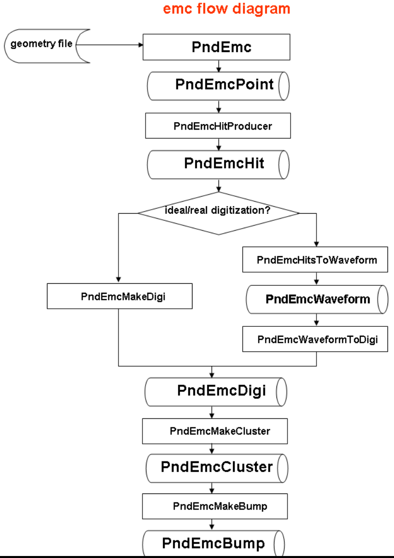
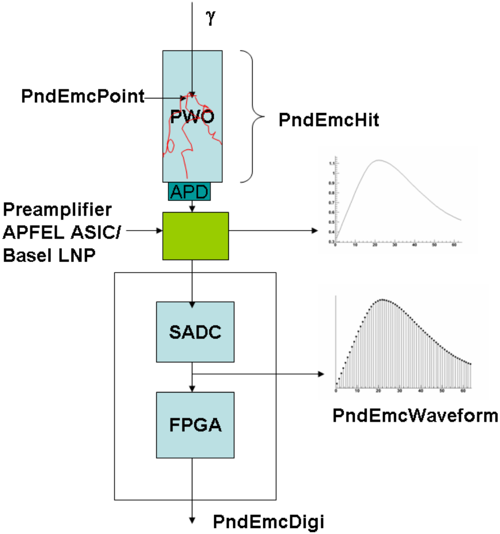
Digitization of EMC signal in pandaroot simulations

D. Melnychuk, SINS Warsaw

Panda Collaboration meeting, Stokholm, 15.06.2010

General structure of EMC simulation





Structure of data classes

PndEmcWaveform.h

```
protected:
    Int_t fTrackId;
    Int_t fDetectorId;              // detector index
    Int_t fWaveformLength;
    std::vector<Double_t>  fSignal; // Signal after FADC
```

PndEmcDigi.h

```
protected:
    Double_t fEnergy;    // digi amplitude
    Double_t fTime;      // digi time
    Int_t fTrackId;
    Int_t fDetectorId;   // detector index
    Int_t fThetaInd;
    Int_t fPhiInd;
    TVector3 fWhere;     // digi position
    Double_t fTheta;
    Double_t fPhi;
```

Two options for digitization

- Simple Digitization

  Energy of PndEmcHit is copied to digi or effective smearing is applied:

$$\frac{\sigma}{E} = \sqrt{\frac{F}{N_{p.e.} \cdot E(MeV)}} \oplus \frac{E_{noise}(MeV)}{E(MeV)} \quad (1)$$

  or for shashlyk according to prototype tests

$$\frac{\sigma}{E} = \frac{5.6\%}{E(GeV)} \oplus \frac{2.4\%}{\sqrt{E(GeV)}} \oplus 1.3\% \quad (2)$$

- Full digitization

  Pulseshapes, which correspond to output of preamplifier are produced (exponential, CR-RC, CR-2RC options) and are digitized with Nbits. Their amplitude is smeared with photon statistics and electronic noise is added to each bin. Then pulseshapes are analysed and their amplitude stored to PndEmcDigi::fEnergy (scaled by amplitude of 1 GeV signal). Two options for pulseshape analysis exist: simple parabolic fit and digital filters.

Conversion of deposited energy to digi amplitude. Relevant parameters

```
NBits:Int_t 14
DetectedPhotonsPerMeV:Double_t 500
SensitiveAreaAPD:Double_t 200
SensitiveAreaVPT:Double_t 200
QuantumEfficiencyAPD:Double_t 0.7
QuantumEfficiencyVPT:Double_t 0.22
ExcessNoiseFactorAPD:Double_t 1.38
ExcessNoiseFactorVPT:Double_t 2.2
Incoherent_elec_noise_width_GeV_APD:Double_t 1.0e-3
Incoherent_elec_noise_width_GeV_VPT:Double_t 1.0e-3
EnergyRange:Double_t 15
EnergyRangeBW:Double_t 3
FirstSamplePhase:Double_t 0
Number_of_samples_in_waveform:Int_t 64
Shaping_diff_time:Double_t 100e-9
Shaping_int_time:Double_t 500e-9
Crystal_time_constant:Double_t 12e-9
SampleRate:Double_t 80e6
Use_shaped_noise:Int_t 1
Use_photon_statistic:Int_t 1
EnergyDigiThreshold:Double_t 3.0e-3
UseDigiEffectiveSmearing:Int_t 0
NoiseAllChannels:Int_t 0
Use_nonuniformity:Int_t 0
```

- Nbits is set by default 14 bit which is rather optimistic estimate vs. 12 bits x 2 channel = 13 bit (realistic)
- Number of detected photons per MeV = 500 correspond to the measured value at $-25^{\circ}C$ 120 divided by QE =18%
- Area of APD correspond to 2 APDs $7 \times 14\ mm^2$, $2 \times 14 \times 7 = 196\ mm^2$
- VPT area corresponds to diameter 16 mm
- Electronics noise for APD (ASIC preamp): ENC=4150 e^- (rms) at $C_{det} = 270\ pF$ corresponds to 0.9 MeV.
- Electronics noise for VPT (LNP): ENC=235 e^- at $C_{VPTanode} = 22\ pF$ corresponds to 0.78 MeV (650 ns peaking time). With 200 ns peaking time - 1 MeV.

- Parabolic fit

```
Double_t pValue = signal[peakBin];
Double_t pPosition = Double_t(peakBin);
Double_t leftValue = signal[theBin-1];
Double_t rightValue = signal[theBin+1];
if (leftValue<pValue && rightValue<pValue) {
    Double_t d = 0.25*(rightValue-leftValue);
    Double_t b = pValue-0.5*(leftValue+rightValue);
    pValue += d*d/b;
    pPosition += d/b;
}
```

- Digital FIR (finite impulse response) filter
  Two possible implementation direct convolution and recursive equation. With signal x and filter kernel h convolution operation:

$$y[i] = \sum_{j=0}^{M-1} h[j]x[i-j] \tag{3}$$

For Moving Window Deconvolution (MWD) filter

$$MWD_M[i] = D_M[i] + \frac{1}{\tau} MA_M[i], \tag{4}$$

where $D_M[n] = x[n] - x[n-M]$ and $MA_M[n] = \sum_{k=n-M}^{n-1} x[k]$
Class PndEmcFadcFilter allow to set weights of kernel directly with SetData() methods or use one of predefined filters, e.g. SetupBipolarTriangle().

What is missing?

- Limited implementation of 2 APD per crystal option. Only number of photoelectrons is doubled but not summation of the signal from two APDs is implemented
- No proper timing algorithm is implemented in pandaroot
- Nuclear Counter Effect (direct hit of APD by passing particles) is not yet implemented (in progress).