	Document #	Date effective
	**LAT-TD-06994-01**	July 20, 2005
	Author(s) Shantha Condamoor	Supersedes
	Subsystem/Office Electronics & DAQ Subsystem	

Document Title

LAT FSW Qualification Test Procedure:

MEMMGT_001: Memory Management Verification

CHANGE HISTORY LOG

Revision	Effective Date	Description of Changes
01	July 20, 2005	Initial Release.

TABLE OF CONTENTS

1. SCOPE

This document describes the procedure followed for qualification testing of the LAT FSW. This Qualification Test Procedure document describes one of the qualification tests executed to verify compliance with the requirements defined in the "Flight Software Specification – Level III" (LAT-SS-00399).

1.1 Test Suite

MEMMGT

The Test Suite is a logical grouping of a set of tests based on common functionalities and can be executed as a suite of tests in some order when the pre-conditions and post-conditions for each test within the suite have been met successfully.

This test procedure *MEMMGT_001* belongs in the Test Suite *MEMMGT* under the *FST* Project.

The MEMMGT tests verify the capability of FSW, on command, to successfully upload data into a specified block of SIU or EPU memory and dump the contents of specific blocks of SIU or EPU memory.

1.1.1 Relationship of MEMMGT_001 to Other Test Suites

Responsibility for verifying 1553 interface and telecommand validation and processing requirements is shared between the MEMMGT_001 test and the CMDFNC test suite. In particular, MEMMGT_001 verifies that the subset of telecommands and telemetry associated with the memory management (MEM) FSW have the correct format, can be received over the 1553 interface and processed, and that memory management telecommands with valid parameters are correctly processed. By contrast, the CMDFNC suite tests these memory management telecommands and telemetry when packet structures and parameters are *invalid*. The specific requirements shared between these test suites are identified in Section 1.3 below. See the CMDFNC test procedure documents for a complete discussion of how responsibility for testing the entire telecommand and telemetry interface is mapped among various qualification test suites.

Responsibility for verifying the ability of FSW to transmit diagnostic telemetry is shared between the MEMMGT_001 test and the NBTLMV test suite. The specific requirements shared between these tests are identified in 1.3 below. Again, see the NBTLMV test procedure documents for a complete discussion of how responsibility for testing diagnostic telemetry is partitioned among various qualification test suites.

The following MEM telecommands and telemetry are used in the MEMMGT test.

MEM Telecommands

APID	FC	Cmd. Packet (L)	Description (L)
0x644	0	LMEMDUMPMEM	Memory Data Dump
0x644	1	LMEMDUMPCANCEL	Memory Dump Cancel
0x644	2	LMEMDUMPPCI	PCI Device Header Dump
0x644	3	LMEMDUMPREG	Processor Register Dump
0x644	4	LMEMLOADMEM	Memory Write
0x644	5	LMEMLOADPCI	PCI Device Header Write
0x644	6	LMEMLOADREG	Processor Register Write
0x644	7	LMEMDUMPPOOL	Memory Pool Status Dump
0x644	8	LMEMDUMPSYMVAL	Memory Symbol Lookup
0x644	9	LMEMDUMPSYMREL	Memory Dump Symbol Relative

MEM Telemetry

APID	Tlm. Packet (L)	Description (L)
0x311	LMEMPOOLDATA	Memory Pool Statistics Dump
0x312	LMEMSYMVAL	Symbol Value Dump
0x314	LMEMSIUDATA	SIU Memory Dump Data
0x315	LMEMEPU0DATA	EPU 0 Memory Dump Data
0x316	LMEMEPU1DATA	EPU 1 Memory Dump Data

1.2 Test ID

MEMMGT_001

Test *MEMMGT_001* contains one main script, MEMMGT_001.py, which can be run when the SIU or an EPU is in the Boot or Application Mode. This test primarily verifies that specific blocks of memory can be written to or read from SIU or EPU memory.

1.3 Requirement(s) Tested

The Qualification Test Procedure described herein is performed to verify that the FSW satisfies the following requirement(s), quoted from the Flight Software Specification – Level III:

Requirement Number	Requirement Name	Requirement	Level of Requirements Verification in This Test
5.3.7.12.1	EPU Memory Dumps	After entering the boot shell, the EPU shall be commandable by the SIU to perform memory dumps.	Full
5.3.7.12.2	SIU Memory Dumps	After entering the boot shell, the SIU shall be commandable from the ground to perform memory dumps.	Full
5.3.7.12.3	Memory Dump Data	Upon receipt of a command to perform a memory dump, the FSW shall transmit the requested data to the spacecraft via the CTDB.	Full
5.3.7.12.4	Memory Dump Cancel	The FSW shall process a command to cancel a memory dump.	Full
5.3.7.12.5	Memory Loads	In order to perform memory writes, the SIU FSW shall process commands, from the SC via the CTDB, that include unit identifier, memory address, memory size, and memory data.	Full
5.3.7.11	Memory Pool Status Dump	The FSW shall receive as input, from the spacecraft via the CTDB, a command to return information on the memory pool, including allocated and free memory, for a specified unit and memory pool.	Full

Requirement Number	Requirement Name	Requirement	Level of Requirements Verification in This Test
5.3.3.8	Multiple Block Commands	The FSW shall execute block commands for uploads or dumps serially (i.e., not simultaneously, not interleaved with other block commands).	Partial (MEMMGT_001 tests block command handling of memory management telecommands. Handling of file management telecommand blocks is covered in the FILMGT suite)

This test also shares verification of certain requirements with the NBTLMV suite. The requirements partially verified in this test are listed below.

Requirement Number	Requirement Name	Requirement	Level of Requirements Verification in This Test
5.3.5.2	Diagnostic Telemetry	[Derived] The FSW shall send diagnostic telemetry messages, as defined in [9], to the SC via the CTDB. Diagnostic telemetry is sent in response to a command and downlinked immediately while the SC is in contact with the ground. If not in contact, the telemetry is written to the solid-state recorder on the spacecraft.	Partial (MEMMGT_001 verifies that telemetry packets transmitted by the memory management FSW have APIDs in the diagnostic telemetry APID range and that these packets are delivered over the CTDB, as required. Transmission of other diagnostic telemetry is covered by the NBTLMV test suite)

This test also shares verification of requirements with the CMDFNC suite. The requirements partially verified in this test are listed below.

Requirement Number	Requirement Name	Requirement	Level of Requirements Verification in This Test
5.2.2.1.1	Command, Telemetry, and Data Bus Protocol – SC	[2] (3.2.5.1.1) [3] (5.3.4) [4] (5.3.1) The SIU FSW shall exchange commands, low rate telemetry, time messages, and ancillary data with the SC C&DH across a MIL-STD-1553B bus, using the MIL-STD-1553B physical layer protocol.	Partial (Also see the CMDFNC suite)

Requirement Number	Requirement Name	Requirement	Level of Requirements Verification in This Test
5.2.2.1.3	Data Format	[1] (3.1.2.5.1.1) [2] (3.2.5.4.6) All data exchanged across the CTDB interface shall be formatted as CCSDS telecommand or telemetry packets, as specified in [7].	Partial (Also see the CMDFNC suite)
5.3.3.5	Command Format	[Derived] The FSW shall be able to process commands of the format described in [9].	Partial (Also see the CMDFNC suite)
5.3.3.9.1	Data Integrity Errors	[Derived] To reduce the chance of executing a corrupted command, the FSW shall validate commands prior to execution.	Partial (Also see the CMDFNC suite)
5.3.3.9.2	Parameter Validation	[Derived] The SIU FSW shall validate command parameters prior to execution.	Partial (Also see the CMDFNC suite)
5.3.3.1	SC Command Processing by SIU	[2] (3.2.6.1.1.2) [4] (5.3.1) The SIU FSW shall process LAT commands from the SC, received via the CTDB, in the order that they are received.	Partial (Also see the CMDFNC suite)
5.3.3.2	Command Execution Notification	[Derived] For all LAT commands from the SC, the FSW shall be capable of generating time-tagged telemetry notification of their dispatch for execution.	Partial (Also see the CMDFNC suite)
5.3.3.3	Command Completion Status	[Derived] The FSW shall be capable of generating time-tagged telemetry notification of command completion status (success or general cause of failure) for ground-initiated commands at the time such information becomes available. As a goal, in addition to completion status, each successful command should result in a modification of a telemetry parameter (e.g., Off to On, Open to Closed, etc.) to further confirm success.	Partial (Also see the CMDFNC suite)

Requirement Number	Requirement Name	Requirement	Level of Requirements Verification in This Test
5.3.3.6	Command Execution	[4] (5.3.2) The SIU FSW shall be able to reconfigure and direct the operation of the instrument through direct execution of commands from the SC.	Partial (Also see the CMDFNC suite)

If the requirement(s) quoted above cite external documents (e.g., "…Further details are provided in [11]"), consult LAT-SS-00399 for the list of citations.

2. DEFINITIONS AND ACRONYMS

The following terms, abbreviations, and acronyms are used in this document:

2.1 Definitions

Hz	Hertz, unit of frequency
s, sec	seconds
V	Volt
W	Watt

2.2 Acronyms

CAL	Calorimeter
EGSE	Electrical Ground Support Equipment
GASU	Global trigger Anti-collision Spacecraft Unit
PTR	Post Test Review
TEM	Tower Electronics Module
TKR	Tracker
TPS	Tower Power Supply
TRR	Test Readiness Review
QAE	Quality Assurance Engineer
TE	Test Engineer

3. REFERENCES

The list below provides documents that are to be used as references for this procedure:

3.1 Applicable Documents

Document Number	Description
SPECIFICATIONS	
LAT-SS-00399	LAT Flight Software Level III Specification
1196 EI-S46310-000	GLAST 1553 Bus Protocol Interface Control
LAT-TD-02659	LAT Flight Software Telecommand and Telemetry Formats
LAT-TD-0561	The Virtual Spacecraft (VSC)
PROCEDURES	
LAT-TD-TBD	CMDFNC_003: 1553 Interface and Command Functional Verification
PLANS	
LAT-MD-00039	Performance Assurance Implementation Plan
LAT-MD-00078	GLAST LAT System Safety Program Plan
LAT-MD-00404	LAT Contamination Control Plan
LAT-MD-00408	LAT Program Instrument Performance Verification Plan
LAT-SS-00296	T & DF Test Plan
LAT-TD-00297	LAT Electronics Test Plan
LAT-TD-00786	LAT Flight Software Test Plan
DRAWINGS	
N/A	
OTHER	
LAT-MD-00091	GLAST Quality Manual
LAT-MD-00471	Control of Nonconforming Product
LAT-MD-00472	Corrective and Preventative Action

4. REQUIREMENTS

This section lists the requirements that shall be followed during the LAT FSW Qualification Testing process.

The Performance Assurance Implementation Plan, LAT-MD-00039, shall be utilized to ensure that the products produced by the GLAST LAT project intended for design qualification, flight and critical ground support equipment usage meet the required levels of quality and functionality for their intended purposes.

The LAT Program Instrument Performance Verification Plan, LAT-MD-00408, shall be utilized to address the testing to be performed at the unit/subsystem and instrument level for flight qualification, proto-flight and acceptance testing phases. Also included are the necessary processes/procedures and systems assurance activities.

4.1 Test Data, Equipment and Software

This procedure shall follow the requirements found in the Control of Nonconforming Product, LAT-MD-00471. This document establishes the method to identify and control nonconforming product developed by the LAT project team.

4.2 Quality Assurance

This procedure shall follow the requirements found in the Corrective and Preventative Action document, LAT-MD-00472 and the GLAST Quality Manual, LAT-MD-00091.

The Corrective and Preventative Action document establishes the method to be used to initiate, implement, evaluate and record corrective and preventive actions. The GLAST Quality Manual defines the methods implemented by the GLAST LAT project to ensure consistent quality of all processes for procurement, design, development and production of flight hardware, flight software and all associated ground support equipment interfacing with flight hardware and software.

4.3 Safety

This procedure shall follow the requirements found in the GLAST LAT System Safety Program Plan, LAT-MD-00078. This document defines all phases of the LAT program including: design, development, fabrication, handling, transportation, storage, test, assembly and operation.

WARNING: When high voltages are present extreme care should be exercised.

4.4 Warnings, Cautions, and Notes

The following SAFETY ALERTS are intended to create awareness of the potential safety hazards and the steps that must be taken to avoid accidents. These same alerts are used throughout this document to identify specific hazards that may endanger personnel and/or equipment.

Identification of every conceivable hazardous situation is impossible. Therefore, all personnel have the responsibility to diligently exercise safe practices whenever exposed to this equipment.

WARNING: Indicates a potential hazardous situation which, if not avoided, could result in death or injury.

CAUTION: Indicates a potential hazardous situation which, if not avoided, could result in damage to equipment.

Note: Indicates a notification of information that is important, but not hazard related.

4.5 General Instructions

This qualification test procedure shall be conducted on a formal basis to its latest approved and released version. The designated Software QAE shall be notified 24 hours prior to the start of this procedure. Software QAE may monitor the execution of all or part of this procedure should they elect to do so.

The Test Engineer conducting this test shall read this document in its entirety and resolve any apparent ambiguities before beginning the procedures described herein.

Deviations from the procedures described in this document and breaks in hardware or software configuration can only be initiated by the Test Engineer, must be approved by QA, and must be documented in Appendix A.

Any nonconformance/defect/anomaly is to be reported in JIRA. Refer to the LAT Flight Software Test Plan LAT-TD-00786 for guidance. Do not alter or break configuration if a failure occurs. Notify Software Quality Assurance.

All success conditions for a test must be met for the test to pass.

5. SETUP

This section describes the hardware and software configuration used for the qualification test described later in this document. Any break from configuration or deviation from a particular procedure must be authorized by the Quality Assurance Engineer and documented in Appendix A.

5.1 Hardware Setup

The list below indicates the equipment that is used to execute the tests described in this document.

Hardware Unit	Manufacturer	GLAT Number (and Hardware Sub-Units by GLAT Number)	Firmware Version (where applicable)
Virtual Spacecraft (VSC)	SLAC		
Spacecraft Interface Unit (SIU)	SLAC		
Event Processing Unit (EPU): 2 (EPU0 and EPU1)	SLAC		
Global trigger, ACD, DAQ, and Signal distribution unit (GASU)	SLAC		
Power Distribution Unit (PDU)	SLAC		
1553 cables and couplers	SLAC		
Unix or Linux Host Establishes connection between VSC and the terminal from which the test is run.			
Power supply for SIU	SLAC		

The Figure below depicts the Testbed on which this qualification test is performed. The particular hardware units utilized in this test are itemized by GLAT number and firmware version in the

preceding table.

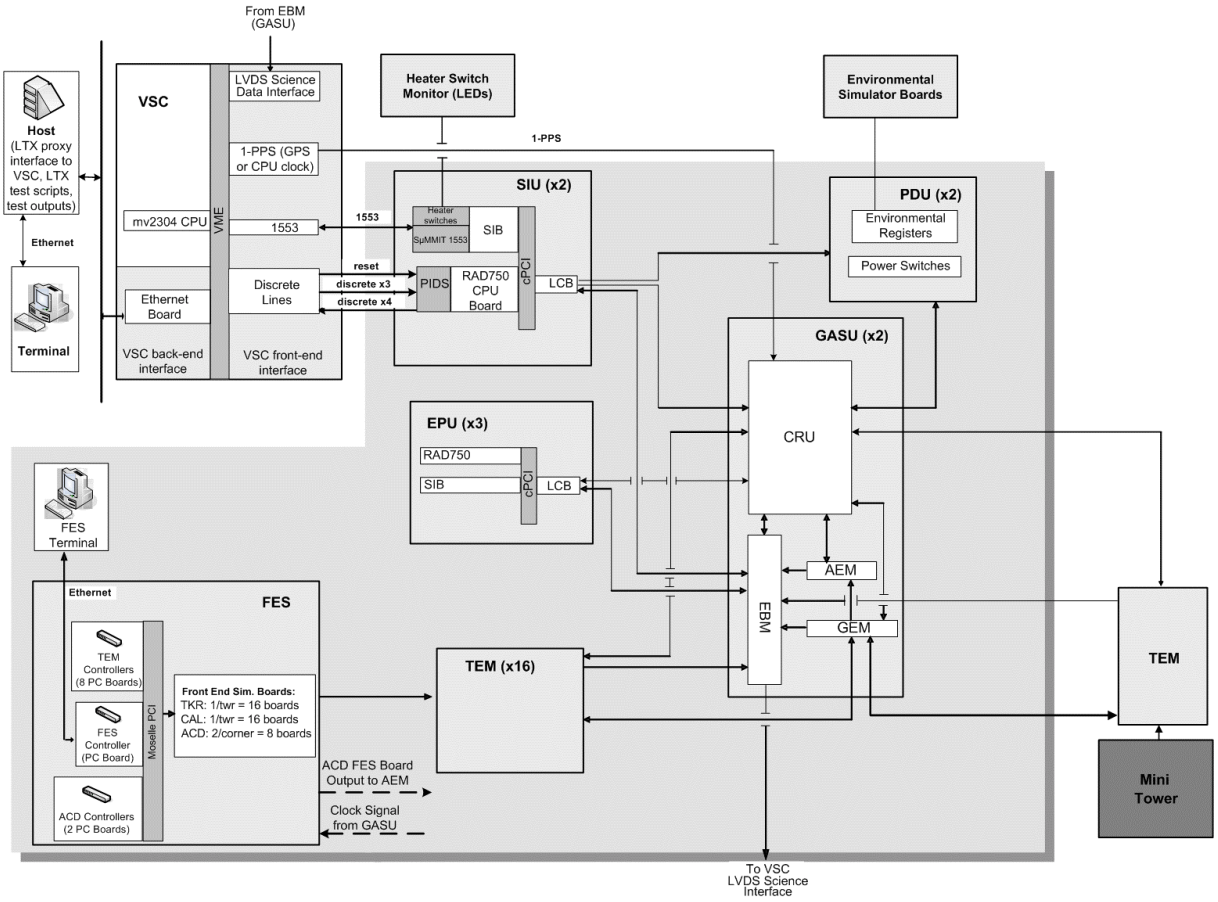

Figure 1. FSW Testbed

5.2 Software Setup

The software required to prepare for and execute the tests described in this Qualification Test Procedure document is itemized in this section.

5.2.1 Test Tools

The following table specifies the test executive used to run this qualification test, and identifies the other software tools used to support the execution of the test. The “Software Version Number” column identifies the version number of the test tool being used; alternatively, this column identifies the hardcopy attachment to this document that records the version of the tool being used (e.g., “Attachment 1”). The “Path to Attachment” column identifies the directory in which the electronic copy of any hardcopy attachment is saved (if applicable).

Software	Description of Software	Software Version Number (or Specify Attachment Number)	Path to Attachment (If Applicable)
LTX	LAT Test Executive		
VSC	Virtual Spacecraft system software		
VPI	VSC Python/Proxy Interface		

5.2.2 Test Scripts

The following table identifies the test scripts that are run to execute this qualification test. The "Script Version Number" column identifies which version of the script is being used; alternatively, this column identifies the hardcopy attachment to this document that records the version of the script being used (e.g., "Attachment 1"). The "Path to Attachment" column identifies the directory in which the electronic copy of any hardcopy attachment is saved (if applicable).

Test Script	Description of Test Script	Script Version Number (or Specify Attachment Number)	Path to Attachment (If Applicable)
MEMMGT_001.py	This script, the main test execution script, can be run in the Boot or Application Modes. It executes tests of SIU FSW, EPU0 FSW, and EPU1 FSW. The script controls the entire test process, detecting the state of the hardware and FSW and advancing SIU and EPU FSW through the different operational modes covered during the test.		

5.2.3 Flight Software

This qualification test is performed on a complete, integrated Candidate Release of FSW. All FSW libraries under test are final Flight Unit Candidate versions. The test described in this document is designed to evaluate the particular FSW packages and constituents listed in the following table; for clarity, only those packages and constituents that are the focus of the test are listed below.

FSW Package	Constituent(s)
MEM	
PBC	

FSW Package	Constituent(s)

The Candidate Release into which these FSW constituents are integrated is identified in the following table. The "Candidate Release Build Number" column specifies the build number of the FSW Candidate Release under test (e.g., "B0-0-1"). The "Candidate Release Attachment Number" column identifies which hardcopy attachment to this document confirms the name and version number of the full set of FSW packages and constituents used in the test (e.g., "Attachment 2"). The "Path to Attachment" column specifies where in the test repository an electronic copy of the hardcopy attachment has been saved.

Candidate Release Build Number	Candidate Release Attachment Number	Path to Attachment

5.3 Setup Validation

5.3.1 Hardware Validation

The following signatures confirm that the Test Engineer and Quality Assurance Engineer have verified the GLAT numbers, firmware version numbers, and the proper connection of all hardware listed in the table in Section 5.1.

_____	_____	_____	_____
Date	Time	Test Engineer	QAE

5.3.2 Software Validation

The Test Engineer performs the following procedure to validate the software setup for this qualification test and records completion of the setup validation steps in the space provided.

Step No.	Description of Step	Step Outcome
1	Record the version numbers of all test tools used to perform this qualification test in the table in Section 5.2.1	Complete/ Not Complete

Step No.	Description of Step	Step Outcome
2	Record the version numbers of all test scripts used to perform this qualification test in the table in Section 5.2.2	Complete/ Not Complete
3	Record the version numbers of the FSW constituents and the Candidate Release on which this test is performed in Section 5.2.3.	Complete/ Not Complete

The following signatures confirm that, using the procedure described in the previous table, the Test Engineer and Quality Assurance Engineer have verified that all versions of test support software, test scripts, and FSW constituents match those identified in Section 5.2.

Date	Time	Test Engineer	QAE

6. TEST PROCEDURE FOR MEMMGT_001

6.1 Test Objective

This test verifies that the SIU and EPU FSW is commandable by ground to perform Memory Management Operations listed in the Requirement(s) Tested section.

This Test Objective is broken down into the following Test Sub-Objectives.

Number	Test Sub-Objective
1	Verify memory management functionality on SIU, EPU0 and EPU1.
2	Verify functionality in both the Boot and Application Modes.
3	Verify Memory Dump commands by issuing the commands and analyzing the resultant dump telemetry. Memory Dump LMEMDUMPMEM PCI Device Header Dump LMEMDUMPPCI Processor Register Dump LMEMDUMPREG Memory Symbol Lookup dump LMEMDUMSYMVAL Memory Dump Symbol Relative LMEMDUMSYMREL Memory Pool Status Dump LMEMDUMPPOOL Verifies dump of few hundreds of bytes spanning few telemetry packets.
4	Verify Memory Dump Cancel Command by issuing the LMEMDUMPCANCEL command while each of the above dumps is in progress and noticing that the dump stops.
5	Verify Memory Load commands by issuing the commands: Memory Load LMEMLOADMEM

Number	Test Sub-Objective
	PCI Device Header LMEMLOADPCI Processor Register Load LMEMLOADREG
6	Verify that FSW executes block commands for uploads or dumps serially (i.e., not simultaneously, not interleaved with other block commands). Verify interleaved Memory Dump Commands are rejected. Verify that interleaving of non-block and block commands is successful.

As explained in Section 1, the MEMMGT_001 test is related to tests in the NBTLMV suite. MEMMGT_001 partially verifies the diagnostic telemetry requirements, thus the following are also listed as Test Objectives and are verified in this test.

Number	Test Sub-Objective
7	Verify that memory management FSW transmits telemetry packets with APIDs in the diagnostic telemetry range and that this telemetry is delivered over the CTDB interface.

In addition, the MEMMGT_001 test is related to tests in the CMDFNC suite. MEMMGT_001 partially verifies some of the 1553 interface and command validation requirements, thus the following are also listed as Test Objectives and are verified in this test.

Number	Test Sub-Objective
8	Verify that the MEM Telecommands and Telemetry exchanged across the 1553 Interface conform to CCSDS packet structure:
9	Verify that all supported valid MEM Telecommands issued by S/C with LAT as their destination are received correctly and processed.
10	Verify that SIU FSW processes LAT commands from the SC in the order that they are received.
11	Verify that for all MEM commands from SC, the FSW is capable of generating time-tagged telemetry notification of their dispatch for execution.
12	Verifies that FSW is capable of generating time-tagged telemetry notification of command completion status (success or general cause of failure) for all MEM commands.
13	Verifies that SIU FSW receives commands at the rate of up to 20 commands per second.

Analysis of results is performed as and when data arrives in telemetry. Typically analysis includes verifying the telemetry values against expected values and tagging the sub-objectives as either “PASS” or “FAIL”.

6.2 Test Input Files

The following table identifies all auxiliary files (e.g., Front End Simulator data files, GLEAM data files) used as inputs to this qualification test. Note that not all qualification tests use input data of this type. The "Input File Version Number" column identifies the version number of the auxiliary file being used; alternatively, this column identifies the hardcopy attachment to this document that records the version of the file being used (e.g., "Attachment 1"). The "Path to Attachment" column specifies where in the test repository an electronic copy of the hardcopy attachment has been saved.

Input File	Description of Input File	Input File Version Number (or Specify Attachment Number)	Path to Attachment (If Applicable)
N/A	N/A	N/A	N/A

6.3 Test Preparation

After the hardware and software setup has been validated, steps may be required to place the hardware and FSW in an operational mode in which the qualification test can be performed or otherwise complete preparations for the test to begin.

The Test Engineer carries out the following procedure to prepare for qualification testing and records completion of the test preparation steps in the space provided.

Step No.	Description of Step	Step Outcome
1	Confirm that the VSC is powered up.	Complete/ Not Complete

6.4 Test Procedure and Test Analysis

This section describes the step by step procedure performed once the test preparation is complete. The test as well as analysis for each of the sub-objectives is conducted by the main test script MEMMGT_001.

The Test Engineer proceeds with the qualification test procedure itself, as described below, and records the outcome of each step during test execution. The outcome of each step is either

"Complete" or "Not Complete" (for steps which involve no analysis or verification); or, "Pass" or "Fail" (for steps involved in verifying completion of test objectives and sub-objectives).

The test procedure is executed on the SIU and EPUs, in both Boot Mode and Application Mode, in 7 major parts:

- SIU Boot Mode Testing
- SIU Application Mode Testing
- EPU0 Boot Mode Testing
- EPU0 Application Mode Testing
- EPU1 Boot Mode Testing
- EPU1 Application Mode Testing
- Review of Test Outputs

6.4.1 Part 1: SIU Boot Mode Testing

This part of the test exercises Boot Mode memory management functionality on the SIU.

Step No.	Description of Step	Step Outcome
1	At the test terminal, run the script *MEMMGT_001* under LTX through the VSC with the following command: **$ ltx run MEMMGT_001**	Complete/ Not Complete
2	The test script determines whether the SIU is powered on by checking whether telemetry is being transmitted. Regardless of the SIU's current operational mode, the script sends the SIU the LPBCRESET telecommand to reboot the unit. The script then checks whether the SIU FSW is operating in Boot Mode by detecting if boot housekeeping is being transmitted. If not, the script sends the LPBCRESET command and checks again. If the SIU FSW cannot be placed in Boot Mode, LTX prints an error message to the screen and exits, aborting the test. If the test is NOT aborted, mark "Complete" for the Step Outcome.	Complete/ Not Complete

Step No.	Description of Step	Step Outcome
3	With FSW on the SIU advanced to Boot Mode, the test script begins executing the main MEMMGT test script, starting with the SIU Boot Mode Memory Dumps phase. The test script issues the LMEMDUMPMEM, LMEMDUMPPCI, and LMEMDUMPREG telecommands in sequence to the SIU. After each of these telecommands is sent, the script examines the SIU boot housekeeping (LBTHKP) packets, confirming that the current dump command was received and processed (by checking the LPBCTOTALERRCNT, LPBCNEXTERRWORD, LPBCTCRECVCNT, LPBCTCACCCNT, and LPBCLASTERRWORD fields) and that the commanded memory dump was properly executed (by checking the LPBCLASTFUNC, LPBCLASTAPID, LPBCMEMDUMPWC, LPBCMEMDUMPADDR, LPBCMEMDUMPDAT** fields). In the Boot Mode, PBC code continuously dumps data from fixed addresses and hence a proper memory dump is verified by looking for specific packets containing the expected addresses. If all expected data words in expected address ranges have been collected, the test concludes that the Memory Dump operations were successful. If not, then the output log file contains descriptions of why the test failed.	N/A
4	The test script next executes its SIU Boot Mode Memory Dump Cancellations phase. The test script re-issues the LMEMDUMPMEM telecommand with a large dump value to the SIU. After the command is issued and while it is executing, the script sends the LMEMDUMPCANCEL command to cancel the ongoing dump. While the dumps are in progress and after the LMEMDUMPCANCEL command is sent, the script examines the SIU boot housekeeping (LBTHKP) packets, confirming that the cancel command was received and processed (by checking the LPBCTOTALERRCNT, LPBCNEXTERRWORD, LPBCTCRECVCNT, LPBCTCACCCNT, and LPBCLASTERRWORD fields) and that the ongoing dump was indeed cancelled (by checking the LPBCLASTFUNC, LPBCLASTAPID, LPBCMEMDUMPWC, LPBCMEMDUMPADDR, LPBCMEMDUMPDAT** fields). It is verified that the ongoing dump was cancelled by noticing that the boot housekeeping telemetry does not contain memory dumps in the requested address range. Care is taken to ensure that the Requested Dump address ranges are not in the boot diagnostic address range that boot housekeeping telemetry constantly dumps.	N/A
5	The test script next executes its SIU Boot Mode Memory Loads phase. Memory load payloads are a few hundred bytes in size and thus span several telecommand packets. First, the test script issues the LMEMDUMPMEM telecommand to the SIU to establish the baseline contents of selected memory areas. Next, the script issues the LMEMLOADMEM telecommand to the SIU to load test values to that memory region. The script re-issues the LMEMDUMPMEM telecommand to request a memory dump to confirm that the requested values were loaded to memory. The script then repeats this process for the LMEMDUMPPCI/LMEMLOADPCI/LMEMDUMPPCI and LMEMDUMPREG/LMEMLOADREG/LMEMDUMPREG command sets to confirm that these types of memory loads are also correctly performed. As the telecommands are sent, the script examines the SIU boot housekeeping (LBTHKP) packets, confirming that the current memory load command was received and processed (by checking the LPBCTOTALERRCNT, LPBCNEXTERRWORD, LPBCTCRECVCNT, LPBCTCACCCNT, and LPBCLASTERRWORD fields) and that the commanded load was properly executed (by checking the LPBCLASTFUNC, LPBCLASTAPID, LPBCTCRECVCNT, LPBCTCACCCNT, LPBCTOTALERRCNT fields, and other fields). Note: After this Load sub-test has been run, the original values in all the memory locations prior to the test are restored to their original values by reissuing the LOAD commands and verifying that the original values were restored with the DUMP commands.	N/A

Step No.	Description of Step	Step Outcome
6	The test script next executes its SIU Boot Mode Multiple Block Commands phase. The script sends an LMEMDUMPMEM command while a memory dump is in progress, and reviews the same fields in boot housekeeping mentioned in the previous steps to confirm that the second LMEMDUMPMEM command packet is rejected. This verifies that interleaved block commands are rejected. Note: Correct handling of non-interleaved (sequential) block commands is verified in Steps 3, 4, and 5 described above. In those steps, the test confirms that sequentially transmitted MEM DUMP and LOAD telecommands are properly executed, and that a DUMP/CANCEL/DUMP sequence is also properly executed. The complete analysis of block command handling based on the results of Steps 3, 4, 5, and 6 is recorded in the output log printed and reviewed at the end of the test.	N/A
7	Next, the test script executes its SIU Boot Mode MEM Command Data Format and Integrity Errors phase. The script confirms that the SIU boot housekeeping telemetry (LBTHKP) packets conform to the CCSDS standard. Note: In Steps 3, 4, and 5, the test verifies that all Boot Mode MEM telecommands transmitted in those steps have the proper data format and integrity and conform to the CCSDS standard. The complete analysis of SIU Boot Mode command and telemetry formatting based on the results of Steps 3, 4, 5, and 7 is recorded in the output log printed and reviewed at the end of the test.	N/A
8	Next, the test script executes its SIU Boot Mode MEM Command Format and Parameter Validation phase. Proper execution of telecommands that have the correct format and valid parameters is tested in Steps 3, 4, and 5. The complete analysis of command parameter validation based on the results of Steps 3, 4, and 5 is recorded in the output log printed and reviewed at the end of the test.	N/A
9	Next, the test script executes its SIU Boot Mode MEM Command Processing Order phase. The script sends a series of LMEMDUMPMEM commands to the SIU, each command specifying a different address range (for instance, incrementing the address range by 0x100). The script examines the LPBCLASTFUNC and LPBCLASTAPID fields of SIU boot housekeeping telemetry (LBTHKP packets) to confirm that the LMEMDUMPMEM/LMEMDUMPPCI/LMEMDUMPREG/LMEMDUMPSYMREL commands were received and processed in the order in which they were sent. The script also confirms that the address range in dumped memory data telemetry changes in the same order in which those address ranges were requested in the series of LMEMDUMPMEM commands.	N/A
10	Next, the test script executes its SIU Boot Mode MEM Command Completion Status phase. In Steps 3, 4, 5, and 6, the test checks that command completion status is reported by FSW, using the telemetry fields described in those steps. The complete analysis of command completion status notification based on the results of Steps 3-6 is recorded in the output log printed and reviewed at the end of the test.	N/A
11	Next, the test script executes its SIU Boot Mode Command Rates phase. The test script sends an alternating series of LMEMDUMPMEM and LMEMDUMPCANCEL commands at the rate of 20 commands per second to the SIU and examines SIU boot housekeeping telemetry (LBTHKP packets) to verify that the telemetry contains the specified dumps.	N/A

6.4.2 Part 2: SIU Application Mode Testing

Once SIU Boot Mode testing is complete, the test continues to memory management functionality in SIU Application Mode.

For each of the MEM-related commands sent to the SIU during this part of the test, a Command Verification (CmdConfirm) telemetry packet is received, which is further analyzed to verify the status of the reception and execution of the commands. The test script examines the ITC_NODE, ITC_TASK, and ITC_EXESTATUS fields in these packets, as well as the CmdHeader field (in which the CCSDS header of each MEM telecommand sent to SIU FSW is reflected).

Additionally, the test script sends the ReqDiagPacket (APID:0x650, FC:0) command to request "demand" versions of Memory Load/Dump Statistics Housekeeping Packets, DiagMemStats0 (APID: 0x28C) and DiagMemStats1 (APID:0x28D), which contain statistics about the memory loads or writes just performed. The various LHKSMEM* fields in these 2 packets are used for analysis.

Step No.	Description of Step	Step Outcome
12	The test script advances the SIU FSW to Application Mode by sending the LPBCRTOSEXEC telecommand, which commands the SIU to perform a secondary boot. The script checks that the SIU is operating in Application Mode by detecting whether Application Mode housekeeping telemetry is being transmitted. If not, the script sends the LPBCRTOSEXEC command and checks again. If the SIU FSW cannot be placed in Application Mode, LTX prints an error message to the screen and exits, aborting the test. If the test is NOT aborted, mark "Complete" for the Step Outcome.	Complete/ Not Complete
13	The test script begins executing its SIU Application Mode Memory Dump Commands phase. The test script issues the LMEMDUMPMEM telecommand to the SIU. After the telecommand is sent, the script examines the Command Verification telemetry (CmdConfirm packets), confirming that SUCCESS is reported in the ITC_EXESTATUS field. The script also examines the LMEMSIUDATA dump data packets (LMEMTSIUADDRESS, LMEMTSIUWORDCNT, LMEMTSIUCMDFUNC, and LMEMTSIUDATA* fields) to verify the memory was dumped as requested. The script continues with the rest of the memory dump functionality, again examining the CmdConfirm telemetry packets, and issuing/receiving the following command/telemetry pairs: LMEMDUMPPCI/LMEMSIUDATA; LMEMDUMPREG/LMEMSIUDATA; LMEMDUMPSYMVAL/LMEMSYMVAL, LMEMDUMPSYMREL/LMEMSIUDATA, and finally, LMEMDUMPPOOL/LMEMPOOLDATA.	N/A

Step No.	Description of Step	Step Outcome
14	Next, the script executes its SIU Application Mode Memory Dump Cancellation Commands phase. The test script sends the LMEMDUMPMEM telecommand to the SIU. After the telecommand is sent and while it is executing, the script sends the LMEMDUMPCANCEL command to cancel the current dump. The script checks for the SUCCESS status code in the ITC_EXESTATUS field of CmdConfirm telemetry and confirms that LMEMSIUDATA packets cease to be transmitted, confirming that the LMEMDUMPCANCEL command was successfully executed.	N/A
15	Next, the script executes its SIU Application Mode Memory Load Commands phase. Memory load payloads are a few hundred bytes in size and thus span several telecommand packets. The script issues the LMEMDUMPMEM telecommand to the SIU and examines the LMEMTSIUADDRESS, LMEMTSIUWORDCNT, LMEMTSIUCMDFUNC, and LMEMTSIUDATA* fields in the dump telemetry to establish the baseline contents of the corresponding memory areas. Next, the script issues the LMEMLOADMEM telecommand to the SIU to load test values to memory. The script re-issues the LMEMDUMPMEM telecommand to request a memory dump to confirm that the requested values were loaded to memory. The script then repeats this process for the LMEMDUMPPCI/LMEMLOADPCI/LMEMDUMPPCI and LMEMDUMPREG/LMEMLOADREG/LMEMDUMPREG sets to confirm that the requested memory loads were indeed performed. As the telecommands are sent, the script also examines the SIU CmdConfirm telemetry to verify that SUCCESS is reported in the ITC_EXESTATUS field. Note: After this Load sub-test has been run, the original values in all the memory locations prior to the test are restored to their original values by reissuing the LOAD commands and verifying that the original values were restored with the DUMP commands	N/A
16	Next, the script executes its SIU Application Mode Multiple Block Commands phase. The script sends an LMEMDUMPMEM command while an LMEMDUMPMEM command is already being executed, and reviews the same fields in telemetry used in the preceding steps to confirm that the second LMEMDUMPMEM command packet is rejected. Note: Correct handling of non-interleaved block commands in Application Mode is verified in Steps 13, 14, and 15 described above. In those steps, the test confirms that sequentially transmitted (non-overlapping) MEM telecommands are properly executed and that DUMP/CANCEL/DUMP sequences are also properly processed. The complete analysis of block command handling based on the results of Steps 13, 14, 15, and 16 is recorded in the output log printed and reviewed at the end of the test.	N/A
17	Next, the script executes the SIU Application Mode MEM Command and Telemetry Data Format and Integrity Errors phase. In Steps 13, 14, and 15, the test verifies that all Application Mode MEM telecommands transmitted in those steps have the proper data format and conform to the CCSDS standard. Those steps also confirm that the LMEMSIUDATA, LMEMPOOLDATA, LMEMSYMVAL telemetry packets returned in response to those commands conform to the CCSDS standard. The complete analysis of command and telemetry data format and integrity based on the results of Steps 13, 14, 15, and 16 is recorded in the output log printed and reviewed at the end of the test.	N/A

Step No.	Description of Step	Step Outcome
18	Next the script executes the SIU Application Mode MEM Command Format and Parameter Validation phase. Proper execution of telecommands with correct format and valid parameters is tested in Steps 13, 14, and 15. The complete analysis of command formatting and parameter validation based on the results of Steps 13, 14, and 15 is recorded in the output log printed and reviewed at the end of the test.	N/A
19	Next the script executes the SIU Application Mode Command Processing Order phase. The test script sends all Application Mode memory management telecommands (LMEMDUMPMEM, LMEMDUMPCANCEL, LMEMDUMPPCI, LMEMDUMPREG, LMEMLOADMEM, LMEMLOADPCI, LMEMLOADREG, LMEMDUMPPOOL, LMEMDUMPSYMVAL, and LMEMDUMPSYMREL) in sequence to the SIU. After sending each command, the script examines the ITC_DEQTIMHI and ITC_DEQTIMLO fields in CmdConfirm telemetry to confirm that the time at which each command is "dequed" increases in parallel order with the time at which each command was sent.	N/A
20	Next the script executes the SIU Application Mode Command Execution Notification phase. In Steps 13, 14, 15, and 16, after sending each command, the script examines the ITC_DEQTIMHI and ITC_DEQTIMLO fields and the ITC_EXETIMHI and ITC_EXETIMLO fields in CmdConfirm telemetry to confirm that FSW reports when each command is "dequed" and executed. The complete analysis of command execution notification based on the results of Steps 13-16 is recorded in the output log printed and reviewed at the end of the test.	N/A
21	Next the script executes the SIU Application Mode Command Completion Status phase. In Steps 13, 14, 15, and 16, the test checks that command completion status is reported by FSW, using the telemetry fields described in those steps as well as the Timestamp field in the CCSDS header of CmdConfirm telemetry. The complete analysis of command completion status based on the results of Steps 13-16 is recorded in the output log printed and reviewed at the end of the test.	N/A
22	Next the script executes the SIU Application Mode Command Rates phase. The script issues all Application Mode memory management telecommands in conjunction with Magic 7 commands (LLSMSIATTITUDE, LLSMSIANCILLARY, and LLSMSITIMETONE) at the specified rate of 20 Hz and examines the ITC_DEQTIMHI and ITC_DEQTIMLO fields, ITC_EXETIMHI and ITC_EXETIMLO fields, and ITC_EXESTATUS field in CmdConfirm telemetry to determine the commands were sent at the required rate and were successfully received and dispatched for execution.	N/A

6.4.3 Part 3: EPU0 Boot Mode Testing

The test continues with Boot Mode memory management functionality on EPU0. As described below, at this stage, the test procedure moves through a series of steps to place the GASU, PDU, EPUs, and instrument power in a "clean" state for testing. Once this is accomplished, the memory management testing continues.

Step No.	Description of Step	Step Outcome
23	The test script issues an LIMLOADSHED command. The SIU reboots and the GASU, PDU, EPU, and instrument power are turned off for a fresh test start.	N/A
24	The test script determines whether the SIU is now operating in Boot Mode as a result of the reboot by detecting whether boot housekeeping telemetry is being transmitted. If not, the script sends the LPBCRESET command and checks again. If the SIU FSW cannot be placed in Boot Mode, LTX prints an error message to the screen and exits, aborting the test. If the test is NOT aborted, mark "Complete" for the Step Outcome.	Complete/ Not Complete
25	The test script advances the SIU FSW to Application Mode by sending the LPBCRTOSEXEC telecommand, which commands the SIU to perform a secondary boot. The script checks that the SIU is operating in Application Mode by detecting whether Application Mode housekeeping telemetry is being transmitted. If not, the script sends the LPBCRTOSEXEC command and checks again. If the SIU FSW cannot be advanced to Application Mode, LTX prints an error message to the screen and exits, aborting the test. If the test is NOT aborted, mark "Complete" for the Step Outcome.	Complete/ Not Complete
26	The script sends the LIMMAINFEEDON telecommand to power up the PDU and GASU.	N/A
27	The script then sends the LIMPOWERON command to power up EPU0. In response to a powerup, EPU0 should begin a primary boot. The test script determines whether EPU0 is operating in Boot Mode as a result of the powerup and primary boot by detecting whether the EPU is transmitting boot housekeeping telemetry. If not, the script sends the LPBCRESET command to EPU0 and checks again. If the EPU0 FSW cannot be placed in Boot Mode, LTX prints an error message to the screen and exits, aborting the test. If the test is NOT aborted, mark "Complete" for the Step Outcome.	Complete/ Not Complete
28	With FSW on EPU0 advanced to Boot Mode, the test continues with the EPU0 Boot Mode Memory Dumps phase. The test script issues the LMEMDUMPMEM, LMEMDUMPPCI, and LMEMDUMPREG telecommands in sequence to EPU0. After each of these telecommands is sent, the script examines the EPU0 boot housekeeping (LBTEPU0HKP) packets, confirming that the current dump command was received and processed (by checking the LPBC0TOTALERRCNT, LPBC0NEXTERRWORD, LPBC0TCRECVCNT, LPBC0TCACCCNT, and LPBC0LASTERRWORD fields) and that the commanded dump was properly executed (by checking the LPBC0LASTFUNC, LPBC0LASTAPID, LPBC0MEMDUMPWC, LPBC0MEMDUMPADDR, LPBC0MEMDUMPDAT** fields). In the Boot Mode, PBC code continuously dumps data from fixed addresses and hence dump is verified by looking for specific packets containing the expected addresses. If all expected data words in expected address ranges have been collected, the test concludes that the Memory Dump operations were successful. If not, then the output log file contains descriptions of why the test failed.	N/A

Step No.	Description of Step	Step Outcome
29	The test script next executes its EPU0 Boot Mode Memory Dump Cancellations phase. The test script re-issues the LMEMDUMPMEM, LMEMDUMPPCI, LMEMDUMPREG in sequence to EPU0. After each command is issued and while it is executing, the script sends the LMEMDUMPCANCEL command to cancel the ongoing dump. While the dumps are in progress and after the LMEMDUMPCANCEL command is sent, the script examines EPU0 boot housekeeping (LBTEPU0HKP) packets, confirming that the cancel command was received and processed (by checking the LPBC0TOTALERRCNT, LPBC0NEXTERRWORD, LPBC0TCRECVCNT, LPBC0TCACCCNT, and LPBC0LASTERRWORD fields) and that the ongoing dump was indeed cancelled (by checking the LPBC0LASTFUNC, LPBC0LASTAPID, LPBC0MEMDUMPWC, LPBC0MEMDUMPADDR, LPBC0MEMDUMPDAT** fields). It is verified that the ongoing dump was cancelled by noticing that the EPU0 boot housekeeping telemetry does not contain memory dumps in the requested address range. Care is taken to ensure that the Requested Dump address ranges are not in the boot diagnostic address range that boot housekeeping telemetry constantly dumps.	N/A
30	The test script next executes its EPU0 Boot Mode Memory Loads phase. Memory load payloads are a few hundred bytes in size and thus span several telecommand packets. First, the test script issues the LMEMDUMPMEM telecommand to EPU0 to establish the baseline contents of selected memory areas. Next, the script issues the LMEMLOADMEM telecommand to EPU0 to load test values to that memory region. The script re-issues the LMEMDUMPMEM telecommand to request a memory dump to confirm that the requested values were loaded to memory. The script then repeats this process for the LMEMDUMPPCI/LMEMLOADPCI/LMEMDUMPPCI and LMEMDUMPREG/LMEMLOADREG/LMEMDUMPREG command sets to confirm that these types of memory loads are also correctly performed. As the telecommands are sent, the script examines EPU0 boot housekeeping (LBTEPU0HKP) packets, confirming that the current memory load command was received and processed (by checking the LPBC0TOTALERRCNT, LPBC0NEXTERRWORD, LPBC0TCRECVCNT, LPBC0TCACCCNT, and LPBC0LASTERRWORD fields) and that the commanded load was properly executed (by checking the LPBC0LASTFUNC, LPBC0LASTAPID, LPBC0TCRECVCNT, LPBC0TCACCCNT, LPBC0TOTALERRCNT fields, and other fields in telemetry). Note: After this Load sub-test has been run, the original values in all the memory locations prior to the test are restored to their original values by reissuing the LOAD commands and verifying that the original values were restored with the DUMP commands.	N/A
31	The test script next executes its EPU0 Boot Mode Multiple Block Commands phase. The script sends an LMEMDUMPMEM command to EPU0 while a memory dump is in progress, and reviews the same fields in boot housekeeping mentioned in the previous steps to confirm that the second LMEMDUMPMEM command packet is rejected. This verifies that interleaved block commands are rejected. Note: Correct handling of non-interleaved block commands is verified in Steps 28, 29, and 30 described above. In those steps, the test confirms that sequentially transmitted MEM DUMP and LOAD telecommands are properly executed, and that a DUMP/CANCEL/DUMP sequences is also properly executed. The complete analysis of block command handling based on the results of Steps 28, 29, 30 and 31 is recorded in the output log printed and reviewed at the end of the test.	N/A

Step No.	Description of Step	Step Outcome
32	Next, the test script executes its EPU0 Boot Mode MEM Command Data Format and Integrity Errors phase. The script confirms that EPU0 boot housekeeping telemetry (LBTEPU0HKP) packets conform to the CCSDS standard. Note: In Steps 28, 29, and 30, the test verifies that all Boot Mode MEM telecommands transmitted in those steps have the proper data format and integrity and conform to the CCSDS standard. The complete analysis of EPU0 Boot Mode command and telemetry formatting based on the results of Steps 28, 29, 30, and 32 is recorded in the output log printed and reviewed at the end of the test.	N/A
33	Next, the test script executes its EPU0 Boot Mode MEM Command Format and Parameter Validation phase. Proper execution of telecommands that have the correct format and valid parameters is tested in Steps 28, 29, and 30. The complete analysis of command parameter validation based on the results of Steps 28, 29, and 30 is recorded in the output log printed and reviewed at the end of the test.	N/A
34	Next, the test script executes its EPU0 Boot Mode MEM Command Processing Order phase. The script sends a series of LMEMDUMPMEM commands to EPU0, each command specifying a different address range (for instance, incrementing the address range by 0x100).The script examines the LPBCLASTFUNC and LPBCLASTAPID fields of SIU boot housekeeping telemetry (LBTHKP packets) to confirm that the LMEMDUMPMEM/LMEMDUMPPCI/LMEMDUMPREG/LMEMDUMPSYMREL commands were received and processed in the order in which they were sent. The script also confirms that the address range in dumped memory data telemetry changes in the same order in which those address ranges were requested in the series of LMEMDUMPMEM commands.	N/A
35	Next, the test script executes its EPU0 Boot Mode MEM Command Completion Status phase. In Steps 28, 29, 30, and 31, the test checks that command completion status is reported by FSW, using the telemetry fields described in those steps. The complete analysis of command parameter validation based on the results of Steps 28, 29, 30 and 31 is recorded in the output log printed and reviewed at the end of the test.	N/A

6.4.4 Part 4: EPU0 Application Mode Testing

The test moves on to Application Mode memory management functionality on EPU0.

For each of the MEM-related commands sent to EPU0 during this part of the test, a Command Verification (CmdConfirm) telemetry packet is received, which is further analyzed to verify the status of the reception and execution of the commands. The test script examines the ITC_NODE, ITC_TASK, and ITC_EXESTATUS fields in these packets, as well as the CmdHeader field (in which the CCSDS header of each MEM telecommand sent to EPU FSW is reflected).

Additionally, the test script sends the ReqDiagPacket (APID:0x650, FC:0) command to request "demand" versions of Memory Load/Dump Statistics Housekeeping Packets, DiagMemStats0 (APID: 0x28C) and DiagMemStats1 (APID:0x28D), which contain statistics about the memory loads or writes just performed. The LHKSMEM* fields in these 2 packets is used for analysis.

Step No.	Description of Step	Step Outcome
36	The test script advances EPU0 FSW to Application Mode by sending the LPBCRTOSEXEC telecommand, which commands the EPU to perform a secondary boot. The script checks that EPU0 is operating in Application Mode by detecting whether the EPU is transmitting Application Mode housekeeping telemetry. If not, the script sends the LPBCRTOSEXEC command and checks again. If the EPU0 FSW cannot be advanced to Application Mode, LTX prints an error message to the screen and exits, aborting the test. If the test is NOT aborted, mark "Complete" for the Step Outcome.	Complete/ Not Complete
37	The test script begins executing its EPU0 Application Mode Memory Dump Commands phase. The test script issues the LMEMDUMPMEM telecommand to EPU0. After the telecommand is sent, the script examines the Command Verification telemetry (CmdConfirm packets), confirming that SUCCESS is reported in the ITC_EXESTATUS field. The script also examines the LMEMEPU0DATA dump data packets (LMEMTEPU0ADDRESS, LMEMTEPU0WORDCNT, LMEMTEPU0CMDFUNC, and LMEMTEPU0DATA* fields) to verify the memory was dumped as requested. The script continues with the rest of the memory dump functionality, again examining the CmdConfirm telemetry packets, and issuing/receiving the following command/telemetry pairs: LMEMDUMPPCI/LMEMEPU0DATA; LMEMDUMPREG/LMEMEPU0DATA; LMEMDUMPSYMVAL/LMEMSYMVAL, LMEMDUMPSYMREL/LMEMEPU0DATA, and finally, LMEMDUMPPOOL/LMEMPOOLDATA.	N/A
38	Next, the script executes its EPU0 Application Mode Memory Dump Cancellation Commands phase. The test script sends the LMEMDUMPMEM telecommand to EPU0. After the telecommand is sent and while it is executing, the script sends the LMEMDUMPCANCEL command to cancel the current dump. The script checks for the SUCCESS status code in the ITC_EXESTATUS field of CmdConfirm telemetry and confirms that LMEMEPU0DATA packets cease to be transmitted, confirming that the LMEMDUMPCANCEL command was successfully executed.	N/A

Step No.	Description of Step	Step Outcome
39	Next, the script executes its EPU0 Application Mode Memory Load Commands phase. Memory load payloads are a few hundred bytes in size and thus span several telecommand packets. The script issues the LMEMDUMPMEM telecommand to EPU0 and examines the LMEMTEPU0ADDRESS, LMEMTEPU0WORDCNT, LMEMTEPU0CMDFUNC, and LMEMTEPU0DATA* fields in the dump telemetry to establish the baseline contents of the corresponding memory areas. Next, the script issues the LMEMLOADMEM telecommand to EPU0 to load test values to memory. The script re-issues the LMEMDUMPMEM telecommand to request a memory dump to confirm that the requested values were loaded to memory. The script then repeats this process for the LMEMDUMPPCI/LMEMLOADPCI/LMEMDUMPPCI and LMEMDUMPREG/LMEMLOADREG/LMEMDUMPREG sets to confirm that the requested memory loads were indeed performed. As the telecommands are sent, the script also examines the EPU0 CmdConfirm telemetry to verify that SUCCESS is reported in the ITC_EXESTATUS field. Note: After this Load sub-test has been run, the original values in all the memory locations prior to the test are restored to their original values by reissuing the LOAD commands and verifying that the original values were restored with the DUMP commands	N/A
40	Next, the script executes its EPU0 Application Mode Multiple Block Commands phase. The script sends an LMEMDUMPMEM command to EPU0 while an LMEMDUMPMEM command is already being executed, and reviews the same fields in telemetry to confirm that the second LMEMDUMPMEM command packet is rejected. Note: The test verifies other aspects of Application Mode memory management block command handling in Steps 37, 38, and 39 described above. Namely, the test confirms that sequentially transmitted (non-overlapping) MEM telecommands are properly executed and that DUMP/CANCEL/DUMP sequences are also properly processed. In the output log printed and reviewed at the end of the test, entries corresponding to steps 37, 38, 39, and 40 clearly spell out how these various aspects of block command handling are verified.	N/A
41	Next, the script executes the EPU0 Application Mode MEM Command and Telemetry Data Format and Integrity Errors phase. Note: In Steps 37, 38, and 39, the test verifies that all Application Mode MEM telecommands transmitted in those steps have the proper data format and conform to the CCSDS standard. Those steps also confirm that the EPU0 LMEMSIUDATA, LMEMPOOLDATA, LMEMSYMVAL telemetry packets returned in response to those commands conform to the CCSDS standard. In the output log printed and reviewed at the end of the test, entries corresponding to steps 37, 38, and 39 clearly spell out how command and telemetry formatting and integrity are verified.	N/A
42	Next the script executes the EPU0 Application Mode MEM Command Format and Parameter Validation phase. Proper execution of telecommands with correct format and valid parameters is tested in Steps 37, 38, and 39. In the output log printed and reviewed at the end of the test, entries clearly spell out how parameter validation is verified during these steps.	N/A

Step No.	Description of Step	Step Outcome
43	Next the script executes the EPU0 Application Mode Command Processing Order phase. The test script sends all Application Mode memory management telecommands (LMEMDUMPMEM, LMEMDUMPCANCEL, LMEMDUMPPCI, LMEMDUMPREG, LMEMLOADMEM, LMEMLOADPCI, LMEMLOADREG, LMEMDUMPPOOL, LMEMDUMPSYMVAL, and LMEMDUMPSYMREL) in sequence to EPU0. After sending each command, the script examines the ITC_DEQTIMHI and ITC_DEQTIMLO fields in CmdConfirm telemetry to confirm that the time at which each command is "dequed" increases in parallel order with the time at which each command was sent.	N/A
44	Next the script executes the EPU0 Application Mode Command Execution Notification phase. In Steps 37, 38, 39, and 40, after sending each command, the script examines the ITC_DEQTIMHI and ITC_DEQTIMLO fields and the ITC_EXETIMHI and ITC_EXETIMLO fields in CmdConfirm telemetry to confirm that FSW reports when each command is "dequed" and executed.	N/A
45	Next the script executes the EPU0 Application Mode Command Completion Status phase. In Steps 37, 38, 39, and 40, the test checks that command completion status is reported by FSW, using the telemetry fields described in those steps as well as the Timestamp field in the CCSDS header of CmdConfirm telemetry. In the output log printed and reviewed at the end of the test, entries clearly spell out how command completion status is verified during these steps.	N/A
46	The test script sends the LPBCRESET command to EPU0 to reboot that unit. It then powers down EPU0 by sending the LIMPOWEROFF command.	N/A

6.4.5 Part 5: EPU1 Boot Mode Testing

The test continues with Boot Mode memory management functionality on EPU1. In the last step of the previous part of the procedure, EPU0 is rebooted for a clean exit from the test and is then shut down.

Step No.	Description of Step	Step Outcome
47	The script then sends the LIMPOWERON command to power up EPU1. In response to a powerup, EPU1 should begin a primary boot. The test script determines whether EPU1 is operating in Boot Mode as a result of the powerup and primary boot by detecting whether EPU1 is transmitted boot housekeeping telemetry. If not, the script sends the LPBCRESET command to EPU1 and checks again. If the EPU1 FSW cannot be placed in Boot Mode, LTX prints an error message to the screen and exits, aborting the test. If the test is NOT aborted, mark "Complete" for the Step Outcome.	Complete/ Not Complete

Step No.	Description of Step	Step Outcome
48	With FSW on EPU1 advanced to Boot Mode, the test continues with the EPU1 Boot Mode Memory Dumps phase. The test script issues the LMEMDUMPMEM, LMEMDUMPPCI, and LMEMDUMPREG telecommands in sequence to EPU1. After each of these telecommands is sent, the script examines the EPU1 boot housekeeping (LBTEPU1HKP) packets, confirming that the current dump command was received and processed (by checking the LPBC1TOTALERRCNT, LPBC1NEXTERRWORD, LPBC1TCRECVCNT, LPBC1TCACCCNT, and LPBC1LASTERRWORD fields) and that the commanded dump was properly executed (by checking the LPBC1LASTFUNC, LPBC1LASTAPID, LPBC1MEMDUMPWC, LPBC1MEMDUMPADDR, LPBC1MEMDUMPDAT** fields). In the Boot Mode, PBC code continuously dumps data from fixed addresses and hence dump is verified by looking for specific packets containing the expected addresses. If all expected data words in expected address ranges have been collected, the test concludes that the Memory Dump operations were successful. If not, then the output log file contains descriptions of why the test failed.	N/A
49	The test script next executes its EPU1 Boot Mode Memory Dump Cancellations phase. The test script re-issues the LMEMDUMPMEM, LMEMDUMPPCI, LMEMDUMPREG in sequence to EPU1. After each command is issued and while it is executing, the script sends the LMEMDUMPCANCEL command to cancel the ongoing dump. While the dumps are in progress and after the LMEMDUMPCANCEL command is sent, the script examines EPU1 boot housekeeping (LBTEPU1HKP) packets, confirming that the cancel command was received and processed (by checking the LPBC1TOTALERRCNT, LPBC1NEXTERRWORD, LPBC1TCRECVCNT, LPBC1TCACCCNT, and LPBC1LASTERRWORD fields) and that the ongoing dump was indeed cancelled (by checking the LPBC1LASTFUNC, LPBC1LASTAPID, LPBC1MEMDUMPWC, LPBC1MEMDUMPADDR, LPBC1MEMDUMPDAT** fields). It is verified that the ongoing dump was cancelled by noticing that the EPU1 boot housekeeping telemetry does not contain memory dumps in the requested address range. Care is taken to ensure that the Requested Dump address ranges are not in the boot diagnostic address range that boot housekeeping telemetry constantly dumps.	N/A

Step No.	Description of Step	Step Outcome
50	The test script next executes its EPU1 Boot Mode Memory Loads phase. Memory load payloads are a few hundred bytes in size and thus span several telecommand packets. First, the test script issues the LMEMDUMPMEM telecommand to EPU1 to establish the baseline contents of selected memory areas. Next, the script issues the LMEMLOADMEM telecommand to EPU1 to load test values to that memory region. The script re-issues the LMEMDUMPMEM telecommand to request a memory dump to confirm that the requested values were loaded to memory. The script then repeats this process for the LMEMDUMPPCI/LMEMLOADPCI/LMEMDUMPPCI and LMEMDUMPREG/LMEMLOADREG/LMEMDUMPREG command sets to confirm that these types of memory loads are also correctly performed. As the telecommands are sent, the script examines EPU1 boot housekeeping (LBTEPU1HKP) packets, confirming that the current memory load command was received and processed (by checking the LPBC1TOTALERRCNT, LPBC1NEXTERRWORD, LPBC1TCRECVCNT, LPBC1TCACCCNT, and LPBC1LASTERRWORD fields) and that the commanded load was properly executed (by checking the LPBC1LASTFUNC, LPBC1LASTAPID, LPBC1TCRECVCNT, LPBC1TCACCCNT, LPBC1TOTALERRCNT fields, and other fields in telemetry). Note: After this Load sub-test has been run, the original values in all the memory locations prior to the test are restored to their original values by reissuing the LOAD commands and verifying that the original values were restored with the DUMP commands.	N/A
51	The test script next executes its EPU1 Boot Mode Multiple Block Commands phase. The script sends an LMEMDUMPMEM command to EPU1 while a memory dump is in progress, and reviews the same fields in boot housekeeping mentioned in the previous steps to confirm that the second LMEMDUMPMEM command packet is rejected. This verifies that interleaved block commands are rejected. Note: The test verifies other aspects of block command handling in Steps 48, 49, and 50 described above. Namely, the test confirms that sequentially transmitted (non-overlapping) MEM DUMP and LOAD telecommands are properly executed and that DUMP/CANCEL/SUMP sequences are also properly processed. In the output log printed and reviewed at the end of the test, entries corresponding to steps 48, 49, 50, and 51 clearly spell out how these various aspects of block command handling are verified.	N/A
52	Next, the test script executes its EPU1 Boot Mode MEM Command Data Format and Integrity Errors phase. The script confirms that EPU1 boot housekeeping telemetry (LBTEPU1HKP) packets also conform to the CCSDS standard. Note: In Steps 48, 49, and 50, the test verifies that all Boot Mode MEM telecommands transmitted to EPU1 in those steps have the proper data format and integrity and conform to the CCSDS standard. In the output log printed and reviewed at the end of the test, entries corresponding to steps 48, 49, 50, and 52 clearly spell out how command and telemetry formatting and integrity are verified	N/A
53	Next, the test script executes its EPU1 Boot Mode MEM Command Format and Parameter Validation phase. Proper execution of telecommands with correct format and valid parameters is tested in Steps 48, 49, and 50. In the output log printed and reviewed at the end of the test, entries clearly spell out how parameter validation is verified during these steps.	N/A

Step No.	Description of Step	Step Outcome
54	Next, the test script executes its EPU1 Boot Mode MEM Command Processing Order phase. The script sends a series of LMEMDUMPMEM commands to EPU1, each command specifying a different address range (for instance, incrementing the address range by 0x100).The script examines the LPBCLASTFUNC and LPBCLASTAPID fields of SIU boot housekeeping telemetry (LBTHKP packets) to confirm that the LMEMDUMPMEM/LMEMDUMPPCI/LMEMDUMPREG/LMEMDUMPSYMREL commands were received and processed in the order in which they were sent. The script also confirms that the address range in dumped memory data telemetry changes in the same order in which those address ranges were requested in the series of LMEMDUMPMEM commands.	N/A
55	Next, the test script executes its EPU1 Boot Mode MEM Command Completion Status phase. In Steps 48, 49, 50, and 51, the test checks that command completion status is reported by FSW, using the telemetry fields described in those steps. In the output log printed and reviewed at the end of the test, entries clearly spell out how command completion status is verified during these steps	N/A

6.4.6 Part 6: EPU1 Application Mode Testing

The test moves on to Application Mode memory management functionality on EPU1. The same fields examined in CmdConfirm, DiagMemStats0, and DiagMemStats1 packets in EPU0 Application Mode testing are examined in EPU1 Application Mode testing.

Step No.	Description of Step	Step Outcome
56	The test script advances EPU1 FSW to Application Mode by sending the LPBCRTOSEXEC telecommand, which commands the EPU to perform a secondary boot. The script checks that EPU1 is operating in Application Mode by detecting whether EPU1 is transmitting Application Mode housekeeping telemetry. If not, the script sends the LPBCRTOSEXEC command and checks again. If the EPU1 FSW cannot be advanced to Application Mode, LTX prints an error message to the screen and exits, aborting the test. If the test is NOT aborted, mark “Complete” for the Step Outcome.	Complete/ Not Complete
57	The test script begins executing its EPU1 Application Mode Memory Dump Commands phase. The test script issues the LMEMDUMPMEM telecommand to EPU1. After the telecommand is sent, the script examines the Command Verification telemetry (CmdConfirm packets), confirming that SUCCESS is reported in the ITC_EXESTATUS field. The script also examines the LMEMEPU1DATA dump data packets (LMEMTEPU1ADDRESS, LMEMTEPU1WORDCNT, LMEMTEPU1CMDFUNC, and LMEMTEPU1DATA* fields) to verify the memory was dumped as requested. The script continues with the rest of the memory dump functionality, again examining the CmdConfirm telemetry packets, and issuing/receiving the following command/telemetry pairs: LMEMDUMPPCI/LMEMEPU1DATA; LMEMDUMPREG/LMEMEPU1DATA; LMEMDUMPSYMVAL/LMEMSYMVAL, LMEMDUMPSYMREL/LMEMEPU1DATA, and finally, LMEMDUMPPOOL/LMEMPOOLDATA.	N/A

Step No.	Description of Step	Step Outcome
58	Next, the script executes its EPU1 Application Mode Memory Dump Cancellation Commands phase. The test script sends the LMEMDUMPMEM telecommand to EPU1. After the telecommand is sent and while it is executing, the script sends the LMEMDUMPCANCEL command to cancel the current dump. The script checks for the SUCCESS status code in the ITC_EXESTATUS field of CmdConfirm telemetry and confirms that LMEMEPU1DATA packets cease to be transmitted, confirming that the LMEMDUMPCANCEL command was successfully executed.	N/A
59	Next, the script executes its EPU1 Application Mode Memory Load Commands phase. Memory load payloads are a few hundred bytes in size and thus span several telecommand packets. The script issues the LMEMDUMPMEM telecommand to EPU1 and examines the LMEMTEPU1ADDRESS, LMEMTEPU1WORDCNT, LMEMTEPU1CMDFUNC, and LMEMTEPU1DATA* fields in the dump telemetry to establish the baseline contents of the corresponding memory areas. Next, the script issues the LMEMLOADMEM telecommand to EPU1 to load test values to memory. The script re-issues the LMEMDUMPMEM telecommand to request a memory dump to confirm that the requested values were loaded to memory. The script then repeats this process for the LMEMDUMPPCI/LMEMLOADPCI/LMEMDUMPPCI and LMEMDUMPREG/LMEMLOADREG/LMEMDUMPREG sets to confirm that the requested memory loads were indeed performed. As the telecommands are sent, the script also examines the EPU1 CmdConfirm telemetry to verify that SUCCESS is reported in the ITC_EXESTATUS field. Note: After this Load sub-test has been run, the original values in all the memory locations prior to the test are restored to their original values by reissuing the LOAD commands and verifying that the original values were restored with the DUMP commands	N/A
60	Next, the script executes its EPU1 Application Mode Multiple Block Commands phase. The script sends an LMEMDUMPMEM command to EPU1 while an LMEMDUMPMEM command is already being executed, and reviews the same fields in telemetry to confirm that the second LMEMDUMPMEM command packet is rejected. Note: The test verifies other aspects of Application Mode memory management block command handling in Steps 57, 58, and 59 described above. Namely, the test confirms that sequentially transmitted (non-overlapping) MEM telecommands are properly executed and that DUMP/CANCEL/DUMP sequences are also properly processed. In the output log printed and reviewed at the end of the test, entries corresponding to steps 57, 58, 59, and 60 clearly spell out how these various aspects of block command handling are verified.	N/A
61	Next, the script executes the EPU1 Application Mode MEM Command and Telemetry Data Format and Integrity Errors phase. Note: In Steps 57, 58, and 59, the test verifies that all Application Mode MEM telecommands transmitted in those steps have the proper data format and integrity by examining the structure of each telecommand packet as it is transmitted, including the packet headers and checksum, and confirming that the packets conform to the CCSDS standard. Those steps also confirm that the EPU1 LMEMSIUDATA, LMEMPOOLDATA, LMEMSYMVAL telemetry packets returned in response to those commands confirm to the CCSDS standard. In the output log printed and reviewed at the end of the test, entries corresponding to steps 57, 58, and 59 clearly spell out how command and telemetry formatting and integrity are verified.	N/A

Step No.	Description of Step	Step Outcome
62	Next the script executes the EPU1 Application Mode MEM Command Format and Parameter Validation phase. Proper execution of telecommands with correct format and valid parameters is tested in Steps 57, 58, and 59. In the output log printed and reviewed at the end of the test, entries clearly spell out how parameter validation is verified during these steps.	N/A
63	Next the script executes the EPU1 Application Mode Command Processing Order phase. The test script sends all Application Mode memory management telecommands (LMEMDUMPMEM, LMEMDUMPCANCEL, LMEMDUMPPCI, LMEMDUMPREG, LMEMLOADMEM, LMEMLOADPCI, LMEMLOADREG, LMEMDUMPPOOL, LMEMDUMPSYMVAL, and LMEMDUMPSYMREL) in sequence to EPU1. After sending each command, the script examines the ITC_DEQTIMHI and ITC_DEQTIMLO fields in CmdConfirm telemetry to confirm that the time at which each command is "dequed" increases in parallel order with the time at which each command was sent.	N/A
64	Next the script executes the EPU1 Application Mode Command Execution Notification phase. In Steps 57, 58, 59, and 60, after sending each command, the script examines the ITC_DEQTIMHI and ITC_DEQTIMLO fields and the ITC_EXETIMHI and ITC_EXETIMLO fields in CmdConfirm telemetry to confirm that FSW reports when each command is "dequed" and executed.	N/A
65	Next the script executes the EPU1 Application Mode Command Completion Status phase. In Steps 57, 58, 59, and 60, the test checks that command completion status is reported by FSW, using the telemetry fields described in those steps as well as the Timestamp field in the CCSDS header of CmdConfirm telemetry. In the output log printed and reviewed at the end of the test, entries clearly spell out how command completion status is verified during these steps.	N/A
66	The test script sends the LPBCRESET command to EPU1 to reboot that unit. It then powers down EPU1 by sending the LIMPOWEROFF command.	N/A
67	The script next sends the LIMLOADSHED telecommand to power off the GASU and PDU.	N/A
68	If the SIU FSW cannot be placed in Boot Mode, LTX prints an error message to the screen and exits. Confirm that the script has placed the SIU in Boot Mode ready for another test.	Complete/ Not Complete

6.4.7 Part 7: Review of Test Outputs

This final part of the test covers detailed review of the MEMMGT script output file and verification of PASS/FAIL criteria.

Step No.	Description of Step	Step Outcome
69	After the MEMMGT test script has executed all parts of the test described in the previous sections, a dialogue box appears, reporting the directory location to which the test log output file is saved and instructing the test engineer to print the file for review. Print the log file, retrieve it from the printer, label it, and attach it to this test procedure. Record the label written on the printed log file: _____ Record the full path and filename of the soft copy of this output file: _____	Complete/ Not Complete
70	Review the log file. The file contains detailed outputs of the test script, with the names of telecommands being issued, telemetry packets being received, telemetry fields being examined, and status summaries of each major step in the test The following log file entries appear interspersed through the log file, followed by "PASS" or "FAIL". **Boot Mode Memory Dumps from SIU, EPU0 and EPU1** **Boot Mode Memory Dump Cancellations from SIU, EPU0 and EPU1** **Boot Mode MEM Load Commands** **Boot Mode Memory Pool Status Dump Command from SIU, EPU0 and EPU1** **Boot Mode Multiple Block commands.** **Boot Mode Memory Load Data** **Boot Mode MEM command Data Format and Integrity Errors** **Boot Mode MEM Command Format and Parameter Validation** **Boot Mode MEM Command Processing order** **Boot Mode MEM Command Completion Status** **Boot Mode Command Rates** **Boot Mode Diagnostics Telemetry Verification** **MEMMGT_001(1): Passed** Confirm that each of the above entries appears in the log file with a "PASS" indication. Successful completion of this step verifies all test objectives and sub-objectives for Boot Mode.	Pass/ Fail

Step No.	Description of Step	Step Outcome
71	Review the log file. The file contains detailed outputs of the test script, with the names of telecommands being issued, telemetry packets being received, telemetry fields being examined, and status summaries of each major step in the test The following log file entries appear interspersed through the log file, followed by "PASS" or "FAIL". **Application Mode Memory Dumps from SIU, EPU0 and EPU1** **Application Mode Memory Dump Cancellations from SIU, EPU0 and EPU1** **Application Mode Memory Pool Status Dump Command from SIU, EPU0 and EPU1** **Application Mode Memory Load Commands** **Application Mode Multiple Block Commands** **Application Mode Memory Load Data** **Application Mode MEM command and MEM Telemetry Data Format and Integrity Errors** **Application Mode MEM Command Format and Parameter Validation** **Application Mode MEM Command Processing order** **Application Mode MEM Command Execution Notification** **Application Command MEM Completion Status** **Application Mode Command Rates** **Application Mode Diagnostics Telemetry Verification** **MEMMGT_001(2): Passed** Confirm that each of the above entries appears in the log file with a "PASS" indication. Successful completion of this step verifies all test objectives and sub-objectives for Application Mode.	Pass/ Fail

Initial to confirm.

_____	_____	_____	_____
Date	Time	Test Engineer	QAE

7. TEST POST_CONDITIONS AND OVERALL OUTCOME

7.1 Test Post-Conditions

The following post-conditions are analyzed and verified by the test script as described in "Test Procedure and Test Analysis":

No.	Post-Condition	Post-Condition Met? (Yes/No)
1	SIU operating in Boot Mode	
2	DAQ, GASU, PDU, and EPUs are powered off.	

The Test Engineer and Quality Assurance Engineer verify that all test post-conditions are met.

_____	_____	_____	_____
Date	Time	Test Engineer	QAE

7.2 Overall Outcome of MEMMGT_001

Based on the analysis of the test results, the overall outcome of Test MEMMGT_001 is as follows:

☐ **Passed** - all of the expected outcomes for the test were confirmed

☐ **Failed** - one or more of the test outcomes were not confirmed

_____	_____	_____
Date	Test Engineer	QAE

8. CERTIFICATION

I certify that the information obtained under this test procedure is as represented and the information recorded in this document is complete and correct. Any deviations from test procedures described herein are identified in Appendix A.

_____	_____	_____
Date	Test Engineer (Print Name)	Test Engineer (Signature)

I certify that the information obtained through execution of this test procedure is as represented and the information recorded in this document is complete and correct. Execution of the test, storage of the results, and verification of outcomes were carried out in accordance with quality standards defined in the GLAST Quality Manual (LAT-MD-00091).

_____	_____	_____
Date	Software QA Engineer (Print Name)	Software QA Engineer (Sign)

I certify that the information obtained under this test procedure is as represented and the information recorded in this document is complete and correct. The test procedure, as designed and executed, does indeed verify that the FSW functionality under test satisfies the corresponding requirements from the Flight Software Specification – Level III.

_____	_____	_____
Date	FSW Manager (Print Name)	FSW Manager (Signature)

APPENDIX A: DEVIATIONS FROM THE QUALIFICATION TEST PROCEDURE

This section details any deviations from the hardware configuration, software configuration, or test procedure followed during the execution of the test or tests described in this Qualification Test Procedure document. All deviations from the approved procedure are agreed to by the Test Engineer and the Software Quality Engineer during the test execution session. All deviations must be reported during the Post Qualification Test Review, where their impact on the test results will be evaluated.

Hardware Deviations

Describe any deviations from the hardware configuration defined in Section 5.1. Name the hardware that was modified and describe the modifications. If hardware is *replaced* during execution of the test, name the replaced hardware, the manufacturer, and list an identification number (e.g., GLAT ID number).

_____

_____

_____

_____

_____

_____

Software Deviations

Describe any changes made to the software configuration under test or the software configuration used to support test execution, as defined in Section 5.2. Give version numbers of all FSW packages and test packages that were modified. Describe how the contents of the modified software load were verified. Describe these deviations for each test that was modified.

_____

_____

_____

_____

_____

_____

Procedural Deviations

Specify any deviations from the test procedure for the test being executed. If this document contains more than one test procedure, list the procedure by number (e.g., "MEMMGT_001"). List by number the steps modified or skipped. Provide a numbered sequence listing any added steps. Describe these deviations for each test that was modified.