BWEC EMC digitization in PandaRoot

Guang Zhao (zhaog@ihep.ac.cn), Oliver Noll

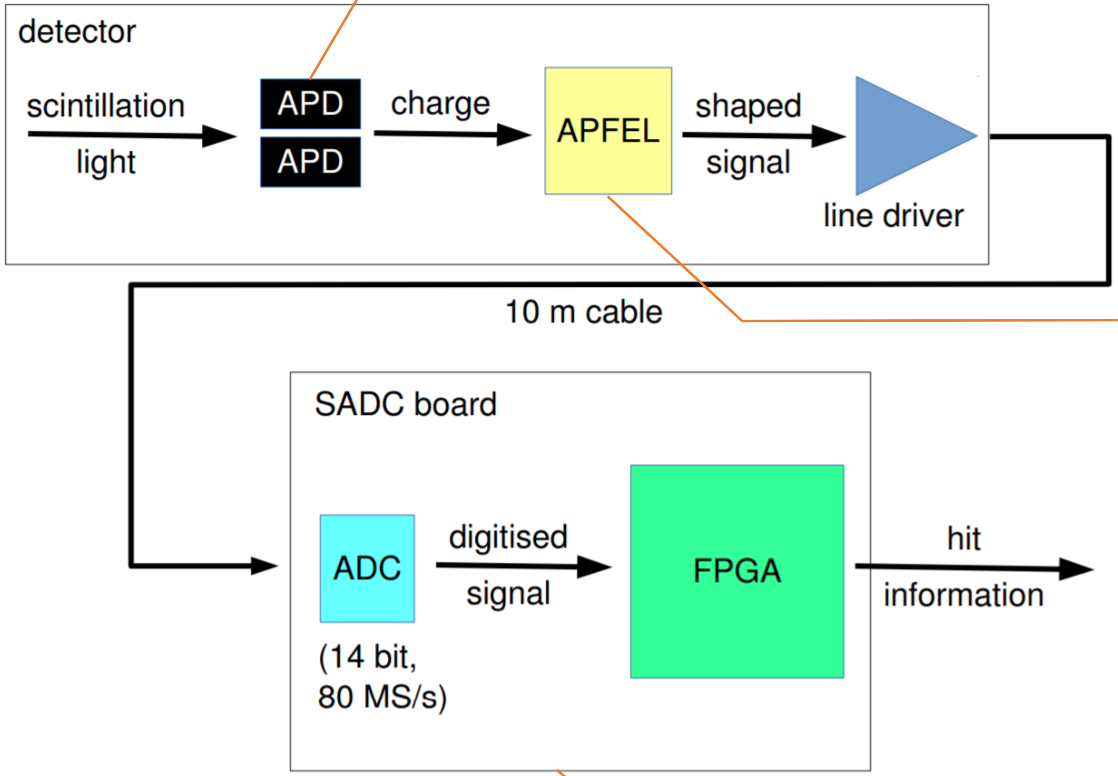
Combined EMC Hardware/Software Meeting

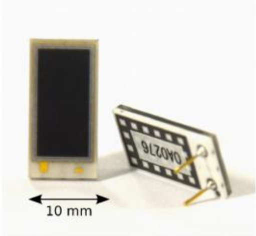
Jan 17th, 2020

Outline

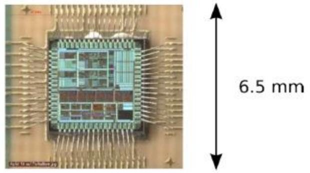
- **Introduction**
- **Digitization implementation**
  - New from the last CM: Optimized noise model, duo-APD output, checks/bug fixing
- **Code development in PandaRoot**
- **Summary**

Introduction: BWEC readout

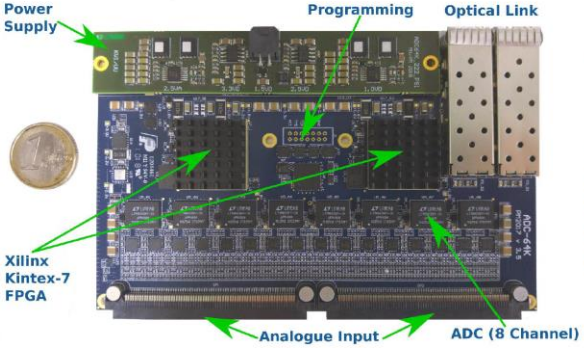




- Large area APD (7x14 mm²)
- Capacitance: 270 pF (full depletion)
- Operated at gain ~200
- Two APD per crystal

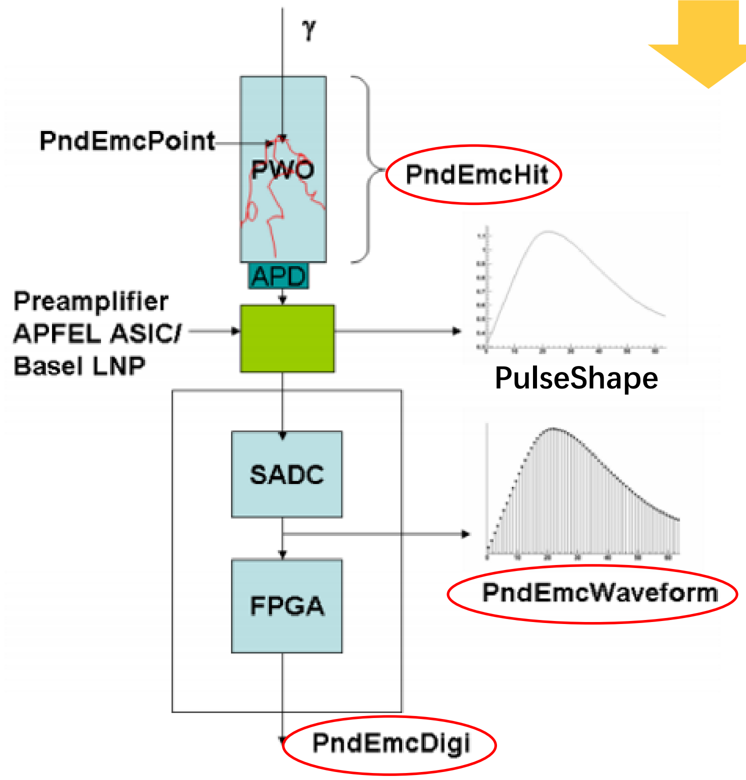


- Charge sensitive preamplifier: APFEL (ASIC for the PANDA Frontend ELectronics)
- Reads out two APD (one crystal)
- Low noise input stage
- Shaper (~1 µs shaping time)
- Two main amplifier (gain 1 and 10)
- 4 output signals (2 APD x 2 gains)
- Low power consumption (~100 mW)



- Developed at University of Uppsala by Pawel Marciniewski
- 64 ADC channel
- 14 bit
- 80 MHz
- Two FPGA's
- Two optical links

Introduction: Digitization process in PandaRoot



Signal Generator (SG)

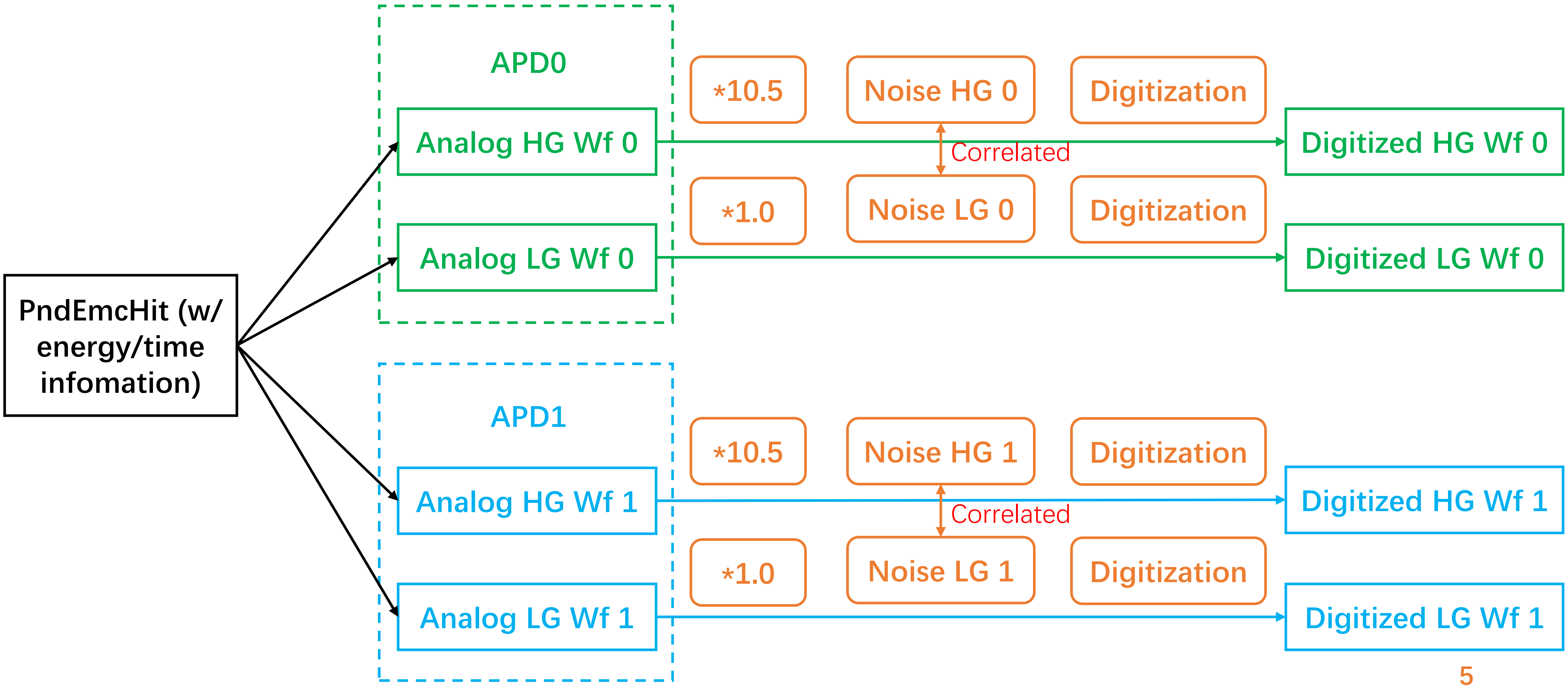
- Analog waveforms creation
- Noises generation
- Digitization
- Pile-up waveforms creation

Feature Extraction (FE)

- Hit detection
- Amplitude/time extraction
- Pile-up recovery

Time-based simulation

Signal Generator

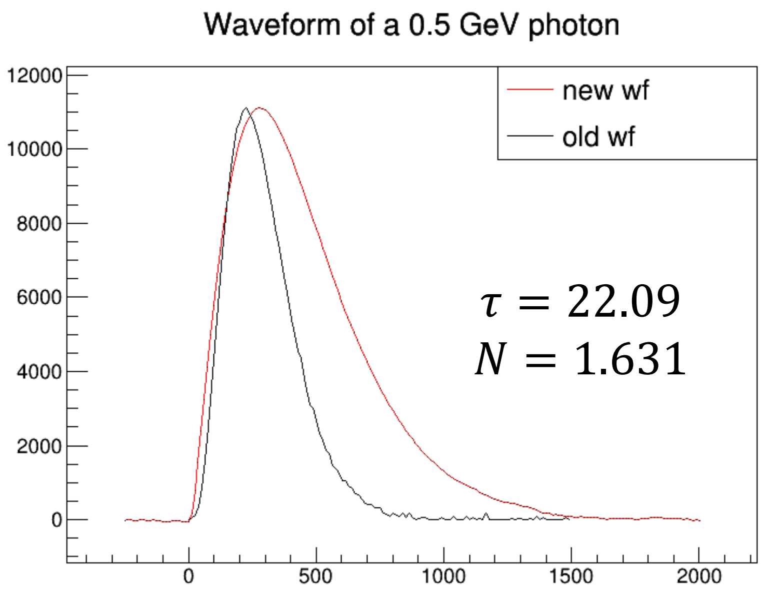


Signal Generator: Pulses

$$f(x) = -A \cdot e^{\frac{-N(x-\delta)}{\tau}} \cdot \left(\frac{x-\delta}{\tau}\right)^N \tag{2.1}$$

Whereby τ is describing the decay behavior. N has an impact on the rising and decay ratio. δ shifts the pulse in time. A is proportional to the pulse hight H:

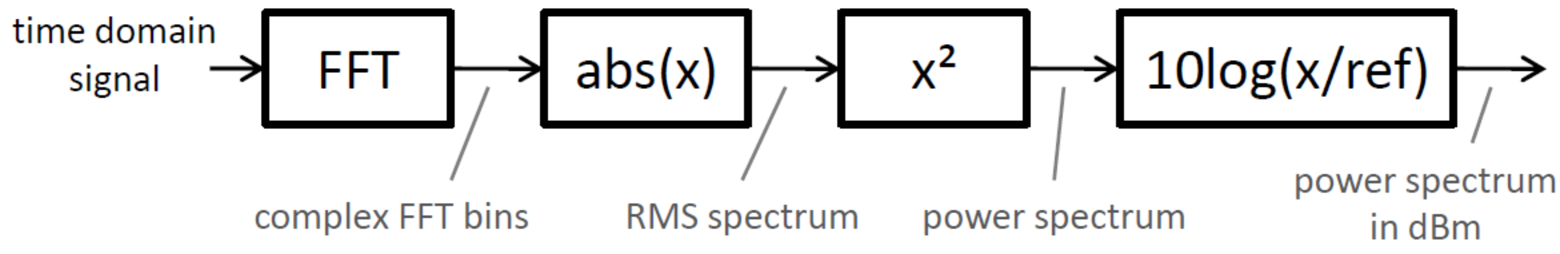
$$A = H \cdot e^N \tag{2.2}$$

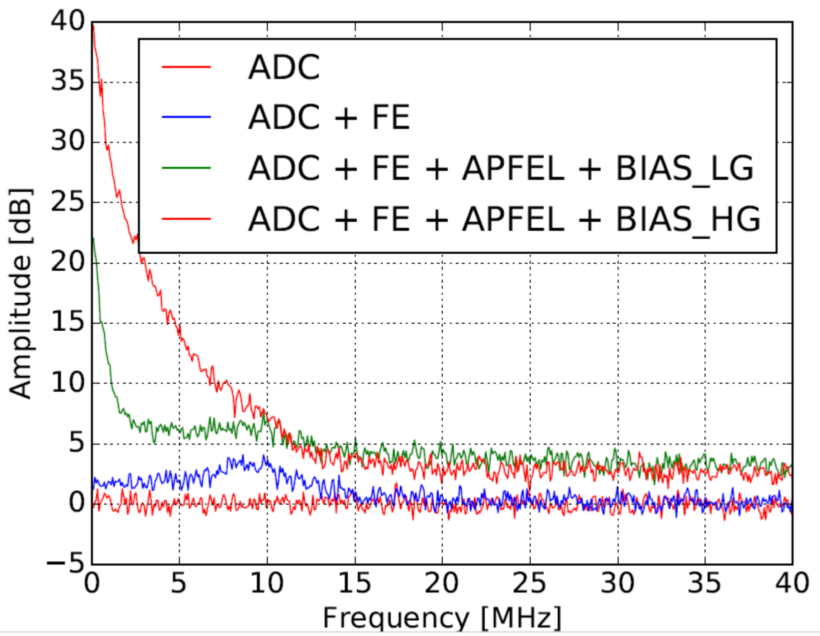


- APD gain = 200
- APFEL amplifier: 2 gains
  - HG/LG = 10.5
- Full pulse width: ~1700 ns
- Rising time: ~300 ns
- APFEL ASIC pulse digitized by the SADC

Signal Generator: Noises

FFT analysis for the noises



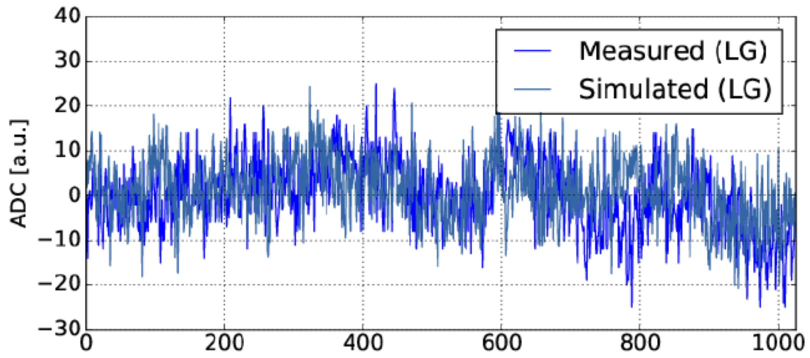


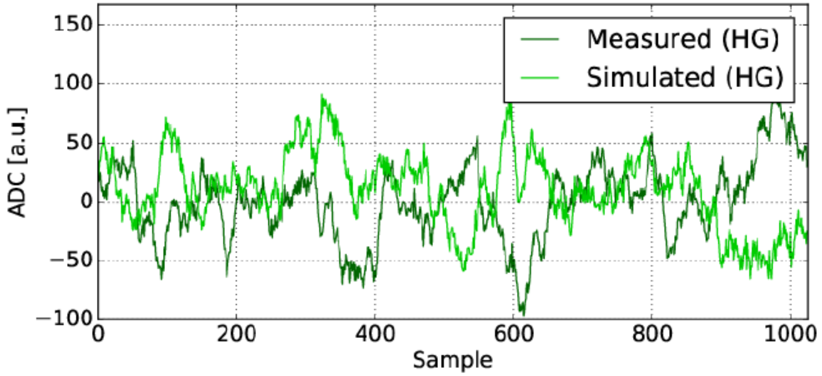
- **Noise components**
  - Biased APD, APFEL preamplifier at low/high gain
  - Open ADC entrance
  - Front-end electronics transmission
- **Noise measurement**
  - FTT analysis of the noises
- **Noise simulation**
  - iFFT of the power spectrum to obtain time-domain noises

Signal Generator: Noise (II)

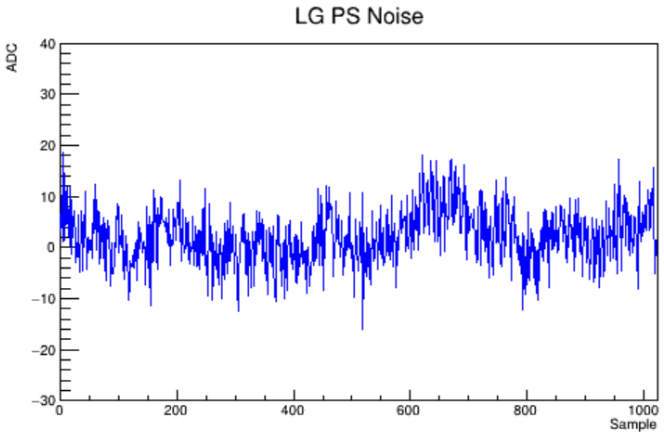
Biased APD, APFEL preamplifier for low/high gain (correlated)

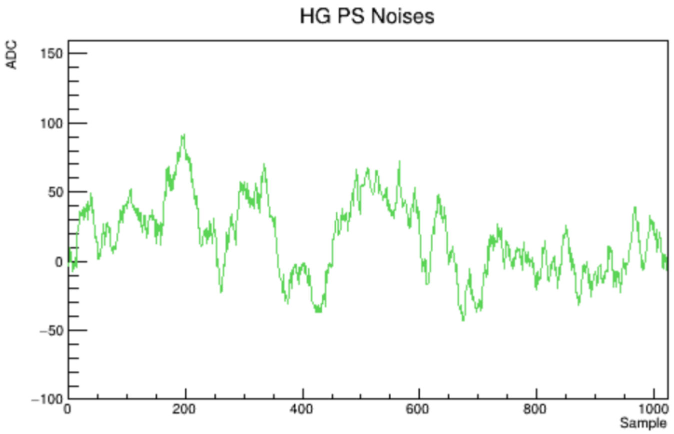
Measurement





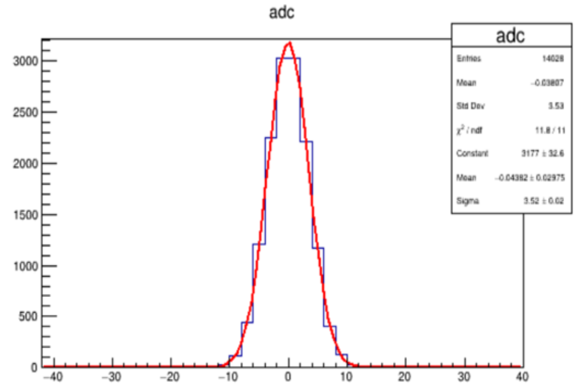
PandaRoot Sim





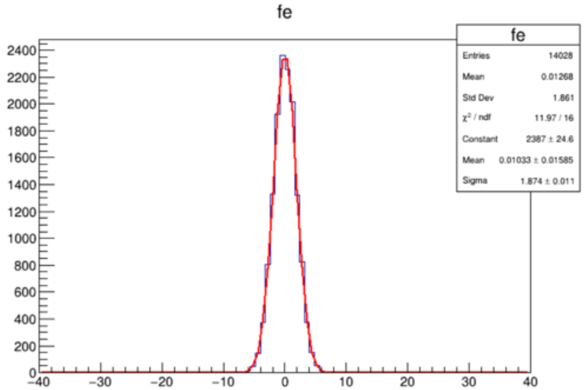
✓ Good agreement between simulation and measurement

ADC & FE Transmission



Measured value: 3.5

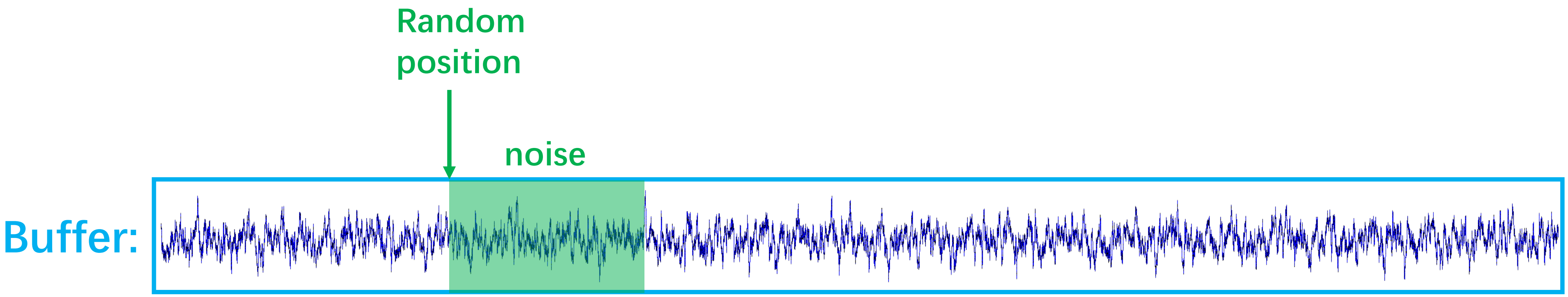
Simulated value: 3.52 +/- 0.02



Measured value: 1.89

Simulated value: 1.874 +/- 0.011

An approximated noise model



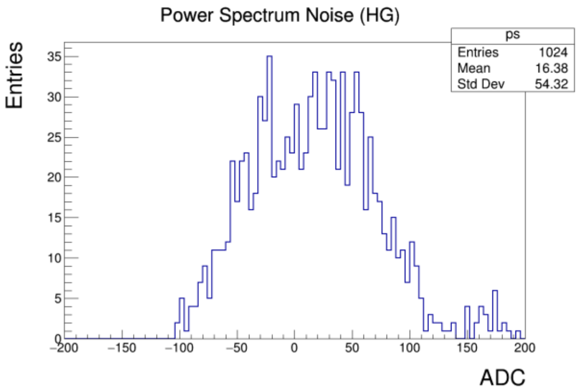
- ✓ **<u>Noise modeling</u>**
  - ✓ Pre-generate a big noise buffer
  - ✓ Pick up the noise of a waveform from a random position in the buffer.

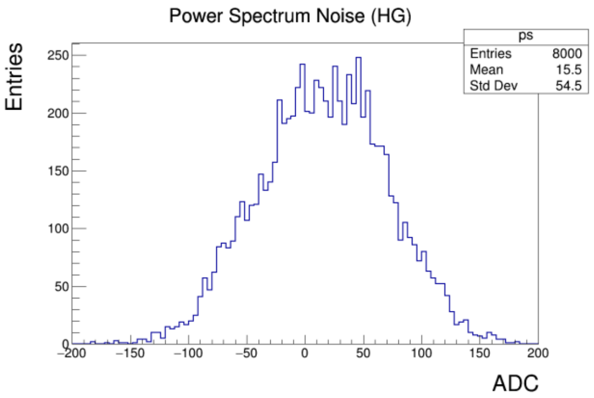
- ✓ **<u>Pros:</u>** Much faster
- ✓ **<u>Cons:</u>** Loose some randomness

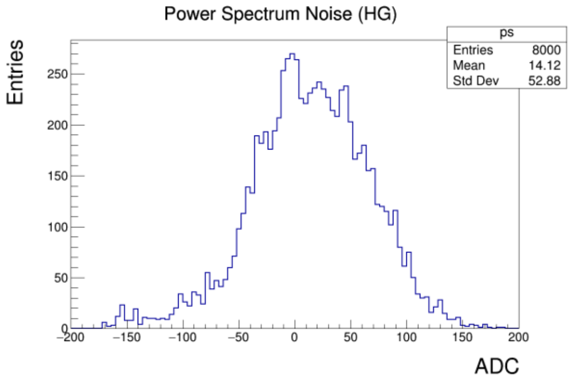
# of waveforms	CPU Time (sec)	
	Full iFFT	Reduced iFFT
100	1.218	1.106
200	1.405	1.079
500	2.413	1.047
1000	4.147	1.105
5000	18.096	1.193

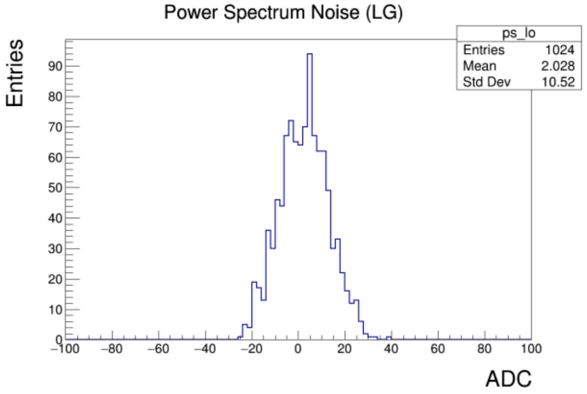
Noise comparisons

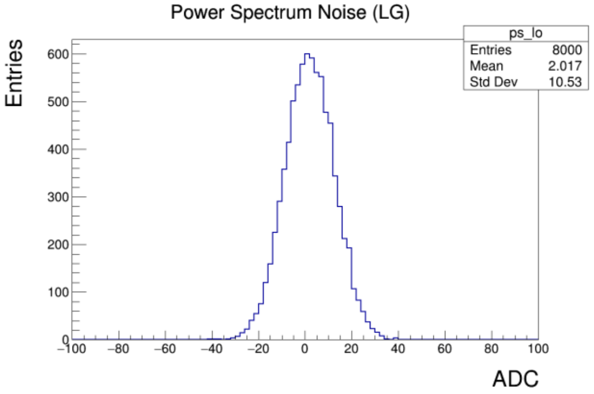
From Oliver	w/ full iFFT	w/ reduced iFFT

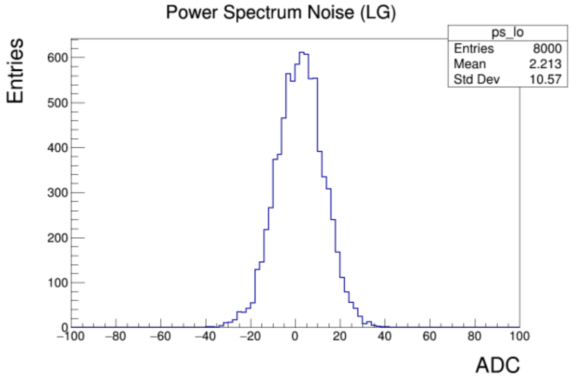










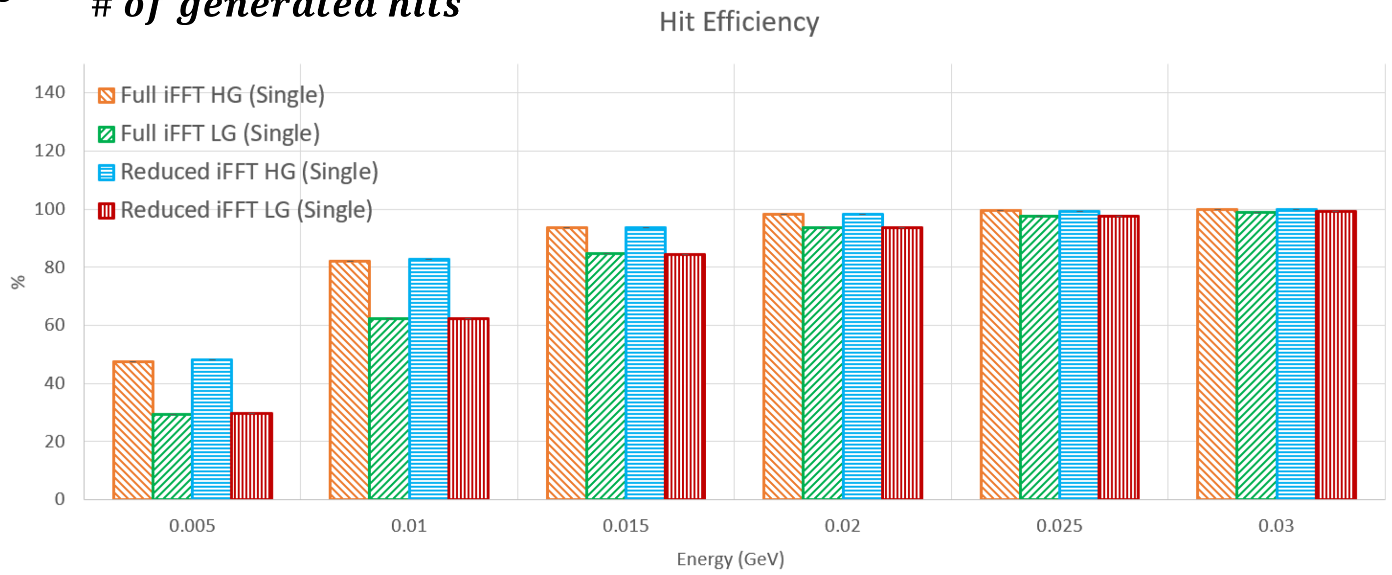


No obvious discrepancies on mean/rms distributions for the reduced iFFT

Noise rate/hit efficiency check (single APD)

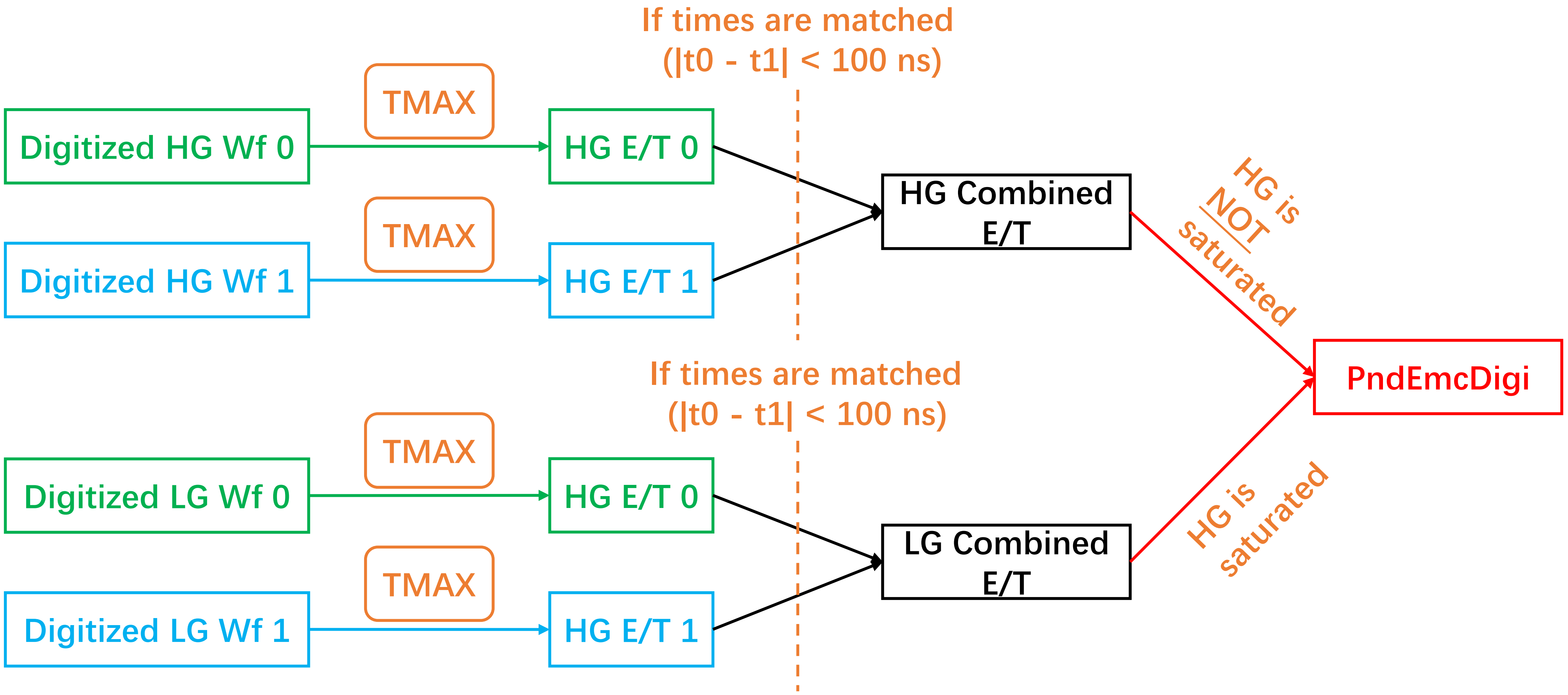
- **Noise rate = # of noises / sec [Hz]**
  - Full iFFT: 160.5 +/- 1.3 kHz (HG), 332.4 +/- 2.6 kHz (LG)
  - Reduced iFFT: 166.0 +/- 1.3 kHz (HG), 332.9 +/- 2.6 kHz (LG)

- **Hit efficiency =** $\frac{\#\ of\ detected\ hits}{\#\ of\ generated\ hits}$

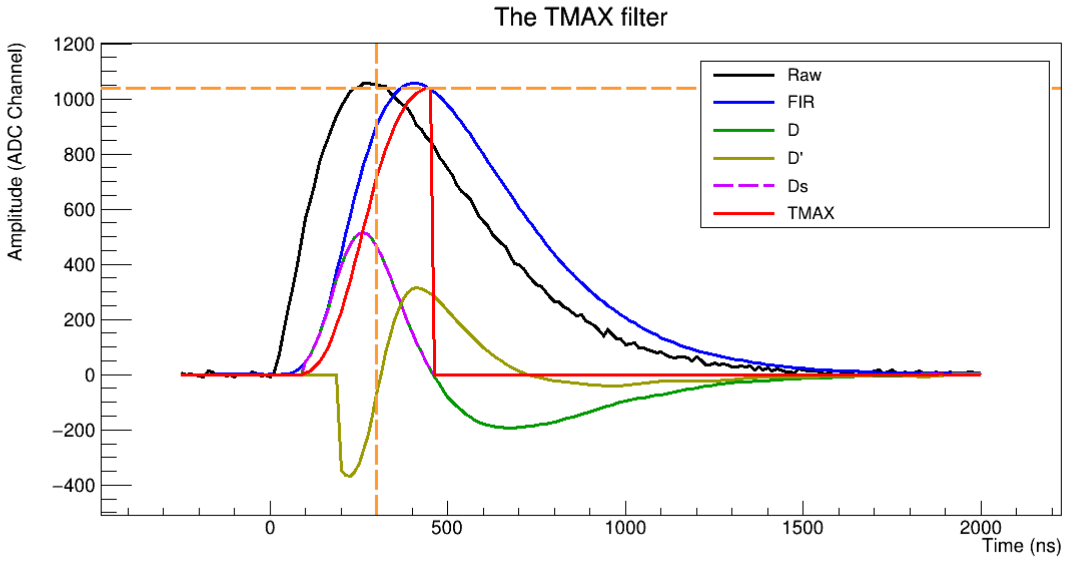


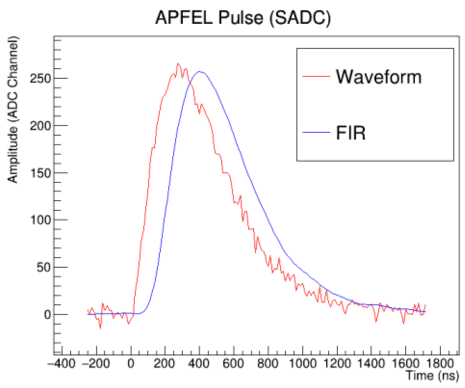
Similar noise rates and hit efficiencies for the reduced iFFT

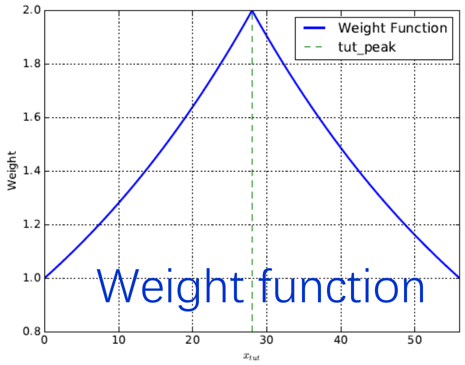
Feature Extraction



TMAX filter







FIR: 20-coefficient filtering

Derivative (D)

- D[i] = T[i + r] – T[i]

Second derivative (D')

- D'[i] = D[i + r] – D[i] → time: zero transit of D'

TMAX

- TMAX[i] = Σ_i(D[i]) – Θ[-D[i]] * D[i] → amplitude: TMAX peak
- where Θ(x) =
  - 0 (x < 0)
  - 1 (otherwise)

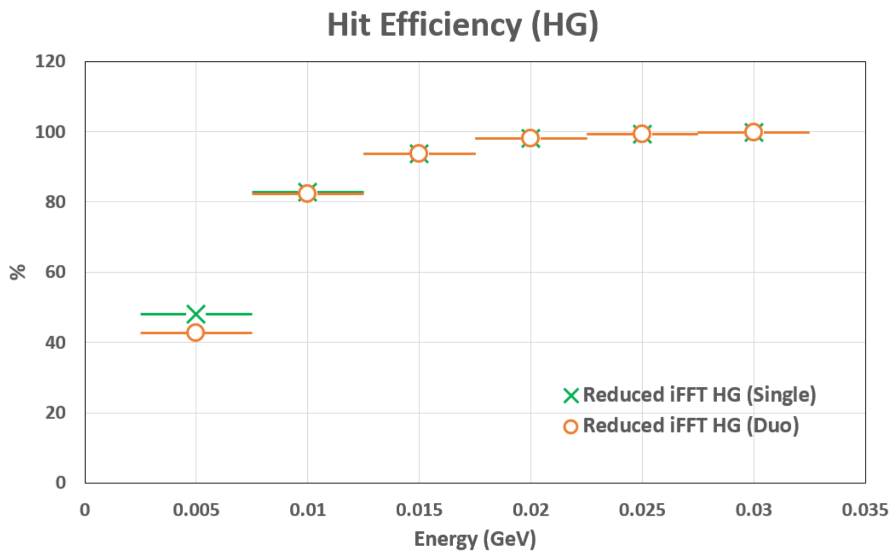
- **Hit detection**
  - Threshold: A weight function convoluted with the TMAX (1.35 MeV)

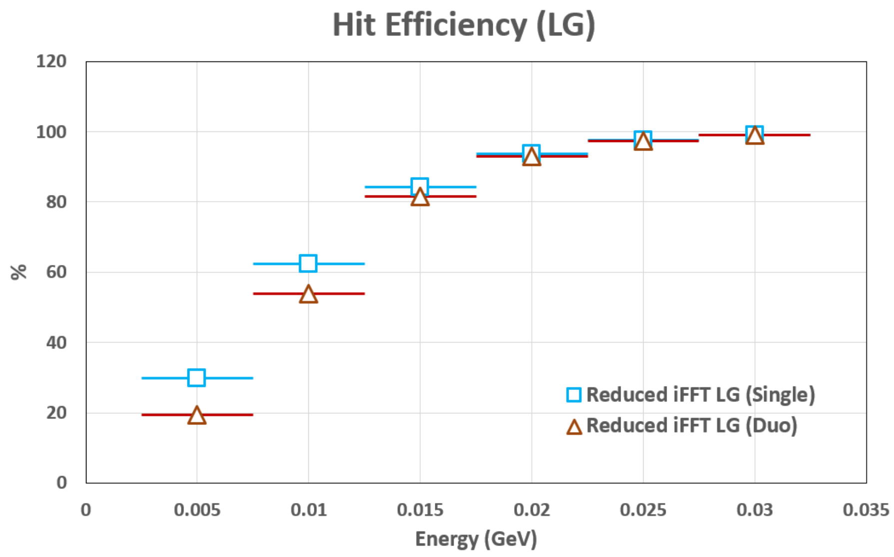
Noise rate/hit efficiency check (duo APD)

- Noise rate = # of noises / sec [Hz]
  - Single APD: 166.0 +/- 1.3 kHz (HG), 332.9 +/- 2.6 kHz (LG)
  - Duo APD: 22.1 +/- 0.2 kHz (HG), 84.4 +/- 0.7 kHz (LG)

➡ The duo-APD output can effectively suppress the noise rate

- Hit efficiency = $\frac{\# \text{ of detected hits}}{\# \text{ of generated hits}}$

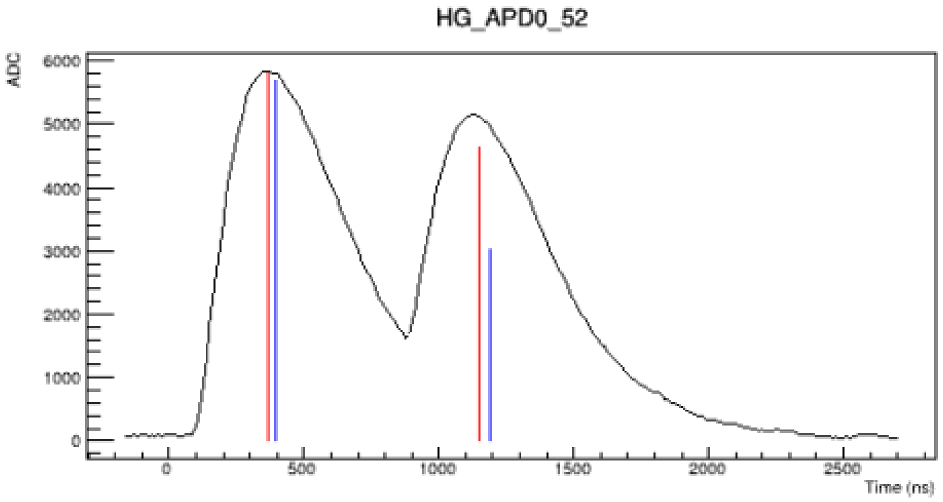


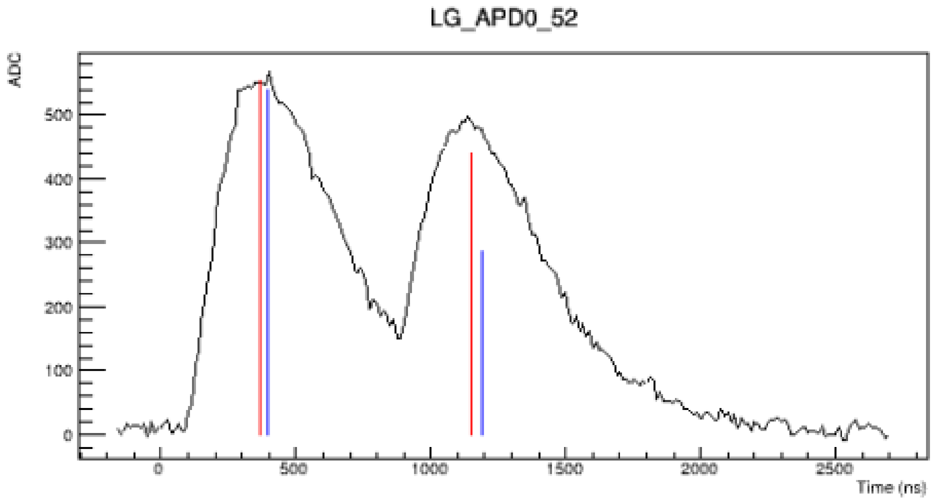


Time-based simulation

- **Signal generator**
  - Set a time window ([t_{start}, t_{active}]) to each waveform. Pile-up waveforms are generated when any time-overlapping is observed in a single crystal
- **Feature extraction**
  - The TMAX filter is capable of separating pile-up waveforms
- **Sorting digis**
  - Digis are sorted according to the time stamps
- **Reconstruction**
  - Clustering using digis in a time gap

Pile-up waveforms

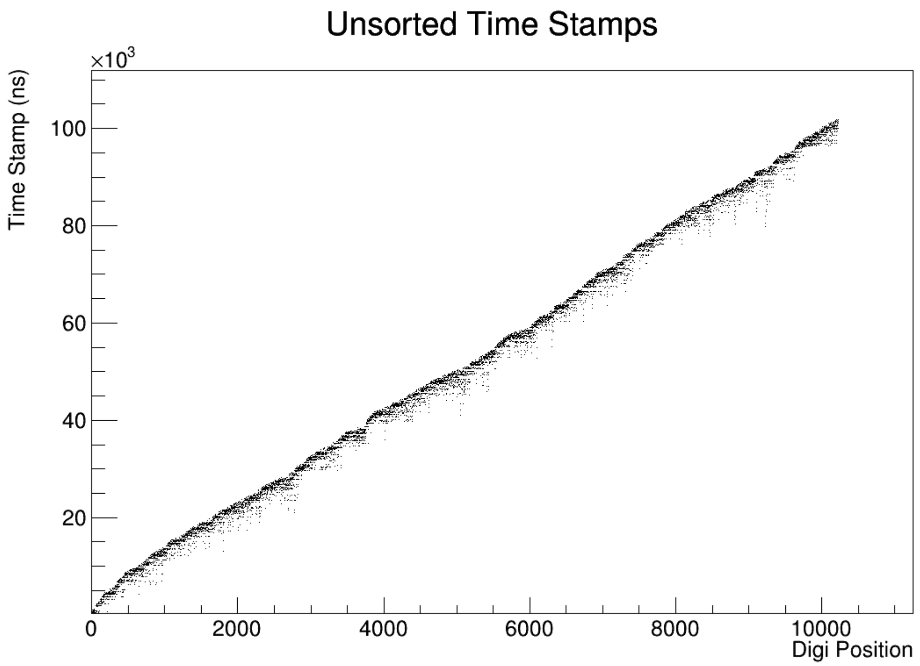


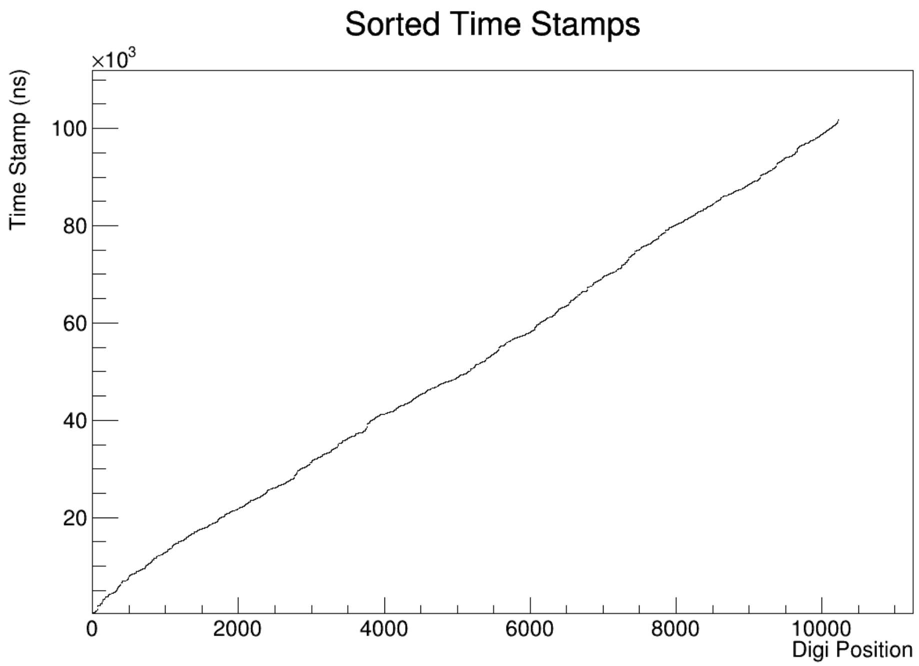


(red line: input time/amplitude, blue line: extracted time/amplitude)

- ✓ We are able to produce the pile-up waveforms, and are able to separate them
- ✓ The amplitude of the secondary waveforms need to be corrected, because the amplitude of the rising edge does not start from 0

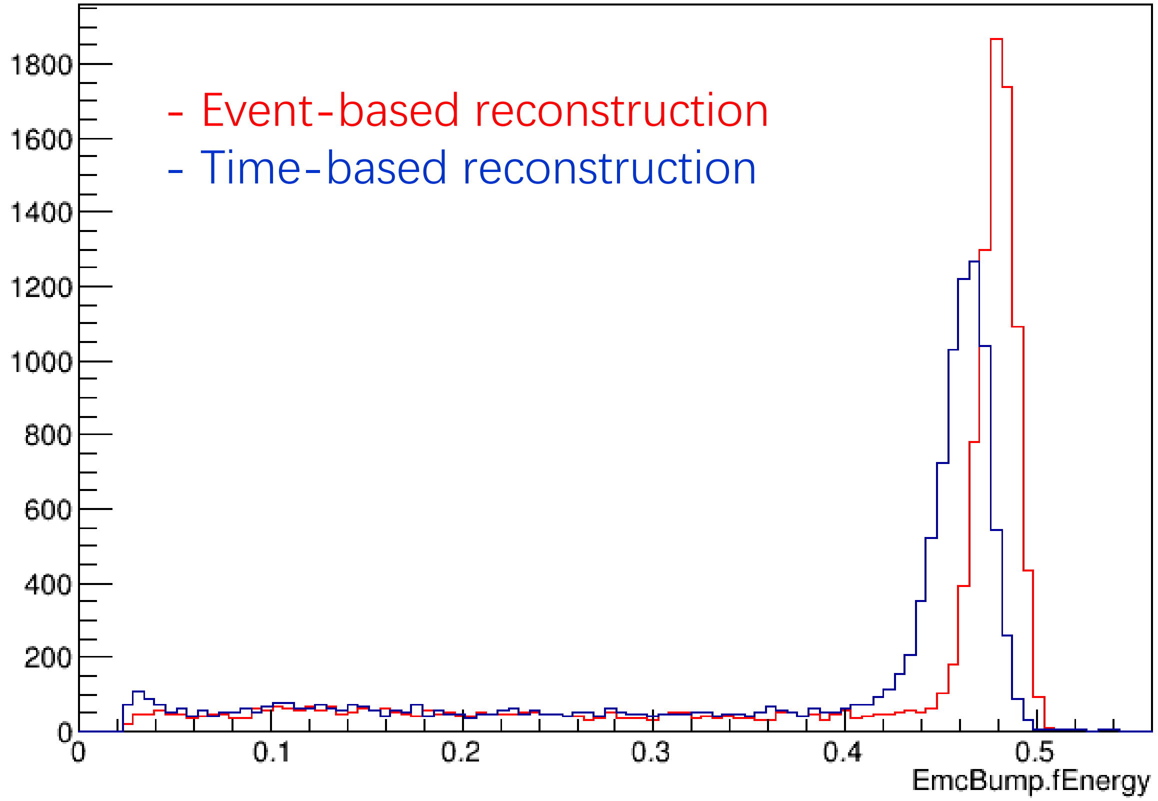
Digi sorting





✓ **Digis are sorted according to the time stamps as expected**

Reconstructed photon energy (GeV)



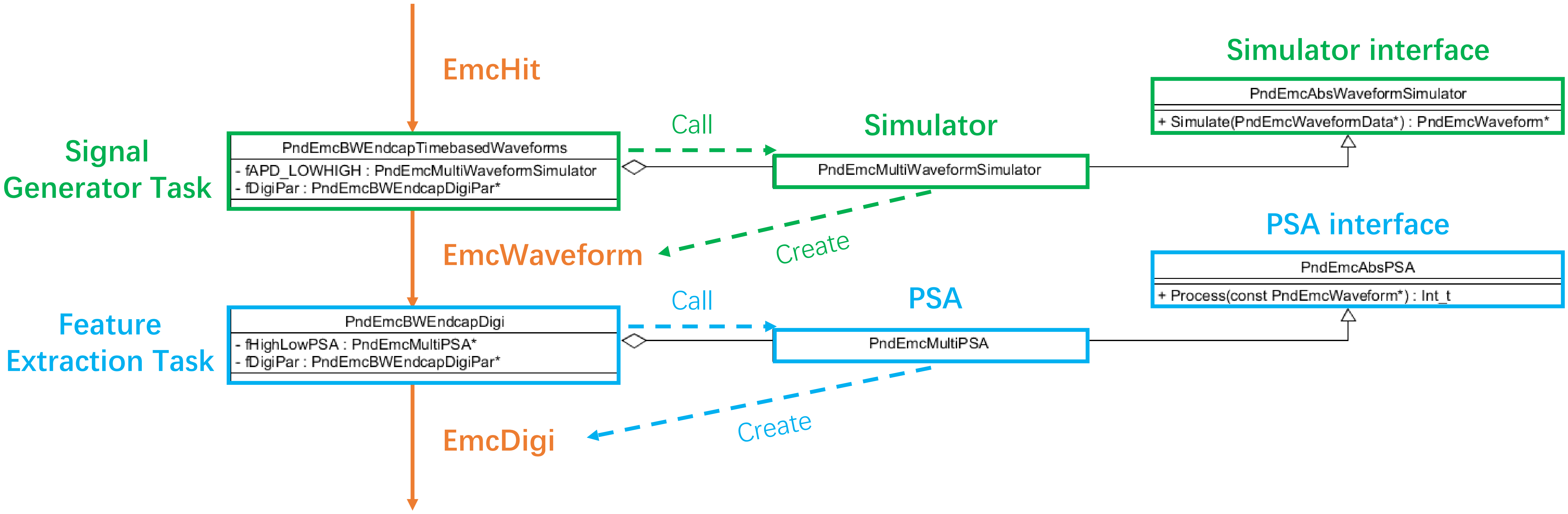
- ✓ Single photon (box generator)
  - ✓ Energy: 0.5 GeV
  - ✓ Theta: 155 deg / Phi: 0-360 deg
  - ✓ Event mean time: 500 ns
  - ✓ Time gap in clustering: 20 ns
- ✓ Time-based simulation gives larger energy resolution, and smaller energy
- ✓ Need to correct the secondary pile-up energy, and to check the time-based clustering algorithm

Code development in PandaRoot

- **A standalone package for the bwec**
  - Implementation of “signal generator”/”feature extraction” features mentioned in this talk
  - Support time-based simulation
  - Signal generator code is based on fwec code by Philipp Mahlberg

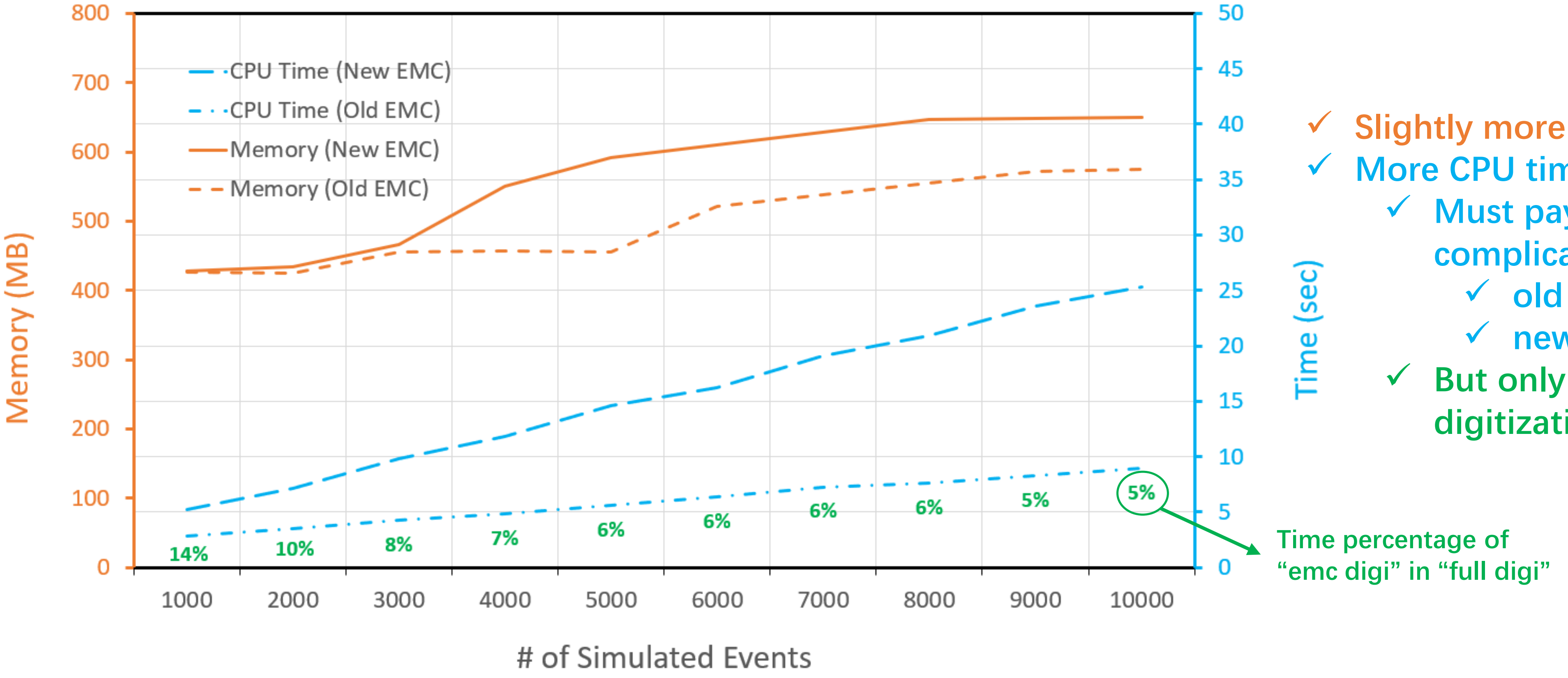
- **Design**
  - Separation of tasks and algorithms → Flexibility
  - Use common interfaces for the algorithm classes → Scalability

Code structure



- Two tasks for signal generator and feature extraction respectively
- Simulator and Pulse Shape Analyzer (PSA) as the “algorithms”
- The algorithms inherit from the “interfaces”

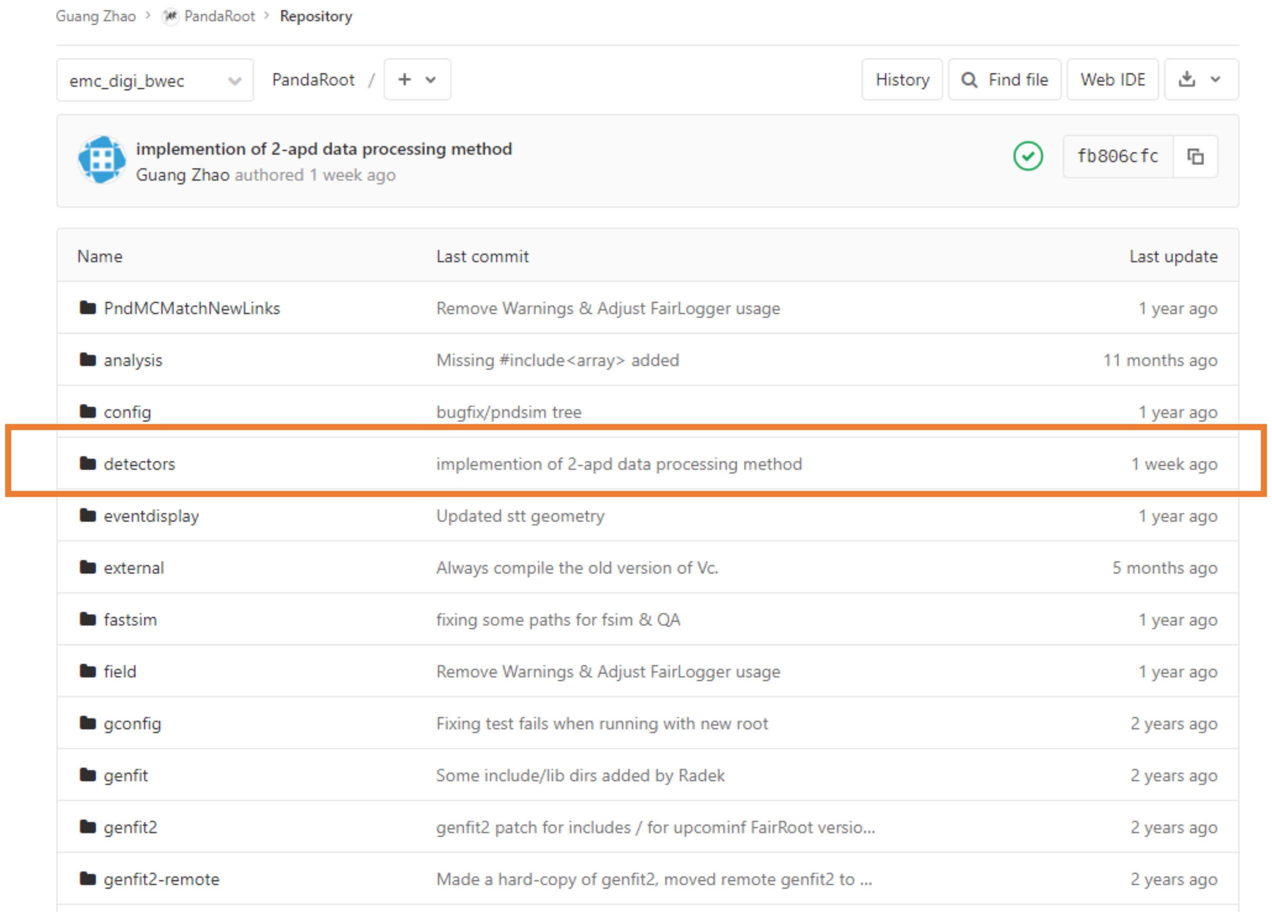
Performance test



- ✓ Slightly more memory
- ✓ More CPU time
  - ✓ Must pay for the much more complicated noise model ☹
    - ✓ old code: Gaussian noise*1/xtal
    - ✓ new code: Realistic noise*4/xtal
  - ✓ But only a small fraction in full digitization ☺

Code in git repository

https://git.panda.gsi.de/zhaog/PandaRoot/tree/emc_digi_bwec



Guang Zhao > PandaRoot > Repository

emc_digi_bwec PandaRoot / + History Find file Web IDE

implemention of 2-apd data processing method
Guang Zhao authored 1 week ago
fb806cfc

Name	Last commit	Last update
PndMCMatchNewLinks	Remove Warnings & Adjust FairLogger usage	1 year ago
analysis	Missing #include<array> added	11 months ago
config	bugfix/pndsim tree	1 year ago
detectors	implemention of 2-apd data processing method	1 week ago
eventdisplay	Updated stt geometry	1 year ago
external	Always compile the old version of Vc.	5 months ago
fastsim	fixing some paths for fsim & QA	1 year ago
field	Remove Warnings & Adjust FairLogger usage	1 year ago
gconfig	Fixing test fails when running with new root	2 years ago
genfit	Some include/lib dirs added by Radek	2 years ago
genfit2	genfit2 patch for includes / for upcominf FairRoot versio...	2 years ago
genfit2-remote	Made a hard-copy of genfit2, moved remote genfit2 to ...	2 years ago

✓ Almost ready to check in to the pandaroot main repository

Summary

- **Digitization implementation**
  - Signal generator: provide digitized waveforms with realistic noises
  - Feature extraction: extract digi information from the waveforms using the TMAX filter
  - Duo APD signal processing to suppress noises
  - Capable of the time-based simulation

- **Code development in PandaRoot**
  - A new package for the bwec digitization
  - OO design
    - Task/algorithm separation: Easy to migrate to the new framework
    - Algorithms w/ well defined interfaces: Easy to scale to other sub-detectors
  - Performance test: acceptable speed and memory use
  - To check in very soon