Onboard Event Processing Scheme for the HET and LET1, and LET2 Sensors on Epi-Hi

Andrew Davis, Version 3, Sept 10 2013

Changelog

Version 3: See Appendix A for proposed species ID Algorithm

Introduction

This document is intended to document the event processing scheme for the LET1, LET2 and HET sensors on the Epi-Hi instrument. This version (1) is intended to be a draft proposal for the general charge (Z) and mass (M) particle identification algorithms. Other aspects of the event processing will be fleshed-out in later versions.

Event Processing Requirements:

- Identify ion Charge (Z), and (for selected Zs) Mass (M), given the energy (dE, in MeV) it deposits in a detector of thickness L of silicon, and its residual energy (Ep, in MeV) when it exits that detector. i.e., we know dE, Ep, and L. Energy range ~1 – 100 MeV/nuc. (species dependent energy-range, exact ranges TBR).
- Identify electrons with an energy range ~0.5 – 6 MeV (exact range TBR).
- Calculate the energy of each identified ion and electron with the accuracy required to fulfill the L4 requirements.
- Sort ions and count electrons into energy, angular-distribution, and time bins per the L4 requirements.
- Be able to process at least [300, TBD] helium events per second, including both Z and M calculations. Since helium (and neon) event processing requires the most CPU resources, other mixes of incoming events will be processed faster than this. Note: the maximum coincidence event rate incoming through the front-end electronics FIFO will be ~1000 - 2000/sec (exact number TBD).

CPU and RAM Resources

RAM: 512Kwords total available (each word is 24 bits wide)

CPU: Perhaps it is best to illustrate the MISC CPU constraints with examples. These examples apply to the current version of the MISC floating-point library. The MISC can perform the following operations at the following rates:

- Integer multiply (or divide): ~218,000/sec
- Floating-point multiply (or divide): ~59,000/sec
- Floating-point add (or subtract): ~40,000/sec
- Natural Logarithm: ~714/sec
- A^x: ~550/sec

These are rough numbers; there is some dependence on the values of the inputs. Clearly, logs and powers, if needed, must be implemented using table lookup or approximation. Floating-point add/subtract and multiply/divide operations must be minimized.

Event Processing Overview

The overall architecture of the event processing task, and the method by which it is called from the main round-robin multitasking loop, is the same as for STEREO LET.

Within the flight software of each of the Epi-Hi sensors, the event processing software exists as a task (or module). Let's call the task EPROC. Periodically, a governing task polls the incoming event FIFO, and, if an event is available in the FIFO, the EPROC task is called to process the event. Based on analysis of the event, EPROC increments certain counters, and sets the "priority" of the event. The EPROC task then exits.

Other tasks manage the counters, and manage the selection of events for telemetry based on the event priorities set by EPROC. These tasks are not discussed here.

Ignoring STIM events and HAZARD flags for brevity, The EPROC Task can be broken into a series of sub-tasks. A proposed set of subtasks (derived from the STEREO LET event processing) follows:

- XTALK: check cross-talk flags/data generated by the PHASICs for this event. Check for other known cross-talk features.
- SORT: sort events into categories, e.g. RNG2A, RNG3B, etc.
- MULTI: handle events with multiple hits in a detector layer.
- EUNPACK: Select dE, Ep, L, values, for use by SCALC task
    - For deeper ranges, there may be multiple sets of dE, Ep, L
- 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, …)
- CRIT2: apply consistency criteria for events with multiple sets of dE, Ep, L
- ECALC: For electrons and protons, calculate ETOT. For Z>=2, calculate ETOT/M.

These sub-tasks are called sequentially within EPROC. At each stage in the sequence, a decision is made whether to continue to the next stage, or to halt the processing of the event. If the decision is to halt, then the appropriate counters are incremented, and the appropriate event priority is set. No events go uncounted.

We now discuss each of these sub-tasks in detail.

XTALK

The crosstalk information contained in the raw event data generated by the PHASICs is examined. See the latest version of the PHASIC manual for details. Based on the results, the processing of the event may be halted.

Experience may dictate that further analysis of actual event data may be needed to filter out crosstalk. This analysis would be similar to that done for STEREO LET, but might be more complicated for Epi-Hi. This is TBD.

SORT

Each raw event has some number N ADC values that have been read out from the PHASICs. Only a subset of the possible permutations of ADC hits that generate valid coincidences will be valid for the purposes of further onboard event processing.

The SORT task assigns an event a category based on the ADC hit pattern. Example categories for valid events are “Range 2 A”, “Range 3 B” etc.

For STEREO LET, there were just 64 possible detector-layer permutations (ignoring the multiple segments in each layer). For the Epi-Hi sensors, there are many more possible permutations, plus there are the guards to consider. Also, different ranges will have different rules for “problematic” detector segment combinations. In January 2013, I made a first stab at categorizing the different permutations for LET1. More work is required here.

MULTI

The flight software must handle multiple hits in the same detector layer. For STEREO LET, multiple non-crosstalk hits in a layer resulted in a halt of further processing of the event. We must decide what the appropriate actions are for the Epi-Hi sensors. There may be different rules for each layer in each sensor. All this is TBD.

EUNPACK

Fully unpack the raw event data into data structures suitable for further event processing. Identify dE, Ep, L, values, for use by SCALC module. For deeper ranges, there may be multiple sets of dE, Ep, L, values.

CRIT

Apply any trajectory or other consistency criteria (these will be range-dependent). These are TBD.

SCALC

For each set of dE, Ep, L values, separate electrons from ions. For ions, identify the species.

See Appendix A for the proposed flight software species ID algorithm.

CRIT2

For events with multiple sets of dE, Ep, L values, apply consistency criteria. These are TBD.

ECALC

For electrons and protons, calculate ETOT. For Z>=2, calculate ETOT/M.

Appendix A: Proposed Flight Software Species Identification Algorithm

We assume that the algorithm has the following inputs: dE, Ep, and L.

Recall that L is the amount of material traversed in the dE detector. (It is NOT the thickness of the detector.) Ignoring any detector thickness variations, L is quantized. For LET1 (and LET2), there are 4 values of L for each Range. Thus, if there are 7 Ranges in LET1, there could be up to 28 unique Ls to consider. In practice, many of these 28 Ls will have the same nominal value.

For STEREO HET/LET, for a given Range, we identified species by mapping the dE and Ep signals into a 400x128 2D matrix. We would ideally have used a unique matrix for each L, but there was not enough RAM, so we had to “angle-correct” the dE and Ep signals, so that we could use a single 2D matrix for all Ls in a given Range.

In this algorithm, instead of defining a 2D matrix for each L, we instead define a set of 1D “boundary tables” in dE-Ep space. For each species, two 1D boundary tables are required to define the region in dE-Ep space occupied by that species. A 3rd boundary table is also required, but it is common to all species, and defines the (roughly) diagonal cut-line in dE-Ep space that separates forward-going from backward-going particles. In some cases, adjacent species may share a boundary table.

For several reasons, it is useful to work in a 2D “log-space”, as was done for STEREO. So, for instance, we transform an L1 signal to log-space as follows:

logL1 = INT*(N (log(L1) - log(L1THRESH))),   where N = DIM/log(DYNAMIC_RANGE)

L1THRESH is the nominal lowest-possible L1 threshold, DYNAMIC_RANGE = Full_scale/L1THRESH, and DIM is the value that we want for logL1 at full-scale. INT* indicates an integer function with proper rounding.

Note that the transformation of dE and Ep to log-space is done by table-lookup in flight software. There are no logarithm calculations done in the flight software. A separate log-lookup table is required or each (Full_scale, THRESH) pair, but each table needs only 2048 entries (we have 11-bit ADCs).

Once a dE or Ep signal is transformed to log-space, any required gain or detector-thickness corrections are done as “add” operations, rather than “multiply” operations.

The 1D boundary tables are also defined in this dE-Ep log-space. To 1st order, these boundary tables can be calculated from L, using range-energy software, but further empirical tweaking will be required to optimally fit empirical data. The number of entries in each boundary table is set be the DIM used in the transformation of Ep signals to log-space, and the Ep energies used to compute the tables are chosen to be evenly-spaced in logEp space.

Figures 1 and 2 show what this all looks like, for LET 1, Range=2 (L=25um), and with DIM=2048 for dE, and DIM=512 for Ep. The boundaries shown are calculated from Mark's range-energy software, and are only approximately correct at this time.

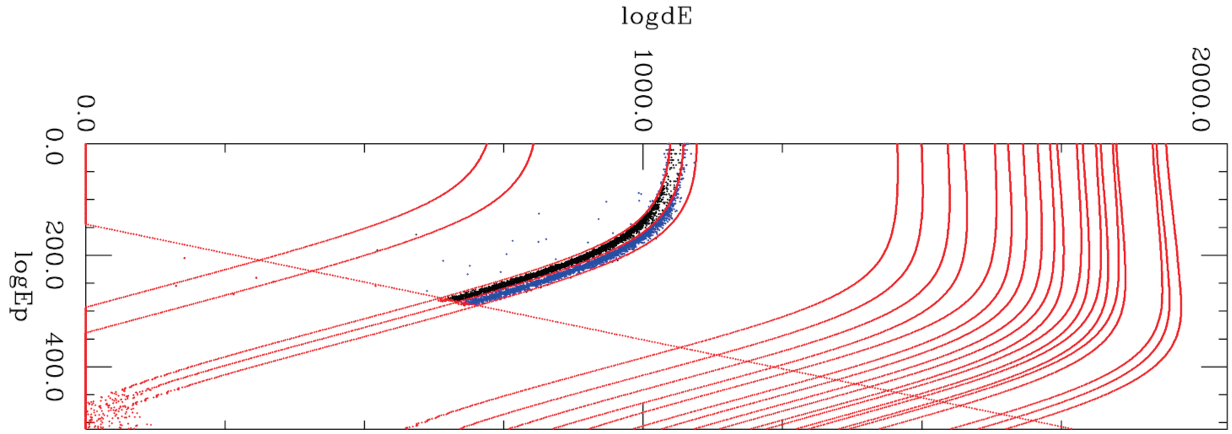

Figure 1: *EPI-Hi LET Monte Carlo Range 2 He events, plotted in 512×2048 log-space, with species boundary-curves superimposed. The boundaries shown would allow identification of p, 3He, 4He, C, N, O, Ne, Na, Mg, Al, Si, Ar, S, Ca, Fe, and Ni. Boundaries for e^- not shown. NOTE: the figure is rotated to fit!*

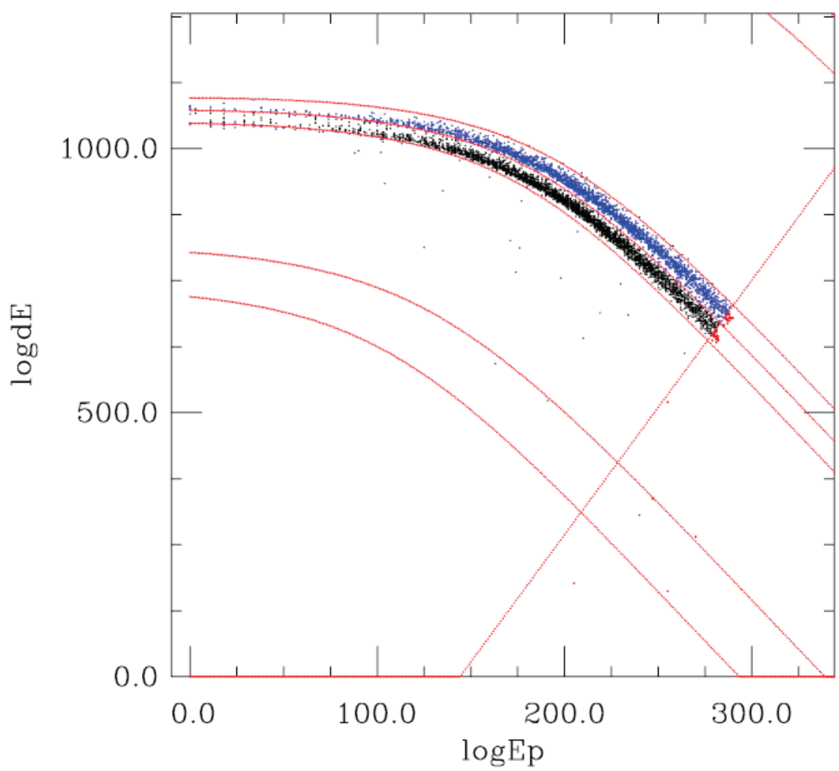

Figure 2: *EPI-Hi LET Monte Carlo Range 2 He events, plotted in 512×2048 log-space, with species boundary-curves superimposed. Same as Figure 1, but zoomed (and rotated) to show He data.*

The data plotted in Figures 1 and 2 are results from Forth event-processing software running on a MISC test unit, with “hi-res” He data from Mark’s Monte Carlo used as input. Raw dE and Ep signals (channels) from the Monte Carlo output were fed to the Forth software on the MISC. For each event, the software calculated logdE, logEP, and identified the species. Events identified as 4He by the MISC software are colored blue in the plots. Events identified as 3He are colored black. Events failing the “penetrating” cut are colored red.

MISC Performance and Memory requirements

From figures 1 and 2, it appears that ~512 entries in each boundary-table will be adequate (DIM=512 for logEp). Thus, each 1D boundary-table will require 512 words. Each species will require 2 boundary tables. Assuming we define ~22 species, the boundary-table RAM requirements for a given L will be 22×2×512 = 22.5Kwords. Let’s round up to 23Kwords to allow for defining the penetrating cut-line. This overstates the RAM requirements, since some adjacent species will have a common boundary.

We have 512Kwords available. Thus, we may define boundaries for 12 unique Ls using a little more than half the available RAM.

Note: we could conserve RAM (at expense of CPU cycles) by storing two table-entries per 24-bit word.

Given dE, Ep (raw channel values), and Range, the current MISC Forth software can process events at a rate of ~9,600/second.

The current event-processing test-code looks like this:

```
FOR Each Event i
    From dE, calculate logdE
    From EP, calculate logEp
    IF (Penetrating) Then
        Species(i) = PEN
    ELSE
        Species(i) = FindBoundary
    ENDIF
ENDFOR
```

Calculating logdE (or logEp) includes the following operations:

- Subtract channel offset
- Check ADC bounds (0 < ADC < 2047)
- Lookup log(dE) via table lookup
- Apply Gain and thickness corrections (ADDs in log-space)
- Subtract log(THRESH)

The FindBoundary routine performs a binary-search through the boundary-tables, using logEp as the index. Z=1 boundaries are checked prior to beginning the binary-search.