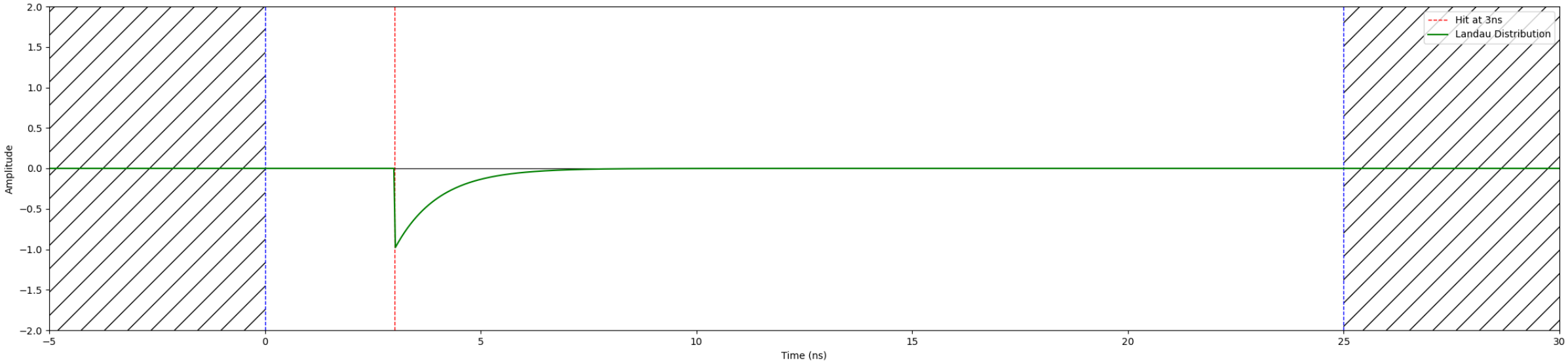
Current Implementation:

- Time between 0-25ns considered for digitization.
- BTOF hit time converted into Landau distribution sampled in 20ps steps

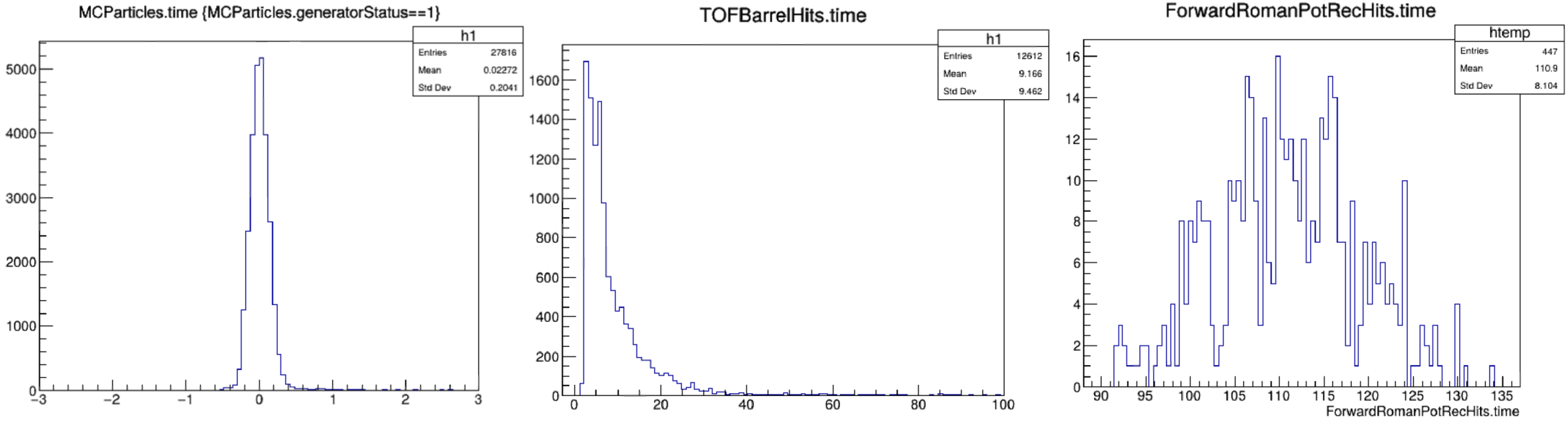


Example file

24.08.1.pythia_ep_noradcor_5x41_q2_0.000000001_1.0_run9.ab.0397.eicrecon.tree.edm4eic.root

- MC origin time from single bunch crossing ~100ps FWHM spread.
- TOFBarrel hits mostly within the first 25 ns but not 100%
- RomanPot hits100+ ns downstream of IP

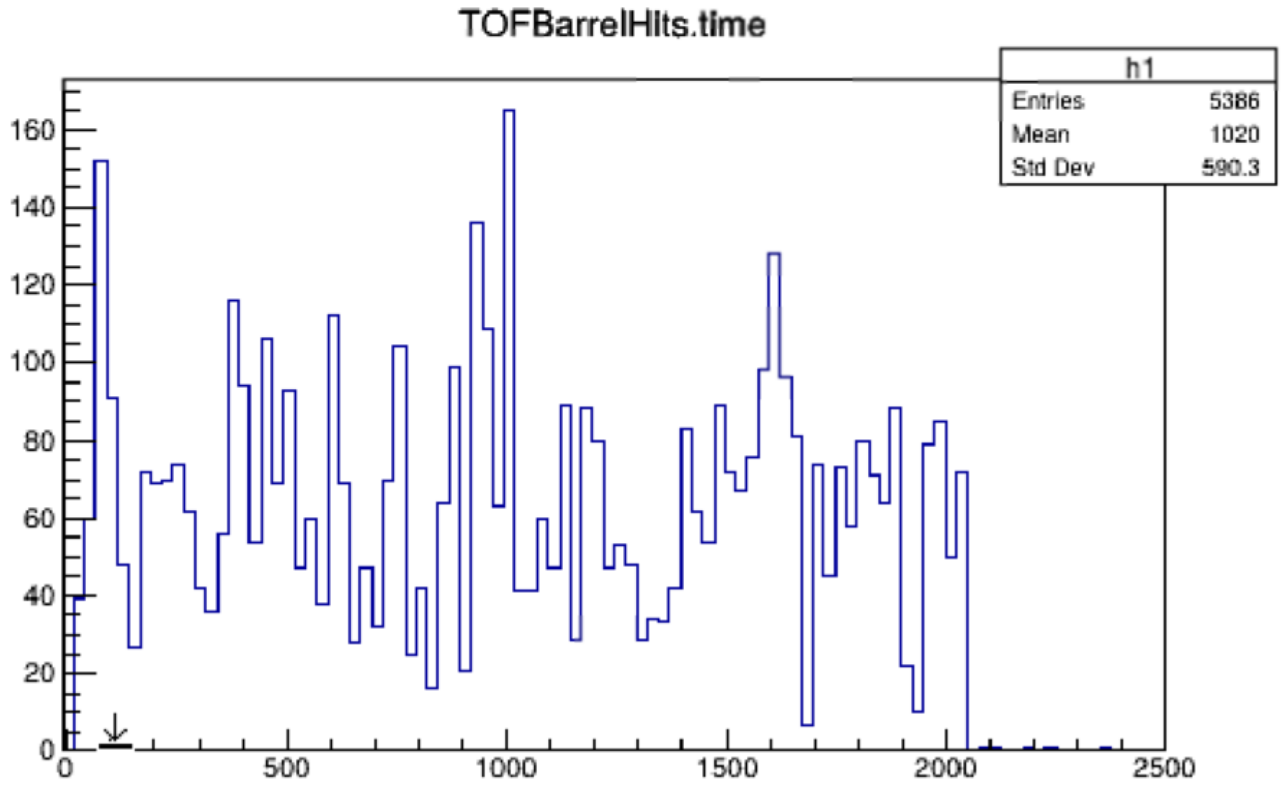
Initial vertex start times



Timeframe File

HEPMC_merger-1.0.2_bgmerged_RealisticSignalPerFrame_MinBias_pythia6_10x100_egas_bgas.0030.eicrecon.tree.edm4eic.root

- TOF Barrel hits distributed “evenly” throughout 2us timeframe
- Algorithm needs to work for a hit at any time, t=0 is in no way special.



Problems

- Any hits occurring outside of the 0-25ns window are lost
- Hits near the end of the 25ns period will lose their amplitude.

Assumptions

- The 10 bit fast timing resolution of the EICROC is supplemented by more bits from the global clock
- 40MHz clock needs to be free running (synchronized to 100MHz accelerator).

Solution

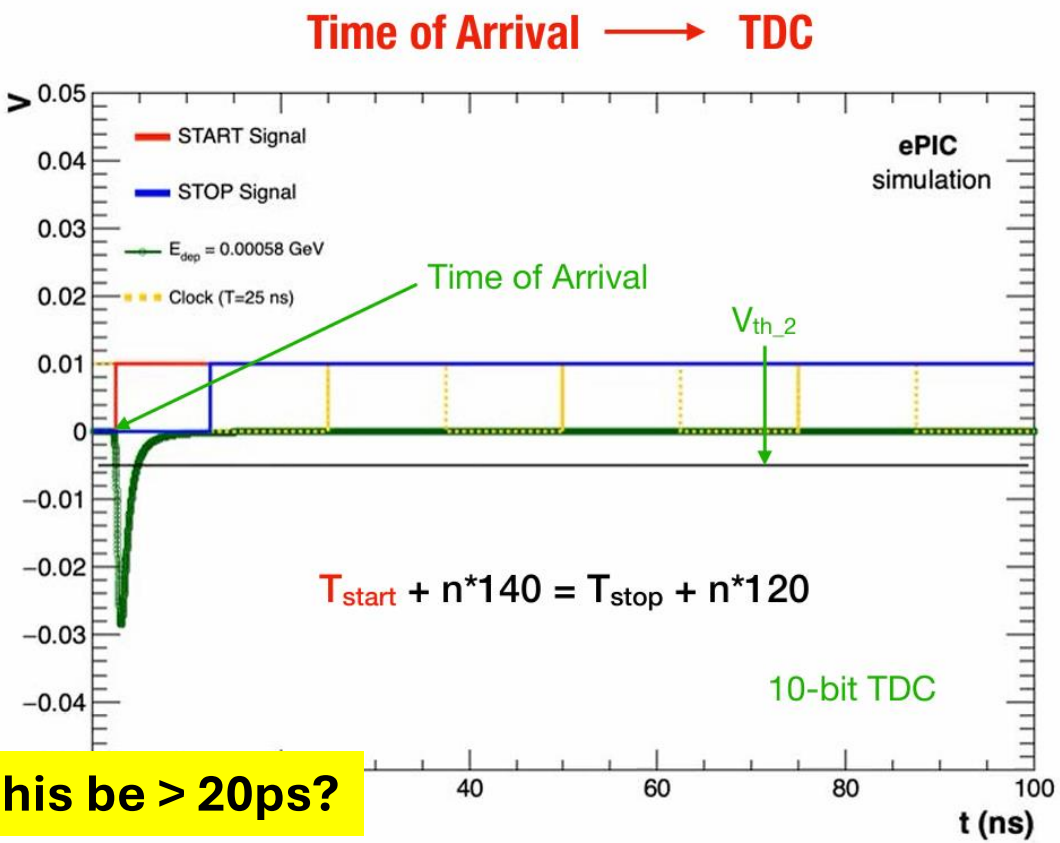
- Generate the pulse starting from the simulated hit time.
  - The 20ps samplings should be aligned so pulses can be summed easily
  - Round the earliest sample time to an integer multiple of 20ps
- The pulse doesn’t need to span a whole 25ns if it is back at 0 much sooner
- Pulse summing can be shifted to a separate algorithm which checks if the range of the pulse overlaps

Other questions - (maybe for EICROC designers)

I cannot see how the two half periods cannot be distinguished in this way, particularly with 100MHz bunch crossing.

Slide from presentation by Prithwish in Lehigh: digitization_aclgad

- Almost all hits occur in the 1st half-period of the clock(f = 40MHz). When the analog signal crosses V_{th_2} (Time of Arrival), the **START** signal flips from 0 to 1. When the clock cycle flips from the 1st to the 2nd half-period, the **STOP** signal flips from 0 to 1.
- Consecutive delay cells propagate the START signal (140 ps delay) and the STOP signal (120 ps delay) in parallel until the START signal crosses the STOP signal (Mathematically, **|START-STOP| < 20 ps**).



Quantization time of ePIC ToF detector

Should this be > 20ps?

- The number of times the signals move ($1 \le n \le 1024$) is converted to a 10-bit TDC code.

Other questions - (maybe for EICROC designers)

- During the time: $T_{start} + n*140 = T_{stop} + n*120$
  - Is 1024*140ps, the pixel deadtime or is it reset when n is reached?
  - Or are there multiple circuits able to handle multiple hits?
  - Is the amplitude taken as the max value crossed before the lowest threshold is crossed again, max value before n is reached or max value in 1024*140ps?
- The 140 and 120ps clocks presumably are free running too, or do they get started by the threshold crossing.