GLAST Large Area Telescope

Instrument Flight Software
Flight Unit Design Review
16 September 2004

Primary Boot Code (PBC)

D. Wood
Naval Research Laboratory / Praxis, Inc.

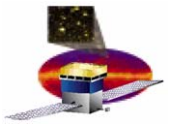
dwood@xip.nrl.navy.mil

Summary of PBC Requirements

- 5.3.1.2: All PBC code and data shall reside in the RAD750 EEPROM.
- 5.3.1.1: The PBC code load shall support the SIU and EPU.
- 5.3.1.4.3, 5.3.1.4.1, 5.3.1.4.2: Power-up, external reset, watchdog reset, and soft reset shall cause execution of the PBC residing in the system's PROM in the RAD750. EPUs perform a primary boot in response to commands from the SIU.
- 5.3.1.3: PBC shall disable the hardware watchdog during the boot process.
- 5.3.1.9: The PBC shall generate housekeeping telemetry.
- 5.3.1.8: PBC shall use discrete interface lines to signal boot status.
- 5.3.1.6: The PBC shall automatically dump the boot diagnostics to HKP telemetry.
- 5.3.1.5, 5.3.1.7, 5.3.1.4.4: PBC shall store system errors in an error log. The log will also record the cause/source of any resets. This log can be sent via telemetry, on command.
- 5.3.7.1-8, 11-12: FSW shall respond to commands to create, dump, and delete files and file system directories; load and dump memory; and, send status information about such objects.
- 5.3.7.9, 5.3.7.10: After entering the boot shell, the FSW processors shall be commandable by the SIU to perform file loads. This load can be cancelled.
- 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 process the SIANCILLARY command packet from the spacecraft for timing.
- The PBC shall recover from errors and exceptions.

Bridge Chip Initialization

- The bridge chip contains an embedded micro-controller (EMC) used to initialize the PPCI bridge chip.
- The EMC code is resident in the PROM.
- The EMC software configures the PPCI bridge chip memory controller and PCI interface and starts the PPC processor.
- The EMC software also provides exception and error processing support, including the watchdog timer.



Power PC Processor Initialization

- First code to run on Power PC.
- Written in PPC assembly language.
- Overview of processing:
  - Setup memory management unit. The PPC 4 DBATS are used to cover RAM, PPCI registers, PCI memory and PROM.
  - Configure PPC processor features.
  - Test 12 MB RAM used by PBC on cold boot. Attempt one level of remediation if failures are detected.
  - Test 128MB RAM used by application on request. Attempt one level of remediation if failures are detected.
  - Copy PBC data segment from PROM to RAM.
  - Initialize BSS segment and stack.
  - Start boot shell (first C code).

Memory Test

- Memory tested in 4 segments: Boot Diagnostics Area, Low Boot Memory, High Boot Memory and Application Memory.
- Tests memory addressing of each 32-bit value and each bit in each memory location.
- On first failure, RAD750 PID is set to indicate failure, the spare nibble column is swapped in and the test is restarted.
- On second failure, after remediation, the PID remains set, the test stops and returns to attempt to run PBC code.
- Test results are stored in Boot Diagnostics Area.
- Remediation swaps spare SDRAM byte column at the expense of the SDRAM EDAC capabilities

Boot Shell Configuration

- PBC is provided in 4 images, loaded separately. These images define both the "personality" of the machine (SIU or EPU) and the interface (1553 or LCB) to be used as the command and telemetry link:

C&T interface	Engineering model image	Flight image
1553	siu_pbc_em	siu_pbc_flight
LCB	epu_pbc_em	epu_pbc_flight

- The PBC code will configure the devices on the PCI bus.

Boot Shell

- Boot shell is main command processing code.
- Polls active communications interface for commands.
- Sends boot housekeeping telemetry
  - When scheduled on 1553 interface
  - Every 250ms on LCB interface
- Boot shell responds to Boot RTOS Execute telecommand by executing the secondary boot code and initializing applications.

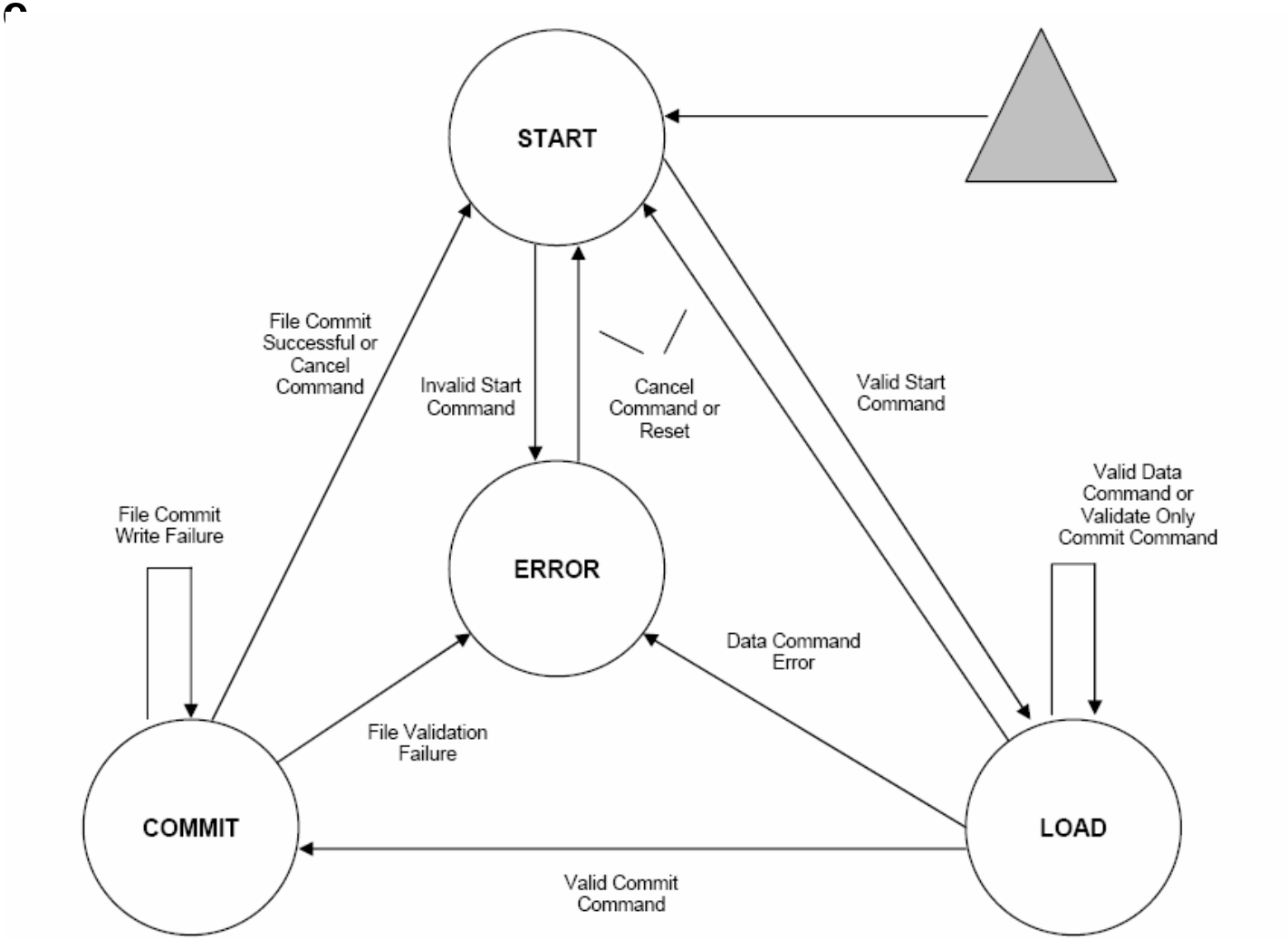
Boot File Operations

- **PBC provides access to load the 9 boot file objects as shown here.**
- **No file dump capability is provided, memory dump operations are available to dump file data.**

File Number	Description
0	RAM RTOS buffer.
1	RAM secondary boot module 0 buffer.
2	RAM secondary boot module 1 buffer
3	SIB EEPROM lower bank boot partition RTOS file.
4	SIB EEPROM lower bank boot partition secondary boot module 0 file
5	SIB EEPROM lower bank boot partition secondary boot module 1 file
6	SIB EEPROM upper bank boot partition RTOS file.
7	SIB EEPROM upper bank boot partition secondary boot module 0 file
8	SIB EEPROM upper bank boot partition secondary boot module 1 file

File Load State Machine

- PBC uses the same state machine for file load as the application code



Memory Load and Dump

- PBC provides the capability to load data to memory mapped devices (RAM, PROM, PPCI registers and PCI memory), PCI device headers and PPC processor registers.
- PBC provides the capability to dump data from memory mapped devices, PCI device headers and PPC processor registers.
- No state machine for memory load, data is committed upon acceptance of memory load command.
- Memory dump data is split up and inserted into boot housekeeping telemetry packets.

Power PC Exception Processing

- PBC provides two sets of PPC exception vectors: ROM based and RAM based.
- ROM based exceptions are used until the PBC boot shell is initialized and configured. These exception vectors record the exception information in the boot diagnostics region and attempt to "plow through" the problem.
- RAM based exceptions are installed at boot shell initialization and used after boot shell is configured. These exception vectors record the exception information in the boot diagnostics region and reset the PBC code.
- Two sets of critical errors are implemented by the RAD750 as interrupts: PCI bus errors and memory controller errors. This requires PBC to poll status bit in boot shell heap.

Boot PROM Memory Map

- **PROM contains all EMC and PPC code & data.**
- **PPC data is copied to RAM at initialization.**

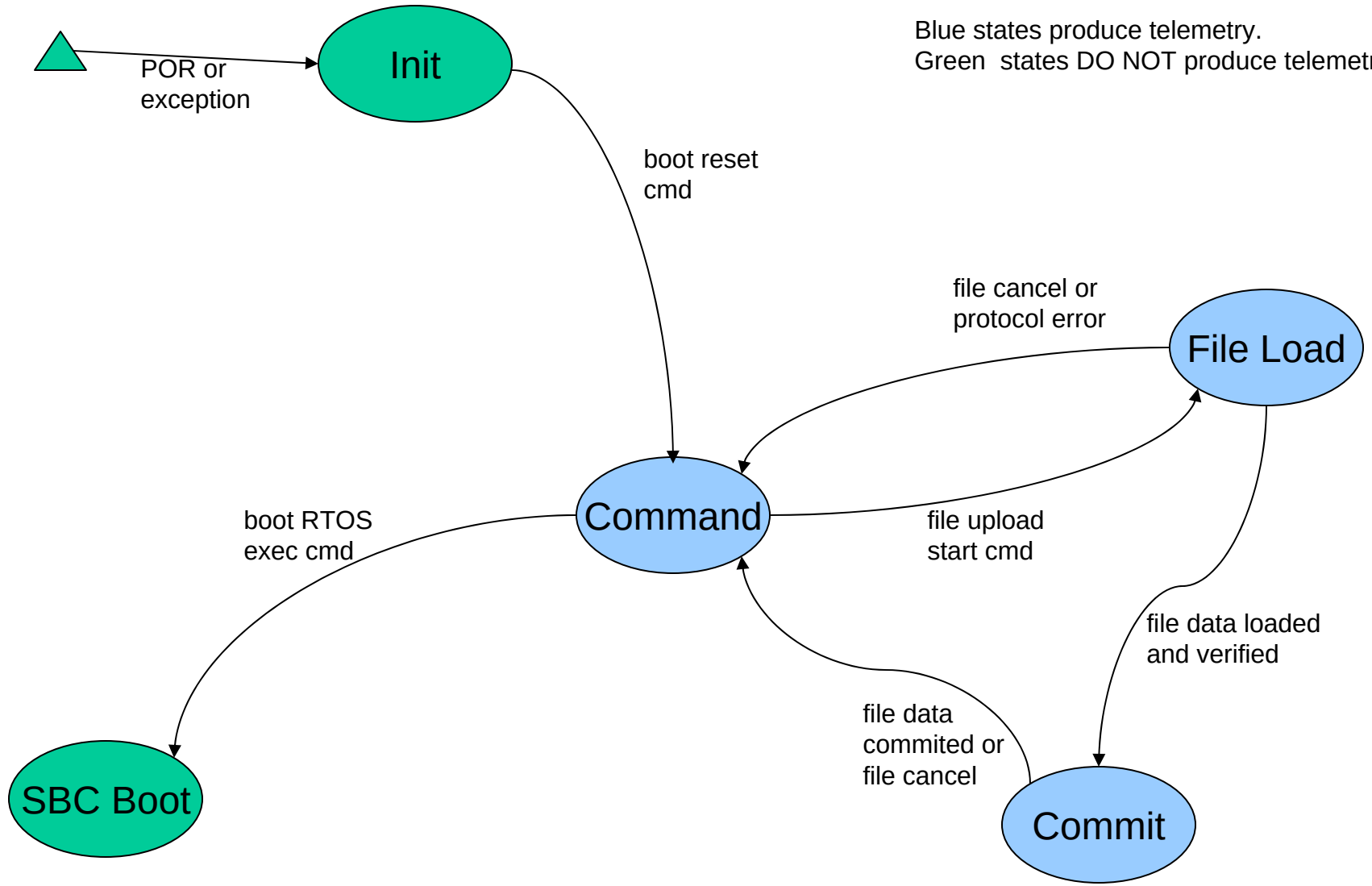
EMC Vector Table	**FFF00000**
EMC Reset Stub	**FFF00020**
PPC Reset Stub	**FFF00100**
PPC Boot Exception Vectors	**FFF00200**
EMC Startup and Vector Code	**FFF01000**
PPC Primary Boot Code Segment (*.text and .rodata*)	**FFF02000**
PPC Primary Boot Initialized Data Segment (*.data*)	***etext***
Unused	***FFF20000 (estimated)***
PBC Configuration Data	***FFF3FC00***

Boot RAM Memory Map

- PBC limited to first 3 megabytes of memory.
- Provides storage for:
  - Ram exception vectors
  - Boot diagnostics region
  - File upload buffers
  - Dynamic allocation heap
  - PBC code program data (copied from PROM at initialization)
  - Hardware I/O buffers
- Dynamic allocation needed by decompression routines. Allocation is "one way" only (i.e no recovery of resources).
- All other memory reserved by RTOS and application code.

Reserved	00000000
Boot Shell Reset Vector	00000100
Boot Shell Exception Vectors	00000200
Boot Code Stack	00003000
EMC Parameter Region	0000FF00
Boot Diagnostics Region	0000FF80
Second Stage Boot RAM Module 0	00010000
Second Stage Boot RAM Module 1	00070000
VxWorks RTOS RAM Load Region	00080000
Boot Shell File Upload Buffer	00140000
LCB Input Ring Buffer	00200000
I/O Output Buffer	002A0000
I/O Input Buffer	002A1000
Boot Shell Memory Heap	002A2000
Boot Code Program Data	002E8000
VxWorks RTOS Image	00300000 00BFFFFF

Boot States


Blue states produce telemetry.
Green states DO NOT produce telemetry
POR or exception
Init
boot reset cmd
file cancel or protocol error
File Load
Command
boot RTOS exec cmd
file upload start cmd
file data loaded and verified
file data commited or file cancel
SBC Boot
Commit

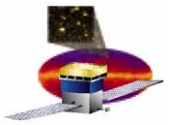
Boot Telecommands

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
0x644	0	Memory Data Dump
	1	Memory Dump Cancel
	2	PCI Device Header Dump
	3	Processor Register Dump
	4	Memory Write
	5	PCI Device Header Write
	6	Processor Register Write

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.

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															
Accepted Telecommand Count															
Latest Error Word MSW															
Latest Error Word LSW															
Last Command Function Code									Last Command APID						
Spare															
Spare															
File Upload State															
File Upload Packet Count															
Scrub Address MSW															
Boot Type															
Memory Dump Word Count															
Memory Dump Address MSW															
Memory Dump Address LSW															
Memory Dump Data Word [0-15]															



Development Environment

- Using GNU tools as supplied by Wind River on a Sun workstation for PPC assembly and C development.
- Using EMC tools as supplied by BAE for EMC code development.
- Using Corelis in-circuit emulator for instruction level debugging on PPC processor (source-level debugging has been problematic)
- Using Corelis emulator and a combination of BAE supplied and NRL created tools to program the PROM.
- Substitute for SDIS on SIU and equivalent GSE for EPU are needed for testing the 1553 interface.

Forward Work

- SIU code complete. EPU code in progress.
- Both primary and secondary boot have already been demonstrated on SIU
  - Boot process involving both SIU and EPU scheduled for October 2004
- Code and unit test complete 10/30/04