LAPPD Digital Model

Location and components

https://github.com/mwetstein/LAPPDresponse

- **LAPPDpulse**
  - a signal on a stripline – stores the time, stripnumber transit times to get to the two ends, peak value of the signal
  - if charge from a single photon spreads over three strips, then there will be 3 “pulses”
- **LAPPDpulseCluter**
  - Stores a collection of pulses
  - Keeps track of how many pulses per strip and indexes where each of the pulses for a given strip are located
- **LAPPDresponse**
  - Takes a photon hit, breaks it into “pulses”, modeling the charge sharing and transit times to the two ends of the stripline. Stores these pulses for as many photons as given.
  - On request, produces a digital trace for each channel, summing over all pulse in a given time window and at a given sampling rate.

TestLAPPDResponse.C: A simple root script for testing the above classes

pulsecharacteristics.root: A root file storing a template pulse shape, the pulse height distribution, and a table for the charge spread across multiple strips.

Basic idea

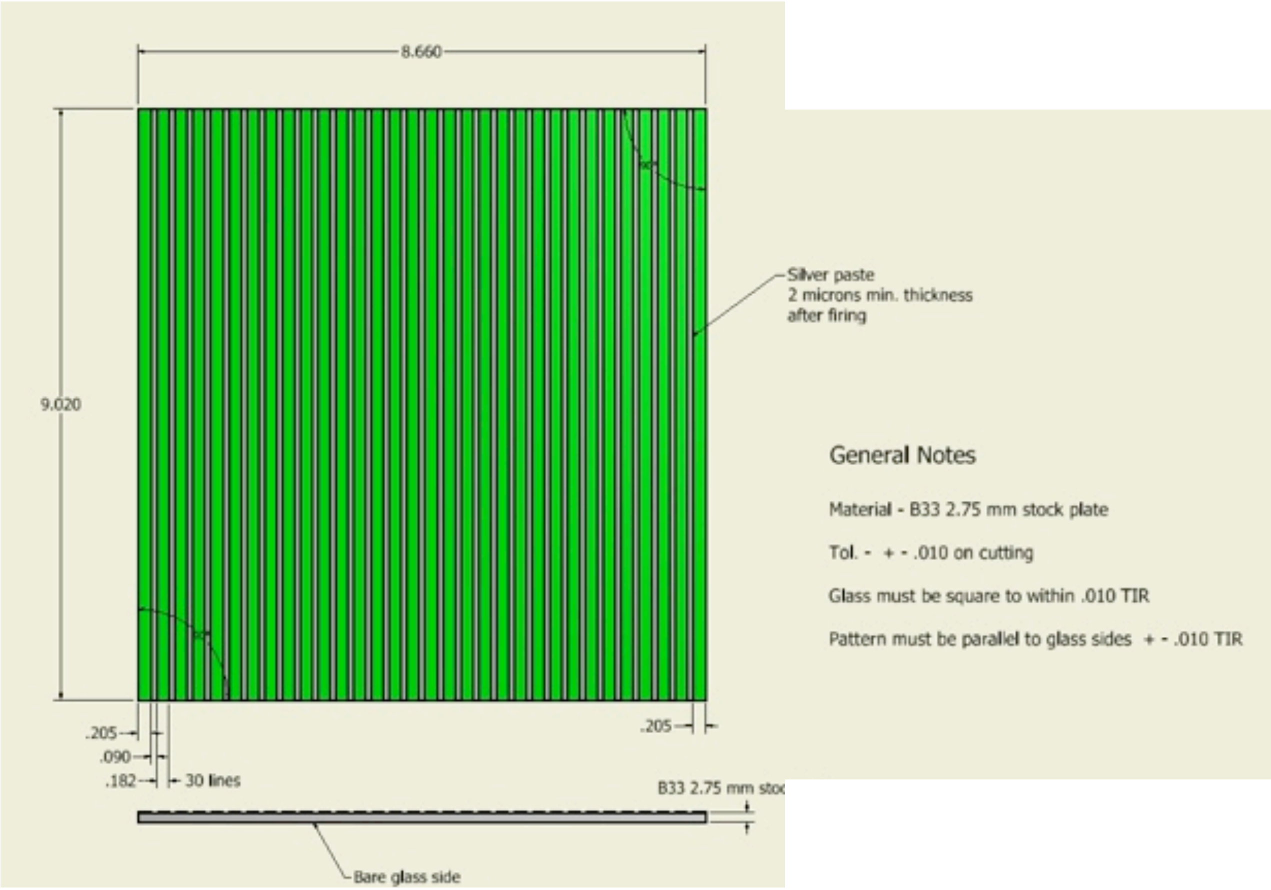
Storing pulse information

- For a given photon hit, chose a random peak value based on pulse height distribution
- Determine the nearest strip
- Based on how close the hit was to that strip, determine the spread of the charge in the transverse direction
- Create a “pulse” for every strip with peak signal above 0.5 mV; determine the time of the pulse, peak value on that strip, transit time to get to each end of the strip; write pulse to LAPPDpulseCluster class
- Repeat process for all photon hits

Generating “traces” (basic digital model)

- provide a buffer size, start time for the acquisition, sample size and noise
- code will loop over all pulses for a given channel. Pulses that fall within timing window are summed interpolating from template pulse shape, each offset by its respective arrival time
- add in white noise

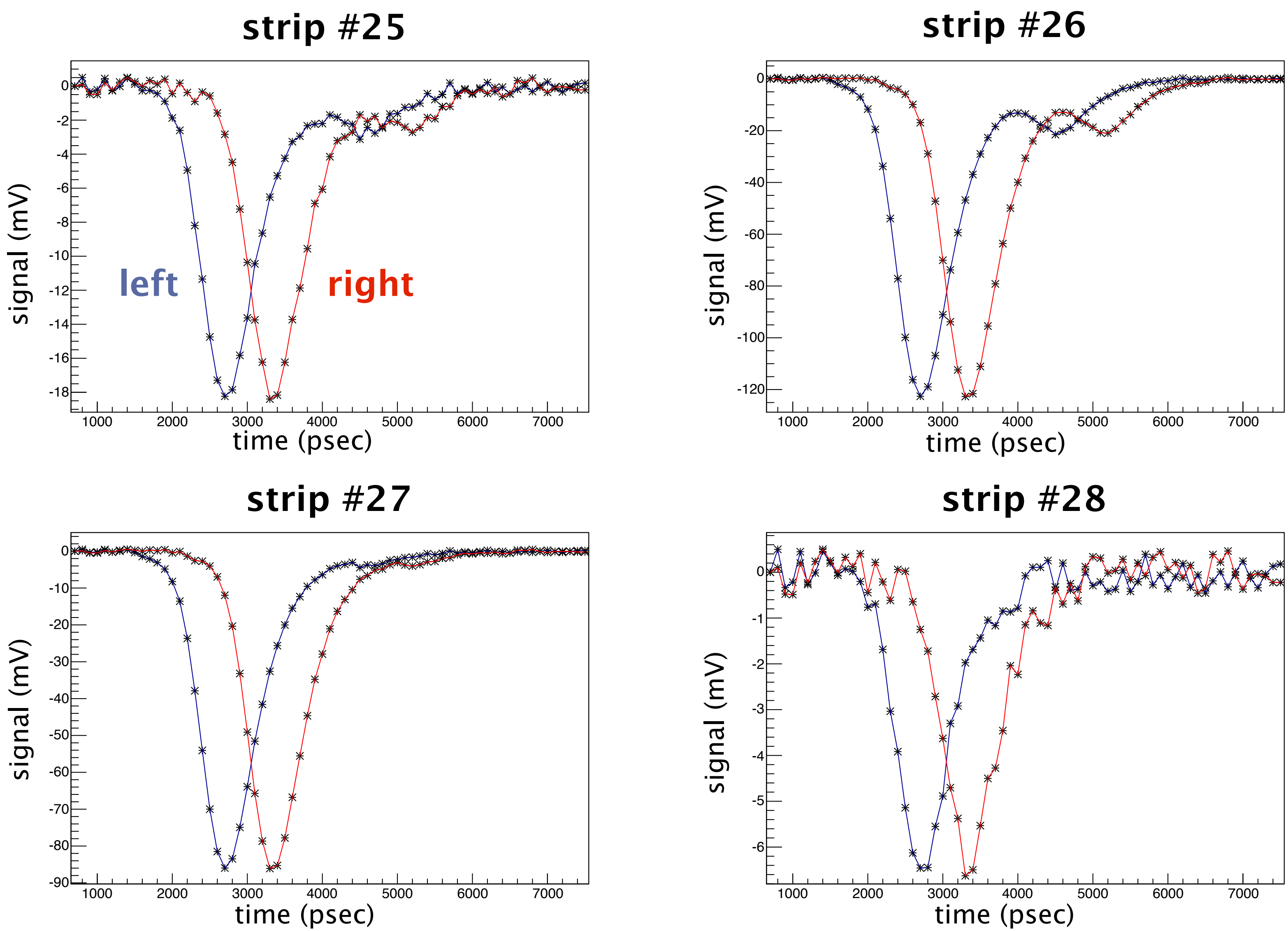
The anode



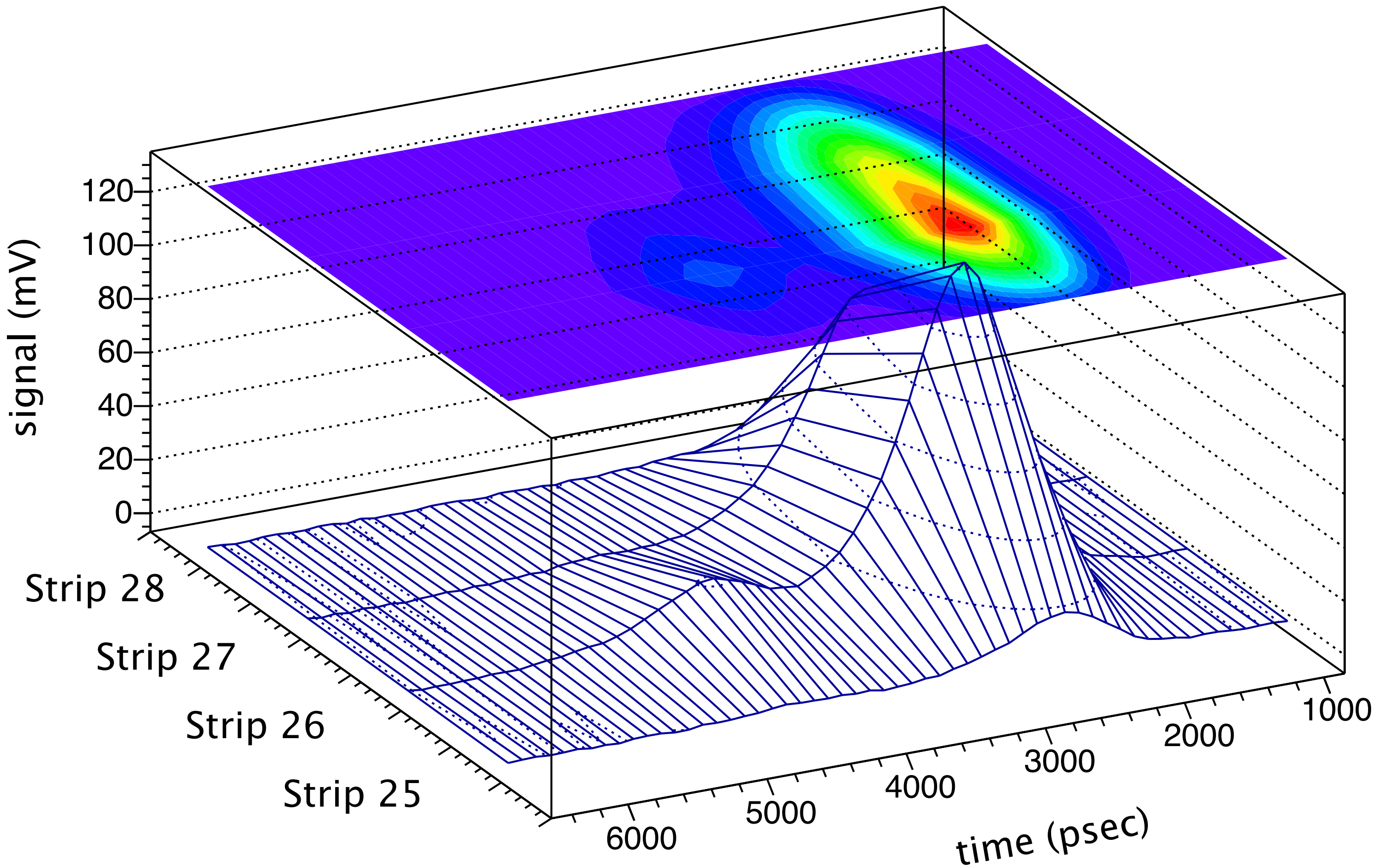
- Longer in the direction of the striplines ("parallel" direction)
- Active area is slightly less than 8"x8" (I need to code that)
- Grid spacers reduce the active area (also needs to be coded)
- First striplines in the transverse direction are wider than the rest
- Measured speed of signal propagation is 0.57c

1D response

Test case: 2 pulses in different places, at different times

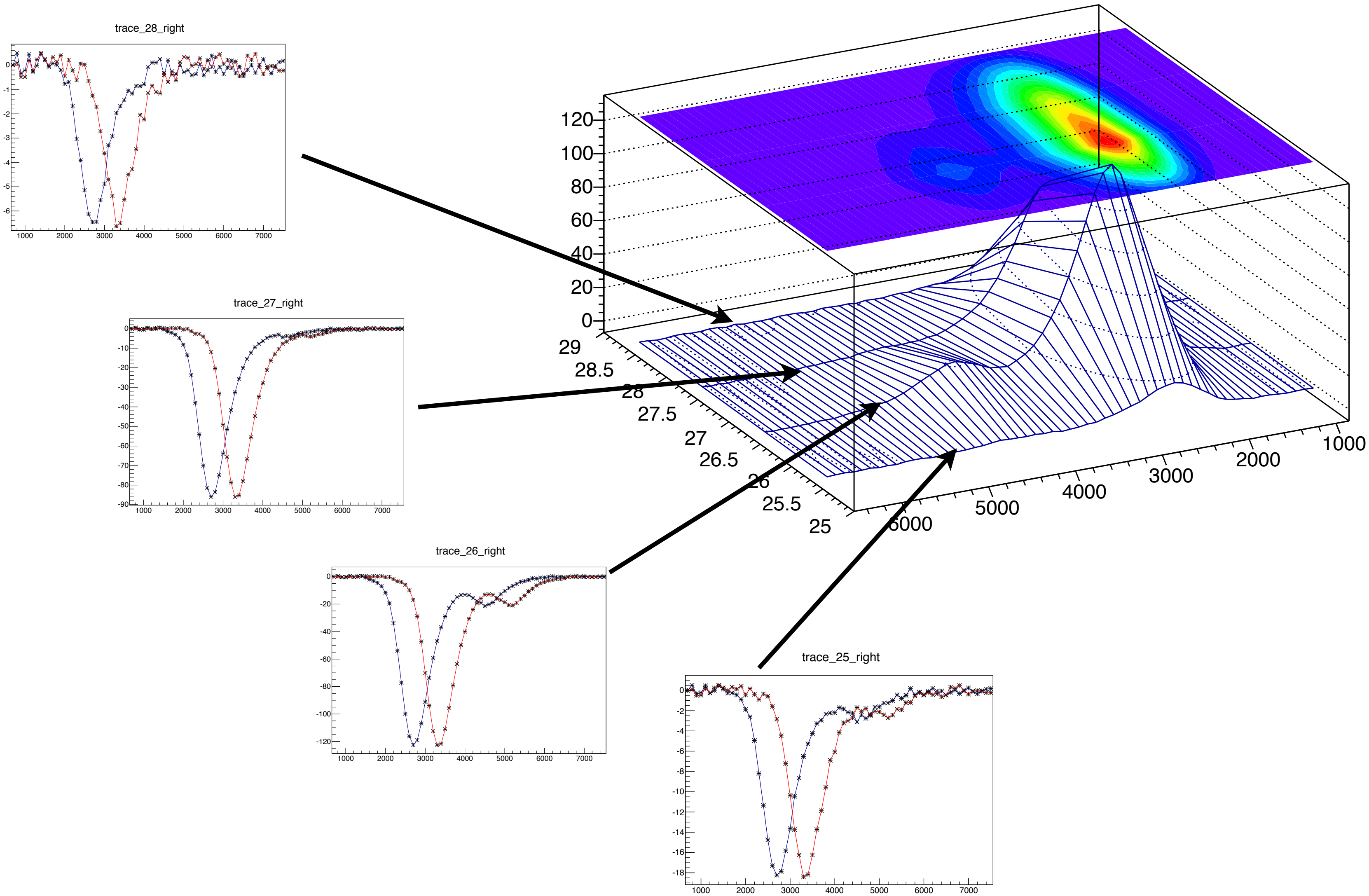


2D response



signal (mV)
120
100
80
60
40
20
0
Strip 28
Strip 27
Strip 26
Strip 25
6000
5000
4000
3000
2000
1000
time (psec)

3D response



Next step

Storage and processing: From a simulations standpoint, LAPPDs multiply the number of channels by 30. But, I have designed the code so it should be possible to process and store only the channels with signal. We need to hash this out.

Digitization, triggering, and dead time: The PSEC4 chip has a shallow buffer (~30 nsec) with significant deadtime (~10s microseconds) to read the traces. One can read each chip (6 channels) separately, leaving the other chips active. One can also place a hold on a trace until a more sophisticated trigger decision is made to read the SCA. At a low enough rate, these games can keep the PSEC electronics streaming, but they need to be simulated. PSEC5 will address these challenges.

The data structures in LAPPDresponse are independent of the chip architecture, so it should be possible to build different models for different electronics architectures.