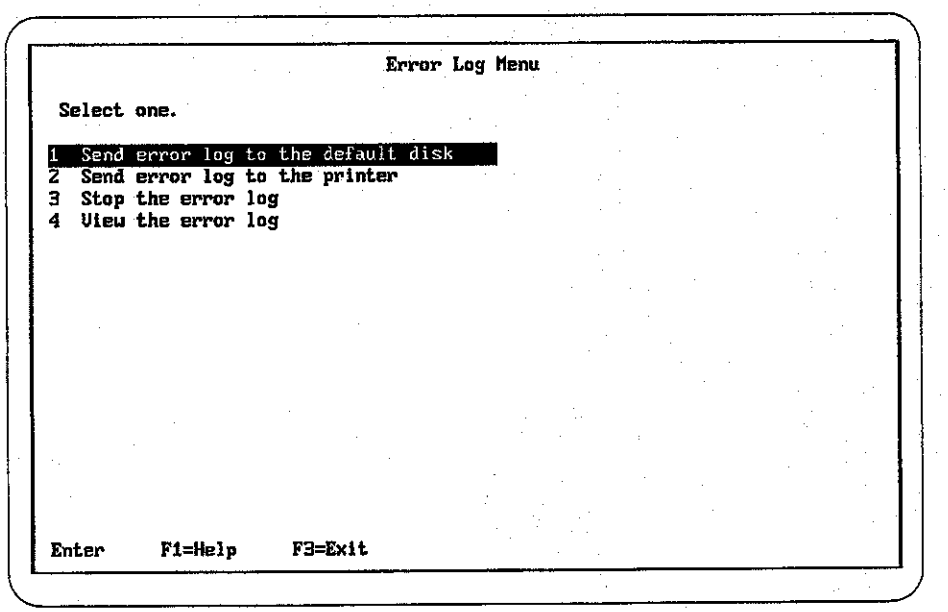


Figure 4-5
Error log menu example

If you select item number 4 from the Test Selection Menu (Figure 4-4), the program will forward you back to the device list menu (See Figure 4-2).

Memory/CPU Diagnostics Tests

When the user selects Memory/CPU Diagnostics (item 2) from the Main Menu (Figure 4-1), the screen shown in Figure 4-6 is displayed. Note that both items (1 and 2) displayed are for a single processor system, whereas, a multiprocessor system will additionally display choices for "Memory test on remote processor" and "Multiprocessor diagnostic".

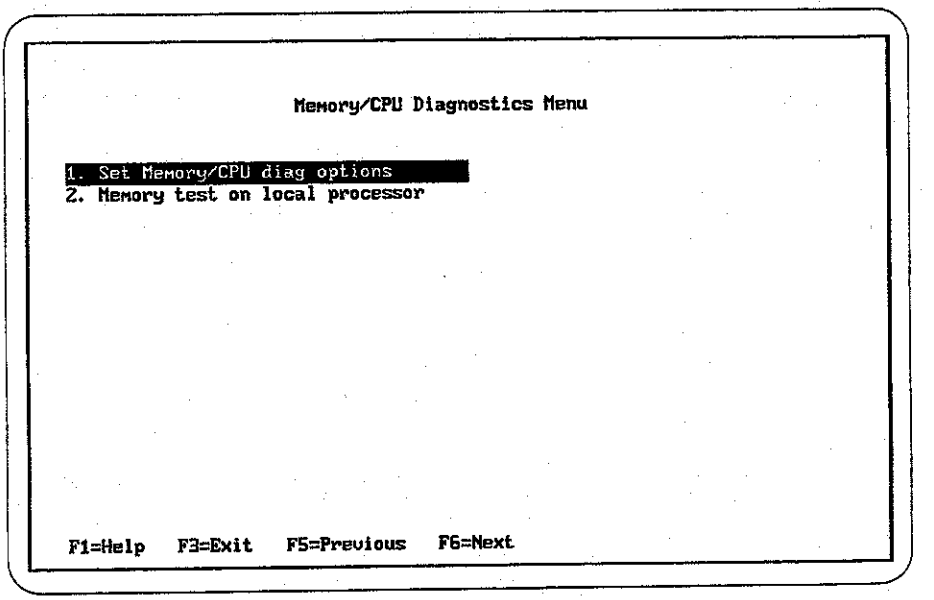


Figure 4-6
Memory/CPU diagnostics menu example

When Set Memory/CPU diag options (item 1) is selected, the screen shown in Figure 4-7 is displayed that allows the user to set IP diagnostic parameters.

If Memory test on local processor (item 2) is selected, it will test the complete memory of the system. Note that this test will only identify the bank locations.

Setting IP Diagnostic Parameters

Figure 4-7 below displays the various IP Diagnostic Parameters that can be set by the user as a prerequisite to the type of error log reporting desired. To change the default settings, toggle from (Y) "YES" to (N) "NO"(and vice versa) using the F5 and F6 keys.

If you select "0" for the "Number of test iterations", the test will run continuously.

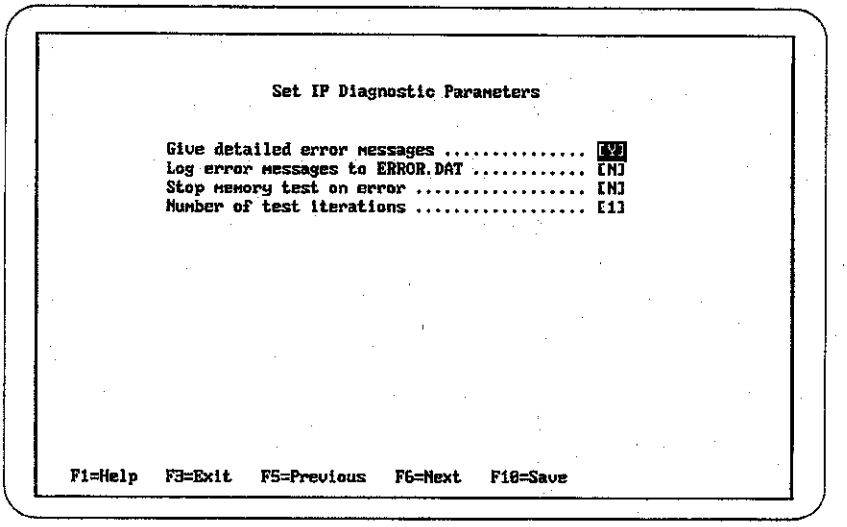


Figure 4-7
Set IP Diagnostic Parameters Example

Diagnostic Monitor Selection

When the user selects the Enter Diagnostics Monitor option (item 3) shown in the Main Menu (Figure 4-1), the server is automatically placed in DIAGMON mode.

Error log reporting

The ECC memory error log is generated by selecting Display ECC Log (Item 4) on the Main Memu shown in Figure 4-1. The ECC Memory Error Log displayed is shown in Figure 4-8 below.

The NMI Log is generated by selecting Display NMI Log (item 5) from the Main Menu shown in Figure 4-1. The NMI Log displayed is shown in Figure 4-9.

ECC Error Log Pg 1 of 2

#	ECC syndrome	Location	Info	Time Stamp
1	data bit 21	0x8a220	double bit	Tue Mar 23 20:08:45 1993
2	data bit 21	0x8a220	hard error	Tue Mar 23 20:08:41 1993
3	data bit 21	0x8a220	hard error	Tue Mar 23 20:08:41 1993
4	data bit 21	0x8a220	hard error	Tue Mar 23 20:08:41 1993
5	data bit 21	0x8a220	hard error	Tue Mar 23 20:08:41 1993
6	data bit 21	0x8a220	hard error	Tue Mar 23 20:08:41 1993
7	double bit	0x8a400	double bit	Fri Mar 19 21:29:26 1993
8	double bit	0x8a400	hard error	Fri Mar 19 21:28:22 1993
9	double bit	0x8a400	hard error	Fri Mar 19 21:28:22 1993
10	double bit	0x8a400	hard error	Fri Mar 19 21:28:22 1993
11	double bit	0x8a400	hard error	Fri Mar 19 21:28:22 1993
12	double bit	0x8a400	hard error	Fri Mar 19 21:28:22 1993
13	data bit 21	0xc4dce0	double bit	Tue Mar 16 11:22:14 1993
14	data bit 21	0x8a260	double bit	Mon Mar 15 23:13:59 1993
15	data bit 21	0x0	double bit	Mon Mar 15 18:05:49 1993

F3=Exit F8=Fwd

Figure 4-8
ECC Memory Error Log Example

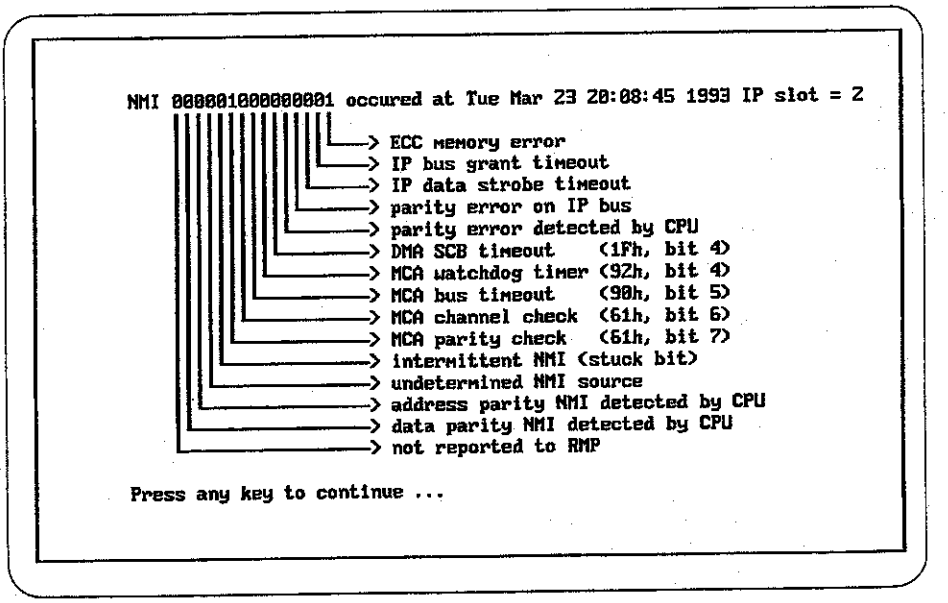


Figure 4-9
NMI Log Example

Diagnostic Monitor Test Batch Files

Diagnostic tests on the PS/2 Server Diagnostic Monitor Tests diskette is divided into categories of single or multiple tests. Single tests are executable files which are indicated on the diskette with a .ABS suffix. Multiple tests (group tests) can be run by executing batch files (.BAT suffix). There are seven different batch files on the diskette. These are:

- ✓ system.bat
- ✓ cpu.bat
- ✓ ip.bat
- ✓ mca.bat (only used for testing the 3Com Etherlink MC-32 Micro Channel adapter)
- ✓ memory.bat
- ✓ rmp.bat
- ✓ scsi.bat

Table A-1 lists the contents of these batch files. Note that these batch files can only be loaded from the floppy diskette.

Table A-1
DIAGMON diagnostic test batch file content

SYSTEM.BAT

```
dev fd
ld printer
go
ld serail
go
ld wrbuff
go
ld cmos
go
ld cachet
go
ld cachem
go
ld pic
go
ld dma
go
ld rtc
go
ld timer
go
ld ipmem
go
ld edac
go
ld dram
go
ld cpumem
go
ld fdc
```

CPU.BAT

```
dev fd
ld serial
go
ld printer
go
ld cachem
go
```

Table A-1 (Continued)
DIAGMON diagnostic test batch file content

```
                    CPU.BAT (Continued)
ld cachet
go
ld cachem
go
ld dma
go
ld cmos
go
ld rtc
go
ld timer
go

                      IP.BAT
dev fd
ld wrbuff
go
ld ipmem
go

                     MCA.BAT
ld fd mcplus
go


                     MEMORY.BAT
dev fd
ld edac
go
ld dram
go
mv
cache on
go
```

Table A-1 (Continued)
DIAGMON diagnostic test batch file content

```
                              RMP.BAT
ld fd iprmp
go

                              SCSI.BAT
ld fd ipscsi
go sec 2
ld fd siop
go sec 1
```

Printed in U.S.A.
71G2213