Series 37

Self Test and Maintenance Mode

Reference Manual

HEWLETT PACKARD
8010 Foothills Blvd., CA 95014

Part No. 30457-90003
E1184

Printed in U.S.A. 11/84

PRINTING HISTORY

New editions are complete revisions of the manual. Update packages, which are issued between editions, contain additional and replacement pages to be merged into the manual by the customer. The dates on the title page change only when a new edition or a new update is published. No information is incorporated into a reprinting unless it appears as a prior update; the edition does not change when an update is incorporated.

The software code printed alongside the date indicates the version level of the software product at the time the manual or update was issued. Many product updates and fixes do not require manual changes and, conversely, manual corrections may be done without accompanying product changes. Therefore, do not expect a one to one correspondence between product updates and manual updates.

First Edition . . . . . Nov 1984

LIST OF EFFECTIVE PAGES

The List of Effective Pages gives the date of the most recent version of each page in the manual. To verify that your manual contains the most current information, check the dates printed at the bottom of each page with those listed below. The date on the bottom of each page reflects the edition or subsequent update in which that page was printed.

Effective Pages	Date
all	Nov 1984

CONTENTS

CONTENTS (continued)

FIGURES AND TABLES

LIST OF TABLES

GENERAL INFORMATION

SECTION 1

INTRODUCTION

This manual describes the Self Test for the HP 3000 Series 37 in both power-on (PON) mode and maintenance mode.

The Self Test is the primary turn-on test. It tests for a functioning CPU, Memory, Termianl Interface Controllers (TICs), Peripheral Interface Channels (PICs), and Synchronous Intermodule Bus (SIMB). Self Test will also verify the cold load path.

The TIC is part of the Advanced Terminal Processor for the Series 37 (ATP37).

Self Test executes when any of the following happens:

- power-on via the keyswitch
- invoked by the operator while in maintenance mode
- restart after power failure

REQUIRED HARDWARE

The minimum hardware required to run Self Test to completion is a console (connected to TIC channel 1/port 0) and the HP 3000 Series 37 which includes:

CPU
Memory
TIC

The power-on self test sequence is CPU, Memory, TIC, and PIC. If a required peripheral is not connected, all tests performed before that peripheral is required are valid.

To use certain features of the Softpanel a breakpoint board is required.

SELF TEST ROM CODE

The ROM code is made up of three subsets: executable ROM code and two types of loadable ROM code - executable code and ROM-based messages.

The executable ROM code loads the first series of self tests from loadable ROM code into Writable Control Store (WCS) upon power-on. Control is then passed to the code loaded into WCS.

This code tests the CPU chip and slow WCS. It then loads the Code Loader and the second series of tests into Slow WCS.

The second series of tests check additional portions of the CPU chip, fast WCS, and the CPU register file.

When these tests are successfully completed, the maintenance panel code is loaded into WCS and enabled. Then the Self Test Executive is loaded into WCS and control is passed to it.

SELF TEST EXECUTIVE

The Self Test Executive determines if an Auto Restart after power failure is to be performed. Auto Restart tests a subset of the memory tests.

If a normal power-on sequence occurred, the Self Test Executive tests all of memory.

The Self Test Executive then tests all of the I/O cards installed in the system and speed senses the system console. The slot numbers of all cards tested are displayed on the console. The failing cards have the error code displayed in inverse video. A prompt is issued, unless Auto Warmstart is enabled. If Auto Warmstart is specified, control is passed to the Loader Code and the Autowarmstart prompt is displayed.

POWER-ON SELF TEST

When power is applied to the SPU, the executable ROM code loads into Writable Control Store (WCS) the initial part of Power On Selftest (PON) and a ROM Code Loader. The code in WCS is then executed.

The basic CPU chip test, a general WCS test, and a ROM code loader are in ROM. This code tests some of the Series 37 CPU chip functions. The last CPU chip function tested is all of Slow WCS and then additional CPU test modules are loaded. These modules test the remaining CPU chip functions. When these tests are complete, the Maintenance Panel module, the Control module, and the Self Test Executive module are loaded. Control is then passed to the Self Test Executive.

The Test Executor loads the memory tests, the TIC test, and the PIC test. The memory tests are then executed. If the memory tests are successfully executed, the console TIC and all of the other channels in the SPU are tested. The results are displayed on the LED display and the console if speed sensing was successful.

The Control module initializes the TIC and displays the following prompt:

```
H for help->
```

MAINTENANCE MODE

The maintenance mode microcode contains the code for Maintenance Mode, and Test Mode commands. Unlike earlier versions of HP 3000 computers, there is no special maintenance processor. The maintenance microcode resides on the CPU board. The Control B detection logic is enabled on the TIC that is in slot 1 channel 1 of the SPU card cage. Only the console connected to Channel 1/Port 0 or the remote console can issue Control B (B^C) to enter Maintenance Mode.

When you enter Maintenance Mode software execution halts. Note that the MPE environment is protected only if you execute a DUMP, DW, EW, HELP, RUN, or SPEED command. If any other commands are executed, MPE is not protected.

WARNING

The following procedure should be attempted only by trained Hewlett Packard Service Personnel.

In some cases, Control B will not be detected when entered on the System Console or the remote console. In this case, Maintenance Mode can sometimes be entered by closing the ATTN switch on the main card cage. It is on the right hand side looking in the front, and is on the bottom printed-circuit assembly (PCA) in the cardcage, directly below the display board.

THE REMOTE CONSOLE

The remote console capability allows the system to be halted or started, and Maintenance Mode commands to be executed from a remote location. When the remote console is in operation the local console can be used to monitor all activity.

Establishing The Remote Console Link

There are three steps to establishing a link to the remote console.

1. Issue a speed command to set the System Console to a speed that is compatible with both the modem and the terminal to be used in the remote location.

2. Set the keyswitch to the remote position. This causes the Data Terminal Ready (DTR) to temporarily drop, disconnecting any pre-established links. MPE will log off the session on Port 7.

3. The remote site must establish the modem link. The REMOTE light will light indicating that REMOTE mode has been established. This will disable the keyboard on the local console and enable the keyboard on the remote console. Both screens will display identical information.

While in remote mode the local keyboard can be enabled by turning the keyswitch to position "2" (LOCAL). As long as the REMOTE LED remains lit, the remote console will mirror the activity of the local console. The remote keyboard can be enabled by turning the keyswitch back to position "3" (REMOTE). Thus the keyswitch acts as a toggle to enable either the local or remote console's keyboard and as long as the connnection has not be broken (by turning the keyswitch to either position "0" or "1") the screens on the local and remote consoles will display the same information.

Disconnecting the Remote Console

Remote mode is exited by turning the keyswitch to position "1" (NORMAL). This causes the DTR to drop temporarily (3-4 seconds) which disconnects the modem. MPE will log off the session on Port 7 and the REMOTE light will go out. The remote console will no longer display the same information as the local console.

OPERATING INSTRUCTIONS

SECTION 2

INTRODUCTION

This section describes how to operate both the Power-on Self Test and the Maintenance Mode Microcode.

The primary function of the Power-on Self Test (PON) is to verify the correct operation of the logic necessary to load the system.

The maintenance mode microcode contains the code for interpreting maintenance commands and the Series 37 Self Test. Since the Series 37 does not have a special maintenance processor, the maintenance microcode resides on the CPU board. The Control B detection logic is on the TIC. Only the TIC in slot 1 (channel 1) can have the the Control B logic enabled. To enable the logic, the keyswitch must be in either the Local or Remote position. Note also that only the System Console connected to port 0 (local) or port 7 (remote) can execute the Maintenance Mode commands.

KEYSWITCH CAPABILITIES

The Keyswitch directly effects the function of certain Maintenance Mode commands. These are described in Table 2-1 Keyswitch Capabilities.

Table 2-1. Keyswitch Capabilities

Keyswitch Position	DC	Looping Enabled	PON Command Allowed	TIC SLOT 1	CONTROL B Enabled
0	NO	N/A	N/A	N/A	N/A
1 NORMAL	YES	NO	YES	YES	NO
2 LOCAL (from NORMAL)	YES	YES	YES	YES	YES
2 LOCAL (from REMOTE)	YES	YES	NO	NO	YES (Local Console)
3 REMOTE	YES	YES	NO	NO	YES (Remote Console)

POWER-ON SELF TEST EXECUTION

Immediately upon powering on the CPU, Power-on Self Test begins execution. The LED display will flash "0", "1", and "2" (the "2" may not be visible) as the first Series of tests are loaded into the WCS. Then a "5", indicating card 5 of the CPU, is displayed as the CPU tests are executed. Upon successful completion of the CPU tests, a "B" will be displayed as the memory tests are executed. Upon successful completion of the memory test, a "C" will be displayed as all of the I/O cards in the system are tested and the System Console is speed sensed. If any of the tests fail, the failing assembly will be indicated in the LEDs as well as on the System Console if the console path is operational. Table 2-2 lists the LED indications and error conditions.

If PON completed without error, and Auto Warmstart was enabled, control will be passed to the loader code.

If PON looping was previously specified from the test mode, the Power-on Self Test will be looped until the loop count reaches 0. However, if the keyswitch was changed to the "1" (NORMAL) position, the looping feature is disabled and PON will be executed only once. When all tests have been successfully completed Maintenance Mode is entered. The following prompt is displayed:

```
H for help->
```

If any errors are reported on either the LED or the System Console contact your local HP Service Representative.

Table 2-2. LED Indications and Error Conditions

LED Indication	Meaning
Flashing "B"	Memory Test failed.
Flashing "C"	Console Failed to Speed Sense.
Flashing "C" followed by Flashing "1"	The Console TIC is bad.
Flashing "5" or Steady "5"	The CPU card failed.
Flashing "n"	The Card in Slot n failed.
Flashing "E" followed by Flashing "n"	For each card in slot n of the extender that failed.
Steady "0"	Processor is not executing micro-instructions or looping on first ones.
Steady "1"	Processor hung in the initial WCS bootup from ROM.
Steady "2"	Initial WCS bootup from ROM failed the data integrity test. A retry is being attempted.

MAINTENANCE MODE

Maintenance Mode allows the System Operator to perform a number of functions. The major functions which can be performed are the LOAD/START/DUMP and Test commands. LOAD and START load MPE from tape or disc respectively, and the DUMP command loads the dump software from the START device. The TEST command allows the System Operator to run the Power-on Self Tests with failures displayed on the System Console. The Test Mode also allows a limited looping feature. Tests can be looped with the *count* parameter. *Count* must be an integer between 1 and 9999, with 1 being the default. The looping feature is disabled when the keyswitch is in the position "1" (NORMAL).

The LOAD, START, and DUMP commands require a channel and device specification unless the default device has been set up by the ",P" and ",C" options. The DUMP command always defaults to the START device. The ",P" option updates the LOAD or START device data in the TOC and initiates the LOAD or START. The TOC is a permanent storage area for system information which includes the default START and LOAD device numbers. The TOC LOAD or START data can also be updated with the ",C" option but the LOAD or START is not performed. The valid channels are 1, 3, or 4 for main and 9-13 for the extender. RUN attempts to "run" the system if it arrived in Maintenance Mode via a Halt or Control B. The "AR" command allows the System Operator to try another Autorestart if the system has given up (5 tries of approximately 90 seconds each). The other LOAD/START type commands (COO, COL, DIS, NEW, RELO, TAPE, UPDA, and WAR) use the LOAD or START deivce as stored in the TOC. Table 2-3 lists the valid Maintenance Mode Commands. For details on the correct syntax of these commands refer to Section III "COMMAND DESCRIPTIONS".

Table 2-3. Maintenance Mode Commands.

Command	DESCRIPTION
AR	Retry Auto Restart.
COLDSTART	Perform COLDSTART using LOAD device. Refer to Note 1 below.
COOLSTART	Perform COOLSTART using START device. Refer to Note 1 below.
DISC	Perform START using START device. Refer to Note 2 below.
DUMP	Perform DUMP using indicated or START device.
DW	Disable auto Warmstart.
EW	Enable auto Warmstart.
HELP	Display HELP messages.
LOAD	Perform LOAD (has options). Refer to Note 2 below.
NEWSYSTEM	Perform initial software installation.
PANEL	Enter Soft Panel.
RELOAD	Perform RELOAD using LOAD device. Refer to Note 1 below.
RUN	RUN system after Control B halt.
SPEED	Allow the System Operator to change System Console speed.
START	Perform START (has options). Refer to Note 2 below.
TAPE	Perform LOAD using LOAD device. Refer to Note 2 below.
TEST	Go to Self Test mode.
UPDATE	Perform UPDATE using LOAD device. Refer to Note 1 below.
WARMSTART	Perform WARMSTART using START device. Refer to Note 1 below.

Notes:

1. These commands do not allow changes to the software configuration (no INITIAL dialogue).
2. These commands do allow changes to software configuration (INITIAL dialogue).

TEST MODE

The Test Mode allows the System Operator to select portions of the Power-on Self Test or an additional CPU test with failures displayed on the System Console. The Test Mode also allows for looping up to 9999 times, the default is 1.

The Maintenance mode TEST command accesses the Self Test mode. Table 2-4 lists the commands available in Test Mode.

REMOTE OPERATOR INTERFACE

The remote System Operator interface is enabled if all of the following conditions are met:

- The Console is connected to Port 7 of TIC in Slot 1 via modem or hardwire configuration.
- The keyswitch is in Remote (or Local after having been in Remote, but not switched through Normal) mode and the REMOTE LED is on.
- The System Console, the modem, and the remote console are all operating at the same baud rate.
- The hardwire configuration requires a standard 25 pin straight through modem cable with pins 4 and 22 to be tied together.

Table 2-4. Test Mode Commands

Command	Description
ALL	Run all self tests 1-9999 times.
CHANNEL	Loop Test of channel 1-9999 times.
CPU	Loop CPU test 1-9999 times.
EXIT	Return to Maintenance Mode.
HELP	Lists Test Mode Commands and display ROM version numbers.
IOMAP	Loop IOMAP 1-9999 times.
MEMORY	Loop memory test 1-9999 times.
PON	Run (or loop) Power-on Self Test.

COMMAND DESCRIPTIONS

SECTION 3

INTRODUCTION

This section provides descriptions of each section of the Series 37 Self Test code. Included here are definitions of the two modes, Maintenance Mode and Test Mode, and their associated commands.

Examples of error messages can be found in Appendix A.

MAINTENANCE MODE COMMANDS

Maintenance Mode is indicated by the "H for help->" prompt. Maintenance Mode initializes the TIC for communication with the System Operator and allows the following commands:

```
AR (Retry AutoRestart)
Coldstart
Coolstart
Disc
Dump
DW
EW
Help
Load
Newsystem
Panel
Reload
Run
Speed
Start
Tape
Test
Update
Warmstart
```

Automatic Warmstart

The Autowarmstart feature of Series 37 systems is enabled by the EW (Enable Warmstart) command. Automatic warmstart is disabled with DW (Disable Warmstart). The state of the Automatic Warmstart is maintained during power off. If Autowarmstart is enabled, the Series 37 will issue the following prompt after being powered on and PON is successfully completed:

```
WARMSTART? (Y/N)
```

The user has 15 seconds to abort the automatic warmstart. If the user types anything other than "Y" (RETURN), or (RETURN) alone, the system will abort the automatic warmstart, return to Maintenance Mode, and display the "H for Help" prompt. If the user types nothing the system will perform an automatic warmstart after waiting 15 seconds. The commands that affect the Automatic Warmstart feature are:

EW	Enable autoWarmstart
DW	Disable autoWarmstart

NOTE

The Automatic Warmstart feature is indepedent of the keyswitch position.

Auto Restart

The AUTO RESTART (AR) command allows the System Operator to retry the Power Fail Auto Restart capability if the Auto Restart was not successful because a disc was down or broken. To initiate an Auto Restart, the cause for the Auto Restart failure must be corrected, and then the System Operator may attempt an Auto Restart. Auto Restart should be successful as long as the memory has been powered, the Auto Restart flag in memory location 0.0358 (hex) is AAAA (hex), and the data in memory on the disc is uncorrupted.

Following the Power-on CPU test, an abbreviated version of the memory test is executed to find the size of memory if the Auto Restart flag at $0.0358 is set to $AAAA. Upon successful execution of this test, all of the I/O cards are tested and communication is established with the System Console. When these steps have completed, execution is transferred to the Auto Restart entry $5001 in the Loader code module. The WCS boot code is executed with the Auto Restart flag and LOAD/Start flags with the start device set into register file location $EF.

If the Auto Restart should fail because the disc was not ready, up to 4 retries will be made. If the retries are not successful, return will be to the Maintenance Mode "H for help->" prompt. The System Operator may type AR for up to 5 retries when the disc is made ready.

The correct syntax for this command is:

```
AR
```

Coldstart

The COLDSTART command loads the system microcode and software from the default tape unit and allows the System Operator to perform the COLDSTART operation for MPE or run DUS. Auto coldload can be run without intervention by the System Operator.

COLDSTART differs from LOAD in the following ways:

- No *channel* or *device* parameters are allowed.
- The default LOAD device always used.
- No opportunity is given to change the software configuration (no INITIAL dialogue).

The correct syntax for this command is:

```
COL[dstart]
```

Coolstart

COOLSTART loads the system microcode and software from the default disc unit and allows the System Operator to perform the COOLSTART operation for MPE.

COOLSTART differs from START in the following ways:

- No *channel* or *device* parameters are allowed.
- The default START device is always used.
- No opportunity is given to change the software configuration (no INITIAL dialogue).

The correct syntax for this command is:

```
COO[lstart]
```

Disc

DISC loads the system microcode and software from the START device (normally the system disc) and allows the System Operator to perform COOLSTART or WARMSTART operations for MPE or run DUS.

The DISC command differs from START in the following ways:

- No channel or device parameters are allowed.
- The default Start device is always used.

The correct syntax for this command is:

```
DIS[{c}{k}]
```

Dump

Dump loads the Dump software from the indicated device and transfers execution to the dump software. The DUMP command allows optional *channel* and *device* parameters to indicate the location of the dump software (normally the system disc). If no parameters are supplied, the dump software is loaded from the default START device.

DUMP uses the START default device number stored in the TOC RAM, or the System Operator may specify a different deivce. If WCS is not valid, control store is restored from the specified device. The softdump software is then loaded and control is transferred to it.

The load execution code test for valid parameters and transfers control to $5FFF if the microcode resides in WCS. If no microcode is loaded, control is passed back to the Maintenance Mode.

The correct syntax for this command is:

```
H for help->DU[mp] [chan,dev]
```

Help

The HELP command lists all of the commands along with their parameters.

HELP displays the version code of the currently loaded microcode or "mainset microcode is NOT loaded" if no microcode is loaded and then displays all of the commands available in Maintenance Mode.

```
2-LOCAL (from Normal)
H for help->Help

mainset microcode is NOT loaded
Auto warmstart disabled
last system stop code (hex)=0022

AR      - Re-attempt Auto Restart,      PA[nel] - Enter soft-panel,
COL[dstart],                            REL[oad],
COO[lstart],                            RUN,
DI[sc] - Start from Disc,               SP[eed],
DU[mp] [chan, dev],                     ST[art] [chan, dev[,{P[erm]   ]],
DW      - disable auto warmstart,                          C[hange]}
EW      - enable auto warmstart,        TA[pe]  - Load from Tape,
L[oad] [chan, dev[,{P[erm]   ]],        TE[st],
                    C[hange]}           UPD[ate],
NEW[system],                            WAR[mstart],

2 LOCAL (from Normal)
H for help->
```

Load

The LOAD command loads the system microcode and software from the indicated or default tape unit and allows the System Operator to perform COLDSTART, RELOAD, or UPDATE operations for MPE or to run DUS.

If *channel* and *device* are specified, LOAD uses these parameters as the channel and device for the LOAD. The default LOAD channel and device are also updated in the TOC register C if the optional *perm* or *change* parameters are specified.

The LOAD command allows the System Operator to perform a LOAD using the default LOAD device number stored in the TOC RAM, specify a different device, or just change the default device. The LOAD is performed by loading the load execution code, the WCS boot code, and transferring control to the load execution code.

The load execution code tests for the valid LOAD parameter and updates the load device in the TOC RAM if requested. Control is then transferred to the WCS boot code. If a parameter error occurred or if only a change to parameters was requested, control is passed back to Maintenance Mode.

If the parameters are valid and if a LOAD or START is to be done, control is passed to the WCS boot code which loads the HP3000 instruction set from the disc or tape and proceeds with the LOAD or START.

The load is performed with the normal operator (INITIAL) dialogue allowing for changes to the software configuration.

The correct syntax for this command is:

```
LOAD [chan,device][{P[erm]},{C[hange]}]
```

Newsystem

The NEWSYSTEM command operates like the LOAD command with the following exceptions:

- No *channel* or *device* parameters are allowed.
- The default LOAD device is always used.
- This command will only work with a FOS tape built especially for the Series 37. If NEW is attempted with other RELOAD tapes, the operator (INITIAL) dialogue will be invoked by INITIAL.

CAUTION

This command performs a RELOAD function erasing any existing user files on the system discs. It should be used only for the initial software installation.

The correct syntax for this command is:

```
NE[wsystem]
```

Panel

The PANEL command enters the Softpanel Mode. The commands available in Softpanel Mode are described in Section IV "SOFTPANEL".

Reload

The RELOAD command operates exactly like the LOAD command with the following exceptions:

- No *channel* or *device* parameters are allowed.
- The default LOAD device is always used.
- No opportunity is given to change the software configuration (no INITIAL dialogue).

The correct syntax for this command is:

```
REL[oad]
```

Run

The RUN command returns execution to MPE or DUS if Maintenance Mode was entered via Control B or Halt and MPE/DUS is executable,

The correct syntax for this command is:

```
RUN
```

Speed

The SPEED command allows the System Operator to change the System Console speed simply by changing the baud rate; no other interaction is required.

To abort the speed command, type any character except Control F with the System Console baud rate unaltered.

NOTE

When changing speeds at low baud rates, it may take up to a minute to get a prompt.

The correct syntax for this command is:

```
SP[eed]
```

Start

The START command loads the system microcode and software from the indicated, or default, disc unit and allows the System Operator to perform the COOLSTART or WARMSTART operations for MPE.

If the START *channel* and *device* are specified, the START command uses these parameters as the *channel* and *device* for the START. The default START *channel* and *device* are also updated in the TOC register B if the optional *perm* or *change* parameters are specified.

The START command allows the operator to perform a START using the default START channel numbers stored in the TOC RAM, specify a different device, or just change the default device. The START is performed by loading the load execution code, the WCS boot code, and transferring control to the load execution code.

The load execution code tests for the valid parameters and updates the TOC RAM if requested. Control is then transferred to the to the WCS boot code. If a parameter error occurred or if only a change to parameters was requested, control is passed back to the Maintenance Mode.

If the parameters are valid and if a START is to be done, control is passed to the WCS boot code which loads the HP 3000 instruction set from the disc and proceeds with the START.

The START is performed with the normal operator (INITIAL) dialogue, allowing for changes to the software configuration or selection of WARMSTART.

The correct syntax for this command is:

```
ST[art] [chan,device][{P[erm]},{C[hange]}]
```

Tape

The TAPE command operates exactly like the LOAD command with the following exceptions:

- No *channel* or *device* parameters are allowed.
- The default LOAD device is always used.
- This command is run with the normal operator (INITIAL) dialogue allowing for changes to the software configuration.

The correct syntax for this command is:

```
TA[pe]
```

Test

The TEST command transfers control to the Self Test Executive which displays all of the Self Test capabilities and the Test Mode prompt.

If software was running when Maintenance Mode was entered, the operator will be prompted with "Do you want to abort the system? (Y/N)". If a "Y" is entered, (or software was not running) then the Test Executor module is loaded and control is transferred to the Test Mode entry point in the Test Executor module. If not confirmed, control is transferred back to Maintenance Mode.

The Test Mode entry point in the Test Executor module displays the ROM version numbers, the commands available in Test Mode, and the "Test ->" prompt. The following commands are allowed in Test Mode:

```
AL[l] [ count ]
CH[an] [ count [ ,chan ] ]
CP[u] [ count ]
E[xit]
H[elp]
I[omap] [ count ]
M[emory] [ count ]
PON [ count ]
```

For more details, refer to "TEST MODE COMMANDS" latter in this section.

Update

The UPDATE command operates exactly like the LOAD command with the following exceptions:

- No *channel* or *device* parameters are allowed.
- The default LOAD device always used.
- No opportunity is given to change the software configuration (no INITIAL dialogue).

The correct syntax for this command is:

```
UPD[ate]
```

Warmstart

The WARMSTART command operates exactly like the START command with the following exceptions:

- No *channel* or *device* parameters are allowed.
- The default START device is always used.
- A warmstart can be run without any interaction by the System Operator (no INITIAL dialogue).

The correct syntax for this command is:

```
WAR[mstart]
```

TEST MODE COMMANDS

Test Mode is entered with the Maintenance Mode TEST command. The Test Mode allows all of the Self Test steps to be manually directed. The following Test Mode commands will be discussed in greater detail:

```
All                    Help
Channel                Iomap
CPU                    Memory
Exit                   PON
```

NOTE

Looping is disabled when the keyswitch is in position "1" (NORMAL).

All

ALL runs all of the manually directed self tests except PON test in the following order:

- CPU test
- Memory Test
- Channel Test
- IOMAP

The test may be looped by specifying the desired number of loops in *count*. *Count* must be an integer between 1 and 9999 with 1 being the default. A space is required before specifing *count*.

The correct syntax for this command is:

```
AL[L] [ count]
```

The following illustrates the use of the ALL command:

```
1-NORMAL
Test ->AL

 TOC RAM
Addr Data
0008 0000
0009 0000
000A 0000
000B 0000
000C 0004
000D 000E
000E 0000
000F 0000
Observe LED display cycle 0-F
CPU test passed
Memory Test passed
Slot   1 Channel  1  - Terminal Interface Controller
Slot   4 Channel  4  - Peripheral Interface Channel
Test Passed
                    System I/O Configuration
------------------------------------------------------------
Number of banks =  4
Load:       Channel  4 Device 3
Start/Dump: Channel  4 Device 1
------------------------------------------------------------
Slot   1 Channel  1  ID=4  - Terminal Interface Controller
------------------------------------------------------------
Slot   4 Channel  4  ID=2  - Peripheral Interface Channel
  Device 3 ID=0260  - 9144 Cartridge Tape Unit
------------------------------------------------------------

1-NORMAL
Test ->
```

Channel

The test may be looped by specifying the desired number of loops in *count*. *Count* must be an integer between 1 and 9999 with 1 being the default. A space is required before specifing *count*. *Count* must be specified whenever a specific channel is selected with *channel*. The failure codes for the TIC/PIC cards are listed in Appendix A. If a channel is not specified, all channels are tested.

The appropriate test (TIC, PIC, or none) is run for each card installed in the CPU and if a failure occurs, the failure code is printed on the System Console next to the card description.

The CHANNEL test performs the TIC test and the PIC test.

The TIC test has six sections:

- Init check. Perform an initialization of the TIC and test that the registers contain the proper data.
- Basic I/O operations. Issue OBII, IPOLL, SMSK, and RMSK and verify that they respond properly.
- Port register tests. Write patterns to registers 0-7 of ports 0-7 and verify the data.
- Diagnostic loopback using DMA sequencer ROM.
- Initiate PCC tests on all 8 ports.
- Perform DMA data loopback test on all 8 ports.

In addition the console TIC is both speed sensed and communication lines are tested with the local console.

The PIC test has nine sections:

- Init check. Perform an initialization of the PIC and test that the registers contain the proper data.
- Basic I/O operations. Issue OBII, IPOLL, SMSK, and RMSK and verify that they respond properly.
- Test the ABI status register (PIC register 1).
- Test the ABI interrupt and interrupt mask registers (PIC registers 2 and 3, respectively).
- Test PIC registers 4 and 5 using data patterns.
- Test PIC register 6.
- Test PIC register 7.
- Test PIC registers 8, 9, and A using data patterns.
- Fill ABI FIFO and do DMA to memory and DMA from memory. Test PIC registers associated with DMA transfers.

Return is to the Test Mode prompt.

The correct syntax for this command is:

```
CH[an][ count[,channel] ]
```

The following illustrates the use of CHAN:

```
1-NORMAL
Test ->CH
~
Slot   1 Channel  1  - Terminal Interface Controller
Slot   4 Channel  4  - Peripheral Interface Channel
Test Passed

1-NORMAL
Test ->
```

CPU

The Test Mode CPU command executes tests of the CPU not run at power-on. The following Tests are run by this CPU test:

- P, D, S, A Bank register testing.
- LED display testng.
- TOC RAM testing.
- TOC counting verified.
- MPE timer counting verified.
- Watchdog Timer Force Condition verified.

The tests not performed by this CPU test, but that are executed by the Power-on CPU test are:

- ROM Checksum Test.
- Full Nezumi chip test.
- Full Fast and Slow WCS address and Data Test.
- Register File Address and Data Test.

Test all of the Bank registers (P, D, S, A), tests the TOC RAM locations, ensures that the TOC and MPE timers are counting, displays LED codes 0-F, and tests the Watchdog timer/FMD capability.

The test may be looped by specifying the desired number of loops in *count*. *Count* must be an integer between 1 and 9999 with 1 being the default. A space is required before specifing *count*.

The correct syntax for this command is:

```
CPU [ count]
```

The following illustrates the use of CPU:

```
1-NORMAL
Test ->CP

 TOC RAM
Addr Data
0008 0000
0009 0000
000A 0000
000B 0000
000C 0004
000D 000E
000E 0000
000F 0000
Observe LED display cycle 0-F
CPU test passed

1-NORMAL
Test ->
```

Exit

The Test Mode EXIT command returns execution to the Maintenance Mode and displays the "H for help->" prompt.

The correct syntax for this command is:

```
E[xit]
```

The following illustrates the use of EXIT:

```
Test ->E
H for help->
```

Help

This command does not appear in the Test Mode menu. When issued HELP will display the available Test Mode commands and the ROM version numbers.

The correct syntax for this command is:

```
HELP
```

The following illustrate the use of HELP:

```
1-NORMAL
Test ->H


ROM Versions:1.nnnn  2.nnnn  3.nnnn  4.nnnn

Selftest Menu:

AL[l] [ count ]
CH[an] [ count [ ,chan ] ]
CP[u] [ count ]
E[xit]
I[omap] [ count ]
M[emory] [ count ]
PON [ count ]


1-NORMAL
Test ->
```

Iomap

The IOMAP command executes a version of IOMAP which is contained in the Self Test ROM. This version of IOMAP displays the number of banks of memory physically installed in the system and identifies all of the I/O cards installed in the system. For PIC cards, all supported HPIB devices attached to the PIC are identified and their ID code is displayed along with a device description.

This command runs the memory size portion of the memory test and displays the size of memory, lists the LOAD and START/DUMP devices, and then displays the types of cards installed in the system. The types of devices on the PICs are also displayed.

The test may be looped by specifying the desired number of loops in *count*. *Count* must be an integer between 1 and 9999 with 1 being the default. A space is required before specifing *count*.

The correct syntax for this command is:

```
I[omap] [ count]
```

The following illustrates the use of IOMAP:

```
1-NORMAL
Test ->I

                    System I/O Configuration
------------------------------------------------------------
Number of banks =  4
Load:       Channel  4 Device 3
Start/Dump: Channel  4 Device 1
------------------------------------------------------------
Slot   1 Channel  1  ID=4  - Terminal Interface Controller
------------------------------------------------------------
Slot   4 Channel  4  ID=2  - Peripheral Interface Channel
  Device 3 ID=0260  - 9144 Cartridge Tape Unit
------------------------------------------------------------

1-NORMAL
Test ->
```

Memory

The Test Mode **MEMORY** command executes the Power-on Memory Test which first determines the amount of memory installed, initializes the memory, performs an Address Test, performs a Pattern Test, and then performs an Error Detection And Correction Test (EDAC). The memory is left with 30F8 (halt 8) in all locations.

The full memory test is run and a pass/fail message is sent to the System Console. If a failure occurrs the number of the failing section is displayed.

The test may be looped by specifying the desired number of loops in *count*. *Count* must be an integer between 1 and 9999 with 1 being the default. A space is required before specifing *count*.

Return is to the Test Mode prompt.

The correct syntax for this command is:

```
M[emory] [ count]
```

The following illustrates the use of MEMORY:

```
1-NORMAL
Test ->M

Memory Test passed

1-NORMAL
Test ->
```

PON

The Test Mode PON command executes the Power-on Self Test *count* times. This command is much like the ALL Test Mode command with the exception that the Power-on CPU test is executed in place of the manually executed CPU test and IOMAP is not executed. The PON test is initiated by toggling the PON line.

NOTE

This command can not be run from keyswitch positions "3" or "1" from "3" (REMOTE and LOCAL from REMOTE, respectively). The correct syntax for this command is:

PON [*count*]

The following illustrates the use of PON:

```
2-LOCAL (from Normal)
Self Test ->PON
~

Power on Self Test
Memory Test passed
Number of banks =  4
Slot    1 Channel  1  - Terminal Interface Controller
Slot    4 Channel  4  - Peripheral Interface Channel

2-LOCAL (from Normal)
Self Test ->
```

SOFTPANEL

SECTION 4

INTRODUCTION

Softpanel is a diagnostic tool to examine software. Softpanel allows the user to display and modify memory, perform register and I/O operations, set memory breakpoints, and perform other necessary functions. All commands requiring parameters must have a "+", "-", or a space between the the command and the parameter. The following commands are allowed in softpanel:

Display Memory

Modify Memory

Register Operations
- DR
- MR

Execution Control
- E
- RUN

Input/Output Operations
- RIO
- WIO

Other/Misc
- T
- ENV
- RTOC
- WTOC
- RDX
- ST

Memory Break Points
- SMB
- CMB
- SMD
- SB

The parameters for these commands are defined as follows:

bank — One of the following numeric fields limited to a range of 0 - 255 (8 bits): The current radix numeric field, a hexadecimal numeric field preceded by a $ or a digit, or an octal numeric field preceded by a %.

count — One of the following numeric fields limited to a 16 bit maximum: The current radix numeric field, a hexadecimal numeric field preceded by a $ or a digit, or an octal numeric field preceded by a %.

expr — Is a combination of *numeric* and *op*. Operations are performed from left to right with no precedence.

ioaddr — One of the following numeric fields limited to a 16 bit maximum: The current radix numeric field, a hexadecimal numeric field preceded by a $ or a digit, or an octal numeric field preceded by a %.

iodata — One of the following numeric fields limited to a 16 bit maximum: The current radix numeric field, a hexadecimal numeric field preceded by a $ or a digit, or an octal numeric field preceded by a %.

numeric — One of the following numeric fields limited to a 16 bit maximum: The current radix numeric field, a hexadecimal numeric field preceded by a $ or a digit, or an octal numeric field preceded by a %.

op	One of: a +, -, or *. Operations done on numeric fields are done signed. Bit (0:1) is the sign bit. : is the indirection operator.
reg	One of the following registers: DB, DL, Q, S, PB, PL, Z, STA (the HP 3000 status register), SB (the split bank flag, 1 bit wide), CIR, X, SW (the switch register containing load/boot device drt), Dbank, Sbank, Pbank, LPFlg, DISP, or ICS.
regfile	One of the following numeric fields limited to a range of 0 - 255 (8 bits): The current radix numeric field, a hexadecimal numeric field preceded by a $ or a digit, or an octal numeric field preceded by a %.
tocaddr	One of the following numeric fields limited to a 16 bit maximum: The current radix numeric field, a hexadecimal numeric field preceded by a $ or a digit, or an octal numeric field preceded by a %.
tocdata	One of the following numeric fields limited to a 16 bit maximum: The current radix numeric field, a hexadecimal numeric field preceded by a $ or a digit, or an octal numeric field preceded by a %.

COMMAND DESCRIPTIONS

Display Memory

The D command continues from where we last left off and displays another half screen full of data. All display commands display in the current radix (refer to SDM [Set Display Mode] command). The display command will always show multiples of 8 words in the current radix and in ASCII. The options are:

DA *expr*[:[+|-]*expr*]][,*count*]	Displays the absolute address of bank 0.
DEA *bank.expr*[:[+|-]*expr*]][,*count*]	Displays the absolute address relative to the specified bank.
DSY[+|-|[ ]*expr*]][:[+|-]*expr*]][,*count*]	Displays the absolute address relative to sysglobal.
DDB[+|-|[ ]*expr*]][:[+|-]*expr*]][,*count*]	Displays the absolute address relative to the DB register.
DDL[+|-|[ ]*expr*]][:[+|-]*expr*]][,*count*]	Displays the absolute address relative to the DL register.
DQ [+|-|[ ]*expr*]][:[+|-]*expr*]][,*count*]	Displays the absolute address relative to the Q register.

DPB[+ -expr [expr]][:[+ -]expr]][,count]	Displays the absolute address relative to the PB register.
DS [+ -expr [expr]][:[+ -]expr]][,count]	Displays the absolute address relative to the S register.
DZ [+ -expr [expr]][:[+ -]expr]][,count]	Displays the absolute address relative to the Z register.
DP [+ -expr [expr]][:[+ -]expr]][,count]	Displays the absolute address relative to the P register.
DPL[+ -expr [expr]][:[+ -]expr]][,count]	Displays the absolute address relative to the PL register.

Modify Memory

Modify memory commands will display the current address, current contents, and waits for the user to input a new value. This value can be input using a numeric field using the current default radix or may be forced using the radix forces ("%" or "$"). The command will terminate when the user inputs either "." or "//" in reponse to the prompt.

The options are:

MA *expr*[:[+ -]*expr*]]	Modify the absolute address in bank 0.
MEA *bank.expr*[:[+ -]*expr*]]	Modify the absolute address in the specified bank.
MSY[+ -expr [expr]][:[+ -]expr]]	Modify the absolute address in sysglobal.
MDB[+ -expr [expr]][:[+ -]expr]]	Modify the absolute address in the DB register.
MDL[+ -expr [expr]][:[+ -]expr]]	Modify the absolute address in the DL register.
MQ [+ -expr [expr]][:[+ -]expr]]	Modify the absolute address in the Q register.

MS [+ -expr [[expr]]][:[[+ -]expr]] Modify the absolute address in the S register.

MZ [+ -expr [[expr]]][:[[+ -]expr]] Modify the absolute address in the Z register.

MPB[+ -expr [[expr]]][:[[+ -]expr]] Modify the absolute address in the MPB register.

MP [+ -expr [[expr]]][:[[+ -]expr]] Modify the absolute address in the P register.

MPL[+ -expr [[expr]]][:[[+ -]expr]] Modify the absolute address in the PL register.

Register Operations

The DR command will display the common registers (i.e., P, PB, PL, CIR, DB, Q, S, etc.). If no field is specified to DR then all common registers will be displayed.

The options are:

BR [reg / regfile[,count]]

MR {reg / regfile}

Execution Control

The options are :

E Exit back to Maintenance Mode

RUN Run (return to software)

Input/Output Operations

The forms of addresses and data paterns can be found in the HP3000 Series 37 Hardware Design Document Revision 'A'.

The options are:

RIO *ioaddr*	Read I/O from address *ioaddr*
WIO *ioaddr,iodata*	Write I/O address *ioaddr* with data *iodata*

Miscellaneous Commands

The T (Trace) command allows the user to trace the current (or specified) stack. ENV allows the user to move back markers of the current stack and access data there as if it were at the current marker. RDX allows the user to change the current Radix. Softpanel starts with the Radix set to octal. The ENV command specified with no parameters will turn ENV off (q-relative addresses revert to the current environment).

The options are:

T[[*numeric.*]*numeric*]	Trace stack.
ENV[*numeric*]	Change the environment.
RTOC *tocaddr*	Read TOC RAM address.
WTOC *tocaddr,tocdata*	Write TOC RAM with data.
RDX {*H* / *O*}	Change the current radix. (*H* - Hex; *O* - Octal)
ST	Give Softpanel status.

Memory Breakpoints

Using the SIMB breakpoint board in conjunction with the softpanel allows for breaking on absolute memory addresses for software debugging or tracing. The breakpoint capability includes two addresses, designated as A and B. These addresses can be set up as a range, allowing for breakpointing of a region of memory.

A data pattern is also available, which allows for breaking only on a particular pattern being written (read data patterns are not supported) to a memory word. This data pattern can consist of ones, zeros, and don't cares.

All memory breakpoint commands require the SIMB breakpoint board.

The commands for the memory breakpoint board are as follows:

dSMB [A:/B:] *bank.numeric*[,{*RD*/*WR*}][,*RA*]

Sets a memory breakpoint. Defaults to breakpoint A if not specified. *RD* and *WR* are to specify read or write only. The default is read/write. *RA* sets up range mode, and must be specified on the last address entered. For example, to set a range breakpoint for reads to address range 1000 to 2000 in bank 0, enter the sequence SMB A:0.1000 followed by SMB B:0.2000,RA.

CMB

Clears ALL memory breakpoints. (A and B can only be cleared together).

SMD *bit pattern*

Sets a data pattern for the breakpoint. This feature is particularly useful for the case of a particular data pattern showing up at random in a table. A single 16-bit word can be specified, made up of 0's, 1's, and X's (don't cares). For example, if the data pattern %20040 is showing up at random in some resident table in MPE, (virtual tables must be frozen using System Debug) then the sequence would be to do SMB A:<*address of start of table*> followed by SMB B: <address of end of table>, followed by SMD 0010000000100000. The bit pattern MUST contain a sequence of 16 ones, zeros, and xs.

The SMD command, when used, must follow the setting up of the range.

SB

Shows the current state of the breakpoints A and B, the ranging flag, and the data pattern.

CPU ROMS Date Code 3444 Exceptions

1. The Softpanel ENV command. One would expect ENV 0 to turn the ENV feature off. Instead, it prints "Whoops. S below Q! Env Aborted!". The workaround is to type ENV with no parameter to turn ENV off.

2. In the Softpanel: the RTOC command does not give a new line after printing the value of the requested TOC register. The display produced is thus somewhat confusing.

3. In the Softpanel: Any hexadecimal value starting with an alphabetic character (for example, "A","A00","C0") must be preceeded by either a zero or a dollar sign, even if the current radix is set to Hexadecimal. Imbedded letters in a hex value starting with a numeric (for example, "1C00") do not have this restriction.

The reason for this qualification for *numeric* in the hex radix is to distinguish between the DB register and the value "$DB".

4. In the Softpanel, the SMB command catches the condition where the SIMB breakpoint board is missing, but the SMD command does not.

5. In the Softpanel, the T (Trace) command to a non-existent s-bank does not print an error message.

6. In the Softpanel, the T (Trace) command will print the last user stack if the current stack is the ICS, but there is no way to use the ENV command to move back to the user stack.

7. In the Softpanel, using the RIO command to a non-existent register produces a watchdog timer interrupt, and the console comes back to the Maintenance Mode prompt.

8. In the Softpanel, the RTOC command issued to values outside the range $0 to $1F will return some value. It seems to be the TOC register address modulo $20 (decimal 32). (i.e. RTOC $5, RTOC $25, RTOC $45, etc. all return the value from TOC register 5).

9. In the Softpanel, the Help text makes reference to a DZ register. DZ really refers to the Z register in this line.

10. The switch register (SR) is not in the default register display. It may be displayed manually.

11. In the Software Display, the status register flags M, I, T, R, O, and C indicate a value of 1 with an upper case letter and a value of 0 with a lower case letter.

Error Codes

APPENDIX A

This appendix will cover all error codes associated with the Series 37 Self Test. The error codes are listed according to the test sequence they are associated with: TIC, PIC, CPU, or MEMORY.

TIC ERROR CODES

The TIC error codes are divided according to one of the nine sections of the TIC test that detected the error.

The Slot Test will display one of the following error codes if the TIC test fails. Note that the Tests are run in the same sequence as the test sections and that the test stops on the first failure found.

TIC Test Section 1

This is the initialization test. It will initialize the TIC and then check that the registers contain expected data.

Code	Error
0108	Register 8 initialization error (expected $0800)
0109	Register 9 initialization error (expected $0800)
010A	Register A initialization error (expected $FF00)
010C	Register C initialization error (expected $0000)
010D	Register D initialization error (expected $0000)
010E	Register E initialization error (expected $5004)
010F	Register F initialization error (expected $5004)

TIC Test Section 2

This section performs tests on: basic operations, test OBII, IPOLL, SMSK & RMSK.

Code	Error
0201	SMSK/RMSK test, TIC did not respond with mask bit set
0202	SMSK/RMSK test, TIC did not respond with mask bit cleared
0203	TIC did not set IRQ
0204	Improper IPOLL response (no, wrong, or multiple channels)
0205	Improper OBII response

TIC Test Section 3

The PORT register test. Test register 0-7 on ports 0-7.

Code	Error
0300	Port 0, register 0 pattern test failure
0301	Port 0, register 1 pattern test failure
.	
.	
0307	Port 0, register 7 pattern test failure
0310	Port 1, Register 0 pattern test failure
.	
.	
0377	Port 7, register 7 pattern test failure

TIC Test Section 4

DMA test, IMB write Word

Code	Error
0401	DMA state machine failed to go to state 4
0402	DMA state machine failed to go to state 3
0403	DMA write to memory transferred improper data
0404	DMA read from memory to RBYTE transferred improper data
0405	DMA read from memory to LBYTE transferred improper data
0406	DMA counter test failed

TIC Test Section 5

Port Self Test

Code	Error
0500	Port 0 Self Test failure
0501	Port 1 Self Test failure
.	
.	
0507	Port 7 Self Test failure

TIC Test Section 6

DMA loopback to port test

Code	Error
0600	Port 0 loopback failure
0601	Port 1 loopback failure
.	
.	
0607	Port 7 loopback failure

TIC Test Section 7

DMA loopback data test

Code	Error
0700	Port 0 loopback data failure
0701	Port 1 loopback data failure
.	
.	
0707	Port 7 loopback data failure

PIC TEST ERROR CODES

The Slot Test will display one of the following error codes if the PIC test fails. The Tests are run in the same sequence as the test sections and the test stops on the first failure found.

PIC Test Section 1

Initialization test, Initialize the PIC and then check that the registers contain expected data.

Code	Error
0104	Register 4 initialization error (expected $0000)
0105	Register 5 initialization error (expected $0000)
0106	Register 6 initialization error (expected $0020)
0107	Register 7 initialization error (expected $0000)
0108	Register 8 initialization error (expected $0000)
010A	Register A initialization error (expected $0000)
010F	Register F initialization error (expected $0087+(8*channel#))

PIC Test Section 2

Basic operations, test OBII, IPOLL, SMSK & RMSK.

Code	Error
0201	SMSK/RMSK test, PIC did not respond with mask bit set
0202	SMSK/RMSK test, PIC did not respond with mask bit cleared
0203	PIC did not set IRQ after SMSK on selected channel
0204	Improper IPOLL response after SMSK (no response, wrong response, or multiple channels responding)
0205	Improper OBII data from this channel after SMSK
0206	Improper Register D response after SMSK
0207	Improper IPOLL response (should have been clear)
0208	Recieved no CSRQ after issuing a SIOP
0209	Wrong channel is responding to SPOLL
020A	Improper OBSI data after SPOLL
020B	Improper Register F data after SPOLL
020C	Improper CSRQ response after HIOP (should have been Zero)
020D	Improper SPOLL after HIOP (should have been Zero)
020E	Improper OBSI response to SPOLL after HIOP (should have pointed to device 7)

PIC Test Section 3

Test ABI Status Register (PIC Register 1)

Code	Error
0301	Initialization error - bits 0 to 7 should be Zero
0302	Initialization error - bits 10, 13, and 14 should be Zero and bits 11 and 12 should be One
0303	Bit 13 should be 1, bit 14 should be 0 when the ABI is addressed to Talk, but not Listen.
0304	Bits 13 and 14 should both be 1 when the ABI is addressed to Talk and Listen.
0305	Bit 14 should be 1, bit 13 should be 0 when the ABI is addressed to Listen, but not to Talk.
0306	Bits 13 and 14 should both be 0 when the ABI is not addressed to Talk or to Listen.

PIC Test Section 4

Test ABI Interrupt Register (PIC Register 2) and Interrupt Mask Register (PIC Register 3)

Code	Error
0401	Reg2, Bit 0 should be set (an interrupt is pending).
0402	Reg2, Bits 9 and 13 should be clear (no handshake abort and inbound FIFO empty).
0403	Reg2, Bit 12 should be set (outbound FIFO room available).
0404	Reg2, Bit 14 should be set (outbound FIFO idle).
0405	Reg2, Bit 8 should be clear (no status change).
0406	Reg2, Bit 0 should be clear (no interrupt pending).
0407	Reg2, Bit 8 should be set (status change occurred).
0408	Reg2, Bit 9 should be clear (no handshake abort).
0409	Reg2, Bits 12 and 14 should be clear. Bit 13 should be set (outbound FIFO room not available, outbound FIFO not idle, and inbound FIFO not empty. ==> FIFO bits in opposite state).
040A	Reg2, Bit 9 should be set (handskake abort).
040B	Reg2, Bit 9 should be clear (no handshake abort).

PIC Test Section 5

Test PIC registers 4 & 5 by reading and writing data. Functions are not tested.

Code	Error
0501	Register 4 fails to show data patterns $00AA.
0502	Register 5 fails to show data patterns $00AA.
0503	Register 4 fails to show data patterns $0055.
0504	Register 5 fails to show data patterns $0055.

PIC Test Section 6

Test PIC register 6 (HPIB control register).

Code	Error
0601	Register 6 fails to show patterns $802A for Read/Write test.
0602	Register 6 fails to show patterns $4054 for Read/Write test.
0603	Bits 12 and 14 of the ABI Interrupt Register (PIC register 2) should be set (outbound FIFO NOT full and idle).
0604	Bits 12 and 14 of the ABI Interrupt Register (PIC register 2) should be clear (outbound FIFO full and NOT idle).
0605	Bit 15 of Register 6 should be set (clear outbound FIFO) via the ABI Interrupt Register test. Bits 12 and 14 of the ABI Interrupt Register should be clear.

PIC Test Section 7

Test PIC register 7 (HPIB address).

Code	Error
0701	Register 7 fails to show patterns $800A for Read/Write test.
0702	Register 7 fails to show patterns $4015 for Read/Write test.
0703	Bit 9 of Reg 7 should be set (Talk always) via the ABI Status Register (PIC reg 1) test. Bits 13 or 14 of the ABI Status Register should be off to indicate NOT a Talk or a Listen.
0704	Bit 10 of Reg 7 should be set (Listen always) via the ABI Status Register test. Bit 13 of the ABI chip Status Register must be on to indicate a Talk always.
0705	Same as 0704 except bit 14 of the ABI chip Status Register must be off to indicate NOT a Listen always.
0706	Same as 0704 except the ABI Status Register is again updated and bit 13 of the ABI Status Register must be on to indicate NOT a Listen always.

PIC Test Section 8

Test PIC register 8, 9 & 10 with read/write data only. Register 8 is only tested for lower 8 bits. Register 9 & 10 ($A) are tested for all 16 bits.

0801	Register 8 fails to show data patterns $00AA.
0802	Register 9 fails to show data patterns $AAAA.
0803	Register 10 fails to show data patterns $AAAA.
0804	Register 8 fails to show data patterns $0055.
0805	Register 9 fails to show data patterns $5555.
0806	Register 10 fails to show data patterns $5555.

PIC Test Section 9

Test DMA Write/Read/Abort from memory to PIC FIFO and from PIC FIFO to memory.

Code	Error
0901	CSRQ response & test via OBSI failed the DMA Write to PIC.
0903	Bit 9 of the Interrupt Register (PIC reg 2) should be clear (no handshake abort) the DMA Write Abort (PIC reg E) test.
0902	Data transferred to the PIC FIFO by the above transfer (assuming the DMA Write abort test passed) is incorrect.
0904	CSRQ response & test via OBSI failed the DMA Write Abort.
0905	Bit 5 of PIC register B should be set for the DMA Write Abort termination test.
0906	Bit 6 of PIC register B should be set for the DMA Write Abort termination test.
0907	CSRQ response & test via OBSI failed the DMA Read from PIC.
0908	Microword at $0390 fails to match the data patterns $1011.
0908	Microword at $0391 fails to match the data patterns $1213.
0908	Microword at $0392 fails to match the data patterns $1415.
0908	Microword at $0393 fails to match the data patterns $1617.
090A	CSRQ response & test via OBSI failed for the DMA Read Abort.
090B	Bit 5 of PIC register B should be set for the DMA Read Abort termination test.
090C	Bit 6 of PIC register B should be set for the DMA Read Abort termination test.

CPU TEST ERROR CODES

The CPU Test will display one of the following error codes if the CPU test fails. Note that the Tests are run in the same sequence as the test sections and that the test stops on the first failure found.

Test Section 1 - Bank Register tests

0101 Pbank read/Abank write failure
0102 Dbank read/Sbank write failure
0103 Sbank read/Dbank write failure
0104 Abank read/Pbank write failure

Test Section 2 - TOC RAM tests

0201 TOC RAM test not done, Pfail
0202 TOC RAM data failure

Test Section 3 - TOC count verification

0203 TOC not counting

Test Section 4 - MPE timer verification

0301 TOC not counting

Test Section 5 - LED display

There are no failure codes in this section.

Test Section 6 - Watchdog timer verification

0501 Watchdog timer did not rollover

MEMORY TEST ERROR CODES

The Memory Test will display one of the following error codes if the Memory Test fails. Note that the Tests are run in the same sequence as the test sections and that the test stops on the first failure found.

Code	Error
0001	Memory size test failed
0002	Memory Initialization/test failed
0003	unassigned
0004	Error Detection and Correction (EDAC) test failed
0005	Memory address test failed
0006	Memory pattern test failed
F0F0	Memory dead (watchdog timer failure)

Part No. 30457-90003
Printed in U.S.A. 11/84
E1184

HEWLETT PACKARD

HP 3000 Series 37

SERIES 37 MEMORY

Diagnostic Manual

HEWLETT PACKARD
8010 Foothills Blvd., Roseville, CA 95678

Part No. 30457-90007
E0984

Printed in U.S.A. 09/84

PRINTING HISTORY

New editions are complete revisions of the manual. Update packages, which are issued between editions, contain additional and replacement pages to be merged into the manual by the customer. The dates on the title page change only when a new edition or a new update is published. No information is incorporated into a reprinting unless it appears as a prior update; the edition does not change when an update is incorporated.

First Edition..... SEP 1984

LIST OF EFFECTIVE PAGES

The List of Effective Pages gives the date of the most recent version of each page in the manual. To verify that your manual contains the most current information, check the dates printed at the bottom of each page with those listed below. The date on the bottom of each page reflects the edition or subsequent update in which that page was printed.

Effective Pages	Date
all	SEP 1984

Table of Contents

GENERAL INFORMATION

SECTION 1

1.0 INTRODUCTION

This manual describes the diagnostic program for the Memory in the HP 3000 Series 37. The MDIAG37 will test all of memory and all control functions, and will force single and double-bit error detection and single-bit error correction.

This diagnostic is a tool that can be used to locate bad memory chips, verify memory control circuitry, and verify the correct operation of the Error Detection and Correction circuit (EDAC).

MDIAG37 is written in SPL and is run under the Diagnostic/ Utility System (DUS).

1.1 REQUIRED HARDWARE

The hardware required to run MDIAG37 is the HP 3000 Series 37 minimum configuration:

Console (connected to the TIC at slot 1, port 0)
HP-IB Tape Drive
SPU (with the following board set): CPU
Memory*
TIC
PIC**

*located in slot 2 of the SPU -- must work well enough to initiate MDIAG37
**needed as a coldload path to load DUS

1.2 REQUIRED SOFTWARE

The most recent revision of the following software is required:

DUS

1.3 DIAGNOSTIC PROGRAM STRUCTURE

The MDIAG37 is composed of tests that can be run individually or in any combination that you select. The selected tests can be looped a specific number of times or until you halt them. The tests always run in ascending numerical order.

Tests 1, 8, and 9 do NOT allow error looping.

1.4 TEST LIMITATIONS

This diagnostic is intended to isolate specific RAM and/or EDAC failures. The system to be tested needs to be functioning well enough to load DUS.

Multiple-bit errors will cause MDIAG37 to lose control over error reporting and the ability to recover from the interrupt, EXCEPT during Test Section 2. All other test sections will display the following message:

```
System Halt 6
CPU memory parity error -- multi-bit error.
```

The number of the test section in which the halt occurred will also be displayed.

1.5 TEST EXECUTION TIMES

The test execution times vary with the amount of memory.

OPERATING INSTRUCTIONS

SECTION 2

2.0 INTRODUCTION

This section of the manual describes how to operate the memory diagnostic (MDIAG37).

The primary function of MDIAG37 is to test and diagnose the HP 3000 Series 37 memory. The most important feature of MDIAG37 is that you control test selection, looping, error handling, and printing messages.

2.1 TEST SELECTION

MDIAG37 is divided into tests that can be run individually or in combinations. The order in which the tests run, however, is fixed. Use the TEST command to select tests.

Use the List Diagnostic Status (LDS) command to list which tests are currently selected. A "1" under the test number indicates that test section is enabled. A "0" indicates that test section is disabled.

The standard default selects all tests. If you want to run a simple "go/no go" version of MDIAG37, select the following test list:

1) Test Section 1 - Low Memory/Diagnostic Compatability Test

2) Test Section 2 - EDAC Test

3) Test Section 3 - Address Test

4) Test Section 6 - Move Data Test

5) Test Section 9 - Log Test

To change the test selection, enter TEST, then enter a "-" in front of the test number(s) to be deleted, or enter a "+" in front of the test number(s) to be added. Enter TEST and a list of test numbers without using a "-" or a "+" to select only those tests that you list.

2.2 LOOPING

You can select two types of looping when you configure MDIAG37.

The first type loops on the selected test list the number of times that you specify. If you do not enter a parameter, the selected tests will be looped until you halt them. Set the LOOPOFF option to discontinue looping. LOOPOFF also turns off the loop on error (LOOPERR) option. The loop count can be any number from 1 to 32,767.

The second type loops on failure. If specified, the failing test will loop continuously until you halt it with CNTRL-Y. At this point, the LOOPERR option is also turned off.

2.3 ERROR HANDLING

Error messages cannot be suppressed. However, you have the option of displaying the error messages on the console (default) or of routing them to a printer (PEMP). Make certain that the printer is connected before DUS is initialized. The printer connection is not verified by MDIAG37.

You also have the option to pause on error (default) or to continue with the diagnostic (SEPS).

Only the most recent single-bit error is saved because there is only one location per memory board for error logging. Compare data errors or errors found in the EDAC test are all displayed.

2.4 PRINTING MESSAGES

There are two types of messages: error and information. Error messages tell you that the memory has failed to respond properly to a test. Error messages cannot be suppressed, but can be directed to a printer instead of to the console (PEMP). Information messages can be suppressed (SNDP).

2.5 HOW TO RUN MDIAG37

Input to MDIAG37 is through the system console, after Test Section 1 has run or during program pauses.

Run the memory diagnostic in the following manner:

1) Perform MPE shutdown to log off all users, if applicable.

2) Run the console selftest.

3) Fully reset the console.

4) Install a DUS tape in the cold-load device.

5) If the system is off, power it on by turning the keyswitch to the LOCAL or REMOTE position. If the system is already on, verify that the keyswitch is in the LOCAL or REMOTE position. Use the appropriate load command to load DUS.

6) DUS displays a welcome message and a prompt:

```
DIAGNOSTIC/UTILITY SYSTEM (REVISION nn.nn)
ENTER YOUR PROGRAM NAME (TYPE HELP FOR PROGRAM INFORMATION)
:
```

7) Type MDIAG37 in response to the prompt.

8) The memory diagnostic loads and executes Test Section 1. The following message is then displayed:

```
HP 3000 Series 37 Memory Diagnostic - (MDIAG37 nn.nn XX/XX/XX)
Begin Section 1

Begin Step 11
If step 11 terminates with an error then there are fatal
multiple-bit errors in low memory and the diagnostic will fail
to run properly.

Step 11 completed

Begin Step 12

Step 12 completed

Begin Step 13

Detected X banks of memory on X boards using 64K RAMS

Step 13 completed

Begin Step 14

Step 14 completed

End of Section 1

Type 'GO' to Continue (HELP to list commands)
>
```

9) At this point, GO will execute the default tests (all tests) after Section 1. To change the test selection, enter TEST with a "-" in front of the test numbers to delete those tests, or a "+" in front of the test numbers to add those tests. Without a "-" or a "+" MDIAG37 will perform just those test sections listed.

Examples: TEST -5,6,7 or TEST -5/7 will delete tests 5-7
TEST +2,3,4 or TEST +2/4 will add tests 2-4
TEST 1,2,3,6,9 to run tests 1,2,3,6, and 9 (go/no go)

If you inadvertently select no tests by deleting test sections, MDIAG37 will select all tests (default).

10) Unless the End of Program Pause has been enabled, control is returned to DUS when the diagnostic has been executed.

2.6 HOW TO CONFIGURE MDIAG37

You can configure and reconfigure the diagnostic according to your needs.

MDIAG performs initialization and Section 1, followed by a prompt for an operator command. If you are going to use the default commands (indicated with an asterisk in Table 2-1), type GO in response to the prompt. If you want to reconfigure the diagnostic, you can do so at this point or after you interrupt the diagnostic. Table 2-1 provides the available commands.

Table 2-1. DIAGNOSTIC COMMANDS

COMMAND	NAME	DESCRIPTION
EEOPP*	Enable End Of Program Pause	allows end of program pause
EEPS*	Enable Error Pause	halts after error occurs
ENDP*	Enable Non-error Display	allows information messages
EXIT	Exits the diagnostic	returns to DUS
GO	resumes program at current step	allows diagnostic to continue at current step
HELP	lists commands	lists available commands with a brief description
LDS	List Diagnostic Status	Lists the following information: which test sections are enabled ENDP flag: true or false EEPS flag: true or false EEOPP flag: true or false LOOPTST flag: true or false LOOPERR flag: true or false PEMP flag: true or false number of executed diagnostic passes
LOOPTST (n)	Loop on current Test list	May enter a repetition factor. If a parameter is blank, the loop continues until LOOPOFF is set.
LOOPERR	Loop on test with Error	Loops on failed test section. Halt LOOPERR with CNTRL-Y. LOOPERR is not available in test sections 1, 8, and 9.

Table 2-1. DIAGNOSTIC COMMANDS

COMMAND	NAME	DESCRIPTION
LOOPOFF*	Do not Loop	overrides LOOPTST and LOOPERR
LT	Lists Tests	lists available tests with a brief description
PEMP	Print Error Messages to Printer	uses printer that is defined in DUS I/O table
PEMC*	Print Error Messages to Console	
RUN	restart program from section 1	allows you to restart the diagnostic
SEOPP*	Suppress End of Program Pause	returns to DUS
SEPS	Suppress Error Pause	does not pause after error
SNDP	Suppress Non-error Display	suppresses informational messages
TEST (n)	Specify Test(s) to be executed	allows you to select tests with a +, delete tests with a - NOTE: Refer to the following section for test descriptions.

*default configuration

TEST DESCRIPTIONS

SECTION 3

3.0 INTRODUCTION

This section of the manual describes the test sections of the memory diagnostic (MDIAG37).

3.1 TEST SECTION 1

Low Memory/Diagnostic Compatability Test

This section tests low memory and diagnostic compatability, reads the syndrome information, and checks memory configuration. It automatically executes when you call up MDIAG37. After completion, this test section allows the operator to change the configuration of the diagnostic.

Appendix A contains a table that decodes the syndrome codes.

NOTE

This section does NOT allow the Loop on Error option.

Step 11. Low Memory Test -- This step reads the lowest 256K bytes of memory (banks 0 and 1) to check for double-bit hardware errors. If there are any double-bit errors in bank 0 or bank 1, MDIAG37 will not run. This step does NOT have the Loop on Error option. The displayed failure indication will be a multiple-bit error system halt:

```
System Halt 6
CPU memory parity error -- multi-bit error.
```

Step 12. Diagnostic Compatability -- This step checks that the system you are testing has the correct CPU. This step does NOT have the Loop on Error option. If an incorrect CPU is detected, the following message is displayed:

```
This diagnostic is only for an HP 3000 Series 37 System.
```

Step 13. Memory Configuration -- This step determines the size of memory by reading a location in each consecutive bank until a Memory Bounds Violation occurs. This information is displayed by the following message:

```
Detected XX banks of memory on Y boards using ZZZK Rams.
```

Where: XX = 1-32 (indicating the number of banks)
Y = 1/2
ZZZ = 64/256

If no valid last bank number is found, the following error message is displayed:

```
Last Bank No. Invalid
```

If a Memory Bounds Violation is not detected, the following error message is displayed:

```
Memory Bounds Violation not detected - default ending bank = 3.
```

This step does NOT have the Loop on Error option.

Step 14. Clear Memory Status -- This step reads the memory status of 1 or 2 boards to clear the error syndrome code. It does NOT have the Loop on Error option.

3.2 TEST SECTION 2

EDAC Test

This section performs a simple pattern test on all memory boards present. It then checks the Error Detection and Correction Circuit (EDAC) to ensure that the board will function correctly with single-bit errors present.

Step 21. Simple Pattern Test -- This step does a simple pattern test before beginning the EDAC test to detect faulty RAMs. Since the EDAC facility cannot be shut off, this is the only way to ensure that the EDAC test is using locations that are free of errors.

Error message:

```
Single-bit error detected
Board:  X
Syndrome Code:  %YYY
Chip Number:  UZZZZ
```

Where: X = 0/1
YYY = Octal Syndrome Code
UZZZZ = Reference Designator of faulty RAM

Step 22. EDAC Test -- This step verifies that single-bit error detection and correction is performed correctly. The 32 single-bit errors that can occur are generated. Proper error correction is then checked. The syndrome latch is checked for proper error logging, causing the latch to clear. Finally, a check is made to verify that the syndrome latch is cleared.

Error messages:

- `Multi-error was detected during single-bit error test;`
  `Board: Y; Block: Y`

  Where: Y = 0/1

- `Error in test word was not corrected; data bit XX.`
  `Board: Y; Block: Y`
  `Data expected: %AAAAAA; Data received: %ZZZZZZ`

  Where: 0 = < = XX < = 31
  Y = 0/1
  AAAAAA and ZZZZZZ = octal data

- `Syndrome code expected was not returned; data bit XX.`
  `Board: Y; Block: Y`
  `Expected: %AAA; Received: %ZZZ`

  Where: 0<=XX<=31
  Y = 0/1
  AAA and ZZZ = octal syndrome codes

- `Syndrome latch was not cleared after status was reported;`
  Board: Y

  Where: Y = 0/1

Step 23. EDAC Test - Double-bit Errors -- This step will verify double-bit error detection and correction. Thirty random double-bit errors are generated and checked for proper handling. The syndrome latch is checked for proper error logging.

Error messages:

- `Multi-error was detected during double-bit error test.`
  `Board: Y; Block: Y`

  Where: Y = 0/1

- `Syndrome code expected was not returned;`
  `Board: Y; Block: Y`
  `Expected: %AAA; Received: %ZZZ`

  Where: Y = 0/1
  AAA and ZZZ = octal syndrome codes

- `Syndrome latch was not cleared after status was reported;`
  `Board: Y`

  Where: Y = 0/1

3.3 TEST SECTION 3

Address Test

Step 31. Address Test -- This step will write the "exclusive or" of the bank and address of every available location into itself. Each location is then read and verified. This procedure is repeated, using the compliment of the "exclusive or" of the bank and address of every location as data. If this test fails, the unique address capability may have failed.

The following messages will appear during test execution:

```
Begin Section 3

Begin Step 31

All of tested memory has been written

Pass 1 completed - Begin Pass 2

All of tested memory has been written

Step 31 completed

End of Section 3
```

Error message:

```
Expected:  %XXXXXX; Received:  %YYYYYY
Address = %ZZZZZZ
Bank = W
Board = A
```

Where: XXXXXX = data expected in octal
YYYYYY = data received in octal
ZZZZZZ = address of error in octal 0 < = Z < = 177777
W = bank with error 0 < = W < = 31
A = 0/1

3.4 TEST SECTION 4

Alternating Ones and Zeros Test

Step 41. Alternating Ones and Zeros Test -- This step writes an alternate one and zero pattern into all available memory locations in ascending address order and then reads them back. The complement pattern is then written and read back. The error latch is read after testing each pattern to check whether any single bit errors were detected.

The following message will appear during test execution:

```
Begin Section 4

Begin Step 41

All of tested memory has been written

Pass 1 completed - Begin Pass 2

All of tested memory has been written

Step 41 completed

End of Section 4
```

The following error message will be displayed only if error correction is NOT working:

```
Expected:  %XXXXXX; Received:  %YYYYYY
Address = %ZZZZZZ
Bank = W
Board = A
```

Where: XXXXXX = data expected in octal
YYYYYY = data received in octal
ZZZZZZ = address of error in octal 0 < = Z = 177777
W = bank with error 0 < = W < = 31
A = 0/1

The error latch information will be displayed if an error was detected during the test. Bad memory chips are identified only by the error latch information.

Error message:

```
Single-bit error detected
Board:  X
Syndrome Code:  %YYY
Chip Number:  UZZZZ
```

Where: X = 0/1
YYY = octal syndrome code
UZZZZ = Reference Designator of faulty RAM

3.5 TEST SECTION 5

Data Pattern Test

Step 51. Data Pattern Test -- This step writes 64K data patterns into one 32-bit memory word in each block. This function attempts to access each chip on each board in order to point out any malfunctioning chips. The data patterns are generated by using the 64K possible patterns in the lower 16 bits and the one's complement of the pattern in the upper 16 bits.

The following message will be displayed during test execution:

```
Begin Section 5

 Begin Step 51

 Test is half way to completion

 Step 51 completed

 End of Section 5
```

The following error message will be displayed only if error correction is NOT working:

```
Expected:  %XXXXXX XXXXXX;  Received:  %YYYYYY YYYYYY
Address = %ZZZZZZ
Bank = W
Board = A
```

Where: XXXXXXXXXXX = data expected in octal
YYYYYYYYYYY = data received in octal
ZZZZZZ = address of error in octal 0 < = Z < = 177777
W = bank with error 0 < = W < = 31
A = 0/1

The error latch information is displayed if an error was detected during the test. Bad memory chips are identified only by the error latch information.

```
Single-bit error detected
Board:  X
Syndrome Code: %YYY
Chip Number:  UZZZZ
```

Where: X = 0/1
YYY = octal syndrome code
UZZZZ = Reference Designator of faulty RAM

3.6 TEST SECTION 6

Move Data Test

Step 61. Move Data Test -- This step uses the lower 32K-bytes of memory as a data bank and copies it to every other 32K-byte block. The error latch is checked after each 32K-byte move operation. This step uses the Move Absolute Instruction for speed and simplicity. This step is repeated, using the 32K-bytes of each block as the data to write back into that block.

Error message:

```
Single-bit error detected
Board:  X
Syndrome Code:  %YYY
Chip Number:  UZZZZ
```

Where: X = 0/1
YYY = octal syndrome
UZZZZ = Reference Designator of faulty RAM

3.7 TEST SECTION 7

March Ones and Zeros Test

Step 71. March Ones/Zeros -- This step marches first ones and then zeros through each 32K-byte block of memory. Each 32K-byte block of memory is written to all zeros. Then each location is read for a zero and then written to all ones. When the block contains all ones, the process is repeated by reading each location for all ones and then writing a zero. The error latch is checked after each 32K-byte block of memory is completed.

The following error messsage will be displayed only is error correction is NOT working:

```
Expected: XXXXXX;Received: YYYYYY
Address:  %ZZZZZZ
Bank = W
Board = A
```

Where: XXXXXX = data expected in octal
YYYYYY = data received in octal
ZZZZZZ = address of error in octal 0 < = Z < = 177777
W = bank with error 0 < = W < = 31
A = 0/1

The error latch information will be displayed if an error was detected during the test. Bad memory chips are identified only by the error latch information.

```
Single-bit error detected
Board:  X
Syndrome Code:  %YYY
Chip Number:  UZZZZ
```

Where: X = 0/1
YYY = octal syndrome code
UZZZZ = Reference Designator of faulty RAM

3.8 TEST SECTION 8

Low Memory Test

This section relocates the program and checks the memory area where the program was originally located (banks 0 and 1). It is then relocated back. This section will NOT respond to the Loop on Error option and does NOT provide error messages until the last step is completed. In addition, it will NOT respond to CNTRL Y or any other type of I/O.

Step 81. Program Relocation -- This step will relocate MDIAG37, DUS, stacks, and the Device Reference Table (DRTS) into Banks 2 and 3 in order to test the area in lower main memory, where the program was originally located.

Step 82. Marching Ones and Zeros -- This step is identical to Step 71. These locations are written and read by the diagnostics. Any errors are recorded in the error latch. Because of limitations imposed by the relocation of the diagnostic, only the last error encountered by the marching test will be reported.

Step 83. Program Re-Relocation -- This step relocates the program back to its original area in lower main memory. The memory status is checked and any errors that were encountered in the marching test (Step 82) are now displayed.

3.9 TEST SECTION 9

This text section executes only upon exiting MDIAG.

Log Test

Step 91. Log Test -- This step will check the error latch to make sure it has been cleared before the diagnostic returns control to DUS. This section will NOT respond to the Loop on Error option.

Error message:

```
Single-bit Error Detected
Board:  X
Syndrome Code:  %YYY
Chip Number:  UZZZZ
```

Where: X = 0/1
YYY = octal syndrome code
UZZZZ = Reference Designator of faulty RAM

SYNDROME CODES

APPENDIX
A

Table A-1. Single-bit Error Syndrome Codes

OCTAL SYNDROME CODE	BIT NUMBER
% 14	data bit 0
% 254	data bit 1
% 264	data bit 2
% 324	data bit 3
% 24	data bit 4
% 344	data bit 5
% 44	data bit 6
% 104	data bit 7
% 72	data bit 8
% 132	data bit 9
% 232	data bit 10
% 152	data bit 11
% 252	data bit 12
% 312	data bit 13
% 162	data bit 14
% 322	data bit 15
% 216	data bit 16
% 56	data bit 17
% 66	data bit 18
% 126	data bit 19
% 226	data bit 20

Table A-1. Single-bit Error Syndrome Codes

OCTAL SYNDROME CODE	BIT NUMBER
% 146	data bit 21
% 246	data bit 22
% 306	data bit 23
% 270	data bit 24
% 330	data bit 25
% 30	data bit 26
% 350	data bit 27
% 50	data bit 28
% 110	data bit 29
% 360	data bit 30
% 120	data bit 31
% 176	check bit 0
% 276	check bit 1
% 336	check bit 2
% 356	check bit 3
% 366	check bit 4
% 372	check bit 5
% 374	check bit 6

NOTE: (1) There is a decimal/octal/hexidecimal conversion chart at the beginning of this diagnostic manual set.

(2) The syndrome codes in this table are transposed and then shifted one bit to the left from those listed in the manufacturer's data sheet for the EDAC chip. This octal syndrome code is the same for block 0 and block 1.

HP 3000 Series 37

TERMINAL INTERFACE CONTROLLER

Diagnostic Manual

HEWLETT PACKARD
8010 Foothills Blvd., Roseville, CA 95678

Part No. 30457-90004
E0984

Printed in U.S.A. 09/84

PRINTING HISTORY

New editions are complete revisions of the manual. Update packages, which are issued between editions, contain additional and replacement pages to be merged into the manual by the customer. The dates on the title page change only when a new edition or a new update is published. No information is incorporated into a reprinting unless it appears as a prior update; the edition does not change when an update is incorporated.

First Edition..... SEP 1984.....

LIST OF EFFECTIVE PAGES

The List of Effective Pages gives the date of the most recent version of each page in the manual. To verify that your manual contains the most current information, check the dates printed at the bottom of each page with those listed below. The date on the bottom of each page reflects the edition or subsequent update in which that page was printed.

Effective Pages	Date
all	SEP 1984

Table of Contents

GENERAL INFORMATION

SECTION 1

1.0 INTRODUCTION

This manual describes the diagnostic program for the Terminal Interface Controller (TIC). The TICDIAG is designed to verify correct operation of all functions of the TIC. The TIC is part of the Advanced Terminal Processor for the Series 37 (ATP37).

The TICDIAG is written in SPL and runs under Diagnostic/Utility System (DUS).

1.1 REQUIRED HARDWARE

The hardware required to run the TICDIAG is the HP 3000 Series 37 minimum configuration:

Console (connected to the TIC at slot 1, port 0)
HP-IB Tape Drive
SPU (with the following board set): CPU
Memory
PIC*
TIC**

*needed as a coldload path to load DUS
**must work well enough to initiate the TICDIAG

To test port 7 (the modem port) on the TIC, a modem loopback hood (HP 30146-60002) is required. A different hood is required for each type of interface. The RS-232 interface requires part number HP 30148-60002. The RS-422 interface requires part number HP 30147-60002.

1.2 REQUIRED SOFTWARE

The most recent revision of the following software is required:

DUS

1.3 DIAGNOSTIC PROGRAM STRUCTURE

Each test that is executed by the TICDIAG is a separate module. This enables you to select the tests to be run.

The TICDIAG routines are divided into two major groups in order to adequately test the hardware. The two groups are the tests for the System Interface Board (SIB - actually a section of the TIC board) and the tests for the Asynchronous Interface Board (AIB - actually a section of the TIC board). The tests in the SIB section test the circuitry between the SIMB and the L-Bus. They affect all ports. The tests in the AIB section test the circuitry from the L-Bus to the cable connector and affect only one selectable port at a time.

The TICDIAG loop is very simple. It calls Get'Test'Data to get input from the user. The SIB is then initialized and the SIB tests, if selected, are run. If the SIB section is successfully completed, the AIB is initialized for testing and the selected tests are run. Some of the data from Get'Test'Data is used by the main program, but most of the data is passed directly to the Test' SIB and Test'AIB sections.

Figure 1-1 is a structure chart of the main program body.

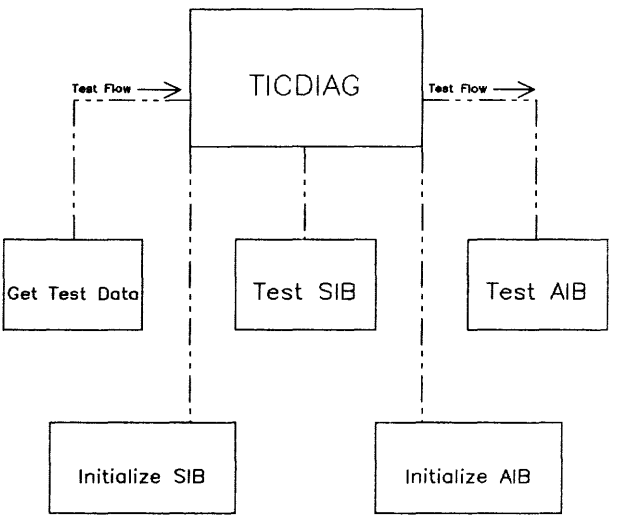

Figure 1-1. TICDIAG Structure

1.4 TEST FLEXIBILITY

The TIC PCA can be installed in any of the HP 3000 Series 37 system configurations. The TICDIAG provides the environment to test the TIC PCA while in any system configuration. Figure 1-2 describes which TICDIAG tests are allowed with a particular configuration.

Channel Number of TIC Under Test	Output Directed To	Channel Number of Console	Tests Allowed
1	Console	1	All AIB Tests Except Port 0 (Stop on error allowed)
Other than 1	Console	1	All Tests (SIB and AIB) (Stop on error allowed)
1	Printer	1	All Tests (SIB and AIB) (No control-y break)
Other than 1	Printer	1	All Tests (SIB and AIB)
1	Console	Other than 1	All Tests (SIB and AIB) (Stop on error allowed)

Figure 1-2. TICDIAG Configurations

1.5 TEST LIMITATIONS

The Remote Operator Interface (ROI) circuitry of the TIC PCA is not tested by the TICDIAG. This circuitry includes the LOCAL/REMOTE circuitry and the Disconnect circuitry. The functions of the ROI circuitry can be tested through a manual procedure.

Control B and warmstart circuitry are not tested by the TICDIAG.

OPERATING INSTRUCTIONS

SECTION 2

2.0 INTRODUCTION

This section describes how to operate the TICDIAG.

The TICDIAG is composed of routines that test a specific function of the TIC PCA. These routines can be run individually or in various combinations. The SIB tests, if selected, run before the AIB tests. The order of the tests is maintained regardless of which tests are run because each test relies on the circuitry that has been tested by the previous routine.

You can specify the type of information to be received from the TICDIAG and can display that information at either the console or the printer. The options are to print pass messages, failure messages, or troubleshooting messages. Field personnel will most likely use the pass and failure messages. The troubleshooting messages aid in the isolation of faulty components.

All input to the TICDIAG is done via the system console.

2.1 LOOPING

You can select one of two types of looping.

The first type loops the specified number of times that you enter in response to a prompt. The loop count can be from 1 to 32,767. Continuous looping can be specified by entering a 0. The SIB tests are run the specified number of times and then the AIB tests are run the specified number of times. Consequently, continuous looping cannot be specified if both SIB and AIB tests are selected.

The second type loops on failure. If specified, the failing test will be looped continuously.

2.2 TIC DIAGNOSTIC COMMANDS

The TICDIAG prompts you for the testing parameters. At each prompt, you can type HELP and a help message will be displayed. The help message displays the acceptable responses and the resulting action of those responses. You can also type EXIT in response to any prompt to have the TICDIAG stop testing.

Input to the TICDIAG is through the system console.

Run the TICDIAG in the following manner:

1) Perform MPE shutdown to log off all users, if applicable.

2) Run the console selftest.

3) Fully reset the console.

4) Install a DUS tape.

5) If loopback testing is desired, attach a hardware jumper between the loop test point and the ground test point near the backplane connector or attach the loopback hood(s) on the desired port(s).

6) Power-on the system by turning the keyswitch to one of the On positions. Use NORMAL if the console is port 0 of channel 1. Use LOCAL or REMOTE only if a TIC other than the one in channel 1 is to be tested. The microdiagnostic will run. You must then reply to the prompt. Type H for HELP. Type L (LOAD) channel number, device number. DUS will load.

   Or, switch to LOCAL and use a Control B sequence to load DUS, following the above procedure. Return the keyswitch back to NORMAL.

7) The Diagnostic/Utility system will display a welcome message and a prompt:

```
DIAGNOSTIC/UTILITY SYSTEM (REVISION nn.nn)
ENTER YOUR PROGRAM NAME (TYPE HELP FOR PROGRAM INFORMATION)
:
```

8) Type TICDIAG in response to the prompt.

9) The TICDIAG loads and executes. The following message is displayed:

   ```
   TERMINAL INTERFACE CARD OFFLINE DIAGNOSTIC        V - nn.nn
   ```

   Enter HELP in response to any question to receive an explanation of what is required.

   Enter EXIT in response to any question to terminate the program.

   Enter STATUS in response to any question to display the current diagnostic configuration.

10) The TICDIAG then prompts you for all information needed to run the tests. The first prompt asks for the channel number of the TIC under test. The channel number will be 1-5 for a basic system, or 1-5,9-13 if an extender is used. If the channel number is 1, the TICDIAG will later prompt you to redirect the printout. If redirected, you must select Port 0.

    ```
    ENTER THE CHANNEL NUMBER OF THE TIC UNDER TEST:
    ```

11) TICDIAG next prompts for the amount of information that you wish to have printed. Answer these prompts with a YES or a NO:

```
PRINT FAILURE MESSAGES  (DEFAULT = NO)?

PRINT PASS MESSAGES  (DEFAULT = NO)?

PRINT TROUBLESHOOTING MESSAGES  (DEFAULT = NO)?
```

12) The TICDIAG will then prompt you for the list device. If the line printer is not specified by responding YES to the off-line prompt, the listing is sent to the system console. You can change the system console if the TIC under test has the system console on it and there is another TIC board in the system.

```
OUTPUT RESULTS TO LINE-PRINTER  (DEFAULT = NO)?
```

If the system console is on the TIC under test and the output is not directed to the line printer, TICDIAG will ask if you wish to change the system console to another TIC and port.

```
CHANGE THE SYSTEM CONSOLE  (DEFAULT = NO)?
```

If answered YES:

```
CHANNEL NUMBER OF TIC TO SWITCH THE CONSOLE TO:
```

13) If the console is the list device, you can select to stop on errors.

```
STOP ON ERRORS  (DEFAULT = NO)?
```

If you answered Yes, then when an error is detected, you are prompted to either cancel testing or to continue.

```
DO YOU WISH TO HALT THE TESTS  (DEFAULT = NO)?
```

14) You can select to loop on a failure. The TICDIAG will continuously loop the test that detected a failure.

```
LOOP ON ERROR  (DEFAULT = NO)?
```

15) The user is prompted for a loop count to specify the number of times that each test is repeated. The maximum value is 32,767. Zero indicates continuous looping.

```
LOOP COUNT - (ZERO FOR CONTINUOUS LOOPING/DEFAULT = 1):
```

16) TICDIAG prompts for the SIB tests to run. If the console is on the TIC under test, no SIB test can be run. Enter the tests by name or specify ALL or NONE.

```
ENTER SIB TESTS TO BE RUN: (DEFAULT = THE LAST SET OF TESTS RUN, IF ANY)
 >
```

The possible test names are:

```
IMB-SIB COMMUNICATIONS (IM)
INITIAL REGISTER VALUES (IN)
ERROR CIRCUITRY (E)
RESET/INIT (RES)
IRQ LOGIC (IR)
MEMORY (M)
SEQUENCING (SE)
TIMEOUT (TIM)
BOARD ENABLE (BO)
PORT POINTER (P)
FREEZE (FR)
TIC BUS BUFFERS (TIC)
DIRECT COMMAND (DI)
FIFO (FI)
STATE COUNTER (ST)
DMA ADDRESS COUNTER (DM)
COMP/CTR LOOPBACK (C)
READ IMB (REA)
WRITE IMB (W)
BEGIN FLAG/LEFT-RIGHT FLAG (BE)

ALL (AL)
NONE (N)
(RETURN)
```

The test names can be abbreviated as indicated. You can enter a list of tests, separated by semicolons. ALL indicates that all available SIB tests are to be run. NONE indicates that testing of the SIB will not be done. If you press Carriage Return, the last SIB tests specified (if any) will be retained.

ALL is the recommended response when using the diagnostic in the field.

17) TICDIAG prompts for the AIB tests to be run.

```
ENTER AIB TESTS TO BE RUN: DEFAULT = THE LAST SET OF TESTS RUN, IF ANY)
 >
```

The possible tests are:

```
PCC SELF TEST (PCCS)
PCC DUMP (PCCD)
MCC SELF TEST (MCCS)
MCC DUMP (MCCD)
PCC MCC COMMUNICATION (PCCM)
DIAGNOSTIC LOOPBACK (D)
BAUD RATE (B)
MODEM SIGNALS (MO)
MSC PORT ADDRESSING (MS)
LOOPBACK (L)

ALL (A)
ALL NO LOOPBACK (ALLN)
NONE (N)
(RETURN)
```

The test names may be abbreviated as indicated. You can enter a list of tests, separated by semicolons. ALL indicates that all available AIB tests are to be run. ALL NO LOOPBACK indicates all AIB tests are to be run, except those that require a loopback hood (Modem Signals test, Modem Address test, and Loopback test). NONE indicates that testing of the AIB is not to be done. Pressing carriage return in response to this question will cause the last AIB tests specified (if any) to be retained.

ALL NO LOOPBACK is the recommended response when using the diagnostic in the field. Enter ALL if loopback connectors have been installed.

18) If any AIB tests are selected, you are prompted for the numbers of the ports that need to be tested. Those ports which still have devices attached to them should not be selected. If the console is on the TIC under test, port 0 cannot be specified.

```
ENTER PORTS TO BE TESTED: (DEFAULT = THE LAST SET OF PORTS TESTED)
 >
```

Ports to be tested are specified by selecting a list of the port numbers. The format is:

<integer>[/<integer>][,<integer>[/<integer>],...]

If the list of ports is omitted, the last set of ports that you selected will be tested. The lists of ports can include individual numbers and ranges of numbers.

For example: 1/3,5,7 would be a valid list of numbers. It would specify that ports 1,2,3,5, and 7 are to be tested.

19) TICDIAG then proceeds to test the TIC PCA. Upon completion, a message is displayed. During the execution of the TICDIAG, the messages specified by you are printed to the list device.

```
TESTING COMPLETED.  CHANNEL/LOOP COUNT/OUTPUT CHANGES  (DEFAULT = NO)?
```

You can run the test again and change the parameters by responding YES and then answering the prompts.

TEST DESCRIPTIONS

SECTION 3

3.0 INTRODUCTION

This section of the manual describes the test sections.

3.1 GET TEST DATA

Get'Test'Data asks for the test parameters that you want to input. Because this may not be the first time that this routine is called, you are given the option to retain most of the parameters that you have previously specified. You are then prompted for the tests to be run and the ports to be tested.

3.2 SYSTEM INTERFACE BOARD (SIB) TESTS

This test section is composed of tests that relate to the SIB. The SIB is actually the system interface section of the TIC board. The test returns a logical value that indicates whether the tests succeeded and if the AIB tests should begin.

Test'SIB calls the procedure for each test that is specified by the parameters passed from Get'Test' Data. Check'Error, Print'Error'Prefix, and Print'Error'Status are utility procedures that are invoked by all of the tests. Test'SIB loops through its testing sequence the number of times specified by the loop count passed from Get'Test'Data.

Test 1. IMB-SIB COMMUNICATION -- Tests the SIB to insure that all allowable Intermodule Bus (IMB) commands work. As each IMB command is executed, two types of error messages can be printed. The first type indicates that the instruction was not successful. The second type indicates that the SIB response was not the expected response.

Test 2. INITIAL REGISTER VALUES -- Tests the values that are obtained from the SIMB registers after initialization. The value of each register to be checked is read. This value is then compared with the expected value for the register. If there is not a match, an error message is output.

Test 3. ERROR CIRCUITRY -- Tests the error circuitry on the SIB. It issues an illegal command and verifies that it is detected. It then checks the Error Clear mechanism of Register !A.

Test 4. RESET/INIT -- Checks to make certain that an INIT command causes a reset of the registers and of the error detection circuitry. First, the loopback bit of Register !A, the Diagnostic Control Register, is written with a "0". An INIT command is issued and should set the loopback bit. Register !A is read to confirm this. Next, an illegal address command is issued and the error bit is set. An INIT is issued to clear the error bit. This tests the operation of the MRSTA line.

Test 5. IRQ LOGIC -- Tests the interrupt logic on the SIB. First, the external interrupts are turned off. An SMSK command is issued, disabling the TIC, and then an illegal address command is issued. The illegal command should generate an error (but not an interrupt, since the mask is disabled). The mask bit is checked by performing an RIOA of the channel with an RMSK command. This results in the hardware mask being returned and not the memory image. An IPOLL is then issued to verify that the IRQ line is not set. The interrupt mask is now set to enable the TIC. The process is repeated, and the IRQ is checked to verify that it is set.

Test 6. MEMORY -- Tests the SIB memory by running a checkerboard test (writing to each register of each port with the Memory Diagnostic bit set and then cleared) and an address test. The SIB memory is used as the DMA registers for the I/O ports.

Test 7. SEQUENCING -- Single-steps the L-Bus State Machine and then checks the state after each cycle. An error message is output if the test fails.

Test 8. TIMEOUT -- Tests the L-Bus timeout logic, which terminates the wait for either BusEnd from the L-Bus, or HALT from the DMA state machine. The timeout occurs when the L-Bus handshake does not finish in the required time limit.

Test 9. BOARD ENABLE -- Tests the board enable logic that is on the SIB section of the TIC. This logic includes the board enable register and the L-Bus State Machine. The tests are as follows:

1) An INIT command is issued to clear the board.

2) The Diagnostic Control Register is set up to single-step and turn LOOPEN off.

3) Single-steps the board and checks that the FPLA (L-Bus State Machine) cycles between States 0 and 4.

4) Enables the board by writing to Register 8, the Board Enable Register.

5) Single-steps to make certain that the L-Bus State Machine goes to State 1 after State 4.

6) Turns on LOOPEN and completes the L-Bus Cycle.

Test 10. PORT POINTER -- Tests the Port Pointer logic. This is done as follows:

1) An INIT command is issued to clear the board.

2) Sets up the Diagnostic Control Register for single-stepping and LOOPEN off.

3) Sets up an illegal port pointer in the Port Pointer Register (8).

4) Single-steps the L-Bus State Machine and checks that BUSGO is not asserted.

5) Waits an appropriate amount of time and checks that a timeout has occurred. The L-Bus State Machine waits for the handshake in reply to the BUSGO.

Test 11. FREEZE -- Tests the actions of the freeze bit of Register 1A (the Diagnostic Control Register). This bit causes the L-Bus State Machine to cycle between states 0 and 4, and causes the PCCs to halt.

Test 12. TIC BUS BUFFERS -- Tests the L-Bus buffers. It uses DMA test code to write data to the L-Bus and read the looped back data. Then the looped back port pointer data is checked for a match with the value written into the Port Pointer Register

Test 13. DIRECT COMMAND -- Tests the data paths that carry a direct command to the L-Bus. The testing sequence follows:

1) Writes an INIT to reset the board and clear the FIFO.

2) Sets up the Diagnostic Control Register (Register !A) with timeouts disabled, ROM Test 2, LOOPEN off, and Free Running. The State Machine switches between L-Bus States 0 and 4 because Board Enable is off.

3) Writes an illegal port pointer to the Port Pointer Register. This prevents the L-Bus handshake by holding off the BUSGO signal on the TIC.

4) Issues the Direct command. This takes the L-Bus State Machine out of State 0.

5) The L-Bus State Machine then halts in State 5, waiting for a BUSEND. The BUSEND will not occur and a timeout will result, but the timeouts are disabled so that the error bits are not set. Checks that the POLL bit indicates that a direct command is being executed (POLL not active).

6) Writes to the Diagnostic Control Register (!A) and sets LOOPEN. This allows the L-Bus State Machine to continue and ROM Test 2 will be executed, writing the Direct command back into the SIB memory where it may be checked for accuracy.

Test 14. FIFO -- Tests the operation of the FIFO. It checks the following:

1) that an interrupt is not present after channel INIT;

2) that an interrupt is present after State 0 of the L-Bus State Machine;

3) that the interrupt goes away after Register 9 is read. A read of Register 9 will clear the current interrupts.

4) that the correct port data is passed through the FIFO; and

5) that an indication is received when the FIFO is full.

The FIFO stores the number of each interrupting port and is three deep.

Test 15. STATE COUNTER -- Checks that the DMA state counter increments each time until it reaches the halt command, at which point it resets to zero.

Test 16. DMA ADDRESS COUNTER -- Tests the DMA address counter by checking each bit carry as it counts.

Test 17. COMP/CTR LOOPBACK -- Tests the DMA logic's COMP register and address counter. It loads a pattern into memory, runs the DMA test that loads the DMA registers from memory, and then runs the test which dumps the registers back into memory.

Test 18. READ IMB -- Verifies proper operation during an IMB read. It uses the diagnostic DMA routine (DMA test 4) that reads a single word from the IMB and puts it in SIB RAM. The data is then checked.

Test 19. WRITE IMB -- Verifies proper operation during an IMB write. It uses the diagnostic DMA routine (DMA test 5) that writes a single word from SIB RAM into main memory.

Test 20. BEGIN FLAG/LEFT-RIGHT FLAG -- Tests the operation of the Begin Flag and Left/Right Flag Circuitry on the SIB. It uses the diagnostic DMA routine that increments the address counter a different number of times for each state of BF and LRF. It then stores the count in memory. The test is used to check that the flags are automatically updated.

3.3 ASYNCHRONOUS INTERFACE BOARD (AIB) TESTS

This test section tests the AIB. The AIB is the asynchronous interface section of the TIC board. The main program determines which ports to test and what tests to run on those ports. The loop count is passed to Test'AIB and indicates the number of times to run each test.

The AIB tests are divided into two groups: (1) the tests that run on one port at a time, and (2) the test that runs on all of the ports to be tested (real loopback). This test section first executes the first group of tests on each port. The tests are executed the number of times specified by the loop count. The MCC Dump, MCC Self-Test, MCC-PCC Communications test, Modem Signal test, and Modem Port Addressing tests execute loop count times if port 7 is specified as a port to be tested. Finally, the real loopback test is executed on all selected ports. After the testing is completed, a report is printed that indicates which ports passed and which ports failed.

Test 21. PCC SELF TEST -- Performs a self-test on one port and checks the results returned from that port's PCC. It checks:

1) that an interrupt is received;

2) that it is the correct interrupt;

3) that the interrupt came from the correct port;

4) that the correct number of bytes were transferred; and

5) that the results of self-test are correct.

Test 22. PCC DUMP -- Checks the operation of a dump on a PCC. It returns a logical value to indicate that the test passed. It performs the following:

1) The Read Data DMA pointers for the port that is being dumped are set up.

2) A Halt I/O is sent to the port so that the initiation of the dump will be controlled by a later Start I/O.

3) A Freeze Port and a Start I/O are performed to initiate the dump.

4) Checks are made to insure that an interrupt occurred, that the interrupt was the correct one, and that the interrupt came from the correct port.

Test 23. MCC SELF-TEST -- Initiates a self-test of the MCC, PCC, and MSC if port 7 is to be tested on the TIC. The self-tests are initiated by disabling the MCC and MSC and then issuing an MCC self-test control order to set the most significant bit. This routine waits for an interrupt, checks that it is the correct one (15), and checks that it came from the port that had been sent the control order. Finally, the pass/fail bit of each byte returned is checked for any problems during selftest.

Test 24. MCC DUMP -- Checks the operation of an MCC dump if port 7 is to be tested on the TIC. The dump is performed by first issuing a Freeze Modem Direct command and then starting a Dump MCC/MSC control program. The interrupt code is 19 (PCC or MCC/MSC dump completed). If the interrupt is correct, the beginning of the MCC and MSC dump areas are checked for MCC error codes that indicate problems while dumping MCC/MSC. The error code in MCC dump will be Non-Maskable Interrupt In Progress. The error code in MSC will be Link Error.

Test 25. PCC MCC COMMUNICATION -- Tests the communication between the PCC and the MCC if port 7 is to be tested on the TIC. It requests a dump of the MCC/MSC without having issued a modem disable. The PCC sends a single byte to the MCC that rejects the request. The expected interrupt is the same as that for a successful dump (19). The receive buffer contains the MCC's error code "invalid PCC message".

Test 26. DIAGNOSTIC LOOPBACK -- Executes the PCC's diagnostic loopback test. The diagnostic loopback control is used to test the hardware path from main memory to the PCC and back to main memory. A block of 256 characters are read by the PCC from main memory and then are written back into memory. The PCC checks that each byte is one greater than the previous byte and will interrupt if an error is detected. This routine verifies that the data that is written back into memory by the PCC is correct. It also checks that a valid interrupt is received from the correct port and that the correct number of bytes are transferred.

Test 27. BAUD RATE -- Tests the baud rate logic on the AIB section of the TIC. It tests the baud rates one port at a time by testing how long it takes to do a blind write of a given number of characters. The baud rate is calculated and compared to an acceptable range. If it falls outside the given range, an error is reported.

Test 28. MODEM SIGNALS -- Performs a loopback of the modem control signals for port 7 of the TIC. For each pattern to be tested, a control program is created and sent to the PCC. This control program consists of a Set Port Protocol control order followed by a Set Modem Controls control order followed by a Perform I/O control order. The pattern to be looped back is placed in the output field of the Set Modem Controls order. These outputs will be looped back to the inputs by a loopback connector. To guarantee that a modem change interrupt occurs immediately after the Perform I/O is executed, the reference mask is set to the complement of the outputs and the control mask is set to FF. This routine also checks that the interrupt actually occurred and that it came from the correct port.

Test 29. MSC PORT ADDRESSING -- Tests the addressing of port 7 by the MSC. In the first part of the test, a control program that contains the Set Modem control order is used to set the modem outputs for each port to that port's address. A loopback connector loops back to the outputs to the modem inputs. In the second part of the test, a Read Modem Inputs control program reads back the inputs from each port. If an addressing error has occurred, it is detected when the wrong address is read from one or more ports.

Test 30. LOOPBACK -- Executes a loopback through either a loopback hood or through the loopback circuitry on the TIC. This loopback circuitry is activated by placing a jumper on the test points near the backplane connector of the TIC PCA. If there is a failure, using the jumper enables you to determine if the TIC or the connector board is failing. The AIB main procedure initiates the test by starting a control program for each port to be tested. This control program contains a Perform I/O control order that specifies concurrent read/write without flush. These control programs will generate interrupts that are then processed by this routine. If no interrupts occur before a timer pops, errors for all ports with pending control programs are reported. If an interrupt does occur, the cause of the interrupt is checked. Next, the Receive Buffer is checked to make certain that it matches the Send Buffer. Because

the Perform I/O is done without flush, the Receive Buffer may begin with some data from the previous operation. The last step is to try to find individual ports that have not interrupted in a long period of time. This is done by maintaining a timer for each port. The timers for all ports are decremented each time any port interrupts. If the timer for a port reaches zero, an error is reported on that port.

ERROR INTERPRETATION

SECTION 4

4.0 INTRODUCTION

The TICDIAG traps many errors that would normally result in interrupts and system failures. In many cases, the TICDIAG creates the circumstances in which these errors occur.

Each failure detected by the TICDIAG results in an error message. The error message contains the following information:

1) the test that detected the error;

2) the symptoms of the error (e.g. incorrect register contents);

3) the functional block of circuitry that failed; and

4) an error number that indexes the possible failed component(s).

EXAMPLE:

```
Lynx bus logic not in state 000 after INIT.
Current State is:  001.
LBus State Machine/Reset Logic failure.
Error number 22.

Lynx Bus Sequencing test failed.
```

If an error occurs during execution of the TICDIAG and it is not an error trapped by the TICDIAG program, a DUS error message is displayed. This should only occur under unusual circumstances (e.g., some part of the system other than the TIC under test failed to function properly).

HP 3000 Series 37

PERIPHERAL INTERFACE CONTROLLER

Diagnostic Manual

HEWLETT PACKARD
8010 Foothills Blvd., Roseville, CA 95678

Part No. 30457-90005
E0984

Printed in U.S.A. 09/84

PRINTING HISTORY

New editions are complete revisions of the manual. Update packages, which are issued between editions, contain additional and replacement pages to be merged into the manual by the customer. The dates on the title page change only when a new edition or a new update is published. No information is incorporated into a reprinting unless it appears as a prior update; the edition does not change when an update is incorporated.

First Edition..... SEP 1984.....

LIST OF EFFECTIVE PAGES

The List of Effective Pages gives the date of the most recent version of each page in the manual. To verify that your manual contains the most current information, check the dates printed at the bottom of each page with those listed below. The date on the bottom of each page reflects the edition or subsequent update in which that page was printed.

Effective Pages	Date
all	SEP 1984

Table of Contents

GENERAL INFORMATION

SECTION 1

1.0 INTRODUCTION

This manual describes the diagnostic program for the Peripheral Interface Controller (PIC). The PIC Diagnostic (PICDIAG) is designed to verify correct operation of all functions of the PIC. A second PIC is required in order to fully test the HP-IB circuitry and the non-controller functions of the Advanced Bus Interface (ABI) chip.

The PICDIAG is written in SPL-II and runs under DUS.

1.1 REQUIRED HARDWARE

The hardware required to run the PICDIAG is the HP 3000 Series 37 minimum configuration:

Console (connected to the TIC at slot 1, port 0)
HP-IB Tape Drive
SPU (with the following board set): CPU
Memory
TIC (in slot 1)
PIC (as a coldload path)
a second PIC

1.2 REQUIRED SOFTWARE

The most recent revision of the following software is required:

DUS

1.3 DIAGNOSTIC PROGRAM STRUCTURE

The PICDIAG is composed of tests that can be run individually or in combination. The selected tests can be looped a specific number of times or until you halt them.

1.4 TEST LIMITATIONS

The PICDIAG requires a second PIC Printed Circuit Assembly (PCA) in order to fully test the HP-IB circuitry and the non-controller functions of the ABI chip. If you have only one PIC PCA, you can only run test steps 1 through 40. Test steps 41 through 45 require a second PIC PCA.

OPERATING INSTRUCTIONS

SECTION 2

2.0 INTRODUCTION

This section of the manual describes how to operate the PICDIAG.

The primary function of the PICDIAG is to test and diagnose the HP 3000 Series 37 Peripheral Interface Card. The PICDIAG will isolate faults to the component level wherever possible.

You control test selection, looping, error handling, and printing messages.

2.1 TEST SELECTION

The PICDIAG is divided into tests that can be run individually or in various combinations.

Use the STATE command to list which tests and commands are currently selected. When the PICDIAG is in the default configuration, the STATE command will display the following message:

```
PICDIAG STATUS:

PATH:      X      X
CHANNEL:   X      X
EEPR:      ON
EEPS:      ON
ENPR:      ON
LOOP:      OFF
PRINT:     OFF
TRACE:     OFF
Sections Selected:  1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,
   23,25,28,29,30,32,33,34,35,36,37,39,41,42,43,44,45
```

To change the test selection, type TEST, then use a '-' in front of the test number(s) to be deleted, a '+' in front of the test number(s) to be added, or simply the test number(s) to be executed.

EXAMPLE:

TEST - 3,6 deletes test sections 3 and 6
TEST + 1,5,6 adds test sections 1, 5, and 6
TEST 1/5,8 executes test sections 1-5, and 8

Test 41 through 45 require a second PIC.

2.2 LOOPING

You can loop on selected tests by entering the LOOP command with the number of times that you want the tests to loop.

2.3 ERROR HANDLING

The PICDIAG generates three types of error messages: user input error, test failure, and system failure.

1. User Input Error Messages - Displayed in response to an invalid user input.

   EXAMPLE: When PICDIAG requests the channel number of the PIC to be tested. the expected response is a decimal number between 1 and 15. If your input is different, the following message is displayed:

   ```
   1<=CHANNEL<=15
   ```

   The request for a channel number is repeated until you enter a valid response.

2. Test Failure Error Messages - Displayed when the PICDIAG detects a malfunction of the PIC PCA. The purpose of this type of error message is (a) to let you know that the PIC under test is NOT functioning correctly, and (b) to identify the failure.

   The test failure error message has three parts. The first part identifies the step number that detects the error. The second part describes the discrepancy between the actual and expected response to the test. The third part identifies the probable cause of the error.

   EXAMPLE:

   ```
   Error in Step 38:
   No time-out from CSRQDIS.
   Check U403, U505.
   ```

3. System Error Messages - Displayed when an error is probably caused by something other than the PIC under test.

   EXAMPLE:

   ```
   System in Step 41:
   ????????????????????????????????????
   ```

2.4 PRINTING MESSAGES

There are two types of messages: error and information.

The three types of error messages are described above. The default configuration enables error messages (EEPR). To suppress error messages, type SEPR.

Information messages describe the progress of the diagnostic or tell you to perform an operation. The default configuration enables information messages (ENPR). To suppress information messages, type SNPR.

The messages are displayed on the console by default. To direct the error messages to a hardcopy device, type PRINTER. Non-error messages cannot be directed to the printer.

2.5 HOW TO RUN PICDIAG

Input to the PICDIAG is through the system console.

Run PICDIAG in the following manner:

1) Perform MPE shutdown to log off all users, if applicable.

2) Run the console selftest.

3) Fully reset the console.

4) Install a DUS tape in the coldload device.

5) If the system is off, power it on by turning the keyswitch to the LOCAL or REMOTE position. If the system is already on, verify that the keyswitch is in the LOCAL or REMOTE position. Use the appropriate load command to load the DUS.

6) DUS displays a welcome message and a prompt:

```
DIAGNOSTIC/UTILITY SYSTEM (REVISION nn.nn)
ENTER YOUR PROGRAM NAME (TYPE HELP FOR PROGRAM INFORMATION)
:
```

7) Type PICDIAG in response to the prompt.

8) PICDIAG loads, then displays the following message:

```
PICDIAG xx.xx

Default tests are 1-39,41-45.  Optional test is 40.
```

```
Tests 40/45 require manual intervention.  Tests 41/45
require a second PIC and HP-IB cable.


Enter 'GO' to continue.
>
```

The greater-than symbol (>) is the PICDIAG prompt.

9) At this point, you can select the diagnostic options by entering any of the commands described in Table 2-1.

The following commands control the execution of PICDIAG:

GO - Continues diagnostic execution from a pause.

EXIT - Stops diagnostic execution and returns control to DUS.

RUN - Restarts execution of the diagnostic at the beginning.

CNTRL Y - Interrupts execution of DUS. During the execution of a test, this command brings the PICDIAG back to the prompt. Entering GO will resume the program from the point where you interrupted it.

10) When you enter GO, the following prompt will be displayed:

```
The PATH number (0-2) of the PIC to be tested is?:
```

Enter the correct path number. You then receive the following prompt:

```
What is the PIC address (1-15)?:
```

Enter the correct PIC address, you will receive the following prompt:

```
The second PIC is on what path number?
What is the second PIC's address?
```

Enter 0 if there is no second PIC.

NOTE

Steps 41 through 45 require a second PIC.

2.6 HOW TO CONFIGURE PICDIAG

You can configure and reconfigure the diagnostic according to your needs.

Table 2-1 describes the available PICDIAG commands. Those commands indicated with an asterisk (*) are part of the default configuration. The PICDIAG provides this information when you enter the HELP command.

Table 2-1. DIAGNOSTIC COMMANDS

COMMAND	PARAMETERS	DESCRIPTION
EEPR*		Enable Error Messages
EEPS*		Enable Error Pause
ENPR*		Enable non-error messages
EXIT		Return to DUS
GO		Resume diagnostic execution
LOOP		Loop on selected steps
NOLOOP		Stop Loop
NOTRACE		Suppress I/O trace
PRINTER		Display errors on printer
RST		Enable pauses and messages also suppress printer
RUN		Restart diagnostic execution
SEPS		Suppress error pauses
SNPR		Suppress non-error messages
STATE		List status of program
TEST	step(s)	Select test steps to run
TRACE		Enable I/O activity trace

* default

TEST DESCRIPTIONS

SECTION 3

3.0 INTRODUCTION

This section of the manual describes the test sections of the PICDIAG.

3.1 REGISTER TEST

The register test section is composed of four steps that verify the basic PIC functions:

Step 1. Roll Call (ROCL) Test - Verifies that the PIC responds correctly to the ROCL SIMB command. This step tests part of the global command decoder and part of the poll response circuitry. ROCL is verified at all available channel addresses in the Channel Address Test section.

Step 2. Channel Address Test verifies:

a. That the channel address can be read from registers 13 and 15. Tests the addressed command decoder, the register select, the data drivers, and the slave handshake.

b. The channel address read with the OBSI SIMB command. Tests an additional portion of the command decoder.

c. The channel address read with the OBII SIMB command.

Step 3. Register Initialization Test - Verifies that PIC registers 8, 11, 12, 13, 14, and 15 are correctly read after issuing an INIT channel SIMB command. Only the non-ABI registers are tested at this point because the data and control paths to them are simpler than those to the ABI. The ABI registers are tested by the ABI verification.

Step 4. Storage Register Test - The PICDIAG performs a memory test on registers 9, 10, and part of 8 by using random data patterns to test the greatest range of data patterns in a reasonable amount of time. The pattern halts when an error is encountered so that the error can be repeated. A number of different error messages are displayed that attempt to identify the error to the smallest set of components.

The bit error log shows a map of errors by register and bit.

1. Bit line errors: Any data bits that are in error for all registers tested will cause the following error message to be displayed:

```
    Error in Step 4:
    Bit line error.

bit 0    4    8    12
    v    v    v    v
R8  .... .... 0100 0000
R9  0000 0000 0100 0000
R10 0000 0000 0100 0000
```

2. Four-bit register errors: If data errors are limited to a four-bit section of a register, then the following error message is displayed:

```
Error in Step 4:
Latch/buffer error.

bit 0    4    8    12
    v    v    v    v
R8  .... .... 0000 0000
R9  0000 1110 0000 0000
R10 0000 0000 0000 0000
```

3. Register select line errors: When one register fails the register test, the following error message is displayed:

```
Error in Step 4:
Select line errors.

bit 0    4    8    12
    v    v    v    v
R8  .... .... 0000 0000
R9  1111 1111 1111 1111
R10 0000 0000 0000 0000
```

4. Handshake or write enable errors: If all registers fail without a discernible pattern, the following error message is displayed:

```
Error in Step 4:
Check slave handshake logic.

bit 0    4    8    12
    v    v    v    v
R8  .... .... 1101 1111
R9  1111 1011 1110 1101
R10 1101 1111 1111 0111
```

3.2 IRQ TEST

This test section verifies the correct operation of the PIC Interrupt Request logic.

Step 5. Interrupt Initialization Test - Verifies that the interrupt circuitry is initialized correctly.

Step 6. Interrupt Pending Register Test - Verifies that all eight INTPEN bits can be set and cleared.

Step 7. Interrupt Priority Encoder Test - Verifies that INTDEV is the highest priority bit set in the INTPEN register.

Step 8. Interrupt Mask Bit Test - Verifies that MASKF can be set and cleared. MASKF is also tested at all channel addresses in the Channel Address Test Section.

Step 9. Interrupt Poll Test - Verifies that the SIMB IPOLL command functions. IPOLL uses circuitry shared with ROCL, SPOLL, IPOLL, and RMSKL. The shared circuitry is tested at all channel addresses in the Channel Address Test section.

Step 10. SIMB Interrupt Request Test - Verifies that the SIMB IRQ line functions.

3.3 CONFIGURATION TEST

This section of the PICDIAG tests the correct operation of the PIC's Configuration Register bits.

Step 11. Configuration Bits Zero Test - Tests all the configuration bits in their zero state.

Step 12. Configuration Bits One Test - Tests all the configuration bits in their one state.

Step 13. Channel Address Test - The PICDIAG tests all possible channel addresses, except those that are occupied by other channels on the SIMB. At each selected channel address, the channel address decoder, registers 13 and 15, and the global poll response are tested.

3.4 ABI CHIP VERIFICATION

This test section verifies the correct operation of the Advanced Bus Interface (ABI) chip. The ABI supports all the functions of the PHI (LSI chip) plus Cyclic Redundancy Check (CRC) generation, command parity selection, and other features. The PICDIAG determines that the ABI and not the PHI is present, reports the result to the user, and then proceeds with the tests if the ABI is present.

Step 14. PHI/ABI Type Determination - This test determines and indicates whether a PHI or ABI is installed on the PIC being tested. It also indicates the revision level of the device. If the test indicates than an ABI is not installed on the PIC, an error message is generated and the diagnostic is terminated.

Step 15. ABI Register Initialization Test - Verifies that the ABI registers are correctly initialized following an SIMB INIT channel command.

Step 16. Data Paths and Register Addressability Test - Performs a memory test of ABI storage registers by using a random data pattern. The error messages separate errors into three categories:

- PIC or ABI data line failure
- PIC or ABI register addressing failure
- ABI storage register failure

Step 17. ABI HP-IB Functions Test - Tests all possible ABI and HP-IB state changes:

- Eight-bit Processor Feature Test
- System Controller Functions Test
- Status Change Detection Test
- Outbound FIFO Status Test
- Addresses Status Indicators Test
- Device Clear Test
- Remote and Local Test
- Listen and Talk Addressability Test
- Parallel Poll Response Test
- Parallel Poll Configure and Unconfigure Test
- Serial Poll Test
- Counted Transfers Test

- Inbound FIFO Test
- Passing Control Test
- Parity Error Test
- CRC Generation and Functions Test
- ABI Interrupt Functions Test
- DMA Request Functions Test

ABI Test Limitations

Due to hardware limitations, the following functions of the ABI are NOT tested in this test section.

- ABI to HP-IB data paths - tested in the HP-IB Interface Drivers test section.
- Command parity error detection - tested on the ABI chip that has the programmable command parity feature.
- Command parity error freeze - tested in the HP-IB Interface Drivers test section.
- Outbound data freeze - cannot be tested offline because it is only set when data enters the inbound FIFO from the HP-IB, not from the outbound FIFO. This test is performed in the HP-IB Interface Drivers test section.
- ABI Interrupt - tested in the Service Request (CSRQ) test section.
- ABI DMA request - tested in the DMA State Machine test section.

3.5 DMA STATE MACHINE TEST

This test section verifies the correct operation of the Direct Memory Address (DMA) state machine. Parts of it use the diagnostic DMA clocking feature by utilizing the PIC's diagnostic hardware. This test section verifies:

- The clocking and initialization of the DMA state machine.
- All transitions of the DMA state machine, including SIMB transactions. There are two main state loops in the DMA state transition diagram: one for input DMA transfers, the other for output DMA transfers. Each of these loops has two paths entering it from the initial state. Each also has two normal termination exits and two abort termination exits. The tests are performed in the following sequence: Can the initial state be asynchronously accessed? Can the four entry points to the transfer loops be accessed? Do the loops function? Do the loops exit properly under normal termination conditions? Does the channel service request sequence function properly?
- The outputs of each state.

- Correct error and abort status.
- DMA termination status.
- Byte packing and unpacking.

Step 18 DMA Initialization - Starts the DMA state machine and verifies that INIT resets the DMA state machine to state 0. Also checks that the DMA state machine clocks out of state 0 after writing to register 11.

Step 19. CSRQ and OBSI Test - Verifies that at state %12 CHANRQ is asserted, that an OBSI causes transition to state 0, and that DMA machine stops in state 0.

Step 20. DMA Output State Transitions verifies-

- That the state machine can be started and that the output data transfer loops are accessible. If either of the data transfer loops are not accessible, the rest of the DMA state machine test aborts.
- The output transfer loop (%20-%21-%23-%26-%27-%25) starting at both state %20 and state %22.
- The end transitions at states %23 and %25.
- That a data transfer can be aborted at states %21 and %27.

Step 21. DMA State Outputs Test - Verifies the outputs or the effects of the outputs for each state. For the address and count register tests, a random number sequence is used to test different combinations of starting address and count, then the DMA is run to completion and the final count and address are verified.

Step 22. DMA Input State Transitions verifies-

- That the state machine can be started and that the input data transfer loops are accessible. If either of the data transfer loops are not accessible, the rest of the DMA state machine test aborts.
- The input transfer loop (%1-%3-%14-%15-%17) starting at both state %1 and state %4.
- The end transitions at states %3 and %17.
- That a data transfer can be aborted at states %1 and %14.

Step 23. DMCNT, DMADR Tests verifies-

- That the DMA byte counter logic functions correctly.
- That the DMA address register functions correctly.

Step 25. DMA Error and Abort Test - Verifies the address overflow abort.

Step 28. DMA Termination Status Test -

- 00 - count and end
- 11 - error
- 10 - count no end
- 01 - count subrecord

Step 29. DMA OBYTE Data Paths Test - Verifies that the LEFTOUT and RIGHTOUT registers correctly unpack words into bytes.

Step 30. DMA IBYTE Data Paths Test - Verifies that the LEFTIN and RIGHTIN registers correctly pack data bytes into words.

Step 32. DMA Timeout Test - Verifies that the DMA timeout abort functions correctly.

3.6 CSRQ TEST

This test section verifies the correct operation of the channel request logic.

Step 33. Device Request Test - Verifies that DEVRQ is correctly asserted from each of the following inputs:

- Parallel Poll
- New Status
- CSRQDIS
- DMINACT
- CIC
- RIOC
- OBSI

Step 34. Channel Request Test - Verifies that CHANRQ is correctly asserted from each of the following inputs:

- DMINT
- PHIINT
- Parallel Poll
- DMIN
- DMOFF

- OBSI

Step 35. Local Channel Service Request Test - Verifies the correct operation of MYCSRQ in response to:

- CHANRQ
- DEVRQ

Step 36. SIMB Channel Service Request Test - Verifies the correct operation of CSRQ in response to:

- CSRQ
- DIAG
- MYCSRQ

Step 37. Service Poll Response Test - Verifies the correct response for SPOLL.

Step 38. Channel Service Request Timeout Test - Verifies the correct operation of TORESET during channel service request.

Step 39. Obtain Service Information Test - Verifies that the device number supplied in response to the SIMB OBSI command is correctly generated from each of the following inputs:

- DMDEV
- Parallel Poll Response
- NEW STATUS
- RIOC and OBSI

3.7 HP-IB INTERFACE DRIVERS TEST

This test section verifies the correct operation of the ABI to HP-IB logic which consists of four HP-IB transceivers and two inverters. It is assumed at this point that the PIC and the ABI have been completely tested offline.

A second PIC is required to perform this test section. The second PIC is assumed to be functioning correctly, although some diagnostics are performed on it to avoid catastrophic failures. The two PICs are connected together via HP-IB cables with no other HP-IB devices connected. Since some of the HP-IB lines can only be asserted by the system controller, it is necessary for the PICDIAG to alternate the System Controller function between the two PICs. This test is broken down into three groups so that failures can be easily located.

The first group tests the HP-IB lines that function independently of the handshake. The IFC and SRQ lines fall into this category. The ATN, EOI, and DIO lines also fall into this category in the limited case of executing a parallel poll. Executing a parallel poll verifies that ATN and EOI can both be asserted low, that one data line at a time can be asserted low, and that at least one of the ATN or EOI lines can be asserted high to stop the parallel poll.

The second group tests the HP-IB handshake lines and the other HP-IB lines that function in conjunction with the handshake. This group first verifies that the controller receives NRFD and NDAC high when no other HP-IB devices are online, that device receives controller's handshake, and that device receives REN. Then it verifies the combinations of ATN and EOI that were not conclusively verified with the parallel poll test.

At this point, all of the HP-IB interface lines have been tested. The third group verifies that all of these lines can function harmoniously together at full operating speed. Basically, blocks of random data are passed in each direction between the two PICs, and the other control lines are exercised frequently.

Step 41. Bus Control Lines Test - This test verifies the correct operation of IFC, SRQ, and parallel poll, which all function independently of the HP-IB handshake.

- IFC test
- SRQ test

Step 42. Data Lines Test -

- Data line verification with parallel poll response. Tests passive-low assertion of data lines, ATN and EOI.
- ATN or EOI high assertion test. Tests whether ATN, EOI, or both can be asserted high.

Step 43. Handshake Lines Test - Tests the HP-IB handshake lines DAV, NRFD, and NDAC. These lines can be individually tested to a limited extent. Also tests REN, ATN and EOI.

- NRFD and NDAC passive high test - Verifies that setting the device offline causes NRFD and NDAC to float high, completing any controller handshake.
- NDAC test - Verifies that when the device is online, NDAC is low and NRFD is bringing NDAC high.
- NRFD test - Verifies that when the device in FIFO is full, NRFD is high so that the byte written into controller out FIFO will not go out on the HP-IB.
- REN test - Verifies that when the controller asserts REN, the device goes to REM at next acceptor handshake, and that REM goes false when REN goes false or if IFC goes true.

Step 44. Data Modifiers Test - This tests the two data modifier lines ATN and EOI.

- Tests not ATN, not EOI, and handshake.
- Tests ATN, not EOI, and handshake.
- Tests not ATN, EOI, and handshake.

3.8 ONLINE DATA TEST

This test section verifies the correct operation of the ABI to HP-IB logic (which consists of four HP-IB transceivers and two inverters). It is assumed at this point that the PIC and the ABI have been completely tested offline, so that all that remains to test is that the PIC can correctly communicate with other HP-IB devices over the standard HP-IB.

A second PIC is required to perform this test section. The second PIC is assumed to be functioning correctly, although some diagnostics are performed on it to avoid catastrophic failures. The two PICs are connected together via HP-IB cables with no other HP-IB devices connected. Since some of the HP-IB lines can only be asserted by the system controller, it is necessary for the PICDIAG to alternate the System Controller function between the two PICs. This test is broken down into three groups so that failures can be easily pinpointed.

Step 45. HP-IB Data and Control Test - At this point the PIC, the ABI, and all of the HP-IB drivers have been tested. This test verifies that the HP-IB drivers work at high speed and tests the pattern sensitivity of HP-IB interface lines. Additionally, it performs a general data and control test similar to the DMA exerciser program.