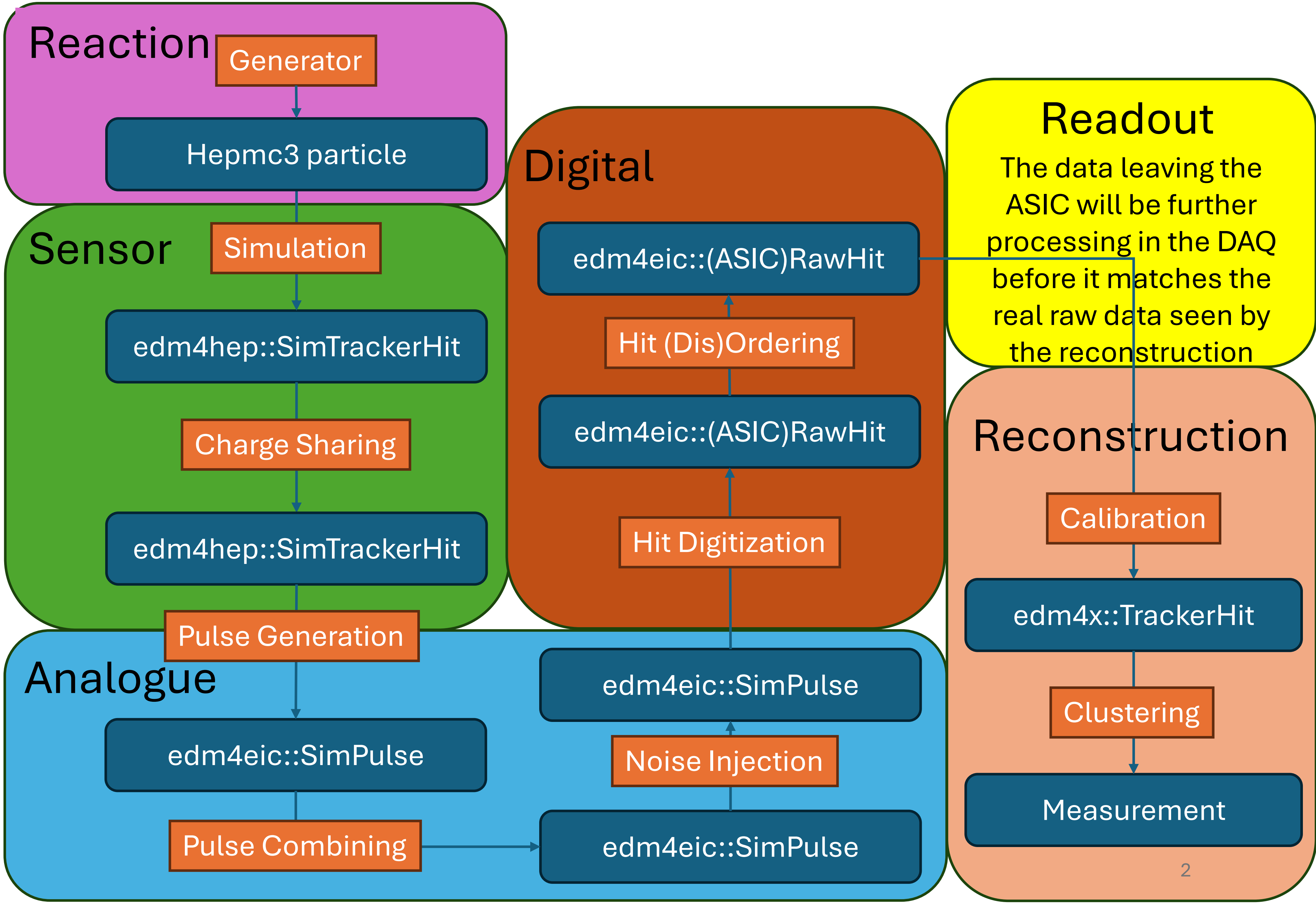
EICrecon Digitization developments

Tracking Meeting

Simon Gardner

17/04/2025



Reaction

Generator

Hepmc3 particle

Sensor

Simulation

edm4hep::SimTrackerHit

Charge Sharing

edm4hep::SimTrackerHit

Pulse Generation

Analogue

edm4eic::SimPulse

Pulse Combining

edm4eic::SimPulse

Noise Injection

edm4eic::SimPulse

Digital

Hit Digitization

edm4eic::(ASIC)RawHit

Hit (Dis)Ordering

edm4eic::(ASIC)RawHit

Readout

The data leaving the ASIC will be further processing in the DAQ before it matches the real raw data seen by the reconstruction

Reconstruction

Calibration

edm4x::TrackerHit

Clustering

Measurement

Focused on analogue pulse manipulation which is generalizable between detectors.

Currently available

- edm4eic::SimPulse data structure
- Pulse generation from SimTrackerHit
- Merging of pulses
- Adding noise to pulses

- BTOF (AC-LGAD) Charge sharing
- EICROC pulse digitization

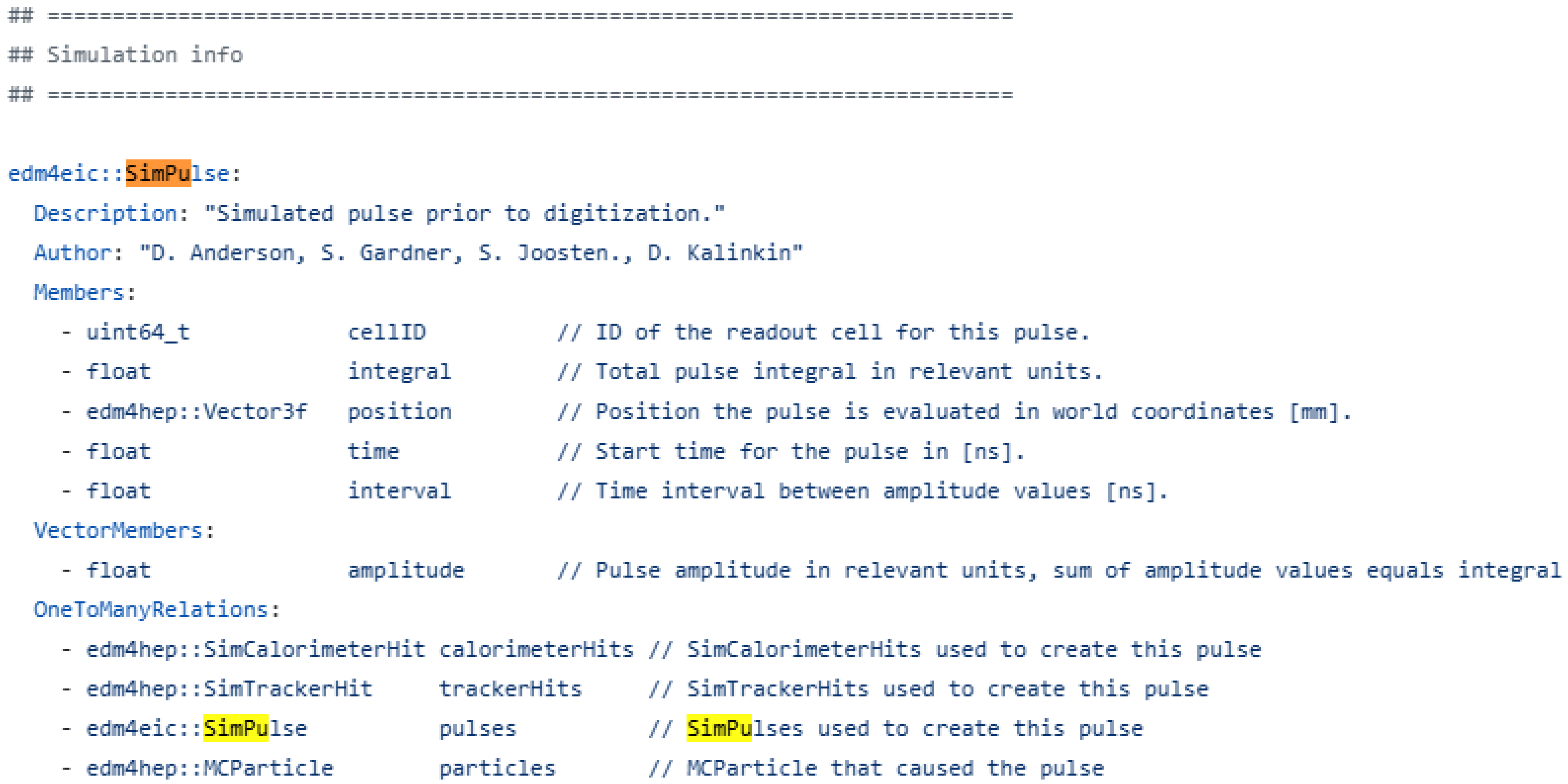
Planned developments

- More generalized charge sharing.
- Pulse shape and noise parameter variation.
- Timepix4 specific digitization (and hit ordering.)
- Other detectors

- MC+ML based approaches to charge sharing and pulse generation.
- Split Digitization steps from EICrecon.

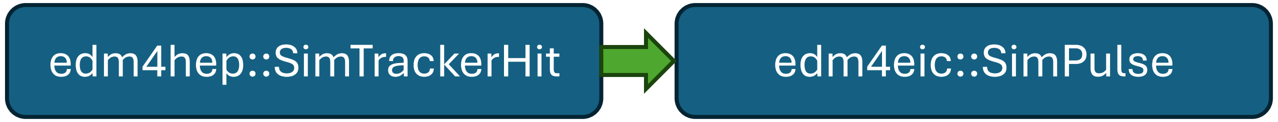
Data model update

- edm4eic::SimPulse introduced to represent analogue signals prior to digitization
- Keeps track of relationships through digitization chain (edm4hep equivalents do not)

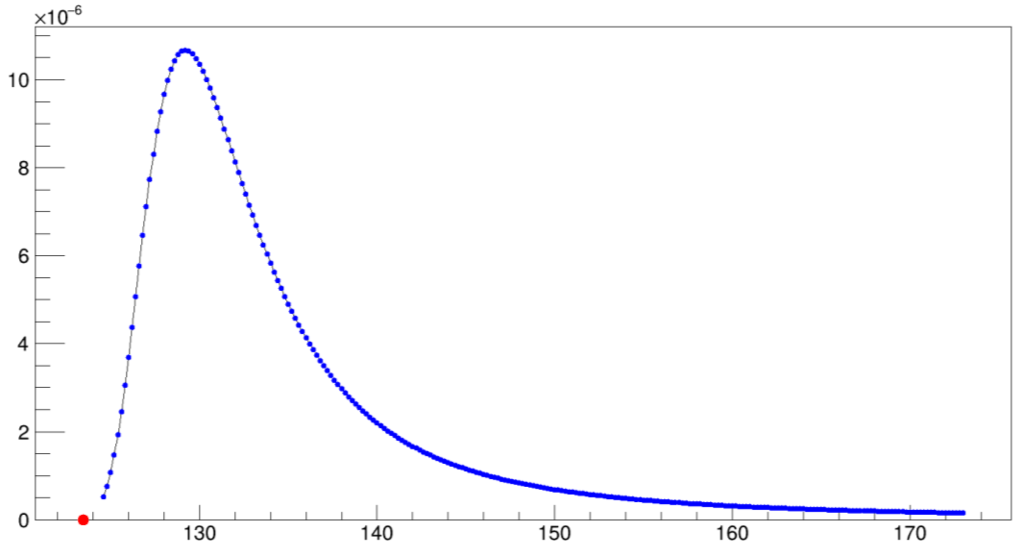


```
## ==========================================================================
## Simulation info
## ==========================================================================

edm4eic::SimPulse:
  Description: "Simulated pulse prior to digitization."
  Author: "D. Anderson, S. Gardner, S. Joosten., D. Kalinkin"
  Members:
    - uint64_t            cellID          // ID of the readout cell for this pulse.
    - float               integral        // Total pulse integral in relevant units.
    - edm4hep::Vector3f   position        // Position the pulse is evaluated in world coordinates [mm].
    - float               time            // Start time for the pulse in [ns].
    - float               interval        // Time interval between amplitude values [ns].
  VectorMembers:
    - float               amplitude       // Pulse amplitude in relevant units, sum of amplitude values equals integral
  OneToManyRelations:
    - edm4hep::SimCalorimeterHit calorimeterHits // SimCalorimeterHits used to create this pulse
    - edm4hep::SimTrackerHit     trackerHits     // SimTrackerHits used to create this pulse
    - edm4eic::SimPulse          pulses          // SimPulses used to create this pulse
    - edm4hep::MCParticle        particles       // MCParticle that caused the pulse
```

Generic Pulse Generation



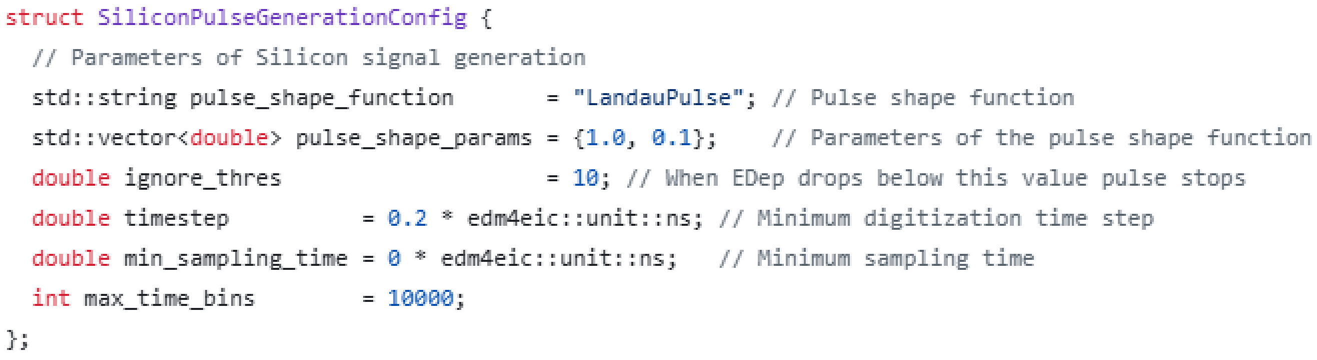
Landau Pulse - Hit (red point) time and pulse (blue).



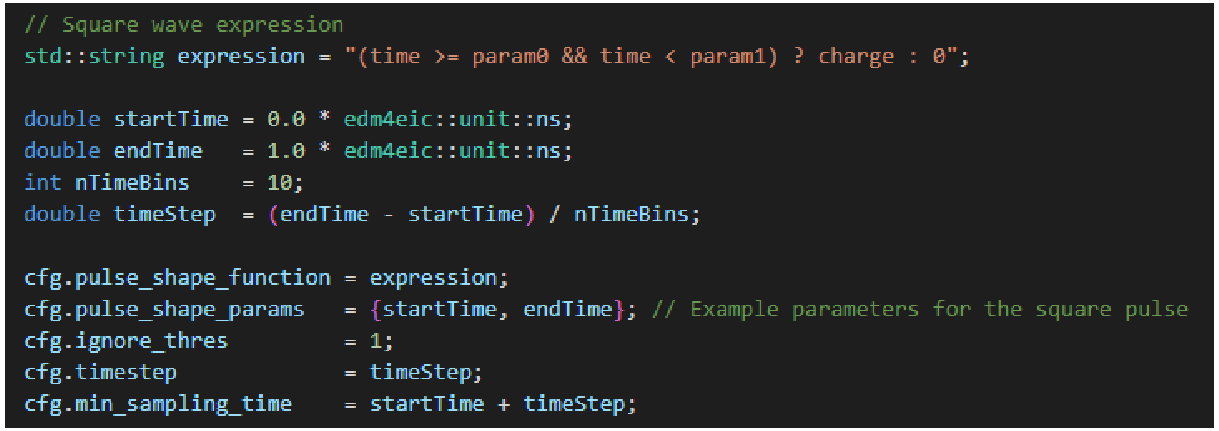
- Pulse shape currently parameterized by hit time and energy
- Pulse step needs to be equal to or smaller than fastest clock.
- Minimum pulse Threshold needs to be below any digitization threshold.
- More complex pulse functions could use position in cell and vector.
- Plans to add parameter variation functions to allow variation in parameterization between readout channels. E.g. a fraction of dead channels.
- EICrecon/src/algorithms/digi/SiliconPulseGeneration.cc at main · eic/EICrecon

Configuration allows function selection through dictionary or EvaluatorSvc

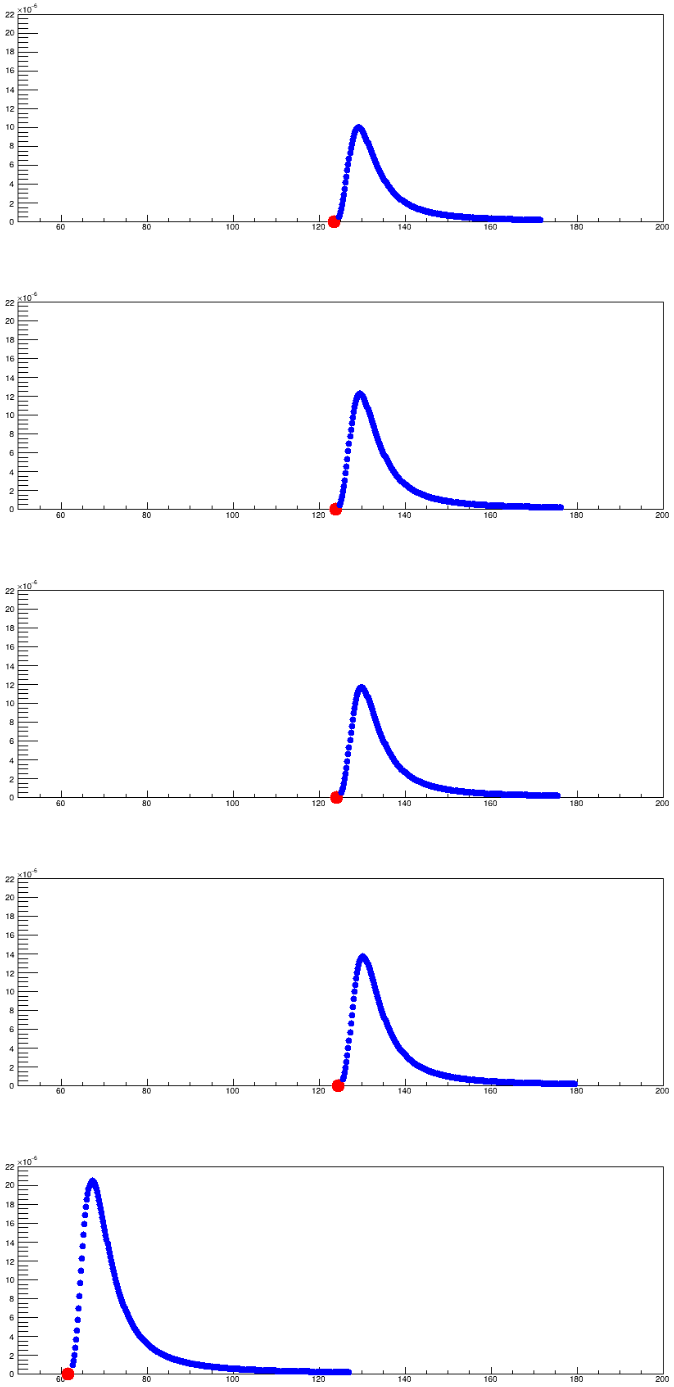
Dictionary currently only contains Landau but should be extended



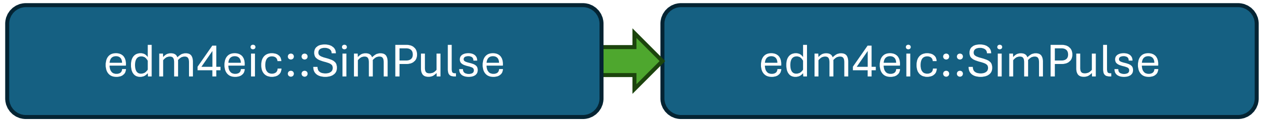
```
struct SiliconPulseGenerationConfig {
  // Parameters of Silicon signal generation
  std::string pulse_shape_function       = "LandauPulse"; // Pulse shape function
  std::vector<double> pulse_shape_params = {1.0, 0.1};    // Parameters of the pulse shape function
  double ignore_thres                    = 10; // When EDep drops below this value pulse stops
  double timestep          = 0.2 * edm4eic::unit::ns; // Minimum digitization time step
  double min_sampling_time = 0 * edm4eic::unit::ns;   // Minimum sampling time
  int max_time_bins        = 10000;
};
```

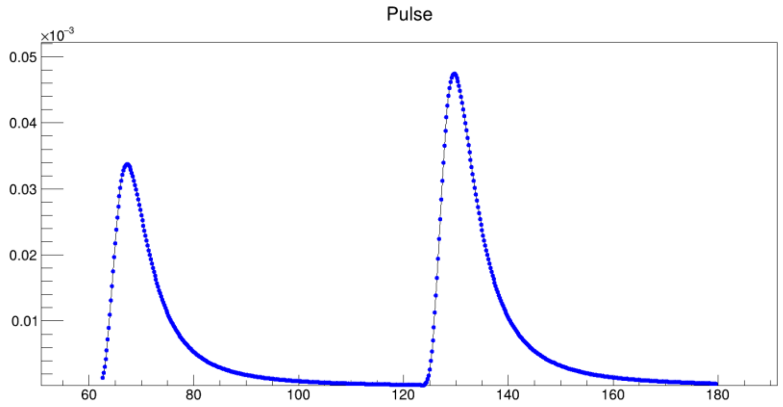


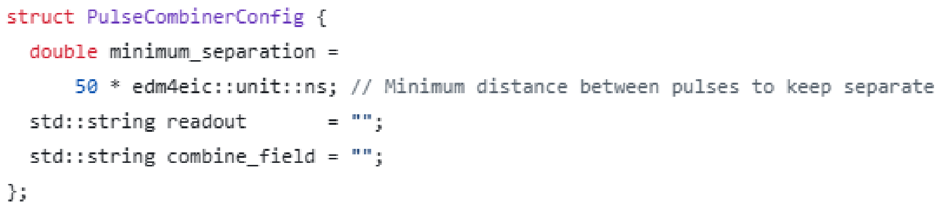
```
// Square wave expression
std::string expression = "(time >= param0 && time < param1) ? charge : 0";

double startTime = 0.0 * edm4eic::unit::ns;
double endTime   = 1.0 * edm4eic::unit::ns;
int nTimeBins    = 10;
double timeStep  = (endTime - startTime) / nTimeBins;

cfg.pulse_shape_function = expression;
cfg.pulse_shape_params   = {startTime, endTime}; // Example parameters for the square pulse
cfg.ignore_thres         = 1;
cfg.timestep             = timeStep;
cfg.min_sampling_time    = startTime + timeStep;
```



Pulse Merging







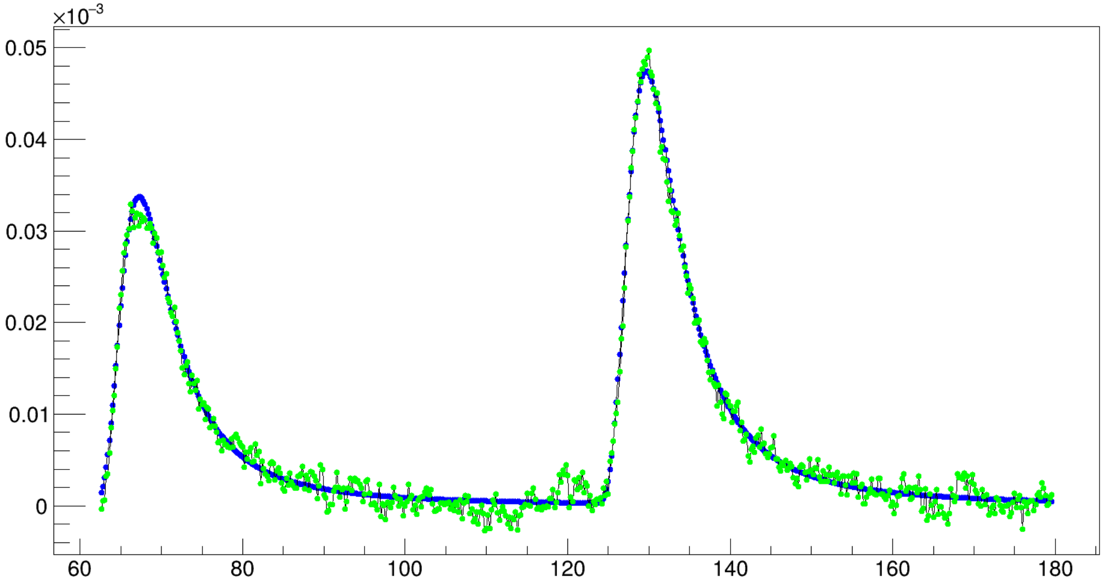
```
struct PulseCombinerConfig {
  double minimum_separation =
      50 * edm4eic::unit::ns; // Minimum distance between pulses to keep separate
  std::string readout       = "";
  std::string combine_field = "";
};
```

- Pulses summed together when the end of one pulse is within x time of the start of the next.
- x needs to be the maximum time a signal can influence another
  - This is guided by the slower clocks responsible for readout of the data.
- Linear summing of pulses - could extend to include non-linear responses.
- Configuration allows merging at any level of the readout field.
- EICrecon/src/algorithms/digi/PulseCombiner.cc at main · eic/EICrecon

Noise Injection

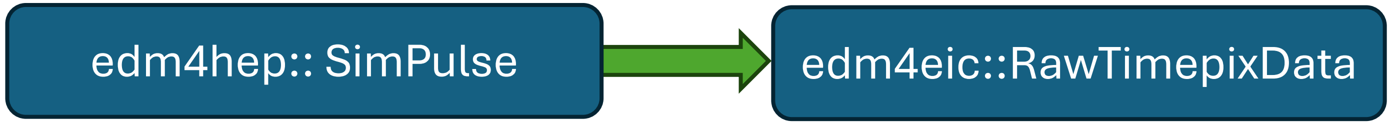
edm4hep:: SimPulse

edm4hep:: SimPulse



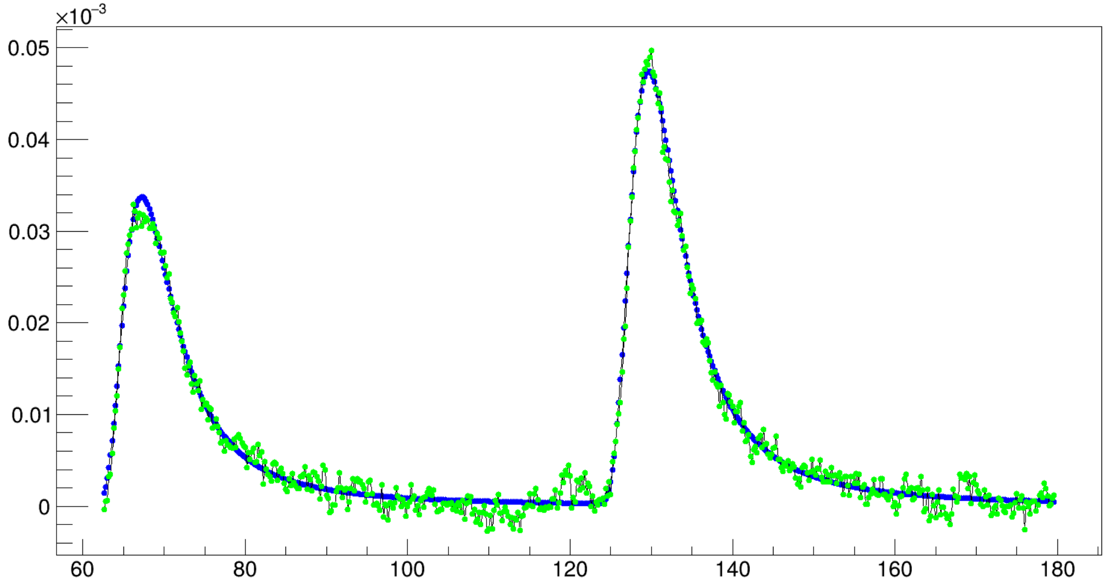
- Alpha noise injection into the pulse across frequencies, provided by DDDigi: DD4hep/DDDigi/src/noise/FalphaNoise.cpp at master · AIDASoft/DD4hep
- Will not add independent noise hits.
- Plans to add parameter variation functions to allow variation in parameterization between readout channels.
- Types of noise other than Alpha could be included.
- EICrecon/src/algorithms/digi/PulseNoise.cc at main · eic/EICrecon

ASIC specific digitization (Timepix4)



edm4hep:: SimPulse

edm4eic::RawTimepixData



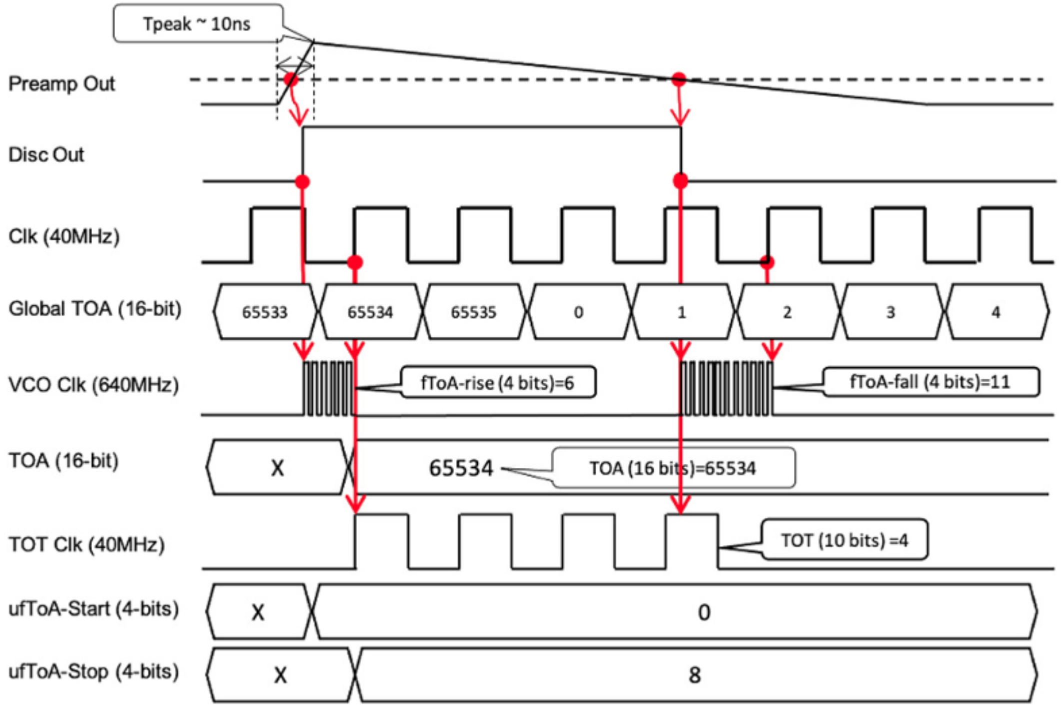
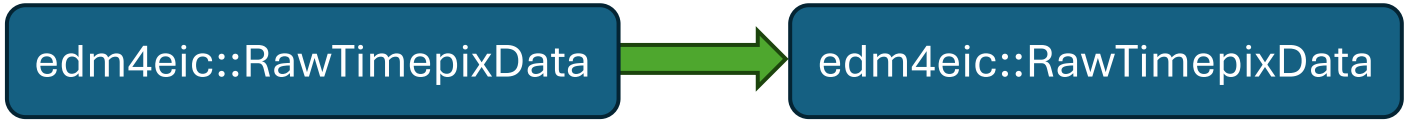


Figure 4. Timing diagram for the Timepix4 pixel cell in data driven mode.

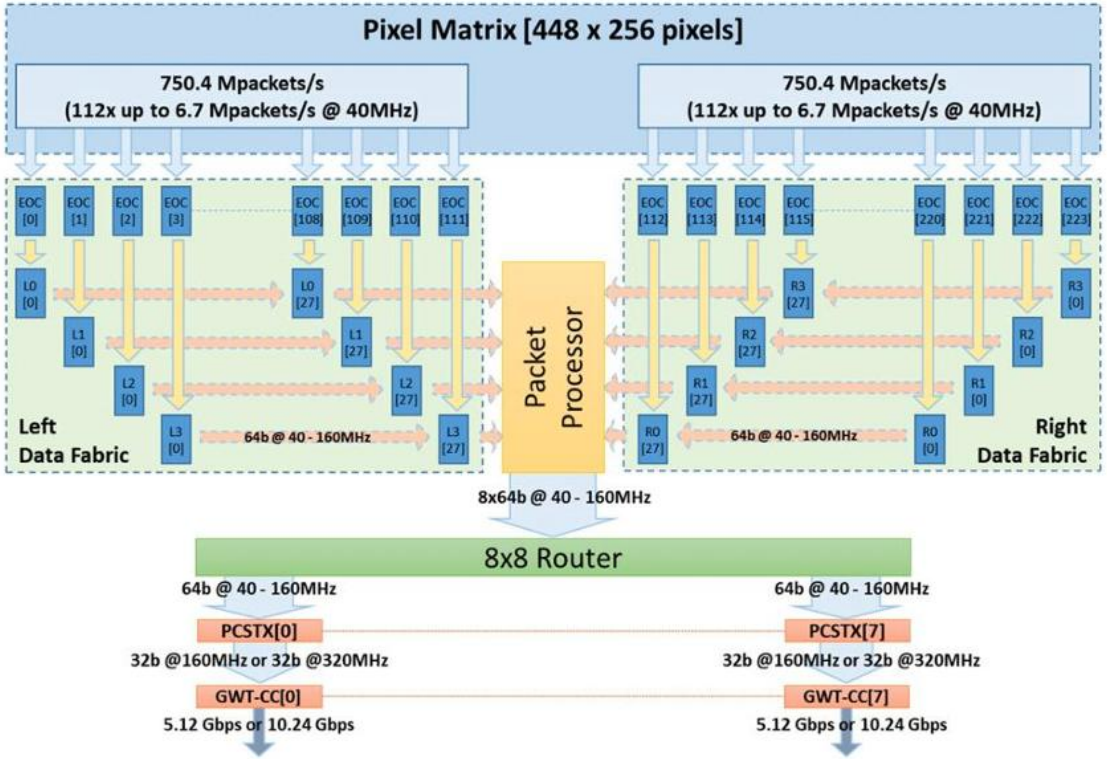
Timepix4 - X. Llopart et al 2022

- Implementation not ready but relatively simple
- Overlays clocks and threshold onto the pulse to determine ToA, ToT, fToA, ufToA.
- Should encode into Timepix4 64bit readout.
- VCO frequency varies slightly across chip.

ASIC specific Hit ordering



- The order of the hits in the datastream will not be ordered by ToA
    - E.g. Hits with a longer ToT will take longer to be processed
    - Hits in different areas of the ASIC take preference.
    - Sorting has to be done to some degree at some point in the DAQ so understanding how the events are disordered is important.
- Might require buffering beyond the scale of Timeframes/Super-timeframes.



Timepix4 - X. Llopart et al 2022

Questions?

Low-Q^2 Tagger and Timepix4

- A realistic digitization scheme and background model are perhaps more important for the Low-Q^2 Tagger than other detectors at this point.
  - Regions of phase space might be impossible to cover at certain beam conditions.
- High Bremsstrahlung rates increasing with Z^2 of ion.
  - Unavoidable overlap with Low-Q2 physics electrons.
- High Synchrotron backgrounds
  - Highly concerning but design of beampipe, magnets and exit window can mitigate the problem.
- **The Timepix4 ASIC already exists and is well understood so is a good example to build from.**
- Related work on MC/ML based digitization using Allpix2 is also being developed.

What is a edm4hep::SimTrackerHit from dd4hep?

- A sim tracker hit is recorded at a weighted central point of particle steps through the sensitive detector.
- Usually only a single hit will be recorded.
- Multiple hits in the same cell will occur when another particle takes a step in the detector.
  - When secondaries created near/in the element summing of signals to reproduce a realistic size is important
- Some detectors with hot spots might see multiple hits in a time frame from physics or other backgrounds
  - Having a realistic digitization which will demonstrate the detectors' ability to separate the hits is important.