GLAST LAT Flight Software Demonstration

SLAC Campus
Building 84, Central Lab Annex

November 3, 2004
Overview Presentation 9:00 AM (Room B-259)
Demonstrations 9:15 AM

1 Demonstration Overview

1.1 Agenda for the Demonstrations

The demonstration will take place in the Central Laboratory Annex (Building 84), Room B-101.

Demo Agenda Item	Presenter(s)
1. Overview of the Demonstration (in Group C Conference Room, Room B-259)	Don May
2. Boot Status Signaling Over Discretes	Don May
3. EPU Boot	Don May
4. Questions from Attendees	NA

Feel free to jot questions and comments down in the margins of this document or in the space provided on page 16.

1.2 Goals of the Demonstration

The October 2004 FSW Demonstration (actually held on November 3) covers two major areas of functionality.

Boot Status Signaling Over Discrete Lines. This demonstration shows that the boot code can use the discrete lines to report detection of errors in the boot region of SDRAM and report any failures to initialize communications.

EPU Boot Process. This demonstration covers the ability of EPU FSW to: (a) provide a controllable boot process (through commanded reboots and commanded entry into secondary boot), (b) to store boot errors and other diagnostic data for use in troubleshooting and make this data available for retrieval on command, (c) to issue boot housekeeping telemetry, and (d) to provide a memory scrubbing capability.

For today, 3 CPU crates and a GASU are used to represent the final flight hardware. One CPU acts as the Spacecraft, the second acts as the SIU, and the third acts as the EPU. See Figure 1 on page 6 for the hardware configuration.

1.3 Verification of Requirements

The following table cites the software requirements to be demonstrated today.

FSW Requirements Demonstrated	Description	Demonstration Procedure
5.3.1.4.3: Event Processor Reboot	[Derived] The EPU FSW shall perform a primary reboot on command from the SIU.	See Section 2.3.3 on page 11.
5.3.1.4.4: Reset Source	[Derived] SIU and EPU FSW shall store the source or cause of a reboot in an error log that can be retrieved after primary boot is complete.	See Section 2.2.2 on page 7, Section 2.3.3 on page 11, and Section 2.3.4 on page 12.
5.3.1.5: Store System Errors	[Derived] SIU and EPU FSW shall store any system errors occurring during primary boot in an error log.	See Section 2.3.2 on page 10.
5.3.1.6: Automatically Send Errors	[Derived] In addition, FSW shall include data in the boot housekeeping telemetry regarding system errors occurring during primary boot.	See Section 2.3.2 on page 10.
5.3.1.7: Retrieve System Errors	[Derived] Upon command, the log of system errors occurring during primary boot shall be sent to the ground via telemetry.	See Section 2.3.2 on page 10.
5.3.1.8: Boot Status	[Derived] SIU FSW shall indicate boot status over the discrete interface lines. The value states of the discrete lines and their meanings are defined in Table 5-1 of FSW Software Requirements Specification (LAT-SS-00399-04).	See Section 2.2.2 on page 7.
5.3.1.9: Boot Housekeeping	[2] (3.2.6.2.2), [4] (5.5.3) During primary boot of a unit, the FSW shall send Boot Housekeeping telemetry to the spacecraft via the CTDB, as defined in [9]. Note that EPUs depend on SIUs to forward this data to the spacecraft.	See Section 2.2.2 on page 7 and Section 2.3.1 on page 9.

FSW Requirements Demonstrated	Description	Demonstration Procedure
5.3.1.10.1: Initiating EPU Secondary Boot	[Derived] At the conclusion of the primary boot, the EPU shall be commandable by the SIU to perform the secondary boot. This command may be issued by the ground and implemented via the SIU.	See Section 2.3.4 on page 12.
5.3.1.12: LAT Memory Scrubbing	[Derived] FSW shall configure memory scrubbing of processor memory. Details may be found in the Appendix of the RAD750 Board Hardware User's Manual.	See Section 2.3.1 on page 9.

2 Demonstration Procedure

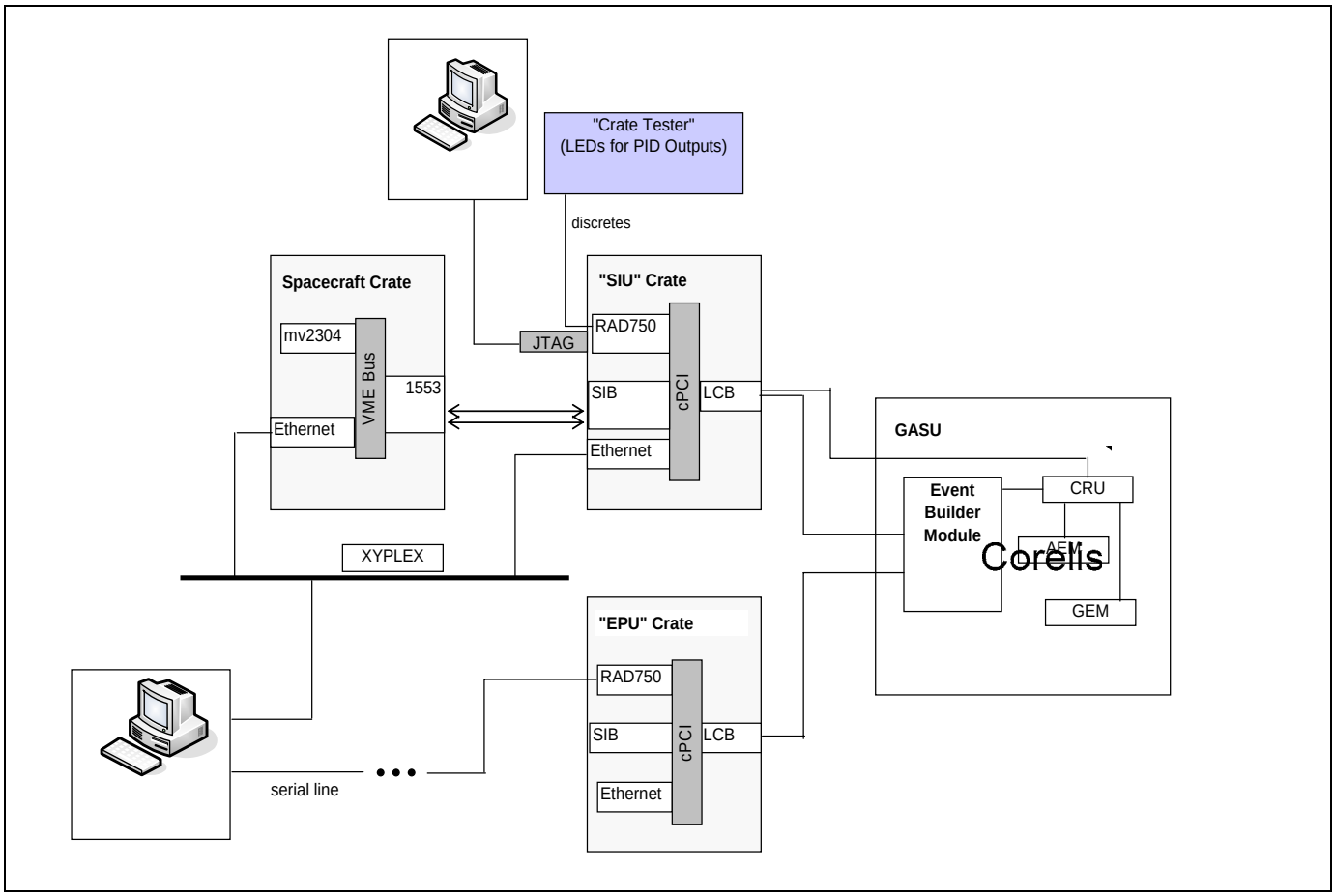
2.1 Context of the Demonstration

2.1.1 Hardware Context

The EPU boot demonstration makes use of four major hardware components set up in the Dataflow laboratory (Room B-101). See Figure 1 below.

- A GASU. The demo GASU contains a complete set of subunits: Command/Response Unit (CRU), ACD Electronics Module (AEM), Global Trigger Electronics Module (GEM), and Event Builder Module (EBM) .
    - The SIU crate and EPU crate (see below) communicate via the EBM.
- A “Spacecraft” CPU crate. This CPU acts the bus controller on the 1553 bus. It is configured as follows:
    - VME chassis
    - mv2304 processor
    - 1553 interface board
    - Ethernet board
- An “SIU” crate. Acts as the remote terminal on the 1553 bus. Configured as follows:
    - cPCI chassis
    - RAD750 processor
    - cPCI LAT Communications Board (LCB), connected to the GASU as node “SIU_0”
    - cPCI Spacecraft Interface Board (SIB), housing the EEPROM and 1553 interface hardware. Connected to Spacecraft crate via the 1553.
    - Ethernet board
- An “EPU” crate. Connected to the SIU crate via the GASU’s EBM. Configured as follows:
    - cPCI chassis
    - RAD750 processor
    - cPCI LCB, connected to the GASU as node “EPU_0”
    - cPCI SIB board, with EEPROM and 1553 interface. The 1553 hardware is not used during the demonstration.

Figure 1: Hardware Setup for November 3, 2004 FSW Demonstration



2.1.2 FSW Context

The following software is loaded on the Spacecraft crate before the demonstration begins:

> CMX/V2-2-2/libcmx_asBuiltSpy.o, PBS/V2-5-0/libpbs.o, MSG/V1-2-2/libmsg_mt.o, MSG/V1-2-2/libmsg_print.o, SIB/V1-2-1/libsib.o, ZLIB/V2-1-0/libzlib_inflate.o, CCSDS/V3-3-0/libccsds_pkt.o, CCSDS/V3-3-0/libccsds_swap.o, CTDB/V5-2-0/libsumt_bc_sib.o, ITC/test/libitc_dump.o, ITC/test/libitc.o, CTS/test/libctx_scp_sumt.o, CTS/test/libcts_scp.o, MEM/3-0-0/libmem_scp.o, PBC/3-0-0/libpbc_scp.o, LSM/V1-0-1/liblsm_dump.o, LSM/V1-0-1/liblsm_scp.o, FILE/test/libfile_path.o, FILE/test/libfile_dump.o, FILE/test/libfile_scp.o, SCP/test/libscp.o

The following software libraries are loaded on the SIU crate before the demonstration begins:

> CMX/V2-2-2/libcmx_asBuiltSpy.o, PBS/V2-5-0/libpbs.o, MSG/V1-2-2/libmsg_mt.o, MSG/V1-2-2/libmsg_print.o, ZLIB/V2-1-0/libzlib_inflate.o, CCSDS/V3-3-0/libccsds_pkt.o, CCSDS/V3-3-0/libccsds_swap.o, CTDB/V5-2-0/libsumt_rt_pmc1553.o, ITC/test/libitc_dump.o, ITC/test/libitc.o, CTS/test/libctx_lcp_sumt.o, CTS/test/libcts_lcp.o, LCBD/V1-2-1/liblcbd.o, LCS/test/liblcsb.o, LCS/test/liblcs.o, MEM/3-0-0/libmem.o, MEM/3-0-0/libmem_scp.o, PBC/3-0-0/libpbc.o, LSM/V1-0-1/liblsm_dump.o, LSM/V1-0-1/liblsm.o, FILE/test/libfile_hdr.o, FILE/test/libfile_sys.o, FILE/libfile_path.o,

FILE/file_upl/libfile_upl.o, FILE/test/libfile_lcp.o, LCB/V4-1-1/libmsg_init.o, LCP/test/liblcp.o

The PBC package is preburned into the EPU SUROM; standalone and Tornado RTOS images are preburned into the SIU and EPU EEPROM.

2.2 Boot Status Signaling Over Discrete Lines Demonstration

Using the hardware and software configuration defined above, the Boot Status demonstration proceeds through the following 2 major steps:

- Step One: Signaling a Hard Error in SDRAM
- Step Two: Signaling Failure to Initialize 1553 Communications

2.2.1 Step 1: Signaling a Hard Error in SDRAM Test

Part One proceeds as follows:

- The demonstrator powers up the Spacecraft crate, then, from the demonstration terminal, opens a VxWorks shell window to the Spacecraft mv2304.
    - From the VxWorks shell, the demonstrator issues the **scp_start** command to initialize and start the Spacecraft Control Program environment.
- The demonstrator powers up the SIU crate. The SIU enters primary boot.
    - As part of the boot process, the Primary Boot Code (PBC) begins to perform SDRAM tests.
- The demonstrator powers up the GASU and EPU.
    - The EPU begins its primary boot process
    - The EPU begins issuing boot telemetry. However, this telemetry is not delivered to the SIU and the Spacecraft crates until a later software initialization step.
- At the demonstration terminal, the demonstrator opens a Corelis application window. Using a debugging tool built into the Corelis application:
    - The demonstrator pauses the SDRAM test.
    - The demonstrator deliberately introduces a memory register mismatch that the PBC memory test code will process as a memory error.
    - The demonstrator resumes the SDRAM test.
- The PBC continues its memory tests, detects the memory error, asserts a signal on PID 6 indicating that a memory error was detected.
    - On the crate tester, the “IN1” LED lights to indicate a signal on PID 6.
    - See Figure 2 on page 14.

2.2.2 Step 2: Signaling Failure to Initialize 1553 Communications

Part Two proceeds as follows:

- Again, using the Corelis application, the demonstrator:
    - Sets a break at a memory location to simulate 1553 driver initialization failure.
- PBC, having failed to detect an operational 1553 communications channel, asserts a signal on PID 5 indicating that 1553 communications are not available.
    - See Figure 3 on page 14.
- The SIU is allowed to complete the primary boot process.
- In the Spacecraft terminal window, the demonstrator points out values reported in the Total Err Cnt (BOOTTOTALERR), Queued Err Cnt (BOOTQERR), Queued Error Word (BOOTERRWORD), and Latest Err Word (BOOTLASTERR) fields of boot housekeeping telemetry packets
    - These fields record codes describing the 1553 communications errors encountered earlier.
    - See Figure 4 on page 15.

Steps 1 and 2 of the Boot Status Signaling Over Discretes demonstration show that FSW provides functionality to satisfy the following requirements from Version 4 of the LAT Flight Software Specification – Level III:

FSW Requirements Demonstrated	Description	Monitor Verification
5.3.1.8: Boot Status	[Derived] SIU FSW shall indicate boot status over the discrete interface lines. The value states of the discrete lines and their meanings are defined in Table 5-1 of FSW Software Requirements Specification (LAT-SS-00399-04).	This requirement was successfully demonstrated _____ Monitor Initials If only partially successful, record deviations on page 16.
5.3.1.9: Boot Housekeeping	[2] (3.2.6.2.2), [4] (5.5.3) During primary boot of a unit, the FSW shall send Boot Housekeeping telemetry to the spacecraft via the CTDB, as defined in [9]. Note that EPUs depend on SIUs to forward this data to the spacecraft.	This requirement was successfully demonstrated _____ Monitor Initials If only partially successful, record deviations on page 16.

2.3 EPU Boot Demonstration

The EPU demonstration proceeds in the following 4 major steps:

- Step 1: Boot the EPU, View EPU Boot Housekeeping, and Observe Memory Scrubbing Progress
- Step 2: Retrieve Errors from EPU Boot Error Log
- Step 3: Reboot the EPU

- Step 4: Initiate EPU Secondary Boot

2.3.1 Step 1: Boot the EPU, View EPU Boot Housekeeping, and Observe Memory Scrubbing Progress

Step 1 of the EPU Boot Demo proceeds as follows:

- From the demonstration terminal, the demonstrator opens a VxWorks shell window to the SIU crate.
- The demonstrator reboots the SIU and waits for SIU boot housekeeping to be displayed in the Spacecraft VxWorks shell window.
- At the Spacecraft VxWorks shell window, the demonstrator issues the **PBC_sendBoot** command (with arguments 0,1,0,0) to boot the SIU RTOS image.
- At the SIU VxWorks shell window, the demonstrator issues the **lcp_start** command (with arguments lat_elf4 rad750 sib) to load FSW libraries to the SIU.
- Again at the SIU VxWorks shell, the demonstrator issues the **lcp_init** command to initialize and start the LAT Control Program (LCP) environment.
    - This command also initializes the GASU and configures the EPU with an address on the LATp communications fabric.
    - Throughout the demo, the EPU boot code has been generating boot housekeeping telemetry. With the LCP software infrastructure in place, at this stage, EPU boot housekeeping telemetry is actually *delivered* to the SIU crate and on to the Spacecraft crate for display.
    - See Figure 4 on page 15
- The demonstrator points out boot housekeeping telemetry packets being sent by the EPU to the SIU crate, which forwards them to the Spacecraft crate.
    - The Boot Type (BOOTTYPE) field in the telemetry specifies that a reset of type ‘Cold Start’ occurred. See Figure 4 on page 15.
- The demonstrator issues the **PBC_setDisplayLevelTlm** command (with an argument of 5) to automatically refresh the display of housekeeping telemetry packets
    - As the display updates, the Scrub Address Hi (BOOTSCRUBADDRHI) field increments to reflect the progress of the memory scrubbing operation

This part of the demonstration shows that FSW provides functionality to satisfy the following requirements from Version 4 of the LAT Flight Software Specification – Level III:

FSW Requirements Demonstrated	Description	Monitor Verification
5.3.1.12: LAT Memory Scrubbing	[Derived] FSW shall configure memory scrubbing of processor memory. Details may be found in the Appendix of the RAD750 Board Hardware User's Manual.	This requirement was successfully demonstrated _____ Monitor Initials If only partially successful, record deviations on page 16.
5.3.1.4.4: Reset Source	[Derived] SIU and EPU FSW shall store the source or cause of a reboot in an error log that can be retrieved after primary boot is complete.	This requirement was successfully demonstrated _____ Monitor Initials If only partially successful, record deviations on page 16.
5.3.1.9: Boot Housekeeping	[2] (3.2.6.2.2), [4] (5.5.3) During primary boot of a unit, the FSW shall send Boot Housekeeping telemetry to the spacecraft via the CTDB, as defined in [9]. Note that EPUs depend on SIUs to forward this data to the spacecraft.	This requirement was successfully demonstrated _____ Monitor Initials If only partially successful, record deviations on page 16.

2.3.2 Step 2: Retrieve Errors from EPU Boot Error Log

The EPU Boot Demo continues to Step 2, which proceeds as follows:

- The demonstrator opens another VxWorks shell window, this time over serial connection to the EPU.
- The demonstrator issues the **PBC_setDisplayLevelTlm** command again (with an argument of 4) to reduce the verbosity of the boot housekeeping telemetry display.
- To deliberately instruct the EPU to look in an invalid location for an RTOS image to load, the demonstrator issues the **PBC_sendBoot** command (with arguments 1,0,0,0), instructing the boot code to look in RAM for the RTOS (no RTOS is stored in RAM).
    - The EPU displays 3 error messages to the EPU VxWorks shell window
    - The EPU also reports this error in the housekeeping telemetry stream. The Total Err Cnt (BOOTTOTALERR) and Queued Err Cnt (BOOTQERR) counts are incremented by 3 and the Queued Error Word (BOOTERRWORD) field shows a code describing the first of the 3 errors stored in the error queue.
    - See Figure 4 on page 15

- The demonstrator issues the **PBC_sendErrDump** command three times to retrieve the error codes one at a time from the error queue.
    - The boot housekeeping telemetry packets displayed in the Spacecraft terminal window change in response to the command to “pop” the error codes from the queue
- To access the errors by dumping the memory region containing the boot error queue, the demonstrator issues the **MEM_sendDump** command (with arguments 1, 0x2ec640, 256) to instruct the EPU to dump 256 words of memory starting at address 0x2ec640.

This part of the demonstration shows that FSW provides functionality to satisfy the following requirements from Version 4 of the LAT Flight Software Specification – Level III:

FSW Requirements Demonstrated	Description	Monitor Verification
5.3.1.5: Store System Errors	[Derived] SIU and EPU FSW shall store any system errors occurring during primary boot in an error log.	This requirement was successfully demonstrated _____ Monitor Initials If only partially successful, record deviations on page 16.
5.3.1.6: Automatically Send Errors	[Derived] In addition, FSW shall include data in the boot housekeeping telemetry regarding system errors occurring during primary boot.	This requirement was successfully demonstrated _____ Monitor Initials If only partially successful, record deviations on page 16.
5.3.1.7: Retrieve System Errors	[Derived] Upon command, the log of system errors occurring during primary boot shall be sent to the ground via telemetry.	This requirement was successfully demonstrated _____ Monitor Initials If only partially successful, record deviations on page 16.

2.3.3 Step 3: Reboot the EPU

The EPU Boot Demo continues to Step 3, which proceeds as follows:

- At the Spacecraft terminal window, the demonstrator issues the **PBC_sendReset** command (with argument 1) to command the EPU to reboot
    - When the EPU reboots, the Boot Type (BOOTTYPE) field in housekeeping telemetry reflects that a reboot of type ‘COMMANDED’ occurred.

- See Figure 4 on page 15.

This part of the demonstration shows that FSW provides functionality to satisfy the following requirements from Version 4 of the LAT Flight Software Specification – Level III:

FSW Requirements Demonstrated	Description	Monitor Verification
5.3.1.4.3: Event Processor Reboot	[Derived] The EPU FSW shall perform a primary reboot on command from the SIU.	This requirement was successfully demonstrated _____ Monitor Initials If only partially successful, record deviations on page 16.
5.3.1.4.4: Reset Source	[Derived] SIU and EPU FSW shall store the source or cause of a reboot in an error log that can be retrieved after primary boot is complete.	This requirement was successfully demonstrated _____ Monitor Initials If only partially successful, record deviations on page 16.

2.3.4 Step 4: Initiate EPU Secondary Boot

The EPU Boot Demo continues to Step 4, which proceeds as follows:

- At the end of Step 3, the EPU has been rebooted and awaits the command to boot the RTOS.
- At the Spacecraft terminal window, the demonstrator issues the **PBC_sendBoot** command (with arguments 1,1,0,0) to command the EPU PBC to boot the RTOS image stored in the lower bank of EEPROM.
    - The progress of RTOS boot is displayed at the EPU VxWorks terminal
- Optionally, the demonstrator reboots the EPU with a keyboard command (“^x”) at the EPU serial console window.
    - Boot housekeeping telemetry reports the reboot type as ‘VXWORKS’.
    - See Figure 4 on page 15.

This part of the demonstration shows that FSW provides functionality to satisfy the following requirements from Version 4 of the LAT Flight Software Specification – Level III:

FSW Requirements Demonstrated	Description	Monitor Verification
5.3.1.10.1: Initiating EPU Secondary Boot	[Derived] At the conclusion of the primary boot, the EPU shall be commandable by the SIU to perform the secondary boot. This command may be issued by the ground and implemented via the SIU.	This requirement was successfully demonstrated _____ Monitor Initials If only partially successful, record deviations on page 16.
5.3.1.4.4: Reset Source	[Derived] SIU and EPU FSW shall store the source or cause of a reboot in an error log that can be retrieved after primary boot is complete.	This requirement was successfully demonstrated _____ Monitor Initials If only partially successful, record deviations on page 16.

3 Outputs of the Demonstration

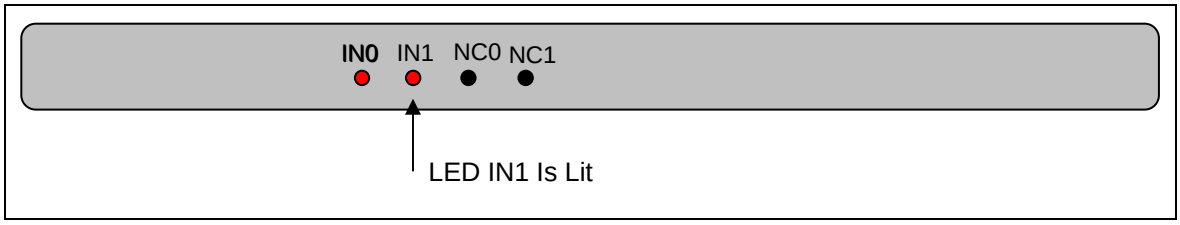
This section contains the screen captures, terminal outputs, and other raw results of the demonstrations run today. These figures are provided to assist monitors in verifying that the FSW successfully demonstrated compliance with the software requirements cited above.

3.1 Outputs of the Boot Status Signaling Over Discrete Lines Demonstration

The following figures show examples of output from Steps 1 and 2 of the Boot Status demonstration.

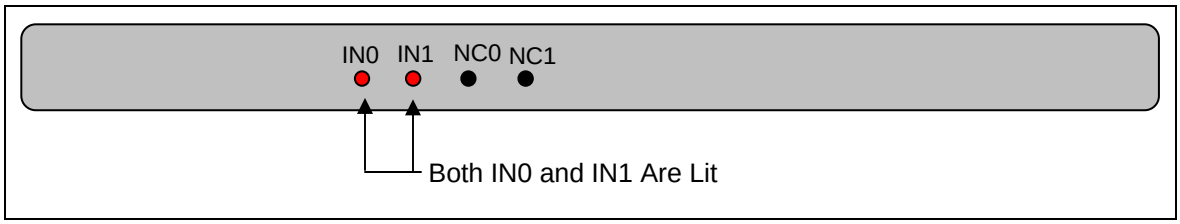
3.1.1 Output of Step 1

Figure 2: Crate Tester LED Settings on Successful Completion of Boot Status Demo, Step 1



3.1.2 Output of Step 2

Figure 3: Crate Tester LED Settings on Successful Completion of Boot Status Demo, Step 2



3.2 Outputs of EPU Boot Demonstration

In Steps 1 through 4 of the EPU Boot demonstration, boot code generates boot housekeeping telemetry. A sample boot housekeeping packet is shown in Figure 4 below. In this figure, fields referred to throughout the demonstration are highlighted.

Figure 4: Boot Housekeeping Telemetry Packet (Fields Referred to During Demonstration Highlighted)

```
*************************************************************************
 CCSDS header (big-endian byte order)
 -------------------------------------------------------------------------
 Offset  Literal    Bits  Value      Description      Interpretation
 ------- ---------- ----- ---------- ---------------- -------------------------
   0-  1 0x0a61      0- 2 0x00       CCSDS version    CCSDS version 1 packet
                     3- 3 0x00       Type             Telemetry
                     4- 4 0x01       Sec. header      Yes
                     5-15 0x0261     APID             609 (decimal)
   2-  3 0xc2ef      0- 1 0x03       Seq. flags       Standalone packet
                     2-15 0x02ef     Seq. count       751 (decimal)
   4-  5 0x006d                      CCSDS length     -> 116 byte total length
   6-  9 0x00000000                  Date/time        2001-01-01 00:00:00
  10- 13 0x00000000                  Microsec         .000000

 =========================================================================
 EPU 0 Boot Housekeeping telemetry (big-endian byte order)
 (Changed values indicated by 'c' in the 'C' column)
 (Use PBC_setDisplayLevelTlm to change the frequency of these telemetry dumps)
 -------------------------------------------------------------------------
 Offset  Literal    C Bits  Value      Description      Interpretation
 ------- ---------- - ----- ---------- ---------------- -------------------------
  14- 15 0x0002                       Boot Mode        Command
  16- 17 0x0000                       Total Err Cnt
  18- 19 0x0000                       Queued Err Cnt
  20- 23 0x00000000                   Queued Err Word
  24- 25 0x0001     c                 Cmd Rcv Cnt
  26- 27 0x0001     c                 Cmd Accept Cnt
  28- 31 0x00000000                   Latest Err Word
  32- 33 0x0640        0- 4 0x00      Last Func Code
                    c  5-15 0x0640    Last APID
  34- 35 0x0000                       Spare 1
  36- 37 0x0000                       Spare 2
  38- 39 0x0000                       File Upl State   Idle
  40- 41 0x0000                       File Upl Pkt Cnt
  42- 43 0x000d     c                 Scrub Address Hi
  44- 45 0x0001                       Boot Type        Cold Start
  46- 47 0x0010                       Dump Word Cnt
  48- 51 0x0000ffc0 c                 Dump Address     Background Dump
*************************************************************************
```

4 Demo Wrapup and Summary

The FSW Team thanks you for attending the demonstration and welcomes any questions about the software and hardware systems showcased. Comments or questions can be written in the following space.

5 Glossary

1553 – MIL-STD-1553B. Serial data bus specification; in particular, the serial data bus and data protocol implemented for the GLAST mission.

APID. CCSDS packet application identifier. A numerical code indicating the general type of data in a CCSDS packet.

Crate. Fond, generic term for development versions of Spacecraft Interface Units (SIUs) or Event-Processor Units (EPUs): custom-built, standalone on-board FSW processors and communications hardware units that control the LAT and communicate with the spacecraft (SIU), and process/filter instrument events (EPU). Crates are used for development purposes and will be replaced in the flight unit with single board computers with the same functionality.

GASU (Global-Trigger/ACD-EM/Signal-Distribution Unit). Portion of the FSW hardware suite that serves as the major hardware interface between data acquisition electronics on the LAT and other hardware and electronics that make up the FSW hardware package. The GASU contains the GEM, EBM, AEM, and CRU.

HKP. Real-time housekeeping telemetry data; telemetry data which relates to the health and safety of the LAT instrument.

LCB – The LAT Communications Board. A cPCI board that allows the internal components of the LAT to communicate with one another.

PID – Programmable Discrete. The RAD750 CPU board contains 32 channels of digital I/O. The primary boot code uses two channels configured as outputs.

PDU (Power Distribution Unit). Portion of the FSW hardware suite that manages power distribution from the spacecraft and monitors the health of other FSW hardware.

RTOS – Real Time Operating System. In particular the VxWorks 5.4 operating system used by the LAT.

SC – The GLAST Spacecraft. As built by Spectrum Astro. Refer to the GLAST LAT Instrument – Spacecraft Interface Control Document for the formal specifications of the SC as seen by the LAT.

SDIS – Spacecraft Data Interface Simulator. The SDIS represents significant portions of the interface between the LAT and the Spacecraft. The SDIS comprises one side of the SC C&DH system with a 1553 bus, LVDS Science Interface, and Discrete monitors and controls.

SDRAM. The RAD750 CPU board 128 MB of synchronous DRAM; the SDRAM serves as the RAD750 main memory.

SIB – Spacecraft Interface Board. The board in the SIU crates that contains the LAT 1553 remote terminal hardware.

SIU – Spacecraft Interface Unit. A type of single board computer (SBC) in the FSW hardware suite that acts as an interface between the spacecraft and the LAT.

SUROM – Startup ROM. 256 KB of EEPROM memory on the RAD750 CPU boards that holds the primary boot code; the SUROM is only programmable on the bench through the PPCI JTAG interface.

T&DF (Trigger and Dataflow System). Large LAT subsystem that provides gamma-ray identification, readout of the detector measurements, assembly of gamma-ray source location and energy measurements, and the streaming of data to the spacecraft. The T&DF subsystem contains the TKR, CAL, and ACD front-end electronics, Tower Electronics Modules (TEMs), ACD Electronics Modules (AEMs), Event Builder Module (EBM), Global Trigger (GLT), Global Trigger Electronics Module (GEM), and CPUs used for instrument configuration and data processing.

VxWorks. Computer operating system used on board the RAD750 processor.