

HP 3000 Computer System

General I/O (Channel GIC) Diagnostic Manual



COMPUTER SYSTEMS DIVISION, 19447 PRUNERIDGE AVE., CUPERTINO, CALIF. 95014

Part No. 30341-90009
Index No. 3HDWR.070.30341-90009

Printed in U.S.A 5/81

NOTICE

The information contained in this document is subject to change without notice.

HEWLETT-PACKARD MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance or use of this material.

This document contains proprietary information which is protected by copyright. All rights are reserved. No part of this document may be photocopied or reproduced without the prior written consent of Hewlett-Packard Company.

LIST OF EFFECTIVE PAGES

The List of Effective Pages gives the date of the current edition and of any pages changed in updates to that edition. Within the manual, any page changed since the last edition is indicated by printing the date the changes were made on the bottom of the page. Changes are marked with a vertical bar in the margin. If an update is incorporated when an edition is reprinted, these bars are removed but the dates remain.

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 date of the title page of the manual changes only when a new edition is published. When an edition is reprinted, all the prior updates to the edition are incorporated.

CONTENTS

SECTION I - GENERAL INFORMATION

Paragraph	Page
INTRODUCTION	836-1
REQUIRED HARDWARE	836-1
REQUIRED SOFTWARE	836-4
DIAGNOSTIC PROGRAM STRUCTURE	836-4
TESTING PHILOSOPHY	836-4
TEST LIMITATIONS	836-5

SECTION II - OPERATING INSTRUCTIONS

Paragraph	Page
INTRODUCTION	836-7
LOADING THE SYSTEM.....	836-7
STANDARD MODE	836-7
EXTENDED MODE	836-8
GIC DIAGNOSTIC COMMANDS	836-8

SECTION III - EXECUTION TIMES

Paragraph	Page
INTRODUCTION	836-9
NOTES	836-10

SECTION IV - TEST DESCRIPTIONS

Paragraph	Page
INTRODUCTION	836-11
SECTION 1 - SWITCHES AND INTERRUPTS	836-11
SECTION 2 - PHI VERIFICATION	836-14
SECTION 3 - PARTIAL CSRQ TEST	836-14
SECTION 4 - PARTIAL CSRQ TEST	836-15
SECTION 5 - PARALLEL POLL PRIORITY	836-15
SECTION 6 - PARTIAL CSRQ TEST	836-16
SECTION 7 - REGISTERS 8, 9, AND A	836-16
SECTION 8 - DMA STATE MACHINE	836-17
SECTION 9 - RIGHT OUTPUT DMA TRANSFER	836-20
SECTION 10 - RIGHT OUTPUT DMA TRANSFER	836-22
SECTION 11 - LEFT INPUT DMA TRANSFER	836-22
SECTION 12 - DMA INPUT TRANSFERS	836-24
SECTION 13 - RIGHT TO LEFT BYTE DMA PATH	836-25

CONTENTS (Continued)

SECTION 14 - WAIT STATES	836-25
SECTION 15 - DMA WAIT STATES	836-26
SECTION 16 - ADDRESS ROLLOVER	836-26
SECTION 17 - IMB MEMORY TIMEOUT	836-26
SECTION 18 - PARITY ERROR ABORT	836-26
SECTION 19 - DNV ASSERTION	836-27
SECTION 20 - DNV ASSERTION	836-27
SECTION 21 - NEW STATUS WHEN NOT CACS	836-27
SECTION 22 - DNV ASSERTION	836-27
SECTION 23 - DMA LOOPBACK TESTS	836-28
SECTION 24 - GIC TIMEOUT LOGIC CHECK	836-28
SECTION 25 - GIC-TO-GIC TRANSFERS	836-29

SECTION V - ERROR INTERPRETATION

SECTION VI - GLOSSARY OF TERMS

ILLUSTRATIONS

Title	Page
Required Hardware Summary	836-2
Hardware Configuration For Modes 2a And 2b	836-3
Table Of Sections	836-9
Function/Block Reference Table	836-32

GENERAL INFORMATION	SECTION I
---------------------	--------------

1.0 INTRODUCTION

This document describes the diagnostic test program for the General I/O Channel (hereinafter referred to as the GIC). This program is designed to verify correct operation of all functions of the GIC provided that all sections, both standard and extended, are executed. This is accomplished by exhaustively activating and checking each logical function of the unit under test. This program is written in the SPL-II language.

1.1 REQUIRED HARDWARE

Special logic functions have been included on the GIC to permit the state machines to be exercised and meaningful failure data to be obtained. Activation of these functions via Mode Switch S5, however, makes the unit under test unsuitable as a device to communicate with other I/O devices, such as the flexible disc or magnetic tape unit. Several methods of operation, each requiring different equipment, are possible.

In configuring the hardware for this diagnostic, there are four considerations:

- (1) Hardware. An HP3000 HP-IB version computer system with 128 kbytes of memory.
- (2) Loading. Either the GIC under test must work well enough to load DUSIII or an additional GIC is required.
- (3) Isolation from HP-IB. The GIC under test must be disconnected from the HP-IB except in Section 25.
- (4) Section 25. If the GIC-to-GIC test is to be run, a second GIC is required to communicate with the GIC under test via the HP-IB.

This is summarized in Figures 1.1 and 1.2

GIC Diagnostic

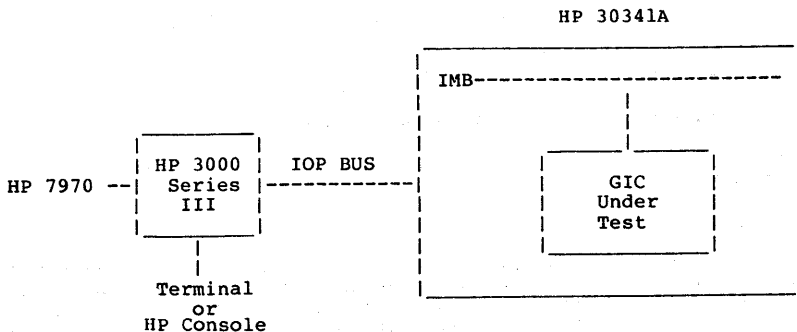
Mode	GIC Under Test	Add'l Good GIC	Comment	Test Sect 25*
2a	X		Only if GIC under test can load	
2b	X	X	Load through good GIC	X

*There is enough hardware to run optional Test Section 25.

Figure 1.1 Required Hardware Summary

GIC Diagnostic

Mode 2a - Loading and Running Test Sections 1 through 24



Mode 2b - Running Test Section 25

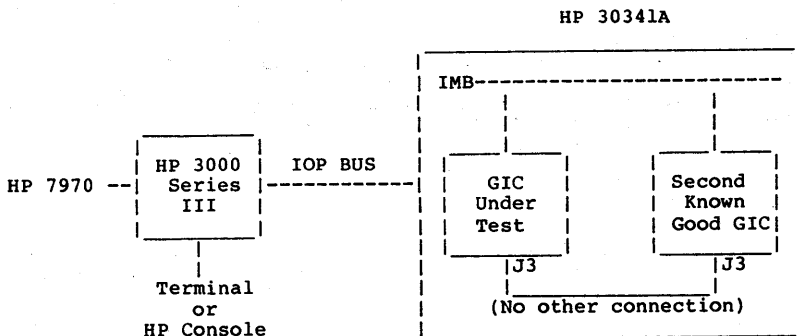


Figure 1.2 Hardware Configuration for Modes 2a and 2b

GIC Diagnostic

1.2 REQUIRED SOFTWARE

DUSIII file system plus the GIC diagnostic program.

1.3 DIAGNOSTIC PROGRAM STRUCTURE

The heirarchy for this diagnostic program is:

Program
 Section
 Step
 Case

where a step is the smallest entity (not necessarily loopable) which contains at least one case and verifies operation of one conceptually separate function. Steps include one or more cases when a single stimulus sets up multiple conditions to be verified.

A case is defined as the entity which verifies proper circuit conditions along a distinct path in the logic of the unit under test. Associated with each case is a subsection of the hardware which is required to be functional for that case to succeed. Cases occur in the program flow wherever program statements have established proper conditions in the unit under test to verify that a function has or has not operated properly. As some cases are used in several steps, case numbers do not appear in a strictly ascending order from the beginning to the end of the diagnostic program. Step numbers, however, are in strictly ascending order, with each step assigned a unique number.

1.4 TESTING PHILOSOPHY

The GIC circuitry is not divisible into simple blocks for diagnosis; many logical paths contribute to proper operation of several board functions.

This diagnostic program is substantially different from other programs of this nature previously produced for Hewlett-Packard data products. This unique design is an attempt to achieve the goal of providing the diagnostic program user with the intimate circuit knowledge of the design engineer. The program and documentation is optimized to provide the user with a maximum amount of useable data about the tests being performed, the circuitry involved in these tests, the circuit elements which are suspected of failure, as well as symptom data about each failure.

This program is used with the "board-swap" method of trouble-shooting. By running the diagnostic, a GIC board is tested to determine if any error occurs. If one does, the board is replaced.

The goals of this diagnostic are achieved by first testing a minimum subset of the logic which must be functional before any data can be obtained from the unit under test. When this section of logic has been verified, another small section is added to this original kernel and tested with it, assuming that the first section tested operates properly. Additional logic functions are included in the cases, each piece being verified before it is used to test adjacent functions. As errors are detected, they are reported in two ways:

- (1) Error messages are generated at the time the errors are detected. These may contain expected/obtained values, as well as reporting other symptoms of the malfunction.
- (2) Each case which fails or passes is recorded. After the diagnostic has run to completion, the numbers of the cases which have failed are supplied by the program to the user in the Fault Data Summary. One may then correlate this data through the use of the GIC Diagnostic IMS to determine which hardware module is most likely to have failed, which modules are suspected of failure, and which modules have not been involved in cases which failed. One may also use this information with the IMS to trace on a schematic, in a follow-the-dots fashion, the exact path traversed through the logic in the cases that failed.

1.5 TEST LIMITATIONS

Running the standard set of Sections rather than the complete set results in some limitations.

- a. Section 25 tests the GIC by communicating on the HP-IB back and forth to a known-good GIC, thus adding tests for:

- (1) The pin-drivers on the PHI chip
- (2) The HP-IB Transceiver chips
- (3) Master-slave handshake contention on the IMB
- (4) Running the DMA machine at higher speed than the remainder of the diagnostic.

Conditions 3 and 4 cannot be checked by the 9570 test for the GIC.

- b. The SWITCHES command adds steps to Section 1 which require the operator to set all switches to all positions, thus adding tests for:

- (1) channel response at CHAN ADDRs not otherwise used (console etc.)
- (2) DEVICE TYPE switch in both positions

GIC Diagnostic

(3) PROCESSOR switch in both positions

The limitations of the complete program come under these general headings: limitations due to the test cases used and limitations resulting from the inability to test certain logic.

a. Limitations due to inability to test certain logic:

- (1) Bus Request Logic is not checked.
- (2) Race conditions involving asynchronous circuitry cannot be checked.
- (3) Output signals from certain DMA states cannot be verified.
- (4) Special PHI handshakes included to provide for early asynchronous PHI handshake assertion and completion cannot be checked.
- (5) Logic to prevent unassertion of memory read requests before completion of the master handshake cannot be verified.
- (6) HSEN on the HP-IB transceivers is not checked.
- (7) Data rate on the HP-IB is never checked.

Conditions 2, 3, 4, 5, 6, and 7 cannot be checked by the 9570 test for the GIC.

b. Limitations due to test cases

Because the test cases were generated using the idea that each IC output should be "active" at least once (see the GIC Diagnostic IMS for definition of active), no rigorous verification was ever made that all outputs had been verified in each state which they can assume. No rigorous verification was made to determine that adjacent IC pins or bus lines on some internal data paths were not shorted together. The test cases do, however, provide a very thorough and strenuous exercise of the PCA through as many of its possible states as is reasonably practical.

2.0 INTRODUCTION

Before running this diagnostic be sure that the physical configuration (switches on the frontplane) matches the logical configuration required by the operating system.

There are two modes of operation possible for this diagnostic program, the standard (default) mode and the optional mode.

To operate in either mode, the following steps must be executed:

- (1) Set the GIC board Mode Switch S5 to the TEST position and bring up DUSIII.
- (2) When the DUSIII prompt character (:) is displayed, respond 'GICDIAG' to load the diagnostic.
- (3) The GIC Diagnostic Program displays its title message and prompt character '>'.

2.1 LOADING THE SYSTEM

- (1) Perform an MPE 'SHUTDOWN' to properly logoff every current session, if applicable.
- (2) Run the terminal Self-Test and verify the displayed results.
- (3) Fully reset the terminal.
- (4) Ensure that the terminal is in REMOTE.
- (5) Install a GIC Tape on the Cold-Load Tape Unit.
- (6) If the cold-load device is an HP 7970B/E, set the Control Panel Switch Register to 83006. If the cold-load device is an HP 7976, set the Switch Register to:

```
0 CCCC DDD 01 111 101
```

where: CCCC = channel number (binary) to which the device is attached and DDD = device number (binary) of the device.

- (7) Press HALT; then press ENABLE and LOAD.
- (8) Set Switch Register to 1.
- (9) Press RUN.
- (10) GIC will be loaded from tape.
- (11) Press the RETURN key on the terminal to speed-sense the terminal.
- (12) The welcome message and prompt are displayed:

General I/O Channel revision XX.XX

Enter your program name (Type HELP for program information)
(The revision is determined by the latest release date of the FMGR program.)

(HELP is an AID program that presents file and command information.)

2.2 STANDARD MODE

- (1) If the standard mode is to be executed, the operator responds 'GO' and diagnostic execution begins.

The standard mode is defined as follows:

- (a) Execute all Diagnostic Sections except Sections 24 and 25.
- (b) Display error, information, and prompt messages.
- (c) Pause on errors and prompts.

2.3 EXTENDED MODE

- (1) If the optional mode is to be executed, the operator may input one or more of the commands discussed in paragraph 2.3.

After all desired options have been entered, the operator enters 'GO', and the diagnostic will begin execution.

The diagnostic will run until an error condition is detected or all selected sections have been executed.

2.4 GIC DIAGNOSTIC COMMANDS

These commands are used when the GIC diagnostic is not in the run mode (i.e., it is in a pause or wait state) and they are usually executed before the GO command.

OUTPUT AND PAUSES

PRINTER- print error messages and the Fault Data Summary.

- *EEPR - enable error messages.
- SEPR - suppress error messages.
- *ENPR - enable non-error messages.
- SNPR - suppress non-error messages.
- *EEPS - enable pauses after error messages.
- SEPS - suppress pauses after error messages.
- RST - reset message and pause commands to default state;
supercede PRINTER command.

*Default Value

TEST SELECTION

SWITCHES- execute switch test portions of Section 1

NOSWITCHES- supercedes SWITCHES command

- TEST - change from the default set of section execution:
- 'TEST 1,5,8' -- execute sections 1,5 and 8.
 - 'TEST 1/3,8' -- execute sections 1,2,3, and 8.
 - 'TEST +3,6' -- add sections 3 and 6.
 - 'TEST -3,6' -- remove sections 3 and 6.

PROGRAM CONTROL

- GO - continue diagnostic execution from a pause.
- EXIT - stop diagnostic execution and return to DUSIII.
- RUN - restart execution of diagnostic at the beginning.
(supercedes SWITCHES and PRINTER commands;
does not affect LOOP and TEST commands)
- LOOP - loop on the selected sections.
- LOOPOFF- supercede the LOOP command.

EXECUTION TIMES	SECTION III
-----------------	----------------

3.0 INTRODUCTION

The following information provides timing information for all test sections and information regarding the execution status of each test section.

Std.	Sect	Steps	Description	Time	Notes
*	1	1-19	Switches and Interrupts		4
*	2	20-21	PHI Verification	3 sec	
*	3	22-24	Partial CSRQ Test	ms	
*	4	25-27	Partial CSRQ Test	ms	
*	5	28-29	Parallel Poll Priority Encoder	ms	
*	6	30-34	Partial CSRQ Test	ms	
*	7	35-37	Registers 8, 9, and 10	ms	
*	8	38-57	DMA State Machine	ms	1
*	9	58-62	Right Output DMA Transfer	ms	1
*	10	63-65	Right Output DMA Transfer	ms	1
*	11	66-77	Left Input DMA Transfer	ms	1
*	12	78-82	DMA Input Transfers	ms	1
*	13	83-85	Right to Left Byte DMA Path	ms	1
*	14	86-87	DMA Wait States	ms	1
*	15	88-89	DMA Wait States	ms	1
*	16	90-91	Address Rollover	ms	1
*	17	92	IMB Memory Timeout	ms	1
*	18	93-94	Parity Error Abort	ms	1
*	19	95	DNV Assertion	ms	1
*	20	96	DNV Assertion	ms	1
*	21	97	New Status when not CACS	ms	1
*	22	98	DNV Assertion	ms	1
*	23	99-100	DMA Loopback Tests	ms	1
	24	101-104	Timeout Logic	4 sec	2
	25	105	GIC-to-GIC Transfers	40 sec	2,3

Figure 3.1 Table of Sections

GIC Diagnostic

3.1 NOTES

total running time per pass:
standard set = 3 seconds
complete set = 5 minutes including operator cabling

* Part of standard set of Sections

- (1) DMA machine is single-stepped by the slave handshake. Refer to Section VI for details.
- (2) Channel program microcode utilized.
- (3) Requires both the GIC board under test and a known good GIC board dedicated to use as a communication target for the GIC-to-GIC transfers. Refer to paragraph 1.1 for details.
- (4) If "SWITCHES" option is selected, operator intervention is required to test switch positions.

TEST DESCRIPTIONS	SECTION IV
-------------------	---------------

4.0 INTRODUCTION

This section of the manual will provide you with a fair description of the test sections and the steps within each step along with possible error conditions, error messages, and prompt messages.

4.1 SECTION 1 - SWITCHES AND INTERRUPTS

The portions of this section which require all switches to be set to all positions are optional. They are run only when the operator desires to verify the ability of all the switches mounted on the GIC to correctly respond in all positions. These portions also test the circuitry associated with these switches. This option is selected by the "SWITCHES" option at runtime. Data for the Fault Data Summary is collected during this section only when the "SWITCHES" option has been selected.

Step 1 - Verifies that the channel under test responds to ROCL at the address to which its CHAN ADDR switch is set. It also verifies that the channel responds at only one address.

Case 1 Verifies channel addresses 2 through 7.
(Channel 1 is not used due to IMBA PCA.)

Case 2 Verifies channel addresses 8 through 14.
(Channel 15 is not used due to Mailbox Location 0.)

Step 2 - Case 3 Verifies that the channel number set on the CHAN ADDR switch is the same as that read from register D by OBII.

Step 3 - Case 4 Verifies that the channel number received by OBSI is the same as that set on the CHAN ADDR switch.

Step 4 - Verifies that bits 0 and 3-15 of register E (CHANNEL CONFIGURATION REGISTER) are read as zeroes and that the DEVICE TYPE and PROCESSOR switches operate properly. Should this step indicate that both switches are inoperable (a highly unlikely situation), the diagnostic is exited. Should manual checking indicate that the switches are operational, it is likely that register E cannot be read.

Case 5 Verifies bits 0 and 3-15 of register E.

Case 6 Verifies both switches.

Case 7 Verifies the DEVICE TYPE switch.

GIC Diagnostic

- Case 8 Verifies the PROCESSOR switch.
- Step 5 - Verifies SYS CTRL switch.
- Step 6 - Case 9 Executed only if interrupts are found to be set in register C at this point by reading this register. Verifies that such interrupts are cleared by the IMB INIT command.
- Step 7 - Verifies that interrupts may be set for all device numbers and that each device greater in priority than device 7 has priority over device 7. Device numbers not responding properly are reported through error messages.
 - Case 10 Verifies devices 0 through 3
 - Case 11 Verifies devices 4 through 7
 - Case 14 Verifies that the proper device number is received from OBII and that the NOT VAL bit is clear indicating valid interrupt data.
- Step 8 - Case 12 Verifies that the GIC does not assert IRQ when interrupts have been set but SMSK has disabled IRQ.
- Step 9 - Case 13 Verifies that the GIC under test responds properly to IPOLL when interrupts have been set and SMSK enables IRQ. Interrupts are turned off at the CPU for this step.
- Step 10 - Case 12 Verifies that the GIC does assert IRQ when interrupts have been set and SMSK has enabled IRQ.
 - Case 14 Verifies that the proper device number is received from OBII.
- Step 11 - Checks that IRQ was cleared by interrupt service.
- Step 12 - Case 9 Verifies that IMB INIT clears pending interrupts.
 - Case 15 Verifies that writing to register C does not assert DNV on the IMB.
- Step 13 - Case 16 Verifies that IMB INIT clears the interrupt mask for the channel under test by setting an interrupt for device 7 and verifying that IRQ is not asserted on the IMB.

Step 14 - Verifies that setting NEW STATUS using SIOP clears the NOT VAL bit in register F. Were the MODE switch in the OPERATE (in) position at this time, clearing bit 12 of register F would not inhibit setting CSRQ when NOT VAL is cleared by this step, causing a system crash. Therefore, before the SIOP is executed to set the NEW STATUS, the value of this switch is read, and if incorrectly set, the operator is prompted until he corrects the setting (to the out position) before the diagnostic will proceed. The NEW STATUS is not set if NOT VAL, register F, is already cleared when tested before SIOP, but the remainder of this step is executed.

The remainder of this step verifies that NOT VAL is cleared by SIOP setting NEW STATUS, that the proper device number appears in register F, that only a DEV REQ is indicated, and that bits 0-4, register F, are zeroes.

This step is executed eight times, starting with device number 7 and counting down, and therefore verifies that the priority encoder functions properly for this particular sequence of NEW STATUS bits set.

Case 17 Verifies the NOT VAL bit is cleared.

Case 20 Verifies proper device number (checks the priority encoder).

Case 18 Verifies SRQ, CHAN REQ cleared, DEV REQ set.

Case 19 Verifies that bits 0-4 are zero.

Step 15 - Case 20 Verifies that SIOP can clear the NEW STATUS previously set for device 0.

Step 16 - Case 20 Verifies that HIOP can clear the NEW STATUS for device 1.

Step 17 - Case 21 Verifies that the GIC under test responds to SPOLL2 at its proper channel address when MYCSRQ is asserted.

Step 18 - Case 22 Verifies that the GIC under test responds to SPOLL1 at its proper channel address when MYCSRQ is asserted.

Step 19 - Case 23 Verifies that OBSI returns the proper information previously set up by the series of steps preceding this one. The data returned by OBSI should be 1027X where X may be !8, !9 or !A. If all tests involving SIOP and HIOP have passed, OBSI should return 1027A.

GIC Diagnostic

4.2 SECTION 2 - PHI VERIFICATION

Section 2 contains the PHI diagnostic.

Step 20 - Case 24 Verifies that INIT clears the NEW STATUS register. Note, however, that the conditions that are being cleared are the conditions set by section 1. Although section 2 may be looped and no error condition created by not preceding it with section 1, neither will this step provide any meaningful data about the ability of INIT to clear NEW STATUS.

Step 21 - Case 25 Checks the PHI in off-line mode by writing all bit combinations to the registers and reading the results. None of the non-data lines whose outputs drive GIC circuitry (DMARQ, etc.) are checked. This test runs approximately one minute. Failure of this test does not necessarily indicate a bad PHI, as much of the GIC logic is involved in the reads and writes to the PHI. If a replacement PHI yields the same symptoms, the problem is probably elsewhere on the PCA. If this step fails, the diagnostic is automatically exited, as many of the tests performed after this point would be meaningless.

4.3 SECTION 3 - PARTIAL CSRQ TEST

This section tests part of the CSRQ logic.

Step 22 - Case 26 Verifies that the DMA BUSY and PHI INT bits in register B are both cleared. If this step fails, further testing is probably meaningless, so the diagnostic is exited automatically.

Step 23 - Case 27 Verifies that no service requests are pending on the GIC under test. Should this test fail, the diagnostic is exited, as further testing is probably meaningless.

Step 24 - Verifies that setting the DO SRQ bit in register F causes only the SRQ bit in register F to be set, the NOT VAL bit cleared, and device 7 to be indicated. This tests a portion of the CSRQ logic.

Case 28 Verifies SRQ set, CHAN REQ, DEV REQ cleared.

Case 29 Verifies NOT VAL cleared.

Case 30 Verifies device number equals 7.

4.4 SECTION 4 - PARTIAL CSRQ TEST

This section tests more of the CSRQ logic.

Step 25 - Case 31 Verifies that writing device number 3 to register F causes device number 3 to be read from register B.

Step 26 - Causes a simulated parallel poll response for device 7 on the HP-IB and then verifies that this response causes NOT VAL to be cleared, device 3 to be indicated, and only CHAN REQ to be asserted in register F.

Case 32 Verifies NOT VAL is cleared.

Case 33 Verifies device number equals 3.

Case 34 Verifies only CHAN REQ asserted.

Step 27 - Performs an OBSI, which should receive the parallel poll response device number 7 instead of the DMA device 3. The OBSI should show the NOT VAL bit cleared and a DEV REQ only.

Case 38 Verifies device 7.

Case 35 Verifies NOT VAL cleared.

Case 36 Verifies DEV REQ only.

4.5 SECTION 5 - PARALLEL POLL PRIORITY

The parallel poll priority encoder is checked by this section.

Step 28 - Case 37 Verifies that the INIT command clears the DMA device number read from register B.

Step 29 - Case 38 Partially tests the parallel poll priority encoder. To do this, a parallel poll response is generated for device 7 with each of the other device numbers, one at a time. Since device 7 has lowest priority, the other device numbers will be read by OBSI. This is a partial test only, since there are many conditions in the priority encoder not checked by this test.

GIC Diagnostic

4.6 SECTION 6 - PARTIAL CSRQ TEST

This section tests certain paths in the CSRQ logic. Refer to the figure entitled "CSRQ Logic Flowchart" of the GIC ERS. The condition of the channel is altered and the effect observed by noting the state of the NOT VAL bit in register F.

- Step 30 - Case 39 SRQ is asserted on the HP-IB. This should clear the NOT VAL bit in register F. If the NOT VAL bit is not cleared, either SRQ cannot be asserted or the CSRQ logic is inoperable. The remainder of this section is bypassed.
- Step 31 - Case 40 The NO POL bit, register F, is set and verified by reading register B. If this bit cannot be set, step 32 is bypassed as the CSRQ function it tests cannot be activated.
- Step 32 - Case 41 Reads register F to verify that NO POL, set by the previous step, set NOT VAL. Since the ability to clear the NOT VAL bit by asserting SRQ has already been tested by Step 30, the only logic included in this test is the data path from the NO POL bit to the IMB through the read register F logic.
- Step 33 - Case 42 Verifies that asserting DMARQ causes the CSRQ logic to assert NOT VAL in register F. First the NO POL bit, register B, is cleared, then register 6 is written with !0002 to generate DMARQ from the PHI. Register F is read to verify NOT VAL set.
- Step 34 - Case 43 Clears DMARQ by writing !0 to register 6 to clear NOT VAL, then sets NOT VAL by clearing EOI (by writing !405C to register 0). Reading register F verifies that NOT VAL is set.

4.7 SECTION 7 - REGISTERS 8,9, AND A

This section verifies that each bit in registers 8, 9 and A can be written and read independently of all other bits in those registers. Testing these registers also checks the IMB and GIC data buffers for adjacent data line shorts.

- Step 35 - Verifies that bits in register 8 are independent and operational.
 - Case 44 Verifies bits 12-15, register 8.
 - Case 45 Verifies bits 8-11, register 8.
- Step 36 - Verifies that bits in register 9 are independent and operational.

Case 46 Verifies bits 12-15, register 9.

Case 47 Verifies bits 8-11, register 9.

Case 48 Verifies bits 4-7, register 9.

Case 49 Verifies bits 0-3, register 9.

Step 37 - Verifies that bits in register A are independent and operational.

Case 50 Verifies bits 12-15, register A.

Case 51 Verifies bits 8-11, register A.

Case 52 Verifies bits 4-7, register A.

Case 53 Verifies bits 0-3, register A.

4.8 SECTION 8 - DMA STATE MACHINE

This section begins the test of the DMA state machine. To test this sequential state machine, which is normally driven by an independent oscillator, the DIAGNOS bit in register F is used to disable the state machine oscillator, and allow the timing of the DMA machine to be synchronized with the slave handshake. The state machine is clocked by the slave handshake, one clock to the DMA machine for every slave handshake performed. Three slave handshakes are required for one state time of the DMA machine. Before starting DMA, the DMA control register is checked to see that all bits operate properly. Short DMA transfers are relied upon for all synchronized DMA operations in this and later sections.

Step 38 - Case 54 Verifies that the DMA EN bit, register 8, is not set when this section of the diagnostic is begun.

Step 39 - Verifies that bits 9-11 of register B can be set by writing to register F, and that writing to register F does not start DMA (indicated by DMA BUSY, register B, not being set).

Case 55 Verifies DMA BUSY is not set.

Case 58 Verifies RT BYT bit.

Case 56 Verifies NO END bit.

Case 57 Verifies DMA OUT bit.

Step 40 - Verifies that bits 9-11, register B, can be cleared by writing to register F.

GIC Diagnostic

- Case 58 Verifies RT BYT bit.
- Case 56 Verifies NO END bit.
- Case 57 Verifies DMA OUT bit.

Step 41 - Sets up a synchronized DMA transfer, an output transfer starting on the left byte, with initial byte count of zero, with NO END set, NO POL cleared, for device 3. The starting memory address is !FFFE. DMARQ s asserted throughout the transfer by loading register 6 with !0002. The DMA EN bit, register 8, is checked to verify that starting DMA sets it. The DMA BSY bit, register B, is checked to verify that it is set. PAR ERR, ADR OVF, MEM TIM, and DMA STATUS are verified to be cleared. Register E is written to the DMA transfer to verify the abort path. Register 8 is read to assure that state 26 is reached within 10 slave hand-shake operations of the write to register B.

Case 59 Verifies that write register B set DMA EN, register 8.

Case 60 Verifies that write register B set DMA BSY.

Case 61 Verifies that the PAR ERR bit can be cleared.

Case 62 Verifies that the ADR OVF bit is cleared.

Case 63 Verifies that the MEM TIM bit is cleared.

Case 64 Verifies DMA STATUS can be cleared.

Case 65 Verifies DMA state progression from state 0 to 26. Failure of this test terminates diagnostic execution.

Step 42 - Case 66 Verifies the CSRQ information from register F at this point in the DMA transfer. Although DMA was aborted between Case 64 and Case 65, DMA ABT is not yet set, as the DMA machine has not yet entered state 5. Contents of register F should be !00FF.

Step 43 - Verifies that DMA BSY is still set, but that DMA STATUS indicated 11 (abort). The abort occurred between Case 64 and Case 65.

Case 67 Verifies DMA BSY still set.

Case 68 Verifies DMA STATUS is 11.

Step 44 - Verifies state progression from state 26 to state 5. It also verifies that writing to register E cleared the DMA EN bit in register 8.

Case 69 Verifies DMA EN cleared.

Case 70 Verifies DMA state progression from state 26 to state 5. Failure of this test causes the diagnostic to be exited.

- Step 45 - Case 71 Verifies the CSRQ information in register F at this point in the DMA transfer. Register F should contain !09FB. For this test to pass, the DMA machine must have reached state 5, and the PHI have asserted its INTERRUPT line. To insure that PHI INTERRUPT will be asserted, register 3 was loaded with !FFFF before this transfer was initiated. If this test fails, the diagnostic is exited, since the CSRQ logic has now been thoroughly tested and should be diagnosable. This allows for simplification of the following test data.
- Step 46 - Case 72 Verifies that the RT BYT bit was toggled during state 26.
- Step 47 - Verifies that the DMA BYTE COUNT register, register A, previously loaded with !0, was decremented to !FFFF following state 26. This value was chosen to verify that the ripple clock outputs properly decremented the succeeding clock stages. This also verifies that the counter bits may be set by the clock inputs. This does not verify that the bits in the counter are completely functional, as some bits may conceivably set but not clear when the counter is clocked.
- Case 73 Verifies bits 12-15, register A, for !F.
- Case 74 Verifies bits 8-11, register A, for !F.
- Case 75 Verifies bits 4-7, register A, for !F.
- Case 76 Verifies bits 0-3, register A, for !F.
- Step 48 - Case 77 Verifies that the state machine remains in state 5 for multiple state times until OBSI is executed to the channel. If register 8 indicates state 5, this is verified, since it has been more than one state time since state 5 was executed. (State 5 was entered before test 71). If this step fails because the wait for OBSI fails, the state read will be either 0 or 4. If this test fails, diagnostic execution is terminated.
- Step 49 - Case 79 Performs a read of register 2 while DMA is active, and verifies that this causes DNV to be asserted in the IMB.
- Step 50 - Case 78 Issues OBSI to the channel under test, then clocks the state machine one state time, and verifies that state 4 is then entered. This checks that OBSI allows progression from state 5 to state 4. If this

GIC Diagnostic

test fails, diagnostic execution is terminated.

- Step 51 - Case 80 Performs a read of register F while DMA is active and verifies that DNV is not set on the IMB.
- Step 52 - Case 81 Verifies that three DMA clock cycles after state 4 was entered, state 0 has been reached. If this test fails, diagnostic execution is terminated.
- Step 53 - Case 82 Verifies that starting a left byte output DMA transfer drives the state machine through state 26 to state 27 from state 0. If either state 26 or 27 is not reached, diagnostic execution is terminated.
- Step 54 - Verifies the state machine progression from state 27 through state 17 to state 16. The transition from state 17 to 16 depends on the assertion of MEMDN and the unassertion of CNT0. During this step, DMA is aborted by a write to register E. If improper states are reached diagnostic execution terminates.
 - Case 81 Verifies transition from state 27 to 17.
 - Case 83 Verifies state 17 to 16 transition.
- Step 55 - Case 84 Verifies DMA state machine progression from state 16 to state 24. This requires assertion of IOEND, indicating a PHI handshake completion. However, no check of the data is made so this step could succeed and the PHI data handshake still be defective. If an incorrect state is reached, execution is terminated.
- Step 56 - Case 85 Verifies that register A (DMA BYTE COUNT) has been decremented twice since this DMA started, from !FFFF to !FFFD. The starting value was the result of the preceding transfer in this section. If !FFFD is not read, execution terminates.
- Step 57 - Case 86 Verifies that the write to register 14 executed during step 54 causes the state machine to progress from state 24 to state 5. If state 5 is not reached, the diagnostic is terminated. OBSI is also issued and the progression from state 5 to state 4 is verified. Failure of the state machine to reach state 4 causes diagnostic execution to terminate, but test 86 is not marked as failing in that case.

4.9 SECTION 9 - RIGHT OUTPUT DMA TRANSFER

This section sets up and executes a synchronized output DMA transfer of two bytes, starting with the right byte, which terminates normally on a count-equal-zero condition. Refer to Section VI for an explanation of synchronized DMA transfers.

GIC Diagnostic

- Step 58 - Case 87 Asserts DMARQ from the PHI by writing 12 to register 6 to cause DMARQ to look for outbound FIFO not full (always true for a two-byte transfer), sets up registers 8,9, and A to 10, 1CFFF, 1FFFD respectively, starts the DMA transfer, and synchronizes the program with the DMA clock. It verifies that DMA goes from state 0 to state 28. Reaching an incorrect state terminates the diagnostic.
- Step 59 - Case 88 Verifies state progression from state 28 to 24. This progression requires the assertion of MEMDN by the master handshake. This test does not verify that any data has been read as a result of this handshake. Failure to reach state 24 terminates diagnostic execution.
- Step 60 - Case 89 Verifies that DMA will go from state 24 to 25. This requires assertion of both DMARQ and DMAENF. If this test fails, execution terminates.
- Step 61 - Verifies DMA state machine progression from state 25 through state 29 to state 30, and verifies that register 9 (DMA ADDRESS REGISTER) was incremented to !D000. Arrival in incorrect states will terminate diagnostic execution.
- Case 81 Verifies progression from state 25 to 29.
- Case 90 Verifies bits 12-15, register 9, are !0.
- Case 91 Verifies bits 8-11, register 9, are !0.
- Case 92 Verifies bits 4-7, register 9, are !0.
- Case 93 Verifies bits 0-3, register 9, are !D.
- Case 94 Verifies progression from state 29 to 30.
- Step 62 - Transfers the second byte of this transfer to the PHI. It verifies state changes from 30 through 26, 27, 17, 1 and 5 to 4. State changes occurring incorrectly terminate the diagnostic. Fault data is not entered for state changes that have already been verified, but the failures are reported through error messages. The byte count equal zero condition to terminate the transfer is created artificially by writing !0 to register A during this step.
- Case 95 Verifies change from state 30 to 26.
- Case 83 Verifies change from state 17 to 1.
- Case 95 Verifies change from state 1 to 5.

GIC Diagnostic

Case 96 Verifies DMA STATUS is 0, indicating normal termination.

4.10 SECTION 10 - RIGHT OUTPUT DMA TRANSFER

This section performs a test of the DMA state machine to verify state transitions in the output, right byte branch of the DMA state diagram. State transitions, once successfully completed, but which now fail are reported by error messages, but no data is entered into the Fault Data Summary. All incorrect state changes terminate execution of the diagnostic. Refer to Section VI for an explanation of synchronized DMA transfers.

Step 63 - Case 94 An output transfer of 1 byte, to start on the right byte, is initiated by a write to register B. DMARQ is held asserted by writing !2 to register 6. States 28, 24, 25 and 29 are traversed without FAULT DATA reporting, but errors are reported through error messages. These transitions depend on assertion of MEMDN, DMA EN, and DMARQ. The change from state 29 to 7 requires assertion of CNT0. Failure of this transition is reported in the FAULT DATA SUMMARY if this failure occurs. If any of these state transitions fail, execution is terminated.

Step 64 - Case 95 Verifies the state transition from state 7 to state 5. This transition depends on the assertion of IOEND. However, the data output is not verified, and this test succeeding does not indicate that the PHI will properly receive data. Failure of this step terminates execution.

Step 65 - Case 97 Executes OBSI to the channel under test and clocks the state machine to state 4. It verifies that DMA EN is cleared when state 4 is entered. Failure to enter state 4 terminates diagnostic execution.

4.11 SECTION 11 - LEFT INPUT DMA TRANSFER

This section performs a test of a DMA input transfer, receiving and writing only 1 byte. This requires that a whole word be read from memory, the proper byte be replaced by the byte input from the PHI, and the word be written back into memory. This allows testing of some of the byte packing and unpacking logic, decrementing of the byte count, and setting of proper status.

Step 66 - Case 87 Sets up the DMA transfer by programming the PHI for loop-back (!60 to register 7), and loading !CC into the inbound FIFO, by loading memory address !CFFF with !3333 and by writing the proper data to register B to start the transfer. The DMA clock is synchronized. Arrival in state 8 from state 0 is verified. If this state is not entered, the diagnostic is exited.

GIC Diagnostic

- Step 67 - Case 98 Verifies that the state machine can go from state 8 to state 9. This depends on assertion of DMA EN and DMARQ. Special circuitry is required to keep DMARQ asserted after DMIOG01 is asserted when operating a synchronized transfer. This logic may cause failure in the synchronized DIAGNOS mode but not in normal operation. Failure of this state transition will terminate program execution.
- Step 68 - Case 73 Verifies that state 8 decremented the byte count to 10.
- Step 69 - Case 81 Verifies the state change from state 9 to 20. Failure of this test will terminate program execution.
- Step 70 - Case 99 Verifies assertion of IOEND as a result of DMIOG01 in state 20. State 22 cannot be reached until IOEND is asserted. Data from this PHI handshake is checked later in the diagnostic, but failure of the master handshake or byte packing circuitry could mask a correct data transfer. Failure will terminate execution.
- Step 71 - Case 100 Verifies the state transition from state 22 to 18. This is dependent on assertion of CNT0, indicating the detection of the byte count being decremented to 0. Failure of this test terminates diagnostic execution.
- Step 72 - Case 101 Verifies that DMA STATUS, set during step 71, is 10. If this test fails, it might indicate PHI malfunction.
- Step 73 - Case 81 Verifies state transition from state 18 to 19. If this test fails, the diagnostic terminates.
- Step 74 - Case 88 Verifies the state transition from state 19 to 23. This requires the assertion of MEMDN in response to DMRDRQ. Failure of this test terminates diagnostic execution.
- Step 75 - Verifies state machine progression from state 23 through 21 to 5. Arrival in state 5 indicates that DMWRRQ generated MEMDN. This step does not check the validity of the data sent. Failure to arrive in the correct states will cause termination of the diagnostic.
- Case 81 Verifies transition from 23 to 21.
- Case 102 Verifies transition from 21 to 5.
- Step 76 - Verifies that the word in memory accessed by this DMA (location !CFFF) has had the left byte written as !CC and the right byte left untouched as !33. This is the first time that the data from DMA reads and writes has

GIC Diagnostic

been checked so extensive blocks of circuitry are implicated should this step fail. If it does fail, diagnostic execution is terminated.

Case 103 Verifies left byte is 1CC

Case 104 Verifies right byte is 133

Step 77 - Case 105 Verifies that the IMB command INIT will clear the state to zero when executed.

4.12 SECTION 12 - DMA INPUT TRANSFERS

This section performs several DMA transfers to test various paths through the DMA state diagram.

Step 78 - Verifies that register 9 is incremented, register A is decremented and DMA STATUS set to 10 when this right-byte input transfer is performed. States traversed are 6, 2, 10, 11, 15, 13, 5, 4 and 0, in that order. State transition failures are reported through error messages, but data on these failures is not reported in the FAULT DATA SUMMARY. However, when an incorrect state is entered, the diagnostic execution is terminated. Correct decrementing of register A is indicated by a correct transition from state 15 to 13.

Case 101 Verifies DMA STATUS is 10

Case 106 Verifies DMA ADDRESS is 1D000

Step 79 - Case 107 Verifies that the right data byte is properly read from the PHI and written into the proper word in memory (location 1CFFF). The right byte should be 144. Failure terminates execution.

Step 80 - Case 108 Verifies that the left data byte of the memory word (location 1CFFF) was restored by the DMA machine as it was before the transfer. The left data byte should be 133.

Step 81 - Case 70 Initiates a DMA input transfer, starting on the left byte, and then aborts it by writing to register E to verify that the state machine can go from state 8 to 5. OBSI is then issued, and DMA driven to state 4. Incorrect state transitions cause execution to terminate.

Step 82 - Verifies that the DMA state machine detects the condition where CNT0 is not asserted by starting an input transfer of more than 1 byte on the left byte, allowing it to progress from state 0 through states 8, 9, 20, and 22 to state 10. Arrival in state 10 indicates CNT0 not asserted was detected. A write register

GIC Diagnostic

E in state 20 causes the next state after 10 to be state 18, indicating the abort succeeded.

Case 100 Verifies CNT0 unasserted

Case 70 Verifies state 10 to 18 transition due to WREG E.

4.13 SECTION 13 - RIGHT TO LEFT BYTE DMA PATH

This section checks the path from state 15 to state 14. (Refer to Section VI for notes on synchronized DMA.)

Step 83 - Case 109 Verifies DMA state transitions from state 0 through states 6, 2, 10, 11, 15, and 14. Arrival in state 14 requires that CNT0 be unasserted and MEMDN be asserted. Only the arrival in state 14 is checked, and failure to arrive in state 14 will cause termination of the diagnostic. This path is checked by an input transfer starting on the right byte.

Step 84 - Case 81 Verifies the state transition from state 14 to state 8. Failure to reach state 8 terminates execution of the diagnostic.

Step 85 - Case 106 Verifies that state 14 incremented register 9 (DMA address register) to !D000. This checks the ripple clock propagation to all states.

4.14 SECTION 14 - DMA WAIT STATES

This section checks various wait states where the DMA state machine is to wait for the assertion of DMARQ. These are checked by removing DMARQ and then entering those states and verifying that those states are not exited. (Refer to Section VI for notes on synchronized DMA.)

Step 86 - Case 110 Starts a right byte output transfer and verifies that the state machine hangs in state 24. Failure to remain in state 24 terminates program execution.

Step 87 - Case 110 Starts a left byte output transfer and verifies that the state machine hangs in state 26. Failure to remain in state 26 causes diagnostic execution to be terminated.

GIC Diagnostic

4.15 SECTION 15 - DMA WAIT STATES

This section checks various wait states where the DMA state machine is to wait for the assertion of DMARQ. These are checked by removing DMARQ and then entering those states and verifying that the state machine is still in the same state one state time later. (Refer to Section VI for notes on synchronized DMA.)

Step 88 - Case 110 Starts a left byte input transfer and verifies that the state machine hangs in state 8. Failure to do so results in termination of the diagnostic.

Step 89 - Case 110 Starts a right byte input transfer and verifies that the state machine hangs in state 10 when DMARQ is not asserted. Failure to remain in state 10 causes the diagnostic to terminate.

4.16 SECTION 16 - ADDRESS ROLLOVER

This section generates address rollover, which should abort the DMA transfer and set DMA states to 11. (Refer to Section VI for notes on synchronized DMA).

Step 90 - Case 111 Starts a right byte output transfer with address equal to !FFFF. States 0, 28, 24, and 25 are traversed, and an address roll-over is generated. DMA EN is checked to verify that it has been cleared.

Step 91 - Case 112 Verifies that the address rollover generated in step 90 set DMA STATUS to 11 and set the ADR OVF bit in register B.

4.17 SECTION 17 - IMB MEMORY TIMEOUT

This section tests the memory timeout logic by accessing non-existent memory. (Refer to Section VI for notes on synchronized DMA.)

Step 92 - Case 113 Starts a DMA transfer that accesses memory location !0FF 0003 ; an address that is hopefully non-existent. This causes a memory timeout to be generated. This occurrence is verified by reading register B and seeing that DMA STATUS is 11 and MEM TIM is set.

4.18 SECTION 18 - PARITY ERROR ABORT

This section checks the parity error circuitry in the master handshake and clearing of abort DMA STATUS. (Refer to Section VI notes on synchronized DMA.)

- Step 93 - Case 114 Checks the parity error logic in the master handshake by creating a word in memory with incorrect parity and reading it with a DMA transfer. The PAR ERR bit, register B, is read to be sure it is set.
- Step 94 - Case 115 Verifies that a write to register B clears DMA STATUS from 11 to 00.

4.19 SECTION 19 - DNV ASSERTION

(Refer to Section VI for notes on synchronized DMA.)

- Step 95 - Case 116 Verifies that DNV is asserted on the IMB when a read of register 0 is attempted with no parallel poll in progress and no byte in the inbound FIFO.

4.20 SECTION 20 - DNV ASSERTION

(Refer to Section VI for notes on synchronized DMA.)

- Step 96 - Case 117 Verifies that DNV is not asserted on the IMB when register 0 is read and a parallel poll is in progress on the HP-IB.

4.21 SECTION 21 - NEW STATUS WHEN NOT CACS

(Refer to Section VI for notes on synchronized DMA.)

- Step 97 - Case 118 Determines that NEW STATUS may be set by SIOP when the channel is not CACS by verifying that NOT VAL is cleared following the operation.

4.22 SECTION 22 - DNV ASSERTION

(Refer to Section VI for notes on synchronized DMA.)

- Step 98 - Case 119 Verifies that DNV is not asserted when register 0 is read, no parallel poll is in progress but DMARQ is asserted (signifying a byte in the inbound FIFO).

GIC Diagnostic

4.23 SECTION 23 - DMA LOOPBACK TESTS

Loopback tests of DMA. (Refer to Section VI for notes on synchronized DMA).

Step 99 - Case 121 Verifies that the last byte of an output transfer has EOI appended to it when this function has been enabled. This is checked by doing an output transfer with the PHI in loop-back mode and then reading the byte transferred from the inbound FIFO. Bits 0 and 1 should be 11.

Step 100 - Case 122 Verifies that the right byte of the last word read before an address rollover is detected by the state machine is not cleared before it is transferred to the HP-IB. If both right and left bytes are incorrect (reported by error messages), a more serious problem is indicated than only the alteration of the right byte. Perhaps memory reads or PHI writes are not operating properly.

4.24 SECTION 24 - GIC TIMEOUT LOGIC CHECK

This section verifies correct operation of the GIC timeout logic. The section is done in four steps.

Step 101 - Normal interrupt, halt

The channel program:

```
WREG 0, !405E          <<address chan to talk>>
INT H, 1, F           <<end >>
```

is executed. Normal channel program processing is expected. After allowing sufficient time (milliseconds) for the channel program to complete, the second CPVA word is checked to insure a value of !800F, the expected halt code.

Step 102 - Receive data request sent and no data

The channel program:

```
WREG 0, !C000          <<receive data request>>
INT H, 1, 1           <<end>>
```

is executed. This program should not complete normally, but should be aborted by the CPU because the inbound FIFO of PHI has been told to receive data then no data is transferred. The 4th DRT word is checked during the timeout to insure it contains the value !8004 (the program is suspended waiting for another service request), a sufficient time (1 second) is

allowed for the timeout abort, then CPVA word 0 is checked to insure a value of !E004 (timeout).

Step 103 - Receive data request - DMA INPUT

The channel program:

```
WREG 0, !C000          <<receive data request>>
EXECUTE DMA           <<byte count=9, input>>
END
```

is executed. This program should time out because more bytes are expected by the channel program (9 bytes) than can be provided by the inbound FIFO. The 4th DRT word is verified to be !8002 (channel program and DMA active) a sufficient time (1 second) is allowed for the timeout to occur, then CPVA word 0 is verified to be !E004 (timeout abort).

Step 104 - Receive data request - DMA output

The channel program:

```
WREG, 0, !C000        <<receive data request>>
EXECUTE DMA           <<output, byte count=2>>
END
```

is executed. This program should time out because a read data request is sent, then an OUTPUT transfer is executed. The same tests of the DRT and CPVA words described in the previous step are executed.

4.25 SECTION 25 - GIC-TO-GIC TRANSFERS

Step 105 - Case 120

This section uses a known good second GIC to provide stimulus to, and to respond to stimulus provided by the GIC under test. (Refer to paragraph 1.1 for cabling instructions.)

The GIC under test, using HP-IB device address 5, transfers a 256 byte buffer on the HP-IB to device address 3 on the known good GIC. Then, the direction of transfer is reversed and 256 bytes are transferred on the HP-IB from device address 3 of the known good GIC to device address 5 of the GIC under test. This sequence is repeated 100 times. A dot appears on the display every other time. The DRT and CPVA areas of both channel/device number are tested to insure completion of these channel programs. Then, the transfer buffers are checked to verify the ability of the GIC under test to transfer data in and out of memory. Interface Clear signal is sent via the second GIC to the one under test to check its response.

GIC Diagnostic

ERROR INTERPRETATION	SECTION V
----------------------	--------------

5.0 INTRODUCTION

The table on the following page relates functions and blocks tested (e.g., channel program execution; PHI) to specific tests (Sections in the Diagnostic) and subsystems (GIC, IMB, CPU, and Memory).

There are two ways to use the table:

- (1) Enter the table at the failing section. This tells you:
 - a. which function or block was under test
 - b. the most likely failing subsystems in order of probability
- (2) Enter the table at a block or function. This tells you:
 - a. which test section explicitly tests that block or function
 - b. which test sections depend on that block or function in order to pass
 - c. which subsystems cause failure of that block or function

If every block or function were independent of the others and could be tested independently, then the table would have only a diagonal line of 'X's. Since this is not the case, the table provides a compact guide to these interrelationships; to deal with the cases in which the subsystem appears to fail due to the failure of other subsystems.

(See separate Specialist-level document, the GIC Diagnostic IMS, for a component-level table.)

GIC Diagnostic

Block or Function Tested	Section Number										Subsystems In Order Of Failure Probability
	1	2	3	8	17	18	19	23	24	25	
Switches & Interrupts	X	-	-	-	-	-	-	-	-	-	GIC, IMB, CPU
PHI	-	X	-	-	-	-	-	X	-	-	PHI, GIC
CSRQ and Registers	-	-	X	-	-	-	-	-	-	-	GIC, PHI, IMB
DMA State Machine	-	-	-	X	-	-	-	-	-	-	GIC, PHI, IMB, Memory
Memory Timeout	-	-	-	-	X	-	-	-	-	-	GIC, IMB, Memory, CPU
Mem Parity Error	-	-	-	-	-	X	-	-	-	-	GIC, IMB, Memory
DNV on IMB	-	-	-	-	-	-	X	-	-	-	GIC, IMB, CPU
Channel Programs	-	-	-	-	-	-	-	-	X	X	GIC, CPU, Memory
GIC-GIC Transfer	-	-	-	-	-	-	-	-	-	X	PHI, GIC, HP-IB

Key: X = function or block under test
 - = function or block must work to pass test

Figure 5.1 Function/Block Reference Table

- Case - The entity which verifies proper circuit conditions along a distinct path in the logic, which has a unique case number and has associated with it one or more location designators. Cases verify portions of the board which correspond to the location designators associated with all signals which are active during the test.
- Channel - In this diagnostic, this refers to the 31262A General I/O Channel.
- Error Message - A message output when an error condition is detected in response to a diagnostic stimulation. Error Messages generally contain symptom data specific to the error encountered.
- PHI - Processor to HP-IB Interface chip.
- Synchronized DMA Transfer - A DMA transfer which is run in synchronization with the slave handshake, where the DMA state variables are updated once every three slave handshake operations of the channel being diagnosed. To enable the DMA state clock to become properly synchronized, so that the state variable change occurs after the third slave handshake, the state variables are sampled until the change is detected by a diagnostic program sub-routine before any section requiring synchronized transfers is executed.

