



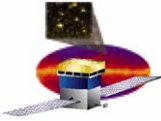
GLAST Large Area Telescope

**Instrument Flight Software
Flight Unit Design Review
16 September 2004**

Instrument Configuration

**James Swain
Stanford Linear Accelerator Center**

**jswain@slac.stanford.edu
650-926-2721**



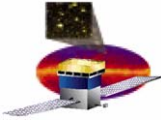
Instrument Configuration: Requirements

- **Flight Software General Requirements:**
 - **Configuration of Subsystems (5.3.17.1)**
 - Upon command, the SIU FSW shall configure the LAT power distribution subsystem and the CAL, TKR, ACD, and T&DF subsystems by writing to the T&DF-provided configuration registers.
 - **Configuration by File (5.3.17.2)**
 - Upon command, the SIU FSW shall configure the LAT power distribution subsystem and the CAL, TKR, ACD, and T&DF subsystems by writing to the T&DF-provided configuration registers specified in a configuration file that identifies the registers to be written and the corresponding value of the registers.
 - **Readback of Subsystems (5.3.17.3)**
 - Upon command, the SIU FSW shall read back and record the configuration of the LAT power distribution subsystem and the CAL, TKR, ACD, and T&DF subsystems by reading back the T&DF-provided configuration registers.



Instrument Configuration: Overview

- **As one can see from the previous slide, the SRS describes instrument configuration in 2 ways:**
 - **As the configuration and readback of individual registers on LAT subsystems**
 - **As the configuration of large collections of registers using configuration files.**
- **Correspondingly, this presentation is divided into two pieces:**
 - **“Instrument Configuration”, which covers configuration of individual registers using the LEM package**
 - **“Instrument Configuration by File”, which addresses file-based configuration using LATC**



Instrument Configuration: Functional Components

- **Functional Inputs**
 - **FSW receives different inputs for its write and read operations:**
 - **Write:**
 - Addressing data that identifies registers of LAT subsystems (TKR, CAL, ACD, TEM, CRU, EBM, GEM, and PDU)
 - Data to write to the register
 - **Read:**
 - Addressing data
- **Functional Processing**
 - **FSW produces bit streams that are sent to the registers of subsystems on the LAT**
 - **FSW decodes responses received from registers of subsystems on the LAT**
 - **FSW validates responses by checking parity**
- **Functional Outputs**
 - **FSW outputs values read from registers on the LAT (TKR, CAL, ACD, TEM, AEM, CRU, EBM, GEM, and PDU)**
 - **FSW outputs I/O completion status and time stamps**



Instrument Configuration: Overview

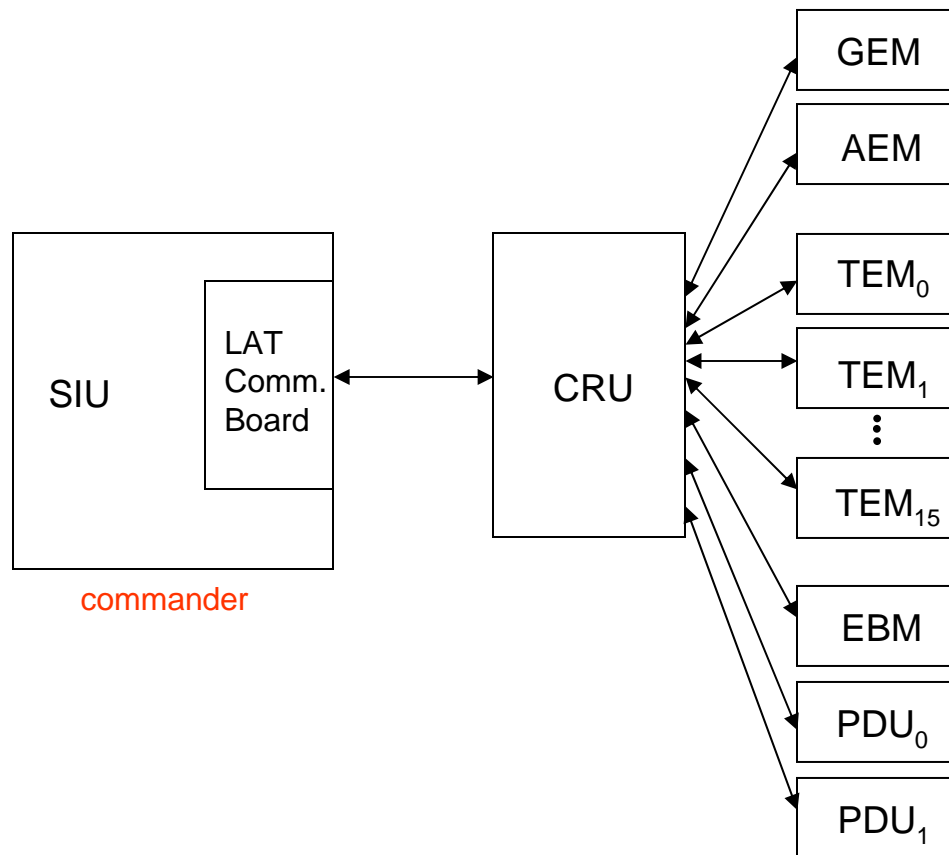
- **Purpose**
 - To place and maintain the instrument in a known configuration
 - To report the configuration
 - Knowledge of instrument configuration is *essential* to a successful physics analysis
- The instrument appears to flight software as a collection of registers
 - A large collection

Subsystem / Block	Chip / Sub-block	Number in Instrument	Register Count	Register documentation
TKR	GTFE	13824	69120	LAT-SS-00169
	GTRC	1152	2304	LAT-SS-00170
CAL	GCFE	3072	21504	LAT-TD-00424
	GCRC	256	2048	LAT-SS-00208
ACD	GAFE	216	2376	LAT-SS-00363
	GARC	12	528	
TEM	GTCC	128	768	LAT-TD-00605
	GCCC	64	448	
	GTIC	16	272	
	Common	16	112	
GASU	AEM	1 (primary and redundant)	6	LAT-TD-00639
	CRU	1 (primary and redundant)	4	LAT-TD-01547
	EBM	1 (primary and redundant)	31	LAT-TD-01546
	GEM	1 (primary and redundant)	59	LAT-SS-01596
PDU			7	LAT-SS-01543
			99587	



Instrument Configuration: Communications Architecture

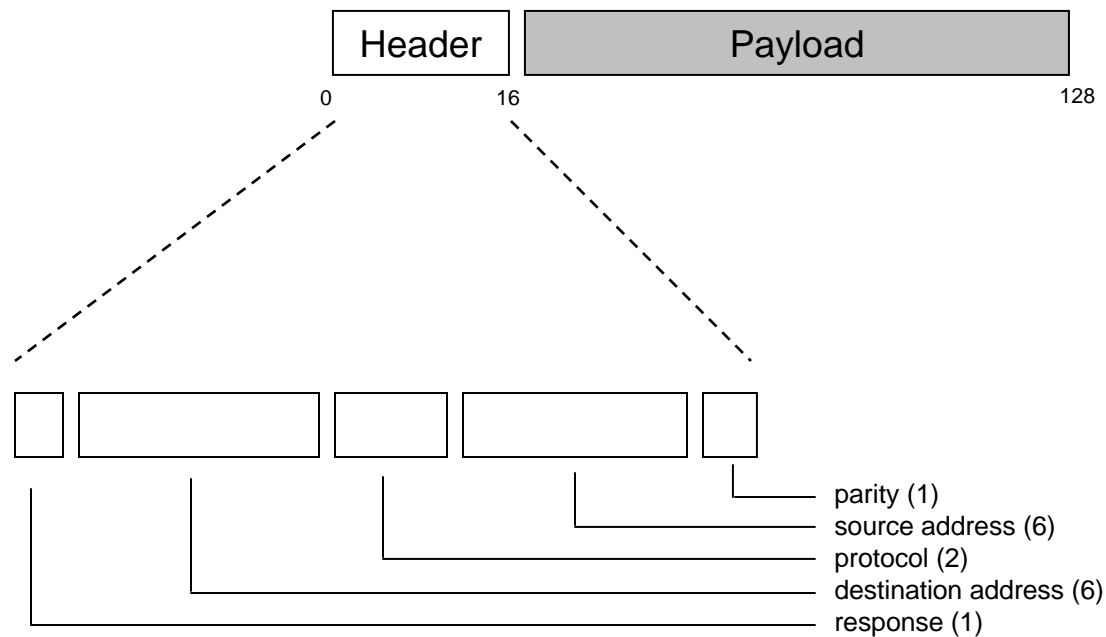
- Communications with registers on the CAL, TKR, ACD, T&DF and LAT power supply subsystems occur over a network fabric using the LATp communications protocol:
 - Certain nodes (SIUs) are *commanders*. Other nodes (EPUs, Tower Electronics Modules, ACD Electronics Modules, Global Triggers) are *responders*.

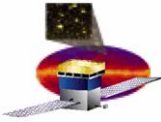




Instrument Configuration: Communications Architecture (cont'd)

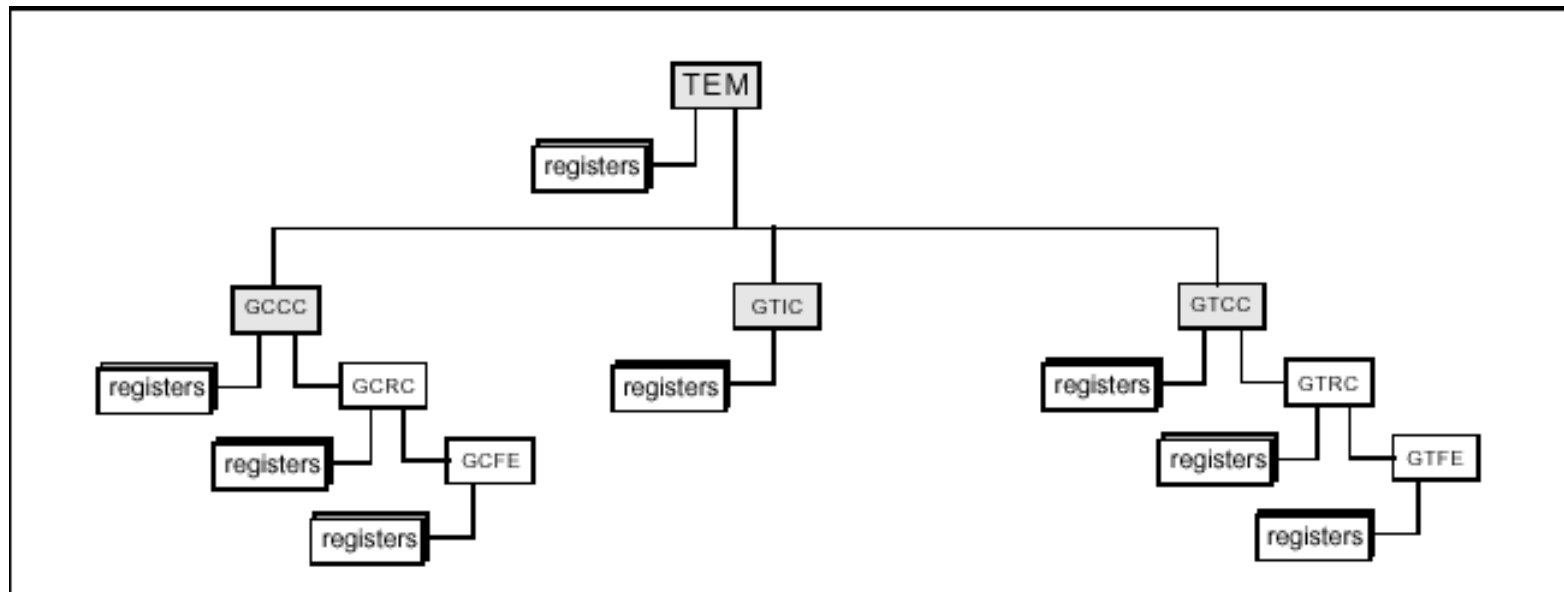
- The LATp protocol provides a type of communication packet called a *command packet*. Command packets are used to send read and write commands to registers on the LAT.
- LATp command packets are 128 bits long. They are broken down into a header of 16 bits, followed by 112 node-specific bits.

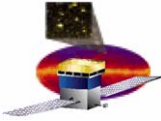




Instrument Configuration: Reaching the Registers

- The registers in a LATp node (e.g., a TEM) are arranged in a hierarchy of function blocks (e.g., a TEM CAL cable controller), each block containing registers and possibly more function blocks.
- To access the registers of a function block, one specifies the hierarchical path to the register.



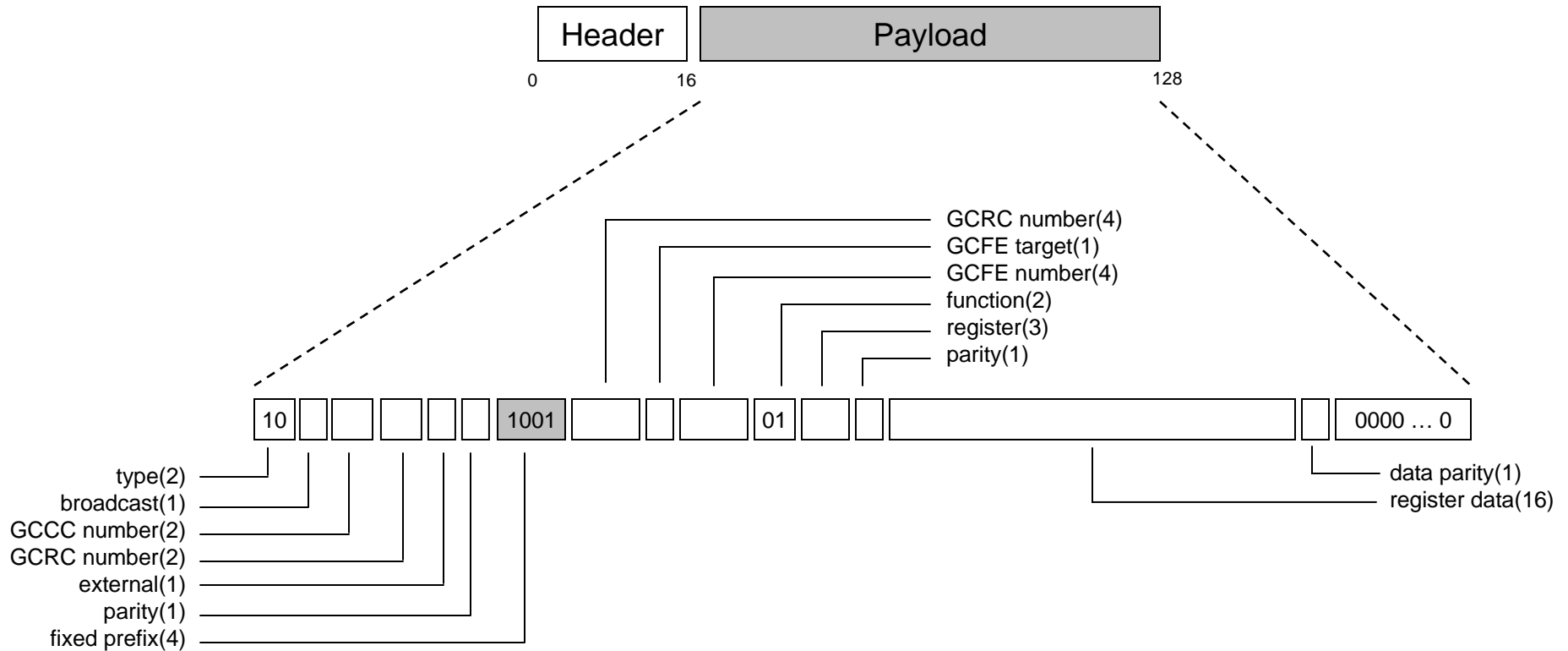


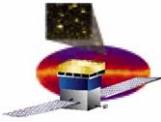
Instrument Configuration: Reaching the Registers (cont'd)

- For instance, to write a register in the CAL front-end ASIC, one specifies the path through the hierarchy, indicating:
 - Which of the 16 TEMs (LATp nodes) to address
 - Which of the 4 GCCCs (TEM board ASICs) to address
 - Which of the 4 GCRCs (off-board CAL layer readout controller ASICs) to address
 - Which of the 12 GCFEs (off-board CAL front-end ASICs) to address
 - Which of the 7 registers on that GCFE to write to
- The next slide illustrates the packet structure used to write to a register on the CAL front-end ASIC.



Instrument Configuration: Reaching the Registers (cont'd)

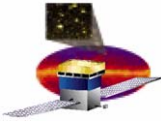




Instrument Configuration: Reaching the Registers (cont'd)

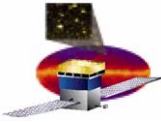
- The same messaging and addressing system is used to write and read registers on all subsystems.

Subsystem / Block	Chip / Sub-block	Number in Instrument	Register Count	Register documentation
TKR	GTFE	13824	69120	LAT-SS-00169
	GTRC	1152	2304	LAT-SS-00170
CAL	GCFE	3072	21504	LAT-TD-00424
	GCRC	256	2048	LAT-SS-00208
ACD	GAFE	216	2376	LAT-SS-00363
	GARC	12	528	
TEM	GTCC	128	768	LAT-TD-00605
	GCCC	64	448	
	GTIC	16	272	
	Common	16	112	
GASU	AEM	1 (primary and redundant)	6	LAT-TD-00639
	CRU	1 (primary and redundant)	4	LAT-TD-01547
	EBM	1 (primary and redundant)	31	LAT-TD-01546
	GEM	1 (primary and redundant)	59	LAT-SS-01596
PDU			7	LAT-SS-01543
			99587	

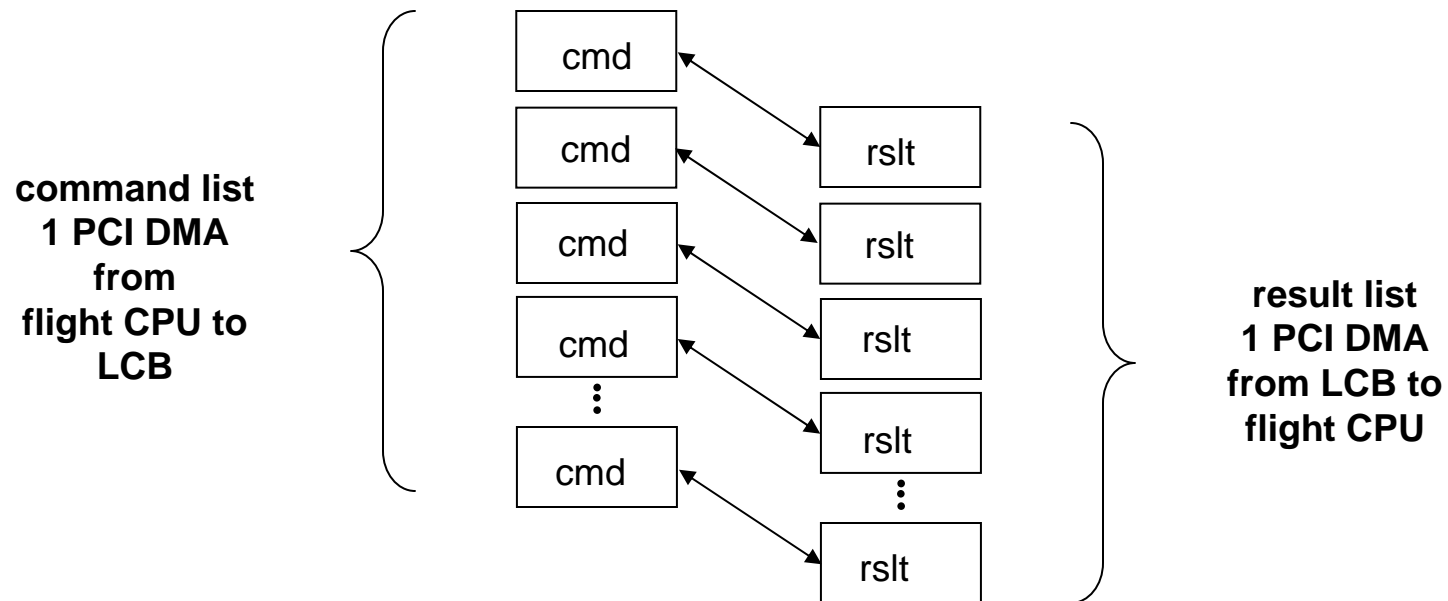


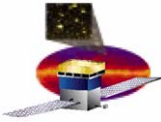
Instrument Configuration: Sending Multiple Commands

- To speed up configuration, multiple commands and responses can be bundled into DMA transfer operations between a flight CPU (e.g., an SIU) and the LAT Communications Board (LCB):
 - Multiple command items in a “command list” are DMA’d to the LCB.
 - There the command items are sent one by one to LAT subsystems.
 - The command list can be up to 4 KB long, enough space to hold 150 register write commands.
 - Multiple responses are assembled into a “results list.”
 - All result items have a timestamp and a status word.
 - For command items that expect a response, the result item also contains the data read back from the instrument subsystem.
 - The results list can also be up to 4 KB long.
 - A one-to-one relationship between items on the command and results lists is maintained.
 - The results list is DMA’d back to the flight CPU, and LCB raises an interrupt to signal the transfer is complete.



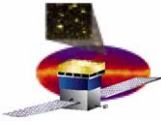
Instrument Configuration: Sending Multiple Commands (cont'd)





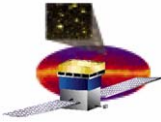
Instrument Configuration: Software Implementation

- **Communications via LATp and transmission of bit strings carrying register settings are handled by two FSW packages:**
 - **The LCBD package controls the LAT Communications Board (PCI register access) and handles LATp communications**
 - **LCBD is covered in more detail by JJ Russell and Ed Bacho.**
 - **The LEM package constructs bit strings bound for registers on all LAT subsystems: CRU, EBM, GEM, PDU, ACD, CAL, & TKR, as well as the TEM and AEM**



LEM Package Components

- **lem**
 - This constituent defines the structures of all the Command/Response LATp command items. The structures are joined together in a typedef'd union called LEM_ci and are public. This allows application developers to construct LATp commands by hand by setting the various bit fields.
 - Various macros are also defined that perform the bit packing and determine the various parity bits.
 - A functional interface, which is just a wrapper around the macros, is also provided. Developers are able to use whichever interface seems sensible.
 - The constituent also defines structures describing the various responses to a read command along with functions to break apart result item payloads into more user-friendly forms.
 - This constituent is complete and tested.
- **cli**
 - This constituent provides the command line interface allowing interaction with the LAT from the VxWorks command line. Functions in this constituent break up the payload (if necessary) and copy it into user supplied memory locations.
- **bci**
 - This constituent is a wrapper around the command line interface intended to provide run-time and compile time backwards compatibility for preexisting software that is required for testing. It will be used temporarily to give developers breathing space as hardware is upgraded.



LEM Software Package Organization

- Shareables (flight)
 - [liblem](#) – main LEM library
- Shareables (testing and ground support)
 - [liblem_cli](#) – command-line interface libraries
 - [liblem_bci](#) – interface libraries (for backwards compatibility)
 - [libt_lem](#) – LEM unit test library

- LEM directly uses the following packages/constituents:
 - [CMX](#)
 - [LCBD](#)
 - [MSG](#)
- The dependency tree, constituent list, command and telemetry list, and other configuration management information related to LSW are published dynamically on the FSW Web site at http://www.slac.stanford.edu/exp/glast/flight/web/a_pnp/Pack_LEM.shtml