

Hodoscope Trigger And Simulation

HPS Collaboration Meeting

May 2019

Introduction

- Many updates were performed on the readout system to enable to support for the hodoscope subsystem.
 - This includes new, hodoscope-specific drivers, and also updates to old calorimeter drivers.
- The new code is installed, merged, and running.
 - Trigger studies Monte Carlo is undergoing processing at this time.
- A multi-level hodoscope trigger has been designed.
 - Preliminary trigger parameters have been selected. Final parameters are pending completion of studies using new A' signal Monte Carlo.

Readout Step I – SLiC

- SLiC is not set up to handle distributing energy across several channels on the same active component.
 - This means that hits will be reported once for a given scintillator, regardless of how many physical FADC channels are actually present.
 - As such, SLiC hits have to have their energies manually distributed to the actual FADC channels.
- To better enable this, SLiC hits are stored in `SimTrackerHit` objects, which store an exact interaction position.
 - This allows the hit's position on the scintillator to be known to a high degree of accuracy, so that its energy can better be distributed.

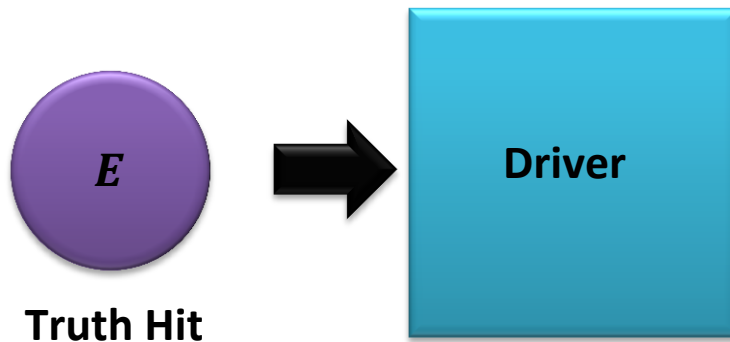
Readout Step II – Preprocessing Driver

- Preprocessing is necessary before the readout chain can properly handle the SLiC hits.
 - SLiC hits do not present their energy on each physical channel of the hodoscope – instead they are only by scintillator.
 - This energy needs to be distributed across the physical channels for scintillators with more than one channel.
 - SLiC hits are `SimTrackerHit` objects – the readout chain expects `CalorimeterHit` objects.

- To handle these tasks, a preprocessing driver is implemented.

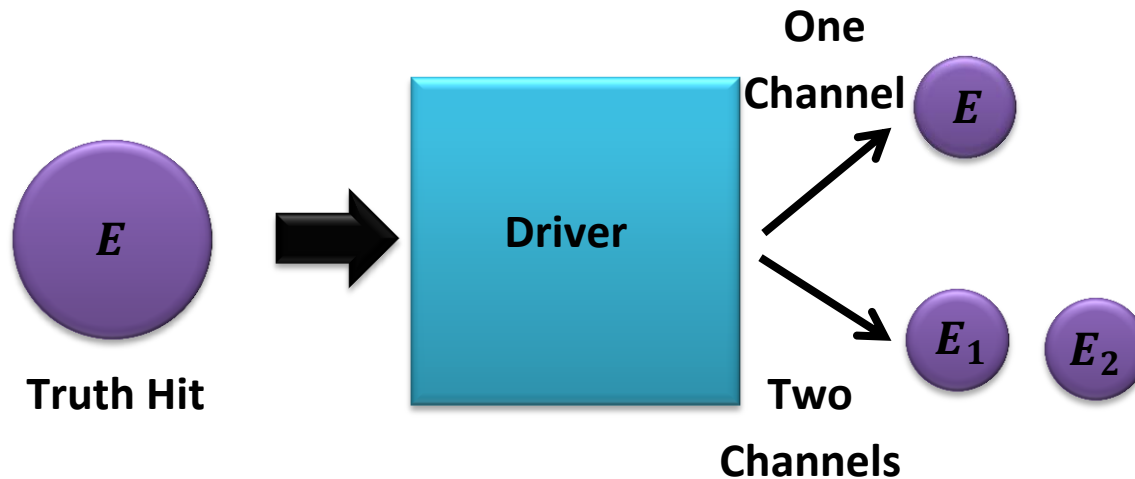
Readout Step II – Preprocessing Driver

- The preprocessing driver first pulls in all truth hits from a beam bunch.
 - Hit energies are then processed. Hits on a one-channel scintillator are unchanged. Hits on a two-channel scintillator have their energies split.



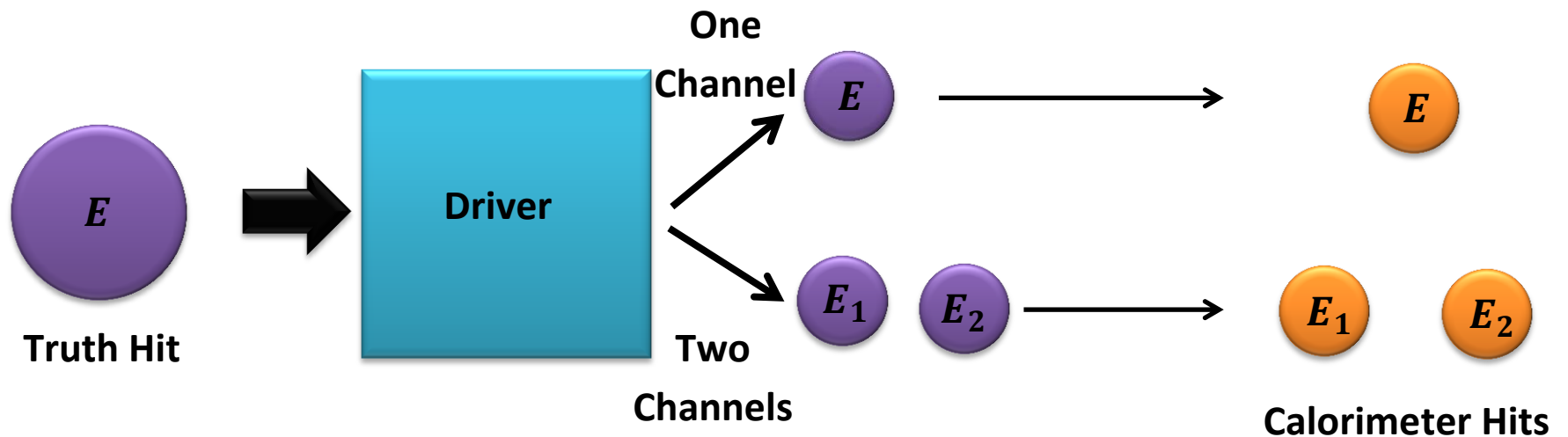
Readout Step II – Preprocessing Driver

- The preprocessing driver first pulls in all truth hits from a beam bunch.
 - Hit energies are then processed. Hits on a one-channel scintillator are unchanged. Hits on a two-channel scintillator have their energies split.



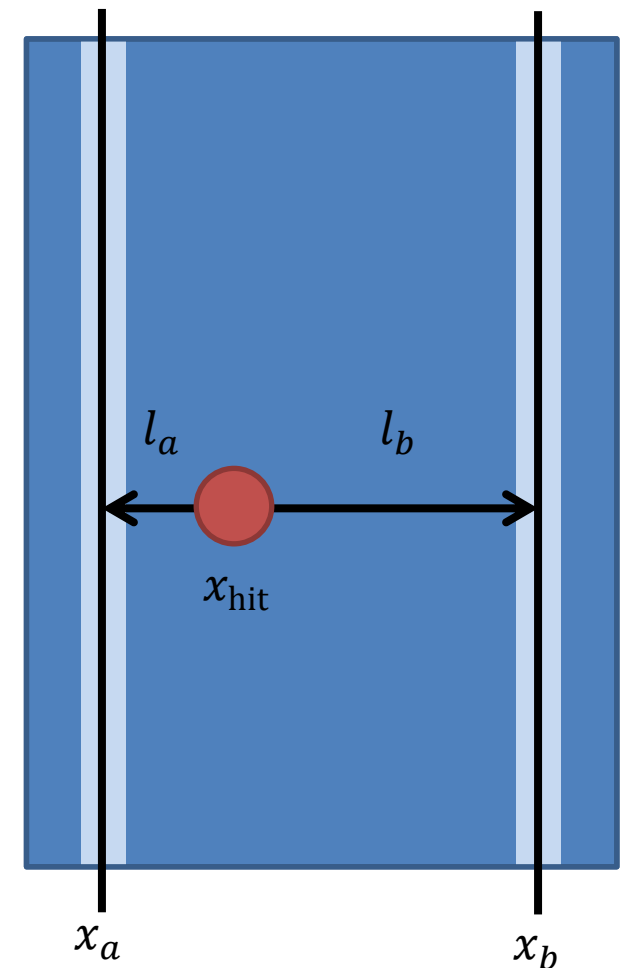
Readout Step II – Preprocessing Driver

- The preprocessing driver first pulls in all truth hits from a beam bunch.
 - Hit energies are then processed. Hits on a one-channel scintillator are unchanged. Hits on a two-channel scintillator have their energies split.
 - The resultant objects are then replaced by `SimCalorimeterHit` objects, which retain the original's time and truth information, but use the new FADC channel and energy.



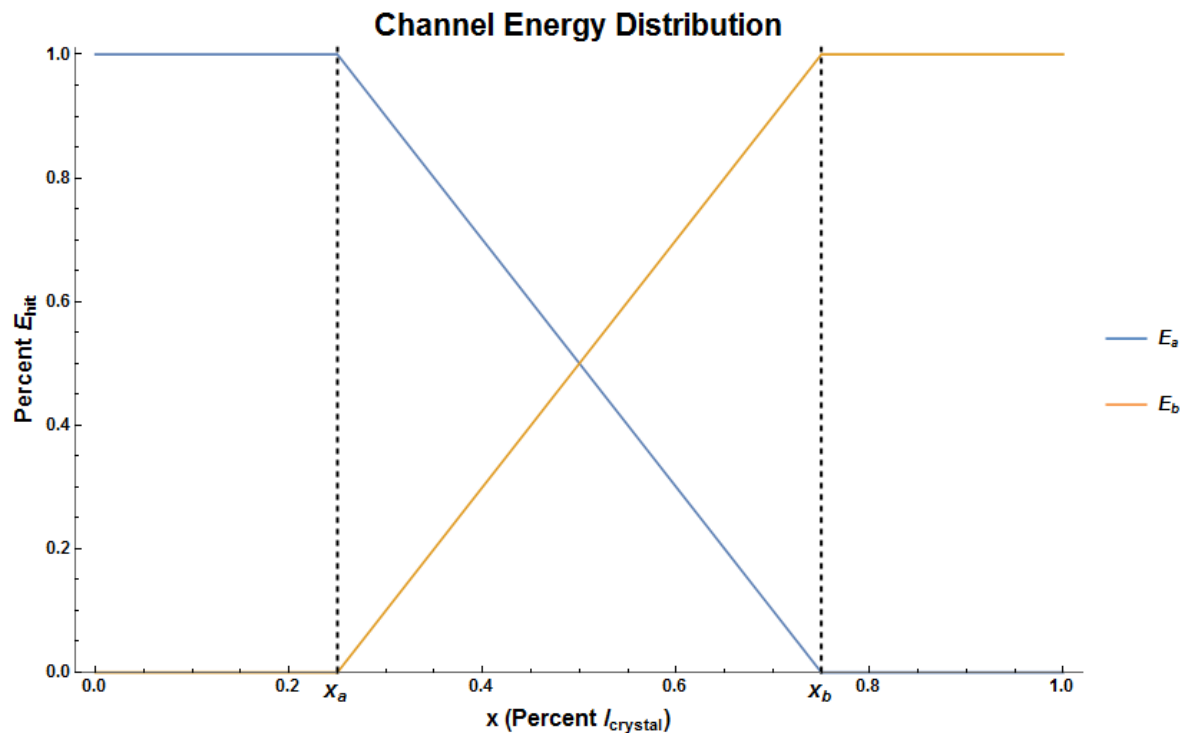
Readout Step II – Preprocessing Driver

- The energy splitting is performed by a simple linear interpolation based on the distance of the hit from each of the optical fiber holes.
- Let:
 - $l_a = |x_a - x_{\text{hit}}|$ be the distance between the first hole and the hit
 - $l_b = |x_b - x_{\text{hit}}|$ be the distance between the second hole and the hit
 - $E_a = E_{\text{hit}} \frac{l_b}{l_a + l_b}$ be the energy of the first optical fiber channel
 - $E_b = E_{\text{hit}} \frac{l_a}{l_a + l_b}$ be the energy of the second optical fiber channel



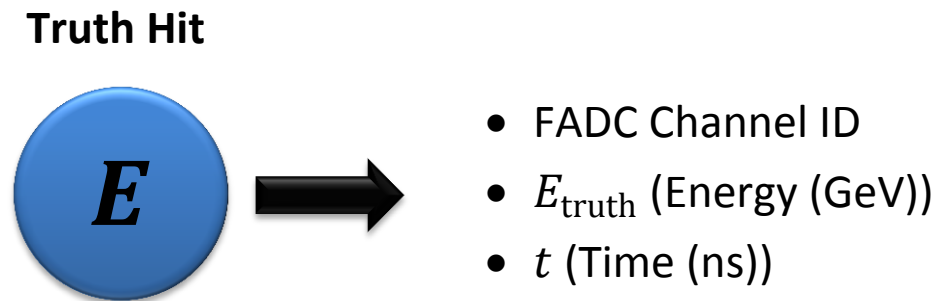
Readout Step II – Preprocessing Driver

- Additionally, any hit that falls between the edge of the scintillator and an optical fiber hole will contribute all of its energy to the closest optical fiber.
- This gives the following energy distribution:



Readout Step III – Digitization Driver

- The digitization driver is responsible for taking truth hits and converting them into ADC pulses which can be integrated in the same manner as actual hits on the hardware.
- This process begins by taking in a SLiC truth hit.

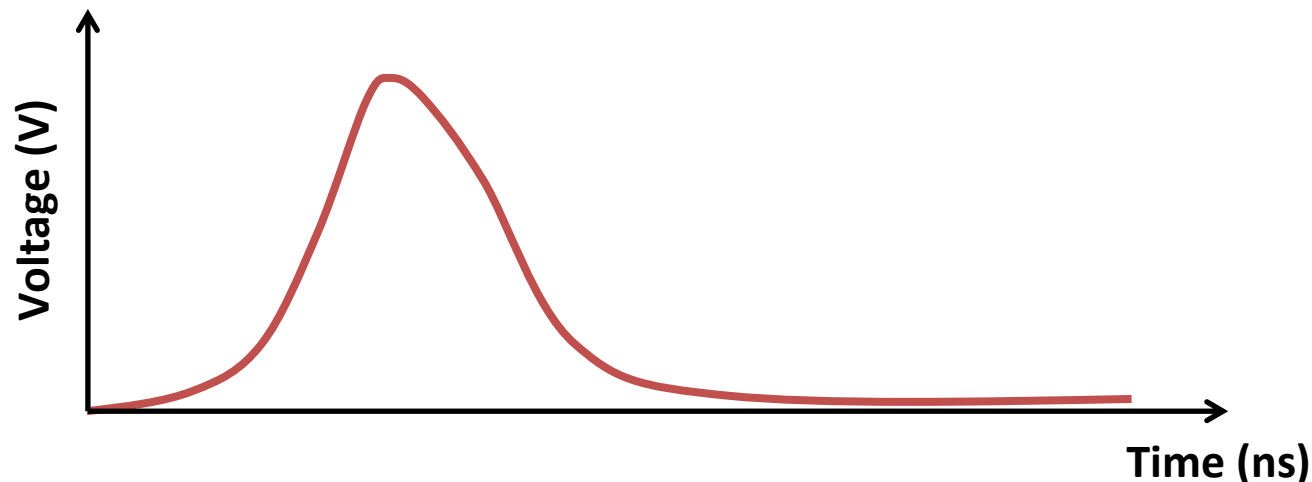


Readout Step III – Digitization Driver

- There exists both a voltage buffer and an ADC buffer for each FADC channel. The buffer matching the hit's channel is selected.
- The hit is converted to a voltage pulse using the “three-pole function”:

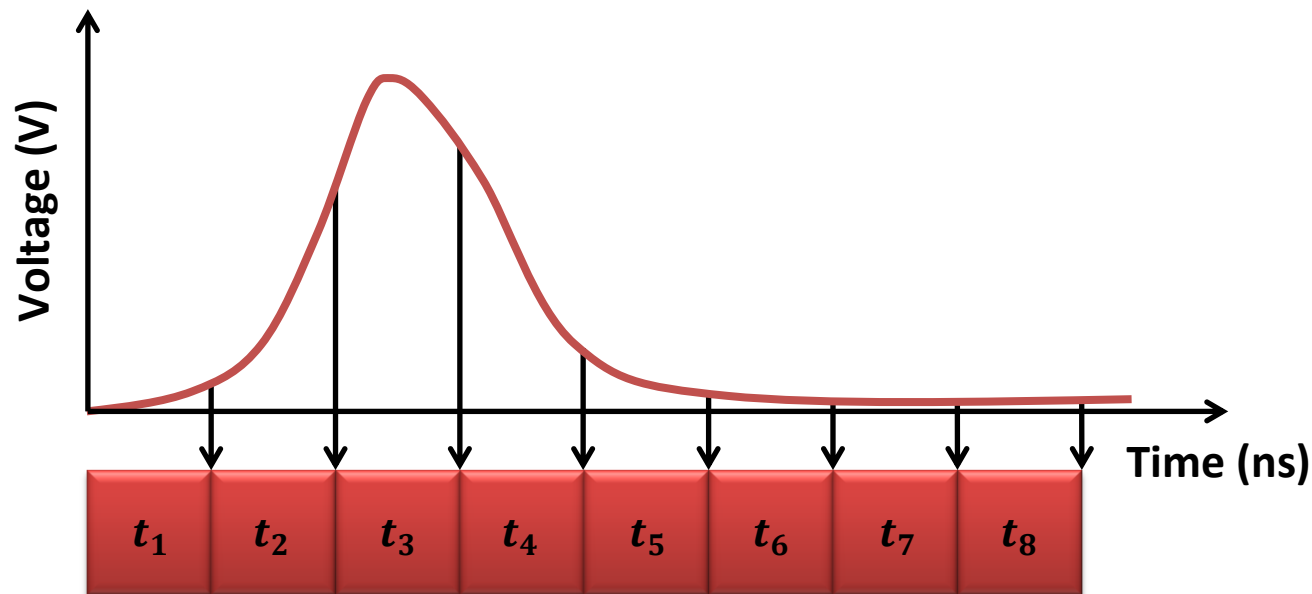
$$V(t) = \frac{t^2}{t_{sh}^3} e^{-t/t_{sh}}$$

- Here, t is the time in ns and t_{sh} is the shaping time. For the hodoscope, this is set to $t_{sh} = 4$ ns.



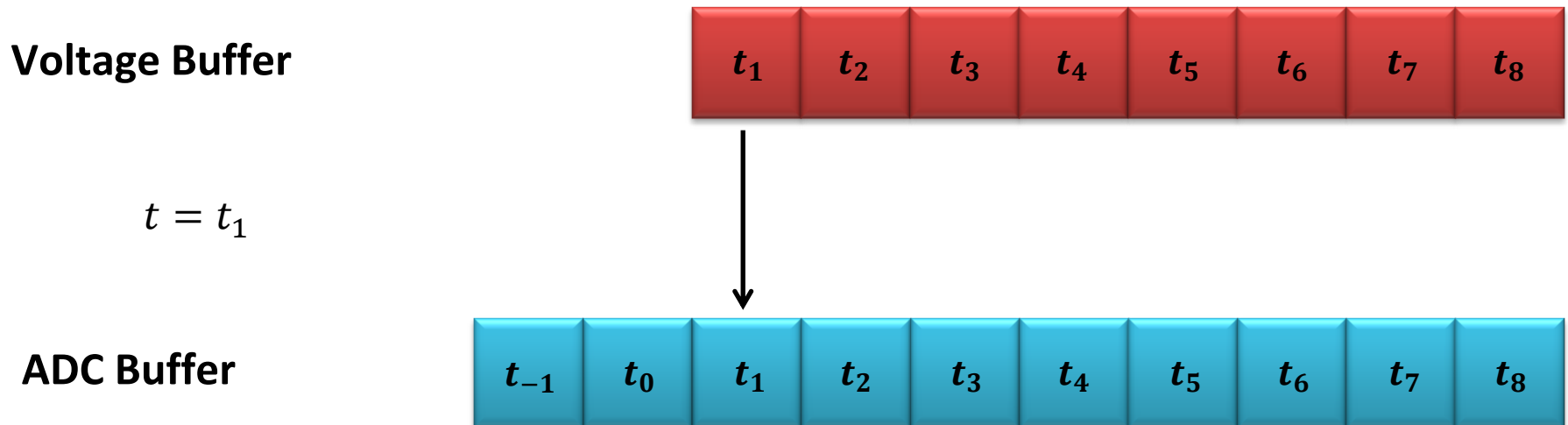
Readout Step III – Digitization Driver

- The voltage pulse is then sampled every clock-cycle (4 ns) and the value at that point added to the voltage buffer.
 - The voltage buffer is cumulative – if multiple hits occur on the same channel in close enough proximity, they will add to one another.
- The voltage pulse for a hit is propagated across the entire voltage buffer immediately upon reading the hit.



Readout Step III – Digitization Driver

- Next, voltage samples are converted to ADC. Unlike the voltage pulse propagation, this only occurs every two beam bunches (4 ns).
 - At this time, the “current” voltage sample for each channel is sampled and converted to ADC.
 - The pedestal is added at this stage.

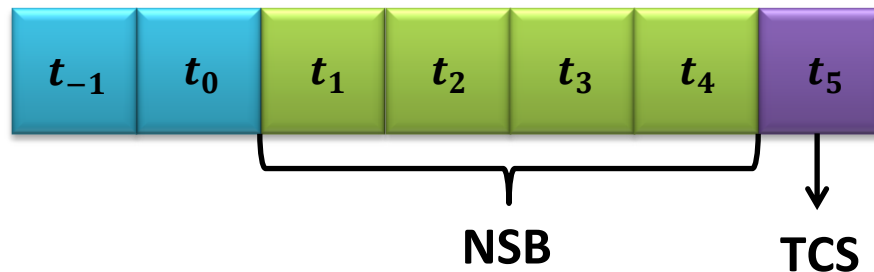


Readout Step III – Digitization Driver

- Once all of the voltage samples for the present time are converted to ADC, each channel is tested.
- If the ADC buffer value exceeds the integration threshold, integration is initiated for that channel.
 - A number of samples before the threshold-crossing sample (TCS) equal to NSB (number of samples before) are then summed together.
 - The TCS itself is also added to the sum.
- Each event thereafter, when a new cell of the ADC buffer is written, that cell is added to the total integrated ADC sum for the channel.
 - This continues until a total number of samples equal to NSA (number of samples after) have been added.
- The total integrated value is then output as a RawCalorimeterHit to be used by the trigger simulation.
- Only one integration may occur at a time, per channel.

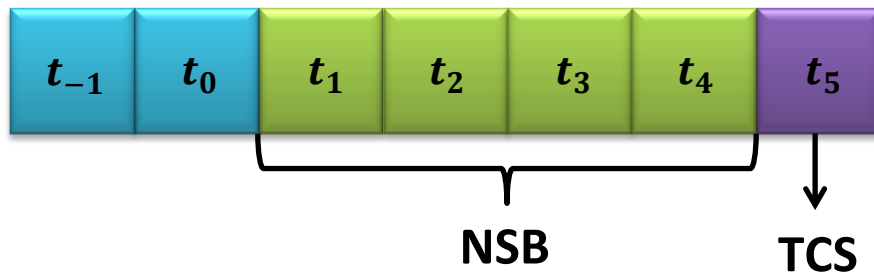
Readout Step III – Digitization Driver

- If the TCS occurs at $t = t_5$, and NSB is 4, the driver immediately sums the first set of ADC sample that already in the buffer.

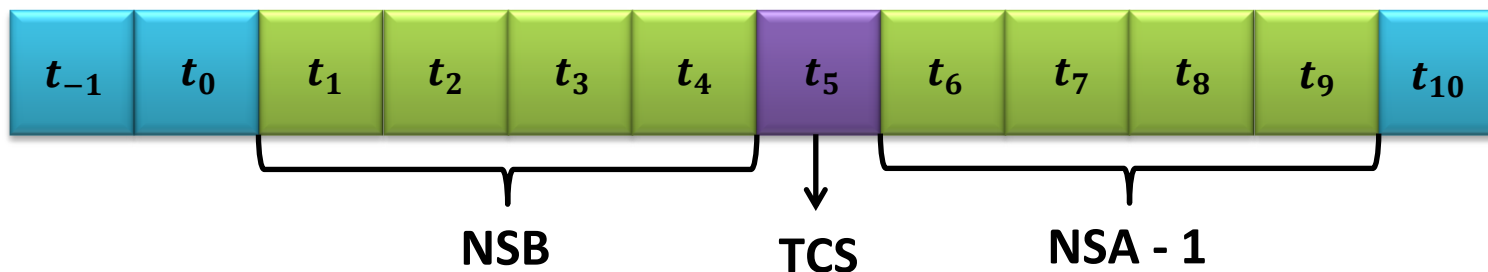


Readout Step III – Digitization Driver

- If the TCS occurs at $t = t_5$, and NSB is 4, the driver immediately sums the first set of ADC sample that already in the buffer.



- If NSA is then 5, the simulation will have wait until the enough beam bunches have been processed to add the remaining samples to the ADC buffer. These will then also be added to produce the full hit.



Readout Step III – Digitization Driver

- At the end of each 4 ns clock-cycle (i.e. two beam bunches processed), the driver will output Mode-3 hits for all completed channel integrations.
- This hits contains:
 - Total integrated ADC, including pedestal.
 - A timestamp equal to the time of the threshold-crossing sample.
 - A channel.
- Pedestal-subtraction, gains, and energy-conversion are handled separately.

Readout Step IV – Raw Converter

- The raw converter is responsible for taking an integrated ADC `RawCalorimeterHit` from the digitization driver and converting it to an energy-valued `CalorimeterHit` that can be used by the trigger.
 - The raw converter takes in each ADC hit and performs pedestal-subtraction, applies gains, and applies time shifts.
 - The resultant energy-valued hits are then passed to the trigger.

Readout Step V – The Trigger Driver

- The trigger begins by taking in all hodoscope hits and GTP clusters in a programmable time window (4 – 32 ns).
- The hodoscope only cares whether or not a hit occurs – not its energy. As such, the highest-energy hit on each scintillator is selected.
- Scintillators with hits above a programmable threshold are considered to register a hit. Those with either no hit at all, or with energy depositions below the threshold, are ignored.
 - This eliminates low-energy noise.

Top	Layer 1	—	0.58 MeV	0.43 MeV	—	—
	Layer 2	—	—	1.02 MeV	0.05 MeV	—
Bottom	Layer 1	—	—	—	0.33 MeV	—
	Layer 2	—	—	0.02 MeV	0.26 MeV	—

Readout Step V – The Trigger Driver

- Next, a series of tables are populated showing the total values for different configurations of hits.
 - Secondly, more stringent cut is applied to select hit configurations that are considered sufficient for triggering.

	Top		Bottom	
	Layer 1	Layer 2	Layer 1	Layer 2
ix = 1	—	—	—	—
(ix = 1) + (ix = 2)	0.58 MeV	—	—	—
ix = 2	0.58 MeV	—	—	—
(ix = 2) + (ix = 3)	1.01 MeV	1.02 MeV	—	—
ix = 3	0.43 MeV	1.02 MeV	—	—
(ix = 3) + (ix = 4)	0.43 MeV	1.02 MeV	0.33 MeV	0.26 MeV
ix = 4	—	—	0.33 MeV	0.26 MeV
(ix = 4) + (ix = 5)	—	—	0.33 MeV	0.26 MeV
ix = 5	—	—	—	—

Readout Step V – The Trigger Driver

- The hodoscope has three trigger levels.
- **Level I:** There exists a hit in one or both layers of the hodoscope and a cluster at $x \geq 100$ mm in the calorimeter.
 - This requires that some hit configuration on the previous table passes the trigger threshold.
 - A cluster is also required to exist with an energy exceeding some threshold and a position greater than the specified x . If the hit is a top hit, the cluster must be a top cluster and vice versa for a bottom hit.

Readout Step V – The Trigger Driver

- The hodoscope has three trigger levels.
- **Level II:** There exists a hit in layer 1 (L1) and Layer 2 (L2) of the hodoscope, both hits are either on the top or both on the bottom of the hodoscope, and they meet the correlation conditions. Additionally, there exists a cluster at $x \geq 100$ mm in the calorimeter.
 - This requires that a hit on layer 1 and layer 2 pass the trigger threshold on the table above. They must be either both top or both bottom hits.
 - Those hits must pass the hodoscope layer correlation table.
 - A cluster is also required to exist with an energy exceeding some threshold and a position greater than the specified x . If the hit is a top hit, the cluster must be a top cluster and vice versa for a bottom hit.

Readout Step V – The Trigger Driver

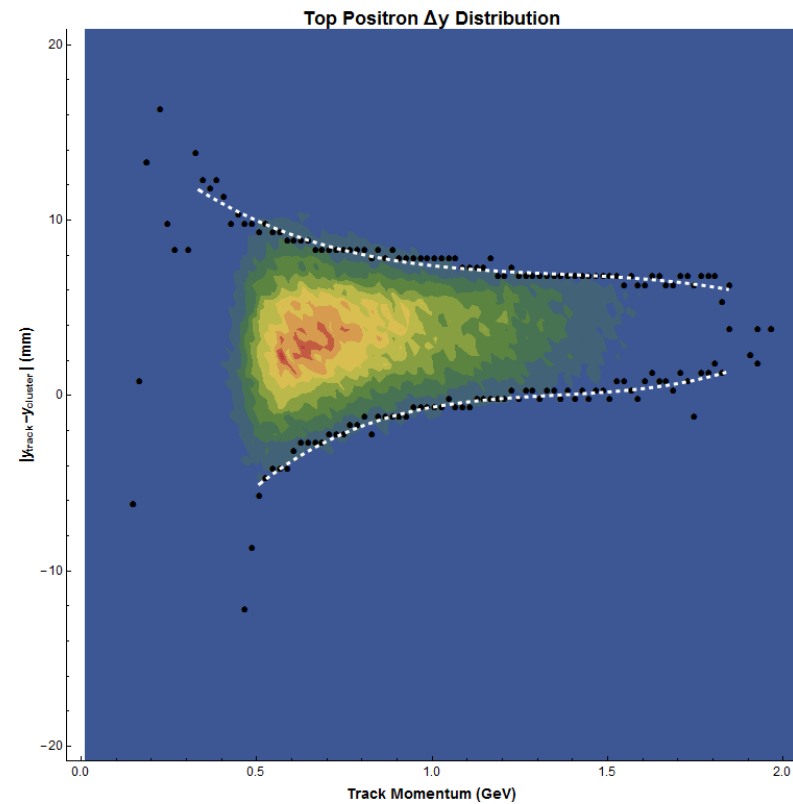
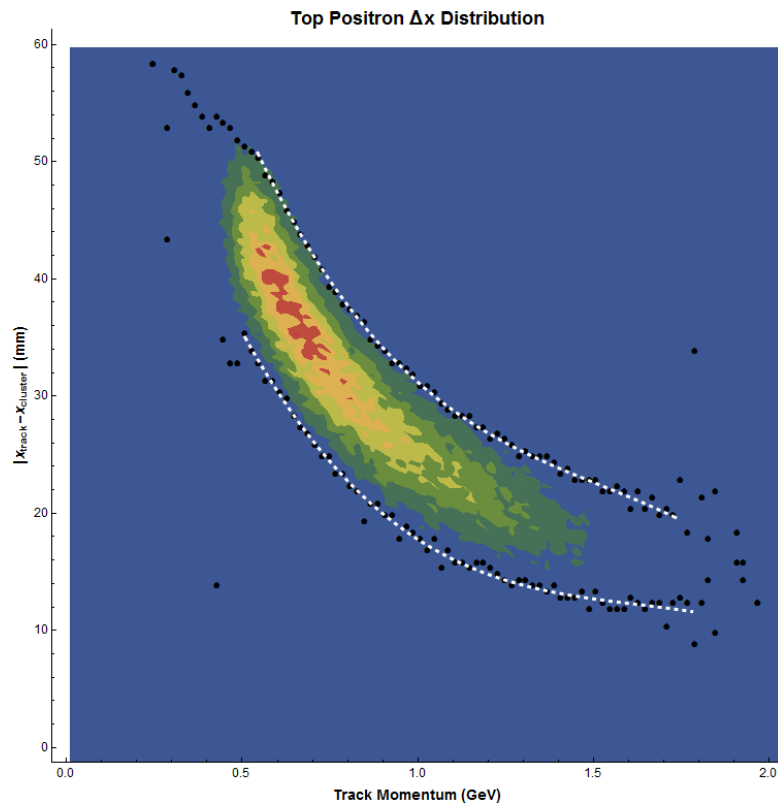
- The hodoscope has three trigger levels.
- **Level III:** As Level II, but the cluster must additionally be correlated with each of the hodoscope hits.
 - This requires that a hit on layer 1 and layer 2 pass the trigger threshold on the table above. They must be either both top or both bottom hits.
 - Those hits must pass the hodoscope layer correlation table.
 - A cluster is also required to exist with an energy exceeding some threshold and a position greater than the specified x . If the hit is a top hit, the cluster must be a top cluster and vice versa for a bottom hit.
 - Each of the correlated hits must individually meet the conditions on the hodoscope-calorimeter correlation table.

Readout Step V – The Trigger Driver

- The hodoscope has three trigger levels.
- **Level III:** As Level II, but the cluster must additionally be correlated with each of the hodoscope hits.
 - This requires that a hit on layer 1 and layer 2 pass the trigger threshold on the table above. They must be either both top or both bottom hits.
 - Those hits must pass the hodoscope layer correlation table.
 - A cluster is also required to exist with an energy exceeding some threshold and a position greater than the specified x . If the hit is a top hit, the cluster must be a top cluster and vice versa for a bottom hit.
 - Each of the correlated hits must individually meet the conditions on the hodoscope-calorimeter correlation table.
- Additionally, the calorimeter-only positron trigger (COP Trigger) may be activated. This excludes any clusters that do not meet its conditions from triggering.

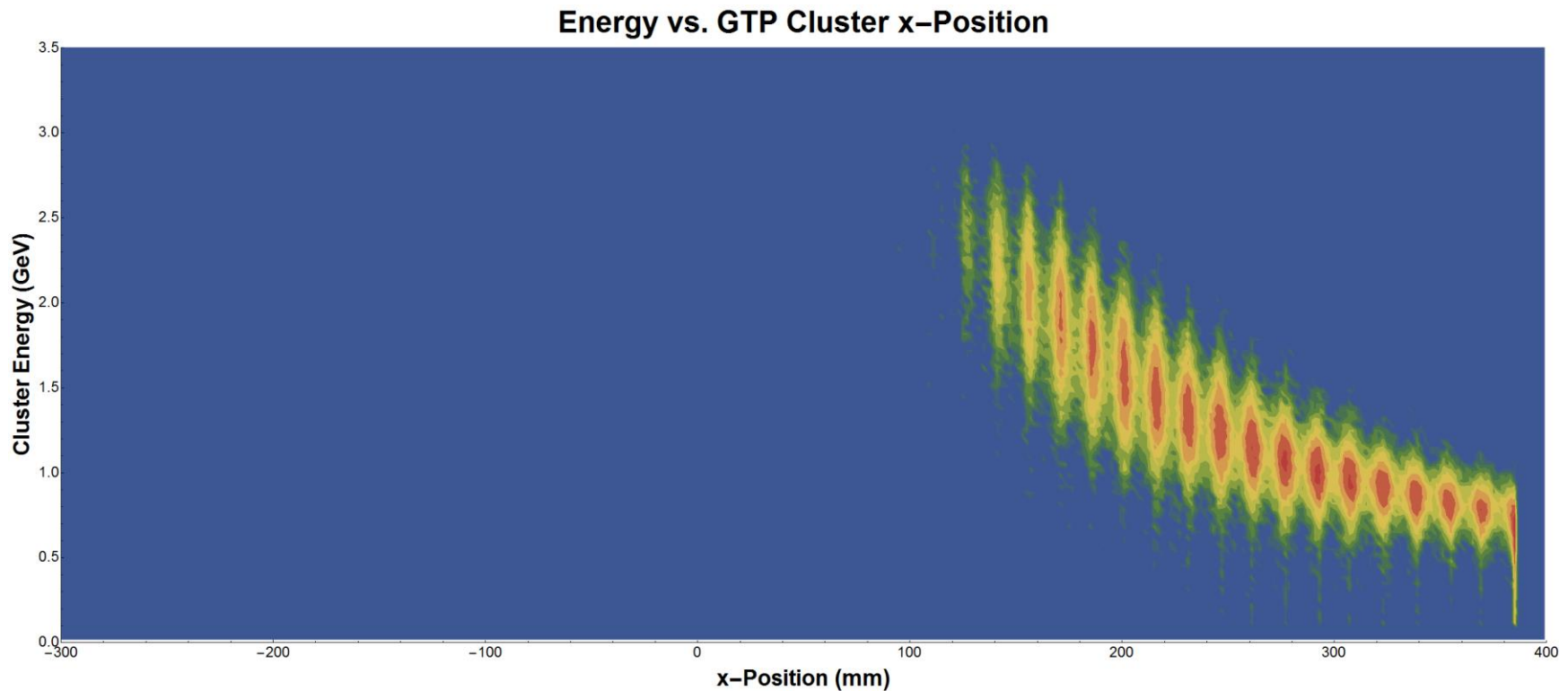
Trigger Tuning Overview

- The COP-Trigger is tuned over several steps:
 - An algorithm is established for cluster-track matching between tracks and trigger-level GTP clusters.



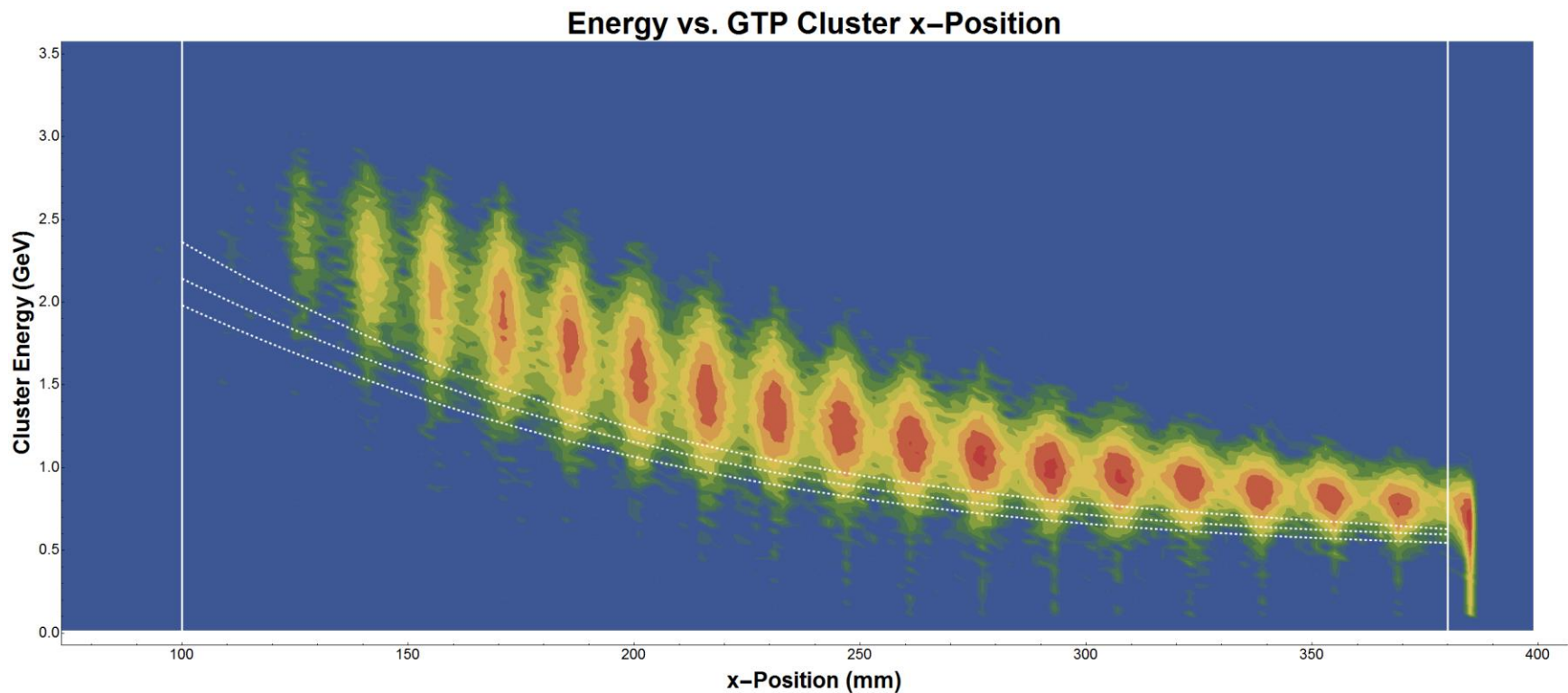
Trigger Tuning Overview

- The COP-Trigger is tuned over several steps:
 - This is used to match all clusters with tracks and select only positrons.



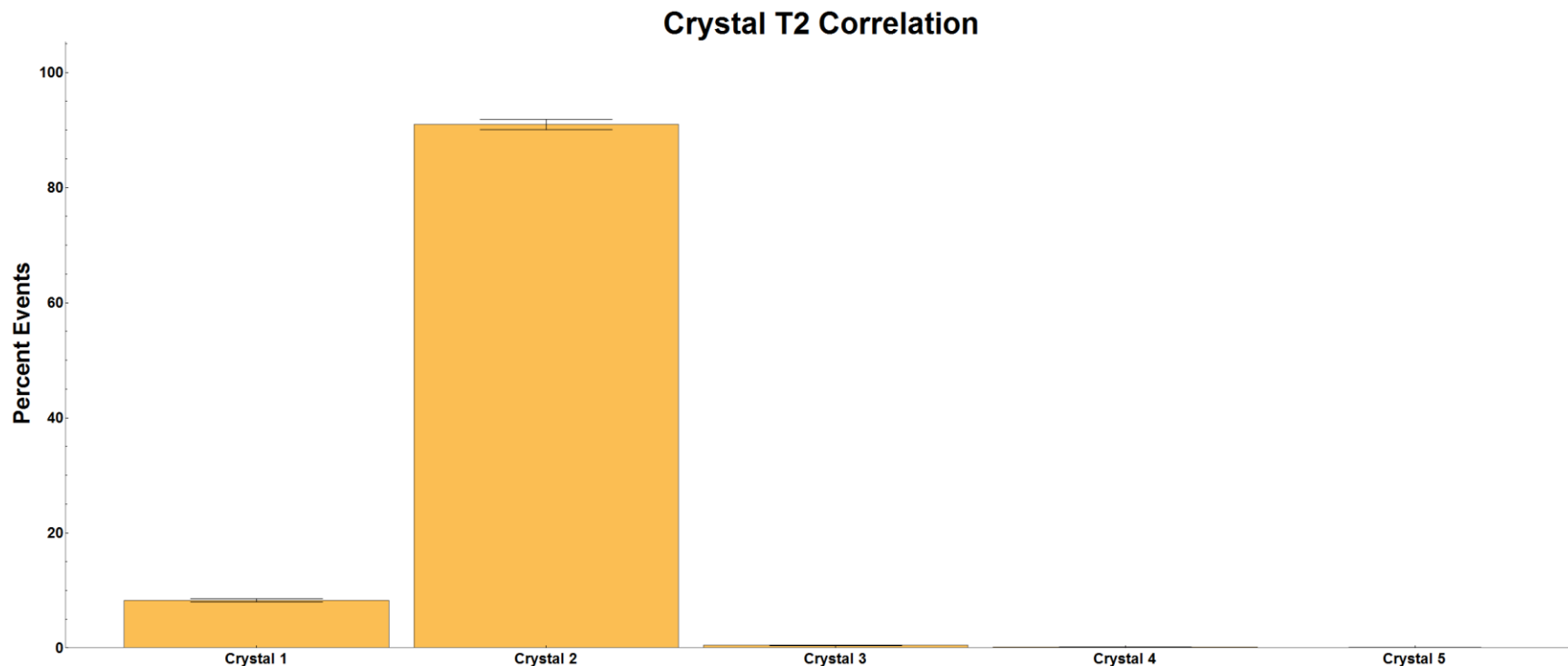
Trigger Tuning Overview

- The COP-Trigger is tuned over several steps:
 - Bounds are then defined to select 91%, 95%, and 97% of all positron clusters.



Trigger Tuning Overview

- The hodoscope trigger itself is then tuned.
 - First, a plot correlating all layer 1 hits with the “best” layer 2 is created for each layer 1 index.
 - This shows which L2 hits correspond with a given L1 hit.



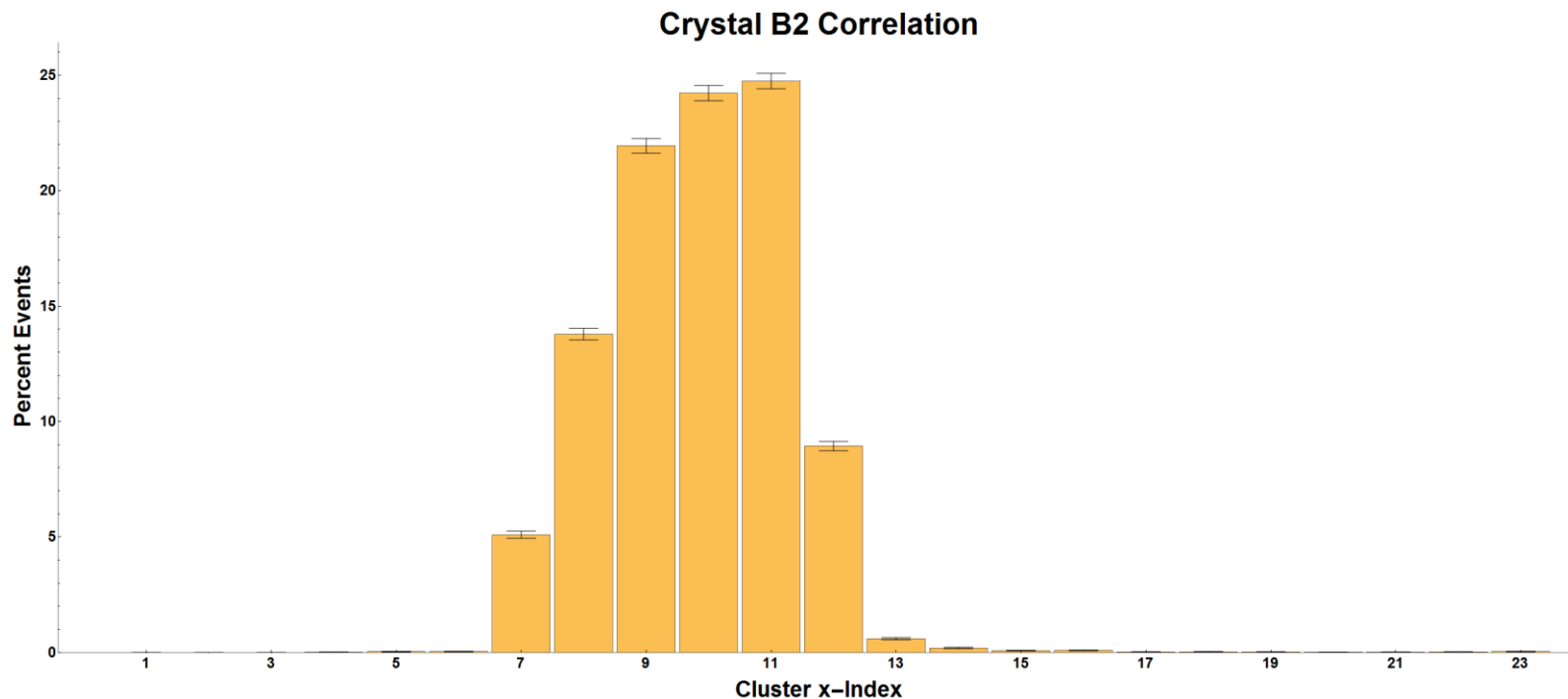
Trigger Tuning Overview

- The hodoscope trigger itself is then tuned.
 - First, a plot correlating all layer 1 hits with the “best” layer 2 is created for each layer 1 index.
 - This shows which layer 2 hits correspond with a given layer 1 hit.
- From this, a correlation table is created that defines which layer 2 hits are valid matches for a layer 1 hit. This is used for the L1 hodoscope trigger.

	L1-1	L1-2	L1-3	L1-4	L1-5
Top	L2-1	L2-[1-2]	L2-[2-3]	L2-[3-4]	L2-[4-5]
Bottom	L2-1	L2-[1-2]	L2-[2-3]	L2-[3-4]	L2-[4-5]

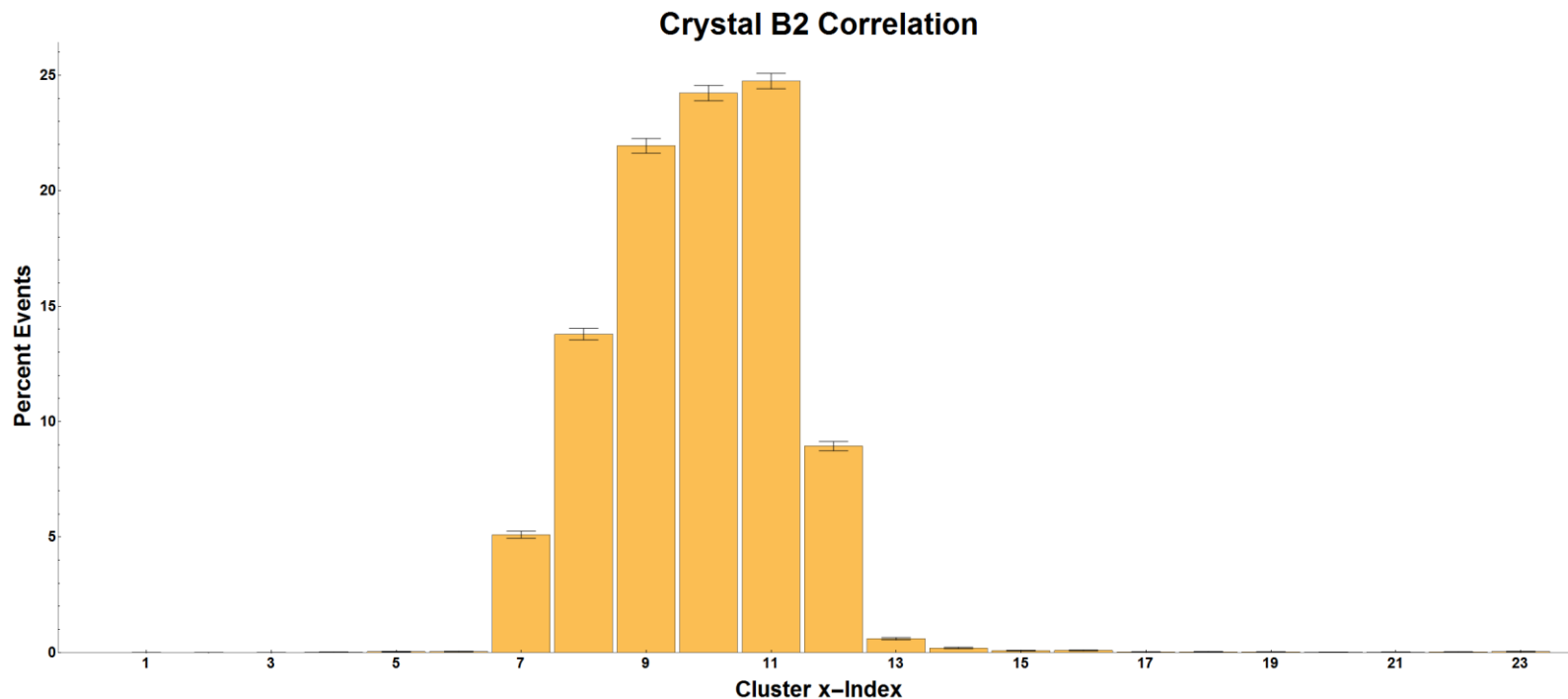
Trigger Tuning Overview

- The hodoscope trigger itself is then tuned.
 - The same process is then performed for each hodoscope scintillator and each calorimeter crystal.



Trigger Tuning Overview

- The hodoscope trigger itself is then tuned.
 - The same process is then performed for each hodoscope scintillator and each calorimeter crystal.



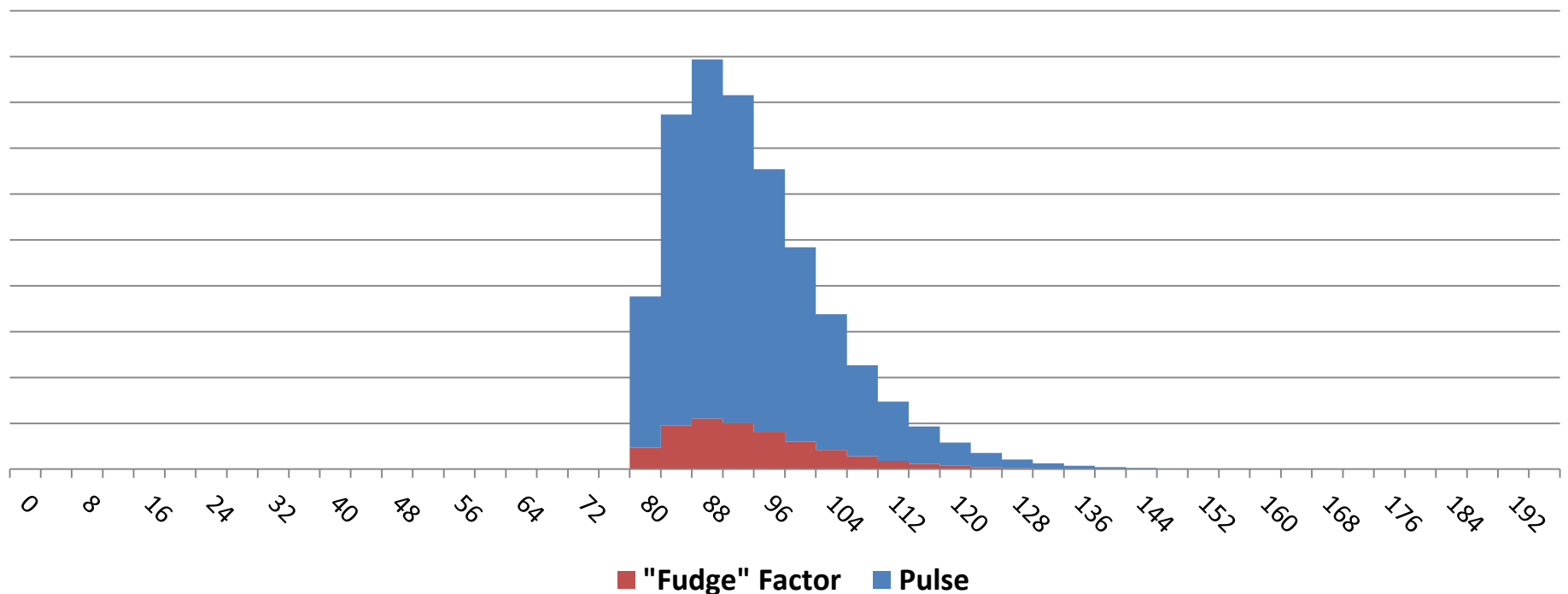
- This is used to produce a correlation table for the LIII trigger.

Conclusions

- The hodoscope readout chain is completed.
 - The hodoscope subsystem is now fully supported in HPS-Java throughout the readout chain.
 - Recon is still under construction.
 - Some framework exists, but there is more to-be-done.
 - Rafo is working on this.
- Work towards final trigger studies is underway.
 - Readout and recon is being processed for trigger studies MC on pure A' sample data.
 - The analysis code to produce the trigger cuts already exists.
 - The trigger itself is already designed; it just needs the final values.
 - Finally, rate estimates will be generated upon acquisition of final trigger parameters.

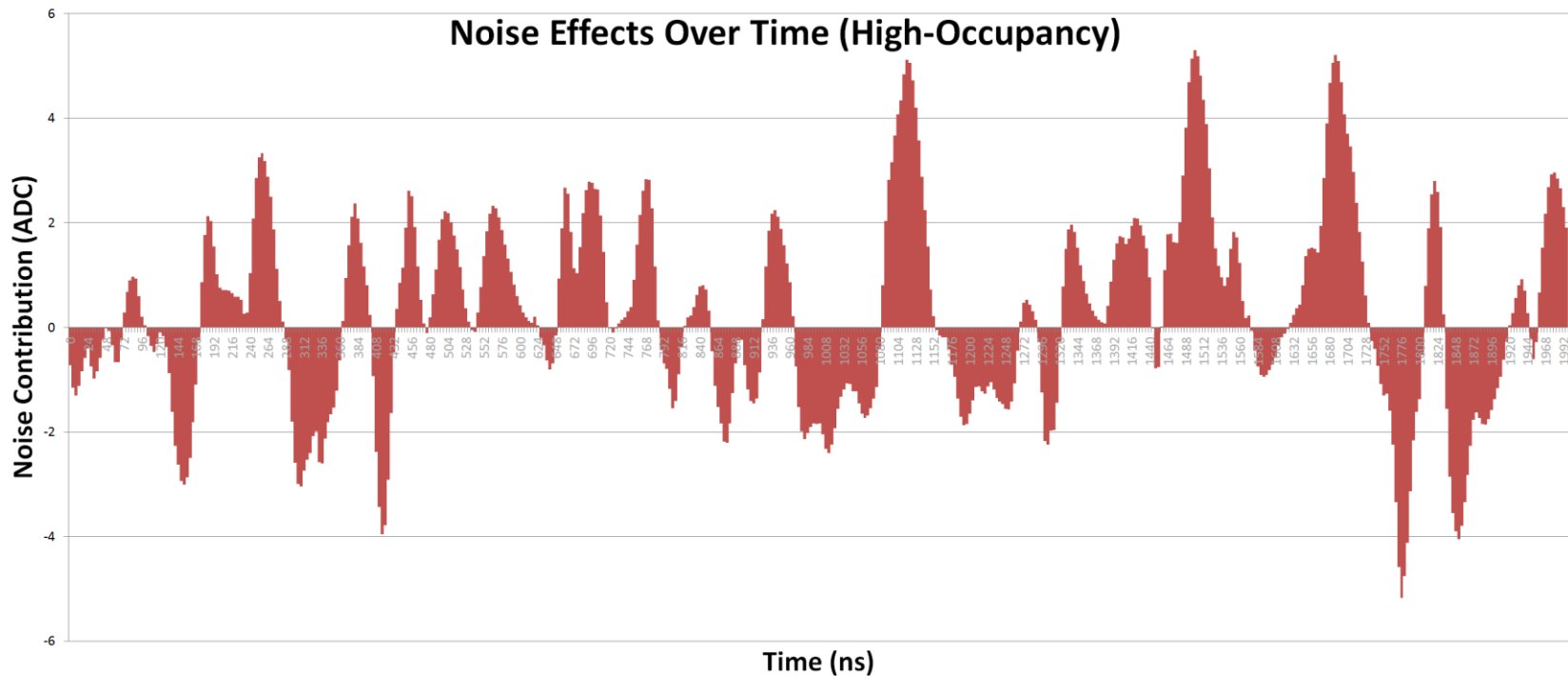
Aside – Noise Emulation

- Noise emulation is incorrectly handled in the digitization driver.
 - Noise is handled by adding a Gaussian “fudge” to the input hit energy.
 - This is wrong because it results in a uniform shift in energy in all affected voltage buffer samples.



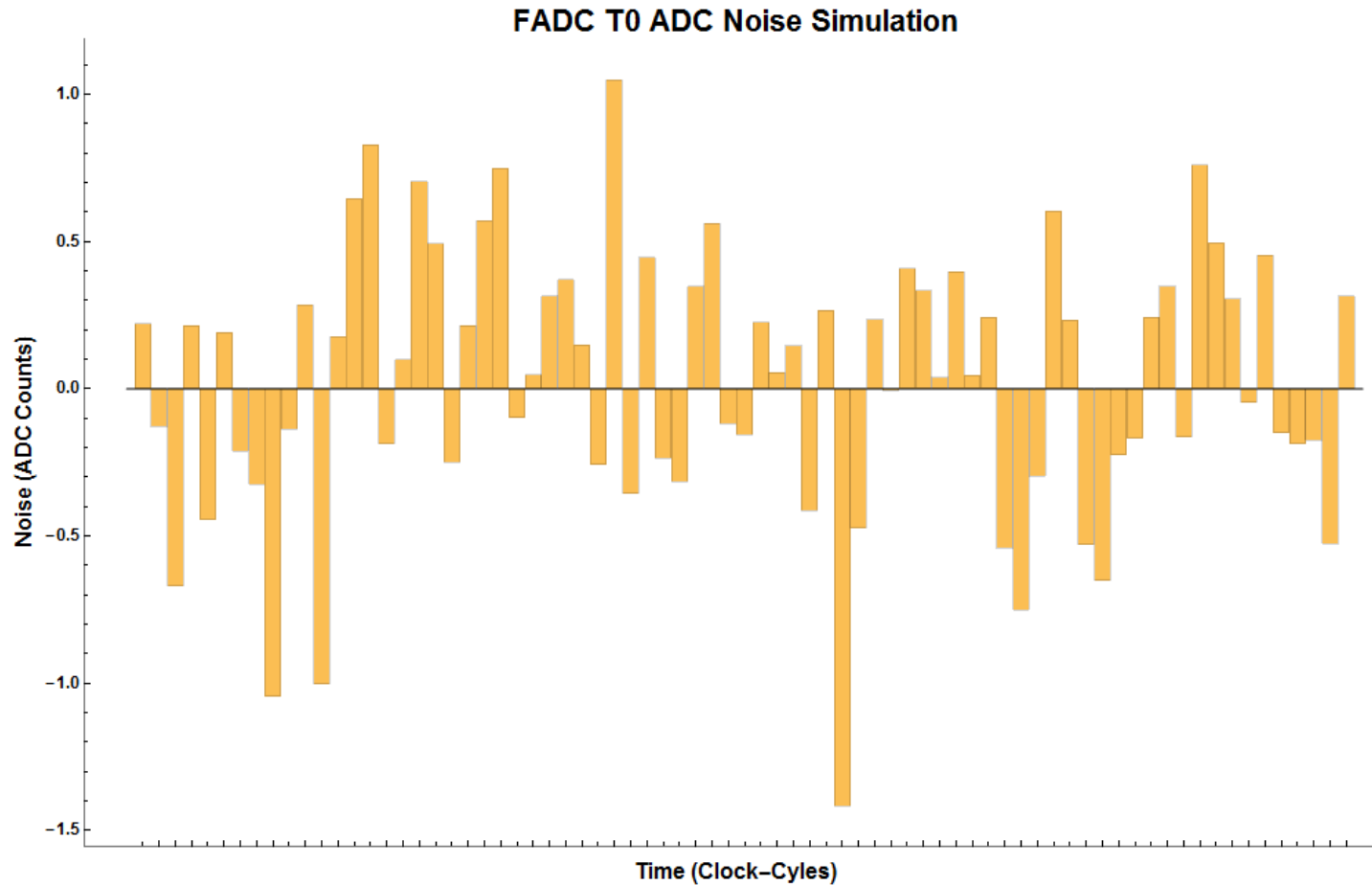
Aside – Noise Emulation

- This results in a series of localized energy shifts across the entire buffer in-time with truth hits.
 - Additionally, this causes “noise” to be occupancy-dependent.



Aside – Noise Emulation

- True noise is random – it more accurately resembles the below plot.



Aside – Noise Emulation

- This is an issue that was inherited from the calorimeter digitization driver.
 - While it affects the hodoscope readout, it is not directly a part of the simulation chain implementation.
 - As such, the noise issue has not been corrected.

- Because the noise is improperly handled from the start, it is disabled in the hodoscope simulation chain.