GLAST Large Area Telescope

Instrument Flight Software
EM2 Review
26 February 2004

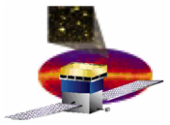
Overview / Management / Schedule

V3

Gunther Haller
Manager, Electronics, DAQ & FSW
LAT Chief Electronics Engineer

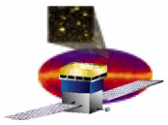
Stanford Linear Accelerator Center

haller@slac.stanford.edu



Agenda

08:00-08:45	Overview / Mgmt. / Schedule	Gunther Haller
08:45-09:45	Software Architecture	JJ Russell
09:45-10:05	Function to Req. Mapping	Mike DeKlotz
	Break	
10:20-11:05	Primary Boot	Dan Wood
11:05-11:30	Secondary Boot	Dan Wood
11:30-12:00	1553 Transport	Dan Wood
12:00-13:00	Lunch & FSW Testbed Visit	
13:00-13:40	Instrument Configuration (GASU, PDU, TKR, CAL)	Curt Brune
13:40-14:00	File-Based Configuration	James Swain
14:00-14:20	Event Delivery	JJ Russell
	Break	
14:30-14:50	Inter-Task Communication	JJ Russell
14:50-15:15	Housekeeping	Sergio Maldonado
15:15-15:45	C&T Database Services	Sergio Maldonado
	Break	
16:00-16:45	Test Plans	Eric Hansen
16:45-17:00	Scheduled Demos & Summary	Gunther Haller



Content

- Instrument overview
- Design strategy
- FSW team organization
- Quick-Look-Review update
- Engineering Model 1 (EM1) redux
- What is added for EM2?
- What remains after EM2?
- Milestones/schedule

- In backup slides
  - Code management/code organization
  - Development environment
  - Software fault handling
  - Documentation list

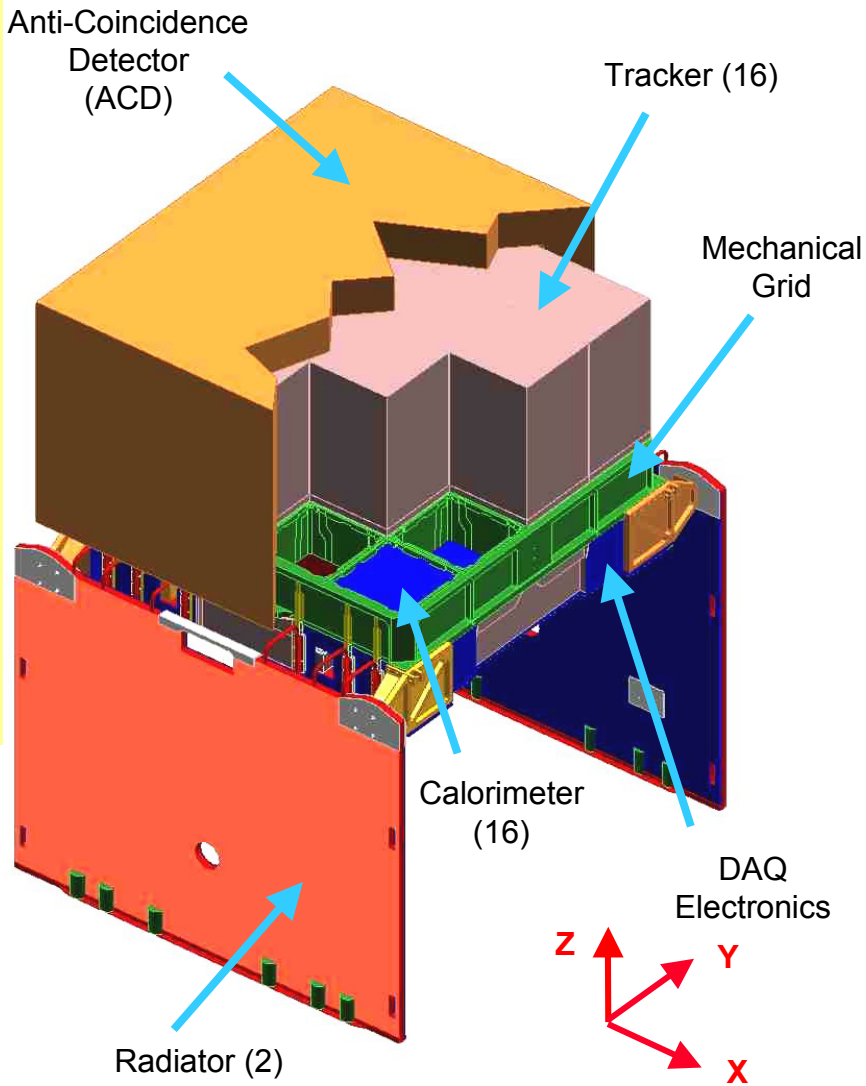
On the Road to Discovery

Large Area Telescope

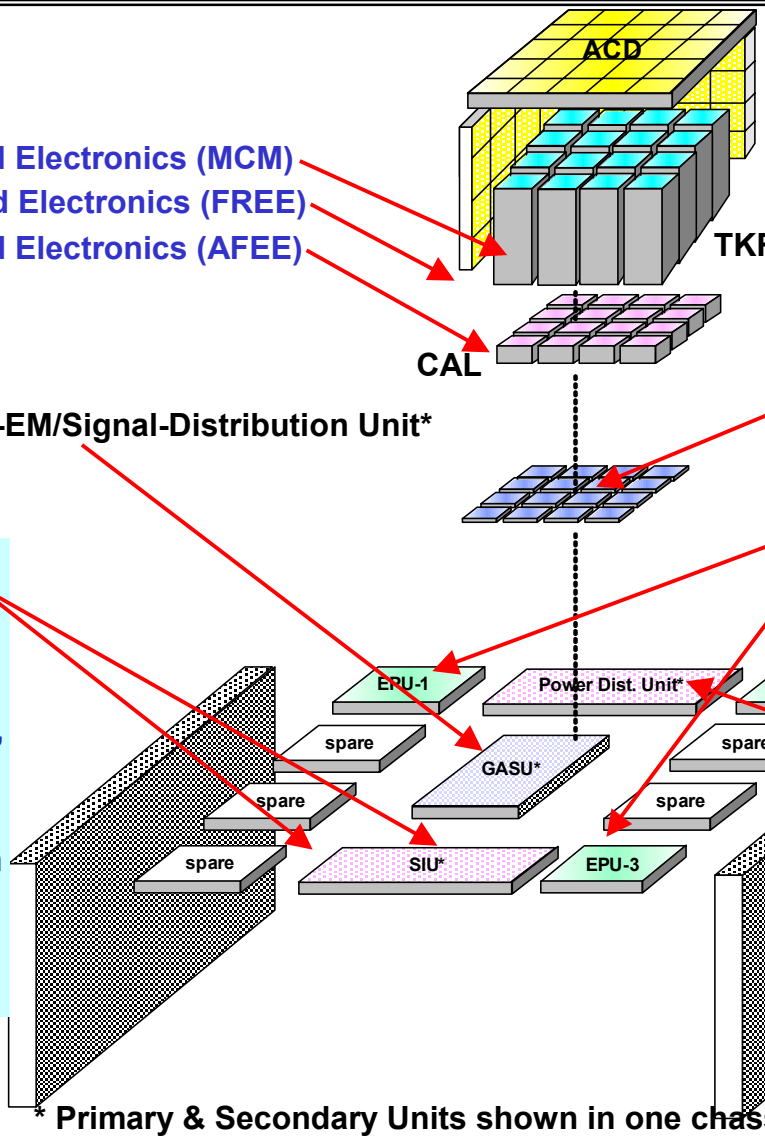
Mass:	**3000 kg**
Instrument Size:	**180 x 180 x 280 cm (X-Y-Z)**
Radiator Area:	**2.83 m^2 each side**
Science FOV:	**2π sr, +Z, above I/F plane**
Mounting:	**4-point mounting I/F at grid**

Key LAT Accommodation Requirements:

- **Pointing Knowledge (10 arcsec, 1σ-r)**
- **Science Data I/F and Peak Rate (40 Mbps LVDS)**
- **Operational Power (650 W OAP)**
- **Clear Radiator FOV on +/- Y faces from LAT I/F Plane (LIP) to L/V Interface Plane (PAF)**
- **Significant Mass Drives Observatory Axial CG**



LAT FSW: Executes on SIU and EPU



FSW is an integral part of the data acquisition (DAQ) subsystem and is managed, budgeted, and scheduled as part of the DAQ subsystem

3 Event-Processor Units (EPU) (2 + 1 spare)
- Event processing CPU
- LAT Communication Board
- SIB

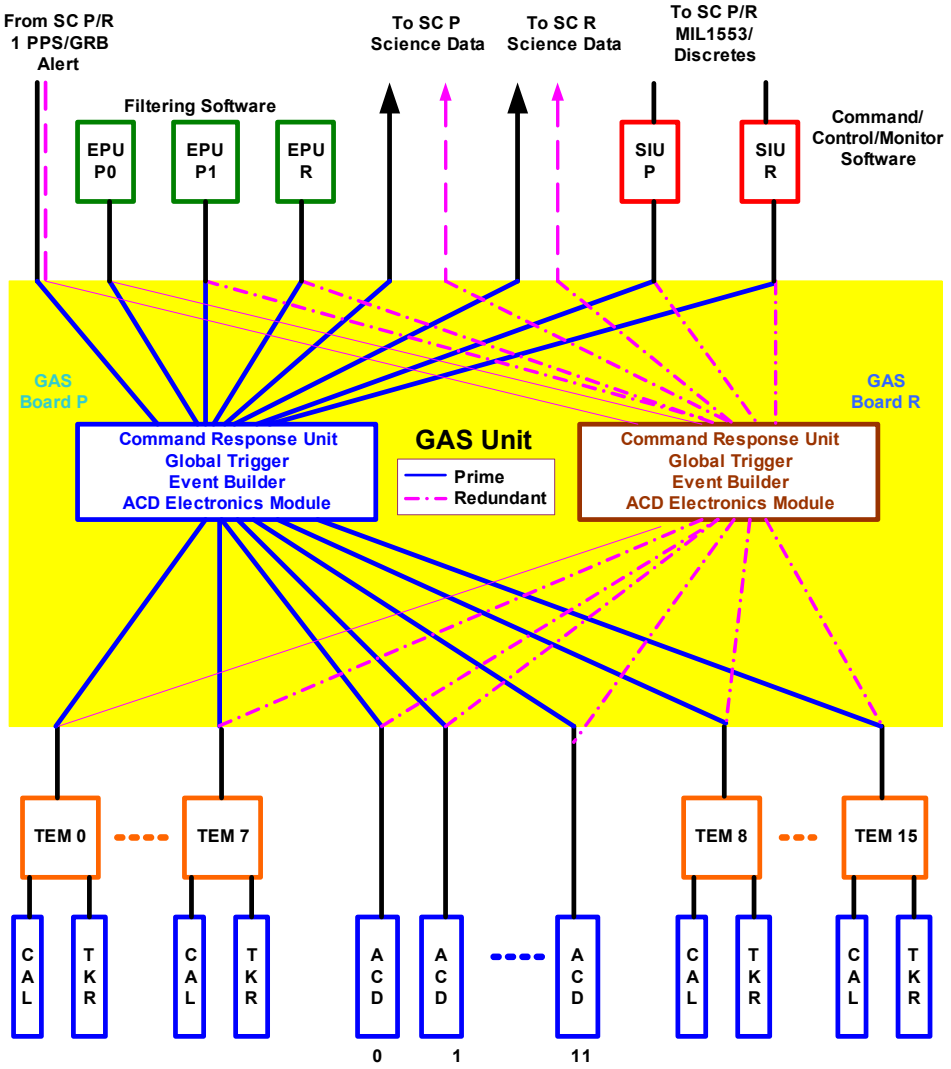
Spacecraft Interface Units (SIU)*
- Storage Interface Board (SIB): Spacecraft interface, control & telemetry
- LAT control CPU
- LAT Communication Board (LCB): LAT command and data interface

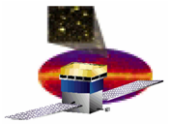
Power-Distribution Unit (PDU)*
- Spacecraft interface, power
- LAT power distribution
- LAT health monitoring

* Primary & Secondary Units shown in one chassis

LAT Electronics

- TKR: Tracker
- CAL: Calorimeter
- ACD: Anti-Coincidence Detector
- TEM: Tower Electronics Module
- EPU: Event Processor Unit
- SIU: Spacecraft Interface Unit
- GAS Unit: Global Trigger-ACD-Signal Distribution Unit
  - CRU: Command/Response Unit
  - EBM: Event Builder Module
  - GEM: Global trigger Electronics Module
  - AEM: ACD Electronics Module
- Flight Software running on RAD750 processors in SIU and EPU cPCI crates

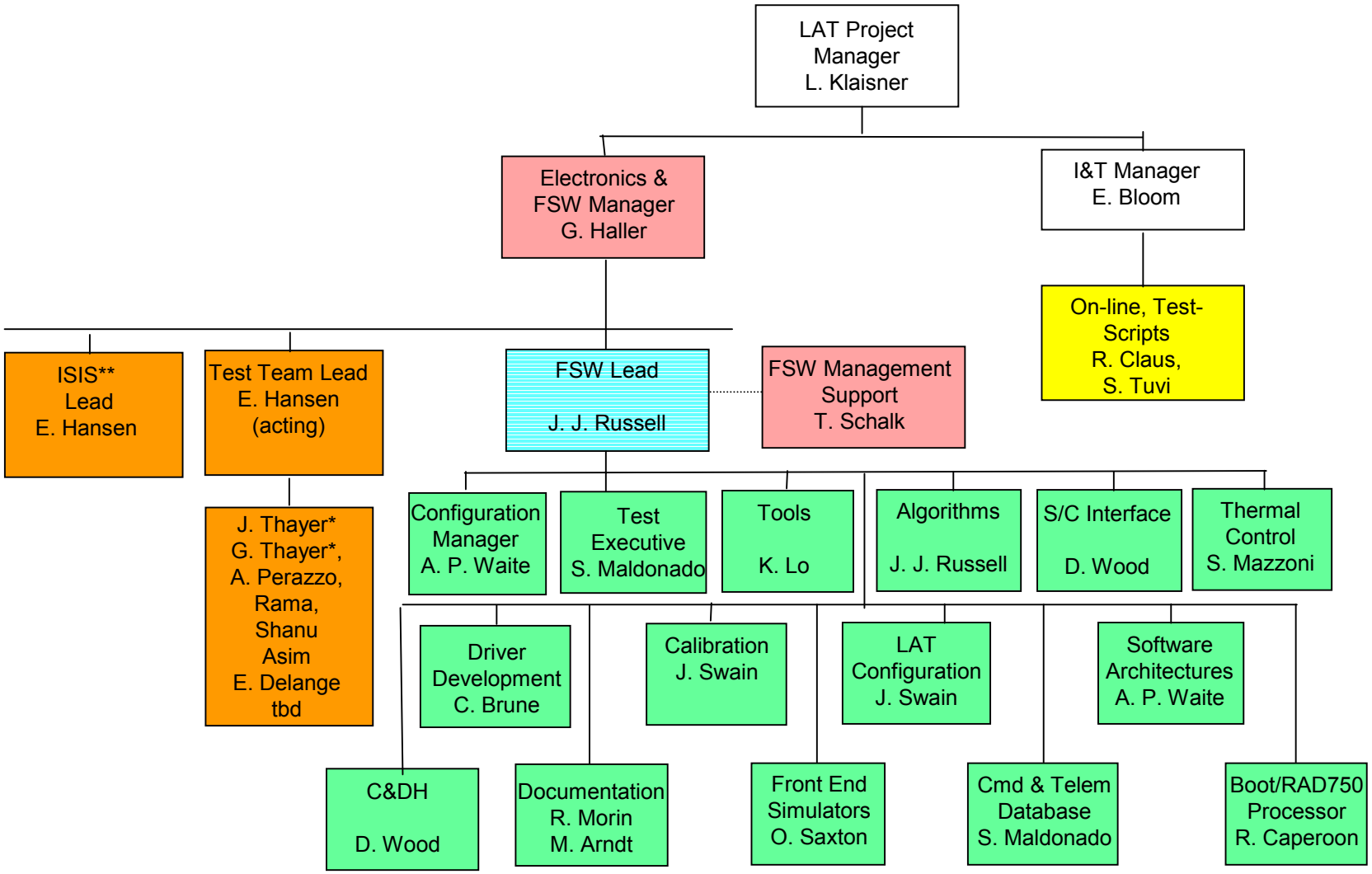




Software Design for Safety

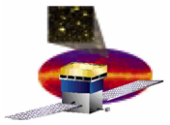
- The software safety environment
  - Software cannot damage hardware (hardware protects itself)
  - Reprogrammable on orbit (except for primary boot code)
- The software safety philosophy during development
  - Leverage the fact that software cannot damage hardware
  - Make unexplained conditions “fatal but not serious” and reboot
    - Decreases complexity
    - Increases reliability / robustness
    - Immediate and graceful exit quickly identifies code weaknesses
      - Improves efficiency for producing reliable / robust final code
  - On a case by case basis, develop recovery strategies
    - Not recoverable and CPU compromised: Stay with reboot strategy
    - Not recoverable but CPU integrity good: Report to ground and await intervention
    - Fully recoverable: Perform recovery action, continue operation

FSW Organization Chart



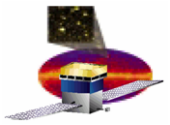
•Start April 1

** ISIS: Instrument Spacecraft Interface Simulator



LAT FSW Development Overview

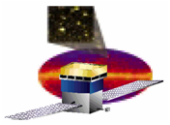
- FSW is an integral part of the data acquisition (DAQ) subsystem and is managed, budgeted, and scheduled as part of the DAQ subsystem
  - > 3 million configuration bits to configure system and to take event data properly
  - Majority of software is to configure & setup hardware (boot, TKR, CAL, ACD, DAQ trigger, event-builder, sub-system setup, etc)
  - Rest is filtering, data monitoring, health & safety, telemetry
  - Once running, most is done in hardware
- FSW builds are coordinated with the DAQ hardware builds in both timing and content



LAT FSW Strategy

- Incremental FSW builds to coincide with the natural hardware builds as follows:
- Major Builds
  - Engineering Model 1
    - Run single tower, single CPU
      - Used for EGSE test-stands for hardware development (built, tested, done)
      - Used for LAT single-tower engineering unit by I&T group (built, tested, done)
  - Engineering Model 2
    - Multiple towers, GASU*, single CPU
      - Used for EGSE test-stands for hardware development (in progress)
      - Used as test-bed for hardware and software development & test (in progress)
- Releases
  - ISIS (Instrument Spacecraft Interface Simulator) Build & Release
      - To be delivered to the Spacecraft vendor (Formal Release)
  - Full LAT Build & Release
    - Complete set of 16 towers, GASU*, full set of CPU's
      - Used as test-bed for hardware and software development & test
      - To be delivered to I&T for full-LAT testing (Formal Release)

* GASU includes LAT Global Trigger, LAT Event-Builder, Command-Response Unit, ACD Electronics Module



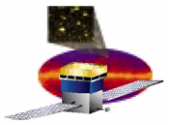
Quick-Look-Review Update

- QLR review due to concern about high productivity estimate by LAT
  - Was held in December 03
  - Discovered that metrics used were different than assumed by project
  - Charge to recalculate Line-Of-Code productivity
    - Count LOC's as defined by project office (see back-up slides for details)
    - Include all man-power working on FSW (see back-up slides for details)
  - Results in average of 9.5 LOC/man/day for LAT which is in-line with NASA experience
- Change Control Board (CCB) at a project level (not just 4.1.7) to prevent implementing of code for new requirements without oversight and impact/benefit/risk evaluation
  - Is in place
- Tracking of progress
  - See schedule/demo's
- Detailed QLR RFA's and responses are listed on review web-page

FSW Resource Usage Current Estimates

Resource	Total Available	Anticipated Usage	Margin Factor
EPU Boot PROM	256 kB	128 kB	2
SIU Boot PROM	256 kB	128 kB	2
EPU EEPROM	6 MB	1.5 MB	4
SIU EEPROM	6 MB	1.5-2.5 MB	3
EPU CPU cycles	200% in 2 EPUs	30%	> 6
SIU CPU cycles	100% in 1 SIU	25%	4
EPU memory	128 MB	16-32 MB	4-8
SIU memory	128 MB	< 16 MB	8
Bandwidth – instrument to EBM	45 MB/sec	10 MB/sec	4.5
Bandwidth – EBM to CPU	20 MB/sec	5 MB/sec	4
Bandwidth – CPU to EBM	2.5 MB/sec	0.075 MB/sec	33
Bandwidth – EBM to SSR	5 MB/sec	0.15 MB/sec	33

Change due to increase in LAT-SC allocation



EM1, Single Tower (delivered)

- Developed
  - Hardware (full single tower of tracker and calorimeter including TEM, PPC processor)
  - Software to run full tower
    - Configuration of tracker, calorimeter, TEM
    - Solicited Housekeeping
    - Event Delivery
  - Also worked on: primary boot of RAD750
- Tested with Python LATTE scripts (by Rama, test group member)
- Delivered to I&T
  - Tested with their own scripts
  - Took cosmic data, Van-de-Graff data, etc successfully

Single Complete Tower

Tower Electronics Module with Tower Power Supply

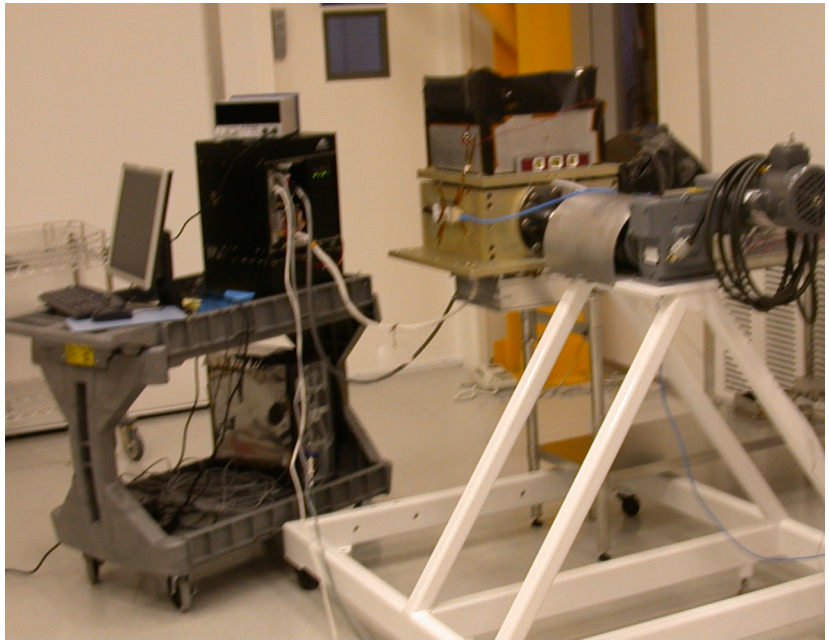
VME CPU with LAT Communication Board (LCB)

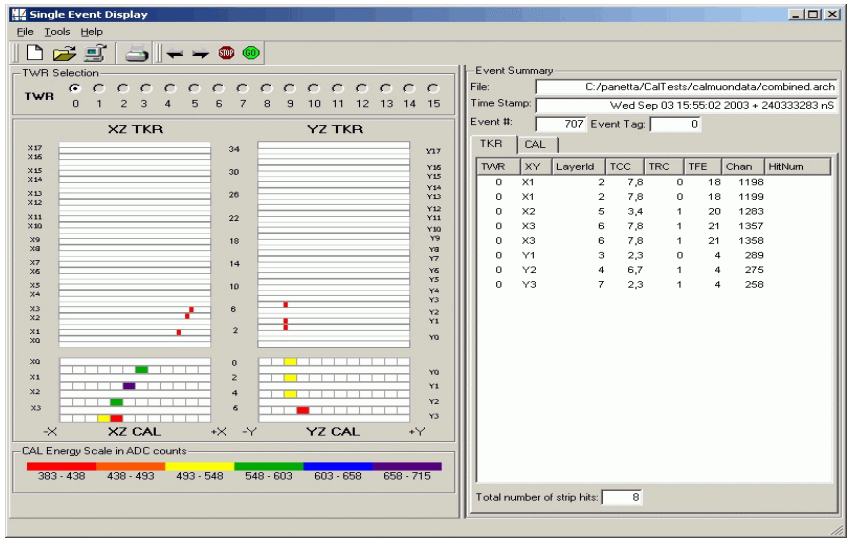
Full set of 4 CAL AFEE boards, (4 sides, 1 each)

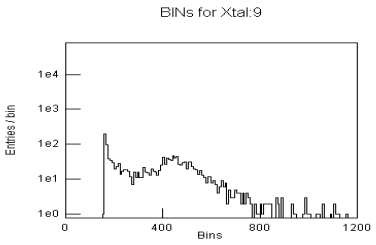
Full set of 36 TKR MCMs (4 sides, 9 each)

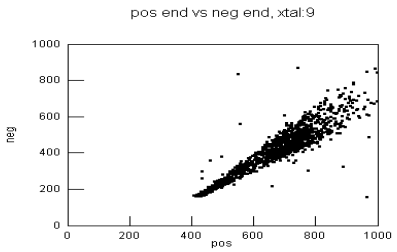
Data-Taking with EM1

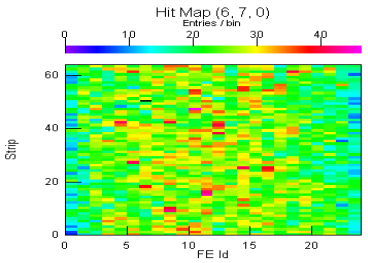
- From EM1 testing as performed by I&T in the SLAC clean room
  - Displays courtesy of the I&T group and LATTE

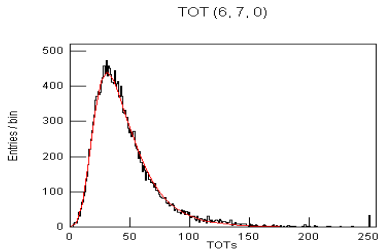


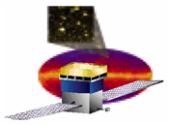






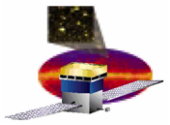






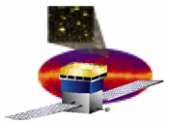
What is added for EM 2?

- Hardware:
  - GASU (global trigger, event builder, command-response unit, ACD electronics module)
  - Power Distribution Unit (PDU)
  - More towers/TEM's
- Software
  - GASU configuration
    - Global trigger
    - Event builder
    - Command-response unit
    - ACD electronics module
  - PDU configuration
  - Streaming housekeeping
  - Inter-task communication
  - Primary/secondary boot of RAD750
  - MIL 1553 communication
  - Telecommand / telemetry
  - Emulated event delivery (ISIS only)
  - Software watchdog



So what is going to be presented today?

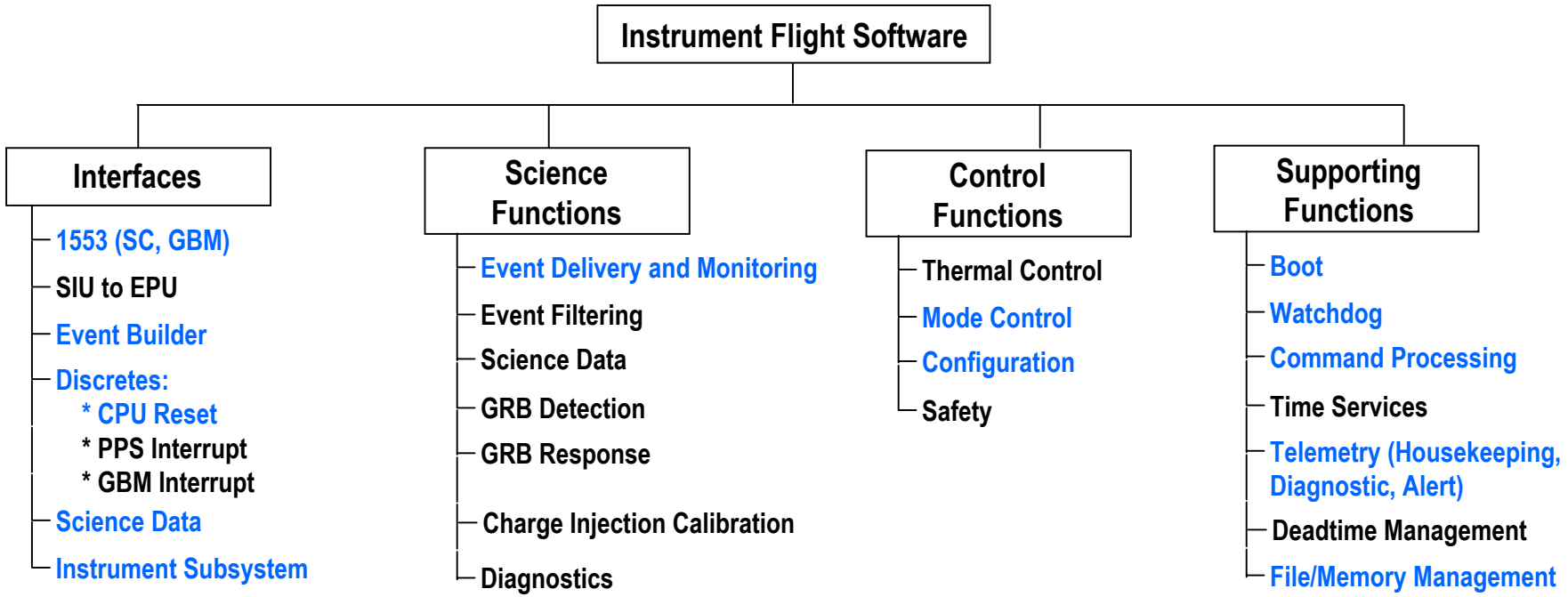
- Overall architecture of FSW
- Specifics of EM2 software
  - Functions
    - Configuration
    - Housekeeping
    - Inter-task communication
    - Primary/secondary boot
    - MIL1553
    - Telecommand & telemetry
    - Event delivery
  - For each function:
    - Requirements
    - Design
    - Testing
- Overall testing of releases



What is remaining after EM2 to get to FU?

- Hardware
  - Multi-CPU system
- Software
  - Event-filtering
  - Event-compression
  - Event monitoring
  - Charge injection calibration
  - Diagnostics
  - Thermal control
  - Full command and telemetry support
  - Time services
  - Gamma-Ray-Burst (GRB) detection/alert
  - SC repointing request

Engineering Model 2 Elements



Blue: EM2

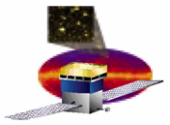
Note:

— This is instrument flight software, not spacecraft flight software

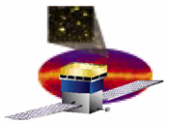
Software Key Milestones for I&T/ISIS

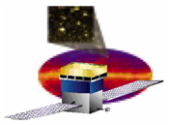
- August 03
  - Tested Software to control/read-out single tower with Calorimeter and Tracker sub-system (done, EM1), delivered to I&T
- Feb 04
  - SW peer review
- March '04 –
  - Lehman review demo.
- May '04 –
  - ISIS test ready review.
- June 04
  - I&T requires software to control/readout multi-tower (i.e. GASU) configuration (EM2)
- June 04 (est)
  - Instrument Spacecraft Interface Simulator (ISIS) to Spectrum Astro
- August 04
  - Demo to LAT system engineering on fully instrumented test-bed
- December 04
  - I&T requires tested software to control/readout full LAT
  - FU SW build to I&T
- December 04
  - FU build finish and transition to formal test
  - FU Formal test complete Feb 05
- February 05
  - I&T requires FU software release to operate/test (whole LAT is integrated)
  - Start of system testing
- May 05
  - End of system test
- July 05
  - Ship LAT to NRL for environmental test

- FUNCTIONAL DEMONSTRATIONS every month (details at end of today's presentations)
  - with ISIS / FU having additional formality



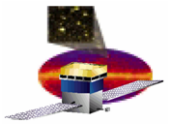
Back-Up





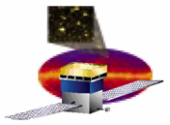
Software Fault Handling

- Fault Recovery
  - On basis of fault identified
    - Reboot
      - Always attempt to save a block of information describing the fault condition in a known fixed memory location so that it can be picked up and sent to ground after the reboot
    - Notify ground and await intervention
    - Execute recovery procedure and continue operation



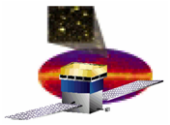
Development Environment

- **Embedded System**
  - Processor / operating system: BAE RAD750 / VxWorks
  - Toolset (Wind River Systems):
    - Language: C
    - Development platform: Sun / Solaris
    - Compiler / linker / binutils: GNU cross compiler suite
    - Debugger: Crosswind
- **Host System**
  - Processor / operating system: Sun / Solaris or Intel / Linux
  - Toolset (host simulation or cooperating processes):
    - Language: C
    - Development platform: Sun / Solaris or Intel / Linux
    - Compiler / linker / binutils: GNU compiler suite
    - Debugger: GDB / DDD
  - Toolset (test executive and scripting): Python / XML / MySQL / Qt / Perl
- **Other Tools**
  - Requirements management: DOORS
  - Code / configuration management: CMX / CMT / CVS
  - Autogeneration of documentation: Doxygen, in-house developed tools
  - Documentation: Microsoft office suite (also Adobe / Framemaker, etc.)



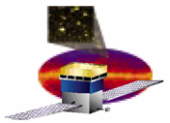
Management & Configuration

- Version numbers follow strict naming conventions indicating:
  - Major: non-backwardly compatible interface changes
  - Minor: backwardly compatible interface changes
  - Patch: bug-fixes and performance enhancements
- A traditional build is a coherent collection of tagged packages
- Changes to builds delivered for formal testing (e.g. ISIS, FU), must be approved through configuration manager and CCB process
- Configuration management
  - Formal control through project management tools
    - LATDocs System
    - Non conformance reporting system



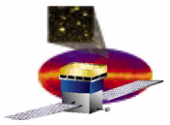
SW Code Management

- Packages exist at 3 levels
  - Test
    - "Sand-box" in the developer's private area
  - Development
    - Public area used to exercise code in a non-critical environment
  - Production
    - Public area used for official, tagged versions
    - A number of these can exist at a given time
      - One is declared the *current* production version
      - Back copies exist for comparison and other purposes
    - All production versions are tagged in CVS



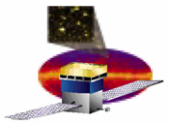
Code Organization

- FSW partitioned into functional blocks based on the Software Requirements Specification (SRS)
  - Functional blocks are then mapped into packages, the fundamental unit of the code management system
- Package code is (and has been) version/configuration controlled via Code Management System
  - Package structure includes
    - Source code
    - Documentation directory (manuals, developer guides, …)
    - Software Development Folder (a directory)
      - Development notes, version log, running log, to do lists, …
    - Package test directory
      - Pure test code
      - LTX test definitions/scripts
- Build: a collection of version-tagged packages
  - For an example for EM1 build see
    - https://oraweb.slac.stanford.edu:8080/pls/slacquery/bbrdownload/FSW-EM1-release-VDD.pdf?P_FRAME=GLAST&P_DOC_ID=10444



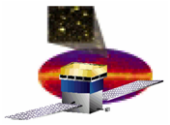
Software Development Approach

- Software lifecycle model
  - Iterative / incremental development model
  - Multiple builds with increased capability with each build
  - Regression testing on each build
- Requirements flowdown, analysis, review
  - Flowdown from program and system specs
  - Peer reviews
- Continuous cycle of development and test
- Code management
  - Formal control through the CMX / CMT / CVS toolchain
- Configuration management
  - Formal control through project management tools
    - LATdocs
- FSW CCB



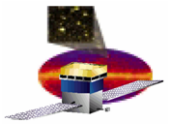
Code Metrics

- ***Summary:***
  - **Identify productivity metrics and definition of Source Line Of Code**
- ***Action Plan –***
  - **Action 3A: GPO, SLAC: Develop a mutually understood and agreed upon definition for metrics measurement. Specifically for SLOC and SLOC/day. Use this to put the IRT concerns to bed once and for all. Report on these metrics to Project . Recommend that after we get a good "baseline" estimate, that we ask for updates of 'actual' vs. 'estimated' at the package/unit level at build/release milestones. (See Test Process Action (#4/5) for more on Test Metrics.)**
  - **Determine whether there is adequate validity to providing estimated productivity metrics to build/release milestones. If we use them, evaluate productivity/performance needs to support planned future milestones from the past performance.**
  - **Action 3B: GPO, SLAC: Work a baseline to report out, using that baseline on our overall project estimates.**



Productivity Definitions

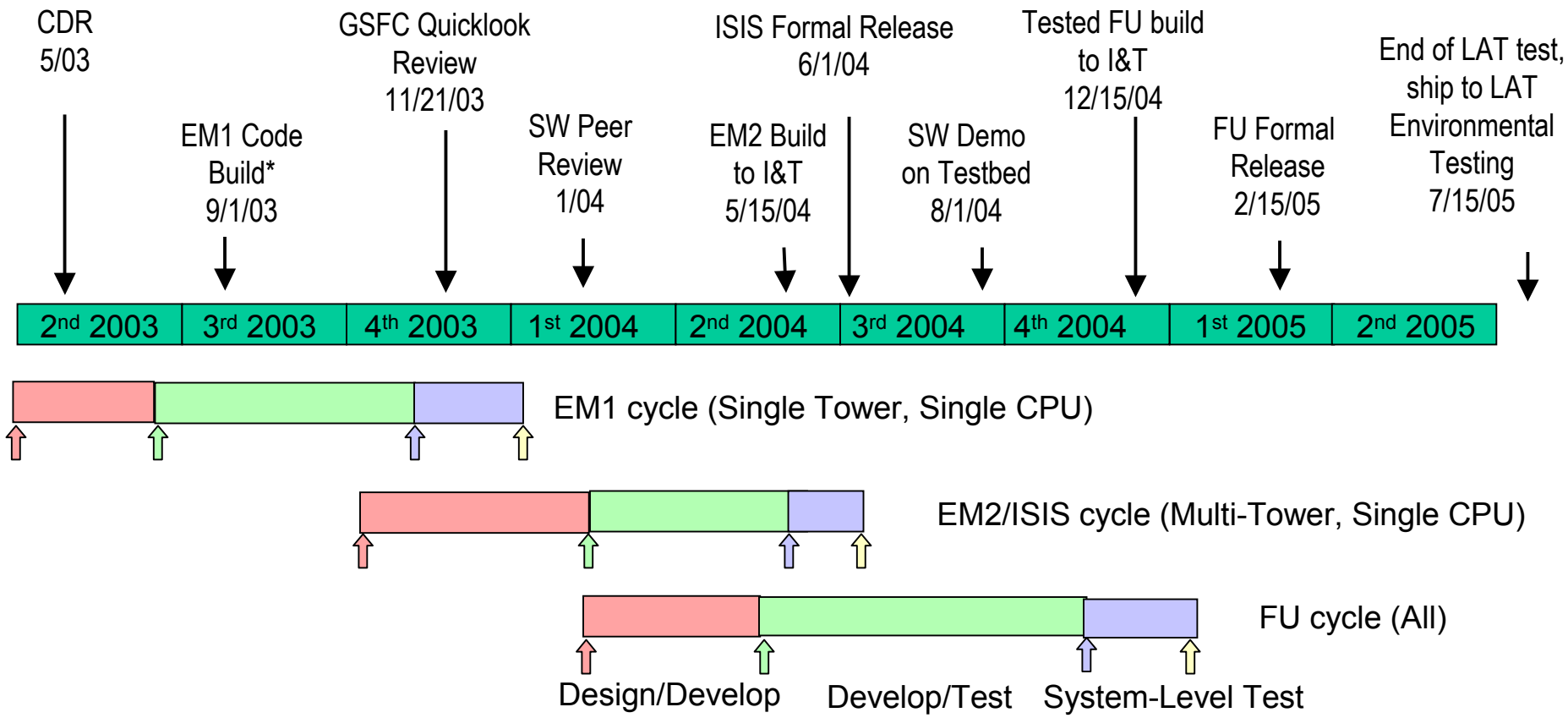
- Implicitly a Source Line Of Code is tested and verified (I guess this sentence makes it explicit! :-)
- Duration of Project Life for which to measure amount of effort associated with a piece of software - From the Requirements Analysis Phase through FQT. Since FQT often reveals problems which need to be corrected, the effort associated with the correction and retest of problems found in the initial FQTs is included. Maintenance/Enhancements following FQT are not considered part of the project duration (and therefore the staffing for these efforts are not included in productivity estimates).
- Due to the duration of the effort on a project which is used to measure productivity, intermediate measurements of SLOC/day are inherently risky metrics to take stock in.
  There may be some value in assessing iterations (Build 1, Build 2, etc) for tracking purposes, but the parameters for the assessment must be made clear and the error bars on this type of intermediate assessment are typically large (eg. 2X).
- Note: In all cases (LAT, GBM, SC), we are utilizing the C programming language (except in the cases where assembly language is required for specific tasks. These are very few.)
- C Source Line Of Code - Semi-colon delimited statement of lexically valid programming language.
- Flight Software. Only lines of code that are compiled into executable images for the flight processor and are placed into memory on the spacecraft/instrument are considered Flight Software. ALL flight software must be tested to the same rigorous standard and held to meet requirements.
- Productivity. Productivity estimates will be greatly influenced by what tasks are included in establishing the amount of effort (measured in staff-months) to perform the FSW tasks. Here at GSFC, and for GLAST, the effort associated with the FSW task includes:
  - (1) FSW Management/Leadership.
  - (2) FSW Dev't & Test Engineering Support.
  - (3) FSW Design: Includes all analysis, design, development and Unit Test.
  - (4) FSW Testing: Beyond the Unit Testing, Through Build and Release Testing.
  - (5) H/W Support.



Productivity based on definitions

Person	Months		Person	Months	
JJ	43		Curt B	40	
Tony	43		Sergio	30	
GXH	22		Kim Lo	18	
Erik A	21		James S	22	
Eric Hansen	15		Owen S	22	
Amadeo P.	18		Dan W	43	
Huffer/TTL	22		Ray C	20	
Rama	18		Brian D	6	
Shanu	16		Steve M.	16	
Eric D/Asim	15		Mike D	15	
Mark A.	15		R Morin	15	
Ric & Selim	18		TBD SE	15	
Total Man Months On Project:				528	
Total Man DAYS on Project				10560	
Estimated SLOC on LAT:				100000	
Estimated Productivity (SLOC/day):				**9.47**	

Breakdown of Development Cycles



Design/Develop: Start design, code small prototypes, no hardware available, only descriptions
Develop/Test: Code and test against real hardware, take snap-shot at end (i.e. define build)
System-Level test: Build Test

*: EM1 was built/tested/delivered, group is now integrating/testing another recently available piece of hardware for EM1, the LCB.

LAT FSW Team Heritage

- Very experienced people
  - HEP
  - NRL
- Successful track record
  - Architected, designed, implemented, tested, and commissioned major large experiments
- Leads are developers
- Leads are scientists

Experiment	# CPUs	Man-years	Time
SLD	> 500	12	4 years
Babar	> 200	15	3 years
LAT	3	20	In progress

SLD: SLC Large Detector

BaBar: B-B Detector (Matter-Antimatter Exp.)

BFEM: GLAST LAT Balloon-Flight (proof-of principle for GLAST LAT detectors and DAQ)

FSW LATDocs Document Index

SS - System Specification (Design To Specification)

Doc. No.	V#	DCS	Ext.	S	Primary Author / Document Title
SS-00053		4IW			Russell, J.J. Balloon Flight Data Format
SS-00169	??	4SO	PDF		Johnson, Robert P. Tracker Front-end Readout ASIC Specifications
SS-00170	6	4SO	PDF		Johnson, W. Neil Conceptual Design of the GLAST Tracker Readout Controller Electronics ASIC (GTRC)
SS-00208	4	4IW	PDF		Johnson, W. Neil Calorimeter Readout Control, ASIC - Conceptual Design
SS-00286	0	4IW	PDF		Russell, J.J. GLAST LAT - Conceptual Design of the Global Trigger
SS-00363	5	4SO	PDF		Bielawski, Rich LAT Dataflow Subsystem Specification - ACD-AEM Interface
SS-00399	2	4SO	PDF		Waite, Anthony P. LAT Flight Software Level IV Specification

FSW LATDocs Document Index, cont.

SS-00424		4IW		Johnson, W. Neil Calorimeter Front End ASIC - GCFE V9 Design Description
SS-00443	??	AIW	PDF	Wallace, James L Conceptual Design of the Instrument Simulator
SS-00461	1	4IW	PDF	Huffer, Michael E. LAT TEM-GASU to CPU Data Formats
SS-00912	0	4IW	PDF	Brune, Curtis LAT Dataflow Simulation
SS-01596	0	4IW	PDF	Huffer, Michael E. The GLAST Global Trigger - Design Specification
SS-01597	2	3IW	PDF	Russell, J.J. Configuration Data: Storage and Transmission
SS-01792	0	4IW	PDF	Russell, J.J. SIB - The GLAST/LAT Spacecraft Interface Board, Hardware Specification

FSW LATDocs Document Index, cont.

TD - Technical Document

Doc. No.	V#	DCS	Ext.	S	Primary Author / Document Title
TD-00331		4IW			Russell, J.J. LAT Flight Software Preliminary Design Report
TD-00560	1	3SO	DOC		Huffer, Michael E. LAT Global Trigger and ACD Hit Maps
TD-00593	4	4IW	PDF		Brune, Curtis LAT Comm I/O Board -- Response FIFO
TD-00597	4	4IW	PDF		Brune, Curtis LAT Comm I/O Board -- Triggering
TD-00605	2	4SO	PDF		Huffer, Michael E. The Tower Electronics Module (TEM) - A Primer
TD-00606	1	4IW	PDF		Huffer, Michael E. LAT Inter-module Communications
TD-00639	1	3SO	PDF		Haller, Gunther The ACD Electronics Module (AEM)
TD-00712	1	3SO	DOC		Russell, J.J. LAT Auxiliary Data Survey
TD-00786	2	4SO	PDF		 LAT Flight Software Test Plan

FSW LATDocs Document Index, cont.

TD-00861	1	4IW	PDF	Huffer, Michael E. Test-stand Architecture Redux
TD-00862	1	4IW	PDF	Huffer, Michael E. Test-stand Update
TD-00863	1	4IW	PDF	Huffer, Michael E. LAT Custom Processor Specification
TD-01121	1	3SO	DOC	Russell, J.J. Trigger and Dataflow Resource Usage
TD-01199	10	4IW	PDF	Brune, Curtis LAT Test Stand Communications Interface
TD-01380	3	4IW	PDF	Brune, Curtis LAT Communication Board Driver
TD-01536	1	3SO	DOC	Russell, J.J. SC / LAT ICD for Startup Procedures
TD-01543	1	3SO	PDF	Huffer, Michael E. The Power Distribution Unit Programming ICD Specification
TD-01545	1	3SO	PDF	Huffer, Michael E. The GLT Electronics Module Programming ICD Specification
TD-01546	1	3SO	PDF	Huffer, Michael E. The Event Builder Module Programming ICD Specification
TD-01547	1	3SO	PDF	Huffer, Michael E. The Command/Response Unit Programming ICD Specification

FSW LATDocs Document Index, cont.

TD-01553		3IW		Russell, J.J. Instrument Damage Protection against LAT Processor Hardware or Software Malfunction
TD-01781	1	3SO	DOC	Russell, J.J. LAT Flight Software Package Descriptions and LOC Basis of Estimate
TD-01806	5	3IW	PDF	Russell, J.J. Primary Boot Code
TD-02067	1	4IW	PDF	Waite, Anthony P. Glossary of Flight Software Terms
TD-02150	2	4IW	PDF	Waite, Anthony P. LAT Flight Software Secondary Boot Code
TD-02620	1	AIW	PDF	Swain, James E. LAT Instrument Startup
TD-02627		AIW		Swain, James E. Trigger Test Plan
TD-02634	??	4IW	PDF	Waite, Anthony P. Em1 Code Release Version Description
TD-02659	1	AIW	PDF	Wood, Daniel Lee LAT Flight Software Telecommand and Telemetry Formats

FSW LATDocs Document Index, cont.

XR - Change Record

Doc. No.	V#	DCS	Ext.	S	Primary Author / Document Title
XR-02038	1	3IW	PDF		Russell, J.J. Dcn for LAT Daq Trigger and Dataflow Resource Usage
XR-02619		AIW			Swain, James E. Dcn for LAT Instrument Startup

Legend

Heading	Description
Doc. No.	LATDocs document number
V#	current version number
DCS	Document Control & Status (abridged)
Ext	File Extension
S	Status (? indicates new document)
Author / Title	First author listed by LATDocs, Title

This page last modified on January 22, 2004 at 14:16 PST.
(Included material may have been updated at a different time.)
Contact fsw-web for help requests, comments, bug reports, etc.