GLAST Large Area Telescope

Instrument Flight Software

Functional Demonstration
April 2004

Stanford Linear Accelerator Center

Demo Agenda

Demo Agenda Item	Presenter
1. Overview of the Demonstrations	Dan Wood
2. Boot Commands and Telemetry Demo	Dan Wood
3. Questions from Attendees	NA

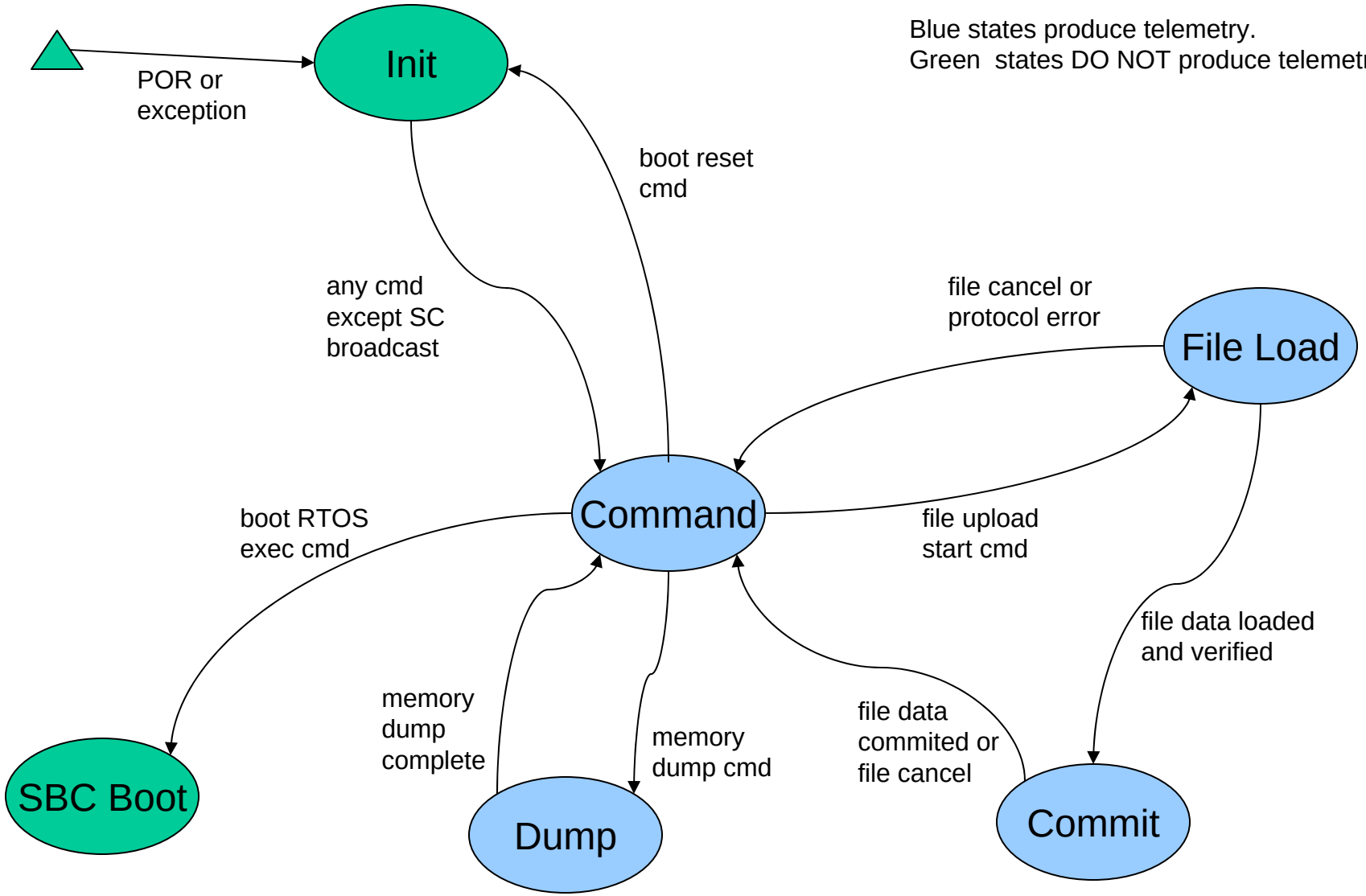
CPU Boot Process

- The LAT RAD750 CPU boards employ a two-stage boot process that covers the time span between when the board is reset or powered on and when the application code is initialized.
- The primary boot code (PBC package) is responsible for:
  - Low level initialization of the RAD750 board
  - Execution of the primary boot code shell
  - Loading and execution of a VxWorks RTOS executable image.
- The secondary boot code (SBC package) covers:
  - Initialization of VxWorks RTOS
  - Loading and initialization of the LAT application code modules.
- The boot processes have been made as independent as possible, so that modifications to the secondary boot code modules may be made without any changes to the primary boot code.

Summary of PBC (Primary Boot) Requirements

- All PBC code and data shall reside in the RAD750 SUROM.
- The PBC code load shall support the SIU and EPU.
- The PBC shall use minimal processor resources.
- PBC shall initialize the RAD750 CPU, memory controller, and PCI bridge to a known state.
- The PBC shall provide the capability to reload the SIB EEPROM.
- The PBC shall generate housekeeping telemetry.
- The PBC shall process the SIANCILLARY command packet from the spacecraft for timing.
- The PBC shall recover from errors and exceptions.
- The PBC shall automatically dump the boot diagnostics to HKP telemetry

Primary Boot States



Boot Telecommands

- Commands demonstrated today are shown in red.

APID	Func Code	Description
0x640	0	Boot Start
	1	Boot Reset
	2	Boot Error Dump
	3	Boot RTOS Execute
0x641	0	File Upload Start
	1	File Upload Cancel
	2	File Upload Commit
	3	File Upload Data
0x642	0	Memory Write
	1,2	PCI Device Header Write, Processor Register Write
0x644	0	Memory Data Dump
	1	Memory Dump Cancel
	5,6	PCI Header Dump, Processor Register Dump

Boot Housekeeping Telemetry

- Boot sends out one telemetry packet 4 times per second.
- Contains command, file load and boot state information.
- Contains boot error codes.
- Contains memory dump data.

- During today's demonstration, boot housekeeping telemetry is being generated. However, it will not be displayed at the demonstration terminal.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Version = 0			T = 0	SH = 1	APID = 0x200										
SF=3	Sequence Count														
Packet Length = 109															
Timestamp Seconds MSW															
Timestamp Seconds LSW															
Timestamp Sub-Seconds MSW															
Timestamp Sub-Seconds LSW															
Software Mode															
Total Error Count															
Queued Error Count															
Error Word MSW															
Error Word LSW															
Received Telecommand Count															
Boot Telecommand Acceptance Count															
File Telecommand Acceptance Count															
Boot Telecommand Sequence Count															
Last Command APID														Last Command Function Code	
Latest Error Word MSW															
Latest Error Word LSW															
File Upload State															
File Upload Packet Count															
Latest File Upload Error MSW															
Latest File Upload Error LSW															
Memory Dump Word Count															
Memory Dump Address MSW															
Memory Dump Address LSW															
Memory Dump Data Word 0-15 MSW															
Memory Dump Data Word 0-15 LSW															

Summary of SBC (Secondary Boot) Requirements

- Configure and initialize VxWorks RTOS
  - VxWorks 5.5 kernel
  - BSP and drivers for RAD750 CPU board hardware
- Provide file system non-volatile storage
  - Small code object modules (variable length)
  - Small configuration objects (variable length)
- Load application object modules
- Call application initialization functions
- Report diagnostics and error from secondary boot process
  - Fatal errors reboot and leave behind information in boot diagnostics area

What's Covered in the Demonstration

- The demonstrations show that:
  - PBC and SBC, the FSW packages responsible for the CPU boot process, are nearly complete and have essentially all of their functionality.
  - The packages that make up the entire low-level infrastructure used by a 1553 bus controller (ultimately, the Spacecraft) and remote terminal to command the boot process, upload files, and retrieve information about the boot process on a flight CPU are in place: ZLIB, FILE, CCSDS, CTDB, and PBC watchdog code.
  - 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.
  - The secondary boot process executes successfully. SBC successfully initializes the RTOS, creates an EEPROM file system, and defines, installs, and initializes application modules.
  - Data from the boot diagnostics area of 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 discrete line.
  - The boot shell responds to commands that allow selection of RTOS images and SBC executables. It also responds to boot commands that allow files to up uploaded and committed to EEPROM. A total of boot telecommands will be demonstrated.
- What's left for future demonstrations:
  - Remaining 10 boot telecommands
  - Boot status over discrete line
  - EPU boot and reboot control
  - A SIIS and AstroRT tools must be used to display housekeping telemetry in a human-readable way

Evaluating the Demonstrations

- **Links to documentation for the systems demonstrated today are available on the FSW Web site. Follow the "Monthly Functional Demonstration" link.**

- **Refer to the Demonstration Guidebook for details:**
  - **Requirements partially or fully demonstrated during the session**
  - **Expected outputs of each part of the demonstration**
  - **Hardware and software context of each demonstration**
  - **Procedures followed for each demonstration**

- **Comments and evaluation are welcome.**