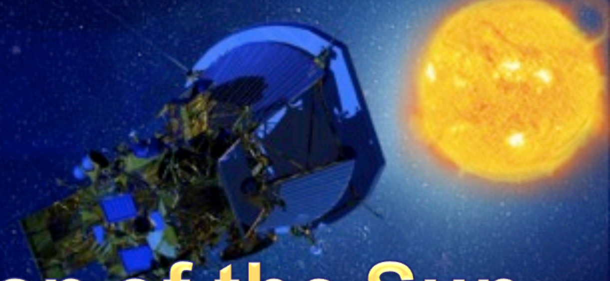


Solar Probe Plus

A NASA Mission to Touch the Sun

Integrated Science Investigation of the Sun Energetic Particles

EPI-Hi Autonomy Review

11 Nov 2015

EPI-Hi Flight Software Overview

Andrew Davis & Rick Cook

EPI-Hi Software Engineering (Caltech)

Outline

- Personnel
- Flight Software Overview – Requirements
- Operations Concept
- Flight Software Context – Architecture, main tasks
- Detailed Software Design
  - Functional description
  - Structural decomposition
  - Timing
- Commands and Telemetry
- Resource Margins
- Software Verification and Testing
- Configuration Management
- Status of Engineering Model Software

Personnel

- High-level science requirements and algorithms
  - Mark Wiedenbeck
  - Dick Mewaldt
  - Tycho von Rosenvinge
- Electrical engineer, low-level and hardware-oriented flight software
  - Rick Cook (Backup: Jill Burnham)
- Science flight software
  - Andrew Davis (Backup: Hiromasa Miyasaka)

Engineering and flight software personnel have all worked together in same roles for STEREO and NuSTAR missions

Applicable Documents

- 7434-9066 SPP_GI_ICD
- 7434-9058 ISIS_ICD
- SPP-CIT-010 EPI-Hi Flight Software Development Plan
- SPP-CIT-011 EPI-Hi Flight Software Requirements Doc
- SPP-CIT-012 EPI-Hi Instrument Flight Software Design Doc
- SPP-CIT-013 EPI-Hi Instrument Flight Software Test Plan
- SPP-CIT-014 EPI-Hi Instrument Flight Software Verification/Test Matrix
- SPP-CIT-015 EPI-Hi Telemetry Data Format & Command Dictionary

Caltech Internal Docs/Memos:

- PHASIC User Manual
- HKchip User Manual
- P24 MISC Processor Manual
- EPI-Hi Event Processing and Particle Identification Scheme
- LET & HET Dynamic Threshold Recommendations
- LET & HET Rate Data Definitions

Software Requirements

- The EPI-Hi Flight Software Requirements Document details the flow-down from higher-level requirements, and from the GI and ISIS ICDs
- These requirements are quite similar to the requirements on our recent NuSTAR and STEREO missions
- The same modular flight processor and software architecture that we used for NuSTAR and STEREO will meet the EPI-Hi requirements
- The requirements are mapped into a set of modular software tasks, implemented on the EPI-Hi flight processors

Key/Driving Requirements

- See the FSW Reqs Doc, where we find requirements related to
  - Boot/initialization
  - Time synchronization
  - Housekeeping telemetry
  - Fault protection and autonomy
  - Command processing and S/C Interface Management
  - Processing rate (events processed/second)
  - Thermal control
  - Bias supply, LVPS control
  - Inter-MISC Communications
  - Science Measurement Requirements
    - Instrument Livetime
    - Species and Energy Coverage
    - Energy Binning
    - Look-direction binning
    - Helium Isotope Identification
    - Electron Identification
    - Measurement Cadences
    - Dynamic Range in Particle Intensities

FSW Action Items from CDR (1)

1. RFA06: Should the time-tagged commanding capability be resident in the EPI-Hi DPU only, or in the DPU and each of the three telescopes? RATIONALE: This should be resolved quickly and documented. FWIW, it seems to me that the DPU capability is probably sufficient, and, being simpler, is the preferred solution.
   - Response: The EPI-Hi time-tagged commanding capability will be resident in the EPI-Hi DPU only. This is currently documented in the EPI-Hi FSW Requirements (L5-INSTSW-162), and the EPI-Hi FSW Design Document (section 3.4.3.7).
   - Status: Complete, Concurred by Requestor

FSW Action Items from CDR (2)

2. RFA07: Identify specific requirements for macros that must be provided as tested functional elements of the EPI-Hi deliverable. Identify responsible developer, schedule, and verification processes. RATIONALE: The EPI-Hi software includes significant capability to perform time-tagged macro execution. These macro sequences are software, but they do not appear to be covered by the EPI-Hi s/w development plan.

- Response: EPI-Hi is intended to operate as a fully-autonomous instrument under normal circumstances. All of the FSW capabilities and tasks required to fulfill this requirement are described in the EPI-Hi FSW Design Document.

  The time-tagged command capability for EPI-Hi is intended to fulfill a *possible* need to set a parameter or change a mode at some specific time in the future. We have not as yet identified any situation where we will need this capability, but we are implementing the capability in the EPI-Hi FSW because the SPP S/C FSW is not providing this capability. Thus there are at this point no macros defined that will be "provided as tested functional elements of the EPI-Hi deliverable".

  The algorithm for implementing the time-tagged command capability is simple, and is described in the EPI-HI FSW Design Document in Section 3.4.3.7. It uses existing capabilities of the resident Forth operating system. The capability will be tested during instrument comprehensive performance tests, as specified in the EPI-Hi FSW Verification & Test Matrix. Within the Comprehensive Performance Test Script, a few simple macros will be defined that will set a few selected instrument FSW parameters to a new value at a specific time in the future.

  Should we decide that a time-tagged command is required at some stage during the mission (or during I&T), the macro that implements the command will be thoroughly tested on our EPI-Hi EM-unit prior to uploading it to the flight unit.
- Status: Complete, Concurred by Requestor

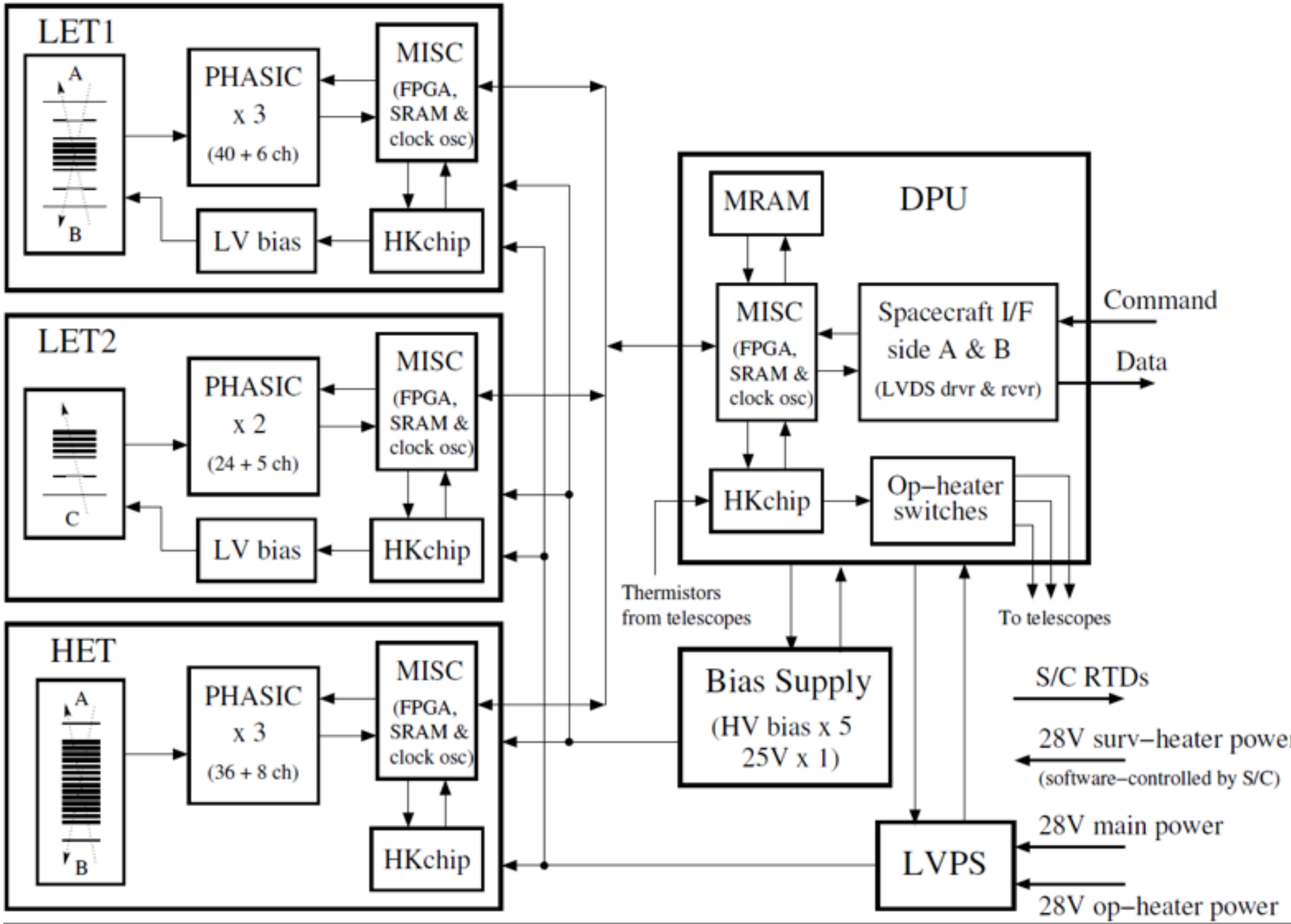
FSW Action Items from CDR (3)

3. RFA08: EPI-Hi Provide a SW Verification matrix that maps SW requirements to a verification test. RATIONALE: The EPI-Hi FSW Test Plan describes the testing phases and process, but not the content of the tests.
   - Response: EPI-Hi did in fact deliver a FSW Verification matrix at CDR, but perhaps it was not provided to the reviewers. The matrix maps each FSW Requirement to a FSW task described in the FSW Design Document, and to a verification test. Current version of FSW Verification matrix forwarded to Project.
   - Status: Complete, Concurred by Requestor

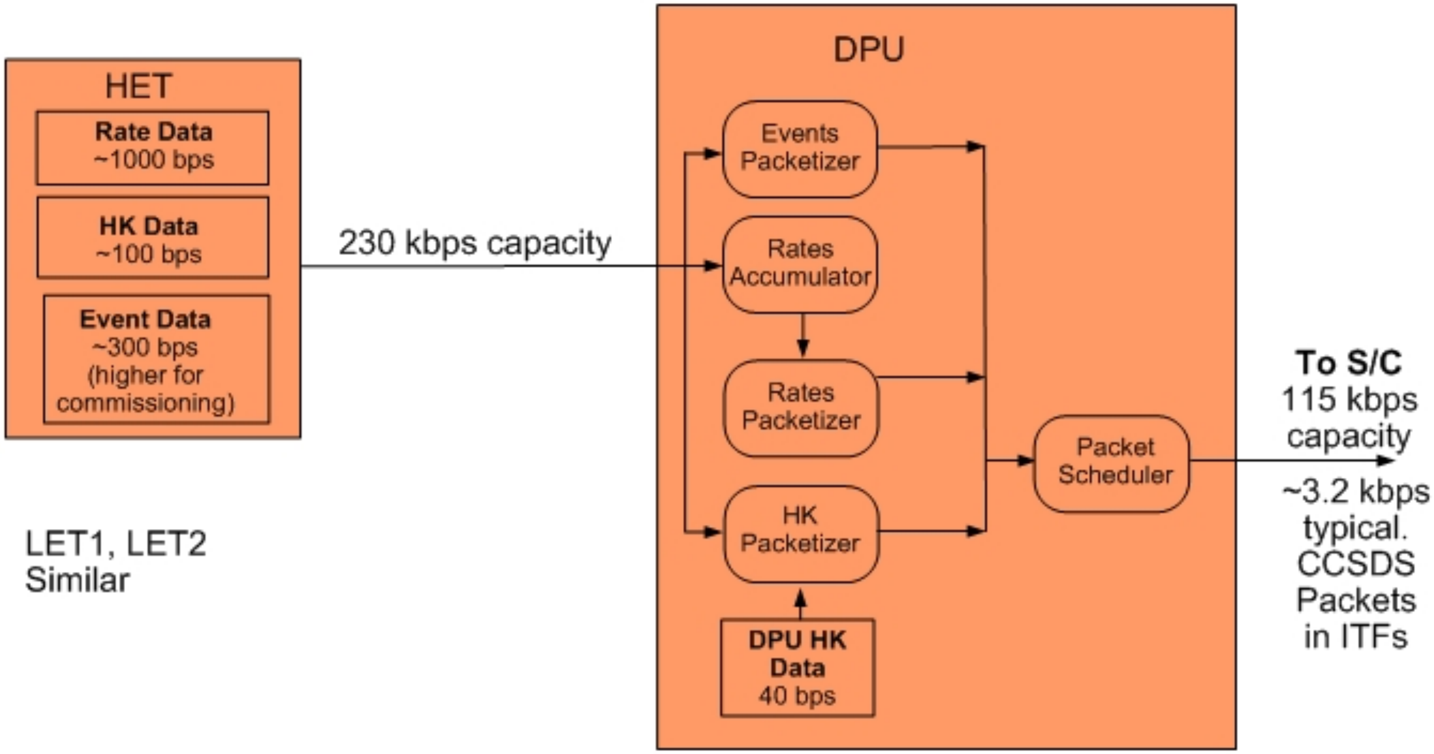
Processor and Software Architecture (1/2)

- Software distributed across 4 Minimal Instruction Set Computers (MISCs)
- 1 MISC in DPU, 1 each in LET1, LET2, HET
- Custom FPGA-embedded micro-controller
- Design implemented in Actel RTAX250SL
- Architecture used in STEREO and NuSTAR space missions
- 24-bit CPU, up to 512 kwords SRAM, Forth operating system, can boot from MRAM or serial interface
- MISC design and Forth operating system stable since 2002
- Only DPU MISC has MRAM (2M×8).
- DPU can boot from MRAM, or via serial uplink from ground
- The three “telescope” MISCs communicate with DPU MISC via RS-422 serial links, 460.8kbps
- Telescope MISCs booted via serial link by DPU MISC
- Telescope MISCs use boot images from DPU MRAM

Processor and Software Architecture (2/2)



Instrument Data Flow



Note: Typical data rates are preliminary, and subject to change based on our SSR allocation

Operations Concept (1)

- The DPU FSW orchestrates the operation of the instrument and controls the command/data interfaces with the spacecraft.
- The detector module FSW controls detector operation, data acquisition, data processing, and manages the command/data interfaces with the DPU.
- The instrument has only two normal operating modes:
  - Spacecraft-Sun Distance R<=0.25 AU (Encounter Science Mode)
  - Spacecraft-Sun Distance R > 0.25 AU (Low-rate Science Mode)
- These two modes differ only in the data transmission rate from the DPU to the spacecraft, and in how the DPU responds to fault conditions.
- The data collection rate and the operating mode of the three detector modules remains the same in these two modes.
- In both modes, the detector modules transmit the same data products at the same rate to the DPU. The DPU decides, based on the mode, what data are to be transmitted to the spacecraft to be stored on the SSR.
- Essentially, each of the detector modules has only one operating mode.

Operations Concept (2)

- Four special operational modes are currently defined. These modes are:
  - Commissioning mode: Very similar to Low-rate Science Mode. In Commissioning mode, the data transmission rate from the DPU to the spacecraft is tailored to match the real-time downlink baud-rate available during commissioning, thus optimizing the capability of the instrument team to monitor the instrument during commissioning.
  - Software upload mode: Data acquisition and some non-essential functions will be halted.
  - Calibration Science mode: Statistically significant samples of Pulse Height Analysis (PHA) event data will be accumulated and returned to validate onboard assignments of species, energy, and incidence angle, and for assessing instrument backgrounds. This mode will be activated outside 0.25 AU by command one or more times early in the mission when sufficiently high intensities of solar energetic particles are present for calibration purposes. This mode, which will be used for a few days after activation of the mode, will require the return of a data volume significantly greater than that provided by the Low-rate Science Mode.
  - Safe mode is a mode where the instrument sits in a “quiet” state, awaiting commands from the ground, performing no data acquisition functions. This mode is reserved for instrument I&T, commissioning, and instances where the flight software encounters some irrecoverable fault.
- As in the two normal operating modes, the mode change occurs in the DPU, not in the detector modules.

DPU Software Tasks

All DPU FSW requirements are mapped into one or more of the following tasks:

- Forth operating system and low-level I/O routines
- Power-on and initialization sequence management
- Telescope MISC serial boot sequence management
- Command table maintenance
- Housekeeping data collection
- Inter-MISC communication management, and data acquisition/buffering from satellite MISCs
- Support time-tagged command capability (macros), to support autonomous instrument operations during encounters
- Setup and control of instrument LVPS, bias supply
- Active control of operational heaters
- Receive/Monitor status and time-synchronization data from the spacecraft, and perform autonomous mode adjustments as needed
- Receive and process ground commands
- Management of software uploads and MRAM “burns”
- Accumulation of science rates data
- Compression of science rates data
- Formatting/packetizing and transfer of science and housekeeping data and command responses to the spacecraft
- Monitor heartbeat from peripheral MISCs, and perform autonomous diagnostics/reboot as needed
- Manage volume of instrument data transferred to SSRs on S/C as function of time

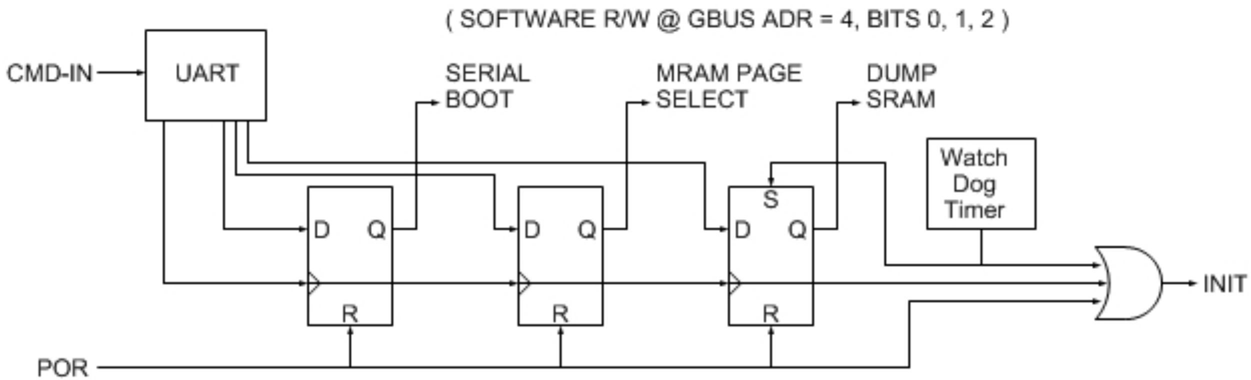
Telescope MISC Software Tasks

All telescope MISC FSW requirements are mapped into one or more of the following tasks:

- Forth operating system and low-level I/O routines
- Science data acquisition
- Detection and handling of crosstalk signals from PHASICs
- Science data processing and reduction (particle ID)
- Science and diagnostic rates data accumulation
- Command table maintenance
- Housekeeping data acquisition
- Processing of status, time-synchronization, and command data from the EPI-Hi DPU
- Send heartbeat message to DPU once per second
- Monitor key counting rates and adjust the telescope operating condition to optimize data quality (dynamic adj. of detector thresholds)
- Monitor temperatures of PHASICS and detector electronics
- Formatting and transfer of science & housekeeping data and command responses to the EPI-Hi DPU
- Setup and control of PHASICs and HKchips

Boot Sequence and Initialization

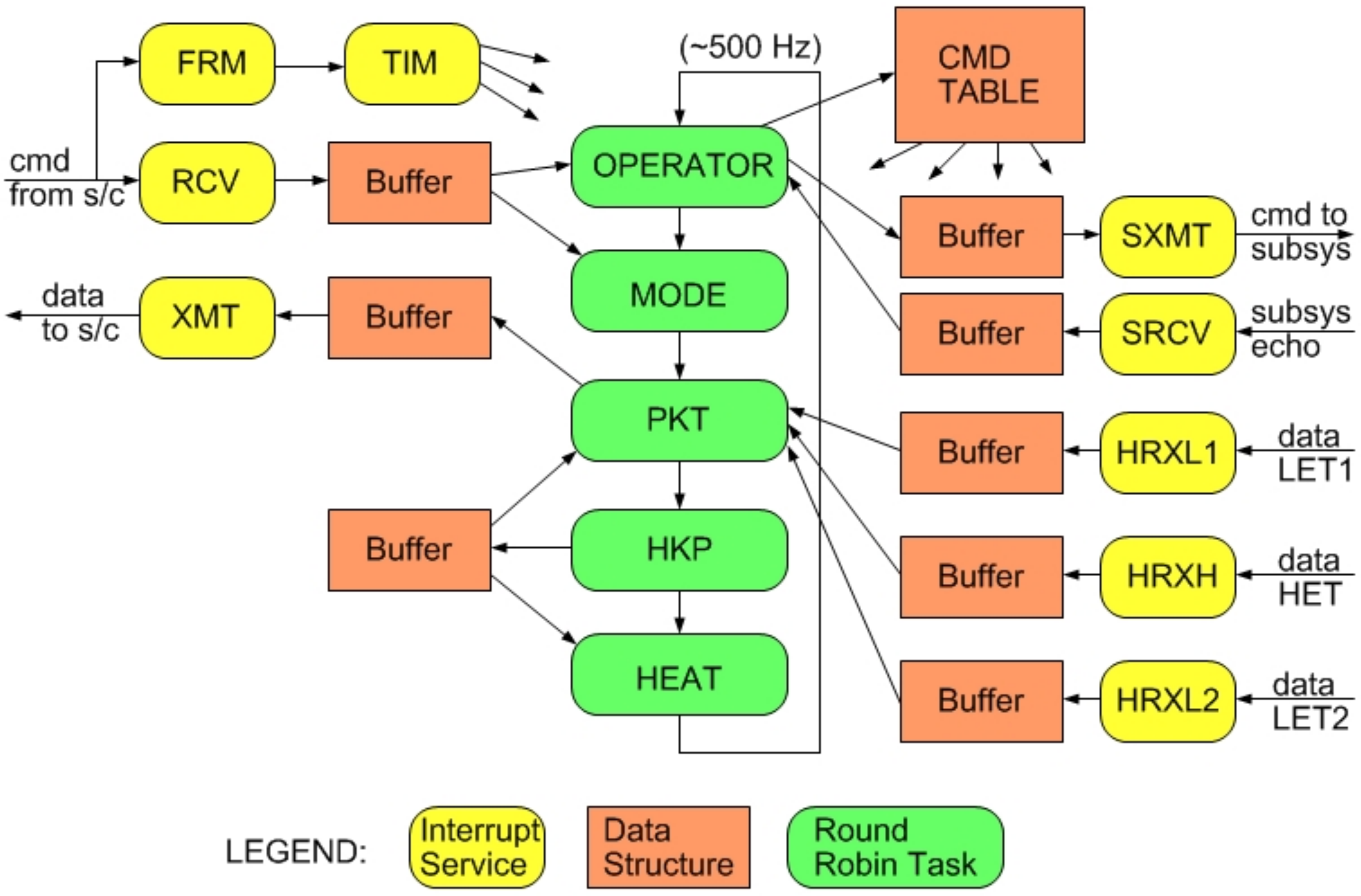
DPU MISC Reset Circuitry Block Diagram



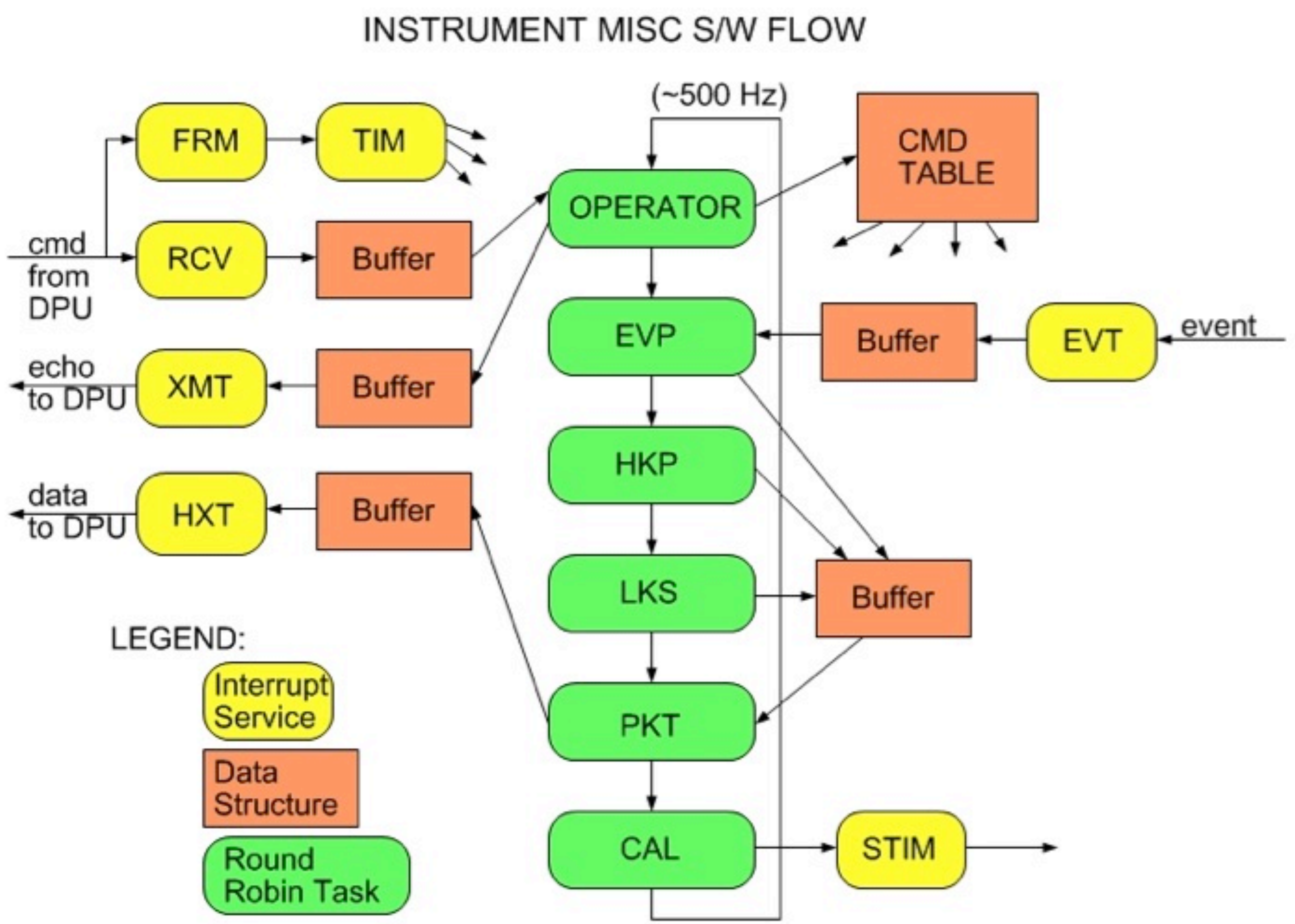
- Supports three ways to initiate boot: Power On Reset (POR), CMD-IN and Watch Dog Timer.
- “Reset Bit Field” is writable by CMD-IN and R/W by software.
- “Serial Boot” bit determines if boot is to be performed serially via CMD-IN or from MRAM.
- “MRAM Page Select” bit determined which page (0 or 1) of MRAM is source of boot images.
- “Dump SRAM” bit determines if SRAM is dumped prior to boot. This bit is set by the watch dog timeout.
- Details of instrument boot process described in Autonomy Presentation

Software Flow – DPU MISC

DPU MISC S/W FLOW



Software Flow – Telescope MISC



Inter-MISC Communication

- There are two standard RS-422 serial interfaces between the DPU and each of the telescope MISCs.
- The first is a bi-directional interface for the transfer of commands and boot code from the DPU to the module, and the transfer of command responses from the module to the DPU. This interface is multiplexed between the modules.
  - If a command message is destined for HET, LET1, or LET2, then the DPU initializes the multiplexed serial command link to the appropriate sensor based on a routing command contained in the command message, and forwards the command to the appropriate sensor.
  - This command routing software is the same as used on NuSTAR, and the SEP instrument suite on STEREO.
- The second serial interface is unidirectional, for the transfer of data from the subsystem to the DPU. The data interface is not multiplexed between the subsystems – each subsystem has its own data interface with the DPU.
  - Double-buffering is implemented in each telescope MISC: data accumulated during period N is transferred to the DPU during period N+1.
  - Double-buffering is implemented in the DPU: data received during period N is packetized and transferred to the S/C during period N+1.

Electron and Ion Event Processing

The EVP task in the main round-robin multitasking loop in each telescope MISC polls the incoming event FIFO. If an event is available in the FIFO, the EPROC task is called to process the event. EPROC consists of a set of subtasks:

- XTALK: check cross-talk flags generated by the PHASICs for each pulse-height reported for an event. Check for other known signal cross-talk features
- SORT: assign a category for each event, e.g. RANGE 2A, RANGE 3B, etc.
- MULTI: resolve any cases of multiple hits in a detector layer for an event
- EUNPACK: Select dE, Ep, and L values, for use by SCALC task
- CRIT: apply any trajectory or other consistency criteria (these will be range-dependent)
- SCALC: for each (dE, Ep, L), identify the species (electron, proton, 3He, 4He, C, …)
- ECALC: for electrons and protons, calculate energy (ETOT). For Z>=2, calculate ETOT/M
- CTR_INC: increment appropriate rate counters based on results of SCALC and ECALC

Electron and Ion Event Processing Performance

Max execution times for each EPROC subtask (usecs):

XTALK	220
SORT	35
MULTI	96
EUNPACK	50
CRIT	25
SCALC	107
ECALC	130
CTR_INC	20

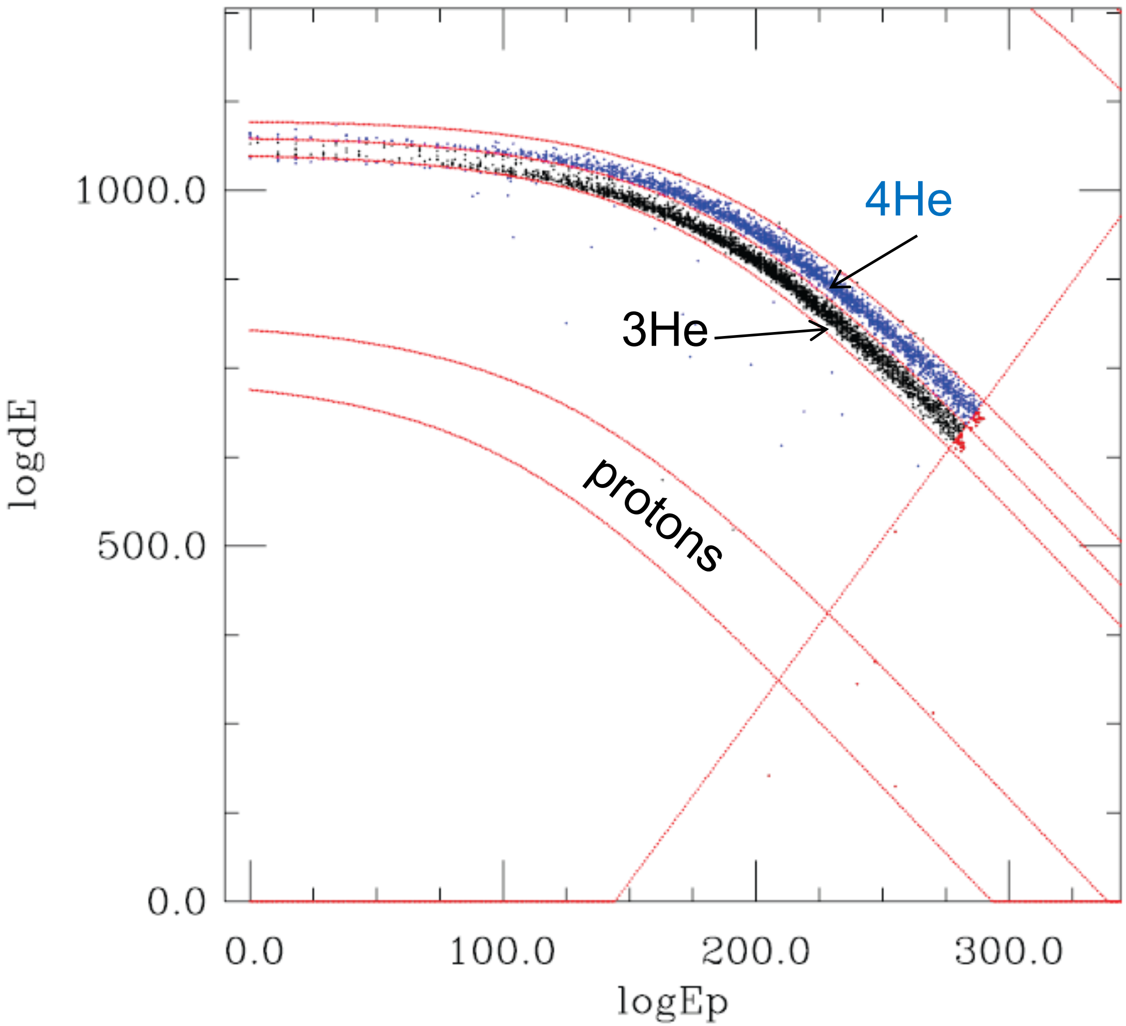
⇒ Max execution time for EPROC is 683 usecs

⇒1464 events processed/second.

Execution times are for an event with a large number of pulse-heights. Most events will be short-range, with only a few pulse-heights. Short-range events can be processed at a rate of ~2500/second.

The requirement is to process proton and electron events at a rate of at least 1000 events/second.

Onboard Particle Identification (SCALC)



- Output from a LET1 Monte Carlo simulation of 3He and 4He events was used as input to the latest version of our onboard particle ID algorithm, implemented on a MISC test board
- Onboard algorithm maps events into a 2D dE-Eprime space, where boundary-curves define different regions for different species
- SCALC processes events at ~ 9400 events/second

Science Rates Accumulation and Compression

We have specified the division of labor between the telescope MISCs and the DPU MISC for these tasks as follows:

- Telescope MISCs perform all rate accumulations up to 1 minute cadence
  - 1 minute ion rates
  - 1 minute sectored rates
  - 1 minute ENA rates
  - 1 minute Neutral rates
  - 1 minute Pixel rates
  - 10 second electron, proton, He, Z>=6 rates
  - 10 second Neutral rates
  - 1 second electron and proton rates
- DPU MISC performs all rate accumulations for cadences > 1 minute
  - 1 hour science rates
  - 5 minute sectored rates
- DPU MISC implements compression of all rates prior to packetization
  - Take a list of N fixed-length rates and telemeter only those that are non-zero; a list of N tag bits is attached as a header to each block of N rates to indicate which rates made the cut.
  - Tested on STEREO rates data, this simple scheme results in compression factors of ~1.4 during large SEP events, to ~5 during solar quiet times.

DPU Timing Cycles

- Asynchronous
  - Receive command echoes from telescope MISCs
- Every 1 second
  - Read Time/Status from S/C Command ITF
  - Check for/interpret commands in S/C Command ITF
  - Send SYNC to telescope MISCs
  - Read rates/events/HK data blocks from telescope MISCs
  - Packetize/transfer command echoes to S/C (as needed)
  - Packetize/transfer event data to S/C
  - Update heartbeat status for each telescope MISC
- Every 4 seconds
  - Acquire DPU HK data
  - Update op. heater control parameters
- Every minute
  - Packetize/transfer 1sec, 10sec, and 1min rates to S/C
  - Packetize/transfer selected HK data to S/C
  - Accumulate 5min and 1hr rates (from 1min rates)
  - Do command table checksum check
- Every 5 minutes
  - Packetize/transfer 5min rates to S/C
- Every 10 minutes
  - Packetize/transfer 10min rates to S/C
- Every hour
  - Update data->SSR metering parameters
  - Packetize/transfer 1hr rates to S/C

Telescope MISC Timing Cycles

- Asynchronous
  - Process events in the event FIFO from the front-end electronics
  - Check for/interpret commands from DPU
  - Xfer Command responses to DPU, if needed
- Every 1 second
  - Receive/decode SYNC from DPU
  - Xfer 1s rates for previous second to DPU
  - Xfer prioritized event data to DPU
  - Update CAL/STIM pulser settings
- Every 4 seconds
  - Acquire HK data, transfer to DPU
- Every 10 seconds
  - Xfer 10s rates for previous 10 seconds to DPU
- Every minute
  - Xfer 1minute rates for previous minute to DPU
  - Update dynamic threshold state parameters
  - Do command table checksum check

Telemetry and Commands

The EPI-Hi Telemetry Format Document contains the following:

- Packet ApID assignments for instrument CCSDS packets. EPI-HI is assigned 96 ApIDs.
- Telemetry data formats and instrument CCSDS packet definitions
- Event data format
- Cadences for Rates data telemetry for each operating mode
- The routine command dictionary

Predicted Memory, CPU Margins

- **LET1 MISC (LET2, HET similar)**
  - SRAM 512kwords available (512k x 24 bits)
    - Basic OS, HK, I/O code + tables 20kwords
    - Event processing, particle ID 230kwords
    - Rates counters, double-buffered 10kwords
    - Event priority buffers 25kwords

    Total SRAM usage ≈ 285/512 = 56%
  - MIPs 12 **60% used** (Maximum, dominated by event processing during intense SEP events)
- **DPU MISC:**
  - SRAM 128kwords available (128k x 24 bits)
    - Basic OS, HK, I/O code + tables 20kwords
    - Detector module data buffers, & rate accum, double-buffered 30kwords
    - CCSDS Packet buffers 30kwords

    Total SRAM usage ≈ 80/128 = 63%
  - MRAM 2M x 8 bits **~70% used** (2 sets of boot images)
  - MIPs 12 **30% used**

Inline with APL Space Exploration Sector guidelines for processor and memory margins

Integration, Testing and Verification

The EPI-Hi Flight Software Test Plan document defines this plan in detail. The plan includes a software requirements verification matrix.

- Software tests will start at the module level. As the code builds up, system-level testing will start, and the majority of the test time will be targeted at the system level.
- A Test Readiness Review will precede formal acceptance testing. SPP Project and ISIS personnel will oversee/participate in this review.
- Formal acceptance tests will be performed on the EPI-Hi instrument, including the flight software, during the EPI-Hi integration and test phase.
- During environmental tests and SPP I&T, more experience will be gained with the flight software, real sensor data, and with controlling the instrument via the SOC-MOC interfaces. Flight software changes may be expected as a result. Any changes in the flight software at this stage will result in CCB review, and a repeat of these acceptance tests.
- The acceptance tests for the EPI-Hi flight software will be designed to verify each of the software functional requirements as called out in the requirements documents and also the functions provided by the Forth operating system, the I/O API, and the multitasking software running on the MISC processors.
- Most of the acceptance tests are enshrined in a **scripted Comprehensive Performance Test** that is run at regular intervals during environmental tests and I&T by the EPI-Hi team.
- **A software requirements verification matrix is maintained by the EPI-Hi team** to aid in verifying that the software meets the requirements. The matrix is available for review by the SPP Project. A preliminary software requirements verification matrix will be presented at CDR and a final version will be presented at the Software Acceptance Review, following successful completion of the acceptance tests.
- A Software Acceptance Review will follow the successful completion of the acceptance tests, and the EPI-Hi team will present a test report. SPP Project will oversee/participate in this review.

Software Maintenance, Configuration Management

Solar Probe Plus
A NASA Mission to Touch the Sun

A Configuration Management Plan is contained in Software Development Plan

- Version control and software archiving: achieved via subversion version control system (SVN) (http://subversion.apache.org/). The subversion server is hosted and maintained by Caltech Space Radiation Lab IT personnel. It is access-controlled for security, and is backed-up regularly.
- The lead engineer will maintain the structure of the SVN repository, including separate source trees for each of the MISC's flight software, separate directories for different software builds, etc.
- All problems and change requests will be documented/tracked in the Software Development Log by the lead engineer.
- After the beginning of I&T with flight hardware, all flight software changes will be approved by a CCB prior to being loaded into flight hardware. Also after this point, Version Description Documents (VDDs) will accompany all software releases. The VDD will contain the functionality of the release, a list of closed software problem reports, a list of any liens/workarounds, installation instructions, and a list of deliverable source code files.
- The CCB membership will consist of the EPI-Hi lead engineer and software developer, and the EPI-Hi Project Manager, plus any other ISIS or SPP Project engineers that the QA team at SwRI deems necessary.

Status of Engineering Model Software

- FSW running on EPI-Hi EM-unit in “Flatsat” configuration
- GSEOS EGSE communicating with EM-Unit via either mini or “full” spacecraft emulators
- HK, event, rates, and command-response data flowing from Detector modules to DPU
- Data in CCSDS packets, encapsulated in ITFs, flowing from DPU to GSEOS
- DPU interpreting virtual 1 PPS time/status ITF from S/C emulator
- Commands and command responses flowing

DPU FSW Status

DPU MISC Flight Software Task	Heritage	% Done	Assignee	Notes
Forth operating system and low-level I/O routines	95%	100%	WRC	
Power-on and initialization sequence management	60%	80%	WRC	
Detector module MISC serial boot sequence management	80%	100%	WRC	
Command Table Maintenance	100%	100%	WRC	
Housekeeping data acquisition	50%*	100%	WRC	
Inter-MISC communication management, data acquisition and buffering from Peripheral MISCs	90%	100%	WRC	
Support time-tagged command capability (macros)	0%	40%	WRC	Algorithm drafted, under review by science team
Setup and control of instrument LVPS and bias supply	50%*	100%	WRC	
Active control of operational heaters	80%*	100%	WRC	
Receive/Monitor status, and time-synchronization data from the spacecraft, and perform autonomous mode adjustments as needed	10%	40%	WRC	Rec/monitor status and time-synch data part of task is 100% done. Perform autonomous mode adjustments part is 0% done.
Receive and process ground commands	100%	100%	WRC	
Management of software uploads and MRAM burns	75%	50%	WRC	
Accumulation of science rates data	75%*	60%	AJD	Acquisition of rates data from telescope modules is done. Accumulation of rates into longer-cadences is not done.
Compression of science rates data	0%	50%	AJD	
Formatting/packetizing and transfer of science & housekeeping data and command responses to the Spacecraft	50% *	80%	WRC and AJD	General data packetization scheme done. HK, events, Cmd-response and preliminary rates data packetization done. Final rates packets awaiting full definition of science rates
Monitor heartbeat from peripheral MISCs, and perform autonomous diagnostics/reboot as needed	60%	60%	WRC	
Manage volume of instrument data transferred to SSRs on S/C as function of time	0%	40%	AJD	Algorithm drafted, under review by science team
* general scheme is inherited, specifics are unique				

Detector Module FSW Status

LET1, LET2, and HET MISC Flight Software Tasks – each MISC performs a similar set of functions:	Heritage	% Done	Assignee	Notes
Forth operating system and low-level I/O routines	95%	100%	WRC	
Power on and initialization sequence management	60%	100%	WRC	
Science data acquisition	50% *	100%	WRC	
Detection and handling of crosstalk signals from PHASICs	50%*	60%	WRC/AJD	Algorithm for implementing this task is done. Awaiting specifics of characteristics of 16-bit crosstalk register from PHASIC
Science data processing and reduction (particle ID)	20%	70%	AJD	Software for analysis of ions/electrons is written and partially tested. Needs further testing and refinement with real particle event data. Algorithms for ENA, Neutrals, and penetrating particles handling needs input from science team.
Science and diagnostic rates data accumulation	50%*	80%	AJD	Need to implement particle species/energy bins
Command Table Maintenance	100%	100%	WRC	
Housekeeping data acquisition	50% *	100%	WRC	
Processing of status, time-synchronization, and command data from the EPI-Hi DPU	50% *	100%	WRC	
Send heartbeat message to DPU once per second	50%	100%	WRC	
Monitor key counting rates and adjust the telescope operating condition to optimize data quality	80%	70%	WRC	Refinement of STEREO dynamic-threshold software for EPI-Hi
Monitor temperatures of detector electronics and PHASICs	100%	100%	WRC	
Formatting and transfer of science & housekeeping data and command responses to the EPI-Hi DPU	50% *	80%	WRC and AJD	Need final definition of rates data blocks to complete this task
Setup and control of PHASICs and HK chips	50% *	100%	WRC	
Detector Leakage Current Monitoring and Balancing	100%	100%	WRC	
Control of Electronics Test Pulser Cycling	92.8%	71.3%	WRC	
* general scheme is inherited, specifics are unique				

FSW Schedule

APL WBS	Task Name	% Complete	Duration	Start Slack	Start	Finish
2.4.G	EPI-Hi SW Support 3 for TX A&M test	100%	41 days	0 days	Mon 15-07-13	Tue 15-09-08
2.4.G	EPI-Hi SW Build 1 to EPI-Hi Continued Test and Integration	100%	5 days	0 days	Mon 15-09-14	Fri 15-09-18
2.4.G	EPI-Hi SW Build 2 to EPI-HI Continued Test and Integration	0%	44 days	84 days	Mon 15-10-19	Mon 15-12-21
2.4.G	EPI-Hi SW Build 3 to EPI-Hi Continued Test and Integration	0%	44 days	84 days	Thu 16-01-21	Tue 16-03-22
2.4.G	EPI-Hi SW Build 4 to EPI-Hi Continued Test and Integration	0%	44 days	84 days	Wed 16-04-20	Tue 16-06-21
2.4.G	EPI-Hi Archive Prepare and Document	0%	44 days	84 days	Thu 16-07-14	Wed 16-09-14
2.4.G	EPI-Hi SW Archive Final	0%	30 days	84 days	Thu 16-10-06	Wed 16-11-16

Backup Slides

DPU Interrupt Analysis

DPU Interrupt Analysis

description	priority	maximum execution time (clocks)	maximum execution time (usec)	min interrupt period (usec)	max allowed servicing delay (usec)	maximum cpu load (%)	average rate (/sec)	average (*) cpu load (%)
timer	10	475	32.1	2000	1965	1.6	500	1.6
gse tx uart	9	105	7.1	95.5		7.4	0	0.0
gse rcv uart	8	90	6.1	95.5	763.9	6.4	0	0.0
sc tx uart	7	130	8.8	95.5		9.2	500	0.4
sc rcv uart	6	179	12.1	95.5	763.9	12.7	500	0.6
tele tx uart	5	60	4.1	47.7		8.5	0	0.0
tele rcv uart	4	70	4.7	47.7	381.9	9.9	0	0.0
het rcv uart	3	51	3.4	47.7	381.9	7.2	200	0.1
let2 rcv uart	2	51	3.4	47.7	381.9	7.2	200	0.1
let1 rcv uart	1	51	3.4	47.7	381.9	7.2	200	0.1
frame	0	212	14.3	1000000		0.0		0.0
						--------------		--------------
						77.4		2.9

(*) overestimated due to use of MAX execution times

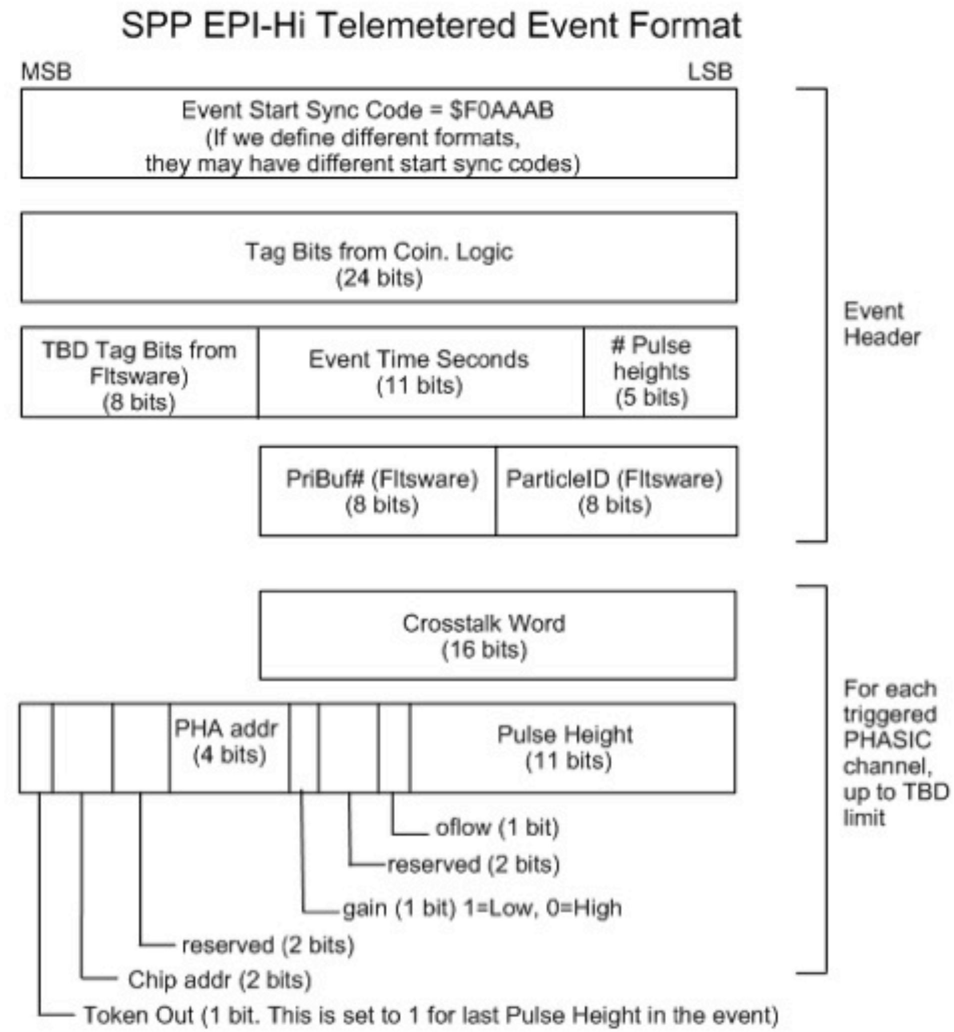
Telescope Board Interrupt Analysis

Telescope Board Interrupt Analysis

desciption	priority	maximum execution time (clocks)	maximum execution time (usec)	min interrupt period (usec)	max allowed servicing delay (usec)	maximum cpu load (%)	average rate (/sec)	average (*) cpu load (%)	
timer	10	475	32.1	2000	1965	1.6	500	1.6	
cmd tx uart	9	105	7.1	47.7		14.9	0	0.0	
cmd rcv uart	8	90	6.1	47.7	381.9	12.7	0	0.0	
data tx uart	7	63	4.3	47.7		8.9	200	0.1	
event	6	300	20.3	500		4.1	2000	4.1	(**)
frame	5	212	14.3	1000000		0.0	1	0.0	
						--------------		--------------	
						42.2		5.7	

(**) this interrupt rate is throttled
the max execution time is actually more like 700 cycles, but 300 is a practical limit for realistic event lengths

Telemetered Event Data Format



Notes: Events are packed into fixed-length CCSDS packets with multiple events per packet. Events may span packets, i.e. an event starting near the end of packet N may continue into the beginning of packet N+1.