Emulation and calibration of the SALT readout chip for the UT tracker for modernised LHCb detector

Adam Dendek

AGH-UST Krakow

Epiphany Conference, January 9 2015

Outline

- General motivations for the LHCb detector upgrade
- Emulation software platform overview
- Description of the DSP algorithms
- Summary

General motivation for the upgrade

- Heavy flavor physics has a great discovery potential:

  - many “theory clean” measurements

  - statistical error is dominant in many cases

  - much larger statistics is crucial for new physics searches beyond the energy scale of the LHC

- Present LHCb detector cannot operate at higher luminosity

  - limited discriminating power of L0 trigger

  - limit of 1.1 MHz for full detector readout rate vs 40 MHz beam crossing rate

Expected upgrade results

- Target luminosity $2 \cdot 10^{33}$ cm^{-2}s^{-1}.
- Plan to collect 50 fb^{-1} in 10 years
- Signal yield 10 (20) times larger for muonic (hadronic) B decays wrt 2011
- Use 40 MHz readout electronics for all subdetectors

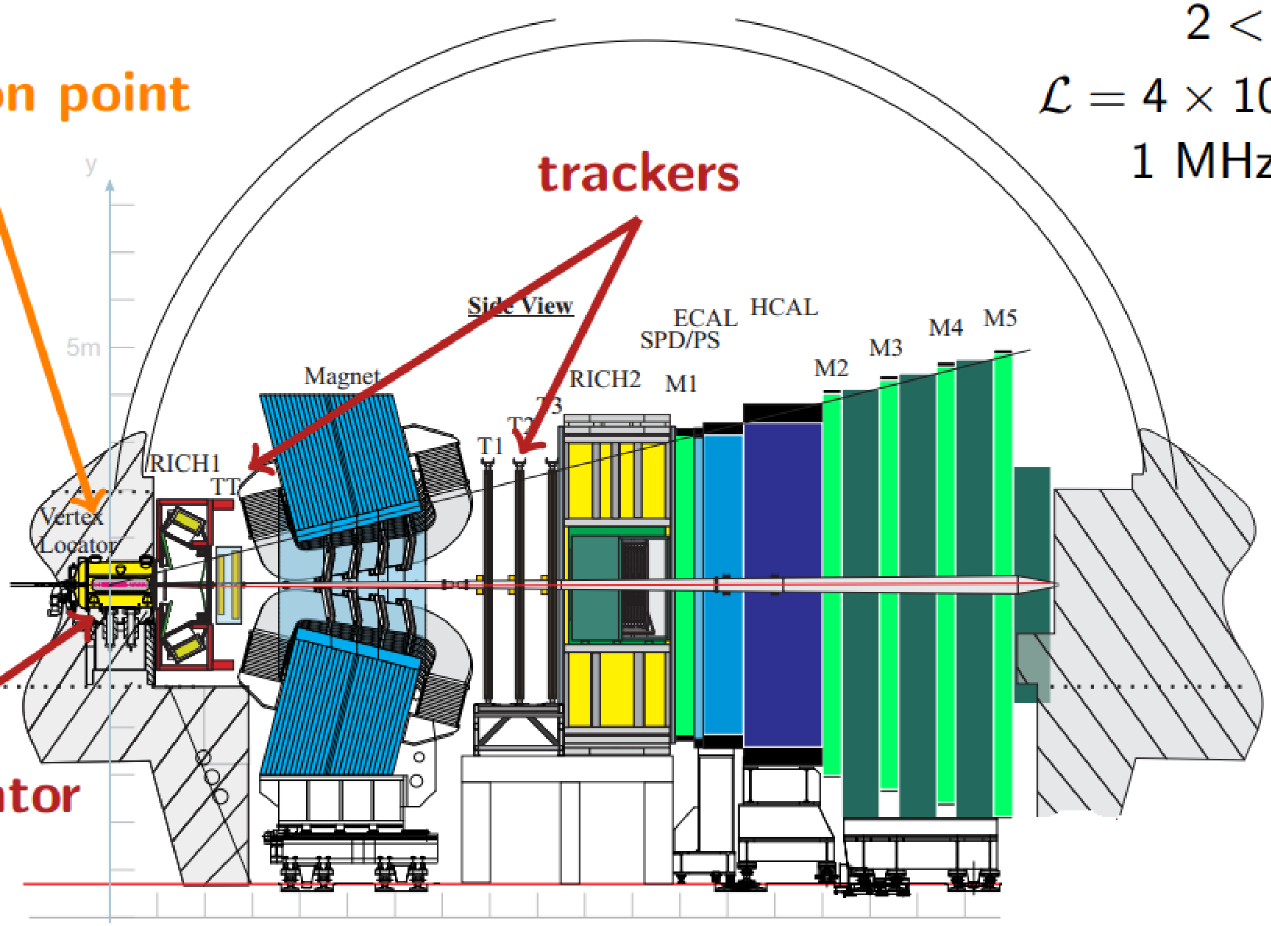
  - optimize detector design to cope with higher particle rates
- Adopt new highly flexible software trigger architecture

The current LHCb detector

$2 < \eta < 5$

$\mathcal{L} = 4 \times 10^{32}\ \mathrm{cm}^{-2}\ \mathrm{s}^{-1}$

1 MHz readout



interaction point

trackers

vertex locator

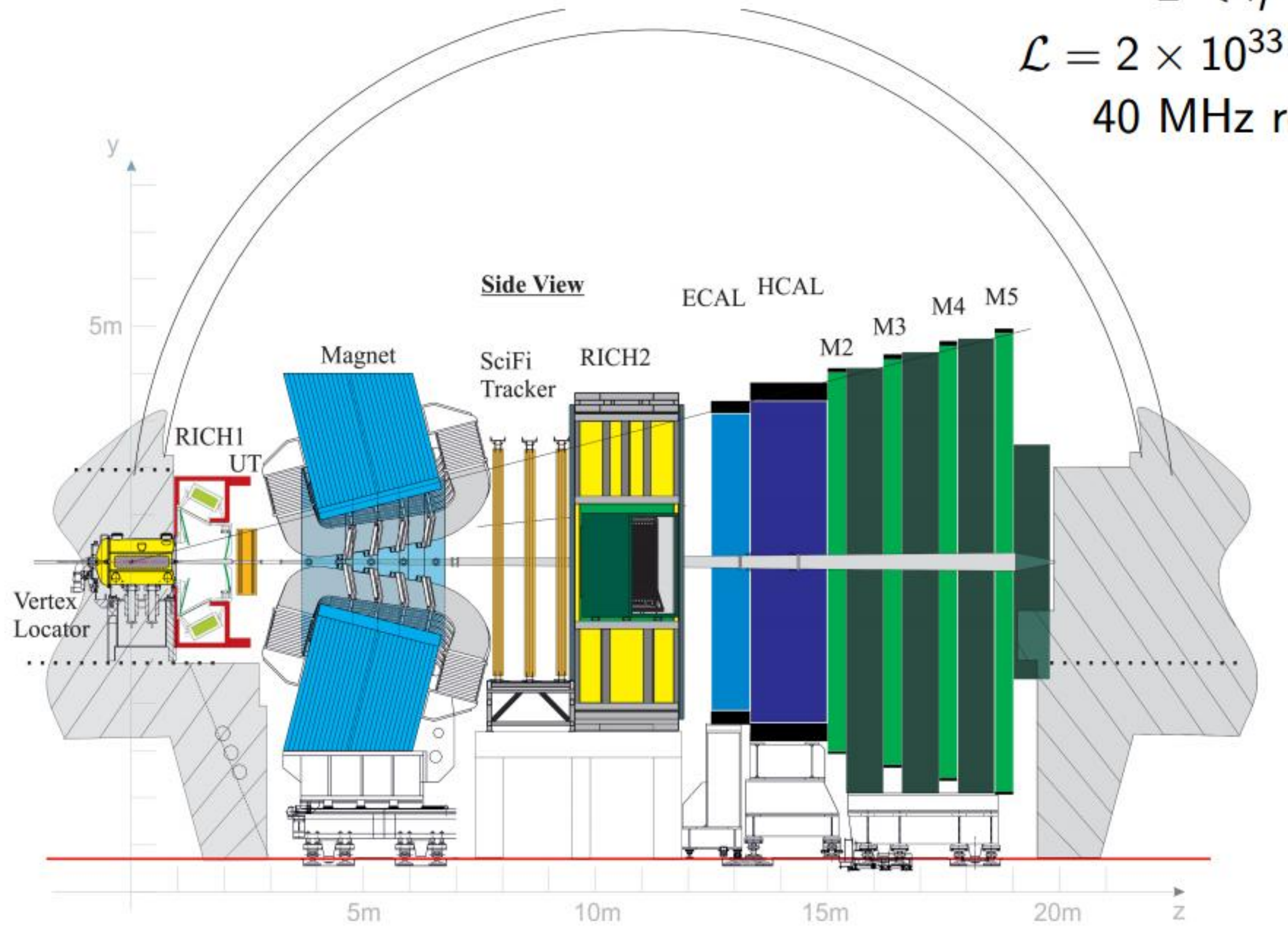
LHCb before and after the Upgrade cont.

The upgraded LHCb detector

$2 < \eta < 5$

$\mathcal{L} = 2 \times 10^{33}\ \mathrm{cm}^{-2}\ \mathrm{s}^{-1}$

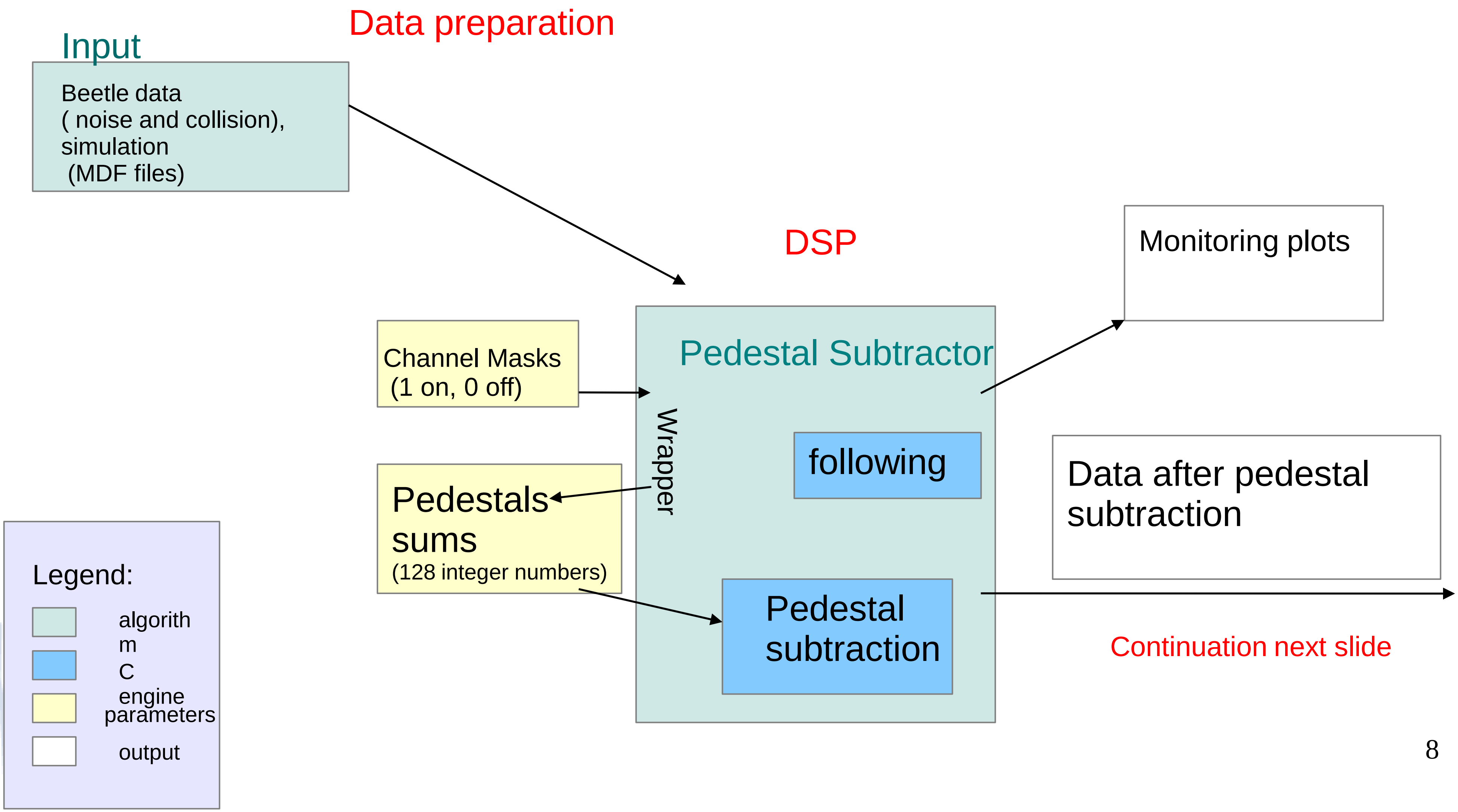
40 MHz readout



SaltLib architecture overview

- SaltLib is an architecture that allow to emulate SALT chips algorithms.
- Based on idea of KISS ("Keep it simple, stupid").
- Provide possibility of easily and complex testing designed algorithms.
- Returned lot of controlling plots.
- Will be useful when designing HDL version of algorithm and essential in detector maintenance
- Easy to add some functionality

SALTLib architecture data flow chart



Data preparation

Input

Beetle data
(noise and collision),
simulation
(MDF files)

DSP

Channel Masks
(1 on, 0 off)

Pedestals sums
(128 integer numbers)

Pedestal Subtractor

Wrapper

following

Pedestal subtraction

Monitoring plots

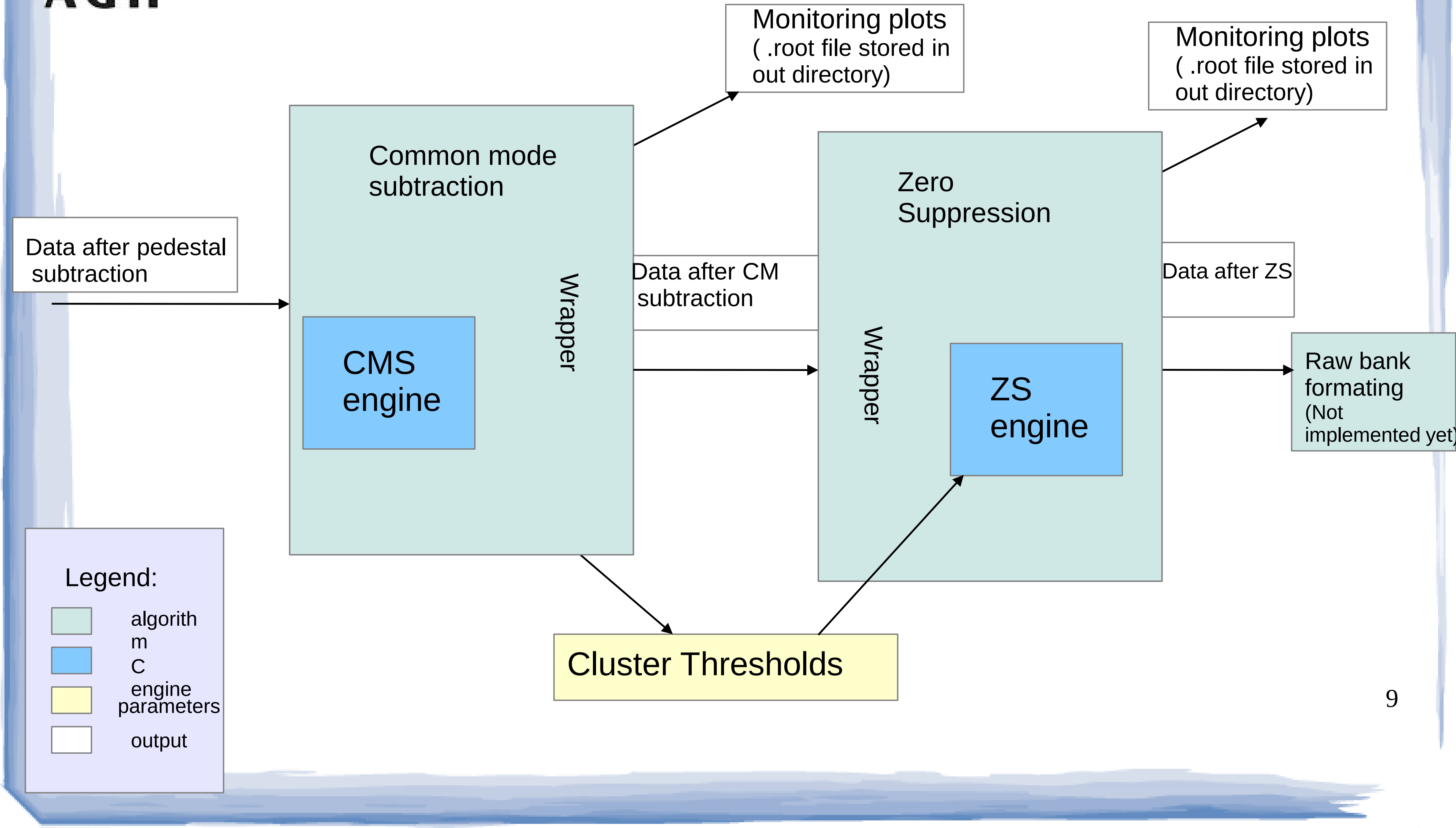
Data after pedestal subtraction

Continuation next slide

Legend:

🟩	algorithm
🟦	C engine
🟨	parameters
□	output

SALTLib architecture data flow chart cont.



Monitoring plots (.root file stored in out directory)

Monitoring plots (.root file stored in out directory)

Common mode subtraction

Wrapper

CMS engine

Data after pedestal subtraction

Data after CM subtraction

Zero Suppression

Wrapper

ZS engine

Data after ZS

Raw bank formating (Not implemented yet)

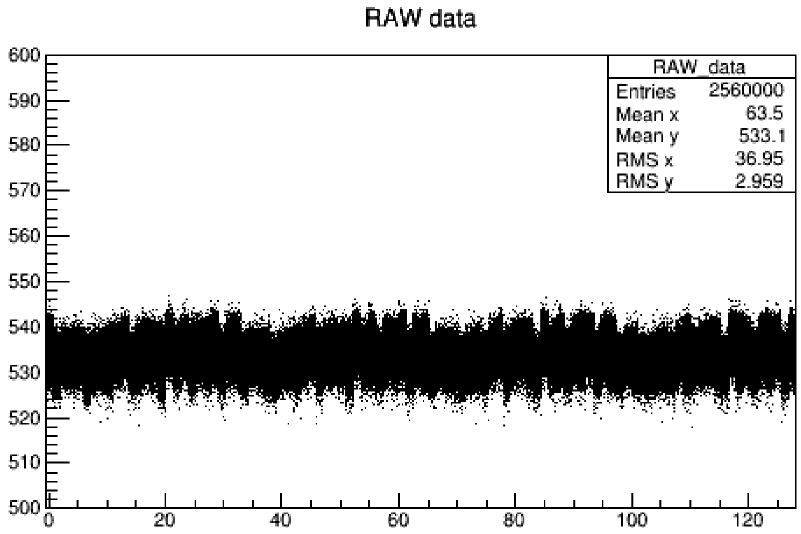
Cluster Thresholds

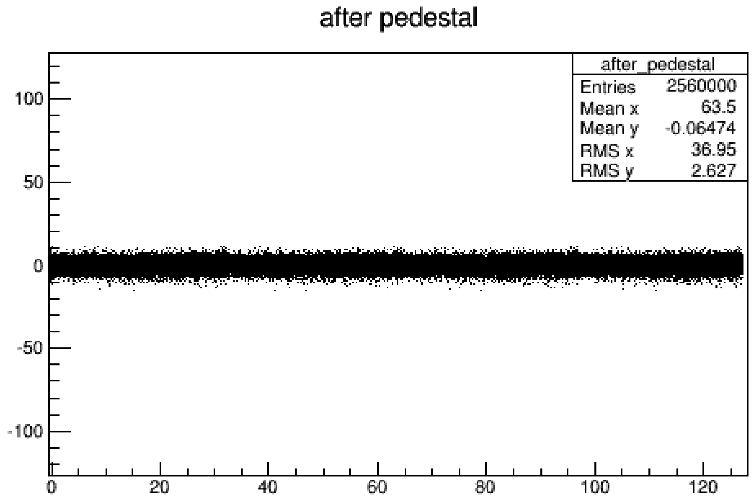
Legend:

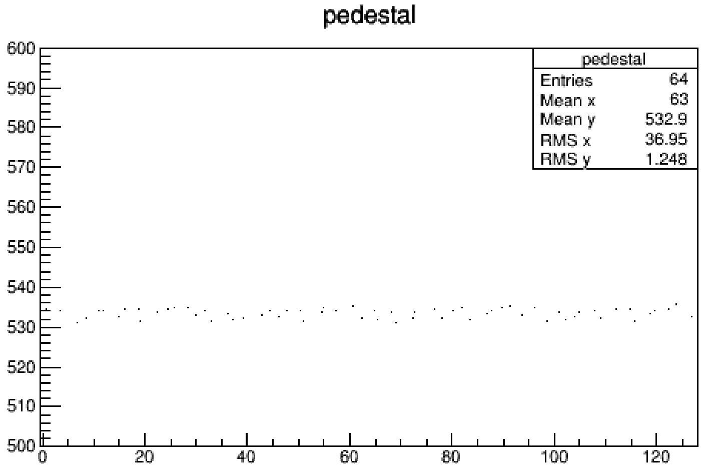
- 🟩 algorithm
- 🟦 C engine
- 🟨 parameters
- □ output

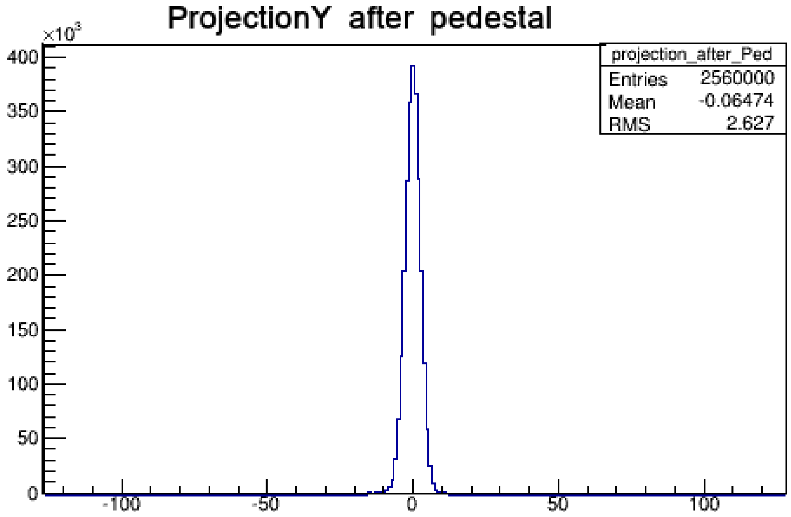
Pedestal subtractor- monitoring

It is possible to look at raw data, pedestal sums, values of pedestal and the most important data processed by the algorithm.





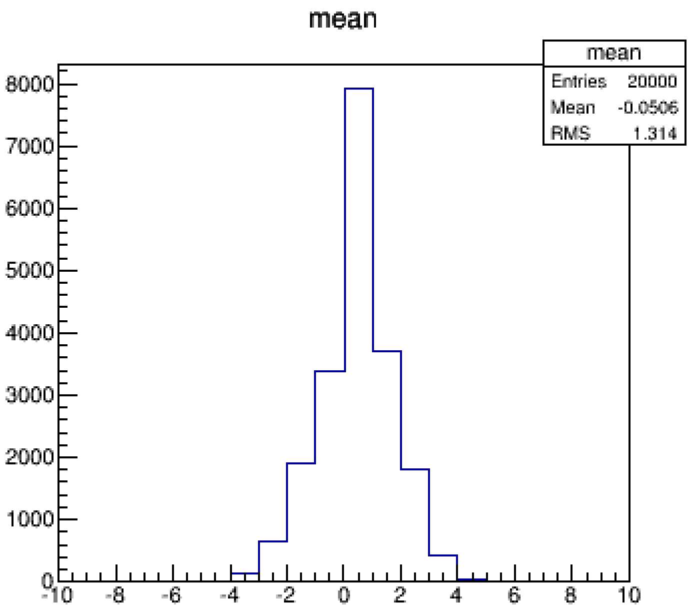




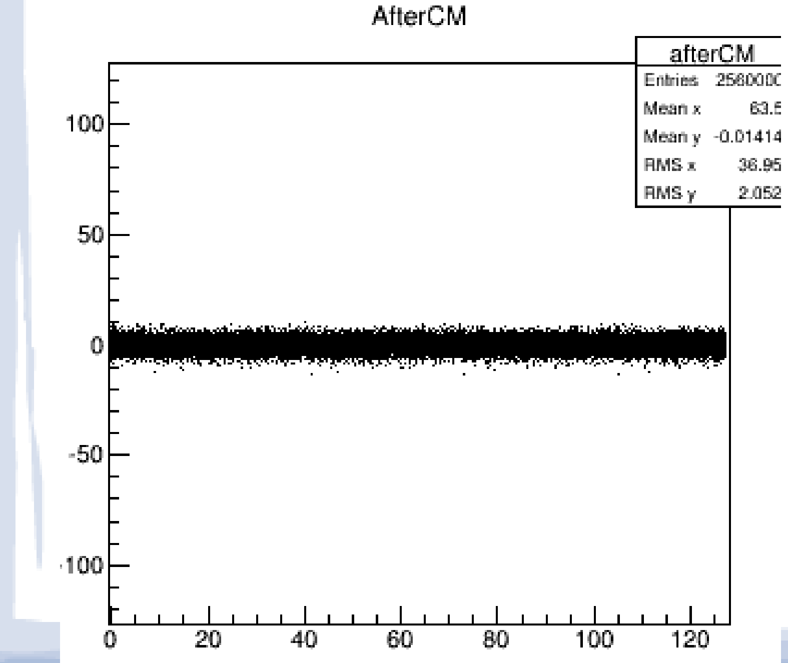
Common mode subtraction

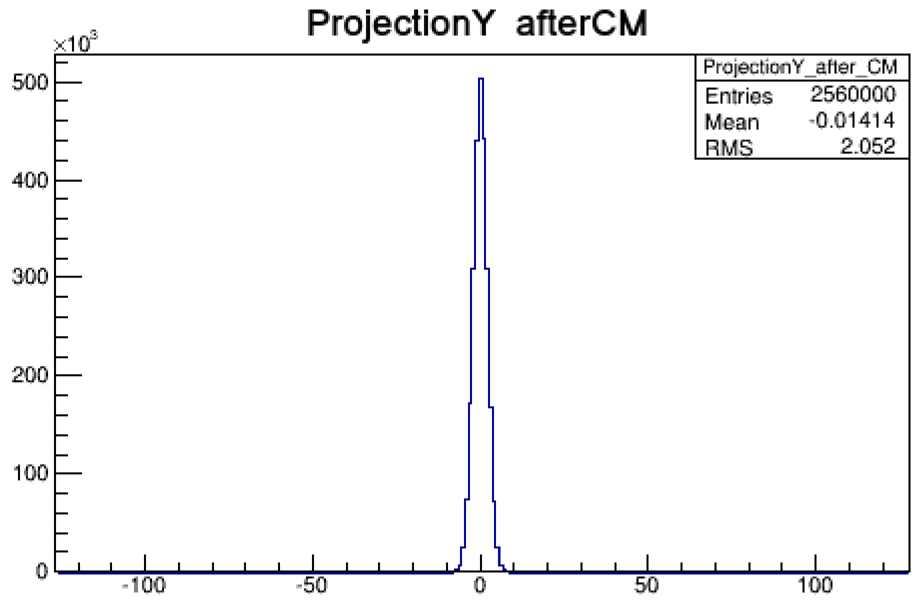
- The algorithm subtracts a constant value
- Hit rejection using a symmetric window

Distribution of noise correction value



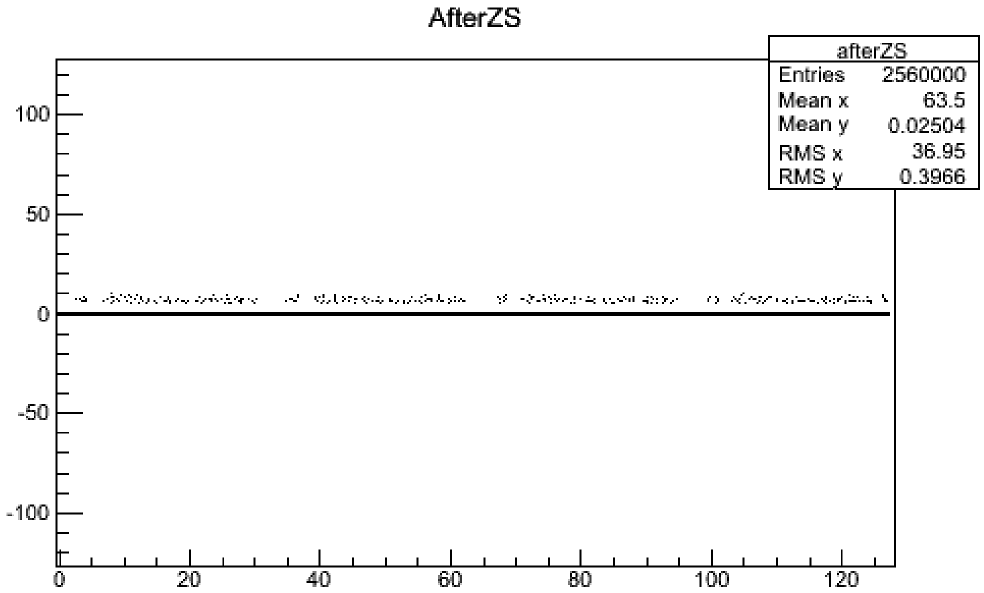
Noise after the CMS (Beetle 8 bit data)



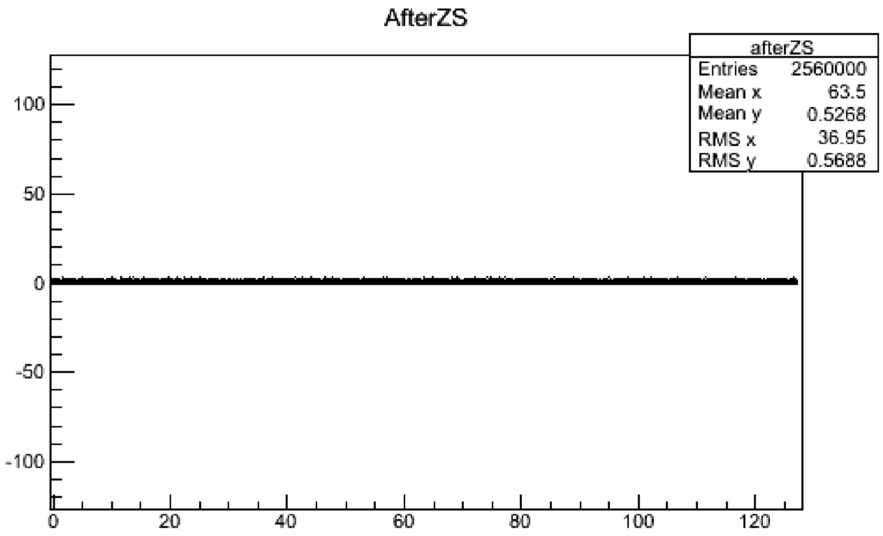


Zero Suppression

- Using loose thresholds for hit discrimination
- The threshold is tuned for each channel



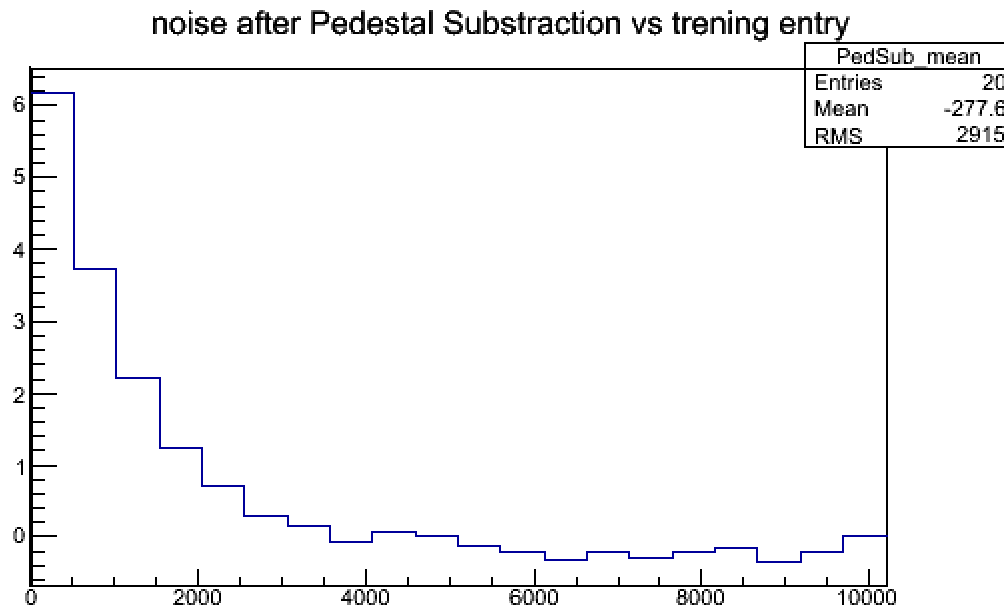
Noise after ZS (Beetle data)

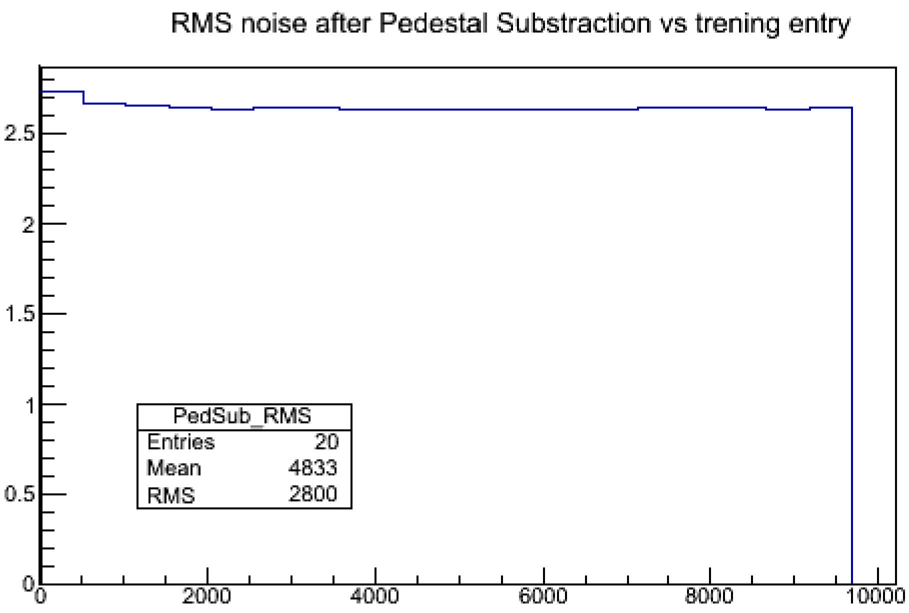


Noise after ZS (6 bit data)

Parametric scan- Pedestal

- Parametric Scan allows to optimalize value of event requited to properly calculate the pedestals sums.
- Based on this studies deduced that suffvicient value of event is 4096.





Summary

- The SaltLib give possibility of emulation data as well as tuning run parameters.
- This software is critical part in of maintenance SALT readout chip process.
- First verification of system using testbeam data has been already done!

Thank you for yours attention!