Honeywell

Process Manager Module Test System

PM13-505

PM/APM/HPM Service - 2

Process Manager Module Test System

PM13-505
9/95

Copyright, Trademarks, and Notices

Revision 01 – September 15, 1995

About This Publication

This Reference Manual documents the individual test programs that make up the Process Manager Module Test System. It is intended for use by either Honeywell or customer service technicians who are responsible for the isolation of those TDC 3000X hardware failures that are not fully identified by the firmware and on-line software diagnostics. This publication is a reference manual for trained technicians and is intended to supplement TDC 3000X service training, not replace it.

This manual supports both the Process Manager (PM) and the Advanced Process Manager (APM).

In order to use this manual, you must first be familiar with the content of the *Process Manager Test Executive (PMEX).*

NOTE

Hexadecimal numeric values are indicated in this manual by the use of a preceding "$"; for example, $0000 or $FFFF.

Table of Contents

1 INTRODUCTION

1.1 What PMMTS is and How it Works
1.1.1 PMMTS Packaging
1.1.2 Test Program Executive
1.2 Uses of PMMTS
1.3 References

2 COMMON CPU TESTS (CCPU)

2.1 Common CPU Test Elements
2.2 Use Information
2.3 Setup Parameters
2.3.1 CCPU-Specific Parameters
2.3.2 General Parameters
2.4 Preset Parameter Values for CCPU
2.5 Mode Characteristics for CCPU
2.6 Individual CCPU Tests

3 GLOBAL MEMORY TEST PROGRAMS (GMEM)

3.1 Global Memory Test Elements
3.2 Use Information
3.3 Setup Parameters
3.3.1 GMEM-Specific Parameters
3.3.2 General Parameters
3.4 Preset Parameter Values for GMEM
3.5 Mode Characteristics for GMEM
3.6 Individual Global Memory Tests

4 <reserved>

5 LOCAL MEMORY TESTS (LMEM)

5.1 Local Memory Test Elements
5.2 Use Information
5.3 Setup Parameters
5.3.1 LMEM-Specific Parameters
5.3.2 General Parameters
5.4 Preset Parameter Values for LMEM
5.5 Mode Characteristics for LMEM
5.6 Individual Local Memory Tests

Table of Contents

6 I/0 LINK TESTS (LNKI)

6.1 I/O Link Tests Elements
6.2 Use Information
6.3 Setup Parameters
6.3.1 LNKI-Specific Parameters
6.3.2 General Parameters
6.4 Preset Parameter Values for LNKI
6.5 Mode Characteristics for LNKI
6.6 Individual I/O Link Tests

7 CONTROL REDUNDANCY TESTS (RDUN)

7.1 Control Redundancy Tests Elements
7.2 Use Information
7.3 Setup Parameters
7.3.1 RDUN-Specific Parameters
7.3.2 General Parameters
7.4 Preset Parameter Values for RDUN
7.5 Mode Characteristics for RDUN
7.6 Individual RDUN Tests

8 TOKEN BUS CONTROLLER STATISTICS (TBCS)

8.1 Token Bus Controller Test Elements
8.2 Use Information
8.3 Setup Parameters
8.3.1 TBCS-Specific Parameters
8.3.2 General Parameters
8.4 Preset Parameter Values for TBCS
8.5 Mode Characteristics for TBCS
8.6 Individual Token Bus Controller Tests

9 PROCESS SEQUENCE OF EVENTS (PSOE)

9.1 Process Sequence of Events Test Elements
9.2 Use Information
9.3 Setup Parameters
9.3.1 PSOE-Specific Parameters
9.3.2 General Parameters
9.4 Preset Parameter Values for PSOE
9.5 Mode Characteristics for PSOE
9.6 Individual Process Sequence of Events Tests
9.7 Alarm Messages and Descriptions

Table of Contents

APPENDIX A — HARDWARE INTERACTIONS USED BY LINKI

A.1 Interactions of Common Events
A.2 Interactions of All Tests
A.3 Interactions of Test 1 (Reset and Self-Test)
A.4 Interactions of Test 2 (Shared RAM Arbitration)
A.5 Interactions of Test 3 (SRAM Data Pattern)
A.6 Interactions of Test 4 (SRAM Address)
A.7 Interactions of Test 5 (SRAM Bus Error)
A.8 Interactions of Test 6 (SRAM Parity)
A.9 Interactions of Test 7 (Sync Timer Gating)
A.10 Interactions of Test 8 (Gap Has Occurred Timer)
A.11 Interactions of Test 9 (Anti-Jabber Timer)
A.12 Interactions of Test 10 (Anti-Jabber Reset)
A.13 Interactions of Test 11 (Watchdog Timer)
A.14 Interactions of Tests 12-14 (Transfer Tests)
A.15 Interactions of Test 15 (Restarter)

INTRODUCTION
Section 1

This section introduces you to the characteristics and uses of PMMTS. It also provides a list of related publications.

1.1 WHAT PMMTS IS AND HOW IT WORKS

The Process Manager Module Test System (PMMTS) is a collection of test programs that work under the Process Manager Test Executive (PMEX) and are used to verify the correct operation of TDC 3000X Process Manager Modules (APMMs and PMMs). Each test program is concerned with a specific Advanced Process Manager or Process Manager Module subsystem. Table 1-1 lists the PMMTS test programs by the 4-character names accepted by PMEX.

All references to PMM in this document apply to the Advanced Process Manager Module (APMM) as well.

Table 1-1 — PMMTS Test Program Names

CCPU	—	Common CPU Tests
GMEM	—	Global Memory Tests
LMEM	—	Local Memory Tests
LNKI	—	I/O Link Tests
PSOE	—	Sequence of Events Tests
RDUN	—	Control Redundancy Tests
TBCS	—	Token Bus Controller Statistics

The test programs are subdivided into individual tests that are designed to verify correct operation of specific functions within the subsystem. You control the operation of PMMTS by keyboard entry of commands that specify the hardware to be tested and the number and sequence of tests to be run.

1.1.1 PMMTS Packaging

PMMTS test programs are packaged with PMEX on either floppy diskette number 51150700 or cartridge disk 51152049. Follow the loading and test system setup instructions in the *Process Manager Test Executive* manual.

1.1.2 Test Program Execution

As defined in the individual test writeups, some PMMTS tests run in only one of the two primary PMM processor board types (COMM and CTRL), while other tests can run in either (or both) processor type(s). The five defined processor testing combinations are:

COMM—The test program runs in only the Communication processor type.
CTRL—The test program runs in only the Control processor type.
Either—The test program runs in either processor type and can run simultaneously in both processors within a node.
COMM+ —The test program runs in only the Communication processor type, but requires interaction with the Control Processor in the same node.
Either+ —The test program runs in either processor type (or both simultaneously), but requires interaction with the other processor type in the same node.

For those tests that require interaction between COMM and CTRL, only the execution processor is specified; a "server" program in the other processor automatically responds.

Normally, a Process Manager Module contains both processor types, but the tests indicated to be "COMM" or "Either" (not "COMM+" or "Either+") can run in a PMM with only a COMM processor.

Simple single-node commands to execute any individual test in an inappropriate processor type are rejected and a error message is output at the TOCS. Similar errors found in a test list (assuming at least one test in the list is acceptable) are ignored.

1.2 USES OF PMMTS

PMMTS is useful during system installation and initial hardware validation.

1.3 REFERENCES

The following are other TDC 3000X test program publications that will be of use during installation and maintenance of TDC 3000X Process Manager Modules.

Publication Title	Publication Number	Binder Title	Binder Number
Test System Executive	SW13-510	LCN Service - 3	3060-3
Process Manager Test Executive (PMEX)	PM13-520	PM/APM/HPM Service - 2	3061-2
Process Manager Test System (PMTS)	PM13-510	PM/APM/HPM Service - 2	3061-2

COMMON CPU TESTS (CCPU)
Section 2

2.1 COMMON CPU TEST ELEMENTS

These tests provide verification of the nonmaskable interrupts, local and global timers, register contents, and the 68 k murder circuitry on the COMM and CTRL boards. It does not verify EPROM hardware.

Previous versions of LMEM did not check the Tracked Ram parts of Local Memory. LMEM now checks the Tracked Ram, so the CCPU and RDUN test must share the Tracked Ram. If LMEM halts because the error limit is reached, LMEM holds on to the Tracked Ram and this can cause errors in CCPU and RDUN.

2.2 USE INFORMATION

Tests 4 and 6 are "fatal" tests and require system reset/reload. They also are "Utility" tests and therefore must be the only test running in the node.

2.3 SETUP PARAMETERS

2.3.1 CCPU-Specific Parameters

none

2.3.2 General Parameters

TESTS	Allowed values are 1 through 6
ERROR_LIMIT	Allowed values are 1 through 32767 or "-"
PASS_LIMIT	Allowed values are 1 through 32767 or "-"
REPORT	Allowed values are PASSNUMBER, TESTNUMBER, LOG, SUSPENDED
INHIBIT	Allowed values are PASSNUMBER, TESTNUMBER, LOG, SUSPENDED
SCALE	Allowed values are 0 through 999
MINUTES_LIMIT	Allowed values are 0 through 32767 or "-"
ABBREVIATION	Allowed values are 0 through 4
REMOVED	Allowed value is YES

2.4 PRESET PARAMETER VALUES FOR CCPU

TESTS = 1-6
ERROR_LIMIT = 5
PASS_LIMIT = -
REPORT = LOG
INHIBIT = PASSNUMBER,TESTNUMBER,SUSPENDED
SCALE = 100
MINUTES_LIMIT = -
ABBREVIATION = 4

2.5 MODE CHARACTERISTICS FOR CCPU

System Mode	Programs Running	Tests Running	Delay Between Tests
Subsystem	Only one	One or more	None
Module	One or more	One or more	Random

2.6 INDIVIDUAL CCPU TESTS

(Processor) Run Modes

Test 01—Powerdown, GMEM & LMEM Parity (Either) SUB, MOD

a) Reads status register (COMM=$600002, CTRL=$60001C) and checks for no regulator power down nonmaskable interrupt.
b) Writes to Global memory at $83FFFE, then reads from the same location to cause a Global parity error. Reads status register (COMM=$600002, CTRL=$60001C) to check for the Global parity error nonmaskable interrupt. Writes a zero to $600001 in COMM or CTRL to acknowledge the interrupt and then writes a one to the same location to enable the parity checkers.
c) Writes to the EPROM location $000002, then reads from the same location to cause a Local RAM/EPROM parity error. Reads status register (COMM=$600002, CTRL=$60001C) to check for the RAM/EPROM parity error nonmaskable interrupt. Writes a zero to $600001 in COMM or CTRL to acknowledge the interrupt, then writes a one to the same location to enable the parity checkers.
d) Reads from EPROM starting at location 0000 and reads through location FFFF checking for parity and calculates a checksum upon completion. Either error is reported.

Test 02—Shared RAM Parity Error (COMM) SUB, MOD

Writes to Shared memory at $50FFFD, then reads from the same location to cause a Shared RAM parity error. Reads status register (COMM=$600002, CTRL=$60001A) to check for the Shared RAM parity error nonmaskable interrupt. Writes a zero to $600001 in COMM or CTRL to acknowledge the interrupt, then writes a one to the same location to enable the parity checkers.

Test 03—7/8 Memory Full and Bus Lock Error (CTRL) SUB

This test will not run if the redundancy daughter card is not present.

a) Writes zero to CTRL memory at $600008 to get out of the bus lock mode. Writes zero to $60000A to reset daughter card counter, then writes a one to it so it will store the correct eavesdrop image. Writes $FF to CTRL memory starting at first address of Tracked Ram for (7*32 k/8)+1 times to cause the 7/8 daughter memory full nonmaskable interrupt. Reads the CTRL status register at $60001C and checks for the interrupt. Writes a zero to $600001 to acknowledge the interrupt, then writes a one to the same location to enable the parity checkers.

b) Writes one to CTRL memory at $600008 to set daughter card into bus lock mode. Writes $FF to CTRL memory at first address of Tracked Ram to cause daughter access (bus lock error) nonmaskable interrupt. Reads CTRL status register at $60001A and checks for the interrupt. Writes a zero to $600001 to acknowledge the interrupt, then writes a one to the same location to enable the parity checkers.

Test 04—Watchdog Timer (Either+) SUB

COMM: Disables the CTRL 68 k watchdog timer, then reads status register at $600000, to check if the CTRL 68 k watchdog timer has expired. Writes a zero to $600001 to acknowledge the interrupt, then writes a one to the same location to enable the parity checkers.

CTRL: Disables the COMM 68 k watchdog timer, then reads status register at $60001C, to check if the COMM 68 k watchdog timer has expired. Writes a zero to $600001 to acknowledge the interrupt, then writes a one to the same location to enable the parity checkers.

This is a "fatal" test; requiring system reset/reload after execution.

Test 05—Local Timer and Global Timer (Either) SUB, MOD, EXER

COMM: Writes $94 (Timer 2, Mode 2, Low write) to COMM memory at $400007; writes a count of $OF to Local timer 2 (COMM memory at $400005); writes a $80 (read-back count mode) to COMM memory at $400007; reads COMM memory at $400005 (Global timer 2) for range count check.

CTRL:Writes $14 (Timer 0, Mode 2, Low write) to CTRL memory at $C00007; writes $54 (Timer 1, Mode 2, Low write) to CTRL memory at $C00007; writes a count of $0F to CTRL memory at $C00001 (Global timer 0); writes a count of $0F to CTRL memory at $C00003 (Global timer 1); writes $00 (read-back count mode) to CTRL memory at $C00007; writes $40 (read-back count mode) to CTRL memory at $C00007; reads CTRL memory at $C00001 (Global timer 0) for range count check; reads CTRL memory at $C00003 (Global timer 1) for range count check.

Note that Local timers 0 and 1 and Global timer 2 are not checked by this test.

Test 06—Murder Circuitry (Either+) SUB

COMM: Writes a one to CTRL memory at $600005 to enable the murder circuitry, then writes a one to EPROM CTRL memory at 0 to murder the CTRL processor. Reads COMM memory at $600000 and checks for the CTRL 68 k Watchdog timer expired nonmaskable interrupt. Acknowledges the interrupt by writing a 0 to COMM memory at $600001; then writes a one to the same address to enable the parity checkers.

CTRL: Writes a one to COMM memory at $600005 to enable the murder circuitry, then writes a one to EPROM COMM memory at $000000 to murder the CTRL processor. Reads CTRL memory at $60001C and checks for the COMM 68 k Watchdog timer expired nonmaskable interrupt. Acknowledges the interrupt by writing a 0 to CTRL memory at $600001; then writes a one to the same address to enable the parity checkers.

This is a "fatal" test; requiring system reset/reload after execution.

GLOBAL MEMORY TESTS (GMEM)
Section 3

3.1 GLOBAL MEMORY TEST ELEMENTS

These tests provide verification of the SRAM Global memory board.

3.2 USE INFORMATION

The COMM card and the CTRL card can both be running tests in the global memory and will arbitrate for use of Global memory areas.

3.3 SETUP PARAMETERS

3.3.1 GMEM-Specific Parameters

LOOP_CT_T5 Allowed values are 0 through 32767. This value defines the number of times the Test 5 loop is executed per pass.

LO_ADRS_T5 This is a hexadecimal address value. This value defines the address "read from" by Test 5.

HI_ADRS_T5 This is a hexadecimal address value. This value defines the address "written to and read from" by Test 5. IMPORTANT: Do not change this value indiscriminately; it can interfere with program operation.

3.3.2 General Parameters

TESTS	Allowed values are 1 through 5
ERROR_LIMIT	Allowed values are 1 through 32767 or "-"
PASS_LIMIT	Allowed values are 1 through 32767 or "-"
REPORT	Allowed values are PASSNUMBER, TESTNUMBER, LOG, SUSPENDED
INHIBIT	Allowed values are PASSNUMBER, TESTNUMBER, LOG, SUSPENDED
SCALE	Allowed values are 0 through 999
MINUTES_LIMIT	Allowed values are 0 through 32767 or "-"
ABBREVIATION	Allowed values are 0 through 4

3.4 PRESET PARAMETER VALUES FOR GMEM

LOOP_CT_T5 = 32767
LO_ADRS_T5 = $800000
HI_ADRS_T5 = $83FFF0
TESTS = 1-4
ERROR_LIMIT = 5
PASS_LIMIT = -
REPORT = LOG
INHIBIT = PASSNUMBER,TESTNUMBER,SUSPENDED
SCALE = 100
MINUTES_LIMIT = -
ABBREVIATION = 4

3.5 MODE CHARACTERISTICS FOR GMEM

System Mode	Programs Running	Tests Running	Delay Between Tests
Subsystem	Only one	One or more	None
Module	One or more	One or more	Random
Exerciser	One or more	Only one	Random

3.6 INDIVIDUAL GLOBAL MEMORY TESTS

(Processor) Run Modes

Test 01—Memory Functions Test (Either) SUB, MOD

One word in each Global memory chip is selected as a test word and the functions of chip select, data access, parity, and the TAS operation are performed on the selected word in each chip.

a) The chip select part of the test writes, then reads the test patterns $0000 and $FFFF.
b) The data access part of the test performs upper and lower data byte writes and reads and word reads, using the data patterns $AA and $55.
c) The parity part of the test performs word/byte writes and reads, using the data patterns $AA55, $8C55, $468C, $AA23, $55AA, $0000, and $0123. These patterns are written and read, using valid parity, then are written and read with invalid parity (both word and byte parity are tested). When invalid parity is tested, a local processor status word (COMM=$600002, CTRL=$60001C) is used to indicate valid parity (bit 12=1) and invalid parity (bit 12=0).
d) The TAS instruction operation part writes test patterns to the test word, then tests the TAS instruction on the upper and lower bytes of the test word. The test patterns used are $0000, $0080, $8000, and $8080.

Test 02—Communications/Control Arbitration Test (Either+) SUB

Tests the arbitration circuitry for access to Global memory by the COMM card and the CTRL card. The test can be started in either card, but both cards must be present and loaded.

Sixteen contiguous words in Global memory are selected as the test words. A pattern of $FFFF is written to word 1 and words 4 thru 16. The COMM card repeatedly writes the pattern $AAA2 to and reads from the second word of the 16 words. The CTRL card repeatedly writes the pattern $5455 to and reads from the third word of the 16 words. Writes and reads are independently performed by each processor 10,000 times, in parallel with each other. Any address or data conflicts show up as bus errors or as invalid data patterns within the 16 data words.

Test 03—Chip Cell Test (Either) SUB, MOD

Writes test patterns to and reads from every Global memory address and data cell that is available for testing. The test patterns used are $0000, $FFFF, $AAAA, $5555, $0123, $468C, $AA23, and $8C55.

Test 04—Exerciser Test (Either+) SUB, MOD,EXER

This is a modified version of Test 2, using random data for maximum interaction. The test can be started in either card, but both cards must be present and loaded. Sixteen contiguous words in Global memory are selected as the test words. A pattern of $FFFF is written to word 1 and words 4 through 16. The COMM card repeatedly writes a randomly selected pattern to and reads from the second word of the 16 words. The CTRL card repeatedly writes a randomly selected pattern to and reads from the third word of the 16 words. Writes and reads are independently performed 1 to 255 times by each processor, in parallel with one another. Any address or data conflicts are expected to show up as bus errors or as invalid data patterns within the 16 data words.

Test 05—Address/Data Line Changes Test (Either+) SUB, MOD

This is a specialized test used to aggravate a COMM/CTRL and backplane noise condition that was manifested by unexpected vector 31, level 7 interrupts. The condition shows up in COMM and CTRL boards of hardware revision 'F' or greater, installed in backplanes of revision 'A' or 'B'. This test alternates the address and data lines to maximize the potential for noise. Test 5 uses parameters LOOP_CT_T5, LO_ADRS_T5, and HI_ADRS_T5. Caution must be exercised if you need to change HI_ADRS_T5 from the default value since this address is written to and you could destroy program instructions or test data.

Both COMM and CTRL boards are required for this test to run. The test reads from LO_ADRS_T5, writes $0000 to HI_ADRS_T5, reads from HI_ADRS_T5, writes $FFFF to HI_ADRS_T5, and reads from HI_ADRS_T5. This sequence is repeated the number of times specified by LOOP_CT_T5 each pass. See Heading 3.4 for the default values of these parameters.

Section 4

Reserved for future use.

LOCAL MEMORY TESTS (LMEM)
Section 5

5.1 LOCAL MEMORY TEST ELEMENTS

These tests provide verification of the local SRAM and byte parity bits on the COMM board and its daughter board and on the CTRL card.

Previous versions of LMEM did not check the Tracked Ram parts of Local Memory. LMEM now checks the Tracked Ram, so the CCPU and RDUN test must share the Tracked Ram. If LMEM halts because the error limit is reached, LMEM holds on to the Tracked Ram and this can cause errors in CCPU and RDUN.

5.2 USE INFORMATION

There is no limitation on running these tests simultaneously in both the COMM card and the CTRL card.

5.3 SETUP PARAMETERS

5.3.1 LMEM-Specific Parameters

LIMITED_MEMORY_CHECK Allowed values are YES or NO.

5.3.2 General Parameters

TESTS	Allowed values are 1 through 2
ERROR_LIMIT	Allowed values are 1 through 32767 or "-"
PASS_LIMIT	Allowed values are 1 through 32767 or "-"
REPORT	Allowed values are PASSNUMBER, TESTNUMBER, LOG, SUSPENDED
INHIBIT	Allowed values are PASSNUMBER, TESTNUMBER, LOG, SUSPENDED
SCALE	Allowed values are 0 through 999
MINUTES_LIMIT	Allowed values are 0 through 32767 or "-"
ABBREVIATION	Allowed values are 0 through 4

5.4 PRESET PARAMETER VALUES FOR LMEM

LIMITED_MEMORY_CHECK = NO
TESTS = 1-2
ERROR_LIMIT = 5
PASS_LIMIT = -
REPORT = LOG
INHIBIT = PASSNUMBER,TESTNUMBER,SUSPENDED
SCALE = 100
MINUTES_LIMIT = -
ABBREVIATION = 4

5.5 MODE CHARACTERISTICS FOR LMEM

System Mode	Programs Running	Tests Running	Delay Between Tests
Subsystem	Only one	One or more	None
Module	One or more	One or more	Random

5.6 INDIVIDUAL LOCAL MEMORY TESTS

(Processor) Run Modes

Test 01—Memory Functions Test (Either) SUB, MOD

One word in each local memory chip is selected as a test word and the functions of chip select, data access, and the TAS operation are performed on the selected word in each chip.

a) The chip select part of the test writes, then reads the test patterns $0000 and $FFFF.
b) The data access part of the test performs upper and lower data byte writes and reads and word reads, using the data patterns $AA and $55.
c) The TAS instruction operation part writes test patterns to the test word, then tests the TAS instruction on the upper and lower bytes of the test word. The test patterns used are $0000, $0080, $8000, and $8080.

Test 02—Chip Cell Test (Either) SUB, MOD

Writes test patterns to and reads from every local memory address and data cell that is available for testing. If LIMITED_MEMORY_CHECK is set to NO, the test patterns used are $0000, $FFFF, and a random number and its complements. If LIMITED_MEMORY is set to YES, the test patterns used are $0000 and $FFFF.

I/O LINK TESTS (LNKI)
Section 6

6.1 I/O LINK TESTS ELEMENTS

These tests verify the integrity of the PMM I/O link hardware. This includes the shared RAM, the 8031 processor and its associated circuitry, the twisted pair interface, integrity circuits (such as Anti-Jabber and Watchdog Timer), and signals between the 8031 I/O processor and the COMM and CTRL processors.

6.2 USE INFORMATION

This test program runs with either a single PMM (see parameter "PAIR_INSTALLED") or with a redundant PMM pair. The following use guidelines should be observed when testing a redundant pair.

- Module mode tests cannot be run in a PMM node while its redundant "partner" PMM node is running a subsystem mode test.
- While running the subsystem-only tests (1-4, 6, and 8 through 11) in a PMM, its partner PMM node—if present—should be running test 15 only. Because test 15 reserves and holds Level 3 (link) and Level 7(NMI) interrupts, no other tests can be running on that node. This is necessary because each of these tests "kills" one or the other of the 8031 I/O link processors and this could interfere with other tests running in the partner. Running test 15 in the partner while tests 10 and 11 are running also ensures that the partner's 8031 I/O link processor is restarted.
- Transfer tests (12, 13 and 14) should be run in both the primary and redundant PMM for maximum network activity.
- An example use scenario for a redundant PMM is, start in module mode on both nodes and run all tests. Then run test 15 in one node and subsystem tests in the other node. Reverse the roles of the nodes and again run the subsystem tests. Run the Exerciser test in both nodes.

All timer tests (8, 9, and 11) use microseconds and incorporate sufficient range for specified accuracy of the test. Error messages show actual, minimum, and maximum times in microseconds.

Tests 8 through 14 use I/O link transactions and thus require a redundant PMM pair.

NOTE

The Exerciser test (Test 14) will cause the node to be dropped if PAIR_INSTALLED is set to YES and there is not a pair installed (i.e., a partner).

6.3 SETUP PARAMETERS

6.3.1 LNKI-Specific Parameters

PAIR_INSTALLED Allowed values are YES or NO

When there is **not** a redundant "partner" PMM, you should set this parameter's value to "NO" to eliminate meaningless alarms and to prevent tests 8 through 14 from running.

6.3.2 General Parameters

TESTS	Allowed values are 1 through 15
ERROR_LIMIT	Allowed values are 1 through 32767 or "-"
PASS_LIMIT	Allowed values are 1 through 32767 or "-"
REPORT	Allowed values are PASSNUMBER, TESTNUMBER, LOG, SUSPENDED
INHIBIT	Allowed values are PASSNUMBER, TESTNUMBER, LOG, SUSPENDED
SCALE	Allowed values are 0 through 999
MINUTES_LIMIT	Allowed values are 0 through 32767 or "-"
ABBREVIATION	Allowed values are 0 through 4

6.4 PRESET PARAMETER VALUES FOR LNKI

PAIR_INSTALLED = YES
TESTS = 1-15
ERROR_LIMIT = 5
PASS_LIMIT = -
REPORT = LOG
INHIBIT = PASSNUMBER,TESTNUMBER,SUSPENDED
SCALE = 100
MINUTES_LIMIT = -
ABBREVIATION = 4

6.5 MODE CHARACTERISTICS FOR LNKI

System Mode	Programs Running	Tests Running	Delay Between Tests
Subsystem	Only one	One or more	None
Module	One or more	One or more	Random
Exerciser	One or more	Only one	Random

NOTE

The hardware interactions tested by LNKI are summarized in Appendix A of this document.

6.6 INDIVIDUAL I/O LINK TESTS

(Processor) Run Modes

Test 01—Reset and Self-Test (COMM) SUB
Resets the I/O link from the Communications board and checks the following areas: ability of the Communications board to reset the I/O link; ability of I/O link to run self-test at operating temperature; ability of I/O link to be reset normally without causing unexpected interrupts to the Communications and Control boards.

Test 02—Shared RAM Arbitration (Either+) SUB
Tests the shared bus arbitration logic.

Test 03—SRAM Data Pattern (Either) SUB
Tests the data lines and integrity of the Shared RAM.

Test 04—SRAM Address (Either) SUB
Tests address logic of the Shared RAM.

Test 05—SRAM Bus Error (Either) SUB, MOD
Tests the ability of the Shared RAM to create a bus error as the result of an access to an even memory location.

Test 06—SRAM Parity (COMM) SUB
Verifies Shared RAM parity checking by 8031.

Test 07—Sync Timer Gating (COMM) SUB, MOD
Verifies the 8031's ability to gate (stop) the I/O Link Timer

Test 08—Gap Has Occurred Timer (COMM) SUB
Verifies Gap Has Occurred (GHO) detection and timer logic of the I/O link on the redundant PMM. This test requires a redundant pair of PMMs with Test 15 running on the partner node.

Test 09—Anti-Jabber Timer (COMM) SUB
Verifies the Anti-Jabber Timer (AJT) detection, timer, and driver disable logic of the I/O link on the redundant PMM. This test requires a redundant pair of PMMs with Test 15 running on the partner node.

Test 10—Anti-Jabber Reset (COMM) SUB
Verifies the Anti-Jabber Timer's ability to reset the 8031 (if the 8031 requests Jabber reset and drive together) on the redundant PMM. This test requires a redundant pair of PMMs with Test 15 running on the partner node.

Test 11—Watchdog Timer (COMM) SUB
Verifies Watchdog Timer logic of the I/O link on the redundant PMM. This test requires a redundant pair of PMMs with Test 15 running on the partner node.

Test 12—Transfer (Either) SUB, MOD

Checks the ability of the I/O link to perform data transfers to the redundant twin PMM's 8031 data base, using polled FIFOs and no interrupts. When run in both nodes, one of the nodes builds the token ring and token passing occurs automatically.

Test 13—Transfer and Interrupt (Either) SUB, MOD

Checks the ability of the I/O link to perform data transfers to the redundant twin PMM's 8031 data base and to interrupt the appropriate 68 k processor. When run in both nodes, one of the nodes builds the token ring and token passing occurs automatically.

Test 14—Exerciser (Either) EXER

Performs random data transfer tests to/from the redundant PMMs 8031 data base. The transfers are of random size and content.

Test 15—Restarter (COMM) SUB

Restarts the 8031 I/O link processor in the partner node after it is "killed" by test 10 or test 11. This test should be running in the partner node while any of the Subsystem mode tests are running (to prevent the possibility of test conflicts between the two nodes). This is a "Utility" test and therefore must be the only test running in the node.

CONTROL REDUNDANCY TESTS (RDUN)
Section 7

7.1 CONTROL REDUNDANCY TESTS ELEMENTS

These tests verify integrity of the redundancy hardware in a PMM/APMM and the Private Path TBC in a PMM/APMM pair. A pair consists of two PMM/APMMs physically linked as a primary and a secondary, which may or may not be in the same backplane. The Private Path TBC testing performs random data transfers from one redundancy daughter card to the other.

Previous versions of LMEM did not check the Tracked Ram parts of Local Memory. LMEM now checks the Tracked Ram, so the CCPU and RDUN test must share the Tracked Ram. If LMEM halts because the error limit is reached, LMEM holds on to the Tracked Ram and this may cause errors in CCPU and RDUN.

7.2 USE INFORMATION

When a chassis contains a redundant PMM/APMM pair, they are referred to as the "left-side" and "right-side" nodes (as you view them). In other words, the "left-side" PMM/APMM is located in slots 1 through 5, and the "right-side" PMM/APMM is located in slots 6 through 10. In normal operation, either "partner" in a PMM/APMM pair can be the primary. In these tests, however, the left-side PMM/APMM is treated as the primary (i.e., the tests are run from the left-side PMM/APMM with a "server" responding from the right side).

When a redundant pair does not reside in the same chassis, they are referred to as upper and lower (as you view them). Both PMM/APMMs occupy slots 1 through 5. A Redundancy driver card may occupy slot 5, and must be installed with the associated redundancy cable linking the upper and lower PMM/APMMs.

MMTS must be loaded and running in the CTRL processors of both the left-side and the right-side or upper and lower PMM/APMMs, respectively, in order to run Private Path TBC test (Test 1). Test 1 must then be run on both PMM/APMMs in the pair at the same time.

The Partner_Installed parameter must be set to “no” in all units (PMMs/APMMs) that are nonredundant, and set to "yes" in all units (PMMs/APMMs) that are redundant, whether they are upper/lower or right/left to successfully run Test 1.

In normal operation, either partner in a PMM/APMM pair can be the primary.

RDUN tests compete with LMEM and CCPU tests for the tracked RAM located in Local Memory. If RDUN is unable to reserve the memory because LMEM or CPU are not releasing the memory, RDUN will abort. In a case where the operator receives a message indicating the resource could not be reserved, LMEM and CCPU should be checked (e.g. LMEM is halted on error and is not releasing memory).

7.3 SETUP PARAMETERS

7.3.1 RDUN-Specific Parameters

Partner_Installed	Allowed values are YES or NO When set to a NO, Test 1 will not run. When set to a YES, Test 1 will run, automatically looking for the proper partner.

7.3.2 General Parameters

TESTS	Allowed values are 1 through 5
ERROR_LIMIT	Allowed values are 1 through 32767 or "-"
PASS_LIMIT	Allowed values are 1 through 32767 or "-"
REPORT	Allowed values are PASSNUMBER, TESTNUMBER, LOG, SUSPENDED
INHIBIT	Allowed values are PASSNUMBER, TESTNUMBER, LOG, SUSPENDED
SCALE	Allowed values are 0 through 999
MINUTES_LIMIT	Allowed values are 0 through 32767 or "-"
ABBREVIATION	Allowed values are 0 through 4

7.4 PRESET PARAMETER VALUES FOR RDUN

PARTNER_INSTALLED = NO
TESTS = 1–5
ERROR_LIMIT = 5
PASS_LIMIT = -
INHIBIT = PASSNUMBER,TESTNUMBER,SUSPENDED
SCALE = 100
MINUTES_LIMIT = -
ABBREVIATION = 4

7.5 MODE CHARACTERISTICS FOR RDUN

System Mode	Programs Running	Tests Running	Delay Between Tests
Subsystem	Only one	One or more	None
Module	One or more	One or more	Random
Exerciser	One or more	Only one	Random

7.6 INDIVIDUAL RDUN TESTS (Processor) Run Modes

Test 01—Private Path/Token Bus Controller Test (CTRL) SUB, MOD

Both PMM/APMMs reserve resources (Tracked RAM and Daughter Card Memory). Then a primary and a secondary PMM/APMM are established based on the node number. The primary is the lowest node number. The PMM/APMMs establish communications by way of mail at this point. Once this is done, the secondary PMM/APMM reserves up to 128 words of heap and fills it with random data. This data is then transmitted from heap 4 times to the primary PMM/APMM and stored in the tracking memory by the Private Path TBC. When the primary PMM/APMM is told by mail that the data has been transmitted, it then transmits the data back to the secondary PMM/APMM. The secondary PMM/APMM then compares the 4 receive buffers (one at a time) to the 128 words of heap. If a miscompare is found, an error is generated. The two PMM/APMMs then reverse roles by mail and now the secondary becomes primary and the primary becomes the secondary. The test is run again and terminates at the end of this pass.

It should be noted that it is virtually impossible to tell if the receiver or the transmitter has failed on any miscompare. All that can be said is that the data is not being transmitted or received on the Private Path TBC.

Test 02—Tracked RAM Test (CTRL) SUB, MOD, EXER

This test requires no partner PMM/APMM. The test first reserves the resources (Tracked RAM and Daughter Card Memory) required to test tracked RAM. The Daughter Card Counter is reset and tracking is enabled. One byte of tracked memory is written and the daughter card is checked for the proper update of the tracked address, data, and data strobe. The daughter card counter is reset and the next byte of the first word is written into and verified.

After the first two bytes of memory are verified and memory strobes have been verified as changing correctly, the test starts word writes to THE FIRST WORD OF EACH 16K word block and increments through the rest of tracked RAM memory. After each word write, the daughter card is verified for the correct tracked address, data, and data strobe.

After verification, the daughter card counter is reset. The counter is reset after every write so that tracking can be verified in the first word of each bank of memory on the daughter card. After 16K words of tracked memory have been checked, the resources are released so other tests may reserve them and the test attempts to reserve the resource again. Once the resources have been reserved again, the test continues from where it left off and verifies the next 16K words. This is done until all of tracked RAM is tested.

During the first pass of the test, the data pattern of 5555 is used. The second pass uses AAAA. These patterns alternate on subsequent passes.

Test 03—7/8's Full Interrupt Test (CTRL) SUB, MOD

This test requires no partner PMM/APMM. The test reserves the required resources (Tracked RAM and Daughter Card RAM). The test then writes to tracked RAM 28671 times. It then reserves the level 7 interrupt and writes to tracked RAM one time. The interrupt should be generated on this write (28672). If for some reason the interrupt is generated prematurely, the system will receive an unexpected interrupt. The test will look for the interrupt at the proper time and report if there is no interrupt or if the interrupt occurred at the wrong time. It also checks the daughter card counter for the proper count of 28672. Any error is reported.

Test 04—Tracking RAM Test (CTRL) SUB, MOD

This test requires no partner PMM/APMM. The test first reserves the resources (Tracked RAM and Daughter Card Memory) required to test tracking RAM. The Daughter Card Counter is reset and tracking is disabled Tracking memory is then filled with random data in each location. This memory is then read back starting at the beginning and verified to have the correct data. If an error is found, then an error message is displayed reporting the Address, WAS data, and SHOULD BE data.

Test 05—Tracking RAM Scrub (CTRL) SUB, MOD

This test requires no partner PMM/APMM. The test first reserves the resources (Tracked RAM and Daughter Card Memory) required to test tracking RAM. The Daughter Card Counter is reset and tracking is disabled Tracking memory is then read in 4K blocks starting at the beginning until all of memory has been read. If any parity errors are encountered they are handled by the Unexpected Interrupt handler in PMEX.

TOKEN BUS CONTROLLER STATISTICS (TBCS)
Section 8

8.1 TOKEN BUS CONTROLLER TEST ELEMENTS

This test program monitors the UCN event statistics kept by the COMM boards in the PM and APMM modules on a Universal Control Network.

8.2 USE INFORMATION

TBCS consists of a single test that monitors the set of statistics kept by the Token Bus Controller firmware in the COMM boards of PM and APM modules as events occur on the common UCN. These statistics are counters that start out at zero when TBCS is run, and then accumulate UCN events as seen by the individual PM and APM nodes.

The Rate parameter specifies the frequency in seconds that statistics are to be sampled. The default value of Rate is five seconds.

The LOOPS_PER_PASS parameter specifies how many samples comprise a pass, and thus in conjunction with the value of RATE determines the duration of each pass. The default value of LOOPS_PER_PASS is one. Thus, with default five second RATE parameter, each pass defaults to five seconds long.

The PRINT_ALL_STATISTICS parameter allows the reporting of all nonzero statistics at the end of each pass. Its preset value is NO. Note that if set to YES with the default RATE and LOOPS_PER_PASS values, all nonzero statistics will be reported every five seconds. Therefore, to reduce the number of reports, it is suggested that a large value be entered for LOOPS_PER_PASS and/or RATE, such that the duration of each pass is long enough to minimize these reports.

An alternative means of seeing the statistics is the STATUS command. This shows all of the statistics in three columns, including those that are zero. Unlike Program-Specific Parameters of the other test programs that can be commanded by name to see individual values, these TBCS statistic names are in lower case letters (for easier reading) and cannot be commanded by name.

The CLEAR_ALL_STATISTCS parameter can be used at any time to reset all statistic counts and the ERROR COUNT to zero. Whenever TBCS is being run, this parameter is automatically set to YES removing whatever value the operator entered before commanding the RUN. After clearing the counts, the CLEAR_ALL_STATISTICS value is automatically set to NO so that the counts are not continuously cleared. The operator may specify a YES anytime after that for a fresh start.

Like other test programs, alarming stops after the ERROR_LIMIT parameter value is reached, and “halted after nn errors” follows the final alarm. However, TBCS continues to run and sample new statistics. Thus, subsequent STATUS requests will report the latest statistical values and PRINT_ALL_STATISTICS YES will report the latest nonzero statistical values, even after alarming has stopped because of reaching the ERROR_LIMIT. Alarming will resume if ERROR_LIMIT is increased or if CLEAR_ALL_STATISTICS YES is commanded.

Token Bus Controller firmware returns total cable swaps in the counter “Number of network cable swaps,” including operator, periodic, and fault-induced swaps. However, operator requested and periodic cable swaps are normal and should not be alarmed. The TBCS test program attempts to subtract these from the firmware count, leaving only fault-induced cable swaps reported in this counter.

8.3 SETUP PARAMETERS

8.3.1 TBCS-Specific Parameters

CLEAR_ALL_STATISTICS Allowed values are: YES, NO. When set to YES, resets all statistics counters and ERROR_COUNT to zero and then automatically sets the parameter value back to NO.

PRINT_ALL_STATISTICS Allowed values are: YES, NO. If YES, initiates the display of all nonzero statistics at the end of each pass .

RATE Allowed values are 1 to 300. This specifies the pause in seconds between sampling of the TBC statistics, i.e., the number of seconds between test loops.

LOOPS_PER_PASS Allowed values are 1 to 1000. This specifies the number of statistic samples that comprise one test pass, i.e., the number of test loops per test pass.

8.3.2 General Parameters

TESTS	Allowed value is 1 or a series of ones; for example: 1,1,1,1,1
ERROR_LIMIT	Allowed values are 1 through 32767 or "-"
PASS_LIMIT	Allowed values are 1 through 32767 or "-"
REPORT	Allowed values are PASSNUMBER, TESTNUMBER, LOG, SUSPENDED
INHIBIT	Allowed values are PASSNUMBER, TESTNUMBER, LOG, SUSPENDED
SCALE	Allowed values are 0 through 999. This is not used by TBCS.
MINUTES_LIMIT	Allowed values are 0 through 32767 or "-"
ABBREVIATION	Allowed values are 0 through 4

8.4 PRESET PARAMETER VALUES FOR TBCS

CLEAR_ALL_STATISTICS = NO (automatically set to YES when TBCS starts running)
PRINT_ALL_STATISTICS = NO
RATE = 5
LOOPS_PER_PASS = 1
TESTS = 1
ERROR_LIMIT = 5
PASS_LIMIT = -
REPORT = LOG
INHIBIT = PASSNUMBER,TESTNUMBER,SUSPENDED
SCALE = 0
MINUTES_LIMIT = -
ABBREVIATION = 4

8.5 MODE CHARACTERISTICS FOR TBCS

System Mode	Programs Running	Tests Running	Delay Between Tests
Subsystem	Only one	Only one	None
Module	One or more	Only one	None
Exerciser	One or more	Only one	None

8.6 INDIVIDUAL TOKEN BUS CONTROLLER TESTS (Processor) Run Modes

Test 01—Monitor Token Bus Statistics (COMM) SUB, MOD EXER

On each test loop, this test samples each of the 22 Token Bus statistics counters and reports all changes; then delays for the number of seconds specified by the RATE parameter. It continues for the number of test loops specified by the LOOPS_PER_PASS parameter. Then, if the value of PRINT_ALL_STATISTICS = YES, the values of all nonzero counters are reported at the completion of each pass.

PROCESS SEQUENCE OF EVENTS (PSOE)
Section 9

9.1 PROCESS SEQUENCE OF EVENTS TEST ELEMENTS

These tests provide verification of Mode Control Register, Synch Counter/Latch, Synch Counter Carry, Master Clock, and Message Decoder functions.

These tests are run on a APMM COMM board only.

9.2 USE INFORMATION

A pass of this test consists of one execution of each of the five tests.

9.3 SETUP PARAMETERS

9.3.1 PSOE-Specific Parameters

FIRST_RANDOM Preset to 0 and will be used as the Seed for the next test that uses a random number.

LOOP_LIMIT Preset to 16 and determines the number of random test loops to be executed during each pass.

9.3.2 General Parameters

TESTS	Allowed values are 1 through 5
ERROR_LIMIT	Allowed values are 1 through 32767 or "-"
PASS_LIMIT	Allowed values are 1 through 32767 or "-"
REPORT	Allowed values are PASSNUMBER, TESTNUMBER, LOG
INHIBIT	Allowed values are PASSNUMBER, TESTNUMBER, LOG
SCALE	Allowed values are 0 through 999
MINUTES_LIMIT	Allowed values are 0 through 32767 or "-"
ABBREVIATION	Allowed values are 0 through 4

9.4 PRESET PARAMETER VALUES FOR PSOE

TESTS = 1–5
ERROR_LIMIT = -
PASS_LIMIT = -
REPORT = LOG
INHIBIT = PASSNUMBER,TESTNUMBER, SUSPENDED
SCALE = 100
MINUTES_LIMIT = -
ABBREVIATION = 4

If Pass_Limit is “-” and Test S is run in Subsystem Mode, the Pass_Limit will be reset to 5.

9.5 MODE CHARACTERISTICS FOR PSOE

System Mode	Programs Running	Tests Running	Delay Between Tests
Subsystem	Only one	One or more	None
Module	One or more	One or more	Random
Exerciser	One or more	Only one	Random

9.6 INDIVIDUAL PROCESS SEQUENCE OF EVENTS TESTS (System Modes)

Test 01 Mode Control Register Test SUB, MOD

Various control commands are written to the Time Synch gate array command addresses and are read from the Time Synch gate array Status Register to verify that the mode changes have been made.

a) Set the mode to transmit.
b) Set the mode to receive (mode is left in receive).
c) Write the Reference Crystal select command address.
d) Set Reference Crystal source to 5 MHz (leave at this setting).
e) Write Clear All Conditions command address.

Test 02 Synch Counter/Latch Test SUB, MOD

Verifies the functionality of Synch Counter Diagnostic command control bits, UCN Latch and the Aux Latch.

a) An initial check is made of the Time Synch gate array status register for unexpected changes while the test was idle.
b) Clear the synch counter to the Synch Counter Diagnostic command address of the Time Synch gate array, then check the Time Synch gate array status register.
c) Load different (loop limit) values in the Synch Counter.
d) Clear the UCN Latch Synched bit, UCN Latch Error bit, Aux Latch Synched bit and Aux Latch Error bit and check the Time Synch gate array Status Register for the proper value.
e) Command the Time Synch gate array to latch the Synch Counter with the UCN Synch Latch and the Time Synch gate array Status Register, then check the UCN Latch for proper value. This test is run for the Loop-Limit random values of the Synch Counter.
f) Command the Time Synch gate array to latch the Synch Counter with the Aux Synch Latch and the Time Synch gate array Status Register, then check the Aux Latch for proper value. This test is run for the Loop-Limit random values of the Synch Counter.
g) Latch the UCN Latch and the Aux Latch a second time to test the latch error bits.
h) The test will write to the Clear All Conditions command address of the Time Synch gate array and the Time Synch gate array Status Register will be checked for the proper values of all bits.

Test 03 Synch Counter Carry Test SUB, MOD

Verifies the functionality of the Synch Counter carry bits and the Synch Counter Diagnostic command control bits.

a) An initial check is made of the Time Synch gate array status register for unexpected changes while the test was idle.
b) Preload the Synch Counter with all Fs.
c) Command the Time Synch gate array to latch the Synch Counter with the Aux Synch Latch.
d) Check the Time Synch gate array Aux Synch Latch and the Status Register for proper value. The Aux Synch Latch should contain all Fs.
e) The AUX Latch is cleared and the associated bit in the Time Synch gate array Status Register is checked for proper value.
f) Write to the synch Counter Diagnostic command address of the Time Synch gate array to increment the least significant nibble of the Synch Counter with the carry flag enabled.
g) Command the Time Synch gate array to latch the Synch Counter with the Aux Synch Latch. The Time Synch gate array Aux Synch Latch and the Status Register are checked for proper values. The Aux Synch Latch should contain all zero's.
h) Write to the Clear All Conditions command address of the Time Synch gate array and the Time Synch gate array Status Register is checked for the proper values of all bits.

Test 04 Master Clock Test SUB, MOD

Compares the Time Synch master clock to the PMEX Mil_Sec hardware clock.

a) Make an initial check of the Time Synch gate array Status Register for any unexpected changes while idle.
b) Set the Transmit/Receive mode to receive and the Reference Crystal source to 5 Mhz. Check both modes in the Time Synch gate array Status Register.
c) Write to Clear All Conditions command address of the Time Synch gate array. Check the Time Synch gate array Status Register for proper values of all bits.
d) Pause 50 milliseconds to synchronize the test to the PMEX Mil_Sec hardware clock.
e) Command the Time Synch gate array to latch the free-running Synch Counter with the UCN Synch Latch and the Time Synch gate array Status Register is checked for the proper values.
f) Pause for one second timed by PMEX_Sec clock.
g) Command the Time Synch gate array to latch the free-running Synch Counter with the UCN Synch Latch and the Time Synch gate array Status Register is checked for the proper values.
h) Test the two synches for a difference in counts greater than 40,000 (25 microsecond Master Clock counts plus or minus a test mode dependent tolerance).
i) Write to Clear all Conditions command address of the Time Synch gate array and check the Time Synch gate array Status Register for proper values.

Test 05 Message Decoder Test SUB

Verify the operation of the UCN Decoder logic in the Time Synch gate array by sending a TBC transmit loopback frame on the UCN.

a) Check the Time Synch gate array Status Register for unexpected changes while idle.
b) Set the Transmit/Receive mode to receive and the Reference Crystal source to 5 Mhz and check the results of both in the Time Synch gate array Status Register.
c) For each bit in the Message Decoder Register, Test 5 will perform the following :
    1) Execute a Time Synch Message Sequence of actions which will set the correct bit and verify that only that bit is set.
    2) Execute a Time Synch Message Sequence of actions which will set one other random bit in the register and verify that both bits are set.
    3) Execute a Time Synch Message Sequence of actions which will reset the bit set in 1) above, but sets the bit set in 2) above, and verify that the correct bit is set.

9.7 ALARM MESSAGES AND DESCRIPTIONS

If the Time Synch gate array Status Register contents do not match the expected values, an alarm is displayed.

DD-MMM-YYYY HH:MM:SS NMA nn COMM nn Alarm TEST#nn
WHILE EXECUTING SOME TEST ACTION - TIME SYNCH STATUS REGISTER
WAS – $XXXX
EXPECTED – $YYYY
MASK – $ZZZZ

If one or both of the Time Synch gate array Synch Latches' contents do not match the expected values, alarms will be displayed.

DD-MMM-YYYY HH:MM:SS NMA nn COMM nn Alarm TEST#nn
WHILE LATCHING TYPT SYNCH LATCH - TYPE SYNCH LATCH
WAS – $XXXX XXXX
EXPECTED – $YYYY YYYY
MASK – $ZZZZ ZZZZ

Where TYPT is the type of Synch Counter/Latch under test and TYPE is the Synch Latch exhibiting an error. The possible values to each Synch Counter/Latch TYPx are AUX or UCN.

If the difference in Master Clock counts in test 4 doesn't match the expected value, an alarm is displayed.

DD-MMM-YYYY HH:MM:SS NMA nn COMM nn Alarm 4nn
MASTER CLOCK TEST - MASTER CLOCK COUNT
UCNWAS – $XXXX XXXX
AUX LATCH WAS – $YYYY YYYY
DIFFERENCE WAS – $ZZZZ ZZZZ
EXPECTED DIFFERENCE WAS 2000 + OR - TOLERANCE

If Random numbers were used for a test that caused an alarm message, the following message will be appended to the alarm message.

FIRST_RANDOM USED FOR THIS TEST - $XXXX

HARDWARE INTERACTIONS USED BY LINKI
Appendix A

A.1 INTERACTIONS OF COMMON EVENTS

INTERRUPT of the 8031 is done by writing 1 then 0 to $600009.

Entry of a Special Command Semaphore (SCS) consists of writing one or two bytes into a FIFO in shared RAM and interrupting the 8031 (which then writes a data link message to the partner PMM).

Entry of a Solicited Message Structure (SMS) consists of writing it (up to 256 bytes) to shared RAM in the next free SMS buffer, entering the page number of the buffer (the page number is the upper byte of the address from the 8031 side) into a transmit FIFO and interrupting the 8031.

RESET of the 8031 consists of writing a one to $600001, then after a 50 millisecond delay writing a zero. If the control card needs to reset the 8031, it uses the Server function of PMEX to tell the COMM to perform the reset.

ACKNOWLEDGE of interrupts:

- Death of the 8031—caused by Shared RAM parity error, Jabber reset, or Watchdog timeout—causes a Level 7 nonmaskable interrupt to the COMM and CTRL processors. Writing 0 then 1 to $600001 acknowledges the NMI.
- Level 3 interrupts occur any time the 8031 has placed something in a reply FIFO waiting to be read. Reset of the 8031 also can cause a level 3 interrupt to the COMM and CTRL processors (this is a problem in which port pins of the 8031 sometimes have a 200 nanosecond pulse when you reset the 8031). Writing 0 then 1 to $600007 acknowledges level 3 interrupts.

A.2 INTERACTIONS OF ALL TESTS

Before any tests have run—An SCS is entered from the COMM board to determine the I/O Link supervisor. An SCS is entered to place the I/O Link in Factory Test Mode.

Before each test runs—Local RAM is reserved for Input, Output, and Error buffers. The link is reserved. The NMI (level 7) and Link (level 3) interrupts are reserved on both boards. After each test, these resources are released.

A.3 INTERACTIONS OF TEST 1 (Reset and Self-Test)

Resets the 8031. Enters RAM_OK_Cleared SCS (this tells the 8031 to proceed with self-test). Watches for IDLE (self-test passed) response in SCS reply FIFO. Repeats six times.

As above, resets the 8031 and brings it through self-test. Enters Determine Supervisor SCS, and interrupts the 8031. If the result is "supervisor," builds ring by entering an SMS to tell the partner node that this node is the next token holder, then enters an SCS to tell this node that the partner node will be the next token holder (after this one).

All entries in SCS or SMS FIFO have a corresponding interrupt to the 8031 instructing it to look in the FIFOs. If a reply comes from the 8031, it consists of an entry in the appropriate FIFO and an interrupt to the test program. The interrupt is acknowledged as above.

A.4 INTERACTIONS OF TEST 2 (Shared RAM Arbitration)

Resets the 8031 to keep it out of Shared RAM (does not give it RAM_OK signal).

Uses 16 bytes of Shared RAM to test arbitration of SRAM: Paints the 16 bytes with a known pattern, then writes repeatedly to one byte of the shared RAM while the server task on the partner board writes to the adjacent byte. After 10,000 writes, checks all 16 bytes for appropriate data.

Resets the 8031, brings it through self-test, and builds the ring.

A.5 INTERACTIONS OF TEST 3 (SRAM Data Pattern)

Resets the 8031 to keep it out of Shared RAM (does not give it RAM_OK signal).

Coats the entire Shared RAM and checks it 34 times, once for each of the walking ones/zeroes patterns.

Resets the 8031, brings it through self-test, and builds the ring.

A.6 INTERACTIONS OF TEST 4 (SRAM Address)

Resets the 8031 to keep it out of Shared RAM (does not give it RAM_OK signal).

Coats the entire Shared RAM in ascending order with random data and checks it, then coats it in descending order with random data and checks it.

Resets the 8031, brings it through self-test, and builds the ring.

A.7 INTERACTIONS OF TEST 5 (SRAM Bus Error)

After the bus error vector is reserved, writes to an even byte of the Shared RAM address space to force a bus error.

A.8 INTERACTIONS OF TEST 6 (SRAM Parity)

Enters an SMS with bad parity (in the next free SMS), the page number is placed in the COMM high priority transmit FIFO and the 8031 is interrupted to force it to look into the FIFO and process the SMS. The 8031 should die when it reads the bad parity and should produce an NMI to both processors.

Resets the 8031, brings it through self-test, and builds the ring.

A.9 INTERACTIONS OF TEST 7 (Sync Timer Gating)

Writes to the control and data registers of the Global timer channel 0 and reads the data register to make sure that it is counting down.

Enters a TSG SCS on the appropriate FIFO and the 8031 is interrupted. The 8031 should then gate/stop the timer. Reads the timer data register to make sure that it is NOT counting down.

Enters a TSG SCS on the appropriate FIFO and the 8031 is interrupted. The 8031 should then start the timer. Reads the timer data register to make sure that it is counting down.

A.10 INTERACTIONS OF TEST 8 (Gap Has Occurred Timer)

The "host" PMM enters a Test Hardware Functions GHO command SMS.

The partner node—which must be running Test 15—times the silence after receipt of the command before it responds and records the time in the THF Time parameter.

The host node writes a Read Slot Parameter SMS (for the THF Time parameter). The value sent back is checked to make sure it is within the specification limits.

A.11 INTERACTIONS OF TEST 9 (Anti-Jabber Timer)

The "host" PMM enters a Test Hardware Functions AJT command SMS.

The partner node—which must be running Test 15—sends back a long response (which exceeds the maximum allowed size and shows up with a bad checksum) until the Anti-Jabber Timer times out and stops the transmission (Jabber Lock flipflop).

The host node enters a Test Hardware Functions Jabber-Reset command SMS. The host node writes a Read Slot Parameter SMS (for the THF Time parameter). The value returned is checked to make sure it is within specification limits.

A.12 INTERACTIONS OF TEST 10 (Anti-Jabber Reset)

The "host" PMM enters a Test Hardware Functions AJR command SMS.

The partner node (which must be running test 15) attempts to unlock the Jabber Lock flipflop and continue to drive. This should kill the partner's 8031 (by holding down its reset pin). After detecting the death of the 8031, the partner node pauses to allow the host to detect the partner 8031's death, then restarts the 8031 and reissues the Determine Supervisor SCS

Meanwhile, the host node polls the partner node until it detects the partner 8031's death (it does not respond). If the partner node 8031 does not die within a reasonable time, the test will time out and alarm. After the host node detects the partner 8031's demise, it pauses to allow the partner to restart it. It then polls the partner's 8031 until it does respond after restarting.

A.13 INTERACTIONS OF TEST 11 (Watchdog Timer)

The "host" PMM enters a Test Hardware Functions WDT command SMS.

The partner node (which must be running test 15) should respond with a message indicating reception of the message, then stop servicing its Watchdog Timer to cause the death of its 8031. After detecting the death of the 8031, the partner node pauses to allow the host to detect the partner 8031's death, then restarts the 8031 and reissues the Determine Supervisor SCS

Meanwhile, the host node polls the partner node until it detects the partner 8031's death (it does not respond). If the partner node 8031 does not die within a reasonable time, the test will time out and alarm. After the host node detects the partner 8031's demise, it pauses to allow the partner to restart it. It polls the partner's 8031 until it does respond after restarting, then checks the value of the THF Time parameter to make sure the Watchdog timer is within specification.

A.14 INTERACTIONS OF TESTS 12-14 (Transfer Tests)

All Transfer tests generate random data in local RAM, build it into a Transmit SMS, enter the SMS, and send the data to the partner node, which responds with whether the data was good or not.

A.15 INTERACTIONS OF TEST 15 (Restarter)

Test 15 waits for the death of the 8031, then pauses and resets the 8031, brings it through self-test, enters a Determine Supervisor SCS, and then enters a Factory Test mode SCS.

READER COMMENTS

Honeywell IAC Automation College welcomes your comments and suggestions to improve future editions of this and other publications.

You can communicate your thoughts to us by fax, mail, or toll-free telephone call. We would like to acknowledge your comments; please include your complet name and address

BY FAX: Use this form; and fax to us at (602) 313-4108

BY TELEPHONE: In the U.S.A. use our toll-free number 1*800-822-7673 (available in the 48 contiguous states except Arizona; in Arizona dial 1-602-313-5558).

BY MAIL: Use this form; detach, fold, tape closed, and mail to us.

Title of Publication: **Process Manager Module Test System**I Issue Date: **9/95**

Publication Number: **PM13-505**

Writer: **C. Phillips**

COMMENTS: _____

_____

_____

_____

_____

_____

_____

_____

RECOMMENDATIONS:_____

_____

_____

_____

_____

_____

_____

_____

NAME _____ DATE _____

TITLE _____

COMPANY _____

ADDRESS _____

CITY _____ STATE _____ ZIP _____

TELEPHONE _____ FAX _____

(If returning by mail, please tape closed; Postal regulations prohibit use of staples.)

Communications concerning technical publications should be directed to:

Automation College
Industrial Automation and Control
Honeywell Inc.
2820 West Kelton Lane
Phoenix, Arizona 85023-3028

FOLD FOLD

From: _____

NO POSTAGE
NECESSARY
IF MAILED
IN THE USA

BUSINESS REPLY MAIL

FIRST CLASS **PERMIT NO. 4332** PHOENIX, ARIZONA

POSTAGE WILL BE PAID BY ....

Honeywell

Industrial Automation and Control
2820 West Kelton Lane
Phoenix, Arizona 85023-3028

Attention: Manager, Quality

FOLD FOLD

Cut Along Line

Additional Comments:

Honeywell

Industrial Automation and Control
Honeywell Inc.
16404 North Black Canyon Highway
Phoenix, Arizona 85023-3033

Helping You Control Your World