

EEEMCal Sim Digitization/Resolutions

Justin Frantz (Ohio U),
ePIC EEEMCal Sim Mtg
Feb 16, 2024



From [ElCrecon](#) / [src](#) / [algorithms](#) / [calorimetry](#) /CalorimeterHitDigi.cc

- `decltype(CalorimeterHitDigiConfig::capADC) EcalEndcapN_capADC = 16384; //65536, 16bit ADC`
`decltype(CalorimeterHitDigiConfig::dyRangeADC) EcalEndcapN_dyRangeADC = 20.0 * dd4hep::GeV;`
 - `decltype(CalorimeterHitDigiConfig::pedMeanADC) EcalEndcapN_pedMeanADC = 20;`
 - `decltype(CalorimeterHitDigiConfig::pedSigmaADC) EcalEndcapN_pedSigmaADC = 1;`
 - `decltype(CalorimeterHitDigiConfig::resolutionTDC) EcalEndcapN_resolutionTDC = 10 *
dd4hep::picosecond;`
 - `.capADC = EcalEndcapN_capADC,`
 - `.dyRangeADC = EcalEndcapN_dyRangeADC,`
 - `.pedMeanADC = EcalEndcapN_pedMeanADC,`
 - `.pedSigmaADC = EcalEndcapN_pedSigmaADC,`
 - `.resolutionTDC = EcalEndcapN_resolutionTDC,`
 - `.thresholdFactor = 0.0,`
 - `.thresholdValue = 4.0, // (20. GeV / 16384) * 4 ~= 5 MeV`
 - `.sampFrac = 0.998,`
 - `.readout = "EcalEndcapNHits",`
 - `.sectorField = "sector",`
- `.eRes = {0.0 * sqrt(dd4hep::GeV), 0.02, 0.0 *
dd4hep::GeV},`
`.tRes = 0.0 * dd4hep::ns,`
`.threshold = 0.0 * dd4hep::MeV, // Use ADC
cut instead`

Thank you
Dmitry, he pointed
us to these files, I did
not find them myself

From ElCrecon/src/detectors/EEMC/EEMC.cc

- `unsigned long long adc = std::llround(ped + edep *
m_cfg.corrMeanScale * (1.0 + eResRel) / m_cfg.dyRangeADC *
m_cfg.capADC);`
- `unsigned long long tdc = std::llround((time +
m_rng.gaussian<double>(0., 1.) * tRes) * stepTDC);`

Conclusion / My status

- Time resolution needs updated to something worse (?) but we don't know what yet...
- E Resolution:
 - no $A/\sqrt{E}$ term [comes from sim] ,
 - constant term [calibration?] 2% too high?
 - what is the $1/E$ term for ? From: https://wiki.jlab.org/cuawiki/index.php/OVERVIEW_OF_SPECIFICATIONS
 - [anyway probably also covered from sim?] Answer from Sasha during talk; usually should be noise term
- Leave simple single ADC digi scheme for now... [?]
 - Real digitization with real waveform (discrete ADC) will have much better performance for *energy*
 - ...but for time resolution who knows..
 - Still needing to understand how HGCROC relates..
- I am trying to run single particle ddsim / npsim to verify these codes are being run

Backup