GLAST LAT Flight Software Demonstration

SLAC Campus
Building 84, Central Lab Annex

April 30, 2004
Overview Presentation 10:00 AM (Room B-188)
Demonstrations 10:15 AM

1 Demonstration Overview

1.1 Agenda for the Demonstrations

The demonstration will take place in the Central Laboratory Annex (Building 84).

Demo Agenda Item	Presenter(s)
1. Overview of the Demonstration (in Kavli Conference Room, Room B-188)	Dan Wood
2. Boot Commands Demo	Dan Wood
3. 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 20.

1.2 Goals of the Demonstrations

The Boot Commands Demo will show the following:

• PBC and SBC, the FSW packages responsible for the CPU boot process, are nearly complete and have essentially all of their functionality. ZLIB, FILE, PBC watchdog code, CCSDS, CTDB - the packages that make up the entire low-level infrastructure used by a 1553 bus controller (ultimately, the Spacecraft) to command the boot process, upload files, and retrieve information about the boot process on a flight CPU - are in place.

• 5 out of 5 required boot and reset conditions run correctly: cold boot in response to power on and power on reset signal, cold boot in response to watchdog timeout, warm reboot from a software command, and warm reboot on a secondary boot exception.

• Primary boot can be completed to a point at which unit testing on a RAD750 under flight conditions is possible.

• Data from the boot diagnostics area of crate SDRAM can be captured and used to record errors not otherwise reportable in telemetry.

• The flight crate can be rebooted on a signal from the Spacecraft over a discrete line.

• The boot shell responds to commands that allow selection of different RTOS images and SBC executables. It also responds to boot commands that allow files to up uploaded and committed to EEPROM. A total of 5 boot telecommands will be demonstrated.

• The secondary boot process executes successfully. SBC successfully initializes the RTOS, mounts an EEPROM file system, and defines, installs, and initializes application modules.

1.3 Expected Outputs of the Demonstrations; Verification of Requirements

This section cites the expected outputs of each of the systems tested today, and provides a space for the demonstration monitors to verify by signature that the demonstration ran successfully.

Demo Agenda Item	Relevant FSW Requirements (SRS Version 3)	Expected Output of the Demonstration
Boot Commands Demonstration	5.3.1.1, 5.3.1.2, 5.3.1.3, 5.3.1.4.1, 5.3.1.4.2, 5.3.1.4.4, 5.3.1.10.2, 5.3.1.11, 5.3.7.9, 5.3.7.10	For a full description of the outputs of the Boot Commands Demo, see Section 1.3.2 starting on page 4. For the demonstration procedure and the demo's software and hardware context, see Section 2.1 on page 15.
		Monitor Signature: _____

1.3.1 Relevant CPU Boot Functionality and Requirements to be Addressed in Future Demonstrations

1.3.1.1 Boot Commands and Telemetry

5 boot commands will be demonstrated today, as described in detail later in this document. The following boot commands, identified by APID and function code, will be demonstrated at a future date:

APID	Function Code	Description
0x640	0	Boot Start
	1	Boot Reset
	2	Boot Error Dump
0x642	0	Memory Write
	1	PCI Device Header Write
	2	Processor Register Write
0x644	0	Memory Dump Data
	1	Memory Dump Cancel
	5	PCI Device Header Dump

APID	Function Code	Description
	6	Processor Register Dump

Currently, boot housekeeping telemetry can be displayed using AstroRT, but cannot be conveniently formatted and displayed at a terminal window. Future followup demonstrations of boot telemetry can be conducted, making use of a SIIS and AstroRT software to be provided by Spectrum Astro for use at SLAC in building and testing the ISIS deliverable. This hardware and software is expected to be available by June.

1.3.2 Output of the Boot Commands Demo

This demonstration is conducted in four parts:

- Cold Boot and Testing Using LTX
- Warm Reboot, Execute Secondary Boot Code, Initialize Application Modules, and Upload Files
- Warm Reboot on SBC Exception
- Cold Reboot on Spacecraft Command Through Discrete Line

Outputs for these parts are described in the following sections.

1.3.2.1 Part One: Cold Boot and Testing using LTX

Part One of the demonstration results in three types of output: boot diagnostics, a log of RTOS image extraction and checksum, and LTX test reports.

1.3.2.1.1 Boot Diagnostic Information

During primary boot, the PBC package returns five categories of boot information:

- Data stored in the boot diagnostics region of SDRAM: a mnemonic for the type of boot (in this case COLD), primary boot flags, secondary boot flags (empty on cold boot), memory test results, exception counts, secondary boot error codes (empty on cold boot).

```
    boot type = COLD (1)

       Primary Boot Flags: 0x20100000
     Secondary Boot Flags: 0x00000000

                 BIST 750: 0xFFF00000
          BIST PPCI MISR0: 0x82F10CB6
          BIST PPCI MISR1: 0x68A3EA9C

         Mem Test Results: 0x111F0000
         Mem Test Address: 0x00800000
          Mem Test Actual: 0x007FFFFC
    Post Mem Test Results: 0x00000000
    Post Mem Test Address: 0x00000000
     Post Mem Test Actual: 0x00000000
          MemBank1Sparing: 0x0000FF00
```

```
                 Exc Count: 0x00000000
               Exc Vector: 0x00000000
                 Exc SRR0: 0x00000000
                 Exc SRR1: 0x00000000
                  Exc DAR: 0x00000000
                Exc DSISR: 0x00000000
         Exc PCI Status 2: 0x00000000
        Exc Memory Status: 0x00000000
               Exc TASKID: 0x00000000

              App Info  0: 0x00000000
              App Info  1: 0x00000000
              App Info  2: 0x00000000
              App Info  3: 0x00000000
              App Info  4: 0x00000000
              App Info  5: 0x00000000
              App Info  6: 0x00000000
              App Info  7: 0x00000000

      Sec Boot Error Code: 0x00000000
     Sec Boot Error Index: 0x00000000
      Sec Boot Error Data: 0x00000000
```

- PBC package version information. This information will be reformatted as a single string and can be sent back as telemetry in the flight version.

```
    BUILD_SITE: nrlgoose
    BUILD_USER: dmay
    BUILD_TIME: Fri Apr 16 17:54:15 EDT 2004
    BUILD_DIR:  /data/glast/flight/OS/source/PBC/V1-1-0
    BUILD_LIB:  /data/glast/flight/SVC/binary/FILE/V1-0-
1/rad750/file_hdr_boot/libfile_hdr_boot.ro
    BUILD_LIB:  /data/glast/flight/SVC/binary/FILE/V1-0-
1/rad750/file_upl_boot/libfile_upl_boot.ro
    BUILD_LIB:  /data/glast/flight/SVC/binary/CCSDS/V3-2-
1/rad750/ccsds_pkt_boot/libccsds_pkt_boot.ro
    BUILD_LIB:  /data/glast/flight/SVC/binary/ZLIB/V2-0-
0/rad750/zlib_inflate_boot/libzlib_inflate_boot.r
    o
    BUILD_LIB:  /data/glast/flight/SVC/binary/MEM/V1-0-
0/rad750/memboot/libmemboot.ro
    BUILD_LIB:  /data/glast/flight/DRV/binary/CTDB/V4-0-
2/rad750/sumt_rt_poll_sib_boot/libsumt_rt_poll_si
    b_boot.ro
    BUILD_LIB:  /data/glast/flight/SVC/binary/PBS/V2-2-
0/rad750/pbs_boot/libpbs_boot.ro
```

- PBC 1553 driver configuration information and 1553 communications parameters.

```
    ------------------------------------------------------------
    Driver: sumt_rt_poll_sib
    Board: LAT SIB
    PCI Device ID: 0844
    Hardware Revision: 3
    PCI Bus: 0
    PCI Device: 30
```

```
PCI Function: 0
PCI Hardware Summit Registers Address: 00400000
PCI Logical Summit Registers Address: c0400000
PCI Hardware Shared Memory Address: 00600000
PCI Logical Shared Memory Address: c0600000
Shared Memory Size: 131072
SIB Control Register: 00000000
SIB Status Register: 00000002
------------------------------------------------------------
------------------------------------------------------------
RT Address: 3
CmdRx Subaddress: 27
CmdTx Subaddress: 29
Telem Base Subaddress: 11
Telem Message Count: 15
Wrap Subaddress: 30

RT Descriptor Table Base Address: c0600000
CmdRx Descriptor Address: c06001b0
Telem Complete Descriptor Address: c06001a0

Data Buffer Base Address: c0600800
CmdRx Data Address: c0601660
Telem Data 0 Address: c0601ee0
Telem Data 1 Address: c0601f68
Telem Data 2 Address: c0601ff0
Telem Data 3 Address: c0602078
Telem Data 4 Address: c0602100
Telem Data 5 Address: c0602188
Telem Data 6 Address: c0602210
Telem Data 7 Address: c0602298
Telem Data 8 Address: c0602320
Telem Data 9 Address: c06023a8
Telem Data 10 Address: c0602430
Telem Data 11 Address: c06024b8
Telem Data 12 Address: c0602540
Telem Data 13 Address: c06025c8
Telem Data 14 Address: c0602650
```

- Register locations, followed by a notice that the boot shell is up and running.

```
Control Register: 1800
Status Register: 1d02
Command Register: 0000
IRQ Mask Register: cc80
IRQ Pending Register: 0000
IRQ Log Register: 0000
BIT Word Register: 0000
Descriptor Register: 0000
1553 Status Register: 0000
Illegalization Register 0: ffff
Illegalization Register 1: b3ff
Illegalization Register 2: 07ff
Illegalization Register 3: 9c00
Illegalization Register 4: ffff
```

```
    Illegalization Register 5: ffff
    Illegalization Register 6: ffff
    Illegalization Register 7: ffff
    Illegalization Register 8: ffff
    Illegalization Register 9: ffff
    Illegalization Register 10: fecb
    Illegalization Register 11: ffff
    Illegalization Register 12: ffff
    Illegalization Register 13: ffff
    Illegalization Register 14: ffff
    Illegalization Register 15: ffff
    ------------------------------------------------------------
    boot version 0x04161754 (4/16 17:54) running
```

- Once primary boot is complete, PBC issues housekeeping telemetry packets (4 per second). Since boot code has been written to output these packets in AstroRT format, the packets can only be shown as hex data at a terminal window. Therefore, the telemetry data is not displayed in today's demo.

1.3.2.1.2 RTOS Extraction and Installation Log

During Part One of the demo, when the tester issues the **bc_rtos_start** command, PBC records and reports the process of extracting and checksumming the vxw_tornado RTOS image.

```
    Bank Header @ 0xc0800000
    0x00000000: 0x2243012F 0x00000200 0x000DFE00 0x000E0000
    0x00000004: 0x00010000 0x000F0000 0x00010000 0x00000100
    0x00000008: 0x00000100 0x00000000 0x00000000 0x00000000
    0x0000000C: 0x00000000 0x00000000 0x00000000 0x00000000
    0x00000010: 0x00000000 0x00000000 0x00000000 0x00000000
    0x00000014: 0x00000000 0x00000000 0x00000000 0x00000000
    0x00000018: 0x00000000 0x00000000 0x00000000 0x00000000
    0x0000001C: 0x00000000 0x00000000 0x00000000 0x00000000
    0x00000020: 0x00000000 0x00000000 0x00000000 0x00000000
    0x00000024: 0x00000000 0x00000000 0x00000000 0x00000000
    0x00000028: 0x00000000 0x00000000 0x00000000 0x00000000
    0x0000002C: 0x00000000 0x00000000 0x00000000 0x00000000
    0x00000030: 0x00000000 0x00000000 0x00000000 0x00000000
    0x00000034: 0x00000000 0x00000000 0x00000000 0x00000000
    0x00000038: 0x00000000 0x00000000 0x00000000 0x00000000
    0x0000003C: 0x00000000 0x00000000 0x00000000 0x00000000
    Bank Header @ 0xc0800000
    0x00000000: 0x2243012F 0x00000200 0x000DFE00 0x000E0000
    0x00000004: 0x00010000 0x000F0000 0x00010000 0x00000100
    0x00000008: 0x00000100 0x00000000 0x00000000 0x00000000
    0x0000000C: 0x00000000 0x00000000 0x00000000 0x00000000
    0x00000010: 0x00000000 0x00000000 0x00000000 0x00000000
    0x00000014: 0x00000000 0x00000000 0x00000000 0x00000000
    0x00000018: 0x00000000 0x00000000 0x00000000 0x00000000
    0x0000001C: 0x00000000 0x00000000 0x00000000 0x00000000
    0x00000020: 0x00000000 0x00000000 0x00000000 0x00000000
    0x00000024: 0x00000000 0x00000000 0x00000000 0x00000000
    0x00000028: 0x00000000 0x00000000 0x00000000 0x00000000
    0x0000002C: 0x00000000 0x00000000 0x00000000 0x00000000
    0x00000030: 0x00000000 0x00000000 0x00000000 0x00000000
```

```
0x00000034: 0x00000000 0x00000000 0x00000000 0x00000000
0x00000038: 0x00000000 0x00000000 0x00000000 0x00000000
0x0000003C: 0x00000000 0x00000000 0x00000000 0x00000000
File Header @ 0xc0800200
0x00000000: 0xDCFC0CE3 0x501CFFFF 0xFFFFFFFF 0x154236F9
0x00000004: 0x0005FB6A 0x063F0DCD 0x56585752 0x544F5320
File Header @ 0x00200000
0x00000000: 0xDCFC0CE3 0x501CFFFF 0xFFFFFFFF 0x154236F9
0x00000004: 0x0005FB6A 0x063F0DCD 0x56585752 0x544F5320
compressed data @ 0x  200024, uncompressed length is 0x   E30B0
Comp Data @ 0x00200024
0x00000000: 0x78DAAC7D 0x7B7C54D5 0xB5FF3E99 0x493243A2
0x00000004: 0x4E346290 0x402612BD 0x898C7450 0x6C27E4C1
0x00000008: 0xA4623B18 0x6CE9BDC3 0x0DB670EF 0xE49750C9
0x0000000C: 0x455BAC70 0x3BB46798 0x3310DA49 0xD576D210
0x00000010: 0x1A14347C 0x4AB86045 0x41E8FD69 0xCB23C8A3
0x00000014: 0x70457F58 0x1F3F5494 0x70C10A45 0x2C5E45B1
0x00000018: 0x40CEEFBB 0xF6599339 0x339307F8 0xF38FF9EC
0x0000001C: 0xB3BFFB71 0xD65E6BED 0xB5D77E9C 0x3DEA0F46
RTOS @ 0x00030000
0x00000000: 0x7C6E1B78 0x7C631A78 0x7C600124 0x7C000278
0x00000004: 0x7C1043A6 0x7C1143A6 0x7C1243A6 0x7C1343A6
0x00000008: 0x70630000 0x60632000 0x7C0004AC 0x7C600124
0x0000000C: 0x4C00012C 0xFF80010C 0xFF00010C 0xFE80010C
0x00000010: 0xFE00010C 0xFD80010C 0xFD00010C 0xFC80010C
0x00000014: 0xFC00010C 0x4C00012C 0x48000009 0x3F800000
0x00000018: 0x7C6802A6 0xC0030000 0xC0230000 0xC0430000
0x0000001C: 0xC0630000 0xC0830000 0xC0A30000 0xC0C30000
SIB board found: bus=0, dev=30, id=0844
Target Name: vxTarget
User: target
Attached TCP/IP interface to RX unit 0
Attaching network interface lo0... done.


                VxWorks

Copyright 1984-2002  Wind River Systems, Inc.

            CPU: BAE SYSTEMS - RAD750
   Runtime Name: VxWorks
Runtime Version: 5.5
    BSP version: 1.0/0
        Created: Apr 20 2004, 14:24:13
  WDB Comm Type: WDB_COMM_END
            WDB: Ready.
```

1.3.2.1.3 LTX Test Reports

Finally, Part One of the demo makes use of the LTX test suite to run ZLIB through its paces. Refer to the LTX documentation and the demo guidebook for the February 2004 demonstration for examples of LTX output.

1.3.2.2 Part Two: Warm Reboot, Execute Secondary Boot Code, Initialize Application Modules, and Upload Files

Part Two results in three types of output: a log of the secondary boot process (including SBC code inflation and checksum, EEPROM file system mounting and verification, and application module installation and initialization), a log recording results of the file upload process, and directory listings for the EEPROM file system.

1.3.2.2.1 Secondary Boot Log

During Part Two of the demo, when the tester issues the **bc_rtos_start** command, PBC records and reports the process of extracting and checksumming the vxw_standalone RTOS image.

```
    Bank Header @ 0xc0c00000
    0x00000000: 0x2243012F 0x00000200 0x000DFE00 0x000E0000
    0x00000004: 0x00010000 0x000F0000 0x00010000 0x00000100
    0x00000008: 0x00000100 0x00000000 0x00000000 0x00000000
    0x0000000C: 0x00000000 0x00000000 0x00000000 0x00000000
    0x00000010: 0x00000000 0x00000000 0x00000000 0x00000000
    0x00000014: 0x00000000 0x00000000 0x00000000 0x00000000
    0x00000018: 0x00000000 0x00000000 0x00000000 0x00000000
    0x0000001C: 0x00000000 0x00000000 0x00000000 0x00000000
    0x00000020: 0x00000000 0x00000000 0x00000000 0x00000000
    0x00000024: 0x00000000 0x00000000 0x00000000 0x00000000
    0x00000028: 0x00000000 0x00000000 0x00000000 0x00000000
    0x0000002C: 0x00000000 0x00000000 0x00000000 0x00000000
    0x00000030: 0x00000000 0x00000000 0x00000000 0x00000000
    0x00000034: 0x00000000 0x00000000 0x00000000 0x00000000
    0x00000038: 0x00000000 0x00000000 0x00000000 0x00000000
    0x0000003C: 0x00000000 0x00000000 0x00000000 0x00000000
    Bank Header @ 0xc0c00000
    0x00000000: 0x2243012F 0x00000200 0x000DFE00 0x000E0000
    0x00000004: 0x00010000 0x000F0000 0x00010000 0x00000100
    0x00000008: 0x00000100 0x00000000 0x00000000 0x00000000
    0x0000000C: 0x00000000 0x00000000 0x00000000 0x00000000
    0x00000010: 0x00000000 0x00000000 0x00000000 0x00000000
    0x00000014: 0x00000000 0x00000000 0x00000000 0x00000000
    0x00000018: 0x00000000 0x00000000 0x00000000 0x00000000
    0x0000001C: 0x00000000 0x00000000 0x00000000 0x00000000
    0x00000020: 0x00000000 0x00000000 0x00000000 0x00000000
    0x00000024: 0x00000000 0x00000000 0x00000000 0x00000000
    0x00000028: 0x00000000 0x00000000 0x00000000 0x00000000
    0x0000002C: 0x00000000 0x00000000 0x00000000 0x00000000
    0x00000030: 0x00000000 0x00000000 0x00000000 0x00000000
    0x00000034: 0x00000000 0x00000000 0x00000000 0x00000000
    0x00000038: 0x00000000 0x00000000 0x00000000 0x00000000
    0x0000003C: 0x00000000 0x00000000 0x00000000 0x00000000
    File Header @ 0xc0c00200
    0x00000000: 0xF5CA0E69 0x501CFFFF 0xFFFFFFFF 0xA3ADE088
    0x00000004: 0x00042ED6 0x0639CEC8 0x56585752 0x544F5320
    File Header @ 0x00200000
    0x00000000: 0xF5CA0E69 0x501CFFFF 0xFFFFFFFF 0xA3ADE088
    0x00000004: 0x00042ED6 0x0639CEC8 0x56585752 0x544F5320
    compressed data @ 0x  200024, uncompressed length is 0x   9CCA0
```

```
    Comp Data @ 0x00200024
    0x00000000: 0x78DAAC7D 0x7B7854D5 0xB9F7DA99 0x49989080
    0x00000004: 0x83460C26 0x2113899E 0x44461A2B 0xF64CC885
    0x00000008: 0x49C57630 0xD8E239C3 0x09B670CE 0xE424B6A4
    0x0000000C: 0x688B1574 0xD03D993D 0x10EAA4EA 0xF72427FA
    0x00000010: 0x7DD08A86 0xA7C423A7 0xDA07BFD2 0xEFF10624
    0x00000014: 0xDC0A1230 0x2A7A40B9 0x24075428 0xA2F12815
    0x00000018: 0x0A647FBF 0x77ED7766 0xF6CCE402 0x1CFE9867
    0x0000001C: 0xEDFD5B97 0xFDDED6BB 0xDE75D97B 0xD49FE7F8
    RTOS @ 0x00030000
    0x00000000: 0x7C6E1B78 0x7C631A78 0x7C600124 0x7C000278
    0x00000004: 0x7C1043A6 0x7C1143A6 0x7C1243A6 0x7C1343A6
    0x00000008: 0x70630000 0x60632000 0x7C0004AC 0x7C600124
    0x0000000C: 0x4C00012C 0xFF80010C 0xFF00010C 0xFE80010C
    0x00000010: 0xFE00010C 0xFD80010C 0xFD00010C 0xFC80010C
    0x00000014: 0xFC00010C 0x4C00012C 0x48000009 0x3F800000
    0x00000018: 0x7C6802A6 0xC0030000 0xC0230000 0xC0430000
    0x0000001C: 0xC0630000 0xC0830000 0xC0A30000 0xC0C30000
```

- Log of TFFS EEPROM file system mounting and validation.

```
    Retrieved old volume params with %75 confidence:
    Volume Parameters: FAT type: FAT16, sectors per cluster 110
      0 FAT copies, 0 clusters, 5246 sectors per FAT
      Sectors reserved 3198, hidden 478047744, FAT sectors 0
      Root dir entries 32366, sysId (null)  , serial number 7e6e0024
      Label:"        0  " ...
    Disk with 2048 sectors of 512 bytes will be formatted with:
    Volume Parameters: FAT type: FAT12, sectors per cluster 1
      2 FAT copies, 2028 clusters, 6 sectors per FAT
      Sectors reserved 1, hidden 0, FAT sectors 12
      Root dir entries 112, sysId VXDOS12 , serial number 7e6e0024
      Label:"           " ...




    /ee0/  - disk check in progress ...
    dosChkLib : CLOCK_REALTIME is being reset to THU DEC 27 00:05:18
1990
    Value obtained from file system volume descriptor pointer: 0x7ea5f0
    The old setting was THU JAN 01 00:00:00 1970
    Accepted system dates are greater than THU DEC 27 00:00:00 1990
    /ee0/  - Volume is OK

              total # of clusters:  3,679
               # of free clusters:  3,542
                # of bad clusters:  0
                  total free space:  1,771 Kb
         max contiguous free space:  1,813,504 bytes
                       # of files:  11
                     # of folders:  16
                total bytes in files:  56,886
                    # of lost chains:  0
           total bytes in lost chains:  0
```

```
/ee1/  - disk check in progress ...
/ee1/  - Volume is OK

          total # of clusters:  3,679
           # of free clusters:  3,659
            # of bad clusters:  0
              total free space:  1,829 Kb
     max contiguous free space:  1,873,408 bytes
                   # of files:  2
                 # of folders:  16
            total bytes in files:  472
               # of lost chains:  0
   total bytes in lost chains:  0
```

- Log of application module inflation and checksum.

```
SBC database: size=204 checksum=583019c0 compress=0000
Header word: 53424301
Version: 1

11 modules
MODULE id=40400000
Loading file /ee0/d001/f0000000 (dev=2 dir=1 file=0)
module: size=1986 checksum=65d415c0 compress=1000
module inflation size: 5292 bytes
MODULE id=40400001
Loading file /ee0/d001/f0000001 (dev=2 dir=1 file=1)
module: size=15109 checksum=c13a6658 compress=1000
module inflation size: 39100 bytes
MODULE id=40400002
Loading file /ee0/d001/f0000002 (dev=2 dir=1 file=2)
module: size=6309 checksum=c6ced85c compress=1000
module inflation size: 13492 bytes
MODULE id=40400003
Loading file /ee0/d001/f0000003 (dev=2 dir=1 file=3)
module: size=1672 checksum=c93c4cba compress=1000
module inflation size: 4240 bytes
MODULE id=40400004
Loading file /ee0/d001/f0000004 (dev=2 dir=1 file=4)
module: size=3097 checksum=73c250f9 compress=1000
module inflation size: 9336 bytes
MODULE id=40400005
Loading file /ee0/d001/f0000005 (dev=2 dir=1 file=5)
module: size=5613 checksum=a0615f43 compress=1000
module inflation size: 15028 bytes
MODULE id=40400006
Loading file /ee0/d001/f0000006 (dev=2 dir=1 file=6)
module: size=9871 checksum=fe4c436e compress=1000
module inflation size: 27064 bytes
MODULE id=40400007
Loading file /ee0/d001/f0000007 (dev=2 dir=1 file=7)
module: size=3620 checksum=baa06f95 compress=1000
module inflation size: 8900 bytes
MODULE id=40400008
Loading file /ee0/d001/f0000008 (dev=2 dir=1 file=8)
```

```
module: size=2244 checksum=c9c18d03 compress=1000
module inflation size: 5736 bytes
MODULE id=40400009
Loading file /ee0/d001/f0000009 (dev=2 dir=1 file=9)
module: size=2845 checksum=b480b46f compress=1000
module inflation size: 7540 bytes
MODULE id=4040000a
Loading file /ee0/d001/f0000010 (dev=2 dir=1 file=10)
module: size=4168 checksum=82c48db0 compress=1000
module inflation size: 10528 bytes
```

- Record of application module initialization.

```
5 functions
FUNCTION CMX_asBuiltPrint p0=00000000 p1=00000000 p2=00000000 p3=00000000
Calling function CMX_asBuiltPrint with parameters p0=0 p1=0 p2=0 p3=0

Package     Constituent     CVS tag     Built by  Built at  Build Time (UTC)
----------- --------------- ----------- --------- --------- ------------------
CCSDS       ccsds_pkt       V3-2-1      kiml      slac      2004-04-09 22:50:02
CMX         cmx_asBuilt     <test>      kiml      slac      2004-03-19 18:24:28
            cmx_asBuiltSpy  V2-0-8      kiml      slac      2004-03-19 18:34:40
CTDB        co1553_rt       V4-0-2      kiml      slac      2004-04-09 23:01:44
            sumt_rt_sib     V4-0-2      kiml      slac      2004-04-09 23:01:44
FILE        file_hdr_rtos   V1-0-1      kiml      slac      2004-04-13 17:44:26
            file_path       V1-0-1      kiml      slac      2004-04-13 17:44:26
            file_sys        V1-0-1      kiml      slac      2004-04-13 17:44:26
            file_upl        V1-0-1      kiml      slac      2004-04-13 17:44:26
LFS         lfs_rt          V0-0-2      kiml      slac      2004-04-15 17:24:13
MSG         msg_mt          V1-1-4      apw       slac      2004-04-01 05:57:09
            msg_print       V1-1-4      apw       slac      2004-04-01 05:57:09
PBS         pbs             V2-2-0      russell   slac      2004-03-25 00:06:11
SBC         sbc_nominal     <dev>       dwood     slac      2004-04-27 16:03:30
VXW         vxw_standalone  V5-0-1      kiml      slac      2004-04-20 21:23:44
ZLIB        zlib_inflate    V2-0-0      kiml      slac      2004-01-07 22:08:00

FUNCTION rt_init p0=00100000 p1=00000000 p2=00000000 p3=00000000
Calling function rt_init with parameters p0=1048576 p1=0 p2=0 p3=0
MBA_initialize
  Partition Begin: 0x00800000
             Size: 0x07800000
               Id: 0x006c5f90
PBS initialization
------------------
Number of wut tmrs: 256
Keepalive   period: 20000000
Update      period: 1000000000
FUNCTION rt_start p0=00000000 p1=00000000 p2=00000000 p3=00000000
Calling function rt_start with parameters p0=0 p1=0 p2=0 p3=0


FUNCTION upl_info p0=00000000 p1=00000000 p2=00000000 p3=00000000
Calling function upl_info with parameters p0=0 p1=0 p2=0 p3=0
Current state: START
Current error: FILE_SUCCESS
```

```
Error count: 0
Expected file size: 0 bytes
Current file receieved: 0 bytes
Current data packet count: 0
Current data offset: 00000000
File buffer address: 07ef6b40
File commit size: 0
FUNCTION rt_info p0=00000000 p1=00000000 p2=00000000 p3=00000000


Calling function rt_info with parameters p0=0 p1=0 p2=0 p3=0
RT error count: 0
CmdRx packet count: 0
CmdRx byte count: 0
CmdTx packet count: 0
CmdTx byte count: 0
HKP packet count: 0
HKP byte count: 0
Telem packet count: 0
Telem byte count: 0
```

1.3.2.2.2 Record of File Upload Process; Directory and File Listings

When the tester uses the **bc_upload_file** and **bc_upl_cancel** commands to load a file from the Spacecraft crate to a TFFS EEPROM directory on the flight crate, the following type of record is generated:

```
FILE_uplPktStart FILE-I-IUPLSTAR File upload start command received 240
FILE_uplPktData FILE-I-IUPLDATA File upload data command received 00000000
FILE_uplPktData FILE-I-IUPLDATA File upload data command received 00000030
FILE_uplPktData FILE-I-IUPLDATA File upload data command received 00000060
FILE_uplPktData FILE-I-IUPLDATA File upload data command received 00000090
FILE_uplPktData FILE-I-IUPLDATA File upload data command received 000000c0
Commit TC: path=/ee0/d002/f0000005
FILE_uplPktCommit FILE-I-IUPLCOMM File upload commit 40800005
File write complete
FILE_uplReset FILE-I-IUPLREST File upload reset 2
```

Directory and file listings are performed with Unix commands to verify that files were transferred.

```
-> ll
drwxrwxrwx  1 0       0              512 Dec 27 00:10 ./
drwxrwxrwx  1 0       0              512 Jan  1  1970 ../
-rwxrwxrwx  1 0       0              236 Dec 27 00:10 f0000005
```

1.3.2.3 Part Three: Warm Reboot, Generate SBC Exception

The goal of Part Three is to show that an exception during secondary boot leads to a warm reboot, with the reason for reboot and applicable secondary boot errors recorded in the boot diagnostics area of SDRAM. Thus, the output of Part Three is the boot diagnostics log reported by PBC. Only the values drawn from the boot diagnostics area are shown in the figure below. Note the boot type (EXCEPTION) and the errors recorded in the exception and sec boot lines.

```
Adding 765 symbols for standalone.
Generating an exception at address e0000000 in 10 seconds
```

```
    Entry address SBC_init() = 007f1c60

     boot type = EXCEPTION (4)

       Primary Boot Flags: 0x00008000
     Secondary Boot Flags: 0x50000000

                 BIST 750: 0xFFF00000
          BIST PPCI MISR0: 0x82F10CB6
          BIST PPCI MISR1: 0x68A3EA9C

         Mem Test Results: 0xFF0F0000
         Mem Test Address: 0x00000000
          Mem Test Actual: 0x00000000
    Post Mem Test Results: 0x00000000
    Post Mem Test Address: 0x00000000
     Post Mem Test Actual: 0x00000000
          MemBank1Sparing: 0x0000FF00

                Exc Count: 0x00000000
               Exc Vector: 0x00000300
                 Exc SRR0: 0x007F1CB0
                 Exc SRR1: 0x0000B030
                  Exc DAR: 0xE0000000
                Exc DSISR: 0x40000000
         Exc PCI Status 2: 0x22000000
       Exc Memory Status: 0x00000000
              Exc TASKID: 0x007FFDF0

              App Info  0: 0x00000000
              App Info  1: 0x00000000
              App Info  2: 0x00000000
              App Info  3: 0x00000000
              App Info  4: 0x00000000
              App Info  5: 0x00000000
              App Info  6: 0x00000000
              App Info  7: 0x00000000

      Sec Boot Error Code: 0x39862467
     Sec Boot Error Index: 0x398265DE
      Sec Boot Error Data: 0x00000000
```

1.3.2.4 Part Four: Cold Reboot on Spacecraft Command Via Discrete Line

From the command prompt, the tester instructs the Spacecraft crate to issue a cold boot command through a discrete line. The flight crate reboots, and PBC generates log information very similar to that shown in Part One of the demonstration. See Section 1.3.2.1 on page 4.

2 Demonstration Procedure

2.1 Boot Commands Demo

The Boot Commands Demo will be run at a demonstration terminal in the B-101 laboratory in Building 84.

2.1.1 Context of the Boot Commands Demo

- Two crates are configured and connected together for this demonstration:
  - The first crate is configured with an mv2304 processor and a PMC 1553 communications board (operating as a Bus Controller).
  - The second crate (the “flight crate”) is configured with a RAD750 processor, a 1553-capable Spacecraft Interface Board (SIB), a (cPCI) LAT Communications Board, and a Ramix Ethernet board.
- The PBC package is preburned into the RAD750 SUROM.
- The SIB’s two EEPROM banks are preloaded with software. The lower bank contains one RTOS image (vxw_tornado), a secondary boot executable file in ELF format (SBC_nominal), and a secondary boot application database (lfs_rt.db). The upper bank of EEPROM contains a second RTOS image (vxw_standalone) and a second secondary boot executable (SBC_exception).
- The SBC_nominal secondary boot executable is an ELF object file that bundles several application module constituents: CMX/cmx_asBuiltSpy, PBS/pbs, MSG/libmsg_mt, MSG/msg_print, CCSDS/ccsds_pkt, CTDB/co1553_rt, CTDB/sumt_rt_sib, FILE/file_upl, FILE/file_path, FILE/file_sys. The vxw_standalone version of the RTOS also makes the following constituents available: LFS/lfs_rt, ZLIB/zlib_inflate, FILE/file_hdr, and CMX/cmx_asBuilt.
- The following package constituents are loaded onto the mv2304 “Spacecraft” crate: CMX/cmx_asBuiltSpy, CMX/pbs, MSG/msg_mt, MSG/msg_print, CCSDS/ccsds_pkt, CTDB/co1553_bc, CTDB/sumt_bc_sib, FILE/file_path, LFS/lfs_bc.
- During Part Three of the demo, a third RTOS image, vxw_flight, is uploaded to the flight crate.

2.1.2 Boot Commands Demo Procedure

This demonstration is conducted in four “passes” through the boot process. Outputs for each of these passes are described in Section 1.3.2 starting on page 4.

2.1.2.1 Part One: Cold Boot and Testing using LTX

- The “Spacecraft” mv2304 crate is powered up before the demonstration session begins.
- The tester turns on power to the flight crate. The crate undergoes a COLD primary boot, controlled by the PBC package. When primary boot is complete, the flight crate holds indefinitely in the boot shell. At this stage, with primary boot complete, the PBC package has

checked available RAM and provided a basic 1553 driver package to set the flight crate up as a 1553 remote terminal. Also, the flight crate is sending housekeeping telemetry at 4 Hz, and kicking its watchdog timer.

- From the command line controlling the Spacecraft crate (the 1553 bus controller), the tester issues the **bc_rtos_start** command, a wrapper for sending the telecommand Boot RTOS Execute (APID 0x640, function code 3) via CCSDS. Using **bc_rtos_start**, the tester instructs the flight crate to select and initialize the vxw_tornado image from the lower bank of EEPROM. The crate does NOT continue to secondary boot.
- When the Tornado shell becomes available, the tester runs the LTX test suite; LTX has been discussed in previous demonstrations. The zlib_unit_test is run under LTX control.

2.1.2.2 Part Two: Warm Reboot, Execute Secondary Boot Code, Initialize Application Modules, and Upload Files

- At the terminal, the tester reboots the flight crate. The crate executes a warm reboot.
- When primary boot is complete, and the crate is waiting in boot shell, the tester again issues the **bc_rtos_start** command to send Boot RTOS Exec, this time instructing the flight crate to select and initialize the vxw_standalone RTOS image from the upper bank of EEPROM. This command also instructs the crate to choose the SBC_nominal secondary boot executable and the lfs_rt.db application database from the lower bank of EEPROM to control secondary boot. The XML file that defines the structure of lfs_rt.db is shown below:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SBCdoc SYSTEM "D0-0-0.dtd">


<SBCdoc modified="4/16/04" created="4/16/04">

  <application_database version="1">

        <modules>

              <module key="1">
                    <package> CMX </package>
                    <constituent> cmx_asBuiltSpy </constituent>
                    <version> V2-0-8 </version>
                    <device> 2 </device>
                    <directory> 1 </directory>
                    <file> 0 </file>
              </module>

              <module key="2">
                    <package> PBS </package>
                    <constituent> pbs </constituent>
                    <version> V2-2-0 </version>
                    <device> 2 </device>
                    <directory> 1 </directory>
                    <file> 1 </file>
              </module>

              <module key="3">
                    <package> MSG </package>
```

```
                        <constituent> msg_mt </constituent>
                        <version> V1-1-4 </version>
                        <device> 2 </device>
                        <directory> 1 </directory>
                        <file> 2 </file>
                    </module>

                    <module key="4">
                        <package> MSG </package>
                        <constituent> msg_print </constituent>
                        <version> V1-1-4 </version>
                        <device> 2 </device>
                        <directory> 1 </directory>
                        <file> 3 </file>
                    </module>

                    <module key="5">
                        <package> CCSDS </package>
                        <constituent> ccsds_pkt </constituent>
                        <version> V3-2-1 </version>
                        <device> 2 </device>
                        <directory> 1 </directory>
                        <file> 4 </file>
                    </module>

                    <module key="6">
                        <package> CTDB </package>
                        <constituent> co1553_rt </constituent>
                        <version> V4-0-2 </version>
                        <device> 2 </device>
                        <directory> 1 </directory>
                        <file> 5 </file>
                    </module>

                    <module key="7">
                        <package> CTDB </package>
                        <constituent> sumt_rt_sib </constituent>
                        <version> V4-0-2 </version>
                        <device> 2 </device>
                        <directory> 1 </directory>
                        <file> 6 </file>
                    </module>

                    <module key="8">
                        <package> FILE </package>
                        <constituent> file_upl </constituent>
                        <version> V1-0-1 </version>
                        <device> 2 </device>
                        <directory> 1 </directory>
                        <file> 7 </file>
                    </module>

                    <module key="9">
                        <package> FILE </package>
                        <constituent> file_path </constituent>
```

```
                        <version> V1-0-1 </version>
                        <device> 2 </device>
                        <directory> 1 </directory>
                        <file> 8 </file>
                  </module>

                  <module key="10">
                        <package> FILE </package>
                        <constituent> fule_sys </constituent>
                        <version> V1-0-1 </version>
                        <device> 2 </device>
                        <directory> 1 </directory>
                        <file> 9 </file>
                  </module>

                  <module key="11">
                        <package> LFS </package>
                        <constituent> lfs_rt </constituent>
                        <version> V0-0-0 </version>
                        <device> 2 </device>
                        <directory> 1 </directory>
                        <file> 10 </file>
                  </module>

            </modules>

            <functions>

                  <function name="CMX_asBuiltPrint">

                  </function>

                  <function name= "rt_init">
                        <parameter number="0"> 1048576 </parameter>

                  </function>

                  <function name="rt_start">

                  </function>

                  <function name="upl_info">

                  </function>

                  <function name="rt_info">

                  </function>

            </functions>

      </application_database>


    </SBCdoc>
```

- The “standalone” RTOS is installed and initialized. As soon as it is initialized, the crate continues on to the secondary boot process, using SBC_nominal and lfs_rt.db. The secondary boot process mounts the TFFS file system partition in spare EEPROM. The basic primary boot 1553 driver is replaced with application-level 1553 support in the form of the CCSDS/ccsds_pkt, CTDB/co1553_rt constituents. Secondary boot also installs the application modules defined in the lfs_rt application database, namely all FSW packages used to run the LFS RT (“LAT File System – Remote Terminal”) test application.
- The tester uses standard shell commands (**ls**, **cd**, etc.) to move through TFFS EEPROM directories, list directory contents, etc. These shell commands will be replaced with telecommands in the flight system.
- The tester uses the **bc_upl_file, bc_upl_cancel** commands from the LFS application to load files from the Spacecraft crate to a TFFS EEPROM directory on the flight crate and check on the status of the upload operation. These commands are wrappers for the File Upload Start (APID 0x641, function code 0), File Upload Commit (APID 0x641, function code 2), File Upload Data (APID 0x641, function code 3), and File Upload Cancel (APID 0x641, function code 1) boot telecommands.

2.1.2.3 Part Three: Warm Reboot on SBC Exception

- At the terminal, the tester reboots the flight crate. The crate executes a warm reboot.
- When primary boot is complete, the tester uses the **bc_upl_file** command to load a third RTOS image, vxw_flight, into the upper bank of EEPROM.
- The tester then issues the **bc_rtos_start** command, instructing the crate to select and initialize the vxw_flight image and the SBC_exception secondary boot executable stored in the upper bank of EEPROM.
- The crate attempts to perform secondary boot. However, the SBC_exception executable is designed, as its name implies, to cause an exception during the secondary boot process. When the exception occurs, the crate reboots.

2.1.2.4 Part Four: Cold Reboot on Spacecraft Command Through Discrete Line

- From the Spacecraft crate, the tester issues a cold boot command through a discrete line. The flight crate reboots.

3 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.

4 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.

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

LAT – GLAST Large Area Telescope. One of two science instruments for the GLAST mission.

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

PCI – Peripheral Component Interconnect. General purpose parallel data bus. For the purposes of the SIU boot code, the CompactPCI (cPCI) variant is employed.

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

PPC – The PowerPC microprocessor. In particular the RAD750 processor provided on the BAE RAD750 cPCI CPU board.

PPCI – PowerPCI bridge chip. An ASIC on the BAE RAD750 board which provides a memory controller and a PCI host bridge.

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.

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.

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