AESOP-Lite Tracker Boards

Robert Johnson
June 26, 2015
Legacy

Fermi-LAT Tracker

- 36 SSD layers
- 73 m² of Si
- 885,000 readout channels
- 160 Watts power

Launched in June 2008.

Still operating flawlessly in orbit after 7 years.

6/26/2015 AESOP-Lite Tracker
Fermi-LAT: Complex Electronics Integration

Top view of 4 Tracker Modules

<18 mm from active Si to active Si!

1 Tracker Tray

MCM

Right-angle interconnect
Very tight space for electronics
High precision carbon-composite structure to maintain 2.5 mm gaps between modules

6/26/2015 AESOP-Lite Tracker
Fermi-LAT Tracker Electronics

- ASIC based, for minimum power (180 μW/ch).
- Digitize on chip:
  - No coherent noise or pedestal variation!
- Single threshold (0 or 1).
- ToT on trigger OR.
- Internal calibration system.
- Threshold & Cal DACs.
- Redundant 20 MHz serial control and readout paths.
- 4 event buffers at front end → negligible deadtime (few μs).
Fermi-LAT Tracker Performance

Hit efficiency (in active area) >99.4%
Overall Tracker active area fraction: 89.4%
Noise occupancy <5×10⁻⁷
(with small number of noisy channels masked)
Power consumption 160 W (180 μW/ch)
Time-over-threshold 43% FWHM

Note: the pCT system does not do this, but we could program it into AESOP-Lite, using the FPGA.

Muon time-over-threshold (OR of all channels per layer)
The proton beam is tuned such that the protons stop in the tumor, depositing most of their energy there.

Compared with photon radiation (X-ray or γ-ray), a higher dose can be delivered to the tumor while minimizing exposure to surrounding tissue.
Proton Computed Tomography

X-ray CT use in proton cancer therapy treatment planning can lead to large uncertainties in range determination, which limits its use in the case of some tumors located close to critical healthy tissue.

Range Uncertainties (measured with PTR)
- > 5 mm
- > 10 mm
- > 15 mm


Proton CT can measure directly the density distribution needed for range calculation and is less affected by intervening dense structures.
pCT Scanner Concept

The tracking planes are composed of 8 layers of silicon-strip detectors:
- Fermi-LAT surplus SSDs
- New custom ASIC
Example pCT Scanner Proton Event

Si Tracker

PMT Signals

Event 7

Stage 0

Stage 1

Stage 2

Stage 3

Stage 4

6/26/2015 AESOP-Lite Tracker
Note: The SSDs had their edges sawn off to reduce the gaps. This resulted in high leakage current, but the current is all on the edges and does not affect the strip noise.
Through buffering and high-speed serial communication, the system sustains an event output rate of a bit more than a million per second.

The ASICs were specifically designed to support this high rate and thus are gross overkill for AESOP-Lite! As used for pCT (100 MHz clock and 200 ns peaking time), they consume about 3.3 mW/channel.

But there are important advantages over the Fermi-LAT ASIC set:

- For pCT I did all of the ASIC, board, DAQ, and firmware design myself in the past few years. Therefore I am intimately familiar with all of the technical details and can readily adapt the design to the AESOP-Lite needs.

- The pCT interface is FPGA based, allowing easy reprogramming.
pCT ASIC Design

SPICE Simulations

Preamplifier Output

Shaping Amplifier Output

Discriminator Output

200 ns
pCT Scanner ASIC Layout

- Two gain settings (we will use the high setting)
- Two shaping time settings (200 ns or 400 ns; we will use the latter)
- Two polarity settings (we will use one for + signals, appropriate for Fermi-LAT SSDs)
pCT Tracker SSD Performance

- Except for 2 or 3 channels with bad SSD strips, the hit occupancy from electronics noise is essentially zero.
- The proton hit efficiency is well above 99%, with inefficiency dominated by gaps between sensors. Note that the efficiency does drop slightly at very high rates from pileup, but that is not relevant to AESOP-Lite.

**Efficiency versus V for layer 1**

V layer hit efficiency with gaps regions excluded = 99.9% (99.5% when gap regions are included)

**Efficiency versus T for layer 1**

T layer hit efficiency with gaps regions excluded = 99.9% (99.2% when gap regions are included)
Reducing the System Power

• Lower the system clock frequency from 100 MHz to 10 MHz.
• Lower the quiescent current in the ASIC I/O circuits (there are 4 programmable settings built in, and with the slow clock the lowest setting should be fine).
• Lower the quiescent currents in the entire ASIC analog section. These are set by 3 resistors per chip on the PCB.
  – $\times 0.33$ for the input transistor; $\times 0.6$ for other circuits
  – This results in a slower rise time of the shaping amplifier, by about a factor of two (starting from the 400 ns peaking time setting).
  – I don’t expect a noise penalty. There is a lot of margin, and relative to pCT the integration time will be quadrupled.
• Use a switching regulator to derive the digital power (1.2 V, 1.8V, and 2.5V). A low-dropout linear regulator will still be used for the 3.3V analog power.
Nominal Settings

400 ns
Low-Current Setting

800 ns
Power Estimates

<table>
<thead>
<tr>
<th></th>
<th>Nominal (pCT)</th>
<th>Reduced Power</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analog Power/Chan</td>
<td>1.02 mW</td>
<td>0.44 mW</td>
</tr>
<tr>
<td>Digital Power/Chan</td>
<td>2.30 mW</td>
<td>0.46 mW</td>
</tr>
<tr>
<td>FPGA Power/Chan</td>
<td>1.46 mW</td>
<td>0.29 mW</td>
</tr>
<tr>
<td>Total/Chan</td>
<td>5.06 mW</td>
<td>1.19 mW</td>
</tr>
<tr>
<td># channels = 7 × 12 × 64 = 5376</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total dissipation</td>
<td></td>
<td>6.4 W</td>
</tr>
<tr>
<td>With 80% efficiency</td>
<td></td>
<td>8.0 W</td>
</tr>
</tbody>
</table>

Here the digital ASIC power and FPGA power are as measured from currents in the pCT hardware at 100 MHz, while for the reduced case I conservatively divided by 5, i.e. assuming a 20 MHz clock (and also reduced current settings in the ASIC I/O drivers). Finally, I assumed 80% efficiency for the regulators, switching in the digital case and low-drop-out linear in the analog case. That results in a prediction of 8 Watts of power for the 7-plane AESOP-Lite system.
AESOP-Lite Tracker Board Mechanical Interface

• Working assumption: all 7 boards plus spares are identical. This is a huge labor and cost savings for me!
  – Rotate the non-bending-view boards 90 degrees relative to the others.

• Mount the board to an aluminum frame, as in pCT:
  – Alignment via two pins, one in a hole in the PCB, the other in a slot.
  – Several screws around the edges to hold the board flat on the frame.
  – Keep all the tall parts on the SSD side of the board. A few tiny SMT resistors and capacitors will be on the back side.
  – May need aluminum stiffeners on the SSD side to hold the board flat when it is not mounted in the frame, to avoid stress on the SSDs.
  – Still an open question whether to support the interior SSD edges. I think we can get away without it.

• Wire bonds will not be potted, so the frame needs removable acrylic covers to protect the board when not mounted in the instrument.
pCT “Cassette”

Pin and Hole

Pin and Slot
Aluminum stiffeners along these edges?

SSD butt glue joint and wire bonds

4 pitch Adapter circuits go in here.

SSD active area cut out in the PCB, except for about 1 mm around the edges for support and conductive epoxy.
AESOP-Lite Tracker Interfaces

- The electrical signal interface with the outside world will be only the bottom board.
  - USB cable carrying a bidirectional UART connection (19,200 baud?). Should be plenty fast for the low rates that have been described to me.
  - LVDS twist-pair carrying the asynchronous trigger signal to the Tracker.
- 4-wire power cable with 4V, 100V, and ground goes to each board.
- Daisy-chained CAT-5 cables carrying LVDS over twisted pairs provide 10 Mbit/s communication between boards.

Note: each board also has a 14-pin connector socket for FPGA programming.

Note: the board ordering is for illustration and is probably not up to date.
USB Data Acquisition Interface

• Implemented by a Silicon Labs CP2103 USB to UART bridge chip.
  • Copied from the Xilinx ML605 evaluation board schematic that we use for the pCT event builder.
  • Already have the firmware to use on the FPGA end of the link.
  • Can plug a board into any PC or Linux box (with Silicon Labs drivers) and communicate with it via simple software such as pySerial.
• We will define a set of commands consisting of a sequence of Byte codes to communicate with the FPGA firmware. Board and ASIC addresses will allow the commands to address any part of the system.
• We will define a serial output stream for event data, as well as for monitoring information.

Note: the USB interface is present on all boards but powered down on all but the bottom board.
Internal Tracker Communication

- Each cable has 4 LVDS pairs:
  - From each board to the next higher board:
    - 10 MHz clock
    - Serial commands
    - Trigger (now clock aligned)
  - From each board to the next lower board:
    - Serial data during acquisition; otherwise monitoring information

- The bottom board will generate the 10 MHz clock, by a circuit that is powered down on the other boards.

- I may include an additional twist pair, used to implement a cosmic-ray trigger on the lab bench, for testing.
Tracker Monitoring

- All of the ASIC configuration registers have a non-destructive read mechanism. All FPGA configuration parameters will also be readable by the data acquisition computer. This verification is normally done at the time of initial configuration, to ensure that the correct bits were loaded.
- Each board will include monitoring (when polled by the computer) of the voltage and current of every on-board regulator, the leakage current of the SSDs, and the temperature.
  - This is done over a local I²C bus on each board.
  - The board FPGA will be the I²C bus master and will relay information and commands between the computer and the I²C bus.
  - Note that we don’t want any activity on this monitoring bus while the trigger is active, as it is a single-ended signal that could interfere with the sensitive signal amplifiers.
- My idea is to send the monitoring information over the data link, so that data acquisition must be interrupted when monitoring is done, as is necessary anyway for noise reasons.
Board Design

• I’ve already made a first cut of the schematic, starting from the pCT V board and making a few modifications, the main ones being:
  – 20 MHz clock generation, based on the Silicon Labs Si500D chip. Powered off unless the 4th bit of the address switch is set. The FPGA can be programmed to generate practically any other frequency from this. The Xilinx ML605 board uses this chip at 200 MHz.
  – USB and Trigger interface instead of DVI cable.
    • Silicon Labs CP2103 USB to UART bridge chip, powered off unless plugged in (the chip is powered by the USB cable).
    • Mini-USB plug (MINI-USB-UX60-MB-5ST)
  – CAT-5 8P8C connectors for the daisy chain communication.
  – Analog Devices ADP2370 current buck (switching) regulator for the 1.2V and 2.5V digital power (linear regulators are used for the 1.8V Flash memory power and the I2C chip power, because those draw current only for short time spans).
    • Operate at (up to) 1.2 MHz and are synchronized by a clock provided by the FPGA (they clock themselves until the FPGA is up and running).
  – New resistors for programming the ASIC analog currents.
  – Included over-voltage protection (crowbar) based on pCT experience.
Parts

• The SSDs and ASICs are all in hand (surplus).
  – I’m still waiting on our tech to test the SSDs (simple IV curves; we won’t test individual strips).
  – We will not test the ASICs prior to mounting them. The yield in the pCT project (around 200 chips in the system plus spares plus prototypes) was 100%. And if a couple of chips have to be removed and replaced, that is not difficult.
• We don’t have enough of the pitch adapter circuits to finish the job. Our tech is looking for a new vendor.
• None of the other parts is particularly expensive (the FPGAs cost around $30 each), and all are readily available with a week or two lead time.
• The printed circuit boards will be manufactured and loaded in Silicon Valley (probably by the vendor who did the pCT boards).
• I’m assuming that the aluminum frames will be designed and machined by Delaware, not UCSC. But we can do it if necessary.
Schedule

• Once the schematic is final (it needs some careful checking), the PCB design will take a month or two. The most difficult part of the layout can be largely copied from the pCT V board design.

• The board manufacture and loading takes two or three weeks at most.

• A UCSC technician will mount the SSDs and ASICs and wire bond them. He will also form the two-SSD ladders by edge gluing and wire bonding (just as he did for the pCT boards). This will take a month or two, depending on how busy he is with other projects.

• The biggest job is the firmware coding (Verilog) and testing. Much of it can be copied from the pCT tracker board firmware and other parts from the pCT event builder. I don’t see anything particularly difficult. I did all of the pCT firmware design and coding, for lack of a capable student or postdoc, but I hope to get a graduate student (or undergraduate) to help with at least the testing.

• I expect that we can have all of the boards built and working on the bench by a year from now.
Post-Meeting Action Items

• Include an option to output a Tracker trigger-OR primitive, for example to use in test-bench cosmic-ray tests
• Include trigger-OR time-over-threshold in the event readout
• Include mechanical support under the gap between SSD ladders
• UCSC will define a logical interface (commands and data format) between the Tracker and the micro-controller and then will program a Xilinx evaluation board for UD to use to test their system. The evaluation board uses the same USB/UART interface as proposed for the Tracker.